Cassaforte 2.0

Di

Piscitelli Federico

Leggio Giuseppe

Perego Daniele

Rosselli Mattia

Sommario

[Introduzione 2](#_Toc485485210)

[Obiettivi 3](#_Toc485485211)

[Strumenti 4](#_Toc485485212)

[Metodologia e sviluppo 5](#_Toc485485213)

[Beta test 10](#_Toc485485214)

[Funzionamento 11](#_Toc485485215)

[Ringraziamenti 12](#_Toc485485216)

[Bibliografia e sitografia 12](#_Toc485485217)

# Introduzione

L’ idea

L’ idea di creare una cassaforte smart è nata dall’ esigenza di creare un prodotto innovativo non ancora presente sul mercato. Un altro aspetto che ci ha spinto alla realizzazione di questo progetto è la possibilità di estendere il meccanismo di sicurezza ad altri ambiti, come ad esempio in quello della domotica.

Quali funzioni offre

Oltre ad essere una normale cassaforte, il nostro prototipo offre diverse funzionalità che rappresentano sicuramente delle innovazioni nel campo della sicurezza. Prime fra tutte:

* Sensore di impronte digitali
* Applicazione android
* Comunicazione bluetooth
* Registrazione con successiva consultazione tramite app

# Obiettivi

La Cassaforte 2.0 si pone come obiettivo quello di creare un progetto sicuro, affidabile e intuitivo, con l’aggiunta di alcune funzioni “smart”.

*INNOVAZIONE*

Questo progetto pone come primo obiettivo quello di trasformare un oggetto millenario in qualcosa di moderno, al passo con i tempi.

Il primo passo per la modernizzazione della cassaforte avviene tramite l’aggiunta di un sensore per il rilevamento dell’impronta digitale, diventato ormai un must della sicurezza.

Per rendere ancora più all’avanguardia questo progetto ci si è chiesto: “Quale oggetto oggi rappresenta l’innovazione?” Subito è saltato all’occhio lo smartphone che al giorno d’ oggi è parte integrante della nostra vita.

Il modo migliore per combinare questi due mondi è stato quello di sviluppare un’applicazione dedicata alla gestione della cassaforte.

Nonostante il progetto preveda l’utilizzo dello smartphone per alcune funzioni, la cassaforte può svolgere la sua normale funzione di custodia anche senza l’obbligo di utilizzo dell’applicazione, rendendo accessibile a tutti , grandi e piccoli, le sue funzioni, ampliando così il range di persone raggiungibili nel mercato.

*SICUREZZA*

Uno degli obiettivi fondamentali della Cassaforte 2.0 è quello di garantire una sicurezza costante ed efficiente. Per fare ciò il prototipo propone due livelli di sicurezza:

1. Tastierino numerico
2. Sensore d’ impronte digitali

Proprio grazie a questo doppio livello di sicurezza, anche scoperto il codice di apertura, risulta impossibile l’apertura in maniera fraudolenta della cassaforte.

# Strumenti

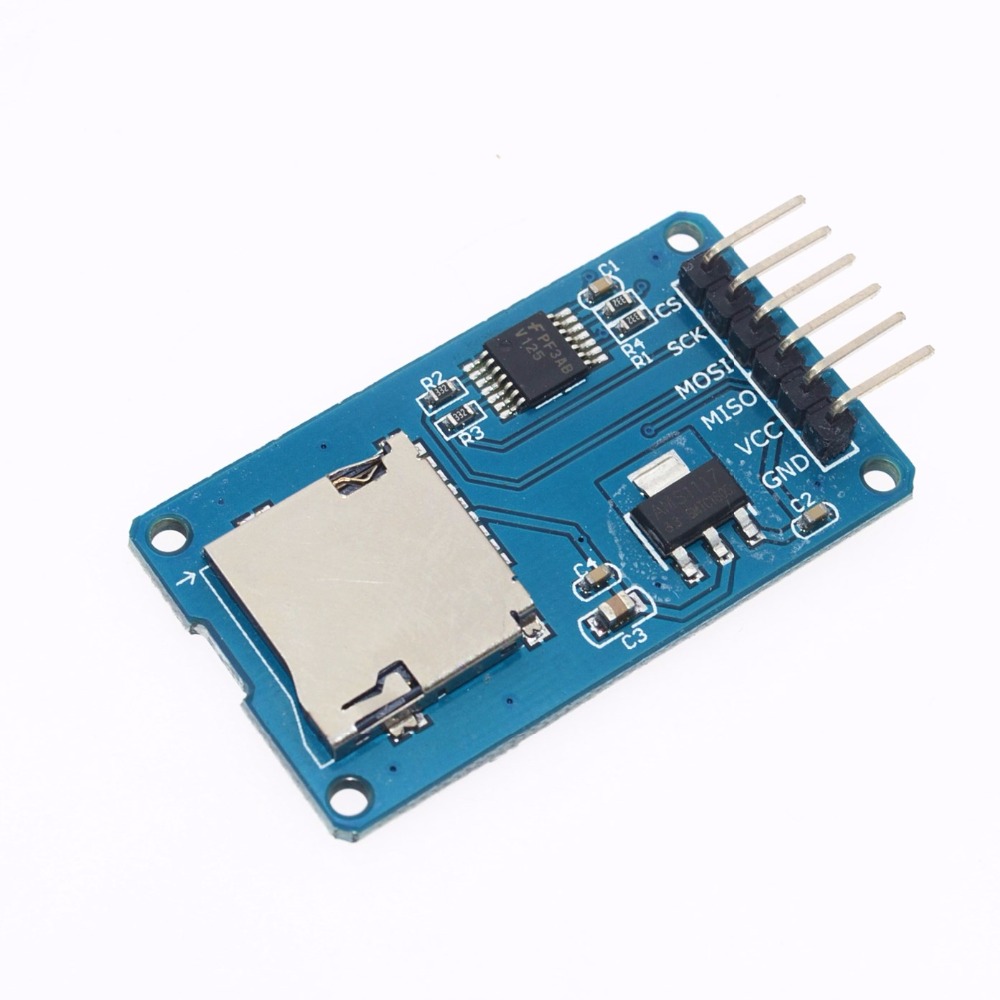
La Cassaforte 2.0 utilizza un Arduino che permette una completa adattabilità, data dalla possibilità di poter compilare il codice in base alle necessità del cliente, dando la possibilità di aggiungere o modificare la varia componentistica.

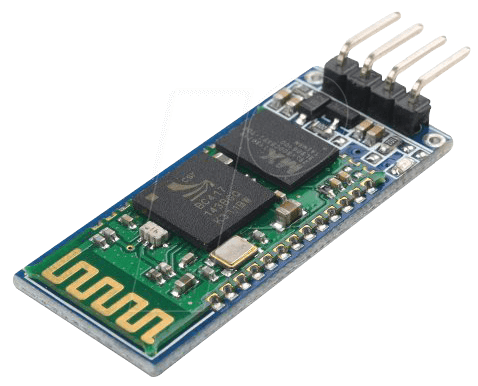
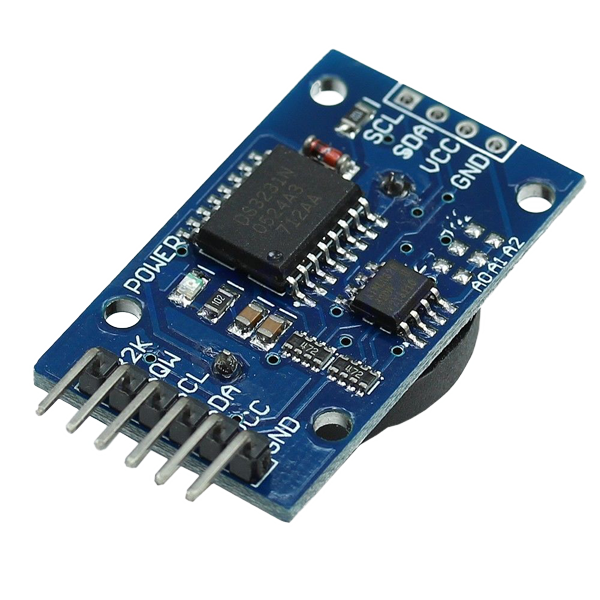
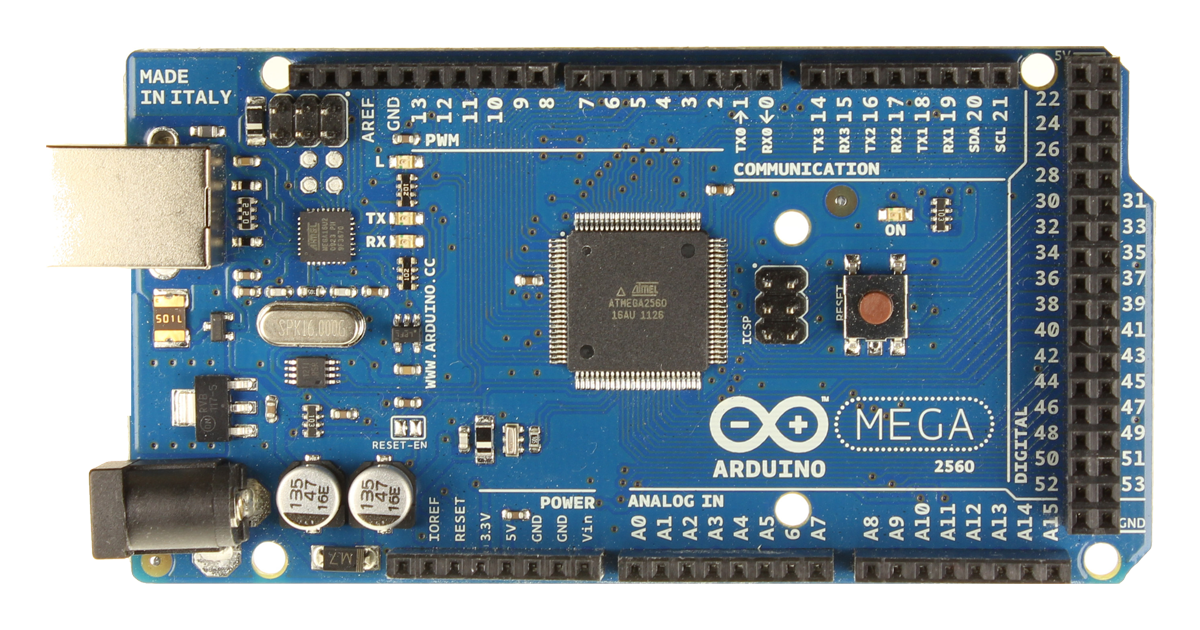
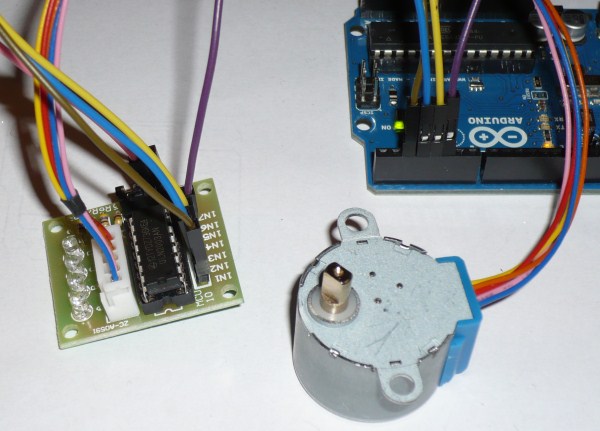
*STRUMENTI UTILIZZATI*

* Arduino Mega
* Modulo RTC (Real Time Clock, utile per tenere ora e giorno)
* Modulo bluetooth HC-06
* Modulo per l’ espansione della memoria
* Sensore d’ impronte digitale
* Schermo LCD
* Motorino Step to Step
* Cassaforte in legno per simulare un reale caso d’uso.
* Android Studio e Arduino IDE come ambienti di sviluppo per la programmazione.
* Librerie fornite dai vari costruttori dei componenti

Durante la fase di progettazione si è deciso di utilizzare un Arduino Mega, a dispetto di un Arduino UNO, per la posssibilità di scalabilità che offriva il primo, data la quantità di pin disponibili.







# Metodologia e sviluppo

Essendoci due ambienti di sviluppo differenti si è optato per la divisione del team in due rami.

Il primo gruppo, composto da Leggio e Piscitelli, si è impegnato nella programmazione e nello sviluppo dell’applicazione android.

Il secondo gruppo, composto da Perego e Rosselli, ha avuto come incarico il montaggio dei vari componenti hardware e la programmazione delle sue funzionalità.

Android

Lo sviluppo dell’ applicazione Android si è divisa in due parti:

* Progettazione e realizzazione della grafica
* Programmazione delle funzionalità dell’ applicazione

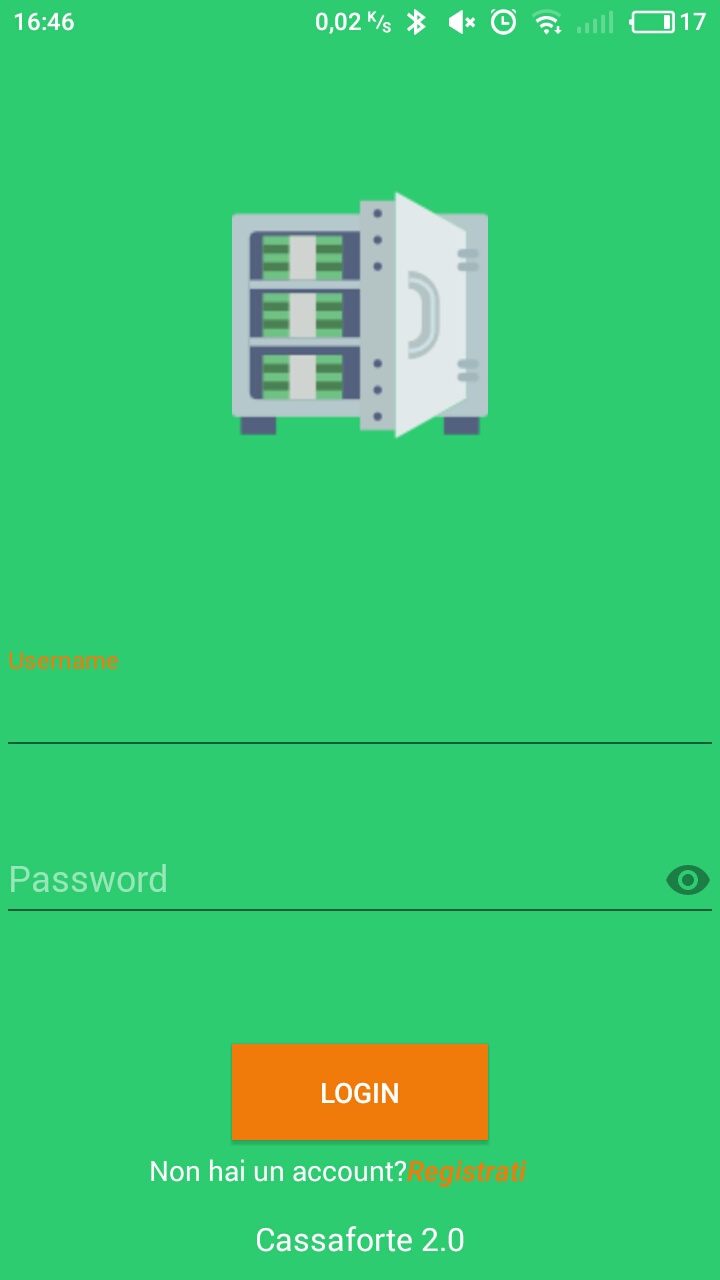
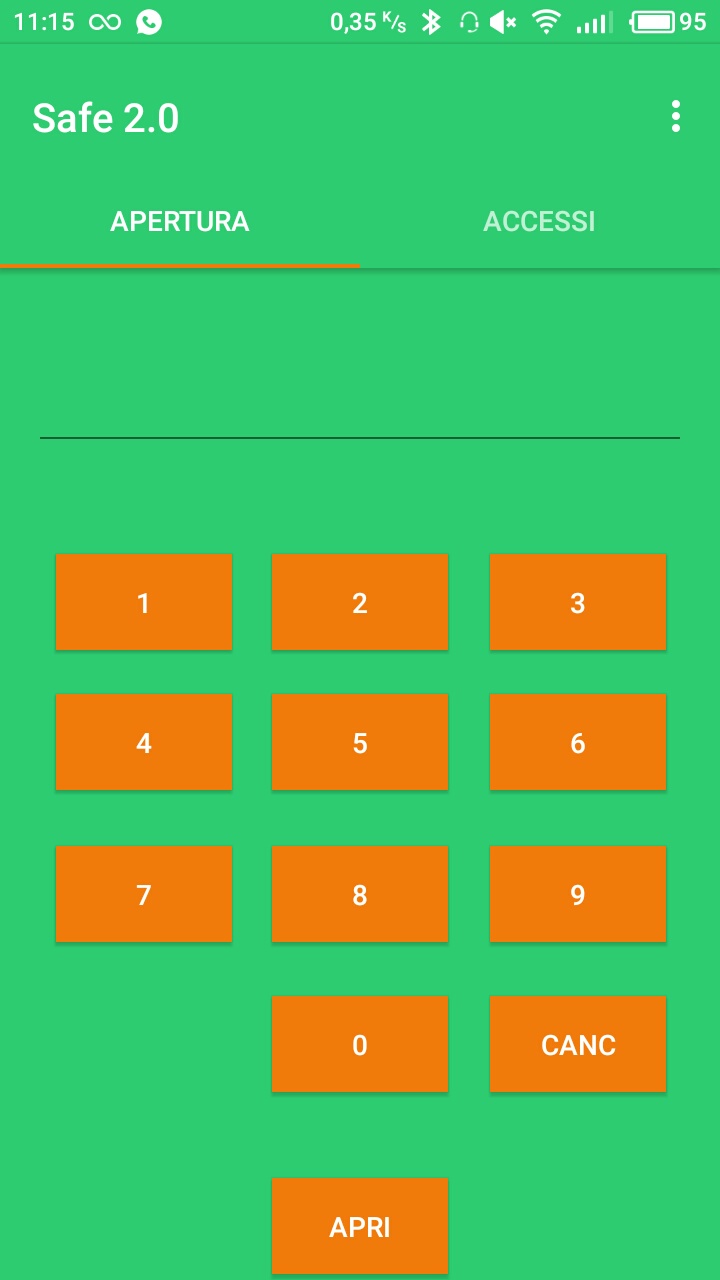
Durante la prima fase, si è realizzato un “bozzetto” in Android Studio di come sarebbe dovuta essere l’ applicazione. Si è passati poi alla creazione dei colori per i bottoni e per lo sfondo e solo successivamente sono state implementate tutte le animazioni nelle varie activity.

La parte più impegnativa a livello di design è arrivata quando si è dovuto implementare il Fragment. Abbiamo imparato che il Fragment è un tipo particolare di activity che permette di contenere al suo interno più “schermate”. Questo ne rende più difficile la gestione in quanto tutti i metodi per richiamare dei componenti grafici dal file xml sono differenti. Ad esempio per richiamare un bottone in un’ activity semplice si scrive questa linea di codice:

*Button btn = (Button) findViewById(R.id.button);*

Mentre in un Fragment:*View myFragmentView = inflater.inflate(R.layout.fragment, container,false);*

*Button btn = (Button) myFragmentView.findViewById(R.id.button);*

**

Successivamente si è passati allo sviluppo delle funzionalità dell’ applicazione. In primo luogo ci si è occupati della comunicazione tra la cassaforte e l’ applicazione, che ci ha portato via gran parte del tempo.

Per comunicare con la cassaforte utilizziamo la classe “BluetoothComIno”, creata da Leggio. Questa classe al suo interno contiene, l’ indirizzo MAC del modulo HC-06, l’ adapter per usare il bluetooth dello smartphone e una socket di tipo BluetoothSocket che a sua volta ha al suo interno due stream:

* InputStream per ricevere i dati dalla cassaforte
* OutputStream per inviare dati alla cassaforte

Questa classe viene utilizzata in tutte le activity, poichè in tutte si effettua una comunicazione con la cassaforte.

Durante la programmazione della comunicazione bluetooth sono stati riscontrati diversi problemi, soprattutto riguardanti la sincronizzazione. Molte volte infatti venivano persi o duplicati caratteri in fase di ricezione. Il problema deriva principalmente dal fatto che l’ Arduino veniva utilizzato spesso in modalità “Debug” e ciò creava degli scompensi nella comunicazione. Se l’ Arduino è collegato invece ad una fonte di alimentazione esterna, che non sia una porta USB del pc, questo problema svanisce.

arduino

Anche lo sviluppo di Arduino e dei componenti è stato diviso in due momenti:

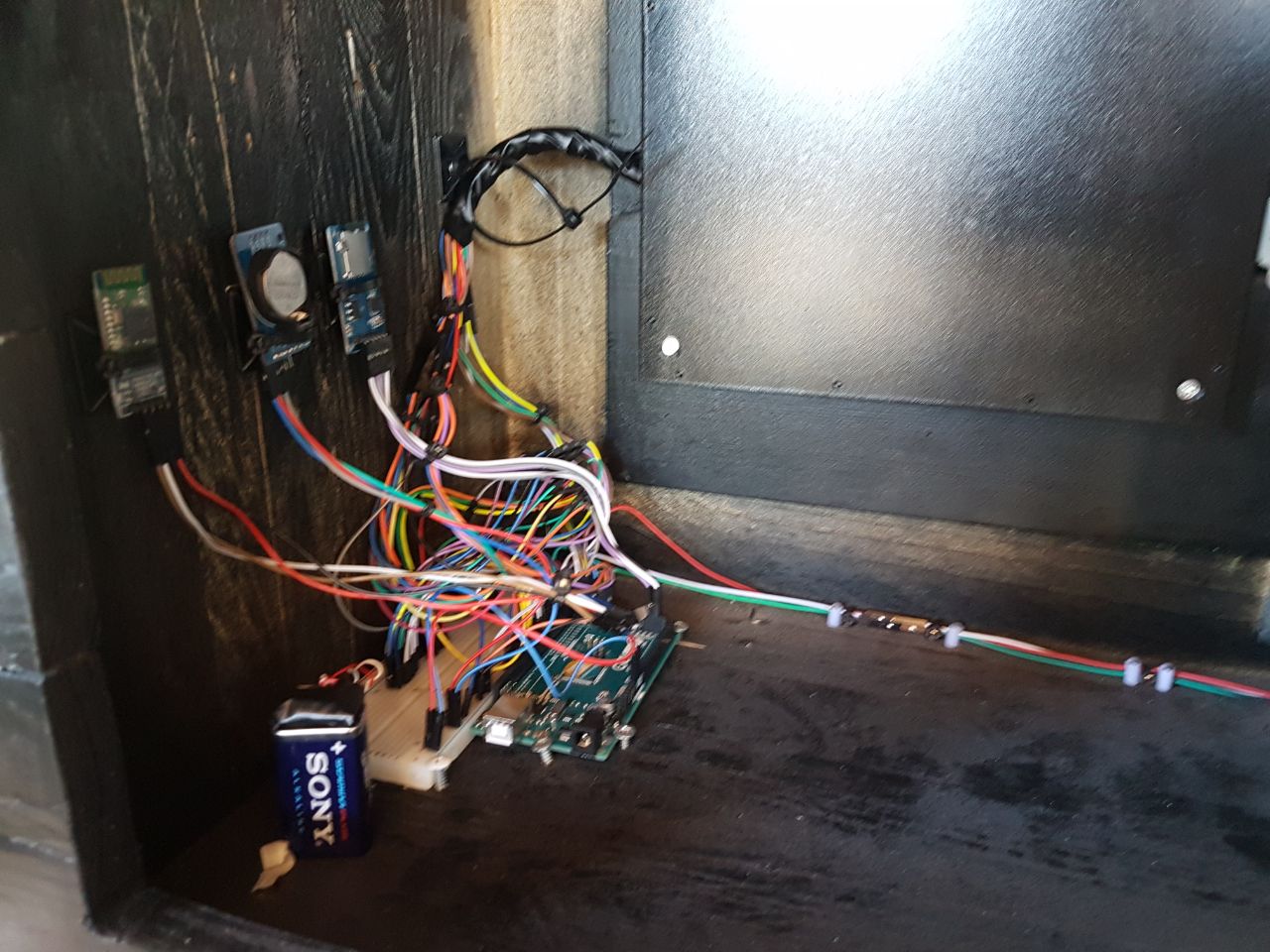
* Programmazione e test dei singoli componenti
* Assemblaggio e test complessivo

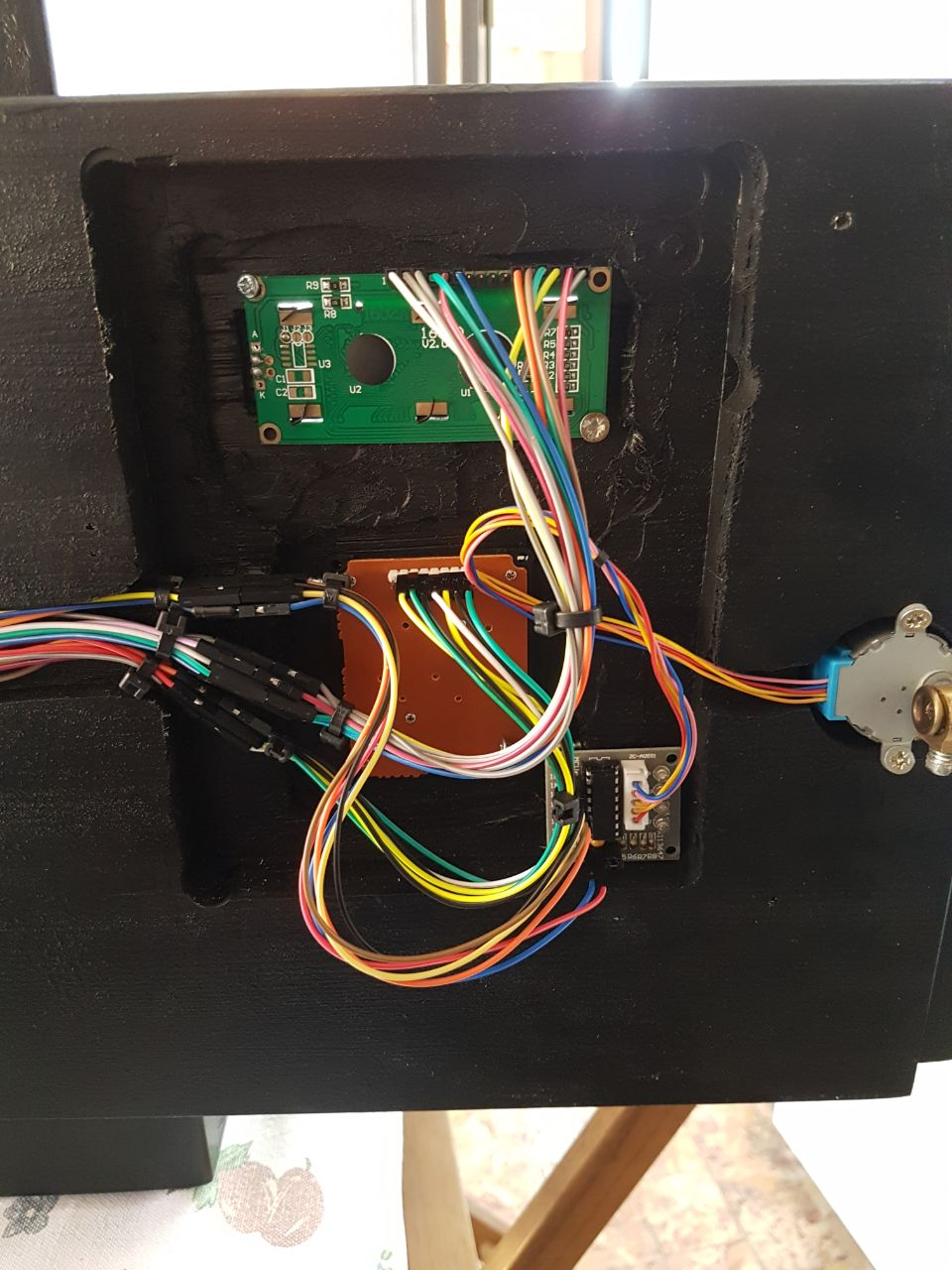
Per la parte fisica abbiamo subito riscontrato delle problematiche, siccome abbiamo dovuto attendere più di un mese dall’ acquisto dei componenti affinchè questi arrivassero. Una volta arrivata la componentistica, il team hardware si è diviso in due e ha cominciato a testare e a programmare uno ad uno ogni modulo.

Arrivati al test del sensore d’ impronte digitali abbiamo avuto altri problemi, dato che la libreria fornitaci dal costruttore non era in grado di rilevare il sensore. Per ovviare a questo inconveniente abbiamo dovuto ricercare una libreria che si adattasse al nostro sensore. Fortunatamente abbiamo trovato la libreria “AdaFruit-Fingerprint” che pur non essendo quella apposita per il modello del sensore, funzionava perfettamente.

Dopo aver testato e programmato ogni componente singolarmente il team preposto per lo sviluppo di Arduino ha montato su una breadboard tutti i componenti e ha cominciato lo sviluppo del “Menu”, ossia il programma che gestisce le funzionalità della cassaforte sincronizzando il lavoro di tutti i componenti.

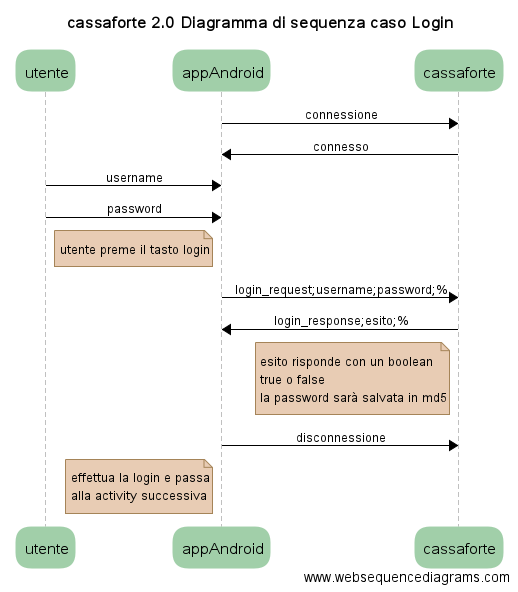
Durante questa fase ci si è accorti di un problema di alimentazione. Tutti i componenti infatti non riuscivano ad essere alimentati dall’ Arduino. Questo inconveniente è stato risolto facilmente inserendo una batteria e collegandola alla breadboard in modo da dare un supplemento di energia a tutti i componenti collegati.

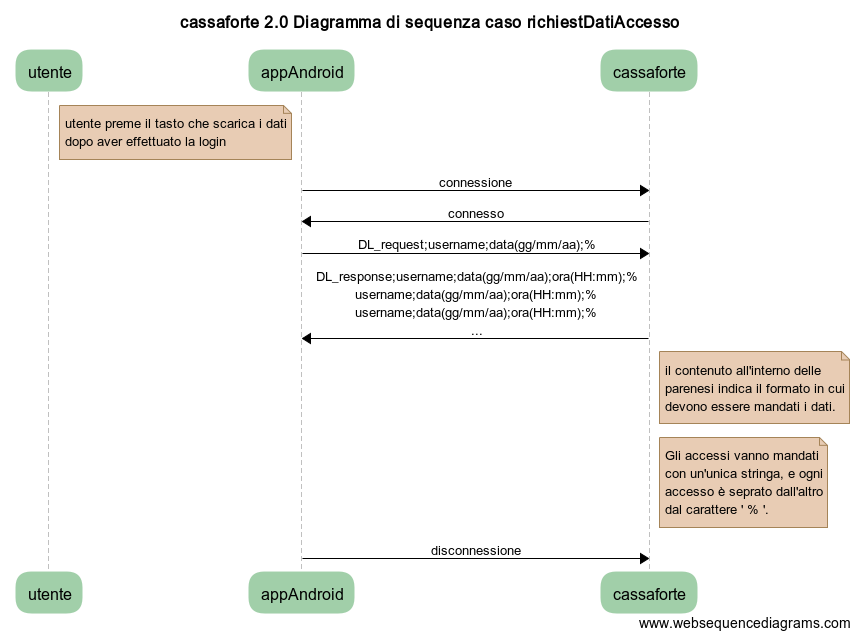
Successivamente ci si è occupati del montaggio di tutti i componenti all’ interno della cassaforte e il risultato finale è il seguente:

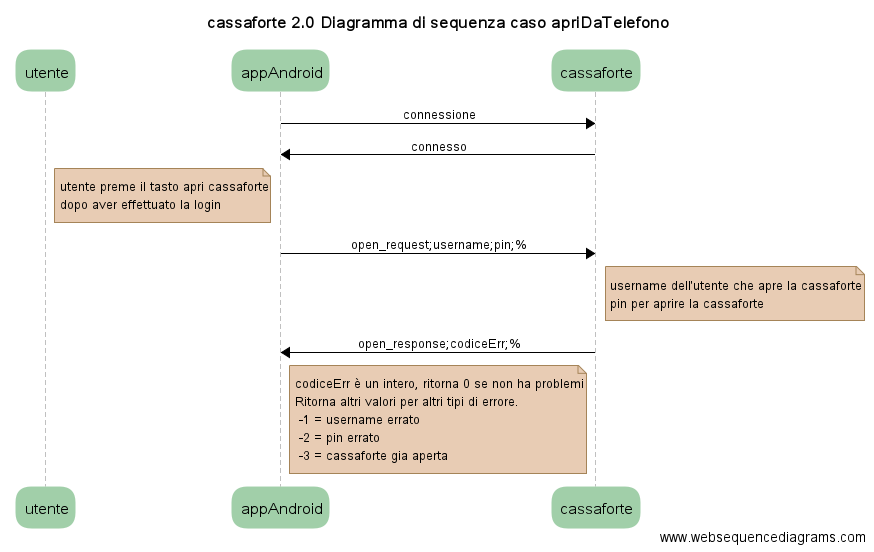


PROTOCOLLI DI COMUNICAZIONE

Per gestire in maniera ottimale la comunicazione sono stati creati dei protocolli di comunicazione grazie al sito “websequencediagrams.com”. I principali sono riportati di seguito:







# Beta test

Durante il beta test, il gruppo si è riunito e ha provato ogni funzionalità della Cassaforte 2.0, risolvendo eventuale errori nella comunicazione e aumentando le prestazioni in termini di velocità.

Principalmente sono stati corretti:

* Ritardi nella comunicazione
* Errori nella comunicazione
* Bug dell’ applicazione
* Errori nella registrazione degli utenti su Arduino
* Errori nell’ invio degli accessi in base alla data

POSSIBILI MIGLIORIE

Durante la fase del beta test, il team ha anche pensato a dei possibili miglioramenti che potrebbero rendere ancora più efficiente il progetto:

* Aggiungere una fotocamera per scattare delle foto a chi effettua gli accessi
* Usare Raspberry al posto di Arduino Mega per gestire al meglio le richieste dell’ utente e per potersi connettere alla rete e salvare gli accessi su un database, contattabile anche da applicazione tramite un web service.
* Utilizzare un modulo wifi al posto del modulo bluetooth per velocizzare la comunicazione e per renderla più stabile.

# Funzionamento

PRIMO AVVIO SENZA APPLICAZIONE

Se si vuole utilizzare la cassaforte senza applicazione al primo avvio bisognerà salvare l’ impronta digitale dell’ utente digitando il codice \*5 e premendo successivamente # per inviare. Verrà richiesto il pin della cassaforte (fornito dagli sviluppatori) e successivamente partirà la sequenza per salvare l’ impronta. Il sensore d’ impronte (in basso a sinistrà) comincerà a lampeggiare. Bisognerà appoggiare il dito e tenerlo fermo per 5 secondi. La luce smetterà di lampeggiare. Ripetere la sequenza fino a quando sullo schermo non comparirà la scritta “Impronta registrata”.

La cassaforte è pronta ora per l’ utilizzo.

Ogni qualvolta che si vorrà accedere alla cassaforte bisognerà digitare il pin e premere #. Inserire l’ impronta appena il sensore comincerà a lampeggiare e attendere l’ apertura della cassaforte. Una volta finito l’ utilizzo, si dovrà chiudere manualmente lo sportello e si dovrà premere # per avviare la chiusura della cassaforte.

PRIMO AVVIO CON APPLICAZIONE

Se si vuole utilizzare la cassaforte con l’ applicazione bisognerà, come prima cosa, creare un nuovo utente. Quindi una volta aperta l’ applicazione l’ utente dovrà andare nell’ apposita activity e inserire uno username(minimo 6 caratteri), una password (minimo 5 caratteri) e il pin della cassaforte. Inviando tutto alla cassaforte verrà richiesto di salvare l’ impronta dell’ utente. Il sensore comincerà a lampeggiare e la procedura da seguire è la medesima descritta sopra. Una volta che l’ utente ha l’ account registrato, può usufruire dei servizi dell’ applicazione effettuando la login. Ad esempio potrà aprire la cassaforte da telefono o potrà controllare gli accessi che sono stati effettuati in un determinato giorno.

# Ringraziamenti

Il progetto ci ha portato via molto tempo negli ultimi mesi e nonostante i nostri sforzi continui ci sono persone esterne al nostro gruppo che ci hanno sostenuto ed aiutato nella realizzazione di questo prototipo.  
In particolare vogliamo ringraziare Perego Attilio per il tempo dedicatoci per la realizzazione della cassaforte in legno e i prof per la disponibilità nel fornire consigli e aiuti quando necessario.

# Bibliografia e sitografia

[www.stackoverflow.com](http://www.stackoverflow.com)

[www.arduino.com](http://www.arduino.com)

[www.github.com](http://www.github.com)