

FORMACIÓN EN AMBIENTE DE TRABAJO II

SAERTA

(Sistema de Asistencia Electrónica y Registro
de Tiempo Automatizado)

PROFESOR: PESOA DARÍO

ALUMNOS: MANCUELLO JOSHUA,
GONZÁLEZ TOBÍAS

CURSO: 7MO AÑO C.S.T ELECTRÓNICA



AGRADECIMIENTOS

Nos gustaría expresar nuestro más sincero agradecimiento a todas las personas que han contribuido al éxito de este proyecto, brindando su apoyo, conocimientos y recursos:

Agradecemos a los Profesores Gonzalez Matías y Gonzalez Julián por su invaluable apoyo desde el inicio de este proyecto. Han compartido con nosotros ideas innovadoras, nos han guiado en el desarrollo de la programación y han contribuido con su experiencia en el diseño en 3D. Su colaboración ha sido fundamental para lograr el resultado final.

Expresamos nuestro agradecimiento a la Profesora Dávila por su invaluable ayuda y orientación en los primeros pasos del proyecto. Sus conocimientos y guía nos permitieron tener una idea clara de cómo crear el sistema y sentar las bases para el desarrollo posterior.

Agradecemos sinceramente al Profesor Alfonso por su generosidad al brindarnos acceso a la impresora 3D de su taller. Su apoyo ha sido fundamental para la fabricación de las piezas necesarias en nuestro proyecto.

Queremos agradecer al Profesor Darío Pesoa por su generosidad al poner a disposición su impresora 3D para la fabricación de las piezas necesarias en nuestro proyecto. Su apoyo ha sido fundamental para hacer realidad la construcción de la carcasa y otros componentes del dispositivo.

Queremos expresar nuestro agradecimiento al Profesor Riquelme por su disposición y por brindarnos acceso a las herramientas del taller. Su apoyo ha sido invaluable para llevar a cabo las etapas de prototipado y montaje del proyecto.

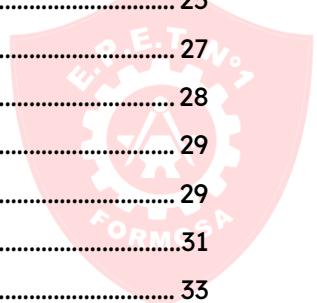
Además, queremos extender nuestro agradecimiento a todos los miembros de nuestra institución, la EPET N°1, por su apoyo y motivación constante durante todo el proceso. Su respaldo ha sido fundamental para superar desafíos y alcanzar nuestros objetivos.

Estamos sinceramente agradecidos por el apoyo recibido de todas estas personas y reconocemos la importancia de su contribución en nuestro proyecto. Sin su ayuda, no hubiéramos logrado los resultados que hoy presentamos.

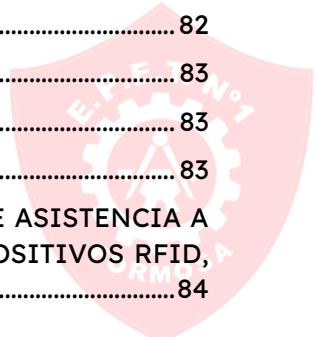


Tabla de Contenido

AGRADECIMIENTOS	2
1. INTRODUCCIÓN	6
2. FUNDAMENTACIÓN	6
3. OBJETIVOS.....	7
GENERAL	7
ESPECÍFICOS	7
4. DESTINATARIO	7
5. MARCO TEÓRICO.....	8
LEY DE OHM.....	8
FÓRMULA.....	8
ONDAS	8
ESPECTRO ELECTROMAGNÉTICO.....	10
TECNOLOGÍA RFID	11
Lector RFID.....	11
Tarjetas RFID.....	13
Chip RFID.....	14
Antena	14
Sustrato.....	14
REGISTROS	15
VENTAJAS DE USAR RFID.....	17
CLORURO FÉRRICO	18
Usos del cloruro férrico	19
Medidas de seguridad para manipular cloruro férrico	20
6. FUNCIONAMIENTO	20
MENÚ PRINCIPAL.....	22
MENÚ ASISTENCIA	22
CURSO.....	23
TURNO	23
ARCHIVOS Y SD.....	23
MODO AHORRO.....	24
7. DIAGRAMAS.....	24
Diagrama de Gantt (2022).....	24
Diagrama de Gantt (2023)	25
Red de Pert	27
Diagrama de Flujo.....	28
DIAGRAMA ELÉCTRICO	29
CIRCUITO ELECTRÓNICO	29
ÁRBOL DE FALLAS.....	31
8. DISEÑO.....	33



DISEÑO 3D.....	33
Vista anterior	33
Vista lateral izquierda.....	33
Vista lateral derecha.....	33
Vista posterior	34
Vista interior.....	34
Vista superior.....	34
DISEÑO DEL CIRCUITO.....	35
PCB Layout.....	35
Lado de pistas	35
Lado de componentes	35
9. PROCESO	36
Investigación.....	36
Programación.....	38
Diseño 3D.....	39
Diseño del circuito.....	39
Fabricación de PCB	41
Impresiones 3D.....	43
Armado del circuito	44
Fabricación del mecanismo de seguridad	46
Instalación	47
10. LISTA DE MATERIALES	49
11. HERRAMIENTAS DIGITALES	50
IDE Arduino.....	50
SketchUp.....	50
Ultimaker Cura.....	52
12. CÓDIGO FUENTE.....	53
13. CONCLUSIONES.....	81
Resumen	81
Logros Alcanzados.....	81
14. ANEXO	82
DISEÑO (PRIMER PROTOTIPO 2022).....	82
VISTA LATERAL DERECHA.....	82
VISTA LATERAL IZQUIERDA.....	82
VISTA ANTERIOR	83
VISTA POSTERIOR	83
VISTA INTERIOR	83
● SISTEMA INFORMÁTICO PARA AUTOMATIZAR EL CONTROL DE ASISTENCIA A CLASES DE DOCENTES Y ESTUDIANTES DE LA ESPOCH CON DISPOSITIVOS RFID, APLICANDO LA METODOLOGÍA SCRUMBAN.....	84



● SISTEMA PARA EL CONTROL DE ASISTENCIA, SOLICITUD-ENTREGA Y RECEPCIÓN DE EQUIPOS DEL LABORATORIO DE ELECTRÓNICA DE LA UNIDAD CENTRAL DEL VALLE DEL CAUCA, TULUÁ-VALLE, COLOMBIA :.....	85
COMPONENTES UTILIZADOS	86
Lectores RFID	86
Modulo Reloj Tiempo Real.....	87
Modulo Lector SD	88
Bibliografía	90



1. INTRODUCCIÓN

El proyecto “Sistema de Asistencia Electrónica y Registro de Tiempo Automatizado” o bien SAERTA, consiste en realizar la asistencia diaria de los alumnos del taller de electrónica, tanto para la entrada como para la salida, a través de un sistema basado en Arduino que utiliza la lectura de tarjetas RFID, cuyo archivo es almacenado en una tarjeta SD.

2. FUNDAMENTACIÓN

En la E.P.E.T N°1 y en la mayoría de establecimientos educativos, el sistema de asistencia de los alumnos se realiza de forma manual mediante un profesor o preceptor, donde por medio de la lectura de las planillas de cada curso se registran los presentes de cada día. Sin embargo, existe ineficiencia e ineficacia en este método de asistencia, ya que la mayoría de veces este procedimiento se realiza sólo una vez en cada turno en un momento determinado y esto lleva a no tener un control preciso en el caso de los alumnos que llegan fuera de las horas reglamentadas, o que se retiren sin autorización.

Este proyecto busca aumentar la eficiencia y la eficacia del proceso de registro de asistencia de los alumnos: el uso de tarjetas RFID y la plataforma de hardware y software de Arduino permitirá agilizar y automatizar el proceso de registro de asistencia de los alumnos, reduciendo el tiempo y esfuerzo requeridos por el personal encargado de hacerlo manualmente.

Además, busca mejorar la seguridad dentro del taller, permitiendo tener un registro detallado de quiénes ingresaron y salieron del lugar, y detectar el horario en que lo hicieron, lo que puede ser útil en caso de emergencias o situaciones imprevistas.



3. OBJETIVOS

GENERAL

Desarrollar un sistema de asistencia electrónica que permita registrar y monitorear la asistencia de los alumnos del taller de Electrónica de la E.P.E.T. N°1, mediante el uso de tarjetas RFID y la plataforma de hardware y software de Arduino.

ESPECÍFICOS

- ✓ Diseñar un modelo de sistema de identificación por tarjeta para controlar la asistencia en el taller de electrónica de la E.P.E.T N°1.
- ✓ Programar el microcontrolador Arduino para recibir los datos de las tarjetas RFID y almacenarlos en una base de datos, dentro de una tarjeta SD.
- ✓ Realizar pruebas y ajustes en el sistema para garantizar su correcto funcionamiento y fiabilidad.
- ✓ Documentar el proceso de diseño, desarrollo y pruebas del sistema, incluyendo su instalación y mantenimiento.
- ✓ Implementar el sistema de asistencia electrónica en clases reales y evaluar su desempeño en términos de eficacia y eficiencia.
- ✓ Identificar posibles mejoras al sistema y proponer nuevas funcionalidades para su futura evolución.
- ✓ Contribuir a la automatización del aula del taller de electrónica.

4. DESTINATARIO

La implementación de un sistema de asistencia mediante el uso de tarjetas está dirigida para los alumnos de 4to a 7mo de la especialidad Electrónica de la E.P.E.T N°1, de forma que todo los alumnos, docentes y preceptores del taller se beneficien por este servicio que mejora el sistema de asistencia.

La aplicación del dispositivo, ayudará a realizar un control más riguroso sobre la presencia de las personas que concurren al establecimiento. Esto brindará más seguridad tanto a los alumnos, como a directivos y docentes.

Aunque también es factible fabricar el mismo dispositivo y adaptarlo para los talleres de las otras especialidades de la institución.



5. MARCO TEÓRICO

LEY DE OHM

La ley de Ohm, postulada por el físico y matemático alemán [Georg Simon Ohm](#), es una ley básica para entender los fundamentos principales de los [circuitos eléctricos](#) que describe la correlación de los parámetros de resistencia, corriente, [voltaje](#) con los que varían, a través del siguiente enunciado: “la intensidad de corriente que atraviesa un circuito es directamente proporcional al voltaje o tensión del mismo e inversamente proporcional a la resistencia que presenta”.

FÓRMULA

En términos matemáticos la ley de Ohm se aplica mediante la ecuación:

$$I = V/R$$

Dónde:

- I es la [intensidad de una corriente eléctrica](#) que atraviesa un conductor expresada en Amperios (A).
- V es el voltaje. Por voltaje en cambio nos referimos a la diferencia de potencial entre un punto con respecto a otro expresado en Voltios (V).
- R es la [resistencia eléctrica](#). Por resistencia nos referimos al obstáculo que la corriente encuentra en su camino, cuanto más alto sea, más difícil será que la corriente lo atraviese. La unidad de medida de la resistencia son los ohmios, simbolizados por la letra griega omega (Ω).

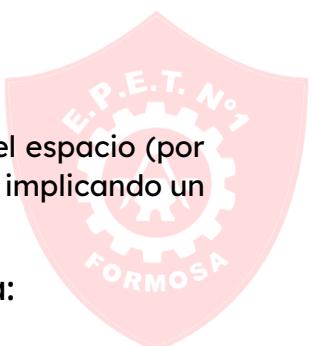
La [intensidad de corriente](#) de las [cargas eléctricas](#) es directamente proporcional al voltaje aplicado e inversamente proporcional a la resistencia.

Para el voltaje, en cambio, cuanto mayor sea, mayor será la fuerza de atracción que genera para mover las cargas, por lo que para el mismo valor resistivo será directamente proporcional a la corriente.

ONDAS

Onda: Propagación de una perturbación de alguna propiedad del espacio (por ejemplo, densidad, presión, campo eléctrico o campo magnético) implicando un transporte de energía sin transporte de materia.

Una función es una onda si cumple la ecuación de onda:



$$\nabla^2 \psi(\vec{r}, t) = \frac{1}{v^2} \frac{\partial^2 \psi}{\partial t^2}(\vec{r}, t)$$

Pero habitualmente se usa una ecuación más sencilla:

$$Onda(t) = A \operatorname{sen}(\omega t + \phi)$$

Parámetros de una onda: $y(t) = A \operatorname{sen}(\omega t + \phi)$

A = Amplitud

ω = Frecuencia angular

ϕ = Fase

f = Frecuencia $\rightarrow 2\pi f = \omega$

T = Período $\rightarrow T = 1/f$

λ = Longitud de onda $\rightarrow \lambda = c/f$

c = velocidad de la luz = 300.000.000 m/s

Se pueden clasificar en 2 grupos:

- Ondas Mecánicas
- Ondas Electromagnéticas (EM)

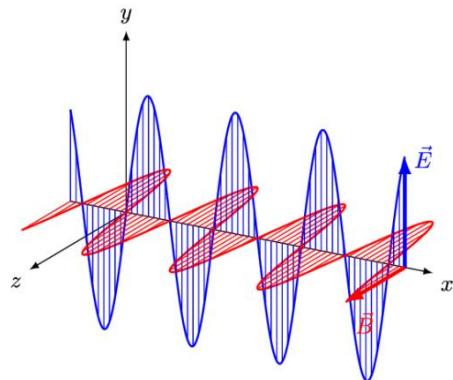
Las ondas mecánicas se definen como una oscilación de materia (posición, velocidad y energía de sus átomos) que propaga una perturbación en las propiedades mecánicas de un medio material. Al ser una oscilación en la materia, las ondas mecánicas no se pueden producir en el vacío. (un ejemplo de este tipo de ondas es el sonido que se propaga a través del aire)



Las ondas electromagnéticas son las que presentan una oscilación en el campo electromagnético. Como su nombre lo dice, están compuestas por dos ondas (una Eléctrica y una Magnética), ambas tienen un comportamiento similar, por lo que no es necesario describir ni graficar las dos. Se pueden propagar a través

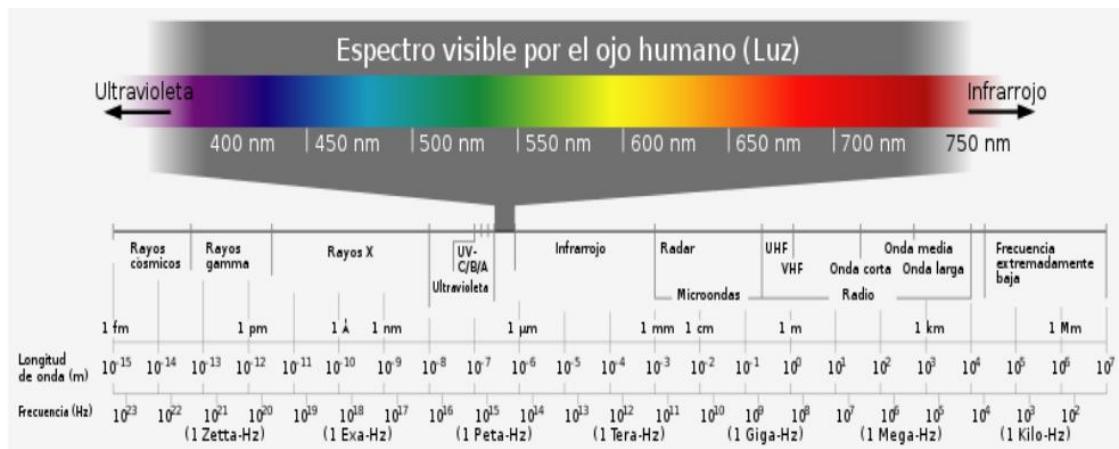


del vacío; no requieren un medio para propagarse, aunque propagarse a través de uno.



ESPECTRO ELECTROMAGNÉTICO

El espectro electromagnético es el conjunto de longitudes de onda de todas las radiaciones electromagnéticas. Dicho en otras palabras, es una forma de representar todas las ondas EM de manera ordenada, acomodándolas en función de su frecuencia. Incluye:



Los rayos gamma, tienen las longitudes de onda más cortas y las frecuencias más altas conocidas. Son ondas de alta energía capaces de viajar a larga distancia a través del aire y son las más penetrantes.

Los rayos X que tienen longitudes de onda más largas que los rayos gamma, pero menores que la radiación ultravioleta y por lo tanto su energía es mayor que la de estos últimos. Se utilizan en diversas aplicaciones científicas e industriales, pero principalmente se utilizan en la medicina como la radiografía. Consisten en una forma

de radiación ionizante y como tal pueden ser peligrosos. Los rayos X son emitidos por electrones del exterior del núcleo, mientras que los rayos gamma son emitidos por el núcleo.

La radiación ultravioleta (UV) se define como la porción del espectro electromagnético que se encuentra entre los rayos X y la luz visible. Para más información haga clic aquí.

La luz visible —también espectro visible— es la parte de espectro electromagnético que los ojos humanos son capaces de detectar. Cubre todos los colores del azul a 400 nm al rojo a 700 nm. La luz azul contiene más energía que la roja.

La radiación infrarroja (IR) —también radiación térmica— es la parte del espectro electromagnético que se encuentra entre la luz visible y las microondas. La fuente natural más importante de radiación infrarroja es el Sol.

Las ondas radioeléctricas tienen longitudes de onda largas que varían unos pocos centímetros a miles de kilómetros de longitud. Sus principales usos son en la televisión, los teléfonos móviles y las comunicaciones por radio

TECNOLOGÍA RFID

La identificación por radiofrecuencia o RFID por sus siglas en inglés (Radio Frequency Identification), es una tecnología de identificación remota e inalámbrica en la cual un dispositivo lector es vinculado a un equipo de cómputo y se comunica a través de una antena con un transponder (también conocido como tag o etiqueta) mediante ondas de radio.

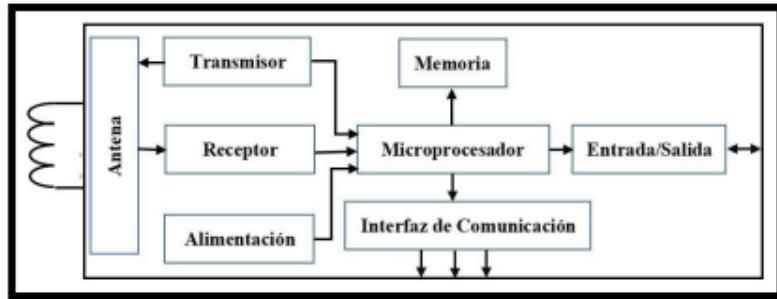
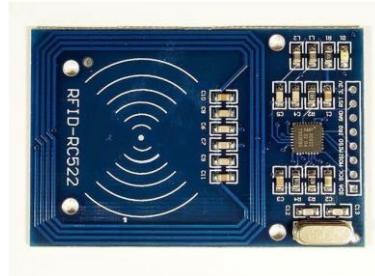
Las etiquetas o tag RFID consisten en un circuito integrado, que va adjunto a una antena de radio, el cual transmite los datos almacenados en la etiqueta o tag.

Es una forma de comunicación inalámbrica que incorpora el uso de acoplamiento electromagnético o electrostático en la porción de radiofrecuencia del espectro electromagnético para identificar de manera única un objeto, animal o persona.

Lector RFID

El lector RFID es un dispositivo conectado a la red que puede ser portátil o estar conectado de forma permanente. Utiliza ondas de radio para transmitir señales que activan la etiqueta que se encuentra dentro de la tarjeta.





Al sistema lector lo conforman varios subsistemas que se mencionan a continuación:

- Transmisor
- Receptor
- Microprocesador
- Memoria
- Entradas y salidas
- Controlador
- Interfaz de comunicación
- Fuente de alimentación

Transmisor El transmisor es el encargado de la emisión de ondas electromagnéticas que contengan información codificada, además de la energía necesaria para poner en funcionamiento a las etiquetas, esto a través de antenas.

2.3.1.3.2. Receptor El receptor a través de las antenas se encarga de recibir las señales análogas provenientes de las etiquetas para su posterior procesamiento en el microprocesador.

22 2.3.1.3.3. Microprocesador Dentro del microprocesador se realiza un procesamiento de bajo nivel para la información que se haya recibido por parte del tag, además de la decodificación de los datos así como también de la verificación de errores en la señal captada.

2.3.1.3.4. Memoria A través de la memoria se podrá almacenar datos importantes tales como: parámetros de configuración del sistema lector o códigos de distintas tarjetas que hayan sido leídas por el sistema.

2.3.1.3.5. Entradas y salidas / actuadores Mediante este tipo de sistemas se puede hacer que el lector RFID interactúe con dispositivos externos como: sensores y actuadores, los mismos que permitan implementar

mecanismos de encendido y apagado, evitando que el sistema se encienda únicamente cuando exista una lectura del tag.

Controlador Permite al usuario poder comunicarse con el sistema lector mediante la ayuda de un computador o un programa, en caso de que exista más de un lector en el sistema a través de una interfaz de comunicación.

Interfaz de comunicación Es el medio por el cual se va a realizar la comunicación entre el usuario y el controlador del sistema lector. En el sistema lector se pueden tener distintos tipos de interfaces de comunicación, de las cuales se puede mencionar: RS-232, RS-485, adaptador de red, etc.

Fuente de alimentación Se encarga de proveer la energía eléctrica necesaria para poner en funcionamiento al sistema en general. Dependiendo del tipo de lector (fijo o móvil) se puede tener una fuente de alimentación conectada a un tomacorriente o una batería recargable.

Antenas: Las antenas son los elementos más sensibles dentro de un sistema RFID, ya que son las encargadas de enviar y recibir las señales que contienen la información que se desea procesar. Su tamaño y forma varía, dependiendo de la frecuencia de operación y el área de cobertura que generen, así como también de la aplicación que vayan a tener.

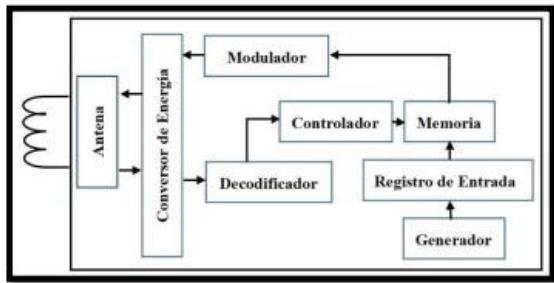
Tarjetas RFID

Son tarjetas sin contacto que contienen un [Chip RFID](#) que almacena todos los datos necesarios. La información se transmite a través de ondas de radio y es recogida por un lector RFID.



El circuito integrado que se encuentra dentro del chip tiene algunas funcionalidades entre las cuales están: la conversión de energía, el control lógico, el almacenamiento y recuperación de datos y la modulación requerida para devolver los datos al lector.





Las ondas de radio generadas por el lector y captadas por la antena son procesadas y pasan a convertirse no solo en peticiones sino también en energía para el resto de unidades que conforman el chip a través de un conversor ADC, el cual provee de energía al resto de dispositivos. El chip contiene además un decodificador y codificador para convertir la señal analógica en digital y viceversa. Posteriormente esta señal va al bloque de control lógico en el cual se realiza un procesamiento de bajo nivel para leer el código único que se encuentra grabado en la memoria y 19 que fue codificado por el fabricante

Tienen tres partes principales:

Chip RFID

Es lo que hace que las tarjetas RFID sean diferentes de las tarjetas de plástico promedio. Es un pequeño chip de computadora que está incrustado en la tarjeta y contiene información sobre su identidad y detalles de la cuenta. No contiene ninguna fuente de energía, por lo que necesita una fuente de alimentación para funcionar.

Antena

Esta es una pequeña bobina de metal incrustada en la tarjeta que recibe y envía ondas de radio. Es un componente crítico que permite que el chip se comunique con un lector.

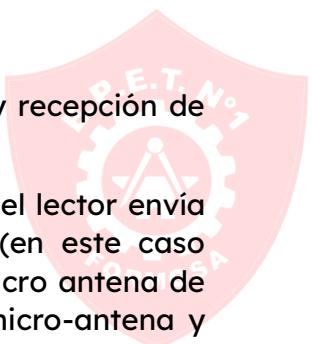
Una vez que recibe una señal de radio del lector, convierte esta información en energía eléctrica. Esto permite que el chip funcione y procese datos.

Sustrato

Esta es la parte de la tarjeta que ves y tocas físicamente. Mantiene el chip y la antena juntos y proporciona soporte estructural. Los sustratos generalmente están hechos de plástico, aunque algunas tarjetas RFID premium usan PVC, PET, ABS, madera y otros materiales resistentes para mayor durabilidad.

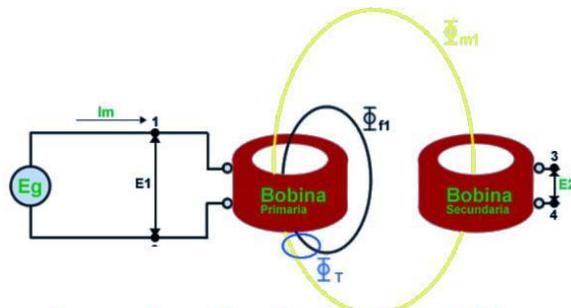
Su principio de funcionamiento está basado en la transmisión y recepción de ondas electromagnéticas:

En el esquema general de funcionamiento de un sistema RFID, el lector envía una serie de ondas de radiofrecuencia a la etiqueta o tag (en este caso almacenada dentro de una tarjeta), que son captadas por la micro antena de ésta. Dichas ondas activan el microchip, que a través de la micro-antena y



mediante ondas de radiofrecuencia, transmite al lector la información que tenga en su memoria. Finalmente, el lector recibe la información que tiene el tag y lo envía a una base de datos en la que previamente se han registrado las características del producto o puede procesarlo según convenga a cada aplicación.

La antena es un tag pasivo que funciona captando la energía de las ondas de radio emitidas por el interrogador (lector) en una determinada frecuencia para la que está especialmente construida, de tal forma que resuena esa frecuencia y capta las ondas de radio solamente en ese espectro



Acoplamiento magnético

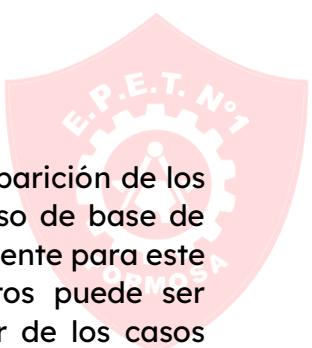
Dependiendo de la frecuencia a la que estemos trabajando la antena será físicamente distinta, debido a que ha de acoplarse a determinadas longitudes de onda. Así, en LF (baja frecuencia) la antena suele estar compuesta por bobinas con un elevado número de espiras realizadas con hilo esmaltado bobinado en anillos o bien sobre núcleos de ferrita, lo que permite unas menores dimensiones físicas de la antena, siendo el acoplamiento básicamente inductivo.

En frecuencias más altas, HF (alta frecuencia) y sobre todo UHF (ultra alta frecuencia), el acoplamiento pasa a ser capacitivo. En HF con acoplamiento inductivo, las antenas están normalmente formadas por pequeñas bobinas de un corto número de espiras laminadas sobre un sustrato plano, normalmente de circuito impreso o plástico flexible, con espiras de láminas de cobre o de aluminio

En frecuencias UHF se trabaja siempre con acoplamiento por el campo eléctrico, con lo que normalmente estamos trabajando con antenas de tipo dipolo realizadas sobre sustratos plásticos en finas láminas de aluminio impreso, o incluso de tinta conductora. Este tipo de antena presenta una determinada direccionalidad y polarización, por lo que a veces se usan varios dipolos o incluso antenas fractales para permitir la radiación con una cierta independencia de la orientación entre la antena del tag y la antena del interrogador.

REGISTROS

La historia de registros se divide en dos etapas, ya que con la aparición de los computadores se pasó a llevar a cabo registros mediante el uso de base de datos, lo que está dejando relegado el uso de papelería estrictamente para este fin, ya que la información depositada en documentos escritos puede ser modificada, registrada de una manera incorrecta o en el peor de los casos



perderse, a continuación se mostrará de una manera más detallada en qué consiste el registro manual y el registro computacional

Registros manuales: Desde la perspectiva de un aula se habla básicamente de libros de registro/asistencia o simplemente planillas, que son los únicos soportes existentes de manera física en donde se registra, valga la redundancia la asistencia de los alumnos.

Registros computacionales: En 1952 IBM presenta su primera cinta magnética para el almacenamiento de datos, conocida como "7 pistas". Este formato ha evolucionado hasta almacenar varios cientos de Megabytes.

Con el paso del tiempo, las unidades de cinta de carrete abierto dieron paso a unidades de cartuchos. Así en 1972 la empresa 3M presenta el QIC (Quarter Inch Cartridge). En la actualidad existen cartuchos de hasta 80m Gigabytes de capacidad 14.

A partir de los años 60, con el apogeo de la invención de la C.P.U, se introdujeron los conceptos de base de datos y de sistemas de gerencia de base de datos en la primera mitad de esta década, en donde los datos se almacenaban en múltiples archivos independientes entre sí sin tener ningún tipo de relación entre ellos y teniendo en cuenta la cantidad de información que se maneja en las organizaciones es cada vez mayor, es por ello que se da la necesidad de implementar un sistema de almacenamiento que pueda almacenar gran volumen de información de forma rápida y se pueda generar consultas ágiles que optimicen los tiempos en los procesos .

En 1970 que se comienza a hablar de base de datos relacionales, todo gracias al trabajo teórico fundamental de Edgar F. Codd, el cual con el objetivo de evitar un conjunto de anomalías y permitir una representación sin problemas se introduce la normalización, pero sin aun darse la adopción generalizada de estas tipo de base de datos. Posteriormente en los años 80 el crecimiento de materiales electrónicos en su inventario, muchos laboratorios se vieron con la necesidad de administrar su información de forma más dinámica, así que digitalizaron la forma de almacenar sus registros ya que a partir del año 1987 se contaban con programas de registro confiables como Microsoft Excel y Microsoft Word, los cuales fueron adoptados de manera más rápida que las base de datos anteriormente mencionadas; de esta manera se sistematizó la forma en cómo se registraba la información.

Hasta la primera década de los 90 donde se encontró en las base de datos relacionales una manera más eficiente del almacenamiento de la información, en donde la gran diferencia entre estas base de datos relacionales y las de archivos independientes se daría por los sistemas de gerencia de base de datos en donde muchas o todas de las tareas importantes de administración y de navegación en los datos son realizadas en forma automática realizadas por estos sistemas.



VENTAJAS DE USAR RFID

Un sistema RFID digital identifica datos en tiempo real sin intervenciones humanas manuales y visuales, agregando ganancias de tiempo, seguridad y eficacia en diversas aplicaciones en el proceso de adquisición de información. Esta tecnología es utilizada para identificar, rastrear y gestionar productos, documentos, objetos, personas, animales – sin contacto y sin la necesidad de campo visual. Proyectos-piloto alrededor del mundo tienen probada su funcionalidad, que puede ser aplicada en seguridad, control de acceso y tráfico en los sectores público, farmacéutico, médico-hospitalaria, de automóviles, comercio de venta al por menor, entre otros. La tecnología RFID tiene amplia aplicación en la gestión de la cadena logística. Además de hacer el control de movimiento y almacenamiento de materiales a lo largo de los procesos logísticos, optimizar tiempo, Reducir costos con distribución y movimiento de materiales, mejorar servicios y la atención al cliente. La funcionalidad de la tecnología RFID facilita la gestión y agrega valor a los productos y servicios. Los colaboradores pueden actuar con enfoque en actividades que agregan mayor valor, lo que tendrá impacto directo en los resultados de la empresa con mejoría en la productividad y atención al consumidor.

La aplicabilidad RFID funciona para almacenar datos de productos; enviar indicadores de humedad, temperatura y, combinada a un GPS, informar la ubicación precisa. Logra leer y grabar datos y la lectura no necesita ser estática y en línea recta como en el código de barras, además de capturar radiofrecuencia de objetos en movimientos no uniformes. La distancia de captura es superior al lector óptico Y con eso logra grabar un código único y no alterable de productos y piezas.

La tecnología RFID asociada a sensores de temperatura posibilita, por ejemplo, informar la temperatura en pequeños intervalos de tiempo. Estos datos pueden ser monitoreados por software y enviar alertas para control de la situación de pacientes en hospitales. El monitoreo de temperaturas también es útil para procesos de fabricación y logísticas de productos. En la industria de alimentos, puede funcionar el monitoreo de la temperatura de los productos perecibles durante la distribución y la entrega, como forma de garantizar condiciones adecuadas de consumo.

Otra aplicación muy común en el sistema, es la identificación de bagajes en los aeropuertos. La etiqueta RFID ubicada en las pertenencias del cliente logra relacionar el dueño del bagaje, su vuelo y su destino. Ese monitoreo ayuda a evitar pérdidas y atrasos, además de aumentar la seguridad y agilidad en caso de operaciones de vuelo y redirecciones. En el campo de la logística, la aplicabilidad funciona para agregar flexibilidad, Inteligencia y seguridad a los procesos. Los principales beneficios de la tecnología RFID en la cadena logística son: Optimización de los procesos de fabricación con el recibimiento automatizado integrado y planeado, el control de la producción sin necesidad de aguardar notificaciones; producción personalizada de productos en las líneas de montaje; automatización, control y abastecimiento de estoques; registro automático de las entradas y salidas de materiales; optimización de chequeo de precios, ítems en falta, devoluciones y fecha de caducidad de los productos; practicidad en la proporción de contenidos ubicados fuera del alcance visual;

Agilidad y reducción de errores en las entregas de pedidos de los clientes; ganancias de tiempo en movimientos y ubicaciones de materiales; captura ágil y simultánea de varias tags con eliminación de los procesos de lectura “uno a uno” de código de barras ; garantía de autenticidad de los productos con grabación de código único no alterable; rastreo de productos, personas, animales y objetos; garantía de captura del tag RFID con grabación y registro de fecha y hora de la lectura.

Agilidad y reducción de mano de obra para hacer el inventario de productos y materiales. Con amplia funcionalidad y múltiples beneficios, la tecnología RFID puede ser un diferencial competitivo en la sustentabilidad de los negocios, pero necesita de soluciones específicas en diferentes situaciones, como faja de radiofrecuencia, alcance, interferencia, barreras a las ondas de radio, compatibilidad de hardware y software, fuentes de energía y estructuras de códigos estandarizados. Algunos factores pueden dificultar el uso, como obstáculos en la propagación de las ondas de radio y metales que afectan en la distancia de la captura, además de líquidos y el propio cuerpo humano, que puede impedir la captura. La aplicabilidad efectiva en canteras y mercados globalizados requiere estandarización de hardware y software, procesos y estructuras de sistemas de información para que exista escala necesaria y compatibilidad de sistemas. De esa forma, para disfrutar de los numerosos beneficios de la tecnología RFID, es importante elaborar un amplio proyecto, considerando todos los procesos involucrados, además de hacer este depurado antes de aplicar las tags en los materiales.

CLORURO FÉRRICO

El cloruro de hierro (III), más conocido por la comunidad técnica como cloruro férrico o tricloruro de hierro, como decíamos, es un componente químico con base de hierro y con fórmula química FeCl_3 . Se origina a partir del mineral molisita, en algunas fumarolas.

Usualmente, el [cloruro de hierro](#) se vende en forma de polvo (aunque también tenemos una versión de [cloruro férrico líquido](#)). Los polvos de cloruro de hierro son de color pardo, aunque cuando se lo ilumina con intensidad puede dar reflejos de color verde oscuro.



Aunque es soluble en disolventes orgánicos, con el agua reacciona exotérmicamente, liberando calor y puede ser sumamente corrosivo. El resultado de esta reacción es utilizado como coagulante en el tratamiento de aguas residuales, potables y para el grabado químico en la industria electrónica. Cuando se disuelve en alcohol se le conoce como tintura de hierro, y cuando entra en contacto con metales, forma hidrógeno gaseoso, que es un compuesto inflamable.



Usos del cloruro férrico

Los usos del cloruro férrico van desde el tratamiento de aguas y desechos industriales hasta la fabricación de placas de circuito impreso. Estos son algunos de los más relevantes:

• Tratamiento de aguas

Como se mencionó anteriormente, la venta de cloruro férrico más extendida es para el tratamiento de aguas residuales y potables domésticas e industriales debido a que con sus eficaces componentes y aditivos remueve materia orgánica, sólidos disueltos, turbiedad, metales pesados y olores de manera eficaz y funcional.

• Electrónica

En el campo de la electrónica, el cloruro férrico es utilizado para la fabricación de placas de circuito impreso en bajas cantidades, ya que este compuesto reacciona con el cobre generando cloruro ferroso y cloruro cúprico.

Las placas de circuito impreso o PCB (por sus siglas en inglés) son soportes físicos en los cuales se instalan componentes electrónicos y eléctricos para formar chips, condensadores, diodos y demás elementos indispensables para la fabricación de computadores, monitores, teléfonos inteligentes, entre muchos otros más artículos tecnológicos.

• Laboratorios

El cloruro férrico también tiene varios usos en los laboratorios químicos como ácido para crear reacciones de catálisis como cloración y reacción de Friedel-Crafts de compuestos aromáticos. Además, se emplea para el control de corrosión, evitar la formación de estruvita en reactores anaerobios, eliminación de flúor y arsénico, como reactivo deshidratante en algunas reacciones y para la remoción de sulfatos y fósforos.

Medidas de seguridad para manipular cloruro férrico

El cloruro férrico o cloruro de hierro es un [producto químico muy utilizado](#) debido a sus diversas propiedades, sin embargo, debe manipularse con ciertas precauciones ya que, al ser una sustancia química corrosiva, puede tener efectos negativos en la salud humana.

Algunas de las afecciones que puede sufrir si se expone al cloruro férrico son graves quemaduras e irritaciones en la piel; irritación en garganta, pulmones y nariz y dificultad para respirar por inhalación de la sustancia; y cambios en la coloración de los ojos por contacto prolongado.

Estas son algunas de las medidas de seguridad que debe tomar para el almacenamiento y manipulación de cloruro férrico:

- Cualquier persona encargada de la manipulación de cloruro férrico o cloruro de hierro debe contar con los elementos de protección personal adecuados, a saber: guantes, máscara y respirador, gafas protectoras (en caso de ser necesario), ropa adecuada y que cubra la mayor parte del cuerpo, etc. Lo ideal será contar con equipo especializado que esté fabricado con los materiales y las normas requeridas para este tipo de trabajo.
- Las personas expuestas a cloruro férrico deben cambiar sus ropas y lavar manos y cara tan pronto como sea posible
- Almacene el cloruro férrico en recipientes bien cerrados, en espacios frescos, bien ventilados y lejos de la luz o materiales combustibles.
- En caso de exposición, utilice el protocolo de primeros auxilios. Para contacto con ojos o piel, retire el equipo de protección y enjuague con abundante agua la zona afectada por un mínimo de 15 minutos. Para contacto por vías respiratorias o inhalación del compuesto, retire a la persona del lugar de exposición, proporcione primeros auxilios (si tiene el entrenamiento para ello) y traslade a la persona inmediatamente a un centro de atención médica.

6. FUNCIONAMIENTO



Para configurar inicialmente el dispositivo, dentro del programa se deben registrar las UID de las tarjetas de cada alumno por separado y asignar esa información a su nombre y apellido.

Para que el dispositivo funcione, la tarjeta SD debe estar colocada en su módulo, de otra manera, aparecerá una pantalla con un mensaje diciendo “TARJETA SD NO INSERTADA” y no nos permitirá utilizar el dispositivo.

El preceptor o el profesor, selecciona el curso y el turno de los alumnos y habilita la entrada mediante su tarjeta (tarjeta maestra). Los datos de esta tarjeta se comparan con los almacenados y de no ser iguales, aparece un mensaje “Tarjeta maestra no encontrada”. Si un alumno intenta habilitar el ingreso con su propia tarjeta, también aparece un mensaje de alerta: “Alumno no autorizado” junto con el apellido de quien lo hizo.

Habilitado el ingreso, los alumnos deberán colocar su tarjeta, para corroborar su presencia. La información obtenida se compara con la que ya está almacenada y, de ser la misma, se guarda el nombre y apellido del presente en una tarjeta SD en formato de documento de texto (.txt).

Al final del día, el preceptor o el profesor deberán habilitar la salida, donde cada alumno pasa su tarjeta y registra su egreso. Se realiza una comparación entre los datos de los alumnos que ingresaron y los que egresaron para verificar que nadie haya salido de la escuela sin autorización.

Una vez que hayan salido todos los alumnos, se debe seleccionar la opción de generar archivo, donde se vuelve a pedir la tarjeta maestra. Esta opción, genera un archivo del curso y lo almacena en la carpeta del día correspondiente dentro de la tarjeta SD.

Para extraer la tarjeta SD y poder leerla, existe una opción llamada EXTRAER SD, en la que se solicita la tarjeta maestra y, de ser correcta, se abre una compuerta que permite sacar la tarjeta del módulo

El módulo RTC DS1302 es utilizado para que el sistema pueda ubicarse temporalmente, es decir, que pueda obtener los datos de fecha y hora exactas. Con estos datos se realiza tanto la creación de los archivos de asistencia como el control en los horarios de dicha asistencia que se ejecuta diariamente.

El control de mando para navegar a través de los diferentes menús del dispositivo, se encuentra en la parte derecha inferior de la cara anterior, el nombre técnico es encoder



- **ENCODER:** Es un dispositivo que detecta la posición del eje generando pulsos digitales. Se utiliza para navegar entre las líneas de cada pantalla del menú.



MENÚ PRINCIPAL

Dentro del menú principal se encontrarán las opciones que presenta el dispositivo. Las acciones que puede realizar la asistencia electrónica son:

- Asistencia
- Opciones SD
- Curso
- Turno
- Modo Ahorro



A continuación, se explicarán en detalle cada uno de los apartados del menú:

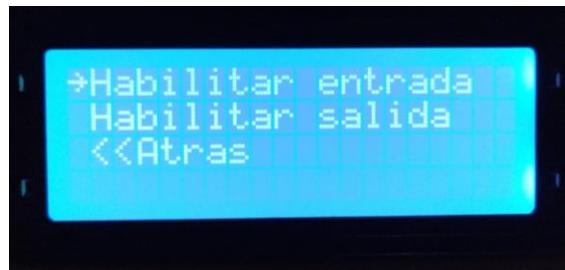
MENÚ ASISTENCIA

Contiene la opción para poder tomar la asistencia del curso, tanto en el horario de entrada como en el de salida. Se necesita la tarjeta del preceptor para acceder a ambas opciones:

- Habilitar ingreso: Permite que los alumnos registren su llegada al aula colocando sus tarjetas
- Habilitar egreso: Permite que los alumnos registren su salida del aula al colocar sus tarjetas.

Si no se ha seleccionado el curso y/o turno de los alumnos a los cuales se le va a realizar la asistencia, el sistema se redirecciona automáticamente al menú de curso o turno, dependiendo del caso.





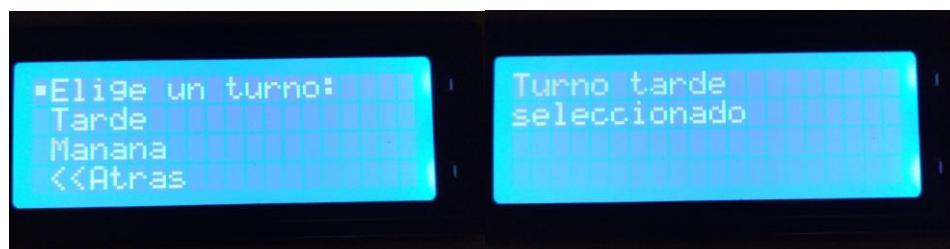
CURSO

Permite seleccionar el curso (7mo, 6to, 5to y 4to) al cual se va a realizar la asistencia y redirecciona automáticamente al menú de Turno



TURNO

Permite seleccionar el turno al cual se va a realizar la asistencia y redirecciona automáticamente al menú Asistencia o al menú Curso si no se ha elegido alguno aún.



ARCHIVOS Y SD

Dentro de las opciones de la memoria SD, se encuentran las siguientes acciones disponibles:

- Extraer SD: Esta opción permite la extracción de la memoria, activando el mecanismo de la compuerta.
- Generar archivo: Con esta acción se genera manualmente el archivo de asistencia en la memoria SD que se guarda en un directorio de carpetas con el día, el mes y el turno correspondiente a cada curso.
- ATRÁS: Regresa al menú principal.





MODO AHORRO

Como indica el apartado, permite activar o desactivar el display para el ahorro de energía mientras que el dispositivo no se encuentra en uso.

7. DIAGRAMAS

Diagrama de Gantt (2022)

Actividades	Encargado	Julio	Agosto	Sep.	Octubre	Nov.	Dic.	2023
Investigaci ón	Agüero Mancuello Senes Melgarejo							
Lista de materiales	González Mancuello Szterenzos							
Carpeta preliminar	Senes Melgarejo							
Croquis y Diseño 3D	Agüero Mancuello							
Programa	Gonzalez Mancuello Szterenszo							
Construcci ón del circuito	Gonzalez Mancuello Szterenzos							
Impresione s 3D	Agüero							

Ensamblaje del proyecto	TODOS							
Carpeta final	Agüero Melgarejo Senes							
Instalación del proyecto	TODOS							

Diagrama de Gantt (2023)

Actividades	Encargado (s)	Marzo	1° de Abril	15 de Abril	1° de Mayo	15 de Mayo.	1° de Junio.	15 de Junio
Carpeta	Gonzalez							
Diseño 3D	Mancuello							
Parte anterior,								
Lado izquierdo,								
Lado derecho,								
Parte superior,								
Parte inferior,								
Parte posterior								
Recopilación sobre información de alumnos (Datos personales, UID de la tarjeta)	González Mancuello							

Programa	Mancuello								
Control de salida	Mancuello	Yellow	Yellow	Yellow	Green	Green	Green	Green	Green
Selector de turno/curso	Mancuello	Yellow	Yellow	Yellow	Yellow	Green	Green	Green	Green
Creación de carpetas	Mancuello	Yellow	Yellow	Yellow	Yellow	Green	Green	Green	Green
Impresiones 3D	Mancuello González	Red	Red	Red	Red	Yellow	Green	Green	Green
Mecanismo de seguridad	González Mancuello	Red	Red	Yellow	Yellow	Green	Green	Green	Green
Diseño del circuito	González	Red	Red	Yellow	Yellow	Green	Green	Green	Green
Armado del circuito	Gonzalez Mancuello	Red	Red	Red	Red	Red	Yellow	Green	Green
Instalación	Gonzalez Mancuello	Red	Red	Red	Red	Red	Red	Red	Green

En Proceso
Completo
No Iniciado



Red de Pert

ACTIVIDAD	RESPONSABLE
A (Diseño 3D)	Mancuello
B (Programación)	Mancuello
C (Impresiones 3D)	Mancuello
D (Diseño del circuito)	Gonzalez
E (Armado del circuito)	Gonzalez
F (Mecanismo de seguridad)	Gonzalez - Mancuello
G (Recopilacion de la informacion de alumnos)	Gonzalez - Mancuello
H (Instalación)	Gonzalez - Mancuello
I (Pruebas y evaluacion)	Gonzalez - Mancuello

Actividad	Estimaciones de tiempo (semanas)			Actividad	Actividad	Tiempo estimado (TE=To+4Tm+Tp/6)	Predecesora inmediata
	Optimista (To)	Más probable (Tm)	Pesimista (Tp)				
A	4	6	10	-	A	6,333333333	-
B	6	8	10	-	B	8	-
C	1	2	4	B	C	2,166666667	B
D	2	4	6	-	D	4	-
E	1	2	3	D	E	2	D
F	1	2	3	A,E	F	2	A,E
G	0,5	1	1,5	A	G	1	A
H	1	1,5	3	A,B,C,D,E,F,G	H	1,666666667	A,B,C,D,E,F
I	0,5	1	1,5	H	I	1	H

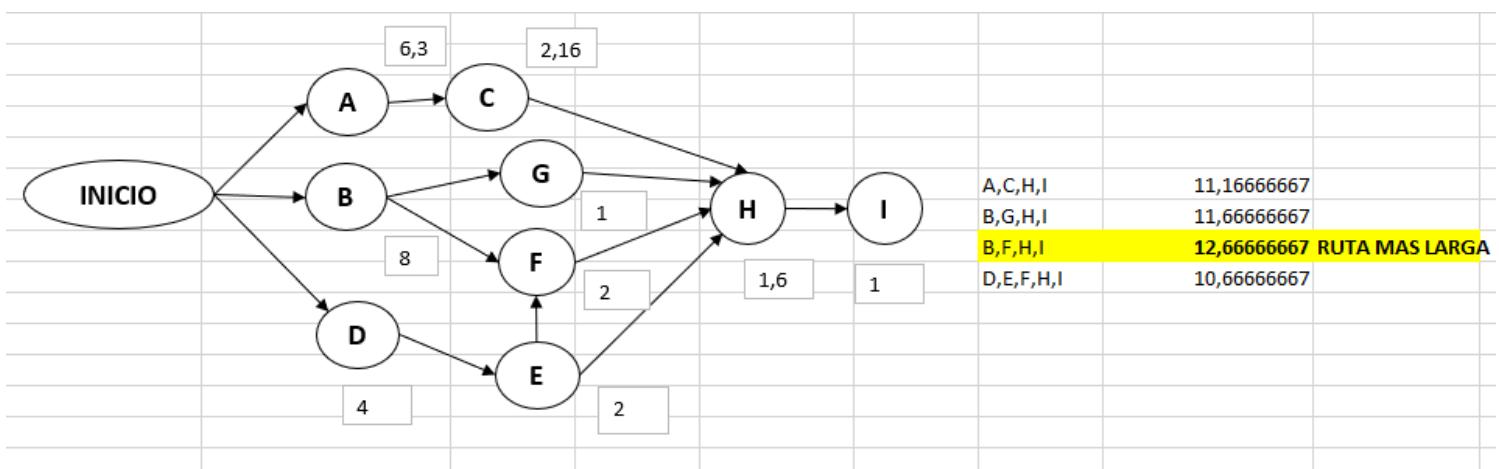


Diagrama de Flujo

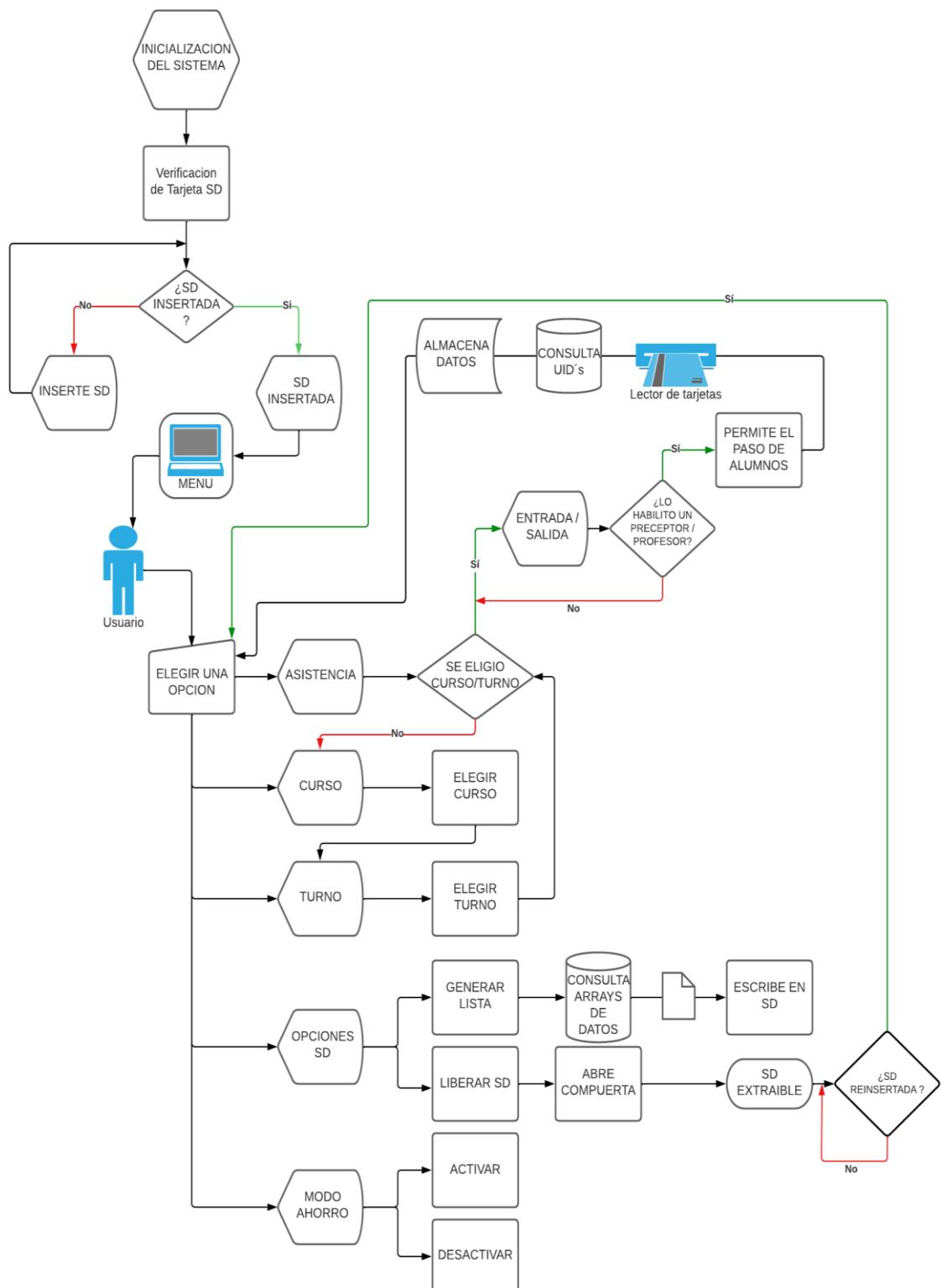
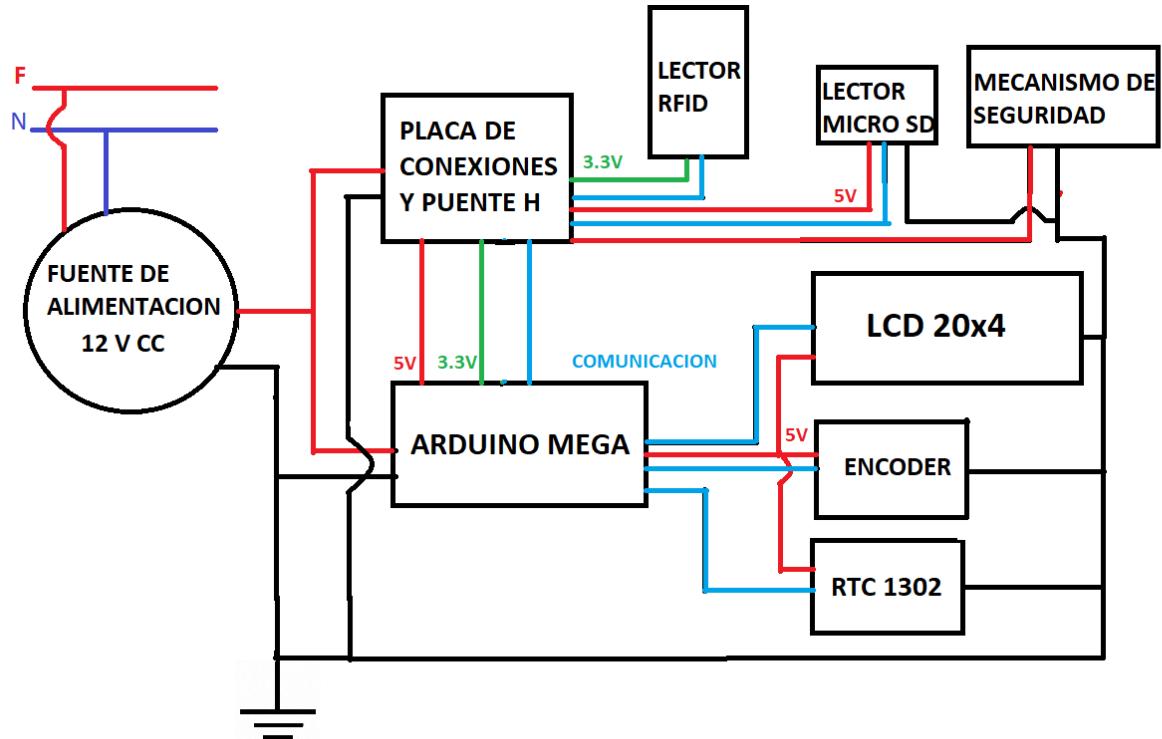
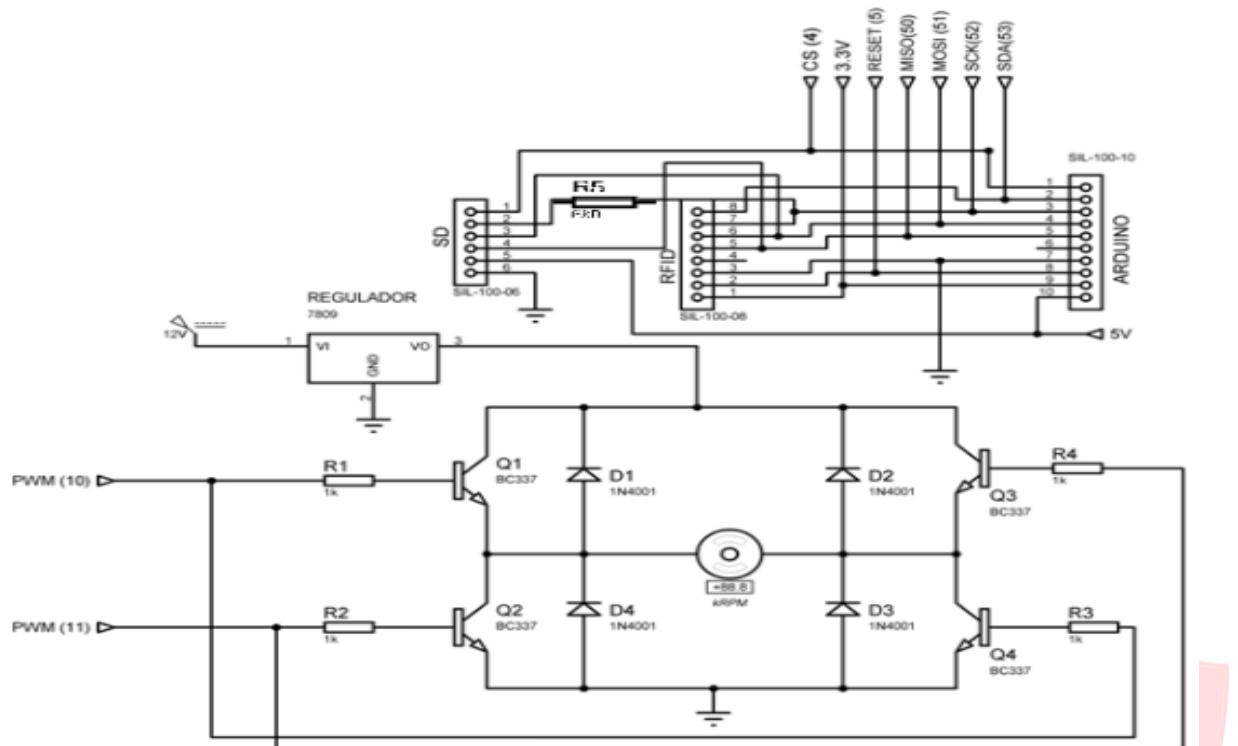
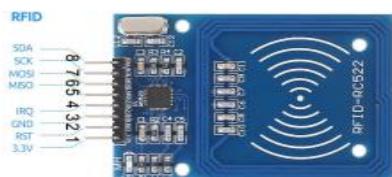
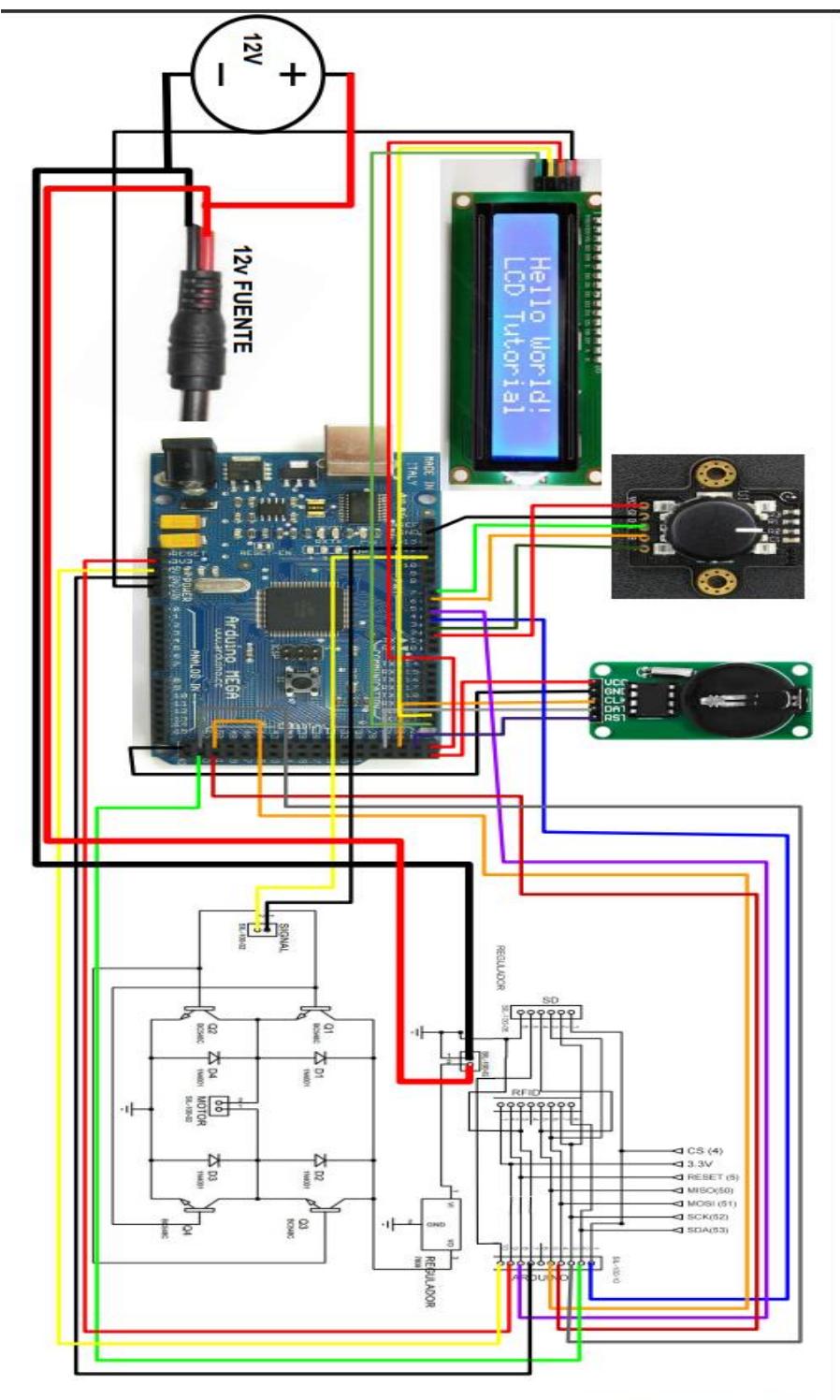


DIAGRAMA ELÉCTRICO



CIRCUITO ELECTRÓNICO





ÁRBOL DE FALLAS

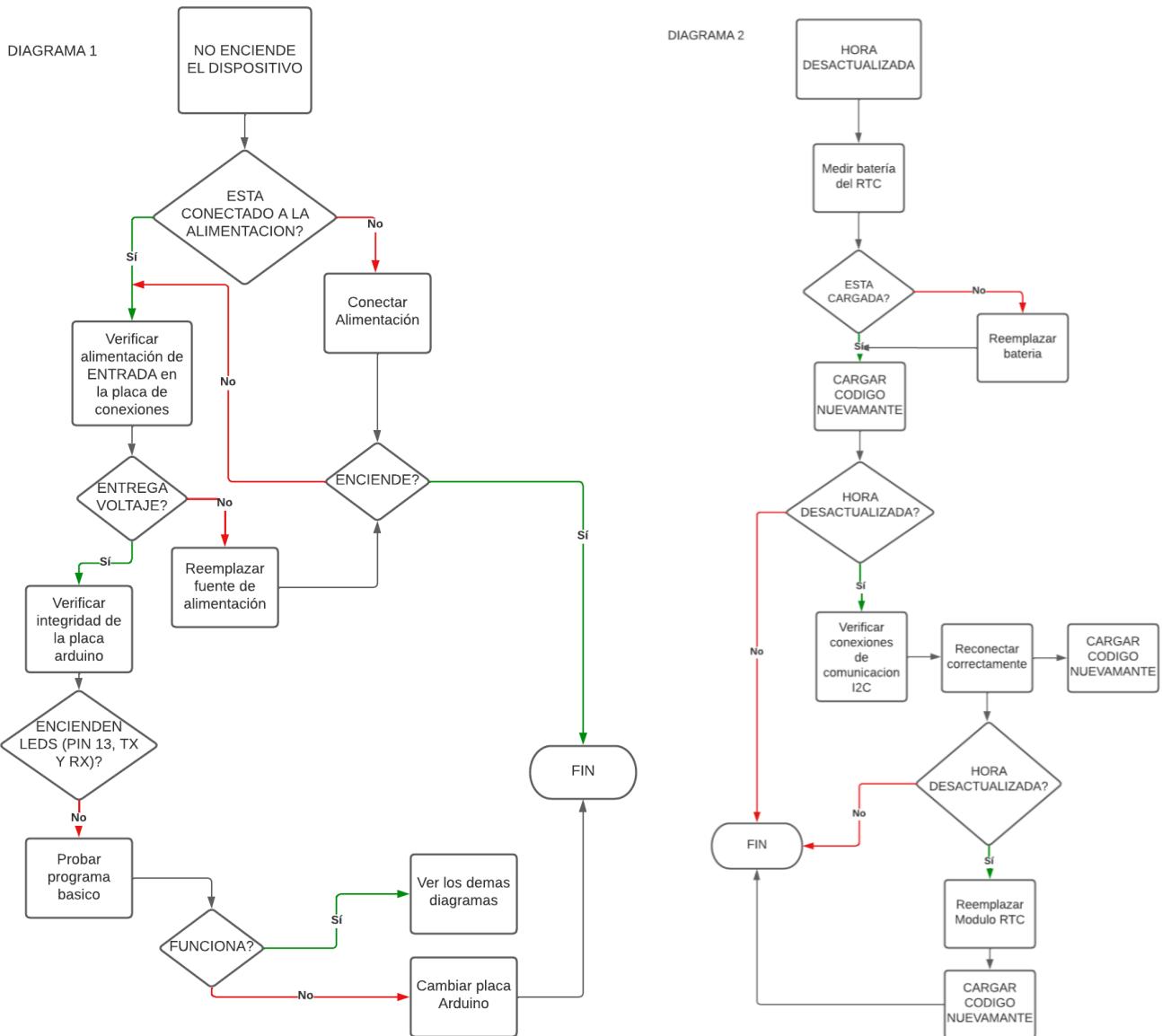


DIAGRAMA 3

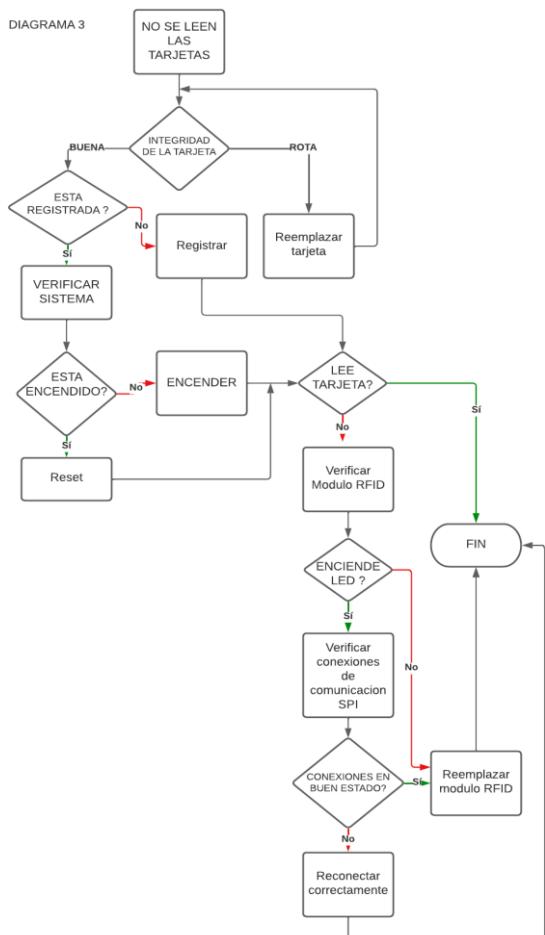
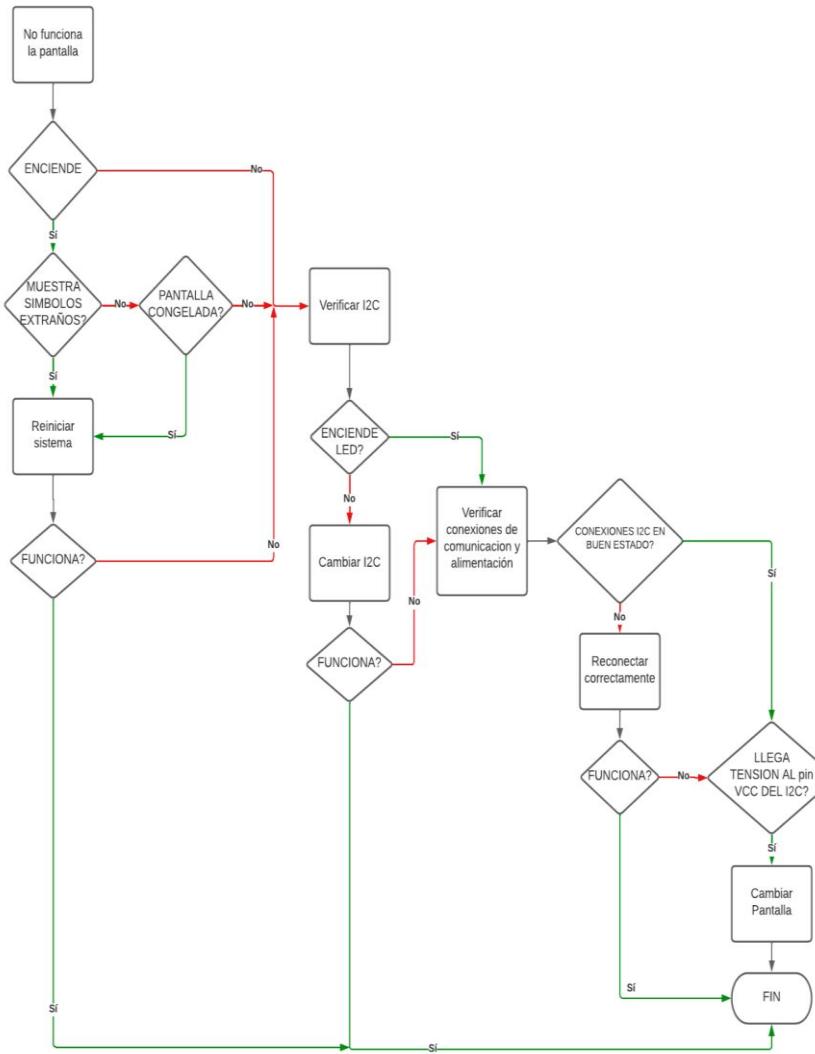


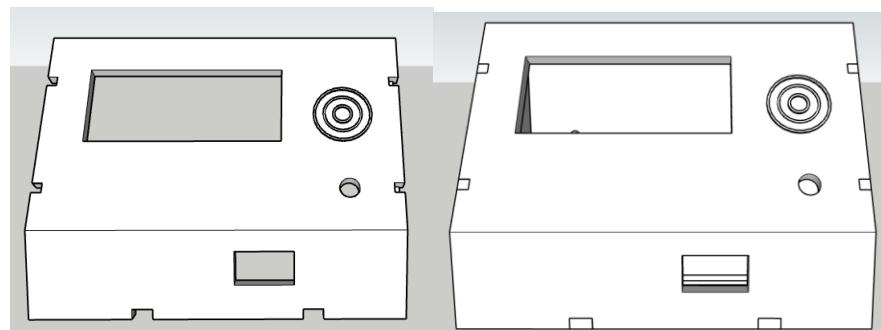
DIAGRAMA 4



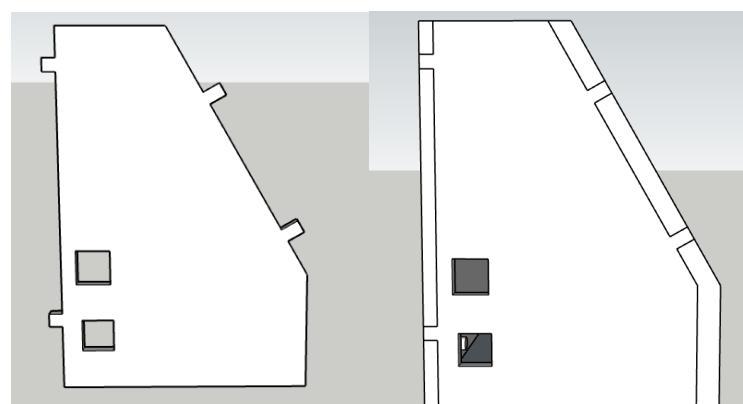
8. DISEÑO

DISEÑO 3D

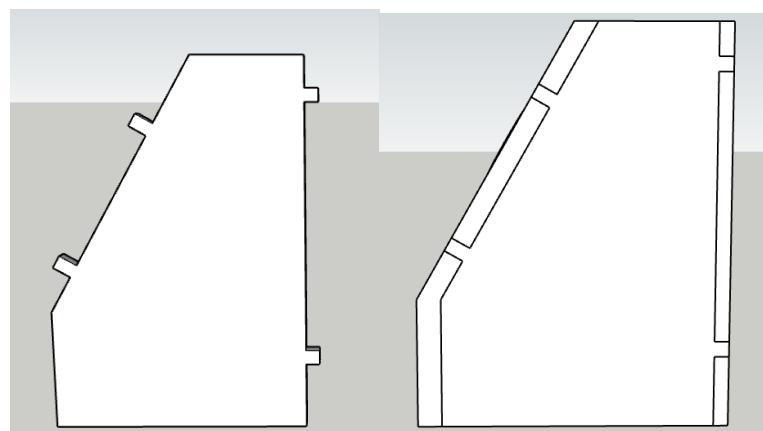
Vista anterior



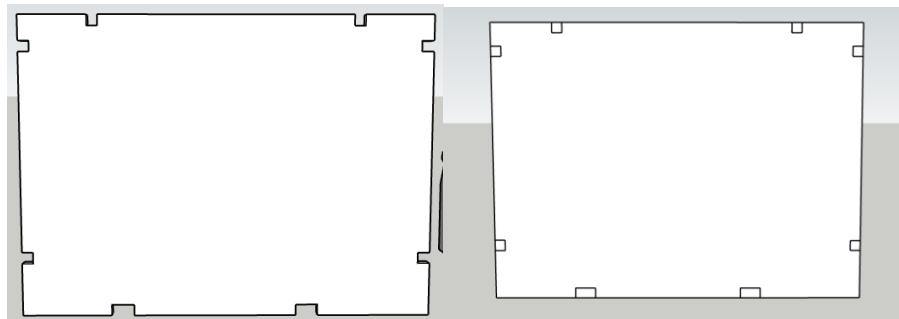
Vista lateral izquierda



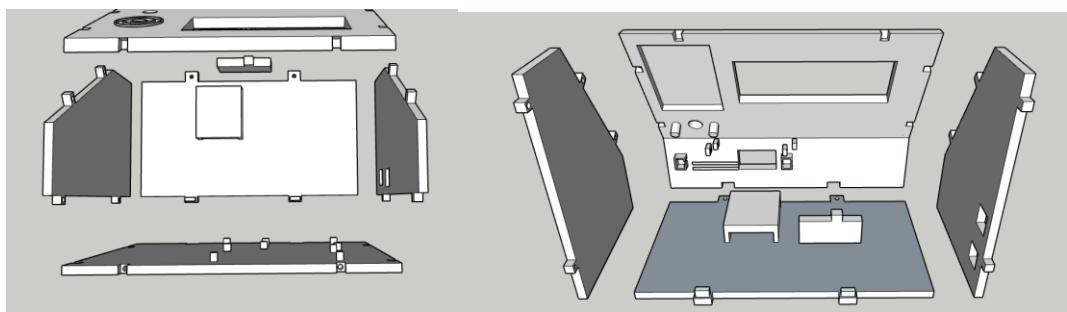
Vista lateral derecha



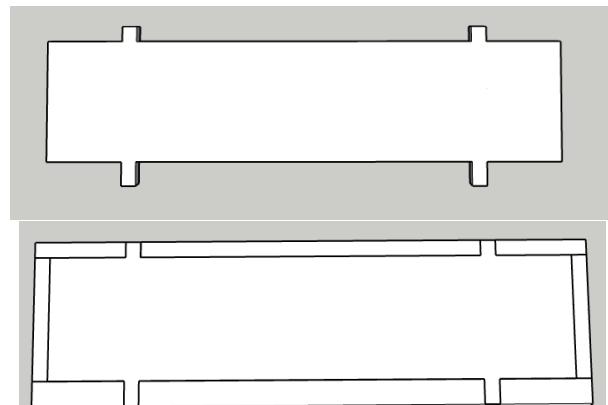
Vista posterior



Vista interior

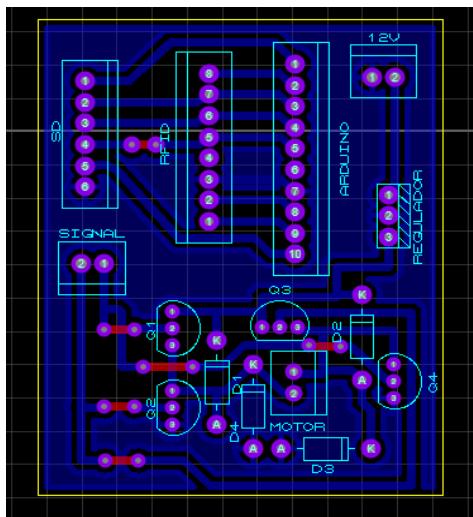


Vista superior

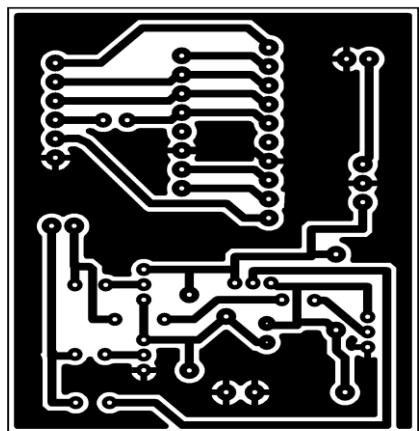


DISEÑO DEL CIRCUITO

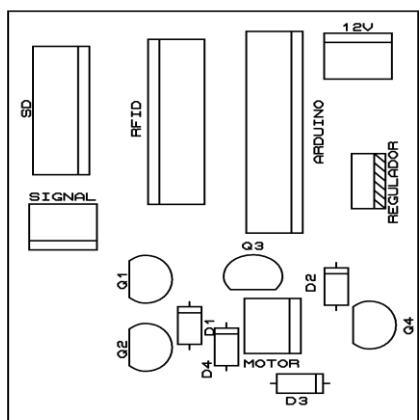
PCB Layout



Lado de pistas



Lado de componentes



9. PROCESO

Investigación

Cuando se decidió hacer este proyecto de sistema para el control de accesos, lo primero que se hizo fue buscar cómo hacer un menú que se pueda visualizar en una pantalla LCD y encoder rotativo para que el usuario interactúe de la mejor manera con el dispositivo.

Casi todas las opciones que se encontraron eran las de hacer un menú escribiendo detalladamente línea por línea todas las condiciones, opciones, funciones y muestreos en la pantalla. Pero se encontró una librería llamada LiquidMenu que tiene toda esa programación incluida y sólo hay que especificar las líneas, las pantallas y la función que se ejecuta según la opción que se elija, además de lo relacionado al encoder rotativo.

Una vez solucionado el menú, había que definir qué tipo de identificación utilizar. Existían dos opciones:

- Identificación por huella digital
- Identificación por tarjeta RFID

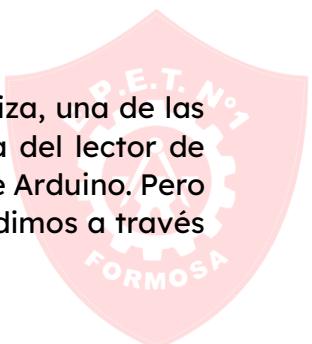
Empezamos investigando acerca del sensor de huellas digitales, específicamente el Lector UART de huellas dactilares con procesador STM32F205 integrado, ya que admite registrar más de 1000 huellas diferentes, pero la información y las librerías que se encontró no fue suficiente para hacerlo funcionar correctamente.



Observando esta situación, se decidió utilizar las tarjetas RFID como método de identificación.



Además de las conexiones y que protocolo de comunicación utiliza, una de las cosas más importantes a investigar fue como utilizar la librería del lector de tarjetas RFID. Se encuentra fácilmente en el gestor de librerías de Arduino. Pero debido a la sintaxis su uso es bastante tedioso por lo que aprendimos a través de guías, foros y videos de internet a como usarla.



```

23
24 #include <SPI.h>
25 #include <MFRC522.h>
26
27 #define RST_PIN      9      // Configurable, see typical pin layout above
28 #define SS_PIN       10     // Configurable, see typical pin layout above
29
30 MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
31
32 //***** *****
33 void setup() {
34   Serial.begin(9600);           // Initialize serial communications with the PC
35   SPI.begin();                 // Init SPI bus
36   mfrc522.PCD_Init();          // Init MFRC522 card
37   Serial.println(F("Read personal data on a MIFARE PICC:")); //shows in serial that it is ready to read
38 }
39

```

```

80
81 //----- GET FIRST NAME
82 status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 4, &key, &(mfrc522.uid)); //line 834 of MFRC522.cpp file
83 if (status != MFRC522::STATUS_OK) {
84   Serial.print(F("Authentication failed: "));
85   Serial.println(mfrc522.GetStatusCodeName(status));
86   return;
87 }
88
89 status = mfrc522.MIFARE_Read(block, buffer1, &len);
90 if (status != MFRC522::STATUS_OK) {
91   Serial.print(F("Reading failed: "));
92   Serial.println(mfrc522.GetStatusCodeName(status));
93   return;
94 }
95
96 //PRINT FIRST NAME
97 for (uint8_t i = 0; i < 16; i++)
98 {
99   if (buffer1[i] != 32)
100   {
101     Serial.write(buffer1[i]);
102   }
103 }
104 Serial.print(" ");
105

```

Código de ejemplo obtenido del repositorio oficial de la librería MFRC522

Luego de analizar diferentes opciones para gestionar los datos de la asistencia diaria, como transmisión por bluetooth, Wi-Fi y almacenamiento en tarjeta SD, se decidió que esta última era la manera más viable

Se utilizó el módulo Lector de Tarjeta Micro SD, por lo que nos pusimos a investigar en foros y guías acerca de su librería: SD.h. Lo que más nos interesaba saber era: la creación de archivos, la lectura y escritura de los mismos y la creación de carpetas y subcarpetas.

SD - open()

Abre un archivo en la tarjeta SD. Si el archivo se abre para escribir, se creará si aún no existe (pero el directorio que lo contiene ya debe existir).

Sintaxis

```
SD.open(filepath)
SD.open(filepath, mode)
```

Parámetros

- **filepath:** el nombre del archivo que se va a abrir, que puede incluir directorios (delimitados por barras diagonales, /).
- **mode (opcional):** el modo en el que abrir el archivo. El modo puede ser (abrir el archivo para leer, comenzando al principio del archivo) o (abrir el archivo para leer y escribir, comenzando al final del archivo).FILE_READFILE_WRITE



SD - mkdir()

Cree un directorio en la tarjeta SD. Esto también creará directorios intermedios que aún no existen; p. ej.. creará A, B y C.
C.SD.mkdir("a/b/c")

Sintaxis

```
SD.mkdir(filename)
```

Parámetros

- **filename:** el nombre del directorio que se va a crear, con subdirectorios separados por barras diagonales, /.

Devuelve

1 si la creación del directorio se realizó correctamente, 0 si no.

Un inconveniente que tuvimos fue que tanto el lector de tarjetas RFID como el lector de tarjetas SD utilizan el protocolo de comunicación SPI, por lo que no se pueden utilizar simultáneamente en condiciones normales, entonces se investigó una forma de poder usarlos a la vez. La solución encontrada fue conectar todos los pines de comunicación SPI de los módulos (RFID Y SD) en los pines especificados (en el Arduino Mega) por las librerías, pero esta vez, entre el pin de MISO del lector RFID y el pin MISO del lector de tarjetas SD, se colocó una resistencia de 680 ohm en serie. Con esto se logró igualar los voltajes de funcionamiento de ambos módulos, lo cual impedía el correcto funcionamiento. Informacion obtenida del siguiente link : <https://forum.arduino.cc/t/sd-card-and-rfid-not-working-together/492955/15>

Programación

- Identificación de los requisitos: Se inició por comprender los requisitos del proyecto, como el objetivo principal y las funcionalidades necesarias para el sistema de asistencia de alumnos de 7mo, 6to, 5to y 4to año.
- Diseño del sistema: Se realizó un diseño de alto nivel del sistema, considerando la arquitectura general, las interacciones entre componentes y los flujos de información.
- Selección de plataforma y lenguaje de programación: Se eligió la plataforma adecuada para el sistema embebido y se seleccionó el lenguaje de programación.
- Implementación del código: Se comenzó a escribir el código, dividiendo el sistema en módulos y funciones para mejorar la legibilidad y facilitar el mantenimiento.
- Pruebas y depuración: Se realizaron pruebas exhaustivas del código para garantizar su correcto funcionamiento y para identificar y corregir errores o comportamientos inesperados.
- Optimización y mejora: Se analizó el código en busca de posibles mejoras en términos de rendimiento, eficiencia o legibilidad. Se realizaron ajustes y optimizaciones según fuera necesario.
- Documentación: Se documentó el código, proporcionando descripciones claras de las funciones, explicando el propósito de cada componente y

detallando el flujo general del programa. También se documentaron los requisitos, las dependencias y las instrucciones de instalación si era necesario.

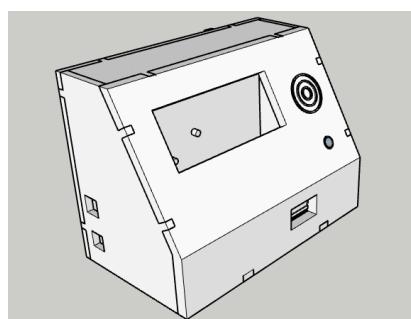
- Pruebas finales y validación: Se realizaron pruebas finales en el sistema completo para asegurarse de que todas las funcionalidades se ejecutarán correctamente y se cumplieran los requisitos establecidos.
- Implementación en el hardware externo: Finalmente, se realizó la implementación del código en el hardware objetivo, asegurándose de que el sistema embebido funcionara correctamente en el entorno real.

Diseño 3D

Una vez se establecieron todas las necesidades técnicas del proyecto, se empezó a diseñar lo que contendría más tarde a nuestro sistema. Para ello optamos por diseñar e imprimir en 3D, piezas que funcionan como carcasa para los componentes. Como si se tratase de un gabinete para una PC

Nuestro diseño sufrió diversos cambios a lo largo del proyecto, desde modificaciones simples como sus dimensiones generales, hasta algunas más especializadas como las referentes a los mecanismos y disposiciones internas de los componentes. Véase Anexo (primer prototipo).

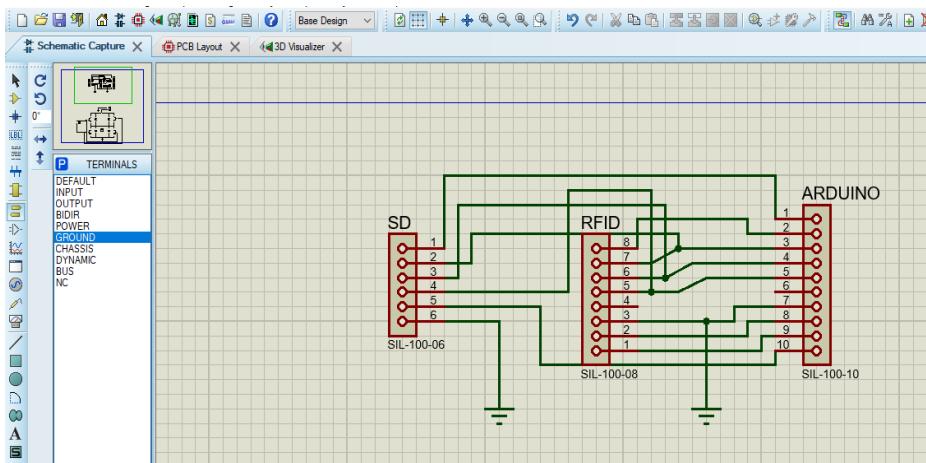
El diseño final contempla todo lo necesario para el correcto funcionamiento del sistema y la instalación del mismo en el aula. Pero también funciona como punto de partida para mejoras a futuro que pueda sufrir el proyecto.



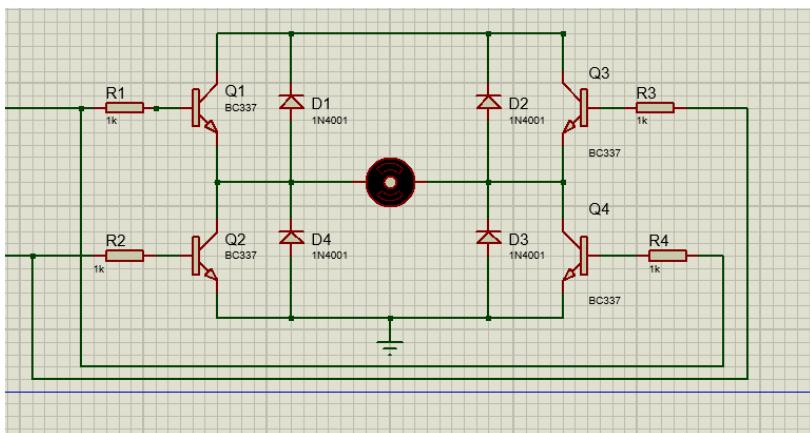
Diseño del circuito

Se inició el diseño ubicando los conectores para el lector RFID y el lector de tarjetas SD, de tal manera que queden interconectados los pines de comunicación SPI, dejando un lugar para la resistencia ubicada en la línea de comunicación de MISO. Luego, se colocó un conector de 10 terminales para la conexión con la placa Arduino MEGA, destinado no solamente a la comunicación con la placa sino también para la alimentación de ambos lectores.

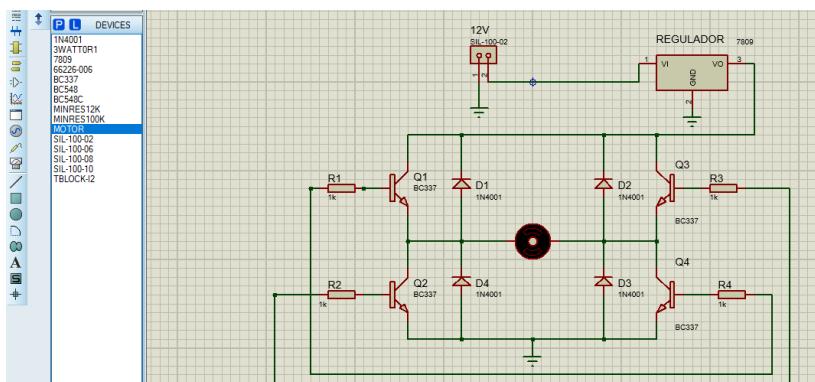




En otro lugar separado de la placa se diseñó un circuito puente H, con la finalidad de controlar el sentido de giro del motor del mecanismo de seguridad. Cuenta con cuatro transistores NPN BC337, cada uno con su resistencia de protección, y cuatro diodos rectificadores polarizados en inversa (uno por cada colector del transistor).

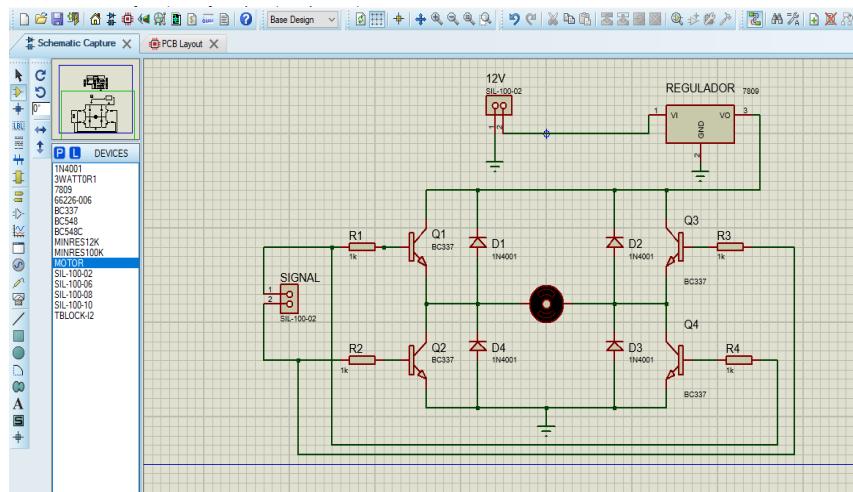


Para alimentar al motor, se colocó un regulador de 9V, cuya entrada son los 12V provenientes de la fuente, y su salida está conectada directamente al colector del puente H, y por lo tanto al motor.



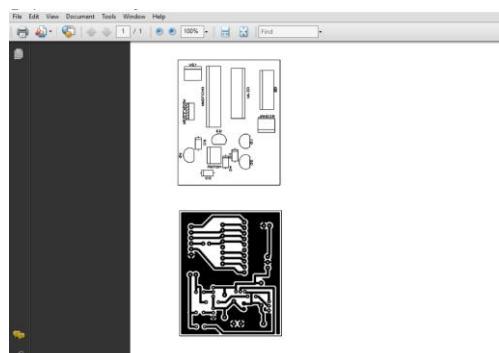
Por último, se dejó lugar para un conector de dos terminales destinados a la conexión con el Arduino.





Fabricación de PCB

- Se exportó el archivo de la placa, tanto del lado de pistas como el lado componentes y se imprimió en papel fotográfico.



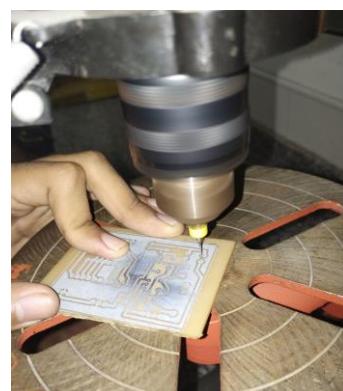
- Después, se colocó la impresión del lado de pistas sobre la placa de cobre y se pasó repetidas veces por la laminadora.



- Una vez adherida la tinta a la placa, se procedió a retirar el papel y a colocar la placa en un recipiente con ácido férreo para eliminar el cobre sobrante.



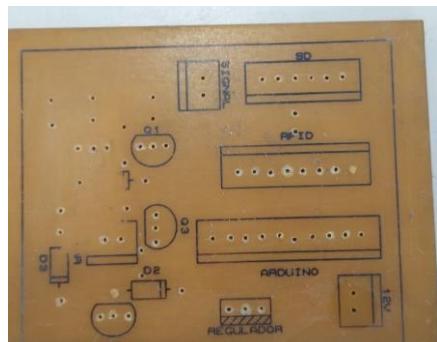
- Se hicieron los orificios en los lugares correspondientes.



- Para deshacerse de la tinta que quedaba en la placa se la limpió con virulana y alcohol.



- Luego, se repitió el proceso para el lado de componentes, pero sin colocar la placa en ácido férrico.

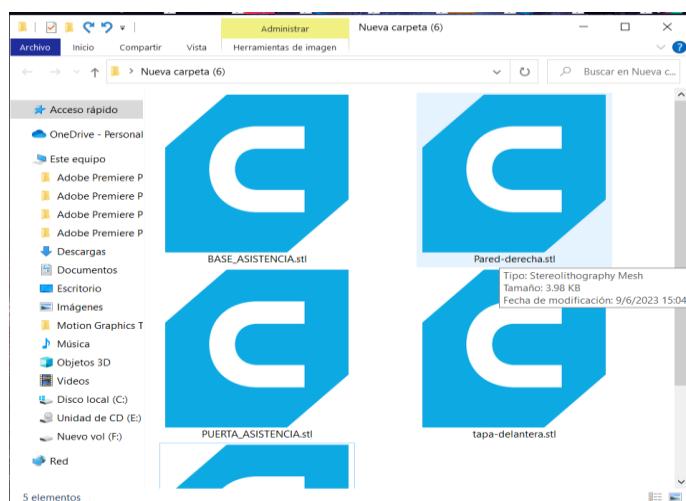


- Por último, se soldaron todos los componentes



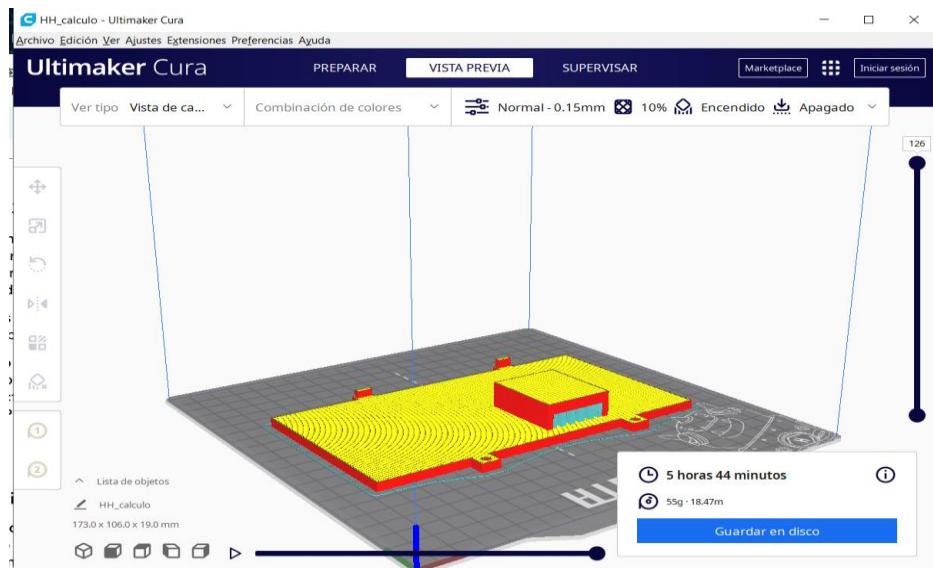
Impresiones 3D

Con el diseño terminado, se procedió a exportar cada parte del mismo en archivos “.stl” para poder pasarlo por el software de corte Ultimaker Cura. Con este programa especificamos y seleccionamos los parámetros de impresión de cada pieza.



Una vez finalizamos la configuración exportamos los archivos a formato “gcode” que es el tipo de archivo que admiten las impresoras 3D.





En cuanto al tiempo que se necesita, se calcularon aproximadamente 30 horas, aunque el lapso real de impresión puede presentar variaciones dependiendo de factores como la impresora 3D, la configuración o el material utilizado (PLA/ABS).

Armado del circuito

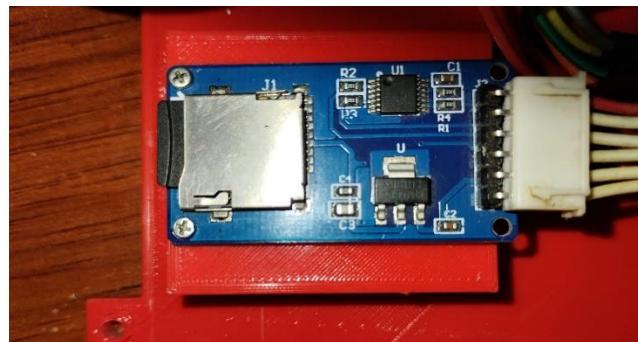
En la parte trasera de la carcasa, se colocó con tornillos la placa PCB y la placa de Arduino Mega. A un lado, se pegó con silicona el módulo RTC 1302 y se lo conectó directamente a la placa Arduino



Se colocó el lector RFID en la parte interna de la cara anterior y se lo conectó a la placa PCB en su conector correspondiente.



Se hizo lo mismo con el lector SD, colocándolo en su soporte ubicado en la base de la carcasa y conectándolo a la placa PCB



La pantalla LCD se colocó con tornillos en el frente de la carcasa y se conectó directamente a la placa Arduino.



El encoder rotativo se ubicó en el orificio de la parte inferior derecha del frente de la carcasa y se lo aseguró con dos tornillos.



En el conector de Arduino de la placa PCB se conectaron directamente cables macho-macho a la placa Arduino MEGA a los pines correspondientes



Fabricación del mecanismo de seguridad

Para el mecanismo de seguridad que resguarda la tarjeta SD en el interior de la carcasa nos basamos en la estructura de unas disqueras que pudimos encontrar en el taller.



Se desmontaron los mecanismos de los cuales obtuvimos los siguientes componentes :

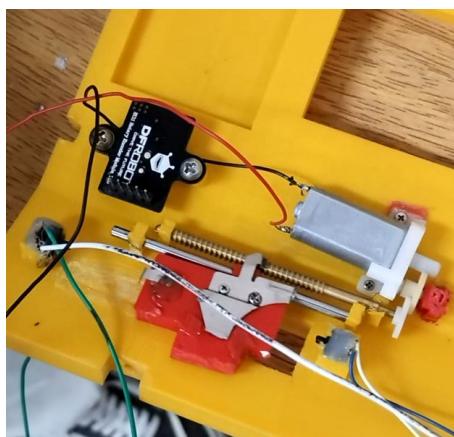
- Riel liso
- Riel roscado
- Motor DC
- Final de carrera (x2)

Teniendo en cuenta la disposición del mecanismo original, tratamos de replicarlo y adaptarlo a nuestro diseño.

Contando con las piezas impresas nos dispusimos a montar el mecanismo y realizar los ajustes necesarios para el correcto funcionamiento. Una vez finalizadas las conexiones eléctricas se realizaron pruebas (ver video) para asegurar el funcionamiento.

[VID_20230616_203513401.mp4 - Google Drive](#)

El resultado final del montaje del mecanismo de seguridad fue el siguiente :



Instalación

Para la instalación del dispositivo se optó por asegurarla en la pared adyacente a la puerta del taller, para ello utilizamos una ménsula atornillada a la pared.

Para hacer la ménsula dónde se colocó la asistencia, se cortó una pieza metálica que era parte de un equipo de música y se le realizaron tres orificios.



Luego se marcó la pared y se hicieron orificios de modo que coincidieran con los existentes en la ménsula y se colocaron tarugos dentro de ellos. Después, se colocaron tornillos para asegurar la ménsula a la pared.



Lo primero que se hizo en la carcasa fue perforar dos orificios en la pared trasera, para colocar los tornillos que la asegurarían a la pared.

Una vez armado el circuito, se encastró y se pegó la parte posterior a la base de la carcasa.

Luego, se encastraron y se pegaron las dos paredes laterales a la pared posterior





Se realizaron dos orificios en los encastres anteriores de la base, atravesando también los encastres de la parte anterior de la carcasa. Luego, se colocaron tornillos en ambos orificios, haciendo coincidir la pared frontal con el resto de la carcasa y se colocó la tapa superior



Por último, se aseguró el dispositivo con tres tornillos (uno en la ménsula y dos en la pared).



10. LISTA DE MATERIALES

MATERIALES	CANTIDADES
Arduino mega	1
Módulo micro SD	1
Módulo Rfid	1
Módulo Rtc DS1302	1
resistencia 1k	1
Transistor BC547	1
Transistor BC337	3
Encoder rotativo	1
LCD 20x4	1
Motor de corriente continua	1
Engranajes	2
Tornillos M3x16	2
Cables hembra macho	cantidad a confirmar
Plastico PLA	350g aprox



11. HERRAMIENTAS DIGITALES

A continuación se detallarán las herramientas digitales utilizadas para llevar a cabo este Proyecto:

IDE Arduino



El IDE de Arduino es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) que se utiliza para escribir, compilar y cargar el código en la placa Arduino. Es una aplicación que proporciona una interfaz gráfica de usuario (GUI) que simplifica el proceso de programación y depuración de las placas Arduino.

Es una herramienta de software libre y de código abierto que se puede descargar e instalar de forma gratuita desde el sitio web de Arduino. Una vez instalado, el IDE proporciona un editor de texto enriquecido con funciones de resaltado de sintaxis, auto-completado y corrección de errores, así como herramientas para compilar y cargar el código en la placa Arduino.

El IDE de Arduino utiliza una versión simplificada de C++ para programar la placa. Además, también incluye una biblioteca estándar de funciones para interactuar con los diferentes componentes de la placa Arduino.

SketchUp



SketchUp es un software de modelado 3D que permite crear modelos tridimensionales de objetos, edificios, paisajes, entre otros. Este software es



ampliamente utilizado por arquitectos, ingenieros, diseñadores de interiores y profesionales de la construcción para diseñar y visualizar modelos 3D.

SketchUp es conocido por su interfaz amigable y fácil de usar, lo que lo hace accesible para usuarios con diferentes niveles de experiencia en modelado 3D. Además, cuenta con herramientas de dibujo y modelado intuitivas, así como una gran biblioteca de objetos y materiales predefinidos que se pueden utilizar en los modelos.

Las principales funciones de SketchUp incluyen:

- **Modelado 3D:** SketchUp permite crear modelos 3D precisos y detallados utilizando herramientas intuitivas, como la extrusión, la rotación y la escala. También es posible importar modelos 3D de otros programas y trabajar en ellos dentro de SketchUp.
- **Renderizado:** SketchUp incluye herramientas para renderizar y visualizar modelos 3D con efectos de iluminación realistas y sombras precisas. También se pueden agregar materiales, texturas y fondos para mejorar la calidad de la imagen.
- **Documentación:** SketchUp permite crear planos detallados y dimensionados a partir de los modelos 3D. También se pueden agregar notas y etiquetas para especificar detalles y requerimientos de construcción.
- **Colaboración:** SketchUp permite compartir modelos 3D y documentos con otros usuarios, ya sea mediante la exportación de archivos o el uso de servicios en línea. También es posible colaborar en tiempo real en un mismo modelo mediante el uso de la función "SketchUp Live".
- **Extensibilidad:** SketchUp cuenta con una amplia biblioteca de extensiones y complementos que permiten extender las funcionalidades del programa. Estas extensiones pueden ser desarrolladas por terceros o por el equipo de SketchUp, y se pueden descargar e instalar desde la plataforma Extension Warehouse.



Ultimaker Cura



Ultimaker Cura es un software de preparación de modelos 3D diseñado para utilizarse específicamente con impresoras 3D FDM (Fused Deposition Modeling), como las impresoras 3D de Ultimaker. El software es de código abierto y gratuito para descargar, lo que significa que cualquier persona puede utilizarlo y modificarlo para adaptarse a sus necesidades específicas.

El software permite cargar modelos 3D en varios formatos (STL, OBJ, 3MF, etc.)

Una de las principales ventajas de Ultimaker Cura es que incluye una biblioteca de perfiles de impresora para varias impresoras 3D, incluyendo las impresoras 3D de Ultimaker. Los usuarios pueden ajustar los parámetros de impresión, como la velocidad, la temperatura y la resolución, para adaptarse a sus necesidades específicas. Además, se pueden generar soportes para modelos 3D complejos para asegurar una impresión exitosa, y los usuarios pueden ajustar el tamaño, la forma y la densidad de los soportes para adaptarse a las necesidades específicas del modelo 3D.

Las principales funciones de Ultimaker Cura incluyen:

- **Preparación de modelos 3D:** Cura permite cargar modelos 3D en varios formatos (STL, OBJ, 3MF, etc.) y prepararlos para su impresión en 3D. Los usuarios pueden realizar ajustes en la orientación, escalado, posición y soportes para optimizar el proceso de impresión.
- **Configuración de impresora:** Cura incluye una biblioteca de perfiles de impresora para varias impresoras 3D, incluyendo las impresoras 3D de Ultimaker. Los usuarios pueden ajustar los parámetros de impresión, como la velocidad, la temperatura y la resolución, para adaptarse a sus necesidades específicas.
- **Visualización de la impresión:** Cura proporciona una vista previa en 3D del modelo 3D y de cómo se imprimirá. Los usuarios pueden ver los soportes, las capas y la ruta de impresión para asegurarse de que la impresión se realizará correctamente.
- **Generación de soportes:** Cura permite generar soportes para modelos 3D complejos para asegurar una impresión exitosa. Los usuarios pueden ajustar el tamaño, la forma y la densidad de los soportes para adaptarse a las necesidades específicas del modelo 3D.

- Optimización de la impresión: Cura proporciona una variedad de herramientas para optimizar el proceso de impresión, incluyendo la capacidad de ajustar la velocidad, la resolución y otros parámetros para equilibrar la calidad de la impresión y el tiempo de impresión.
- Previsualización y simulación de la impresión: Cura incluye una función de previsualización y simulación de la impresión, que permite a los usuarios ver cómo se imprimirá el modelo 3D antes de imprimirla en realidad. Esto ayuda a evitar errores y a ahorrar tiempo y materiales.

12. CÓDIGO FUENTE

```

/* REFERENCIA:
0 == AUSENTES
1 == PRESENTE
2 == TARDANZA
3 == AUSENTES CON PERMANENCIA EN EL AULA

*/
#include <SPI.h>
#include <SD.h>
File myFile;
const byte chipSelect = 4;
#include <MFRC522.h>
String Preceptor[8][2] = {
  {"Lopez R.", "60 0B 8D 65"}, 
  {"Pesoa", "60 8C 28 2B"}, 
  {"Gonzalez J", "80 13 41 D5"}, 
  {"Gonzalez M", "63 C7 27 2B"}, 
  {"Victor", "9D 70 0F 2B"}, 
  {"Natalia", "80 84 8C 65"}, 
  {"Villasanti", "31 F9 8C 65"}, 
  {"Davila S.", "F6 53 2E 2B"}};

bool flag_salida = 0;
String dataString;           // Cadena para almacenar datos leídos del archivo
String names, id;           // Cadena para almacenar nombres e IDs
const byte numRows = 30;     // Número de filas de la matriz
const byte numCols = 2;       // Número de columnas de la matriz
byte encoderPin = 3;
boolean septimo;
boolean sexto;
boolean quinto;
boolean cuarto;
boolean tarde;
boolean manana;
boolean archivoCursoSelected = 0;
boolean curso_seleccionado = false;
String alumno_estado;
String dataArray[numRows][numCols];
MFRC522 rfid(53, 5);
byte Indice;
byte Indice_2 = -1;
boolean prueba = 0;
boolean alerta = 0;
///////////////////////////////
#include <ThreeWire.h>
#include <RtcDS1302.h>

ThreeWire myWire(26, 24, 22); // DAT, CLK, RST
RtcDS1302<ThreeWire> Rtc(myWire);

char Ano[20];
char Mes[20];
char Dia[20];
char Hora[20];

```




```

LiquidLine linea5(1, 6, "Archivos y SD");
LiquidLine linea3(1, 4, "Extras");
LiquidLine linea4(1, 5, "Modo Ahorro");

LiquidScreen pantalla1;

//////////////////LINEAS
EXTRA////////////////////

LiquidLine RESET(1, 0, "Resetear archivo");
LiquidLine atras1(1, 1, "<<Atras");
LiquidScreen pantalla2;

//////////////////LINEAS Turno ///////////////////
LiquidLine Turnos(1, 0, "Elige un turno:");
LiquidLine Turnos_T(1, 1, "Tarde");
LiquidLine Turnos_M(1, 2, "Manana");
LiquidLine T_Atras(1, 3, "<<Atras");
LiquidScreen pantalla6;

//////////////////LINEAS SD/////////////////
LiquidLine SDMENU1(1, 0, "Liberar SD ");
LiquidLine SDMENU2(1, 1, "Generar archivo");
LiquidLine SDMENU3(1, 2, "<<Atras");
LiquidScreen pantalla3;

//////////////////LINEAS AHORRO/////////////////
LiquidLine OnLight(1, 0, "Desactivar ");
LiquidLine OffLight(1, 1, "Activar ");
LiquidLine atras(1, 2, "<<Atras");
LiquidScreen pantalla4;

//////////////////LINEAS LECTOR/////////////////
LiquidLine lec1(1, 0, "Habilitar entrada");
LiquidLine lec2(1, 1, "Habilitar salida");
LiquidLine lec3(1, 2, "<<Atras");
LiquidScreen pantalla5;

//////////////////LINEAS CURSO ///////////////////
LiquidLine CURSOS(1, 0, "Elige un curso:");
LiquidLine CURSOS_4(1, 1, "Cuarto U");
LiquidLine CURSOS_5(1, 2, "Quinto U");
LiquidLine CURSOS_6(1, 3, "Sexto U");
LiquidLine CURSOS_7(1, 4, "Septimo U");
LiquidLine CURSO_ATRAS(1, 5, "<<Atras");
LiquidScreen pantalla7;

//////////////////RTC/////////////////
LiquidMenu menu(lcd);

void setup() {

Serial.begin(57600);

myFile = SD.open("ej.txt", FILE_WRITE);
myFile.close();

//////////////////RTC/////////////////
Serial.print("compiled: ");
Serial.print(__DATE__);
Serial.println(__TIME__);

Rtc.Begin();

RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__);
printDateTime(compiled);
}

```



```

Serial.println();

if (!Rtc.IsDateTimeValid())
{
    // Common Causes:
    //      1) first time you ran and the device wasn't running yet
    //      2) the battery on the device is low or even missing

    Serial.println("RTC lost confidence in the DateTime!");
    Rtc.SetDateTime(compiled);
}

if (Rtc.GetIsWriteProtected())
{
    Serial.println("RTC was write protected, enabling writing now");
    Rtc.SetIsWriteProtected(false);
}

if (!Rtc.GetIsRunning())
{
    Serial.println("RTC was not actively running, starting now");
    Rtc.SetIsRunning(true);
}

RtcDateTime now = Rtc.GetDateTime();
if (now < compiled)
{
    Serial.println("RTC is older than compile time! (Updating DateTime)");
    Rtc.SetDateTime(compiled);
}
else if (now > compiled)
{
    Serial.println("RTC is newer than compile time. (this is expected)");
}
else if (now == compiled)
{
    Serial.println("RTC is the same as compile time! (not expected but all is
fine)");
}

//////////////////SD//////////////////SD//////////////////SD//////////////////SD

// Open serial communications and wait for port to open:
while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
}

lcd.init();
lcd.backlight();
starttime = millis();
while ((millis() - starttime) < 4000) {
    lcd.setCursor(3, 1);
    lcd.print("BIENVENIDOS!");
    lcd.setCursor(3, 2);
    lcd.print("ELECTRONICOS");
}

limpia();

starttime = millis();
while ((millis() - starttime) < 2000) {
    lcd.setCursor(0, 1);
    lcd.print("INICIANDO SISTEMA...");
}
limpia();

//////////////////CASO ESPECIAL si la sd no esta al iniciar////////////////

while (!SD.begin(4)) {

    lcd.setCursor(0, 0);
    lcd.print("TARJETA INEXISTENTE");
    lcd.setCursor(5, 1);
    lcd.print("INSERTE SD");
    lcd.setCursor(4, 2);
    lcd.print("PARA INICIAR");
}

limpia();

```



```

msjsd();
limpia();
///////////////////////////////
if (SD.begin(4)) {
    Serial.println("initialization correct!");
}
///////////////////////////////
pinMode(encoderPin, OUTPUT);
pinMode(sw, INPUT_PULLUP);
pinMode(fcarrera1, INPUT_PULLUP);
pinMode(fcarrera2, INPUT_PULLUP);
// pinMode(pulsadordsr, INPUT_PULLUP);
///////////////////////////////PANTALLA PRINCIPAL/////////////////////////////
pantalla1.add_line(linea1_P);
pantalla1.add_line(linea1);
pantalla1.add_line(linea6);
pantalla1.add_line(linea2);
pantalla1.add_line(linea5);
pantalla1.add_line(linea3);
pantalla1.add_line(linea4);

linea1_P.set_focusPosition(Position::CUSTOM);
linea1.set_focusPosition(Position::LEFT);
linea6.set_focusPosition(Position::LEFT);
linea2.set_focusPosition(Position::LEFT);
linea5.set_focusPosition(Position::LEFT);
linea3.set_focusPosition(Position::LEFT);
linea4.set_focusPosition(Position::LEFT);

linea1_P.attach_function(1, fn_atras );
linea1.attach_function(1, LECTOR);
linea2.attach_function(1, TURNO);
linea3.attach_function(1, EXTRAS);
linea4.attach_function(1, MenuAhorro);
linea5.attach_function(1, EXSD);
linea6.attach_function(1, curso_selector);
menu.add_screen(pantalla1);
///////////////////////////////PANTALLA EXTRAS/////////////////////////////
pantalla2.add_line(RESET);
pantalla2.add_line(atras1);
RESET.set_focusPosition(Position::LEFT);
atras.set_focusPosition(Position::RIGHT);
RESET.attach_function(1, RES);
atras1.attach_function(1, fn_atras);
menu.add_screen(pantalla2);
///////////////////////////////PANTALLA SD/////////////////////////////
pantalla3.add_line(SDMENU1);
pantalla3.add_line(SDMENU2);
pantalla3.add_line(SDMENU3);
SDMENU1.set_focusPosition(Position::LEFT);
SDMENU2.set_focusPosition(Position::LEFT);
SDMENU3.set_focusPosition(Position::RIGHT);
SDMENU1.attach_function(1, SERVOSD);
SDMENU2.attach_function(1, GEN);
SDMENU3.attach_function(1, fn_atras);
menu.add_screen(pantalla3);
///////////////////////////////PANTALLA AHORRO/////////////////////////////
pantalla4.add_line(OnLight);
pantalla4.add_line(OffLight);
pantalla4.add_line(atras);
OnLight.set_focusPosition(Position::LEFT);
OffLight.set_focusPosition(Position::LEFT);
atras.set_focusPosition(Position::RIGHT);
OnLight.attach_function(1, ON);
OffLight.attach_function(1, OFF);
atras.attach_function(1, fn_atras);
menu.add_screen(pantalla4);
///////////////////////////////PANTALLA LECTOR/////////////////////////////
pantalla5.add_line(lec1);
pantalla5.add_line(lec2);
pantalla5.add_line(lec3);
lec1.set_focusPosition(Position::LEFT);
lec2.set_focusPosition(Position::LEFT);
lec3.set_focusPosition(Position::RIGHT);
lec1.attach_function(1, lec_f);
lec2.attach_function(1, LEC_SALIDA);
lec3.attach_function(1, fn_atras);

```



```

menu.add_screen(pantalla5);
//////////////////PANTALLA TURNO///////////////
pantalla6.add_line(Turnos);
pantalla6.add_line(Turnos_T);
pantalla6.add_line(Turnos_M);
pantalla6.add_line(T_Atras);
Turnos.set_focusPosition(Position::CUSTOM);
Turnos_T.set_focusPosition(Position::LEFT);
Turnos_M.set_focusPosition(Position::LEFT);
T_Atras.set_focusPosition(Position::LEFT);
Turnos.attach_function(1, V);
Turnos_T.attach_function(1, T_tarde);
Turnos_M.attach_function(1, T_man);
T_Atras.attach_function(1, fn_atras);
menu.add_screen(pantalla6);
//////////////////PANTALLA CURSOS/////////////
pantalla7.add_line(CURSOS);
pantalla7.add_line(CURSOS_4);
pantalla7.add_line(CURSOS_5);
pantalla7.add_line(CURSOS_6);
pantalla7.add_line(CURSOS_7);
pantalla7.add_line(CURSO_ATRAS);
CURSOS.set_focusPosition(Position::CUSTOM);
CURSOS_4.set_focusPosition(Position::LEFT);
CURSOS_5.set_focusPosition(Position::LEFT);
CURSOS_6.set_focusPosition(Position::LEFT);
CURSOS_7.set_focusPosition(Position::LEFT);
CURSO_ATRAS.set_focusPosition(Position::LEFT);
CURSOS.attach_function(1, V);
CURSOS_4.attach_function(1, CURSOS_cuarto);
CURSOS_5.attach_function(1, CURSOS_quinto);
CURSOS_6.attach_function(1, CURSOS_sexto);
CURSOS_7.attach_function(1, CURSOS_septimo);
CURSO_ATRAS.attach_function(1, fn_atras);
menu.add_screen(pantalla7);
pantalla1.set_displayLineCount(4);
pantalla2.set_displayLineCount(4);
pantalla3.set_displayLineCount(3);
pantalla4.set_displayLineCount(3);
pantalla5.set_displayLineCount(3);
pantalla6.set_displayLineCount(4);
pantalla7.set_displayLineCount(4);
menu.init();
menu.set_focusedLine(0);
menu.update();
SPI.begin();
rfid.PCD_Init();
}
boolean msj = 0;
boolean sdarchive = false;
String file; ///////////nombre del archivo
String carpeta; //carpeta principal
String dataFile; ///////////nombre del archivo con los datos de los alumnos
boolean var = 0;
boolean FOLDER_EXIST = false;
boolean FOLDER_EXIST2 = false;
String ruta;
void loop() {
  digitalWrite(encoderPin, HIGH);
  RtcDateTime now = Rtc.GetDateTime();
  printDateTimeDia(now);
  printDateTimeFecha(now);
  printDateTimeTiempo(now);
  String contenedor = String(Fecha); //// guarda dato rtc (char) en variable
  string
  if ( carpeta == contenedor) { ////////// si el nombre del archivo = dato
    rtc inicia creacion del files sd
    if (sdarchive == false) { //////////crea archivo si aun no fue creado
      if (!FOLDER_EXIST) {
        if (SD.mkdir("/") + carpeta)) {
          Serial.println("dia creada correctamente");
          FOLDER_EXIST = true;
        } else {
          Serial.println("Error al crear dia");
        }
      }
    Serial.println(carpeta);
    delay(20);
  }
}

```



```

if (curso_seleccionado == true) {
    Serial.println("CURSO SELECCIONADO");
    if (septimo == 1) {
        file = "7mo.txt";
        if (tarde == 1) {
            ruta = "/" + carpeta + "/" + "7mo" + "/" + "tarde";
            if (!FOLDER_EXIST2) {
                if (SD.mkdir(ruta)) {
                    Serial.println("tarde correctamente");
                    FOLDER_EXIST2 = true;
                }
            }
        } else if (manana == 1) {
            ruta = "/" + carpeta + "/" + "7mo" + "/" + "manana";
            if (!FOLDER_EXIST2) {
                if (SD.mkdir(ruta)) {
                    Serial.println("mañana correctamente");
                    FOLDER_EXIST2 = true;
                }
            }
        }
        myFile = SD.open(ruta + "/" + file, FILE_WRITE);
        myFile.print("Asistencia 7To 'U'---");
        myFile.println(Ano);
        myFile.println(" ");
        myFile.close();
        Serial.println("Archivo 7mo ya creado");
        FOLDER_EXIST2 = false;
    }
} else if (sexta == 1) {
    file = "6to.txt";
    if (tarde == 1) {
        ruta = "/" + carpeta + "/" + "6to" + "/" + "tarde";
        if (!FOLDER_EXIST2) {
            if (SD.mkdir(ruta)) {
                Serial.println("tarde correctamente");
                FOLDER_EXIST2 = true;
            }
        }
    } else if (manana == 1) {
        ruta = "/" + carpeta + "/" + "6to" + "/" + "manana";
        if (!FOLDER_EXIST2) {
            if (SD.mkdir(ruta)) {
                Serial.println("mañana correctamente");
                FOLDER_EXIST2 = true;
            }
        }
    }
}

myFile = SD.open(ruta + "/" + file, FILE_WRITE);
myFile.print("Asistencia 6To 'U'---");
myFile.println(Ano);
myFile.println(" ");
myFile.close();
Serial.println("Archivo 6to ya creado");
}

else if (quinto == 1) {
    if (tarde == 1) {
        ruta = "/" + carpeta + "/" + "5to" + "/" + "tarde";
        if (!FOLDER_EXIST2) {
            if (SD.mkdir(ruta)) {
                Serial.println("tarde correctamente");
                FOLDER_EXIST2 = true;
            }
        }
    } else if (manana == 1) {
        ruta = "/" + carpeta + "/" + "5to" + "/" + "manana";
        if (!FOLDER_EXIST2) {
            if (SD.mkdir(ruta)) {

```



```

        Serial.println("mañana correctamente");
        FOLDER_EXIST2 = true;
    }

}

file = "5to.txt";
myFile = SD.open(ruta + "/" + file, FILE_WRITE);
myFile.print("Asistencia 5To 'U'---");
myFile.println(Ano);
myFile.println(" ");
myFile.close();
Serial.println("Archivo 5to ya creado");
}

else if (cuarto == 1) {
    file = "4to.txt";
    myFile = SD.open(ruta + "/" + file, FILE_WRITE);
    myFile.print("Asistencia 4To 'U'---");
    myFile.println(Ano);
    myFile.println(" ");
    myFile.close();
    Serial.println("Archivo 4to ya creado");
}
} else {
    Serial.println("SIN CURSO");
}

/*
myFile = SD.open("/" + file + "/" + "7mo", FILE_WRITE);
myFile.print("Asistencia 7To 'U'---");
myFile.println(Ano);
myFile.println(" ");
myFile.close();*/
}

//file.remove(2, 4); //////////remueve el .txt para hacer la comparacion
posterior
sdarchive = true; //////////variable que indica que el archivo fue creado
protec = false; //permite la generacion de una nueva lista
resetoide();

}

else if (sdarchive == true) //////////verifica si ya se creo un archivo
{
    if (msj == 0) {

        msj = 1;/////////impide que el mensaje se repita mucho en en serial
    }
}

else if (carpeta != contenedor) { // verifica si el dia ha cambiado
    carpeta = contenedor; //vuelve a igualar file name al dato rtc

    Serial.println("Fecha actualizada");
    FOLDER_EXIST = false; // habilita la creacion de una nueva carpeta del dia
    msj = 0; ////////// msj no importante
    archivoCursoSelected = 0 ;
    sdarchive = false; // vuelve a permitir la creacion de un archivo devolviendo
la variable a false
}

///////////
selectOption();

aState = digitalRead(outputA);
if (aState != aLastState) {

    if (digitalRead(outputB) != aState) {

        incremento++;
        if (incremento > 1) {
            incremento = 0;
            menu.switch_focus(false);
        }
    }
}

```

```

        }

    } else {

        incremento++;
        if (incremento > 1) {
            incremento = 0;
            menu.switch_focus(true);
        }

        menu.softUpdate();
        aLastState = aState;
    }

printDateTimeMinuto(now);
printDateTimeHora(now);

hora = String(Hora).toInt();
minut = String(Minuto).toInt();

if (VARM != minut) {
    VARM = minut;
    menu.softUpdate();
}

if (VARH != hora) {
    VARH = hora;
    menu.softUpdate();
}

}

//Funciones:::::
void selectOption() {
    if (digitalRead(sw) == LOW) {
        menu.call_function(1);
        delay(300);
    }
}
////////////////LECTOR///////////////
void LECTOR() {

    if (archivoCursoSelected == 1) {
        printDatosMatriz();
        menu.change_screen(5);
        menu.set_focusedLine(0);
    } else {
        lcd.clear();
        starttime = millis();
        while ((millis() - starttime) < 3000) {
            lcd.setCursor(6, 0);
            lcd.print("CURSO NO");
            lcd.setCursor(4, 1);
            lcd.print("SELECCIONADO");
            lcd.setCursor(1, 2);
            lcd.print("PRIMERO SELECCIONE");
            lcd.setCursor(6, 3);
            lcd.print("UN CURSO");

        }
        menu.change_screen(7);
        menu.set_focusedLine(0);
    }
}

boolean rfid_main;
boolean rfid_main2 = 0;
void lec_f(){
    limpia();
    starttime = millis();
    auto_asis();
}

```



```

if (cosita == 0) { // si aun no es la hora se requiere si o si tarjeta maestra
    while ((millis() - starttime) < 5000 and rfid_main == 0 and alerta == 0) {

        lcd.setCursor(3, 0);
        lcd.print("Esperando al");
        lcd.setCursor(4, 1);
        lcd.print("Preceptor");

        RFID_FUNCTION();

    }

} else {
    rfid_main2 = 1;
}
///////////////////
if ( alerta == 1) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 4000) {
        lcd.setCursor(0, 0);
        lcd.print("Tarjeta incorrecta");
        lcd.setCursor(0, 1);
        lcd.print("ALUMNO NO AUTORIZADO");
        lcd.setCursor(0, 2);
        lcd.print(dataArray[Indice_2][0]);
    }
}
///////////////////

if (rfid_main == 1) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 5000) {
        lcd.setCursor(0, 0);
        lcd.print("Tarjeta Correcta");
        lcd.setCursor(0, 1);
        lcd.print("Bienvenido/a ");
        lcd.setCursor(0, 2);
        lcd.print(Preceptor[Indice][0]);

    }
    RtcDateTime now = Rtc.GetDateTime();
    printDateTimeTiempo(now);
    String tiempito = String(Tiempo);

    myFile = SD.open(ruta + "/" + file, FILE_WRITE);
    myFile.print("El preceptor/a ");
    myFile.print(Preceptor[Indice][0]);
    myFile.print(" habilito la asistencia a las: ");
    myFile.println(tiempito);
    myFile.println(" ");
    //////////////////

    Serial.print("El/La preceptor/a ");
    Serial.print(Preceptor[Indice][0]);
    Serial.print(" habilito la asistencia a las: ");
    Serial.println(tiempito);
    Serial.println(" ");

    //////////////////
    myFile.close();
}

Indice = 0;
Indice_2 = 0;
alumnos_main = 0;
if ( rfid_main == 1 or rfid_main2 == 1 ) {

    limpia();
    starttime = millis();
    lcd.setCursor(0, 0);
    lcd.print("Esperando alumnos");
    lcd.setCursor(0, 1);
    lcd.print("apoyar tarjetas");

    while ((millis() - starttime) < 5000 and prueba == 0) {

```



```

lcd.setCursor(0, 0);
lcd.print("Esperando alumnos");
lcd.setCursor(0, 1);
lcd.print("apoya tarjetas");
alumnos();
///////////
if ( alumnos_main == 1) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 3000) {
        lcd.setCursor(0, 0);
        lcd.print("Tarjeta Correcta");
        lcd.setCursor(0, 1);
        lcd.print("Bienvenido/a ");
        lcd.setCursor(5, 2);
        lcd.print(dataArray[Indice_2][0]);
    }
    alumnos_main = 0;
    starttime = millis();
    limpia();
    Indice_2 = 0;
}
///////////
if ( prueba == 1) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 4000) {
        lcd.setCursor(0, 0);
        lcd.print("Tarjeta incorrecta");
        lcd.setCursor(0, 1);
        lcd.print("Se requiere tarjeta");
        lcd.setCursor(0, 2);
        lcd.print("de alumno");
    }
}
if ( rfid_main == 0 and rfid_main2 == 0 ) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 3000 ) {
        Nodetec();
    }
}
Indice = 0;
Indice_2 = -1;
rfid_main = 0;
rfid_main2 = 0;
cosita = 0;
prueba = 0;
alerta = 0;
lcd.clear();
menu.change_screen(5);
menu.setFocusedLine(0);

}

void LEC_SALIDA() {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 5000 and rfid_main == 0 and alerta == 0) {

        lcd.setCursor(3, 0);
        lcd.print("Esperando al");
        lcd.setCursor(4, 1);
        lcd.print("Preceptor");

        RFID_FUNCTION();

    }
}
///////////
if ( alerta == 1) {

```



```

limpia();
starttime = millis();
while ((millis() - starttime) < 4000) {
    lcd.setCursor(0, 0);
    lcd.print("Tarjeta incorrecta");
    lcd.setCursor(0, 1);
    lcd.print("ALUMNO NO AUTORIZADO");
    lcd.setCursor(0, 2);
    lcd.print(dataArray[Indice_2][0]);
}
///////////////////////
if (rfid_main == 1) {
    flag_salida = 1;
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 5000) {
        lcd.setCursor(0, 0);
        lcd.print("Tarjeta Correcta");
        lcd.setCursor(0, 1);
        lcd.print("Salida habilitada");
        lcd.setCursor(0, 2);
        lcd.print(Preceptor[Indice][0]);
    }
    RtcDateTime now = Rtc.GetDateTime();
    printDateTimeTiempo(now);
    String tiempito = String(Tiempo);

    myFile = SD.open(ruta + "/" + file, FILE_WRITE);
    myFile.print("El preceptor/a ");
    myFile.print(Preceptor[Indice][0]);
    myFile.print(" habilito la SALIDA a las: ");
    myFile.println(tiempito);
    myFile.println(" ");
    ///////////////////
    Serial.print("El/La preceptor/a ");
    Serial.print(Preceptor[Indice][0]);
    Serial.print(" habilito la SALIDA a las: ");
    Serial.println(tiempito);
    Serial.println(" ");
    ///////////////////
    myFile.close();
}
Indice = 0;
Indice_2 = 0;
alumnos_main = 0;
if ( rfid_main == 1 or rfid_main2 == 1 ) {

    limpia();
    starttime = millis();
    lcd.setCursor(0, 0);
    lcd.print("Esperando alumnos");
    lcd.setCursor(0, 1);
    lcd.print("apoyar tarjetas");

    while ((millis() - starttime) < 5000 and prueba == 0) {

        lcd.setCursor(0, 0);
        lcd.print("Esperando alumnos");
        lcd.setCursor(0, 1);
        lcd.print("apoya tarjetas");
        alumnos_salida();
        ///////////////////
        if ( alumnos_main == 1) {
            limpia();
            starttime = millis();
            while ((millis() - starttime) < 3000) {
                lcd.setCursor(0, 0);
                lcd.print("Tarjeta Correcta");
                lcd.setCursor(0, 1);
                lcd.print("ADIOS");
                lcd.setCursor(5, 2);
                lcd.print(dataArray[Indice_2][0]));
            }
        }
    }
}

```



```

        }
        alumnos_main = 0;
        starttime = millis();
        limpia();
        Indice_2 = 0;
    }
    ///////////////
}
if ( prueba == 1) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 4000) {
        lcd.setCursor(0, 0);
        lcd.print("Tarjeta incorrecta");
        lcd.setCursor(0, 1);
        lcd.print("Se requiere tarjeta");
        lcd.setCursor(0, 2);
        lcd.print("de alumno");
    }
}
if ( rfid_main == 0 and rfid_main2 == 0 ) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 3000 ) {
        Nodetec();
    }
}
Indice = 0;
Indice_2 = -1;
rfid_main = 0;
rfid_main2 = 0;
cosita = 0;
prueba = 0;
alerta = 0;
lcd.clear();
menu.change_screen(5);
menu.setFocusedLine(0);

}
////////////////Lector preceptores///////////////////////////
void RFID_FUNCTION() {
    if ( ! rfid.PICC_IsNewCardPresent()) {
        return;
    }

    if ( ! rfid.PICC_ReadCardSerial()) {
        return;
    }
    String tag = "";
    for (byte i = 0; i < rfid.uid.size; i++)
    {
        tag.concat(String(rfid.uid.uidByte[i] < 0x10 ? " 0" : " "));
        tag.concat(String(rfid.uid.uidByte[i], HEX));
    }

    tag.toUpperCase();
    if (tag.substring(1) != dataArray[Indice_2][1])
    {
        Indice_2++;
    }
    if (tag.substring(1) != Preceptor[Indice][1])
    {
        Indice++;
    }
    if (tag.substring(1) == Preceptor[Indice][1])
    {
        Serial.println("Access Granted " + Preceptor[Indice][0]);
        rfid_main = 1;
        rfid_main2 = 1;
    }
    if (tag.substring(1) == dataArray[Indice_2][1])
    {

```



```

        Serial.println("Acceso denegado, alumno no autorizado : " +
dataArray[Indice_2][0]);

        alerta = 1;
    }
    else {
        Serial.println(".");
    }
}
////////////////////////////Lector
alumnos///////////////////////////

```



```

void alumnos() {
    if ( ! rfid.PICC_IsNewCardPresent()) {
        return;
    }

    if ( ! rfid.PICC_ReadCardSerial()) {
        return;
    }
    String tag = "";
    for (byte i = 0; i < rfid.uid.size; i++)
    {
        tag.concat(String(rfid.uid.uidByte[i] < 0x10 ? " 0" : " "));
        tag.concat(String(rfid.uid.uidByte[i], HEX));
    }

    tag.toUpperCase();

    if (tag.substring(1) == Preceptor[0][1] or tag.substring(1) == Preceptor[1][1])
    {
        prueba = 1;
    }
    if (tag.substring(1) != dataArray[Indice_2][1])
    {
        Indice_2++;
    }

    if (tag.substring(1) == dataArray[Indice_2][1])
    {

        asistencia();//////////////////SETEA ESTADO DE ALUMNO ENTRADA
        ALUMNOS[Indice_2] = pres;
        Serial.println("Alumno registrado: " + dataArray[Indice_2][0]);
        SET_ESTADO();
        Serial.println(dataArray[Indice_2][0] + " " + alumno_estado);
        alumnos_main = 1;
    }
    /*else {
        Serial.println(".");
    }*/
}

```



```

///////////////////////////////
void alumnos_salida() {
    if ( ! rfid.PICC_IsNewCardPresent()) {
        return;
    }

    if ( ! rfid.PICC_ReadCardSerial()) {
        return;
    }
    String tag = "";
    for (byte i = 0; i < rfid.uid.size; i++)
    {
        tag.concat(String(rfid.uid.uidByte[i] < 0x10 ? " 0" : " "));
        tag.concat(String(rfid.uid.uidByte[i], HEX));
    }

    tag.toUpperCase();
}

```



```

        if (tag.substring(1) == Preceptor[0][1] or tag.substring(1) == Preceptor[1][1])
    ) {
        prueba = 1;
    }
    if (tag.substring(1) != dataArray[Indice_2][1])
    {
        Indice_2++;
    }

    if (tag.substring(1) == dataArray[Indice_2][1])
    {
        Salida_alumn[Indice_2] = 1; ////////////////SETEA ESTADO ALUMNO SALIDA
        Serial.println("Alumno registrado: " + dataArray[Indice_2][0] + " PRESENTE A LA SALIDA");
        alumnos_main = 1;
    }
}

////////////////////SD FUNCTIONS////////////////////

void EXSD() {
    menu.change_screen(3);
    menu.set_focusedLine(0);
}

/////////////////////
void SERVOSD() {
    lcd.clear();
    limpia();
    starttime = millis();

    while ((millis() - starttime) < 4000 and rfid_main == 0 and alerta == 0) {
        lcd.setCursor(3, 0);
        lcd.print("Esperando al");
        lcd.setCursor(4, 1);
        lcd.print("Preceptor");

        RFID_FUNCTION();
    }
}

if (rfid_main == 1) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 3000 ) {
        lcd.setCursor(3, 0);
        lcd.print("Tarjeta leida");
        lcd.setCursor(4, 1);
        lcd.print("Correcta!");
    }
}

while (digitalRead(fcarreral) == HIGH) {
    lcd.setCursor(3, 0);
    lcd.print("ABRIENDO");
    lcd.setCursor(3, 1);
    lcd.print("COMPUERTA");

    analogWrite(OPEN, 140);
    //servo.write(0);
}

if (digitalRead(fcarreral) == LOW) {
    //puerta abierta
    analogWrite(OPEN, 0);
}
delay(150);
COMopen();

```



```

//////////FUNCIONES AHORRO///////////
void MenuAhorro() {
    menu.change_screen(4);
    menu.set_focusedLine(0);
}

//////////FUNCIONES AHORRO/////////
lcd.clear();
starttime2 = millis();
while ((millis() - starttime2) < 10000 and SD.begin(4)) // verifica sd y tiempo
de espera
{
    lcd.setCursor(5, 1);
    lcd.print("EXTRAIGA");
    lcd.setCursor(8, 2);
    lcd.print("SD");
}
lcd.clear();
if (SD.begin(4)) {
    lcd.clear();
    starttime = millis();
    while ((millis() - starttime) < 3000) //
    {
        lcd.setCursor(6, 1);
        lcd.print("TARJETA");
        lcd.setCursor(4, 2);
        lcd.print("NO EXTRAIDA");
    }
}
while (!SD.begin(4)) {// entra aca si no esta la sd
    msj1();
    msj2();
    sd_state = 0;
}
mensaje siguiente si se reinserto la sd
delay(150);
if (SD.begin(4) and sd_state == 0) {
    msjsd();
    sd_state = 1;
}

//////////FUNCIONES AHORRO/////////
lcd.clear();

delay(150);
while (digitalRead(fcarrera2) == HIGH && SD.begin(4) ) {//cerrando puertas
    //servo.write(180);

    lcd.setCursor(6, 1);
    lcd.print("CERRANDO");
    lcd.setCursor(6, 2);
    lcd.print("COMPUERTA");
    analogWrite(CLOSE, 150);
}

if (digitalRead(fcarrera2) == LOW) {
    //puerta cerrada
    analogWrite(CLOSE, 0);
}

delay(250);
COMclose();

lcd.clear();

menu.change_screen(1);
menu.set_focusedLine(0);
}
rfid_main = 0;
}

//////////FUNCIONES AHORRO/////////

```



```

void OFF() {
    lcd.noBacklight();
}

void ON() {
    lcd.backlight();
}

}

////////////////// FUNCIONES EXTRA //////////////////////////////

void EXTRAS() {
    menu.change_screen(2);
    menu.set_focusedLine(0);
}

void Credits() {

}

///////////////////////////////ATRAS //////////////////////////////

void fn_atras() {
    menu.change_screen(1);
    menu.set_focusedLine(0);
}

/////////////////////////////msj1////////////////////////////

void msj1() {
    lcd.clear();
    msjt1 = millis();
    while ((millis() - msjt1) < 1000) {
        lcd.setCursor(6, 1);
        lcd.print("TARJETA");
        lcd.setCursor(6, 2);
        lcd.print("EXTRAIDA");
    }
}

void msj2() {
    lcd.clear();
    msjt2 = millis();
    while ((millis() - msjt2) < 1000) {
        lcd.setCursor(5, 0);
        lcd.print("REINERTE");
        lcd.setCursor(6, 1);
        lcd.print("TARJETA");
        lcd.setCursor(5, 2);
        lcd.print("POR FAVOR");
    }
}

//////////////////COMclose////////////////////////////

void COMclose() {
    lcd.clear();
    starttime = millis();
    while ((millis() - starttime) < 2000) {
        lcd.setCursor(5, 1);
        lcd.print("COMPUERTA");
        lcd.setCursor(6, 2);
        lcd.print("CERRADA");
    }
}

//////////////////COMopen////////////////////////////

void COMopen() {
    lcd.clear();
    startime1 = millis();
    while ((millis() - startime1) < 2000) {
        lcd.setCursor(5, 1);
        lcd.print("COMPUERTA");
        lcd.setCursor(6, 2);
        lcd.print("ABIERTA");
    }
}

```



```

    }
}

///////////////////////////////
void msjsd() {
    lcd.clear();
    starttime = millis();
    while ((millis() - starttime) < 3000) {
        lcd.setCursor(6, 1);
        lcd.print("TARJETA");
        lcd.setCursor(5, 2);
        lcd.print("INSERTADA!");
        sd_state = 1;
    }
}

/////////////////////////////
ASISTENCIA
////////////////////////////

void tardanza() {
    if (manana == 1) {
        if (hora <= 7) { //los valores son de prueba
            ESTADO = 1;
        }
    } else if (tarde == 1) {
        if (hora <= 14) { //los valores son de prueba
            ESTADO = 1;
        }
    }
    else if (tarde == 1 and septimo == 1) {
        if (hora <= 17) { //los valores son de prueba
            ESTADO = 1;
        }
    }
}

void veriftime() {
    if (manana == 1) {
        if (hora == 7 && minut > 35 and minut < 50) { //los valores son de prueba
            ESTADO = 2;
        }
    } else if (tarde == 1) {
        if (hora == 14 && minut > 35 and minut < 50) { //los valores son de prueba
            ESTADO = 2;
        }
    } else if (tarde == 1 and septimo == 1) {
        if (hora == 17 && minut > 35 and minut < 50) { //los valores son de prueba
            ESTADO = 2;
        }
    }
}

void asistencia() {
    if (manana == 1) {
        if (hora > 7) {
            ESTADO = 3;
        }
    }
    if (tarde == 1) {
        if (hora > 14) {
            ESTADO = 3;
        }
    }
    if (tarde == 1 and septimo == 1) {
        if (hora > 17) {
            ESTADO = 3;
        }
    }
}

tardanza();
veriftime();

if (ESTADO == 1) {
    pres = 1;
}

```



```

        }
    else if (ESTADO == 2) {
        pres = 2;
    }
    else if (ESTADO == 3) {
        pres = 3;
    }
    ESTADO = 0;
}

///////////////////////////////
void Nodetect() {
    lcd.setCursor(0, 0);
    lcd.print("Tarjeta Maestra");
    lcd.setCursor(0, 1);
    lcd.print("No encontrada!");
}

void limpia() {
    lcd.setCursor(0, 0);
    lcd.print("                ");
    lcd.setCursor(0, 1);
    lcd.print("                ");
    lcd.setCursor(0, 2);
    lcd.print("                ");
    lcd.setCursor(0, 3);
    lcd.print("                ");
}
//////////////////RTC
VOIDs/////////////////
#define countof(a) (sizeof(a) / sizeof(a[0]))
void printDateTime(const RtcDateTime & dt)
{
    //char datestring[20];

    snprintf_P(datestring,
               countof(datestring),
               PSTR("%02u/%02u/%04u %02u:%02u:%02u"),
               dt.Month(),
               dt.Day(),
               dt.Year(),
               dt.Hour(),
               dt.Minute(),
               dt.Second());
    Serial.print(datestring);
    return datestring;
}

void printDateTimeTiempo(const RtcDateTime & dt)
{
    //char datestring[20];

    snprintf_P(Tiempo,
               countof(Tiempo),
               PSTR("%02u:%02u:%02u"),
               dt.Hour(),
               dt.Minute(),
               dt.Second());
}

void printDateTimeFecha(const RtcDateTime & dt) {

    snprintf_P(Fecha,
               countof(Fecha),
               PSTR("%02u-%02u"),
               dt.Day(),
               dt.Month()
               );
}

```



```

    return Fecha;
}

void printDateMes(const RtcDateTime & dt)
{
    //char datestring[20];

    snprintf_P(Mes,
               countof(Mes),
               PSTR("%02u/%02u/%04u %02u:%02u:%02u"),
               dt.Month() );

    return Mes;
}

void printDateTimeDia(const RtcDateTime & dt)
{
    //char datestring[20];

    snprintf_P(Dia,
               countof(Dia),
               PSTR("%02u"),
               dt.Day() );

    return Dia;
}

void printDateTimeHora(const RtcDateTime & dt)
{
    //char datestring[20];

    snprintf_P(Hora,
               countof(Hora),
               PSTR("%02u"),
               dt.Hour() );

    return Hora;
}

void printDateTimeMinuto(const RtcDateTime & dt)
{
    //char datestring[20];

    snprintf_P(Minuto,
               countof(Minuto),
               PSTR("%02u"),
               dt.Minute() );

    return Minuto;
}

///////////
void GEN() {
//
if (archivoCursoSelected == 1) {
    if (flag_salida == 1) {
        limpia();
        starttime = millis();

        while ((millis() - starttime) < 4000 and rfid_main == 0 and alerta == 0) {

            lcd.setCursor(3, 0);
            lcd.print("Esperando al");
            lcd.setCursor(4, 1);
            lcd.print("Preceptor");

            RFID_FUNCTION();

        }
    }
}
}

```



```

///////////
if (rfid_main == 1) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 3000 ) {

        lcd.setCursor(3, 0);
        lcd.print("Tarjeta leida");
        lcd.setCursor(4, 1);
        lcd.print("Correcta!");

    }
}
///////////



if (protec == false and rfid_main == 1 ) {///escritura de la asistencia

    Indice_2 = -1;
    Indice = -1;

    myFile = SD.open(ruta + "/" + file, FILE_WRITE);
    Serial.println("ENTRADA: ");
    myFile.println("ENTRADA: ");
    for (byte i = 0; i < 25; i ++ ) {
        Indice_2++;
        SET_ESTADO();
        myFile.print(dataArray[Indice_2][0]);
        myFile.print(" ");

        myFile.println(alumno_estado);
        myFile.println(" ");

        Serial.print(dataArray[Indice_2][0]);
        Serial.print(" ");
        //1V32

        Serial.println(alumno_estado);
        Serial.println(" ");

    }
    myFile.println("/////////");
    myFile.println("SALIDA: ");

    Serial.println("/////////");
    Serial.println("SALIDA: ");
    entrance_compare();

    myFile.println("/////////");
    myFile.println("ALUMNOS FALTANTES A LA SALIDA:");

    Serial.println("/////////");
    Serial.println("ALUMNOS FALTANTES A LA SALIDA:");

    print_faltantes();
    Serial.println("");
    Serial.println("/////////ASISTENCIA CERRADA/////////");

    myFile.println("");
    myFile.println("/////////ASISTENCIA CERRADA/////////");
    myFile.close();

    protec = true;
    Indice_2 = -1;
    Indice = -1;

}
else if (protec == true and rfid_main == 1) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 4000) {
        lcd.setCursor(0, 0);
        lcd.print("Ya ha generado el");
        lcd.setCursor(0, 1);
        lcd.print("archivo de hoy");
    }
}

```



```

if ( rfid_main == 0 and alerta == 0 ) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 3000 ) {
        Nodetec();
    }
}

if ( alerta == 1) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 4000) {
        lcd.setCursor(0, 0);
        lcd.print("Tarjeta incorrecta");
        lcd.setCursor(0, 1);
        lcd.print("ALUMNO NO AUTORIZADO");
        lcd.setCursor(0, 2);
        lcd.print(dataArray[Indice_2][0]);
    }
}

lcd.clear();
rfid_main = 0;
rfid_main2 = 0;
alerta = 0;
Indice_2 = -1;
Indice = -1;
} else {
    lcd.clear();
    starttime = millis();
    while ((millis() - starttime) < 3000) {
        lcd.setCursor(6, 0);
        lcd.print("NO HA");
        lcd.setCursor(4, 1);
        lcd.print("HABILITADO");
        lcd.setCursor(5, 2);
        lcd.print("LA SALIDA");
    }
    menu.change_screen(5);
    menu.setFocusedLine(0);
}
} else {
    lcd.clear();
    starttime = millis();
    while ((millis() - starttime) < 3000) {
        lcd.setCursor(6, 0);
        lcd.print("CURSO NO");
        lcd.setCursor(4, 1);
        lcd.print("SELECCIONADO");
        lcd.setCursor(1, 2);
        lcd.print("PRIMERO SELECCIONE");
        lcd.setCursor(6, 3);
        lcd.print("UN CURSO");
    }
    menu.change_screen(7);
    menu.setFocusedLine(0);
}

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////void para activacion sin
tarjeta maestra despues de cierta hora
void auto_asis() {
    if (tarde == 1) {
        if (hora == 2 && minut > 30 and minut < 59) {
            cosita = 1; // boolean para operar en el void del lector
        }
    }

    if (tarde == 1 and septimo == 1) {
        if (hora == 5 && minut > 20 and minut < 59) {
            cosita = 1; // boolean para operar en el void del lector
        }
    }

    if (manana == 1) {
        if (hora == 7 && minut > 50 and minut < 59) {
            cosita = 1; // boolean para operar en el void del lector
        }
    }
}

```



```

        }
    }
////////////////////////////////////////////////////////////////RESET FILE////////////////////////////////////////////////////////////////
//resetea manualmente el archivo
void RES() {
    limpia();
    starttime = millis();

    while ((millis() - starttime) < 4000 and rfid_main == 0 and alerta == 0) {

        lcd.setCursor(3, 0);
        lcd.print("Esperando al");
        lcd.setCursor(4, 1);
        lcd.print("Preceptor");

        RFID_FUNCTION();

    }

    ///////////////////
    if (rfid_main == 1) {
        limpia();
        starttime = millis();
        while ((millis() - starttime) < 3000 ) {

            lcd.setCursor(3, 0);
            lcd.print("Tarjeta leida");
            lcd.setCursor(4, 1);
            lcd.print("Correcta!");

        }
    }
    ///////////////////

    Indice = 0;
    delay(400);
    if (rfid_main == 1) {
        limpia();
        starttime = millis();
        rfid_main = 0;

        while ((millis() - starttime) < 6000 and rfid_main == 0 ) {

            lcd.setCursor(3, 0);
            lcd.print("Se requiere");
            lcd.setCursor(4, 1);
            lcd.print("confirmacion!");
            RFID_FUNCTION();

        }
    }
    Indice = 0;

    limpia();
    starttime = millis();
    if (rfid_main == 1 ) {
        protec = false;
        sdarchive = false;
        msj = 0;
        SD.remove(ruta + "/" + file);

        if (!SD.exists(ruta + "/" + file)) {
            Serial.println("archivo borrado");
        }
        else {
            Serial.println("archivo no borrado");
        }
    }

    while ((millis() - starttime) < 4000 ) {

        lcd.setCursor(3, 0);
        lcd.print("Reseteo realizado");
        lcd.setCursor(3, 1);
        lcd.print("archivo removido!");

    }
}

```



```

}
///////////////////////////////
if ( rfid_main == 0 and alerta == 0 ) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 3000) {
        Nodetec();
    }
}
///////////////////////////////

if ( alerta == 1) {
    limpia();
    starttime = millis();
    while ((millis() - starttime) < 4000) {
        lcd.setCursor(0, 0);
        lcd.print("Tarjeta incorrecta");
        lcd.setCursor(0, 1);
        lcd.print("ALUMNO NO AUTORIZADO");
        lcd.setCursor(0, 2);
        lcd.print(dataArray[Indice_2][0]);
    }
}

lcd.clear();
rfid_main = 0;
rfid_main2 = 0;
Indice = 0;
Indice_2 = -1;
alerta = 0;
}
//////////////////FUNCION      PARA      REINICIAR      EL      ESTADO      DE      LOS
ALUMNOS////////////////

void resetoide() {
    Indice_2 = -1;
    for (byte i = 0; i < 40 ; i++) {
        Indice_2++;
        ALUMNOS[Indice_2] = 0;
        Salida_alumn[Indice_2] = 0;
        FALTANTES[Indice_2] = " ";
    }
    Indice_2 = -1;
}
}

//////////////////TURNOS MENUS////////////////

```

```

void TURNO() {

if (archivoCursoSelected == 1) {
    menu.change_screen(6);
    menu.set_focusedLine(0);
} else {
    lcd.clear();
    starttime = millis();
    while ((millis() - starttime) < 3000) {
        lcd.setCursor(6, 0);
        lcd.print("CURSO NO");
        lcd.setCursor(4, 1);
        lcd.print("SELECCIONADO");
        lcd.setCursor(1, 2);
        lcd.print("PRIMERO SELECCIONE");
        lcd.setCursor(6, 3);
        lcd.print("UN CURSO");

    }
    menu.change_screen(7);
    menu.set_focusedLine(0);
}
}

```



```

void T_tarde() {
    manana = 0;
    tarde = 1;
    sdarchive = false;
    lcd.clear();
    starttime = millis();
    Serial.println("Turno tarde");
    while ((millis() - starttime) < 3000) {
        lcd.setCursor(0, 0);
        lcd.print("Turno tarde");
        lcd.setCursor(0, 1);
        lcd.print("seleccionado");
    }
    menu.change_screen(5);
    menu.setFocusedLine(0);
}

void T_man() {
    manana = 1;
    tarde = 0;
    sdarchive = false;
    Serial.println("Turno manana");
    lcd.clear();
    starttime = millis();
    while ((millis() - starttime) < 3000) {
        lcd.setCursor(0, 0);
        lcd.print("Turno manana");
        lcd.setCursor(0, 1);
        lcd.print("seleccionado");
    }
    menu.change_screen(5);
    menu.setFocusedLine(0);
}

//////////////////comparacion entrada-salida///////////////////
void entrance_compare() {
    Indice_2 = -1;
    String nameF;
    for (int j = 0; j < 40; j++) {
        Indice_2++;

        if (ALUMNOS[Indice_2] == 1 and Salida_alumn[Indice_2] == 1) {
            Serial.println(dataArray[Indice_2][0] + " PRESENTE A LA SALIDA");
            myFile.println(dataArray[Indice_2][0] + " PRESENTE A LA SALIDA");

        }
        else if (ALUMNOS[Indice_2] == 0 and Salida_alumn[Indice_2] == 0 and
dataArray[Indice_2][0] != "") {
            Serial.println(dataArray[Indice_2][0] + " ALUMNO AUSENTE");
            myFile.println(dataArray[Indice_2][0] + " ALUMNO AUSENTE");

        }

        else if (ALUMNOS[Indice_2] == 2 and Salida_alumn[Indice_2] == 0) { //<-----
        ----
            Serial.println(dataArray[Indice_2][0] + " ALUMNO ESCAPADO");
            myFile.println(dataArray[Indice_2][0] + " ALUMNO ESCAPADO");
            nameF = dataArray[Indice_2][0];
            FALTANTES[Indice_2] = nameF;
        }
        else if (ALUMNOS[Indice_2] == 3 and Salida_alumn[Indice_2] == 0) { //<-----
        ----
            Serial.println(dataArray[Indice_2][0] + " ALUMNO ESCAPADO");
            myFile.println(dataArray[Indice_2][0] + " ALUMNO ESCAPADO");
            FALTANTES[Indice_2] = nameF;
        }
        else if (ALUMNOS[Indice_2] == 1 and Salida_alumn[Indice_2] == 0) { //<-----
        ----
            Serial.println(dataArray[Indice_2][0] + " ALUMNO ESCAPADO");
            myFile.println(dataArray[Indice_2][0] + " ALUMNO ESCAPADO");
            nameF = dataArray[Indice_2][0];
        }
    }
}

```



```

        Serial.println(nameF);

        FALTANTES[Indice_2] = nameF;
    }
    else if (ALUMNOS[Indice_2] == 2 and Salida_alumn[Indice_2] == 1) {
        Serial.println(dataArray[Indice_2][0] + " ALUMNO PRESENTE");
        myFile.println(dataArray[Indice_2][0] + " ALUMNO PRESENTE");
    }
    else if (ALUMNOS[Indice_2] == 3 and Salida_alumn[Indice_2] == 1) {
        Serial.println(dataArray[Indice_2][0] + " ALUMNO AUSENTE C/ PERMANENCIA
EN EL AULA ");
        myFile.println(dataArray[Indice_2][0] + " ALUMNO AUSENTE C/ PERMANENCIA
EN EL AULA ");
    }

}
Indice_2 = -1;
}

void print_faltantes() {
Indice_2 = -1;
for (byte i = 0; i < 40; i++) {
    Indice_2++;
    if (FALTANTES[Indice_2] == "o") {
        Serial.print("a");
        myFile.print("a");
    }
    else {
        Serial.print(FALTANTES[Indice_2]);
        Serial.print("-");
        myFile.print(FALTANTES[Indice_2]);
        myFile.print("-");
    }
}
Indice_2 = -1;
}

//////////////////////Curso MENUS///////////////////

```

```

void curso_selector() {
menu.change_screen(7);
menu.setFocusedLine(0);
}

void CURSOS_septimo() {
septimo = 1;
curso_seleccionado = true;

cuarto = quinto = sexto = 0;
lcd.clear();
starttime = millis();
while ((millis() - starttime) < 3000) {
    lcd.setCursor(0, 0);
    lcd.print("CURSO: SEPTIMO");
    lcd.setCursor(1, 1);
    lcd.print("seleccionado");

}
file_selector();
menu.change_screen(6);
menu.setFocusedLine(0);
}

void CURSOS_cuarto() {
cuarto = 1;
curso_seleccionado = true;

septimo = quinto = sexto = 0;
lcd.clear();
starttime = millis();

```



```

while ((millis() - starttime) < 3000) {
    lcd.setCursor(0, 0);
    lcd.print("CURSO: CUARTO");
    lcd.setCursor(1, 1);
    lcd.print("seleccionado");

}

file_selector();
menu.change_screen(6);
menu.setFocusedLine(0);

}

void CURSOS_quinto() {
    quinto = 1;
    curso_seleccionado = true;

    cuarto = septimo = sexto = 0;
    lcd.clear();
    starttime = millis();
    while ((millis() - starttime) < 3000) {
        lcd.setCursor(0, 0);
        lcd.print("CURSO: QUINTO");
        lcd.setCursor(1, 1);
        lcd.print("seleccionado");

    }

    file_selector();
    menu.change_screen(6);
    menu.setFocusedLine(0);

}

void CURSOS_sexto() {
    sexto = 1;
    curso_seleccionado = true;

    cuarto = quinto = septimo = 0;
    lcd.clear();
    starttime = millis();
    while ((millis() - starttime) < 3000) {
        lcd.setCursor(0, 0);
        lcd.print("CURSO: SEXTO");
        lcd.setCursor(1, 1);
        lcd.print("seleccionado");

    }

    file_selector();
    menu.change_screen(6);
    menu.setFocusedLine(0);

}

void V() {

}

void SET_ESTADO() {
    if (ALUMNOS[Indice_2] == 1) {
        alumno_estado = "PRESENTE";
    }
    else if (ALUMNOS[Indice_2] == 2) {
        alumno_estado = "TARDANZA";
    }
    else if (ALUMNOS[Indice_2] == 3) {
        alumno_estado = "AUSENTE C/PERMANENCIA EN EL AULA";
    }
    else if (ALUMNOS[Indice_2] == 0) {
        alumno_estado = "AUSENTE";
    }
}

void file_selector() {
    eliminarDatosMatriz();
    if (septimo == 1) {
        myFile = SD.open("LF_7mo.txt");
    }
}

```



```

        Serial.println("archivo septimo");
    } else if ( cuarto == 1 ) {
        myFile = SD.open("4to.txt");
        Serial.println("archivo cuarto");
    } else if ( quinto == 1 ) {
        myFile = SD.open("5to.txt");
        Serial.println("archivo quinto");
    } else if ( sexto == 1 ) {
        myFile = SD.open("6to.txt");
        Serial.println("archivo sexto");
    }
    archivoCursoSelected = 1;

    // Leer los datos del archivo y colocarlos en la matriz
    if (myFile) {
        byte row = 0; // Fila actual de la matriz
        while (myFile.available() && row < numRows) {
            // Leer la siguiente línea de datos del archivo
            dataString = myFile.readStringUntil('\n');

            // Separar los nombres y los IDs utilizando la función de división de
            // cadena
            byte separatorIndex = dataString.indexOf(',');
            names = dataString.substring(0, separatorIndex);
            id = dataString.substring(separatorIndex + 1);

            // Colocar los nombres y los IDs en la matriz
            dataArray[row][0] = names;
            dataArray[row][1] = id;

            row++; // Avanzar a la siguiente fila de la matriz
        }

        // Cerrar el archivo de datos
        myFile.close();
    }
    Serial.println("TERMINO LA LECTURA");
    printDatosMatriz();
    flag_salida = 0;
}
void eliminarDatosMatriz() {
    for (byte row = 0; row < numRows; row++) {
        dataArray[row][0] = ""; // Elimina los datos asignando una cadena vacía
        dataArray[row][1] = ""; // Elimina los datos asignando una cadena vacía
    }
    Serial.println("Matriz reiniciada");
}
void printDatosMatriz() {
    for (byte row = 0; row < numRows; row++) {
        Serial.print(dataArray[row][0]);
        Serial.println(dataArray[row][1]);
    }
}

```



13. CONCLUSIONES

Resumen

SAERTA ha permitido optimizar el tiempo empleado en la asistencia de los alumnos, eliminando la necesidad de recopilar manualmente la información y reduciendo la posibilidad de errores humanos. Mediante el uso de tecnología Arduino y RFID, el sistema ofrece una forma rápida y eficiente de identificar a los estudiantes, registrando su asistencia de manera automática y precisa.

La utilidad del sistema en el laboratorio de electrónica radica en su capacidad para simplificar y acelerar el proceso de asistencia, liberando tiempo valioso para actividades más productivas. Además, al eliminar la necesidad de registros manuales, se garantiza una mayor fiabilidad en los datos recopilados, minimizando posibles discrepancias o confusiones.

En resumen, el sistema de control de asistencia automatizado SAERTA ha demostrado ser una solución altamente útil en el laboratorio de electrónica. Su capacidad para aumentar la rapidez del proceso de asistencia y garantizar su fiabilidad ha mejorado significativamente la eficiencia y la productividad en el entorno educativo.

Logros Alcanzados

Durante el desarrollo del proyecto SAERTA, hemos logrado alcanzar diversos hitos significativos que han contribuido a nuestro crecimiento y aprendizaje en el campo de la programación y el manejo de dispositivos tecnológicos. A continuación, resumimos algunos de los logros más destacados:

1. Ampliación de conocimientos de programación: A través de este proyecto, hemos consolidado y ampliado nuestros conocimientos de programación, aplicando conceptos y técnicas avanzadas en lenguajes como C++ y Arduino. Hemos adquirido habilidades en el manejo de datos, manipulación de arrays y mejora de nuestra lógica de programación.
2. Manejo de dispositivos: Durante el desarrollo de SAERTA, hemos tenido la oportunidad de trabajar con diferentes dispositivos tecnológicos, como tarjetas SD y módulos RFID. Esta experiencia nos ha permitido adquirir conocimientos prácticos sobre su funcionamiento, configuración e integración en el sistema.
3. Diseño en 3D con SketchUp: Para mejorar la presentación y funcionalidad de nuestro proyecto, hemos aprendido a utilizar SketchUp, una herramienta de diseño en 3D. Gracias a esto, pudimos diseñar una carcasa personalizada para el dispositivo SAERTA, adaptándola a nuestras necesidades y requisitos específicos.

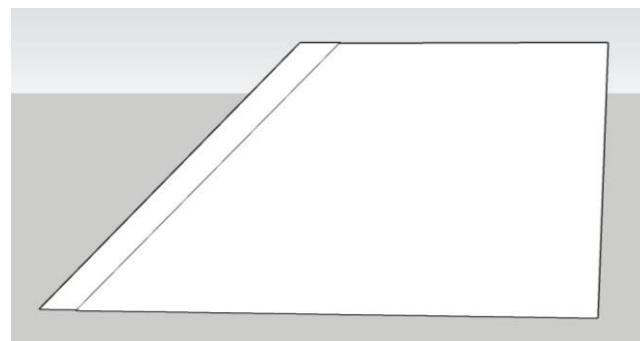


Estos logros nos han proporcionado un sólido bagaje de habilidades técnicas y conocimientos prácticos, que nos serán útiles en futuros proyectos y desafíos en el campo de la tecnología. Nos sentimos orgullosos de haber superado obstáculos, adquirido nuevos conocimientos y alcanzado los objetivos que nos propusimos en este proyecto.

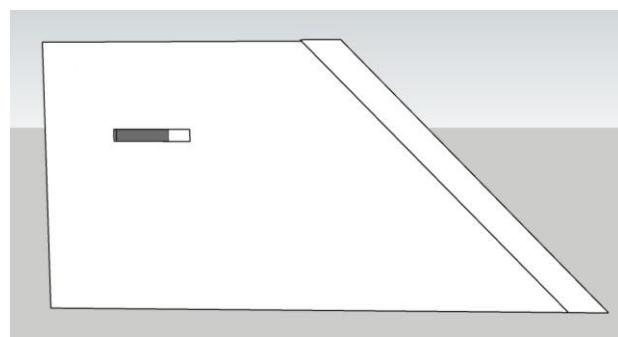
14. ANEXO

DISEÑO (PRIMER PROTOTIPO 2022)

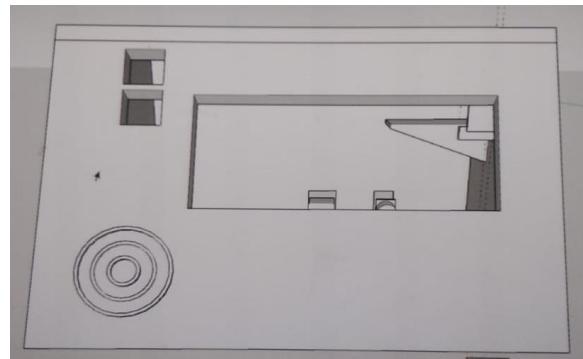
VISTA LATERAL DERECHA



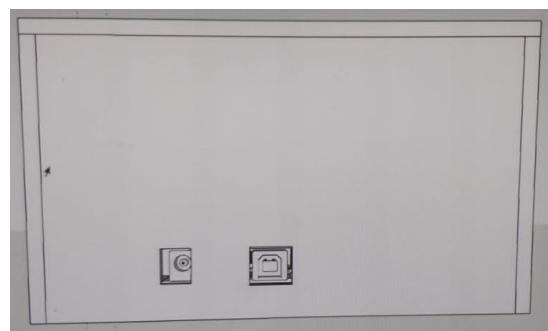
VISTA LATERAL IZQUIERDA



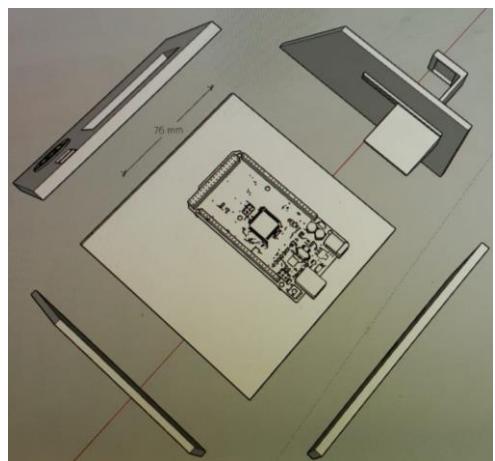
VISTA ANTERIOR



VISTA POSTERIOR



VISTA INTERIOR



ANTECEDENTES

- **SISTEMA INFORMÁTICO PARA AUTOMATIZAR EL CONTROL DE ASISTENCIA A CLASES DE DOCENTES Y ESTUDIANTES DE LA ESPOCH CON DISPOSITIVOS RFID, APLICANDO LA METODOLOGÍA SCRUMBAN**

AUTOR(ES):

- Bryan Alexander Baldeón Hermida
- Steven Alejandro

TUTOR: ING. BYRON ERNESTO VACA BARAHONA. PhD

Escuela Superior Politécnica de Chimborazo (ESPOCH) Riobamba-Ecuador

AÑO: 2019

El presente trabajo tuvo como objetivo el desarrollo de un sistema informático para automatizar el control de asistencia a clases de docentes y estudiantes de la Escuela Superior Politécnica de Chimborazo (ESPOCH) con dispositivos de identificación por radio frecuencia (RFID), a fin de optimizar los tiempos de registro de asistencia de estudiantes y docentes, también la generación de reportes que incluye el porcentaje de asistencia a la fecha actual.

Para identificar la problemática principal se usaron técnicas como la observación y la entrevista, aplicadas en la Escuela de Ingeniería en Sistemas de la ESPOCH, permitiendo recolectar los requerimientos del sistema.

Con respecto al hardware se realizó un prototipo de un dispositivo RFID, el equipo desarrollador utilizó ARDUINO MEGA y un lector RFID. La codificación del sistema se ejecutó mediante tecnologías tales como: lenguaje de programación java y C++, el servidor de aplicaciones Payara, mientras que para el almacenamiento de los datos se manejó PostgreSQL.

Para evaluar la calidad del software se enfocó en la característica productividad, se establecieron dos métricas: tiempo en completar una tarea y proporción productiva; basadas en el estándar de calidad de uso ISO/IEC 9126-4, para lo cual se aplicó el método de captura de tiempos, los datos obtenidos fueron evaluados mediante la prueba estadística T-student, dando como resultado la aceptación de la hipótesis alternativa la cual dice que: el sistema informático para automatizar el control de asistencia a clases de docente y estudiantes reduce el tiempo al realizar el proceso de registro de asistencia, mejorando la productividad en la institución

Fuente: [DSpace ESPOCH.: Sistema informático para automatizar el control de asistencia a clases de docentes y estudiantes de la ESPOCH con dispositivos RFID, aplicando la metodología SCRUMBAN.](#)

[18T00792.pdf \(esepoch.edu.ec\)](#)

- **SISTEMA PARA EL CONTROL DE ASISTENCIA,
SOLICITUD-ENTREGA Y RECEPCIÓN DE EQUIPOS
DEL LABORATORIO DE ELECTRÓNICA DE LA
UNIDAD CENTRAL DEL VALLE DEL CAUCA,
TULUÁ-VALLE, COLOMBIA :**

AUTOR(ES):

- Angela Marcela Goyes Ramírez
- William Andrés Herrera Loaiza

FACULTAD: Ingeniería

PROGRAMA: Ingeniería Electrónica

DIRECTOR: Ing. Andrés Rey Piedrahita

AÑO 2018

SIRCLab (Sistema de registro y control del laboratorio) es un dispositivo que facilita llevar el control de asistencia de los estudiantes de electrónica que realizan prácticas en este laboratorio además de permitir efectuar la solicitud de equipos para realizar dichas prácticas.

Este dispositivo cuenta con una interfaz gráfica desarrollada para que los usuarios puedan interactuar firmando su asistencia por medio de tarjetas RFID, realizar la solicitud de equipos y entrega de los mismos; el dispositivo se construye sobre una Raspberry PI 3 modelo B que se encarga de ejecutar la interfaz y hacer la comunicación serial a 9600 baudios entre el software y el módulo RFID RDM 6300, cada tarjeta RFID será asignada a un estudiante de electrónica con sus datos personales así cada vez que ingresa al laboratorio solo deberá pasar su tarjeta por el dispositivo y este se encargara de llenar en una base de datos en PostgreSQL la hora y fecha de asistencia de la misma manera cuando requiera solicitar un equipo para su práctica pasará su tarjeta por el módulo en una ventana de solicitud y escogerá de la interfaz los equipos y la cantidad requerida y esa solicitud se cargará a el estudiante dueño de la tarjeta.

El dispositivo permitirá llevar un control digital de la información que el laboratorista requiere como el control de asistencia y solicitud de equipos, de esta manera será más sencillo llevar los reportes que él debe presentar a la universidad sobre estas actividades.

Fuente: [Sistema para el control de asistencia, solicitud- entrega y recepción de equipos del laboratorio de electrónica de la Unidad Central del Valle del Cauca, Tuluá- Valle. \(uceva.edu.co\)](#)

[T00030917.pdf \(uceva.edu.co\)](#)



CONTINUACIONES

1. **Integración en la nube:** En lugar de almacenar los registros de asistencia en una tarjeta SD, se puede considerar enviarlos directamente a un servidor en la nube. Esto permitiría un acceso más fácil a los datos desde diferentes ubicaciones y dispositivos, y también posibilitaría la generación de informes automatizados, estadísticas y análisis adicionales. Lo que se debe hacer es. O bien reemplazar la placa Arduino MEGA por una placa ESP ya que permite la comunicación inalámbrica de datos a través de WiFi y bluetooth y su programación no es muy diferente a la de Arduino.
2. **Aplicación móvil:** Desarrollar una aplicación móvil que funcione en conjunto con el sistema de asistencia RFID. La aplicación podría permitir a los estudiantes verificar su asistencia en tiempo real, recibir notificaciones de cambios en el horario o anuncios importantes, y acceder a recursos adicionales relacionados con el laboratorio de electrónica.
3. **Integración con sistemas de gestión educativa:** Conectar el sistema de asistencia a los sistemas de gestión educativa existentes de la institución, como bases de datos estudiantiles. Esto podría facilitar la generación automática de informes de asistencia, sincronizar la información con otras aplicaciones y simplificar la administración de los datos del estudiante. Esto ya tiene otro grado de complejidad, porque primero se debe crear un sistema de gestión de datos de la institución alojado en la nube, para después vincularlo al dispositivo

COMPONENTES UTILIZADOS

Lectores RFID

Los lectores son los dispositivos que envían y reciben ondas de radio contra el Tag-RFID u otro dispositivo RFID.

Podemos encontrar lectores fijos y Móviles:

- **Lectores Fijos:** Instalados en lugares fijos como controles de acceso, paredes, arcos de entrada, o en el interior de un autobús.
- **Lectores Móviles:** También llamados dispositivos de mano. Entrarían en esta clasificación todos los móviles con lector RFID así como PDA con lector específico para la lectura de Tags en el almacén logístico.

El módulo lector RFID utilizado para este proyecto es el RC522 RF





El módulo lector RFID-RC522 RF utiliza 3.3V como voltaje de alimentación y se controla a través del protocolo SPI. También puede ser controlada con un puerto UART. Entonces, podemos decir que es compatible con casi cualquier microcontrolador, Arduino o tarjeta de desarrollo. El RC522 utiliza un sistema avanzado de modulación y demodulación para todo tipo de dispositivos pasivos de 13.56Mhz. Incluso, puesto que se hará una lectura y escritura de la tarjeta, es necesario conocer las características de los bloques de memoria una tarjeta:

La tarjeta que viene con el módulo RFID cuenta con 64 bloques de memoria (0-63) donde se hace lectura y/o escritura. Cada bloque de memoria tiene la capacidad de almacenar sobre todo hasta 16 Bytes. Finalmente el número de serie consiste de 5 valores hexadecimales, se podría utilizar esto para hacer una operación dependiendo del número de serie.

Características:

- Modelo: MF522-ED.
- Corriente de operación: 13-26mA a 3.3V.
- Isb de stand by: 10-13mA a 3.3V.
- Ism de sleep-mode: <80uA.
- Im máxima: 30mA.
- Frecuencia de operación: 13.56Mhz.
- Distancia de lectura: 0 a 60mm.
- Protocolo de comunicación: SPI.
- Velocidad de datos máxima: 10Mbit/s.
- Dimensiones: 40 x 60 mm.
- Temperatura de operación: -20 a 80°C.
- Humedad de operación: 5%-95%.
- Máxima velocidad de SPI: 10Mbit/s.
- Incluye pines, llavero y tarjeta.

Modulo Reloj Tiempo Real

Un módulo RTC (Real Time Clock) o "Reloj de tiempo real" consiste en un circuito integrado alimentado por una batería el cual, en todo momento, registra la fecha, día de la semana y hora al igual que un reloj digital convencional.

El módulo RTC usado para este proyecto es el DS1302.



El chip DS1302 contiene un reloj/calendario de tiempo real y 31 bytes de RAM estática. Se comunica con un microprocesador mediante una interfaz serial simple.

Solamente se requieren 3 alambres para comunicarse con el reloj y la RAM: CE (chip enable), I/O (data line) y SCLK (serial clock). Los datos pueden ser transferidos desde y hacia el reloj/RAM 1byte a la vez o 31 bytes de una sola vez. El DS1302 está diseñado para operar con muy baja potencia y retiene los datos y la información del reloj con menos de 1 µW.

Para su funcionamiento se necesita un cristal de 32.768 kHz con capacitancia de carga de 6 pF (vendido por separado). Recuerde que hablamos del Chip, el modulo si tiene el cristal.

Características:

- Cuenta segundos, minutos, horas, mes, día de la semana y año
- Año bisiesto válido hasta el 2100
- Soporta modo de 24 o 12 horas con indicador AM/PM
- RAM estática de 31 bytes
- Empaque: 8-pin DIP
- Especificaciones Clave:
- Requerimiento de potencia: 2 a 5 VDC; menos de ~300 nA
- Comunicación: TTL compatible serial sincrónico de 3 líneas
- Temperatura de Operación: -32° a +185° F (-0° a +70° C)
- Dimensiones: 0.4 x .025 in (10 x 6.4 mm)

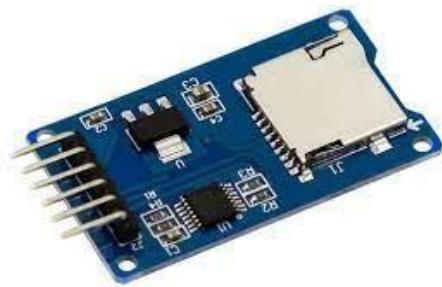
Modulo Lector SD

Un lector SD es un dispositivo que permite emplear como almacenamiento una tarjeta SD.

La lectura puede realizarse a través del bus SPI. Aunque pueden disponer de otros interfaces, como bus I2C o UART, normalmente es preferible emplear SPI por su alta tasa de transferencia. Respecto a las tarjetas empleadas, podemos emplear tarjetas SD o SDSC (Standard Capacity) o SDHC (High Capacity), pero no SDXC (Extended Capacity).

Deberá estar formateada en sistema de archivos FAT16 o FAT32. La tensión de alimentación es de 3.3V, pero en la mayoría de los módulos se incorpora la electrónica necesaria para conectarlo de forma sencilla a Arduino, lo que frecuentemente incluye un regulador de voltaje que permite alimentar directamente a 5V

- El modelo utilizado es el MicroSD Card Adapter Module



Características:

- Voltaje de Operación: 4.5V ~ 5.5V
- Voltaje en la Interfaz SPI: 3.3V ~ 5V
- Corriente de Operación: 200 uA ~ 200 mA
- Soporta Memoria uSD clásica de 2 GB
- Soporta Memoria uSD de hasta 32 GB si la tarjeta es de alta velocidad - Micro SDHC
- Tipo de Comunicación: SPI
- Dimensiones: 42 mm x 24 mm



Bibliografía

Cloruro férrico en la industria: Usos y aplicaciones:
<https://www.vadequimica.com/blog/todos-los-articulos/cloruro-ferrico.html>

Cloruro férrico o cloruro de hierro : características, usos y medidas de seguridad:
<https://www.amoquimicos.com/propiedades-del-cloruro-ferrico>

Cintas magnéticas:

<https://www.umadivulga.uma.es/museo-virtual/informatica/cintas-magneticas/>

-<https://www.umes/docencia/barzana/II/Ii04.html>

Libreria MFRC522 :

<https://reference.arduino.cc/reference/en/libraries/mfrc522/>

<https://github.com/miguelbalboa/rfid> - Repositorio Oficial

Tecnología RFID:

https://driv.in/blog/tecnologia-rfid-cuales-son-sus-ventajas?utm_source=Google-Search&utm_medium=CPC&utm_campaign=S-Blog&utm_term=&utm_campaign=S-BLOG&utm_source=adwords&utm_medium=ppc&hsa_acc=8759955348&hsa_cam=15800357032&hsa_grp=136297261193&hsa_ad=609767959863&hsa_src=g&hsa_tgt=dsa-2031047569864&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad=1&gclid=Cj0KCQjw1rqkBhCTARIzAAHz7K1pPgAavusWUXte1WddTu8ok-BKuYJEJHMQfyOwGzsy3j3ZWt_aOkIaAjQGEALw_wcB

[Funcionamiento básico de la RFID \(sistemas-rfid.es\)](https://sistemas-rfid.es/)

Módulo Lector RFID 13.56MHz RC522:

[https://naylampmechatronics.com/rfid-nfc/80-modulo-lector-rfid-1356mhz-rc522.html.](https://naylampmechatronics.com/rfid-nfc/80-modulo-lector-rfid-1356mhz-rc522.html)

Etiquetas RFID: qué son y qué aplicaciones tienen:

<https://www.dipolerfid.es/blog-rfid/etiquetas-rfid-y-aplicaciones>

[Todo lo que necesita saber sobre las tarjetas RFID - WXR \(rfidfuture.com\)](https://www.rfidfuture.com/Tutorial/RFID/RFID-Card-Technology/RFID-Card-Technology-1/RFID-Card-Technology-1.aspx)

SD, conexiones, librerías y uso:

<https://reference.arduino.cc/reference/en/libraries/sd/>

https://naylampmechatronics.com/blog/38_tutorial-arduino-y-memoria-sd-y-micro-sd.html

PUENTE H - Funcionamiento y Explicación:

<https://sensoricx.com/circuitos-para-armar/puente-h-funcionamiento-explicacion-detallada/>



Librería Liquidmenu :

<https://reference.arduino.cc/reference/en/libraries/liquidmenu/>

<https://www.youtube.com/watch?v=lxlumFsKpg0> - Video explicativo

<https://github.com/VaSe7u/LiquidMenu> - Repositorio Oficial

Funcionamiento y aplicación de encoder Rotativo:

<https://www.youtube.com/watch?v=fnDHQ6YcxTs> - Video explicativo

Transistor BC547 Datasheet :

<https://pdf1.alldatasheet.com/datasheet-pdf/view/11551/ONSEMI/BC547.html>

Transistor BC337 Datasheet :

<https://pdf1.alldatasheet.com/datasheet-pdf/view/643709/FAIRCHILD/BC337.html>

