



UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea triennale di Ingegneria Biomedica

**Sviluppo di un sistema di raccolta ed analisi dati per
sensori indossabili**

**Development of a data collection and analysis system for
wearable sensors**

Relatore:

Prof. Lorenzo Scalise

Tesi di Laurea di:

Giulia Passacantando

Correlatrice:

Dott.ssa Ing. Nicole Morresi

A.A. 2021/2022

Sommario

1. INTRODUZIONE	3
2. STATO DELL'ARTE	4
2.1 Dashboard per il monitoraggio.....	4
2.1.1 Trend e layout.....	5
2.2 Dispositivi Indossabili.....	8
2.3 Importanza della misura dei parametri vitali tramite smartwatch.....	10
3. MATERIALI E METODI.....	11
3.1 Garmin Venu Sq.....	11
3.1.1 Parametri misurati.....	12
3.1.2 Sensori installati.....	13
3.2 Creazione Dashboard	14
3.2.1 Garmin Connect.....	15
3.2.2 Fitrockr	17
3.2.3 Analisi dei dati e implementazione della dashboard	21
4. RISULTATI	30
4.1 Visualizzazione dei risultati	30
5. CONCLUSIONI	34
6. BIBLIOGRAFIA E SITOGRAFIA.....	35

1. INTRODUZIONE

Lo scopo di questa tesi di laurea è quello di definire e sviluppare una dashboard che possa essere di sostegno per i caregiver, al monitoraggio e diagnosi di pazienti da remoto, tramite la registrazione continua di parametri fisiologici effettuata dallo smartwatch Garmin Venu Sq.

La necessità di utilizzare questo dispositivo indossabile nasce dal bisogno di monitorare la salute in modo intuitivo, semplice e soprattutto autonomo. Infatti, monitorando parametri base come la frequenza cardiaca, la saturazione del sangue, la frequenza respiratoria o la temperatura è possibile capire se l'utente monitorato sta passando da una condizione fisiologica ad una patologica.

Questo lavoro di tesi è iniziato con un'analisi della letteratura riguardante gli smartwatch, in particolare il Garmin Venu Sq, le loro caratteristiche, il loro funzionamento e il loro utilizzo in diversi ambiti, sia lo sviluppo di dashboard per la salute, quindi come vengono definite, i trend che vanno a rappresentare e con quale scopo vengono utilizzate nell'ambito medico.

Da questa analisi iniziale è stato possibile comprendere che i dati disponibili registrati dal Garmin erano quelli riguardanti la frequenza cardiaca, la variabilità di quest'ultima, la frequenza respiratoria, la saturazione, lo stress ed in base a questi è stata implementata una dashboard.

Inizialmente si utilizza lo smartwatch per la registrazione dei dati, salvati automaticamente nel sito Garmin Connect. Attraverso Fitrockr, un'implementazione tecnologica integrata dell'API e dell'SDK di Garmin Health, è stato possibile scaricare tutti i dati di interesse.

Successivamente si è passato allo sviluppo software della dashboard che ha permesso di processare questi ultimi e creare poi una dashboard tramite Dash, un micro-framework Python.

Come risultato finale si ottiene una dashboard che rappresenta l'andamento, i valori di massimo, minimo e medio, la distribuzione, la correlazione dei dati di chi indossa l'orologio, utili per visualizzare lo stato di salute del paziente in un determinato periodo di tempo che si può estendere da una singola giornata all'intero periodo di registrazione.

2. STATO DELL'ARTE

2.1 Dashboard per il monitoraggio

Una dashboard è un record interattivo che visualizza, sottoforma di tabelle o grafici, e analizza dei dati in tempo reale al fine di permettere un monitoraggio migliore di questi ultimi [1]. In base ai dati presi in considerazione, si possono definire diverse tipologie di dashboard, da quelle aziendali a quelle cliniche.

Argomento di questa tesi sono le dashboard utilizzate per il monitoraggio di dati riguardanti l'ambito clinico.

In ambiente sanitario le dashboard possono essere utilizzate per il monitoraggio da remoto dei pazienti al fine di ottenere una visione globale della salute di questi ultimi in tempo reale, evitando così un sovraccollamento causato da un numero elevato di visite e quindi rendendo più efficace ed efficiente il sistema sanitario nazionale [2].

Il controllo dei parametri vitali, come temperatura corporea, frequenza cardiaca e frequenza respiratoria, infatti, determina le condizioni di salute dei pazienti e, attuando un monitoraggio da remoto di questi, è possibile prevenire, gestire e trattare la salute dei pazienti, che sono ricoverati o che si trovano nel loro domicilio, in modo più tempestivo [3].

Inoltre un esempio importante di come l'utilizzo delle dashboard si è rivelato efficiente ed efficace riguarda il monitoraggio di pazienti in auto-isolamento da COVID-19; ha permesso infatti da una parte di sorvegliare i parametri vitali al fine di poter intervenire in modo rapido nel paziente ed evitare il peggioramento dei sintomi [4], dall'altra di registrare i dati da ogni paziente per studiare l'evoluzione della malattia, conteggiare i nuovi casi, il tasso di peggioramento e di mortalità [5].

2.1.1 Trend e layout

Come detto in precedenza, esaminare i parametri clinici permette di definire la salute di un individuo.

Ovviamente i parametri visualizzati devono essere tali da poter essere misurati senza il bisogno di un operatore sanitario, quindi, in una dashboard, si potranno osservare dati come frequenza cardiaca, frequenza respiratoria, saturazione, temperatura corporea, tutte variabili fisiologiche che si possono ottenere mediante l'utilizzo di dispositivi indossabili.

Di rilevante importanza è anche come questi dati vengono visualizzati.

L'interfaccia utente varia in base alla destinazione d'uso della dashboard stessa; nel caso in cui deve essere utilizzata come auto-monitoraggio sarà molto simile alla **Figura 1**. In questo esempio, i dati sanitari visualizzati, provenienti da più fonti, sono integrati nell'EHDViz (Electronic Health Data Visualization), cioè il software di cartelle cliniche elettroniche, di un solo paziente. È presente un'interfaccia per la gestione dell'utente (1), in cui è possibile selezionare i dati da visualizzare, ottenuti da visite ambulatoriali (2). Dati che provengono da misurazioni attuate da un dispositivo indossabile (in questo caso Fitbit) (3) e da misurazioni effettuate manualmente dall'utente, come peso e pressione sanguigna, immesse nel sistema, effettuate autonomamente dal paziente stesso (4).

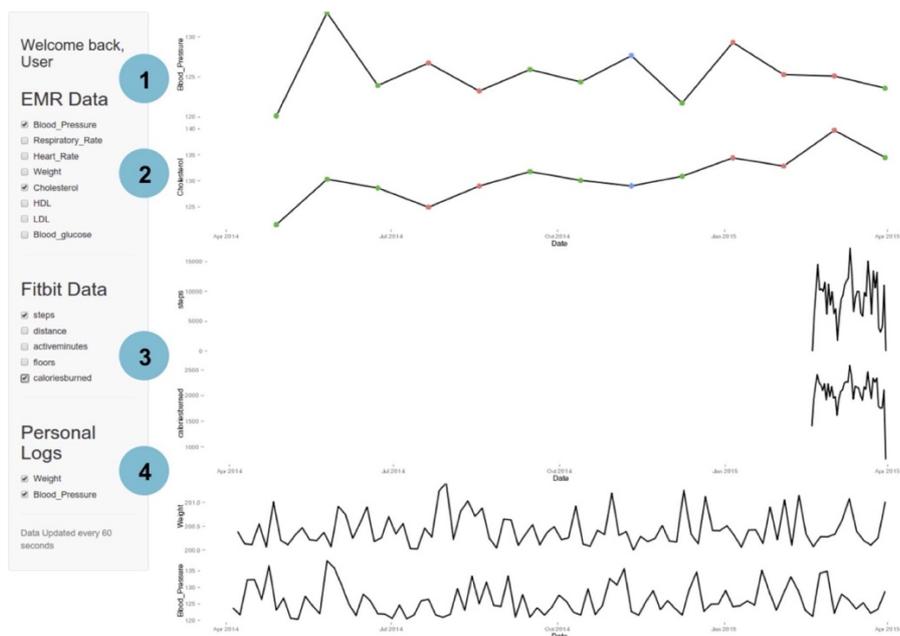


Figura 1. Dashboard clinica per monitoraggio di un singolo paziente

Rispetto al precedente esempio, si ottengono dati più dinamici se si considerano quelli contenuti in una nelle cartelle cliniche elettroniche di pazienti durante una degenza ospedaliera. In questo caso cambiano anche i parametri visualizzati (**Figura 2**). Si può notare infatti come è presente una barra laterale, in cui è possibile selezionare tramite configurazione drop-down il paziente e l'intervallo di durata del ricovero (**A-B**) o il reparto di degenza (**C-D**), e un grafico che rappresenta la scala MEWS (Modified Early Warning Score), che indica il grado di instabilità clinica, in funzione dei giorni di degenza, per uno o più pazienti.

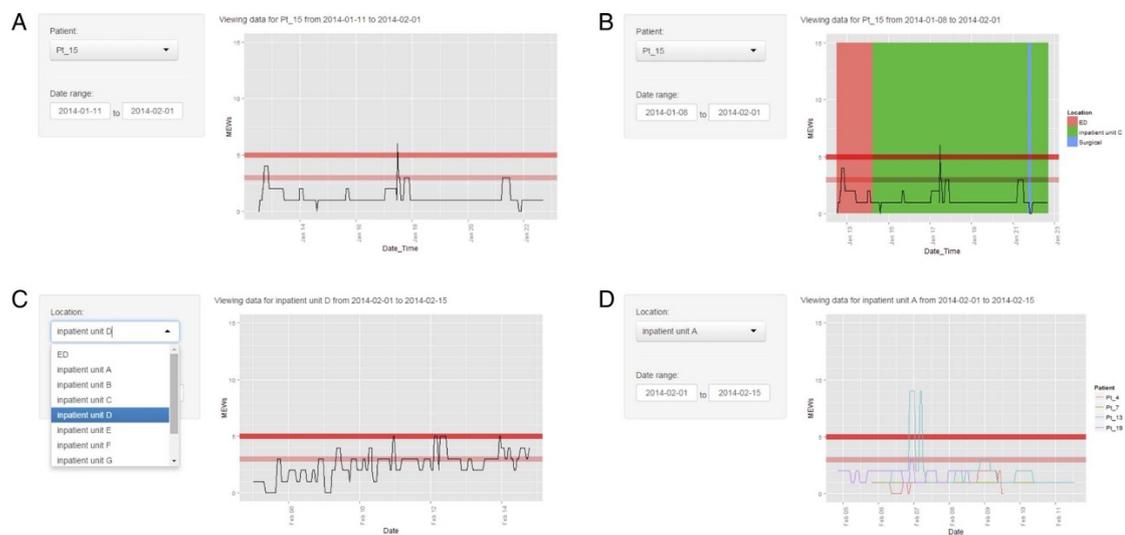


Figura 2. Esempi di dashboard per monitoraggio in ambiente ospedaliero

Attraverso l'interfaccia grafica di queste dashboard, il personale sanitario è in grado di comprendere e studiare le condizioni di un singolo paziente durante il periodo di degenza (**Figura 2-A**), evidenziando lo stato di salute in diversi reparti ospedalieri, come pronto soccorso, sala operatoria (**Figura 2-B**), oppure di visualizzare le tendenze della scala MEWS in diversi reparti di degenza (**Figura 2-C**) e per diversi pazienti in uno stesso reparto (**Figura 2-D**).

Se si considera in particolare un unico reparto, selezionato direttamente dall'utente, si può prendere come riferimento la dashboard rappresentata nella **Figura 3**. In questo caso la parte in alto a sinistra è costituita da un riquadro in cui l'operatore sceglie il reparto da studiare (1), il modo in cui è configurato il layout (in una o più colonne) (2) e i dati che vengono visualizzati (specifici enzimi) (3). Questi ultimi sono rappresentati mediante grafici in funzione del tempo e sono riferiti a più pazienti (3 nel caso della Figura 3).

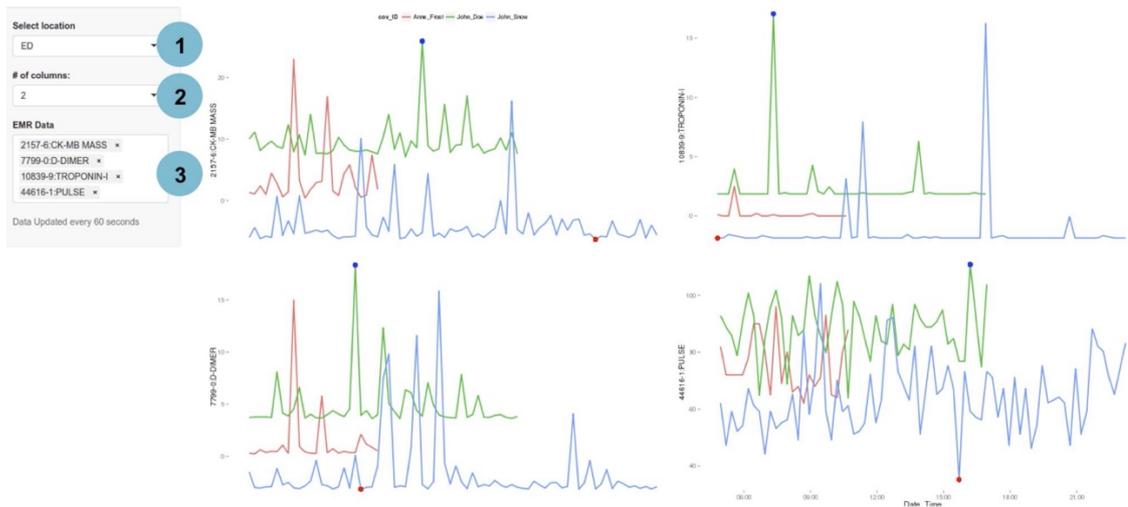


Figura 3. Dashboard per il monitoraggio ospedaliero di più pazienti

Quando invece il personale sanitario preferisce porre l'attenzione sul singolo paziente, può far riferimento alla dashboard in **Figura 4**, molto simile alle precedenti, ma con differenti scelte nel menù drop-down presente a destra: l'operatore sanitario può scegliere il nome del paziente (1), la classificazione ICD (International Classification of Diseases) (2), che è la classificazione internazionale delle malattie e, in seguito, visualizzare i parametri correlati mediante dei grafici (3). [6]

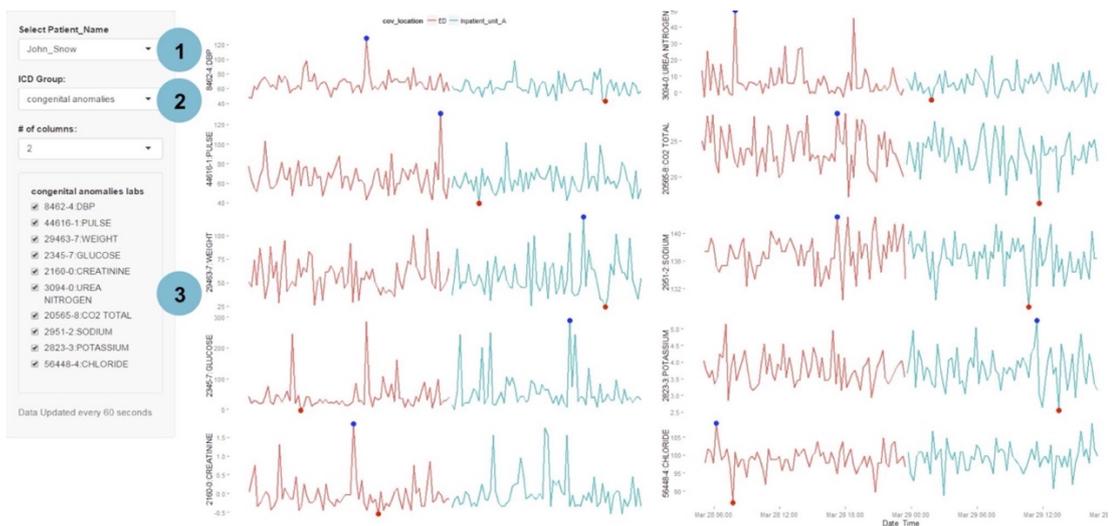


Figura 4. Dashboard relativa al monitoraggio di un singolo paziente

Si può notare che come minimo comun denominatore di ogni dashboard ci sia la semplicità con cui i dati vengono rappresentati, tramite grafici, evidenziando con

diversi colori i diversi pazienti o le diverse condizioni di valutazione dei dati stessi. Questo perché il fine ultimo è quello di permettere al personale medico di poter controllare e valutare in modo veloce le condizioni cliniche di uno o più pazienti.

2.2 Dispositivi Indossabili

La maggior parte dei dati clinici visualizzati in precedenza viene misurata con dispositivi indossabili. L'utilizzo di questi ultimi ne permette un'efficiente raccolta perché, prima di tutto, si esclude la presenza di operatori sanitari che misurano manualmente i parametri vitali, dal momento che, attraverso i sensori installati, lo smartwatch è in grado di raccogliere i dati automaticamente; inoltre, si aumenta il numero di pazienti coinvolti nella misurazione, utilizzando più dispositivi alla volta. Questo, oltre a permettere un controllo migliore e più veloce su un maggiore gruppo di pazienti, rende più precisi gli studi clinici, perché attuati su più campioni.

In uno studio pubblicato nel "Journal of Ideas in Health" nel 2021 [7], sono presi in considerazione vari dispositivi indossabili per monitorare l'evoluzione del COVID-19, misurando parametri vitali come frequenza cardiaca (HR), variazione della frequenza cardiaca (HRV), saturazione di ossigeno (SpO2) e frequenza respiratoria (RR).

Questi dati non vengono però misurati uno alla volta attraverso i metodi tradizionali, come l'utilizzo di un sensore applicato sul petto per trovare la frequenza respiratoria (**Figura 5-1**), dell'elettrocardiografo per valutare sia frequenza cardiaca, che la sua variazione (**Figura 5-2**), di sensori infrarossi per misurare la temperatura corporea (**Figura 5-3**) o pulsossimetri per ottenere la saturazione d'ossigeno (**Figura 5-4**), ma si utilizza unicamente lo smartwatch che colleziona tutti questi parametri simultaneamente. Tra i differenti tipi di dispositivo considerati nel precedente studio, è presente il Garmin Venu Sq, di interesse per questo specifico lavoro di tesi.

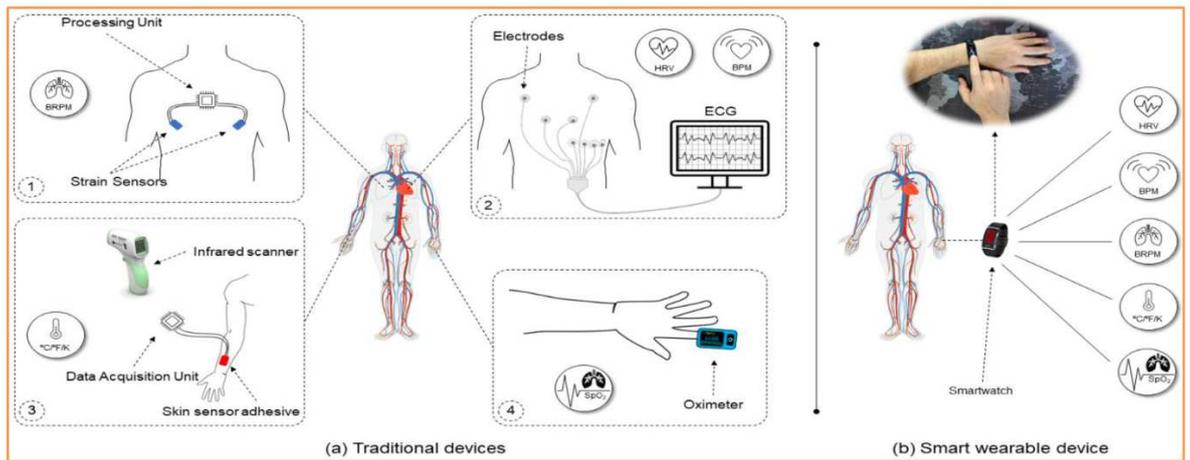


Figura 5. Differenza di acquisizione dei dati da dispositivi tradizionali e dispositivi indossabili

In letteratura sono presenti vari studi che utilizzano il Garmin per diversi scopi: per esempio è stato utilizzato per estrarre valori di HR a riposo, variabilità della frequenza cardiaca, frequenza respiratoria, saturazione di ossigeno nel sangue e qualità del sonno nelle donne incinta, indispensabili per una diagnosi precoce di condizioni correlate alla gravidanza come ipertensione gestazionale (aumento transitorio dei valori di pressione arteriosa sopra i 140/90 mmHg) e preeclampsia, ovvero la condizione di ipertensione spesso in combinazione con il riscontro di una quantità significativa di proteine nelle urine [8].

I dati estratti dall'orologio possono essere utilizzati come supporto per studi di carattere psicologico-cognitivo. Infatti, viene misurata la durata del sonno, al fine di effettuare uno studio relativo alla qualità e durata di questo correlata all'autostima dei giovani islandesi [9].

L'acquisizione dei valori di frequenza cardiaca risulta invece di supporto a segnali neurali ed espressioni facciali per la valutazione dello stato cognitivo degli studenti durante l'orario di lezione [10].

Da un'analisi sulla letteratura si può notare come lo smartwatch in questione viene utilizzato per differenti studi che però hanno come punto di convergenza il tipo dei dati raccolti dall'orologio stesso.

2.3 Importanza della misura dei parametri vitali tramite smartwatch

Monitorare i dati fisiologici per uno o più pazienti è importante per diversi motivi. Innanzitutto, il fatto che a monitorarli sia un dispositivo indossabile rende più semplice e veloce l'acquisizione dei valori, perché non è necessario né l'ausilio di macchinari biomedicali, né la presenza di personale specifico, dal momento in cui è tutto svolto autonomamente dallo smartwatch, inoltre si ottiene una collezione di più valori, ottenuti da diversi sensori, in maniera simultanea.

Effetto di ciò è una migliore efficienza nella gestione dei dati biomedici e dei pazienti, visto che il medico è in grado di accedere in qualsiasi momento ad una visione globale della salute di uno o più pazienti e stabilire poi a posteriori se ci sono o no condizioni patologiche e come intervenire.

Strutturando dashboard per monitoraggio comprensibili visivamente anche a persone che non hanno conoscenze in ambito medico, è possibile che i parametri misurati dai dispositivi possono essere controllati anche direttamente da chi indossa l'orologio.

Questo è un altro aspetto che va a semplificare ancora di più la gestione sanitaria, perché l'auto-monitoraggio permette al singolo individuo di controllare se è in salute e se ci sono anomalie riguardanti i parametri fisiologici, le quali, se presenti, potrebbero essere la causa di altre patologie.

Oltre ad avere un impatto sull'aspetto decisionale e organizzativo a livello ambulatoriale o ospedaliero, la quantità di dati raccolti dai vari dispositivi indossabili permette di fare un'analisi cumulativa sulla salute di più pazienti o sull'andamento di una malattia, per esempio il COVID-19.

Il monitoraggio tramite smartwatch è fondamentale se si considera che qualsiasi persona in qualunque momento, è in grado di controllare i propri parametri vitali ed è in grado di comprendere se questi sono o no nella norma. Quindi fornisce un supporto tempestivo, di supporto per la salute dell'individuo.

3. MATERIALI E METODI

3.1 Garmin Venu Sq

Il Garmin Venu Sq (**Figura 6**) [11] è un dispositivo smart indossabile progettato specificamente per lo sport e per il monitoraggio giornaliero dei parametri vitali, acquistabile direttamente dal sito Garmin a partire da 199,99 € [12] o negli specifici negozi. Le sue caratteristiche generali e le funzioni, elencate rispettivamente nella **Figura 7** e **Figura 8**, lo rendono adatto per un utilizzo quotidiano.



Figura 6. Garmin Venu Sq

MATERIALE DELLALENTE	Corning® Gorilla® Glass 3
MATERIALE DELLA GHIERA	Alluminio anodizzato
MATERIALE DEL RIVESTIMENTO	Polimeri fibrorinforzati
CINTURINI A RILASCIO RAPIDO	Sì (20 mm, standard del settore)
MATERIALE DEL CINTURINO	Silicone
DIMENSIONI DELLA CASSA, LXLXA	40,6 x 37,0 x 11,5 mm Adatto a polsi con una circonferenza di 125 - 190 mm
TOUCHSCREEN	✓
SCHERMO A COLORI	✓
DIMENSIONI SCHERMO, LXA	1,3" (33,1 mm) diagonale
RISOLUZIONE SCHERMO, LXA	240 x 240 pixel
TIPO DI SCHERMO	Cristalli liquidi Modalità sempre attiva opzionale
PESO	37,6 g
CLASSIFICAZIONE DI IMPERMEABILITÀ	Nuoto, 5 ATM
DURATA DELLA BATTERIA	Modalità smartwatch: fino a 6 giorni Modalità GPS: fino a 14 ore
CHARGING METHOD	Garmin proprietary plug charger
MEMORIA/CRONOLOGIA	200 ore di dati sulle attività

Figura 7. Caratteristiche generali Garmin Venu Sq

ORA/DATA	✓
SINCRONIZZAZIONE DELL'ORA TRAMITE GPS	✓
ORA LEGALE AUTOMATICA	✓
SVEGLIA	✓
TIMER	✓
CRONOMETRO	✓
ORARI DI ALBA/TRAMONTO	✓

Figura 8. Funzioni Garmin Venu Sq

3.1.1 Parametri misurati

I parametri che vengono misurati direttamente dal Garmin sono:

- Frequenza cardiaca al polso (costante, ogni secondo)
- Frequenza cardiaca a riposo giornaliera
- Avvisi sulla frequenza cardiaca anomala
- Frequenza respiratoria (24x7)
- Saturazione di ossigeno nel sangue, Pulse Ox
- Monitoraggio dell'energia body battery
- Tracciamento dello stress giornaliero
- Promemoria rilassamento
- Timer promemoria rilassamento
- Sonno
- Idratazione
- Monitoraggio della salute della donna

Tra questi possibili tipi di dato da acquisire sono presenti quelli maggiormente estratti per l'utilizzo di ricerche e studi di carattere clinico visti in precedenza e quelli che sono importanti per stabilire le condizioni di salute di un individuo, come saturazione, frequenza cardiaca e respiratoria.

3.1.2 Sensori installati

Il Garmin Venu Sq possiede diversi sensori, ognuno dei quali è impiegato per estrarre una tipologia di dato differente:

- **GPS (Global Positioning System):** Sistema di Posizionamento Globale che è in grado di determinare con precisione la posizione (latitudine e longitudine) sul pianeta Terra di un ricevitore (in questo caso lo smartwatch) e di fornirla in coordinate geografiche
- **Glonass:** Sistema di Posizionamento Satellitare russo
- **Galileo:** Sistema di Posizionamento e Navigazione Satellitare civile
- **Tecnologia cardio da polso Garmin Elevate:** sensore ottico in grado di misurare la frequenza cardiaca in modo continuo
- **Bussola:** strumento per l'individuazione dei punti cardinali
- **Accelerometro:** strumento che misura l'accelerazione di un oggetto
- **Sensore della luce ambientale:** regola automaticamente la luminosità del pannello LCD dello schermo dell'orologio, in base ai livelli di luce circostante
- **Acclimatazione del pulsossimetro:** strumento biomedicale che misura la saturazione di ossigeno nel sangue

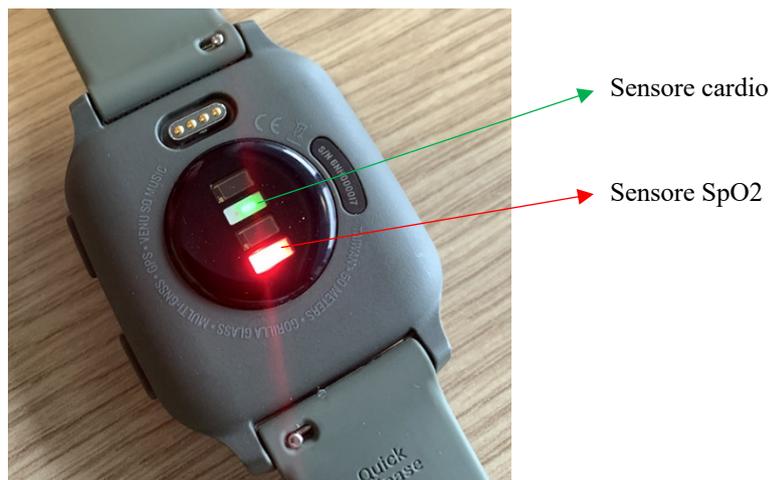


Figura 9. Retro del Garmin Venu Sq

3.2 Creazione Dashboard

Come già affermato, scopo di questa tesi è strutturare una dashboard semplice in grado di monitorare lo stato di salute di un paziente tramite la misurazione di più parametri fisiologici, misurati tramite il Garmin Venu Sq.

Per ottenere i dati utili, lo smartwatch è stato indossato dall'utente per 2 settimane (dal 1° dicembre al 15 dicembre 2022), 24 ore al giorno eccetto i periodi in cui il dispositivo era in carica.

La creazione della dashboard è il risultato di diversi passaggi, raffigurati nello schema a blocchi sottostante (**Figura 10**); tutto parte dalla misurazione continua effettuata dallo smartwatch Garmin e dal successivo salvataggio automatico dei dati all'interno del profilo utente sul sito Garmin Connect.

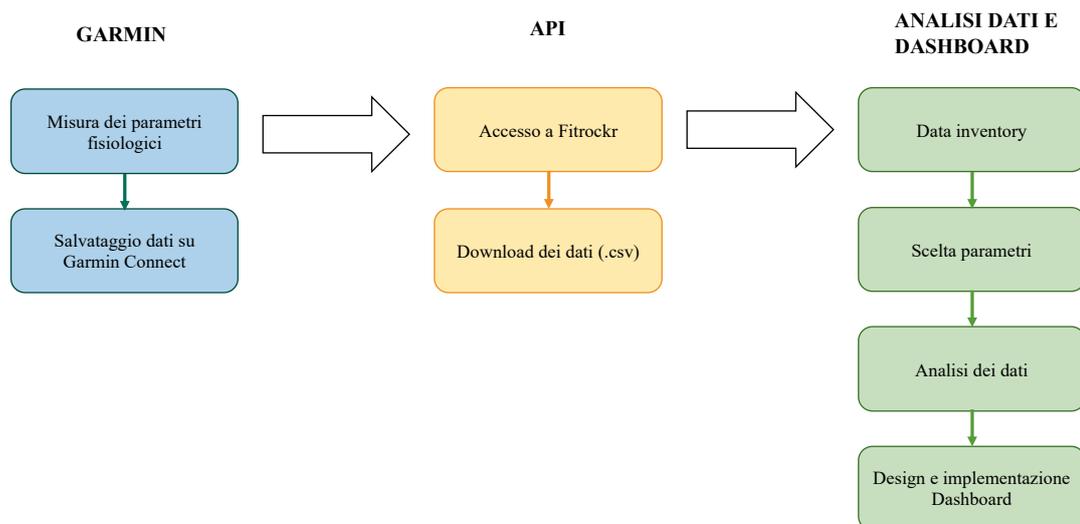


Figura 10. Schema a blocchi per la creazione della dashboard

Per scaricare i dati è necessario l'utilizzo di un'implementazione tecnologica integrata dell'API di Garmin Health: Fitrockr; una volta scaricati, si scelgono quali parametri rappresentare, si analizzano e si passa alla realizzazione della dashboard vera e propria attraverso Python.

3.2.1 Garmin Connect

Garmin Connect è un'applicazione disponibile tramite Web e su smartphone che permette il tracciamento, l'analisi e la condivisione di dati, relativi alla salute o alle attività sportive svolte, registrati da qualsiasi dispositivo Garmin [13].

Per accedervi sono necessari username e password, ottenuti tramite registrazione al sito. Una volta fatto, bisogna sincronizzare il dispositivo indossabile con l'account Garmin affinché si possa accedere alla dashboard relativa a tutti i dati registrati e poi salvati. La dashboard visualizzata è utile per avere un monitoraggio sia giornaliero che periodico delle attività svolte; infatti, è possibile visualizzare parametri e grafici utili relativi alle attività, la cui registrazione è effettuata manualmente dall'utente che indossa lo smartwatch.

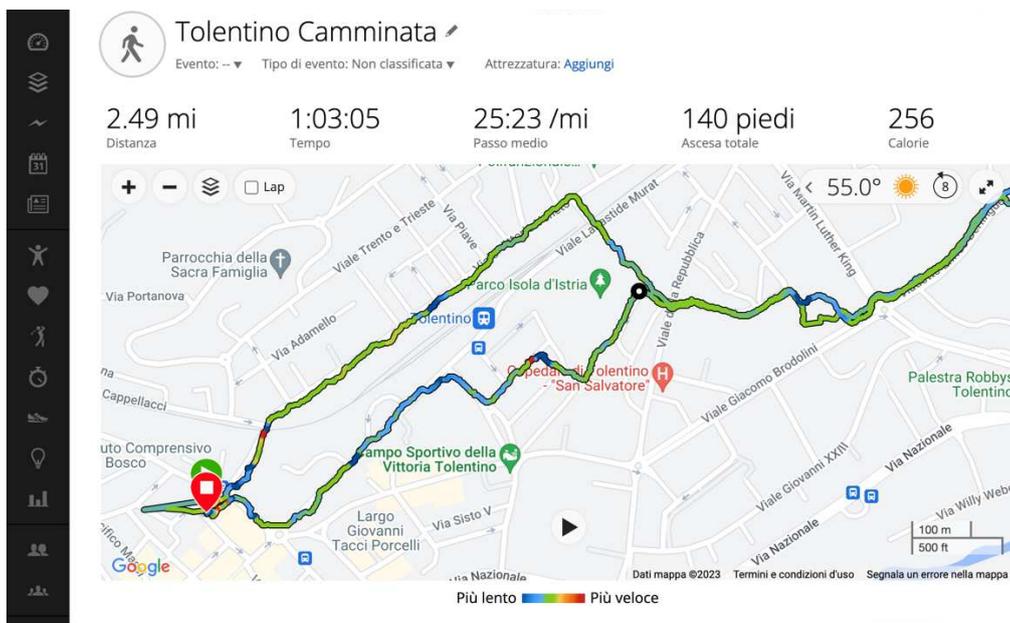


Figura 11-A. Dashboard relativa ad una attività in Garmin Connect

Le **Figure 11** rappresentano come viene visualizzata la dashboard di una attività nel sito Garmin Connect: nella **Figura 11-A** è possibile vedere che l'attività considerata è la camminata, la sua durata e il percorso effettuato, mentre nell'immagine sottostante (**Figura 11-B**) ci sono i grafici relativi a quota, passo, frequenza cardiaca e cadenza, riguardanti tutto il periodo dell'attività.



Figura 11-B. Dashboard relativa ad una attività in Garmin Connect

Nell'ultima immagine (**Figura 11-C**) è possibile notare i valori corrispondenti ai grafici sovrastanti che descrivono, con maggiore dettaglio, l'attività svolta dall'utente, come per esempio le calorie bruciate o attive, la frequenza cardiaca media e massima, informazioni relative alla distanza e al tempo dell'attività, la quota, il passo/velocità e la cadenza.

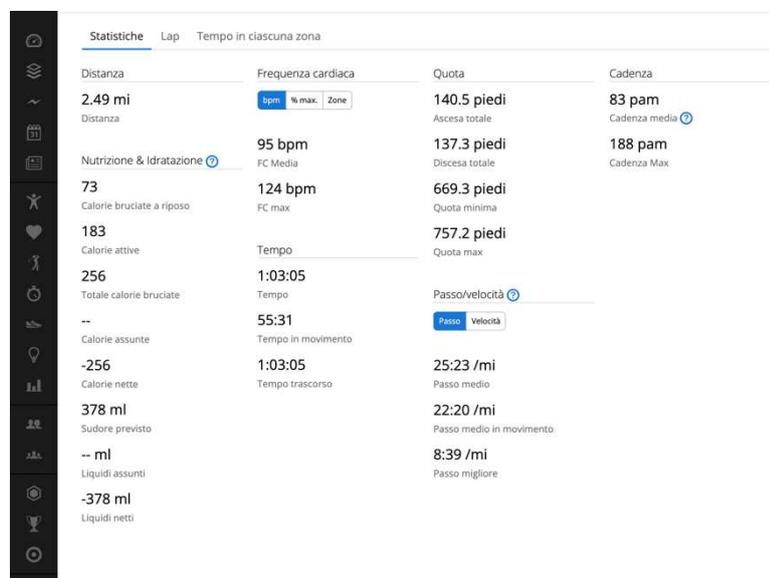


Figura 11-C. Dashboard relativa ad una attività in Garmin Connect

Attraverso Garmin Connect è possibile scaricare esclusivamente i dati relativi all'attività salvata in diversi formati (originale .fit, .tcx, .csv e .gpx) (**Figura 12**), in cui però l'unico parametro fisiologico disponibile è la frequenza cardiaca.



Figura 12. Esportazione dati da Garmin Connect

Limite dell'utilizzo di questo sito è sia la sola possibilità di poter scaricare i dati relativi alle attività salvate manualmente e non quelli relativi all'intero periodo di 2 settimane in cui lo smartwatch è indossato dall'utente, sia l'ottenimento della frequenza cardiaca come unico parametro fisiologico.

Come già detto in precedenza, in questo lavoro di tesi è importante l'acquisizione di più parametri fisiologici affinché si possa strutturare una dashboard precisa che permetta una buona valutazione della salute dell'utente; per questo motivo si andrà ad utilizzare un sito differente da Garmin Connect che in modo efficiente permette il download dei dati necessari.

3.2.2 Fitrockr

Fitrockr [14] è una piattaforma integrata al sito Garmin Health che permette l'accesso, l'analisi e l'estrazione dei dati utili, salvati in Garmin Connect, per la creazione della dashboard.

Primo passo importante è l'iscrizione al sito tramite diverse tipologie di abbonamento (**Figura 13**): è possibile effettuare un periodo di prova gratuito di 2 settimane (TRIAL) o pagare al mese una certa cifra in base al servizio di cui si vuole usufruire (BASIC e PRO). Per questo specifico lavoro di tesi si è utilizzato il periodo di prova di 2 settimane (dal 1° dicembre al 15 dicembre) che coincide con il lasso di tempo in cui l'orologio viene indossato.

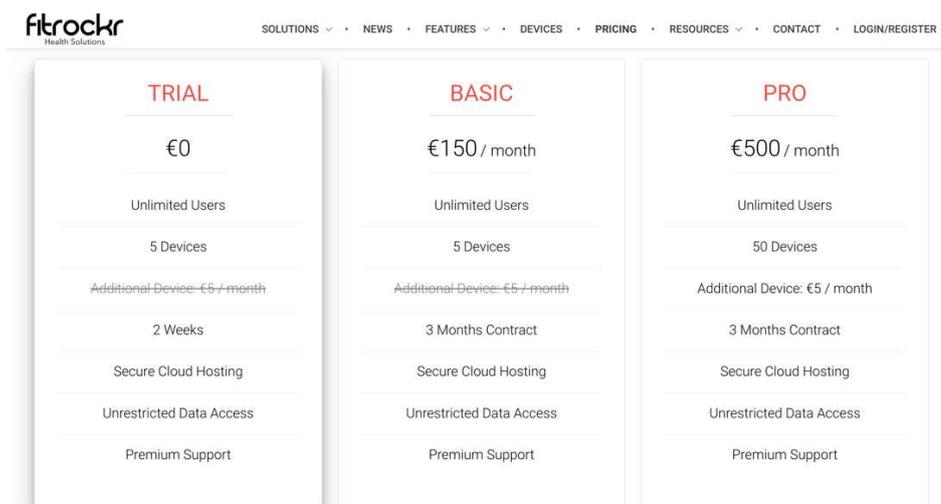


Figura 13. Abbonamenti per l'accesso e l'utilizzo di Fitrockr

Una volta eseguito l'accesso, è necessario scaricare nello smartphone o nel tablet l'applicazione Fitrockr Hub, effettuare l'accoppiamento app e smartwatch Garmin e successivamente sincronizzare i dati acquisiti dall'orologio; questa procedura permette il salvataggio dei dati nel cloud dell'account Fitrockr.

La sincronizzazione può essere fatta sia manualmente premendo la scritta "Sync Garmin Device" (Figura 14) oppure viene svolta automaticamente dallo smartphone (il numero di volte al giorno in cui avviene la sincronizzazione si determina nelle impostazioni). Come si può vedere dalla figura sottostante, l'ultima sincronizzazione è avvenuta il 15 dicembre, ciò significa che nel cloud Fitrockr sono stati salvati tutti i dati registrati dalla prima sincronizzazione all'ultima.



Figura 14. Screenshot app Fitrockr Hub

Passando alla pagina web Fitrockr, per ottenere i dati finali è necessario seguire alcuni passi importanti [15] mostrati in **Figura 15**:

- 1- **SETUP PROJECT**: in questa fase bisogna scegliere quali dati misurare e definire la frequenza di campionamento, ovvero l'intervallo di registrazione di ciascun parametro. Successivamente è necessario effettuare la sincronizzazione che, come visto in precedenza, viene eseguita tramite applicazione dello smartphone e notificata nel sito stesso di Fitrockr, come visto in figura. Questi passaggi vengono svolti subito dopo che avviene l'iscrizione a Fitrockr e danno inizio all'acquisizione dei parametri fisiologici da parte dell'orologio.
- 2- **COLLECT DATA**: una volta effettuato il setup e fino all'ultimo giorno disponibile di Fitrockr, quindi per 2 settimane, è necessario solamente indossare lo smartwatch Garmin, affinché possa registrare tutti i dati relativi all'utente e sincronizzare tramite app per poterli collezionare nel cloud. Durante tutta questa fase è consigliato monitorare l'acquisizione dei dati, scaricandoli e controllare che non ci siano anomalie durante la misurazione.
- 3- **ACCESS RAW DATA**: questo è lo step finale in cui, una volta acquisiti nelle due settimane, si scaricano tutti i dati di interesse nel formato che più si preferisce (excel, .csv e .json).

Name	E-Mail	EntryLang	Tracker	Last Sync
BVAEB-SQ01	bvae01-sq01@hotmail.com	DE / de	Venu Sq	5/27/22, 5:19 PM
BVAEB-SQ02	bvae01-sq02@hotmail.com	DE / de	Venu Sq	5/27/22, 9:46 PM
BVAEB-SQ03	bvae01-sq03@hotmail.com	DE / de	Venu Sq	5/26/22, 9:36 AM
BVAEB-SQ04	bvae01-sq04@hotmail.com	DE / de	Venu Sq	5/27/22, 3:57 PM
BVAEB-SQ05	bvae01-sq05@hotmail.com	DE / de	Venu Sq	5/27/22, 6:19 PM

Figura 15. Step eseguiti in Fitrockr

Nel momento in cui si estraggono i dati, è possibile scegliere il periodo di tempo in cui questi vengono acquisiti, quali dati scaricare e in che formato.

I parametri in questione sono:

- **User:** dati relativi all'utente, come per esempio id, nome, e-mail, anno di nascita, peso, altezza, sesso, ultima sincronizzazione
- **Dailies:** dati sommari registrati giornalmente relativi a user, passi, intensità dell'attività svolta, frequenza cardiaca minima, media e massima, livelli di stress
- **Intraday:** dati simili a dailies, con frequenza di campionamento maggiore
- **Sleep:** dati relativi al sonno, la sua durata, la durata delle varie fasi: rem, sonno leggero, profondo
- **Stress:** dati relativi allo stress
- **Motion intensity:** dati relativi all'intensità del movimento riferiti all'accelerometro
- **Activities:** dati riferiti alle attività svolte, distanza percorsa, numero di passi totali, calorie, frequenza cardiaca massima e media, valori massimi e medi di ogni attività svolta (corsa, camminata, nuoto, bicicletta, ...)
- **Zero crossing:** indica il numero delle volte in cui il segnale di accelerazione attraversa lo zero, quindi, misura la frequenza del movimento
- **BBI (Heart rate variability, HRV):** dati riferiti alla variabilità della frequenza cardiaca, cioè l'intervallo tra un battito e l'altro in ms
- **VO2 max:** definisce il livello cardiorespiratorio e aerobico personale, ovvero il massimo volume di ossigeno consumato per minuto (in ml) per kg di peso
- **Spo2 (Pulse Ox):** misura la saturazione dell'ossigeno nel sangue
- **Spo2 logging:** misura la saturazione dell'ossigeno nel sangue
- **Respiration:** dati relativi alla respirazione (respiro/min)
- **Gamification data:** dati giornalieri relativi alla distanza percorsa, al numero di passi, alle calorie e inoltre ad una media di 30 giorni della distanza, calorie, passi e durata delle attività
- **Heart rate (HR):** misura la frequenza cardiaca in battiti al minuto (bpm)

- **Step log:** dati relativi al numero di passi giornalieri e al numero di passi nell'intervallo di acquisizione
- **Body composition:** dati relativi al peso, a percentuali di grasso corporeo e di acqua nel corpo, all'indice di massa corporea, a massa ossea e muscolare (in g)

Considerando la destinazione d'uso di questi dati, l'acquisizione è relativa al periodo di 2 settimane di prova (1 dicembre - 15 dicembre), mentre il formato scelto è .csv perché, come si vedrà in seguito, è il formato che è possibile leggere in Python.

3.2.3 Analisi dei dati e implementazione della dashboard

Una volta estratti i dati relativi alle misurazioni attuate dallo smartwatch, occorre studiarli per capire quali possono essere utilizzati per realizzare la dashboard.

Un primo passo è quello di esaminare i dati scaricati, costruendo un inventario dei dati (data inventory), mostrato in **Tabella 1**, che consiste in un elenco di tutti gli elementi presenti in ciascun tipo di dato.

csv	Data Elements
activities.csv	Empty
bbi.csv	<ul style="list-style-type: none"> • User ID • User First Name • User Last Name • Calendar Date (Local) • Start Time (Local) • Time Zone (Local) • Calendar Date (UTC) • Start Time (UTC) • Start Time (ms) • Time Zone (s) • Hrv (ms)
body-composition.csv	Empty
dailies.csv	<ul style="list-style-type: none"> • Start Time (s) • End Time (Local) • End Time (UTC) • Duration (s) • Steps • Distance (m) • Moderate Intensity Duration (s) • Heart Rate (min bpm) • Heart Rate (avg bpm) • Heart Rate (max bpm) • Stress Level (avg) • Stress Level (max)

	<ul style="list-style-type: none"> • Stress Duration (s) • Rest Stress Duration (s) • Low Stress Duration (s) • Medium Stress Duration (s) • High Stress Duration (s) • Stress Qualifier
daily-steps-by-date.csv	Empty
heartrate.csv	<ul style="list-style-type: none"> • Start Time (s) • Heart Rate (bpm)
intraday.csv	<ul style="list-style-type: none"> • Start Time (s) • End Time (Local) • End Time (UTC) • Duration (s) • Active Seconds • Active Calories • Distance (m) • Steps • Met Value • Intensity • Stress Level (avg) • Stress Level (max) • Motion Intensity (avg) • Motion Intensity (max)
motion-intensity.csv	<ul style="list-style-type: none"> • Start Time (s) • End Time (Local) • End Time (UTC) • Duration (s) • Activity Type • Motion Intensity
respiration.csv	<ul style="list-style-type: none"> • Start Time (ms) • Respiration (breaths/min)
sleep.csv	<ul style="list-style-type: none"> • Start Time (s) • End Time (Local) • End Time (UTC) • Processing Time • Duration (s) • Rem Sleep (s) • Deep Sleep Duration (s) • Light Sleep Duration (s) • Awake Duration (s) • Sleep Efficiency • SleepPhaseStartTimeSec • SleepPhaseEndTimeSec • SleepPhaseStartTime • SleepPhaseEndTime • SleepPhaseDuration • SleepLevel
spo2-logging.csv	<ul style="list-style-type: none"> • Start Time (ms) • Spo2
spo2.csv	Empty
step-log.csv	<ul style="list-style-type: none"> • Start Time (ms) • Step Count • Total Count
stress.csv	<ul style="list-style-type: none"> • Start Time (s) • Stress Level Value
user.csv	User's information
vo2-max.csv	Empty
zero-crossing.csv	<ul style="list-style-type: none"> • Start Time (ms) • Time Elapsed • Energy Total • Zero Crossing Count

Tabella 1. Data inventory

Il data inventory permette di ottenere una prima panoramica generale sulla struttura dei dati disponibili, alcuni dei quali sono risultati vuoti e quindi conseguentemente non inclusi nell'analisi.

Grazie alla stesura di alcune righe di codice in Python, si riescono a creare diverse tabelle, indicizzate da righe e colonne, che permettono il confronto dei vari dati affinché venga effettuata la sincronizzazione dei segnali acquisiti, perché per ogni tipo di dato c'è un differente periodo di acquisizione. Per semplificare la comparazione, il codice Python permette di graficare i dataframe ottenendo la **Figura 16**.

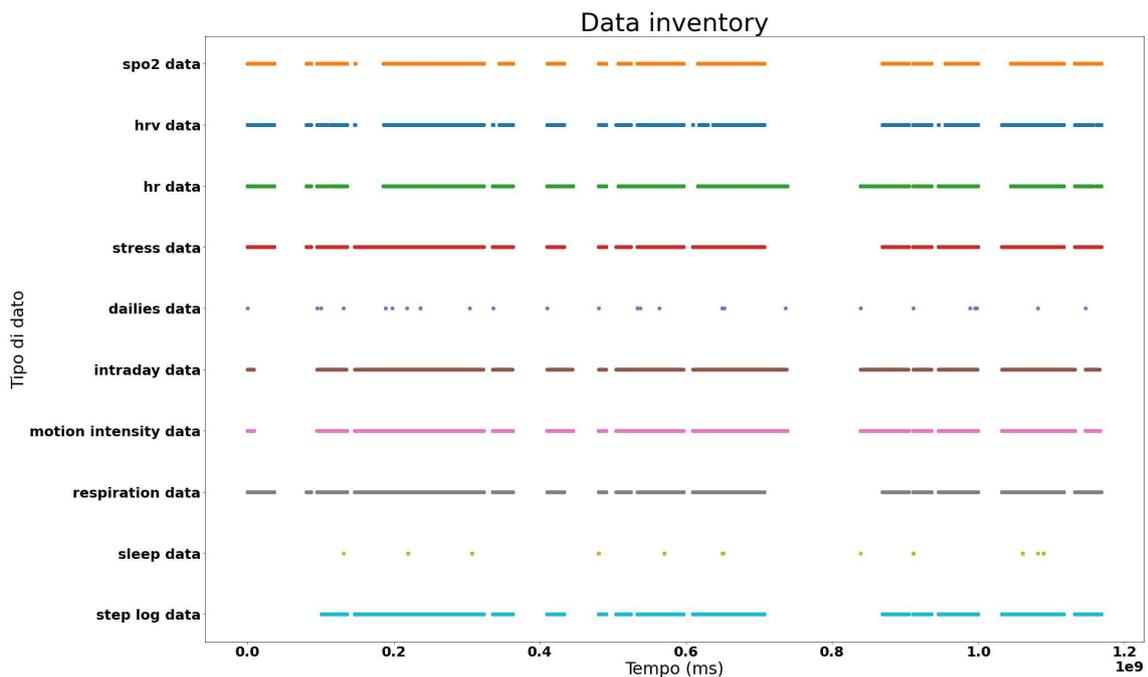


Figura 16. Confronto dei periodi di acquisizione dei dati

Attraverso la **Figura 16**, in cui sono rappresentati nell'asse y i differenti dati, mentre nell'asse x il tempo in millisecondi (ms), è possibile osservare frequenze di campionamento differenti in base al tipo di parametro considerato.

Per questo studio è importante sia essere in possesso di una quantità di dati più grande possibile per ottenere un risultato più accurato, che utilizzare parametri che permettano un monitoraggio veritiero della salute del paziente; per queste motivazioni si prenderanno in considerazione e analizzeranno i dati relativi a frequenza cardiaca (heart rate, HR), variabilità della frequenza cardiaca (heart rate variability, HRV), frequenza respiratoria (respiration), saturazione di ossigeno (saturation, spo2) e livello di stress (stress).

Passo successivo alla scelta di parametri da utilizzare è l'analisi degli stessi svolta mediante la strutturazione di 5 diversi dataframe. È possibile notare che all'interno di questi sono presenti dei valori anomali che stanno ad indicare una mancata o sbagliata misurazione da parte del dispositivo indossabile.

Per superare questa problematica ci si avvale dell'utilizzo di filtri, che teoricamente possono essere considerati passa-banda, perché, come si vedrà in seguito, vanno ad eliminare i valori che si trovano al di fuori del range ammesso dei dati considerati.

La procedura del filtraggio in questo caso consiste nell'eliminazione di tutti i valori anomali all'interno dei dataframe. Se si considera la frequenza cardiaca non filtrata, il dataframe è rappresentato nella **Figura 17**.

Index	Heart Rate (bpm)	Start Time (UTC)	Start Time (s)
0	70	2022-12-01T10:29:00	1669898540
1	0	2022-12-01T10:30:00	1669898600
2	0	2022-12-01T10:31:00	1669898660
3	70	2022-12-01T10:31:59	1669898719
4	0	2022-12-01T10:32:59	1669898779
5	75	2022-12-01T10:33:05	1669898785
6	0	2022-12-01T10:33:59	1669898839
7	0	2022-12-01T10:34:05	1669898845
8	72	2022-12-01T10:34:08	1669898848
9	0	2022-12-01T10:34:59	1669898899
10	0	2022-12-01T10:35:05	1669898905
11	74	2022-12-01T10:35:08	1669898908
12	0	2022-12-01T10:35:59	1669898959

Figura 17. Dataframe non filtrato della frequenza cardiaca

È possibile, infatti, notare che sono presenti nella colonna 'Heart Rate (bpm)' i valori 0. Le righe di codice che si utilizzano per filtrare sono semplici: si crea un nuovo dataframe ottenuto dall'apertura e lettura del file 'heartrate.csv' scaricato da Fitrockr relativo alla frequenza cardiaca, di questo dataframe si considerano solo determinate colonne, successivamente avviene il filtraggio, vengono considerate le righe in cui i valori della frequenza cardiaca sono diverse da 0 e 255, poi si resetta l'indice per ottenere un dataframe ordinato. Il dataframe ottenuto è visualizzato nella **Figura 18**.

Index	Heart Rate (bpm)	Start Time (UTC)	Start Time (s)
0	78	2022-12-01T10:...	1669890548
1	78	2022-12-01T10:...	1669890719
2	75	2022-12-01T10:...	1669890785
3	72	2022-12-01T10:...	1669890848
4	74	2022-12-01T10:...	1669890908
5	71	2022-12-01T10:...	1669890969
6	78	2022-12-01T10:...	1669891032
7	66	2022-12-01T10:...	1669891092
8	68	2022-12-01T10:...	1669891152
9	65	2022-12-01T10:...	1669891212
10	72	2022-12-01T10:...	1669891272
11	74	2022-12-01T10:...	1669891332
12	72	2022-12-01T10:...	1669891392

Figura 18. Dataframe filtrato della frequenza cardiaca

Stesso procedimento è svolto nel filtraggio del dataframe relativo alla saturazione, dove vengono considerati come valori anomali 0 e 240, e alla frequenza respiratoria dove invece vengono eliminate le righe di codice in cui il valore ‘Respiration (breaths/min)’ corrispondeva a 0, 2, 253 e 254.

Il filtraggio precedente permette di ottenere dati fisiologici verosimili, su cui è possibile porre le basi per strutturare una dashboard che permetta un monitoraggio di salute attendibile; si può notare infatti come il filtraggio modifica il dataset primario, ottenendo dei valori che rientrano nei range fisiologici dei parametri misurati. È possibile anche osservare, graficando i dati filtrati, che con l’utilizzo di filtri, la rappresentazione dei parametri è differente e, inoltre, variano anche valori utili per stabilire lo stato di salute del paziente; infatti, dalle figure sottostanti è importante evidenziare come la media giornaliera di saturazione e frequenza respiratoria variano considerevolmente.

Prima di tutto bisogna considerare che i range fisiologici relativi alla saturazione sono tra 96% e 99%, mentre quelli relativi alla frequenza respiratoria, per un adulto in condizioni normali, vanno da 16 a 20 atti respiratori al minuto.

Nel primo caso (**Figura 19**) la saturazione media giornaliera non filtrata, corrisponde a 105 %, valore che addirittura non esiste, mentre la media giornaliera del segnale filtrato equivale a 96%.



Figura 19. Parametro saturazione non filtrato e filtrato

Nel caso riferito alla frequenza respiratoria (**Figura 20**), il dato di media giornaliera è di 52 battiti al minuto, questo in teoria indica una condizione di tachipnea, cioè un aumento della frequenza respiratoria al minuto, ma dalla visualizzazione della media riferita al valore filtrato si evidenzia una situazione pressoché normale dal momento che equivale a 13 battiti al minuto.

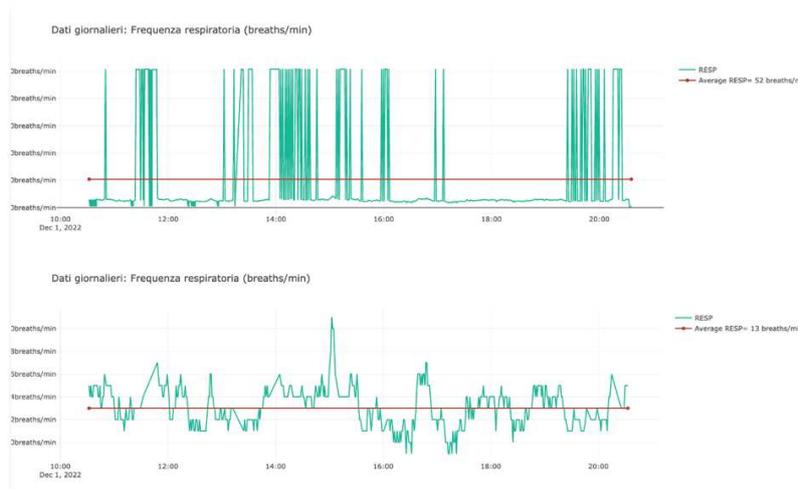


Figura 20. Parametro frequenza respiratoria non filtrato e filtrato

Una volta filtrati i dati, un passo successivo importante è creare 5 dataset (riferiti ai 5 parametri considerati) che siano omologati tra loro, al fine di semplificare la codifica della dashboard.

Questo passaggio viene svolto modificando i dataframe attraverso righe di codice: prendendo come esempio il parametro della frequenza cardiaca, dopo essere stato filtrato, viene rinominata la colonna relativa all'heart rate in 'Data value', vengono aggiunte altre colonne relative a data e ora e un'ulteriore colonna 'Timestamp' in cui sono rappresentati data e ora in datetime, una formattazione specifica dei file in .csv in cui si mantiene il formato data (aaaa/mm/gg) e ora ('%Y/%m/%d %H:%M:%S') e si crea un nuovo file .csv che ha le precedenti caratteristiche.

Index	Data Value	Start Time (UTC)	Start Time (s)	Start Time (ms)	Date	Hour	Timestamp
0	78	2022-12-01T10:29:00	1669898540	1669898540000	2022-12-01	10:29:00	2022-12-01 10:29:00
1	78	2022-12-01T10:31:59	1669898719	1669898719000	2022-12-01	10:31:59	2022-12-01 10:31:59
2	75	2022-12-01T10:33:05	1669898785	1669898785000	2022-12-01	10:33:05	2022-12-01 10:33:05
3	72	2022-12-01T10:34:08	1669898848	1669898848000	2022-12-01	10:34:08	2022-12-01 10:34:08
4	74	2022-12-01T10:35:08	1669898988	1669898988000	2022-12-01	10:35:08	2022-12-01 10:35:08
5	71	2022-12-01T10:36:09	1669899069	1669899069000	2022-12-01	10:36:09	2022-12-01 10:36:09
6	78	2022-12-01T10:37:12	1669899132	1669899132000	2022-12-01	10:37:12	2022-12-01 10:37:12
7	66	2022-12-01T10:38:12	1669899189	1669899189000	2022-12-01	10:38:12	2022-12-01 10:38:12

Figura 21. Dataframe finale della frequenza cardiaca

La **Figura 21** rappresenta uno dei dataset che si ottiene dal procedimento precedente. Analogo sarà per gli altri 4 dataset, con la differenza che all'interno della colonna 'Data Value' non ci sarà la frequenza cardiaca, ma rispettivamente variabilità della frequenza cardiaca, frequenza respiratoria, saturazione e livello di stress.

Dopo che sono stati analizzati e modificati i dati fisiologici estratti, si passa all'implementazione della dashboard vera e propria in Visual Studio, una piattaforma che supporta diversi linguaggi di programmazione, tra cui Python. Si utilizzerà, più precisamente, Dash [16], un micro-framework Python che permette di creare interfacce di visualizzazioni dati.

Prima cosa da fare è progettare la dashboard, scegliere, dunque, cosa visualizzare e come farlo; i parametri che si utilizzeranno sono quelli scelti e filtrati in precedenza, mentre come questi verranno visualizzati dipende dal fine ultimo della dashboard. Visto che lo scopo è il monitoraggio e la diagnosi dei pazienti, sarà reso visibile l'andamento dei vari parametri tramite dei grafici, affiancato anche ad un'analisi cumulativa (valore medio o correlazione tra i dati) riferita ad una singola giornata o ad un range di tempo più vasto.

Prima di iniziare a strutturare la dashboard è importante definire una directory che conterrà sia il progetto che i dati scaricati e filtrati e successivamente creare un virtual environment [17], un ambiente di sviluppo isolato dagli altri, contenente diversi pacchetti installati necessari per sviluppare l'applicazione. Questo passo iniziale si attua direttamente nel terminale del pc o di Visual Studio Code scrivendo semplici righe di codice (**Figura 22**).

```
~ % python -m venv dashenv
~ % source dashenv/bin/activate
```

Figura 22. Creazione virtual environment (dashenv)

Dopo aver elaborato i dati, è importante che si inizializzi la dashboard (riga 91 della **Figura 23**), poi successivamente si passa alla stesura del codice che serve a definirne il contenuto.

```
83 external_stylesheets = [
84     {
85         "href": "https://fonts.googleapis.com/css2?"
86         "family=Lato:wght@400;700&display=swap",
87         "rel": "stylesheet",
88     },
89 ]
90
91 app = dash.Dash(__name__, external_stylesheets=[dbc.themes.BOOTSTRAP])
92 app.title = "Dashboard salute!"
93
```

Figura 23. Righe di codice in Python

Si utilizzano Dash HTML Components [18], cioè moduli per creare il layout della dashboard direttamente in Python senza andare ad utilizzare il linguaggio di programmazione HTML e Dash Core Components [19], cioè moduli che permettono di creare le componenti interattive, come grafici o menù a discesa per la scelta di parametri o date.

L'interattività è data dalla Callback, che viene definita dal glossario online di Python [20] come “una funzione che viene passata come argomento per essere eseguita in un momento futuro”; infatti, durante la codifica, nella definizione del layout della dashboard, le funzioni vengono solamente dichiarate, mentre all'esterno di questo codice viene poi definita la funzione, questo per semplificare il codice stesso.

Per quanto riguarda invece lo stile di tutta la dashboard si fa riferimento ad un file style.css esterno, caricato poi nell'applicazione (righe da 83 a 89 della **Figura 23**),

in cui si vengono stabiliti il font, la dimensione, il colore o la posizione degli elementi all'interno della dashboard. Nella stessa directory in cui è presente style.css, si trova anche il file favicon.ico che contiene l'immagine che si vuole visualizzare nella barra del titolo del browser web insieme al titolo della dashboard stessa (definito nella riga 92 della **Figura 23**).

Per poter poi eseguire l'intera applicazione dash, il codice si conclude con delle righe finali (**Figura 24**).

```
992     if __name__ == "__main__":  
993         app.run_server(debug=True)
```

Figura 24. Righe finali del codice

Una volta conclusa la stesura del codice, è necessario eseguirlo nel terminale di Visual Studio, dove successivamente si visualizzerà un URL (Uniform Resource Locator), una sequenza di caratteri che identifica in modo univoco l'indirizzo web della dashboard.

4. RISULTATI

4.1 Visualizzazione dei risultati

La dashboard ottenuta si apre sulla pagina web, cliccando sull'URL visualizzato nel terminale dopo che si è eseguito tutto il codice.

Inizialmente l'applicazione aperta sarà raffigurata come in **Figura 25 A-B**: è presente una sezione per la scelta del parametro, un primo grafico (**Figura 25-A**) in cui viene rappresentato l'andamento della frequenza cardiaca nelle 2 settimane, un secondo grafico (**Figura 25-B**) rappresenta invece i valori della frequenza cardiaca durante il periodo di acquisizione dei dati, mentre l'ultimo grafico rappresenta la correlazione dei parametri tra loro.

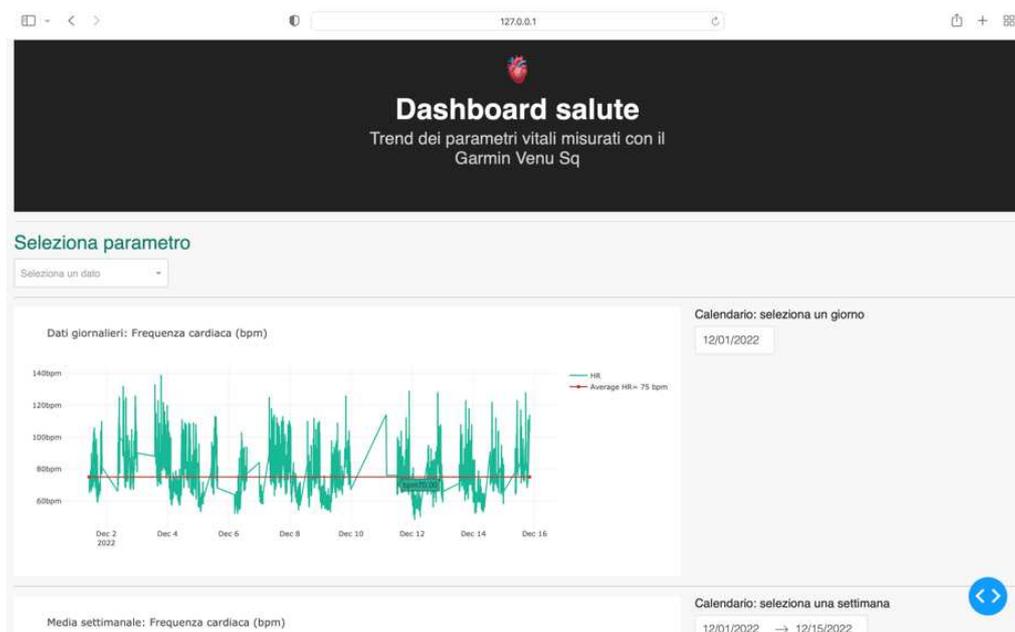


Figura 25-A. Dashboard aperta

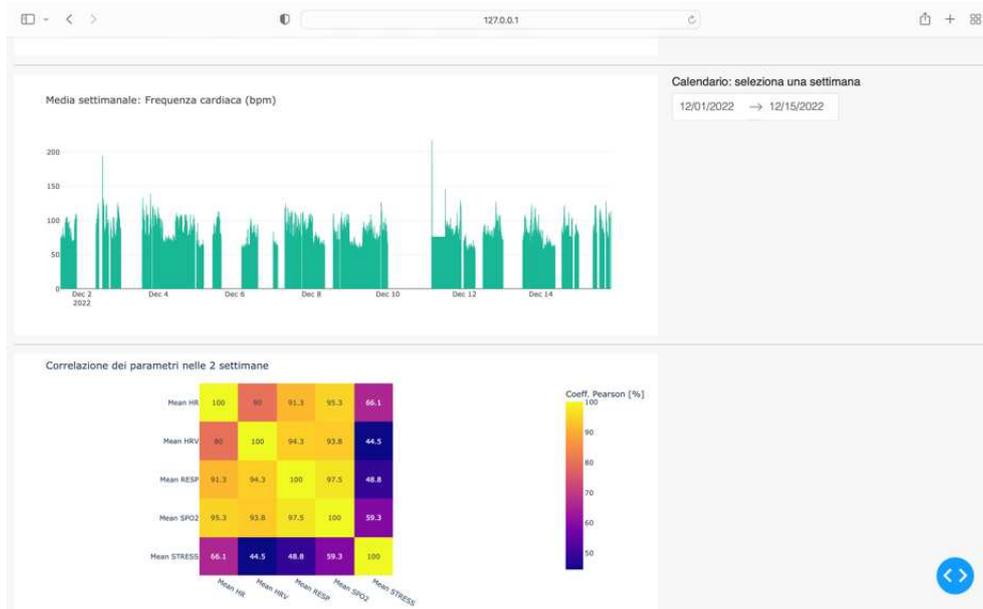


Figura 25-B. Dashboard aperta

Come visto precedentemente, sono stati aggiunti al codice elementi che rendono la dashboard interattiva; è possibile scegliere, attraverso un menù a tendina (Figura 26), il tipo di parametro che si vuole monitorare tra frequenza cardiaca (HR), variabilità della frequenza cardiaca (HRV), saturazione di ossigeno (SPO2), frequenza respiratoria (RESP) e livello di stress (STRESS).

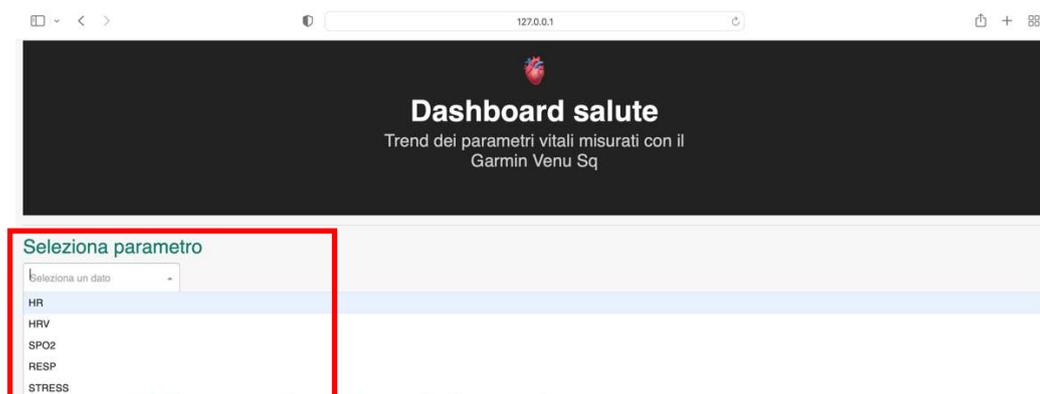


Figura 26. Menù a tendina per selezionare il parametro da monitorare

Si nota un primo grafico in cui si visualizza l'andamento e la media del parametro scelto (frequenza cardiaca) durante una giornata; è importante sottolineare che il giorno viene scelto direttamente dall'utente attraverso il calendario (Figura 27).

È possibile evidenziare come la frequenza cardiaca ha valori minori durante la fascia notturna rispetto all'intera giornata e che invece si nota un picco di intensità che corrisponde all'attività fisica svolta da chi indossa il Garmin.

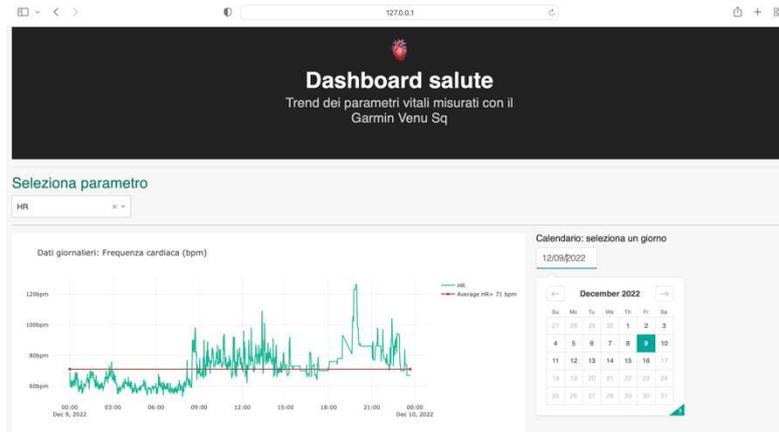


Figura 27. Andamento della frequenza cardiaca durante il 9 dicembre

Il secondo grafico (Figura 28-A) invece rappresenta le medie giornaliere del parametro in considerazione in un range di tempo che è comunque scelto dall'utente sempre attraverso il calendario (in questo caso dal 4 dicembre al 9 dicembre).

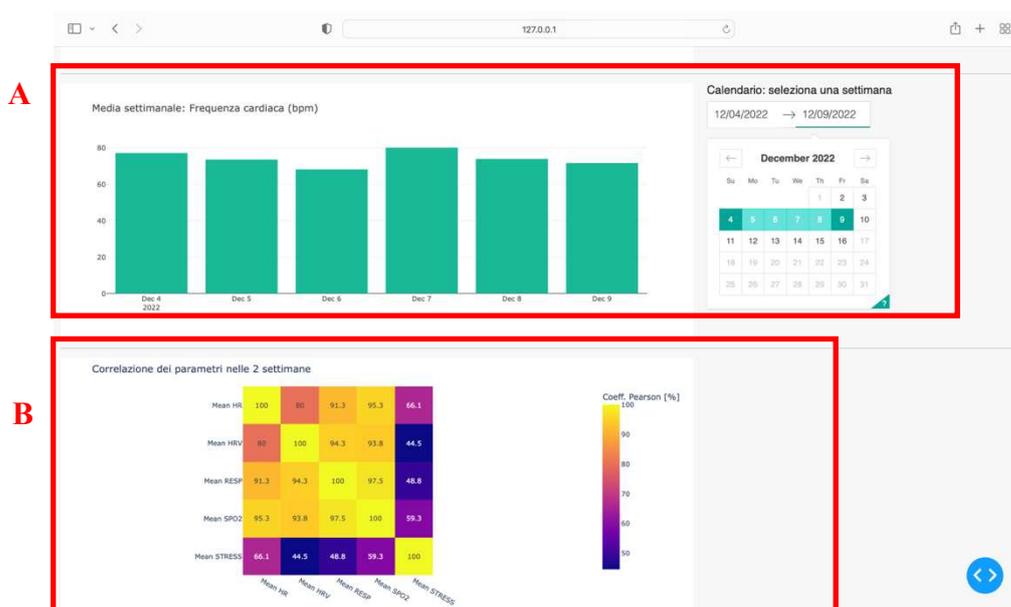


Figura 28. Medie giornaliere della frequenza cardiaca dal 4 dicembre al 9 dicembre

L'ultimo grafico (**Figura 28-B**) invece rappresenta la correlazione in percentuale, tramite heatmap o mappa di calore, della media complessiva di tutti e 5 i parametri durante tutto il periodo di utilizzo dello smartwatch. La correlazione sta ad indicare la relazione che sussiste tra due parametri, quindi, più questa sarà elevata, più è elevata la tendenza di una variabile a cambiare rispetto all'altra. Nel grafico si può evidenziare come c'è una alta correlazione tra frequenza respiratoria e saturazione o frequenza cardiaca e saturazione, questo deriva dal fatto che è proprio grazie alla circolazione sistemica sanguigna e alla respirazione che i livelli di ossigeno del sangue variano.

5. CONCLUSIONI

Lo scopo di questo progetto di tesi è di implementare una dashboard basata sull'acquisizione di parametri fisiologici tramite un dispositivo indossabile che aiuti un medico o un caregiver nel monitoraggio e nella diagnosi della salute di un paziente.

Il sistema di raccolta di dati e la relativa analisi sostituisce la fase in cui avviene una diretta interazione tra medico e paziente durante visite ambulatoriali o domiciliari, necessarie per la valutazione della salute di quest'ultimo. Questo avviene perché i parametri fisiologici che prima venivano misurati dal medico tramite diverse strumentazioni come pulsossimetro, stetoscopio o sfigmomanometro, vengono invece misurati e registrati direttamente dal Garmin Venu Sq.

È compito poi dello sviluppatore dell'applicazione permettere una visualizzazione e un'accessibilità semplice e veloce dei parametri rilevanti, non solo per il medico, ma anche per un eventuale caregiver.

Questo porterebbe ad un miglioramento dell'efficienza ambulatoriale sia perché si consente un accesso veloce ai dati giornalieri o cumulativi relativi al paziente, di conseguenza il medico avrebbe a disposizione più tempo per monitorare più pazienti o per effettuare visite specifiche che necessitano di un appuntamento e sia perché la dashboard è strutturata in maniera tale da poter effettuare anche un monitoraggio autonomo, grazie alla semplicità della rappresentazione dei parametri.

In conclusione, il sistema di raccolta dei dati misurati e l'analisi di questi è un processo che rende più accessibile e semplice il monitoraggio della salute e, grazie alla possibilità di modificare il codice sorgente dell'applicazione implementata, è anche modellabile, dal punto di vista sia di visualizzazione che di parametri considerati, in funzione alle differenti esigenze cliniche di un paziente.

6. BIBLIOGRAFIA E SITOGRAFIA

[1] <https://www.digitalpills.it/cosa-e-la-dashboard>

[2] “Internet of medical things (IoMT)-based heart rate and body temperature monitoring system” (2022), Ida Laila, A Arifin, Bidayatul Armynah

[3] “Sensors Based Smart Healthcare Framework Using Internet Of Things (Iot)” (2020), Hitesh Kumar Sharma, J.C. Patni, Prashant Ahlawat, Siddhratha Sankar Biswas

[4] “Virtual health care for community management of patients with COVID-19.” (2020), Hutchings O., Dearing C., Jagers D., Shaw M., Raffan F., Jones A., Taggart R., Sinclair T., Anderson T., Ritchie AG.

[5] “Features Constituting Actionable COVID-19 Dashboards: Descriptive Assessment and Expert Appraisal of 158 Public Web-Based COVID-19 Dashboards” (2021), Damir Ivanković, MD, MBA; Erica Barbazza, MSc; Véronique Bos, MA; Óscar Brito Fernandes, MSc, MEd; Kendall Jamieson Gilmore, MSc; Tessa Jansen, MPH, PhD; Pinar Kara, MSc; Nicolas Larrain, MA; Shan Lu, PhD; Bernardo Meza-Torres, MSc, MD; Joko Mulyanto, MSc, MD, PhD; Mircha Poldrugovac, MSc, MD; Alexandru Rotar, MSc, PhD; Sophie Wang, MPH; Claire Willmington, MSc; Yuanhang Yang, MSc; Zhamin Yelgezekova, MSc; Sara Allin, MSc, PhD; Niek Klazinga, MD, PhD; Dionne Kringos, MSc, PhD

[6] “EHDViz: clinical dashboard development using open-source technologies” (2016), Marcus A Badgeley, Khader Shameer, Benjamin S Glicksberg, Max S Tomlinson, Matthew A Levin, Patrick J McCormick, Andrew Kasarskis, David L Reich, Joel T Dudley

[7] “On revisiting vital signs IoT sensors for COVID-19 and long COVID-19 monitoring: a condensed updated review and future directions.” (2021), Vinicius

Facco Rodrigues, Rodrigo da Rosa Righi, Lucas Mayer Ceschini, Barbara Canali Locatelli, Bruna Donida, Cristiano André da Costa

[8] “Better Understanding of the Metamorphosis of Pregnancy (BUMP): protocol for a digital feasibility study in women from preconception to postpartum” (2022), S. M. Goodday, E. Karlin, A. Brooks, C. Chapman, D. R. Karlin, L. Foschini, E. Kipping, M. Wildman, M. Francis, H. Greenman, Li Li, E. Schadt, M. Ghassemi, A. Goldenberg, F. Cormack, N. Taptiklis, C. Centen, S. Smith and S. Friend

[9] “Association between sleep duration, sleep quality and self- esteem in adolescents in Iceland” (2021), Heiðdís Vala Þorsteinsdóttir

[10] “A Novel Redundant Validation IoT System for Affective Learning Based on Facial Expressions and Biological Signals” (2022), Antonio Costantino Marceddu, Luigi Pugliese, Jacopo Sini, Gustavo Ramirez Espinosa, Mohammadreza Amel Solouki, Pietro Chiavassa, Edoardo Giusto, Bartolomeo Montrucchio, Massimo Violante and Francesco De Pace

[11] <https://www.garmin.com/it-IT/p/707174/pn/010-02427-11#specs>

[12] <https://www.garmin.com/it-IT/p/707174/pn/010-02427-10#overview>

[13] <https://www.garmin.com/it-IT/p/125677#overview>

[14] <https://www.fitrockr.com/>

[15] <https://www.fitrockr.com/research/>

[16] <https://dash.plotly.com/>

[17] <https://docs.python.org/3/library/venv.html>

[18] <https://dash.plotly.com/dash-html-components>

[19] <https://dash.plotly.com/dash-core-components>

[20] <https://docs.python.org/3/glossary.html#term-callback>