

SMART LIBRARY

Bianco Michael, Bertolini Matteo, Todaro Marco

Introduzione

Smart Library è l'evoluzione del concetto di biblioteca, è un sistema distribuito che consente la gestione completamente autonoma e delocalizzata dei prestiti di libri.

Grazie alla sua apposita applicazione ognuno è in grado di prendere in prestito un libro, di leggerlo e di riconsegnarlo nel punto a lui più vicino, oltre ad ottenere suggerimenti sui prossimi libri da leggere.

I totem sparsi in varie parti di una città forniscono all'utente comodità nel ritiro e nella consegna, oltre a garantire un continuo ricambio dei libri.

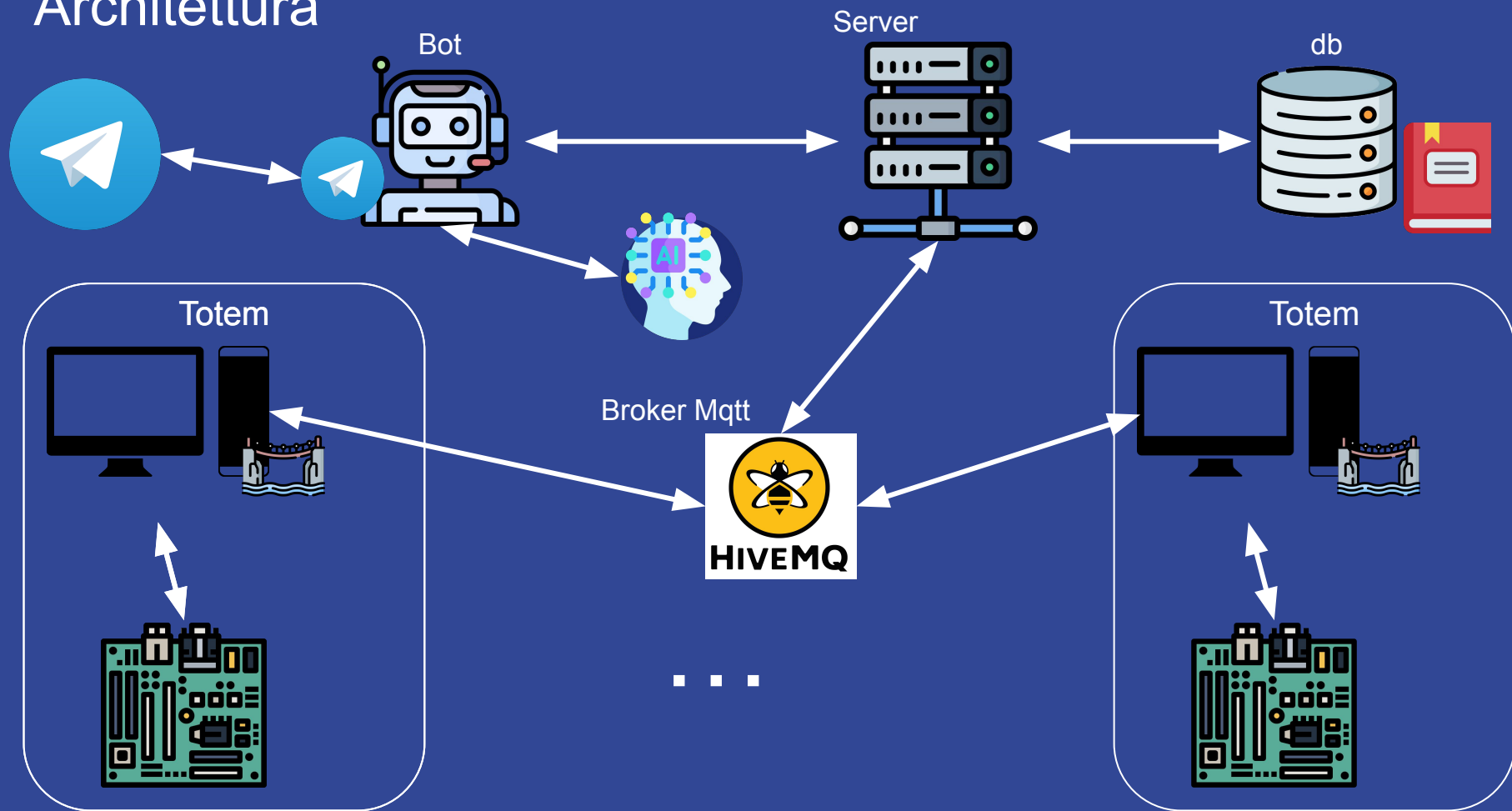
Punti di forza

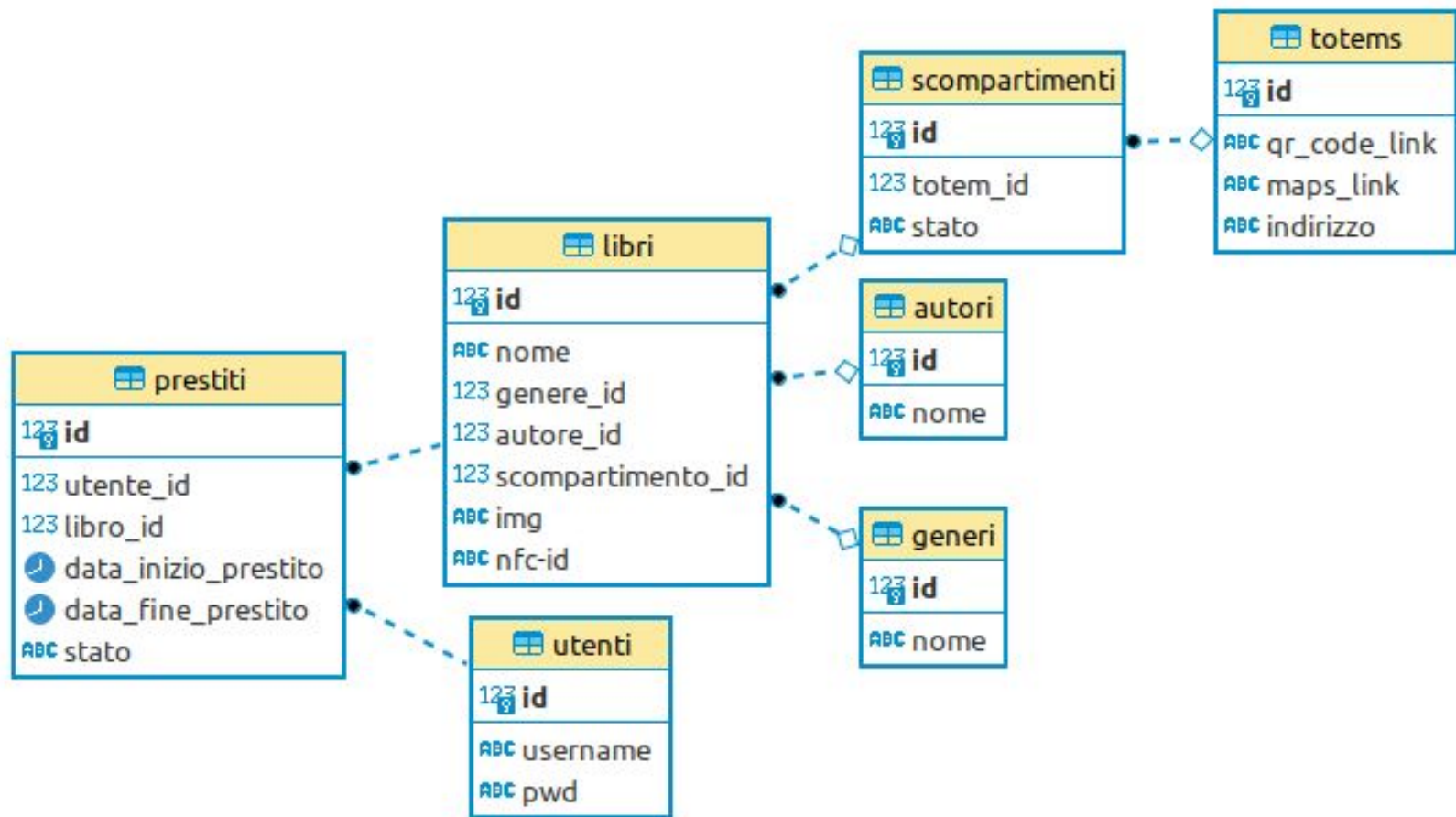
- **Comodità:** una volta finito di leggere un libro un utente può riconsegnarlo nel totem più vicino a lui
- **Condivisione:** persone o enti possono decidere di fornire gratuitamente libri da leggere e diffondere maggiormente la cultura nel proprio paese
- **Semplicità:** l'utente può prenotare il libro da leggere attraverso il suo smartphone ed eseguire tutte le operazioni semplicemente utilizzando il suo codice di prenotazione, senza bisogno di autenticarsi nuovamente ad ogni step

Punti di forza

- **Scalabilità:** si possono aggiungere nuovi totem e nuovi libri senza dover cambiare l'architettura del sistema o riprogrammare alcune parti.

Architettura





Digital Twin

Il Digital Twin di un totem è rappresentato dai suoi dati (ID, indirizzo, link a Google Maps), dagli scompartimenti in cui è diviso e dai libri contenuti al loro interno.

Ogni Digital Twin è accessibile con una GET all'indirizzo
https://.../totems/ID_TOTEM

BOT: prenotazione di un libro

/prenota Nome_Libro

1. GET HTTP a *https://.../totems?nomeLibro=NOME_LIBRO* per verificare la disponibilità del libro
2. Se il libro è disponibile fa una POST a *https://.../prenotazioni* per creare una nuova prenotazione. Ottiene in risposta il numero della prenotazione, l'indirizzo del totem che contiene il libro e il link a Google Maps
3. Stampa a video un riepilogo della prenotazione e la copertina del libro

BOT: visualizzazione libri in un totem

/totem Numero_Totem

1. GET HTTP a *https://.../totems/NUMERO_TOTEM* per ottenere i dati del totem, i libri contenuti al suo interno e la loro disponibilità
2. Viene mostrato a video l'elenco dei libri contenuti nel totem, seguiti dalla loro disponibilità

BOT: ricerca totem in cui consegnare un libro

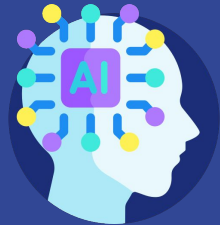
/consegna

1. GET HTTP a *https://.../totems/scompartimentolibero* per ottenere i dati dei totem con almeno uno scompartimento libero in cui è possibile consegnare un libro
2. Viene mostrato a video l'elenco dei totem con almeno uno scompartimento libero, insieme all'indirizzo e al link Google Maps dei totem

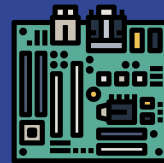
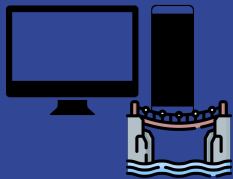
BOT: consiglio prossimo libro da leggere

/consigliami Ultimo_Libro_Letto?

1. GET HTTP /prenotazioni/last/libro?utente='nomeutente' (per i nuovi utenti è obbligatorio selezionare l'ultimo libro letto)
2. GET HTTP al servizio *data.readow.ai*, intelligenza artificiale in grado di suggerire nuovi libri da leggere, dati in input gli ultimi libri letti.
3. Vengono mostrati a video tre libri consigliati, con relative copertine
4. In questo modo l'utente potrà prenotare il suo prossimo libro



Prenotazione



1. Notifica avvenuta
prenotazione



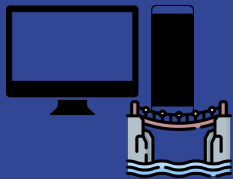
2. Leggi l'id dello
scompartimento e
comunicalo al totem



3. Leggi id scompartimento
e accendi il led rosso per
riservare il libro



Ritiro

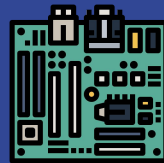


Ritira

1. Avvio richiesta di ritiro



2. Controllo
prenotazione e
abilitazione al ritiro



3. Attendi conferma ritiro

4. Leggi NFC e conferma
ritiro

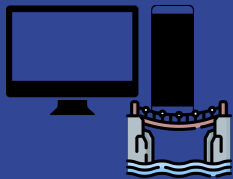


5. Ritiro avvenuto, invio
conferma al server



6. Aggiorna prenotazione

Consegna



Consegna

1. Avvio richiesta di consegna



2. Controllo prenotazione e abilitazione alla consegna

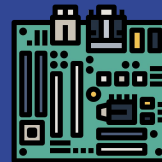


3. Leggi id scompartimento e attendi conferma consegna

5. Consegna avvenuta, invio conferma al server



6. Aggiorna prenotazione



4. Leggi NFC e conferma consegna



Pacchetti MQTT

Pacchetto MQTT da Server a Bridge

Topic: TOTEMS / ID_TOTEM

Messaggio: NFC_LIBRO / ID_SCOMPARTIMENTO / AZIONE / ID_PRENOTAZIONE

L'azione può essere quella di PRENOTAZIONE, RITIRO o CONSEGNA



Pacchetto MQTT da Bridge a Server

Topic: TOTEMS / PRENOTAZIONI

Messaggio: ID_TOTEM / ID_PRENOTAZIONE / AZIONE / ID_SCOMPARTIMENTO

L'azione può essere quella di RITIRO o CONSEGNA



Pacchetti in Seriale

Pacchetto Seriale da Bridge ad Arduino



Messaggio: FF / ID_SCOMPARTIMENTO / AZIONE / NFC_LIBRO / FE

L'azione può essere quella di PRENOTAZIONE, RITIRO o CONSEGNA

Pacchetto Seriale da Arduino a Bridge



Messaggio: FF / ID_SCOMPARTIMENTO / AZIONE / FE

L'azione può essere quella di RITIRO, CONSEGNA, RICHIESTA STATO o ERRORE

Inizializzazione Totem

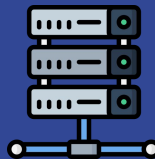
Richiesta HTTP da Bridge a Server

URL: (GET) `https://.../totems/ID_TOTEM`



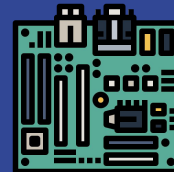
Risposta HTTP da Server a Bridge

Messaggio: JSON contenente i dati del totem, i libri contenuti al suo interno e lo stato dei suoi scompartimenti



Pacchetto Seriale da Bridge ad Arduino

Messaggio: `FF / STATO_SCOMPARTIMENTO1 / STATO_SCOMPARTIMENTO2 / ... / FE`



Pacchetti di Errore

Pacchetto MQTT da Server a Bridge

Topic: TOTEMS / ID_TOTEM

Messaggio: -1 / MESSAGGIO_ERRORE



Pacchetto Seriale da Arduino a Bridge

Messaggio: FF / ID_SCOMPARTIMENTO = 0 / AZIONE = 0 / FE



STRUMENTI UTILIZZATI

Gli strumenti utilizzati sono di natura hardware e software.

HARDWARE



Microcontrollore ARDUINO UNO per la gestione del totem e dei suoi componenti collegati.

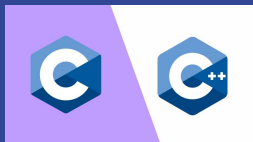
LETTORE NFC, componente attivo che legge le informazioni contenute in un sensore RFID/NFC posto nelle vicinanze. Fa uso della comunicazione SPI e per questo motivo usa, tra gli altri di alimentazione, i pin SDA-SCK-MISO-MOSI.



BUZZER passivo per riprodurre suoni di frequenza e durata differenti. Comandato con un digital pin di ARDUINO UNO.

STRUMENTI UTILIZZATI

SOFTWARE - linguaggi di programmazione



Linguaggi C e C++ per la programmazione dell'MCU, Arduino nel nostro caso.

Linguaggio Python per la programmazione del Bot Telegram, del Bridge e del Server MQTT.



Node JS, libreria di JavaScript per la programmazione delle parti backend dei sistemi. Usato per la creazione del Server HTTP e per la gestione delle richieste/risposte.

Postgre SQL, DBMS per la gestione del database programmato in linguaggio SQL.



PostgreSQL

STRUMENTI UTILIZZATI

SOFTWARE - IDE & repository



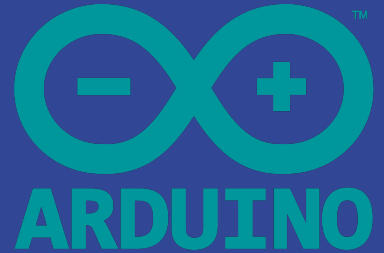
GitHub



Postman



Visual Studio Code



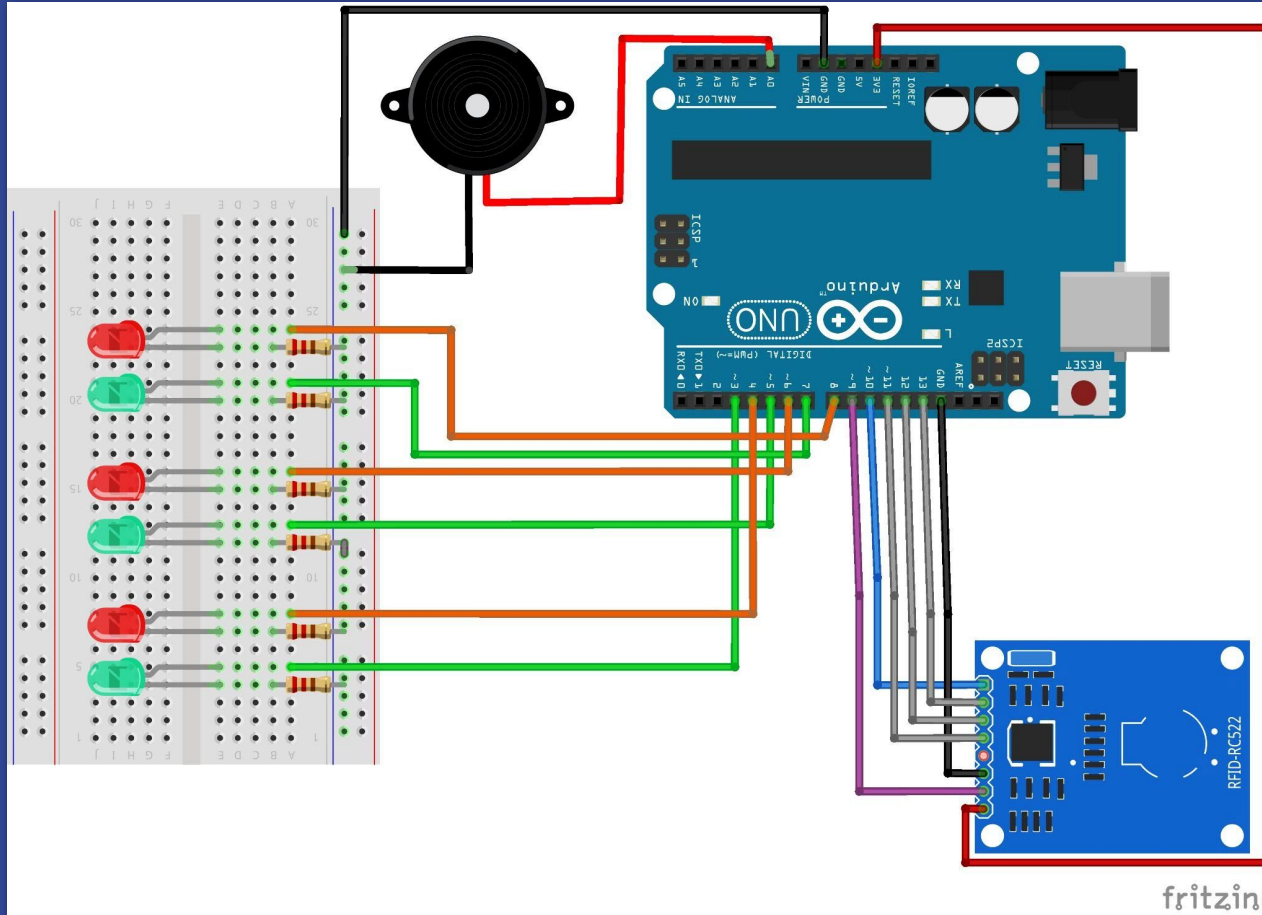
Arduino IDE

Repository



<https://github.com/Tod-dev/SmartLibrary-IOT>

SCHEMA DEL TOTEM



SMART LIBRARY

Grazie per l'attenzione