Immagine che contiene testo, segnale, grafica vettoriale

Descrizione generata automaticamente

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Documentazione**  Unlocker IoT   |  |  | | --- | --- | | Riferimento |  | | Versione | 1.0 | | Data | 15/05/2022 | | Destinatario | Prof. P. Ritrovato  Prof.ssa L. Fotia | | Presentato da | Montefusco Alberto  Spina Gennaro  Oskar Szuba | |

Sommario

[Sommario 2](#_Toc530825397)

[Responsabilità 4](#_Toc530825396)

1. [Introduzione 5](#_Toc530825398)

1.1 Scopo del Sistema 5

1.2 Panoramica ............................................................................................................................................ 9

1. Sistema Proposto ............................................................................................................................................ 9

2.1 Requisiti funzionali ............................................................................................................................. 10

2.2 Requisiti non funzionali .................................................................................................................... 10

1. Strumenti Software e Hardware utilizzati................................................................................................. 20

3.1 Dispositivi Hardware ........................................................................................................................ 12

3.2 Micropython vs C .............................................................................................................................. 12

3.3 GUI ...................................................................................................................................................... 12

Responsabilità

|  |  |
| --- | --- |
| **Artefatto** | **Autori** |
|  |  |
|  |  |

1. Introduzione
   1. Scopo del Sistema

La realizzazione di Unlocker IoT ha come obiettivo quello di facilitare l’utente nell’autenticazione digitale ma anche di assicurare la sicurezza attraverso la sua funzione di password manager. Con questo sistema l’utente non dovrà fare altro che collegarlo ad un dispositivo (Computer, Smartphone, …) tramite Bluetooth e, una volta che si è autenticato tramite il fingerprint di Unlocker IoT, le credenziali scelte saranno inviate automaticamente nei campi che l’utente ha selezionato per la verifica. Quindi, tra gli obiettivi che il sistema propone di assicurare abbiamo: la sicurezza, l’efficienza, la portabilità e la versatilità.

* 1. Panoramica

Nei seguenti capitoli andremo ad analizzare i requisiti funzionali e non funzionali che il sistema deve necessariamente assicurare (Capitolo 2). Inoltre, verranno descritti gli strumenti Software utilizzati per la realizzazione dell’interfaccia grafica che permette il setup del sistema IoT da parte dell’utente, i dispositivi Hardware aggiuntivi integrati nella scheda ESP32 ed i linguaggi utilizzati per la programmazione IoT (Capitolo 3).

1. Sistema Proposto

Il sistema che proponiamo di realizzare è un dispositivo IoT che permette ad un utente di autenticarsi in un sito web oppure in un’applicazione per smartphone in pochi e semplici passi, inoltre, l’utente non dovrà ricordare tutte le password e gli username che possiede poiché Unlocker IoT funge anche da password manager: al suo interno saranno memorizzate un insieme di credenziali (cifrate) che l’utente potrà reperire tramite semplici steps.

Spieghiamo ora il funzionamento del sistema proposto. L’utente potrà accendere Unloacker premendo un piccolo bottone posto sul lato. All’accensione, l’utente visualizzerà su un piccolo schermo OLED una lista di nomi (i siti web o le applicazioni a cui è registrato con username e/o password). L’utente sceglierà un nome dalla lista e avrà accesso a quelle credenziali solo dopo essersi verificato tramite fingerprint posto sul sistema.

Una volta che la verifica ha avuto esito positivo, l’utente collegherà il sistema ad un dispositivo (Computer o Smartphone) tramite il Bluetooth: il collegamento avverrà solo quando l’utente avrà inserito un codice di sicurezza sul suo Computer o Smartphone; nei prossimi accoppiamenti il codice non sarà richiesto fino a quando l’utente non cancellerà l’autenticazione Bluetooth sul dispositivo accoppiato.

Successivamente, l’utente, dopo aver selezionato un campo per l’inserimento dell’username o della password, cliccherà sul primo bottone, posto di fianco lo schermo OLED e automaticamente (tramite una simulazione della tastiera mediante Bluetooth) il sistema invierà la stringa decifrata nel campo selezionato. Per l’inserimento della password la procedura è analoga ma l’invio avverrà solo quando l’utente pigerà il secondo bottone posto sotto al primo di fianco lo schermo OLED.

Dopo che le credenziali sono state inviate il sistema automaticamente mostrerà nuovamente la lista di nomi e, per accedere alle credenziali, l’utente dovrà nuovamente verificarsi tramite fingerprint.

Nel prossimo capitolo spiegheremo come avverrà il setup del sistema IoT tramite GUI realizzata in Java.

* 1. Requisiti funzionali

|  |  |  |
| --- | --- | --- |
| **Identificativo** | **Priorità** | **Descrizione** |
| **RF[1]** |  |  |
| **RF[2]** |  |  |
| **RF[3]** |  |  |
| **RF[4]** |  |  |
| **RF[5]** |  |  |
| **RF[6]** |  |  |
| **RF[7]** |  |  |
| **RF[8]** |  |  |
| **RF[9]** |  |  |
| **RF[10]** |  |  |
| **RF[11]** |  |  |
| **RF[12]** |  |  |
| **RF[13]** |  |  |

* 1. Requisiti non funzionali

|  |  |  |
| --- | --- | --- |
| **Identificativo** | **Priorità** | **Descrizione** |
| **RNF[1]** | 5 | Il sistema deve garantire la sicurezza delle credenziali memorizzate |
| **RNF[2]** | 5 | Il sistema deve collegarsi ad un dispositivo e inviare le credenziali entro pochi secondi |
| **RNF[3]** | 5 | Il sistema deve essere compatibile con tutti i dispositivi che supportino una connessione Bluetooth e una tastiera virtuale |

1. Strumenti Software e Hardware utilizzati
   1. Dispositivi hardware

Il sistema dovrà essere composto dai seguenti componenti hardware:

LISTA HARDWARE PROTOTIPO

* Scheda ESP32 per la gestione del sistema, dei sensori esterni e dell’interfacciamento con i dispositivi su cui sarà necessario inserire le credenziali di accesso (computer, smartphone ecc…);
* Display OLED da 0.96” che permetterà l’interfacciamento dell’utente al dispositivo;

[](https://www.amazon.it/display-KYYKA-caratteri-compatibile-Raspberry/dp/B09N77J4DK/ref=sr_1_4_sspa?crid=M6BDQQMJCE8A&keywords=arduino+oled+display&qid=1649606160&sprefix=arduino+oled+display+%2Caps%2C133&sr=8-4-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExRUJIRDkwUVdNRVNVJmVuY3J5cHRlZElkPUEwNzU1MTIwMzE0TDJYTkJSN0tLSCZlbmNyeXB0ZWRBZElkPUEwNTQzMDE3RFBWT1lDMzZYVVBUJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==)

* Capacitive fingerprint sensor, attraverso il quale sarà possibile autenticare l’utente. Sensore che garantirà la presenza dell’utente a cui apparterrà il dispositivo stesso;

[Immagine che contiene testo

Descrizione generata automaticamente](https://www.amazon.it/Capacitive-Fingerprint-Raspberry-360-degree-Recognition/dp/B07W84DCT7/ref=sr_1_1?crid=1ZKMZ8VO325YJ&keywords=capacitive+fingerprint+sensor&qid=1649605922&sprefix=capacitive+fingerprint+sensor%2Caps%2C114&sr=8-1)

* 2 x interruttori tattili momentanei, essi consentiranno all’utente di navigare attraverso i menù e le credenziali presenti sul dispositivo;

[Immagine che contiene elettronico

Descrizione generata automaticamente](https://www.amazon.it/HUAZIZ-Miniatura-Momentanea-Interruttore-miniaturizzato/dp/B08VDFW4VG/ref=sr_1_3_sspa?crid=1JQKY5XQRVH0Y&keywords=interruttori+tattili+momentanei&qid=1649605837&sprefix=interruttori+tattili+momentanei%2Caps%2C110&sr=8-3-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExTzRXTzhFS1BaTkhNJmVuY3J5cHRlZElkPUEwOTk2NDU4MkJaREVFODdLMkpYViZlbmNyeXB0ZWRBZElkPUEwMzE4MzQyNEg5UzZNMEUyTUlOJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==)

* Batteria LiPo 1 cella 500mah 3,7v permetterà di avere un dispositivo portatile e indipendente. [Immagine che contiene testo

  Descrizione generata automaticamente](https://www.amazon.it/gp/product/B082NZGW3V/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8&psc=1)
* LiPo BMS per la gestione della batteria, quindi della carica, scarica e della prevenzione della salute della batteria.

[Immagine che contiene testo, elettronico

Descrizione generata automaticamente](https://www.amazon.it/HAYATEC-TP4056-batteria-ricarica-confezione/dp/B08KWD643H/ref=sr_1_7?crid=36MIJYMT636G5&keywords=modulo+ricarica+lipo+usb+c&qid=1649605596&sprefix=modulo+ricaricalipo+usb+c%2Caps%2C121&sr=8-7)

LISTA HARDWARE DISPOSITIVO FINALE

* Scheda ESP32 TTGO, per la gestione del sistema, dei sensori esterni e dell’interfacciamento con i dispositivi su cui sarà necessario inserire le credenziali di accesso (computer, smartphone ecc…). La board grazie al display OLED, gli interruttori momentanei e il BMS integrato, permetterà di ridurre significativamente le dimensioni finali del prodotto;

[Immagine che contiene elettronico, circuito

Descrizione generata automaticamente](https://www.amazon.it/gp/product/B07VNG9D52/ref=ppx_yo_dt_b_asin_image_o03_s00?ie=UTF8&psc=1)

* Capacitive fingerprint sensor, attraverso il quale sarà possibile autenticare l’utente. Sensore che garantirà la presenza dell’utente a cui apparterrà il dispositivo stesso;

[Immagine che contiene testo

Descrizione generata automaticamente](https://www.amazon.it/Capacitive-Fingerprint-Raspberry-360-degree-Recognition/dp/B07W84DCT7/ref=sr_1_1?crid=1ZKMZ8VO325YJ&keywords=capacitive+fingerprint+sensor&qid=1649605922&sprefix=capacitive+fingerprint+sensor%2Caps%2C114&sr=8-1)

* Batteria LiPo 1 cella 500mah 3,7v permetterà di avere un dispositivo portatile e indipendente. [Immagine che contiene testo

  Descrizione generata automaticamente](https://www.amazon.it/gp/product/B082NZGW3V/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8&psc=1)
* Micropython VS C

Programmare la board per la realizzazione del progetto utilizzando il **micropython**, comporterà l’utilizzo delle seguenti librerie:

* [MicroPythonBLEHID](https://github.com/Heerkog/MicroPythonBLEHID), questa libreria permette la simulazione di una tastiera bluetooth su diversi dispositivi Windows e Android, ma non è compatibile con i sistemi MacOs, IOS, IpadOs. Il che comporta la rottura del 3° vincolo non funzionale stabilito a priori;
* [MicroPythonOledDisplay](https://github.com/micropython/micropython/blob/master/drivers/display/ssd1306.py), questa libreria permette di gestire la comparsa e la posizione di scritte su uno schermo OLED;
* [AdafruitFingerprint](https://docs.circuitpython.org/projects/fingerprint/en/latest/_modules/adafruit_fingerprint.html), questa libreria permette la gestione di un sensore per le impronte digitali;
* [ESP-Widget](https://github.com/joewez/ESP-Widget), questa libreria permette di mostrare immagini su uno schermo oled, anche se in modo molto limitante, di fatto le immagini per poter essere mostrate devono essere prima convertite in testo [LINK](http://www.eran.io/the-dot-factory-an-lcd-font-and-image-generator/), e poi le si possono utilizzare [LINK](https://forum.micropython.org/viewtopic.php?t=4901). Questo comporta un grado di difficoltà ulteriore per poter gestire testo e immagini, rendendo impossibile mostrare immagini con testo che vari dinamicamente. A meno che non si riscriva la libreria;