1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

UNIVERSITÀ DEGLI STUDI DI URBINO
CARLO BO

DIPARTIMENTO DI SCIENZE PURE E APPLICATE

Laurea Triennale in Informatica Applicata

Sistema di monitoraggio presenze basato su riconoscimento biometrico

Relatore:
Prof. Antonio Della Selva

Correlatore:
Prof. Alessandro Aldini

Candidato:
Luca Neve

ANNO ACCADEMICO 2024/2025

Abbiate fame di curiosità
in questo pasto che è la vita

Indice

Elenco delle figure

Capitolo 1

Introduzione

L'idea di questo progetto è nata nel corso della mia carriera universitaria, a partire dall'esperienza vissuta in prima persona come studente vincitore di borsa di studio. Ho trascorso tre anni alloggiando presso le strutture abitative fornite dalla Regione Marche destinate a studenti con minori opportunità. Durante questa esperienza triennale, ogni anno riemergeva una situazione ricorrente: un numero significativo di studenti che avevano ottenuto l'assegnazione di un posto letto non vi risiedeva effettivamente, lasciando la stanza formalmente occupata ma praticamente inutilizzata. Questo fenomeno comporta uno spreco rilevante di risorse pubbliche, destinate a supportare il percorso universitario di studenti - o potenziali studenti - che ne avrebbero realmente bisogno e che saprebbero sfruttare appieno tali servizi.

Per ovviare a questo problema l'idea progettuale prevede l'introduzione di un sistema di rilevamento di presenze, basato su un meccanismo di autenticazione che lo studente deve periodicamente eseguire per dimostrare all'ente regionale l' effettivo utilizzo del posto letto assegnato. Va precisato che il sistema interviene a valle del problema: non previene l'utilizzo improprio del servizio offerto, ma consente di rilevare successivamente se lo studente ne usufruisce effettivamente, fornendo così agli enti pubblici gli strumenti necessari per adottare eventuali provvedimenti.

1.1 Tipi di autenticazione biometrica

Una volta analizzato il problema dell'utilizzo improprio del servizio da parte degli studenti, si è reso necessario individuare un sistema di autenticazione che fosse efficace, non aggirabile e pratico. La scelta è ricaduta su un'autenticazione biometrica che ha il vantaggio di non richiedere oggetti esterni per essere effettuata poiché si basa esclusivamente su caratteristiche fisiche e/o comportamentali dell'individuo. Questi sistemi si fondano sui **fattori di inerenza**: sono tratti fisici unici di una persona, come il modello dei vasi sanguigni nella retina, la voce, il riconoscimento facciale o l'impronta digitale [4].

Le due principali categorie di biometria sono:

- **Biometria fisica**: include caratteristiche come impronte digitali, il riconoscimento facciale, la retina, la geometria della mano ed altri tratti anatomici.
- **Biometria comportamentale**: basata su comportamenti dinamici come la voce, l'andatura, dinamica di digitazione ed altri fattori.

Nel contesto nel quale deve essere applicato questo progetto, i principali vantaggi di un'autenticazione biometrica attraverso la biometria fisica sono i seguenti:

- **Identificazione univoca**: associa con certezza l'identità del soggetto al tentativo di accesso.
- **Non replicabilità**: i dati biometrici non sono facilmente replicabili e questo riduce il rischio di frodi.
- **Disponibilità immediata**: sono sempre a disposizione del soggetto che deve autenticarsi, senza la necessità di dispositivi esterni.
- **Non cedibilità**: non possono essere prestati a terzi per aggirare il sistema di autenticazione.

1.2 Autenticazione mediante impronte digitali

Tra tutti i tipi di autenticazione biometrica, la scelta è ricaduta sull'utilizzo delle impronte digitali. Queste ultime presentano un ottimo punto di incontro tra affidabilità, sicurezza, costi e facilità di integrazione.

Inoltre, dato che rappresenta un metodo di autenticazione molto diffuso e largamente utilizzato nella vita quotidiana (es. smartphone, servizi bancari), gode di un livello di accettazione sociale generalmente molto alto.

1.2.1 Perchè scegliere le impronte digitali: confronto con altri metodi biometrici

Sebbene l'impronta digitale offra un elevato standard di precisione e sicurezza, è importante sottolineare che non è il metodo di autenticazione biometrica più accurato. Metodi come il riconoscimento dell'iride garantiscono una precisione superiore, ma ciò si traduce in costi notevolmente più elevati per le attrezzature hardware, oltre a una maggiore complessità di integrazione nei sistemi già esistenti. Questi fattori rendono questo tipo di autenticazione non adatta al contesto e agli obiettivi del progetto che si andrà a svolgere.

Dall'altra parte abbiamo metodi come la biometria vocale o comportamentale, questi ultimi presentano il vantaggio di essere meno costosi in termini di prezzo, tuttavia godono di un livello di accuratezza inferiore rispetto alle impronte digitali. Presentano poi altri problemi, nello specifico i sistemi di riconoscimento vocale sono sensibili a fattori esterni come il rumore ambientale, questo li rende poco pratici in contesti affollati o potenzialmente rumorosi.

Capitolo 2

Aspetti normativi e tecnici del trattamento dei dati biometrici

2.1 Dati biometrici e GDPR

Essendo un progetto che lavora con dati biometrici personali è doveroso un approfondimento della sfera legale legata al trattamento dei dati. Per analizzare di che tipo di dati stiamo parlando facciamo riferimento al **Regolamento (UE) 2016/679 del Parlamento Europeo e del Consiglio del 27 Aprile 2016**, relativo alla protezione delle persone fisiche con riguardo al trattamento dei dati personali, nonché alla libera circolazione di tali dati (di seguito, `GDPR` - General Data Protection Regulation). Il GDPR definisce i dati personali come "Qualsiasi informazione riguardante una persona fisica identificata o identificabile" e segue specificando che "si considera identificabile la persona fisica che può essere identificata, direttamente o indirettamente, con particolare riferimento a un identificativo come il nome, un numero di identificazione, dati relativi all'ubicazione, un identificativo online o a uno o più elementi caratteristici della sua identità fisica, fisiologica, genetica, psichica, economica, culturale o sociale"(Art. 4.1)[2]. Dunque l'impronta digitale è un dato personale, inoltre quest'ultima rientra nella sotto-categoria di dati personali che sono i dati biometrici, vengono definiti come: "i dati personali ottenuti da un trattamento tecnico specifico relativi alle caratteristiche fisiche, fisiologiche o comportamentali di una persona fisica che ne consentono o confermano l'identificazione univoca, quali l'immagine facciale o i dati dattiloscopici" (Art. 4.14) [2]. L'articolo 9 del GDPR spiega che in generale il trattamento dei dati sensibili è vietato salvo condizioni specifiche. I dati personali diventano dati sensibili quando rivelano aspetti delicati della vita di una persona, alcuni esempi possono essere i dati biometrici, genetici, l'origine razziale o opinioni politiche.

Ci sono principalmente quattro fondamenti da rispettare quando vengono trattati dati personali biometrici:

- **Minimizzazione**: bisogna trattare solo i dati strettamente necessari rispetto

alle finalità che sono state dichiarate.

- **Limitazione**: i dati non devono essere trattati successivamente in modo incompatibile con le finalità dichiarate.
- **Sicurezza**: il trattamento dei dati deve garantire sicurezza, specialmente contro accessi non autorizzati.
- **Trasparenza**: l'interessato deve necessariamente ricevere un'informativa chiara e completa sul trattamento dei suoi dati.

2.2 Consenso al trattamento di dati personali Biometrici

All'interno del **Provvedimento del Garante della Privacy del 22 Febbraio 2024** [3] si giunge alla conclusione che in un progetto che prevede l'utilizzo di impronte digitali per scopi come il controllo di accesso generico o la rilevazione di presenze è **obbligatorio** fornire un'alternativa meno invasiva, come un badge o un altro sistema di autenticazione non biometrica. Questo perché l'individuo deve sentirsi libero di decidere se acconsentire al trattamento dei propri dati sensibili. Dunque, a questo punto, entriamo in uno scenario determinante per la conformità del progetto al GDPR. Le implicazioni si differenziano a seconda che il consenso per il trattamento dei dati sia fornito o negato; inoltre, è importante ricordare che queste casistiche vengono messe in atto per **questo specifico progetto** che ha una destinazione precisa, non sono regole universali da seguire quando c'è un trattamento di dati biometrici. Il contesto nel quale si va a lavorare è fondamentale.

Al netto di ciò, se l'individuo decide di non dare il proprio consenso al trattamento dei suoi dati biometrici:

- Il sistema **deve** fornire un'alternativa non biometrica (uso di badge, password, PIN).
- L'individuo **non** deve subire svantaggio o discriminazione, non deve subire limitazioni di accesso o qualsiasi altra conseguenza negativa rispetto a chi ha scelto di utilizzare l'impronta.

I due punti sopra elencati sono fondamentali poichè il Garante della Privacy [3] ha più volte sanzionato aziende che non fornivano alternative o che non ottenevano un valido consenso per l'utilizzo della biometria in contesti non eccezionali.

Se, al contrario, l'individuo fornisce il consenso al trattamento della propria impronta digitale, il sistema potrà utilizzare l'autenticazione biometrica. Tuttavia, in questo caso ci sono dei requisiti che rendono il consenso valido. Il consenso deve essere:

- Libero: l'individuo può scegliere di dare o negare il consenso.

- Specifico: non può essere un consenso generico, tutte le finalità devono essere esplicite e ben definite.
- Informato: l'individuo da il consenso dovo aver ricevuto una informativa sulla privacy completa che spiega completamente come verranno gestiti i suoi dati, alcuni esempi sono lo scopo del trattamento, come avverrà il trattamento, chi avrà accesso ai dati o quali sono i suoi diritti.
- Esplicito: il consenso richiede un'azione attiva come la firma di un modulo.
- Revocabile: l'interessato deve poter revocare il proprio consenso in qualsiasi momento senza difficoltà.

Una volta analizzate le condizioni affinché il consenso sia valido, bisogna anche esplicitare come devono essere utilizzati i dati biometrici forniti. Oltre ai quattro fondamenti discussi nella sezione 2.1, per avere la totale conformità con il regolamento del GDPR, deve essere effettuata una valutazione d'impatto sulla protezione dei dati (DPIA), che serve a dimostrare che il titolare del trattamento dei dati ha accuratamente considerato l'impatto delle operazioni che verranno effettuate sui dati personali e ha messo in atto le protezioni necessarie.

2.3 Trattamento locale dei dati biometrici nel sistema progettato

Nel sistema sviluppato, la gestione delle impronte digitali viene affidata al sensore R307, quest'ultimo memorizza al suo interno i dati biometrici associandoli a un ID univoco. È importante sottolineare che il dato salvato non corrisponde all'immagine dell'impronta, ma un *template* numerico generato a partire da essa, non riconvertibile all'originale. Il sistema utilizza un approccio di tipo `match-on-device`, dunque le fasi di acquisizione e confronto delle impronte avvengono esclusivamente all'interno del sensore in maniera locale. Non è previsto alcun trasferimento o elaborazione esterna dei dati biometrici acquisiti, in questo modo si può garantire una gestione locale e conforme ai principi del GDPR. C'è da sottolineare che il sistema utilizza un database locale, separato dal sensore, al cui interno ci sono solo i dati anagrafici degli individui registrati all'interno del sensore di impronte. L'associazione tra i dati biometrici presenti nel lettore e i dati anagrafici all'interno del database avviene tramite l'ID univoco assegnato a ogni impronta al momento della registrazione. Questo approccio permette di separare i dati personali dai dati biometrici, per fare in modo che, anche nel caso in cui un malintenzionato entrasse in possesso:

- solo del database: avrebbe accesso solo ai dati personali ordinari, ma non alle loro impronte.
- solo del sensore: avrebbe accesso a template numerici, ma non saprebbe a chi appartengono.

Questo approccio è coerente con il principio di `"privacy by design"`, che stabilisce che la protezione dei dati deve essere integrata già dalle prime fasi della progettazione del sistema.

Capitolo 3

Scelte implementative e funzionamento del sistema

3.1 Componenti hardware

3.1.1 ESP32

L'ESP32 è una serie di sistemi su chip a basso costo e a basso consumo creata dall'azienda cinese `"Espressif Systems"`. Oltre al processore dual-core, integra anche funzionalità Wi-Fi e Bluetooth, rendendolo una soluzione ideale per programmi IoT. È programmabile in diversi ambienti di sviluppo e in diversi linguaggi di programmazione, e si distingue per la sua versatilità.

Oltre ai numerosi pin GPIO utilizzabili per l'interazione con i componenti elettrici come sensori o display, supporta sia il Wi-Fi(802.11 b/g/n) che il Bluetooth (Classic e Low Energy). Questa doppia connettività permette all'ESP32 di essere utilizzato in una varietà di applicazioni di comunicazione wireless.

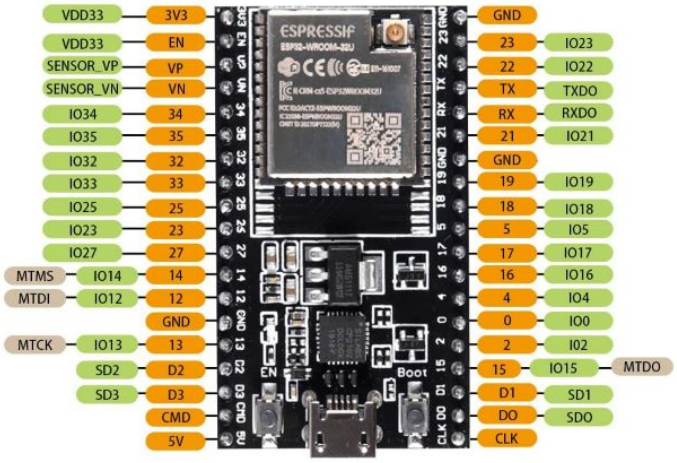

Figura 3.1: ESP32

I pin digitali dell'ESP32 sono programmabili e la loro configurazione può differire in base alle esigenze del progetto; alcuni esempi sono la comunicazione seriale (UART), la generazione di segnali PWM o la gestione di semplici input/output digitali. In questo progetto, il microcontrollore rappresenta il cuore del sistema, impartisce comandi al sensore di impronte digitali e comunica con il database locale.

3.1.2 Sensore R307

Per la gestione delle impronte digitali, la scelta è ricaduta sul sensore R307.

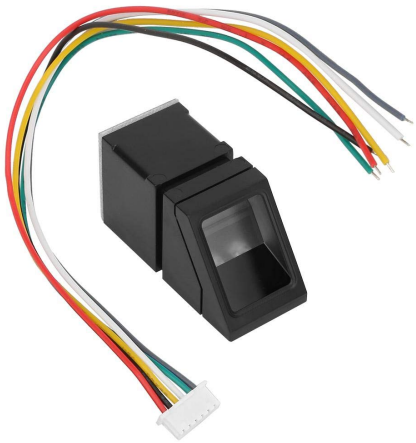

Figura 3.2: Sensore R307

Funzionamento di un lettore di impronte digitali

Le nostre mani hanno la caratteristica di presentare creste e solchi sulla pelle dei palmi e dei polpastrelli. Questa caratteristica è legata al concetto di attrito e aumenta la solidità della nostra presa su superfici ed oggetti. Quando afferriamo un oggetto, l'attrito tra la pelle e la superficie causa il deposito di residui come umidità, sporco e cellule morte. Questi residui si accumulano principalmente nei punti di contatto delle creste cutanee del polpastrello, lasciando così una traccia visibile conosciuta come *impronta digitale* [5]. Un esempio quotidiano di questo fenomeno lo si osserva nei segni lasciati sui bicchieri di vetro dopo averli afferrati o utilizzati, sono proprio le nostre impronte digitali lasciate dalle creste dei nostri polpastrelli.

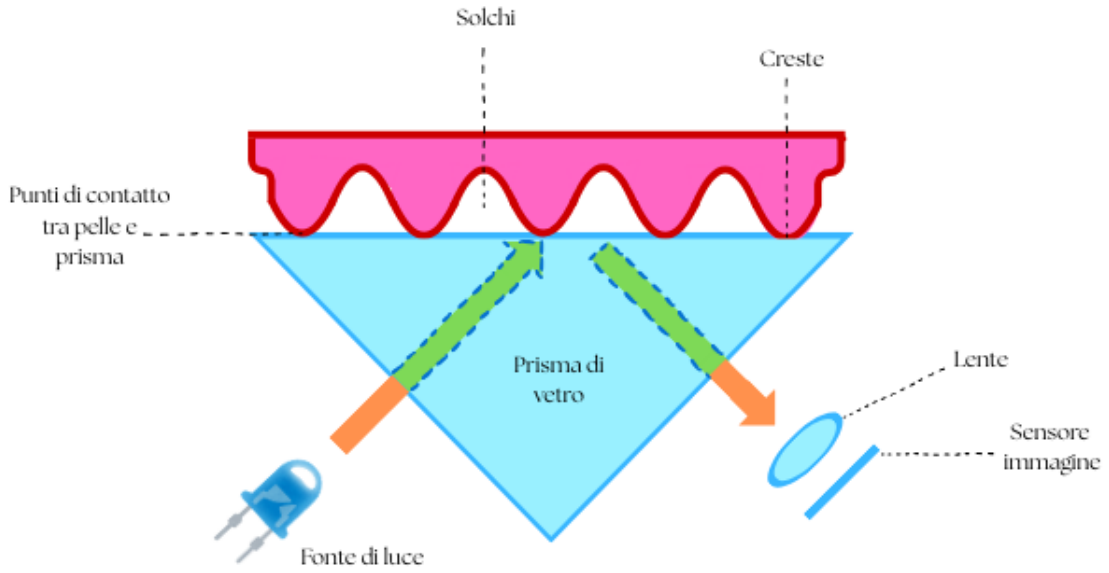

Figura 3.3: Principio di funzionamento di uno scanner ottico per impronte digitali (riflessione interna totale)

Uno scanner ottico, come l'R307, funziona seguendo il principio della **riflessione interna totale**. Viene emessa una luce led con una precisa angolazione in un lato del prisma e, quando il dito è appoggiato sulla superficie, come nella rappresentazione 3.3, la luce viene riflessa ed esce dall'altro lato del prisma. Quando il dito è appoggiato, solo le creste del polpastrello sono a buon contatto con il vetro, mentre i *solchi* rimangono "separati" dalla superficie. Questa differenza di contatto cambia il comportamento della luce riflessa: sulle creste (dove il dito tocca) la luce viene dispersa, sui solchi (dove il dito non tocca) la luce viene riflessa all'interno del prisma fino ad arrivare alla lente. Grazie alla modulazione ottica che viene effettuata dalla lente, il sensore posto dietro cattura un'immagine ad alto contrasto della struttura dell'impronta, che verrà elaborata poi digitalmente per l'identificazione.

A seguito di questa breve introduzione sul funzionamento generale di uno scanner ottico per impronte digitali, è necessario un approfondimento sulle caratteristiche del sensore scelto per questo progetto.

Caratteristiche R307

	R307
Capacità di Memoria	Fino a 1000 template biometrici (memoria flash interna)
Precisione (FAR)	<0,001% (False Acceptance Rate - probabilità di riconoscere un utente sbagliato)
Tempo di Identificazione	<Circa 1 secondo (Match-on-device, senza elaborazione esterna)
Interfaccia di Comunicazione	UART
Compatibilità	Arduino, ESP32, Raspberry Pi
Costo	Basso-Medio (10-20 €)
Facilità di Integrazione	Alta, facile da configurare, supportato da molte librerie Arduino/ESP.
Affidabilità	Molto buona, adatto a usi frequenti, sensibile a dita sporche, tagli, umidità.

Tabella 3.1: Caratteristiche principali del sensore R307

Come possiamo notare dalla tabella 3.1, che specifica tutti i parametri operativi del sensore R307, questo dispositivo è ottimale per l'ambiente nel quale è stato pensato il progetto per 3 motivi principali:

1. **Capacità**: è dotata di una memoria flash al cui interno è possibile memorizzare fino a 1000 impronte digitali [5]. Ideale per una struttura che accoglie un gran numero di utenti.
2. **Compatibilità nativa con ESP32**
3. **Affidabilità per utilizzi intensivi**: essendo adatto ad usi frequenti cade a pennello con lo scopo del progetto, che prevede un uso significativo del sensore.

Ruolo del sensore nel sistema sviluppato

Il sensore R307 si occupa dell'acquisizione, del salvataggio e del confronto di impronte digitali. Come anticipato nel Capitolo 2.3, queste operazioni verranno eseguite all'interno del sensore, sfruttando un approccio `match-on-device`. L'unico componente a dialogare con il sensore è l'ESP32; quest'ultimo impartisce i comandi all'R307 tramite interfaccia UART, decidendo se aggiungere una nuova impronta, autenticarne una già esistente o rimuoverne una salvata.

3.1.3 Computer locale

L'ultimo componente hardware è rappresentato da un computer locale utilizzato esclusivamente per attività di gestione. Al suo interno risiedono il database conte-

nente i dati degli utenti e i loro ID associati alle impronte salvate nel lettore. Inoltre, il computer serve per poter utilizzare l'interfaccia grafica del programma, che permette all'operatore di registrare, eliminare o aggiornare voci nel database e inviare comandi all'ESP32.
La comunicazione tra computer ed ESP32 avviene tramite un collegamento diretto via cavo USB, che consente sia il trasferimento dei comandi seriali sia l'eventuale programmazione del microcontrollore.

3.2 Componenti software

3.2.1 PlatformIO

PlatformIO è un ambiente di sviluppo integrato (IDE) gratuito e open source, ideale per lo sviluppo di sistemi embedded. È integrabile con l'editor Visual Studio Code e supporta una vasta gamma di microcontrollori, tra cui l'ESP32.
In questo progetto è stato utilizzato come ambiente di sviluppo per l'implementazione del firmware dell'ESP32, sfruttando la flessibilità del linguaggio di programmazione C++ e la facile integrazione di molte librerie per la gestione del sensore R307.

3.2.2 XAMPP

XAMPP è una piattaforma che permette di avviare un server locale sul proprio computer, oltre ad essere gratuito ed open source, include:

- Apache: è un software server HTTP che consente di fornire contenuti web in risposta alle richieste dei client.
- MySQL: è il sistema di gestione di database relazionali, utilizzato per memorizzare e gestire i dati nelle tabelle.
- PHP: utilizzato per gestire la logica applicativa di un sito o un servizio web e per generare pagine web dinamiche.

In questo progetto è stato utilizzato XAMPP per avviare il server Apache e il database MySQL sulla macchina locale, così che il backend possa essere gestito senza connessione Internet.
Inoltre, sono presenti dei file PHP che consentono l'interazione tra l'interfaccia grafica dell'utente (GUI) e il database; essi devono risiedere all'interno della directory del server Apache. In definitiva, PHP riceve richieste HTTP e le traduce in operazioni SQL sul database.

3.2.3 Qt Creator

Qt è un framework di sviluppo di applicazioni multipiattaforma, implementato come una libreria in C++. Viene fornito come una raccolta di DLL o librerie condivise.

Qt Creator è l'ambiente di sviluppo ufficiale fornito da Qt. Viene utilizzato principalmente per la progettazione di GUI, ma offre anche strumenti per la gestione del progetto. Grazie a questa risorsa è stata sviluppata un'applicazione Desktop che consente all'operatore di interagire con il sistema in modo semplice ed intuitivo. La GUI funge da ponte tra l'operatore e l'ESP32, oltre ad offrire funzionalità di consultazione e aggiornamento del database.

3.3 Comunicazione tra componenti

I quattro macro-componenti da tenere in considerazione per capire la logica del funzionamento sono: ESP32, R307, Database e la GUI. Suddividiamo il tipo di operazioni che il programma può eseguire in due gruppi:

- Operazioni che coinvolgono sia il database che le impronte:
  1. Registrazione di un utente.
  2. Accesso di un utente.
  3. Rimozione di un utente.
  4. Rimozione di tutti gli utenti.
- Operazioni di gestione del database:
  1. Visualizzazione e modifica dinamica del database dalla GUI.

Entrambi i tipi di operazioni partono dalla GUI, ma solo il primo gruppo passa per il microcontrollore.

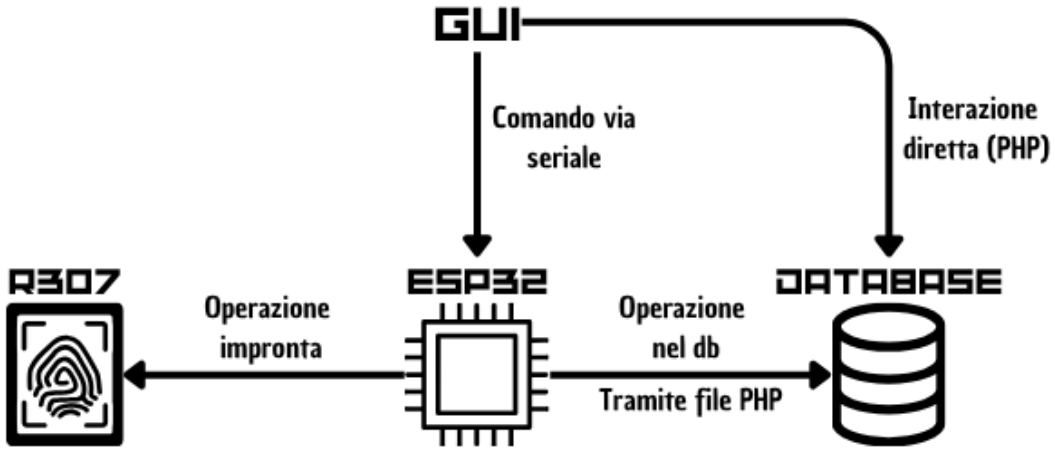

Figura 3.4: Schema della comunicazione tra i componenti principali del sistema

Quando un'operazione coinvolge sia il database che il sensore di impronte digitali, il comando viene inviato dalla GUI all'ESP32 tramite la porta seriale. Quest'ultimo gestisce poi la comunicazione con il sensore R307 (tramite UART) e con il database (tramite script PHP).
Si consideri il caso in cui deve essere eseguita la fase di registrazione di un utente, la GUI invia un comando all'ESP32 come `REGISTRA` insieme ai parametri necessari (`NOME;COGNOME;STANZA`). A questo punto il microcontrollore registra l'impronta nel sensore e i dati dell'utente nel database, utilizzando una richiesta HTTP verso i file PHP 3.4. Per quanto riguarda il secondo gruppo di operazioni, invece, vengono richiamati i file PHP direttamente dalla GUI. Questi servono per restituire il contenuto del database e per gestirlo dinamicamente.

3.3.1 Considerazioni sulla sicurezza dell'ID utente

Per avere un quadro più completo della memorizzazione di un utente all'interno del database e all'interno del sensore, prendiamo l'esempio di un utente già memorizzato al suo interno come in figura 3.5.

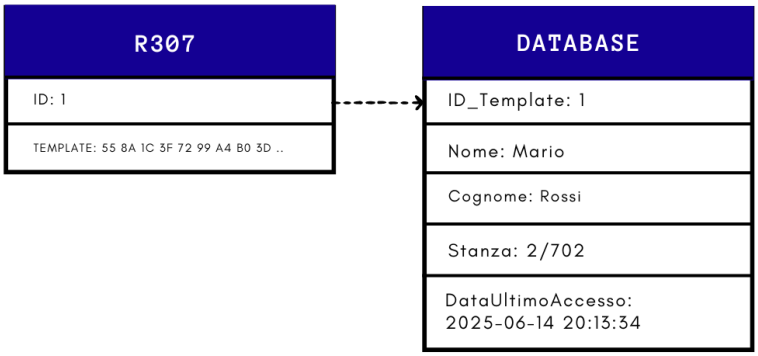

Figura 3.5: Memorizzazione di un utente

Un utente memorizzato all'interno del sistema vede divisi i suoi dati personali all'interno del database con i suoi dati sensibili all'interno del sensore (come già motivato all'interno della Sezione 2.3). Per far corrispondere i due gruppi di dati viene utilizzato l'Id che il sensore assegna automaticamente all'impronta dell'utente, al momento della registrazione, al suo interno. Venendo in possesso di entrambi gli elementi allora, tramite una semplice associazione tra ID del sensore e ID_Template del Database, si potrebbe risalire all'identità dell'utente corrispondente a quel determinato template biometrico.

Nel contesto di questo progetto l'utilizzo di funzioni hash risulterebbe inappropriato poiché l'ID numerico inserito nel sensore deve essere riconducibile all'ID presente nel database e l'hash, per definizione, non è reversibile.

È importante sapere che il sensore offre una possibilità di abilitare un livello di sicurezza più elevato configurando manualmente un parametro chiamato `Security Level`, che consente di attivare una cifratura interna dei dati. Tuttavia, secondo il manuale tecnico ufficiale del dispositivo R307 [1], questa configurazione deve essere fatta esplicitamente dall'utente e, seppur aumentando la sicurezza, non garantisce un livello di protezione avanzato poiché le chiavi sono gestite internamente e non sono personalizzabili. Inoltre, il produttore non fornisce specifiche dettagliate sugli algoritmi o più in generale sul meccanismo di protezione.

3.4 Funzionamento generale

3.4.1 Collegamenti Hardware

I collegamenti hardware del progetto si limitano alla connessione tra l'ESP32 e il sensore R307.

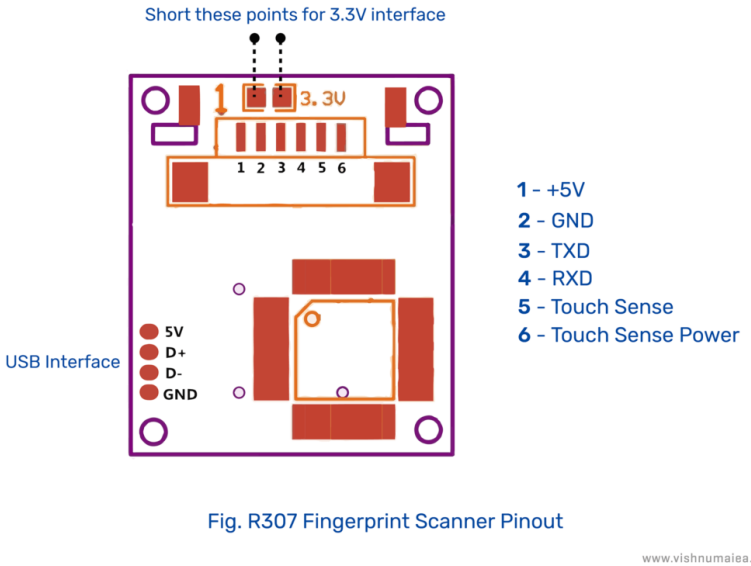

Figura 3.6: Pin del sensore R307

Prendendo in riferimento la Figura 3.6, che mostra come sono configurati di fabbrica i pin del sensore R307, e la Figura 3.1 che mostra tutti i pin del microcontrollore, i collegamenti hardware sono stati eseguiti seguendo la tabella 3.2.

Pin del Sensore R307	Collegamento ESP32
VCC	Pin **5V**
GND	Pin **GND**
TXD	Pin **RX (GPIO16)**
RXD	Pin **TX (GPIO17)**
TOUCH	Pin **GPIO18**
3.3V	**None**

Tabella 3.2: Collegamenti tra il sensore R307 e l'ESP32

N.B. Visto che il sensore si interfaccia con il microcontrollore, che opera a 5V, il pin numero 6 (3,3V) è bene lasciarlo scollegato per evitare eventuali problemi.

Una volta effettuato questo semplice cablaggio non rimane altro che collegare l'ESP32 al PC tramite un semplice cavo USB.

3.4.2 Flusso operativo del sistema

Contesto operativo e modalità di acquisizione

Facendo riferimento al contesto per il quale è stato sviluppato questo sistema, è prevista la presenza di un operatore autorizzato con accesso al software. Gli studenti che devono autenticarsi o registrarsi si presentano allo sportello, e l'operatore, tramite la GUI, invia i comandi al sensore per eseguire l'operazione desiderata.

Nel caso della registrazione, il programma prevede un doppio inserimento dell'impronta digitale al fine di minimizzare gli errori nella creazione del template.

È importante sottolineare come il sensore di impronte digitali sia fisicamente accessibile agli studenti che dovranno utilizzarlo, ma il suo funzionamento è totalmente controllato dall'operatore. Quest'ultimo deve essere l'unica figura ad aver accesso al software e, dunque, al database.

A. Inizializzazione

1. L'ESP32 viene collegato al computer, si avvia e inizializza il sensore di impronte digitali, verificando se i collegamenti sono stati eseguiti correttamente.

2. Il programma si mette in attesa finché non vengono inviati comandi tramite porta seriale.

B. Invio comandi tramite GUI

1. Tramite l'interfaccia grafica realizzata con Qt Creator, il personale può decidere quale operazione eseguire. Di seguito una tabella che abbina ad ogni operazione il relativo comando che verrà inviato all'ESP32 tramite porta seriale:

Operazione	**Comando inviato**
Registrazione utente	`REGISTRA;Nome;Cognome;Stanza\n`
Accesso	`ACCEDI\n`
Eliminazione utente specifico	`ELIMINA;ID\n`
Eliminazione di tutte le impronte	`PULISCI\n`
Navigazione (torna indietro)	`INDIETRO\n`

Tabella 3.3: Comandi inviati all'ESP32 tramite la GUI Qt

C. Risposta del server

1. L'ESP32 riceve il comando, lo interpreta ed esegue:
   - Operazioni locali al sensore di impronte digitali.
   - Invio dei dati al server PHP tramite HTTP per aggiornare il database,
2. Il server invia una risposta all'ESP32 che a sua volta lo invia alla GUI, questo messaggio di risposta serve all'operatore come feedback per l'operazione che ha mandato in esecuzione.

Capitolo 4

Installazione e configurazione del sistema

Per poter utilizzare il programma e applicarlo in qualsiasi ambiente, sono necessarie alcune operazioni preliminari per la configurazione dei componenti hardware e software.

4.1 Posizionamento del progetto

La struttura della directory ProgettoBiometrico deve già contenere le seguenti sottodirectory:

- **ESP32**: contiene il progetto PlatformIO con il firmware inserito nel microcontrollore.
- **GUI_QT**: contiene il file eseguibile Qt e tutte le librerie necessarie all'esecuzione.
- **Server\htdocs\backend**: contiene i file PHP che permettono di dialogare con il server.

Inoltre è presente anche una relazione puramente tecnica sul funzionamento del sistema chiamata `ProgettoBiometrico.pdf`.

L'intera cartella `ProgettoBiometrico` deve essere copiata direttamente all'interno del disco locale `C:\`.
Ottenendo, in questo modo, il seguente percorso: `C:\ProgettoBiometrico\`

4.2 Installazione di XAMPP e configurazione del database

1. Scaricare ed installare XAMPP dal suo sito ufficiale.

2. Aprire il pannello di controllo XAMPP e avviare i servizi Apache e MySQL.
3. Accedere a phpMyAdmin tramite `http://localhost`
4. Creare un nuovo database con il nome: `gestione_impronte`
5. All'interno di tale database, creare la tabella `studenti` con la seguente struttura:

Campo	Tipo
ID_Template	INT(11)
Nome	VARCHAR(20)
Cognome	VARCHAR(20)
Stanza	VARCHAR(20)
DataUltimoAcccesso	TIMESTAMP

Tabella 4.1: Struttura tabella studenti

`N.B.` Il campo DataUltimoAccesso deve essere impostato con valore di default `CURRENT_TIMESTAMP`, questo campo verrà automaticamente aggiornato ad ogni accesso.

La struttura del Database deve apparire come in Figura 4.1.

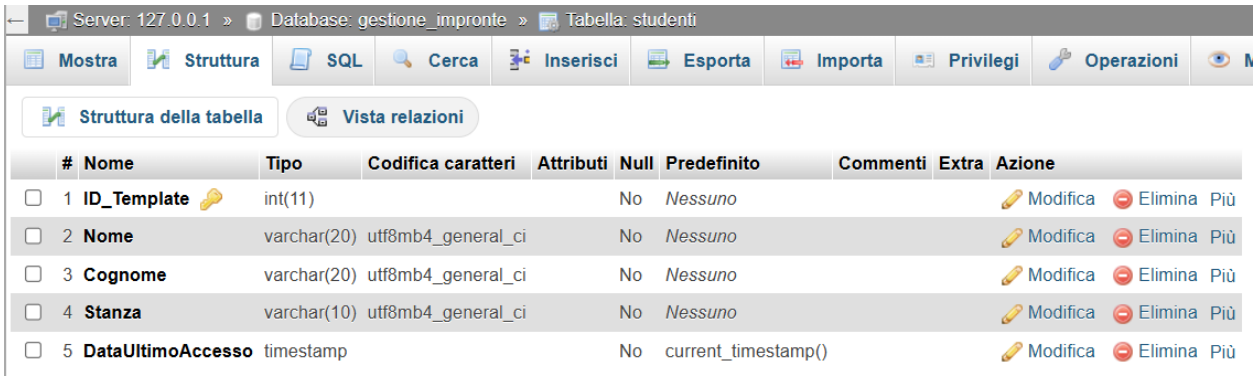

Figura 4.1: Struttura del Database

4.2.1 Configurazione di Apache

Una volta installato XAMPP, Apache ha una cartella predefinita come localhost che si trova nel seguente percorso: `C:\xampp\htdocs`.
Per fare in modo che Apache punti alla cartella corretta di questo progetto, è necessario modificare il file di configurazione principale:

1. Aprire il file: `C:\xampp\apache\conf\httpd.conf`.

2. Cercare le seguenti righe:

```
DocumentRoot "C:/xampp/htdocs"
<Directory "C:/xampp/htdocs">
```

3. Sostituirle con:

```
DocumentRoot "C:/ProgettoBiometrico/Server/htdocs"
 <Directory "C:/ProgettoBiometrico/Server/htdocs">
```

4. Salvare il file `httpd.conf` e riavviare Apache dal pannello XAMPP.

4.3 Configurazione ESP32

Per configurare correttamente il firmware dell'ESP32 bisogna:

1. Accedere al codice del microcontrollore tramite il percorso:
   `C:\ProgettoBiometrico\ESP32` e modificare:

   - Le credenziali corrette del WiFi a cui si desidera collegarsi, modificando le seguenti righe:

     ```
     // Configurazione WiFi
     const char* ssid = "NOME_RETE_WIFI";
     const char* password = "PASSWORD_WIFI";
     ```

   - L'indirizzo IP della macchina sulla quale gira il programma tramite la seguente riga:

     ```
     String serverBase = "http://0.0.0.0/backend/";
     ```

     Questa riga definisce l'indirizzo del server a cui l'ESP32 invia le richieste HTTP. È fondamentale sostituire l'indirizzo IP 0.0.0.0 con l'indirizzo IP effettivo del computer su cui è in esecuzione il server Apache/XAMPP. Si consiglia l'assegnazione di un IP statico al computer per evitare problemi di connettività nel tempo. A questo punto basta caricare il firmware sull'ESP32 utilizzando gli IDE ad esso compatibili (come PlatformIO o, più semplicemente, Arduino IDE).

   Completata la configurazione, il sistema è pronto per l'avvio.

4.4 Avvio del programma

1. Collegare l'ESP32 al computer mediante cavo USB.

2. Aprire il pannello di controllo XAMPP e avviare Apache e MySQL, come in Figura 4.2.

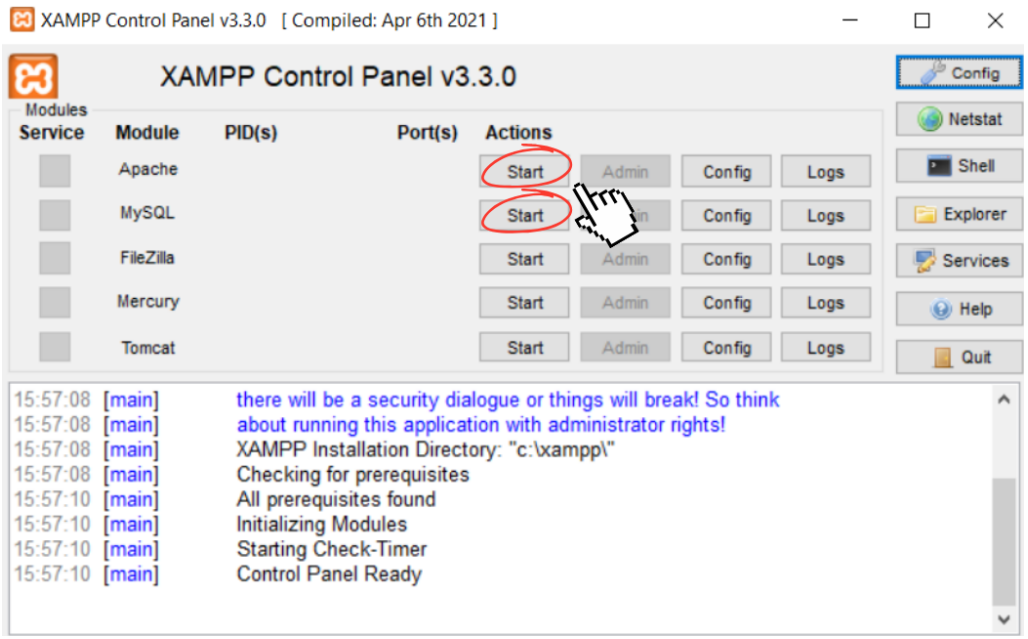

Figura 4.2: XAMPP Control Panel

3. Eseguire l'applicazione GUI disponibile nel percorso:
`C:\ProgettoBiometrico\GUI_Qt\FingerprintGUI.exe`

4. Utilizzare l'interfaccia grafica per registrare nuovi utenti, verificare le presenza o modificare il database.

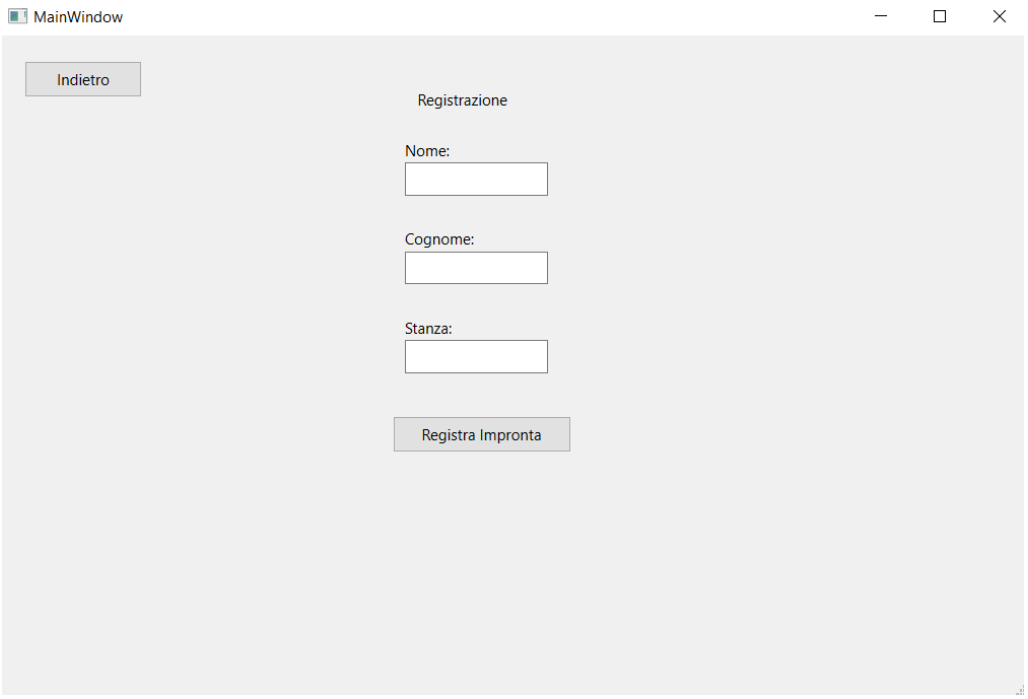

Figura 4.3: Schermata di registrazione

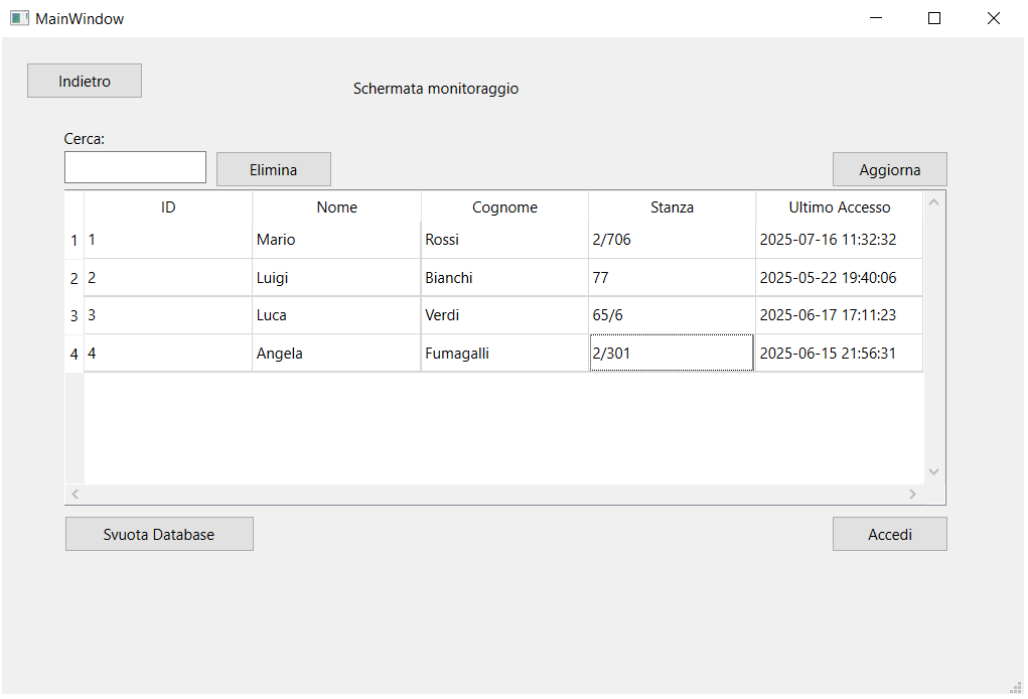

Figura 4.4: Schermata monitoraggio del database

Capitolo 5

Conclusione ed eventuali sviluppi futuri

In conclusione, il sistema sviluppato rappresenta un prototipo funzionale ed efficace per il monitoraggio della presenza effettiva negli alloggi studenteschi. La sua struttura, sia hardware che software, lo rende facilmente adattabile a contesti differenti, ovunque sia utile verificare la presenza fisica delle persone.

L'integrazione tra i diversi componenti ha consentito la realizzazione di una soluzione economica, autonoma e gestibile localmente, dimostrando come, anche con risorse limitate, sia possibile sviluppare progetti IoT utili e concreti.

5.1 Criticità incontrate e sviluppi futuri

La difficoltà nel coordinare GUI ed ESP32 è stato lo scoglio principale del progetto, entrambi comunicano con il database ma per finalità differenti. Questa duplice gestione ha aumentato la complessità dell'architettura.

Uno sviluppo significativo potrebbe consistere nell'affidare l'accesso al database esclusivamente alla GUI e lasciare all'ESP32 solo il compito di comunicare con il sensore biometrico. Questa centralizzazione dell'accesso al database ridurrebbe la complessità architetturale del sistema, lasciando la gestione del database interamente al software desktop.

Un'ulteriore miglioramento futuro riguarda la configurazione dell'indirizzo IP del server. Nel firmware dell'ESP32 non possiamo richiamare il `localhost`, poiché lo interpreterebbe come se stesso e non farebbe riferimento al computer con il quale è collegato. Attualmente l'indirizzo IP deve essere inserito manualmente nel firmware del microcontrollore, risultando poco pratico in assenza di un IP statico. Dunque una possibile soluzione potrebbe essere far sì che la GUI rilevi automaticamente, all'avvio del software, l'indirizzo IP della macchina e lo comunichi via seriale all'ESP32. Questo eliminerebbe la necessità di modifiche manuali, rendendo il sistema più flessibile e facilmente distribuibile.

Sicurezza dati sensibili

Alla luce delle considerazioni emerse nella Sezione 3.3.1 riguardante la sicurezza dei dati biometrici, è possibile scegliere se affidare la cifratura dei template al sensore stesso - delegando la sicurezza all'azienda produttrice - oppure sviluppare un algoritmo crittografico personalizzato, ottenendo un maggiore controllo e una maggiore garanzia della sua efficacia.

Sicurezza del database

Sebbene il sistema sia progettato per funzionare in locale, la protezione dei dati nel database è fondamentale per quanto riguarda l'affidabilità e la resilienza. Un guasto hardware o un errore accidentale può compromettere la disponibilità delle informazioni registrate.

Per questo motivo, potrebbe essere ragionevole considerare l'implementazione di un meccanismo di duplicazione del server, che consiste nella creazione di una copia periodica del database e dell'ambiente di esecuzione. Esistono diversi modi per effettuare questa duplicazione, la scelta ricade in base al contesto nel quale ha vita il sistema:

- **Backup manuale o automatico del database**: si può effettuare un backup manuale tramite strumenti come phpMyAdmin. Se è necessario un backup periodico allora possiamo automatizzarlo pianificandolo ad intervalli regolari.
- **Clonazione del server**: si effettua una clonazione delle directory principali, quali `xampp` e `ProgettoBiometrico`, allocandole in un secondo computer locale (o comunque un dispositivo esterno).
- **Utilizzo di una macchina virtuale**: questa opzione è la più robusta poiché si va a configurare l'intero sistema all'interno di una macchina virtuale. Questo ci da la possibilità di creare snapshot o copie della macchina stessa per un ripristino completo.

Queste sono le principali soluzioni adottabili per contribuire all'aumento dell'affidabilità - e della sicurezza - del sistema.

Bibliografia

[1] R307 fingerprint module user manual. https://www.openhacks.com/uploadsproductos/r307_fingerprint_module_user_manual.pdf, 2015.

[2] Parlamento Europeo e Consiglio dell'Unione Europea. Regolamento (ue) 2016/679 del parlamento europeo e del consiglio del 27 aprile 2016 relativo alla protezione delle persone fisiche con riguardo al trattamento dei dati personali, nonché alla libera circolazione di tali dati (general data protection regulation - gdpr), 2016.

[3] Garante per la protezione dei dati personali. Utilizzo di sistemi biometrici per il controllo degli accessi e la rilevazione delle presenze: il garante detta le regole per le scuole e i datori di lavoro. https://www.garanteprivacy.it/home/docweb/-/docweb-display/docweb/9995680, Febbraio 2024.

[4] IBM. Che ccos'è l'autenticazione biometrica? https://www.ibm.com/it-it/think/topics/biometric-authentication, 2023.

[5] Vishnu Mohanan. Interfacing r307 optical fingerprint scanner with arduino boards for biometric authentication. https://www.circuitstate.com/tutorials/interfacing-r307-optical-fingerprint-scanner-with-arduino-boards-for-biometric-authentication, 2021.

Ringraziamenti

Voglio ringraziare i miei genitori per avermi dato la fiducia necessaria per portare a termine questo progetto, per avermi supportato in questo percorso e per avermi insegnato i valori importanti della vita di una persona. Mi avete insegnato che anche i momenti più bui hanno una fine e che insieme è più facile superarli.

A mio fratello che è sempre stato il mio punto di riferimento, ti ho sempre copiato fin da piccolo su qualsiasi cosa e non lo rimpiango affatto. Sei una certezza, una persona sulla quale sono certo che posso contare in caso di bisogno.

Agli amici di Centobuchi e dintorni, amicizie così salde nel tempo possono diventare scontate, ma la nostra non lo è mai stata e di questo vi ringrazio. Dopo tutto questo tempo potrei ringraziare ognuno per un motivo diverso, ma non lo farò perché non me ne va. Forza samba.

Al 700, che più che un semplice alloggio è diventato una vera e propria famiglia, vi ringrazio per tutti i momenti indelebili che mi avete lasciato, senza di voi l'università non sarebbe mai stata la stessa.

Un ringraziamento va anche a tutte le amicizie create durante questo percorso di studi, ognuna a suo modo ha contribuito alla realizzazione di questo traguardo.

Al gruppo degli Italeñi, avete reso il mio Erasmus incredibile, ognuno di voi mi ha lasciato qualcosa a suo modo, non avrei potuto chiedere compagni migliori e per questo vi ringrazio davvero tanto.