Sommario

[Obiettivo e nascita idea: 2](#_Toc140433493)

[Descrizione: 2](#_Toc140433494)

[Descrizione Funzioni: 2](#_Toc140433495)

[Ecosistema: 3](#_Toc140433496)

[Componenti: 3](#_Toc140433497)

[1. Db: 3](#_Toc140433498)

[1.1. Diagramma er: 3](#_Toc140433499)

[1.2. Tabelle db: 4](#_Toc140433500)

[1.3. Implementazione 4](#_Toc140433501)

[2. Domotica: 5](#_Toc140433502)

[2.1. componenti necessari: 5](#_Toc140433503)

[2.2. Descrizione: 5](#_Toc140433504)

[2.3. Dettagli: 5](#_Toc140433505)

[2.3.1. Web Service 5](#_Toc140433506)

[2.3.2. Schema funzionamento microcontrollori: 5](#_Toc140433507)

[3. WebService 6](#_Toc140433508)

[3.1. Descrizione 6](#_Toc140433509)

[3.2. url 6](#_Toc140433510)

[4. App Mobile 9](#_Toc140433511)

[4.1. Descrizione 9](#_Toc140433512)

[4.2. Pagine 9](#_Toc140433513)

[4.3. Grafica 9](#_Toc140433514)

[4.4. Diagrammi 9](#_Toc140433515)

[5. Sito: 10](#_Toc140433516)

[5.1. Descrizione 10](#_Toc140433517)

[5.2. Pagine 10](#_Toc140433518)

[Stesse della app 10](#_Toc140433519)

[5.3. Grafica 10](#_Toc140433520)

[5.4. Diagrammi 10](#_Toc140433521)

[Idee 10](#_Toc140433522)

# Obiettivo e nascita idea:

Con l’avvento della domotica molte case hanno installato sempre più dispositivi domotici, ma tutti questi dispositivi sono gestiti da app poco user friendly oppure le app disponibili non coprono tutti i dispositivi allo stesso momento, e ci si ritrova con diverse app per gestire i propri dispositivi.

Da questo problema abbiamo avuto una idea di uniformare i dispositivi, e creare un ecosistema sicuro e user friendly che permetta agli utenti di poter gestire tutti i loro dispositivi tramite app o sito.

# Descrizione:

Il nostro progetto permetterà agli utenti di poter gestire tutti i loro pulsanti presenti nella propria casa tramite un agile ecosistema datacentrico che espone all’utente app sia per pc che per mobile oppure tramite un comodo sito.

# Descrizione Funzioni:

Nel progetto ogni persona potrà:

* Creare un account
* Effettuare il login
* Poter resettare la password
* Poter vedere le persone associare alla casa
* Visualizzare il suo profilo
* Spegnere o accendere un pulsante
* Visualizzare info casa
* Visualizzare info pulsanti relativi alla propria casa
* Visualizzare info programmi relativi alla propria casa
* Creare una casa

Una volta creata una casa si diventa admin della casa e allora si può fare anche:

* Modificare Casa
* Associare un account alla propria casa
* Poter ottenere info degli altri utenti
* Dissociare un account dalla propria casa
* Creare un pulsante
* Eliminare un pulsante
* Modificare pulsante
* Creare un programma
* Eliminare un programma
* Modificare un programma

# Ecosistema:

Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente

Il progetto sarà composto da diverse parti che interagiranno tra di loro:

1. Server:
   1. db che manterrà i dati relativi alle case domotiche.
   2. webService che permetterà di accedere a quei dati e poterli gestire.
   3. Un sito web che permette alle persone di interfacciarsi al web service tramite una grafica
2. App:
   1. Una app per mobile che permetta come il sito di poter gestire la propria casa domotica
   2. Una possibile app per computer
3. Domotica:
   1. Un web service per il settaggio
   2. Gestione hardware

# Componenti:

1. Db:
   1. Diagramma er:

Immagine che contiene diagramma, schizzo, Disegno tecnico, Piano

Descrizione generata automaticamente

* 1. Tabelle db:

Utente(Id, Username, Mail, Immagine, Password, Nome, Api\_Key, admin, \**IdCasa*)

Casa(Id, Nome, )

Pulsante(Id, Nome, Stato, *\*IdCasa*)

Programmi(Id, OrarioInizio, OrarioFine, *\*idPulsante*)

Log(Id, Orario, Descrizione, *\*IdUtente*)

Reset(Id, Codice, Orario, Usato, *\*idUtente*)

* 1. Implementazione
     1. Utente

|  |  |  |  |
| --- | --- | --- | --- |
| Attributo | Tipo | Lunghezza | Opzioni extra |
| Id | Int | 11 | Primary\_key |
| Username | Varchar | 32 | Unique |
| Mail | Varchar | 64 | Unique |
| Immagine | Varchar | 64 | Default “user.png” |
| Password | Char | 256 |  |
| Nome | VarChar | 32 |  |
| Api\_Key | Char | 64 | Unique |
| Admin | Int | 1 |  |
| IdCasa | Int | 11 | Foreign\_key |

* + 1. Casa

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Int | 11 | Primary\_Key |
| Nome | Varchar | 32 |  |

* + 1. Pulsante

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Int | 11 | Primary\_key |
| Nome | Varchar | 32 |  |
| Stato | Int | 1 | Default 0 |
| IdCasa | Int | 11 | Foreign\_key |

* + 1. Programmi

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Int | 11 | Primary\_key |
| OrarioInizio | Time |  |  |
| OrarioFine | Time |  |  |
| IdPulsante | Int | 11 | Foreign\_key |

* + 1. Log

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Int | 11 | Primary\_key |
| Orario | Datetime |  | Default now() |
| Descrizione | Varchar | 256 |  |
| IdUtente | Int | 11 | Foreign\_key |

* + 1. Reset

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Int | 11 | Primary\_key |
| Codice | Char | 64 | Unique |
| Orario | DataTime |  | Default now() |
| Usato | Int | 1 | Dafault 0 |

1. Domotica:
   1. componenti necessari:
      1. Arduino uno
      2. Microcontrollore con wi-fi
      3. Sensori
   2. Descrizione:

La domotica composta da un arduino nano 33 iot per gestire le connessioni tramite wifi, e un arduino uno che interrogherà e aggiornerà i pulsanti.

I due microcontrollori comunicheranno tramite seriale.

Arduino nano 33 iot avrà 2 compiti:

* + - * 1. continuare a interrogare il webService per trovare eventuali cambiamenti da comunicare ad arduino uno.
        2. Esporre un url che permetterà di passare i dati dell’utente associato alla casa.

Arduino uno ha 2 compiti:

* + - * 1. Ascoltare la seriale e in caso arrivano cambiamenti da fare li deve applicare.
        2. Deve controllare il circuito e in caso di cambiamenti li comunica ad arduino nano 33 iot.
  1. Dettagli:
     1. Web Service

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | url | Descrizione | Output | parametri |
| Post | \setUser | Permette di associare la domotica a un utente(**solo admin della casa**) | Bool esito | Api\_key |

* + 1. Schema funzionamento microcontrollori:

Immagine che contiene testo, diagramma, Piano, schermata

Descrizione generata automaticamente

Immagine che contiene testo, diagramma, schermata, Piano

Descrizione generata automaticamente

1. WebService
   1. Descrizione

Questo webservice servirà come layer di sicurezza, in quanto maschererà il db al suo interno, e tutte le app dovranno passare da lui per poter accedere ai dati.

Esporrà all’esterno diversi url, ognuno in grado di portare modifiche o restituire tati specifici.

Per utilizzare il web service ci sarà bisogno

* 1. url

restituiranno tutti json seguendo questo pattern:

{

“stato”: seguito dal codice di ritorno seguendo i response status,

in caso sia andato a buon fine

“risposta”: con i dati contenuti nella colonna output

In caso non va bene

“messaggio”: con un messaggio di errore che dica cosa non è andato bene

}

**Tra le [] vuol dire opzionale**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | url | Descrizione | output | parametri |
| Post | \login | Permette di effettuare il login e in caso abbia successo restituisce l’Api\_Key | Api\_key | Username, Password |
| Post | \register | Permette di registrare un account | Un bool per dire l’esito | Username, Mail, Immagine, Nome, Password |
| Get | \utenti | Permette di ottenere tutti gli utenti | Array di utenti | [username anche parziale]  [limiteQuantita] |
| Post | \resetPassword | Permette di richiedere l’invio del reset password(**un reset ogni 1 ora**) | Bool esito | Username |
| Put | \resetPassword | Permette di cambiare la password | Bool esito | Codice, NuovaPassword |
| Get | \Api\_ Key\profilo | Restituisce le informazioni relative all’utente a cui appartiene la Api\_key | Un oggetto con dentro tutte le info dell’utente |  |
| put | \Api\_Key\profilo | Permette di modificare le informazioni relative all’account | Un bool per dire l’esito | [Username,] [Mail,] [Immagine,] [Nome,] [Password] |
| Delete | \Api\_Key\profilo | Elimina l’account | Un bool per dire l’esito |  |
| Get | \Api\_ Key\casa | Restituisce le informazioni della casa | Un oggetto con tutte le info delle case |  |
| Post | \Api\_Key\casa | Permette di inserire una nuova casa  E la associa all’utente che **diventa admin** | Restituisce un bool per dire l’esito | Nome |
| Put | \Api\_Key\casa | Permette di modificare una casa  (**solo admin della casa**) | Restituisce un bool per dire l’esito | Nome |
| Delete | \Api\_Key\casa | Permette di eliminare una casa (**solo admin può eliminarla**, in caso contrario viene dissociato | Bool per dire l’esito |  |
| Put | \Api\_key\associaCasa | Permette di associare alla casa un’altra persona (**solo admin della casa può farlo**) | Bool per dire esito | IdUtente o Username |
| Get | \Api\_key\usersCasa | Permette di ottenere gli utenti associati a quella casa | Array di Utenti | [limit,] [Username] |
| Delete | \Api\_Key\dissociaCasa | Permette di dissociare un utente dalla casa(**solo admin può farlo**) | Bool esito | IdUtente |
| Get | \Api\_Key\pulsanti | Permette di avere tutti i pulsanti relativi alla casa | Array di Pulsanti | [limit] |
| Get | \Api\_Key\pulsante | Permette di ricevere le info di un pulsante | Oggetto con le info di un pulsante | IdPulsante |
| Post | \Api\_Key\pulsante | Permette di aggiungere un pulsante (**solo admin della casa**) | Bool per esito | Nome |
| Put | \Api\_Key\pulsante | Permette di modificare un pulsante(**solo admin della casa**) | Bool per esito | Nome |
| Delete | \Api\_Key\pulsante | Permette di eliminare un pulsante(**solo admin della casa**) | Bool per esito | idPulsante |
| Put | \Api\_Key\premiPulsante | Permette di accendere/spegnere il pulsante | Bool esito | idPulsante, Stato |
| Get | \Api\_Key\IdPulsante\programmi | Restituisce tutti i programmi legati al pulsante | Array di programmi | [limit] |
| Get | \Api\_Key\IdPulsante\programma | Restituisce le informazioni legate a un programma | Oggetto con info programma | IdProgramma |
| Post | \Api\_Key\IdPulsante\programma | Permette di inserire un nuovo programma(**solo admin della casa**) | Bool per esito | OrarioInizio, OrarioFine |
| Put | \Api\_Key\IdPulsante\programma | Permette di modificare un programma(**solo admin della casa**) | Bool esito | idProgramma, [OrarioInizio,] [OrarioFine] |
| Delete | \Api\_Key\IdPulsante\programma | Permette di eliminare un programma (**solo admin della casa**) | Bool esito | idProgramma |

1. App Mobile
   1. Descrizione

Una app che prenderà di manipolare i dati relativi al proprio account tramite pagine user friendly

* 1. Pagine
* Login
* Registrazione
* Cambio password
* Pulsanti
* Casa
* Utente
* Add pulsante
* Edit casa
* Edit utente
  1. Grafica

Immagine che contiene schermata, testo, Software multimediale, software

Descrizione generata automaticamenteImmagine che contiene schermata, elettronica, Software multimediale, software

Descrizione generata automaticamente

Immagine che contiene schermata, Software multimediale, software, testo

Descrizione generata automaticamente

* 1. Diagrammi

Immagine che contiene testo, diagramma, Piano, schematico

Descrizione generata automaticamente

1. Sito:
   1. Descrizione

Questo sito permetterà tramite pagine html di gestire la propria casa domotica

* 1. Pagine

Stesse della app

* 1. Grafica
  2. Diagrammi

# Idee

Casa che gestisce una serie di pulsanti on/off

Utenti, Case, Pulsanti, log, Programmi

Utenti:  
 Password, immagine, username, Api\_Key, nome

Casa:

identificativo, nome, pulsanti

Pulsanti:

identificativo, stato, nome

log:

utente, orario, descrizione

Programma:

pulsante, orarioAvvio, OraTermine

sviluppi futuri:

notifiche

Utente(Username, nome, password, Api\_key, immagine)

App mobile/desktop java/android studio

webService e sito node js

***DB***

Idea iniziale:

Casa che gestisce una serie di pulsanti on/off

Utenti, Case, Pulsanti, log, Programmi

Utenti:  
 Password, immagine, username, Api\_Key, nome

Casa:

identificativo, nome, pulsanti

Pulsanti:

identificativo, stato, nome

log:

utente, orario, descrizione

Programma:

pulsante, orarioAvvio, OraTermine

sviluppi futuri:

notifiche

Diagramma:

Diagramma er:

Immagine che contiene diagramma, schizzo, disegno, Disegno tecnico

Descrizione generata automaticamente

Tabelle db:

Utente(Username,Immagine,Password,Nome,Api\_Key,\**IdCasa*)

Casa(Id, Nome)

Pulsante(Id, Nome, Stato, *\*IdCasa*)

Programmi(Id, OrarioInizio, OrarioFine, *\*idPulsante*)

Log(Id, Orario, Descrizione, *\*UsernameUtente*)