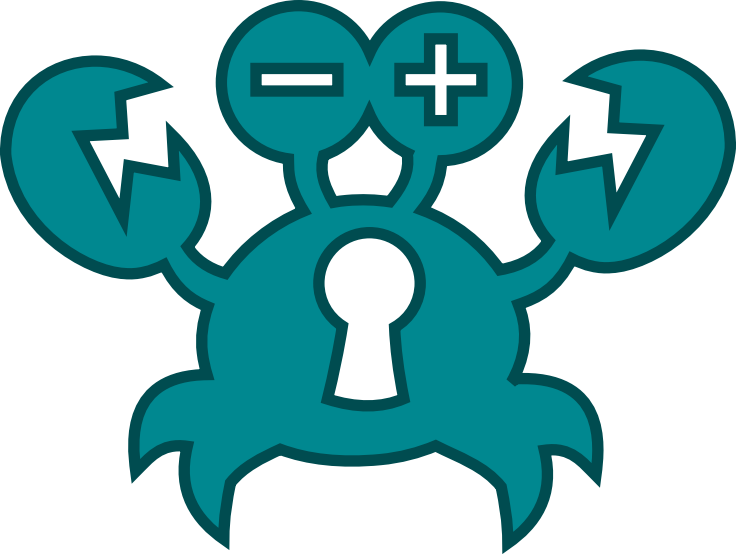
**Institución**: ESI buceo **Grupo**: 3ºBN





**Fecha**: 12/11/2021 **Integrantes**: Matías Cardozo, Joaquin Caballero, Ignacio Gomez, Maicol Arezo, Pablo Camargo

[**Solicitud de tres emprendimientos**](#_heading=h.fegfsokdqvt5) **3**

[**Introducción:**](#_heading=h.f6dinhup9xsy) **4**

[**Resumen**](#_heading=h.30j0zll) **4**

[**Guia Tecnica:**](#_heading=h.m7xnr0sn2i9o) **5**

[**Objetivo:**](#_heading=h.2et92p0) **6**

[**METODOLOGÍA:**](#_heading=h.tyjcwt) **6**

[**DISCUSIÓN**](#_heading=h.3dy6vkm) **6**

[**PÁGINA WEB**](#_heading=h.1t3h5sf) **7**

[Página inicial:](#_heading=h.4d34og8) 7

[Página principal:](#_heading=h.2s8eyo1) 8

[Teclado:](#_heading=h.17dp8vu) 9

[**DIAGRAMAS:**](#_heading=h.3rdcrjn) **10**

[DIAGRAMA DE RED:](#_heading=h.26in1rg) 10

[DIAGRAMA DE BLOQUES :](#_heading=h.9j0nq6fzia5e) 11

[Diagrama de flujo:](#_heading=h.1fob9te) 12

[**Configuracion DNS**](#_heading=h.ig7as4nnmyv6) **13**

[Configuracion pihole:](#_heading=h.xr188mkujdvp) 13

[Configuracion Router:](#_heading=h.7gtwibxu5h1b) 14

[**PRESUPUESTO**](#_heading=h.yusmg74xxg48) **15**

[Costo de producto](#_heading=h.mqs7ygbmjkt) 15

[coste de proyecto:](#_heading=h.dhjwbp7e1vz0) 16

[**Anexo:**](#_heading=h.dvzyc3rctr99) **17**

[**DOCUMENTACIÓN**](#_heading=h.xjbzx09juvg3) **18**

[MARCO NORMATIVO](#_heading=h.1ksv4uv) 18

[PROTOCOLOS](#_heading=h.5nbrpd3m7d20) 20

[RANGO DE OPERACIÓN DE LOS COMPONENTES](#_heading=h.44sinio) 20

[RFID](#_heading=h.oelxmtogtphr) 20

[ESP32](#_heading=h.vjh2vn29dps9) 20

[Cerrojo](#_heading=h.9rqzr3fno5f7) 20

[Reley](#_heading=h.q87ekezbvpeu) 20

[SD](#_heading=h.ngjxog7fk889) 20

[VERSIONES DE LOS S.O](#_heading=h.2jxsxqh) 20

[CONTRATO DE SERVICIO](#_heading=h.z337ya) 20

[UNIT](#_heading=h.3j2qqm3) 20

[EMI](#_heading=h.1y810tw) 20

[RFI](#_heading=h.4i7ojhp) 21

[I.P](#_heading=h.2xcytpi) 21

[AUTONOMÍA](#_heading=h.1ci93xb) 21

[CONDICIONES AMBIENTALES Y OPERATIVAS](#_heading=h.3whwml4) 21

[CONSUMO](#_heading=h.2bn6wsx) 21

[ALIMENTACIÓN](#_heading=h.qsh70q) 21

[NIVELES DE SEGURIDAD](#_heading=h.3as4poj) 21

[-FÍSICO:](#_heading=h.1pxezwc) 21

[USABILIDAD](#_heading=h.2p2csry) 21

[INSTALACIÓN](#_heading=h.147n2zr) 22

[MANTENIMIENTO](#_heading=h.3o7alnk) 22

[APLICACIÓN(MANUAL)](#_heading=h.23ckvvd) 22

[COMPATIBILIDAD](#_heading=h.ihv636) 22

[GARANTÍA](#_heading=h.32hioqz) 22

[CIRCUITO ELECTRICO:](#_heading=h.41mghml) 24

[CIRCUITO PCB:](#_heading=h.2grqrue) 24

[DISEÑO PCB:](#_heading=h.3iggi6mr91cf) 25

[Codigo fuente página web:](#_heading=h.551vwpf277qn) 26

[index.html:](#_heading=h.vx1227) 26

[style.css:](#_heading=h.glush2jfhovv) 27

[main-page.html:](#_heading=h.3fwokq0) 30

[main-page.css:](#_heading=h.o7yyukkmhnn6) 32

[tablet.html:](#_heading=h.4f1mdlm) 36

[Tablet.css:](#_heading=h.68djr1aql61p) 39

[Codigo fuente Esp32:](#_heading=h.2azcsyc077ie) 42

[**Bibliografia:**](#_heading=h.gazfqzqqkuzz) **58**

[Frecuencia:](#_heading=h.q5p4849y4dc5) 59

[Longitud de onda:](#_heading=h.68vjn6wub979) 59

[Frecuencia y longitud de onda:](#_heading=h.30kevvbizptd) 59

[Principio de Huygens:](#_heading=h.ew2x8mtj6pno) 59

[Difracción:](#_heading=h.8d0zlzo0kg22) 59

[Interferencia constructiva:](#_heading=h.11y303tm0sg) 59

[Definición de luz:](#_heading=h.22plwj7w5ko0) 59

[Luz IR:](#_heading=h.mjwlms4noc5j) 59

[Láser:](#_heading=h.jg6o9nw210yr) 60

[Ángulo crítico:](#_heading=h.temyu0c638v3) 60

[Como onda:](#_heading=h.o2er25wgivdd) 60

[Luz como partícula:](#_heading=h.lfq6uanelsfw) 60

[Átomo n:](#_heading=h.n1xjngkz5bj6) 60

[Átomo p:](#_heading=h.qaafs2kywqfq) 60

[Átomo e:](#_heading=h.elf4ts4yxs7b) 60

[Fotón:](#_heading=h.n2iyuswyugdu) 61

[Datasheets](#_heading=h.8p4a6seiw75t) 61

[NodeMCU 32s](#_heading=h.uuggazsupj82) 61

[ESP32](#_heading=h.z54xe9rbeizl) 61

[RFID](#_heading=h.kyclhwdkk1a2) 61

[Modulo SD](#_heading=h.qp95ahd6gidn) 61

[Relay](#_heading=h.5pvft3xnm9n9) 61

# 

# Solicitud de tres emprendimientos

El Consorcio “Innovemos” hace un llamado a emprendedores del área tecnológica, más precisamente de Telecomunicaciones y Robótica, para asignar tres gestiones relacionadas y vinculadas ediliciamente.

Se solicita entonces:

-Un proyecto que resuelva los desafíos de riego del predio destinado al jardín.

-Un proyecto destinado a la identificación y acceso al nivel uno correspondiente al sector de entrada de personal.

-Un proyecto que mejore las prestaciones de la dispensación de los sanitizadores, atendiendo los protocolos Covid-19.

En la fecha a acordar los tres concursantes seleccionados presentarán su propuesta, la cual deberá incluir una carpeta del proyecto, una presentación implementada en maqueta, y un equipo perfectamente identificado capaz de responder a las consultas del grupo que asigne el Consorcio.

El grupo “Tribunal” asumirá el carácter de veedor y asesor de cada propuesta, determinando la calificación para su aprobación final.

Todo aspecto que refiere a la funcionalidad, alcance, seguridad, soporte técnico, procedimientos, modos y medios de comunicación, supervivencia, etc., será discutido conforme avance la etapa de desarrollo de cada proyecto al asesor del consorcio con el expertiz adecuado y validado.

Los tres desafíos deben quedar vinculados a la red informática, aprovechando los medios de cobre, aire y vidrio, ser resueltos en sistema con hardware de código abierto y la etapa de potencia, principalmente, debe ser resuelta en placa impresa.

Se valorará la supervivencia de los sistemas y el uso de energías renovables. Así mismo se considerará el alcance de los emprendimientos atendiendo su capacidad de certificación, cumplimiento de normativas y característica de la garantía, mantenimiento del sistema y servicio post implementación.

Por razones de seguridad, el Consorcio proveerá un puerto con Vlan única a cada emprendimiento para que resuelva su propia red.

# Introducción:

Con la idea de hacer un sistema más higiénico para las oficinas o incluso hogares desarrollamos un sistema de control y registro de acceso a través de un módulo lector de tarjetas y tags RFID. la emergencia sanitaria que estamos viviendo nos lleva a una necesidad por tener la mayor higiene posible en todo momento, según estudios realizados se llegó a la conclusión que uno de los objetos con más contacto humano son las puertas (más específicamente los pestillos), con esta idea en mente desarrollamos un sistema que nos permite abrir las puertas minimizando el contacto con la misma.

# Resumen

Este proyecto tiene como propósito la construcción en el marco de la cultura OPEN HARDWARE(adaptable) de una “cerradura inteligente” la cual permite un registro de cada usuario y hora de acceso del mismo. Integrando diversas técnicas, dispositivos y herramientas tanto de Hardware, Software, Programación y Robótica.

El **CRAB Lock** (Control and Registry Arduino Based Lock):consiste de un lector de tarjetas RFID que a través de arduino permite acceso a determinados usuarios con un horario establecido; la cerradura para su funcionamiento dispone de dos tipos de alimentación: una batería de 12v en caso de apagones, y una fuente la cual suministra 13v para recargar la batería y 5v como alimentación al arduino. Cabe destacar que los usuarios y los horarios de entrada y salida quedarán registrados en un archivo.

The purpose of this project is the construction within the framework of the OPEN HARDWARE (adaptable) culture of an "intelligent lock" which allows a record of each user and time of access. Integrating various techniques, devices and tools from both Hardware, Software, Programming and Robotics.

The **CRAB Lock** (Control and Registry Arduino Based Lock): consists of an RFID card reader that through arduino allows access to certain users with a set schedule; The lock for its operation has two types of power: a 12v battery in case of blackouts, and a source which supplies 13v to recharge the battery and 5v as power to the arduino. It should be noted that users and entry and exit times will be recorded in a file.

# 

# Guia Tecnica:

RFID: Mrfc522 RFID o identificación por radiofrecuencia es un sistema de almacenamiento y recuperación de datos remotos que usa dispositivos denominados etiquetas, tarjetas o transpondedores RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto mediante ondas de radio.

Relay: El relé (en francés, relais 'relevo') es un dispositivo electromagnético. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

Controlador: Esp32 Se trata de un SoC o Sistema en Chip programable. Básicamente consiste en un chip que tiene todo integrado (o casi todo) para que pueda funcionar de forma autónoma como si fuera un ordenador.

SD: Una tarjeta digital segura (SD) es una tarjeta de memoria flash extraíble utilizada para almacenar información digital, como programas y archivos. Las tarjetas SD se utilizan comúnmente en teléfonos celulares y en otros dispositivos portátiles para ampliar el almacenamiento de la memoria ROM.

Cerradura: Se trata de una cerradura basada en un electroimán el cual al ser excitado atrae una pieza metálica liberando así la puerta, al cortar la corriente al electroimán dicha pieza vuelve a su posición cerrando la cerradura.

Alimentación: El dispositivo es suministrado con energía eléctrica a través de una fuente de alimentación DC de 12V 3A.

Autonomía: En caso de apagarse el suministro eléctrico de la red (alimentación) el equipo cuenta con una batería que sustenta el dispositivo manteniéndolo en funcionamiento hasta la reconección.

Circuito de corte: Este circuito impide que la batería se sobrecargue,cortando la energía y evitando que se desplace a la batería al detectar cuando esta se encuentra cargada.

regulador de voltaje: Un regulador de tensión o regulador de voltaje es un dispositivo electrónico diseñado para mantener un nivel de tensión constante o regulable, el dispositivo utiliza un regulador para alimentar el arduino con su voltaje correspondiente.

# Objetivo:

El dispositivo es ser capaz de bloquear y desbloquear una cerradura con el menor contacto posible sobre la misma para conseguir una mayor higiene, permitiendo el ingreso de usuarios autorizados manteniendo un registro de los mismos.

# METODOLOGÍA:

ensayo y prueba.

Investigamos qué diseño de cerradura segura, modificable y más rentable del mercado teniendo en cuenta su durabilidad y gastos energéticos,

para su creación decidimos utilizar sensores y placas de arduino o esp:

sensor RFID(lector de tarjetas), módulo sd, una cerradura eléctrica, esp32, relay, interruptor con llave y baterías de litio 18650.

En cada caso indagamos otros componentes de mejor calibre,

Para el lector de tarjetas descubrimos que la placa PN532 tenía mayor delay que MRC522 a la hora de leer.

Para la cerradura eléctrica investigamos que el mejor método sería usar una cerradura electrónica de perno, sin embargo por motivos de disponibilidad utilizamos un Cerrojo destrabador eléctrico.

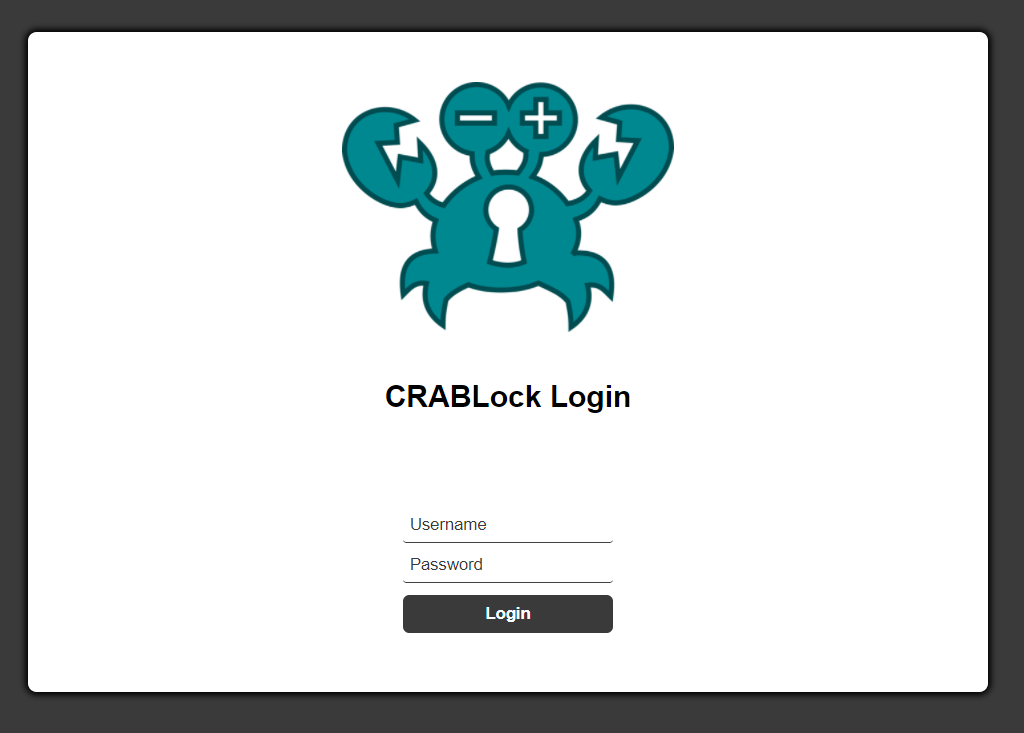
# DISCUSIÓN

Se logró realizar la cerradura la cual al pasar una tarjeta con un usuario registrado por un lector se desbloquea y se vuelve a cerrar una vez la puerta se cierra, también registrando la hora y el nombre del usuario.

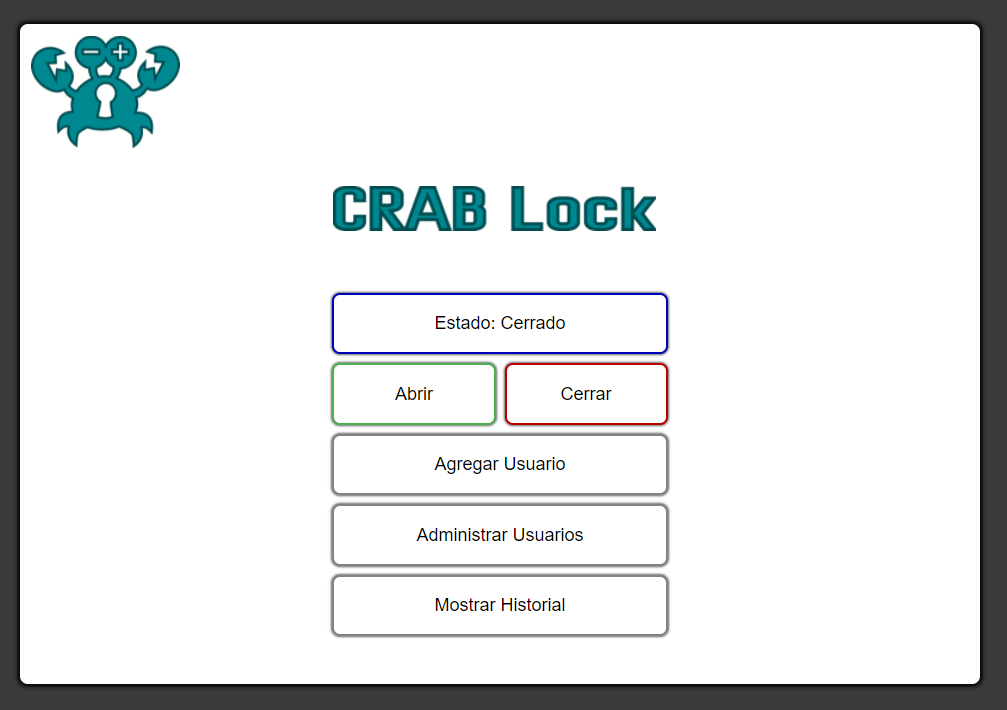
En cada proceso se hizo una prueba y error, se probó con hacer el sistema fuera de una puerta real, pero por practicidad utilizamos una maqueta a ¼ de escala basada en una puerta de datacenter.

# PÁGINA WEB

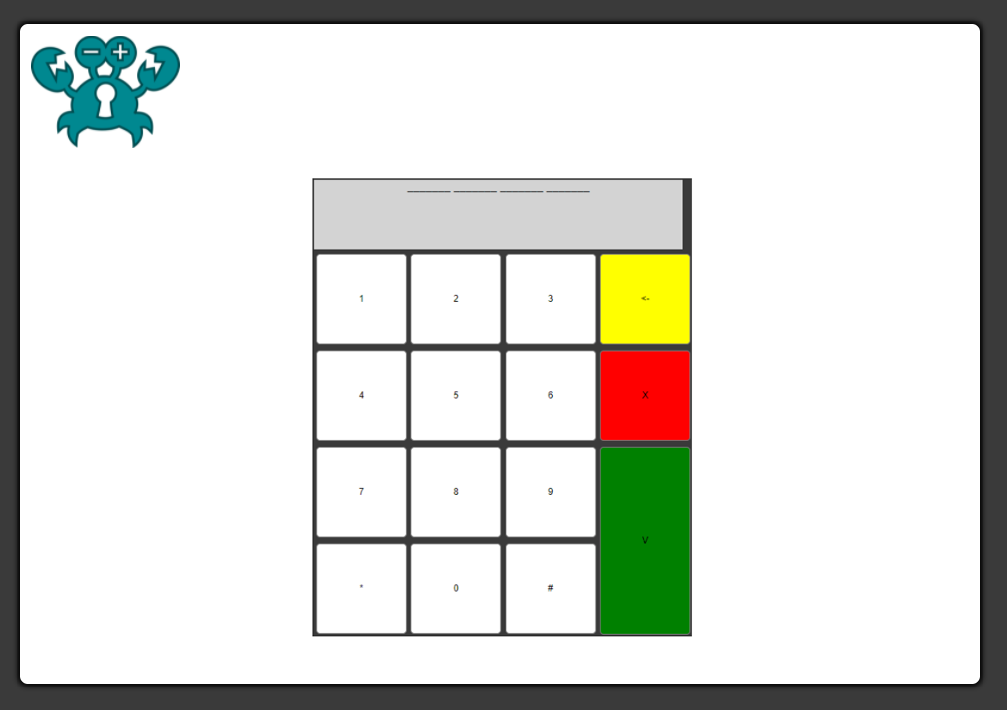
## Página inicial:



## Página principal:

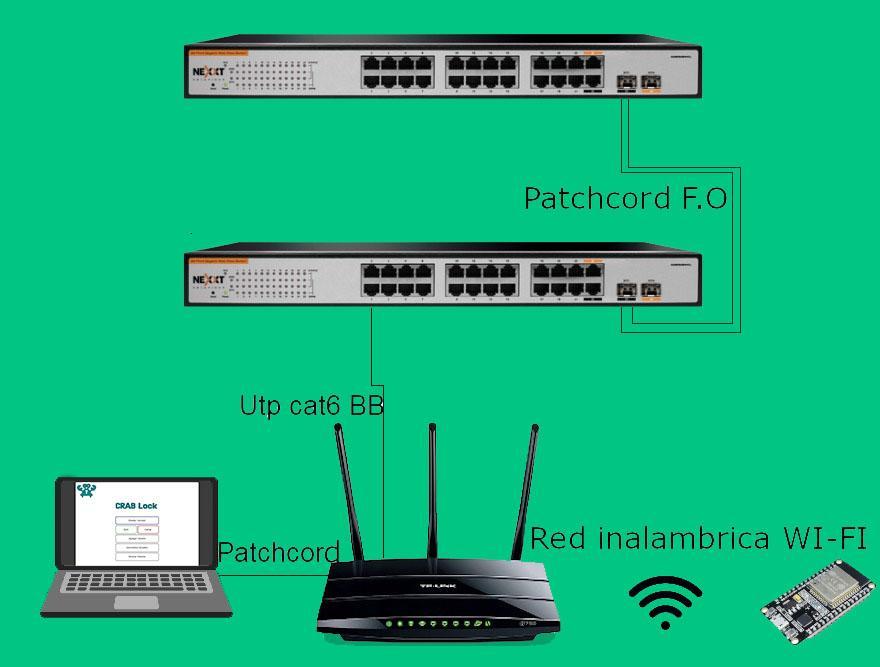


## Teclado:

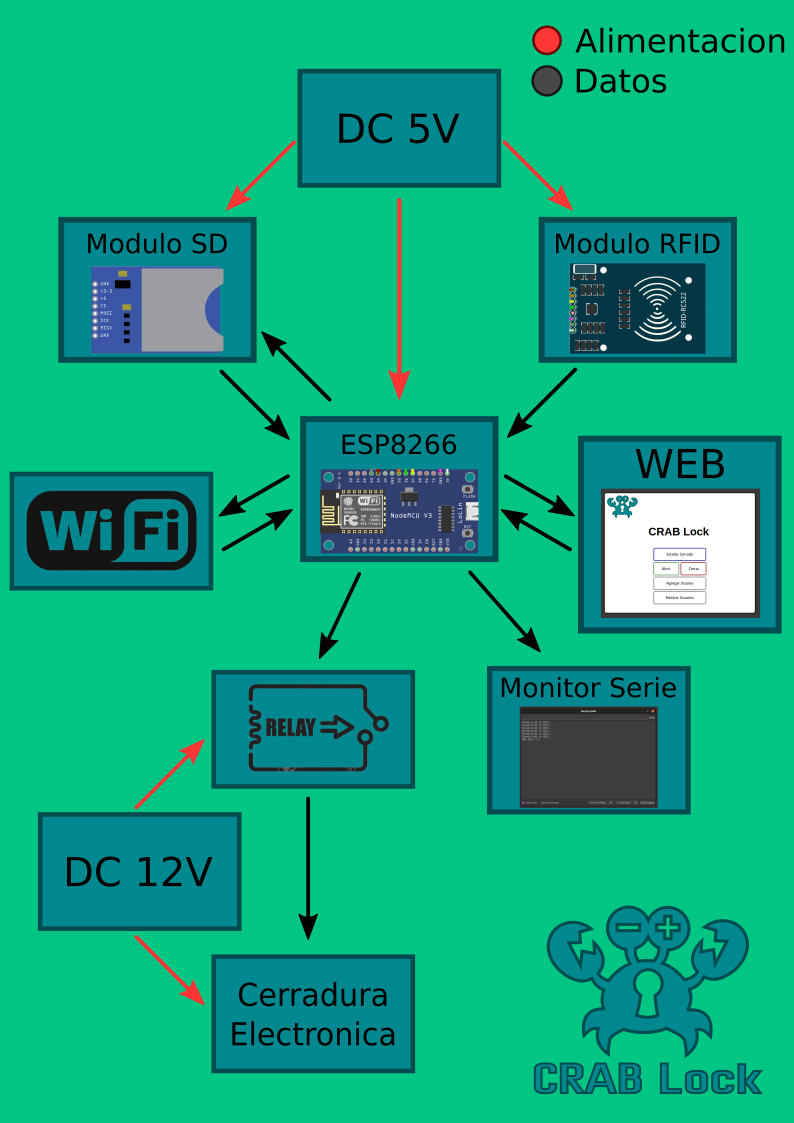


# DIAGRAMAS:

## DIAGRAMA DE RED:

****

## DIAGRAMA DE BLOQUES :

****

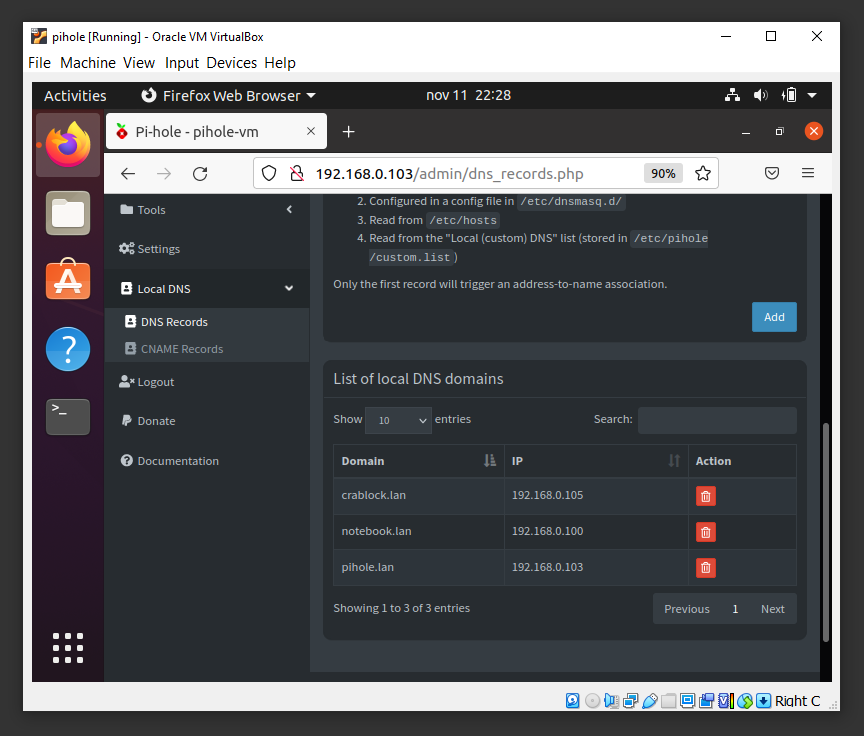
## Diagrama de flujo:

# 

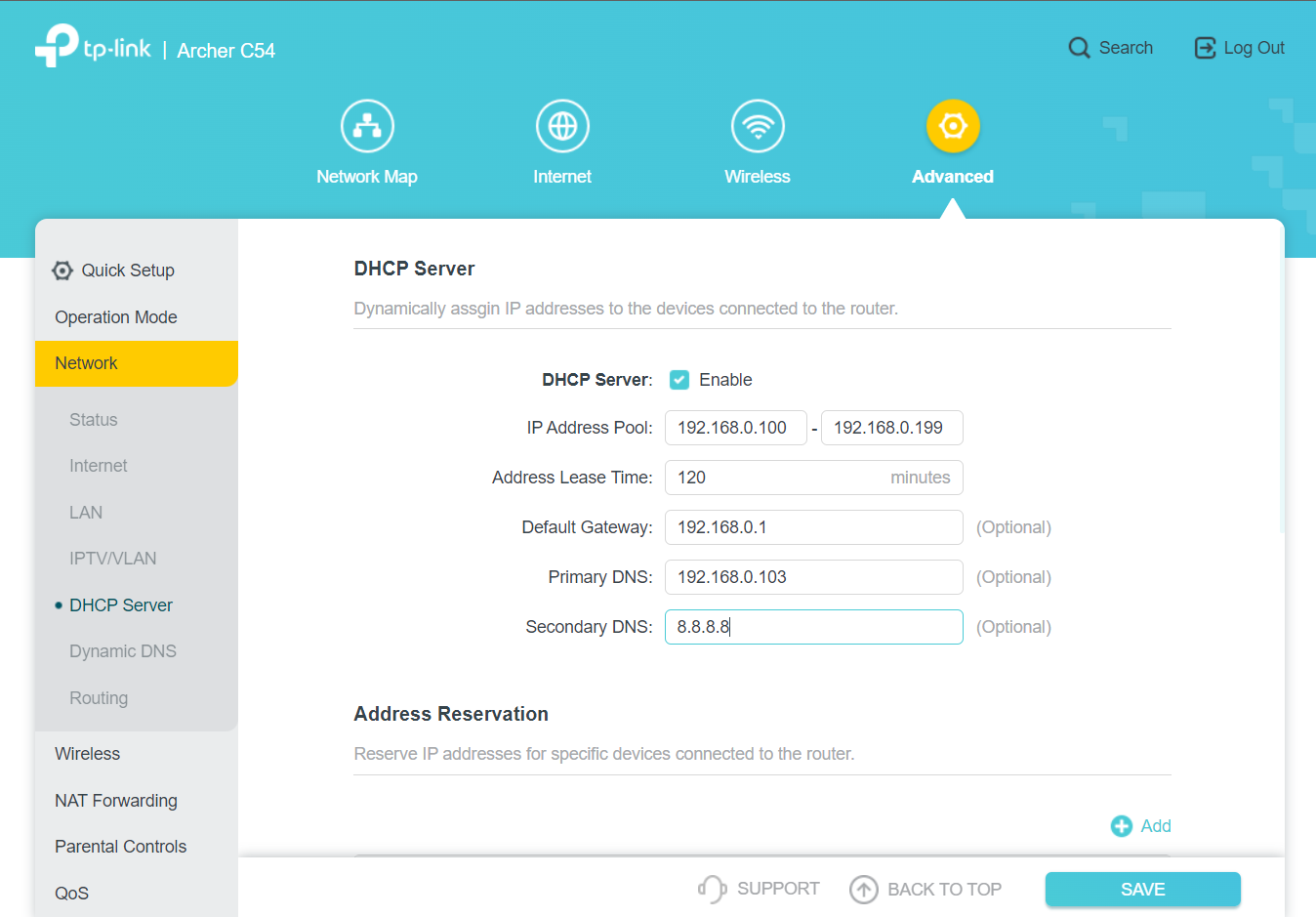
# 

# Configuracion DNS

## Configuracion pihole:



## Configuracion Router:



# 

# 

# PRESUPUESTO

## Costo de producto

| RFID | 1 | 5 |
| --- | --- | --- |
| Lector SD | 1 | 3 |
| Reley | 1 | 3 |
| LED 20U | 1 | 2 |
| LED RGB 5U | 1 | 2.5 |
| ESP32 | 1 | 25 |
| Cerradura | 1 | 18 |
| Baterías | 3 |  |
| Buzzer | 1 | 0.5 |
| Cabezales 40P x 5U | 1 | 4 |
| Transistores | 1 | 1.5 |
| Resistencias | 2 | 1 |
| PCB | 1 | 4 |
| tarjetas | 4 | 4 |
| Instalación | 1 | 30 |
| Mano de obra | 100HS+ | 3 |
| Diseño | 30HS+ | 5 |
| Precio Final | 25.000UYU | 555USD |

## coste de proyecto:

| RFID | 2 | 10 |
| --- | --- | --- |
| Lector SD | 2 | 6 |
| Reley | 2 | 6 |
| LED 20U | 1 | 2 |
| LED RGB 5U | 1 | 2.5 |
| ESP32 | 1 | 25 |
| Cerradura | 1 | 18 |
| Baterías | 3 |  |
| Buzzer | 1 | 0.6 |
| Cabezales 40P x 5U | 1 | 4 |
| Transistores | 1 | 1.5 |
| Resistencias | 2 | 3 |
| PCB | 2 | 8 |
| tarjetas | 4 | 4 |
| ESP8266 | 1 | 14 |
| Pomela | 1 | 4 |
| cables | 1 | 6 |
| maqueta | 1 | 40 |
| Precio final | 6.150UYU | 136.6USD |

## 

# 

# 

# Anexo:

# DOCUMENTACIÓN

## MARCO NORMATIVO

MTB: Ámbito de Aplicación. Las prescripciones contenidas en este Reglamento se refieren a instalaciones definidas por tensiones iguales o inferiores a los 1000 V y particularmente a los servicios suministrados por UTE, que son: En sistemas sin neutro: 50 Hz; 220 V de tensión compuesta. En sistemas con neutro: 50 Hz; 380 V de tensión compuesta y 220 V entre fase y neutro.

Nota: Se aconseja que se utilicen para estos trabajos interruptores termo magnéticos. Si se instalan a la intemperie lo mismo que los fusibles, deben estar colocados en cajas estancas a la humedad.

Medidas de Protección contra Contactos Directos e Indirectos. Las instalaciones eléctricas se establecerán de forma que no supongan riesgo para las personas (y eventualmente para los animales domésticos) tanto en servicio normal como cuando puedan presentarse averías previsibles. En relación con estos riesgos, las instalaciones deberán proyectarse y ejecutarse aplicando las medidas de protección necesarias contra los contactos directos e indirectos. Estas medidas de protección son las señaladas en el Capítulo VI.

Las instalaciones deberán presentar una resistencia de aislamiento mayor a 1.000 × U ohmios, siendo U la máxima tensión de servicio expresada en voltios, con un mínimo de 250.000 ohmios. Este aislamiento se entiende para una instalación en la cual la longitud del conjunto de canalización eléctrica y cualquiera que sea el número de conductores que las componen no exceda de 100 metros de longitud, bien por seccionamiento, desconexión, retirada de fusibles o aperturas de interruptores, cada una de las partes en que la instalación ha sido fraccionada debe presentar el aislamiento que corresponda. Cuando no sea posible efectuar el fraccionamiento citado, se admite que el valor de la resistencia de aislamiento de toda la instalación sea, con relación al mínimo que le corresponda, inversamente proporcional a la longitud de las canalizaciones eléctricas.

Las corrientes máximas admisibles en servicio permanente, para conductores aislados en canalizaciones eléctricas fijas, y a una temperatura ambiente de 25°C, se indican en las distintas Tablas de este reglamento, según sea el tipo de aislamiento, sistema de instalación y medio ambiente.

Corrientes máximas Admisibles. En la Tabla I figuran las corrientes máximas admisibles en régimen permanente para este tipo de cables, en condiciones normales de instalación. Las condiciones normales de instalación se definen como un solo cable tripolar, o tetrapolar, instalado al aire libre en una disposición que permita una eficaz renovación de aire, y a una temperatura ambiente de 25 °C. Para otras condiciones diferentes, en el apartado 2.2 figuran los factores de corrección apropiados.

5.1.- Factores de Corrección. La corriente máxima admisible, deducida de las Tablas X, XI, XII y XIII deberá corregirse teniendo en cuenta las características de la instalación, de forma que el incremento de temperatura provocado por la corriente eléctrica, no dé lugar a una temperatura en el conductor superior a 70°C, en los cables con aislamiento de policloruro de vinilo o de goma y 90°C en los cables con aislamiento de goma butílica, etileno - propileno o polietileno reticulado. Para valores de temperatura ambiente diferente de 25 °C, se aplicarán los factores de corrección de la Tabla XIV, según el tipo de aislamiento. Cuando por un tubo o conducto tengan que pasar más de 3 conductores, normalmente recorridos por la corriente, los valores de la corriente máxima admisible de la última columna, se reducirá aplicando los factores de reducción siguientes: de 4 a 7 conductores = 0.90 más de 7 conductores = 0.70 Para el cómputo de estos conductores no se tendrá en cuenta, en ningún caso, el conductor de protección, ni el neutro, en un suministro trifásico con neutro

Puestas a la Tierra. Definición. La denominación "puesta a tierra" comprende toda la ligazón metálica directa, sin fusible ni protección alguna, de sección suficiente, entre determinados elementos o partes de una instalación y un electrodo, o grupo de electrodos, enterrados en el suelo, con objeto de conseguir que en el conjunto de instalaciones, edificios y superficie próxima del terreno no existan diferencias de potencial peligrosas y que, al mismo tiempo, permita el paso a tierra de las corrientes de falta o la de descarga de origen atmosférico.

Puestas a tierra en pequeños o medianos suministros eléctricos individuales. En el caso de viviendas, pequeños comercios o industrias individuales, ubicados en predios independientes, y cuyas cargas solicitadas no superen los 15 kW en 220 V, o los 20 kW en 380 V, las secciones de los conductores de protección serán los indicados en el numeral 10.- . En estos casos, si el suelo es de conductividad adecuada y las condiciones obtenidas con un electrodo simple son suficientes para alcanzar los objetivos expresados en el numeral 1.- en lugar de utilizar el anillo de enlace con tierra será suficiente con conectar la línea principal de tierra con el electrodo en la respectiva cámara. En el caso de que esta solución no fuese suficiente se deberá adoptar el criterio reglamentario más adecuado. Si fuera necesaria la instalación de pararrayos para proteger un área de pequeñas dimensiones, se deberá respetar, en todo lo expresado en la figura referida al "esquema de un sistema de puesta a tierra".

## 

## PROTOCOLOS

SPI,RFID, WiFi, HTTP.

## RANGO DE OPERACIÓN DE LOS COMPONENTES

### RFID

Rango de frecuencias: Banda ISM de 13,56 Mhz

Voltaje de operación: 2,5V a 3,3V

Corriente máx. de funcionamiento: 13-26 mA

Alcance 35Cm

### ESP32

Corriente de Operación: ~80mA (fuente superior a 500mA)

Voltaje de Alimentación: 3.3V DC (2.7~ 3.6V)

### Cerrojo

alimentación 8V-12V

### Relay

250V AC - 5V DC - 10A

### SD

alimentación 2.7V a 3.6 V

## VERSIONES DE LOS S.O

la versión del sistema operativo usada para la programación del proyecto fue Windows 10 ya que es un sistema operativo muy estable y con un mejor acceso a las herramientas de programación como el arduino IDE

## CONTRATO DE SERVICIO

## UNIT

El conductor es de cobre con un diámetro de 2mm2, y su resistividad es de 0.0176Ω por mm2/m, con una aislación de PVC O assl

## EMI

El dispositivo puede ser afectado por interferencias electromagnéticas de los conductores que tenga las paredes del comercio y vivienda, estas pueden interrumpir o afectar la comunicación wifi.

## RFI

El dispositivo genera cierta interferencia por su campo electromagnético que puede afectar los dispositivos más cercanos

## I.P

Este producto está certificado con una protección de ingreso IP20, lo que significa que está protegido contra cuerpos sólidos de hasta 12mm de diámetro (como lo puede ser un dedo), y no está protegido contra entrada de agua

## AUTONOMÍA

Su autonomía se basa en una batería de 12v 4ah la cual se mantiene cargada y lista para una emergencia. Esta mantendrá al dispositivo almenos unos

## CONDICIONES AMBIENTALES Y OPERATIVAS

Uso exclusivamente interior. No resiste golpes ni sobrecargas o cortocircuito.

## CONSUMO

< 35W

## ALIMENTACIÓN

alimentaremos el mecanismo y los dispositivos mediante un transformador (12V 3A) conectado a la corriente eléctrica

## NIVELES DE SEGURIDAD

### -FÍSICO:

Como seguridad física contamos con una caja hecha a medida que protege el dispositivo y a sus componentes, además que la placa que utilizamos cuenta con un sistema que requiere acceso a la placa para cargar el código dentro de ella, el acceso a la placa programable esp32 será restringido para que solo el personal autorizado pueda acceder

## USABILIDAD

Nuestro dispositivo es muy sencillo y amigable hacia el usuario. Consta de 2 partes fundamentales un lector de tarjetas que simplemente acercando una tarjeta con una ID autorizada accionarias el el sistema o desde una interfaz web dando una serie de opciones de gestión que permiten un fácil control del dispositivo siempre y cuando estés autorizado.

## INSTALACIÓN

La instalación del producto es muy sencilla, simplemente es colocar un cerrojo eléctrico donde iría el cerrojo tradicional y adherir el dispositivo a un lugar cercano y con acceso a corriente y a internet si se quiere usar la interfaz web,

## MANTENIMIENTO

Crablock ofrece un servicio de mantenimiento semestral, el cual consta de testeos y calibración de sensores, control de almacenamiento y control de voltaje de el sistema de autonomía. Siendo gratuito el primer año de uso.

## APLICACIÓN(MANUAL)

El sistema de la cerradura permite la apertura o en casos específicos registro de usuarios, se debe pasar la tarjeta por el lector, en caso de que en el dispositivo se encienda una luz en rojo significa que la tarjeta no está registrada como un usuario, por el contrario si se ilumina una luz verde y suena una bocina la puerta se libera y se podrá abrir empujándola un poco, se cerrará automáticamente, una vez dentro se podrá abrir la puerta sin la necesidad de una tarjeta ya que tiene un pomo normal, en caso de que un administrador quieras agregar quiera agregar un usuario se puede hacer desde la página presionando “agregar usuario” y pasando la tarjeta sobre el lector.

## COMPATIBILIDAD

El dispositivo es compatible con casi cualquier puerta siempre y cuando esté en condiciones óptimas para el correcto funcionamiento del dispositivo. además que puede adaptarse fácilmente si es necesario

## GARANTÍA

La garantia cubre:

-mal funcionamiento de sensores/actuadores por parte del fabricante.

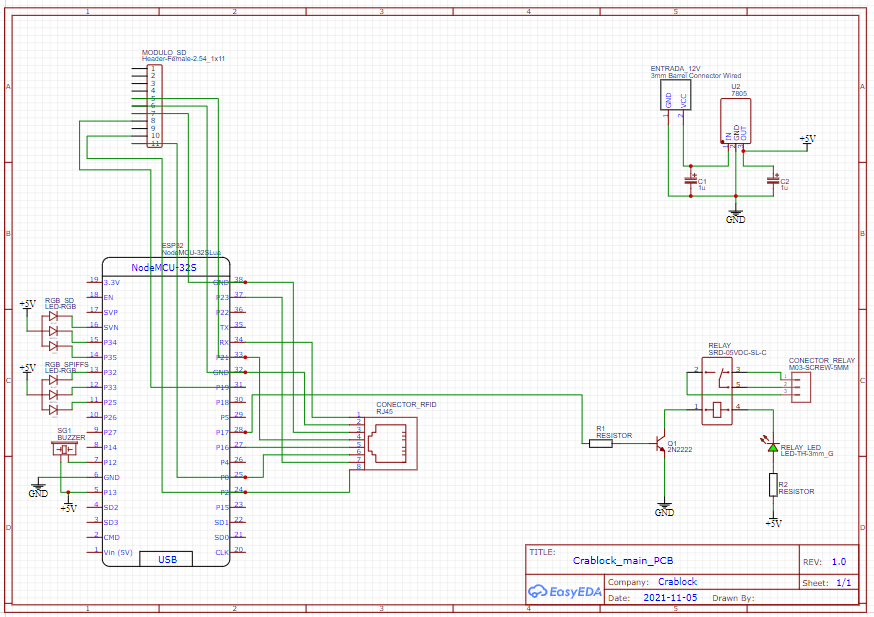
-mal funcionamiento o daños a causa de errores de instalación por parte de un técnico autorizado por Crablock.

La empresa no se hace responsable de daños ocasionados por mal uso o negligencia del usuario.

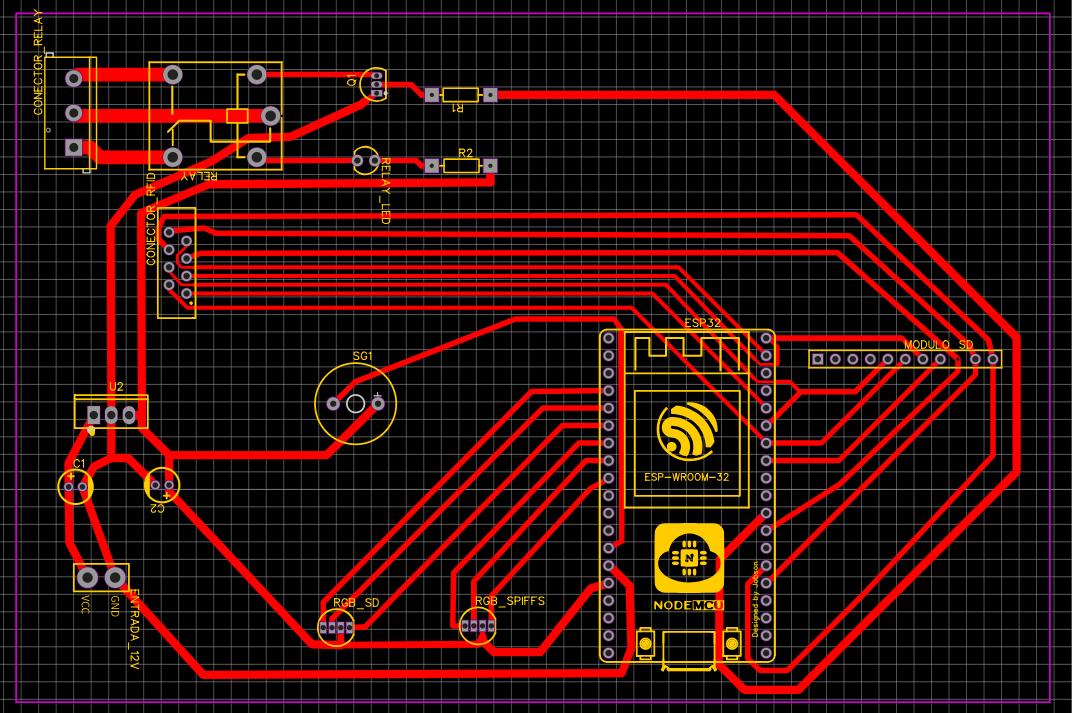
Crablock ofrece el servicio de mantenimiento semestral gratis el primer año de uso.

## 

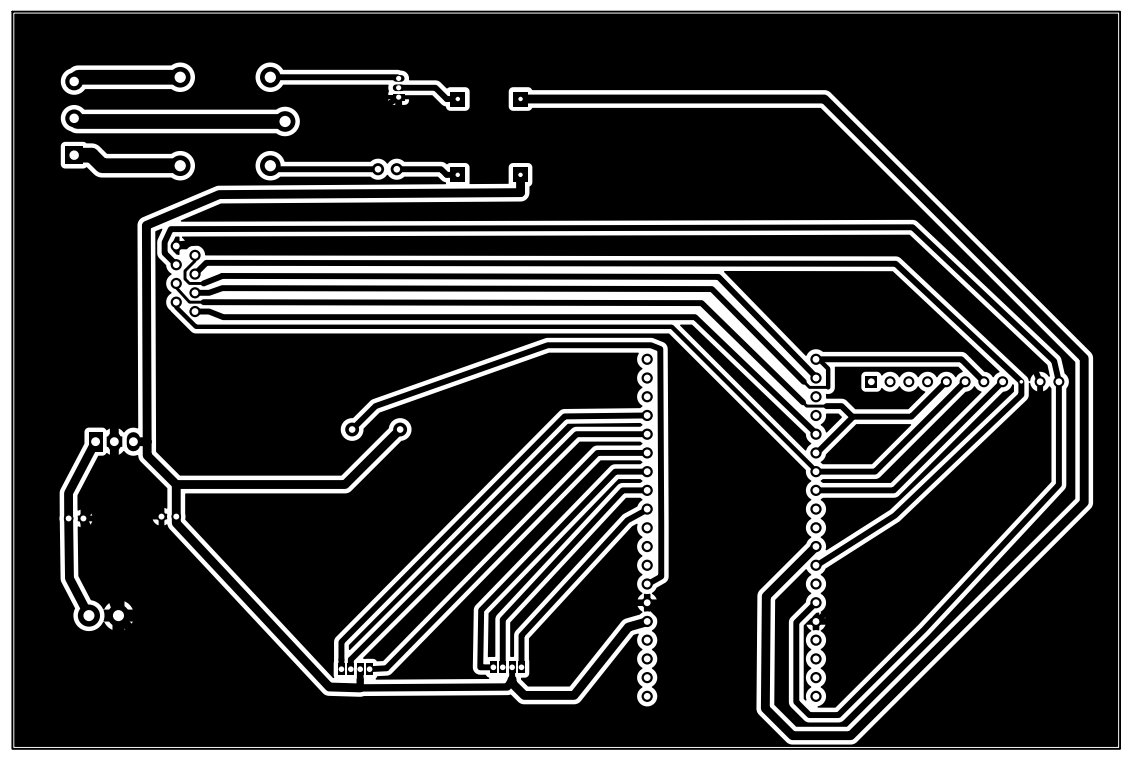
## CIRCUITO ELECTRICO:



## CIRCUITO PCB:



## DISEÑO PCB:



## 

## 

## Codigo fuente página web:

### index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CRABLock Login</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<main id="main-holder">

<br>

<div><img src="https://i.ibb.co/kGJk3RK/CRABLock-logo.png" height=200> </div>

<h2 id="login-header">CRABLock Login</h2>

<div id="login-error-msg-holder">

<p id="login-error-msg">Usuario o pass<span id="error-msg-second-line">invalidos</span></p>

</div>

<form id="login-form">

<input type="text" name="username" id="username-field" class="login-form-field" placeholder="Username">

<input type="password" name="password" id="password-field" class="login-form-field" placeholder="Password">

<input type="submit" value="Login" id="login-form-submit">

</form>

<br>

</main>

</body>

<script>

const loginForm = document.getElementById("login-form");

const loginButton = document.getElementById("login-form-submit");

const loginErrorMsg = document.getElementById("login-error-msg");

loginButton.addEventListener("click", (e) => {

e.preventDefault();

const username = loginForm.username.value;

const password = loginForm.password.value;

if (username === "user" && password === "pass") {

window.location.href = "main-page.html";

} else {

loginErrorMsg.style.opacity = 1;

}

})

</script>

</html>

### style.css:

html {

height: 100%;

}

body {

height: 100%;

margin: 0;

font-family: Arial, Helvetica, sans-serif;

display: grid;

justify-items: center;

align-items: center;

background-color: #3a3a3a;

}

div.inline {

display: inline-block;

vertical-align:middle;

}

#main-holder {

min-width: 300px;

max-width: 500;

min-height: 500px;

width: 50%;

height: 70%;

display: grid;

justify-items: center;

align-items: center;

background-color: white;

border-radius: 7px;

box-shadow: 0px 0px 5px 2px black;

}

#login-error-msg-holder {

width: 100%;

height: 50%;

display: grid;

justify-items: center;

align-items: center;

}

#login-error-msg {

width: 23%;

text-align: center;

margin: 0;

padding: 5px;

font-size: 12px;

font-weight: bold;

color: #8a0000;

border: 1px solid #8a0000;

background-color: #e58f8f;

opacity: 0;

}

#error-msg-second-line {

display: block;

}

#login-form {

align-self: flex-start;

display: grid;

justify-items: center;

align-items: center;

}

.login-form-field::placeholder {

color: #3a3a3a;

}

.login-form-field {

border: none;

border-bottom: 1px solid #3a3a3a;

margin-bottom: 10px;

border-radius: 3px;

outline: none;

padding: 0px 0px 5px 5px;

}

#login-form-submit {

width: 100%;

padding: 7px;

border: none;

border-radius: 5px;

color: white;

font-weight: bold;

background-color: #3a3a3a;

cursor: pointer;

outline: none;

}

### main-page.html:

<!DOCTYPE HTML>

<html>

<head>

<meta content="text/html;charset=utf-8" http-equiv="Content-Type">

<meta content="utf-8" http-equiv="encoding">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="main-page.css">

<title>CRAB Lock</title>

</head>

<body>

<div id="adminUsers">

<span class="helper"></span>

<div>

<div id="popupCloseButton" onclick="mostrarUsers()">&times;</div>

%MOSTRARUSUARIOS%

</div>

</div>

<div id="showLogs">

<span class="helper"></span>

<div>

<div id="popupCloseButton" onclick="mostrarLogs()">&times;</div>

%MOSTRARLOGS%

<a href="/logs.txt" download>Descargar logs.txt</a>

</div>

</div>

<main id="main-holder">

<div id="logo"><a href="/"><img src="https://i.ibb.co/kGJk3RK/CRABLock-logo.png" height=100></a></div>

<h2><img src="https://i.ibb.co/jGgchqP/CRABlock.png" height=40></h2>

%BUTTONPLACEHOLDER%

<br>

<button class="button button1" onclick="abrir()">Abrir</button>

<button class="button button2" onclick="cerrar()">Cerrar</button>

<br>

<button class="button button4" onclick="agregarUsuario()">Agregar Usuario</button>

<br>

<button class="button button5" onclick="mostrarUsers()">Administrar Usuarios</button>

<br>

<button class="button button6" onclick="mostrarLogs()">Mostrar Historial</button>

<br>

</main>

</body>

<script type="text/javascript">

function toggle() {

var xhr = new XMLHttpRequest();

xhr.open("GET", "/update?opcion=2", true);

xhr.send();

location.reload();

}

function abrir() {

var xhr = new XMLHttpRequest();

xhr.open("GET", "/update?opcion=1", true);

xhr.send();

location.reload();

}

function cerrar() {

var xhr = new XMLHttpRequest();

xhr.open("GET", "/update?opcion=0", true);

xhr.send();

location.reload();

}

function agregarUsuario() {

var xhr = new XMLHttpRequest();

xhr.open("GET", "/update?opcion=3", true);

xhr.send();

}

function mostrarUsers() {

var elem = document.getElementById('adminUsers');

if(elem.style.display === "block"){

elem.style.display = "none";

}else{

elem.style.display = "block";

}

}

function descargarLogs() {

var xhr = new XMLHttpRequest();

xhr.open("GET", "/update?opcion=4", true);

xhr.send();

}

function mostrarLogs(){

var elem = document.getElementById('showLogs');

if(elem.style.display === "block"){

elem.style.display = "none";

}else{

elem.style.display = "block";

}

}

function elimUser(x){

var xhr = new XMLHttpRequest();

if (confirm("eliminar usuario " + (x) + "?")) {

xhr.open("GET", `/update?eliminar=${x}`, true);

xhr.send();

setTimeout(() => {

location.reload();

}, 500);

alert("usuario " + (x) + " eliminado");

}

}

</script>

</html>

### main-page.css:

html {

height: 100%;

font-family: Arial;

text-align: center;

}

h2 {font-size: 3.0rem; padding-right: 10px;}

p {font-size: 3.0rem;}

body {

height: 100%;

margin: 0;

font-family: 'Dhurjati';

display: grid;

justify-items: center;

align-items: center;

background-color: #3a3a3a;

}

#logo {

margin-left : 10px;

width: 100px;

height: 100px;

position: relative;

top: 10px;

}

#main-holder {

min-width: 350px;

min-height: 500px;

width: 50%;

height: 70%;

justify-items: center;

align-items: center;

background-color: white;

border-radius: 7px;

box-shadow: 0px 0px 5px 2px black;

}

@media (orientation: portrait) {

.button {

border: none;

color: white;

padding: 16px 32px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

margin: 4px 2px;

transition-duration: 0.4s;

cursor: pointer;width: 70%;

border-radius: 7px;

box-shadow: 0px 0px 2px 1px dimgrey;

}

.button1 {background-color: white;color: black;border: 2px solid #4CAF50;width: 34%}

.button2 {background-color: white;color: black;border: 2px solid #ba0000;width: 34%}

}

@media (orientation: landscape) {

.button {

border: none;

color: white;

padding: 16px 32px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

margin: 4px 2px;

transition-duration: 0.4s;

cursor: pointer;width: 35%;

border-radius: 7px;

box-shadow: 0px 0px 2px 1px dimgrey;

}

.button1 {background-color: white;color: black;border: 2px solid #4CAF50;width: 17%}

.button2 {background-color: white;color: black;border: 2px solid #ba0000;width: 17%}

}

.button3 {background-color: white;color: black;border: 2px solid #0003ba;}

.button4 {background-color: white;color: black;border: 2px solid #828282;}

.button5 {background-color: white;color: black;border: 2px solid #828282;}

.button6 {background-color: white;color: black;border: 2px solid #828282;}

.buttonSesion{background-color: white;}

/\* Popup box BEGIN \*/

#adminUsers{

background:rgba(0,0,0,.4);

cursor:pointer;

display:none;

height:100%;

position:fixed;

text-align:center;

align-items: center;

justify-items: center;

top:0;

width:100%;

z-index:10000;

}

#adminUsers .helper{

display:inline-block;

height:100%;

vertical-align:middle;

}

#adminUsers > div {

background-color: #fff;

box-shadow: 10px 10px 60px #555;

display: inline-block;

height: auto;

min-height: 100px;

vertical-align: middle;

width: 300;

position: relative;

border-radius: 8px;

padding: 15px 5%;

}

#showLogs{

background:rgba(0,0,0,.4);

cursor:pointer;

display:none;

height:100%;

position:fixed;

text-align:center;

align-items: center;

justify-items: center;

top:0;

width:100%;

z-index:10000;

}

#showLogs .helper{

display:inline-block;

height:100%;

vertical-align:middle;

}

#showLogs > div {

background-color: #fff;

box-shadow: 10px 10px 60px #555;

display: inline-block;

height: auto;

min-height: 100px;

vertical-align: middle;

width: 300;

position: relative;

border-radius: 8px;

padding: 15px 5%;

}

#popupCloseButton {

background-color: #fff;

border: 3px solid #999;

border-radius: 50px;

cursor: pointer;

display: inline-block;

font-family: arial;

font-weight: bold;

position: absolute;

top: -20px;

right: -20px;

font-size: 25px;

line-height: 30px;

width: 30px;

height: 30px;

text-align: center;

}

#popupCloseButton:hover {

background-color: #ccc;

}

#abrirPopup {

cursor: pointer;

font-size: 20px;

margin: 20px;

display: inline-block;

font-weight: bold;

}

/\* Popup box BEGIN \*/

## 

### tablet.html:

<!DOCTYPE HTML>

<html>

<head>

<meta content="text/html;charset=utf-8" http-equiv="Content-Type">

<meta content="utf-8" http-equiv="encoding">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="main-page.css">

<title>CRAB Lock</title>

</head>

<body>

<div id="adminUsers">

<span class="helper"></span>

<div>

<div id="popupCloseButton" onclick="mostrarUsers()">&times;</div>

%MOSTRARUSUARIOS%

</div>

</div>

<div id="showLogs">

<span class="helper"></span>

<div>

<div id="popupCloseButton" onclick="mostrarLogs()">&times;</div>

%MOSTRARLOGS%

<a href="/logs.txt" download>Descargar logs.txt</a>

</div>

</div>

<main id="main-holder">

<div id="logo"><a href="/"><img src="https://i.ibb.co/kGJk3RK/CRABLock-logo.png" height=100></a></div>

<h2><img src="https://i.ibb.co/jGgchqP/CRABlock.png" height=40></h2>

%BUTTONPLACEHOLDER%

<br>

<button class="button button1" onclick="abrir()">Abrir</button>

<button class="button button2" onclick="cerrar()">Cerrar</button>

<br>

<button class="button button4" onclick="agregarUsuario()">Agregar Usuario</button>

<br>

<button class="button button5" onclick="mostrarUsers()">Administrar Usuarios</button>

<br>

<button class="button button6" onclick="mostrarLogs()">Mostrar Historial</button>

<br>

</main>

</body>

<script type="text/javascript">

function toggle() {

var xhr = new XMLHttpRequest();

xhr.open("GET", "/update?opcion=2", true);

xhr.send();

location.reload();

}

function abrir() {

var xhr = new XMLHttpRequest();

xhr.open("GET", "/update?opcion=1", true);

xhr.send();

location.reload();

}

function cerrar() {

var xhr = new XMLHttpRequest();

xhr.open("GET", "/update?opcion=0", true);

xhr.send();

location.reload();

}

function agregarUsuario() {

var xhr = new XMLHttpRequest();

xhr.open("GET", "/update?opcion=3", true);

xhr.send();

}

function mostrarUsers() {

var elem = document.getElementById('adminUsers');

if(elem.style.display === "block"){

elem.style.display = "none";

}else{

elem.style.display = "block";

}

}

function descargarLogs() {

var xhr = new XMLHttpRequest();

xhr.open("GET", "/update?opcion=4", true);

xhr.send();

}

function mostrarLogs(){

var elem = document.getElementById('showLogs');

if(elem.style.display === "block"){

elem.style.display = "none";

}else{

elem.style.display = "block";

}

}

function elimUser(x){

var xhr = new XMLHttpRequest();

if (confirm("eliminar usuario " + (x) + "?")) {

xhr.open("GET", `/update?eliminar=${x}`, true);

xhr.send();

setTimeout(() => {

location.reload();

}, 500);

alert("usuario " + (x) + " eliminado");

}

}

</script>

</html>

### Tablet.css:

html {

height: 100%;

font-family: Arial;

text-align: center;

}

body {

height: 100%;

margin: 0;

font-family: Arial, Helvetica, sans-serif;

display: grid;

justify-items: center;

align-items: center;

background-color: #3a3a3a;

}

#row {

width: 130px;

height: 130px;

background-color: white;

color: black;

border: 2px solid #828282;

border-radius: 5px;

justify-items: center;

align-items: center;

text-align: center;

font-size: 30;

margin: 3px;

}

#x {

width: 130px;

height: 130px;

background-color: red;

color: black;

border: 2px solid #828282;

border-radius: 5px;

justify-items: center;

align-items: center;

text-align: center;

font-size: 30;

margin: 3px;

}

#del {

width: 130px;

height: 130px;

background-color: yellow;

color: black;

border: 2px solid #828282;

border-radius: 5px;

justify-items: center;

align-items: center;

text-align: center;

font-size: 30;

margin: 3px;

}

#ok {

width: 130px;

height: 270px;

background-color: green;

color: black;

border: 2px solid #828282;

border-radius: 5px;

justify-items: center;

align-items: center;

text-align: center;

font-size: 30;

margin: 3px;

}

.pass{

width: 530px;

height: 100px;

background-color: lightgrey;

margin:3px;

justify-items: center;

}

.col {

display: flex;

margin:3px;

}

.lastcol{

display: flex;

margin:3px;

margin-top: -140px;

}

#main-holder {

display: grid;

min-width: 350px;

min-height: 500px;

width: 50%;

height: 70%;

justify-items: center;

align-items: center;

background-color: white;

border-radius: 7px;

box-shadow: 0px 0px 5px 2px black;

}

## 

## Codigo fuente Esp32:

#include <FS.h>

#include <SD.h>

#include <SPI.h>

#include <MFRC522.h>

#include <WiFi.h>

#include <AsyncTCP.h>

#include <ArduinoJson.h>

#include <ESPAsyncWebServer.h>

#include "SPIFFS.h"

#include <NTPClient.h>

#include <WiFiUdp.h>

#define RST\_PIN 22

#define SS\_PIN 21

MFRC522 reader(SS\_PIN, RST\_PIN);

MFRC522::MIFARE\_Key key;

WiFiUDP ntpUDP;

NTPClient timeClient(ntpUDP);

File myFile;

String usersJson;

String Users[8];

bool estaAbierto = false;

bool funcionActiva = false;

int cantUsuarios = 6;

byte LecturaUID[4];

byte Usuarios[8][4];

//Servidor Web///////////////////////////////////////////////

const char\* ssid = "Crablock";

const char\* password = "notengoidea";

/\*

const char\* ssid = "JoakoMi8";

const char\* password = "qqqqqqqq";

\*/

const char\* PARAM\_INPUT\_1 = "opcion";

const char\* PARAM\_INPUT\_2 = "eliminar";

AsyncWebServer server(80);

const int cerradura = 15;

String processor(const String& var) {

//Serial.println(var);

if (var == "BUTTONPLACEHOLDER") {

String buttons = "";

String estadoPuerta = "Estado: ";

if (estaAbierto == LOW) {

estadoPuerta = "Estado: Cerrado";

}

else if (estaAbierto == HIGH) {

estadoPuerta = "Estado: Abierto";

}

buttons += "<button class=\"button button3\" onclick=\"toggle()\" >" + estadoPuerta + "</button>";

return buttons;

}

if (var == "FAVICON") {

String favicon = "<div id=\"topright\"> <img src=\"https://i.ibb.co/kGJk3RK/CRABLock-logo.png\" height=100> </div>";

return favicon;

}

if (var == "MOSTRARUSUARIOS") {

String alert = "<br>";

String alertUsers = "";

for (int i = 0; i < cantUsuarios; i++) {

alertUsers = Users[i];

if (alertUsers.length() > 5) {

alertUsers.toUpperCase();

alertUsers.replace("\"", "");

alertUsers.replace(",", ", ");

alertUsers.replace(":", ": ");

alertUsers.replace("{", "");

alertUsers.replace("}", "");

alert += "<a>" + alertUsers + " &nbsp; </a>";

alert += "<a style=\"color:#FF0000; font-size: 20px;\", onclick=\"elimUser(";

alert += i + 1;

alert += " )\"> x </a>";

alert += "<br>";

alert += "<br>";

}

}

return alert;

}

if (var == "MOSTRARLOGS") {

String alert = "<br>";

String log\_line = "";

for (int i = 0; i < 15; i++) {

log\_line = get\_log\_line(i);

alert += "<a>" + log\_line + " &nbsp; </a>";

alert += "<br>";

}

return alert;

}

return String();

}

void setup() {

/\*Users[0] = "{\"id\":1,\"pass\":\"1234\",\"UID\":[211,240,171,002]}";

Users[1] = "{\"id\":2,\"pass\":\"4321\",\"UID\":[249,116,037,179]}";

Users[2] = "{\"id\":3,\"pass\":\"1590\",\"UID\":[014,200,190,141]}";

Users[3] = "{\"id\":4,\"pass\":\"1234\",\"UID\":[211,240,171,002]}";

Users[4] = "{\"id\":5,\"pass\":\"4321\",\"UID\":[249,116,037,179]}";

Users[5] = "{\"id\":6,\"pass\":\"1590\",\"UID\":[014,200,190,141]}";\*/

Serial.begin(9600);

while (!Serial) continue;

if (!SPIFFS.begin()) {

Serial.println("An Error has occurred while mounting SPIFFS");

return;

}

//Cerradura inicia cerrada

pinMode(cerradura, OUTPUT);

digitalWrite(cerradura, LOW);

SPI.begin();

reader.PCD\_Init();

delay(4);

for (byte i = 0; i < 6; i++) {

key.keyByte[i] = 0xFF;

}

if (!SD.begin(5)) {

Serial.println("SD card initialization failed!");

//return;

} else {

// getSdInfo();

Serial.println("Listo");

}

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Connecting to WiFi..");

}

//Mostrar IP

Serial.println(WiFi.localIP());

obtenerUsuarios();

iniciarServerWeb();

timeClient.begin();

timeClient.setTimeOffset(-10800);

showMenu();

}

void loop() {

if (!funcionActiva) {

leerTarjeta();

} else {

agregarUsuario();

}

if (Serial.available()) {

String input = Serial.readString();

int opc = input.toInt();

switch (opc) {

case 0:

showMenu();

break;

case 1:

crearUser();

break;

case 2:

showUsers();

break;

case 3:

delUser();

break;

case 4:

listAllFiles();

break;

default:

Serial.println("Opcion incorrecta");

break;

}

}

delay(50);

}

void showMenu() {

Serial.println("Opciones:");

Serial.println("0-Mostrar Menu");

Serial.println("1-Agregar usuario");

Serial.println("2-Mostrar usuarios");

Serial.println("3-Eliminar usuario");

}

void crearUser() {

Serial.println("Agregar usuario:");

while (!Serial.available()) {

}

if (Serial.available()) {

String input = Serial.readString();

if (input.length() > 15) {

saveString(input);

Users[cantUsuarios] = input;

String idUser = cantUsuarios + "";

Serial.println("Usuario " + idUser + " agregado");

cantUsuarios++;

}

}

}

void showUsers() {

Serial.println("Usuarios:");

for (int i = 0; i < cantUsuarios; i++) {

if (Users[i].length() > 15) {

Serial.println(i + Users[i]);

}

}

}

void delUser() {

Serial.println("Eliminar usuario");

Serial.println("Introduzca Id del usuario a eliminar:");

while (!Serial.available()) {}

if (Serial.available()) {

String input = Serial.readString();

int delId = input.toInt();

if (delId > cantUsuarios) {

Serial.println("Ese ID no existe");

showMenu();

} else {

eliminarUsuario(delId);

}

}

}

void eliminarUsuario(int delId) {

for (int i = delId; i < cantUsuarios; i++) {

Users[i] = Users[i + 1];

for (int x = 0; x < 4; x++) {

Usuarios[i][x] = Usuarios[i + 1][x];

}

}

cantUsuarios--;

saveUsers();

Serial.println("Usuario eliminado");

}

void listAllFiles() {

File root = SPIFFS.open("/");

File file = root.openNextFile();

while (file) {

Serial.print("FILE: ");

Serial.println(file.name());

file = root.openNextFile();

}

}

void iniciarServerWeb() {

server.on("/", HTTP\_GET, [](AsyncWebServerRequest \* request) {

request->send(SPIFFS, "/index.html", String(), false, processor);

});

server.on("/style.css", HTTP\_GET, [](AsyncWebServerRequest \* request) {

request->send(SPIFFS, "/style.css", "text/css");

});

server.on("/main-page.html", HTTP\_GET, [](AsyncWebServerRequest \* request) {

request->send(SPIFFS, "/main-page.html", String(), false, processor);

});

server.on("/main-page.css", HTTP\_GET, [](AsyncWebServerRequest \* request) {

request->send(SPIFFS, "/main-page.css", "text/css");

});

server.on("/logs.txt", HTTP\_GET, [](AsyncWebServerRequest \*request){

request->send(SD, "/logs.txt", "text/css");

});

//////////////////////////////////////////////////////

//Interpretacion HTTP

// Send a GET request to <ESP\_IP>/update?state=<inputMessage>

server.on("/update", HTTP\_GET, [] (AsyncWebServerRequest \* request) {

String inputMessage;

String inputParam;

// GET input1 value on <ESP\_IP>/update?state=<inputMessage>

if (request->hasParam(PARAM\_INPUT\_1)) {

inputMessage = request->getParam(PARAM\_INPUT\_1)->value();

inputParam = PARAM\_INPUT\_1;

if (inputMessage.toInt() == 0) {

Serial.println("Cerrar");

estaAbierto = LOW;

digitalWrite(cerradura, estaAbierto);

} else if (inputMessage.toInt() == 1) {

Serial.println("Abrir");

estaAbierto = HIGH;

digitalWrite(cerradura, estaAbierto);

}

else if (inputMessage.toInt() == 2) {

Serial.print("Toggle - ");

estaAbierto = !estaAbierto;

digitalWrite(cerradura, estaAbierto);

if (estaAbierto) {

Serial.println("Abierto");

} else {

Serial.println("Cerrado");

}

}

else if (inputMessage.toInt() == 3) {

Serial.println("Agregar Usuario");

agregarUsuario();

}

else if (inputMessage.toInt() == 4) {

Serial.println("Descargar logs");

//download\_logs(request);

AsyncWebServerResponse \*response = request->beginResponse(SPIFFS, "/users.txt", String(), true);

response->addHeader("Content-Disposition","attachment; filename=\"users.txt\"");

request->send(response);

//request->send(200, "text/plain", "OK");

}

}

else if (request->hasParam(PARAM\_INPUT\_2)) {

inputMessage = request->getParam(PARAM\_INPUT\_2)->value();

inputParam = PARAM\_INPUT\_2;

int inputId = inputMessage.toInt() - 1;

Serial.println("Eliminar usuario " + inputId);

eliminarUsuario(inputId);

}

else {

inputMessage = "No message sent";

inputParam = "none";

}

//Serial.println(inputMessage);

request->send(200, "text/plain", "OK");

});

// Start server

server.begin();

}

String getDate() {

String formattedDate;

String dayStamp;

String timeStamp;

String date\_time;

while (!timeClient.update()) {

timeClient.forceUpdate();

}

formattedDate = timeClient.getFormattedDate();

// Extract date

int splitT = formattedDate.indexOf("T");

dayStamp = formattedDate.substring(0, splitT);

// Extract time

timeStamp = formattedDate.substring(splitT + 1, formattedDate.length() - 1);

date\_time = dayStamp + " " + timeStamp;

return date\_time;

}

void saveLog(String toSave) {

File log\_file;

if (SD.exists("/logs.txt")) {

log\_file = SD.open("/logs.txt", FILE\_APPEND);

} else {

log\_file = SD.open("/logs.txt", FILE\_WRITE);

}

String date\_time = getDate();

String log\_line = date\_time + " " + toSave;

Serial.println(log\_line);

if (log\_file) {

delay(50);

log\_file.println(log\_line);

delay(50);

log\_file.close();

Serial.println("log guardado en logs.txt");

} else {

Serial.println("error abriendo logs.txt");

}

}

String get\_log\_line(int line) {

File log\_file = SD.open("/logs.txt");

int count = 0;

if (log\_file) {

while (log\_file.available()) {

String log\_line = log\_file.readStringUntil('\r');

if (count == line) {

return log\_line;

}

count++;

}

return "";

}

else {

return "Error abriendo logs.txt";

}

}

void download\_logs(AsyncWebServerRequest \* request) {

File download = SD.open("/logs.txt");

if (download) {

/\* AsyncWebServerResponse \*response = request->beginResponse(SD, download, String(), true);

response->addHeader("Server", "ESP Async Web Server");

request->send(response);\*/

download.close();

} else Serial.println("error abriendo logs.txt");

}

void noAutorizado(byte lectura[4]) {

Serial.println("no estas autorizado");

String tarjeta;

for (int i = 0; i < 4; i++) {

tarjeta += String(lectura[i], HEX) + ":";

}

tarjeta.toUpperCase();

String toSave = tarjeta + " Acceso denegado";

saveLog(toSave);

}

void usuario (byte lectura[4]) {

Serial.println("Bienvenido Usuario");

Serial.println("abierto");

estaAbierto = true;

digitalWrite(cerradura, estaAbierto);

//delay(2000);

String tarjeta;

for (int i = 0; i < 4; i++) {

tarjeta += String(lectura[i], HEX) + ":";

}

tarjeta.toUpperCase();

String toSave = tarjeta + " Acceso autorizado";

saveLog(toSave);

}

void admin() {

Serial.println("Bienvenido administrador ");

Serial.println("abierto");

estaAbierto = true;

digitalWrite(cerradura, estaAbierto);

}

void dentro () {

Serial.println("saliendo ");

estaAbierto = true;

digitalWrite(cerradura, estaAbierto);

//delay(2000);

}

void leerTarjeta() {

if (!reader.PICC\_IsNewCardPresent())

return;

if (!reader.PICC\_ReadCardSerial())

return;

Serial.print("UID:");

for (byte i = 0; i < reader.uid.size; i++) {

if (reader.uid.uidByte[i] < 0x10) {

Serial.print(" 0");

}

else {

Serial.print(" ");

}

Serial.print(reader.uid.uidByte[i], HEX);

LecturaUID[i] = reader.uid.uidByte[i];

}

Serial.print("\t");

bool accesoAutorizado = false;

for (int i = 0; i < cantUsuarios; i++) {

if (comparaUID(LecturaUID, Usuarios[i])) {

accesoAutorizado = true;

}

}

if (accesoAutorizado) {

usuario(LecturaUID);

} else {

noAutorizado(LecturaUID);

}

// Halt PICC

reader.PICC\_HaltA();

// Stop encryption on PCD

reader.PCD\_StopCrypto1();

}

boolean comparaUID(byte lectura[], byte usuario[])

{

for (byte i = 0; i < reader.uid.size; i++) {

if (lectura[i] != usuario[i])

return (false);

}

return (true);

}

void obtenerUsuarios() {

myFile = SPIFFS.open("/users.txt");

int count = 0;

//Obtener usuarios guardados en la SD

if (myFile) {

while (myFile.available()) {

String list = myFile.readStringUntil('\r');

//Serial.println(list);

list.trim();

if (list.length() > 5) {

Users[count] = list;

count++;

}

}

myFile.close();

} else {

Serial.println("Error abriendo el archivo cards.txt");

}

if (count > 0) {

cantUsuarios = count;

}

else {

cantUsuarios = 0;

}

//Ajustar cantidad de usuarios obtenidos e imprimir usuarios

int usuariosVacios = 0;

//Serial.println("Usuarios:");

for (int i = 0; i < cantUsuarios; i++) {

if (Users[i].length() > 5) {

//Serial.println(Users[i]);

parseJson(Users[i], i);

} else {

usuariosVacios++;

}

}

cantUsuarios = cantUsuarios - usuariosVacios;

}

void agregarUsuario() {

if (!funcionActiva) {

Serial.println("Agregar Usuario");

Serial.println("Acerque la tarjeta a ingresar");

}

funcionActiva = true;

delay(50);

if (!reader.PICC\_IsNewCardPresent())

return;

if (!reader.PICC\_ReadCardSerial())

return;

Serial.print("UID:");

for (byte i = 0; i < reader.uid.size; i++) {

if (reader.uid.uidByte[i] < 0x10) {

Serial.print(" 0");

}

else {

Serial.print(" ");

}

Serial.print(reader.uid.uidByte[i], HEX);

LecturaUID[i] = reader.uid.uidByte[i];

}

// Halt PICC

reader.PICC\_HaltA();

// Stop encryption on PCD

reader.PCD\_StopCrypto1();

Serial.print("\t");

bool userEncontrado = false;

for (int i = 0; i < cantUsuarios; i++) {

if (comparaUID(LecturaUID, Usuarios[i])) {

userEncontrado = true;

}

}

if (userEncontrado) {

Serial.println("La tarjeta ya esta en el sistema");

} else {

for (int i = 0; i < 4; i++) {

Usuarios[cantUsuarios][i] = LecturaUID[i];

}

guardarUsuario(cantUsuarios, "0000", LecturaUID);

cantUsuarios++;

}

funcionActiva = false;

}

void guardarUsuario(int jsonId, String jsonPass, byte jsonUID[4]) {

myFile = SPIFFS.open("/users.txt", FILE\_APPEND);

StaticJsonDocument<200> doc;

String jsonString;

doc["id"] = jsonId;

doc["pass"] = jsonPass;

JsonArray data = doc.createNestedArray("UID");

for (int i = 0; i < 4; i++) {

data.add(jsonUID[i]);

}

serializeJson(doc, jsonString);

Serial.println("Usuario a guardar:");

Serial.println(jsonString);

Users[cantUsuarios] = jsonString;

if (myFile) {

delay(50);

myFile.println(jsonString);

Serial.println("Guardando usuario en cards.txt...");

delay(50);

myFile.close();

Serial.println("Guardado.");

} else {

Serial.println("Error abriendo cards.txt");

}

}

void parseJson(String jsonToParse, int userID) {

StaticJsonDocument<200> doc;

//Deserializar Json

DeserializationError error = deserializeJson(doc, jsonToParse);

//Comprobar errores en la deserializacion

if (error) {

/\*Serial.println("");

Serial.print(F("deserializeJson() failed: "));

Serial.println(error.f\_str());\*/

return;

}

//Guardar usuario

int jsonId = doc["id"];

String jsonPass = doc["pass"];

for (int i = 0; i < 4; i++) {

Usuarios[userID][i] = doc["UID"][i];

}

}

void saveUsers() {

if (SPIFFS.exists("/users.txt")) {

if (SPIFFS.exists("/users\_old.txt")) {

SPIFFS.remove("/users\_old.txt");

}

SPIFFS.rename("/users.txt", "/users\_old.txt");

}

for (int i = 0; i < cantUsuarios; i++ ) {

if (Users[i].length() > 5) {

saveString(Users[i]);

}

}

}

void saveString(String datos) {

if (SPIFFS.exists("/users.txt")) {

myFile = SPIFFS.open("/users.txt", FILE\_APPEND);

} else {

myFile = SPIFFS.open("/users.txt", FILE\_WRITE);

}

StaticJsonDocument<200> doc;

if (myFile) {

delay(50);

myFile.println(datos);

//Serial.println("Guardando usuario en users.txt...");

delay(50);

myFile.close();

//Serial.println("Guardado.");

} else {

Serial.println("Error abriendo users.txt");

}

}

# Bibliografia:

## Frecuencia:

Cantidad de crestas que pasan por un determinado punto por unidad de tiempo. Se mide en Hertz, símbolo Hz.

## Longitud de onda:

Distancia entre crestas, valles o parte idéntica sucesiva en la onda, se expresa con la letra griega lambda λ. Son inversamente proporcionales, mientras mayor f menor λ.

## Frecuencia y longitud de onda:

La frecuencia y la longitud de onda son dos fenómenos encontrados en la mecánica ondulatoria. La frecuencia de una oscilación describe qué tan "frecuente" es el evento. La longitud de onda de una onda describe la longitud de una onda unitaria.

## Principio de Huygens:

Los frentes de las ondas luminosas que se propagan desde una fuente puntual se pueden considerar como crestas encimadas de ondas secundarias distintas.

## Difracción:

La desviación de la luz de la luz que pasa en torno a un obstáculo o a través de una rendija delgada, haciendo que se esparza.

## Interferencia constructiva:

Cuando las ondas están en desfase.

## Definición de luz:

Luz es una radiación electromagnética y generalmente nos referimos a la que nos es visible. La luz se transmite en forma de ondas cuyo reflejo ilumina las superficies permitiéndonos, de esa manera, ser los objetos y los colores a nuestro alrededor.

## Luz IR:

Es un tipo de radiación electromagnética, de mayor longitud de onda que la luz visible, pero menor que la de las microondas.

## Láser:

Dispositivo óptico que genera un haz luminoso de una sola frecuencia, monocromática, coherente y muy intenso.

## Ángulo crítico:

La luz es una propagación de radiación de radio electromagnética en forma de una onda.

## Como onda:

La luz es una propagación de radiación electromagnética en forma de onda.

## Luz como partícula:

La luz, como todas las radiaciones electromagnéticas, está formada por partículas elementales desprovistas de masa denominadas fotones.

## Átomo n:

Un átomo neutro es aquél que carece de carga eléctrica debido a una compensación entre el número de sus protones y electrones.

## Átomo p:

El átomo es la unidad constituyente más pequeña de la materia que tiene las propiedades de un elemento químico. Cada sólido, líquido, gas y plasma se compone de átomos neutros o ionizados Los electrones son partículas con carga negativa que conforman una parte de un átomo

## 

## Átomo e:

El átomo es la unidad constituyente más pequeña de la materia que tiene las propiedades de un elemento químico. Cada sólido, líquido, gas y plasma se compone de átomos neutros o ionizados Los electrones son partículas con carga negativa que conforman una parte de un átomo

## Fotón:

Un fotón es el quantum de energía en forma de radiación electromagnética, emitido o absorbido por la materia.

## 

## Datasheets

### NodeMCU 32s

<https://docs.ai-thinker.com/_media/esp32/docs/nodemcu-32s_product_specification.pdf>

### ESP32

<https://html.alldatasheet.com/html-pdf/1148026/ESPRESSIF/ESP32-WROOM-32/4633/8/ESP32-WROOM-32.html>

### RFID

<https://www.alldatasheet.com/datasheet-pdf/pdf/227839/NXP/MFRC522.html>

### Modulo SD

<http://datalogger.pbworks.com/w/file/fetch/89507207/Datalogger%20-%20SD%20Memory%20Reader%20Datasheet.pdf>

### Relay

<https://pdf1.alldatasheet.com/datasheet-pdf/view/1131944/SONGLERELAY/SRD05VDCSLC.html>