

SmartLock

Pany Intel·ligent

The logo of the Universitat Oberta de Catalunya (UOC), consisting of the letters 'UOC' in a bold, blue, sans-serif font.

David Bassa Romera

Grau Multimèdia
Arduino

Tutor/a de TF

Jose Lopez Vicario

**Professor/a responsable de
l'assignatura**

Pere Tuset

Data Lliurament

06/2023

Universitat Oberta
de Catalunya



Aquesta obra està subjecta a una llicència de [Reconeixement-
NoComercial-Compartir Igual 3.0 Espanya de Creative
Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

FITXA DEL TREBALL FINAL

| | |
|--|----------------------------|
| Títol del treball: | <i>Smartlock</i> |
| Nom de l'autor: | <i>David Bassa Romera</i> |
| Nom del consultor/a: | <i>Jose Lopez Vicario</i> |
| Nom del PRA: | <i>Pere Tuset</i> |
| Data de lliurament (mm/aaaa): | <i>06/2023</i> |
| Titulació o programa: | <i>Grau multimèdia</i> |
| Àrea del Treball Final: | <i>Area Arduino</i> |
| Idioma del treball: | <i>Català</i> |
| Paraules clau | <i>Arduino, RFID, Pany</i> |
| Resum del Treball | |
| <p>Hi ha diferents estudis que demostren que una persona pot arribar a perdre fins a 5.000h al llarg de la seva vida buscant objectes per dintre de casa, i entre les que ocupen el TOP 3 del Ranking es troba les claus de casa.</p> <p>I es justament en mig d'aquest problema social on SmartLock es presenta com una solució, on amb l'ajuda de l'ecosistema Arduino (programari i components) i desenvoluparà un pany de porta intel·ligent.</p> <p>Aquest pany permetrà controlar l'apertura de la porta d'entrada de casa amb l'ajuda d'un identificador NFC (targeta, clauer, etc), amb el mòbil o per contrasenya, i on problemes com ara la pèrdua de les claus que poden vulnerar la seguretat de casa formarien part del passat.</p> | |
| Abstract | |
| <p>There are various studies that show that a person can spend up to 5,000 hours during a life time looking for some objects inside the house. Among those objects, one of the TOP 3 in the ranking is the house key.</p> <p>It is precisely a problem where SmartLock can offer a solution as it has developed a smart door lock technology in the Arduino ecosystem (software and components).</p> <p>This lock will make possible to control the opening of the front door of the house with an NFC identifier (card, key fob, etc.). This technology will enable that door can be opened either by mobile phone or by password. By implementing of this system, the problems that existed in the past and could breach the security like the loss of keys, will disappear.</p> | |

Índex

| | |
|--|----|
| 1. Introducció | 1 |
| 1.1 Context i justificació del Treball | 1 |
| 1.2 Objectius del Treball | 2 |
| 1.3 Impacte en sostenibilitat, ètic-social i de diversitat | 3 |
| 1.4 Enfocament i mètode seguit | 4 |
| 1.5 Planificació del Treball | 4 |
| 1.6 Breu sumari de productes obtinguts | 10 |
| 1.7 Estudi de viabilitat | 11 |
| 1.8 Breu descripció dels altres capítols de la memòria | 15 |
| 2. Estat del Art | 16 |
| 2.1 Benchmarking | 16 |
| 2.2 Tecnologies utilitzades: avantatges i inconvenients | 18 |
| 3. Desenvolupament Smartlock | 20 |
| 3.1. Sistema Smartlock | 20 |
| 3.2. Smartlock - Components | 21 |
| 3.3. Llibreries | 31 |
| 3.4. Desenvolupament codi Arduino.Smartlock | 31 |
| 3.5. Desenvolupament codi Arduino xip ESP8266 | 37 |
| 3.6. Desenvolupament Servidor | 40 |
| 3.7. Diagrames funcionals | 46 |
| 4. Validació i campanya de proves | 49 |
| 4.1. Dispositiu Smartlock | 49 |
| 4.2. Servidor | 50 |
| 4.3. Acabat final -Smartlock | 51 |
| 4.4. Acabat final -Servidor | 52 |
| 5. Conclusions i treballs futurs | 53 |
| 5.1. Conclusions | 53 |
| 5.2. Treballs a futur | 53 |
| 6. Glossari | 54 |
| 7. Bibliografia | 55 |
| 8. Referències | 57 |
| 9. Annexos | 58 |
| 9.1 Codi Arduino Smartlock | 58 |
| 9.2 Codi xip ESP8266 | 64 |

Llista de figures

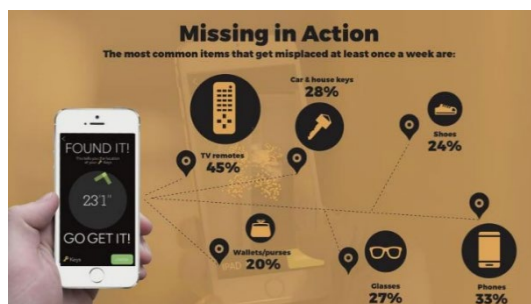
| | |
|--|----|
| Il·lustració 1 Objectes més perduts | 1 |
| Il·lustració 2 Diagrama de Gantt inicial resumit | 8 |
| Il·lustració 3 Diagrama de Gantt detallat | 10 |
| Il·lustració 4 Nuki Smart Lock 3.0 Pro | 16 |
| Il·lustració 5 Tesa Assa Abloy | 16 |
| Il·lustració 6 ZKTeco ZK-AL20DB | 17 |
| Il·lustració 7 Vima Smart Lock | 17 |
| Il·lustració 8 Esquema circuit Smartlock..... | 20 |
| Il·lustració 9 Esquema servidor | 21 |
| Il·lustració 10 Esquema funcionament Smarlock..... | 21 |
| Il·lustració 11 Microcontroladors Arduino | 21 |
| Il·lustració 12 Esquema connexions Arduino ATmega2560 + ESP8266 | 22 |
| Il·lustració 13 Interruptors control de la placa..... | 23 |
| Il·lustració 14 Selector canal de comunicació | 23 |
| Il·lustració 15 Esquema de colors d'una resistència..... | 24 |
| Il·lustració 16 Esquema connexions LED RGB | 24 |
| Il·lustració 17 Esquema de connexions piezoelèctric..... | 25 |
| Il·lustració 18 Esquema de connexions lector RFID RC522 | 26 |
| Il·lustració 19 Esquema RFID Tag | 26 |
| Il·lustració 20 Connexions polsadors d'un teclat matricial | 27 |
| Il·lustració 21 Esquema de connexions d'un teclat matricial 4x4 | 27 |
| Il·lustració 22 Esquema de connexions d'un potenciòmetre..... | 28 |
| Il·lustració 23 Esquema de connexions d'un panell LCD..... | 28 |
| Il·lustració 24 Esquema de mides d'un Servomotor SG90 | 29 |
| Il·lustració 25 Esquema de connexions servomotor SG90 | 29 |
| Il·lustració 26 Passos a seguir carregar llibreria..... | 31 |
| Il·lustració 27 Passos a seguir carregar llibreria ZIP | 31 |
| Il·lustració 28 Tarifes disponibles de 000webhost..... | 40 |
| Il·lustració 29 Pantalla creació projecte..... | 40 |
| Il·lustració 30 Pantalla selecció tipus de hosting | 40 |
| Il·lustració 31 Pantalla explorador d'arxius de 000webhost | 41 |
| Il·lustració 32 Esquema Entitat-Relació de la base de dades..... | 41 |
| Il·lustració 33 Pantalla administració de base de dades de 000webhost..... | 41 |
| Il·lustració 34 Pantalla creació base de dades desde 000webhost..... | 41 |
| Il·lustració 38 Diagrama de connexions | 46 |
| Il·lustració 35 Diagrama de flux Smartlock | 46 |
| Il·lustració 36 Diagrama de flux del servidor | 47 |
| Il·lustració 37 Diagrama d'estats dels sensors i components | 48 |
| Il·lustració 38 Diagrama de seqüència | 48 |
| Il·lustració 39 Esquema d'un voltímetre | 49 |
| Il·lustració 40 Prova de so | 50 |
| Il·lustració 41 Smartlock final frontal | 51 |
| Il·lustració 42 Smartlock final interior | 51 |
| Il·lustració 43 Smartlock final lateral..... | 51 |
| Il·lustració 44 Captures de pantalla panell administrador | 52 |

1. Introducció

1.1 Context i justificació del Treball

Segons diferents estudis [1], un dels objectes que més perd la gent són les claus (de casa o automòbil), ja sigui de manera "definitiva" les perden pel carrer, les roben, etc. O bé no saben on les han deixat. Aquest estudis [1] revelen que al llarg de tota la vida d'una persona es poden perdre fins a 5.000h buscant objectes per dintre de casa (les claus és un dels objectes que encapçalen el rànquing).

A més, dintre dels objectes que més es perden i que més ansietat generen és troben les claus i els telèfons mòbils. El fet de perdre les claus no només genera un problema a l'hora d'accedir al domicili, sinó també posa en perill el benestar i seguretat d'aquesta, ja que s'augmenten les probabilitats que s'efectuïn robatoris. Aquesta inseguretat i ansietat que es genera, fa que finalment l'usuari hagi de canviar el pany de la porta, amb les despeses que això comporta, o fins i tot, hagi de trucar a un manyà perquè obri la porta.



Il·lustració 1 Objectes més perduts

Gràcies a aquests estudis, es pot observar que la majoria d'usuaris de totes les edats han patit aquest problema en algun moment de la seva vida (encara que finalment s'hagin trobat). Cal destacar que aquest problema no es dona en una zona geogràfica concreta, sinó que es troba en l'àmbit global (països desenvolupats), de fet, hi ha estudis realitzats en diferents punts del món; en Espanya, Regne Unit, Austràlia, EUA, etc.

Una vegada detectat aquest problema i analitzat els diferents estudis realitzats, el projecte SmartLock apareix per donar una solució tecnològica. Actualment, la tendència del mercat és estar tot connectat entre sí i a l'lot (internet de les coses), i aquesta és l'orientació que té SmartLock, connectar el pany de la porta dels usuaris a internet o xarxa local amb la finalitat de tenir un major control sobre l'accés a habitatge, així com facilitar i donar solució a les limitacions del sistema tradicional.

La solució que dona SmartLock consisteix en un sistema que permet controlar l'accés a l'habitatge de diferents maneres:

- Per entrar a casa s'utilitzarà una identificació per targeta/clauer/etc. RFID. La idea d'utilitzar aquest sistema envers a d'altres com podrien ser; lector d'empremta dactilar, clau tradicional amb xip (similar als cotxes), accés per codi, etc. Es pels següents motius:
 - Tant el lector RFID com els xips RFID són molt econòmics, per tant si es perd algun xip (targeta, clauer) l'usuari pot adquirir un altre de nou per un baix cost.
 - Es poden crear nous usuaris sense la necessitat d'estar presents (com en el cas de l'empremta digital).
 - L'accés per codi es pot veure vulnerat si algú visualitza la seqüència de codi, així com esborrar-se els dígits del teclat durant el seu ús.
 - L'accés a través de telèfon mòbil, està limitat al fet que el dispositiu no es quedi sense bateria, per poder entrar a l'habitatge.
- L'usuari administrador tindrà l'opció d'habilitar una doble autenticació per determinats usuaris, on a part de disposar d'una targeta RFID amb accés també haurà d'introduir una clau d'accés.
- L'administrador podrà crear, modificar i eliminar l'accés dels diferents usuaris, així com tenir un registre d'accessos.

Després de tot el mencionat anteriorment, aquest projecte SmartLock és rellevant dintre del Grau d'Enginyeria d'informàtica perquè amb l'ajuda de components bàsics d'aprenentatge com són tot l'ecosistema d'Arduino es pot donar solució a un problema social existent. És cert que actualment existeixen diferents sistemes que intenten donar solució a aquest problema, però amb un cost elevat. A més, el seu funcionament és també molt similar al del control de fitxatge dels treballadors d'una empresa, amb el que es podria estendre les seves funcionalitats més enllà de l'ús domèstic, i tot això només amb Arduino, un ecosistema desenvolupat inicialment perquè els alumnes poguessin crear dispositius i que aquest poguessin interactuar amb el seu entorn amb l'ajuda de sensors (Wiring 2003).

1.2 Objectius del Treball

Els objectius d'aquest projecte a part de classificar-los entre principals i secundaris també es categoritzaran en 3 tipus;

- **Imprescindible:** El sistema no pot funcionar sense complir tots els objectius imprescindibles
- **Importants:** Defineixen una part important del sistema, i sense aquest objectiu el sistema perd consistència, i s'hauria de buscar una alternativa.
- **Adicional:** Són objectius que afegeixen funcionalitats addicionals al sistema però que no són necessaris per garantir el seu funcionament.

Principals

| | | |
|------------|---|----------------|
| OBJ01 ✓ | Control total sobre el pany de l'habitatge; El sistema ha de poder controlar totalment el pany de la porta per poder obrir i tancar el pany, sigui elèctric o electromagnètic amb l'ajuda d'un microcontrolador Arduino. | Imprescindible |
| OBJ02 ✓ | Augmentar la seguretat de l'habitatge; El sistema Arduino ha de permetre la lectura de dispositius NFC (targetes, clauers, etc), amb els quals es connectarà a la BBDD i farà una consulta per verificar si aquest identificador té accés a l'habitatge, si és correcte, obrirà el pany, si no, denegarà l'accés. | Imprescindible |
| OBJ03 ✓ | Connexió amb el servidor: El sistema Arduino es connectarà per wifi a internet (amb un xip ESP8266), i aquest a un servidor MySQL on es realitzaran les consultes necessàries per la verificació de l'autenticació del NFC. | Important |
| OBJ04 ✓ | Creació BBDD i instal·lació del servidor: Es dissenyarà i crearà la BBDD amb la finalitat d'emmagatzemar tota la informació necessària pel correcte funcionament d'Smartlock (targetes autoritzades i contrasenyes). | Important |

Secundàries

| | | |
|------------|---|-----------|
| OBJ05 ✓ | Historial; SmartLock disposarà d'un historial d'accés, on es guardarà l'entrada i l'usuari que ha obert la porta. | Adicional |
| OBJ06 ✓ | Funcioni en cas de caiguda de llum: el sistema disposarà d'una bateria que permetrà el seu funcionament durant almenys 2 hores (es podrien emmagatzemar unes targetes mestre que estiguin guardades al codi del sistema o targeta SD i permetin obrir la porta encara que no hi hagi accés al servidor a causa del tall de llum). | Adicional |
| OBJ07 ✓ | Panell administrador; L'administrador podrà tenir al seu abast un panell d'administrador, on podrà donar d'alta i baixa els usuaris, modificar claus d'accés, activar o desactivar la doble autenticació, controlar l'obertura de la porta a distància. | Adicional |
| OBJ08 ✓ | Facilitar la vida de persones amb mobilitat reduïda, discapacitat visual o auditiva: Smartlock permetrà l'obertura amb targeta NFC o targeta NFC+Contrasenya, d'aquesta manera els usuaris que tinguin dificultats per introduir la contrasenya pel teclat podran accedir igualment a l'habitatge. | Adicional |

Observacions: *s'han considerat dos objectius com a importants, que són la part de la verificació de l'accés via servidor amb BBDD, aquest seria el funcionament ideal del sistema. Si no s'aconsegueix crear aquest accés al servidor, s'haurà de buscar una alternativa, com per exemple, emmagatzemar les dades en una targeta SD perquè pugui funcionar sense servidor, connectar-se al servidor amb una targeta Ethernet en comptes de Wi-Fi, etc.*

1.3 Impacte en sostenibilitat, ètic-social i de diversitat

Una vegada identificat el problema social mencionat anteriorment, s'han plantejat diferents solucions per buscar una solució a aquest problema, i durant aquesta cerca de possibles solucions s'ha intentat tenir en compte i reflexionar sobre les tres dimensions que planteja CCEG.

En la dimensió de sostenibilitat, és cert que és un sistema que no és autosuficient, donat que necessita energia externa, ni promou la sostenibilitat, consum responsable, etc. Però, sí, que s'ha tingut en compte alguns aspectes, com ara, d'una banda, que tots els components electrònics utilitzats en el projecte, cap és fet a mida. És a dir, components genèrics que una vegada es retiri el producte, com que no són fets a mida, es poden reciclar/reaprofitar cap a altres productes existents en el mercat, a més de no necessitar una producció de components electrònics específics per aquest projecte. D'altra banda, tampoc fa ús de piles (com en el cas d'alguns dels productes existents del mercat) que donen solució al mateix producte i que afavoreixen que anualment es generin tones de piles [2]. Aquest dos punts, intenten ajudar a complir l'objectiu ODS 12.5 que consisteix a reduir la generació de residus a través de la prevenció i reutilització. A més, s'ha buscat que el seu funcionament sigui de baix consum, encara que originalment no és autosuficient perquè no incorpora cap sistema d'ús d'energia renovable, sí que pot utilitzar-se amb sistemes de plaques solars o similars, a causa del seu baix consum que tenen els seus components, per intentar ajudar a la millora d'eficiència energètica ODS 7.3.

En la dimensió de comportament ètic i de responsabilitat social, aquest projecte té una finalitat tècnica que pretén donar una solució a un problema, el qual no afectaria a nivell ètic ni de responsabilitat social, donat que no promou ni afecta en l'àmbit de l'aigua, l'alimentació mundial, drets laborals, justícia social, etc. Si s'analitza a escala laboral, aquest projecte no busca ni afecta una millora en l'àmbit de drets laborals ni de pau, ja que és un producte per ús domèstic i empresarial de seguretat. No obstant és cert que busca un increment tecnològic a nivell social i, per tant, econòmic, però aquest increment no afectant en l'àmbit de societat o país sinó a nivell de l'empresa creadora de Smartlock, i amb una producció nacional, no es contempla en cap moment la producció en l'estranger.

En la dimensió de diversitat, gènere i drets humans, s'ha tingut en compte diferents factors:

- Els missatges de la pantalla LCD que mostrarà Smartlock, en la qual dóna la benvinguda als usuaris, missatges d'error o informatius, etc. Tots aquests missatges que apareixeran per pantalla no farà distinció de gènere. És a dir, en comptes de mostrar Benvingut o Benvinguda, o prioritzar el gènere masculí "Benvingut" la pantalla mostrarà un missatge sense gènere que digui "Hola!".
- En la creació de la base de dades per a cada usuari, on cada targeta serà assignada a un usuari, es crearà un camp dintre de la taula usuaris per la identificació de gènere. O sigui, que en el moment que tingui accés l'usuari i sigui reconegut pel sistema, aquest donarà un missatge personalitzat en funció del gènere masculí, femení o no binari, amb la finalitat de no fer cap discriminació i agafar el camí de la inclusió social.
- Diferents solucions per a diferents limitacions funcionals, on es té en consideració limitacions de mobilitat, auditives o visuals:
 - En l'àmbit de mobilitat, el sistema permet col·locar Smartlock a qualsevol alçada. Donat que uns cables aniran connectats al pany elèctric (independentment de si és una persona molt alta o molt baixa, o amb cadira de rodes) podrà col·locar Smartlock a l'alçada que més li convingui i no limitar-se a l'alçada del pany. També, permet accedir utilitzant la targeta (té un rang d'acció més gran que el forat de la clau) o teclat per a persones amb dificultats de moviment.

- En l'àmbit auditiu, el sistema disposa d'ajudes visuals, on mostra diferents missatges informatius per pantalla. Un LED que ens avisa si la lectura de la targeta és correcta o incorrecte i si la porta està oberta, etc.
- En l'àmbit visual, el sistema disposa d'una ajuda auditiva que consta d'un altaveu que informa sobre la lectura de la targeta, errors, etc. I d'una marca que sobresurt en el lateral d'Smartlock perquè l'usuari pugui identificar on es troba el lector RFID.

Així doncs, aquest projecte no afavoreix una igualtat social a nivell d'oportunitats, donat que és un projecte que segurament no arribarà al tercer món, ni sigui aquest el target de públic destí, ja que requereix electricitat constant, internet, pany elèctric, etc. A més, tampoc ajuda a promoure canvis a nivell polític afavorint a les dones, nenes, etc. No obstant, aquest projecte sí que està més orientat a ajudar a complir alguns objectius d'ODS 10 (Reduced inequalities), concretament, en objectius com ara el 10.2, que vol empoderar i promoure la inclusió social interdependent d'edat, sexe, discapacitat, etc. On s'intenta, d'una banda, no crear discriminació d'orientació sexual no definint text masculista i afavorint la inclusió social. I d'altra banda, ajudar a persones amb discapacitat en el seu dia a dia, com pot ser arribar i sortir de casa, gràcies als punts mencionats anteriorment.

1.4 Enfocament i mètode seguit

Actualment, existeixen diferents solucions en el mercat que aborden aquest problema amb diferents solucions (obrint la porta amb wifi, Bluetooth, empremta dactilar, targetes identificatives, claus amb xip, etc.). No obstant això, els productes que es distribueixen no són de codi obert, alguns tenen diferents limitacions i els que ofereixen més opcions tenen un cost més elevat. Tenint en compte aquests fets, i que utilitzen circuits impresos propis, es torna molt complicat adaptar el producte existent per millorar-lo i adaptar-lo a les noves necessitats. Per aquest motiu, es pren la decisió de desenvolupar un producte nou, amb un programari lliure (IDE Arduino) i uns components electrònics genèrics, com són els microcontroladors d'Arduino i els components electrònics compatibles amb el seu ecosistema per tal de desenvolupar una nova solució que cobreixi noves necessitats i problemes.

1.5 Planificació del Treball

1.5.1 Anàlisi de requeriments.

- ✓ => Requisit realitzat completament.
- ✓ => Requisit realitzat de manera parcial.
- ✗ => Requisit no realitzat.

| Objectiu | Requisit | Descripció |
|----------|----------|---|
| OBJ01 | 001 ✓ | Quan s'arrenqui el sistema Smartlock, estigui en espera o en procés de verificació d'autenticació, el pany s'ha de mantenir en posició tancada (32°) o sense passar corrent al pany electromagnètic, prioritant la seguretat de l'accés a l'habitatge, el LED RGB apagat i mostrar un missatge amigable per pantalla. |
| | 002 ✓ | S'haurà de desenvolupar una funció encarregada d'obrir la porta (120°), aquesta funció haurà de ser capaç de controlar el pany electrònic, magnètic o servomotor de manera que pugui obrir o tancar l'accés a l'habitatge enviant un senyal al mateix. En el prototip s'enviarà una senyal d'apertura al servomotor. |
| | 003 ✓ | Si Smartlock detecta que hi ha un accés no autoritzat el sistema mantindrà la posició del motor en tancat. |
| OBJ02 | 004 ✓ | S'ha de connectar un lector RFID al sistema Smartlock i programar una funció de lectura que sigui capaç d'identificar el valor d'un RFID Tag a una distància entre 0 i 1.5 cm, i guardar-ho en una variable temporal mentre es fa la comprovació d'autenticació. |
| | 005 ✓ | Si la lectura de la targeta es valida, el sistema haurà d'informar a l'usuari que s'ha detectat una lectura de targeta. Per fer-ho, il·luminarà breument el LED RGB de color verd, i emetrà un so curt de mínim 50db |

| | | |
|-------|----------|--|
| | | perquè es pugui escoltar encara que hi hagi un só ambient i menys de 90dB perquè no sigui molest. |
| | 006 ✓ | El sistema ha de ser capaç de detectar si la targeta té accés. Per fer-ho, disposarà d'una funció encarregada de recórrer els valors de la targeta llegida i comparar amb els valors de les targetes registrades en el sistema. Si troba alguna coincidència, cridarà a la funció encarregada d'autoritzar l'accés. En cas contrari, enviarà un senyal a la funció encarregada de denegar l'accés. |
| OBJ03 | 007 ✓ | S'haurà d'activar el xip ESP8266 de la placa d'Arduino i configurar-la de manera que sigui capaç de connectar-se al wifi de l'usuari, per fer-ho s'hauran d'introduir els paràmetres de nom de la xarxa i la contrasenya. |
| | 008 ✓ | Smartlock amb accés a internet gràcies a la connexió wifi que ofereix el router de l'usuari, ha de connectar-se al servidor Smartlock. De manera que pugui accedir a les funcions del servidor encarregades de comprovar l'autenticació de l'usuari i rebre una resposta en menys de 2 segons. |
| OBJ04 | 009 ✓ | Es farà el disseny de la BBDD del servidor utilitzant un diagrama d'entitat-relació tenint en compte les necessitats del sistema el qual necessitarà com a mínim emmagatzemar informació d'usuari, targeta i un registre històric d'esdeveniments. |
| | 010 ✓ | Es crearà el codi MySQL necessari per a la creació de les taules de la base de dades, així com la relació de les relacions que hi ha entre elles. |
| | 011 ✓ | S'habilitarà un servidor que tingui els requisits necessaris pel correcte funcionament de Smartlock. És a dir, ha de disposar d'un servidor de BBDD MySQL i d'un servidor PHP que permeti interactuar amb la BBDD i efectuar operacions. |
| | 012 ✓ | El servidor ha de tenir carregada les taules de BBDD d'Smartlock i desenvolupades les funcions PHP necessàries per verificar l'autenticació de la targeta de l'usuari, així com, enviar la resposta al sistema Smartlock (Arduino). |
| OBJ05 | 013 ✓ | El servidor d'Smartlock ha de disposar d'una taula on es guardarà tota la informació referent a l'historial d'accessos del sistema. |
| | 014 ✓ | Quan el sistema reconegui un accés autoritzat, crearà un registre a la taula d'historial que contindrà la informació de l'usuari que ha obert la porta, el número de la targeta i el dia i hora d'obertura. |
| OBJ06 | 015 ✓ | Smartlock tindrà muntat un sistema de bateries perquè en cas de tall de subministrament elèctric continuï funcionant, per fer-ho es muntarà unes bateries 18650 que permetin que el sistema funcioni durant almenys 2 h (aquest càlcul es realitzarà analitzant el consum energètic del sistema en estat d'espera i en obertura de porta). |
| | 016 ✓ | Es guardarà la informació d'unest targetes "mestre" que permetin l'accés al sistema sense fer una verificació d'accés a través d'un servidor. Al mateix temps, de manera local, es crearà una funció que si detecta que no hi ha accés al servidor, farà la comprovació d'autenticació de la targeta en l'àmbit local. |
| | 017 ✗ | Es muntarà un accessori de targetes SD que permet guardar l'arxiu d'informació de les targetes que tenen accés a nivell local. |
| OBJ07 | 018 ✓ | Es dissenyarà un panell d'administrador, amb una verificació d'usuari i contrasenya per tenir accés al mateix. |
| | 019 ✓ | Es crearà una opció que permeti a l'administrador crear, modificar i eliminar usuaris de la taula d'usuaris. |

| | | |
|--------------|----------|---|
| | 020 ✓ | Es crearà una opció que permeti activar o desactivar les targetes emmagatzemades al sistema (si un usuari no estar segur si ha perdut una targeta la pot desactivar temporalment). |
| | 021 ✓ | Hi haurà una funció que permetrà obrir la porta a distància des del mateix panell d'administrador. |
| OBJ08 | 022 ✓ | S'habilitarà una opció al sistema que permeti accedir a l'habitatge només utilitzant la targeta RFID o amb una doble autenticació. Primerament s'haurà de verificar la targeta i posteriorment es demanarà un codi d'accés vàlid. La modalitat escollida per l'usuari la transmetrà el servidor en el moment de rebre el valor de la targeta, i Smartlock rebrà la resposta en menys de 2 segons. |
| | 023 ✓ | Quan s'executi la funció de lectura de targeta, el sistema encendrà el LED RGB de color verd i emetrà un soroll superior a 50dB i inferior a 90dB per notificar a l'usuari que s'ha llegit una targeta. |
| | 024 ✓ | Quan s'executi la funció d'accés concedit, el sistema encendrà el LED RGB de color verd i emetrà un so superior a 50dB i inferior a 90dB perquè no sigui molest. |
| | 025 ✓ | Quan s'executi la funció d'accés denegat, el sistema encendrà el LED RGB de color vermell i emetrà un so d'error superior a 50dB i inferior a 90dB perquè no sigui molest. |

1.5.2 Llistat de tasques acotades.

1. Anàlisis (PAC 1 i 2)

En aquesta fase es realitzarà un anàlisis sobre la problemàtica que hi ha al mercat: anàlisis de la competència, anàlisis de les possibles solucions, definició d'objectius, estudis de viabilitat, etc. També, s'adquirirà tot el material necessari per realitzar el prototip.

2. Disseny de la solució

En aquesta fase es realitzarà un estudi de les solucions disponibles actualment en el mercat, així com una anàlisi de les diferents solucions possibles. Tenint en compte factors com ara si la tecnologia actual permet la solució plantejada, si és un projecte viable, etc.

3. Smartlock-Arduino

En aquesta fase es realitzarà tot el muntatge i programació de Smartlock en la part d'Arduino.

a. Muntatge bàsic dels elements

En aquesta tasca es farà l'esquema dels elements bàsics pel funcionament del projecte que seran una placa Arduino Mega2560+wifi, 1 Led RGB, resistències 220 Ohms, 1 boozzer, lector NFC, Teclat matriu 4x4, LCD, Servomotor, cables de connexió, 1 potenciòmetre. Posteriorment, es realitzaran les connexions bàsiques pel seu funcionament, instal·lació de l'IDE d'Arduino i programació de la posada en marxa bàsica (controlar LED, Boozzer, declarar els pins, etc).

b. RFID

En aquesta tasca es connectarà el lector NFC a la placa Arduino, es crearà el codi i carregaran les llibreries necessàries per realitzar la lectura del RFID. Una vegada comprovat que pot dur a terme la lectura dels diferents identificador, es crearan unes variables que continguin la informació del RFID amb la finalitat de crear una funció que s'encarregarà de verificar si l'identificador que s'ha llegit es troba en el sistema o no. En cas afirmatiu, es realitzarà el procés d'obertura de porta que en aquest cas consistirà en encendre el LED de color verd i fer sonar el boozzer. En cas contrari, s'encendrà el LED de color vermell i el boozzer emetrà un só d'error.

- c. Teclat
En aquesta tasca es connectarà el teclat matriu 4x4, es crearà el codi i carregaran les llibreries necessàries per capturar l'entrada del teclat, així com fer una doble autenticació controlant que la contrasenya sigui correcte. Per fer això, prèviament es crearà una variable amb una contrasenya i es modificarà la funció encarregada de comprovar si la targeta llegida existeix en el sistema (tasca b) demanant a l'usuari que introdueixi la contrasenya correcte per donar l'accés. Si tot és correcte, donarà autorització com en el pas anterior (Led verd+boozer) o es denegarà si és incorrecte (LED vermell+Boozer).
 - d. LCD
En aquesta tasca es connectarà un LCD de cristall líquid, es crearà el codi i s'afegiran les llibreries necessàries per mostrar per pantalla missatges d'advertiment i informació sobre l'estat del Smartlock. Inicialment, donarà un missatge de benvinguda, i si es detecta una lectura de targeta, mostrarà per pantalla un missatge, demanant que l'usuari introdueixi la contrasenya. Si és correcte, mostrarà un missatge d'accés concedit, en cas contrari, mostrarà un missatge d'accés denegat. Passat uns segons, el sistema tornarà a mostrar el missatge de benvinguda.
 - e. Servomotor (pany electromagnètic)
En aquesta tasca es connectarà un servomotor que simularà ser un pany elèctric o electromagnètic, i es crearà el codi necessari poder controlar la seva posició, segons si està obert el pany o tancat. Es modificarà la funció encarregada de donar accés i denegar. Ara apart de mostrar el LED de color verd, emet un so i mostra per pantalla el missatge d'accés concedit. També, mourà el servomotor (simularà el pany elèctric) en la posició d'obert i passat un temps tornarà a la seva posició original de tancat. En canvi, si no té accés el servomotor no farà res.
 - f. Wifi
En aquesta tasca es controlarà el xip wifi que té incorporat Arduino, es crearà el codi i carregaran les llibreries necessàries pel seu funcionament, de manera que Smartlock es pugui connectar a la xarxa wifi de casa, així que es realitzaran les proves necessàries per verificar la seva connexió a la xarxa local.
 - g. Tancament del sistema
En aquesta tasca es muntarà tot el circuit de SmartLock dintre d'una capça amb la finalitat d'evitar tot el cablejat vist, així com assegurar que els diferents components del sistema no es desconnectin i es pugui col·locar penjat a l'entrada de la porta.
4. Servidor
- a. Disseny BBDD
En aquesta tasca es definirà el disseny de la BBDD del sistema Smartlock, així com les connexions entre les diferents taules (1-1, 1-N, N-M), les primary keys, etc.
 - b. Instal·lació i creació de la BBDD
Es realitzarà la instal·lació d'un servidor MySQL, posteriorment es desenvoluparà el codi necessari per crear totes les taules de la BBDD tenint en compte el disseny creat en la tasca anterior (a), així com unes primeres entrades a la BBDD per poder fer les proves posteriors, com ara introducció de targetes vàlides, contrasenyes, etc.
 - c. Instal·lació i creació de PHP

Es farà la instal·lació de PHP al servidor. Seguidament, es desenvoluparà el codi necessari (funcions) per fer les consultes de verificació de la targeta, contrasenya, control d'usuaris, historial, etc. De manera que amb el PHP es puguin realitzar totes les gestions necessàries a la BBDD connectant PHP a la BBDD.

d. Connexió

Aquí es connectarà Arduino amb el servidor, i es modificarà el codi de manera que les comprovacions d'autenticació de les targetes ara notificaran a funcions PHP on s'enviaran els paràmetres de la targeta i contrasenya (en cas de doble autenticació) i esperarà la resposta del servidor per donar accés o denegar.

e. Panell de control

Es crearà un panell de control que permeti a l'administrador donar d'alta noves targetes, donar de baixa, veure un historial, etc. Per fer-ho, es desenvoluparà una pàgina web amb HTML, CSS i PHP. En la part HTML i CSS, es desenvoluparà tota la part visual del panell de l'administrador, i es crearan botons que enviaran un senyal a les funcions de PHP desenvolupades en la tasca c, per poder administrar el sistema SmartLock.

5. Testatge i solucions de problemes

a. Testatge

Es realitzarà el testatge del funcionament d'Smartlock amb la finalitat de trobar possibles errors del sistema centrant els esforços en intentar enganyar el sistema. És a dir, provar la lectura de targetes e introduir contrasenyes que pertanyen a un altre, proves de rang de lectura de la targeta, ús de diferents tipus de targetes NFC, control del historial, etc.

b. Solució de problemes

En aquesta tasca es buscaran les solucions dels problemes detectats durant la fase de testatge (a).

6. Memòria del projecte

a. Realització de la memòria del projecte.

Durant tot el projecte s'anirà ampliant, creant i modificant la memòria, afegint observacions, canvis en el cronograma, problemes trobats, etc.

b. Entrega de la memòria.

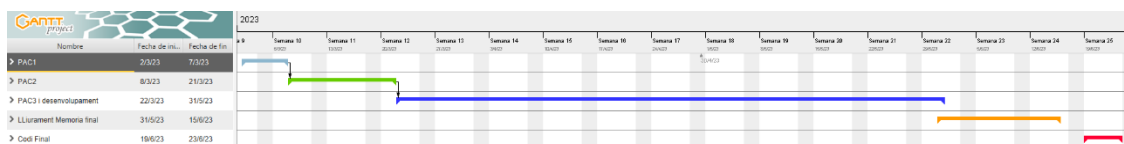
Es realitzarà l'entrega de la memòria de projecte.

c. Entrega de codi.

Es realitzarà l'entrega del codi final del projecte.

1.5.3 Diagrama de Gantt.

A continuació, es pot observar el diagrama de Gantt inicial resumit, que mostra els temps plantejats pel desenvolupament de cada fase:



Il·lustració 2 Diagrama de Gantt inicial resumit

Després, es pot veure una taula amb tot el detall del temps d'execució de cadascuna de les tasques, així com la data d'inici i final previstes en cada cas, tenint en compte les dates claus d'entrega de les diferents PACS.

Aquesta taula conté tant la planificació inicial, com la planificació final, amb les desviacions que s'han esdevingut durant el projecte. D'aquesta manera es pot visualitzar les tasques i desviacions que han aparegut.

- ✓✓ => Planificació inicial i planificació final són iguals. Tasca realitzada amb èxit
 ✗✓ => Diferència entre planificació inicial i planificació final. Tasca realitzada amb èxit
 ✗✗ => Diferència tasca inicial i planificació final. Tasca no realitzada.

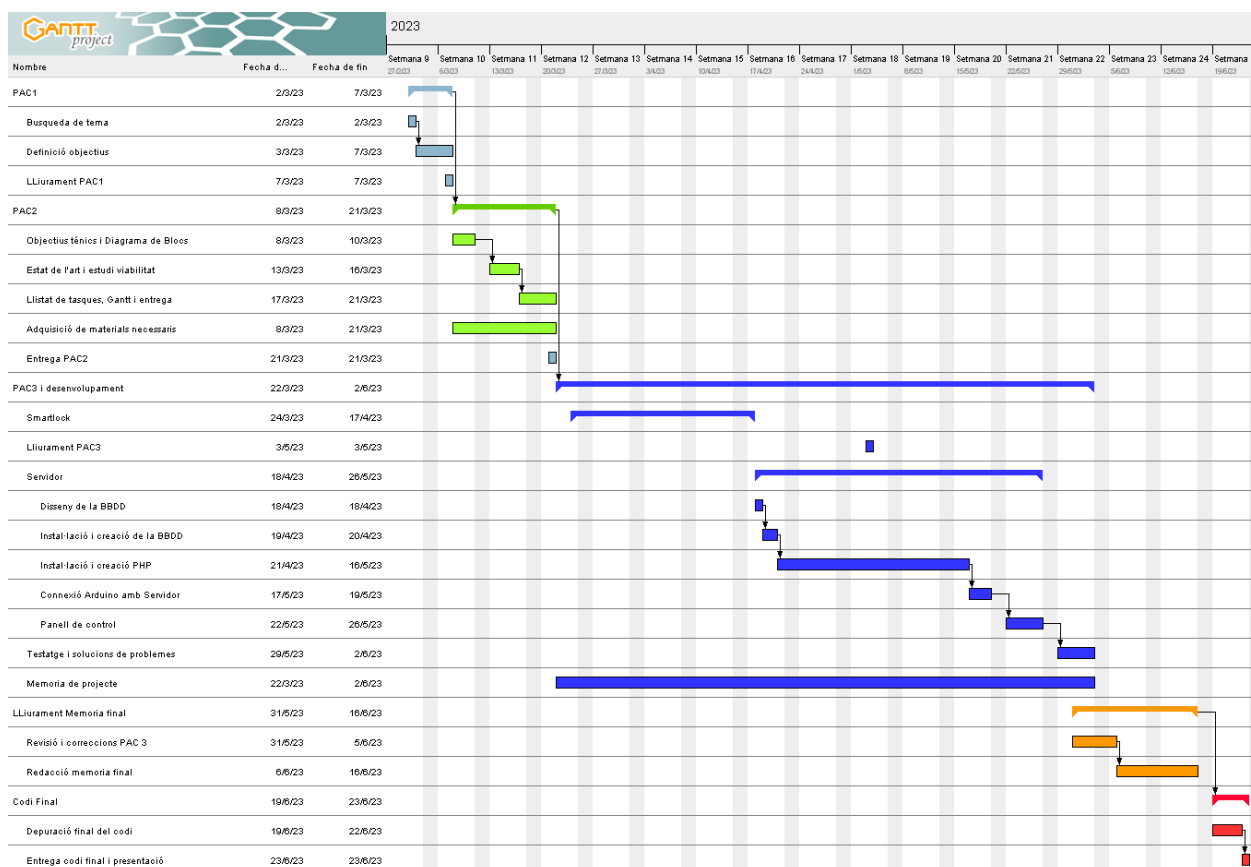
| Tasca | Planificació inicial | | Planificació final | |
|---|----------------------|------------|--------------------|------------|
| | Data inici | Data final | Data inici | Data final |
| PAC1 | 2/3/23 | 7/3/23 | 2/3/23 | 7/3/23 |
| Recerca de tema ✓✓ | 2/3/23 | 2/3/23 | 2/3/23 | 2/3/23 |
| Definició d'objectius ✓✓ | 3/3/23 | 7/3/23 | 3/3/23 | 7/3/23 |
| LLiurament PAC1 ✓✓ | 7/3/23 | 7/3/23 | 7/3/23 | 7/3/23 |
| PAC2 | 8/3/23 | 21/3/23 | 8/3/23 | 21/3/23 |
| Objectius tècnics i Diagrama de Blocs ✓✓ | 8/3/23 | 10/3/23 | 8/3/23 | 10/3/23 |
| Estat de l'art i estudi de viabilitat ✓✓ | 13/3/23 | 16/3/23 | 13/3/23 | 16/3/23 |
| Llistat de tasques, Gantt i entrega ✓✓ | 17/3/23 | 21/3/23 | 17/3/23 | 21/3/23 |
| Adquisició de materials necessaris ✓✓ | 8/3/23 | 21/3/23 | 8/3/23 | 21/3/23 |
| Entrega PAC2 ✓✓ | 21/3/23 | 21/3/23 | 21/3/23 | 21/3/23 |
| PAC3 i desenvolupament | 22/3/23 | 2/6/23 | 22/3/23 | 2/6/23 |
| Smartlock ✓✓ | 24/3/23 | 17/4/23 | 24/3/23 | 17/4/23 |
| Muntatge bàsic dels elements ✓✓ | 24/3/23 | 31/3/23 | 24/3/23 | 31/3/23 |
| RFID ✓✓ | 3/4/23 | 5/4/23 | 3/4/23 | 5/4/23 |
| Teclat ✓✓ | 6/4/23 | 7/4/23 | 6/4/23 | 7/4/23 |
| LCD ✓✓ | 10/4/23 | 11/4/23 | 10/4/23 | 11/4/23 |
| Servomotor ✓✓ | 12/4/23 | 12/4/23 | 12/4/23 | 12/4/23 |
| Wifi ✓✓ | 13/4/23 | 17/4/23 | 13/4/23 | 17/4/23 |
| Lliurament PAC3 ✓✓ | 3/5/23 | 3/5/23 | 3/5/23 | 3/5/23 |
| Servidor ✗✓ | 18/4/23 | 12/5/23 | 18/4/23 | 26/5/23 |
| Disseny de la BBDD ✓✓ | 18/4/23 | 18/4/23 | 18/4/23 | 18/4/23 |
| Instal·lació i creació de la BBDD ✓✓ | 19/4/23 | 20/4/23 | 19/4/23 | 20/4/23 |
| Instal·lació i creació PHP ✗✓ | 21/4/23 | 2/5/23 | 21/4/23 | 16/5/23 |
| Connexió Arduino amb Servidor ✗✓ | 3/5/23 | 5/5/23 | 17/5/23 | 19/5/23 |
| Panell de control ✗✓ | 8/5/23 | 12/5/23 | 22/5/23 | 26/5/23 |
| Testatge i solucions de problemes ✗✓ | 15/5/23 | 19/5/23 | 29/5/23 | 2/6/23 |
| Memòria de projecte ✓✓ | 22/3/23 | 2/6/23 | 22/3/23 | 2/6/23 |
| LLiurament Memòria final | 31/5/23 | 16/6/23 | 31/5/23 | 16/6/23 |
| Revisió i correccions PAC 3 ✓✓ | 31/5/23 | 1/6/23 | 31/5/23 | 5/6/23 |
| Redacció memòria final ✓✓ | 2/6/23 | 16/6/23 | 6/6/23 | 16/6/23 |
| Codi Final | 19/6/23 | 23/6/23 | 19/6/23 | 23/6/23 |
| Depuració final del codi i presentació ✓✓ | 19/6/23 | 22/6/23 | 19/6/23 | 22/6/23 |
| Entrega codi final i presentació ✓✓ | 23/6/23 | 23/6/23 | 23/6/23 | 23/6/23 |

En la taula anterior es pot observar que totes les tasques planificades s'han realitzat, i la majoria dintre de les dates planificades. No obstant això, hi ha una petita modificació en algunes tasques principalment les referents amb el servidor i la connexió d'aquest amb d'Arduino, pels següents motius:

- Van haver-hi complicacions en el desenvolupament de l'arxiu de verificació on no donava els resultats esperats.
- Complicacions en aconseguir la comunicació entre Arduino Mega i el xip ESP8266 (port Serial).
- La redacció de la memòria durant el desenvolupament, va portar més temps de l'esperat. A més, no es va tenir en compte el temps que s'havia de dedicar per aplicar els canvis del feedback de la PAC2.

Per arreglar aquest desviament, i tenint en compte que la memòria de projecte estava molt avançada, es va treure una setmana de tasca de redacció de la memòria final, i es van dedicar més hores diàries de les planificades.

També van haver-hi problemes en la connexió del lector RFID, aquest no venia amb els pins soldats i es desconnectava amb facilitat, no obstant això, per mantenir la planificació inicial es van dedicar més hores cada dia. El diagrama de Gantt final del projecte és el següent:



Il·lustració 3 Diagrama de Gantt detallat

NOTA: En la planificació no s'han tingut en compte els festius i els caps de setmana, donat que el desenvolupament del projecte també es realitzarà durant aquest període.

1.6 Breu sumari de productes obtinguts

Al següent quadre es pot observar tots el productes obtinguts amb la finalitat de desenvolupar el prototip SmarLock, així com una petita fotografia del producte de cada producte.

| Producte | Fotografia |
|--|--|
| Placa Arduino Mega 2560+ ESP8266 |  |
| LED RGB |  |
| Kit MFRC-522 RC522 RFID NFC + Targeta + Clauer |  |
| Teclat Matriu 4x4 |  |
| Piezoelectric Speaker |  |
| Panell Cristall Liquid 16x2 |  |
| Potenciòmetre |  |
| ServoMotor |  |
| Resistencies 220 Ohms |  |
| Connectors |  |
| Protoboard |  |

1.7 Estudi de viabilitat

1.7.1 Viabilitat tècnica

Després de trobar una possible solució al problema social plantejat anteriorment, s'ha de valorar si aquesta solució que es planteja es pot arribar a crear amb la tecnologia existent.

Per verificar-ho, aprofitant que Arduino té darrere una comunitat molt gran d'usuaris, s'han buscat projectes que poguessin fer accions similars com: llegir una targeta RFID, mostrar text per pantalla, utilitzar un teclat, verificar autenticació per RFID, etc. I s'han trobat diferents projectes que ofereixen accions similars a les que es plantegen en Smartlock com són: accionar un relé d'una porta amb l'ajuda d'una autenticació per contrasenya [3], fer un control d'assistència dels alumnes amb l'ajuda d'una targeta RFID [4], un sistema d'assistència connectat a un servidor [5], etc.

Després de trobar-ne diferents projectes que donaven solucions parcials al que ofereix Smartlock, s'ha buscat si existia en el mercat els components elèctrics necessaris per desenvolupar la solució proposada i que fossin compatibles amb l'ecosistema Arduino. Així doncs, s'han trobat tots els components que es necessitaven i que es llistaran en l'apartat de viabilitat econòmica.

Finalment, també s'ha fet un anàlisi en l'àmbit tècnic, els requisits necessaris per a fer la connexió a un servidor i connectar Arduino a aquest servidor. A més, s'ha comprovat que a part d'existir projectes com els mencionats anteriorment, que fan ús de base de dades i es connecten a un servidor, també s'ha comprovat que les possibilitats que ofereix PHP i MySQL permeten àmpliament resoldre les necessitats de Smartlock.

1.7.2 Viabilitat econòmica

Per fer un estudi de viabilitat econòmica del projecte el primer que es farà és realitzar un pressupost de material necessari i a continuació es tindrà en calcularan les hores de desenvolupament. Al ser un projecte tecnològic amb productes que es poden adquirir en

l'àmbit internacional, es faran dos tipus de pressupost, amb compres dintre d'Espanya i amb compres a la Xina, per verificar la diferència de preus així com assegurar que compleixin els estàndard. Tampoc es tindrà en compte els possibles descomptes per volum de compra o marques alternatives.

| Espanya | | |
|--------------------------------|---------------|------------------------|
| Material | Preu | URL |
| Arduino mega2560+ESP8266 | 13.92€ | Enllaç |
| LED RGB | 0.60€ | Enllaç |
| Lector RFID + Targetes | 1.95€ | Enllaç |
| Resistències (100 unitats) [6] | 4.99€ | Enllaç |
| Teclat matriu 4x4 | 4.99€ | Enllaç |
| Boozer | 0.80€ | Enllaç |
| Panell LCD Cristall Liquid | 5.79€ | Enllaç |
| Potenciòmetre | 1.30€ | Enllaç |
| Connectors | 1.82€ | Enllaç |
| Protoboard | 4.99€ | Enllaç |
| TOTAL | 41.15€ | |
| Opcional [7] | | |
| Mòdul targeta SD | 5.49€ | Enllaç |
| Targeta 16gb | 5.73€ | Enllaç |
| Mòdul carrega externa | 11.58€ | Enllaç |
| 2xBateria 18650 3500mAh | 10.99€ | Enllaç |
| TOTAL | 74.94€ | |

| Xina | | |
|----------------------------|---------------|------------------------|
| Material | Preu | URL |
| Arduino mega2560+ESP8266 | 13.86€ | Enllaç |
| LED RGB (Pack de 10) | 1.35€ | Enllaç |
| Lector RFID + Targetes | 1.24€ | Enllaç |
| Resistències (100 unitats) | 0.79€ | Enllaç |
| Teclat matriu 4x4 | 0.37€ | Enllaç |
| Boozer (10 unitats) | 0.49€ | Enllaç |
| Panell LCD Cristall Liquid | 1.50€ | Enllaç |
| Potenciòmetre | 0.48€ | Enllaç |
| Connectors | 1.02€ | Enllaç |
| Protoboard | 1.23€ | Enllaç |
| TOTAL | 22.33€ | |
| Opcional | | |
| Mòdul targeta SD | 0.62€ | Enllaç |
| Targeta 8gb | 2.10€ | Enllaç |
| Mòdul carrega externa | 3.94€ | Enllaç |
| 2xBateria 18650 3500mAh | 5.67€ | Enllaç |
| TOTAL | 34.66€ | |

NOTA: La part del pany elèctric en aquest cas, pel prototip, es farà utilitzant un servomotor el qual no està contemplat al pressupost, donat que hi ha molts sistemes diferents de panys elèctrics i electromagnètics, i es possible que el client ja disposi d'un instal·lat a casa que sigui compatible amb Smartlock. També, es contempla la possibilitat d'afegir a futur un accessori que es fixi en el pany de la porta tradicional, i un servomotor connectat a Smartlock perquè giri la clau per obrir o tancar la porta. No obstant això, aquest accessori no es contempla inicialment en el prototip del projecte. D'altre banda, en la valoració del pressupost de material, en alguns casos hi han despeses d'enviament, aquestes no s'han contemplat donat que a partir de cert import l'enviament és gratuït.

Com el projecte es farà amb el sistema d'Arduino i amb components electrònics molt econòmics, sobretot a la Xina (en els pressupostos anteriors no es tenen en compte possibles descomptes per quantitat), es té un producte que pot tenir un preu molt competitiu. Ara bé, si el producte té acollida, es podrien crear diferents línies de producte:

- **Per a professionals i empreses amb materials de major qualitat:** es podria reaprofitar part del codi i materials, així com fer algunes modificacions per poder utilitzar aquest mateix sistema, però en aquesta cas, per control horari de treballadors (fixatge), de manera que el cost econòmic d'hores de desenvolupament no seria elevat i es podria oferir alhora un servei d'instal·lació i manteniment. Així com, la possibilitat d'afegir funcionalitats a mida, per poder obtenir beneficis més enllà del marge del producte.
- **Per usuaris amb una versió més limitada,** on SmartLock treballaria de manera local amb una targeta SD, la qual limitaria les seves opcions, però no necessitaria tenir coneixements avançats ni contractar un servei d'instal·lació.

A continuació, es farà una valoració econòmica de les despeses d'ús del programari:

| Programari | |
|-----------------|---------|
| Nom | Preu |
| IDE Arduino | Gratuït |
| Notepad++ | Gratuït |
| GanttProject | Gratuït |
| MySQL WorkBench | Gratuït |

Una vegada calculat el programari, per valorar la viabilitat econòmica del projecte es necessitarà tenir en compte el cost de desenvolupament del projecte:

| Cost de mà d'obra | | | |
|---|--------------|-------------|-----------------|
| Tasca | Hores | Preu/Hora | Total |
| Estudi | 48H | 28.12€ | 1.349,76€ |
| Disseny del circuit, connexions, etc | 12H | 28.12€ | 337,44 € |
| Desenvolupament Smartlock (Arduino) | 56H | 28.12€ | 1.574,72 € |
| Servidor + Interacció + Panell de control | 80H | 28.12€ | 2.249,60 € |
| Proves funcionals | 16H | 28.12€ | 449,92 € |
| Redacció documentació | 40H | 28.12€ | 1.124,80 € |
| | Total | 252H | 7086,24€ |

Per calcular el preu hora, s'ha tingut en compte que s'haurà de contractar a un enginyer programador i a un analista. Si es fa una cerca del salari mig brut d'un programador experimentat a Espanya, és de 38.000 € bruts. Llavors, es farà el càlcul de quin cost repercuteix a l'empresa tenint en compte una jornada laboral anual de 1750 hores.

| Càlcul cost preu/hora | |
|------------------------------|---------|
| Salari brut | 38.000€ |
| Seguretat Social [8] | 11.210€ |
| Cost per hora (1750h/anuals) | 28.12€ |

Al mateix temps, com es necessitarà un servidor extern, es valorarà el cost del manteniment anual d'aquest.

| Servidor | |
|-------------------|----------------------|
| Domini Anual | 5€ |
| Hosting Anual [9] | 37.07€ |
| Total | 42.08€/Anuals |

A més, com el programari necessita la introducció d'alguns paràmetres al sistema com són la contrasenya Wi-Fi de l'usuari, la creació de la BBDD i el servidor on es connectarà el sistema domòtic, és molt probable que l'usuari necessiti un instal·lador per a realitzar la seva configuració. Com s'ha de programar el firmware a mida per cada client, d'aquesta

forma es podrà tenir un control sobre la llicència del producte i oferir un producte integral (hardware+software). Tanmateix, es pot optar per una opció més limitada on l'usuari no farà ús d'un servidor extern, sinó que quedarà tot emmagatzemat a la targeta SD del sistema Smartlock i amb unes funcionalitats més limitades on només es podrien activar i desactivar targetes i les contrasenyes. No obstant això, es farà el càlcul tenint en compte que l'usuari utilitzarà el servidor i el producte amb les funcions addicionals comprat a la Xina. En resum:

| Totals | |
|--------------------|------------------|
| Material Smartlock | 34,66€ |
| Software | 0,00€ |
| Mà d'obra | 7.086,24€ |
| Total | 7.120,90€ |

Tenint en compte tot el mencionat anteriorment, així com els preus dels productes de la competència, si s'aplica un marge d'un 100% sobre el cost del producte, es tindrà el següent preu de venda:

$$\text{Preu venda} = 34.66€ \times 100\% = 69.32€$$

Per fer-ho més atractiu, es posarà a un preu de venda de 69.99€. Tot i això, s'haurà de tenir en compte que la majoria d'usuaris podria necessitar una configuració/instal·lació que incrementaria el preu final si es donés el cas.

Finalment, ara es valorarà el nombre d'unitats que s'haurien de vendre el primer any per recuperar la inversió inicial tenint en compte el cost del manteniment dels servidors anualment i un marge de benefici per cada venda de 35.33€.

| Quadre d'amortització | | | | | |
|-----------------------|---------|---------------|---------------|------------------|---------------|
| Vendes | Marge | Cost Servidor | Cost Projecte | Benefici Unitats | Benefici Real |
| 100 | 35,33 € | 42,08 € | 7.120,90 € | 3.533,00 € | -3.545,82 € |
| 150 | 35,33 € | 42,08 € | 7.120,90 € | 5.299,50 € | -1.779,32 € |
| 200 | 35,33 € | 42,08 € | 7.120,90 € | 7.066,00 € | -12,82 € |
| 250 | 35,33 € | 42,08 € | 7.120,90 € | 8.832,50 € | 1.753,68 € |
| 300 | 35,33 € | 42,08 € | 7.120,90 € | 10.599,00 € | 3.520,18 € |
| 350 | 35,33 € | 42,08 € | 7.120,90 € | 12.365,50 € | 5.286,68 € |

Com es pot observar a la taula anterior, a partir de més de 200 unitats venudes en un any, ja es podria recuperar la inversió inicial del producte. Si es valora que l'any 2020, segons l'Institut Nacional d'Estadística d'Espanya, hi havia 25.882.055 d'habitatges, això significaria que per recuperar la inversió inicial del projecte es necessitaria que el 0,0008% dels habitatges totals d'Espanya tinguessin instal·lat un Smartlock.

No obstant això, les xifres anteriors serien en un cas idíl·lic, i no es pot tenir únicament en compte el nombre d'habitatges que hi ha en Espanya, sinó més factors que poden determinar el potencial Smartlock com són:

- **Persones [10]:** El perfil de client de Smartlock es centra principalment entre joves i persones de mitjana edat. És a dir, entre els 18 i els 50 anys, que són la part de la població que més confiança té en les noves tecnologies i l'ús d'internet (un 68.5% dels espanyols afirma confiar molt o bastant en internet).
- **Habitatges amb connexió d'internet [11]:** a Espanya, hi ha més de 5,5 milions de persones que no tenen accés a internet. Així doncs, el 88.1% de la població té punts d'accés a Internet (Smartlock necessita una connexió a internet per funcionar correctament), per tant, no tots els habitatges totals de l'Estat podrien tenir Smartlock.
- **Tendències del mercat [12]:** actualment, el mercat espanyol està en una tendència alcista envers la domòtica, on diferents estudis pronostiquen que la domòtica pot créixer fins a un 300% fins al 2024, on s'estima que el 20% de les noves construccions i ja existents tinguin algun dispositiu connectat. A més, el 20% dels enquestats estarien disposats a invertir en domòtica entre 500-1000 € i un 25% més

de 1.000 €, imports molts inferiors al que necessitarien aquest usuaris per instal·lar Smartlock. Un altre detall a tenir en compte, es que els factors que més valoren els usuaris que volen optar per sistemes domòtics són l'estalvi energètic, la seguretat i la comoditat, aquí Smartlock es situa en un dels factors que més valoren els usuaris, la seguretat.

Si es té en compte tot l'estudi anterior, així com la tendència del mercat, el target d'usuaris, etc. Es considera que el projecte Smartlock és un projecte viable econòmicament, encara que s'ha de tenir en compte que els càlculs estan fets sobre unes despeses i beneficis nets, és a dir, que no s'ha tingut en compte el cost de producció del producte, màrqueting, embalatge, distribució, etc. Per tant, el nombre final de productes venuts necessaris per recuperar la inversió seria superior al mencionat anteriorment, però si es té en compte el potencial de clients del mercat, així com un marge de beneficis per producte elevat. Encara que s'incrementi el cost final del producte pels motius mencionats anteriorment i que aquests poguessin arribar fins a un 30% (no sent conservador) de penalització sobre el marge del producte, es necessitaria unes vendes inferiors a 300 unitats per cobrir la inversió inicial. Una quantitat d'unitats que continua sent molt assumible si es té en compte el nombre de clients potencials.

De cara a la viabilitat a llarg termini del projecte, també es té en compte que si el producte només pot aportar beneficis derivats de les vendes. És possible que amb els anys, les vendes baixin i pugui arribar un moment que el projecte no sigui sostenible econòmicament degut a les despeses dels servidors, actualitzacions del sistema, manteniment, etc. Per aquest motiu, i com s'ha mencionat anteriorment, també es contempla de cara al futur, una línia de producte basada en Smartlock per a petites empreses que vulguin tenir un control d'obertura de portes i alhora un control horari dels treballadors (hora d'arribada, sortida, absència, etc.). Aquest producte seria compatible amb Smartlock i necessitaria unes millores en l'àmbit de Software, les quals implicarien una inversió inicial poc elevada, i alhora es crearien unes tarifes mensuals segons diferents criteris com el nombre de treballadors màxims, opcions, etc.

1.7.3 Viabilitat Legal

Les eines i sistemes de desenvolupament, són l'IDE d'Arduino, el qual és de programari lliure, una base de dades en MySQL (Llicència GPL) i PHP també amb llicència lliure. A més les llibreries que s'utilitzaran per interactuar amb els diferents components seran també de llicència lliure amb la finalitat de no incrementar les despeses del desenvolupament del projecte.

A escala legal, no existeix cap problema de llicències donat que el projecte es realitzarà amb llicències de codi obert i el desenvolupament del projecte serà realitzat per part de l'autor.

1.8 Breu descripció dels altres capítols de la memòria

En el primer punt, s'ha fet una anàlisi dels motius o problemes que s'han trobat a la societat i pel qual s'ha decidit desenvolupar un producte anomenat Smartlock, que intenta donar solució a aquest problema. També, s'han analitzat les possibles solucions, els objectius que té el projecte, la seva viabilitat i la planificació que s'ha fet.

A continuació, en la memòria del projecte, es trobaran altres punts importants que permetran al lector entendre millor el funcionament d'Smartlock, com s'ha desenvolupat, quines conclusions s'han esdevingut després de la realització del prototip del projecte, etc.

Punt 2: es realitzarà una anàlisi de l'estat de l'art, on es farà una cerca de productes de la competència que ofereixen diferents solucions al mateix problema que vol donar solució Smartlock. Així com una anàlisi de la tecnologia que s'utilitzarà en el projecte.

Punt 3: s'entrarà en detall sobre el desenvolupament del projecte, i es dividirà principalment en 3 parts. En primer lloc, s'explicarà i s'entrarà en detall sobre tots els components d'Arduino Smartlock

i en el desenvolupament i funcionament del dispositiu. En segon lloc, es detallarà com s'ha fet la configuració del servidor i com s'ha desenvolupat la part de PHP i MySQL encarregada de fer les comprovacions necessàries d'autenticació. I finalment, en tercer lloc, s'exposarà amb l'ajuda de diagrames funcionals quin és el funcionament d'Smartlock.

Punt 4: es parlarà sobre els resultats finals del projecte, la campanya de proves que s'ha realitzat, etc.

Punt 5: es faran les conclusions finals a les quals s'ha arribat en finalitzar el projecte. I també, dels treballs a futur i oportunitats de millora a les que s'ha arribat al finalitzar el projecte.

Finalment, la memòria del projecte disposarà d'una part de glossari amb l'explicació dels conceptes més destacats del projecte i una bibliografia d'on s'ha extret tota la informació pel desenvolupament del projecte.

2. Estat del Art

Actualment, es poden trobar diferents solucions d'ús domèstic que donen resposta a aquesta problemàtica, on l'afronten de la següent manera:

- Utilitzar el pany actual fixant el pany intel·ligent al pany actual i connectant la clau original al pany intel·ligent (s'encarrega de girar la clau).
- Obertura per comandament a distància.
- Obertura per empremta dactilar.
- Obertura per teclat numèric.
- Obertura per smartphone.
- Obertura amb clau tradicional que té incorporat un xip identificatiu (semblant a les claus del cotxe).
- Connexió per Bluetooth, Wi-Fi, Z-Wave i/o Zigbee.

2.1 Benchmarking

En aquest apartat s'analitzarà les diferents solucions que ofereix la competència, tenint en compte les seves característiques principals, i comparant-les finalment amb la solució Smartlock.

Nuki Smart Lock 3.0 Pro (PVP. Recomanat 279€):



Il·lustració 4 Nuki Smart Lock 3.0 Pro

- Aquest sistema utilitza el pany actual de l'usuari (no és compatible amb tots els panys tradicionals).
- Funciona amb bateria recarregable.
- Permet connexió per Bluetooth i Wi-Fi (xifrat d'extrem a extrem).
- Es pot controlar l'obertura de la porta a través d'un dispositiu mòbil o amb la clau original.
- Permet la creació de rutines i control per assistent de veu.
- Permet connectar un accessori a part nomenat Nuki Keypad (159€), que permet el control d'accés per empremta dactilar i codi.
- Tota la configuració es realitza a través de la App de la companyia.
- Té un sensor a la porta per verificar si es troba oberta o tancada.

Tesa Assa Abloy ENTR (PVP Recomanat 350€):



Il·lustració 5 Tesa Assa Abloy

- Aquest sistema utilitza el seu propi pany (no és compatible amb tots els cilindres 30x30 i es possible que l'usuari hagi de realitzar un nou forat).
- Per obrir es pot utilitzar la clau, comandament a distància (max 20) o smartphone.
- Amb accessoris addicionals es pot obrir amb empremta dactilar i codi.
- Funciona a bateria amb una duració aproximada de 3 mesos.
- Funciona per Bluetooth amb un xifratge AES-128.
- Es pot configurar des del propi pany.

ZKTeco ZK-AL20DB (PVP Recomanat 250€)



II-lustració 6 ZKTeco ZK-AL20DB

- Aquest sistema es pot integrar en la majoria de portes residencials de 30 a 54 mm (s'ha de verificar compatibilitat).
- La identificació és per empremta dactilar, contrasenya o mòbil.
- Utilitza 4 piles AAA.
- Capacitat per a fins a 100 usuaris.
- Connexió per Bluetooth.
- Manilla reversible.

Xiaomi Vima Smart Lock (PVP Recomanat 50€)



II-lustració 7 Vima Smart Lock

- Aquest sistema es pot integrar a la majoria de portes (cilindres 30x30).
- Utilitza únicament claus amb xip integrat (similar al cotxe) inclouen 5.
- Les claus es poden programar per temps, activar i desactivar.
- Utilitza connexió Zigbee, per configurar el pany serà necessari tenir el gateway de Xiaomi, sincronitzar el pany i a continuació, fer tota la configuració des de l'app MiHome.
- Es poden crear rutines.
- Avisa en cas de detectar forceig.
- Funciona a bateria (duració aproximada 6 mesos).

Tot seguit es mostra una taula resum amb les característiques principals dels models analitzats envers d'Smartlock:

| | | Nuki 3.0 | Tesa Assa | Zkteco | VimaSmart | Smartlock |
|--------------|------------------|-----------------|------------------|---------------|------------------|------------------|
| Autenticació | Smartphone | SI | SI | SI | NO | User/Admin |
| | Empremta | Accessori | Accessori | SI | NO | NO |
| | Contrasenya | Accessori | Accessori | SI | NO | SI |
| | Clau | SI | SI | NO | SI | Accessori |
| | RFID | NO | NO | NO | NO | SI |
| | Comandament | NO | SI | NO | NO | NO |
| Alimentació | Cable | NO | NO | NO | NO | SI |
| | Bateria | SI | SI | SI | SI | Accessori |
| Connectivita | Bluetooth | SI | SI | SI | NO | NO |
| | Wifi | SI | NO | NO | NO | SI |
| | Assistent de veu | SI | NO | NO | NO | NO |
| | Zigbee | NO | NO | NO | SI | NO |

Així doncs, després d'analitzar les diferents solucions que es troben en el mercat, es pot treure les següents conclusions envers la competència vs Smartlock.

Avantatges d'Smartlock:

- Tots els panys analitzats funcionen únicament amb bateria o piles, per tant l'usuari ha d'estar pendent de l'estat de la bateria, donat que si es queda sense, no podrà accedir a la habitatge. En canvi, SmartLock funciona amb connexió directa a l'electricitat i incorpora una bateria addicional per evitar qualsevol tall de subministrament elèctric.
- Les solucions que utilitzen el pany actual no solucionen el problema de seguretat en cas de pèrdua de la clau, donat que si l'usuari perd la clau, es veurà obligat igualment a canviar tot el pany per un de nou, per evitar un accés a l'habitatge no autoritzat.
- Les solucions que usen el pany actual no són compatibles amb tots els models de pany.
- Les solucions que usen únicament el mòbil per obrir el pany, corren el risc que si l'usuari s'ha quedat sense bateria al mòbil, no podrà accedir a l'habitatge.

- Les solucions que utilitzen una clau amb xip si que permeten activar i desactivar la clau, però no permeten obrir i tancar el pany a distància.
- Alguns dels sistemes analitzats no disposen d'un històric d'accessos.

Inconvenients d'Smartlock.

- Smartlock necessita tenir un pany elèctric o electromagnètic per funcionar, no serveix amb el pany tradicional a diferència de la solució que dona Nuki o Tesa.
- Smartlock necessita un servidor per funcionar, per realitzar la consulta de verificació d'accés (els panys que funcionen per Wi-Fi o Z-wave necessiten un hub i un servidor on connectar-se si es vol controlar a distància).
- Sistemes com ara el de Nuki, Tesa, VimaSmart o Zkteco són de més fàcil instal·lació. No cal portar cap cable d'alimentació fins al pany de la porta per a la seva alimentació, ja que funcionen a bateria.
- Alguns dels altres sistemes no necessiten cap element o dispositiu específic per la autenticació, utilitzen el mòbil (Nuki i Tesa), la empremta dactilar (Zkteco) o teclats incorporats en el dispositiu (Nuki i Tesa).
- Alguns models com ara, el de Nuki, es pot implementar amb assistents de veus com Google Home o Alexa, els quals permeten crear rutines o interactuar amb el pany amb la veu.
- No funciona amb empremta dactilar (com és el cas de Nuki amb un accessori o Zkteco), aquest sistema d'autenticació és molt segur i pràctic per l'usuari (és recomanable guardar l'empremta de més d'un dit perquè si l'usuari es fa un tall en el dit d'identificació el sistema podria no detectar-lo), ja que no es necessari l'ús d'un dispositiu extern o específic per el control d'accés.

En resum, com es pot observar hi ha diferents productes i empreses que busquen donar una solució a aquest problema social, cadascuna atacant el problema des de diferents solucions. No obstant això, no hi ha cap pany que sigui perfecte, com s'ha pogut comprovar en el punt anterior d'avantatges e inconvenients. Per aquest motiu, s'ha proposat la creació d'Smartlock, un producte que intenta aportar una solució diferent a les que es troben al mercat, una solució que a diferència de la resta aporta:

- Una doble autenticació si és necessari una major seguretat.
- No depèn exclusivament de bateries, té major dificultat d'instal·lació, però una vegada instal·lat, no s'ha d'estar pendent de que s'esgoti la bateria i no es pugui accedir a l'habitatge.
- No depèn únicament d'un smarhphone amb bateria (donat que si l'usuari es queda sense bateria no podrà accedir a l'app que permeti l'apertura de la porta). Al necessitar una identificació RFID si que es pot enganxar darrere del mòbil una enganxina RFID per poder obrir des del mòbil sense la necessitat de tenir bateria.
- És un sistema que permet, a futur, ampliar les seves possibilitats a un nivell més professional, on s'utilitzi alhora per obrir la porta i marcar un fixatge d'inici de la jornada laboral (control horari).
- Les solucions proposades per la competència no permeten controlar panys elèctrics o electromagnètics, només panys mecànics.
- La majoria d'ells no disposen d'un històric d'accessos.

2.2 Tecnologies utilitzades: avantatges i inconvenients

Una vegada analitzades les solucions que ofereix la competència, decidida quina solució aportarà Smartlock i de quines de les tecnologies actuals en farà ús pel seu desenvolupament. A la següent taula s'abordarà quins avantatges i inconvenients tenen les diferents tecnologies de les que estarà dotat Smartlock amb Arduino.

| Wi-Fi | |
|--|---|
| <p>Consum energètic baix.</p> <p>No requereix d'instal·lació de cables per que els elements tinguin comunicació.</p> <p>Es poden configurar diferents nivells de seguretat.</p> <p>Té una distancia de cobertura de senyal superior altres tecnologies com ara el BluetooH.</p> | <p>Si hi ha obstacles com ara parets, portes, finestres, etc poden provocar una atenuació de la senyal.</p> <p>A nivell de seguretat, les xarxes Wi-Fi són més propenses a rebre atacs d'usuaris no autoritzats a diferencies de les de per cable.</p> |
| RFID | |
| <p>Baix consum energètic.</p> <p>Baix cost econòmic de les etiquetes RFID passives.</p> <p>Baix pes i diferents formats disponibles d'etiquetes RFID.</p> <p>Llarga vida, la majoria d'etiquetes RFID no necessiten bateria, per tant, poden durar molt anys.</p> <p>Les plaques de circuit imprès d'acoblament inductiu o radiant gairebé no produeixen so.</p> <p>Fàcils d'utilitzar, no requereix coneixements previs per part de l'usuari.</p> | <p>El rang d'acció és molt curt degut a que no disposen d'una antena activa.</p> <p>Emmagatzematge limitat, al no requerir de bateria integrada, les etiquetes passives no poden emmagatzemar molta informació.</p> <p>Requereixen d'un lector d'alta potència per activar-se.</p> <p>Les etiquetes RFID passives no disposen de sensors (no es poden utilitzar aplicacions per a la seva detecció).</p> |
| LCD | |
| <p>Al ser retroiluminades s'aconsegueixen alts contrastos entre els colors i permet la seva visualització en condicions amb poca il·luminació.</p> <p>Tenen un cost menor que altres pantalles.</p> <p>Tenen un espessor bastant prim (comparat amb tecnologies ja obsoletes com ara la de raig catòdics).</p> <p>No produeixen radiació i tenen una dissipació de calor baixa.</p> | <p>No permet crear colors amb un alt grau de precisió.</p> <p>Si no es mira la pantalla des d'un grau de visibilitat correcte, les tonalitats poden canviar.</p> <p>El temps de resposta de la pantalla és lent.</p> <p>Problemes de manteniment, en general, les pantalles són de vidre que és més fàcil de trencar, i els píxels són molt petits. Això provoca que amb freqüència hi hagin trencaments de píxels individuals.</p> |

També s'han analitzat altres projectes TFG de la UOC (repositori O2) que utilitzessin tecnologies similars o solucions que es podrien aplicar en part, per solucionar aquest problema. S'han trobat els següents:

- “Control de acceso mediante NFC con Arduino”: desenvolupat per Miguel Viñas Gutiérrez, on desenvolupa un control de fixatge d'una empresa.
- “Control de marcatges NFC/RFID”: desenvolupat per Hector López Pujol, on utilitza la tecnologia RFID amb una Raspberry Pi per al control de fixatge d'una empresa.
- “Sistema de control d'accés via Bluetooth per a terminals Android”: desenvolupat per José Luis del Aguila on desenvolupa una aplicació per a Android amb connectivitat Bluetooth per a l'apertura de portes.

Aquests projectes utilitzen tecnologies similars a les de Smartlock i que donen respostes a altres problemes, però Smartlock aporta una solució diferent. En el cas dels projectes desenvolupats per Miguel Viñas Gutiérrez i Hector López Pujol, es centren en un control de fixatges, però no controlen obertura de portes, autenticació d'usuaris, doble autenticació, etc. A més, no permeten controlar l'obertura de la porta, ja que es centra en el control horari dels treballadors i aquest s'emmagatzemen a la base de dades. És cert que aquest concepte es planteja com una possible ampliació a futur d'Smartlock cap a una línia destinada a empreses, on a part del seu funcionament de pany intel·ligent, tingui la funció de control horari de treballadors, si aquesta es situa a l'entrada de l'empresa.

Respecte a la solució que planteja Jose Luis del Aguila, és un sistema més similar al que ofereix Smartlock, però on la interacció entre l'usuari i Smartlock és a través d'un dispositiu mòbil amb connectivitat Bluetooth, i amb funcionament local. És a dir, que no es pot controlar el sistema a una distància més enllà de la distància màxima que admet la connectivitat Bluetooth (10 metres aproximadament). En aquest aspecte, amb Smartlock es vol donar la possibilitat a l'usuari d'accedir a l'habitatge sense dependre d'un dispositiu amb bateria, amb la qual si l'usuari es queda sense bateria no podria accedir a l'habitatge encara que tingui la "clau" (Smartphone). També, vol dotar d'una major seguretat addicional utilitzant un sistema de doble autenticació i un registre d'accessos.

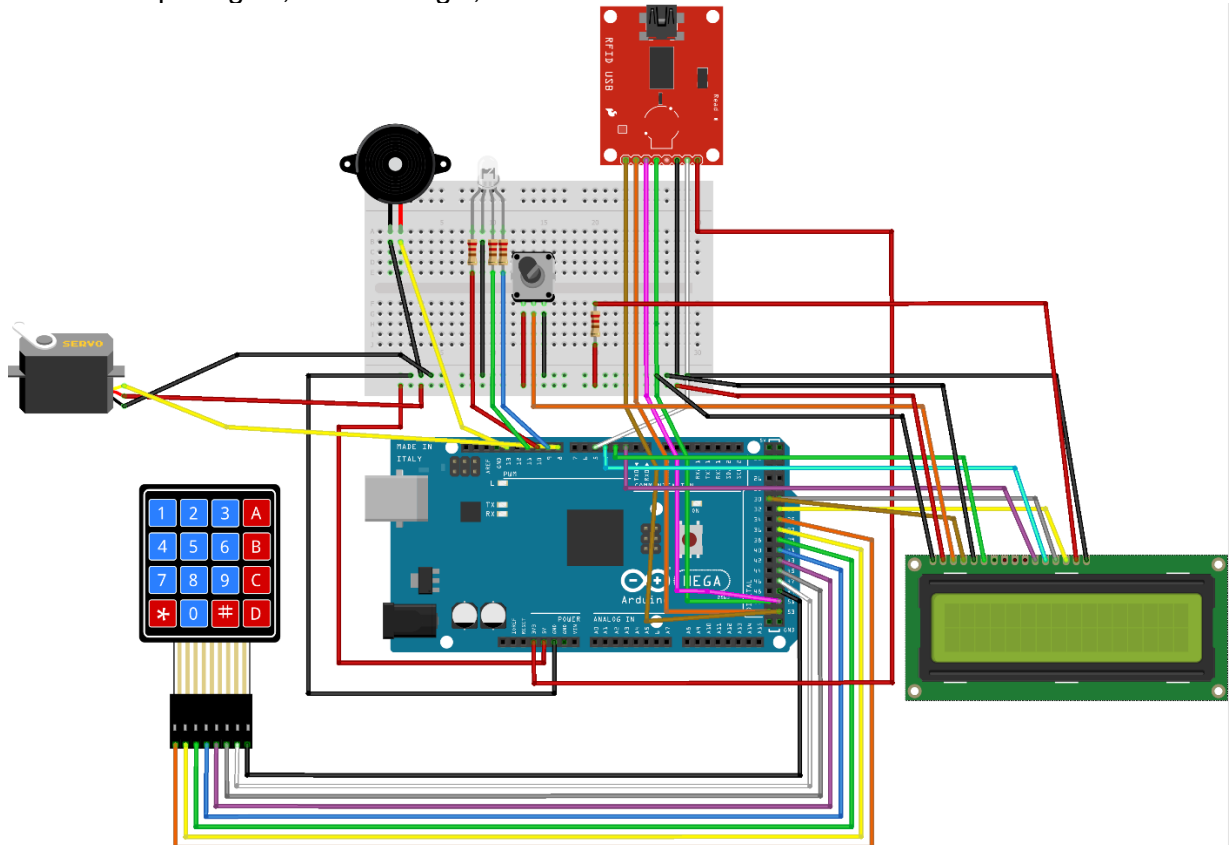
Resumint, si s'analitzen altres solucions similars al problema que es planteja en aquest projecte, es pot veure que es pot donar una solució amb la tecnologia ja existent, de fet aquests projectes que s'han mencionat anteriorment, es van desenvolupar fa uns anys, i actualment han aparegut nous components compatibles amb Arduino. Així com nous formats de plaques Arduino, diferents sistemes de connexions, nous conceptes i idees sobre IOT gràcies a l'expansió de la domòtica a la societat, on com s'ha pogut observar anteriorment, la competència ha intentat donar solució aquest problema atacant des de diferents solucions.

3. Desenvolupament Smartlock

3.1. Sistema Smartlock

El sistema Smartlock constarà de dues parts, d'una banda, la part física, que consistirà en la connexió dels diferents dispositius compatibles amb Arduino i amb la qual l'usuari interactuarà directament. I d'altra banda, el servidor, que serà on es recolzarà Smartlock per fer totes les comprovacions d'autenticació necessàries.

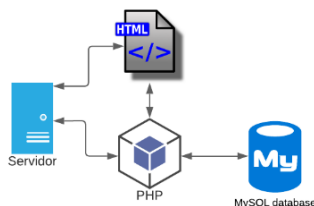
Com es pot observar en el següent esquema de components, s'ha fet ús d'una protoboard per facilitar les connexions dels diferents components, i s'ha tingut en compte (pel seu correcte funcionament) cada connexió que compleixi les seves necessitats, com ara, si el component necessita un pin digital, o un analògic, etc.



Il·lustració 8 Esquema circuit Smartlock

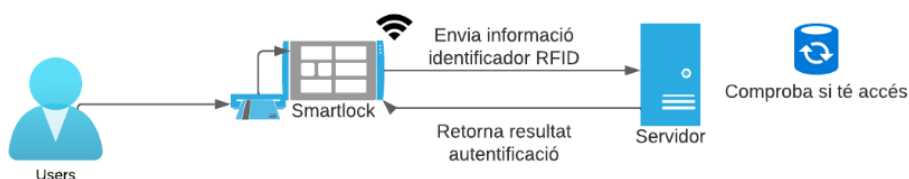
fritzing

Aquesta part física que és Smartlock es recolzarà amb un servidor que constarà d'una part visual HTML que farà de panell d'administrador, i un servidor PHP i MySQL. El servidor PHP s'encarregarà de fer les operacions necessàries amb la base de dades, com són la verificació d'autenticació, guardar noves targetes, usuaris, etc. I el servidor MySQL s'encarregarà de guardar tota la informació necessària del sistema com ara usuaris, log, targetes, etc. En el següent diagrama es pot observar els diferents elements dels quals estarà compost el servidor.



Il·lustració 9 Esquema servidor

Finalment, amb les dues parts desenvolupades, Smartlock ja estarà operatiu. A la il·lustració es pot observar quin seria el seu funcionament principal:



Il·lustració 10 Esquema funcionament Smartlock

3.2. Smartlock - Components

A continuació, en aquest punt es procedirà a analitzar i entrar en detall sobre el desenvolupament del circuit Smartlock.

3.2.1. Arduino

Arduino és una plataforma d'electrònica basada en la filosofia del software lliure, nascuda l'any 2005 com un projecte d'estudiants de l'institut IVREA (Itàlia), en el qual els seus principis eren les de desenvolupar un projecte amb software i hardware fàcils d'utilitzar.

Bàsicament el que ens permet aquesta eina és la generació d'infinitat de tipus de projectes, gràcies a la combinació del seu hardware, el seu software i els components electrònics compatibles amb l'ecosistema Arduino.

En Arduino es poden trobar diferents formats de plaques microcontroladores, entre les més populars que es troben són: Arduino UNO, Arduino DUE, Arduino ATMEGA2560, Arduino Micro, Arduino Nano, Arduino YUN, etc. Cada placa té les seves peculiaritats en l'àmbit de nombre de pins (digitals i analògics), xips, connexions, alimentació, etc.

Seguidament es mostren algunes imatges dels diferents tipus de microcontroladors més típics d'Arduino:



Il·lustració 11 Microcontroladors Arduino

Després d'analitzar el potencial que ofereix aquest sistema, es decideix que serà el sistema a utilitzar per desenvolupar Smartlock, donat que disposa dels components necessaris compatibles així com un software lliure anomenat IDE Arduino per al seu desenvolupament.

D'altra banda, després d'analitzar les necessitats tècniques necessàries per Smartlock es determina que es necessitarà un nombre elevat de connexions (PINS), tant analògics com digitals, a més de necessitar una connexió wifi, ja que aquest tipus de connexió facilitaria molt poder ubicar en qualsevol lloc Smartlock, sense la necessitat de fer instal·lacions de cable Ethernet per l'habitatge. Per aquest motiu, es descarten alguns models de microcontroladors per a desenvolupar Smartlock com ara Arduino UNO, Arduino Leonardo, Arduino Micro, Arduino Zero, etc. per la seva limitació en el nombre de pins, i es decideix agafar com a model Arduino Mega 2560 que està dotat d'una gran quantitat de pins digitals i analògics.

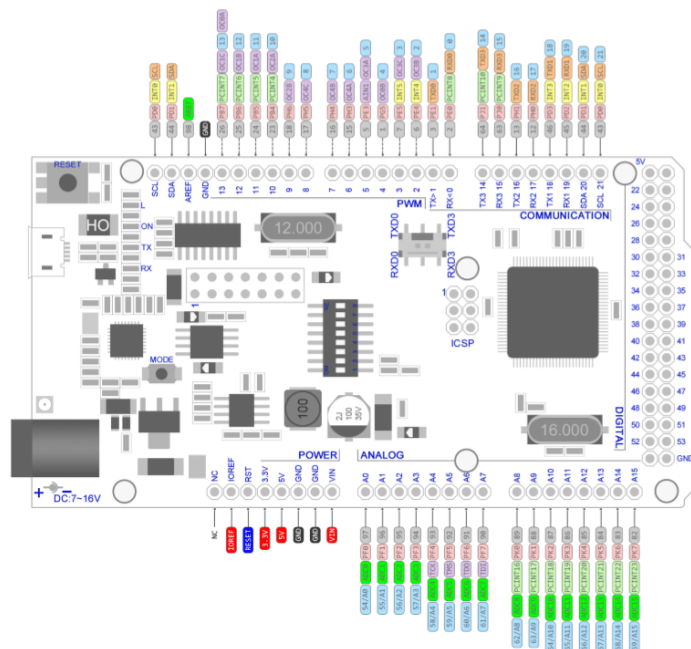
Una vegada definit el microcontrolador necessari pel projecte Smartlock es detecta la necessitat d'afegir algun component que permeti connectar el microcontrolador al servidor, a través d'una connexió Wi-Fi. Per solucionar aquest problema, es troben dues possibles solucions: adquirir un xip Wi-Fi i connectar-lo a la placa o buscar si existeix en el mercat alguna placa que segueixi l'estructura d'Arduino ATmega2560, però amb un xip Wi-Fi integrat, així evitant ocupar pins i reduint l'espai necessari dels components del projecte.

Finalment, es troba que existeix en el mercat una placa que dona solució al problema plantejat, una placa que incorpora Arduino ATmega2560 i un xip8266 per connectar la placa al Wi-Fi. Donat que s'ha d'adquirir un microcontrolador pel projecte es decideix optar per aquesta opció que incorpora tot. No obstant, si es disposés d'una placa Arduino ATmega2560 prèviament, es podria haver optat per adquirir per separat un xip8266 o similar.

3.2.2 Placa Arduino Atmega2560 + ESP8266

Com s'ha mencionat anteriorment dintre de l'ecosistema d'Arduino existeixen diferents formats de plaques, cadascuna amb les seves particularitats, però pel projecte SmartLock s'utilitzarà una placa Arduino modificada, en la qual utilitzarà la base de les plaques Arduino MEGA 2560 amb un xip Wi-Fi integrat ESP8266. Això, permetrà treballar amb Arduino tant a nivell local com amb connexió Wi-Fi sense la necessitat d'afegir un component addicional, ja que el portarà integrat.

A continuació, es pot visualitzar l'esquema de connexions i components d'aquesta placa.



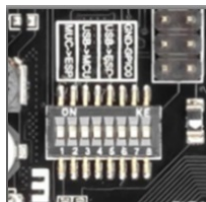
II-lustració 12 Esquema connexions Arduino ATmega2560 + ESP8266

Els motius d'utilitzar aquest model de placa són:

- **Arduino MEGA 2560:** Aquesta placa està dotada de 54 pins digitals entrada/sortida, dels quals 14 es poden utilitzar com a PWM i 16 analògiques. Això permetrà tenir suficients pins entrada/sortida per connectar tots els components necessaris pel funcionament Smartlock,

amb un Arduino UNO es disposa de només 16 entrades digitals i 6 analògiques, les quals són insuficients pel sistema.

- **ESP8266:** Aquest xip integrat permet la connexió Wi-Fi de la placa Arduino, i es compatible amb el protocol TCP/IP. D'aquesta manera, Smartlock podrà tenir comunicació entre el seu microcontrolador i el servidor (BBDD + PHP). A més, entre les característiques més destacables d'aquest xip mencionem el seu baix consum, ideal per projectes que necessiten corrent elèctrica les 24h del dia.



Il·lustració 13
Interruptors control de la placa

Una vegada analitzades les característiques principals d'aquesta placa, pasarem a analitzar una de les seves peculiaritats que la fa diferents a les plaques originals Arduino. El fet de fusionar una placa ATmega 2560 amb el xip 8266, donat que aquesta placa permet configurar el seu funcionament en 6 modes diferents, gràcies a un petit circuit d'interruptors situat a la zona central de la placa.

A la següent taula, es pot visualitzar les 6 combinacions possibles de les que disposa.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | TXD0/TXD3 |
|----------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----------|
| USB Aïllat | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF | TXD0 |
| USB <--> ATmega 2560 | OFF | OFF | ON | ON | OFF | OFF | OFF | OFF | TXD0 |
| USB <--> ESP8266(firmware) | OFF | OFF | OFF | OFF | ON | ON | ON | OFF | TXD0 |
| USB <--> ESP8266(treball) | OFF | OFF | OFF | OFF | ON | ON | OFF | OFF | TXD0 |
| USB <--> ESP8266 | ON | ON | OFF | OFF | OFF | OFF | OFF | OFF | TXD0 |
| USB<--> ATmega 2560 <-->ESP 8266 | ON | ON | ON | ON | OFF | OFF | OFF | OFF | TXD3 |

Un cop hem conegut les diferents combinacions, explicarem en detall cadascuna de les seves opcions:

- **Opció 1:** tots els jumpers en posició OFF, això significa que no hi haurà cap comunicació entre l'USB i la placa
- **Opció 2:** el jumper 3 i 4 en ON, amb això es pot controlar des de l'USB la placa de Arduino Mega, és com si fos un ATmega 2560 normal i corrent sense Wi-Fi.
- **Opció 3:** el jumper 5, 6 i 7 en posició ON, amb això es pot 'accedir a l'ESP8266 des de l'USB però només és per pujar el codi compilat del Arduino IDE.
- **Opció 4:** els jumpers 5 i 6 en ON això significa que hi ha connexió entre l'USB i l'ESP8266, però només a manera de treball, no es pot pujar ni actualitzar el codi que hi ha en la placa ESP8266.
- **Opció 5:** el jumper 1 i 2 en ON, amb aquesta configuració hi ha connexió amb l'USB i la placa ESP8266 com si no existís l'Arduino Mega.
- **Opció 6:** els jumpers 1, 2, 3 i 4 en ON, amb aquesta combinació hi ha una connexió entre l'USB i la placa Mega i al mateix temps també hi ha connexió entre la Mega i l'ESP8266. Aquesta seria la combinació necessària per a poder utilitzar totes les funcionalitats de la placa, és a dir, Arduino Mega amb Wi-Fi (la combinació per Smartlock).



Il·lustració 14
Selector canal de comunicació

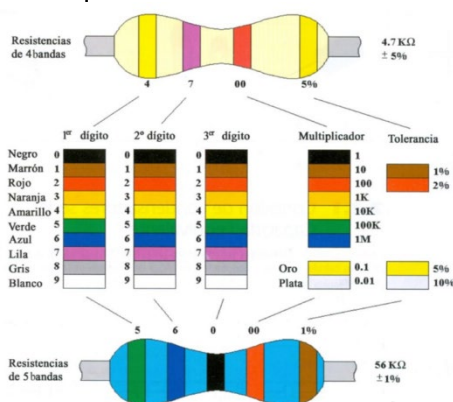
Un altre de les característiques pròpies d'aquest model de placa és que també es pot escollir el canal de comunicació amb l'ajuda d'un interruptor situat prop del panell d'interruptors mencionats anteriorment. Aquest permet escollir entre dues posicions que són el TXD0 o TXD3, per saber quin canal seleccionar s'haurà de seguir les indicacions de la taula anterior. És a dir, segons la modalitat en la qual es trobi la placa s'haurà de col·locar l'interruptor en TXD0 o TXD3 (aquest canal de comunicació només es col·loca a l'opció 6).

Una vegada analitzat totes les combinacions de funcionament de la placa, primerament es situarà a l'opció 2 (Arduino ATmega2560 normal) per fer totes les proves de funcionament bàsic de la placa. Un cop funcioni tot, es passarà a modificar els jumpers a l'opció 6 per utilitzar tot el potencial de la placa, Arduino ATmega2560 + Wi-Fi.

3.2.3 Resistència

Les resistències són un element passiu que té com a particularitat dificultar el pas de corrent, a través seu. Les resistències tenen les següents particularitats:

- La unitat de mesura utilitzada per a les resistències és Ohm.
- Com més gran sigui el valor Ohm, més resistència oferirà al pas de corrent, i per tant també s'escalfarà més.
- Les resistències no tenen polaritat, és a dir, es poden connectar en un sentit o en un altre indistintament.
- Les resistències venen pintades amb una sèrie de marques verticals de diferents colors, aquest és un codi de colors que informa visualment quin és el seu valor de resistència (si l'usuari és daltònic pot fer servir un polímetre). A continuació, es pot observar una taula amb el codi de colors que utilitzen.



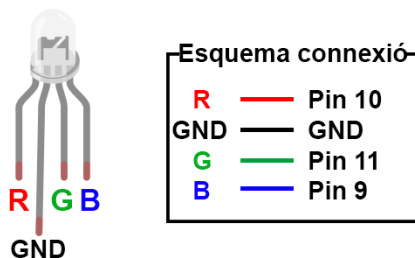
Il·lustració 15 Esquema de colors d'una resistència

En el cas d'Smartlock aquestes resistències s'utilitzaran en dos elements, en el LED RGB, una de 220 Ohms per a cada connector de color (vermell, verd i blau) i una altre pel Boozer.

3.2.4 LED RGB

Els LEDs (Light-emitting Diode) és un component capaç de produir llum quan circula per ell una corrent elèctrica, normalment són fabricats en un color específic on el color i la intensitat de llum que és capaç d'emetre variarà el seu voltatge.

Com es pot visualitzar en el següent esquema, el LED RGB disposa de 4 connexions (4 colors primaris i el més llarg l'ànode o càtode) a diferència dels LEDs tradicionals que tenen 2.



Il·lustració 16 Esquema connexions LED RGB

El seu funcionament és similar als LEDs tradicionals, amb la diferència que en el seu interior disposa de tres LEDs amb els colors primaris (vermell, verd i blau), amb els quals fa ús de la síntesi additiva proposada per James Clek Maxwell, que consisteix a aconseguir tota una gamma de colors gràcies a la il·luminació de diferents intensitats d'aquests LEDs.

Si es té en compte aquest funcionament i seguint l'esquema de connexions del LED RGB a la placa, on s'ha de tenir especial atenció que s'haurà de col·locar resistències de 220 Ohms en les connexions RGB per evitar que es cremin per sobretensió. A més, s'haurà de desenvolupar

el codi necessari pel seu funcionament, on es buscarà il·luminar el LED de color verd si la lectura de la targeta és correcta, si està oberta la porta, etc. I de color vermell, en els casos d'errors.

Per programar el funcionament del LED primerament s'haurà de definir com a pins de sortida el número de pins corresponent a les connexions R,G,B seguint l'esquema anterior, utilitzant la sentència següent on x és el número de pin:

```
pinMode(PIN X, OUTPUT);
```

Una vegada definit els 3 pins de sortida, amb la següent funció es podrà regular la intensitat de llum per a cada pin, és a dir, la intensitat de llum per a cadascun dels 3 colors primaris. Aquesta intensitat es podrà regular des de 0 a 255 amb la finalitat d'aconseguir la gama de colors desitjada.

```
analogWrite (PIN X, 0); // el segon paràmetre pot anar de 0 a 255 (pin, Intensitat)
```

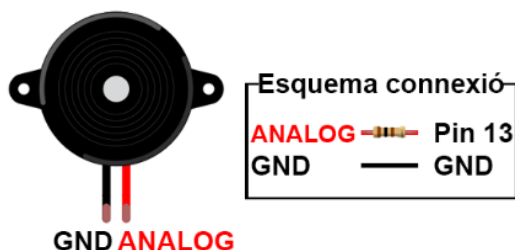
En Smartlock s'utilitzarà principalment dos colors, el color vermell per error i el color verd per la lectura de RFID correcta i en accés autoritzat.

3.2.5 Speaker piezoelèctric

Els buzzers o altaveus passius són dispositius que permeten convertir un senyal elèctric en una ona de so, aquests dispositius no disposen d'una electrònica interna, per aquest motiu s'ha de proporcionar un senyal elèctric per aconseguir el so desitjat. A diferència dels altaveus passius, que aquest emeten so constant en el moment que reben senyal elèctric.

En aquest cas, l'altaveu piezoelèctric que s'anomena així perquè en la seva part interior té un component piezoelèctric que en rebre un senyal elèctric es deforma i produeix so.

L'esquema de connexió és senzill, consta només de dues connexions, una al GND d'Arduino i l'altra a un pin analògic que permet configurar la intensitat del senyal. No obstant, entre Arduino i el pin analògic s'ha de col·locar una resistència, donat que treballa normalment a 16 Ohms. Per aquest motiu, es pot col·locar una resistència de 100 o 200 Ohms, tenint en compte que quan més alta sigui aquesta resistència més baix s'escoltarà el so. A continuació, es pot veure l'esquema de connexió.



Il·lustració 17 Esquema de connexions piezoelèctric

Per utilitzar aquest altaveu en Arduino, primerament s'haurà de definir el Pin d'entrada analògic com a sortida (utilitzant la sentència anterior pinMode), i després, s'utilitzaran les següents funcions per emetre sons:

```
noTone(BuzzerPin); // No emetre cap só  
tone(BuzzerPin, 1000, 200); //Só d'error (pin, freqüència, duració)  
tone(BuzzerPin, 3000, 200); //Só correcte (pin, freqüència, duració)
```

3.2.6 Lector RFID RC522

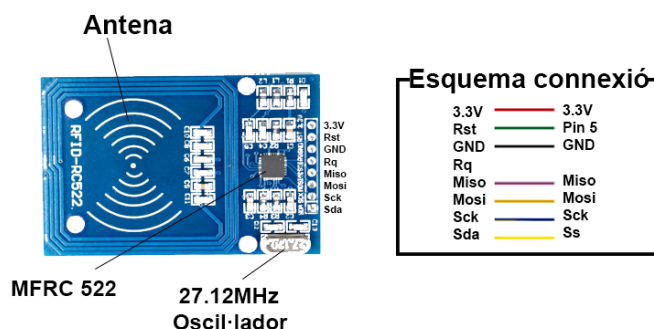
Aquest mòdul RFID-RC522 utilitza un voltatge de 3.3V com alimentació, 64 blocs de memòria (0-63) de 16 Bytes on fa la lectura i/o escriptura, i es controla a través del protocol SPI. Cal destacar que és compatible amb gairebé qualsevol microcontrolador, com ara Arduino.

El RC522 utilitza un sistema avançat de modulació i desmodulació per a tota mena de lectura i escriptura de targetes amb dispositius passius de 13.56Mhz, aquestes targetes disposen d'un número de sèrie que consisteix en 5 valors hexadecimals.

A la següent taula es mostren les característiques principals d'aquest lector:

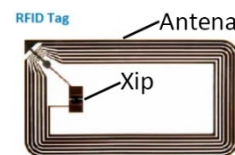
| | |
|-----------------------------|----------------|
| Alimentació | 13-26mA a 3.3V |
| Lsb stand by | 10-13mA a 3.3V |
| Lsm en sleep-mode | <80uA |
| Lm màxim | 30mA |
| Freqüència d'operació | 13.56Mhz |
| Distància de lectura | de 0 a 60mm |
| Protocol de comunicació | SPI |
| Velocitat màxima de dades | 10Mbit/s |
| Dimensions | 40 x 60mm |
| Temperatura de funcionament | -20C a 80C° |
| Humitat de funcionament | 5% a 95% |
| Velocitat màxima SPI | 10Mbit/s |

La següent imatge mostra l'esquema de connexions i parts del lector RFID



II·lustració 18 Esquema de connexions lector RFID RC522

El funcionament d'aquest sistema consisteix en que l'usuari ha d'apropar un RFID Tag (que es pot trobar en diferents formats com ara una targeta, un clauer, etc.). Aquest RFID Tag disposa d'un circuit simple que està compost per dos elements, d'una banda, una antena que envolta la major part de l'identificador, i d'altre el xip que guarda la informació. Aquest RFID Tag, s'ha d'apropar al lector RFID-RC522 a una distància no superior a 60 mm perquè aquest pugui realitzar la lectura/escriptura d'aquest.



II·lustració 19 Esquema RFID Tag

Per fer ús d'aquest component en Smartlock s'utilitzarà una llibreria anomenada "MFRC522.h" la qual ha sigut desenvolupada per Miguel Balboa. Aquesta es carregarà al IDE d'Arduino amb la funció include mencionada anteriorment. Posteriorment, es declararan els Pins del Reset i del SDA seguint l'esquema de connexions.

A continuació, s'ha de crear l'objecte mfr522 per poder utilitzar les funcions de la llibreria:

```
MFR522 mfr522(SDA PIN, RESET PIN);
```

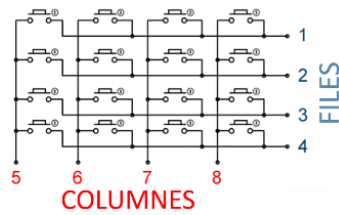
Una vegada creat l'objecte mfr522 s'utilitzaran les següents funcions per operar amb aquest component:

```
SPI.begin(); //Per inicialitzar el bus SPI
mfr522.PCD_Init(); //Per inicialitzar l'objecte mfr522
mfr522.PICC_IsNewCardPresent(); //Per saber si s'ha detectat una targeta
mfr522.PICC_ReadCardSerial(); //Per seleccionar una targeta
mfr522.uid.uidByte[i]; //Per obtenir el valor hexadecimal de cada posició
mfr522.PICC_HaltA(); //Per terminar la lectura del RFID Tag
```

3.2.7 Teclat matriu 4x4

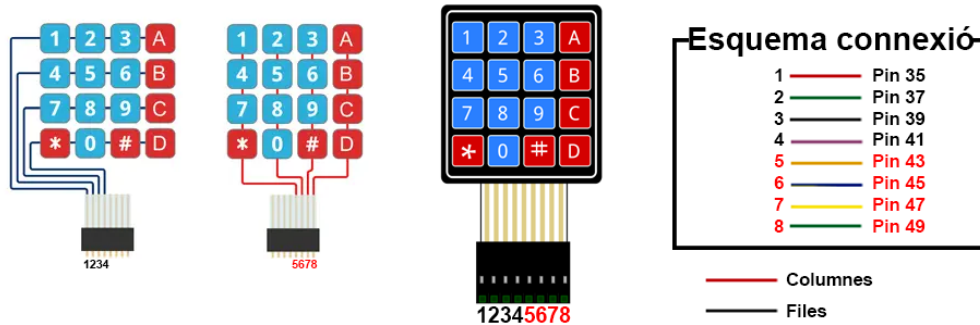
Els teclats matriu són dispositius que agrupen diferents pulsadors i que permeten controlar-los utilitzant un nombre de conductors inferior al que es necessitaria si s'utilitzessin de manera individual. Aquests agrupen els pulsadors en files i columnes formant una matriu de NxM, en aquest cas 4 Files x4 Columnes. Gràcies a aquest sistema matriu, en comptes de necessitar 16 conductors per al teclat de 4x4 només es necessiten 8 conductors (la meitat). No obstant això, aquest sistema no és perfecte, donat que pot donar problemes a l'hora de detectar la

pulsació de diverses tecles alhora, però aquesta limitació, en aquest cas concret, no serà un problema, donat que si es té en compte la mida del teclat, l'usuari només pot fer servir de manera còmoda, un sol dit per fer les pulsacions.



A la següent imatge es pot observar l'esquema intern de connexions del teclat matricial 4x4, així com l'esquema de connexions a la placa Arduino.

II-lustració 20 Connexions pulsadors d'un teclat matricial



II-lustració 21 Esquema de connexions d'un teclat matricial 4x4

Per utilitzar aquest teclat en Smartlock es farà ús de la llibreria Keypad que té en el programari Arduino IDE. Per afegir aquesta llibreria al programa, s'utilitzarà la sentència següent:

```
#include <Keypad.h>
```

Una vegada carregada la llibreria Keypad, s'ha de crear el mapa del teclat. Per fer-ho primerament es definiran els pins que controlen les columnes i les files, seguint l'esquema anterior:

```
byte colPins[4] = {43,45,47,49}; //Els 4 pins de les columnes
byte rowPins[4] = {35,37,39,41}; //Els 4 pins de les files
```

Seguidament, s'ha de definir quins valors té cada tecla:

```
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
```

Una vegada definits els pins corresponents a les columnes i les files, i tenint els valors de cada tecla, es pot crear l'objecte Keypad

```
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS);
//Keypad (MakeKeymap(valors de cada tecla), Pins de les files, Pins de les columnes, nombre de files, nombre de columnes)
```

Finalment, s'utilitzarà la següent funció per a realitzar la lectura del teclat:

```
char key = keypad.getKey(); //La funció retorna el valor de la tecla premuda
```

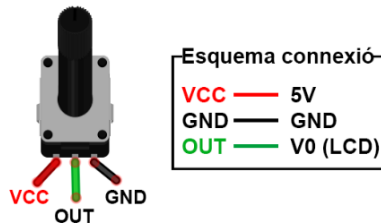
3.2.8 Potenciòmetre

Un potenciòmetre proporciona una resistència variable segons la seva posició. Tenint en compte les següents regles:

- Si està totalment tancat, s'obtindrà el màxim de voltatge que rep per l'entrada.
- Si està totalment obert, s'obtindrà el mínim voltatge, que és 0.
- Si es troba en qualsevol altre posició, s'obtindrà una fracció de voltatge d'entrada, proporcional a la posició on es trobi.

Tenint en compte el funcionament d'un potenciòmetre, mencionat anteriorment, aquests es poden trobar amb diferents resistències (la més típica és la de 10KOhms que és la que s'utilitzarà en Smartlock).

L'esquema de connexions i funcionament d'un potenciòmetre és el següent



Il·lustració 22 Esquema de connexions d'un potenciòmetre

En el cas de Smartlock el potenciòmetre s'utilitzarà per regular la intensitat de contrast de la pantalla LCD, amb la finalitat de millorar la visualització d'aquesta.

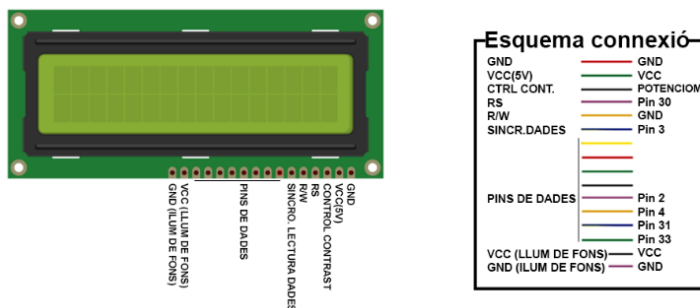
3.2.9 Panell LCD 16x2

El panell de cristall líquid també conegut com LCD (Liquid Crystal Display), utilitza les propietats de la llum polaritzada juntament amb una sèrie de filtres per mostrar informació per pantalla gràcies a la il·luminació de fons.

Hi ha una àmplia gamma de panells LCDs que es poden fer servir per Arduino. En aquest cas, s'utilitzarà una de les més comunes, una LCD 16x2, això vol dir que té dues files de 16 caràcters cadascuna, els quals es podran mostrar de manera simultània. No obstant això, es podrà anar refrescant la pantalla amb nous caràcters però el límit de caràcters que entraran en la pantalla sempre seran 32, 16 a la primera fila i 16 en la segona.

Aquest panell LCD 16x2 consta de 16 connexions per funcionar, en la següent taula es pot visualitzar per a què serveix cada connexió:

| Pin | Funció |
|------|---|
| 1 | GND (Terra) |
| 2 | 5V (Alimentació) |
| 3 | Control de contrast de la pantalla |
| 4 | RS - Selector entre comandos i dades |
| 5 | RW - Escriitura i lectura de comandos i dades |
| 6 | Sincronització de lectura de dades |
| 7-14 | Pins de dades de 8 bits |
| 15 | Alimentació de la llum de fons (5V) |
| 16 | GND (Terra) de la llum de fons |



Il·lustració 23 Esquema de connexions d'un panell LCD

Per implementar el panell LCD en Smartlock, s'utilitzarà la llibreria LiquidCrystal, la qual ja ve inclosa en el IDE d'Arduino, per fer-ho s'utilitzarà la sentència #include com s'ha fet en els casos anteriors.

Una vegada carregada la llibreria corresponent, s'haurà de crear l'objecte LCD 16x2 per que es pugui utilitzar en Smartlock, això es farà amb la següent funció:

```
LiquidCrystal lcd(30,3,2,4,31,33); // lcd (Rs, Sincr.Dades, Pin dades,Pin dades,Pin dades,Pin dades)
Lcd.begin(16,2); //Inicialitzar la pantalla amb el format 16 caràcters i 2 files.
```

Finalment, ara que ja s'ha creat i declarat el format de la pantalla LCD, s'utilitzaran les següents funcions per operar amb la mateixa.

```
Lcd.setCursor(columna, fila); //Situara el cursor en la posición definida
Lcd.print("text per pantalla"); //Introduirà el text desde la posición del cursor.
Lcd.clear(); //Esborrarà tot el contingut de la pantalla.
```

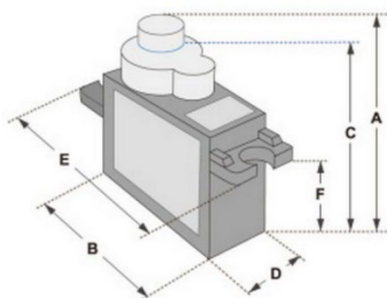
3.2.10 Servomotor

Un servomotor com el seu nom diu és un motor elèctric, però que té dues característiques especials que el diferencien d'un motor elèctric estàndard, que són les següents:

- Permet mantenir una posició determinada que es trobi dintre del rang d'operació del mateix dispositiu.
- Permet controlar la velocitat de gir com ara esperar un temps abans que es mogui a la següent posició.

En Smartlock s'utilitzarà un servomotor Micro Servo 9g SG90 de Tower Pro, que permetrà simular el funcionament d'un pany elèctric, o d'un possible futur accessori que consisteixi a utilitzar el pany de la porta original, col·locar una clau i fer-la girar amb un motor fins a la posició desitjada.

En aquest cas, s'analitzarà el funcionament d'un servomotor de Tower Pro, model Micro Servo 9g SG90. Aquest component permet realitzar un escombratge dels -90° als 90° , és a dir, un gir de 180° . A les següents imatges, es mostren les seves característiques principals:

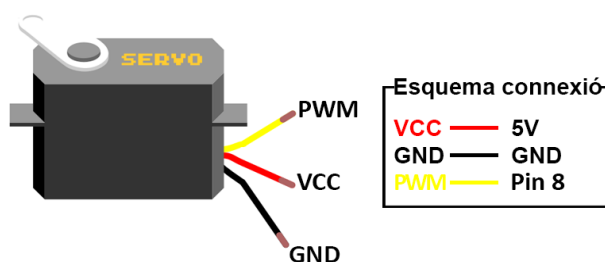


Il·lustració 24 Esquema de mides d'un Servomotor SG90

| Dimensions i especificacions | |
|------------------------------|-------|
| A (mm) | 32 |
| B (mm) | 23 |
| C (mm) | 28.5 |
| D (mm) | 12 |
| E (mm) | 32 |
| F (mm) | 19.5 |
| Velocitat (seg) | 0.1 |
| Esforç de torsió (kg-cm) | 2.5 |
| Pes (g) | 14.7 |
| Voltatge | 4.8-6 |

Aquest motor funciona amb un senyal PWM, amb un puls de treball entre els 1ms i els 2ms en un període de 20ms (50Hz). Això, indica la velocitat màxima que es pot moure aquest motor en Arduino, és a dir, que es pot canviar la posició cada 20ms. Encara que el servomotor es pot moure amb una resolució de més 1 grau, aquest és el màxim de resolució que es pot aconseguir degut a la limitació de la senyal PWM que és capaç de generar Arduino.

Tot seguit, es pot observar l'esquema de connexió del Servomotor



Il·lustració 25 Esquema de connexions servomotor SG90

Per utilitzar aquest servomotor en Smartlock, es farà ús d'una llibreria anomenada Servo, la qual ja es troba incorporada en l'IDE Arduino, per fer-ho s'utilitzarà la sentència #include.

Una vegada incorporada la llibreria es definirà el Pin del servo seguin l'esquema de connexions anterior, i es declararà l'objecte Servo i se li assignarà el pin PWD de treball.

```
Servo myservo; //Declaració de l'objecte Servo
Myservo.attach(Pin Servo PWD) // Inicializar el servo assignant el pin corresponent segons esquema.
```

Finalment, una vegada creat l'objecte Servo, s'utilitzarà la següent funció per moure el servo motor:

```
Myservo.write(Angle); //En Angle s'introduirà un valor entre -90 i 90 graus
```

3.2.11 Wi-Fi ESP8266

Una de les característiques principals d'aquest model de placa Arduino és que porta integrat un xip ESP8266.

Per carregar el Firmware en el xip ESP8266, el primer que s'ha de fer és modificar el panell d'interruptors de mode a l'opció 3 (també el canal de comunicació a la posició TXD0).

| | 11 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | TXD0/TXD3 |
|-------------------|-----|-----|-----|-----|----|----|----|-----|-----------|
| USB | <-> | OFF | OFF | OFF | ON | ON | ON | OFF | TXD0 |
| ESP8266(firmware) | | | | | | | | | |

Ara, la placa ja està a l'opció 3, que permet carregar el Firmware d'ESP8266 i ja es pot connectar a l'IDE Arduino, però s'haurà de modificar el tipus de placa que actualment és la d'Arduino ATmega2560, per una "Generic ESP8266 Module".

Per a desenvolupar el codi necessari per fer la connexió, es necessitarà carregar les llibreries ESP8266HTTPClient.h i ESP8266WiFiMulti.

A continuació, es mostra el codi bàsic utilitzat per al Firmware del xip ESP8266 per connectar-se a la xarxa wifi:

```
//Es carreguen les llibreries necessàries
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFiMulti.h>

//Definició dels credencials de la xarxa Wifi
const char* ssid = "G55_Wifi";
const char* password = "*****";

//Funció que s'executa al carregar el firmware
void setup() {
  Serial.begin(115200); //S'inicialitza el port Serial
  WiFi.begin(ssid, password); //Es realitza la connexió amb la xarxa wifi local
  Serial.println("Connectant..."); //Es mostra un missatge pel port serial
  //Mentre no es realitzi la connexió s'aniran mostrant punts (carregant)
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  //Si s'ha realitzat la connexió correctament es mostrarà un missatge informatiu amb la IP.
  Serial.println("Connexió OK!");
  Serial.print("IP Local: ");
  Serial.println(WiFi.localIP());
}
void loop() {}
```

Una vegada s'ha carregat correctament el Firmware al xip ESP8266, s'haurà de desconnectar l'alimentació de la placa Arduino i modificar el panell de pins de mode a l'opció 4, perquè ESP8266 es posi a treballar.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | TXD0/TXD3 |
|------------------|-----|-----|-----|-----|----|----|-----|-----|-----------|
| USB<-> | OFF | OFF | OFF | OFF | ON | ON | OFF | OFF | TXD0 |
| ESP8266(treball) | | | | | | | | | |

Ara que la placa ja es troba en el mode 4, es connectarà de nou l'alimentació i es visualitzarà si en el port serial de l'IDE Arduino rep alguna resposta.

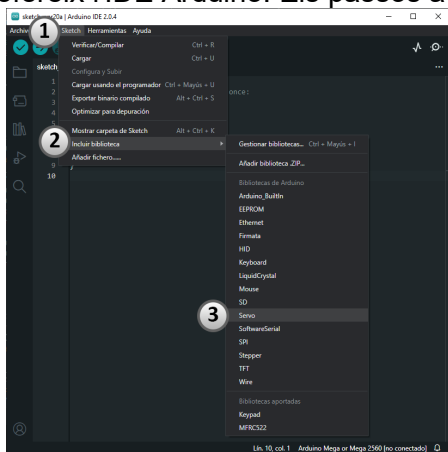
```
Connectant.....Connexió OK!  
IP Local: 192.168.2.108
```

Finalment, ara que ja està en funcionament Smartlock Arduino i el xip ESP8266 es pot habilitar el mode 6 perquè funcioni tot alhora. És important verificar que el port de comunicació, en aquest cas, sigui el port TXD3. També, cal destacar que cada cop que es faci una modificació en el panell d'interruptors de modalitat de funcionament, s'ha de desconnectar la placa de l'alimentació, en cas contrari, a part de poder danyar la placa, els canvis no es detecten i no es realitzarà el canvi de modalitat.

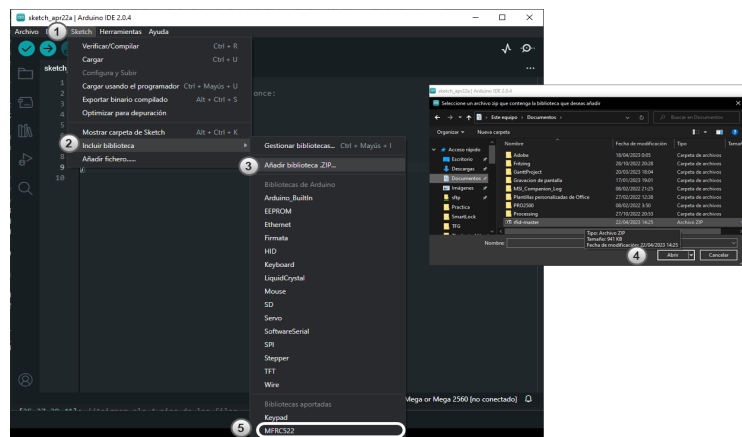
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | TXD0/TXD3 |
|-------------|----|----|----|----|-----|-----|-----|-----|-----------|
| USB<-> | ON | ON | ON | ON | OFF | OFF | OFF | OFF | TXD3 |
| ATmega 2560 | | | | | | | | | |
| <->ESP 8266 | | | | | | | | | |

3.3. Lliberies

Anteriorment s'ha mencionat que una de les formes per afegir les lliberies dels diferents components és utilitzant la sentència #include. Però, també es pot fer utilitzant l'entorn gràfic que ofereix l'IDE Arduino. Els passos a seguir són:



Il·lustració 26 Passos a seguir carregar llibreria



Il·lustració 27 Passos a seguir carregar llibreria ZIP

Finalment, en els casos que es necessiti una llibreria externa com, per exemple, per fer ús del lector RFID RC522, es podrà baixar la llibreria des de la web de l'autor. Aquesta vindrà en un arxiu comprimit .ZIP, i per integrar-la en el IDE Arduino s'hauran de seguir els passos que es mostren a la il·lustració 27.

3.4 Desenvolupament codi Arduino.Smartlock

A l'inici del codi d'Smartlock hi han definits els diferents pins de connexió dels components a la placa Arduino, assignant aquests valors a diferents constants, d'aquesta manera si es modifiquen les connexions dels components a la placa, només es modificarien aquestes constants, no tot el codi:

```
const int redLEDPin = 10; //S'assigna el pin del LED Vermell del LED RGB  
const int greenLEDPin = 9; //S'assigna el pin del LED Verd del LED RGB  
const int blueLEDPin = 11; //S'assigna el pin del LED Blau del LED RGB  
const int BuzzerPin = 13; //S'assigna el pin del Buzzer  
//etc...
```

També hi han variables globals definides a l'inici del codi, aquestes variables serveixen principalment per dues coses, d'una banda, utilitzar una mateixa variable en diferents funcions, com ara el valor de la targeta llegida, el codi d'accés, etc. I, d'altra banda, per als diferents modes de funcionament d'Smartlock, que són, el mode administrador, la doble autenticació, mostrar el valor d'una targeta o el mode offline (sense connexió amb el servidor web):

```

bool mode_admin = false; //Mode administrador
bool mode_control_acces = false; //Mode Doble autenticació
bool mode_nou_pass = false; //Mode mostrar valor targeta
bool mode_offline = false; //Mode sense connexió

```

Una vegada definida tota la configuració inicial, el sistema arrencarà, i el primer que fa sempre Arduino per defecte, és executar la funció `setup()`. En aquesta funció Smartlock iniciarà els ports Serial 1, que l'utilitzarà per mostrar missatges per pantalla a l'ordinador, i el Serial 3, que l'utilitzarà per comunicar-se amb el xip ESP8266.

```

Serial.begin(9600); //S'inicialitza el port Serial d'Arduino
Serial3.begin(9600); //S'inicialitza el port serial 3 (és el port que permet comunicar-se amb el xip ESP8266)

```

A més, es crearan els objectes necessaris per al funcionament dels components connectats, com són el lector RFID, el servomotor (que simula un pany elèctric) i la pantalla. També hi han definits el tipus de pin que utilitzen alguns components, com ara el LED i l'altaveu, que en aquest cas, són de sortida (Output):

```

mfr522.PCD_Init(); //S'inicialitza el RFID RC522
myservo.attach(ServoPin); //S'inicialitza el servoMotor, assignant el pin PWD
lcd.begin(16,2); //S'inicialitza la pantalla LCD amb 16 caràcters en horitzontal i dos en vertical.
pinMode(redLEDPin, OUTPUT);
//etc...

```

Quan Arduino ja ha arrencat del tot, és a dir, que ha carregat les constants, les variables globals i ha executat el `setup`, automàticament executarà la funció `loop()` indefinidament, això vol dir, que aquesta funció s'executarà cada cicle mentre Arduino estigui connectat. Aquesta funció, que és la principal d'Smartlock, farà principalment dues accions, d'una banda, en el moment que detecti una lectura d'un identificador RFID executarà la funció `read_rfid()`, que és l'encarregada de guardar el valor de la targeta i verificar l'accés, i d'altra banda, en funció de la modalitat en la qual es trobi el sistema, executarà una funció o un altre, com ara mostrar el menú d'administrador, demanar la contrasenya de la doble autenticació, etc. Per verificar en quina modalitat es troba, el sistema utilitzarà els condicionals `if`. A continuació es pot veure el codi resultant:

```

void loop() {
  read_rfid(); //Es crida a la funció encarregada de fer la lectura de la targeta RFID
  if (mode_admin == false && mode_nou_pass == false && mode_control_acces == false){
    control_admin_acces(key_admin_access); }
  if (mode_admin == true){ //Si es troba en mode admin
    menu_admin(); } //Es crida a la funció encarregada de mostrar el menu_admin
  if (mode_control_acces == true){ //Si el sistema es troba en el mode de comprovació doble autenticació
    if (intents < 3){ //L'usuari té 3 intents per col·locar bé la contrasenya, si no el sistema haurà de tornar a llegir la targeta
      control_code(code_user); //Es crida a la funció encarregada de verificar si la clau d'accés introduïda per l'usuari
    }else {ini(); mode_control_acces == false; } } //Si l'usuari a esgotat els 3 intents

```

A part de la funció `loop`, un altre que sempre s'executarà en segon pla, serà la de `serialEvent3`, aquesta funció, s'executarà sempre que es detecti una activitat en el port Serial 3, que és el port de comunicació entre Arduino Mega i el xip ESP8266. Aquesta executarà diferents accions depenent del missatge rebut per l'ESP8266, que poden ser 5 missatges diferents;

- **[ON]**: vol dir que hi ha un accés autoritzat i s'ha de permetre l'accés a l'habitatge.
- **[OFF]** informa a l'Smartlock que es denega l'accés.
- **[OFFLINE]** significa que no hi ha accés al servidor, i per tant, ha de fer les comprovacions d'autenticació en l'àmbit local.
- **[ONLINE]** vol dir que ha retornat la connexió amb el servidor i ha de fer les comprovacions d'autenticació amb el servidor.
- **[****]** (on cada * és un dígit), en aquest missatge es trobarà el valor del pin de doble autenticació que consta de 4 dígits numèrics.

Com es pot observar, tots els missatges es troben entre claudàtors, això es fa perquè el port serial només pot enviar missatges (strings) i no pot diferenciar si aquest, és el valor d'una variable, una

ordre, un objecte, etc. Per aquest motiu, es poden filtrar missatges no desitjats, a més, de no poder filtrar-los correctament, per evitar això, la funció serialEvent3 començarà a operar sempre que es detecti una activitat en aquest port, i només gestionarà el missatge que estigui entre els claudàtors. En el següent codi es pot veure com es gestiona aquesta comunicació.

```
void serialEvent3() {
  while (Serial3.available()) { //Es quedarà dintre del bucle mentre el port Serial3 estigui actiu
    char inChar = Serial3.read();//Lectura de dades del Serial3
    Serial.write(inChar);// S'escriu les dades rebudes pel port serial (inChar) en el Serial
    inString += inChar; //Es guarden tots els valor rebuts pel Serial 3 un a un
    // Es busca si hi ha el text [ON], [OFF], [OFFLINE], [ONLINE] o un text entre claudàtors [] (contrasenya TwoFactor)
    if (inChar == ']') { //Si troba un ] vol dir que ha acabat el missatge
      if (inString.indexOf("[ON]")>0) { //Si el valor rebut pel serial és [ON] entra en el if
        acces_correct(); //Es crida a la funció encarregada d'obrir la porta
      }else if (inString.indexOf("[OFF]")>0) { //Si el valor rebut pel serial és [OFF] entra en el if
        acces_denied(); } //Es crida a la funció encarregada d'informar a l'usuari que no té accés
      //etc...
```

Ara Smartlock té definides dues funcions que s'executaran sempre, però aquestes, necessitaran fer una sèrie d'accions, en funció de la interacció rebuda per l'usuari i/o el xip ESP8266. Per aquest motiu, hi han definides tota una sèrie de funcions addicionals que donaran solució a les diferents interaccions.

La funció ini() s'encarregarà de posar tots els components connectats a l'Arduino en l'estat d'espera, és a dir, amb el LED apagat, l'altaveu no emetrà cap so i la pantalla mostrarà un missatge de benvinguda, que en aquest cas, variarà en funció de si Smartlock de la modalitat de connexió, si té accés al servidor, només mostrarà el missatge Hola!, però si no té connexió, mostrarà el missatge Hola! -OFFLINE, per avisar a l'usuari que s'ha perdut la connexió amb el servidor, això vol dir, que només els identificadors guardats en el sistema (en l'àmbit local) podran accedir a l'habitatge.

```
void ini(){
  analogWrite (redLEDPin, 0); //Es posa un valor de sortida 0 (apagat) al pin del LED vermell
  analogWrite (greenLEDPin, 0); //Es posa un valor de sortida 0 (apagat) al pin del LED verd
  analogWrite (blueLEDPin, 0); //Es posa un valor de sortida 0 (apagat) al pin del LED blau
  noTone(BuzzerPin); //S'apaga el buzzer perquè no emeti cap so
  myservo.write(Servo_Close); //Servomotor en posició tancat, la porta per defecte estarà tancada
  lcd.clear(); //S'esborra tot el contingut de text de la pantalla
  if (mode_offline == false){ //Si el sistema està en mode online entra en el if
    lcd.print("Hola!"); //S'escriu un text per la pantalla
  }else { //Si el sistema està en el mode offline entra en el if
    lcd.print("Hola! - OFFLINE"); } //S'escriu un text per la pantalla
  intents = 0; } //Sinicialitza la variable intents a zero
```

Les funcions acces_correct(), acces_denied() i accés_denied_intent(), faran accions molt similars, i que consistiran en:

- **Acces_correct:** Encendre el LED RGB de color verd, emetre un so greu, mostrar un missatge per pantalla i moure el servomotor en posició obert.
- **Acces_denied i accés_denied_intent:** Encendre el LED RGB de color vermell, emetre un so agut, mostrar un missatge per pantalla i posar el servomotor en posició tancat.

A continuació, es pot visualitzar el codi d'una d'aquestes funcions, les altres dues només canviaran alguns dels valors dels components, però la seva estructura serà idèntica;

```
void acces_correct(){
  analogWrite (redLEDPin, 0); //Es posa un valor de sortida 0 (apagat) al pin del LED vermell
  analogWrite (greenLEDPin, 255); //Es posa un valor de sortida 255 (encès) al pin del LED verd
  analogWrite (blueLEDPin, 0); //Es posa un valor de sortida 0 (apagat) al pin del LED Blau
  delay(100); //Es para Arduino durant 100ms
  tone(BuzzerPin, 2000, 1000); //S'emet un altre so més greu per donar a entendre que s'accepta l'accés
```

```
myservo.write(Servo_Open); //Es col·loca el servoMotor en posició oberta (porta oberta)
lcd.setCursor(0,1); //Es situa el cursor de la pantalla en la segona fila de la pantalla
lcd.print("Acces Concedit"); //Es mostra el text d'accés concedit per la pantalla
delay(3000); //Es para Arduino durant 3000ms
noTone(BuzzerPin); //S'apaga el so del so del buzzer
lcd.clear(); //S'esborra tot el contingut de text de la pantalla
if (mode_offline == false){ //Si el sistema està en mode online entra en el if
  lcd.print("Hola!"); //S'escriu un text per la pantalla
}else { //Si el sistema està en el mode offline entra en el if
  lcd.print("Hola! - OFFLINE"); } //S'escriu un text per la pantalla
myservo.write(Servo_Close); //Es col·loca el servoMotor en posició tancat (tanca la porta)
analogWrite (greenLEDPin, 0); //S'apaga el LED verd (valor sortida 0).
mode_control_acces = false; } //S'assigna el valor de mode_acontrol_acces en false
```

La funció read_rfid, s'executarà en el moment que detecti una lectura d'un RFID, aquesta, emetrà un so perquè l'usuari entengui que s'ha llegit d'identificador correctament i procedirà a emmagatzemar el seu valor en dos variables, una de tipus array (comprovació local) i d'altre un string (comprovació online). Finalment, en funció de la modalitat en la qual es trobi el sistema, executarà unes accions o unes altres, concretament poden haver-hi tres modalitats:

- **Amb connexió:** si hi ha connexió amb el servidor, enviarà pel port Serial 3 (Serial3.println) el valor de l'identificador (string), de manera que el xip ESP8266 disposarà d'aquest valor, per fer que el servidor pugui comprovar l'autenticació de l'identificador i retornar un resultat ([ON], [OFF] o [pass]).
- **Sense connexió:** si no hi ha connexió amb el servidor, executarà la funció accés_control, encarregada de verificar localment si la targeta està guardada en el sistema (en aquest cas en variables, però a futur es pot ampliar guardant els diferents identificadors en una targeta SD), si tot és correcte, demanarà una clau d'accés, en cas contrari denegarà l'accés (en l'àmbit local sempre es requereix una doble autenticació).
- **Mode administrador local:** aquesta modalitat permet a l'usuari veure el valor d'un identificador RFID per la pantalla, d'aquesta manera, l'usuari podrà donar d'alta aquest identificador en la base de dades (new_pass).

A continuació es pot veure el codi resultant:

```
void read_rfid(){
  if ( mfr522.PICC_IsNewCardPresent()){ //Si el lector RFID llegeix una targeta retorna un true
    tone(BuzzerPin, 3000, 200); //Al llegir una targeta s'emet un so confirmant la lectura correcte
    if ( mfr522.PICC_ReadCardSerial()) { //Es selecciona una targeta
      targeta = "[]"; //S'assigna a la variable targeta el valor [ a l'inici
      for (byte i = 0; i < mfr522.uid.size; i++) { //Es recorren tots els bytes de la targeta
        TarjetaActual[i]=mfr522.uid.uidByte[i]; //Es guarda el valor de la targeta en la variable TarjetaActual
        //Es guarden els valors de la targeta llegida en un String
        targeta.concat(String(mfr522.uid.uidByte[i] < 0x10 ? " 0" : " "));
        targeta.concat(String(mfr522.uid.uidByte[i], HEX));}
      targeta.concat(String("]")); //Al final de l'string s'afegeix un ]
      mfr522.PICC_HaltA(); // Es tanca la lectura de la targeta
      if (mode_admin == false && mode_nou_pass == false && mode_offline == false){ //Mode ONLINE
        Serial3.println(targeta); } //S'escriu pel port Serial 3 el valor de targeta (xip ESP8266)
      if (mode_admin == false && mode_nou_pass == true){ //Mode Administrador (lectura targeta)
        new_pass (targeta); } //Crida a la funció new_pass encarregada de mostrar el valor de la targeta per pantalla
      if (mode_admin == false && mode_nou_pass == false && mode_offline == true){ //Mode OFFLINE
        acces_control(); } } } //crida a la funció encarregada de verificar l'accés de la targeta a nivell local
```

La funció compareArray rebrà per paràmetres dos arrays (identificadors RFID), i el que farà, és comparar cada valor de l'array, de manera que compara un a un els valors dels dos identificadors, és a dir, que es compara l'identificador que ha llegit Smartlock, amb els identificadors guardats en el sistema, si els dos són iguals, vol dir que aquest usuari està registrat, i pot accedir a l'habitatge.

```
boolean compareArray(byte array1[], byte array2[]){
```

```

if(array1[0] != array2[0])return(false);
if(array1[1] != array2[1])return(false);
if(array1[2] != array2[2])return(false);
if(array1[3] != array2[3])return(false);
return(true); }// Si tots els bytes son iguals retornarà un true

```

La funció access control el que farà és cridar a compareArray per comprovar si l'identificador detectat existeix en el sistema, si és així, posarà Smartlock en la modalitat de doble autenticació (mode_control_acces=true) i assignarà per defecte el password 0000, si per al contrari, no existeix en el sistema, denegarà l'accés. En el mode offline l'usuari sempre haurà de posar la doble autenticació.

```

void access_control(){
if(compareArray(TarjetaActual,Usuari)){//Es comprova si l'identificador existeix en el sistema
lcd.clear();//S'esborra tot el contingut de la pantalla
lcd.print("Codi acces?");//Es mostra un text per pantalla demanant la clau d'accés
mode_control_acces =true;//Es posa el sistema en mode_control_acces, que és l'encarregat de comprovar si la clau
d'accés introduïda és correcte
code_user = "0000";//S'assigna el valor per defecte de la contrasenya de doble autenticació a 0000
}else{//Si la targeta no està registrada al sistema entra en else
acces_denied(); } }//Es crida a la funció acces_denied

```

La funció control code s'executarà sempre que s'esdevinguin dues situacions; d'una banda, que l'identificador existeixi en el sistema i l'usuari tingui activa la doble autenticació, i d'altra banda, que detecti un identificador vàlid i estigui en el mode offline.

En els dos casos, el que farà, es esperar fins que detecti que s'ha premut una tecla del teclat matricial 4x4, si és així, el guardarà en la variable clau[4], i mostrarà un * per pantalla, i així fins a arribar a un total de 4 tecles. En aquest moment, comprovarà si la clau introduïda per l'usuari és igual a la variable password, si és així, donarà accés a l'habitatge i enviarà un avís al xip ESP8266 a través del serial3, amb el missatge [OK2FACTOR], d'aquesta manera es notificarà en l'historial del servidor, "la clau introduïda és correcte", si per al contrari, l'usuari ha esgotat les tres oportunitats que té, per introduir la contrasenya correcta, es denegarà l'accés a l'habitatge i s'informarà el servidor, enviant el missatge [FAIL2FACTOR] pel port serial 3.

```

void control_code(String password){
char key = keypad.getKey();//Variable on es guarda la tecla premuda
char clau[4]; //Variable on es guarden els 4 caràcters seguits que ha premut l'usuari
if (key != NO_KEY){ //Si es detecta que l'usuari ha premut alguna tecla, entra en el if
clau[comptador]=key; // Es guarda caràcter a caràcter cada cop que es toca una tecla en la variable clau[]
lcd.setCursor(comptador,1);//Se situa el cursor de la pantalla en la segona fila i en la posició igual al comptador
lcd.print("*");//S'escriu per la pantalla un * cada cop que l'usuari prem una tecla
comptador=comptador+1; //S'augmenta la variable comptador per guardar cada caràcter en una nova posició
if(comptador==4) { //Si l'usuari ja ha premut 4 tecles seguides es verificarà si la clau es correcte
String claut_str = String (clau); //Es converteix a string i es guarda el valor de clau en la variable claut_str
claut_str = claut_str.substring(0, 4); //S'assegura que en la variable clau_str només tindrà els 4 primers caràcters
if(claut_str.compareTo(password) == 0) { //Es comparen els dos passwords
acces_correct();//Es crida a la funció acces_correct
comptador=0; // Es reseteja la variable a 0 per poder tornar a guardar 4 tecles
intents = 0;
Serial3.println("[OK2FACTOR]");//S'informa al servidor que l'usuari a introduït la contrasenya correctament
} else {acces_denied_intent();intents++; } //Es denega l'accés al sistema si la clau no és correcte
comptador=0; } } //Es reseteja la variable a 0 per poder tornar a guardar 4 tecles
if (intents >=3){ //Si l'usuari ha realitzat més de tres intents erronis s'informarà a l'historial del servidor
Serial3.println("[FAIL2FACTOR]"); } } //S'informa a l'ESP8266 que l'usuari ha introduït malament la contrasenya

```

La funció control admin acces, és similar a l'anterior, però en aquest cas, aquesta funció s'executarà en bucle, similar a la funció loop, però amb la diferència que aquesta, deixarà d'executar-se, en el moment que detecti la lectura d'un RFID. D'aquesta forma, mentre Smartlock

està en repòs, si l'usuari introdueix correctament els 6 dígit de la contrasenya d'administrador (key_admin_access), podrà accedir a aquest mode, en el qual, amb l'ús de les funcions menu_admin i new_pass, l'usuari tindrà la possibilitat de veure el valor d'un identificador. Aquesta és una modalitat oculta que té Smartlock, i que permet a l'usuari accedir al mode d'administrador local, amb la que actualment, té implementada l'opció de lectura RFID. A continuació es pot veure quin seria el procés.

| 1-Pantalla en espera per introduir els 6 dígit | 2-Menú administrador amb dues opcions | 3- Resultat |
|--|---------------------------------------|------------------------------|
| Hola! | Llegir RFID? 1 =>Si 2=>No | Valor RFID [13 20 18 ac] |

Finalment el codi resultant és el següent:

```
void control_admin_acces(char password[]){
  char key = keypad.getKey(); //Variable on es guarda la tecla premuda
  char clau[6]; //Variable on es guarden els 6 caràcters seguits que s'han premut per l'usuari
  if (key != NO_KEY){ //Si es detecta que l'usuari ha premut alguna tecla entra en el if
    clau[comptador]=key; // Es guarda caràcter a caràcter cada cop que es toca una tecla en la variable clau[]
    comptador=comptador+1; //S'augmenta la variable comptador per guardar cada caràcter en una nova posició
    if(comptador==6) { //Si l'usuari ja ha premut 6 tecles seguides es verificarà si la clau és correcte
      if(clau[0]==password[0]&&clau[1]==password[1]&&clau[2]==password[2]&&clau[3]==password[3]&&clau[4]==password[4]&&clau[5]==password[5]){//Es verifica caràcter a caràcter si les dues claus son correctes
        lcd.clear();//S'esborra tot el contingut de text de la pantalla
        lcd.print("Llegir RFID?");//S'escriu un text per pantalla
        lcd.setCursor(0,1); //Es situa el cursor de la pantalla en la segona fila
        lcd.print("1 =>Si 2=>No");//Es mostra per pantalla les dues opcions disponibles de l'usuari.
        comptador=0;//Es reseteja la variable a 0 per poder tornar a guardar 6 tecles
        mode_admin=true;//Es posa la variable mode_admin en True, per fer que el sistema entri en el mode d'administrador
        tone(BuzzerPin, 1000, 200); //S'emetrà un altre só greu per donar a entendre que el password és correcte
        delay(1000); //S'atura Arduino durant 1000ms
        noTone(BuzzerPin); //S'apaga el so de l'altaveu
        menu_admin(); //Es crida a la funció menu_admin encarregada de mostrar el panell de lectura de targeta
        comptador=0; } } //Es reseteja la variable a 0 per poder tornar a guardar 4 tecles
```

La funció menu_admin, que s'utilitza en control_admin_acces, és l'encarregada de mostrar el menú d'administrador per pantalla, i de realitzar unes accions o unes altres, en funció de la tecla premuda per l'usuari, que són la tecla 1 i 2. En el primer cas, si l'usuari vol veure el valor d'un RFID mostrarà per pantalla el text Valor RFID i posarà el sistema en el mode_nou_pass, d'aquesta forma s'executarà la funció new_pass encarregada de mostrar aquest valor. Si per al contrari prem la tecla 2, el sistema sortirà del mode administrador.

```
void menu_admin(){
  char key = keypad.getKey(); //Variable on es guarda la tecla premuda
  if (key != NO_KEY){ //Es verifica si es detecta que l'usuari ha premut alguna tecla, si és així entra en el if
    if (key == '1'){ //Si l'usuari ha premut l'opció 1 podrà veure el valor de la targeta
      lcd.clear();//S'esborra el contingut de text de la pantalla
      lcd.print("Valor RFID"); //S'escriu text per pantalla
      mode_nou_pass = true; //S'assigna a la variable mode_nou_pass valor true per poder llegir el valor d'una targeta
      mode_admin = false; //S'assigna el valor false al mode_admin (surt del panell d'administració)
    }if (key == '2') { //Si l'usuari ha premut l'opció 2 sortirà del mode administrador
      lcd.clear();//S'esborra tot el contingut de la pantalla
      lcd.print("Hola!"); //S'escriu text per pantalla
      mode_admin = false; } } //S'assigna valor false al mode_admin (surt del panell d'administració)
```

La funció new_pass, rebrà el valor de la targeta llegida per paràmetre, i el que farà, és mostrar aquest valor per pantalla, durant 5 segons, i finalment, retornarà Smartlock al seu estat inicial.

```
void new_pass (String targeta){
```

```

mode_admin = false; //S'assigna a la variable mode_admin el valor false, ja que es surt del mode administrador
mode_nou_pass = false; //S'assigna a la variable mode_nou_pass el valor false, ja que es surt del mode new pass
lcd.setCursor(0,1); //Es situa el cursor de la pantalla en la segona fila
lcd.print(targeta); //S'escriu el text amb el valor de la targeta
delay(5000); //S'atura Arduino durant 5 segons perquè l'usuari pugui apuntar el valor
ini(); //Es crida a la funció ini() encarragada d'iniciar el sistema

```

Tot el codi resultant es pot visualitzar en l'annex

3.5 Desenvolupament codi Arduino xip ESP8266

Aquest codi és el firmware que utilitza el xip ESP8266 i que té una estructura similar al de Smartlock, on té declarades unes constants, unes variables globals, la funció setup, la funció loop i tota una sèrie de funcions addicionals que permetran dotar al xip, de les funcionalitats necessàries per fer d'intermediari entre d'Smarlock i el servidor.

Inicialment, estan declarades les constants que tenen l'identificador de la xarxa, la contrasenya i la url del servidor.

```

const char* ssid = "G55_Wifi"; //Nom de la xarxa wifi
const char* password = "contrasenyawifi"; //Contrasenya de la xarxa Wifi
const char* host_server = "www.000webhost.com"; //Direcció del servidor

```

Seguidament, estan declarades les variables globals, que guardaran el contingut de les dues pàgines web del xip, el nombre d'intents de connexió, els missatges rebuts d'Arduino Mega (port Serial) i dues variables booleanes, les quals informaran el sistema si hi ha connexió wifi, i si hi ha connexió amb el servidor.

Quan arrenca el xip ESP8266, el primer que farà per defecte, és executar la funció setup(), on intentarà establir una connexió wifi durant un màxim de 15 cicles (intents), si no ho aconsegueix, informarà a Smartlock que no hi ha connexió ([OFFLINE]) i es posarà en el mode sense connexió (online=false), això vol dir, que a partir d'aquest moment, l'únic que farà el xip, és intentar restablir la connexió wifi, si per al contrari, es connecta, mostrarà pel port serial la IP assignada.

```

void setup() {
  Serial.begin(9600); //S'inicialitza el port serial de comunicació entre ESP8266 i Arduino Mega2560 (serial3)
  intents = 0; //S'inicialitza la variable intents a zero
  Webserver = true; //S'inicialitza la variable webserver a true (servidor en línia)
  targeta = ""; //S'inicialitza la variable targeta
  WiFi.begin(ssid, password); //Es realitza la connexió wifi passant el nom de la xarxa i la contrasenya
  Serial.print("Connectant..."); //S'escriu per serial
  //Es realitza la connexió wifi en arrencar el xip ESP8266
  while (WiFi.status() != WL_CONNECTED && intents < 15) { //Mentre no es detecti una connexió exitosa durant 15 intents
    delay(500); //S'atura Arduino durant 500ms
    Serial.print("."); //S'escriu pel port serial . cada 500ms
    intents++; } //S'augmenta en +1 la variable intents (quan arribi a 15 sortirà del while)
  //Si després de 15 intents el xip no s'ha connectat a la xarxa wifi, informarà el sistema Smartlock que no hi ha connexió
  if (intents >= 15){
    Serial.println("[OFFLINE]"); //S'informa el sistema Smartlock que no hi ha connexió wifi
    online = false; //S'assigna a la variable online el valor false (el sistema entra en la modalitat sense connexió)
  }else{ //Si el sistema s'ha connectat al wifi en menys de 15 intents entrarà en el if
    online = true; //S'assigna a la variable online el valor true (el sistema té connexió wifi)
    Serial.println("Connexió establerta!"); //S'escriu pel port serial
    Serial.print("IP Local: "); //S'escriu pel port serial
    Serial.println(WiFi.localIP()); //S'escriu pel port serial el valor de la IP
  }
}

```

Aquesta funció a més d'intentar establir una connexió wifi, també arrencarà l'objecte server, que permet utilitzar el xip com a servidor web, i carregar diferents pàgines. Per fer-ho incorpora el següent codi:

```

ESP8266WebServer server(80); //Es crea l'objecte server assignant el port 80

```

```
//Es crea la pàgina principal del servidor ESP8266 afegint el contingut a la variable string webPage
webPage += "<script>setTimeout(function(){window.location=\"https://smartlock-uoc.000webhostapp.com/index.php\"},
100); </script>";
server.on("/", []) { //Url principal ip/index.html
  server.send(200, "text/html", webPage); //Es carrega la pàgina principal
});
```

En aquest moment, l'usuari pot accedir al xip ESP8266 amb la seva IP i carregar una pàgina de manera local, per defecte, el servidor redirigirà a l'usuari a la pàgina de login (del servidor web).

D'altra banda, el servidor de l'ESP8266 també disposa d'una altra url; ip\Open, en el moment que l'usuari accedeix a aquesta pàgina, a part de carregar una pàgina nova (webOpen), també enviarà pel port serial, a l'Arduino MEGA, el missatge [ON], això vol dir que s'obrirà la porta. S'ha de tenir en compte, que aquesta funcionalitat només estarà disponible si l'usuari està connectat per wifi a la mateixa xarxa que Smartlock, si no no podrà accedir al servidor. Per fer totes aquestes accions, el xip disposa del codi següent:

```
server.on("/Open", []) { //Url ip/Open
  server.send(200, "text/html", webOpen); //Es carrega la pàgina WebOpen
  Serial.println("[ON]"); //Escriu pel port serial
  delay(1000); //S'atura Arduino durant 1s
};
```

Una vegada finalitzades totes les accions de la funció setup, el xip procedirà a executar la funció loop indefinidament. Aquesta funció s'encarregarà principalment de fer tres accions:

- Si rep el valor d'una targeta pel port Serial, executarà les accions necessàries per fer la comprovació d'autenticació al servidor.
- Si detecta que s'ha perdut la connexió wifi, executarà la funció reconnect encarregada de restablir la connexió.
- Si detecta que s'ha perdut la connexió amb el servidor, executarà la funció verify_server encarregada de fer ping al servidor cada 5 segons fins a aconseguir una resposta positiva (pong).

```
void loop() {
  server.handleClient(); //S'inicialitza el server encarregat de rebre les peticions dels clients i llançar les funcions
  if (Webserver = true){ //Si el servidor està operatiu comprovarà l'accés de la targeta en el servidor
    chekSerialCom(targeta); //Crida a la funció encarregada de guardar en una variable el valor d'una targeta rebuda
    per serial
  }if (WiFi.status() != WL_CONNECTED){ //Si es perd la connexió wifi entrarà en el if
    reconnect (); //Crida a la funció encarregada de realitzar una connexió wifi
  }if (Webserver == false){ //Si el servidor està caigut entra en el if
    verify_server (); } //Crida a la funció encarregada de comprovar si el servidor està operatiu (ping)
```

La funció encarregada de gestionar les comunicacions que rep d'Arduino Mega pel port serial és la checkSerialCom, aquesta funció el que fa, és que en el moment que detecti una comunicació activa pel port serial, esperarà 25ms i guardarà aquesta informació en una variable, posteriorment, executarà unes accions o d'altres en funció del missatge rebut. És molt important fer l'espera de 25ms, donat que el port serial és molt lent, i si no s'atura el xip durant aquest temps, el xip no té temps a rebre el missatge complet, i rep només dos o tres caràcters, en canvi, si s'atura el cicle, aquest missatge es guardarà en el buffer i el rebrà complet.

Hi ha dos tipus d'accions a realitzar en funció del missatge rebut, d'una banda, si rep el missatge [OK2FACTOR] o [FAIL2FACTOR] executarà la funció result2factor, encarregada de guardar en el log del servidor web, el resultat de la doble autenticació, i d'altra banda, si rep qualsevol altre missatge, vol dir que és el valor d'una rfid, i per tant, enviarà aquest valor a la funció check_target, encarregada de fer la connexió amb el servidor i verificar l'accés de l'usuari a l'habitatge.

```
void chekSerialCom(String target) {
  while (Serial.available()) { //Mentre el port serial estigui operatiu entrarà en el while
    delay(25); //S'atura Arduino durant 25ms
    char a = Serial.read(); //Es guarda cada valor rebut pel serial en la variable a
```

```

target.concat(String(a)); //Es guarda cada valor de la variable a en string target
if (a == ']') { //Quan es rep ] vol dir que s'ha acabat el contingut del missatge i entra en el if
  if (target == "[OK2FACTOR]" || target == "[FAIL2FACTOR]") {
    result2factor(target);
  } else {
    check_target(target); } } } //Es crida a la funció encarregada d'enviar el valor de la targeta al servidor

```

Les funcions `check_target` i `result2factor` fan accions molt similars, amb la diferència, que el primer interactua amb l'arxiu `verificacio.php`, encarregat de verificar l'accés de l'usuari, i el segon interactua amb l'arxiu `resultat2factor.php`, encarregat de guardar a l'historial el resultat de la doble autenticació. Aquestes funcions, el primer que fan és crear dos objectes, el `WifiClient` i el `HTTPClient`, que són els dos objectes necessaris per establir la connexió entre l'ESP8266 i el servidor (arxiu php). Una vegada creats es realitzarà la connexió amb l'objecte `HTTPClient`, el qual rebrà per paràmetres el `WifiClient` i la direcció de l'arxiu php. Finalment, es crearan els headers i la variable que s'enviarà pel post.

```

void check_target (String target){
  if(WiFi.status()== WL_CONNECTED){ //si hi ha una connexió wifi entrarà en el if
    String host = "http://smartlock-uoc.000webhostapp.com/verificacio.php"; //Es guarda en la variable host la direcció de
    l'arxiu php encarregat de fer la comprovació
    WifiClient client; //Es crea l'objecte WifiClient
    HTTPClient https; //Es crea l'objecte HttpClient
    https.begin(client, host); //S'inicialitza una connexió http entre el client i el host
    https.addHeader("Content-Type", "application/x-www-form-urlencoded");//S'especifica el content-type header
    //Es guarda en la variable httpRequestData el valor rebut de la targeta (POST)
    String httpRequestData = "targeta=\\" + target + "\\"";

```

En aquest moment, el xip té tot preparat per efectuar la connexió amb l'arxiu php del servidor, per fer-ho utilitzarà la següent sentència, que consisteix a enviar per post, el valor de la targeta a l'arxiu php, i guardar en la variable `httpResponseCode` el resultat de l'operació. Si el resultat és correcte retornarà el valor 200, però si retorna qualsevol altre valor, vol dir que hi ha algun error. En la variable `payload` es guardarà el valor que retorna l'arxiu php qu, que pot ser ser: [ON], [OFF] o [pass]. Finalment, aquest valor l'enviarà pel port serial cap Arduino Mega, i en quant que rebí el missatge, operarà en conseqüència.

```

//Es realitza la connexió https amb el servidor enviant per post el valor de la targeta
int httpResponseCode = https.POST(httpRequestData); //Es guarda en httpResponseCode el codi de la resposta de
l'operació (si el resultat és 200 és que ha sortit correcte)
String payload = https.getString(); //Es guarda en la variable payload el resultat del php ([ON], [OFF] o [pass])
Serial.print("HTTP code: "); //S'escriu pel port serial
Serial.println(httpResponseCode); //S'escriu pel port serial el valor de la variable httpResponseCode
Serial.println(payload); //S'escriu pel port serial el valor de la resposta del servidor ([ON], [OFF] o [pass])
if (httpResponseCode != 200) { //Si httpResponseCode és diferent a 200 entrarà en el if (vol dir que hi ha un error)
  Serial.println("[OFFLINE]"); //S'escriu pel port serial per informar a Smartlock que no hi ha connexió
  Webserver = false; //S'assigna a la variable Webserver el valor false
}https.end();} //Es tanca la connexió https

```

La resta de funcions, `reconnect` i `verify_server`, s'encarregaran únicament de controlar la connectivitat del xip amb el servidor.

Quan s'executa la funció `reconnect`, el primer que farà el xip és informar a Smartlock que no hi ha connexió wifi, per fer-ho, enviarà pel port serial el missatge [OFFLINE]. A continuació intentarà restablir la connexió cada 5 segons, i una vegada ho aconsegueixi, informarà a Smartlock que torna a estar en línia i el firmware passarà a la modalitat online.

```

void reconnect () {
  Serial.println("[OFFLINE]"); //S'escriu pel port serial per informar a Smartlock que no hi ha connexió wifi
  while (WiFi.status() != WL_CONNECTED) { //mentre no hi ha connexió wifi estarà en el while
    WiFi.begin(ssid, password); //S'inicia una connexió wifi amb nom de la xarxa i contrasenya
    delay(5000); //S'atura Arduino durant 5 segons

```

```

}if (WiFi.status() == WL_CONNECTED){//Si es detecta que hi ha una connexió wifi entrarà en el if
  Serial.println("[ONLINE]"); //S'escriu pel port serial per informar a Smartlock que torna a haver-hi connexió wifi
  online = true;} //S'assigna a la variable online el valor true

```

Finalment, la funció `verify_server` el que fa, és enviar un ping cada 5 segons al servidor web, fins que rebí una resposta positiva (pong), una vegada rebí aquesta resposta, informarà a Smartlock que el servidor torna a estar [ONLINE] i posarà el firmware en el mode de WebServer online.

```

void verify_server (){
  while (Ping.ping(host_server) == false){//Mentre no es rebí un pong del servidor, es mantindrà en el while
    delay (5000);}//S'atura Arduino durant 5s
  //Si es rep un pong del servidor, vol dir que torna a estar operatiu
  Serial.println("[ONLINE]"); //S'escriu pel port serial per informar a Smartlock que el servidor està online
  Webserver = true; }//S'assigna a la variable Webserver el valor true

```

Tot el codi resultant es pot visualitzar en l'annex

3.6. Desenvolupament Servidor

3.6.1 Crear un domini i hosting

Per muntar el servidor que necessita Smartlock es necessitarà un domini (on Smartlock es connectarà i passarà els diferents paràmetres per Post i Get) i un hosting amb PHP i MySQL que emmagatzemarà la base de dades de Smartlock i els arxius PHP necessaris per al seu correcte funcionament com ara verificar si la targeta existeix, guardar en un registre els accessos, verificar si un usuari és administrador, etc. En aquest cas, pel prototip s'utilitzarà una opció de domini i hosting gratuït, ja que les limitacions que ofereixen els hostings gratuïts seran suficients per a les necessitats del prototip. En un futur en el qual es produeixi a major escala el producte final, les necessitats seran majors i es passarà a un servidor més segur i que cobreixi totes les necessitats d'un Smartlock a major escala. D'aquesta manera, es reduiran despeses innecessàries durant el desenvolupament del projecte/prototip.

| Free Web Hosting | Single Shared Hosting | Premium Shared Hosting | Business Shared Hosting |
|---|---|---|--|
| \$0 /mo | \$1 /mo | \$2 /mo | \$3 /mo |
| <ul style="list-style-type: none"> ✓ 1 Website ✓ 300 MB Disk Space ✓ Unlimited bandwidth (100 Gb) ✗ No Email Accounts ✗ 24/7/365 Support | <ul style="list-style-type: none"> ✓ 1 Website ✓ 50 GB Disk Space ✓ Limited Bandwidth (100 GB) ✓ 1 Email Account ✓ 24/7/365 Support ✗ Free Domain | <ul style="list-style-type: none"> ✓ 100 Websites ✓ 100 GB Disk Space ✓ Unlimited Bandwidth ✓ 100 Email Accounts ✓ 24/7/365 Support ✓ Free Domain ✓ Weekly Backups | <ul style="list-style-type: none"> ✓ 100 Websites ✓ 200 GB Disk Space ✓ Unlimited Bandwidth ✓ 100 Email Accounts ✓ 24/7/365 Support ✓ Free Domain ✓ Daily Backups |
| See all features | See all features | See all features | See all features |
| FREE SIGN UP | CLAIM DEAL | CLAIM DEAL | CLAIM DEAL |

Il·lustració 28 Tarifes disponibles de 000webhost

En aquest cas, s'utilitzarà un servidor de 000webhost en la versió gratuïta, que permet emmagatzemar fins a 300mb, un lloc web i una base de dades (fins a 1GB, fins a 100 taules), més que suficient per el prototip.

Una vegada seleccionat el pla gratuït de 000webhost, el site demanarà que es creï un compte d'usuari per continuar. Després, demanarà un nom pel projecte i requerirà que es seleccioni una de les tres opcions del hosting, crear un site amb Website Builder, instal·lar un Wordpress o que es puguin pujar directament els arxius al hosting (aquesta serà l'opció escollida).

A great start is half the work

Name Your Project

Smartlock-UJC

Mostrar contrasenya [GENERAR CONTRASEÑA](#)

ENVIAR

Il·lustració 29 Pantalla creació projecte

Here is your tool box

Choose your weapon of mass creation

MÁS VENDIDO

Use Our Website Builder

Instalar WordPress

Upload your site

Seleccionar

Seleccionar

Seleccionar

Il·lustració 30 Pantalla selecció tipus de hosting

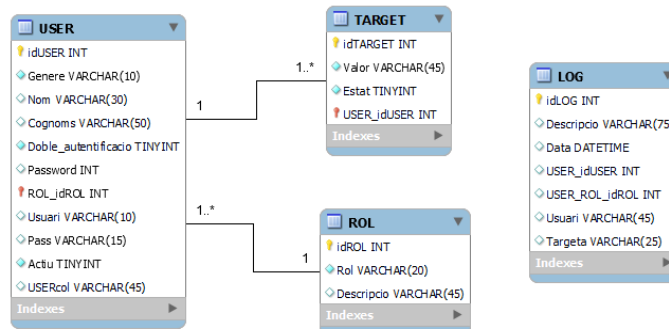
Així doncs, ara ja està configurat el hosting on es guardarà tot el contingut del PHP i tots aquest arxius s'hauran de guardar dintre de la carpeta públic_html perquè es pugui accedir des del domini Smartlock-UOC.



II-lustració 31 Pantalla explorador d'arxius de 000webhost

3.6.2 Disseny de la Base de dades

Un cop que ja es disposa d'un servidor, el primer que es farà abans de crear la base de dades, es dissenyar i crear un diagrama d'entitat relació amb les taules i les connexions entre elles que seran necessàries pel correcte funcionament de Smartlock amb el servidor. A continuació, es pot visualitzar el resultat del mateix.



II-lustració 32 Esquema Entitat-Relació de la base de dades

Taula user: aquesta taula conté tota la informació sobre l'usuari com ara el nom, gènere, si té activada o no la doble autenticació, quines targetes té associades, quin perfil d'usuari té, si és administrador, user, etc.

Taula rol: aquesta taula conté tota la informació dels diferents tipus de rol de Smartlock, que actualment és la d'administrador i user. L'administrador podrà accedir al panell d'administrador per gestionar Smartlock com ara crear nous usuaris, donar de baixa targetes, etc.

Taula target: en aquesta taula es guarda tota la informació de les targetes registrades al sistema, amb el seu número de sèrie i també si estan actives o no.

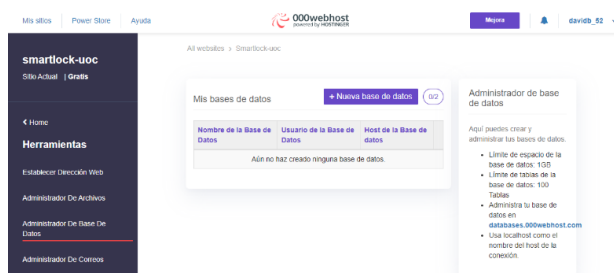
Taula log: en aquesta taula es guardaran tots els accessos al sistema, és a dir, que cada cop s'obri la porta es podrà saber quin usuari ho ha fet i quin dia i hora.

3.6.3 Crear la base de dades en el hosting.

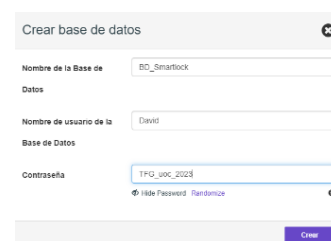
Ara que ja estan definides totes les taules necessàries, els camps de cada taula i les relacions que hi ha entre elles, per tant, ja es pot crear la base de dades en el servidor.

Per fer-ho, s'haurà d'accedir al panell d'administració del site i posteriorment fer clic en eines / administració de la base de dades / Nova base de dades.

Després d'això, s'obrirà una finestra que demanarà introduir les dades del nom de la base de dades, l'usuari i la contrasenya.



II-lustració 33 Pantalla administració de base de dades de 000webhost



II-lustració 34 Pantalla creació base de dades desde 000webhost

Ara que ja està creada la base de dades de Smartlock i des d'aquest hosting es podrà gestionar (carregar SQL, eliminar taules, registres, etc.) des de phpMyAdmin.

3.6.4 Panell d'administrador

El panell d'administrador Smartlock està desenvolupat amb la idea de permetre als propietaris Smartlock tenir un major control sobre els accessos a la habitatge. A part de disposar d'un registre d'accessos (log), l'administrador podrà crear, activar/desactivar i modificar usuaris i targetes del sistema. Per poder accedir a totes aquestes opcions, l'usuari haurà d'iniciar sessió en el sistema i haurà de tenir permisos d'administrador.

El panell d'administrador té tota una sèrie d'arxius php que permeten fer tota la gestió, i tots aquests tenen un codi a l'inici que controlen que hi hagi activa una sessió, és a dir, que si l'usuari no ha obert sessió en el sistema, no podrà accedir a cap dels apartats, i així s'augmenta la seguretat d'Smartlock. A més, també es verificarà la inactivitat de l'usuari, és a dir, que si l'usuari ha iniciat sessió i no carrega la pàgina durant més de 15 minuts, automàticament es tancarà la sessió, d'aquesta forma s'afegeix un altre element de seguretat en cas que l'administrador es deixi oberta una sessió d'usuari. Si la sessió iniciada es d'un rol d'usuari, aquest només podrà accedir a la pàgina user_panel. Per tot això, s'ha introduït a l'inici de totes les pàgines el codi.

```
<?php
//Es comprova si s'ha iniciat sessió, si no està iniciada la sessió llavors dirigeix a l'usuari al login, si està iniciada podrà
veure la pàgina.
ob_start();
session_start(); //s'inicialitza les variables de sessió
if(!isset($_SESSION['user_id'])){
    header("Location: login.php");
    exit;
} else if ($_SESSION['rol'] == '2'){ //Si el rol de la sessió es el de d'usuari accedirà només a l'apertura de porta
    header("Location: user_panel.php");
    exit;
} else if ($_SESSION['rol'] == '1'){ //Si la sessió està inicialitzada com a administrador
    include 'functions.php'; //S'afegeix l'arxiu function.php on estan guardades totes les funcions per smartlock
    //Si es carrega la pàgina web amb la sessió iniciada i hi ha una inactivitat superior a 15 minuts, per
seguretat la sessió es tanca
    echo '<script>
    setTimeout(function(){
    window.location="logout.php"
    }, 900000);
    </script>';
?>
```

3.6.5 Esquema d'arxius

Totes les funcions del panell d'administrador estan dividides en 16 arxius. A continuació, s'explicarà breument la utilitat de cada arxiu, i en l'entrega del codi final, es pot veure tota una sèrie d'anotacions que detallen com s'aconsegueix fer les diferents interaccions entre l'usuari i la base de dades:

- **Style.css:** aquesta és la fulla d'estils del panell d'administració Smartlock, és a dir, l'encarregada de dir al navegador el tipus de lletra utilitzada, mida, fons de pantalla, mides dels contenidors, mides de les imatges, alineació, etc.
- **Functions.php:** aquest php disposa de tota una sèrie de funcions que són necessàries per al funcionament del panell d'administració, aquestes funcions es divideixen principalment en 4 apartats, funcions genèriques (connectar a la base de dades, revisar si una variable es nulla, fer login etc), per a usuaris (mostrar l'informació dels usuaris tots els usuaris, mostrar informació d'un usuari, etc), per a targetes (mostrar informació de totes les targetes, mostrar les targetes propietats d'un usuari, etc) i per al log (mostrar tots els registres, mostrar tots els registres d'una targeta o usuari determinat, etc.).

- **Index.php:** aquesta és la pàgina principal del panell, on l'administrador podrà veure tots els usuaris que estan en actiu i els últims 15 registres d'accés al sistema. També, des del mateix panell podrà fer clic al símbol de la lupa per veure més detall sobre un usuari concret.
- **Login.php:** aquesta pàgina es carregarà sempre que no hi hagi activa una sessió, i permetrà a l'usuari introduir el seu usuari i contrasenya al sistema, i aquest verificarà si els paràmetres són correctes i a més, si aquest usuari disposa de permisos d'administrador o usuari.
- **Logout:** aquesta pàgina permet a l'usuari tancar la sessió.
- **Usuari.php:** aquesta pàgina rebrà per URL (GET) l'idUsuari i mostrarà tota la informació d'aquest en un formulari, el qual podrà modificar i aplicar els canvis a la base de dades, demanant a UpdateUsuari.php. A més, també mostrarà totes les targetes que té aquest usuari i els seus moviments.
- **Usuaris.php:** des d'aquest apartat l'usuari disposarà d'un formulari que li permet crear nous usuaris en el sistema Smartlock, per fer-ho cridarà a la pàgina InsertUser.php. A més, mostrarà en la part inferior tots els usuaris que té el sistema actius e inactius (a diferència d'index.php) i si es fa clic en el símbol de la lupa, es podrà veure tota la informació d'un usuari concret (usuari.php).
- **Tarjeta.php:** aquesta pàgina mostrarà en un formulari tota la informació d'una targeta concreta, que haurà rebut per l'URL (GET) i podrà modificar el seu contingut i aplicar-ho a la base de dades cridant a UpdateTargeta.php. A més, mostrarà informació sobre el propietari de la targeta i els registres d'accessos que s'han fet amb aquesta.
- **Targetes.php:** aquesta pàgina té un formulari que permet a l'usuari crear noves targetes i introduir-les al sistema, per fer-ho es cridarà a la pàgina InsertTarget.php. A més, mostrarà totes les targetes del sistema i si es fa clic en la lupa es podrà veure el detall d'una targeta determinada (targeta.php).
- **InsertTarget.php:** aquesta pàgina s'encarregarà d'afegir una nova targeta a la base de dades, rebrà tots els paràmetres pel formulari de targetes.php a través de POST, i informarà l'usuari si s'ha pogut inserir correctament o hi ha algun error. Finalment, redirigirà a l'usuari de nou a targetes.php
- **InsertUser.php:** aquesta pàgina s'encarregarà d'afegir un nou usuari a la base de dades, rebrà tots els paràmetres pel formulari d'usuaris.php a través de POST, i informarà l'usuari si s'ha pogut inserir correctament o hi ha algun error. Finalment, redirigirà a l'usuari de nou a usuaris.php
- **UpdateTargeta.php:** aquesta pàgina s'encarregarà de fer les modificacions dintre de la taula TARGET de la base de dades, aquestes dades les rebrà per POST des del formulari de la pàgina targeta.php. Finalment, es mostrarà el resultat de la modificació en pantalla informant l'usuari si s'ha realitzat la modificació o hi ha algun error, i redirigirà de nou a targetes.php
- **UpdateUsuari.php:** aquesta pàgina s'encarregarà de fer les modificacions dintre de la taula USER de la base de dades, aquestes dades les rebrà per POST des del formulari de la pàgina usuari.php. Finalment, es mostrarà el resultat de la modificació en pantalla informant l'usuari si s'ha realitzat la modificació o hi ha algun error, i redirigirà de nou a usuaris.php
- **Verificació.php:** aquesta pàgina no s'utilitzarà en el panell d'administració, sinó que serà l'arxiu que utilitzarà Smartlock per fer les comprovacions necessàries (usuari actiu, targeta activa, targeta existent, doble autenticació) fins a determinar si la targeta que ha llegit Smartlock té accés al sistema o no, i retornarà un resultat a Arduino, el qual farà una acció o un altre segons la resposta rebuda. Per acabar, guardarà l'acció realitzada en la taula LOG (accés denegat, accés autoritzat, etc).
- **Result2factor.php:** aquest php rep per POST el missatge [OK2FACTOR] o [FAIL2FACTOR], en funció del missatge rebut, crearà una entrada en l'historial d'Smartlock (taula LOG) amb un missatge o un altre.
- **User_panel:** aquesta pàgina només la podrà visualitzar els usuaris que tinguin el rol user, i amb la qual podran obrir la porta de l'habitatge si estan connectats a la mateixa xarxa wifi.

3.6.6 Comprovació d'accés a través del servidor

Verificació.php que és un arxiu php, rebrà el valor de la targeta llegida en Smartlock per POST, i s'encarregarà de fer diferents comprovacions en la base de dades, fins a determinar si la targeta llegida pot entrar a l'habitatge o no. Per fer-ho, el primer que farà és connectar-se a la base de dades de la següent manera:

```
//Paràmetres per a la connexió a la base de dades
$user = "id20681247_david";
$pass = "TFG_uoc_2023";
$server = "localhost";
$db = "id20681247_bd_smartlock";
$con = mysqli_connect($server, $user, $pass, $db); //Es realitza la connexió a la base de dades
```

Ara que ja es té accés a la base de dades, es realitzaran les següents comprovacions en aquest ordre, si una comprovació no surt positiva, ja no es comprovarà la següent:

1. Que la targeta estigui registrada al sistema i estigui activada.

```
$ID_Smartlock = $_POST['targeta']; //Es guarda el valor rebut per la url en la variable $ID_Smartlock
//Es realitza la consulta per buscar si existeix alguna entrada a la base de dades amb el valor passat pel GET
$result = mysqli_query($con, "SELECT * FROM `TARGET` WHERE `Valor` = $ID_Smartlock AND `Estat` = '1'");
//Es guarda en la variable $total el nombre total de resultats de la consulta si el resultat es 0 vol dir que no hi ha cap
targeta i si es 1 vol dir que existeix la targeta en el sistema
$total = $result->num_rows;
```

2. Que l'usuari propietari de la targeta estigui actiu.

```
$consulta = mysqli_fetch_array($result); //Es guarda la informació de la consulta
$ID_User = $consulta['USER_idUSER']; //Es guarda el camp USER_idUSER la variable ID_USER (BBDD)
//Es fa una consulta per verificar si l'usuari està actiu
$user_info = mysqli_query($con, "SELECT * FROM `USER` WHERE `idUSER` = $ID_User AND `Actiu` = '1'");
//Es guarda en la variable $total el nombre total de resultats de la consulta si el resultat es 0 vol dir que no hi ha cap
usuari no està actiu i si es 1 vol dir que l'usuari està actiu
$total_user = $user_info->num_rows;
```

3. Si l'usuari té activada la doble autenticació.

```
$consulta_user = mysqli_fetch_array($user_info); //Es guarda la informació de la consulta en un array
//Es realitza una consulta amb la intenció de buscar en la taula d'usuaris si l'usuari propietari del identificador RFID té
activada la doble autenticació
$result_auth = mysqli_query($con, "SELECT * FROM `USER` WHERE `idUSER` = $ID_User AND `Doble_autenticacio` =
'1'");
//Es guarda en la variable total_auth el total de resultats de la consulta anterior si el valor es 0 vol dir que no té activa la
doble autenticació i si es 1 vol dir que té activada la doble autenticació
$total_auth = $result_auth->num_rows;
```

En cada comprovació tant si passa a la següent comprovació, com si surt denegat l'accés, es farà un INSERT en la taula LOG, on s'enregistrarà dia i hora de l'accés o intent d'accés, amb quina targeta, quin usuari, quin rol té i una petita descripció amb el resultat (accés concedit, accés denegat, usuari inactiu, doble autenticació, etc). Per fer l'insert es farà de la següent forma:

```
//Es guarda a $sql la sentència del insert a executar a la base de dades
$sql = "INSERT INTO LOG (idLOG, Descripcio, Data, USER_idUSER, USER_ROL_idROL, Usuari, Targeta) VALUES (NULL, 'Accés
autoritzat', '$DateAndTime', '$ID_User', '$ROL', '$nom', '$targeta_valor')";
mysqli_query($con, $sql); //S'executa l'insert a la taula LOG
```

Després de fer les comprovacions i l'entrada al LOG, verificacio.php retornarà tres possibles valors a Smartlock, en funció del resultat de la consulta, si l'usuari té accés retornarà un [ON], si l'usuari no té accés retornarà un [OFF], i si té accés, però té activada la doble autenticació, retornarà el valor del camp USER.Password entre claudàtors.

```
echo "[ON]"; //Accés concedit
echo "[OFF]"; //Accés denegat
echo "[".$consulta_user ['Password']."]"; //Valor del password (TwoFactor)
```

Finalment, es tancarà la connexió amb la base de dades, per evitar consumir recursos innecessàriament, i deixar moltes connexions obertes, per fer-ho s'utilitzarà el següent codi php:

```
mysql_free_result($result); //S'allibera la connexió establerta amb la base de dades
mysql_close($con); // Es tanca la connexió amb la base de dades
```

En l'arxiu php es disposa de tota la documentació necessària per entendre cada pas de comprovació, gràcies a tots els comentaris introduïts en el codi.

3.6.7 Funcions principals de functions.php

En aquest arxiu, tal com s'ha mencionat anteriorment es troben tota una sèrie de funcions que permeten controlar gairebé totes les operacions que es poden realitzar des del panell d'administrador. A continuació s'explicaran algunes de les principals.

- Funció nobuit, s'utilitza principalment en els formularis de creació de nou usuari i modificació, i el que fa és rebre un valor per paràmetre, si aquest valor està buit, retornarà un NULL i si no, retornarà el mateix valor que ha rebut per paràmetre. Aquesta funció s'ha desenvolupat perquè hi ha certs camps del formulari d'usuari que no són d'obligatori compliment, i si es deixaven buits, al fer l'insert o l'update donava error, en canvi, amb aquesta funció si detecta que no s'ha omplert el camp del formulari, s'introduirà un NULL en l'Insert o Update.

```
function nobuit($var){
    if ($var == ""){$var = "NULL";return $var;}
    else {$var = "" . $var . "" ;return $var;}
}
```

- Login: Aquesta funció rep per paràmetre la connexió a la base de dades i el valor d'usuari i contrasenya del formulari login.php. I amb tot això el que farà és comprovar si en la base de dades existeix un usuari amb aquest usuari i contrasenya i a més, es trobi actiu, si és així, crearà tres variables de sessió 'user_id', 'rol' i la 'user_name'.

```
function login($con, $usuari, $pass){
    //Es realitza la consulta a la taula USER
    $result = mysqli_query($con, "SELECT * FROM `USER` WHERE `Actiu` = 1 AND `Usuari` LIKE '$usuari' AND `Pass` LIKE '$pass'");
    $total = $result->num_rows;
    //Si s'ha obtingut algun resultat vol dir que l'usuari existeix al sistema i compleix els requisits.
    if ($total == 1){
        $query = mysqli_fetch_array($result); //Es guarda els resultats de la consulta en un array
        $_SESSION['user_id'] = $query['idUSER']; //Es guarda en una variable de sessió Id de l'usuari
        $_SESSION['user_name'] = $query['Nom']; //Es guarda en una variable de sessió el valor del nom de l'usuari
        $_SESSION['rol'] = $query['ROL_idROL']; //Es guarda en una variable de sessió el valor del nom de l'usuari
    }
}
```

- Les funcions info_targeta i info_usuari són molt similars, i el que fan és rebre per paràmetre la connexió i el valor de la targeta o usuari que es necessita. Amb aquests paràmetres realitzarà una consulta a la base de dades i retornarà un array amb el resultat.

```
function info_usuari($con, $usuari){
    $result = mysqli_query($con, "SELECT * FROM `USER` WHERE `idUSER` = '$usuari'");
    $row = mysqli_fetch_array($result); //Es converteix el resultat en un array
    return $row; }
}
```

- Les funcions mostrar *, hi ha diferents funcions que s'utilitzen per mostrar tots els usuaris, un usuari, totes les targetes, una targeta, tots els registres, etc. Totes aquestes funcions són molt similars, només canvia la taula on es fa la consulta i els camps que té. El que es fa és realitzar una consulta i posteriorment mostrar en una taula tot el contingut de la consulta.

```
function mostrar_[$...]($con){
    $result = mysqli_query($con, "SELECT * FROM `USER`"); //Es fa una consulta
    //Es creen les capçaleres de la taula
    echo "<table border='\\" . $table . "\" cellpadding=2 style='\\" . $style . "\"><tr>";
```

```

echo "<td><font face=\"verdana\"><b>idUsuari</b></font></td>";
      [etc...]
//Es fa un bucle encarregat de mostrar tot el contingut de la taula
while($row = mysqli_fetch_array($result)){ //mentre hi hagi contingut
  echo "<td width=\"25%\"><font face=\"verdana\">".
    $row['Genere'] . "</font></td>";
      [etc...]

```

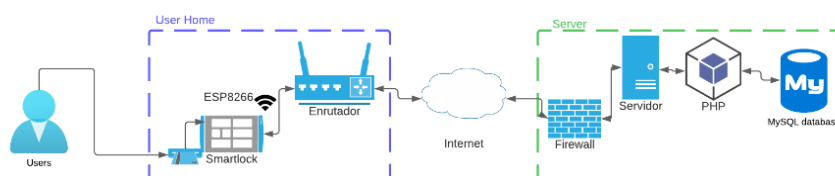
Es pot accedir al panell d'administrador des d'aquest [enllaç](#), amb el nom d'usuari **UOC** i contrasenya **2023**

3.7 Diagrames funcionals

A continuació, per entendre millor el funcionament d'Smartlock es mostraran diferents diagrames funcionals així com una explicació de cadascun dels diagrames.

3.7.1 Diagrama de connexions

A continuació, es pot visualitzar el diagrama de connexions de Smartlock, on l'usuari tindrà accés a Smartlock, i aquest estarà connectat al router de la casa de l'usuari a través d'una connexió Wi-Fi. A més, aquest router li donarà accés a un servidor a través d'internet que li enviarà i rebrà informació.

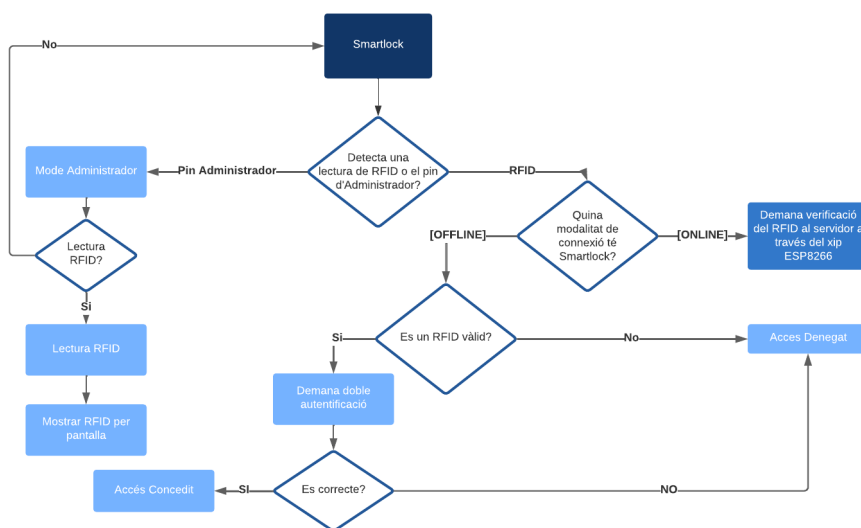


Il·lustració 35 Diagrama de connexions

3.7.2 Diagrama de flux Smartlock

El següent diagrama de flux d'Smartlock, mostra que el dispositiu pot rebre dos tipus d'ordres diferents, que són, el pin d'administrador i la lectura d'un RFID. Si rep el pin d'administrador, el sistema entrarà en la modalitat d'administrador local i apareixerà per pantalla un menú, on l'usuari podrà decidir si vol sortir o vol veure el valor d'un RFID per pantalla.

Si per al contrari, el que rep és la lectura d'un RFID, el primer que farà el sistema és comprovar si té connexió amb el servidor, si és així, enviarà la consulta al servidor, i aquest, retornarà el resultat de l'autenticació. En el cas que no tingui connexió, es farà la comprovació de manera local, en primer lloc, verificarà si la targeta existeix en el sistema, si no és així denegarà l'accés, però si existeix d'identificador, demanarà a l'usuari que introdueixi la contrasenya de la doble autenticació. Finalment, si l'usuari, ha introduït la contrasenya correctament, podrà accedir a l'habitatge.



Il·lustració 36 Diagrama de flux Smartlock

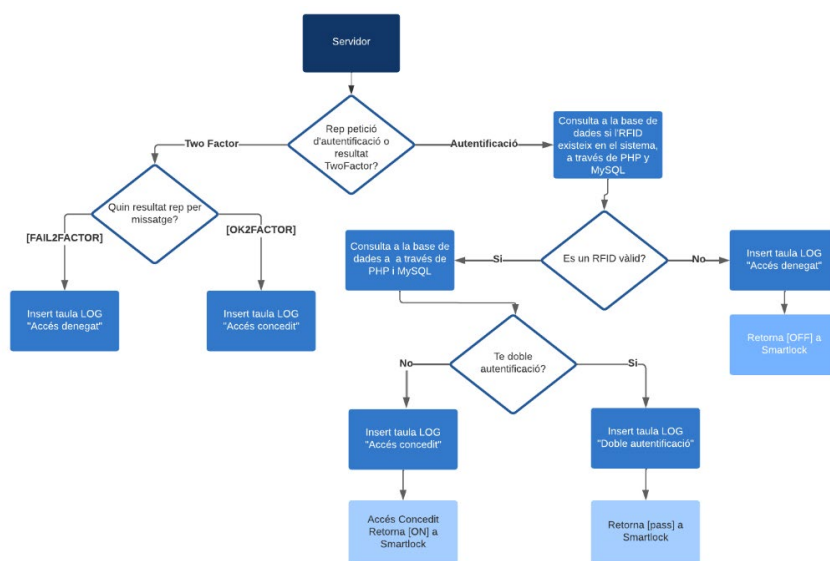
3.7.3 Diagrama de flux del servidor

En el diagrama següent es mostra el diagrama de flux del servidor d'Smartlock, on es pot observar que pot rebre dos tipus de sol·licituds, d'una banda, el resultat d'una doble autenticació, i d'altra banda una sol·licitud de verificació d'autenticació d'un identificador RFID.

En el primer cas, el servidor utilitzarà l'arxiu result2factor.php, on en funció del missatge rebut per paràmetre, guardarà un registre nou en la taula Log de la base de dades.

En el segon cas, el servidor utilitzarà l'arxiu verificacio.php, el qual rebrà per paràmetre el valor d'un identificador.

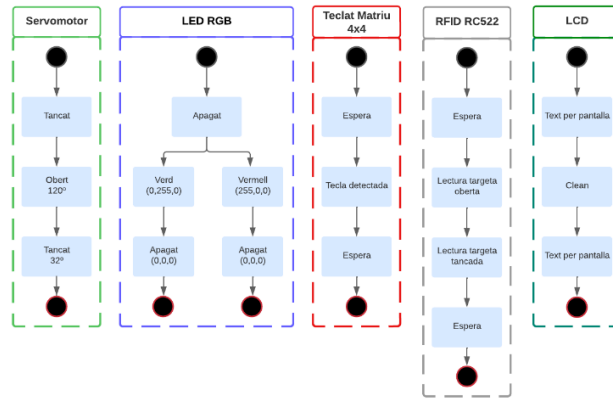
El primer que farà el codi es connectar-se a la base de dades i buscar si aquest identificador existeix en el sistema, si no existeix, crearà un nou registre en la taula LOG i retornarà un [OFF] a Smartlock, perquè aquest denegui l'accés. En canvi, si l'RFID existeix en el sistema, consultarà en la base de dades si l'usuari té activada la doble autenticació, si no està activada, crearà un nou registre a la taula LOG i retornarà un [ON] al dispositiu (Smartlock obrirà la porta). Però, si té activada la doble autenticació, crearà un registre nou en la taula LOG i retornarà els 4 dígits de la contrasenya (entre claudàtors) al dispositiu Smartlock.



Il·lustració 37 Diagrama de flux del servidor

3.7.4 Diagrama d'estats dels sensors i components

- **Servomotor:** El servomotor es pot trobar en dos estats diferents, tancat i obert, al iniciar el sistema, el motor es trobarà a 32° que és la posició de tancat. Quan s'obri (durant uns segons), es mourà a la posició de 120°, la qual permetrà obrir la porta.
- **LED RGB:** El LED RGB es trobarà en 3 estats diferents, a l'iniciar el sistema es trobarà en estat apagat (0,0,0), quan detecti la lectura d'un identificador RFID o un accés correcte s'il·luminarà de color verd (0,255,0) i quan sortí un error s'il·luminarà de color vermell (255,0,0).
- **Teclat matriu:** El teclat matriu té dos estats, quan s'inicia el sistema, estarà en estat d'espera i quan l'usuari premi alguna tecla s'enviarà al sistema quina tecla ha sigut i tornarà de nou a l'estat d'espera.
- **RFID:** El lector RFID es pot trobar en 3 estats diferents. Inicialment, es trobarà en estat d'espera, fins que detecti un identificador RFID on començarà la lectura d'aquest. Una vegada llegit l'identificador, es tancarà la lectura i tornarà a estar en estat d'espera.
- **LCD:** La pantalla es pot trobar en 2 estats diferents. En primer lloc, es mostrarà un missatge per pantalla, però cada cop que es hagi de mostrar un missatge nou, la pantalla farà un clean per esborrar el contingut actual i mostrar el nou contingut.



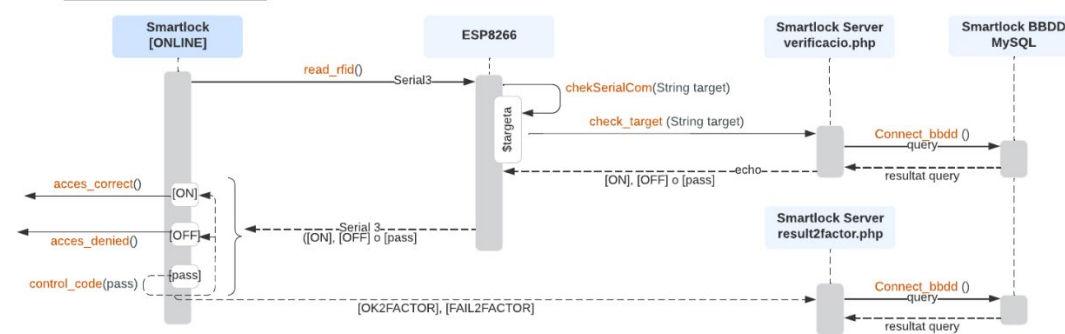
Il·lustració 38 Diagrama d'estats dels sensors i components

3.7.5 Diagrama de seqüència

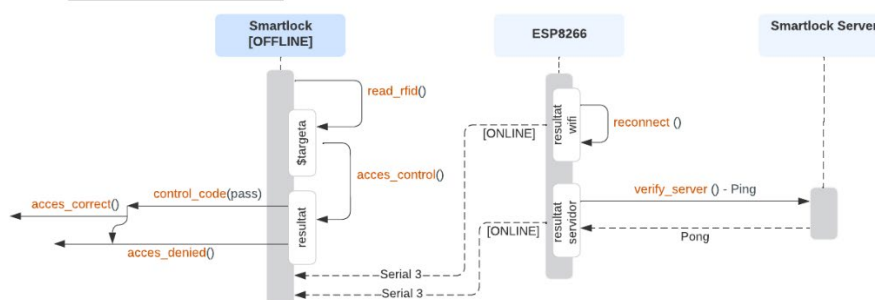
En aquesta imatge es pot veure el diagrama de seqüències que segueix Smartlock quan té connexió amb el servidor. En el moment que detecta un identificador RFID, Smartlock executarà la funció `read_rfid()`, que és l'encarregada de llegir la targeta i guardar-la en una variable. Una vegada guardada, enviarà pel port Serial3, aquest valor al xip ESP8266, i s'executarà la funció `checkSerialCom`, la qual en rebre aquest valor, li donarà el format adequat i el guardarà en la variable `targeta`. Posteriorment, s'enviarà aquesta variable, a la funció `check_target`, que és l'encarregada d'enviar el valor a l'arxiu `verificació.php` pel mètode POST. Ara que el servidor té el valor de la targeta, l'arxiu php es connectarà a la base de dades de Smartlock i consultarà si la targeta és vàlida o no, i també si l'usuari té activada la doble autenticació. Segons el resultat de la consulta, retornarà un [ON] si té accés, un [OFF] si s'ha de denegar l'accés o els quatre dígits de la doble autenticació (en el cas que estigui activada), a més crearà un registre en la taula LOG amb el resultat de la consulta.

Aquesta informació la rebrà el xip ESP8266 i la reenviarà a Smartlock pel port Serial 3. Finalment, Smartlock en funció de la resposta rebuda, obrirà la porta (`access_correct`), denegarà l'accés (`access_denied`) o demanarà que introdueixi el codi d'autenticació (`control_code`). En el cas de la doble autenticació, una vegada l'usuari hagi introduït la contrasenya (màxim 3 intents) informará el xip el resultat de la doble autenticació.

Smartlock amb connexió servidor



Smartlock sense connexió servidor



Il·lustració 39 Diagrama de seqüència

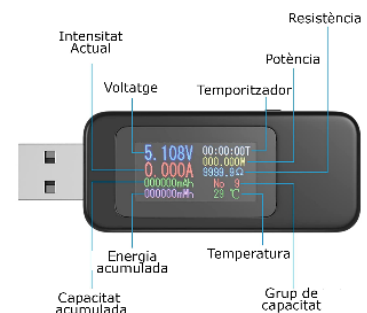
Com es pot observar en el diagrama de seqüència anterior, en el cas que Smartlock no tingui connexió amb el servidor, i detecti una lectura d'un RFID, executarà la funció `read_rfid`, encarregada de guardar el valor de la targeta en una variable, però en aquest cas, enviarà aquest valor, a la funció `accés_control`, el qual comprovarà de manera local i interna, si l'identificador RFID té accés al sistema, si no té accés, executarà la funció `accés_denied`, però si per al contrari, té accés, executarà la funció `control_code`, això farà que l'usuari hagi d'introduir la contrasenya de seguretat per defecte, si finalment la introdueix correctament, obrirà la porta (`accés_correct`) i si no denegarà l'accés (té tres intents). Mentrestant, el xip ESP8266 estarà executant en bucle i de manera independent, dues funcions, `reconnect` i `verify_server`, encarregades d'intentar connectar-se de nou a la xarxa wifi i de comprovar si el servidor torna a estar en línia (fent pings al servidor). En el moment que el xip detecti que tot funciona correctament i hi ha connexió amb el servidor, enviarà pel port serial el missatge [ONLINE], i Smartlock, tornarà a funcionar en el mode online. Cal destacar que quan Smartlock funciona en la modalitat OFFLINE, no s'enregistraran canvis en l'històric, és a dir, durant el període en el que estigui sense connexió no es tindrà constància dels accessos a l'habitatge.

4. Validació i campanya de proves

4.1. Dispositiu Smartlock

4.1.1 Proves de consum

Un dels temes importants d'aquest projecte és també el consum. D'una banda, en l'àmbit mediambiental, ja que és un sistema que està les 24h al dia connectat, i es vol que sigui un sistema de baix consum, i que l'impacte sigui el menor possible. I d'altra banda, també es vol poder dotar al sistema d'autonomia en cas de tall elèctric, una autonomia mínima de 2h (Objectiu 06 requisit 015). Per realitzar les proves, s'ha utilitzat un voltímetre / multímetre que dona diferent informació, tal com es pot veure en la imatge de la dreta.



II·lustració 40 Esquema d'un voltímetre

Per calcular el consum, s'han fet tres proves de 15 minuts cadascuna. A la primera, s'ha fet un ús intensiu de l'aparell amb múltiples obertures, lectures de targeta, etc. A la segona, s'ha fet un ús mitjà del sistema, i en l'última prova, s'ha deixat en repòs (en consum mínim), i aquest són els resultats:

| Prova 1: Ús intensiu | Prova 2: Ús mitjà | Prova 3: en repòs |
|--|--|--|
| 5.035V 00:15:01T 0.167A 000.840W 000041mAh No 4 000209mWh 28 °C | 5.034V 00:15:00T 0.165A 000.830W 000038mAh No 3 000197mWh 28 °C | 5.038V 00:15:01T 0.153A 000.770W 000038mAh No 5 000193mWh 28 °C |
| 41mAh x 4= 164mAh | 38mAh x 4= 152mAh | 38mAh x 4= 152mAh |
| 164mAh + 152mAh + 152mAh = 468mAh / 3 = 156mAh | | |

Com es pot observar, no hi ha grans diferències en el consum. Per fer els càlculs, s'ha multiplicat la capacitat acumulada (mAh) per quatre, i així es tindrà el consum total en una hora, i posteriorment, s'ha fet la mitja entre els tres resultats per tenir un valor més aproximat. A més, es pot observar que el sistema faria un consum de 156mAh cada hora, si s'utilitzen dues bateries model 18650 de 3500mAh cadascuna, 7000mAh totals, donaria un total d'autonomia teòrica de 44 hores, encara que s'ha de tenir en compte el desgast de les mateixes i que a vegades la capacitat total que diu el fabricant no és 100% fiable. No obstant això, la implementació de les bateries no s'ha realitzat en aquesta primera versió per temes econòmics, i ja s'havia mencionat en el pressupost com elements opcionals a futur.

4.1.2 Usabilitat targeta i teclat

4.1.2.1 Targeta o identificador RFID

S'han realitzat múltiples proves amb targetes i clauers, on s'han tret les següents conclusions:

- Superfície de lectura de l'identificador RFID; s'ha observat que en el cas del clauer, la lectura es realitza correctament en tota la seva superfície. En el cas de les targetes, no es pot realitzar la lectura correctament en les extremitats més separades del xip.
- Distància màxima de lectura del lector RFID RC522; s'han aconseguit fer lectures fins a poc més 1 cm, és a dir, que es compleix l'objectiu 2 requisit 4, on es necessitava realitzar una lectura de targeta entre 0 i 1.5cm. En els casos dels clauers, la distància ha sigut menor que en les targetes, i també, s'ha observat que algunes targetes de més qualitat tenen un rang superior a d'altres més senzilles.
- Lectura de dos identificadors alhora; si s'intenta fer una lectura de dos identificadors (targeta o clauer) alhora, el sistema funciona correctament, i fa la lectura d'una sola de les targetes, és a dir, no afecta el correcte funcionament Smartlock.

4.1.2.3 Teclat

S'han realitzat múltiples proves per comprovar el correcte funcionament del teclat, les quals consistien en prémer fort les tecles, prémer més d'una tecla alhora, introduir lletres quan demanava un codi numèric per a la doble autenticació, i en tots els casos el resultat ha sigut satisfactori.

4.1.3 Proves de so



Il·lustració 41
Proba de so

Un dels requisits del projecte (Objectiu 02 Requisit 05 i Objectiu 08 requisits 23,24 i 25), era que l'altaveu instal·lat en Smartlock pogués emetre un so superior a 50dB (so d'una persona parlant) i inferior a 90db (so d'una persona cridant) amb la intenció que l'usuari pugui sentir aquest so. Per realitzar la prova s'ha modificat el codi ampliant el temps de so de l'altaveu, d'aquesta manera: tone(pin, freqüència, duració) i amb una app de mòbil i apropant el micro perquè capti el so s'han realitzat diferents proves i totes elles anaven en l'indiar entre 60-84db, per tant, es compleix amb el requisit del projecte.

4.1.4 Probes de connectivitat

4.1.4.1 Temps de connexió al router i problemes de connexió

S'han realitzat múltiples proves per quantificar el temps que triga Smartlock en realitzar la connexió amb el router, i s'han obtingut resultats entre 4 i 15 segons. En el cas que el xip ESP8266 (Wi-Fi) intenti connectarse al router més de 15 cops i no ho aconsegueixi, envia un avís a Smartlock per notificar que operi en la modalitat OFFLINE. Es compleix l'objectiu 6 requisit 16.

4.1.4.2 Velocitat de resposta

S'han realitzat múltiples proves per quantificar el temps que triga Smartlock entre la lectura de la targeta i la resposta que rep del servidor, que pot ser denegada, accés concedit o demanar la doble autenticació. Durant les diferents proves fetes en diferents dies, amb diferents targetes i amb els tres possibles resultats que retorna el servidor, s'ha comprovat que el temps de resposta oscil·la entre 700ms i 1400ms, és a dir, que es compleix amb l'objectiu 3 requisit 8 i objectiu 8 requisit 23, en el que s'esperava aconseguir una resposta en menys de dos segons.

4.2. Servidor

4.2.1 Consistència de la base de dades

En el disseny inicial de la base de dades, es va relacionar la taula LOG amb els usuaris i targetes, això va generar problemes, ja que no permetia eliminar targetes ni usuaris que estiguessin en algun registre, ni tampoc modificar determinats camps. Finalment, es va arribar a la conclusió que el disseny inicial de la base de dades no era el correcte, i es va decidir aïllar la taula LOG. És a dir, no tindria relacions amb les taules USER i TARGET, d'aquesta manera

es podrien fer modificacions en aquestes taules i a més, si per exemple, un usuari canvia de nom, en el LOG es mantindria el nom de l'usuari que va obrir la porta en el seu moment, i no es modificarien els registres anteriors amb el nou nom. També es va modificar el camp d'usuari de la taula user com a UNIQUE, per evitar conflictes en l'inici de sessió, on diferents usuaris tinguessin el mateix nom d'usuari i contrasenya. Es compleix tot l'objectiu 4.

4.2.2 Proves en el login

S'han fet diferents proves de login, tant deixant els camps buits, omplint amb informació errònia i omplint amb informació correcta, i en tots els casos s'ha obtingut el resultat esperat. També s'ha comprovat si el sistema té en compte que l'usuari estigui actiu en el moment del login, si no és així no deixi iniciar sessió encara que les credencials siguin correctes, i els resultats són els correctes. Alhora s'ha comprovat que els usuaris amb el rol user, només pot accedir a la pàgina d'obertura de porta (user_panel), i no té accés a la resta de pàgines. Es compleix l'objectiu 7 requisit 18.

4.2.3 Proves en els formularis targeta i usuari

En el panell d'administrador s'han fet proves de creació i modificació d'usuaris i contrasenyes, i s'ha detectat un problema en el cas dels usuaris. Aquesta taula té camps que no són d'obligatori compliment (usuari, contrasenya i password) i si no s'omplien aquests camps en el formulari sortia un error tant en la modificació com en la creació d'usuaris. Per resoldre aquest problema, s'ha creat una funció anomenada nobuit, que el que fa és comprovar si el camp del formulari està buit. Si es així, assignarà un valor NULL a aquell camp, i la base de dades en rebre un valor null permetrà fer l'Insert o Update. També, s'ha verificat que els formularis HTML tinguessin els mateixos camps d'obligatori compliment (required) que els de la base de dades (NOT NULL).

4.2.4 Proves verificació.php

Després de realitzar diferents proves de verificació.php, es va comprovar que l'horari que introdueix l'arxiu php no és el correcte, la data i els minuts eren correctes, però l'hora no, per aquest motiu es va afegir una funció php per assignar manualment la zona horària correcta.

```
|date_default_timezone_set("Europe/Madrid");
```

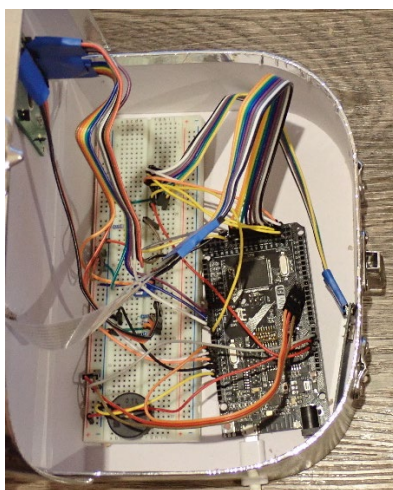
També s'ha comprovat que la verificació funcioni correctament, es a dir, els usuaris inactius i/o targetes inactives no poden obrir la porta. Es compleixen els objectius 2 requisit 6 i objectiu 5 requisit 14.

4.3 Acabat final -Smartlock

Per finalitzar, s'ha ficat tot el circuit Smartlock dintre d'una capsula, per evitar que els cables es desconnectin, entri pols, es pugui penjar, etc. L'acabat final per l'usuari quedaria més compactat que aquest prototip.



Il·lustració 42 Smartlock final frontal



Il·lustració 43 Smartlock final interior



Il·lustració 44 Smartlock final lateral

4.4 Acabat final -Servidor

A continuació es mostren captures de pantalles d'algunes de les pàgines del panell d'administrador.

SMARTLOCK DASHBOARD

Hola, David (Tancar sessió)

Usuaris actius d'Smartlock

| IdUsuari | Gener | Nom | Cognoms | Doble autenticació? | Password | Rol | Actiu | Usuari | Contrasenya |
|----------|----------|---------|--------------|---------------------|----------|-----|-------|--------|-------------|
| 1 | Home | David | Bassa Romera | 1 | 1234 | 1 | 1 | UOC | 2023 |
| 2 | NeBinari | Davínia | Das Santos | 0 | | 2 | 1 | User | 2023 |
| 5 | NeBinari | Julijan | Bassa Romera | 0 | | 3 | 1 | | |
| 8 | Home | Ludá | Cognoms | 0 | 1234 | 2 | 1 | NULL | NULL |
| 9 | Home | Javi | Molina | 0 | | 1 | 1 | | |
| 12 | Home | Javi | asoffasfd | 0 | | 1 | 1 | | |
| 13 | Home | Javi | Molina | 1 | | 1 | 1 | Moll | 1234 |

Total usuaris: 7

Ultims 15 accessos

SMARTLOCK HISTORIAL

Hola, David (Tancar sessió)

Tots els accessos

| IdLOG | Descripció | Data | Nom Usuari | Targeta | Rol | IdUsuari |
|-------|---------------------------------|---------------------|---------------------|----------------|-----|----------|
| 233 | Es demana la doble autenticació | 2023-06-11 15:22:10 | Bassa Romera, David | [13 20 18 ac] | 1 | 1 |
| 232 | Es demana la doble autenticació | 2023-06-11 15:18:45 | Bassa Romera, David | [13 20 18 ac] | 1 | 1 |
| 231 | Clau introduïda correcte | 2023-06-08 19:54:27 | | | | |
| 230 | Es demana la doble autenticació | 2023-06-08 19:54:09 | Bassa Romera, David | [13 20 18 ac] | 1 | 1 |
| 229 | Clau introduïda correcte | 2023-06-08 19:53:26 | | | | |
| 228 | Es demana la doble autenticació | 2023-06-08 19:53:15 | Bassa Romera, David | [13 20 18 ac] | 1 | 1 |
| 227 | Clau introduïda incorrecte | 2023-06-08 19:54:52 | | | | |

SMARTLOCK USUARIS

Hola, David (Tancar sessió)

Afegir nou usuari

Gener: Cognoms:

Rol: Actiu:

Doble autenticació? Activar? Password:

Panel·l administrador Usuari: Contrasenya:

Afegir | Esborrar

Tots els usuaris d'Smartlock

SMARTLOCK TARGETES

Hola, David (Tancar sessió)

Modificar targeta

Valor: Activada? IdUsuari:

Modificat

Usuari propietari

| IdUsuari | Gener | Nom | Cognoms | Doble autenticació? | Password | Rol | Actiu | Usuari | Contrasenya |
|----------|-------|-------|--------------|---------------------|----------|-----|-------|--------|-------------|
| 1 | Home | David | Bassa Romera | 1 | 1234 | 1 | 1 | UOC | 2023 |

Total usuaris: 1

Historial d'accessos de la targeta

SMARTLOCK HISTORIAL de l'usuari

Hola, David (Tancar sessió)

Targetes de l'usuari

| IdTargeta | Valor | Estat | Usuari |
|-----------|------------------------|-------|--------|
| 1 | 0x65, 0xE2, 0xA4, 0x43 | 1 | 1 |
| 3 | [13 20 18 ac] | 1 | 1 |

Total registres: 2

Historial de l'usuari

| IdLOG | Descripció | Data | Nom Usuari | Targeta | Rol | IdUsuari |
|-------|---------------------------------|---------------------|---------------------|----------------|-----|----------|
| 233 | Es demana la doble autenticació | 2023-06-11 15:22:10 | Bassa Romera, David | [13 20 18 ac] | 1 | 1 |
| 232 | Es demana la doble autenticació | 2023-06-11 15:18:45 | Bassa Romera, David | [13 20 18 ac] | 1 | 1 |
| 230 | Es demana la doble autenticació | 2023-06-08 19:54:09 | Bassa Romera, David | [13 20 18 ac] | 1 | 1 |
| 228 | Es demana la doble autenticació | 2023-06-08 19:53:15 | Bassa Romera, David | [13 20 18 ac] | 1 | 1 |
| 226 | Es demana la doble autenticació | 2023-06-08 19:44:27 | Bassa Romera, David | [13 20 18 ac] | 1 | 1 |
| 224 | Es demana la doble autenticació | 2023-06-08 19:44:00 | Bassa Romera, David | [13 20 18 ac] | 1 | 1 |

II·lustració 45 Captures de pantalla panell administrador

5. Conclusions i treballs futurs

5.1 Conclusions

Aquest, ha estat un projecte en el qual he après molt sobre l'ecosistema d'Arduino. Vaig voler buscar un projecte en l'àmbit de la domòtica, donat que en casa tinc muntat un servidor domòtic open source i és un tema que m'agrada.

Quan va sorgir la idea de fer un pany intel·ligent, inicialment, vaig tenir les idees molt clares sobre el que volia que fes, quines limitacions tenien els productes de la competència, quin potencial podia tenir, etc. Però segons vaig anar definint i donant forma al projecte, tot el que tenia tan clar inicialment, es va començar a difuminar, i van sorgir molts dubtes, com ara; com podria ser més segur, que passaria si..., en cas que passi això, com ho hauria d'afrontar Smartlock, soluciona per complet tots els problemes que pot tenir la gent?, etc.

Finalment, després de buscar molta informació, analitzar en profunditat a la competència i sobretot, explicar el projecte a amics i familiars, vaig poder trobar resposta a totes aquestes preguntes que m'havien sorgit, i vaig veure que aquest, era un tipus de producte que a la gent li semblava molt interessant.

Aquest feedback rebut per la gent va ser clau en molts aspectes del projecte, i juntament, amb tota la informació trobada per internet, vaig poder definir tots els objectius i requisits que el projecte havia de complir, els quals, no vaig haver de modificar en el transcurs de les diferents fases, gràcies a aquest anàlisi en detall i feedback dels coneguts. Aquests requisits es van complir, a excepció de la implementació d'un mòdul de targeta SD, que inicialment, ja tenia dubtes sobre si tindria temps per desenvolupar-ho i aconseguir el material, ja que el projecte era extens.

Respecte a la planificació, la part de connexió d'Arduino amb el servidor (xip ESP8266) em va costar molt més del que m'esperava, i li vaig dedicar moltes més hores de les planificades, així que vaig haver de dedicar hores extres per les nits, per evitar desviar-me massa de la planificació. Ara, una vegada està tot desenvolupat, em sembla que la comunicació entre Arduino i l'ESP8266 és molt fàcil d'assolir, però alhora, trobo que està molt limitada, ja que només es pot passar text, i aquest és un fet que no m'esperava.

Finalment, respecte als objectius de la CCEG que s'havien mencionat en el punt 1.3, aquest, no s'han vist afectats al finalitzar el projecte i compleixen amb el que s'havia mencionat en aquest punt. No obstant això, m'ha sorprès el consum tan baix que té el sistema, tenint en compte que té connectats diversos components, i aquest baix consum ajuda a complir ODS 7.3.

5.2 Treballs a futur

Una vegada està desenvolupat tot, i tenint en compte que el mòdul de targeta SD (funcionalitat opcional) no el vaig poder desenvolupar, hi ha una sèrie d'eixos de millora que trobo que serien interessants a futur, com són:

- Desenvolupar el mòdul de targeta SD, on a part de guardar el valor dels identificadors RFID, per utilitzar-los en el mode offline, també guardi un log (watchlog) de tot el que passa en el sistema, amb la finalitat de corregir possibles errors, i alhora que quedi constància dels accessos a l'habitatge quan el sistema operi en la modalitat OFFLINE.
- Realitzar tota una sèrie de canvis, perquè el sistema sigui Plug and Play, com per exemple, crear un assistent (App), que permeti configurar Smartlock perquè es connecti a la xarxa wifi introduint el nom de la xarxa i la contrasenya, que informi l'usuari quina IP té, es puguin modificar paràmetres com ara la contrasenya per defecte, etc.
- Dotar de més seguretat al sistema:
 - Afegint una connexió xifrada tant en el servidor com en la connexió wifi.
 - Separar Smartlock en dues parts, d'una banda, en la part exterior de la porta tingui el teclat, lector rfid i la pantalla, i en la part interior de l'habitatge la placa, el xip, la targeta SD, etc, per evitar que es pugui manipular el pany des de l'exterior.
 - Modificar la forma en la qual s'obre Smartlock de manera online (ip/Open), donat que no és del tot segura.

- Desenvolupar la línia d'Smartlock per a empreses, que es va mencionar anteriorment, on a part d'utilitzar-se coma sistema d'obertura de portes, també faria el control horari dels treballadors d'una empresa.

6. Glossari

- **Wi-Fi** és una tecnologia que permet la interconnexió sense fil de dispositius electrònics com ara portàtils, tablets, Smartphones, etc.
- **RFID** (*Radio Frequency Identification*), és un sistema d'emmagatzematge i recuperació de dades remotes que usa dispositius denominats etiquetes, targetes o transponedors RFID. El propòsit fonamental de la tecnologia RFID és transmetre la identitat d'un objecte mitjançant ones de ràdio
- **CCEG** (Commission expert group on climate change), és una comissió d'experts sobre política de canvi climàtic, on el seu objectiu principal es ajudar al DG Clima en la preparació de propostes legislatives i iniciatives polítiques, així com servir com a fòrum de coordinació i intercanvi de punts de vista en l'àrea política del clima en els estats membres.
- **ODS** (*Objetivos de Desarrollo Sostenible*), són els objectius "goals" que s'ha proposat aconseguir la CCEG envers el canvi climàtic.
- **LCD** (*liquid-crystal display*) és una pantalla prima i plana formada per un nombre de píxels en color o monocroms col·locats davant d'una font de llum o reflectora. Sovint s'utilitza en dispositius electrònics de piles, ja que utilitza quantitats molt petites d'energia elèctrica
- **BBDD** Abreviatura de base de dades. Conjunt de dades comunes emmagatzemades segons les necessitats de l'empresa.
- **PHP** (*Hypertext Preprocessor*) és un llenguatge de programació de codi obert interpretat del costat del servidor i d'ús general que s'adapta especialment al desenvolupament web.
- **MySQL** (*My (cofundador Michael Wideniu's) SQL (Structured Query Language)*) és un sistema de gestió de bases de dades relacional desenvolupat sota llicència dual: Llicència pública general/Llicència comercial per Oracle Corporation i és considerada com la base de dades de codi obert més popular del món, i una de les més populars en general al costat de Oracle i Microsoft SQL Server, tot per a entorns de desenvolupament web.
- **RGB** (*red, green and blue*) es refereix a un sistema que representa els colors utilitzats en una pantalla de visualització digital. El vermell, el verd i el blau es poden combinar en diverses proporcions per a obtenir qualsevol color de l'espectre visible.
- **IDE Arduino** (*Integrated Development Environment Arduino*) és una aplicació multiplataforma que està escrita en el llenguatge de programació Java. S'utilitza per a escriure i carregar programes en plaques compatibles amb Arduino, però també, amb l'ajuda de nuclis de tercers, es pot usar amb plaques de desenvolupament d'altres proveïdors.
- **Zigbee** (*Zonal Intercommunication Global-standard*) es un conjunt de protocols d'alt nivell de comunicació sense fil, per a la seva utilització amb radiodifusió digital de baix consum, basada en l'estàndard IEEE 802.15.4 de xarxes sense fils d'àrea personal (wireless personal area network, WPAN). El seu objectiu són les aplicacions que requereixen comunicacions segures amb baixa taxa d'enviament de dades i maximització de la vida útil de les seves bateries.
- **Bluetooth** és una especificació industrial per a xarxes sense fils d'àrea personal (WPAN) creat per Bluetooth Special Interest Group, Inc. que possibilita la transmissió de veu i dades entre diferents dispositius mitjançant un enllaç per radiofreqüència en la banda ISM dels 2.4 GHz.
- **GND** (*ground*) són els pins a terra de la placa Arduino, el negatiu.
- **VCC** (*Voltage Common Collector*) és el voltatge de subministrament d'alimentació del microcontrolador, el normal d'una placa Arduino convencional és de 5 volts,
- **PWM** (*Pulse Width Modulation*) és un tipus de senyal de voltatge utilitzada per a enviar informació o per a modificar la quantitat d'energia que s'envia a una càrrega
- **Ω** (*Ohm*) és la unitat de resistència elèctrica. La llei d'Ohm, en electricitat, vincula la diferència de potencial, intensitat de corrent i resistència elèctrica.
- **Port Serial** és el que s'utilitza per a la comunicació entre la placa i l'ordinador o altres components de l'ecosistema Arduino, aquest serial permet enviar i rebre missatges, control de la depuració i el control de la placa. Totes les plaques tenen almenys un port sèrie (a aquests se'ls coneix també com UART o USART).

7. Bibliografía

- Projecthub [En línea] [Data de consulta: 04/03/2023]
<https://projecthub.arduino.cc/KABAKA/e93d5bf7-7556-4890-8283-1903c956890e>
https://projecthub.arduino.cc/?ref=search&ref_id=numeric&offset=3
- Elpais [En línea] [Data de consulta: 12/03/2023]
<https://elpais.com/escaparate/2022-12-03/probamos-cuatro-cerraduras-inteligentes-que-se-pueden-manejar-con-el-movil-y-elegimos-la-mejor.html>
- Xataka [En línea] [Data de consulta: 12/03/2023]
<https://www.xataka.com/seleccion/guia-compra-cerraduras-inteligentes-instalacion-conectividad-preguntas-frecuentes-recomendaciones-para-acertar>
<https://www.xataka.com/analisis/abrir-y-cerrar-la-puerta-con-nuestro-smartphone-probamos-la-cerradura-conectada-tesa-entr>
- Nuki [En línea] [Data de consulta: 12/03/2023]
<https://nuki.io/es/smart-lock-pro/>
<https://nuki.io/es/keypad/>
- Ipcenter [En línea] [Data de consulta: 12/03/2023]
<https://www.ipcenter.es/AL20DB.html>
- XatakaHome [En línea] [Data de consulta: 12/03/2023]
<https://www.xatakahome.com/seguridad-en-el-hogar/xiaomi-lanza-una-cerradura-que-no-usa-nuestro-movil-como-llave-y-aun-asi-se-trata-de-un-bombin-inteligente>
- Adslzone [En línea] [Data de consulta: 12/03/2023]
<https://www.adslzone.net/2017/12/13/xiaomi-vima-smart-lock-bombin-inteligente/>
- Luisllamas [En línea] [Data de consulta: 18/03/2023]
<https://www.luisllamas.es/alimentar-arduino-baterias/>
<https://www.luisllamas.es/esquema-de-patillaje-de-arduino-pinout/>
<https://www.luisllamas.es/arduino-rfid-mifare-rc522/>
<https://www.luisllamas.es/arduino-teclado-matricial/>
- Energia.Roams [En línea] [Data de consulta: 18/03/2023]
<https://energia.roams.es/luz/corte-luz/>
- Educaparaelcambio [En línea] [Data de consulta: 17/03/2023]
<https://educaparaelcambio.com/arduino/reto-4-liquid-crystal-display-pantalla-lcd/>
- Electrogeekshop [En línea] [Data de consulta: 17/03/2023]
<https://www.electrogeekshop.com/tutorial-modulo-lector-rfid-rc522/>
- Cursoarduinomega [En línea] [Data de consulta: 17/03/2023]
<http://cursoarduinomega.blogspot.com/2015/03/acceso-por-contrasena-con-el-keypad.html>
- Ingenieriaelectronica [En línea] [Data de consulta: 17/03/2023]
<https://ingenieriaelectronica.org/clave-de-seguridad-con-teclado-matricial-4x4-y-arduino/>
- Robots-argentina[En línea] [Data de consulta: 18/03/2023]
<https://robots-argentina.com.ar/didactica/comparacion-entre-placas-arduino/>
- Arduino.cl[En línea] [Data de consulta: 20/03/2023]
<https://arduino.cl/producto/arduino-mega-2560/#:~:text=El%20Arduino%20Mega%20tiene%2054,ICSP%20y%20bot%C3%B3n%20de%20reset.>
- Wiki.Eurek[En línea] [Data de consulta: 19/03/2023]
<https://wiki.eurek.org/arduino-mega-esp8266/>
- Arduinofacil[En línea] [Data de consulta: 27/03/2023]
<https://arduinofacil.com/la-resistencia/>

- Arduinogetstarted[En línea] [Data de consulta: 27/03/2023]
https://arduinogetstarted.com/tutorials/arduino-piezo-buzzer?utm_content=cmp-true
- Wikipedia[En línea] [Data de consulta: 29/03/2023]
https://es.wikipedia.org/wiki/Resistencia_el%C3%A9ctrica
- LucidApp[En línea] [Data de consulta: 28/03/2023]
<https://lucid.app/>
- Programarfácil [En línea] [Data de consulta: 04/04/2023]
<https://programarfácil.com/esp8266/esp8266-deep-sleep-nodemcu-wemos-d1-mini/>
<https://programarfácil.com/blog/arduino-blog/buzzer-con-arduino-zumbador/>
[https://programarfácil.com/blog/arduino-blog/el-potenciometro-y-arduino/#Potenciometro de variacion lineal](https://programarfácil.com/blog/arduino-blog/el-potenciometro-y-arduino/#Potenciometro_de_variacion_lineal)
<https://programarfácil.com/esp8266/esp8266-deep-sleep-nodemcu-wemos-d1-mini/>
<https://programarfácil.com/podcast/esp8266-wifi-coste-arduino/>
<https://programarfácil.com/blog/arduino-blog/texto-en-movimiento-en-un-lcd-con-arduino/>
- Flexbot[En línea] [Data de consulta: 05/04/2023]
<https://www.flexbot.es/tutorial-tarjetas-rfid/>
- Electrogeekshop [En línea] [Data de consulta: 05/04/2023]
<https://www.electrogeekshop.com/tutorial-modulo-lector-rfid-rc522/>
- Ingenieriaelectronica [En línea] [Data de consulta: 07/04/2023]
<https://ingenieriaelectronica.org/clave-de-seguridad-con-teclado-matricial-4x4-y-arduino/>
- Aranacorp[En línea] [Data de consulta: 10/04/2023]
<https://www.aranacorp.com/es/uso-de-un-pantalla-16x2lcd-con-arduino/>
- Blog.330ohms [En línea] [Data de consulta: 12/04/2023]
<https://blog.330ohms.com/2020/06/12/como-conectar-un-micro-servo-sg90-posicion-180-grados-a-arduino/>
- Desarrolloweb[En línea] [Data de consulta: 20/04/2023]
<https://desarrolloweb.com/articulos/1054.php#numericos>
- EsTalent [En línea] [Data de consulta: 25/04/2023]
<https://es.talent.com/salary?job=programador#:~:text=El%20salario%20programador%20promedio%20en,hasta%20%E2%82%AC%2038.000%20al%20a%C3%B1o.>
- BlogKenjo [En línea] [Data de consulta: 25/04/2023]
<https://blog.kenjo.io/es/cual-es-el-coste-de-la-empresa-al-contratar-a-un-trabajador>
- Tecnoblog [En línea] [Data de consulta: 25/04/2023]
<https://www.tecnoblog.guru/2021/07/como-funciona-la-tecnologia-wifi-ventajas-inconvenientes.html#:~:text=La%20ventaja%20m%C3%A1s%20importante%20del,Bluetooth%20y%20VoIP%2C%20entre%20otros.>
- ResourcesAltium [En línea] [Data de consulta: 25/04/2023]
<https://resources.altium.com/es/p/advantages-and-disadvantages-active-and-passive-rfid-technologies>
- Gigatecno [En línea] [Data de consulta: 26/04/2023]
http://gigatecno.blogspot.com/2013/04/ventajas-y-desventajas-de-las-pantallas_20.html
- Hostinger [En línea] [Data de consulta: 26/04/2023]
https://www.hostinger.es/hosting-web?utm_medium=affiliate&utm_source=aff1015&utm_campaign=14&session=1029a856036b783890051ad02563a6
- 000webhost [En línea] [Data de consulta: 20/04/2023]
<https://co.000webhost.com/>

- MySQL [En línia] [Data de consulta: 18/04/2023]
<https://www.mysql.com/products/workbench/>

8. Referències

- **[1]** Servisatel [En línia] [Data consulta 04/03/2023] Article que parla sobre els perill que té perdre les Claus de casa -
<https://servisatel.com/D/post/peligros-de-perder-llaves-de-la-vivienda-y-soluciones/>
Lavozdegalicia [En línia] [Data de consulta: 04/03/2023]. Article que demostra que una persona arriba a perdre fins a 6 mesos de vida buscant les Claus de casa
<https://www.lavozdegalicia.es/noticia/sociedad/2019/09/13/sabias-pierdes-seis-meses-vida-buscando-llaves-casa/00031568372761377617584.htm>
Apartmenttherapy [En línia] [Data de consulta: 04/03/2023]. Article en anglès que menciona quins són els principals objectes que perd la gent
<https://www.apartmenttherapy.com/study-reveals-the-most-common-items-that-go-missing-at-home-246906>
- **[2]** Efeverde[En línia] [Data de consulta: 20/04/2023] Article que analitza les dades de reciclatge de piles de l'any 2020 - <https://efeverde.com/erp-espana-toneladas-pilas-baterias-2020/>
- **[3]** Projecthub [En línia] [Data de consulta: 04/03/2023]. Projecte realitzat amb Arduino que controla l'accés d'una porta amb contrasenya -
https://projecthub.arduino.cc/SurtrTech/fb3de2bc-cf5c-4288-b5da-5965d97fd668?ref=similar&ref_id=455919&offset=2
- **[4]** Projecthub [En línia] [Data de consulta: 04/03/2023]. Projecte realitzat amb Arduino que controla una alarma amb identificadors RFID -
https://projecthub.arduino.cc/team_chkr/beff9981-1736-4a68-abfa-7e8314810357
- **[5]** Projecthub [En línia] [Data de consulta: 04/03/2023]. Projecte d'Arduino que controla un RFID i guarda la informació en una base de dades -
<https://projecthub.arduino.cc/anshulpareek/online-attendance-system-without-ethernet-708764>
- **[6]** Alguns productes es venen en quantitats grans, és a dir, que amb un paquet, sobraran unitats que es poden utilitzar per a altres SmartLock, amb la qual cosa baixaria el pressupost de cada unitat.
- **[7]** Els articles opcionals són per afegir més funcionalitats de les bàsiques al sistema Smartlock com són el funcionament del sistema en cas de tall elèctric i en el cas d'un problema de connexió amb el servidor.
- **[8]** Cost de la Seguretat Social = (Sou brut x 0,2360) + (Sou brut x 0,055) + (Sou brut x 0,0020) + (Sou brut x 0,0070).
- **[9]** Calculat amb les tarifes de l'empresa Hostinger com a referència.
- **[10]** Ontsi[En línia] [Data de consulta: 10/03/2023]. Document de l'Ontsi realitzat l'any 2022 amb dades sobre l'ús de la tecnologia en les llars Espanyoles - https://www.ontsi.es/sites/ontsi/files/2022-02/usotecnologiahogares_2022_1.pdf
- **[11]** Elindependiente [En línia] [Data de consulta: 14/05/2023]. Article que analitza la xarxa d'internet a Espanya.
<https://www.elindependiente.com/economia/2021/11/11/mas-de-55-millones-de-personas-aun-no-tienen-acceso-a-internet-en-espana/#:~:text=Espa%C3%B1a%2C%20a%20pesar%20de%20ser,seg%C3%BAn%20cifras%20ofrecidas%20por%20Eurona>
- **[12]** Noticias.Habitacalia [En línia] [Data de consulta: 15/05/2023]. Article de la pàgina Habitacalia que analitza el nombre d'habitatges que tenen instal·lat un sistema domòtic, així com les tendències del mercat -
<https://noticias.habitacalia.com/un-48-de-espanoles-cuenta-con-sistemas-de-domotica-en-su-vivienda/>

9. Annexos

9.1 Codi Arduino Smartlock

```
//Es carreguen les llibreries necessàries
#include <SPI.h> //S'importa la llibreria SPI (lector RFID)
#include <MFRC522.h> //S'importa la llibreria MFRC522 del lector
#include <Servo.h> //S'importa la llibreria del Servomotor
#include <LiquidCrystal.h> //S'importa la llibreria de la pantalla
#include <Keypad.h> // S'importa la llibreria del keypad numeric
/*****CONSTANTS*****/
const int redLEDPin = 10; //S'assigna el pin del LED Vermell del LED RGB
const int greenLEDPin = 9; //S'assigna el pin del LED Verd del LED RGB
const int blueLEDPin = 11; //S'assigna el pin del LED Blau del LED RGB
const int BuzzerPin = 13; //S'assigna el pin del Buzzer
//Pins lector RFID RC522
const int RFID_RST_Pin = 5; //S'assigna el pin del Reset del RFID RC522
const int RFID_SS_PIN = 53; // S'assigna el pin del SDA del RFID RC522
//Pins i posició del Servomotor
const int Servo_Open = 120; //S'assigna la posició (angle) del servo quan la porta està oberta
const int Servo_Close = 32; //S'assigna la posició (angle) del servo quan la porta està tancada
const int ServoPin = 8; //S'assigna el pin del Buzzer
//Pins, Nombre de columnes i files del teclat matriu
const byte ROWS = 4; //Es defineix el nombre de files del keypad
const byte COLS = 4; //Es defineix el nombre de columnes del Keypad
byte colPins[4] = {43,45,47,49}; //S'assigna els 4 pins de les columnes
byte rowPins[4] = {35,37,39,41}; //S'assigna els 4 pins de les files
const String code_user = "0000"; //Constant que guarda el valor del pin de doble autenticació de l'usuari
/*****VARIABLES*****/
byte TarjetaActual[4]; //S'emmagatzemara la targeta llegida actual
byte Usuari1[4] = {0x65, 0xE2, 0xAA, 0x43}; //Targeta de l'usuari 1
byte Usuari2[4] = {0xF5, 0xE8, 0xE2, 0x52}; //Targeta de l'usuari 2
LiquidCrystal lcd (30,3,2,4,31,33); //Es crea l'objecte pantalla
MFRC522 mfrc522(RFID_SS_PIN, RFID_RST_Pin); //Objecte per al lector RFID RC522
Servo myservo; //Es crea l'objecte servo
//Es crea un array amb tots els valors del teclat matriu de 4x4
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS ); //Es declara l'objecte keypad per poder utilitzar
el teclat numèric
int comptador=0; //Variable que conté el nombre de tecles premudes per l'usuari
char password_1[]="1234"; //Variable amb la contrasenya de l'usuari per
char key_admin_access[]="11111"; //Variable amb la clau d'accés de admin
int intents; //Variable guarda el nombre d'intents de la doble autenticació
String inString; //Variable on es guarda el text que es rep pel port Serial3
String targeta; //Variable on es guarda el valor de la targeta llegida
```

```

/*****VARIABLES DE MODALITAT*****/
bool mode_admin = false; //Variable booleana mode administrador
bool mode_control_acces = false; //Variable booleana Doble autenticació.
bool mode_nou_pass = false; //Variable booleana mostrar el valor d'una targeta
bool mode_offline = false; //Variable booleana estat connexió
/*****PROGRAMA*****/
//Aquesta funció només s'executa una vegada (a l'arrencar Arduino)
void setup() {
  Serial.begin(9600); //S'inicialitza el port Serial d'Arduino
  Serial3.begin(9600); //S'inicialitza el port serial 3 (ESP8266)
  SPI.begin(); //S'inicialitza el Bus SPI
  mfrc522.PCD_Init(); //S'inicialitza el RFID RC522
  myservo.attach(ServoPin); //S'inicialitza el servoMotor
  lcd.begin(16,2); //S'inicialitza la pantalla LCD amb 16x2
  //S'informa a Arduino que els pins del LED RGB són de sortida.
  pinMode(redLEDPin, OUTPUT);
  pinMode(greenLEDPin, OUTPUT);
  pinMode(blueLEDPin, OUTPUT);
  //S'informa a Arduino que el pin del Buzzer (altaveu) és de sortida.
  pinMode(BuzzerPin, OUTPUT);
  ini(); //Es crida a la funció ini
  //Aquesta funció s'executa en bucle fins a l'infinit i cridarà a unes funcions o d'altres en funció de la modalitat en la qual es
  trobi el sistema
  void loop() {
    read_rfid(); //Es crida a la funció encarregada de fer la lectura de la targeta RFID
    //Si l'usuari prem la combinació de tecles definida a la variable key_admin_access llavors entrarà en el mode admin sinó
    estarà en el mode d'accés per RFID
    if (mode_admin == false && mode_nou_pass == false && mode_control_acces == false){ //Si el mode admin no està actiu,
    el mode nou pass no està activat i el control d'accés per contrasenya tampoc llavors el sistema verificarà quines tecles ha
    premut l'usuari
      control_admin_acces(key_admin_access); //Es crida a la funció control_admin_acces per verificar si l'usuari ha premut la
    combinació correcta de mode admin (6 dígits key_admin_access[])
    }if (mode_admin == true){ // Si el sistema es troba en mode_admin, es mostrarà el menu admin i les opcions que té l'usuari,
    !=> Llegir RFID i 2=>sortir
      menu_admin(); //Es crida a la funció encarregada de mostrar el menu_admin
    }if (mode_control_acces == true){ //Si el sistema es troba en el mode de comprovació de doble autenticació entrarà en el
    if
      if (intents < 3){ //L'usuari té 3 intents per col·locar bé la contrasenya, si no el sistema haurà de tornar a llegir la targeta
      control_code(code_user); //Es crida a la funció encarregada de verificar si la clau d'accés introduïda per l'usuari es la
    correcta i es passa com a paràmetre la contrasenya correcte.
      }else {ini(); mode_control_acces == false;} } //Si l'usuari a esgotat els 3 intents, el sistema és reseteja, surt de la
    modalitat doble autenticació
/*****FUNCIONS*****/
//Aquesta funció s'executa cada cop que hi ha un esdeveniment en el port serial3, és a dir, quan ESP8266 envia informació a
l'Arduino Mega2560
void serialEvent3() {
  while (Serial3.available()) { //Es quedarà dintre del bucle mentre el port Serial3 estigui actiu
    char inChar = Serial3.read(); //Lectura de dades del Serial3
    Serial.write(inChar); // S'escriu les dades rebudes pel port serial
    inString += inChar; //Es guarden tots els valor rebuts pel Serial 3
  }
}

```



```

// Es busca si hi ha el text [ON], [OFF], [OFFLINE], [ONLINE] o un text entre claudàtors [] (contrasenya TwoFactor)
if (inChar == ']') {
  if (inString.indexOf("[ON]")>0) { //Valor rebut pel serial és [ON]      acces_correct(); //Es crida a la funció encarregada
d'obrir la porta
  }else if (inString.indexOf("[OFF]")>0) { //Valor rebut pel serial és [OFF] entra en el if
  acces_denied(); //Es crida a la funció encarregada d'informar a l'usuari que no té accés
  }else if (inString.indexOf("[OFFLINE]")>0) { //Si el valor rebut pel serial és [OFFLINE] entra en el if
  mode_offline = true; //Es posa la constant mode_offline en true per notificar al sistema que s'ha perdut la connexió i
funcionerà a nivell local
  }else if (inString.indexOf("[ONLINE]")>0) { //Si el valor rebut pel serial és [ONLINE] entra en el if
  mode_offline = false; //Es posa la constant mode_offline en false per notificar al sistema que hi ha connexió i funcionarà
en remot (servidor)
  }else {
  String code; //Variable on es guardarà el codi rebut pel port Serial3
  code = inString.substring(inString.length()-5, inString.length()-1); //Es guarden els quatre dígits eliminant els claudàtors
d'inici i final
  code_user = code; //S'assigna el valor de la variable code a la variable global code_user
  mode_control_acces = true; //Es col·loca el sistema en la modalitat de doble autenticació
  lcd.clear(); //S'esborra el contingut de text de la pantalla
  lcd.print("Contrasenya?"); //S'escriu text per pantalla
  }inString = ""; } } //Es reseteja el valor de inString

```

//Aquesta funció s'encarrega de fer una sèrie d'accions si l'usuari té accés a l'habitatge posar el LED en color verd, fer sonar l'altaveu, moure el servomotor en posició oberta enviar un text a la pantalla, etc.

```

void acces_correct(){
  analogWrite (redLEDPin, 0); //Valor de sortida 0 al pin del LED vermell
  analogWrite (greenLEDPin, 255); //Valor de sortida 255 al pin del LED verd
  analogWrite (blueLEDPin, 0); //Valor de sortida 0 al pin del LED Blau
  delay(100); //Es para Arduino durant 100ms
  tone(BuzzerPin, 2000, 1000); //S'emet un altre so més greu per donar a entendre que s'accepta l'accès
  myservo.write(Servo_Open); //Es col·loca el servoMotor en posició oberta
  lcd.setCursor(0,1); //Es situa el cursor de la pantalla en la segona fila
  lcd.print("Acces Concedit"); //Es mostra el text d'accés concedit
  delay(3000); //Es para Arduino durant 3000ms
  noTone(BuzzerPin); //S'apaga el so del so del buzzer
  lcd.clear(); //S'esborra tot el contingut de text de la pantalla
  if (mode_offline == false){ //Si el sistema està en mode online entra  lcd.print("Hola!"); //S'escriu un text per la pantalla
  }else { //Si el sistema està en el mode offline entra en el if
  lcd.print("Hola! - OFFLINE");} //S'escriu un text per la pantalla
  myservo.write(Servo_Close); //Es col·loca el servoMotor en posició tancat
  analogWrite (greenLEDPin, 0); //S'apaga el LED verd (valor sortida 0).
  mode_control_acces = false;} //S'assigna el valor de mode_acontrol_acces en false perquè l'usuari ja ha entrat a l'habitatge

```

//Aquesta funció s'encarrega de fer una sèrie d'accions si l'usuari no té accés al sistema posar el LED en color vermell, fer sonar l'altaveu, posar el Servomotor en posició tancat, i enviar un text per pantalla.

```

void acces_denied(){
  analogWrite (redLEDPin, 255); //Valor de sortida 255 al pin del LED vermell
  analogWrite (greenLEDPin, 0); //Valor de sortida 0 al pin del LED verd
  analogWrite (blueLEDPin, 0); //Valor de sortida 0 al pin del LED blau
  myservo.write(Servo_Close); //Es col·loca el servoMotor en posició tancat

```

```

delay(100); //S'atura arduino durant 100ms
analogWrite (redLEDPin, 0); // S'apaga el LED de color vermell
lcd.setCursor(0,1); //Es situa el cursor de la pantalla en la segona fila la
lcd.print("Acces Denegat");//Es mostra el text d'accés denegat per pantalla
tone(BuzzerPin, 1000, 200); //S'emeta un altre so més agut per donar a entendre que es denega l'accés
delay(1000); //S'atura Arduino durant 1000ms
noTone(BuzzerPin); //S'apaga el Buzzer (no emet so)
delay(3000); //S'atura Arduino durant 3000ms
lcd.clear(); //S'esborra tot el contingut de text de la pantalla
if (mode_offline == false){ //Si el sistema està en mode online entra
  lcd.print("Hola!"); //S'escriu un text per la pantalla
}else { //Si el sistema està en el mode offline entra en el if
  lcd.print("Hola! - OFFLINE"); } //S'escriu un text per la pantalla
mode_control_acces = false; } //S'assigna el valor de mode_acontrol_acces en false perquè l'usuari no té accés

```

//Aquesta funció s'encarrega de fer una sèrie d'accions si l'usuari ha introduït malament la contrasenya de la doble autenticació com posar el LED en color vermell, fer sonar l'altaveu, posar el Servomotor en posició tancat, i enviar un text per pantalla.

```

void acces_denied_intent(){
  analogWrite (redLEDPin, 255); //Valor de sortida 255 al pin del LED vermell
  analogWrite (greenLEDPin, 0); //Valor de sortida 0 al pin del LED verd
  analogWrite (blueLEDPin, 0); //Valor de sortida 0 al pin del LED blau
  myservo.write(Servo_Close); //Es col·loca el servoMotor en posició tancat
  delay(100); //S'atura Arduino durant 100ms
  analogWrite (redLEDPin, 0); // S'apaga el LED de color vermell  lcd.setCursor(0,1); //Es situa el cursor de la pantalla en la
segona fila
  lcd.print("Acces Denegat");//Es mostra el text d'accés denegat per pantalla
  tone(BuzzerPin, 1000, 200); //S'emeta un altre so més agut per donar a entendre que es denega l'accés
  delay(1000); //S'atura Arduino durant 1000ms
  noTone(BuzzerPin); //S'apaga el Boozer (no emet so)
  delay(3000); //S'atura Arduino durant 3000ms
  lcd.clear(); //S'esborra tot el contingut de text de la pantalla
  lcd.print("Contrasenya?"); //S'escriu un text per la pantalla
}

```

//Aquesta funció s'encarrega de preparar el sistema quan arrenca, on no tindrà so, el LED RGB estarà apagat, el servomotor en posició tancat i la pantalla mostra un missatge

```

void ini(){
  analogWrite (redLEDPin, 0); //Valor de sortida 0 al pin del LED vermell
  analogWrite (greenLEDPin, 0); //Valor de sortida 0 al pin del LED verd
  analogWrite (blueLEDPin, 0); //Valor de sortida 0 al pin del LED blau
  noTone(BuzzerPin); //S'apaga el buzzer perquè no emeti cap so
  myservo.write(Servo_Close); //Servomotor en posició tancat
  lcd.clear(); //S'esborra tot el contingut de text de la pantalla
  if (mode_offline == false){ //Si el sistema està en mode online entra  lcd.print("Hola!"); //S'escriu un text per la pantalla
  }else { //Si el sistema està en el mode offline entra en el if
    lcd.print("Hola! - OFFLINE"); } //S'escriu un text per la pantalla
  intents = 0; //S'inicialitza la variable intents a zero
}

```

//Aquesta funció s'encarrega de llegir les targetes del lector RFID, si llegeix una tarjeta emet un so confirmant la lectura, i s'emmagatzema en la variable TarjetaActual el valor de la tarjeta

```

void read_rfid(){

```

```

if ( mfr522.PICC_IsNewCardPresent()){ //Si el lector RFID llegeix una targeta retorna un true
    tone(BuzzerPin, 3000, 200); //Al llegir una targeta s'emet un so confirmant la lectura correcte
    if ( mfr522.PICC_ReadCardSerial()){ //Es selecciona una targeta
        targeta = "["; //S'assigna a la variable targeta el valor [ a l'inici
        for (byte i = 0; i < mfr522.uid.size; i++) { //Es recorren tots els bytes de la targeta
            TarjetaActual[i]=mfr522.uid.uidByte[i]; //Es guarda el valor de la targeta en la variable TarjetaActual
            //Es guarden els valors de la targeta llegida          targeta.concat(String(mfr522.uid.uidByte[i] < 0x10
? " 0" : " "));
            targeta.concat(String(mfr522.uid.uidByte[i], HEX));}
        targeta.concat(String("]")); //Al final de l'string s'afegeix un ]
        mfr522.PICC_HaltA(); // Es tanca la lectura de la targeta
        if (mode_admin == false && mode_nou_pass == false && mode_offline == false){ //Si el sistema no està en mode
administrador i té connexió amb el servidor llavors enviarà el valor de la targeta pel serial3 (xip ESP8266)
            Serial3.println(targeta); //S'escriu pel port Serial 3 el valor de targeta (es passa el valor de la targeta al xip
ESP8266)
        }if (mode_admin == false && mode_nou_pass == true){ //Si el sistema està en mode admin i nou pass (nova
targeta)
            new_pass (targeta); //Crida a la funció new_pass encarregada de mostrar el valor de la targeta per pantalla
        }if (mode_admin == false && mode_nou_pass == false && mode_offline == true){ //Si el sistema no està en mode
administrador i no té connexió amb el servidor, farà la comprovació en local
            acces_control(); } } } //crida a la funció encarregada de verificar l'accés de la targeta a nivell local

//Aquesta funció s'encarregarà de comparar dues targetes (la targeta llegida amb la targeta emmagatzemada) si són iguals
retornarà true i si no falç
boolean compareArray(byte array1[],byte array2[]){
    //Es van comparant els dos arrays que rep per funció, si algun byte es diferent retorna false
    if(array1[0] != array2[0])return(false);
    if(array1[1] != array2[1])return(false);
    if(array1[2] != array2[2])return(false);
    if(array1[3] != array2[3])return(false);
    return(true); } // Si tots els bytes son iguals retornarà un true

//Aquesta funció s'encarrega de comprovar les credencials de la targeta, si té permisos d'accés llavors posa Smartlock en
la modalitat de doble autenticació si introdueix la contrasenya correcta donarà accés al sistema, per al contrari si no en té,
crida a la funció acces denied
void acces_control(){
    if(compareArray(TarjetaActual,Usuari)){ //Es crida a la funció encarregada de verificar si la targeta està al sistema, si
existeix llavors retornarà true i entrarà en el if
        lcd.clear(); //S'esborra tot el contingut de la pantalla
        lcd.print("Codi acces?"); //Es mostra un text per pantalla demanant la clau
        mode_control_acces =true; //Es posa el sistema en mode_control_acces, que és l'encarregat de comprovar si la clau
d'accés introduïda és correcte
        code_user = "0000"; //S'assigna el valor per defecte de la contrasenya de doble autenticació a 0000
    }else{ //Si la targeta no està registrada al sistema entra en else
        acces_denied(); } } //Es crida a la funció acces_denied

//Aquesta funció s'encarrega de verificar si la contrasenya que introdueix l'usuari és correcte o no, si és correcte obre el
pany intel·ligent i
//si no denega l'accés.
void control_code(String password){

```

```

char key = keypad.getKey(); //Variable on es guarda la tecla premuda
char clau[4]; //Variable on es guarden els 4 caràcters seguits que ha premut l'usuari
if (key != NO_KEY){ //Si es detecta que l'usuari ha premut alguna tecla entra
  clau[comptador]=key; // Es guarda caràcter a caràcter cada cop que es toca una tecla en la variable clau[]
  lcd.setCursor(comptador,1); //Se situa el cursor de la pantalla en la segona fila i en la posició igual al comptador
  lcd.print("*"); //S'escriu per la pantalla un * cada cop que l'usuari prem una tecla
  comptador=comptador+1; //S'augmenta la variable comptador per guardar cada caràcter en una nova posició fins arribar
a 4 tecles guardades
  if(comptador==4) { //Si l'usuari ja ha premut 4 tecles seguides es verificarà si la clau es correcte
    String claut_str = String (clau); //Es converteix a string i es guarda el valor de clau en la variable claut_str
    claut_str = claut_str.substring(0, 4); //S'assegura que en la variable clau_str només tindrà els 4 primers caràcters
    if(claut_str.compareTo(password) == 0){ //Es comparen les dues cadenes (password introduir per usuari vs password
correcte) si el valor és 0 vol dir que són iguals
      acces_correct(); //Es crida a la funció acces_correct
      comptador=0; // Es reseteja la variable a 0 per poder tornar a guardar 4 tecles
      intents = 0;
      Serial3.println("[OK2FACTOR]"); //S'escriu pel port Serial 3 per informar el servidor que l'usuari a introduït la
contrasenya correctament
    } else {acces_denied_intent();intents++; } //Es denega l'accés al sistema si la clau no és correcte
    comptador=0; } //Es reseteja la variable a 0 per poder tornar a guardar 4 tecles
  if (intents >=3){ //Si l'usuari ha realitzat més de tres intents erronis s'informarà a l'historial del servidor
    Serial3.println("[FAIL2FACTOR]"); } //S'informa a l'ESP8266 que l'usuari ha introduït malament la contrasenya

//Aquesta funció s'encarrega de verificar si l'usuari a introduït la clau d'accés d'administrador (6 díigits)
void control_admin_acces(char password[]){
  char key = keypad.getKey(); //Variable on es guarda la tecla premuda
  char clau[6]; //Variable on es guarden els 6 caràcters seguits que s'han premut per l'usuari
  if (key != NO_KEY){ //Si es detecta que l'usuari ha premut alguna tecla entra
    clau[comptador]=key; // Es guarda caràcter a caràcter cada cop que es toca una tecla en la variable clau[]
    comptador=comptador+1; //S'augmenta la variable comptador per guardar cada caràcter en una nova posició fins
arribar a 6 tecles guardades
    if(comptador==6) { //Si l'usuari ja ha premut 6 tecles seguides es verificarà si la clau és
correcte if(clau[0]==password[0]&&clau[1]==password[1]&&clau[2]==password[2]&&clau[3]==password[3]&&clau[4]=
=password[4]&&clau[5]==password[5]){ //Es verifica caràcter a caràcter si la variable clau (introduïda per l'usuari) és igual
a la variable on es guarda la clau d'accés de l'administrador (key_admin_access)
      lcd.clear(); //S'esborra tot el contingut de text de la pantalla
      lcd.print("Llegir RFID?"); //S'escriu un text per pantalla
      lcd.setCursor(0,1); //Es situa el cursor en la segona fila
      lcd.print("1 =>Si 2=>No"); //Es mostra per pantalla les dues opcions
      comptador=0; //Reseteja variable a 0 per poder guardar 6 tecles
      mode_admin=true; //Es posa la variable mode_admin en True, per fer que el sistema entri en el mode d'administrador
      tone(BuzzerPin, 1000, 200); //S'emetrà un altre só greu per donar a entendre que el password és correcte
      delay(1000); //S'atura Arduino durant 1000ms
      noTone(BuzzerPin); //S'apaga el so de l'altaveu
      menu_admin(); //Es crida a la funció menu_admin encarregada de mostrar el panell de lectura de targeta
    }comptador=0; } } //Es reseteja la variable a 0 per poder tornar a guardar 4 tecles

//Aquesta funció s'encarrega de mostrar el menú d'administrador per veure el valor d'una targeta
void menu_admin(){
  char key = keypad.getKey(); //Variable on es guarda la tecla premuda

```

```

if (key != NO_KEY){ //Si es detecta que l'usuari ha premut alguna tecla entra

if (key == '1'){ //Si l'usuari ha premut l'opció 1 podrà veure el valor de la targeta
  lcd.clear(); //S'esborra el contingut de text de la pantalla
  lcd.print("Valor RFID"); //S'escriu text per pantalla
  mode_nou_pass = true; //S'assigna a la variable mode_nou_pass valor true per poder llegir el valor d'una targeta
  mode_admin = false; //S'assigna el valor false al mode_admin (surts del panell d'administració)
}if (key == '2') { //Si l'usuari ha premut l'opció 2 sortirà del mode administrador
  lcd.clear(); //S'esborra tot el contingut de la pantalla
  lcd.print("Hola!"); //S'escriu text per pantalla
  mode_admin = false; } } //S'assigna valor false al mode_admin (surts del panell d'administració)

//Aquesta funció permet visualitzar el valor d'una targeta per pantalla, així l'usuari podrà saber el valor d'una targeta per donar-la d'alta al sistema. Rebrà el valor de la targeta pels paràmetres
void new_pass (String targeta){
  mode_admin = false; //S'assigna a la variable mode_admin el valor false, ja que es surt del mode administrador
  mode_nou_pass = false; //S'assigna a la variable mode_nou_pass el valor false, ja que es surt del mode new pass
  lcd.setCursor(0,1); //Es situa el cursor en la segona fila
  lcd.print(targeta); //S'escriu el text amb el valor de la targeta
  delay (5000); //S'atura Arduino durant 5 segons perquè l'usuari pugui apuntar el valor
  ini(); //Es crida a la funció ini() encarragada d'iniciar el sistema
}

```

9.2 Codi xip ESP8266

```

//Es carreguen les llibreries necessàries pel projecte
#include <ESP8266HTTPClient.h> //Llibreria que permet usar el protocol HTTP per a comunicar el controlador amb el servidor web
#include <ESP8266WiFiMulti.h> //Llibreria que permet connectar el controlador a la xarxa Wifi
#include <ESP8266WebServer.h> //Llibreria que s'utilitza per a simplificar la creació d'un servidor web utilitzant un mòdul ESP8266
#include <ESP8266Ping.h> //Llibreria que permet fer ping a un servidor web
/*****CONSTANTS*****/
//Es defineixen les credencials de la connexió wifi
const char* ssid = "G55_Wifi"; //Nom de la xarxa wifi
const char* password = "tearsandrain=89"; //Contrasenya de la xarxa Wifi
const char* host_server = "www.000webhost.com"; //Direcció del servidor
ESP8266WebServer server(80); //Es crea l'objecte server assignant el port 80
/*****VARIABLES*****/
String webPage = ""; //Variable String on es guardarà el contingut de la pàgina principal del servidor ESP8266
String webOpen = ""; //Variable String on es guardarà el contingut de la pàgina d'obertura de la porta del servidor ESP8266
String inString; //Variable on es guardarà el text rebut pel serial
int intents; //Variable que guardarà el nombre d'intents de connexió a la xarxa wifi
char data; //Variable que guardarà cada caràcter rebut pel port serial
String targeta; //Variable que guardarà el valor de la targeta rebuda pel port Serial des d'Arduino Mega
bool online; //Variable booleana que informarà el sistema si està connectat a una xarxa wifi
bool Webserver; //Variable booleana que informarà el sistema si el servidor està operatiu
/*****PROGRAMA*****/
//Aquesta funció només s'executa una vegada i s'encarrega de connectar ESP8266 a la xarxa wifi amb un màxim de 15 intents
void setup() {
  Serial.begin(9600); //S'inicialitza el port serial de comunicació entre ESP8266 i Arduino Mega2560 (serial3)
  intents = 0; //S'inicialitza la variable intents a zero
}

```

```

Webserver = true; //S'inicialitza la variable webserver a true
targeta = ""; //S'inicialitza la variable targeta
WiFi.begin(ssid, password); //Es realitza la connexió wifi passant el nom de la xarxa i la contrasenya
Serial.print("Connectant..."); //S'escriu per serial
//Es realitza la connexió wifi en arrencar el xip ESP8266
while (WiFi.status() != WL_CONNECTED && intents < 15) { //Mentre no es detecti una connexió exitosa al wifi i hi hagi menys
de 15 intents es mantindrà en el while
  delay(500); //S'atura Arduino durant 500ms
  Serial.print("."); //S'escriu pel port serial . cada 500ms
  intents++; //S'augmenta en +1 la variable intents (quan arribi a 15 sortirà del while)
//Si després de 15 intents el xip no s'ha connectat a la xarxa wifi, informarà el sistema Smartlock (port serial) que no hi ha
connexió
}if (intents >=15){
  Serial.println("[OFFLINE]"); //S'informa el sistema Smartlock que no hi ha connexió wifi
  online = false; //S'assigna a la variable online el valor false (el sistema entra en la modalitat sense connexió)
}else{ //Si el sistema s'ha connectat al wifi en menys de 15 intents entrarà
  online = true; //S'assigna a la variable online el valor true (el sistema té connexió wifi)
  Serial.println("Connexió establerta!"); //S'escriu pel port serial
  Serial.print("IP Local: "); //S'escriu pel port serial
  Serial.println(WiFi.localIP()); //S'escriu pel port serial el valor IP
//Es crea la pàgina principal del servidor ESP8266 afegint el contingut a la variable string webPage
webPage += "<script>setTimeout(function(){window.location=\"https://smartlock-uoc.000webhostapp.com/index.php\"},
100); </script>";
//Es crea la pàgina d'obertura de porta del servidor ESP8266 afegint el contingut a la variable string webOpen
webOpen += "<script>setTimeout(function(){window.location=\"https://smartlock-uoc.000webhostapp.com/index.php\"},
3000); </script>";
webOpen += "<link rel=\"stylesheet\" href=\"https://smartlock-uoc.000webhostapp.com/style.css\">";
webOpen += "<div id=\"login\">";
webOpen += "<div id=\"logo\"><img src =\"https://smartlock-uoc.000webhostapp.com/img/smartlock_logo.png\"
alt=\"Smartlock\"/></div>";
webOpen += "<div class=\"notify_door\">";
webOpen += "<p> Porta Oberta</p></div>";
webOpen += "</div>";
//Si s'accedeix al servidor amb la ip carregarà la pàgina principal
server.on("/", []) {
  server.send(200, "text/html", webPage); //Es carrega la pàgina principal
});
//Si s'accedeix al servidor amb la ip/Open es carregarà la pàgina webOpen i es passarà per serial el text [ON] (rebrà la
informació Smartlock)
server.on("/Open", []) {
  server.send(200, "text/html", webOpen); //Es carrega la pàgina WebOpen
  Serial.println("[ON]"); //Escriu pel port serial
  delay(1000); //S'atura Arduino durant 1s
//Si s'accedeix al servidor amb la ip/Close carregarà la pàgina principal (webPage) i es passarà per serial el text [OFF] (rebrà
la informació Smartlock)
server.on("/Close", []) {
  server.send(200, "text/html", webPage); //Es carrega la pàgina principal
  Serial.println("[OFF]"); //Escriu pel port serial
  delay(1000); //S'atura Arduino durant 1s
server.begin(); //S'inicialitza el servidor

```

```

Serial.println("HTTP server started");} //S'escriu pel port serial

//Aquesta funció s'executa en bucle fins a l'infinit, on si no té connexió wifi, intentarà connectar-se de nou, si perd la connexió
al servidor
//estarà fent comprovacions per si torna a estar operatiu, i finalment, si tot es troba online, quan rebí el valor de la targeta
enviarà la informació
//al servidor per comprovar si té accés
void loop() {
  server.handleClient(); //S'inicialitza el server encarregat de rebre les peticions dels clients i llançar les funcions de callback
associades
  if (Webserver = true){ //Si el servidor està operatiu comprovarà l'accés de la targeta en el servidor
    chekSerialCom(targeta); //Crida a la funció encarregada de guardar en una variable el valor d'una targeta rebuda per serial
  }if (WiFi.status() != WL_CONNECTED){ //Si es perd la connexió wifi entrarà
    reconnect (); //Crida a la funció encarregada de realitzar connexió wifi
  }if (Webserver == false){ //Si el servidor està caigut entra en el if
    verify_server ();} //Crida a la funció encarregada de comprovar si el servidor està operatiu (ping)
}
/*****FUNCIONS*****/
//Funció encarregada de guardar en la variable target el valor de la targeta rebuda pel port serial entre claudàtors
void chekSerialCom(String target) {
  while (Serial.available()) { //Mentre el port serial estigui operatiu entrarà en el while
    delay(25); //S'atura Arduino durant 25ms (el port seria és lent, i s'ha de donar temps per guardar el contingut de la targeta
en el buffer si no queda incomplet)
    char a = Serial.read(); //Es guarda cada valor rebut pel serial en la variable a
    target.concat(String(a)); //Es guarda cada valor de la variable a en string target
    if (a == ']') { //Quan es rep ] vol dir que s'ha acabat el contingut del missatge i entra en el if
      if (target == "[OK2FACTOR]" || target == "[FAIL2FACTOR]"){
        result2factor(target);
      }else {
        check_target(target); //Es crida a la funció encarregada d'enviar el valor de la targeta al servidor
      }
    }
  }
}

//Funció encarregada d'enviar el valor de la targeta rebuda per paràmetre al servidor, i enviarà pel port serial la resposta
rebuda per aquest
void check_target (String target){
  if(WiFi.status()== WL_CONNECTED){ //si hi ha una connexió wifi entrarà
    String host = "http://smartlock-uoc.000webhostapp.com/verificacio.php"; //Es guarda en la variable host la direcció de
l'arxiu php encarregat de fer la comprovació
    WiFiClient client; //Es crea l'objecte WifiClient
    HTTPClient https; //Es crea l'objecte HttpClient
    //S'inicialitza una connexió http entre el client i el host
    https.begin(client, host);
    //S'especifica el content-type header
    https.addHeader("Content-Type", "application/x-www-form-urlencoded");
    //Es guarda en la variable httpRequestData el valor rebut de la targeta
    String httpRequestData = "targeta=\" + target + "\"";
    //Es realitza la connexió https amb el servidor enviant per post el valor de la targeta
    int httpResponseCode = https.POST(httpRequestData); //Es guarda en httpResponseCode el codi de la resposta de l'operació
(si el resultat és 200 és que ha sortit correcte)
    String payload = https.getString(); //Es guarda en la variable payload el resultat del php ([ON], [OFF] o [password doble
autenticació])
  }
}

```

```

Serial.print("HTTP code: "); //S'escriu pel port serial
Serial.println(httpResponseCode); //S'escriu pel port serial el valor de la variable httpResponseCode
Serial.println(payload); //S'escriu pel port serial el valor de la resposta del servidor ([ON], [OFF] o [password doble
autenticació])
if (httpResponseCode != 200){ //Si httpResponseCode és diferent a 200 entrarà en el if (vol dir que hi ha un error)
  Serial.println("[OFFLINE]"); //S'escriu pel port serial per informar a Smartlock que no hi ha connexió
  Webserver = false; //S'assigna a la variable Webserver el valor false
}https.end(); } //Es tanca la connexió https

//Funció que s'encarrega d'enviar el resultat de la doble autenticació al servidor perquè creï un nou registre en l'historial
void result2factor (String target){
if(WiFi.status()== WL_CONNECTED){ //si hi ha una connexió wifi entrarà
  String host = "http://smartlock-uoc.000webhostapp.com/resultat2factor.php"; //Es guarda en la variable host la direcció
de l'arxiu php encarregat de guardar el resultat de la doble autenticació
  WiFiClient client; //Es crea l'objecte WifiClient
  HTTPClient https; //Es crea l'objecte HttpClient
  //S'inicialitza una connexió http entre el client i el host
  https.begin(client, host);

  //S'especifica el content-type header
  https.addHeader("Content-Type", "application/x-www-form-urlencoded");
  //Es guarda en la variable httpRequestData el valor del resultat rebut [OK2FACTOR] o [FAIL2FACTOR]
  String httpRequestData = "resultat=\\" + target + "\\"";
  //Es realitza la connexió https amb el servidor enviant per post el valor del resultat
  int httpResponseCode = https.POST(httpRequestData); //Es guarda en httpResponseCode el codi de la resposta de l'operació
(si el resultat és 200 és que ha sortit correcte)
  Serial.print("HTTP code: "); //S'escriu pel port serial
  Serial.println(httpResponseCode); //S'escriu pel port serial el valor de la variable httpResponseCode
  if (httpResponseCode != 200){ //Si httpResponseCode és diferent a 200 entrarà en el if (vol dir que hi ha un error)
    Serial.println("[OFFLINE]"); //S'escriu pel port serial per informar a Smartlock que no hi ha connexió
    Webserver = false; //S'assigna a la variable Webserver el valor false
  }https.end(); } //Es tanca la connexió https

//Funció encarregada d'intentar fer noves connexions wifi cada 5 segons
void reconnect (){
Serial.println("[OFFLINE]"); //S'escriu pel port serial per informar a Smartlock que no hi ha connexió wifi
while (WiFi.status() != WL_CONNECTED) { //mentre no hi ha connexió wifi estarà en el while
  WiFi.begin(ssid, password); //S'inicia una connexió wifi amb nom de la xarxa i contrasenya
  delay(5000); //S'atura Arduino durant 5 segons
}if (WiFi.status() == WL_CONNECTED){ //Si es detecta que hi ha una connexió wifi entrarà en el if
  Serial.println("[ONLINE]"); //S'escriu pel port serial per informar a Smartlock que torna a haver-hi connexió wifi
  online = true; } //S'assigna a la variable online el valor true

//Funció encarregada de verificar si el servidor està online
void verify_server (){
while (Ping.ping(host_server) == false){ //Mentre no es rebí un pong del servidor, es mantindrà en el while
  delay (5000); //S'atura Arduino durant 5s
  //Si es rep un pong del servidor, vol dir que torna a estar operatiu
  Serial.println("[ONLINE]"); //S'escriu pel port serial per informar a Smartlock que el servidor està online
  Webserver = true; } //S'assigna a la variable Webserver el valor true

```