



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di laurea triennale in Ingegneria Elettronica

**STUDIO E SVILUPPO DI SISTEMI COMPLETI PER IL
MONITORAGGIO DOMICILIARE DI PAZIENTI ATTRAVERSO
L'INTERNET OF MEDICAL THINGS**

**Study and development of complete systems for home
monitoring of patients through the Internet of Medical Things**

Relatore:

Prof.ssa Paola Pierleoni

Correlatore:

Dott.ssa Sara Raggiunto

**Candidato:
Sara Bruschi**

Abstract

Lo scopo della seguente tesi consiste nello sviluppo di sistemi che consentono l'elaborazione e la memorizzazione di dati che possono essere prelevati da soggetti sottoposti a misurazioni. In particolare, sono state sfruttate delle acquisizioni effettuate su pazienti affetti da malattia di Parkinson, utili a testare il funzionamento dell'applicazione sviluppata.

L'obiettivo principale, dunque, consiste nel realizzare un'interfaccia grafica che permetta di unire due tipologie di database per conservare sia i dati sensibili di ogni paziente che i dati grezzi provenienti dalle misurazioni effettuate con sensori. I database utilizzati sono stati scelti analizzando il tipo di dato che si trovano a dover ospitare. Allo scopo di migliorare la raccolta di questi ultimi, poiché le acquisizioni sono caratterizzate da un indice temporale, è opportuno ricorrere ad un database per serie temporali. Al contrario, i dati puramente testuali possono essere memorizzati all'interno di un database relazionale, tra questi troviamo: l'anagrafica del paziente; i diari giornalieri compilati dai pazienti stessi sul proprio stato di salute (metodo tradizionale utilizzato fino ad oggi); i report relativi all'elaborazione dei dati grezzi storicizzati nel database per serie temporali.

L'applicazione, sviluppata interamente in Matlab, si pone l'obiettivo di instaurare una comunicazione tra questi due strumenti, andando quindi a costituire un mezzo fondamentale per l'utente che si trovi a dover interagire con dati della tipologia citata.

Nel caso d'uso analizzato che prevede l'analisi del movimento di malati di Parkinson, è possibile sfruttare algoritmi e strumenti di calcolo da cui si possono ottenere informazioni riguardanti la deambulazione dei soggetti sottoposti a misurazione. Si tratta di un sistema di monitoraggio a casa h24 attraverso sensori NGIMU in grado di memorizzare dati all'interno di una SD card. Questi, in seguito all'acquisizione, possono essere elaborati ed utilizzati in modo opportuno.

Il valore aggiunto dell'architettura implementata risiede nei vantaggi che essa fornisce nell'ambito della telemetria. Il dottore attraverso l'utilizzo di questo sistema risulta in grado di accedere alle informazioni su ogni singolo paziente in qualsiasi momento ne abbia bisogno e da qualsiasi luogo in cui si trovi. L'aspetto innovativo può essere identificato anche, e soprattutto, nell'utilizzo delle due tipologie di database citate in precedenza. Essi consentono una notevole riduzione dei documenti cartacei all'interno degli istituti ospedalieri. Inoltre, tutti i dati vengono memorizzati all'interno di strutture ben definite, le quali consentono un migliore e più rapido accesso alle risorse necessarie. Per questo motivo, l'implementazione e l'utilizzo dell'interfaccia grafica illustrata in precedenza rappresenta "un passo avanti" all'interno di quella che oggi viene definita telemedicina.

Indice

Introduzione	IV
Capitolo 1: Stato dell'arte della tecnologia IoT e IoMT	1
1.1 Tecnologia IoT.....	1
1.1.1 Radio Frequency Identification	1
1.1.2 Wireless Sensor Networks.....	2
1.2 Applicazioni dell'IoT nella Telemedicina	3
1.2.1 Dispositivi indossabili.....	3
Capitolo 2: Caso d'uso: la malattia di Parkinson	5
2.1 Descrizione generale della patologia	5
2.2 Gli effetti della malattia.....	6
2.2.1 Sintomi di tipo motorio	7
2.2.2 Sintomi di tipo non motorio	8
2.3 Sistemi di valutazione tradizionali.....	9
2.3.1 Scale di valutazione	9
2.3.2 Sensori IMUs.....	11
Capitolo 3: Sistemi per la telemedicina	12
3.1 Database relazionali	12
3.2 Database per serie temporali.....	13
3.2.1 OpenTSDB.....	14
3.2.2 Apache Druid	14
3.2.3 Graphite.....	15
3.2.4 InfluxDB.....	15
3.2.5 Conclusioni.....	15
3.3 Elasticsearch	16
3.3.1 Elasticsearch vs InfluxDB	18
Capitolo 4: Architettura dei sistemi utilizzati.....	19
4.1 Sensori NGIMU	19
4.2 Database.....	20
4.2.1 MySQL.....	20
4.2.1 InfluxDB	21

4.3 Matlab	23
4.3.1 Database Toolbox	23
4.3.2 App Designer	24
4.3.3 Codice GitHub per il collegamento a InfluxDB	25
Capitolo 5: Medical Data Monitoring App	26
5.1 InfluxDB	26
5.1.1 Guida all'installazione.....	26
5.1.2 Collegamento tra InfluxDB e Matlab	28
5.2 Database MySQL	30
5.2.1 Esportare il database da MySQL Workbench.....	31
5.2.2 Importare il database su MySQL Workbench.....	32
5.2.3 Creazione di un'origine dati ODBC	32
5.2.4 Collegamento tra MySQL e Matlab	33
5.3 Interfaccia grafica Matlab.....	34
5.3.1 La GUI (Graphical User Interface).....	35
5.3.2 Schermata Anagrafica.....	36
5.3.3 Schermata Report.....	37
5.3.4 Schermata Dati	39
Conclusioni	VII
Bibliografia	VIII

Introduzione

Nel mondo odierno, la stragrande maggioranza delle azioni è accompagnata dall'utilizzo di uno strumento o di un oggetto che prevede un'implementazione elettronica ed un collegamento ad Internet. Per questo motivo è importante lo sviluppo di tecnologie di questo tipo anche in ambito medico, dove può fare la differenza nel trattamento di molteplici patologie.

L'Internet of Things (IoT) entra in gioco in questo contesto e si propone di introdurre in medicina dispositivi e sensori caratterizzati da un maggiore comfort per il paziente quando sottoposto a misurazioni. In aggiunta, si hanno anche benefici per i pazienti grazie ad una notevole riduzione degli spostamenti e, dal punto di vista economico, l'utilizzo di sensori IoT permette di ridurre notevolmente i costi di gestione. Oltre a tutti i vantaggi sopra elencati, tali dispositivi "aiutano" il medico nella gestione dei pazienti, permettendogli di avere a disposizione dati più oggettivi per la diagnosi e la cura degli assistiti.

In quella che oggi prende il nome di "Telemedicina" entrano in gioco dei sensori che implementano al loro interno numerosi componenti, tra cui accelerometro, magnetometro, barometro e giroscopio. Facendo indossare questi sensori ai pazienti che si sottopongono a misurazioni ed acquisizioni, è possibile ottenere dati numerici che, dopo un'opportuna elaborazione, sono in grado di fornire indicazioni su parametri riconducibili alla qualità di movimento e all'andatura del soggetto in esame.

Come anticipato in precedenza, i parametri possono essere raccolti sottoforma di report contenenti grafici ed immagini; questi dati costituiscono uno strumento aggiuntivo e di aiuto in campo medico. I dispositivi del cosiddetto Internet of Medical Things (IoMT) forniscono quindi dati grezzi che richiedono di essere memorizzati prima ancora di poter essere elaborati. A questo scopo si possono utilizzare dei sistemi di memorizzazione, i database. Esistono due tipologie di database che si distinguono sia in base alla modalità di memorizzazione che in base alla struttura dei dati salvati al loro interno. Lavorare con dati medici implica anche trattare dati anagrafici di pazienti a cui possono essere associati report derivanti dall'analisi dei segnali acquisiti. Per questo motivo, si è scelto di utilizzare le funzionalità di un database relazionale, come MySQL, che si basa sulle relazioni che possono essere create tra le tabelle che lo compongono. Attraverso la scrittura di query in linguaggio SQL è possibile ottenere le informazioni più importanti relative ad ogni paziente.

Allo stesso modo, trattando acquisizioni di tipo numerico effettuate attraverso sensori e associate ad un indice puramente temporale, è necessario utilizzare un database per serie temporali che risponda all'esigenza di memorizzare grandi quantità di dati senza perdere l'informazione temporale

che determina l'istante in cui ogni campione è stato prelevato. Per questo tipo di applicazione può essere utilizzato InfluxDB, il quale presenta una versione sia client che cloud e che si basa su un linguaggio simil SQL per la scrittura delle query.

Per rendere possibile il collegamento, e dunque un'interazione, tra le due tipologie di database sopra citate, è opportuno sviluppare un'interfaccia grafica che funga da applicazione per il medico, così che possa estrarre ed analizzare i dati di ogni paziente. A questo scopo "entra in gioco" il software Matlab e alcuni suoi toolbox essenziali sia per la creazione dell'applicazione che per la connessione ai database. Il primo prende il nome di "App Designer" e risulta essenziale quando si ha necessità di realizzare una GUI (Graphical User Interface). Il maggior vantaggio di questo toolbox è la possibilità di programmare solo il codice che esplicita le funzioni da compiere a seguito di un'interazione utente-applicazione senza il bisogno di codificare gli elementi grafici. Il "Database Toolbox", invece, si occupa del collegamento al database MySQL e sfrutta un'origine dati ODBC.

Come caso d'uso è stata considerata la malattia di Parkinson e un sistema di acquisizione dati basato su sensori NGIMU che vengono fatti indossare al paziente. Questi sensori vengono collocati in precisi punti anatomici: piede destro, piede sinistro e vita. In questo modo, grazie ai dati acquisiti, è possibile testare il sistema sviluppato per cercare di ottenere le prestazioni migliori possibili.

Con questo lavoro è stato quindi possibile ottenere uno strumento fondamentale per integrare il mondo dell'Internet of Things con l'ambito medico. Tutto ciò, auspicabilmente, potrebbe portare ad un'importante evoluzione per lo sviluppo terapeutico e diagnostico in un mondo dove ormai tutto risulta collegato ad Internet o condiviso all'interno di un cloud.

Capitolo 1: Stato dell'arte della tecnologia IoT e IoMT

1.1 Tecnologia IoT

L'IoT (Internet of Things) è un framework di rete dinamico che ha lo scopo di fondere il dominio fisico e virtuale tramite l'utilizzo di Internet, il quale funge da mezzo per la comunicazione e la trasmissione di dati. In questo modo il mondo fisico e il mondo virtuale sono perfettamente amalgamati all'interno di un'unica rete che fa uso di protocolli di comunicazione.

L'IoT è stato individuato come un insieme di gadget informatici correlati, macchinari di tipo meccanico ed elettronico e creature ed individui aventi l'abilità di scambiare l'informazione tra sistemi senza il bisogno di passare attraverso l'interazione umano-umano o umano-macchina. L'avvento dell'Internet of Things testimonia un'enorme rivoluzione che prevede l'utilizzo di dispositivi autonomi e in grado di utilizzare una grande rete per l'interconnessione tra oggetti. Con l'uso di internet e dell'intelligenza artificiale il mondo sta diventando sempre più intelligente, minimizzando lo sforzo umano, aumentando lo scambio di dati tra macchine e sfruttando l'utilizzo di dispositivi dotati di capacità decisionale e di algoritmi intelligenti. L'IoT forma quindi un fenomeno ciclico che combina l'uso di sensori per creare una connessione tra utente e rete facendo uso di una macchina che aiuta ad analizzare, controllare e agire sui dati a seconda della situazione specifica. La crescita esponenziale del campo che comprende tutta la tecnologia dell'Internet of Things ha permesso oggi di aprire numerose opportunità ingegneristiche e scientifiche, sfruttate per la risoluzione di problemi [1].

1.1.1 Radio Frequency Identification

L'RFID, introdotto inizialmente nel 1945, è un prerequisito dell'IoT ed è un'innovazione che aiuta le macchine e i dispositivi digitali a riconoscere oggetti, registrare metadati e controllare un obiettivo attraverso l'utilizzo di onde radio [1].

Questi sistemi facilitano un'identificazione automatica tramite una combinazione di etichette e lettori; trovano applicazione nelle aree manifatturiere, in agricoltura, nei trasporti e nella sanità.

Le etichette contengono e trasmettono dati ai lettori utilizzando onde radio. I lettori raccolgono le informazioni per trasmetterle ad un server che si occupa dell'analisi e dell'elaborazione delle stesse e, dunque, lo scopo dei server è quello di identificare, monitorare, autenticare o allarmare attraverso lo scambio dei dati tra queste due entità. Per questo motivo possiamo dire che la Radio Frequency Identification facilita l'identificazione remota e automatica. Una tecnologia simile può essere trovata nei codici a barre che però richiedono di essere in vista per poter essere scannerizzati.

Le etichette RFID, al contrario, trasmettono automaticamente i dati al lettore e questo ne facilita l'utilizzo [2].

I dispositivi relativi a questa tecnologia possono essere divisi in due classi: attivi e passivi. I primi richiedono la presenza di una sorgente di alimentazione, di conseguenza sono più costosi e più ingombranti. Gli RFID passivi invece non richiedono batteria né manutenzione, dispongono di una vita operativa quasi illimitata e sono piccoli abbastanza da poter essere inseriti in luoghi in cui i dispositivi attivi non entrerebbero [3].

Le principali applicazioni della Radio Frequency Identification si possono trovare nella sanità, nelle costruzioni, negli ospedali, nell'amministrazione di parcheggi e trasporti e anche nei sistemi di controllo del traffico per monitorare la salute e le performance dei sistemi, aiutando a identificare problemi per riuscire a prevenirli [2].

1.1.2 Wireless Sensor Networks

La Wireless Sensor Networks (WSN) fa parte delle tecnologie IoT e il suo avanzamento è stato ispirato dalle applicazioni militari; comprende dispositivi costituiti di sensori per visualizzare le condizioni fisiche di soggetti in base al contesto in cui sono utilizzati [2].

Una WSN può essere descritta come una rete di nodi che cooperano tra loro e in grado di controllare l'ambiente in cui si trovano, permettendo di creare un'interazione tra uomo e computer. Questa tecnologia permette di dar vita a nuove applicazioni e nuovi mercati; in particolare le maggiori aree di applicazione sono classificate in base al tipo di informazione che viene raccolta e portata dalla rete. La specifica applicazione, infatti, determina i requisiti nella selezione del protocollo che deve essere utilizzato per la trasmissione dei dati.

La rete di nodi che costituisce una WSN può raccogliere informazioni dall'ambiente e comunicarle attraverso collegamenti wireless. I dati vengono inoltrati ad un "sink" che può coincidere con la destinazione finale o che può essere connesso ad altre reti (ad esempio Internet) tramite una porta, detta gateway. Tra le numerose applicazioni troviamo il monitoraggio ambientale, la sanità, il posizionamento e il tracciamento. Come detto in precedenza, la specifica applicazione della rete influenza fortemente la scelta della tecnologia utilizzata, la quale deve soddisfare i requisiti necessari. Le applicazioni si distinguono in base al tipo di dato raccolto all'interno della rete e possono essere classificate in due categorie: rilevamento di eventi e stima spaziale di un processo. Nel primo caso i sensori sono impostati per rilevare un evento, mentre nel secondo caso mirano a stimare un fenomeno fisico. In quest'ultimo caso il problema principale risulta ottenere la stima del

comportamento del processo spaziale sulla base di campioni prelevati da sensori che vengono posizionati in luoghi casuali [4].

1.2 Applicazioni dell' IoT nella Telemedicina

L'ambiente sanitario in questi anni si sta spostando verso una nuova era di diagnosi e di trattamenti particolarizzati. Un'aspettativa di vita più lunga e una sempre più grande quantità di anziani e disabili portano alla necessità di sviluppare strategie in grado di ridurre il costo dell'assistenza sanitaria. Questo deve riguardare anche la popolazione affetta da malattie croniche legate all'età, tra cui rientra anche la malattia di Parkinson. Le nuove tecnologie e l'Internet of Things consentono l'applicazione di dispositivi indossabili nel campo diagnostico e terapeutico. Sistemi che siano realizzati specificamente per la malattia e per il paziente, basati sull'intelligenza artificiale, possono diventare mezzi di supporto per la squadra di medici, la quale si trova ad affrontare la diagnosi e a decidere i trattamenti.

In aggiunta a questo, la valutazione del paziente tramite l'utilizzo della telemedicina può aiutare a ridurre ampiamente i costi di logistica, aumentando anche la tranquillità del paziente. Il monitoraggio e la terapia svolte in casa, infatti, possono consentire al malato di avvertire la presenza dei sintomi in modo più naturale, così da poter ricevere prontamente i medicinali necessari. Nonostante questo, comunque, la diagnosi rimane impegnativa e richiede la competenza di neurologi specializzati [5].

L'approccio dell' IoT nel campo delle malattie che si manifestano in problemi di andatura prevede per lo più l'utilizzo di dispositivi intelligenti che possano connettersi tra loro per scambiare dati ed effettuare un'analisi degli stessi. In questo ambito l'intelligenza computazionale è fondamentale perché viene utilizzata per sfruttare la potenza dei dispositivi impiegati nella misurazione e per permettere un'elaborazione efficace dei dati raccolti [6].

1.2.1 Dispositivi indossabili

Nell'ultimo decennio si è assistito ad un enorme sviluppo dei sensori di movimento miniaturizzati, i quali risultano in grado di raggiungere performance elevate con un minimo consumo di potenza. La ricerca sui sistemi micro-elettromeccanici ha raggiunto il punto in cui i sensori sono talmente leggeri da poter risultare impercettibili quando indossati. Esistono tre principali modalità di applicazione di questi sistemi nel campo medico: sensori specifici bioelettrici, tra cui si identificano sensori per l'elettrocardiografia, l'elettromiografia, l'elettroencefalografia e la foto pletismografia; sensori di

movimento come, ad esempio, accelerometro e giroscopio; sensori ambientali come videocamere o sensori di temperatura e pressione [5].

Ovviamente nell'ambito del monitoraggio di pazienti l'interesse si focalizza sui sensori indossabili, utilizzati allo scopo di effettuare misurazioni in grado di fornire informazioni sull'andatura e sull'equilibrio del soggetto sottoposto alla raccolta dei dati.

Per monitorare la malattia utilizzando un metodo oggettivo e garantendo la possibilità di rimanere in un ambiente familiare al paziente, vengono applicati dei sensori indossabili aventi un costo limitato ma molto efficienti. Oltre a raccogliere e memorizzare dati, tali sensori sono anche in grado di effettuare un minimo di elaborazione su di essi, così da fornire in uscita delle misurazioni che consentono di esplorare nuove modalità di trattamento. Dunque, in un panorama sanitario in cui il paziente deve essere visitato in una clinica specifica per poter ottenere una valutazione sulla sua instabilità posturale e sui livelli di tremore dovuti alla malattia di cui risulta affetto, gli elevati costi che questa operazione prevede possono essere ridotti in modo considerevole attraverso una procedura automatica e impiegabile direttamente in casa. Quest'ultima prevede un controllo del paziente "da remoto", permesso dall'utilizzo di dispositivi IoT intelligenti [6].

Capitolo 2: Caso d'uso: la malattia di Parkinson

2.1 Descrizione generale della patologia

Il Parkinson è una malattia idiopatica del sistema nervoso caratterizzata sia da manifestazioni di tipo motorio che di tipo non motorio, rappresentati in Figura 2.1. Si tratta del secondo più comune disturbo cronico neurodegenerativo progressivo che compare per lo più in soggetti di età avanzata. Fu riconosciuta per la prima volta nei primi anni dell'Ottocento e si capì immediatamente che si trattava di una malattia non frequente in persone sotto i 40 anni di età. Per questo motivo, in tutto il mondo si stima che tra i 7 e i 10 milioni di persone siano affette da Parkinson. La sua definizione patologica si identifica nella perdita o nella degenerazione di neuroni produttori di dopamina, provocando così una marcata compromissione del controllo motorio. Le cause del suo sviluppo sono ancora sconosciute, ma sono stati rilevati numerosi fattori di rischio (mostrati in Figura 2.2), tra cui età, fattori genetici ed esposizione a pesticidi [7].

I sintomi di cui sono affetti i malati di Parkinson sono principalmente tremori a riposo, bradicinesia, mancanza di stabilità e rigidità. A questi si aggiungono sintomi di tipo non motorio che risultano spesso essere tanto debilitanti quanto quelli sopra elencati; tra essi i più comuni si identificano in depressione, ansia, insonnia e problemi gastro intestinali come nausea e dolore addominale [8].

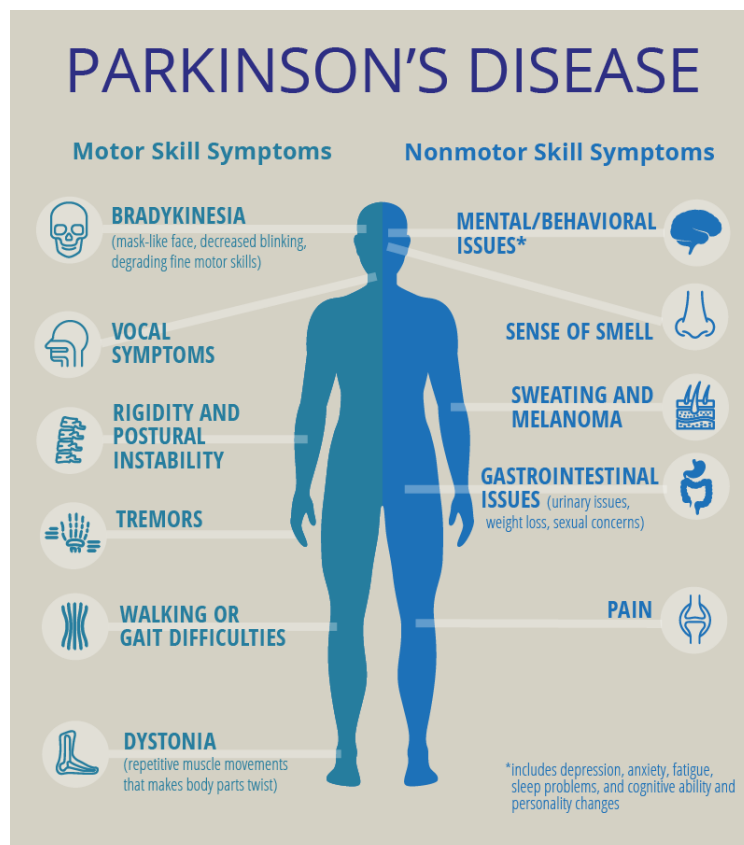


Figura 2.1 Sintomi motori e non della malattia di Parkinson [9].

I disturbi di tipo cognitivo e motorio, interagendo tra loro, risultano avere un grandissimo impatto negativo sulla qualità di vita del paziente affetto da malattia di Parkinson. Alcuni pazienti possono riscontrare difficoltà di deambulazione e perdita di equilibrio, altri possono presentare gravi anomalie posturali senza compromissione nell'andatura. In ogni caso alleviare i disturbi posturali e di andatura rimane un compito molto difficile. Uno dei più grandi rischi per i malati di Parkinson si identifica nel costante pericolo di cadere, il quale risulta essere molto pericoloso nelle condizioni di una persona che già si trova a dover combattere con i numerosi sintomi descritti in precedenza [10]. Attualmente non esiste alcuna cura per il Parkinson, ma sono disponibili diverse terapie che possano aiutare i pazienti a tenere sotto controllo i sintomi più aggressivi. Tra queste: medicine, interventi e terapie di supporto sia fisico che mentale possono risultare d'aiuto. In ogni caso, qualunque tipo di terapia ha come obiettivo quello di alleviare i sintomi e di prevenire l'avanzamento della malattia [11].

2.2 Gli effetti della malattia

In condizioni fisiche come quelle date dal Parkinson i malati subiscono un deterioramento di tipo motorio e cognitivo [12] che può portare ad un sempre più rapido peggioramento nell'abilità di compiere azioni quotidiane.

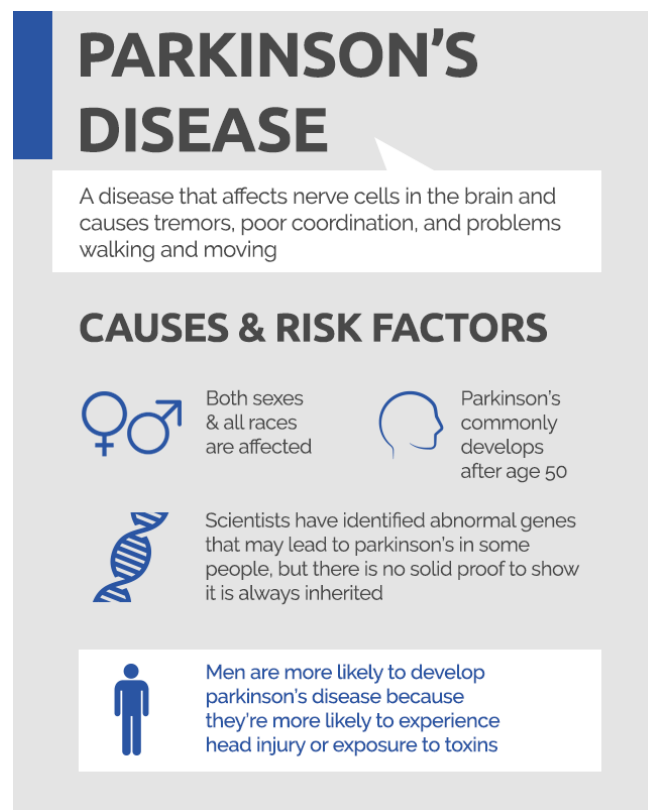


Figura 2.2 Fattori di rischio per la malattia di Parkinson [13].

Il Parkinson è una malattia caratterizzata da cambiamenti di tipo neurologico. Nel cervello di un malato, infatti, avviene la formazione di proteine anormali chiamate corpi di Lewy. È questo il motivo per cui è stato mappato in sei stadi neuropatologici differenti: i primi due sono definiti stadi pre-sintomatici; con l'avanzamento della malattia si raggiungono gli stadi 3 e 4 dove i pazienti iniziano a sviluppare i primi sintomi; negli ultimi due stadi, 5 e 6, si raggiunge infine un'ampia varietà di manifestazioni cliniche. Ad oggi non esiste alcuna cura in grado di fermare la malattia, ma esistono numerose terapie che coinvolgono medicine dopaminergiche in grado di correggere i disturbi motori. La Levodopa è la più utilizzata, ma l'aspetto negativo risiede nel fatto che dopo un lungo periodo di somministrazione tende a non avere più un buon effetto sul paziente in cura [14].

2.2.1 Sintomi di tipo motorio

Viene stimato che circa l'80% delle cellule di dopamina vengano perdute prima che i problemi motori inizino a comparire come sintomi del Parkinson. Inizialmente questi ultimi si manifestano su un solo lato del corpo, comparendo in modo uniforme solo dopo qualche anno. I sintomi di natura motoria comprendono bradicinesia, rigidità muscolare, tremori e instabilità posturale; insieme ad essi possono manifestarsi blocchi, congelamento dei movimenti, problemi nel parlare o nel deglutire e deformità posturali [14].

Il tremore è tipicamente più evidente quando i muscoli sono rilassati e risulta più aggressivo sulle mani e sulle braccia rispetto che sulle gambe, dove invece risulta meno frequente. La bradicinesia è definita come lentezza del movimento, la quale rende progressivamente più difficile compiere attività di tipo quotidiano senza problemi. La rigidità, invece, causa una resistenza passiva ai movimenti degli arti rendendo difficile ai malati di Parkinson riuscire a camminare con un passo non strascicato. L'instabilità è dovuta a problemi con l'equilibrio, questi debilitano i movimenti più semplici come, ad esempio, alzarsi dal letto o da una sedia, aumentando così il rischio di cadute pericolose [15].

I sintomi fino a qui elencati, insieme ad altre condizioni di disagio, contribuiscono ad aumentare il pericolo a cui sono sottoposti i malati, mettendo a rischio la loro incolumità. Per questo motivo è fondamentale riuscire a monitorare la situazione dei pazienti con Parkinson, in modo da poter assicurare loro una buona terapia. Un monitoraggio, inoltre, risulta necessario a valutare la progressione e la gravità della malattia. Per prevedere il tasso di declino della capacità motoria nel Parkinson sono stati introdotti diversi metodi di studio altamente variabili, dipendenti dalla compromissione motoria e dal deterioramento cognitivo [16].

Tra questi oltre all'utilizzo di scale di valutazione e questionari compilabili dal paziente, possono essere impiegati anche dei sensori in grado di rilevare le caratteristiche motorie del soggetto. Entrambi i metodi verranno analizzati più approfonditamente in un prossimo paragrafo.

2.2.2 Sintomi di tipo non motorio

I sintomi di natura non prettamente motoria tendono a comparire addirittura prima della diagnosi della malattia, e possono coincidere con problemi di diverso tipo. Tra questi possono manifestarsi apatia, sonnolenza diurna, problemi del sonno e costipazione; ad essi si aggiungono l'incapacità di godere di situazioni piacevoli, perdita di memoria, perdita di gusto e olfatto, eccessiva sudorazione, fatica e dolore. Ansia e depressione, poi, possono comparire anche molti anni prima della diagnosi di Parkinson. I sintomi pre-motori variano da paziente a paziente, ma continuano a presentarsi e a peggiorare durante il corso della malattia. Con il suo progredire, infatti, i sintomi non motori possono diventare anche più fastidiosi dei sintomi motori [14].

Ansia e depressione si presentano in circa la metà dei malati affetti da malattia di Parkinson, essi risultano tristi, pessimisti e ansiosi dopo la diagnosi ricevuta poiché si preoccupano di un futuro incerto. Insieme a queste reazioni emotive spesso si manifestano anche specifici cambiamenti nel cervello del soggetto, i quali vanno ad aumentare il rischio di finire in depressione. Quando questi sintomi si presentano, influenzano in modo ampiamente negativo la qualità di vita del malato. Per questo motivo si ha la necessità di intervenire con terapie che spesso non coincidono con quelle utilizzate per trattare il Parkinson. Nonostante quest'ultimo venga classificato come disturbo del movimento, i cambiamenti emotivi possono avere un effetto più negativo rispetto a quelli motori, poiché risultano avere un maggiore impatto sulla vita di tutti i giorni del malato di Parkinson. L'apatia viene descritta come una mancanza di emozione e di interesse in situazioni che prima venivano invece considerate interessanti dal soggetto in questione. È un sintomo direttamente collegato ai bassi livelli di dopamina nel cervello che, oltre a causare alterazioni nei movimenti, controlla anche aree responsabili del piacere e della motivazione. Oltre a queste tipologie di sintomi, si è visto che circa un terzo dei malati di Parkinson può sviluppare anche un certo livello di demenza quando la malattia risulta essere ad uno stadio avanzato; insieme ad essa, si manifestano allucinazioni, confusione e delirio [15].

2.3 Sistemi di valutazione tradizionali

2.3.1 Scale di valutazione

Il numero sempre più crescente di diagnosi della malattia di Parkinson evidenzia l'importanza nel poterne misurare gli esiti clinici. Per far questo sono state realizzate scale e questionari in grado di valutare il danno motorio e la disabilità nei pazienti affetti da Parkinson. Informazioni su validità, affidabilità e reattività sono considerate essenziali per assicurare un'utile applicazione di queste tecniche [17].

Problemi di postura, andatura ed equilibrio causano una grande invalidità che può manifestarsi attraverso cadute, immobilità e perdita di indipendenza, andando così a minare la qualità di vita di persone affette da Parkinson. Di conseguenza, un'adeguata e puntuale individuazione dei disturbi nel movimento è importante per ridurre il rischio di gravi ripercussioni da essi derivanti. Per valutare i sintomi che minacciano andatura ed equilibrio nel Parkinson vengono utilizzati test e scale di valutazione, i quali prevedono l'assegnazione di punti da parte del paziente e del medico a seconda della risposta del primo a diversi stimoli motori. Questo comporta però che i risultati sono di difficile

Activities-Specific Balance Confidence (ABC) Scale

Adapted from: Powell LE and Myers AM. The Activities-Specific Balance Confidence (ABC) Scale. *J Gerontol. A. Biol. Sci. Med. Sci.* 1995; 50A:M 28-34

Level of Confidence (0-100%)

1. Walking around the house _____
2. Walking up and down stairs _____
3. Picking up slippers/ something from the floor _____
4. Reaching at your eye level _____
5. Reaching while on your tiptoes _____
6. Reaching while standing on a chair _____
7. Sweeping the floor _____
8. Walking outside to a nearby car _____
9. Getting in/out of a car/transport _____
10. Walking across a parking lot _____
11. Walking up and down a ramp _____
12. Walking in a crowded mall _____
13. Being bumped while walking in a crowd _____
14. Using an escalator while holding the railing _____
15. Using an escalator without holding the railing _____
16. Walking on slippery floors _____

Total Score: _____

MDS UPDRS Score Sheet

Patient Name or Subject ID		Site ID	Assessment Date (mm-dd-yyyy)	Investigator's Initials
1.A	Source of information	<input type="checkbox"/> Patient <input type="checkbox"/> Caregiver <input type="checkbox"/> Patient + Caregiver	3.3b Rigidity-- RUE 3.3c Rigidity-- LUE 3.3d Rigidity-- RLE	
Part I				
1.1	Cognitive impairment		3.3e Rigidity-- LLE	
1.2	Hallucinations and psychosis		3.4a Finger tapping-- Right hand	
1.3	Depressed mood		3.4b Finger tapping-- Left hand	
1.4	Anxious mood		3.5a Hand movements-- Right hand	
1.5	Apathy		3.5b Hand movements-- Left hand	
1.6	Features of DDS		3.6a Pronation- supination movements-- Right hand 3.6b Pronation- supination movements-- Left hand	
1.6a	Who is filling out questionnaire	<input type="checkbox"/> Patient <input type="checkbox"/> Caregiver <input type="checkbox"/> Patient + Caregiver	3.7a Toe tapping-- Right foot 3.7b Toe tapping-- Left foot	
1.7	Sleep problems		3.8a Leg agility-- Right leg 3.8b Leg agility-- Left leg	
1.8	Daytime sleepiness		3.9 Arising from chair	
1.9	Pain and other sensations		3.10 Gait	
1.10	Urinary problems		3.11 Freezing of gait	
1.11	Constipation problems		3.12 Postural stability	
1.12	Light headedness on standing		3.13 Posture	
1.13	Fatigue		3.14 Global spontaneity of movement	
Part II				
2.1	Speech		3.15a Postural tremor-- Right hand 3.15b Postural tremor-- Left hand	
2.2	Saliva and drooling		3.16a Kinetic tremor-- Right hand 3.16b Kinetic tremor-- Left hand	
2.3	Chewing and swallowing		3.17a Rest tremor amplitude-- RUE 3.17b Rest tremor amplitude-- LUE	
2.4	Eating tasks		3.17c Rest tremor amplitude-- RLE 3.17d Rest tremor amplitude-- LLE	
2.5	Dressing		3.17e Rest tremor amplitude-- Lip/jaw	
2.6	Hygiene		3.18 Constancy of rest tremor	
2.7	Handwriting		Were dyskinesias present? <input type="checkbox"/> No <input type="checkbox"/> Yes	
2.8	Doing hobbies and other activities		Did these movements interfere with ratings? <input type="checkbox"/> No <input type="checkbox"/> Yes	
2.9	Turning in bed		Hoehn and Yahr Stage	
2.10	Tremor			
2.11	Getting out of bed			
2.12	Walking and balance			
2.13	Freezing			
3a	Is the patient on medication?	<input type="checkbox"/> No <input type="checkbox"/> Yes		
3b	Patient's clinical state	<input type="checkbox"/> Off <input type="checkbox"/> On		
3c	Is the patient on levodopa?	<input type="checkbox"/> No <input type="checkbox"/> Yes		
3.C1	If yes, minutes since last dose:		4.1 Time spent with dyskinesias 4.2 Functional impact of dyskinesias 4.3 Time spent in the OFF state	
Part III				
3.1	Speech		4.4 Functional impact of fluctuations	
3.2	Facial expression		4.5 Complexity of motor fluctuations	
3.3a	Rigidity-- Neck		4.6 Painful OFF-state dystonia	

Figura 2.3 Scala MDS-UPDRS (a destra) [18] e scala ABC (a sinistra) [19] per la valutazione dei disturbi motori di pazienti affetti da Parkinson

interpretazione a causa della natura soggettiva di tali tecniche. Sotto questo punto di vista, anche i test che prevedono la compilazione di questionari non aiutano a fornire strumenti aggiuntivi nella valutazione del rischio a cui i pazienti si trovano sottoposti [20].

La UPDRS (Unified Parkinson's Disease Rating Scale) è la scala di valutazione clinica più utilizzata per il morbo di Parkinson. Originariamente, insieme ad un'enorme quantità di punti di forza, era caratterizzata anche da un gran numero di punti deboli e dunque necessitava dell'inclusione di ulteriori aree di interesse, così da permettere uno sviluppo scientifico adeguato. Per risolvere queste problematiche venne proposta una versione avanzata, la MDS-UPDRS (visibile in Figura 2.3 a destra), caratterizzata da quattro parti, le quali si focalizzano su diverse caratteristiche della malattia. Tra queste, la prima parte riguarda le esperienze non motorie durante azioni di vita quotidiana, la seconda riguarda invece esperienze di tipo motorio nella vita di ogni giorno, la terza viene utilizzata per l'analisi motoria e la quarta analizza invece le complicazioni nel movimento [18].

Un'altra scala particolarmente utilizzata su pazienti affetti da Parkinson è la ABC (Activities-specific Balance Confidence Scale), la quale consiste in un questionario strutturato in modo da valutare la confidenza dell'individuo nell'eseguire attività di tutti i giorni senza perdere l'equilibrio [21], come si può vedere in Figura 2.3 a sinistra.

Per il controllo dell'equilibrio e della postura, invece, viene comunemente utilizzato il TUG (Timed Up and Go Test) che misura il tempo necessario al paziente per alzarsi da una sedia, camminare e tornare al posto. Insieme a questo si utilizzano anche la BBA (Brunel Balance Assessment), la BBS (Berg Balance Scale) e il FRT (Functional Reach Test) utili a valutare l'avanzamento della malattia sui soggetti sottoposti ad analisi [21].

Chiaramente, come già anticipato, le scale di valutazione e i questionari non consentono di ottenere risultati oggettivi e corrispondenti alla realtà perché si basano su un sistema di punteggi assegnati sia dal dottore di riferimento che dal paziente sottoposto a valutazione, il tutto in una situazione non confortevole per il soggetto su cui vengono eseguiti gli esami. In aggiunta, ogni scala consente di ottenere informazioni relative ad uno specifico aspetto della malattia, dunque per avere un resoconto più generale sarebbe necessario impiegare più di una tecnica contemporaneamente. Risulta quindi indispensabile disporre di strumenti in grado di rilevare in modo più oggettivo l'intensità dei sintomi, riuscendo anche a segnalare in tempo reale la presenza di rischi, facendo sì che possano essere evitate situazioni di pericolo per il malato.

2.3.2 Sensori IMUs

La scienza sta cercando di realizzare dei dispositivi automatici in grado di aiutare nella diagnosi di Parkinson e di valutare il peggioramento nell'andatura del paziente, con l'aiuto di algoritmi applicabili nel caso in cui risulti necessario notificare il personale medico [22].

L'esame posturografico consiste in misurazione e interpretazione sistematica di grandezze che caratterizzano l'oscillazione nella postura quando ci si trova in posizione eretta. In campo medico viene utilizzato per valutare il rischio di caduta in soggetti anziani o affetti da malattie che compromettono il sistema motorio. Questo tipo di esame utilizza una piastra pesante, difficile da trasportare e molto costosa. Per questo motivo, negli ultimi anni, per effettuare tali misurazioni sono stati realizzati dei dispositivi indossabili basati su sensori IMUs (Inertial Measurement Units) miniaturizzati. Il numero di sensori che devono essere applicati al soggetto sottoposto a misurazione varia a seconda dell'applicazione; ogni sensore include accelerometro, giroscopio e magnetometro. Il primo ci permette di misurare l'accelerazione di ogni movimento, il secondo misura la velocità angolare specifica, mentre il magnetometro misura sia ampiezza che direzione del campo magnetico locale. Anche se questi sensori non sono ancora considerati uno standard, il loro utilizzo è ideale per la misurazione dell'equilibrio, grazie soprattutto al costo contenuto che li caratterizza e alla facilità nel trasportarli in posti diversi. Le tecnologie basate sui dispositivi indossabili nel campo del Parkinson, oltre ad essere un importante strumento di appoggio che aiuta i dottori nella diagnosi, rende possibile il controllo del rischio di caduta e la rilevazione di situazioni di pericolo per i pazienti che li indossano [23].

Capitolo 3: Sistemi per la telemedicina

In campo medico si ha a che fare con una grandissima mole di dati, la quale ha necessità di essere archiviata e conservata per poter essere eventualmente elaborata, al fine di ottenere report o grafici utili alla comprensione di quell'insieme di informazioni. A questo scopo si hanno a disposizione i database che possono distinguersi in due tipologie: database relazionali e database per serie temporali. Entrambi presentano la medesima struttura dato che si basano su una raccolta di informazioni organizzata sotto forma di tabelle, formate da righe e colonne, all'interno delle quali è possibile memorizzare i dati forniti in ingresso al database.

Per quanto riguarda il modo in cui è possibile comunicare con tali strutture dati, in entrambi i casi è possibile sfruttare linguaggi basati sulla scrittura di query che permette di interagire con i database e con i dati presenti al loro interno, permettendo all'utilizzatore di accedere ad essi sia in lettura che in scrittura.

La differenza principale che sussiste tra le due forme di database presentate è il fatto che nel caso dei relazionali risulta implementata la possibilità di utilizzare una chiave primaria che non viene creata automaticamente sulla base dai dati inseriti. Questa infatti, se definita, permette all'utente decidere quale dato costituirà proprio il dato di riferimento della tabella. Al contrario, nei database per serie temporali la chiave primaria coincide con il timestamp di raccolta del dato, ovvero l'istante temporale in cui è avvenuta quella precisa misurazione.

È questo il motivo principale che porta l'utente a dover effettuare una scelta, a seconda delle proprie necessità, tra uno o l'altro tipo di database.

3.1 Database relazionali

Un database relazionale, anche detto RDB (Relational Database), è un contenitore di dati, tra i quali esistono relazioni ben definite. Le informazioni vengono memorizzate all'interno di strutture organizzate secondo un sistema di righe e colonne che prendono il nome di tabelle.

Le tabelle di un RDB possono includere un identificatore univoco per ogni riga della tabella chiamato chiave primaria, la quale consente di indicizzare i dati. Quest'ultima viene utilizzata per condividere gli stessi valori tra tabelle diverse all'interno dello stesso database, consentendo così di creare relazioni tra i dati conservati al suo interno. Per interrogare i dati nei database relazionali viene utilizzato il linguaggio SQL (Structured Query Language) con cui è possibile scrivere query per creare, leggere, modificare ed eliminare sia dati che tabelle [24].

Nei primi anni di vita dei database ogni applicazione archiviava dati in una struttura unica quindi recuperarli era un'operazione difficile che non consentiva alle applicazioni di ottenere buone prestazioni. Il modello relazionale venne progettato per risolvere il problema di strutture dati arbitrarie, fornendo una modalità standard per rappresentare i dati, i quali ora risultano facilmente accessibili da qualsiasi applicazione che esegua query su di essi. Un altro punto di forza dei RDB rispetto ai database del passato è proprio il linguaggio SQL utilizzato, il quale adesso risulta basato sull'algebra relazionale che semplifica le operazioni e permette un miglioramento delle performance di tutte le query eseguite su un database. Il semplice ma potente modello relazionale viene utilizzato da organizzazioni di ogni tipo per un'ampia varietà di esigenze e di informazioni; è questo che fa degli RDB i database più utilizzati sin dagli anni '70 [25].

3.2 Database per serie temporali

Al giorno d'oggi, a seguito di un'enorme evoluzione della tecnologia che ha portato ad una elevata produzione di sensori e dispositivi connessi nell'IoT (Internet of Things), e a causa dell'altissima velocità con cui si ha necessità di memorizzare i dati da essi provenienti per poterne ottenere informazioni utili, è grande la rilevanza che hanno ottenuto le serie di dati temporali nell'analisi e nel monitoraggio dei sistemi. Per time series data si intendono i dati forniti sottoforma di serie temporali, in cui il tempo è un asse di riferimento primario e fondamentale. Per questo motivo possiamo dire che questo tipo di dato rappresenta in modo complessivo l'evoluzione di un sistema o di un processo nel tempo [26].

Per immagazzinare ed elaborare i time series data è necessario disporre di un database che possa avere caratteristiche specifiche e che sia in grado di gestire in modo ottimale il fattore tempo. Quest'ultimo risulta di grande importanza, ad esempio, nel caso di misurazioni effettuate attraverso l'uso di sensori, in modo che possano essere discriminate a seconda dell'istante temporale in cui ogni campione è stato prelevato.

In queste applicazioni entrano in gioco i Time Series Database (TSDB) ottimizzati per poter gestire in modo adeguato l'evoluzione temporale dei dati inseriti al suo interno. Infatti, a differenza dei database relazionali, sono in grado di comprimere ed archiviare le informazioni relative a data e ora, di gestire il ciclo di vita dei dati e di effettuare il riepilogo degli stessi. Per far questo, i TSDB dispongono di una fondamentale proprietà, ovvero quella di presentare un unico campo chiave costituito dall'informazione temporale (detta timestamp) che esplicita l'istante in cui quel dato è stato prelevato. Rispetto ad un tradizionale database quelli per serie temporali si distinguono per

due importanti caratteristiche: la scalabilità, molto importante quando si ha a che fare con una grandissima mole di dati che deve essere memorizzata in tempi brevissimi, e l'usabilità, per la quale entrano in gioco una serie di funzioni ed operazioni utili all'analisi dei dati in seguito al caricamento degli stessi all'interno del database di riferimento [26].

È possibile dire quindi che la scelta di un database per serie temporali rispetto ad un database relazionale dipende dalla necessità di focalizzare l'attenzione sul fattore temporale dei dati che si vogliono analizzare, facendo in modo che si possa ottenere una visione d'insieme dell'andamento generale.

È possibile analizzare i database temporali offerti sul mercato, effettuando un confronto dal punto di vista di efficienza, servizi offerti, possibilità di elaborazione dei dati ed altri fattori che potrebbero risultare determinanti nel caso fosse necessario scegliere quello adatto ad una specifica applicazione.

3.2.1 OpenTSDB

OpenTSDB è un database per serie temporali distribuito e scalabile in grado di raggiungere una velocità di scrittura dei dati di qualche milione al secondo. Queste alte prestazioni sono dovute al layer di base che sfrutta un'architettura distribuita, la quale è in grado di garantire altissime velocità. Rispetto ad altri database dello stesso tipo OpenTSDB è maggiormente indirizzato alla comunità open source grazie all'uso della LGPL (Lesser General Public License), ovvero una licenza di software libero. Sebbene la caratteristica che potrebbe portare a preferire questo database sia la notevole velocità di scrittura dei dati, il suo difetto è sicuramente la mancanza della capacità di esecuzione di query per l'analisi dei dati registrati e conservati [27].

3.2.2 APACHE DRUID

Apache Druid è, anch'esso, un database open source per l'analisi di dati real time che è stato sviluppato in modo da offrire un immagazzinamento dei dati di tipo SQL e OLAP (Online Analytical Processing) che consentono il querying dei dati caratterizzati da un timestamp. Questo comporta la possibilità di interagire, in lettura o in scrittura, con i dati memorizzati all'interno di Apache Druid. Questo tipo di database viene scelto quando le necessità sono l'elaborazione dei dati in real time e un alto uptime (il tempo durante il quale il server è attivo e raggiungibile sulla rete) [27].

3.2.3 GRAPHITE

Graphite è un TSDB open source che archivia serie storiche e permette di visualizzarle su richiesta, attraverso la propria interfaccia web, ad una velocità discreta. Viene utilizzato per lo più come un archivio per le metriche che descrivono le performance di sistemi e applicazioni, poiché è in grado di effettuare una rapida analisi delle loro capacità o dei loro problemi. Questo database, com'è possibile dedurre, non fa della velocità la sua più grande priorità ma si focalizza sull'offrire strumenti avanzati per la visualizzazione di serie storiche di dati [27].

3.2.4 INFLUXDB

InfluxDB è un database per serie temporali realizzato proprio per l'analisi di questo tipo di dato; è in grado di archiviare e comprimere l'informazione, fornendo anche un servizio di querying in real time. Una nota positiva di questo database è il tipo di linguaggio utilizzato che prende il nome di InfluxQL (Influx Query Language), molto simile a SQL. Tale meccanismo per l'esecuzione di query permette così a chi è già familiare con i database relazionali di non trovarsi spaesato di fronte ad un primo approccio con un TSDB. InfluxDB è il database che più si sta affermando all'interno del mondo dei database per serie temporali, proprio grazie alla sua grande capacità di gestire milioni di data point al secondo.

InfluxDB è particolarmente indicato per chi ha necessità di estrarre e archiviare grandi moli di dati caratterizzati da un timestamp in pochissimi secondi [27].

3.2.5 Conclusioni

È chiaro che la scelta del TSDB più adatto alle proprie necessità dipende dai dati che si hanno a disposizione e dal tipo di elaborazione che deve essere effettuata su di essi. In Tabella I sono racchiuse tutte le caratteristiche che risultano essere i punti di forza di ogni specifico database rendendo possibile un confronto tra quelli analizzati in precedenza.

OpenTSDB	Apache Druid	Graphite	InfluxDB
Dato memorizzato esattamente come viene inserito	Creato per grandi flussi di dati	Strumento di monitoraggio per aziende	Ideato per dispositivi IoT e applicazioni cloud
Scrittura con precisione in millisecondi	Supporta numerosi formati di file per dati strutturati o non strutturati	Permette di tenere sotto controllo le prestazioni di applicazioni e siti web	Utilizza una potente API per applicazioni real-time
Dati mantenuti senza limiti temporali	Permette di effettuare rapide query per ottenere risultati in pochi secondi	Rende semplice la visualizzazione e l'archiviazione di time series	Presenta un motore per l'analisi di serie storiche di alte prestazioni
Possibilità di generare grafici dalla GUI	Ideato per moderne applicazioni di analisi	Rendering di grafici relativi ai dati su richiesta	Si basa su un linguaggio molto simile a SQL
Open-Source e software gratuito	Open-Source e software gratuito	Open-Source e software gratuito	Open-Source e periodo di prova gratuito di 30 giorni

Tabella I: Confronto tra database per serie temporali

3.3 Elasticsearch

Elasticsearch è un motore di ricerca ed analisi, distribuito e open source, che lavora in real-time consentendo all'utente di esplorare i propri dati ad una velocità particolarmente elevata [28].

È uno strumento che viene utilizzato per ricerche di tipo full text, le quali gli consentono di esaminare tutte le parole in ogni documento archiviato al suo interno, allo scopo di individuare un riscontro tra le parole fornite e i dati a sua disposizione.

La combinazione tra ricerca full text, ricerca strutturata e analisi dei dati, consente ad Elasticsearch di essere uno dei motori di ricerca più utilizzati da grandi corporazioni:

- *Wikipedia* usa Elasticsearch per fornire ricerche full-text con segmenti di ricerca evidenziati e con un meccanismo di suggerimenti basato su ciò che l'utente digita;
- *The Guardian* usa Elasticsearch per combinare gli accessi dei visitatori con i dati presi dai social network, in modo da fornire un riscontro in tempo reale ai suoi editori rispetto la risposta dei lettori ai nuovi articoli;
- *Stack Overflow* combina la ricerca full-text con query di geolocalizzazione e utilizza il meccanismo di similarità per trovare domande e risposte;
- *GitHub* usa Elasticsearch per interrogare innumerevoli linee di codice.

Elasticsearch non è solo destinato a grandi siti web ma è aperto anche a utenti che abbiano dati a disposizione e necessità di utilizzarli per effettuare analisi o elaborazioni sugli stessi. Per permettere questo tipo di utilizzo si basa su un meccanismo che prende il nome di Apache Lucene [28].

Apache Lucene è uno strumento software, più precisamente una libreria open source funzionale all'estrazione di informazioni, definita come API (Application Programming Interface) e progettata in Java. Lucene viene impiegata per l'indicizzazione e la ricerca, per questo costituisce uno strumento fondamentale alla progettazione di motori di ricerca. Inoltre, la sua grande flessibilità permette una ricerca testuale indipendente dal tipo di documento digitale indicizzato [29].

Elasticsearch può essere descritto in diversi modi: può essere identificato come un magazzino di documenti dove ogni campo è indicizzato e ricercabile; come un motore di ricerca distribuito con analisi real-time; come un server capace di ridimensionare petabytes di dati strutturati e non strutturati. Tutte queste funzionalità sono raccolte all'interno di un singolo server con il quale le applicazioni possono dialogare attraverso semplici RESTful API che utilizzano un web client, con il quale si può comunicare attraverso un linguaggio di programmazione o direttamente attraverso il prompt dei comandi del proprio computer [28].

Per quanto riguarda il funzionamento di Elasticsearch, i dati grezzi fluiscono al suo interno da una grande varietà di fonti, che possono essere metriche di sistema o applicazioni web. L'inserimento dei dati è il processo mediante il quale essi vengono analizzati, normalizzati e arricchiti prima di essere indicizzati all'interno del motore di ricerca. Una volta indicizzati, gli utenti saranno in grado di eseguire query complesse su di essi, per effettuare analisi o elaborazioni. Un "index" in Elasticsearch è una raccolta di documenti correlati tra loro. I dati vengono archiviati al suo interno sottoforma di documenti JSON, ognuno dei quali correla una serie di chiavi con i valori corrispondenti. Inoltre, viene utilizzata una struttura di dati chiamata "a indice invertito" progettata per consentire ricerche full-text molto veloci. Tale struttura elenca ogni parola univoca che appare in qualsiasi documento e identifica quelli in cui ciascuna di esse si trova. Dunque, durante il processo di indicizzazione Elasticsearch archivia i documenti e crea un indice invertito per rendere i dati del documento ricercabili quasi in tempo reale [30].

Elasticsearch è quindi una piattaforma di ricerca che prevede una latenza minima tra il momento in cui un documento viene indicizzato e il momento in cui diventa ricercabile. Per questo motivo può essere considerato un motore di ricerca che lavora in real-time.

3.3.1 Elasticsearch vs InfluxDB

InfluxDB è un database ottimizzato per le serie temporali, mentre Elasticsearch è un motore di ricerca costruito per immagazzinare grandi quantità di dati. Ci si chiede quale sia la soluzione migliore tra i due quando l'obiettivo è quello di conservare serie di dati temporali.

InfluxDB è la scelta ottimale per applicazioni che richiedono un'interrogazione in tempo reale perché riesce a gestire un maggior numero di scritture rispetto ad Elasticsearch. Tuttavia, Elasticsearch è più adatto per dati testuali perché è stato ideato per la ricerca di questa tipologia di dato. Da qui ne deriva che esso rimane la scelta migliore per un'interrogazione dei dati attraverso il loro contenuto. Una buona soluzione per lavorare con dati temporali potrebbe essere quella di conservare gli stessi in InfluxDB e utilizzare invece Elasticsearch per i metadati (ovvero le informazioni riguardanti i dati numerici), così da localizzare velocemente i dati testuali e i dati forniti di timestamp per poi effettuare calcoli ed analisi in InfluxDB ad ogni ricezione [31].

In Figura 3.1 è possibile osservare un confronto tra le caratteristiche di InfluxDB e quelle di Elasticsearch.

INFLUXDB	VS	ELASTICSEARCH
<ul style="list-style-type: none">• Database per serie temporali• Linguaggio utile alla creazione di query: InfluxQL e Flux• Formato CSV• Indice <i>temporale</i> (timestamp)• Libreria per la comunicazione con Matlab• Versione Cloud a pagamento		<ul style="list-style-type: none">• Motore di ricerca che lavora in real-time• Ricerca full-text• Formato JSON• Indice <i>testuale</i>• Non disponibile al collegamento con Matlab• Versione Cloud a pagamento

Figura 3.1 Confronto tra InfluxDB ed Elasticsearch

Capitolo 4: Architettura dei sistemi utilizzati

4.1 Sensori NGIMU

I sensori utilizzati per l'acquisizione dei dati necessari all'analisi dell'equilibrio e dell'andatura dei pazienti affetti da malattia di Parkinson sono i NGIMU (Next Generation IMU). Sono dispositivi compatti per l'acquisizione di dati che combinano sia algoritmi per il processamento degli stessi che sensori integrati. Costituiscono una piattaforma versatile adatta sia per applicazioni in tempo reale che per la registrazione dei dati. Il dispositivo, mostrato in Figura 4.1, presenta al proprio interno un insieme di sensori tra cui giroscopio, accelerometro e magnetometro a tre assi, così come anche un sensore di pressione barometrica. Inoltre, un algoritmo di fusione del sensore AHRS integrato permette di combinare misurazioni magnetiche ed inerziali così da fornire una corretta raccolta dati dell'orientamento rispetto alla Terra. Per rafforzare questo algoritmo, ogni dispositivo viene calibrato in modo individuale attraverso l'utilizzo di apparecchiature robotiche. Una particolarità è anche la presenza di un'interfaccia di input analogica, la quale permette l'utilizzo di sensori o segnali esterni; mentre un'interfaccia seriale ausiliaria consente il collegamento di ulteriori dispositivi, come ad esempio il modulo GPS. I dispositivi NGIMU dispongono anche di una sincronizzazione wireless che permette di ottenere i timestamp di ogni misurazione sincronizzati ad un clock comune. In aggiunta a tutte le caratteristiche sopra elencate, questa tipologia di sensori dispone di un'interfaccia grafica, la NGIMU GUI (rappresentata in Figura 4.2), utilizzata per configurare i dispositivi e per esportare i dati ad un software secondario, trasformando il file dal formato XIO al formato CSV [32].

Grazie a quest'ultima funzione si riescono ad ottenere i file nel formato richiesto, così da rendere i dati disponibili al caricamento all'interno di un database per serie temporali adeguato.



Figura 4.1 Sensore NGIMU utilizzato per effettuare le misurazioni su pazienti affetti da malattia di Parkinson [32].

L'innovazione si trova nella dimensione di questi apparecchi che risultano abbastanza piccoli da poter essere indossati dal soggetto su cui vengono effettuate le misurazioni. Nel caso d'uso analizzato vengono utilizzati tre sensori NGIMU: uno per la schiena, uno per il piede destro e uno per il sinistro. Questi sono in grado di valutare orientamento, velocità, pressione, rotazione ed accelerazione, insieme al movimento in tutte le direzioni [33].

In uno step successivo, a seguito di un'elaborazione dei dati attraverso algoritmi studiati appositamente per questo scopo e scritti in ambiente Matlab, si possono ricavare dei parametri fondamentali allo staff medico, i quali aiutano a valutare meglio le condizioni fisiche e motorie del paziente sottoposto a misurazione.

4.2 Database

Per realizzare un sistema che permetta al personale medico di interagire con i dati raccolti dalle misurazioni effettuate attraverso i sensori vengono utilizzati dei database. Si è scelto di lavorare sia con un database di tipo relazionale che con un database in grado di lavorare con le serie temporali. Questa scelta deriva dalla necessità di dover elaborare dati caratterizzati da un'indicazione temporale, nel caso delle misurazioni prelevate dai sensori, i quali devono essere messi in relazione a dati puramente anagrafici che caratterizzano il paziente a cui si riferiscono. Questi ultimi possono essere gestiti attraverso l'utilizzo di un database relazionale.

4.2.1 MySQL

MySQL, rilasciato nel 1995, è un sistema open source di gestione di database relazionali SQL sviluppato e supportato da Oracle, che fa parte dei sistemi RDBMS (Relational Database Management System) [34].

Questo tipo di database è tra le tecnologie più conosciute per la gestione di grandi moli di dati ed è basato sul linguaggio SQL (Structured Query Language). Un database è una collezione di dati strutturata e, in particolare, un database relazionale è una collezione di dati che viene organizzata

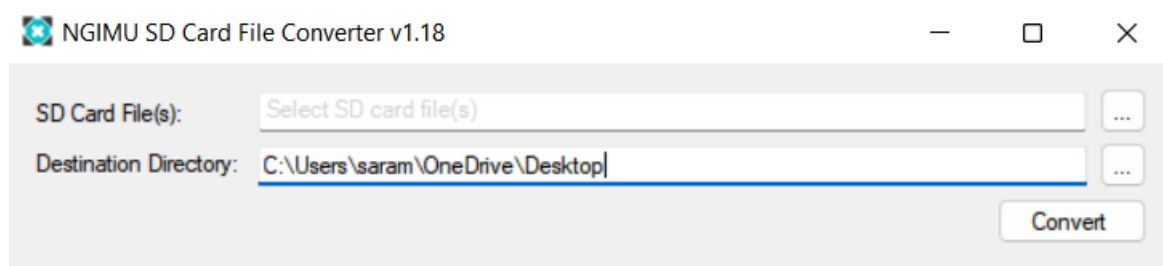


Figura 4.2 Interfaccia per la conversione del file da XIO a CSV

secondo il modello relazionale. In quest'ultimo le tabelle sono costituite da righe, colonne e da relazioni tra i dati realizzate secondo una struttura logica. MySQL viene spesso associato ad applicazioni internet e servizi web, ma è stato creato soprattutto per essere compatibile con altre tecnologie e architetture. Questo RDBMS funziona su tutte le piattaforme informatiche, tra cui sistemi operativi Unix-based, come ad esempio Linux, MacOS e Windows. Il fattore principale che permette di differenziare un database relazionale da uno storage digitale è il modo in cui i dati vengono organizzati ad alto livello. Da questo punto di vista i database come MySQL contengono dati in molteplici tabelle separate e altamente codificate, al contrario di una cartella contenente un insieme di documenti semi-strutturati o non strutturati. Questo consente ai RDBMS di ottimizzare sia la ricerca dei dati al loro interno, sia l'aggiornamento delle informazioni, ma anche altre azioni più complesse. Il modello relazionale risulta ancora oggi molto diffuso soprattutto grazie alla sua intuitività e ad un linguaggio di programmazione dichiarativo che consente di comunicare in modo diretto al database qual è il risultato che si vuole ottenere. MySQL, tra tutti i database relazionali disponibili, rimane una delle tecnologie più semplici da imparare e da utilizzare e, nonostante risulti essere uno tra i più "vecchi" creati riesce ad essere ancora perfettamente attuale, grazie anche ad un'evoluzione nell'elaborazione e nell'archiviazione dei dati nel cloud [35].

Per poter operare con un database strutturato in MySQL è possibile utilizzare il software MySQL Workbench, un programma sviluppato da Oracle attraverso cui è possibile amministrare un database da remoto dopo aver impostato opportunamente una nuova connessione [36].

4.2.2 InfluxDB

InfluxDB è un sistema di gestione di database open source programmato nel linguaggio di programmazione Google Go che per interrogare database esterni utilizza il nuovo linguaggio Flux specifico per database di serie temporali (Time Series Database, TSDB). La sintassi di Flux si basa su JavaScript ed è compatibile con diverse fonti di dati [37].

Un TSDB è un sistema di database ottimizzato per la fruizione di serie di dati temporali e la loro memorizzazione in termini di tempo e valore. Questo tipo di database, infatti, consiste di eventi che vengono monitorati e permette di ottenere un insieme di dati caratterizzati da un timestamp, un elemento fondamentale per l'identificazione della correlazione tra il dato e l'informazione temporale [38].

La colonna del tempo è inclusa in ogni database creato in InfluxDB e contiene il dato temporale associato ad ogni specifica riga. Un'altra colonna che deve essere presente è quella dei field. Ogni

field consiste di una key e un value, quest'ultimo contiene i dati veri e propri che sono sempre associati ad un timestamp, è possibile dire che ogni coppia key-value costituisce un field set.

Un altro concetto importante nella struttura di un database di InfluxDB è quello del measurement (comparabile ad una tabella SQL) che esplicita il contenuto dei field [39].

Le serie temporali di dati, memorizzate con una struttura simile a quella spiegata in precedenza, vengono conservate nei bucket (corrispondenti ai database in MySQL), ai quali viene associata una scadenza che specifica per quanto tempo ogni data point verrà mantenuto all'interno del database. InfluxDB eliminerà ogni data point che presenta dei timestamp più vecchi della data associata al bucket [40].

Ogni bucket appartiene ad una organizzazione, ovvero uno spazio di lavoro dedicato ad un gruppo di utenti a cui è associato l'intero database con i dati contenuti al suo interno [41].

Tra i vantaggi introdotti dall'utilizzo di InfluxDB in applicazioni di IoT Monitoring e Real-Time Analytics:

- È stato progettato per operare con serie temporali di dati in modo più efficiente rispetto a come questa gestione avviene in database relazionali;
- Presenta soluzioni per ogni bisogno, infatti insieme ad InfluxDB possono essere utilizzati anche i componenti del cosiddetto "TICK Stack" (Kapacitor, Telegraf, Chronograf) utili per applicazioni di vario tipo;
- InfluxDB non si limita ad un paio di plugin di input come altri tipi di database per serie temporali, infatti insieme ad HTTP API offre un plugin UDP, un plugin Graphite (per i dati in formato Graphite line protocol) e un plugin CollectD che consente di inserire dati in formato nativo collectd e molti altri;
- È garantito il supporto di Grafana, un software per time series analytics che ha introdotto in InfluxDB delle dashboard per l'analisi di dati;
- C'è la possibilità di utilizzare diversi linguaggi di programmazione tra cui .Net, Java, PHP, Python, Ruby, Scala eccetera;
- Presenta un linguaggio tipo SQL per la costruzione di query, il quale prende il nome di InfluxQL;
- Supporta le query continue che vengono avviate automaticamente e periodicamente su dati in tempo reale, salvando il risultato in una specifica measurement;
- È di facile installazione poiché la versione non Cloud, oltre ad essere gratuita, prevede un file binario senza dipendenze che può essere installato da riga di comando;

- È fornito di un'interfaccia web per semplificarne l'utilizzo per chi non è pratico con software da utilizzare tramite comandi dal prompt del computer;
- Presenta una documentazione estesa che fornisce una guida ad InfluxDB per ottimizzazione ed esecuzione di query.

Grazie a questa grande quantità di vantaggi InfluxDB, per l'elaborazione di serie di dati temporali, risulta essere il TSDB più popolare del momento (secondo DB-Engines Rankings nel 2016) [39].

4.3 Matlab

Per raggiungere l'obiettivo presentato all'inizio della tesi è fondamentale l'utilizzo del software Matlab che ci permette di far comunicare i dati presenti all'interno di entrambi i database scelti, tramite l'impiego di un'interfaccia grafica accuratamente costruita.

Per far questo sono stati utilizzati alcuni strumenti messi a disposizione dalla Mathworks e da utenti esterni, in grado di semplificare il lavoro svolto.

4.3.1 Database Toolbox

Database Toolbox contiene un certo numero di funzioni e un'applicazione utile per scambiare dati con database relazionali e non relazionali. Questo scambio avviene grazie ad una conversione

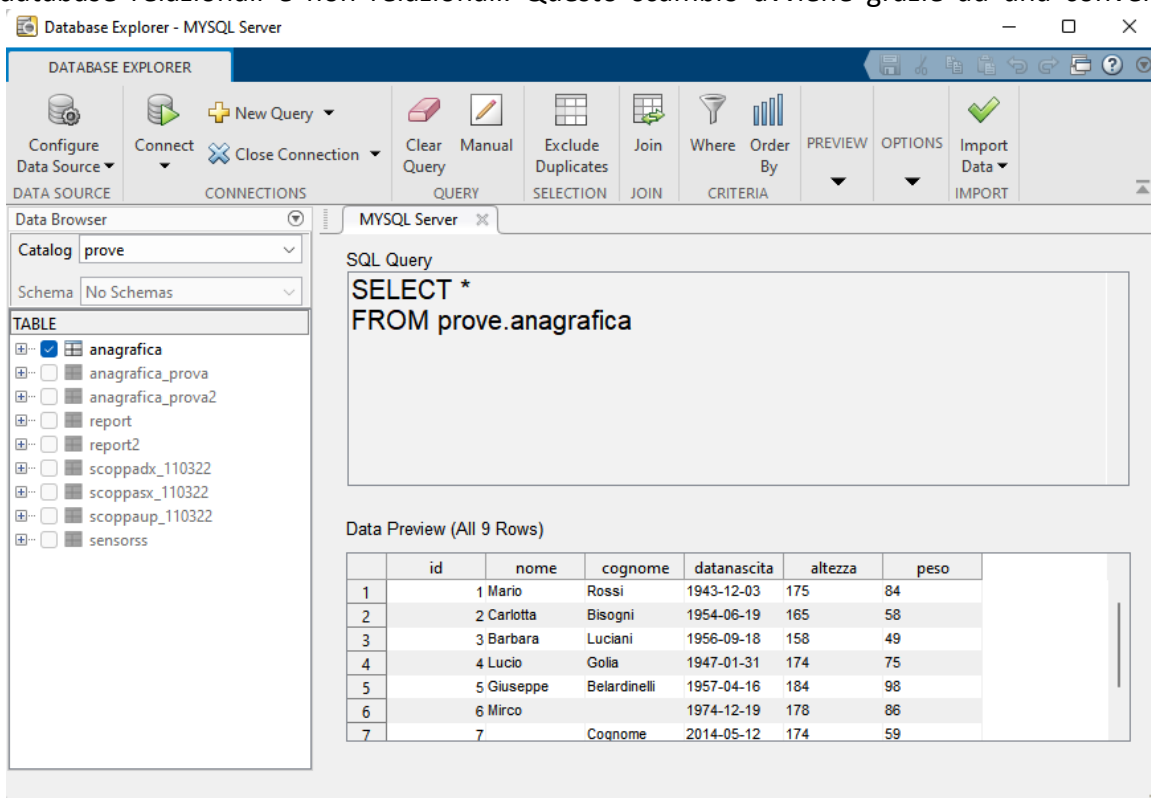


Figura 4.2 Schermata dell'applicazione Database Explorer, facente parte del Database Toolbox di Matlab.

automatica continua tra i tipi di dati Matlab e i tipi di dati dei database. C'è il supporto di ogni tipologia di database compatibile con sorgenti ODBC o JDBC ed è proprio grazie a questa caratteristica che si riescono a realizzare collegamenti con un database in MySQL. Insieme alle funzioni, fondamentale è anche la presenza dell'applicativo Database Explorer, la cui schermata è mostrata in Figura 4.2. Esso permette di esplorare i dati relazionali senza dover scrivere codice. Tramite il suo utilizzo e sfruttando la conoscenza del linguaggio SQL che agisce tramite esecuzione di query, si riescono a filtrare ed importare i dati sottoforma di tabelle all'interno di Matlab. Allo stesso modo, i dati presenti in Matlab sottoforma di tabelle possono essere esportati ed inseriti all'interno del database di riferimento [42].

4.3.2 App Designer

App Designer, mostrato in Figura 4.3, permette di creare applicazioni professionali senza dover necessariamente scrivere codice relativo agli elementi grafici. Per progettare una GUI (Graphical User Interface) si possono semplicemente sfruttare dei componenti di visualizzazione preimpostati che vanno aggiunti alla propria applicazione tramite trascinamento. In questo modo possono essere realizzate applicazioni desktop o applicazioni web autonome sfruttando Matlab Compiler. App Designer, dunque, integra le fasi principali della creazione di applicazioni, ovvero l'ideazione dei componenti e la programmazione del comportamento dell'applicativo. Il tool genera automaticamente il codice orientato a oggetti che specifica layout e design dell'applicazione e, in

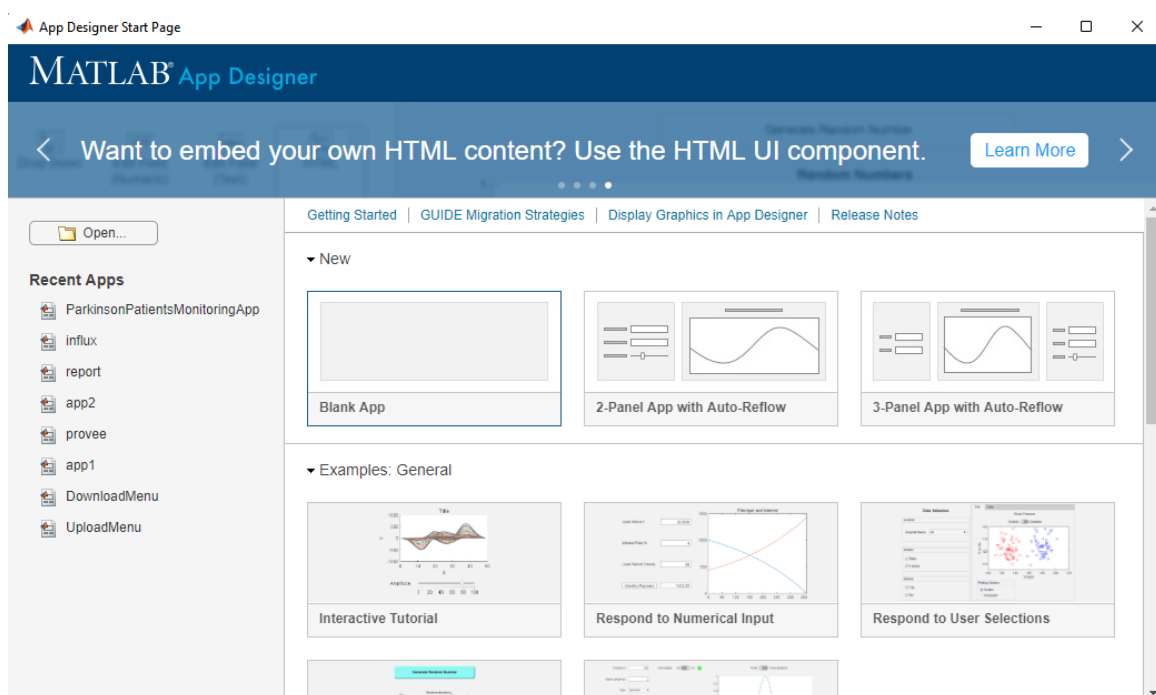


Figura 4.3 Schermata principale di App Designer

una fase successiva, il programmatore può utilizzare una versione dell'editor Matlab per definirne il comportamento. È possibile prevedere la visualizzazione di messaggi di avviso o di errore oppure si può far eseguire una determinata funzione a seguito di un'azione che l'utente compie sull'applicazione. Infatti, ai componenti possono essere assegnate delle callback, le quali devono essere opportunamente programmate per personalizzare le interazioni tra utente e interfaccia. Queste ultime avvengono attraverso l'utilizzo di mouse e tastiera [43].

Tale importante strumento messo a disposizione dal software Matlab viene utilizzato proprio per creare l'applicazione da fornire ad un gruppo di medici che si occupa di un certo numero di pazienti sottoposti a misurazioni effettuate con sensori di movimento, i cui dati devono essere tenuti sotto controllo. L'interfaccia grafica implementata prevede il collegamento tra i due database utilizzati per rendere più semplice la comunicazione tra i dati grezzi derivanti dalle misurazioni dei sensori e i dati relativi all'anagrafica di ogni paziente.

4.3.3 Codice GitHub per il collegamento ad InfluxDB

Per realizzare una comunicazione tra Matlab ed InfluxDB non può essere utilizzato Database Toolbox poiché la Mathworks non ha inserito questo database non SQL tra i compatibili. Per aggirare questo problema, dunque, è stato utilizzato un codice scritto da un utente esterno, il quale lo ha reso disponibile all'interno del social GitHub. Tale codice, inizialmente implementato solo per la versione 1.5 di InfluxDB, è stato poi aggiornato per la versione 2.0, reperibile al sito [44]. Dopo la sua installazione e la sua importazione all'interno di Matlab possono essere utilizzate le funzioni proposte sia per realizzare il collegamento alla propria organizzazione InfluxDB, sia per comunicare con un bucket specifico inserendo o estraendo dati da esso.

Prima di poter lavorare con lo specifico database di InfluxDB all'interno di Matlab è però necessario renderlo compatibile alla versione 1 di questo TSDB tramite l'utilizzo della sua versione client.

Capitolo 5: Medical Data Monitoring App

È stata realizzata un'interfaccia grafica che permette di mettere in comunicazione i database MySQL e InfluxDB. Lo scopo principale risiede nella necessità di consentire al medico di interagire sia con i dati di tipo anagrafico dei pazienti che con i dati grezzi derivanti dalle acquisizioni effettuate con i sensori. Questi ultimi, infatti, vengono memorizzati all'interno di InfluxDB per poter essere poi elaborati in modo da ottenere dei report in formato pdf, i quali verranno anch'essi memorizzati all'interno di MySQL relativamente al paziente a cui fanno riferimento. Il diagramma di flusso rappresentante lo schema logico che si trova dietro la scrittura del codice con cui è stata realizzata l'applicazione è visibile in Figura 5.1.

5.1 InfluxDB

InfluxDB, database per serie temporali, viene utilizzato all'interno dell'applicazione per memorizzare i dati provenienti dalle acquisizioni eseguite attraverso l'uso di sensori per la rilevazione del movimento di un paziente. L'organizzazione di InfluxDB è stata utilizzata allo scopo di conservare i dati grezzi in formato CSV, caratterizzati da un indice temporale che esplicita il momento in cui un preciso dato è stato misurato.

Questi dati potranno essere usati in una successiva elaborazione allo scopo di fornire report riguardanti lo stato di salute del paziente.

5.1.1 Guida all'installazione

Per scaricare InfluxDB in versione non cloud, dunque utilizzabile in locale sul proprio computer, è possibile accedere al sito [45] e selezionare la piattaforma su cui si desidera effettuare il download. A seconda del sistema operativo sarà necessario installare i file binari relativi ad InfluxDB accedendo ai link riportati sotto:

Per Windows: <https://dl.influxdata.com/influxdb/releases/influxdb2-2.1.1-windows-amd64.zip>.

Per MacOS: <https://dl.influxdata.com/influxdb/releases/influxdb2-2.2.0-darwin-amd64.tar.gz>.

Per Linux: <https://dl.influxdata.com/influxdb/releases/influxdb2-2.2.0-linux-amd64.tar.gz>.

A questo punto, attraverso prompt dei comandi, accedendo alla cartella di installazione del programma, scaricata dal link adeguato, può essere lanciata l'installazione e l'esecuzione del software attraverso il comando `influxd`, come mostrato in Figura 5.2. Lavorando in locale, dunque, per comunicare con InfluxDB si può accedere all'indirizzo <http://localhost:8086>, al quale verrà richiesto di creare un utente. Altrimenti, installando la versione client, le operazioni possono essere

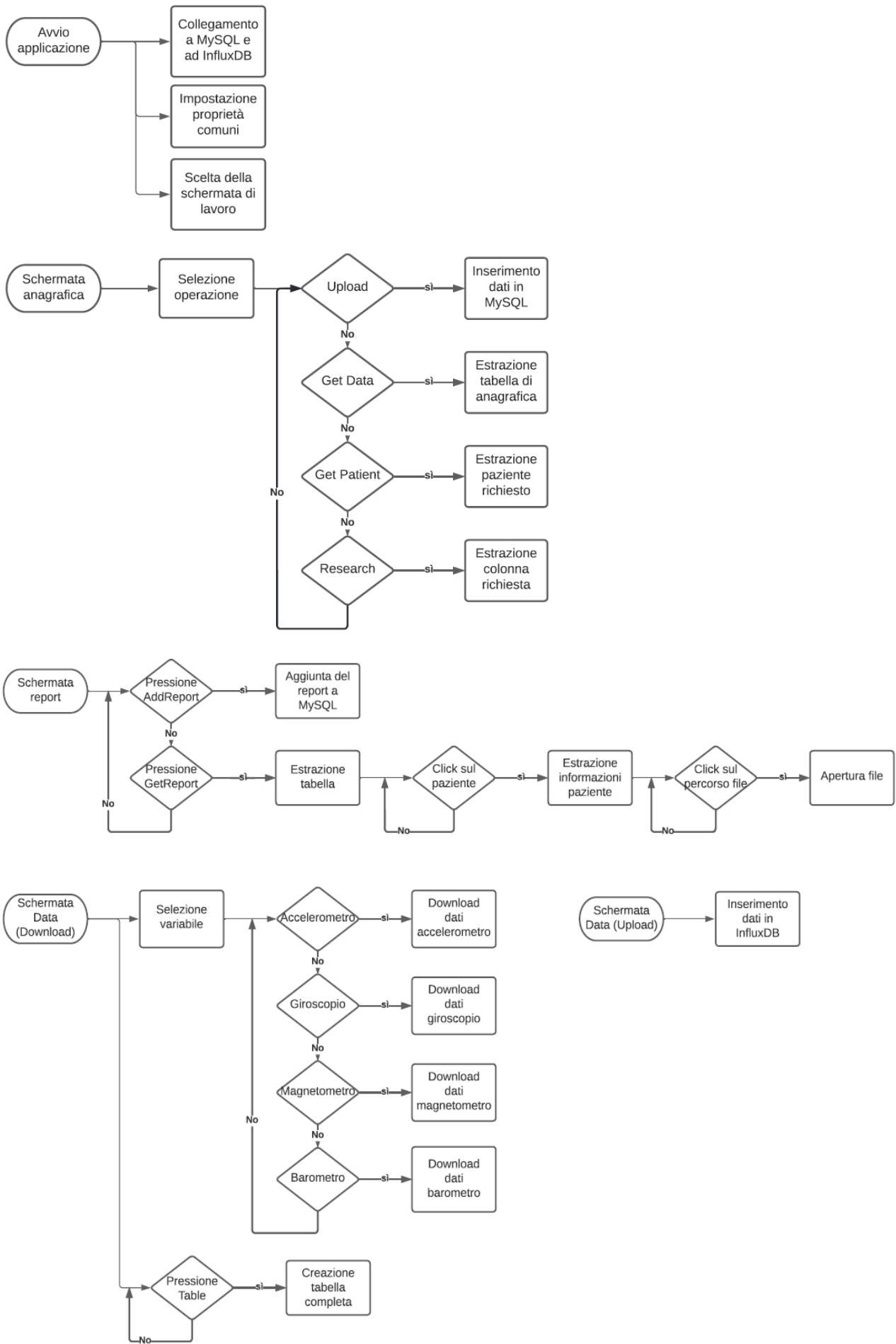


Figura 5.1 Diagramma di flusso del codice relativo all'applicazione Matlab

eseguite interamente all'interno del prompt dei comandi, utilizzando istruzioni che è possibile reperire all'interno della documentazione online.

In particolare, la versione client è utile per rendere compatibili alla versione 1 i bucket che sono stato creati nella versione 2 di InfluxDB; questa operazione viene eseguita con il comando `influx v1 auth create`.

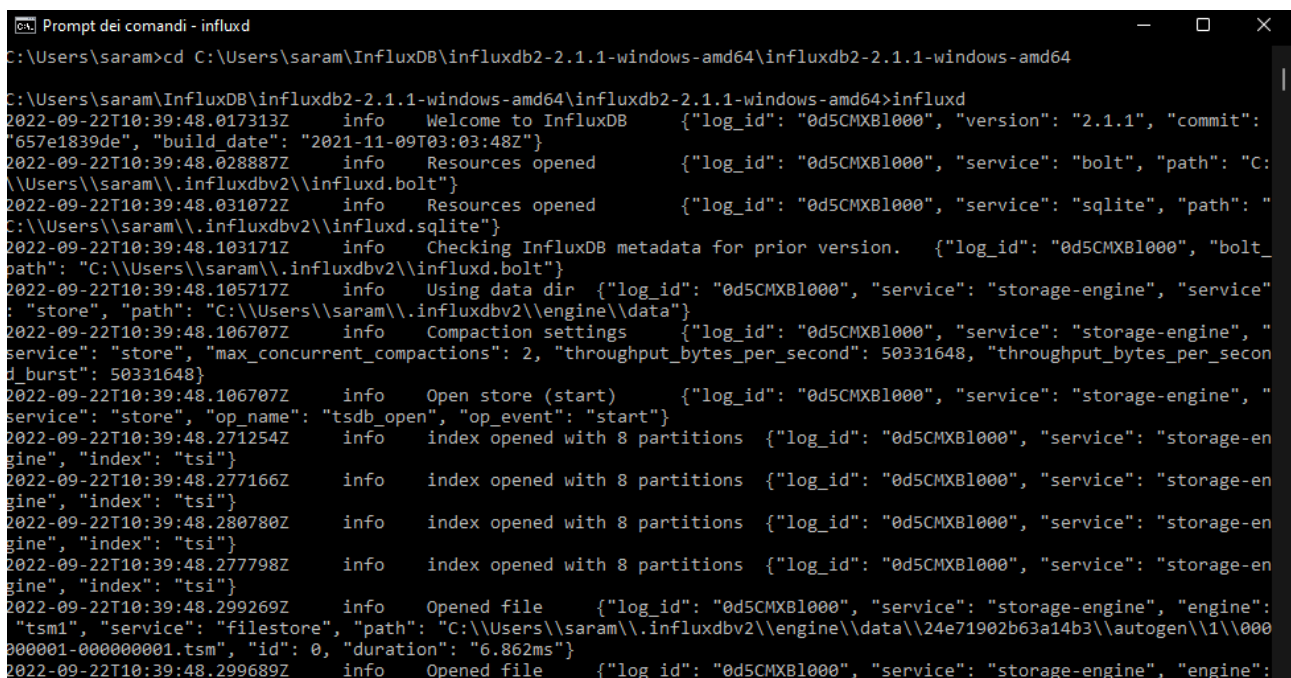
5.1.2 Collegamento tra InfluxDB e Matlab

Per collegare l'organizzazione creata in InfluxDB a Matlab è stata utilizzata una libreria di funzioni dedicate creata appositamente da un utente e condivisa all'interno della piattaforma di hosting GitHub. Dopo aver scaricato la cartella disponibile, è necessario importarla in Matlab attraverso il comando `addpath()` per riuscire a lavorarci.

Attraverso le funzioni messe a disposizione dal codice è stato possibile realizzarne altre che rispondessero alle necessità poste dall'utilizzo di InfluxDB all'interno dell'applicazione.

Tra queste `[influxdb] = influxConn(URL,TOKEN,ORG,DATABASE)` permette di instaurare una connessione alla propria organizzazione di InfluxDB, fornendo in ingresso:

- URL che corrisponde all'indirizzo http sul quale lavora il database, nel caso in cui si stia lavorando con la versione non cloud sarà URL = 'http://localhost:8086';
- TOKEN corrisponde alla chiave dell'organizzazione con cui vogliamo comunicare;
- ORG è il nome dell'organizzazione di InfluxDB con cui stiamo lavorando;



```
Prompt dei comandi - influxd
C:\Users\saram>cd C:\Users\saram\InfluxDB\influxdb2-2.1.1-windows-amd64\influxdb2-2.1.1-windows-amd64

C:\Users\saram\InfluxDB\influxdb2-2.1.1-windows-amd64\influxdb2-2.1.1-windows-amd64>influxd
2022-09-22T10:39:48.017313Z info Welcome to InfluxDB {"log_id": "0d5CMXB1000", "version": "2.1.1", "commit":
"657e1839de", "build_date": "2021-11-09T03:03:48Z"}
2022-09-22T10:39:48.028887Z info Resources opened {"log_id": "0d5CMXB1000", "service": "bolt", "path": "C:
\Users\saram\influxdbv2\influxd.bolt"}
2022-09-22T10:39:48.031072Z info Resources opened {"log_id": "0d5CMXB1000", "service": "sqlite", "path": "
C:\Users\saram\influxdbv2\influxd.sqlite"}
2022-09-22T10:39:48.103171Z info Checking InfluxDB metadata for prior version. {"log_id": "0d5CMXB1000", "bolt_
path": "C:\Users\saram\influxdbv2\influxd.bolt"}
2022-09-22T10:39:48.105717Z info Using data dir {"log_id": "0d5CMXB1000", "service": "storage-engine", "service"
: "store", "path": "C:\Users\saram\influxdbv2\engine\data"}
2022-09-22T10:39:48.106707Z info Compaction settings {"log_id": "0d5CMXB1000", "service": "storage-engine", "
service": "store", "max_concurrent_compactions": 2, "throughput_bytes_per_second": 50331648, "throughput_bytes_per_secon
d_burst": 50331648}
2022-09-22T10:39:48.106707Z info Open store (start) {"log_id": "0d5CMXB1000", "service": "storage-engine", "
service": "store", "op_name": "tsdb_open", "op_event": "start"}
2022-09-22T10:39:48.271254Z info index opened with 8 partitions {"log_id": "0d5CMXB1000", "service": "storage-en
gine", "index": "tsi"}
2022-09-22T10:39:48.277166Z info index opened with 8 partitions {"log_id": "0d5CMXB1000", "service": "storage-en
gine", "index": "tsi"}
2022-09-22T10:39:48.280780Z info index opened with 8 partitions {"log_id": "0d5CMXB1000", "service": "storage-en
gine", "index": "tsi"}
2022-09-22T10:39:48.277798Z info index opened with 8 partitions {"log_id": "0d5CMXB1000", "service": "storage-en
gine", "index": "tsi"}
2022-09-22T10:39:48.299269Z info Opened file {"log_id": "0d5CMXB1000", "service": "storage-engine", "engine":
"tsm1", "service": "filestore", "path": "C:\Users\saram\influxdbv2\engine\data\24e71902b63a14b3\autogen\1\000
000001-000000001.tsm", "id": 0, "duration": "6.862ms"}
2022-09-22T10:39:48.299689Z info Opened file {"log_id": "0d5CMXB1000", "service": "storage-engine", "engine":
```

Figura 5.2 Installazione InfluxDB da riga di comando

- DATABASE è il nome del bucket su cui verranno caricati o scaricati i dati.

Dopo aver correttamente creato una connessione al bucket su cui si deve lavorare, è possibile implementare delle funzioni che permettano di effettuare operazioni di upload e di download di dati all'interno del database.

Nel caso in esame l'obiettivo prevede che InfluxDB funga da "contenitore" per i dati grezzi provenienti dalle acquisizioni effettuate attraverso i sensori di movimento; per far questo vengono utilizzate le funzioni derivanti dal codice fornito tramite GitHub per caricare file CSV all'interno del bucket di riferimento. Prima di effettuare l'effettivo upload è necessario manipolare i dati originali per ottenere una timetable che soddisfi i requisiti imposti da InfluxDB. In particolare, viene modificata la colonna contenente il dato temporale, utilizzando una riga di codice che permette di unire l'informazione riguardante l'orario di acquisizione a quella della data in cui è stata effettuata la misurazione. La funzione che si occupa sia della giusta formattazione della table derivante dalla lettura del CSV, sia dell'upload di tali dati all'interno del bucket è `influxUpload(influxdb, file_path, measurement, date)`. I dati richiesti in ingresso sono:

- `influxdb`, ovvero il collegamento che deve essere effettuato prima dell'operazione di upload attraverso la funzione `influxConn`;
- `file_path` è la destinazione del file csv contenente i dati provenienti dalle acquisizioni che devono essere caricati all'interno del bucket specificato in fase di collegamento;
- `measurement`, ovvero il nome della "tabella" all'interno della quale i file verranno memorizzati;
- `date` che è la data del giorno in cui è stata effettuata l'acquisizione;

La funzione `influxUpload` prevede delle righe di codice che si occupano di prendere il file in ingresso e di modellarlo opportunamente in modo da ottenere una timetable avente il formato richiesto da InfluxDB. L'upload vero e proprio dei dati viene eseguito da un ciclo `for` all'interno del quale vengono utilizzate le funzioni fornite dal codice GitHub. Oltre al ciclo `for`, per file che prevedono un numero di righe superiore ai cinque milioni, è presente una serie di condizioni `if` che continuano a scorrere la timetable fino a caricare tutte le righe che la costituiscono.

Al contrario, per riuscire a scaricare i dati presenti all'interno del bucket specificato, sono state implementate diverse funzioni a seconda della variabile che si ha necessità di estrarre. Quest'ultime prendono il nome di `getAccelometer`, `getMagnetometer`, `getGyroscope` e `getBarometer`; presentano tutte lo stesso tipo di implementazione e richiedono in ingresso `influxdb`, `measurement` e `date`, i quali coincidono con le descrizioni fornite in precedenza. In uscita presentano una tabella

costituita sia dalla colonna dell'informazione temporale che dalle coordinate cartesiane (x,y,z) della variabile selezionata.

Infine, a partire dai singoli parametri estratti, la funzione che viene utilizzata per ricostruire la tabella completa al termine delle operazioni è `parameters = makeTable(Accelerometer, Gyroscope, Magnetometer, Barometer)`, dove in ingresso vengono richieste le tabelle ottenute dalle funzioni elencate precedentemente.

5.2 Database MySQL

Il database creato in MySQL ha il compito di contenere i dati relativi alle informazioni che possono essere attribuite ad ogni singolo paziente sottoposto alla misurazione di parametri attraverso l'utilizzo di sensori specifici da indossare su piede destro, piede sinistro e schiena.

In particolare, una prima tabella che deve essere presente nel database è quella di anagrafica, costruita per contenere i dati sensibili di ogni paziente, ovvero nome, cognome, data di nascita, altezza e peso.

Un'altra tabella è poi quella costruita per contenere i report relativi ad ogni paziente, derivanti dall'analisi di ogni acquisizione, con corrispondente data in cui è avvenuta la misurazione. Un report

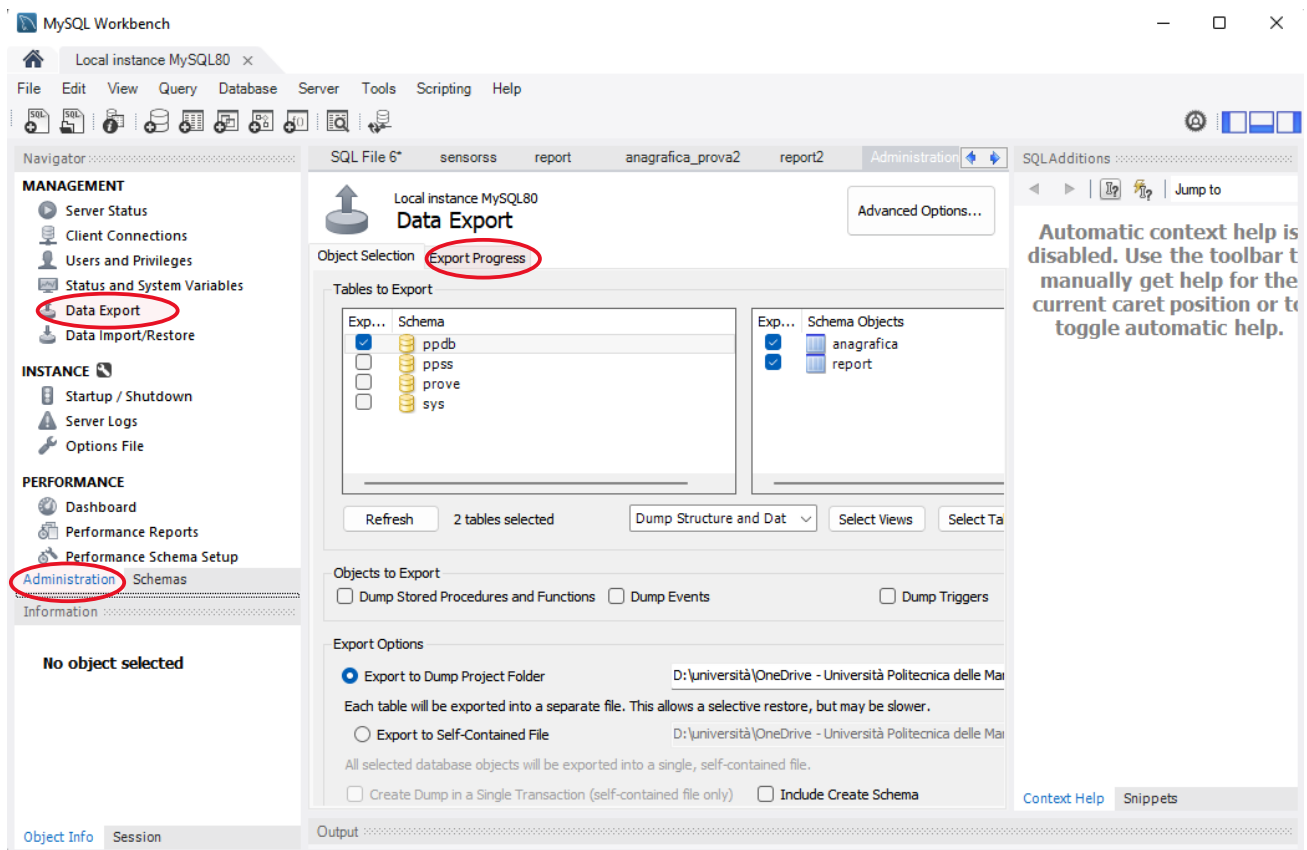


Figura 5.3 Esportazione database da MySQL Workbench

deriva dall'elaborazione dei dati grezzi di ogni acquisizione e costituisce un file pdf che viene poi inserito nella tabella tramite l'applicazione realizzata in Matlab.

5.2.1 Esportare il database da MySQL Workbench

Per esportare un database da MySQL Workbench occorre accedere alla propria istanza locale inserendo nome utente e password.

Dalla sezione di sinistra del software deve essere selezionato "Administration" e successivamente, sotto la voce "Management", è necessario cliccare su "Data Export". A questo punto si aprirà una finestra dedicata all'esportazione del database il quale, per essere esportato, deve essere selezionato.

Per includere anche tutte le sue tabelle occorre fare doppio click sul nome del database selezionato. Dopo aver anche indicato la cartella sulla quale vogliamo che i dati vengano salvati, cliccando su "Export Progress" e poi su "Start Export", come si può vedere in Figura 5.3, si porterà a termine l'operazione.

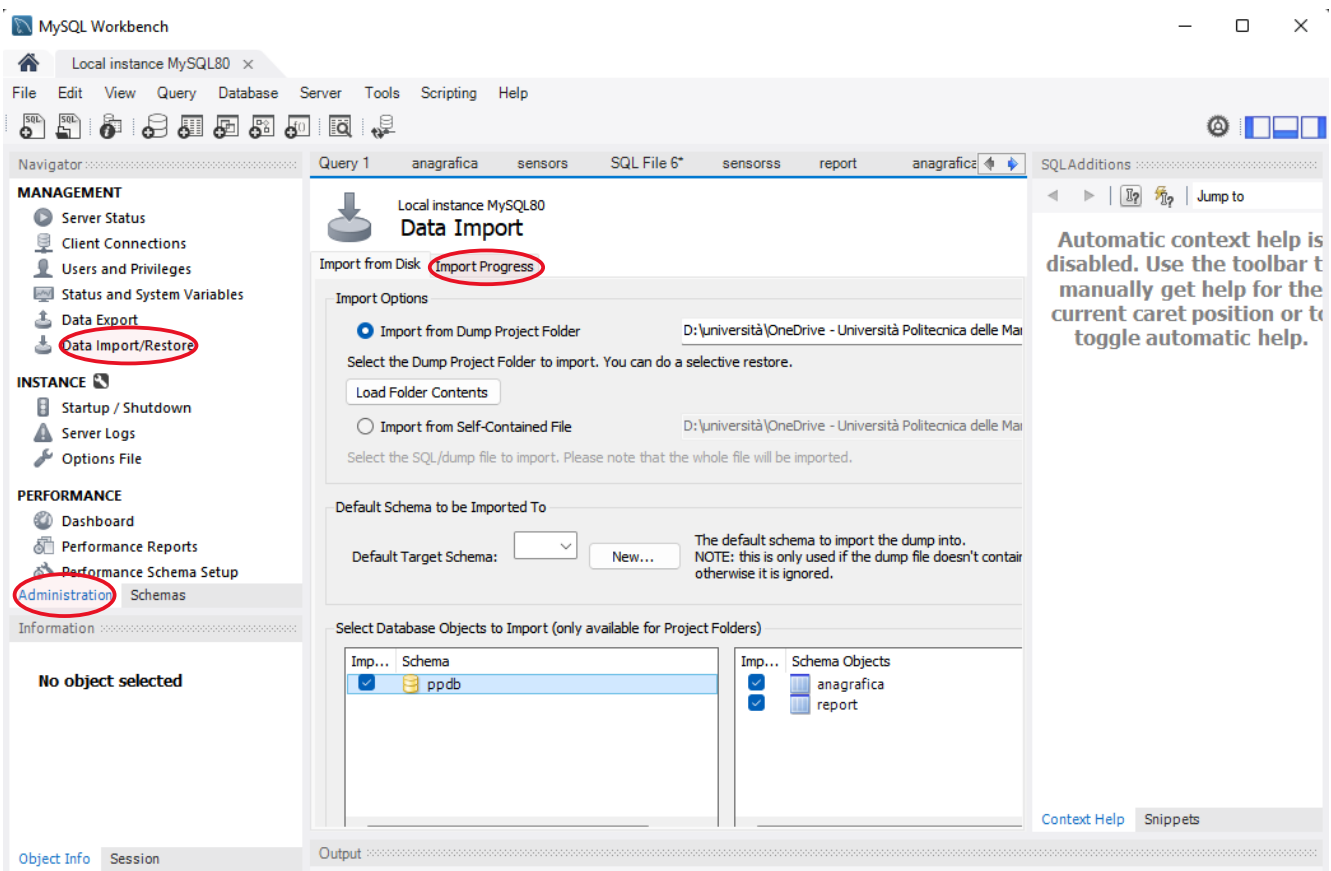


Figura 5.4 Importazione database in MySQL Workbench

5.2.2 Importare il database su MySQL Workbench

Per importare un database in MySQL Workbench è necessario accedere alla propria istanza locale inserendo nome utente e password.

Dalla sezione che si trova sulla sinistra occorre selezionare “Administration” e successivamente, sotto la voce “Management”, cliccare su “Data Import/Restore”.

A questo punto si aprirà una finestra dedicata all’importazione del database: sulla schermata denominata “Import Options” va indicato il percorso file dove si trova la cartella contenente il dump del database che si vuole importare, poi è necessario cliccare su “Load Folder Contents”. Va selezionato il database che comparirà in basso, come si può vedere in Figura 5.4, e per importare anche tutte le tabelle va effettuato un doppio click sul nome del database selezionato.

Nella sezione Default Target Schema, è necessario cliccare su “New” e inserire il nome del database che si sta importando. Dopo aver fatto queste operazioni andando su “Import Progress” e cliccando “Start Import” si può portare a termine l’importazione del database.

5.2.3 Creazione di un’origine dati ODBC

Per collegarsi e per poter comunicare con il database che abbiamo creato in MySQL è necessario creare un’origine dati ODBC, ovvero una sorgente dati che permette di instaurare la comunicazione. Per far questo occorre aver installato il connettore ODBC per MySQL relativo al sistema operativo utilizzato, disponibile al sito [46]. Una volta installato, si deve accedere all’Amministratore origine dati ODBC del proprio computer e andare sulla voce “DNS di sistema”. Da questa pagina, cliccando su “Aggiungi”, si può configurare la propria origine dati, come mostrato in Figura 5.5 a sinistra.

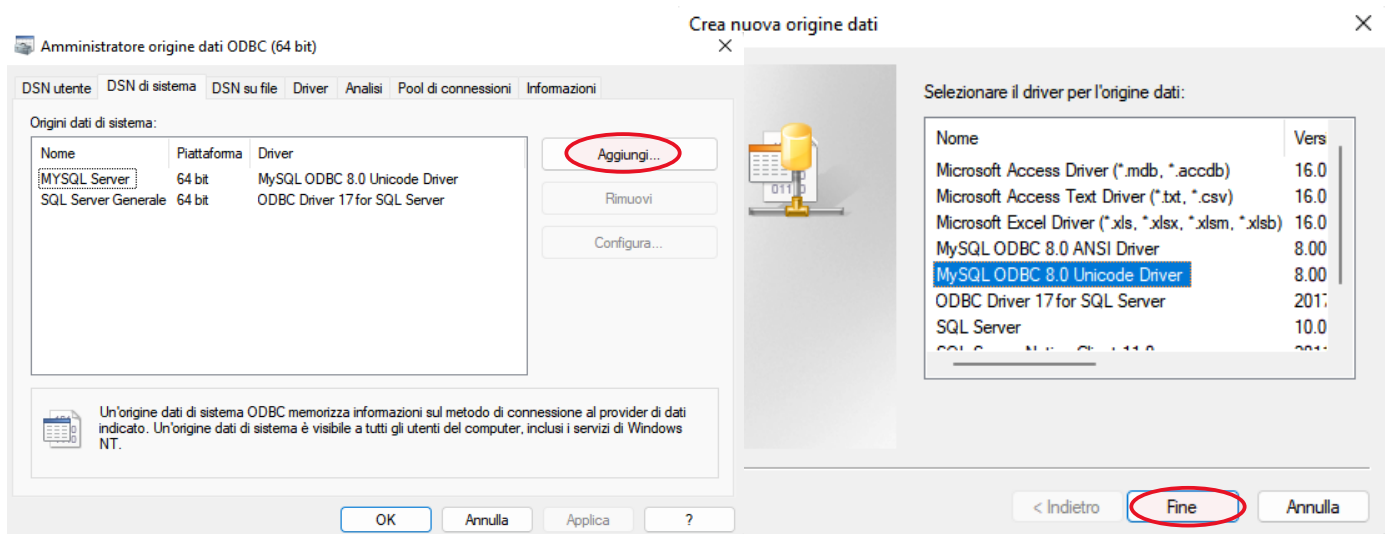


Figura 5.5 Creazione origine dati ODBC

A questo punto selezionando MySQL ODBC 8.0 Unicode Driver e premendo “Fine” (Figura 5.5 a destra) si aprirà un’altra pagina, illustrata in Figura 5.6, in cui sarà necessario andare a definire il nome della propria origine dati, il nome utente e una password necessari a connettersi al database attraverso l’ODBC (si consiglia di utilizzare gli stessi username e password che si utilizzano per accedere alla propria istanza su MySQL Workbench).

Per connettersi a qualunque database creato lasciare vuota la sezione “Database”.

Un’operazione importante di cui tenere conto per riuscire ad effettuare query di importazione file all’interno del nostro database è quella di attivare la voce “Enable LOCAL DATA INFILE statements” che si trova sotto Details→Misc (come indicato in Figura 5.6).

A questo punto cliccando “Ok” la configurazione della propria origine dati sarà completa.

5.2.4 Collegamento tra MySQL e Matlab

Grazie ad un toolbox di matlab chiamato Database Toolbox, il quale dispone di un’applicazione che prende il nome di Database Explorer, è possibile realizzare una connessione a MySQL all’interno di Matlab, ad esempio per riuscire ad eseguire query in linguaggio SQL attraverso codice Matlab.

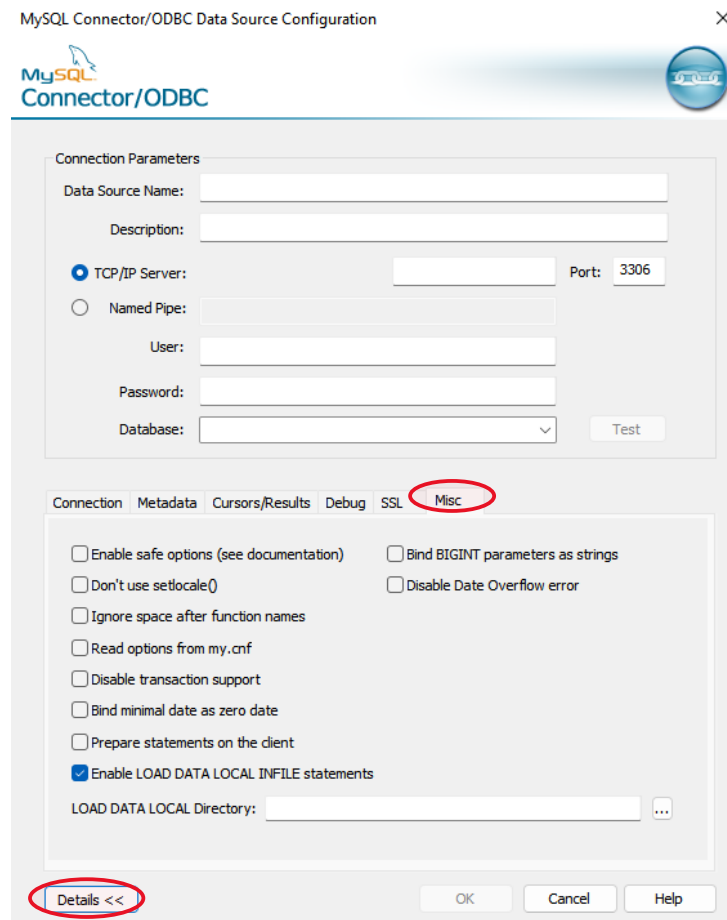


Figura 5.6 Configurazione origine dati ODBC

Dall'interfaccia dell'applicazione citata, cliccando su "Configure Data Source" e selezionando "Configure ODBC Data Source" è possibile connettersi all'origine dati precedentemente creata e, dunque, al proprio database MySQL. In questo modo si riesce ad interagire con il database e, attraverso l'opzione "Import Data" si può ottenere il codice Matlab che descrive la connessione al database e la formattazione delle righe di codice che costituiscono le query.

Tra queste, le più importanti sono:

- `conn = database(nome_ODBC, nome_utente, password);` che è una riga di codice essenziale per instaurare la connessione al database. Nome_ODBC coincide con il nome utilizzato in fase di creazione dell'origine dati, nome_utente e password sono le credenziali che si sono utilizzate sempre durante la fase di creazione dell'ODBC;
- `data = fetch(conn, query);` è la riga di codice che ci permette di lanciare l'esecuzione di una query precedentemente definita;
- `close(conn);` permette di chiudere la connessione al database quando si è terminato il lavoro su di esso.

5.3 Interfaccia grafica Matlab

È stata realizzata un'interfaccia grafica, attraverso l'utilizzo del software Matlab, che permette al dottore di interagire attivamente con il database contenente i dati relativi ai pazienti sotto esame. Il dottore, oltre a poter gestire l'anagrafica di suddetti pazienti, è così in grado di memorizzare i file CSV contenenti le acquisizioni raccolte attraverso l'uso di sensori da indossare ai piedi e sulla schiena, le quali potranno essere eventualmente elaborate per poter ottenere un report dedicato al paziente.

Per realizzare la GUI è stato utilizzato lo strumento di Matlab che prende il nome di App Designer, il quale semplifica la realizzazione e la progettazione di applicazioni consentendo all'utente di non dover scrivere codice riguardante gli oggetti grafici. Il compito del programmatore, dunque, si riduce alla sola scrittura del codice riguardante le azioni che devono essere compiute a seguito di un'interazione utente-applicazione. Tramite la scrittura di callback può essere specificata la risposta che segue, ad esempio, alla pressione di un pulsante o all'inserimento di campi testuali.

Dalla schermata di avvio del tool App Designer occorre innanzitutto selezionare il tipo di applicazione che stiamo per realizzare; in questo caso, è stata selezionata una "Blank App" in modo da poter gestire autonomamente ogni elemento grafico. A questo punto si possono inserire gli elementi e i componenti necessari attraverso il trascinamento degli stessi dalla Component Library,

fino all'interno del rettangolo che rappresenta l'applicazione. In questo modo Matlab genera automaticamente il codice relativo alla realizzazione di quel componente, del quale invece il programmatore deve scrivere il codice riferito ad ogni callback. Per aggiungere la funzione di callback di uno specifico elemento è necessario effettuare un click con il destro su di esso, aprire il menu a tendina relativo alla voce "Callbacks" ed aggiungere quella che si vuole implementare.

Gli elementi fondamentali dell'applicazione sviluppata sono sicuramente le tre schermate che permettono di agire su diversi dati a seconda degli elementi con cui si sta lavorando. Per realizzare delle schermate all'interno della GUI occorre aggiungere il componente "Tab Group" che consente, appunto, di generare delle tab di selezione attraverso cui è possibile passare da una finestra all'altra.

5.3.1 La GUI (Graphical User Interface)

L'interfaccia grafica è stata progettata pensando al ruolo del dottore che si trova ad interfacciarsi con un certo numero di pazienti. Il monitoraggio viene eseguito soprattutto dal punto di vista delle rilevazioni effettuate grazie all'utilizzo di particolari sensori. Questi dispositivi sono in grado di raccogliere dati numerici che, se opportunamente elaborati, forniscono informazioni riguardanti lo stato di salute del paziente sottoposto a misurazione (in particolare sull'andatura, sul rischio di caduta, sulla frequenza con cui avvengono i blocchi eccetera).

L'idea da cui nasce la realizzazione di questa interfaccia è la necessità di rendere semplice l'interazione tra il dottore e i dati sia numerici che testuali; infatti, sono coinvolte le informazioni personali di ogni paziente, raccolte nella tabella di anagrafica del database MySQL. Da queste poi è possibile aggiungere acquisizioni da elaborare, memorizzate all'interno di InfluxDB, utili ad ottenere report complessivi relativi ad ognuno dei pazienti sotto esame.

Nella schermata principale dell'applicazione troviamo la parte dedicata alla gestione dell'anagrafica dove possono essere eseguite quattro diverse operazioni: caricamento di un nuovo paziente, ricerca di un paziente attraverso il cognome, richiesta di tutte le informazioni dei pazienti presenti nel database, ricerca di uno specifico campo della tabella.

In una seconda schermata, indicata con l'etichetta "Report", è presente la parte di gestione dei report di ogni singolo paziente. In questa sezione alla pressione del pulsante la tabella si popola con i nomi dei pazienti ai quali può essere assegnato un report derivante dall'analisi e dall'elaborazione dei dati provenienti dalle acquisizioni caricate in InfluxDB. Tale report è costituito da un file pdf contenente il resoconto dell'elaborazione dei dati, il quale può essere caricato all'interno di MySQL per essere associato al relativo paziente.

Come ultima schermata si ha quella per la gestione dei dati grezzi proveniente dai sensori, raccolti sottoforma di file CSV. Essa permette di effettuare un download o un upload di ogni misurazione all'interno della propria organizzazione di InfluxDB.

5.3.3 Schermata Anagrafica

Nella schermata che permette di gestire l'anagrafica (mostrata in Figura 5.7) si prevede un collegamento al database MySQL, il quale deve essere utilizzato per eseguire le varie query. Per far questo si può sfruttare la definizione di proprietà pubbliche dell'interfaccia, andando ad inserire la riga di codice `conn = database('MYSQL Server', 'root', 'Tirocinio2022')`. Questa serve affinché la comunicazione venga instaurata tramite l'utilizzo della sorgente dati ODBC denominata "MYSQL Server" e con nome utente e password coincidenti rispettivamente con "root" e "Tirocinio2022".

Analogamente all'interno delle proprietà viene definito il collegamento al database InfluxDB, utile per la connessione alla propria organizzazione, e alcune variabili che possono essere condivise tra diverse callback, le quali dovranno essere richiamate con la notazione `app.Variabile`.

Per realizzare la schermata di gestione dell'anagrafica, è stato inserito un gruppo di bottoni, utile alla selezione dell'operazione che si vuole compiere, e dei campi di inserimento testuale. Un componente importante è poi il pulsante, il quale permette di far sapere all'applicazione che, alla

The screenshot shows the 'Parkinson Patients Monitoring App' window with a light blue background and a grey header. The header contains three tabs: 'Registry', 'Report', and 'Data'. The main interface is divided into four quadrants:

- Select Operation:** Features a language toggle for 'Eng' and 'Ita'. A central box contains four radio buttons: 'Upload' (selected), 'Get Data', 'Get Patient', and 'Research'. A blue 'Go' button is positioned below.
- Patient Information:** Contains input fields for 'Name', 'Surname', 'Date of birth' (with a 'yyyy-mm-dd' placeholder), 'Height', 'Weight', and 'Research'.
- Upload File:** Includes a 'File' box with radio buttons for 'Right foot' (selected), 'Left foot', and 'Back'. To the right are input fields for 'Acquisition Date', 'Patient', and 'File path', followed by a blue 'Upload' button.
- Get Report:** Contains input fields for 'Patient', 'Patient Id', 'Acquisition Date', and 'Somministration Hour', with a blue 'Report' button at the bottom.

Figura 5.7 Schermata principale dell'applicazione per la gestione dell'anagrafica

sua pressione, deve svolgere una precisa funzione. Infatti, questa intera finestra viene gestita dalla sola callback relativa alla pressione del pulsante: prende il nome di “GoButtonPushed”.

All'interno di quest'ultima compaiono quattro diversi if che effettuano il controllo condizionale sullo stato del gruppo di bottoni e, alla selezione di una precisa richiesta, l'if che prevede una condizione true esegue diverse operazioni.

Quando viene selezionato l'Upload è necessario riempire i vari campi testuali, poiché all'interno del codice che gestisce l'applicazione vengono utilizzati per eseguire la query “INSERT INTO database.tabella VALUES (...)”. Quando l'operazione giunge al termine viene stampato un messaggio di conferma in lingua italiana, o inglese, a seconda della posizione dello switch utilizzato per impostare la lingua dell'applicazione.

Quando è selezionato il campo Get Data, invece, non deve essere inserita nessuna informazione, perché alla pressione del pulsante viene eseguita la query “SELECT * FROM database.tabella”. Questa ci permette di recuperare l'intero contenuto della tabella di anagrafica, stampandola poi all'interno di una finestra secondaria.

Se ad essere selezionato è il campo di Get Patient, viene richiesto l'inserimento del cognome del paziente di cui si vogliono conoscere tutte le informazioni. Per ottenere tale risultato viene eseguita la query “SELECT * FROM database.tabella WHERE cognome LIKE inserimento”. Anche in questo caso il risultato viene stampato e mostrato attraverso una finestra pop up.

Infine, se viene selezionata l'ultima operazione disponibile, ovvero quella di Research, si chiede all'utente di inserire nell'omonimo campo quale informazione (ovvero colonna) si vuole estrarre dalla tabella. Qui viene eseguita la query “SELECT inserimento FROM database.tabella” che stampa la colonna richiesta in una schermata, la quale comparirà a seguito della pressione del pulsante.

5.3.4 Schermata Report

Anche in questa seconda schermata (mostrata in Figura 5.8) si lavora con MySQL, ma in questo caso oltre alla tabella di anagrafica abbiamo necessità di comunicare anche con quella dei report, all'interno della quale per ogni paziente viene riportato il report derivante dall'elaborazione dei dati grezzi estratti dai sensori, insieme al diario giornaliero compilato dal paziente. Per poter effettuare un collegamento tra le due tabelle è fondamentale l'utilizzo della query di JOIN che però richiede l'utilizzo degli indici di ogni tabella; per ottenere una tabella completa di tutte le informazioni, infatti, si fa uso dell'ID relativo al paziente a cui si vuole assegnare un report.

Anche in questo caso le operazioni da compiere dipendono dalla pressione di uno specifico pulsante. In questa schermata ne troviamo due: uno si occupa dell'inserimento del report di un paziente che sia già esistente nella tabella di anagrafica, mentre l'altro si occupa dell'estrazione della tabella completa.

Con la pressione del pulsante Get Report, la tabella superiore presente nella schermata dell'applicazione si popola dei dati presenti nella tabella di report di MySQL. Questa operazione viene gestita dalla callback "GetReportButtonPushed", all'interno della quale si esegue la query "SELECT (...) FROM database.tabella" e successivamente le operazioni necessarie alla copia dei dati dalla tabella di MySQL a quella di Matlab.

Nel caso in cui venga premuto il pulsante Add Report, la callback in esame prende il nome di "AddReportButtonPushed" e fa sì che le informazioni testuali inserite dall'utente nei vari campi vengano utilizzati per realizzare una query di "INSERT INTO database VALUES (...)". Questa funzione consente anche di aggiornare simultaneamente la tabella di report alla quale si aggiungerà la nuova riga.

Da questa schermata è possibile eseguire anche altre due operazioni, infatti attraverso la callback "UITableCellSelection", se dalla tabella ottenuta dalla pressione di Get Report si effettua un singolo click sul cognome di un paziente, questo permette di ottenere tutte le informazioni di quel preciso

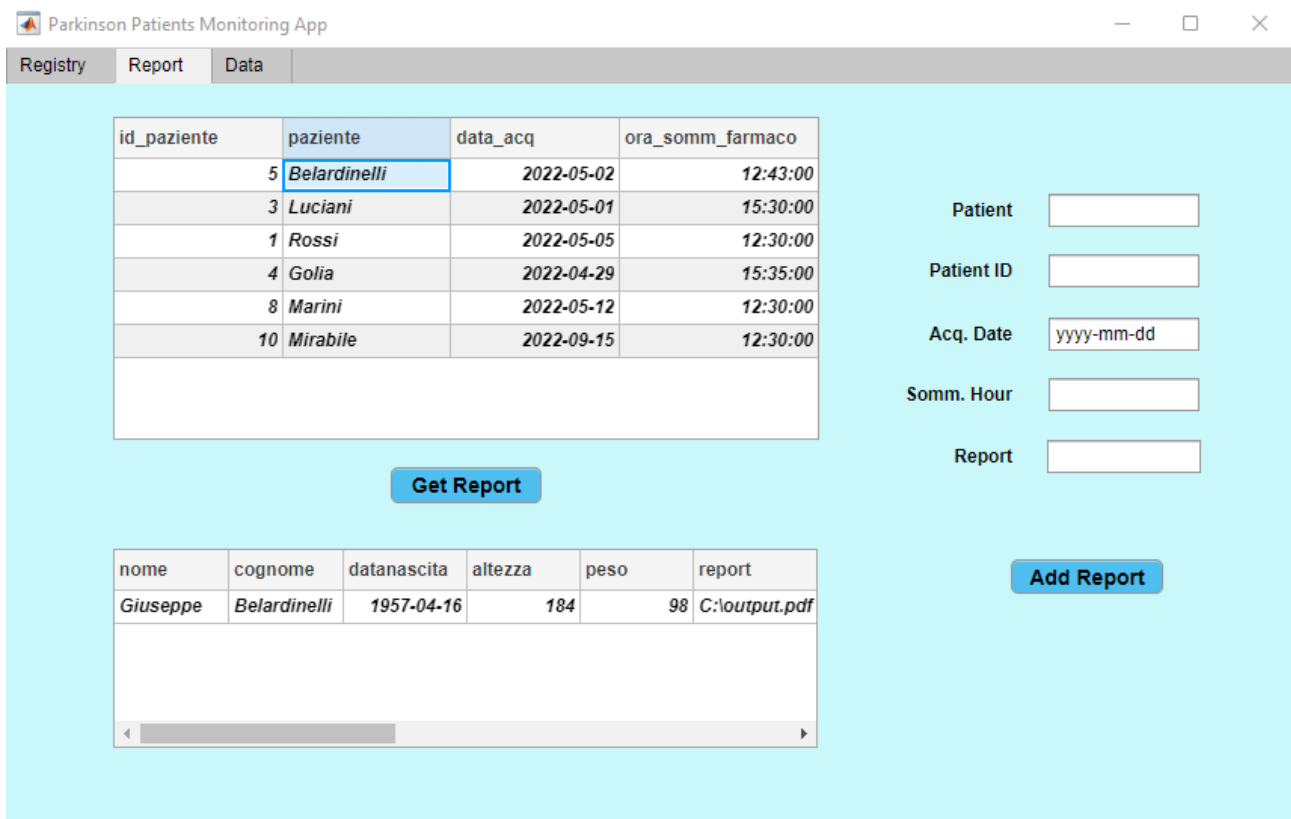


Figura 5.8 Seconda schermata dell'applicazione, utile alla gestione dei report di ogni paziente.

paziente all'interno della tabella in basso presente nella schermata. Questa operazione viene espletata dalla query "SELECT (...) FROM database.tabella a JOIN database.tabella r ON a.id = r.id WHERE a.cognome = cella selezionata". Da questa seconda tabella, poi, grazie alla seconda callback "UITable2CellSelection" a seguito di un singolo click sul percorso file del report, viene aperto il pdf contenente il report del paziente. In questo caso si usa la funzione Matlab winopen() che consente di aprire un file a partire dalla sua destinazione.

5.3.5 Schermata Dati

All'interno della terza ed ultima schermata (visibile in Figura 5.9) avviene la comunicazione con InfluxDB, database per serie temporali all'interno del quale vogliamo che vengano memorizzati i dati grezzi provenienti dalle acquisizioni effettuate con i sensori. Per poter rendere necessaria questa operazione è stata definita come proprietà dell'applicazione "influxdb", attraverso la funzione influxConn() che ci permette di andare ad instaurare una comunicazione tra Matlab e l'organizzazione di InfluxDB. In ingresso a tale funzione è necessario fornire l'URL su cui lavora InfluxDB (nel caso in cui si stia lavorando nella versione non cloud esso corrisponderà a localhost://8086), il TOKEN della propria organizzazione, il nome della propria organizzazione e il nome del database (ovvero il bucket) su cui si vuole lavorare.

Anche in questa schermata si hanno due possibilità: si può lavorare in Upload oppure in Download. Nel caso in cui fosse necessario lavorare in Upload, l'unica operazione possibile è la pressione del pulsante, a cui corrisponde la callback "UploadButton_3Pushed". È richiesto all'utente di inserire il nome del measurement all'interno del quale verranno salvati i dati (si consiglia di utilizzare il formato Cognome_sensore_data), la data relativa all'acquisizione e il percorso file dove si trova il file CSV da caricare all'interno del database. Alla pressione del pulsante, dunque, viene eseguita la funzione influxUpload() che sfrutta il codice reso disponibile su GitHub e in grado di far interagire Matlab ed InfluxDB. Tale funzione richiede in ingresso la connessione al database, il percorso file, il nome del measurement e la data. Essa si occupa di elaborare opportunamente i dati contenuti nel file CSV per ottenere una timetable che rispetti i requisiti richiesti da InfluxDB. L'operazione di upload vera e propria viene eseguita attraverso l'uso delle funzioni messe a disposizione dal codice GitHub.

Per effettuare il Download dei dati già presenti in InfluxDB, anche in questo caso si gestisce il tutto attraverso la callback che prende il nome di "DownloadButtonPushed", ma qui è presente un gruppo di bottoni selezionabili che ci permette di scegliere quale variabile vogliamo estrarre dal

measurement, dovendo necessariamente indicare anche la data di acquisizione a cui tale parametro si riferisce. All'interno della funzione, dunque, vengono usati degli if per verificare quale variabile è stata selezionata e, per ognuna, si esegue la rispettiva funzione di download che prende il nome di getAccelerometer, getBarometer, getMagnetometer o getGyroscope. Ognuna di queste funzioni esegue una query di selezione, sempre sfruttando il codice scaricato da GitHub, per ogni coordinata di quella variabile e costruisce infine una tabella di quattro colonne avente le coordinate (x,y,z) e l'informazione temporale. Se il download va a buon fine, viene data la possibilità all'utente di salvare i dati ottenuti all'interno del workspace di Matlab.

Al termine dell'estrazione di tutte le variabili possibili, premendo il pulsante Table verrà eseguita la funzione che prende il nome di makeTable(), la quale realizza una tabella completa mettendo insieme tutti i parametri estratti e la colonna relativa all'identità temporale dei dati.

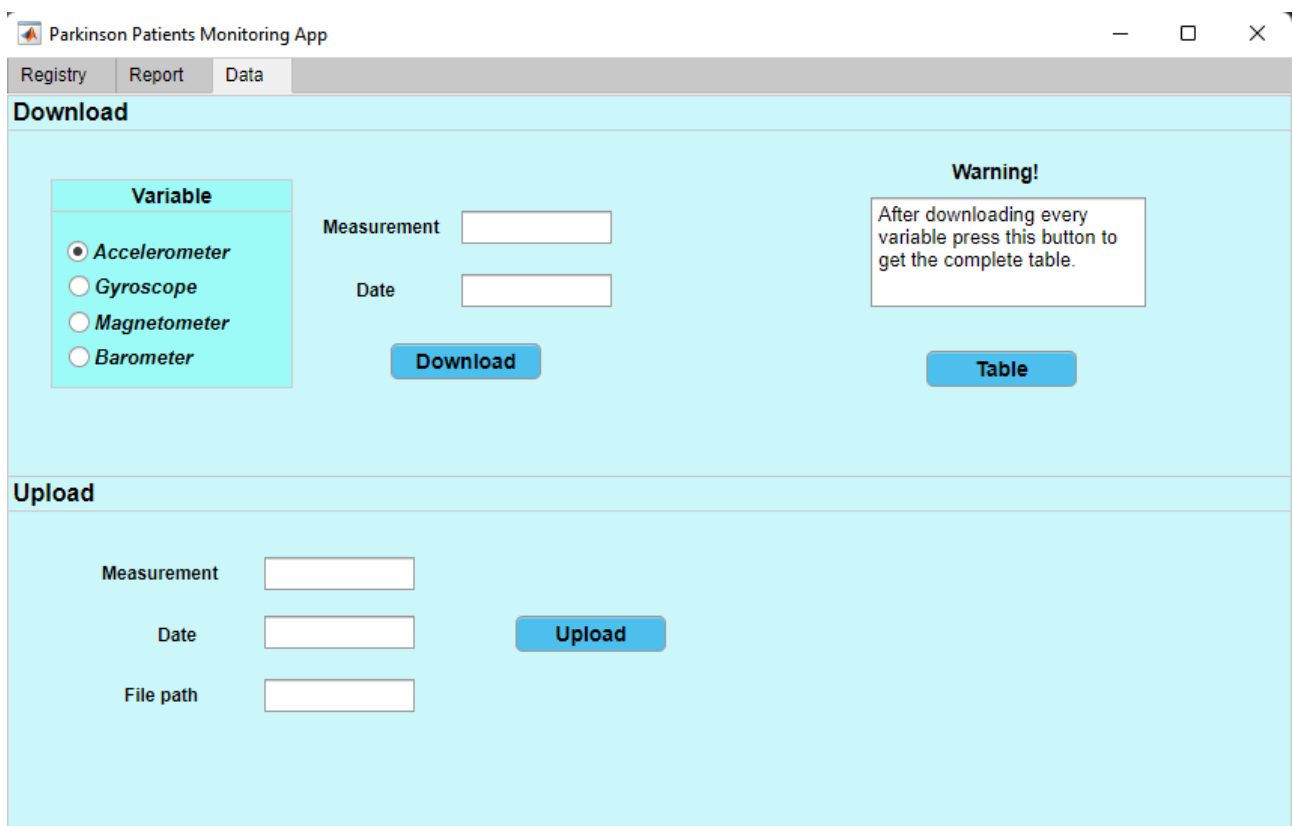


Figura 5.9 Terza schermata dell'applicazione per la gestione dei dati grezzi da inserire e prelevare dal database di InfluxDB.

Conclusioni

In conclusione, l'applicazione sviluppata può risultare un elemento fondamentale nell'ambito della telemedicina, in particolare in situazioni in cui lo staff medico si ritrovi a dover memorizzare e lavorare con una grande mole di dati che descrive lo stato di salute dei pazienti. Il risultato finale è stato ottenuto con un'elaborazione che prevede l'utilizzo delle architetture presentate nel corso della trattazione. In particolare, Matlab è stato fondamentale per creare l'interfaccia grafica in grado di realizzare una comunicazione tra il database relazionale e il database per serie temporali.

Oltre all'interazione uomo-macchina è essenziale la presenza di dispositivi dell'Internet of Medical Things, i quali vengono utilizzati per prelevare dati dalla categoria di pazienti alla quale si sta facendo riferimento. In questa tesi il prodotto è stato testato ed utilizzato nell'ambito della malattia di Parkinson e, in particolare, le misurazioni sono state effettuate tramite sensori indossabili NGIMO programmati per essere posti su entrambi i piedi e sulla vita del soggetto in esame.

Per quanto concerne gli sviluppi futuri, si potrebbe implementare un algoritmo in grado di sfruttare il modulo wi-fi presente nei sensori IoT. Questo può essere utilizzato per far sì che i dati prelevati durante le misurazioni vengano inviati direttamente al database per serie temporali anziché passare attraverso la memorizzazione su scheda SD. Tale operazione comporterebbe una notevole riduzione dei tempi necessari alla memorizzazione dei file che, dopo essere stati estratti dal sensore, tipicamente devono anche essere convertiti dal formato XIO al formato CSV. Inoltre, grazie a questo "upgrade", verrebbe ridotta la mole di dati che passa attraverso l'applicazione sviluppata, evitando al dottore di effettuare un upload dei singoli file in autonomia all'interno di InfluxDB.

Bibliografia

- [1] Balaji, S., Karan Nathani, and R. Santhakumar. "IoT technology, applications and challenges: a contemporary survey." *Wireless personal communications* 108.1 (2019): 363-388.
- [2] Nambiar, Arun N. "RFID technology: A review of its applications." *Proceedings of the world congress on engineering and computer science*. Vol. 2. Hong Kong, China: International Association of Engineers, 2009.
- [3] Want, Roy. "An introduction to RFID technology." *IEEE pervasive computing* 5.1 (2006): 25-33.
- [4] Buratti, Chiara, et al. "An overview on wireless sensor networks technology and evolution." *Sensors* 9.9 (2009): 6869-6896.
- [5] Pasluosta, Cristian F., et al. "An emerging era in the management of Parkinson's disease: wearable technologies and the internet of things." *IEEE journal of biomedical and health informatics* 19.6 (2015): 1873-1881.
- [6] Romero, Luis Eduardo, Parag Chatterjee, and Ricardo Luis Armentano. "An IoT approach for integration of computational intelligence and wearable sensors for Parkinson's disease diagnosis and monitoring." *Health and Technology* 6.3 (2016): 167-172.
- [7] Beitz, Janice M. "Parkinson's disease: a review." *Frontiers in Bioscience-Scholar* 6.1 (2014): 65-74.
- [8] Hayes, Michael T. "Parkinson's disease and parkinsonism." *The American journal of medicine* 132.7 (2019): 802-807.
- [9] Debû, Bettina, et al. "Managing gait, balance, and posture in Parkinson's disease." *Current neurology and neuroscience reports* 18.5 (2018): 1-12.
- [10] "Parkinson's Disease", Dr Prem Pillay. <https://singaporebrain.org/brain/parkinsons-disease/> (Visitato il 04/08/2022).
- [11] "Treatment and Therapies", Parkinson's NSW. <https://www.parkinsonsnsw.org.au/treatment-and-therapies/> (Visitato il 31/07/2022).
- [12] Rucco, Rosaria, et al. "Type and location of wearable sensors for monitoring falls during static and dynamic tasks in healthy elderly: a review." *Sensors* 18.5 (2018): 1613.
- [13] "Causes and risk factors that influence Parkinson's Disease", EliteAyurveda ContestSeo. <https://eliteayurveda.com/blog/causes-and-risk-factors-that-influence-parkinsons-disease/> (Visitato il 04/08/2022).
- [14] Sveinbjornsdottir, Sigurlaug. "The clinical symptoms of Parkinson's disease." *Journal of neurochemistry* 139 (2016): 318-324.

- [15] "Symptoms and Complications", *Parkinson's NSW*.
<https://www.parkinsonsnsw.org.au/symptoms-and-complications/#1634016222546-9188324c-4b6c> (Visitato il 31/07/2022).
- [16] Opara, Józef, et al. "Motor assessment in Parkinsons disease." *Ann Agric Environ Med* 24.3 (2017): 411-415.
- [17] Ramaker, Claudia, et al. "Systematic evaluation of rating scales for impairment and disability in Parkinson's disease." *Movement disorders: official journal of the Movement Disorder Society* 17.5 (2002): 867-876.
- [18] Goetz, Christopher G., et al. "Movement Disorder Society-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): scale presentation and clinimetric testing results." *Movement disorders: official journal of the Movement Disorder Society* 23.15 (2008): 2129-2170.
- [19] "Activities-specific Balance Confidence Scale (ABC Scale)", *Yumpu*.
<https://www.yumpu.com/en/document/read/11600003/activities-specific-balance-confidence-abc-scale> (Visitato il 01/08/2022).
- [20] Bloem, Bastiaan R., et al. "Measurement instruments to assess posture, gait, and balance in Parkinson's disease: Critique and recommendations." *Movement Disorders* 31.9 (2016): 1342-1355.
- [21] "Activities-specific Balance Confidence Scale (ABC Scale)", *Stokengine*.
<https://stokengine.ca/en/assessments/activities-specific-balance-confidence-scale-abc-scale/> (Visitato il 01/08/2022).
- [22] Pierleoni, Paola, et al. "A high reliability wearable device for elderly fall detection." *IEEE Sensors Journal* 15.8 (2015): 4544-4553.
- [23] Ghislieri, Marco, et al. "Wearable inertial sensors to assess standing balance: A systematic review." *Sensors* 19.19 (2019): 4075.
- [24] "Database relazionale", *geekandjob*. <https://www.geekandjob.com/wiki/database-relazionale> (Visitato il 10/06/2022).
- [25] "Che cos'è un database relazionale (RDBMS)?", *Oracle*.
<https://www.oracle.com/it/database/what-is-a-relational-database/> (Visitato il 10/06/2022).
- [26] "Qual è la differenza tra Time Series e Relational Database", *Extrared*.
<https://www.extrasys.it/it/redblog/time-series-vs-relational-database> (Visitato il 28/04/2022).

- [27] “I migliori 4 Time Series Database open source del 2020”, *Extrared*.
<https://www.extrasys.it/it/redblog/i-migliori-4-time-series-database-open-source-del-2020>
(Visitato il 29/04/2022).
- [28] Kuc, Rafal, and Marek Rogozinski. *Elasticsearch server*. Packt Publishing Ltd, 2013.
- [29] “Introduzione a Lucene: Cos’è e come funziona”, *Bucap*.
[https://www.bucap.it/news/approfondimenti-tematici/protocollo_informatico/lucene-cos-e-
come-funziona.htm](https://www.bucap.it/news/approfondimenti-tematici/protocollo_informatico/lucene-cos-e-come-funziona.htm) (Visitato il 05/05/2022).
- [30] “Cos’è Elasticsearch”, *Massimo Ivaldi*. <https://massimoivaldi.com/cose-elasticsearch/> (Visitato il 05/05/2022).
- [31] “Elasticsearch Vs InfluxDB for Time-Series Data”, *Entrance*.
<https://www.entranceconsulting.com/elasticsearch-vs-influxdb-for-time-series-data/> (Visitato il 09/05/2022).
- [32] “Xio Technologies”, *x-io*. <https://x-io.co.uk/ngimu/> (Visitato il 02/08/2022).
- [33] “NGIMU”, *Ignitec*. <https://www.ignitec.com/work/ngimu/> (Visitato il 02/08/2022).
- [34] “Cosa è MySQL? Una spiegazione per principianti”, *Kinsta*.
<https://kinsta.com/it/knowledgebase/cosa-e-mysql/#client-server-model> (Visitato il 10/06/2022).
- [35] “What is MySQL? Everything you need to know”, *Talend*.
[https://www.talend.com/resources/what-is-
mysql/#:~:text=MySQL%20is%20a%20relational%20database,information%20in%20a%20corporate%20network](https://www.talend.com/resources/what-is-mysql/#:~:text=MySQL%20is%20a%20relational%20database,information%20in%20a%20corporate%20network) (Visitato il 14/06/2022).
- [36] “Come usare MySQL Workbench”, *SiteGround*. [https://it.siteground.com/tutorial/php-
mysql/mysql-workbench/](https://it.siteground.com/tutorial/php-mysql/mysql-workbench/) (Visitato il 14/06/2022).
- [37] “InfluxDB: spiegazione, vantaggi e primi passi”, *Ionos*.
<https://www.ionos.it/digitalguide/hosting/tecniche-hosting/cose-influxdb/> (Visitato il 30/06/2022).
- [38] “Introduction to InfluxDB: a time-series database”, *We are community*.
<https://wearecommunity.io/communities/india-java-user-group/articles/891> (Visitato il 30/06/2022).
- [39] Naqvi, Syeda Noor Zehra, Sofia Yfantidou, and Esteban Zimányi. "Time series databases and influxdb." *Studienarbeit, Université Libre de Bruxelles* 12 (2017).
- [40] “Manage buckets”, *influxdata*.
<https://docs.influxdata.com/influxdb/v2.2/organizations/buckets/> (Visitato il 04/07/2022).

- [41] “Manage organizations”, *influxdata*. <https://docs.influxdata.com/influxdb/v2.2/organizations/> (Visitato il 04/07/2022).
- [42] “Database Toolbox”, *MathWorks*. <https://it.mathworks.com/products/database.html> (Visitato il 02/08/2022).
- [43] “MATLAB App Designer”, *MathWorks*. <https://it.mathworks.com/products/matlab/app-designer.html> (Visitato il 03/08/2022).
- [44] “InfluxDB Matlab”, *wardgessens*. <https://github.com/wardgessens/influxdb-matlab/tree/v2-compatibility> (Visitato il 03/08/2022).
- [45] “InfluxDB Downloads”, *influxdata*. <https://portal.influxdata.com/downloads/> (Visitato il 20/08/2022).
- [46] “MySQL Community Downloads”, *MySQL*. <https://dev.mysql.com/downloads/connector/odbc/>. (Visitato il 21/08/2022).