



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Apizaco

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO

“LÁPIZ FÁCIL”

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

**PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERO EN MECATRÓNICA**

**PRESENTA:
PABLO MALDONADO XELHUANTZI**

**NOMBRE DEL ASESOR:
MTRO. GADIRO CANO LIMA**



APIZACO, TLAX., ABRIL 2024



Apizaco, Tlaxcala, 0/abr/2024.

Oficio No. NM/282/2024.

ASUNTO: Liberación de Proyectos para Titulación Integral.

MARTÍN ROJAS RAMÍREZ
JEFE DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E

Por este medio informo que ha sido liberado el siguiente proyecto para Titulación integral:

Alumno (a):	PABLO MALDONADO XELHUANTZI
Carrera:	INGENIERÍA MECATRÓNICA
No. De Control:	16370866
Nombre del Proyecto:	"LÁPIZ FÁCIL"
Producto:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestros egresados.

ATENTAMENTE

*Excepciones en Educación Tecnológica
Pensar para Servir, Servir para Triunfar*



ENRIQUE ACOLTZI BAUTISTA
JEFE DEL DEPTO. DE METAL-MECÁNICA
SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLOGICO NACIONAL
DE MÉXICO
TECNOLOGICO DE APIZACO
DEPARTAMENTO DE
METAL MECÁNICA

GADIRO CANO LIMA
ASESOR

c.p. Expediente
Archivo.

EAB*petry.





Apizaco, Tlax., 10/abr/2024
No. OFICIO: DEP-CT/368/2024

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN

PABLO MALDONADO XELHUANTZI
EGRESADO DE LA CARRERA DE INGENIERÍA MECATRÓNICA
CON NÚMERO DE CONTROL: 16370866
P R E S E N T E

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** cuyo tema es: **"LÁPIZ FÁCIL"**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar **3** empastados del mismo, apropiadamente encuadernados, a la brevedad, para presentar su acto de recepción profesional.

Sin otro particular por el momento, le envió un cordial saludo.

ATENTAMENTE

*Excelencia en Educación Tecnológica»
Pensar para Servir, Servir para Triunfar»*


MARTÍN ROJAS RAMÍREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE
ESTUDIOS PROFESIONALES

c.p. archivo



AGRADECIMIENTOS

Por medio de este espacio agradezco a todas las personas que me brindaron el apoyo incondicional durante todo este proceso de mi experiencia universitaria de arduo esfuerzo en el Instituto Tecnológico de Apizaco.

En primer lugar, les agradezco a mis padres que siempre me han brindado su apoyo incondicional para poder cumplir todos mis objetivos personales y académicos. Ellos son los que con su cariño me han impulsado siempre a perseguir mis metas y nunca abandonarlas frente a las adversidades. También son los que me han brindado el soporte material y económico para poder concentrarme en los estudios y nunca abandonarlos. Agradezco profundamente a mis hermanos por el gran cariño y apoyo durante mi formación profesional. Le agradezco infinitamente el cariño, amor y apoyo de mi esposa por brindarme su paciencia a lo largo de este proyecto. Le agradezco muy profundamente a mi tutor por su dedicación y paciencia. Sin sus palabras y correcciones precisas, no hubiese podido lograr llegar a esta instancia tan anhelada. Gracias por su guía y todos sus consejos, los llevaré grabados para siempre en la memoria en mi futuro profesional. Doy gracias profundamente a todos mis profesores por el gran apoyo y paciencia, por el haberme compartido sus grandes conocimientos y consejos, para seguir delante y poder conseguir esta meta profesional.

ÍNDICE

INTRODUCCIÓN	7
PROBLEMAS A RESOLVER	8
OBJETIVO	9
OBJETIVO GENERAL	9
OBJETIVOS ESPECIFICOS	9
JUSTIFICACION	10
DELIMITACION	11
MARCO TEORICO	12
SISTEMA NFC	12
ANTECEDENTES	12
HISTORIA	12
DEFINICION	13
TIPOS DE SISTEMAS RFID	13
Componentes de un sistema RFID	14
Etiquetas RFID	15
Lectores RFid	17
Tipos de antena	18
NFC	20
Modos y Roles de comunicación	21
Obtención de información Etiquetas NFC	22
Principio de funcionamiento	24
DESARROLLO DE CODIGO DE PROGRAMACIÓN	26
Descripción	26
ESTRUCTURA DE PROGRAMACION	27
ELEMENTOS DE LA PROGRAMACION	27
FUNCION	28
LLAVES {}	29
PUNTO Y COMA (;)	29
BLOQUE DE COMENTARIO	29
VARIABLE	30
DECLARACION DE VARIABLES	31
UTILIZACION DE VARIABLES	31

OPERADORES LOGICOS	32
CONDICIONAL IF	33
CONDICIONAL IF ELSE.....	34
SENTENCIA FOR	35
SENTENCIA WHILE	35
DIAGRAMAS	37
DIAGRAMA DE FLUJO DEL SISTEMA.....	37
BLOQUES DEL SISTEMA	38
RECURSOS UTILIZADOS EN ARDUINO	40
Modulo RFID-RC522 Mifare	40
Características Tarjeta RFID.....	43
Programa generador de código de tarjeta.....	44
Programa de Control y Monitoreo de tarjeta	45
PROGRAMA PLX-DAQ (USO OPCIONAL).....	48
CIRCUITO FINAL.....	49
INSTALACIÓN DEL SISTEMA.....	50
CRITERIO DE IMPLEMENTACION	50
COTIZACION DE PROYECTO.....	51
ESTUDIO ECONOMICO DEL PROYECTO	51
COSTO DE HARDWARE	51
COSTOS DE SOFTWARE	53
Evidencias de desarrollo.....	55
Código para la tarjeta ESP8266	58
CONCLUSIONES, RECOMENDACIONES Y EXPERIENCIA PROFESIONAL ADQUIRIDA	61
Competencias desarrolladas y/o aplicadas.....	62
BIBLIOGRAFIA.....	63

ÍNDICE DE TABLAS

TABLA 1 Tipos de etiqueta NFC.....	23
TABLA 2 Correlación pines Arduino – Lector RFID RC522.	42
TABLA 3 Características etiquetas Mifare.	43
TABLA 4. Costo total de instalación.....	55

INTRODUCCIÓN

Communication (NFC) o comunicación de campo cercano es una tecnología de comunicación inalámbrica, de corto alcance, que permite el intercambio de datos entre dispositivos. Los estándares de NFC cubren protocolos de comunicación y formatos de intercambio de datos. Es una plataforma abierta pensada desde el inicio para teléfonos y dispositivos móviles. Su mayor virtud está en la **velocidad de comunicación**, que es casi instantánea, sin necesidad de emparejamiento previo. Como contrapartida, su alcance es **muy limitado**, con un máximo de **20 cm** de distancia.

Las características que definen los casos para los cuales se está utilizando esta tecnología son de mayor impacto ya que no es complicada de utilizar. Los chips NFC son sumamente livianos. Esto permite que puedan ser usados en llaveros o etiquetas. Su costo es muy bajo. Las **etiquetas NFC** son etiquetas (pueden ser blancas o de colores), que contienen un chip NFC. El chip es de lectura y escritura desde un teléfono celular. **No requieren de baterías**, obtienen energía en la aproximación a un dispositivo NFC cargado (el teléfono). Para dicho proyecto se garantiza la satisfacción del director general por tener una mejora del control de asistencia de una manera única e irrepetible es el identificar las etiquetas (pegatinas), tarjetas NFC, que se le asigna a cada persona una manera sencilla y rápida de saber a quién pertenece una tarjeta NFC es de la siguiente manera asignar un número externo que sea visible para el usuario, para que el módulo ESP8266 identifique que es la persona correcta tiene por defecto un número grabado internamente. Este número le asignamos a una persona y de esta manera se garantiza que no se perderá su registro y su control de acceso.

PROBLEMAS A RESOLVER

La problemática que se presenta en el área de producción y salidas de producto de la empresa KUBIY SOFT SAS de C.V. Monterrey N.L., era un gran número de quejas del cliente por no tener la producción en tiempo y forma, ya que esto se debía al equipo incompleto de personas que conforman una área y cada operador tiene asignado, una máquina para operar y al faltar o llegar tarde una persona existían retardos de producción (afectaba toda una línea de producción o salida de producto), dicho problema estaba generando multas y sanciones por incumplimiento de entrega. Al recibir un número considerable de quejas de los clientes, el índice de satisfacción al cliente no era el esperado por la empresa y también había sanciones económicas hacía el personal, por esta razón la tecnología se implementa a raíz de la demora que existe con el proceso actual de control de asistencia la cual al ser manual. Implicaba la posible lectura incorrecta de la hora de entrada y salida del personal. Por lo tanto, se requiere de un sistema automático que realice la identificación de entrada del personal identificando hora exacta de entrada y salida.

Con esta implementación disminuyo las quejas y se eliminó una gran cantidad de problemas para todos así lo demostró desarrollo organizacional.

OBJETIVOS

OBJETIVO GENERAL

Implementar un control para la asistencia de las personas en el área de producción de la empresa KUBIY SOFT SAS de CV. con el propósito de garantizar un buen servicio a los clientes, satisfacción y así conservar su confianza y permanencia.

OBJETIVOS ESPECIFICOS

Diseñar e implementar un sistema para Arduino que permita realizar un control de asistencia de las personas que ingresan a sus áreas de trabajo.

Utilizar una aplicación que permita almacenar todos los datos recolectados y generados por el sistema de control de asistencia personal.

Identificar cada una de las tarjetas o pulseras utilizadas por el usuario por medio de colores y números externos de manera impresa, para tener un mayor control sobre la posible pérdida, o que alguien más ingrese con la tarjeta que no pertenece a un usuario que la haya olvidado.

JUSTIFICACION

El proyecto tiene como fin implementar un control del personal, actualmente hay áreas de oportunidad respecto a la calidad del servicio, ya que han surgido quejas por parte de los clientes ya sea que no se ha entregado el producto correctamente que tenga los estándares de calidad, etc. Con el proyecto a desarrollar se pretende aumentar la productividad, reducir los defectos en la empresa y así lograr satisfacer las necesidades y expectativas del cliente.

El horario de ingreso y salida de personal dentro de la empresa debe ser controlada, lo cual provoca la constante aglomeración de personal a la hora de registrar su ingreso o salida lo que genera retrasos y malestar en el personal, sin tomar en cuenta el crecimiento de personal lo cual generará aún más demoras en el proceso de marcación de asistencia

Es por ello que el proyecto se justifica, teniendo como argumento la necesidad urgente de mejorar el sistema actual de control de asistencia para el personal, para el cual se propone el ~~de~~ diseño e instalación de un sistema de control de asistencia automática mediante comunicación RFID-NFC. Con la implementación del sistema de control de asistencia automática se simplifican los procesos de control y revisión de asistencia, al generar un archivo digital se tiene una más fácil verificación de la asistencia del personal.

Resumen

El presente proyecto de aplicación consiste en el diseño e implementación de un sistema de control de asistencia personal de estudiantes para la empresa KUBI Y SOFT S.A.S DE C.V. mediante el uso de sistema RFID-NFC. El sistema de control se diseñará e implementará en una base de datos utilizando ARDUINO en su versión Arduino UNO R3, con el uso de sensor RFID modelo RC522 – NFC, así creando una interfaz para la grabación de datos. Se genera una comunicación serial para interconectar los dispositivos esta información es enviada al programa XAMPP de manera automática la base de datos es creada por KUBI Y SOFT, ya que dicha y prestigiada empresa ha diseñado su propia base de datos la cual he recibido una dirección HOST(URL) esta dirección de internet nos enviará los datos a una cuenta creada por la empresa donde será consumida está dirección y mostrar los datos de las personas que ya han sido registradas.

Por otra parte, investigue más a fondo de direcciones IP, URL, la sentencia JONSON, cómo se utiliza cada sentencia, para que funcionen, determinar en qué momento serán utilizadas. Siendo las direcciones de un servidor y de esta manera se mandará toda la información de manera automática que solicita la empresa KUBIY SOFT.

DELIMITACION

La implementación del sistema de control de asistencia será diseñada de manera tal que pueda controlar únicamente TAG's de radiofrecuencia y equipos que cuenten con sensores NFC que se encuentren registrados en el sistema de control.

MARCO TEORICO

SISTEMA NFC

RFID (siglas de Radio Frequency IDentification, en español identificación por radiofrecuencia) es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio. Las tecnologías RFID se agrupan dentro de las denominadas Auto ID (automatic identification, o identificación automática).

Las etiquetas RFID (RFID Tag, en inglés) son unos dispositivos pequeños, similares a una pegatina, que pueden ser adheridas o incorporadas a un producto, un animal o una persona. Contienen antenas para permitirles recibir y responder a peticiones por radiofrecuencia desde un emisor-receptor RFID. Las etiquetas pasivas no necesitan alimentación eléctrica interna, mientras que las activas sí lo requieren. Una de las ventajas del uso de radiofrecuencia (en lugar, por ejemplo, de infrarrojos) es que no se requiere visión directa entre emisor y receptor.

En otras palabras, se trata de una tecnología de transmisión de datos automática que emplea radiofrecuencia para comunicar información entre un lector y una etiqueta electrónica (e-tag); o lo que es lo mismo, utiliza ondas electromagnéticas para enviar la información desde la etiqueta al lector. Este sistema permite la captura y/o grabación de datos sin contacto entre lector y etiqueta, eliminando así la necesidad de un contacto visual directo.

(parzibyte.me, 2020)

ANTECEDENTES

En la actualidad, la tecnología más extendida para la identificación de objetos es la de los códigos de barras. Sin embargo, éstos presentan algunas desventajas, como la escasa cantidad de datos que pueden almacenar y la imposibilidad de ser reprogramados. La idea mejorada constituyó el origen de la tecnología RFID; consistía en usar chips de silicio que pudieran transferir los datos que almacenaban al lector sin contacto físico, de forma equivalente a los lectores de infrarrojos utilizados para leer los códigos de barras.

HISTORIA

Se ha sugerido que el primer dispositivo conocido similar a RFID pudo haber sido una herramienta de espionaje inventada por Léon Theremin para el gobierno soviético en 1945. El dispositivo de Theremin era un dispositivo de escucha secreto

pasivo, no una etiqueta de identificación, por lo que esta aplicación es dudosa. Según algunas fuentes, la tecnología usada en RFID habría existido desde comienzos de los años 1920, desarrollada por el MIT y usada extensivamente por los británicos en la Segunda Guerra Mundial (fuente que establece que los sistemas RFID han existido desde finales de los años 1960 y que sólo recientemente se había popularizado gracias a las reducciones de costos). Una tecnología similar, el transpondedor de IFF, fue inventada por los británicos en 1939, y fue utilizada de forma rutinaria por los aliados en la Segunda Guerra Mundial para identificar los aeroplanos como amigos o enemigos. Se trata probablemente de la tecnología citada por la fuente anterior. Otro trabajo temprano que trata el RFID es el artículo de 1948 de Harry Stockman, titulado "Comunicación por medio de la energía reflejada" (Actas del IRE, pp. 1196-1204, octubre de 1948). Stockman predijo que el trabajo considerable de investigación y de desarrollo tiene que ser realizado antes de que los problemas básicos restantes en la comunicación de la energía reflejada se solucionen, y antes de que el campo de aplicaciones útiles se explore." Hicieron falta treinta años de avances en multitud de campos diversos antes de que RFID se convirtiera en una realidad.

DEFINICIÓN

Un sistema RFID básicamente se compone de una etiqueta RFID que contiene una información y un lector RFID (también puede ser grabador) comunicado con un sistema informático que procesa la información recibida de la etiqueta o tag. En el proceso de lectura, el lector realiza una petición a la etiqueta solicitando que ésta le envíe la información que tiene almacenada. La etiqueta responde a esa solicitud enviando por radiofrecuencia dicha información. El lector recoge esa información y la interpreta. Evidentemente, esta información será útil si detrás del sistema RFID

existe un ordenador o un dispositivo similar capaz de procesar esta información para realizar una acción concreta. El proceso de escritura es muy parecido al de lectura, con la particularidad de que es necesario que el tag se pueda reescribir. La estación grabadora enviará al tag por radiofrecuencia los datos a grabar junto con la instrucción u orden de grabación, y el tag los grabará en el chip que tiene incorporado.

(cucorent, 2018)

TIPOS DE SISTEMAS RFID

Según la frecuencia de trabajo que se utilice en un sistema RFID tendremos distintos tipos de sistemas, lo que a su vez dará lugar a significativas diferencias funcionales. La frecuencia de trabajo es, de este modo, un factor crítico que determina en gran medida el comportamiento de un sistema RFID en cuanto a su manera de funcionar, sus capacidades y sus aplicaciones.

Así, tenemos los siguientes tipos de sistemas RFID:

- Sistemas RFID de baja frecuencia, conocidos como sistemas RFID LF (Low Frequency). Su frecuencia de funcionamiento, en Europa, es 125 KHz.
- Sistemas RFID de alta frecuencia, conocidos como sistemas RFID HF (High Frequency). Su frecuencia de funcionamiento es 13,56 MHz.
- Sistemas RFID UHF, son aquellos sistemas que trabajan en frecuencias de la banda UHF (Ultra High Frequency). Los sistemas RFID UHF trabajan, en España, en 869 MHz.
- Sistemas RFID de microondas, son aquellos sistemas RFID que trabajan a frecuencias mayores de los 2,4 GHz. Estos sistemas actualmente son muy poco utilizados.

Como hemos mencionado, la frecuencia a la que trabaje el sistema RFID influye directamente en su funcionamiento. Estas diferencias funcionales inciden tanto en las características de rango y ratio de lectura (distancia y velocidad), como en el diseño de las antenas (tanto las de los tags como las de estaciones lectoras/grabadoras), así como en la distinta manera en la que el tag y el lector intercambian los datos. Además, estas diferencias funcionales harán que, dependiendo de la frecuencia de trabajo, el sistema sea o no apropiado para una determinada aplicación.

Así, las frecuencias más bajas disponen de un menor rango y ratio de lectura que las frecuencias más altas; en otras palabras, la identificación entre dispositivos es más lenta y se limita a distancias más cortas. Sin embargo, son muy apropiadas para trabajar en entornos húmedos, por lo que son frecuencias idóneas para aplicaciones en las que el grado de humedad del entorno supone un problema (así, por ejemplo, los sistemas RFID LF son los sistemas habituales para la identificación de animales o envases líquidos).

Componentes de un sistema RFID

Un sistema RFID básico consta de dos elementos fundamentales:

- Una etiqueta, tag o transpondedor RFID. Es el elemento en el que se incorpora el chip electrónico que contiene la información necesaria para identificar al producto. Contiene también una micro antena que le permite recibir y transmitir la información a través de radiofrecuencia.
- Un lector RFID. Se trata de un equipo capaz de recibir y procesar la información procedente de los tags RFID. Este equipo cuenta con una antena que le permite recibir y transmitir peticiones de información a través de radiofrecuencia.

Además de estos dos elementos imprescindibles, un sistema RFID puede estar formado por varios componentes más, como estaciones de programación de etiquetas (que podría ser la misma que la de lectura, ya que es habitual encontrar en el mercado estaciones de lectura/escritura), lectores de circulación, bases de datos, etc.

Etiquetas RFID

Básicamente, una etiqueta RFID está compuesta por un chip unido a una antena. Dependiendo de las necesidades de la aplicación puede presentar diferentes formatos (rígido, flexible, adhesivo, etc.), diferentes tamaños, funcionalidades estándares y por lo tanto, diferentes precios. En cuanto a las dimensiones de las etiquetas, existe una amplia gama de ofertas y posibilidades en el mercado que van desde unos pocos milímetros hasta decenas de centímetros, dependiendo de la aplicación a la que se destinen, estas etiquetas RFID pueden ser de tres tipos distintos dependiendo del lugar del que provenga la energía que utilizan para tramitar la respuesta. Así, las etiquetas RFID pueden ser pasivas, si no tienen fuente de alimentación propia, semipasivas, si utilizan una pequeña batería asociada, y activas, si tienen su propia fuente de alimentación.

Las etiquetas RFID pasivas no tienen fuente de alimentación propia. La corriente eléctrica necesaria para su funcionamiento se obtiene por inducción en su antena de la señal de radiofrecuencia procedente de la petición de lectura de la estación lectora. De esta manera, cuando el lector interroga a la etiqueta, genera un campo magnético que produce en la microantena del tag un campo eléctrico suficiente que permite generar la energía necesaria para que el circuito CMOS7, integrado en la etiqueta, pueda transmitir una respuesta. Debido a esto, la señal de respuesta tiene un tiempo de vida bastante corto; además, en la práctica, las distancias de lectura son relativamente reducidas (varían entre unos pocos milímetros hasta distancias cercanas a 1 metro).

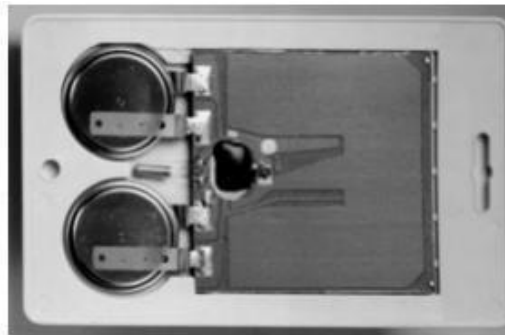
Figura 1. Tarjeta RFID pasiva. Recuperado 13/10/2021 de www.sistemanfc.com



Las etiquetas RFID semipasivas son muy similares a las pasivas, salvo que incorporan además una pequeña batería. Esta batería permite al circuito integrado en la etiqueta estar constantemente alimentado, y elimina la necesidad de diseñar una antena para recoger potencia de una señal entrante (si bien la antena es necesaria para la transmisión/ recepción de las señales). Estas etiquetas responden más rápidamente, por lo que son más fuertes en la ratio de lectura en comparación con las etiquetas pasivas. Asimismo, también disponen de un rango de lectura (distancia) ligeramente superior al de las etiquetas pasivas.

Las etiquetas RFID activas poseen una fuente de alimentación propia y alcanzan rangos de lectura mucho mayores que las etiquetas pasivas y semipasivas. Así, este tipo de etiquetas pueden ser leídas a distancias superiores a los 100 metros. Su tamaño es lógicamente mayor que el de los otros dos tipos, aunque puede ser inferior el tamaño de una moneda. Estas etiquetas activas pueden incorporar también una pequeña memoria adicional que les permite almacenar mayor cantidad de información. La gran limitación que presenta este tipo de etiquetas es que su funcionamiento está ligado a su batería, por lo que su tiempo de vida se limita a la vida de la batería.

Figura 2. Tarjeta RFID activa. Recuperado de: Torrez A. Control de Inventario



Las etiquetas pasivas son mucho más baratas de fabricar, por lo que la gran mayoría de las etiquetas RFID existentes son de ese tipo. Sin embargo, a pesar de la ventaja del costo, las diferencias funcionales de los distintos tipos de etiquetas (rango de lectura, ratio de lectura y capacidad de almacenamiento), hacen que determinadas aplicaciones necesiten usar etiquetas activas o, al menos, semipasivas.

Además de este criterio del origen de su alimentación, las etiquetas RFID también se pueden clasificar según la configuración de su memoria, y así hay etiquetas que únicamente se pueden leer (no se puede escribir en ellas, vienen escritas “de fábrica”), etiquetas que se pueden leer un número ilimitado de veces pero que admiten un único proceso de grabación o escritura, y etiquetas que se pueden leer y escribir un número ilimitado de veces.

Todas las etiquetas almacenan en el microchip un número de identificación único que permite identificar y diferenciar al producto al que acompañan no sólo de otros productos distintos, sino también de otros ítems iguales.

Lectores RFid

Los lectores de RFID o “interrogadores” utilizan ondas de radio para leer la información almacenada en la etiqueta. El lector puede enviar a la etiqueta la orden de transmitir la información que tiene almacenada, o bien la etiqueta puede transmitir la información que contiene periódicamente, en espera de que algún lector la detecte. En definitiva, el lector es el dispositivo que interactúa con el tag o etiqueta. Los lectores de radiofrecuencia constan de:

- Una antena.
- Un módulo de radiofrecuencia o controlador.
- Una unidad de control.

Las antenas del lector son el componente más sensible de un sistema RFID y pueden variar en función de la aplicación concreta. Así, las antenas pueden ser:

- De sobremesa: se incorporan a los puntos de identificación y control o terminales punto de servicio, para la lectura y escritura de las etiquetas.
- De puerta: se disponen en pedestales en el suelo e incluyen elementos de alarma.

Variar la ubicación de la antena del lector es una de las formas más fáciles de ajuste cuando se localizan problemas de un sistema, y al mismo tiempo resulta una de las tareas más difíciles de llevar a cabo de forma correcta.

La antena del lector debe ser colocada en una posición que permita optimizar tanto la transmisión de energía hacia la etiqueta como la recepción de los datos emitidos. Las limitaciones legislativas del nivel de potencia de los lectores hacen que la ubicación de las antenas sea fundamental para alcanzar un alto grado de lectura. El funcionamiento básico del lector suele ser muy sencillo.

La antena del lector crea un campo magnético cuyo radio de acción varía dependiendo de la forma de la antena, de la potencia del lector y de la frecuencia en la que emita. En el caso de etiquetas pasivas, el lector genera un campo de radiofrecuencia que activa la etiqueta alimentando el chip para que pueda transmitir

los datos almacenados. En el caso de etiquetas activas la propia etiqueta puede iniciar la comunicación, pues no necesita la energía del lector.

Cuando la etiqueta entra en contacto con el campo magnético creado por el lector, reacciona automáticamente y envía la información a éste. El lector la decodifica y la pone en contacto con el sistema de control, a través de una aplicación que se ejecuta en su unidad de control y que se conoce como ONS (Object Name Service). Algunos sistemas utilizan encriptación de clave pública para codificar la información que fluye entre la etiqueta y el lector con objeto de conseguir mayor seguridad ante posibles escuchas maliciosas y obtener cierta privacidad.

Un lector puede utilizarse también para reescribir sobre una etiqueta, siempre y cuando el lector esté habilitado para ello y la etiqueta sea “reescribible”. Naturalmente, estos lectores con doble función tienen un precio superior al de aquellos que sólo leen.

Los lectores pueden ser portátiles o fijos, y también pueden estar integrados en otros equipos como terminales handheld (PDAs, Lectores de mano, etc.), grúas horquillas, o cualquier equipo que se considere estratégico.

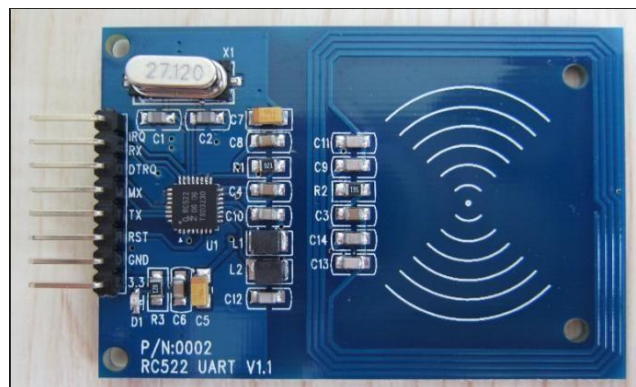


Figura 3. Lector RFID FRC522. Recuperado de www.nfcRC522.com

Tipos de antena

El tipo de antena utilizado en una etiqueta depende de la aplicación para la que está diseñado y de la frecuencia de operación. Las etiquetas de baja frecuencia (LF, del inglés low frequency) normalmente se sirven de la inducción electromagnética. Como el voltaje inducido es proporcional a la frecuencia, se puede producir el necesario para alimentar un circuito integrado utilizando un número suficiente de espiras. Existen etiquetas LF compactas (como las encapsuladas en vidrio, utilizadas para identificación humana y animal) que utilizan una antena en varios niveles (tres de 100-150 espiras cada uno) alrededor de un núcleo de ferrita.

En alta frecuencia (HF, 13.56 MHz) se utiliza una espiral plana con 5-7 vueltas y un factor de forma parecido al de una tarjeta de crédito para lograr distancias de decenas de centímetros. Estas antenas son más baratas que las LF ya que pueden producirse por medio de litografía en lugar de espiración, aunque son necesarias dos superficies de metal y una aislante para realizar la conexión cruzada del nivel exterior al interior de la espiral, donde se encuentran el condensador de resonancia y el circuito integrado.

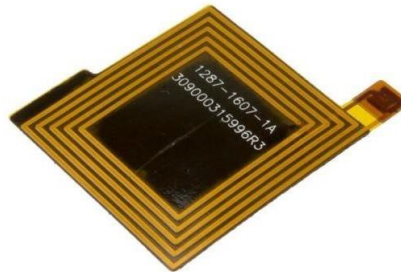


Figura 4. Forma de antena RFID de Smart Phone. Recuperado de www.nfc.com

Las etiquetas pasivas en frecuencias ultra alta (UHF) y de microondas suelen acoplarse por radio a la antena del lector y utilizar antenas clásicas de dipolo. Sólo es necesaria una capa de metal, lo que reduce el coste. Las antenas de dipolo, no obstante, no se ajustan muy bien a las características de los circuitos integrados típicos (con alta impedancia de entrada, ligeramente capacitiva). Se pueden utilizar dipolos plegados o bucles cortos como estructuras inductivas complementarias para mejorar la alimentación. Los dipolos de media onda (16 cm a 900 MHz) son demasiado grandes para la mayoría de aplicaciones (por ejemplo, las etiquetas RFID para uso en etiquetas no pueden medir más de 10 cm), por lo que hay que doblar las antenas para satisfacer las necesidades de tamaño. También pueden usarse estructuras de banda ancha. La ganancia de las antenas compactas suele ser menor que la de un dipolo (menos de 2 dB) y pueden considerarse isótropas en el plano perpendicular a su eje.

Los dipolos experimentan acoplamiento con la radiación que se polariza en sus ejes, por lo que la visibilidad de una etiqueta con una antena de dipolo simple depende de su orientación. Las etiquetas con dos antenas ortogonales (etiquetas de doble dipolo) dependen mucho menos de ella y de la polarización de la antena del lector, pero suelen ser más grandes y caras que sus contrapartidas simples.

Pueden usarse antenas de parche (patch) para dar servicio en las cercanías de superficies metálicas, aunque es necesario un grosor de 3 a 6 mm para lograr un buen ancho de banda, además de que es necesario tener una conexión a tierra que incrementa el coste comparado con estructuras de una capa más sencillas.

Las antenas HF y UHF suelen ser de cobre o aluminio. Se han probado tintas conductoras en algunas antenas encontrando problemas con la adhesión al circuito integrado y la estabilidad del entorno



Figura 5. Forma antena RFID. Recuperado de <https://store.mectronica.it/es/etiqueta-rfid-nfc-e/501-tag-nfc-adhesivo-circular-45-mm-polycarbonato-externo-interno-impermeable.html>

NFC

Near field communication (NFC, comunicación de campo cercano en español) es una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos. Los estándares de NFC cubren protocolos de comunicación y formatos de intercambio de datos, y están basados en ISO 14443 (RFID, radio-frequency identification) y FeliCa.¹ Los estándares incluyen ISO/IEC 18092² y los definidos por el NFC Forum, fundado en 2004 por Nokia, Philips y Sony, y que hoy suma más de 170 miembros.

Las etiquetas NFC son pequeños dispositivos NFC pasivos que pueden contener fragmentos de información. Básicamente se trata de pequeñas espirales de metal a las que se les añaden

Componentes de memoria y comunicación. Su diseño completamente plano las hace ideales para presentarse en formatos como pegatinas, tarjetas de visita, llaveros e incluso pulseras. Su funcionamiento es parecido al de los códigos de barras y de los códigos QR, sin la necesidad del reconocimiento óptico del código. Así, un dispositivo NFC activo (incorporado en un teléfono inteligente, por ejemplo) actúa como lector. Al acercarlo a una etiqueta NFC, el campo de radiofrecuencia que el lector aplica sobre la etiqueta la activa y hace que le transmita los datos que almacena en su interior. Estos datos dependerán, entre otros factores, del tipo de etiqueta NFC, ya que existen varias clases en función de su memoria, su tasa de transferencia de datos y los modos de interacción. Estos tipos fueron estandarizados por el NFC Forum, además, promueven el logotipo N Mark como distintivo de la tecnología.

Figura 6. Logotipo NFC. Recuperado de www.NFC.com



En las transmisiones de dispositivos NFC se utiliza la banda de frecuencia alojada en los 13,56 Mhz, tradicionalmente asignada a las etiquetas RFID en modo pasivo (como etiquetas de identificación de productos farmacéuticos, documentos de identidad, etc.). Esta frecuencia pertenece al conjunto de bandas de radio ISM, que son utilizadas generalmente con fines industriales, científicos y médicos (de ahí su nombre ISM, Industrial, Scientific and Medical). Para el uso de estas bandas no es necesaria una licencia, pero se debe garantizar que no se produzcan interferencias entre los dispositivos. Esto representa una ventaja para la implantación y uso de NFC, ya que el canal de transmisión es libre y no tiene un coste de uso asociado. Otros sistemas que también utilizan estas bandas de frecuencia son, por ejemplo, Bluetooth y Wi-Fi.

Modos y Roles de comunicación

- **Pasivo:** Si el dispositivo no genera su propio campo, sino que recibe alimentación del campo de radiofrecuencia de otro dispositivo. Íntimamente vinculados a estos modos de actuación, se encuentran los roles que un dispositivo NFC puede tomar en una comunicación. Al basarse la transmisión en el concepto de pregunta y respuesta, un dispositivo sólo puede responder a otro si y sólo si ha iniciado una comunicación con él. Estos roles son dos, y en toda comunicación son necesarios ambos, asumiendo cada dispositivo uno de ellos:
- **Iniciador:** El dispositivo que da comienzo a la interacción. Aunque se trate de una comunicación en la que ambos dispositivos lleguen a ejercer un papel activo, el dispositivo que comience la comunicación será el considerado como iniciador hasta que se finalice la comunicación.
- **Objetivo:** El dispositivo que responde a la comunicación establecida por el iniciador.

Según estas definiciones, se puede deducir que hay determinadas combinaciones de modos y roles no compatibles. Un dispositivo activo puede actuar como iniciador o como objetivo en una comunicación, sin embargo, un dispositivo pasivo no puede ser nunca iniciador, ya que no puede generar su propio campo de radiofrecuencia ni, por tanto, emitir una señal a otro dispositivo por sí solo.

Tradicionalmente, una de las configuraciones más usadas es la que comunica un dispositivo activo que actúa como iniciador y uno pasivo que es el objetivo. En este escenario, el elemento activo emite un campo de radiofrecuencia, que alimentará electromagnéticamente al dispositivo pasivo haciendo que este responda

modulando la señal con cierta codificación. Esta configuración es heredada de RFID y la base para construir las etiquetas NFC.

Con la especificación NFCIP-1 se introduce el concepto de la comunicación peer-to-peer entre dos dispositivos activos. En este caso, un dispositivo activo “A” actúa como iniciador de la comunicación mandando un mensaje a un dispositivo “B”. Mientras espera respuesta, el dispositivo “A” desactiva su campo de radiofrecuencia, mientras que “B” lo activa para emitir. Así, durante la comunicación los dispositivos van alternándose en la activación sus campos de radiofrecuencia para enviar y recibir datos entre ellos.

En este último caso es importante no confundir los modos y roles. Aunque desactiven sus campos de radiofrecuencia, ambos dispositivos siguen siendo por definición activos. Además, aunque “B” transmita datos, el iniciador de la comunicación en el origen seguirá siendo siempre el dispositivo “A” hasta que finalice la comunicación.

A pesar de sus posibilidades, el sistema NFC tiene diferentes limitaciones. La limitación principal es que un dispositivo activo no puede comunicarse con varios dispositivos pasivos a la vez, aunque estos sí serían capaces de responder al estar bajo la influencia del campo de radiofrecuencia. Por ello, esta comunicación se limita a la elección de uno de los elementos pasivos antes de iniciar la transferencia de datos. El iniciador debe seleccionar qué dispositivo pasivo es el receptor del mensaje, siendo ignorado por el resto. Así, los mensajes de difusión obroadcast, en los que varios dispositivos reciben información de un mismo emisor, no están permitidos en esta tecnología.

Obtención de información Etiquetas NFC

Las etiquetas NFC son pequeños dispositivos NFC pasivos que pueden contener fragmentos de información. Básicamente se trata de pequeñas espirales de metal a las que se les añaden componentes de memoria y comunicación. Su diseño completamente plano las hace ideales para presentarse en formatos como pegatinas, tarjetas de visita, llaveros e incluso pulseras.

Su funcionamiento es parecido al de los códigos de barras y de los códigos QR, sin la necesidad del reconocimiento óptico del código. Así, un dispositivo NFC activo (incorporado en un teléfono inteligente, por ejemplo) actúa como lector. Al acercarlo a una etiqueta NFC, el campo de radiofrecuencia que el lector aplica sobre la etiqueta la activa y hace que le transmita los datos que almacena en su interior. Estos datos dependerán, entre otros factores, del tipo de etiqueta NFC, ya que existen varias clases en función de su memoria, su tasa de transferencia de

datos y los modos de interacción. Estos tipos fueron estandarizados por el NFC Forum y se pueden observar en la siguiente tabla:

TABLA 1. Índice de tabla 1 Tipos de etiqueta NFC. Recuperado de https://www.kimaldi.com/productos/sistemas_rfid/etiquetas_rfid/tags_etiqueta_nfc/

	Cuadrado 14x14.5mm / 0.6x0.6" 18x18 mm / 0.7x0.7"	Frecuencia 13.56 MHz	Protocolo ISO 15693
	Rectangular 14x31 mm / 0.6x1.2" 18x36mm / 0.7x1.4"	Frecuencia 13.56 MHz	Protocolo ISO 15693 ISO 18000-3 ISO 14443 A - NFC
	Cuadrado 47x47 mm / 1.9x1.9" 50x50 mm / 2.0x2.0"	Frecuencia 13.56 MHz	Protocolo ISO 15693 ISO 18000-3
	Rectangular 45x76 mm / 1.8x3.0" 49x81 mm / 1.9x3.2"	Frecuencia 13.56 MHz	Protocolo ISO 15693 ISO 18000-3 ISO 14443 A / NFC
	Disco 16 mm / 0.6" 19 mm / 0.7" 23 mm / 0.9" 25 mm / 1.0"	Frecuencia 13.56 MHz	Protocolo ISO 15693 ISO 18000-3 ISO 14443 A / NFC
	Disco 33 mm / 1.3" 40 mm / 1.6" 35 mm / 1.4" 38 mm / 1.5"	Frecuencia 13.56 MHz	Protocolo ISO 15693 ISO 18000-3 ISO 14443 A / NFC

En el interior de las etiquetas, la información se almacena en un formato especial llamado Formato de Intercambio de Datos o NDEF (NFC Data Exchange Format). Este formato indica cómo deben ser guardados y encapsulados los datos dentro de las etiquetas u otros dispositivos para asegurar la interoperabilidad, es decir, que cualquier dato NFC pueda ser leído por cualquier lector, independientemente del fabricante de los dispositivos implicados.

Un mensaje NDEF se compone de una serie de registros, los cuales contienen una cabecera y el cuerpo. Esta cabecera contiene información sobre el tipo y la longitud de los datos almacenados en el cuerpo. Posteriormente, se encuentran los datos propiamente dichos o payload.

Estos datos pueden ser de diferentes clases, por ejemplo, URIs o cualquiera de los tipos de datos específicos para NFC identificados por las Definiciones de Tipos de Registros o RTD (Record Type Definition). Los RTD son formatos optimizados para la transmisión entre dispositivos NFC.

RTD Texto: Es el tipo más simple y contiene una cadena de texto en codificación Unicode. Puede utilizarse como si de texto en plano se tratara, aunque realmente se ideó para añadir metadatos de otros registros dentro de la etiqueta.

RTD URI: Se trata de una serie de datos que identifican un recurso en concreto (estas siglas se refieren a Uniform Resource Identifier, identificador uniforme de

recursos). Los recursos a identificar pueden ser de diferentes tipos, ya que las informaciones almacenadas pueden ser teléfonos, direcciones de correo electrónico y web, etc. Así, la etiqueta que incorpore esta información permitiría remitir al usuario a un determinado recurso. Un registro URI puede utilizarse de forma aislada o formando parte de otros registros, como el Smart Poster. Además de la URI en cuestión, en el RTD se define un campo URI Identifier, que contiene el protocolo a utilizar. Están soportados hasta 35 protocolos diferentes, incluyendo HTTP, FTP, telnet, incluso Samba o Bluetooth. Esto hace que los dispositivos no necesiten implementar un protocolo para las URIs, ya que este se encuentra especificado en el propio dato.

RTD Smart Poster: Es el RTD por excelencia y el que engloba la mayor parte de los usos finales. Se define como una estructura que contiene a los RTDs anteriores además de un icono y acciones de control recomendado (hacer una acción, editar un contenido o guardarlo). Por ejemplo, puede contener información de un contacto, con nombre en un RTD texto, correo y página web en URI, foto en un tipo MIME image/jpeg o cualquier otro dato representable con estos tipos.

RTD de control genérico: Da acceso a funciones o aplicaciones que no puedan ser expresadas mediante otros RTD. Además permite enviar órdenes a otros La tecnología NFC: aplicaciones y gestión de la seguridad Página 6 de 21 Instituto Nacional de Tecnologías de la Comunicación (INTECO) La tecnología NFC: aplicaciones y gestión de la seguridad Página 7 de 21 Instituto Nacional de Tecnologías de la Comunicación (INTECO) dispositivos, pudiendo incluso seleccionar qué aplicación desea que ejecute la orden.

RTD firma: Contiene en la etiqueta una firma digital para los contenidos como método de seguridad. Soporta firmados DSA, ECDSA y PKCS#1, que además aporta cifrado, y certificados X.509 y X9.68. Se puede usar una sola firma para todo el contenido, o firmar sólo otros registros o grupos de registros, además de poder usarse diferentes firmas para diferentes registros.

Principio de funcionamiento

Para establecer una comunicación entre un lector y un tag, es decir entre un activo y un pasivo, el lector emite una forma de onda de radiofrecuencia, para cuando el tag se encuentra dentro del rango de acción del lector (es decir dentro del campo magnético generado por el lector), la micro antena de este capta la señal y activa a el microchip, el cual, a través de la micro antena y mediante ondas de radiofrecuencia, transmite al lector la información almacenada en su memoria, es decir, le responde identificándose a sí mismo.

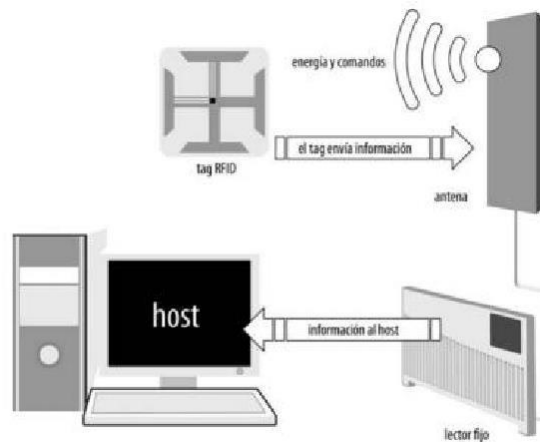


Figura 7. Comunicación entre TAG, lector y Host (PC). Recuperado de: Torrez A. Control de Inventario, 2021

Las etiquetas pueden leerse a cierta distancia dependiendo del diseño sin tener un contacto físico con el lector. El rango de acción determina la distancia dentro del cual el lector puede comunicarse con una TAG, a esta distancia también se le suele llamar rango de lectura.

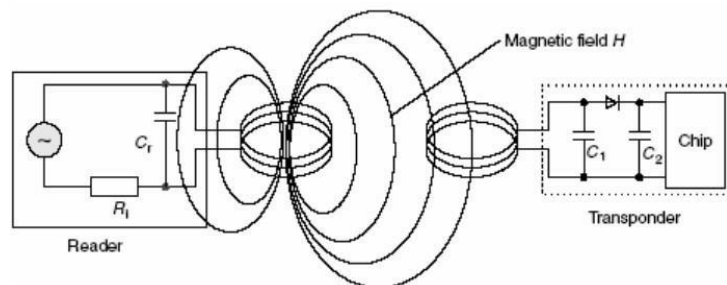


Figura 8. Esquema de acoplamiento inductivo lector transponder. Recuperado de: Torrez A. Control de Inventario, 2021

DESARROLLO DE CODIGO DE PROGRAMACIÓN

Descripción

Arduino (en EEUU, Genuino a nivel internacional) es una compañía de hardware libre y una comunidad tecnológica que diseña y manufactura placas de desarrollo de hardware y software, compuesta respectivamente por circuitos impresos que integran un microcontrolador y un entorno de desarrollo (IDE), en donde se programa cada placa. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Toda la plataforma, tanto para sus componentes de hardware como de software, son liberados con licencia de código abierto que permite libertad de acceso a ellos.

El hardware consiste en una placa de circuito impreso con un microcontrolador, usualmente Atm el AVR, puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a placas de expansión (siles), que amplían las características de funcionamiento de la placa Arduino. Asimismo, posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación con el computador.

Por otro lado, el software consiste en un entorno de desarrollo (IDE) basado en el entorno de Processing y lenguaje de programación basado en Wiring, así como en el cargador de arranque (bootloader) que es ejecutado en la placa.⁴ El microcontrolador de la placa se programa mediante un computador, usando una comunicación serial mediante un convertidor de niveles RS-232 a TTL serial.

La primera placa Arduino fue introducida en 2005, ofreciendo un bajo costo y facilidad de uso para novatos y profesionales. Buscaba desarrollar proyectos interactivos con su entorno mediante el uso de actuadores y sensores. A partir de octubre de 2012, se incorporaron nuevos modelos de placas de desarrollo que usan microcontroladores Cortex M3, ARM de 32 bits,⁵ que coexisten con los originales modelos que integran microcontroladores AVR de 8 bits. ARM y AVR no son plataformas compatibles en cuanto a su arquitectura y por lo que tampoco

es su set de instrucciones, pero se pueden programar y compilar bajo el IDE predeterminado de Arduino sin ningún cambio.

Las placas Arduino están disponibles de dos formas: ensambladas o en forma de kits "Hazlo tú mismo". Los esquemas de diseño del Hardware están disponibles bajo licencia Libre, con lo que se permite que cualquier persona pueda crear su propia placa Arduino sin necesidad de comprar una prefabricada. Adafruit Industries estimó a mediados del año 2011 que, alrededor de 300,000 placas Arduino habían sido producidas comercialmente y en el año 2013 estimó que alrededor de 700.000 placas oficiales de la empresa Arduino estaban en manos de los usuarios.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data, etc. Una tendencia tecnológica es utilizar Arduino como tarjeta de adquisición de datos desarrollando interfaces en software como JAVA, Visual Basic y LabVIEW. Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo

integrado libre se puede descargar gratuitamente.

ESTRUCTURA DE PROGRAMACION

La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones.

```
void setup()  
{
```

```
  estamentos;  
}
```

```
void loop()  
{
```

```
  estamentos;  
}
```

En donde setup() es la parte encargada de recoger la configuración y loop() es la que contienen el programa que se ejecutará cíclicamente (de ahí el termino loop – bucle-). Ambas funciones son necesarias para que el programa trabaje.

La función de configuración debe contener la declaración de las variables. Es la primera función a ejecutar en el programa, se ejecuta sólo una vez, y se utiliza para configurar o inicializar pinMode (modo de trabajo de las E/S), configuración de la comunicación en serie y otras.

La función bucle (loop) siguiente contiene el código que se ejecutara continuamente (lectura de entradas, activación de salidas, etc) Esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo.

La función setup() se invoca una sola vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pins, o el puerto serie. Debe ser incluido en un programa, aunque no haya declaración que ejecutar.

```
void setup()  
{  
  pinMode(pin, OUTPUT); // configura el 'pin' como salida  
}
```

ELEMENTOS DE LA PROGRAMACION

Después de llamar a setup(), la función loop() hace precisamente lo que sugieresu nombre, se ejecuta de forma cíclica, lo que posibilita que el programa este respondiendo continuamente ante los eventos que se produzcan en la tarjeta

```
void loop()
```

```
{
digitalWrite(pin, HIGH); // pone en uno (on, 5v) el 'pin' delay(1000); // espera un
segundo (1000 ms) digitalWrite(pin, LOW); // pone en cero (off, 0v.) el
'pin' delay(1000);
}
```

FUNCIÓN

Una función es un bloque de código que tiene un nombre y un conjunto de estamentos que son ejecutados cuando se llama a la función. Son funciones `setup()` y `loop()` de las que ya se ha hablado. Las funciones de usuario pueden ser

escritas para realizar tareas repetitivas y para reducir el tamaño de un programa. Las funciones se declaran asociadas a un tipo de valor “type”. Este valor será el que devolverá la función, por ejemplo 'int' se utilizará cuando la función devuelva un dato numérico de tipo entero. Si la función no devuelve ningún valor entonces se colocará delante la palabra “void”, que significa “función vacía”. Después de declarar el tipo de dato que devuelve la función se debe escribir el nombre de la función y entre paréntesis se escribirán, si es necesario, los parámetros que se deben pasar a la función para que se ejecute.

```
type nombreFunción(parámetros)
{
estamentos;
}
```

La función siguiente devuelve un número entero, `delayVal()` se utiliza para poner un valor de retraso en un programa que lee una variable analógica de un potenciómetro conectado a una entrada de Arduino. Al principio se declara como una variable local, 'v' recoge el valor leído del potenciómetro que estará comprendido entre 0 y 1023, luego se divide el valor por 4 para ajustarlo a un margen comprendido entre 0 y 255, finalmente se devuelve el valor 'v' y se retornaría al programa principal. Esta función cuando se ejecuta devuelve el valor de tipo entero 'v'.

```
int delayVal()
{
int v; // crea una variable temporal 'v' v= analogRead(pot); //
v /= 4; // convierte 0-1023 a 0-255 return v; // devuelve el valor final
}
```

LLAVES {}

Las llaves sirven para definir el principio y el final de un bloque de instrucciones. Se utilizan para los bloques de programación `setup()`, `loop()`, `if..`, etc.

```
type funcion()
```

```
{  
estamentos;  
}
```

Una llave de apertura “{” siempre debe ir seguida de una llave de cierre “}”, si no es así el programa dará errores.

El entorno de programación de Arduino incluye una herramienta de gran utilidad para comprobar el total de llaves. Sólo tienes que hacer click en el punto de inserción de una llave abierta e inmediatamente se marca el correspondiente cierre de ese bloque (llave cerrada).

PUNTO Y COMA (;)

El punto y coma “;” se utiliza para separar instrucciones en el lenguaje de programación de Arduino. También se utiliza para separar elementos en una instrucción de tipo “bucle for”.

```
int x = 13; // declara la variable 'x' como tipo entero de valor 13
```

Nota: Olvidarse de poner fin a una línea con un punto y coma se traducirá en un error de compilación. El texto de error puede ser obvio, y se referirá a la falta de una coma, o puede que no. Si se produce un error raro y de difícil detección lo primero que debemos hacer es comprobar que los puntos y comas están colocados al final de las instrucciones.

BLOQUE DE COMENTARIO

Los bloques de comentarios, o multi-línea de comentarios, son áreas de texto ignorados por el programa que se utilizan para las descripciones del código o comentarios que ayudan a comprender el programa. Comienzan con `/ *` y terminan con `*/` y pueden abarcar varias líneas.

`/* esto es un bloque de comentario no se debe olvidar cerrar los comentarios estos deben estar equilibrados */`

Debido a que los comentarios son ignorados por el programa y no ocupan espacio en la memoria de Arduino pueden ser utilizados con generosidad y también pueden utilizarse para "comentar" bloques de código con el propósito de anotar informaciones para depuración.

Nota: Dentro de una misma línea de un bloque de comentarios no se puede escribir otra bloque de comentarios (usando /* .. */))

Una línea de comentario empieza con // y terminan con la siguiente línea de código. Al igual que los comentarios de bloque, los de línea son ignoradas por el programa y no ocupan espacio en la memoria.

// esto es un comentario

Una línea de comentario se utiliza a menudo después de una instrucción, para proporcionar más información acerca de lo que hace esta o para recordarla más adelante.

VARIABLE

Una variable es una manera de nombrar y almacenar un valor numérico para su uso posterior por el programa. Como su nombre indica, las variables son números que se pueden variar continuamente en contra de lo que ocurre con las constantes cuyo valor nunca cambia. Una variable debe ser declarada y, opcionalmente, asignarle

un valor. El siguiente código de ejemplo declara una variable llamada variableEntrada y luego le asigna el valor obtenido en la entrada analógica del PIN2:

```
int variableEntrada = 0; // declara una variable y le asigna el valor 0
variableEntrada = analogRead(2); // la variable recoge el valor analógico del PIN2
```

'variableEntrada' es la variable en sí. La primera línea declara que será de tipo entero "int". La segunda línea fija a la variable el valor correspondiente a la entrada analógica PIN2. Esto hace que el valor de PIN2 sea accesible en otras partes del código.

Una vez que una variable ha sido asignada, o re-asignada, usted puede probar su valor para ver si cumple ciertas condiciones (instrucciones if.), o puede utilizar directamente su valor. Como ejemplo ilustrativo veamos tres operaciones útiles con variables: el siguiente código prueba si la variable "entradaVariable" es inferior a 100, si es cierto se asigna el valor 100 a "entradaVariable" y, a continuación, establece un retardo (delay) utilizando como valor "entradaVariable" que ahora será como mínimo de valor 100:

```

if (entradaVariable < 100) // pregunta si la variable es menor de 100
{
  entradaVariable = 100; // si es cierto asigna el valor 100 a esta
}
delay(entradaVariable); // usa el valor como retardo

```

Nota: Las variables deben tomar nombres descriptivos, para hacer el código más legible. Nombres de variables pueden ser “contactoSensor” o “pulsador”, para ayudar al programador y a cualquier otra persona a leer el código y entender lo que representa la variable. Nombres de variables como “var” o “valor”, facilitan muy poco que el código sea inteligible. Una variable puede ser cualquier nombre o palabra que no sea una palabra reservada en el entorno de Arduino.

DECLARACIÓN DE VARIABLES

Todas las variables tienen que declararse antes de que puedan ser utilizadas. Para declarar una variable se comienza por definir su tipo como int (entero), long (largo), float (coma flotante), etc, asignándoles siempre un nombre, y, opcionalmente, un valor inicial. Esto sólo debe hacerse una vez en un programa, pero el valor se puede cambiar en cualquier momento usando aritmética y reasignaciones diversas. El siguiente ejemplo declara la variable entradaVariable como una variable de tipo entero “int”, y asignándole un valor inicial igual a cero. Esto se llama una asignación.

```
int entradaVariable = 0;
```

Una variable puede ser declarada en una serie de lugares del programa y en función del lugar en donde se lleve a cabo la definición esto determinará en que partes del programa se podrá hacer uso de ella.

UTILIZACIÓN DE VARIABLES

Una variable puede ser declarada al inicio del programa antes de la parte de configuración setup(), a nivel local dentro de las funciones, y, a veces, dentro de un bloque, como para los bucles del tipo if.. for.., etc. En función del lugar de declaración de la variable así se determinará el ámbito de aplicación, o la capacidad de ciertas partes de un programa para hacer uso de ella.

Una variable global es aquella que puede ser vista y utilizada por cualquier función y estamento de un programa. Esta variable se declara al comienzo del programa, antes de setup().

Una variable local es aquella que se define dentro de una función o como parte de un bucle. Sólo es visible y sólo puede utilizarse dentro de la función en la que se

declaró. Por lo tanto, es posible tener dos o más variables del mismo nombre en diferentes partes del mismo programa que pueden contener valores diferentes. La garantía de que sólo una función tiene acceso a sus variables dentro del programa simplifica y reduce el potencial de errores de programación.

El siguiente ejemplo muestra cómo declarar a unos tipos diferentes de variables y la visibilidad de cada variable:

```
int value; // 'value' es visible para cualquier función void setup()
{
  // no es necesario configurar
}
void loop()
{
  for (int i=0; i<20;) // 'i' solo es visible
  { // dentro del bucle for i++;
  }
  float f; // 'f' es visible solo
  }      // dentro del bucle
```

OPERADORES DE COMPARACIÓN

Las comparaciones de una variable o constante con otra se utilizan con frecuencia en las estructuras condicionales del tipo if.. para testear si una condición es verdadera. En los ejemplos que siguen en las próximas páginas se verá su utilización práctica usando los siguientes tipos de condicionales:

```
x == y // x es igual a y
x != y // x no es igual a y
x < y // x es menor que y
x > y // x es mayor que y
x <= y // x es menor o igual que y
x >= y // x es mayor o igual que y
```

OPERADORES LÓGICOS

Los operadores lógicos son usualmente una forma de comparar dos expresiones y devolver un VERDADERO o FALSO dependiendo del operador. Existen tres operadores lógicos, AND (&&), OR (||) y NOT (!), que a menudo se utilizan en estamentos de tipo if..:

Logical AND:

```
if (x > 0 && x < 5) // cierto sólo si las dos expresiones son ciertas
```

Logical OR:

if (x > 0 || y > 0) // cierto si una cualquiera de las expresiones es cierta

Logical NOT:

if (!x > 0) // cierto solo si la expresión es falsa

CONDICIONAL IF

if es un estamento que se utiliza para probar si una determinada condición se ha alcanzado, como por ejemplo averiguar si un valor analógico está por encima de un cierto número, y ejecutar una serie de declaraciones (operaciones) que se escriben dentro de llaves, si es verdad. Si es falso (la condición no se cumple) el programa salta y no ejecuta las operaciones que están dentro de las llaves, El formato para if es el siguiente:

```
if (unaVariable ?? valor)
{
    ejecutaInstrucciones;
}
```

En el ejemplo anterior se compara una variable con un valor, el cual puede ser una variable o constante. Si la comparación, o la condición entre paréntesis se cumple (es cierta), las declaraciones dentro de los corchetes se ejecutan. Si no es así, el programa salta sobre ellas y sigue.

Nota: Tenga en cuenta el uso especial del símbolo '=', poner dentro de if (x = 10), podría parecer que es valido pero sin embargo no lo es ya que esa expresión asigna el valor 10 a la variable x, por eso dentro de la estructura if se utilizaría X==10 que en este caso lo que hace el programa es comprobar si el valor de x es 10.. Ambas cosas son distintas por lo tanto dentro de las estructuras if, cuando se pregunte por un valor se debe poner el signo doble de igual "=="

CONDICIONAL IF ELSE

if... else viene a ser una estructura que se ejecuta en respuesta a la idea “si esto no se cumple haz esto otro”. Por ejemplo, si se desea probar una entrada digital, y hacer una cosa si la entrada fue alta o hacer otra cosa si la entrada es baja, usted escribiría que de esta manera:

```
if (inputPin == HIGH) // si el valor de la entrada inputPin es alto
{
instruccionesA; //ejecuta si se cumple la condición
}
Else
{

instruccionesB; //ejecuta si no se cumple la condición
```

Else puede ir precedido de otra condición de manera que se pueden establecer varias estructuras condicionales de tipo unas dentro de las otras (anidamiento) de forma que sean mutuamente excluyentes pudiéndose ejecutar a la vez. Es incluso posible tener un número ilimitado de estos condicionales. Recuerde sin embargo qué sólo un conjunto de declaraciones se llevará a cabo dependiendo de la condición probada:

```
if (inputPin < 500)
{

instruccionesA; // ejecuta las operaciones A
}

else if (inputPin >= 1000)
{
instruccionesB; // ejecuta las operaciones B
}
else
{
instruccionesC; // ejecuta las operaciones C
}
```

Nota: Un estamento de tipo if prueba simplemente si la condición dentro del paréntesis es verdadera o falsa. Esta declaración puede ser cualquier declaración válida. En el anterior ejemplo, si cambiamos y ponemos (inputPin == HIGH). En este caso, el estamento if sólo chequearía si la entrada especificada está en nivel alto (HIGH), o +5 v.

SENTENCIA FOR

La declaración for se usa para repetir un bloque de sentencias encerradas entre llaves un número determinado de veces. Cada vez que se ejecutan las instrucciones del bucle se vuelve a testear la condición. La declaración for tiene tres partes separadas por (;) vemos el ejemplo de su sintaxis:

```
for (inicialización; condición; expresión)
{
```

La inicialización de una variable local se produce una sola vez y la condición se testea cada vez que se termina la ejecución de las instrucciones dentro del bucle. Si la condición sigue cumpliéndose, las instrucciones del bucle se vuelven a ejecutar. Cuando la condición no se cumple, el bucle termina.

Nota: El bucle en el lenguaje C es mucho más flexible que otros bucles encontrados en algunos otros lenguajes de programación, incluyendo BASIC. Cualquiera de los tres elementos de cabecera puede omitirse, aunque el punto y coma es obligatorio. También las declaraciones de inicialización, condición y expresión puede ser cualquier estamento válido en lenguaje C sin relación con las variables declaradas. Estos tipos de estados son raros, pero permiten disponer soluciones a algunos problemas de programación raras.

SENTENCIA WHILE

Un bucle del tipo while es un bucle de ejecución continua mientras se cumpla la expresión colocada entre paréntesis en la cabecera del bucle. La variable de prueba tendrá que cambiar para salir del bucle. La situación podrá cambiar a expensas de una expresión dentro el código del bucle o también por el cambio de un valor en una entrada de un sensor

```
while (unaVariable ?? valor)
{
    ejecutarSentencias;
}
```

El siguiente ejemplo testea si la variable "unaVariable" es inferior a 200 y, si es verdad, ejecuta las declaraciones dentro de los corchetes y continuará ejecutando el bucle hasta que 'unaVariable' no sea inferior a 200.

```
While (unaVariable < 200) // testea si es menor que 200
{
```

```
instrucciones; // ejecuta las instrucciones entre llaves unaVariable++; // incrementa la variable en 1  
}
```

SERIAL

Serial.begin

Abre el puerto serie y fija la velocidad en baudios para la transmisión de datos en serie. El valor típico de velocidad para comunicarse con el ordenador es 9600, aunque otras velocidades pueden ser soportadas.

```
void setup()  
{  
  Serial.begin(9600); // abre el Puerto serie  
} // configurando la velocidad en 9600 bps
```

Nota: Cuando se utiliza la comunicación serie los pins digital 0 (RX) y 1 (TX) no puede utilizarse al mismo tiempo.

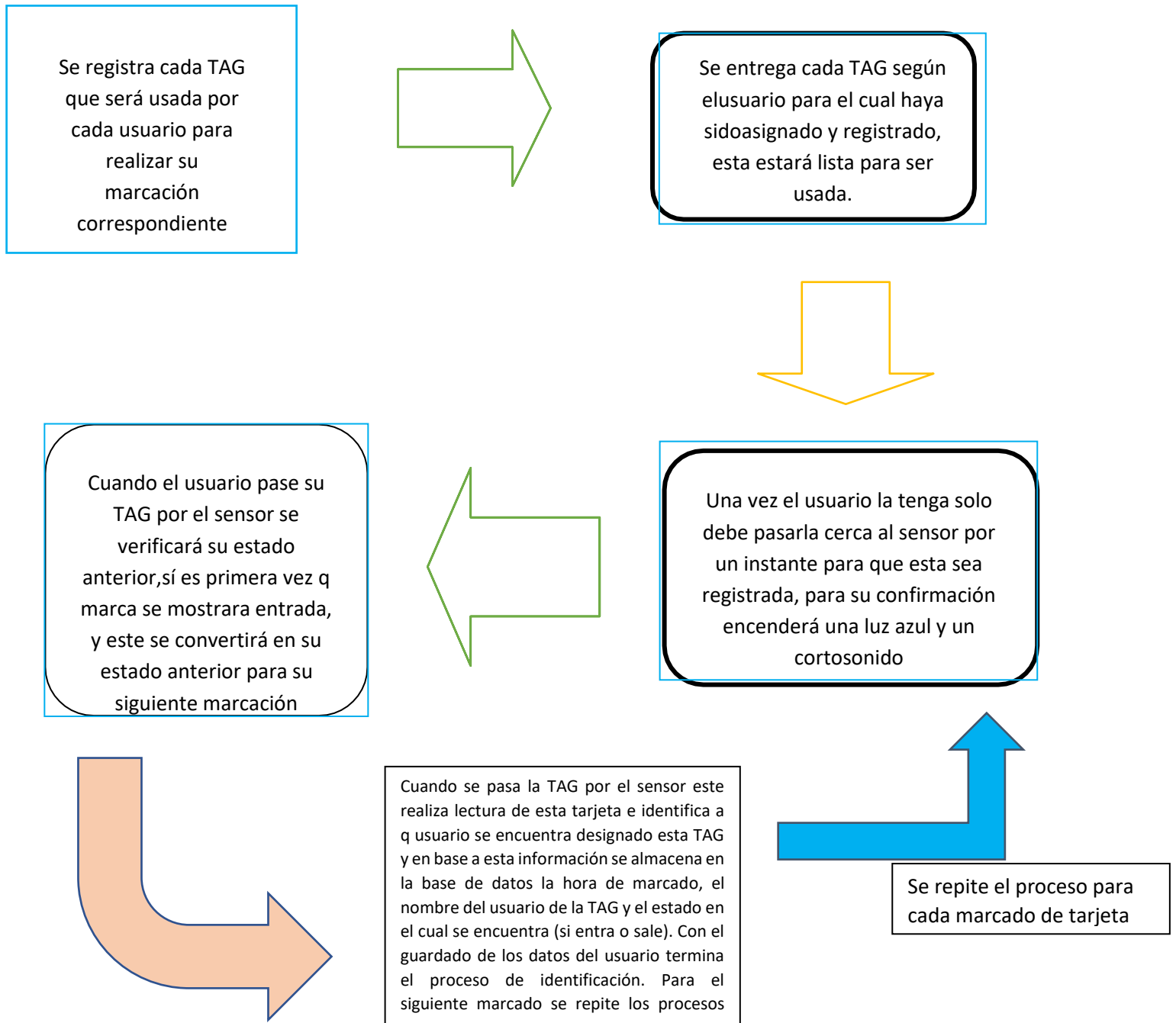
Serial.println(data)

Imprime los datos en el puerto serie, seguido por un retorno de carro automático y salto de línea. Este comando toma la misma forma que Serial.print (), pero es más fácil para la lectura de los datos en el Monitor Serie del software.

Serial.println(analogValue); // envía el valor 'analogValue' al puerto

DIAGRAMAS

DIAGRAMA DE FLUJO DEL SISTEMA



BLOQUES DEL SISTEMA

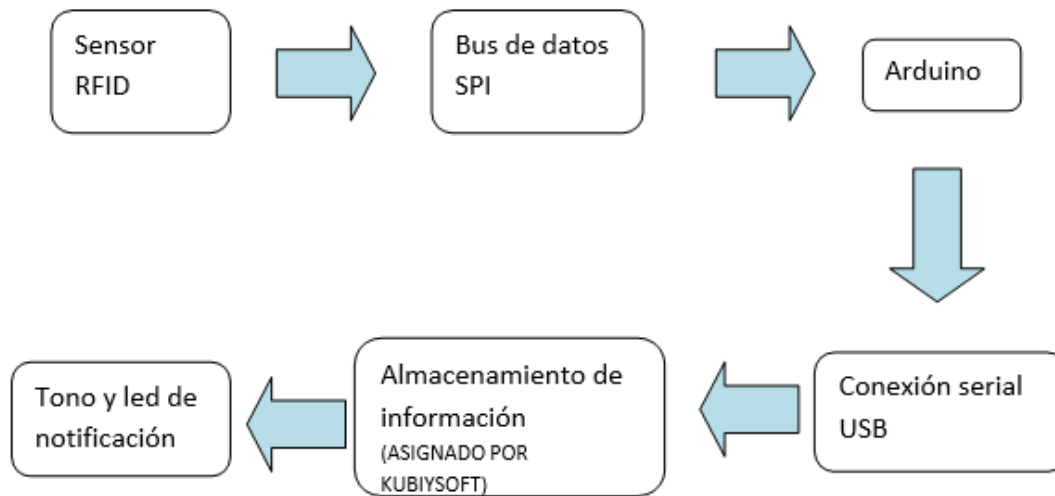


Figura 9. Diagrama de bloques del circuito. Imagen propia.

Bloque 1 (Sensor RFID 522RC).- El sensor RFID se encarga de generar un campo magnético pequeño para poder identificar la tarjeta de identificación, el cual contiene una antena elicoidal para la recepción de este campo magnético, una vez identificada, realiza la lectura de su código serial y la envía a través del bus SPI que va conectado al Arduino.

Bloque 2 (Conexión SPI).- los datos emitidos por el sensor son transmitidos en el bus con la siguiente distribución: pines SDA, SCK, MOSI, MISO, GND, RESET y 3.3v en el sensor, donde los cuatro primeros pines se encargan de realizar la comunicación mediante el protocolo SPI y los últimos 3 se encargan de realizar la alimentación correspondiente para su funcionamiento, el Pin SDA se encarga de verificar control del sensor, SCK se encarga de generar una señal de reloj entre el Arduino y el sensor, MOSI se encarga de recibir la información del Arduino y MISO de enviar información al Arduino, GND, 3.3v son la alimentación del sensor y RESET se encarga de reiniciar los valores del sensor para la siguiente lectura. Por parte del Arduino la conexión en los pines determinados corresponden a los designados por la librería ya proporcionada por la plataforma Arduino.

Bloque 3 (Arduino).- Una vez que el Arduino recibe la información del sensor este procesa la señal convirtiendo una serie de datos ASCII en un código de tipo String con el cual determina a que usuario pertenece, y mediante un pequeño algoritmo de tipo lógico se determina el estado de el usuario (si entra o sale), todo esto ocurre dentro del Arduino, de manera que este realiza todo el proceso principal y una vez obtenido los resultados solamente los envía de manera serial al computador a través de conexión USB directa con la PC.

Bloque 4 (Conexión USB).- todos los datos generados por el Arduino son enviados de manera serial hacia el computador, de manera que este sea capturado por la aplicación PLX-DAQ el cual trabaja sobre comunicación serial, para identificar la separación requiere que sea transmitida el carácter coma (","), de esta manera logra separar cada uno de los datos que son transmitidos por este medio.

Bloque 5 (Base de Datos).- Toda información recibida dentro del computador se almacenará en una base de datos, esto es posible gracias a la aplicación XAMPP la cual tiene la función de transferir todos los datos recibidos vía serial a través de los puertos COM, en tiempo real, logrando de esta manera se tenga una base de datos actualizada en todo momento.

Bloque 6 (Leds y Buzzer).- Este bloque es el que verifica el registro de una tarjeta, cada vez que sea reconocida la tarjeta, este procederá a reproducir un sonido por un momento para que el usuario constate que se realizó correctamente la lectura, además de encender un led, este bloque va conectado al pin digital 8 del Arduino, este pin se encarga de generar el tono y encender el led de notificación después de haberse verificado la lectura de la tarjeta y haberla guardado en la base de datos.

RECURSOS UTILIZADOS EN ARDUINO

Para la creación del programa en Arduino se hizo la utilización de las siguientes librerías:

Librería SPI (`#include <SPI.h>`)

La Interfaz Periférica Serial (SPI) es un protocolo de datos en serie síncrono utilizado por microcontroladores para comunicarse con uno o más dispositivos

periféricos rápidamente en distancias cortas. También se puede utilizar para la comunicación entre dos microcontroladores.

Con una conexión SPI siempre hay un dispositivo maestro (generalmente un microcontrolador) que controla los dispositivos periféricos. Normalmente hay tres líneas comunes a todos los dispositivos:

MISO (Master In Slave Out) - La línea Slave para enviar datos al maestro,

MOSI (Master Out Slave In) - La línea maestra para enviar datos a los periféricos,

SCK (Serial Clock) - Los pulsos de reloj que sincronizan la transmisión de datos generada por el maestro Y una línea específica para cada dispositivo:

SS (Slave Select): el pin de cada dispositivo que el maestro puede usar para habilitar y deshabilitar dispositivos específicos.

Esta librería genera todo lo necesario para poder usar el protocolo de comunicación SPI, incluyendo la designación de pines, el estado de los pines y la velocidad.

Librería MFRC522 (`#include <MFRC522.h>`)

Esta librería es creada por la empresa MIFARE como soporte para la utilización de sus diversos tipos de lectores de RFID, dentro de esta librería al igual que la anterior librería, designa los puertos a ser utilizados, designa la funcionalidad de los pines, además de que dentro de esta se encuentran rutinas para la lectura, escritura, administración de datos que se encuentran en la tarjeta a trabajar.

Modulo RFID-RC522 Mifare

El Módulo Lector RFID-RC522 RF utiliza 3.3V como voltaje de alimentación y se controla a través del protocolo SPI, así como el protocolo UART, por lo 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. Como 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 hasta 16 Bytes. El número de serie consiste de 5 valores hexadecimales, se podría utilizar esto para hacer una operación dependiendo del número de serie.

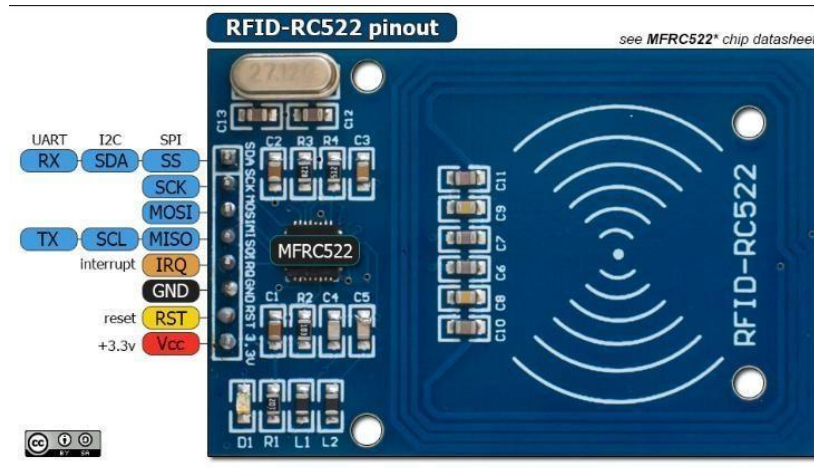


Figura 10. Lector RFID RC522 con designación de pines. Recuperado de <https://ja-bots.com/producto/rfid-rc522/>

Características del Módulo Lector RFID-RC522 RF

- Modelo: MF522-ED
- Corriente de operación: 13-26mA a 3.3V
- Corriente de stand by: 10-13mA a 3.3V
- Corriente de sleep-mode: <80uA
- Corriente máxima: 30mA
- Frecuencia de operación: 13.56Mhz
- Distancia de lectura: 0 a 60mm
- 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:
- Protocolo de comunicación: SPI
- 10Mbit/sLa conexión a realizar

será la siguiente:

ARDUINO UNO	RFID RC522
DIGITAL PIN #10	SDA
DIGITAL PIN #13	SCK
DIGITAL PIN #11	MOSI
DIGITAL PIN #12	MISO
N/A	IRQ
POWER GND	GND
DIGITAL PIN #5	RST
POWER 3.3 V	3.3 V

TABLA 2. Correlación pines Arduino – Lector RFID RC522. Recuperado de www.hetpro.com

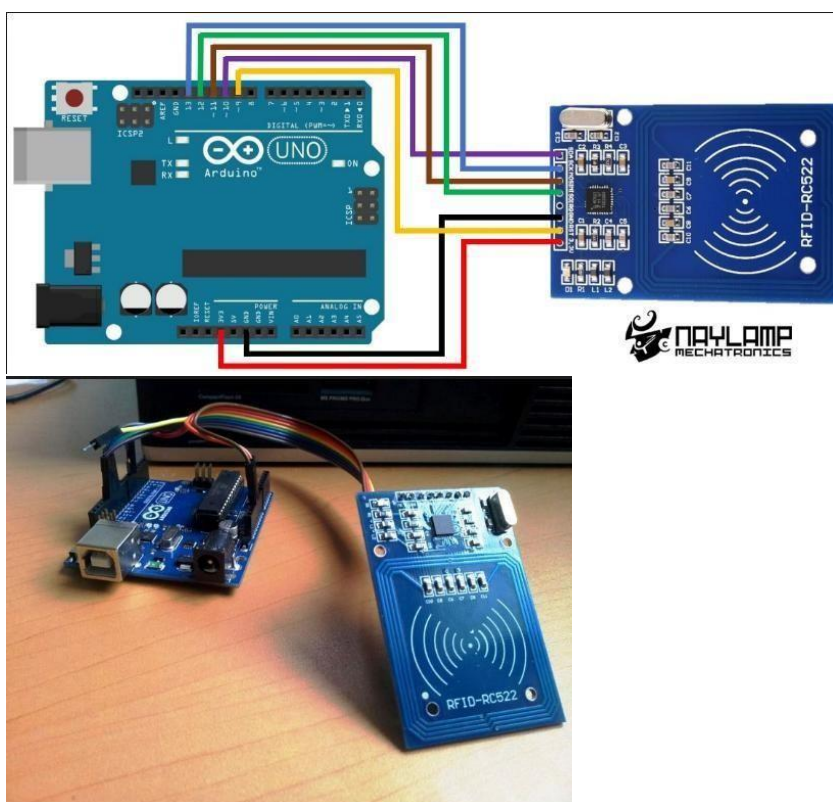


Figura 11. Diagrama de conexión Arduino – Lector RFID RC522. Recuperado de www.nfc.com

Características Tarjeta RFID

Para el sensor RFID RC522 de Mifare, se cuenta con tres versiones de tarjetas disponibles con las cuales puede operar a corta distancia, que es lo que se busca para tener una buena lectura de tarjeta, en la siguiente tabla se detalla las características de estos tres tipos de tarjetas disponibles.

Característica	MFARE UltralightS	NTAG203	NTAG210	NTAG213	NTAG215
Capacidad Memoria	64 bytes	160 bytes	80 bytes	180 bytes	540 bytes
Memoria Usuario	48 bytes	144 bytes	48 bytes	144 bytes	504 bytes
Longitud URL Caracteres	41	132	41	132	492
Compatibilidad Móvil	Si	Si	Si	Si	Si
NFC Forum TS	Si	Si	Si	Si	Si
Número de Serie	Si 7 bytes	Si 7 bytes	Si 7 bytes	Si 7 bytes	Si 7 bytes
Criptografía	No	No	No	No	No
Fortaleza de escaneo	Media	Alta	Alta	Muy Alta	Alta

Tabla 3. Características etiquetas Mifare. Recuperado de <https://www.tangoid.com.ar/que-es-nfc>

Para este proyecto se hará uso de las Etiquetas Mifare 1K por ser una de las más económicas que existe en el mercado, además de que estas cuentan con cifrado para una mayor seguridad ante una posible clonación, y que también es una de las más versátiles por la capacidad de memoria comparada con las etiquetas Ultraligh, lo que permite que con el tiempo se le puedan agregar más funciones a estas etiquetas.



Figura 13. Formas de etiquetas RFID-NFC. Recuperado de www.rfid.com

Programa generador de código de tarjeta.

El siguiente programa es el necesario para la identificación de la tarjeta, este programa generará un código el cual será utilizado en el programa principal para la identificación del usuario y de su estado de ingreso o salida.

```
#include <SPI.h> #include <MFRC522.h>

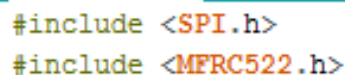
#define RST_PIN 9
#define SS_PIN 10
MFRC522 mfrc522(SS_PIN, RST_PIN); // Crea instancia MFRC522 void setup() {
Serial.begin(9600); // Inicializa comunicacion serial con la
.begin(); // Inicializa bus SPI mfrc522.PCD_Init(); // Inicia tarjeta MFRC522
Serial.println(F("Ingrese Tarjeta Personal"));
}
void loop() { MFRC522::MIFARE_Key key;
for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

if ( ! mfrc522.PICC_IsNewCardPresent()) { return;
}

if ( ! mfrc522.PICC_ReadCardSerial()) return; String codigo ="";
Serial.print(F("Num. de Serie unico:")); //Imprime numero serie de tarjeta for (byte i
= 0; i < mfrc522.uid.size; i++) { Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
Serial.print(mfrc522.uid.uidByte[i], HEX);
codigo = codigo + (mfrc522.uid.uidByte[i]);
}
Serial.println(","); Serial.println(codigo); delay(2000);
}
```

Detalles del programa:

Se incluye las librerías SPI la cual se encarga del protocolo de comunicación SPI y la librería RFID, la cual contiene todas las funciones relacionadas con el lector.



```
#include <SPI.h>
#include <MFRC522.h>
```

Figura 14. Parte del programa, adición de librerías. Imagen propia.

```

void setup() {
  Serial.begin(9600); // Inicializa comunicacion serial con la PC
  SPI.begin(); // Inicia bus SPI
  mfrc522.PCD_Init(); // Inicia tarjeta MFRC522
  Serial.println(F("Ingrese Tarjeta Personal"));
}

```

Figura 15. Parte del programa, inicialización. Imagen propia.

Se comprueba que siga siendo la misma tarjeta. La contraseña por defecto en las tarjetas es 0xFFFFFFFFh, al ingresar esta contraseña se puede leer el código serie de la tarjeta, a medida que se va leyendo se va almacenando los números decaracteres dentro de un String, el cual representara el código de la tarjeta para su posterior identificación.

```

if ( ! mfrc522.PICC_ReadCardSerial()) return;
String codigo = "";
Serial.print(F("Num. de Serie unico:")); //imprime UID
for (byte i = 0; i < mfrc522.uid.size; i++) {
  Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
  Serial.print(mfrc522.uid.uidByte[i], HEX);
  codigo = codigo + (mfrc522.uid.uidByte[i]);
}
Serial.println(",");
Serial.println(codigo);
delay(2000);
}

```

Figura 16. Parte del programa, método de lectura de código serie. Imagen propia.

Una vez ya obtenido el condigo se procederá con el registro de esta en el programa principal, el cual se encargará del control y registro de cada tarjeta que pase por el sensor.

Programa de Control y Monitoreo de tarjeta

Una vez ya obtenido todos los códigos de las tarjetas que serán utilizadas para el control se procederá con la introducción de estos en el sistema para que puedan ser identificados por el sensor, el cual identificara si están ingresando o saliendo de un sitio, además de registrarlo en la base de datos asignada por KUBIYSOFT, la cual será manipulada por el Arduino para la introducción de datos.

El programa principal es el siguiente.

```

#include <SPI.h> #include <MFRC522.h> #define RST_PIN 9
#define SS_PIN 10

```



```

int j=0; int ok =0;
while (ok == 0){
if (tarj==code[j]){ nom = Nom[j];
digitalWrite (8, HIGH); if (st&&esta[j]==1){ est="entrada";
}
else{
est ="salida";
}
esta[j] = ~esta[j]; delay(500);
for (i=fMin; i<=fMax;i++){ tone(7,i,duracion);
}
digitalWrite (8, LOW); ok++;
} j++;
}

```

```

linea++; //

```

```

Serial.print("DATA,TIME,"); //inicia a impresion de datos Serial.print(nom);
Serial.print(","); Serial.println(est);

```

```

if (linea > 1000) //es el limite de registros que realizara
{
linea = 0;
Serial.println("ROW,SET,2"); // posiciona la escritura en la linea dos de la hoja de
Excel
}
delay(1000); // espera 1 segundo
}

```

Detalles del programa

Al igual que en el anterior programa, en este de la misma manera se añaden las librerías de SPI y RFID, para la habilitación del protocolo SPI y de las funciones del lector respectivamente, de la misma manera se hace lectura del código serial de la tarjeta, se almacena la codificación dentro de una String llamada tarj, la cual es comparada con el cada uno de los valores almacenados dentro del vector en el cual se almacenaron todas las tarjetas habilitadas para ser controladas por el sensor.

```

String code[] = {"22712234213", "2", "761038475", "4", "5", "6", "7", "8", "9", "10",
                "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
                "21", "22", "23", "24", "25", "26", "27", "28", "29", "30"};
String Nom[] = {"User1", "User2", "User3", "User4", "User5", "User6", "User7", "User8", "User9", "User10",
               "User11", "User12", "User13", "User14", "User15", "User16", "User17", "User18", "User19", "User20",
               "User21", "User22", "User23", "User24", "User25", "User26", "User27", "User28", "User29", "User30"};

```

Figura 17. Vectores en los cuales se almacena los códigos de las tarjetas. Imagen propia.

Una vez que se la tarjeta detectada es identificada y se verifica su coincidencia con alguna de las almacenadas, se procede con la exportación del nombre del usuario y del estado de ingreso o salida de esta tarjeta a la base a través del programa XAMPP para que sea almacenado junto con la hora exacta en la que el sensor hizo la captura de datos de la tarjeta. Realizadas las pruebas para verificar el correcto funcionamiento se procede con la implementación del sistema completo para la puesta en funcionamiento de este.



Figura 18. Ejemplo de forma de registrar la tarjeta.

PROGRAMA PLX-DAQ (USO OPCIONAL)

Para la comunicación entre el dispositivo Arduino y la hoja de Excel se utilizará el programa PLX-DAQ, la cual es un aporte gratuito por parte de la empresa PARALLAX, la cual permite a cualquier microcontrolador o placa de desarrollo interactuar con una plantilla de Excel en forma serial por medio de cadenas de texto.

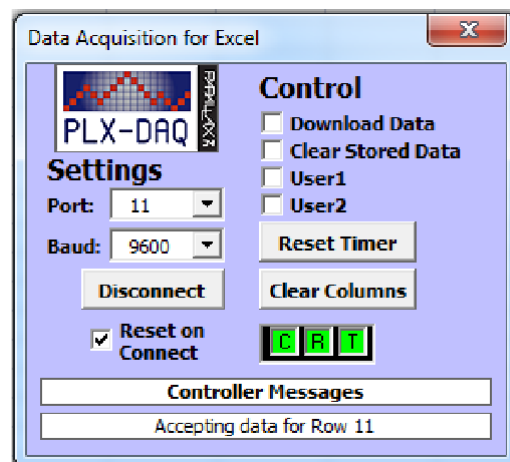


Figura 19. Programa PLX-DAQ de PARALLAX. Recuperado de www.parallax.com

Esta aplicación permite que todos los datos generados por el microcontrolador o placa de desarrollo, en este caso ARDUINO UNO R3, gestionen todos los datos proporcionados por este y almacenarlos dentro de una hoja de cálculo de Excel.

El programa PLX-DAQ es una aplicación gratuita ofrecida por la empresa PARALLAX.inc con fines prácticos a fin de acortar los procesos del microcontrolador para comunicarse con un computador a fin de almacenar datos por medio de una hoja de cálculo (Para Windows en programa Excel) y de esta manera dejar al microprocesador realizar más tareas pudiendo comunicarse de manera serial sin tener que gestionar la comunicación por completo con el computador.

La forma de comunicación es bastante simple, todos los datos que recibe los va introduciendo en una misma celda, hasta que encuentre el carácter “,” (coma), cuando encuentra este valor cambia de celda y de la misma manera todos los valores que vaya recibiendo los almacenara hasta que encuentre la siguiente coma, este carácter llega a ser la clave para el cambio de celda.

CIRCUITO FINAL

Para el circuito final a utilizar se utilizará el Arduino conectado al sensor RFID RC522 y conectado además dos diodos led que serán los que informarán de forma visual que se ha leído una tarjeta y una sirena que sonara para cada lectura de tarjeta, el diagrama físico a utilizar se refleja en la imagen siguiente.

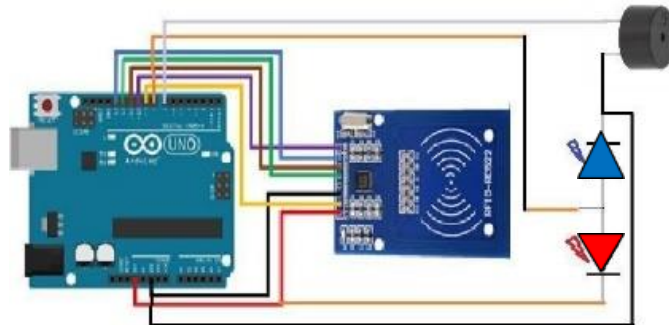


Figura 20. Circuito Final para utilizar. Imagen propia.

INSTALACIÓN DEL SISTEMA

Para la instalación se realizó la utilización de cable UTP para la ubicación e instalación del sensor de radiofrecuencia, por otra parte se tiene conectado directamente el Arduino al equipo que se utilizara como servidor de base de datos para el almacenamiento de la información generada por este sensor. este sistema tiene que estar ubicado de manera que todo tipo de personas puedan acercar su TAG de manera fácil y sencilla para que sea lo más rápido posible, en la siguiente imagen se presenta una imagen del prototipo instalado en el cual se presenta de manera demostrativa.



Figura 21. Implementación de sistema en la puerta principal. Imagen propia.

Una vez ya instalado solo es necesario que se pase con la TAG otra manera de activar el sensor será con una pulsera que tiene la misma función que una tarjeta NFC, cerca del sensor para poder realizar el registro de este.

CRITERIO DE IMPLEMENTACIÓN

Se tomo como principio para el criterio de instalación la necesidad de generar un método de control de movimiento de personal que ingresa y sale del lugar, además de controlar la hora exacta del movimiento, para eso este sistema se implementara justo dentro de la puerta principal para que una vez ingrese no se demore en realizar el registro de la TAG, de esta manera se podrá reducir sustancialmente el tráfico de ingreso a los demás ambientes de la empresa, agilizando el proceso de registro del

personal al hacerse automáticamente gracias al sistema proporcionado.

COTIZACION DE PROYECTO

ESTUDIO ECONOMICO DEL PROYECTO

El análisis costo/beneficio del sistema propuesto pretende demostrar de manera técnica y numérica, la rentabilidad y sostenibilidad del proyecto tomando en cuenta todos los costos que serán generados tanto en la implementación del hardware como en las licencias necesarias correspondientes al software necesario para su correcta ejecución.

COSTO DE HARDWARE

Para todo lo que se refiere en cuanto a HARDWARE los artículos necesarios son los siguientes:

- Arduino UNO R3.....(x2)
- Sensor RFID RC522(x2)
- o TAG's Mifare 1K... (x1)
- Rentabilidad de equipo de cómputo (cargar programa)
- Programa
- Carcasa del sistema (1)
- Equipo de instalación
- Artículos Varios

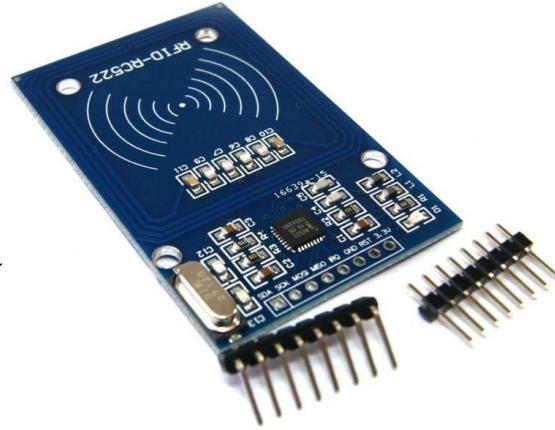
para la evaluación de costos se tomaron en cuenta los costos proporcionados por las principales empresas creadoras o distribuidoras del producto

Arduino UNO R3: tiene un costo unitario de 200 \$.



Figura 22. Arduino UNO R3. Recuperado de www.arduino.cc

SENSOR RFID RC522 : Tiene un costo unitario de 100 \$



Figur

RC522.com

TAG's Mifare 1K: Tiene un costo unitario de 100 \$. (individual)

Figura 24. Etiquetas RFID compatibles con frecuencia NFC. Recuperado de www.rfid-RC522.com



ARTICULOS VARIOS: Dentro de este se encuentran diferentes productos los cuales tienen un costo mínimo por separado, pero en conjunto tiene un costo aproximado de 60 \$ por metro. Entre los artículos más relevantes se encuentra el cable a ser utilizado, diodos led, parlante y otros.

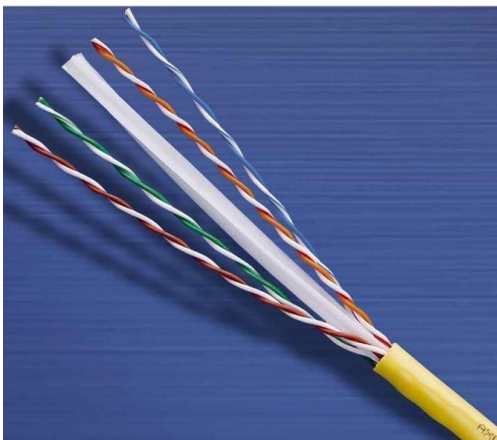


Figura 25. Cable UTP imagen propia



Figura 26. Diodos LED de Bajo Consumo



Figura 27. Buzzer de 8Ω.

COSTOS DE SOFTWARE

Para este proyecto, en cuanto respecta a costos de SOFTWARE se tiene que tomar muy en cuenta que, aunque la descarga y el manejo de algunas aplicaciones sea gratuita, no significa que no tenga un costo para su utilización en sistemas para instituciones, pues estas para su libre utilización dentro de cualquier empresa requiere de la respectiva licencia del programa para uso comercial.

Los software's a utilizar dentro de este proyecto son:

- Arduino IDE
- Microsoft Office
- XAMPP

Arduino IDE: Tiene un costo gratuito la licencia, pero el programa generado con esta aplicación tendrá un costo de 50 \$US.



Figura 28. Imagen del programa ARDUINO IDE. Recuperado de www.arduino.cc

MICROSOFT OFFICE: La licencia para el uso de esta aplicación para fines comerciales está determinado por la cantidad de equipos que van a ser utilizados, el costo de esta única licencia es de 289.99 \$US.



Figura 29. Licencia MICROSOFT OFFICE. Recuperado de www.office.com

PLX-DAQ de PARALLAX.inc: Para el uso comercial de este programa se debe para un monto para su licencia en cual tiene un valor de 10 \$US. (USO OPCIONAL)



Figura 30. Insignia PARALLAX. Recuperado de www.parallax.com

COTIZACIÓN DE PROYECTO

En la siguiente tabla se detallan todos los costos y se obtiene el valor total del costo económico.

Elementos del programa	Cantidad	Costo (\$)
Arduino UNO R3	2	400

Sensor RFID RC522	2	200
TAG's Mifare	1	100
Licencia Microsoft Office	1	289.99
Licencia PARALLAX	1	15 (uso opcional)
Carcasa del sistema	1	200
Equipo de instalación	1	250
Desarrollo del código		500
Artículos Varios		60
	TOTAL	1,999.9

Tabla 4. Costo total de instalación.

Con referencia al costo total de producción, se verifica que los gastos mayoritarios están basados a la licencia de uso del programa Microsoft Office para el uso de su aplicación Excel (USO OPCIOONAL), se debe tener muy en cuenta el costo de las licencias en productos que requieren de estas aplicaciones, la omisión de estos costos puede traer muchos problemas en cuanto a la legalidad del programa puesto que se estaría utilizando de manera ilegal las aplicaciones.

Evidencias de Desarrollo

Figura 31. Presentacion de directivos y asesores

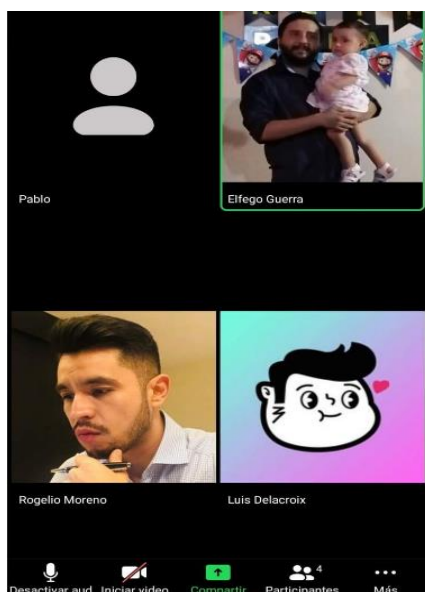


Figura 32. Introducción a la base de datos

(Introducción al proyecto)

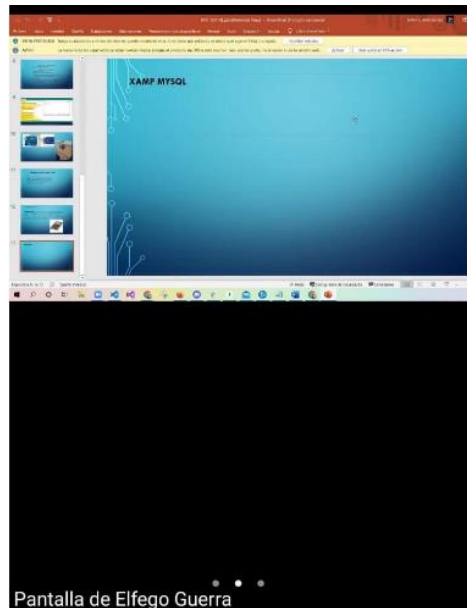


Figura 33. Pruebas de conexión y lector NFC

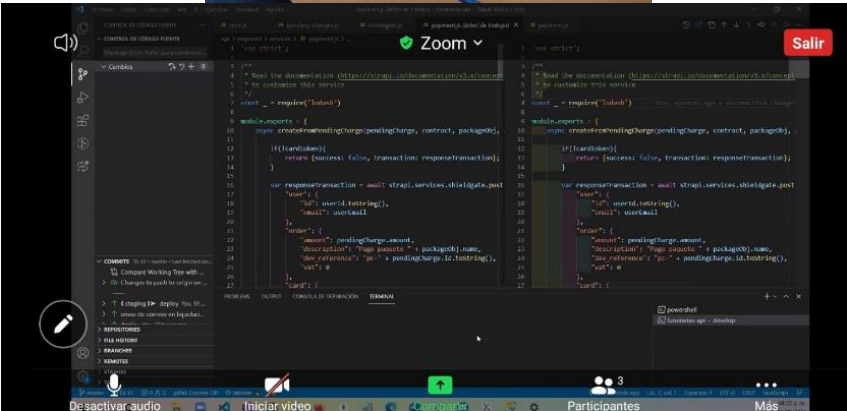
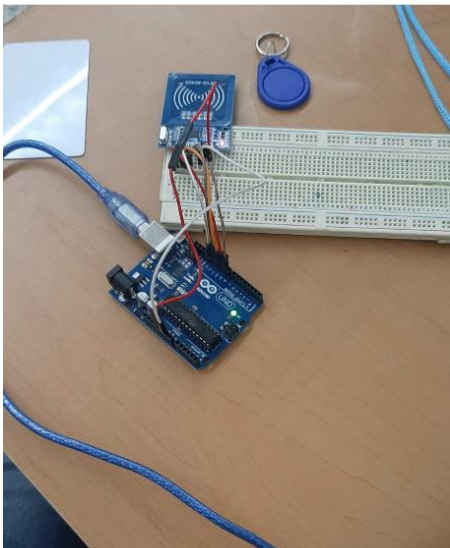


Figura 34. Programa final y funcionando

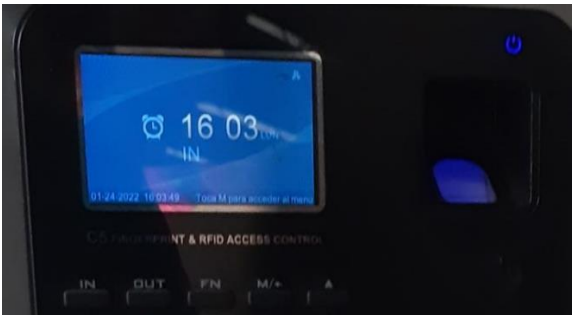


Figura 35. Instalacion final y funcionando.

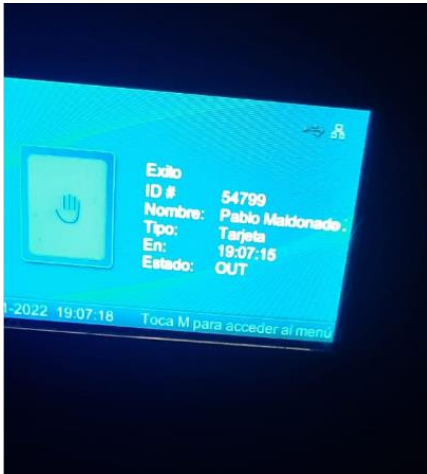


Figura 36. Lectura correcta. Proyecto finalizado.

(POR DERECHOS DE AUTOR Y DE LA EMPRESA EMPRESA NO SE MOSTRARÁ TODO EL CONTENIDO)
PHP y etiquetas RFID.

Tabla de contenido

[1 Lectoí RFID](#)

Lector RFID

Lo que yo usaré es el módulo NodeMCU ESP8266 y el sensor MFRC522 RFID-RC522 y una LCD de 16 x 2 usando un módulo I2C. La comunicación con el servidor la haremos a través del WiFi que ya viene integrado en la ESP8266.

El circuito de conexión queda de la siguiente manera.

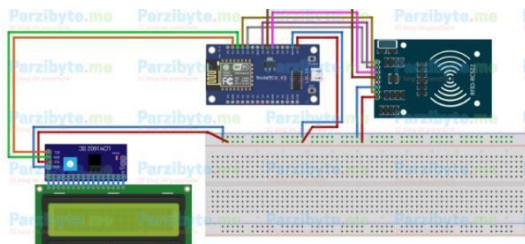


Figura 36. Diagrama de conexión

2 lecturas recomendadas

Visual Studio Code para programar la tarjeta. Vamos a hace peticiones HTTP a PHP, además de leeR y escribir datos en las tarjetas RFID. También vamos a imprimir en la LCD, para lo cual necesitamos conocer la dirección I2C.

3 explicación de la arquitectura

Para realizar una compra o una recarga he escrito el frontend en JavaScript. Lo que se hace es simplemente indicarle al servidor PHP la orden que le tiene que dar a la tarjeta, y después de eso igualmente comenzamos a realizar peticiones cada 2 segundos para ver lo que la tarjeta ha respondido.



Figura 37. LCD

- Recargar saldo: el servidor le ordenará a la tarjeta que recargue saldo al usuario.
- Descontar saldo: el servidor va a indicar que la tarjeta descuente saldo si es que eso es posible.
- Mientras no haya órdenes por parte del servidor, la ESP8266 va a estar consultando cada 5 segundos. tan pronto reciba una orden, la va a ejecutar e indicar los resultados a través de la LCD.
- Para realiza una compra o una recarga he escrito el frontend en JavaScript. Lo que se hace es simplemente indicarle al servidor PHP la orden que le tiene que dar a la tarjeta, y después de eso igualmente comenzamos a realizar peticiones cada 2 segundos para ver lo que la tarjeta ha respondido.

Código para la tarjeta ESP8266

El código completo de la tarjeta. cambiar la dirección del I2C, así como las credenciales para conectar a la red Wifi y la dirección de tu servidor, que puede ser de la red local o un servidor en internet (para esto último se configurará el certificado en caso de usar https). Las funciones de guardarSaldo y obtenerSaldo, pues son muy importantes y se encargan de leer o escribir los bloques de la tarjeta RFID.

4 código

Para la tarjeta ESP8266

NOTA: (CODIGO FUENTE, CON LINEAS RESTRINGIDAS)

```
#include <ESP8266HTTPClient.h>
#include <Arduino.h>
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal_I2C.h>

#define LONGITUD_BYTES 18
#define LONGITUD_BYTES_ESCRITURA 16
#define SEPARADOR_ORDENES_SERVIDOR ";"
#define ORDEN_RECARGAR "r"
#define ORDEN_DESCONTAR "d"
/*
Pines para conectar el lector
*/
#define RST_PIN D3
#define SS_PIN D4

// Credenciales para conectar a la red

const char *ssid = "Aquí va el nombre de tu red";
const char *password = "Aquí va la contraseña de tu red";
const String DIRECCION_SERVIDOR = "http://kubiysoft
id(perzonalizado)/rfid_saldo";

// Conexión a WiFi
ESP8266WiFiMulti wifiMulti;
// La LCD
LiquidCrystal_I2C lcd(0x3f, 16, 2);

// El lector
MFRC522 lector(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key clave;
int tarjetaLeidaDesdeLoop = 0;
```

```
double saldoGlobal;

void obtenerSaldo(double *saldo)
{

    if (!lector.PICC_IsNewCardPresent())
    {

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("SALDO:");
        lcd.setCursor(6, 0);
        lcd.print(saldoGlobal);
        lcd.setCursor(0, 1);
```

CONCLUSIONES, RECOMENDACIONES Y EXPERIENCIA PROFESIONAL ADQUIRIDA

En conclusión, los objetivos del proyecto se cumplieron satisfactoriamente porque gracias al control de asistencia aplicado en el área de producción de la KUBI y Soft se mejoró la productividad, aumentó el índice de satisfacción al cliente, se logró analizar las causas que influían en la mala gestión de las personas que no asistían a sus labores correspondientes. El proyecto que realice queda abierto a mejoras debido a que existe un concepto denominado “mejora continua”, la mejora continua es un método empleado para mejorar la calidad, los procesos y servicios. Por lo que se recomienda implementar un proyecto basado en mejora continua partiendo de la ampliación del control de calidad. Mi estancia en el área de servicio de la empresa KUBIY SOFT fue una experiencia muy buena porque pude poner en práctica los conocimientos adquiridos durante mi formación como ingeniero Mecatrónico. No solo pude poner en práctica lo aprendido en el Instituto Tecnológico de Apizaco, si no también pude adquirir conocimiento práctico porque al supervisar los trabajos realizados por los técnicos aprendí a manipular herramientas y algunas máquinas, entre las herramientas que aprendí a manipular son la llave de la marca HONEYWELL que más adelante dejare una pequeña imagen, el uso de una máquina con sellado al vacío, patín hidráulico, la pistola neumática para soplado, el tornillo de banco, llaves, etc.

Sin olvidar mencionar que también al estar trabajando conjuntamente con los asesores de servicio, se puso en práctica el trabajo en equipo, ya que, si había alguna complejidad del sistema que no tenía una buena comunicación serial debido a la mala colocación de la API, mi función dentro de mis estadías era como apoyo, en caso de entrega de producción, yo entregaba prefecturas a los clientes de la siguiente manera (se crea la cabecera de AA a AC) mientras el asesor no estaba en recepción, para que el cliente pudiera hacer su pago sin estar esperando al asesor para recibir su prefectura, en caso de recepción de unidades yo apoyaba con: hacer los movimientos requeridos para una entrega correcta.

Se pudo realizar el diseño e implementación del sistema de control de asistencia por NFC. Se realizó el programa con éxito necesario para la lectura de información proporcionada por las TAG's RFID y el sistema NFC de tal forma que estas sean utilizadas por un sistema capaz de manipular estos programas para la gestión de control de personal. Se logró realizar el almacenaje de información, alzando un servidor en XAMPP de tal manera que estos datos son guardados dentro de una hoja de cálculo la cual funcionara como Base de datos.

Este proyecto facilitará a la gestión y control de flujo de personal, de manera que se tendrá mejor control acerca de los movimientos que realizará el personal de la empresa, este proyecto contribuirá a la reducción de congestión para el marcado de tarjeta puesto que al ser instantáneo disminuirá la espera para el marcado de asistencia, a su vez reducirá la complejidad a la hora de realizar el control de días y horas asistidas pues al estar en formato digital es más fácil buscar o definir parámetros para realizar el respectivo control.

Como recomendación, al ser este un prototipo aún se pueden añadir muchas más funciones para aumentar el nivel de seguridad, tanto a nivel de software como a nivel de hardware, como el manejo de la memoria ROM de la tarjeta la cual puede ser utilizada para almacenar información por el usuario o empresa y poder bloquearla de manera permanente para que no pueda ser regrabada. Una de las partes más complejas de realizar fue la comunicación serial de la tarjeta NFC a la dirección que se le asignó, este problema se solucionó de la siguiente manera investigando un poco más a fondo como direccionar correctamente la dirección IP al servidor que será consumido.

Competencias desarrolladas y/o aplicadas

Para poder elaborar mi proyecto de residencias se aplicaron los conocimientos adquiridos a lo largo de mi formación como ingeniero Mecatrónico. Las materias impartidas durante toda la carrera con más relevancia en la elaboración de mi proyecto son: Programación básica, Programación avanzada y Programación orientada a objetos. Estas asignaturas han aportado al perfil del ingeniero Mecatrónico con la gran capacidad de controlar el acceso de las personas mediante el uso de un sitio web. Esta asignatura tiene como competencia específica la aplicación de metodologías estadísticas para la identificación y la implementación de estrategias para optimizar los procesos productivos en la industria. De desarrollos tecnológicos. La asignatura de programación avanzada le aporta al perfil del ingeniero Mecatrónico la capacidad de aplicar conocimientos y habilidades generales de ingeniería en las áreas de mantenimiento, manufactura, producción y calidad para fomentar la competitividad en la industria de desarrollo en tecnología e innovación. La competencia a desarrollar en esta asignatura es aplicar los fundamentos de programación básica, diagramas de conexión electrónicos y eléctricos, análisis, operación y control y mantenimiento de los lectores RFID-NFC diagramas de flujo. Elementos electrónicos. La asignatura de electrónica digital le aporta al perfil del ingeniero Mecatrónica la capacidad para comprender el comportamiento de los diferentes elementos electrónicos que conforman las partes del sensor tales como, la tarjeta de memoria RAM, ROM, microcontrolador para aplicar estos conocimientos y habilidades en las áreas de diseño y mantenimiento principalmente. La competencia a desarrollar en esta asignatura es conocer la clasificación y el funcionamiento de los elementos electrónicos abordados en esta asignatura para la identificación de problemas relacionados con los diferentes sistemas de conexión tales como; interpretación de diagramas, esquemas eléctricos, simbología, etc. De acuerdo con las especificaciones de la industria.

BIBLIOGRAFIA

parzibyte. (29 NOV 2020). Sistema de registro de asistencia con tarjeta RFID . 22 Enero 2022, de Parzibyte's Recuperado de: <https://parzibyte.me/blog/2020/11/28/sistema-registro-asistencia-tarjetas-rfid/>

parzibyte. (29 NOV 2020). Lecturas recomendadas . 22 Enero 2022, de Parzibyte's Recuperado de: https://parzibyte.me/blog/2020/11/28/sistema-registro-asistencia-tarjetas-rfid/#Lecturas_recomendadas

parzibyte. (29 NOV 2022). Código del lector RFID. 22 Enero 2022, de Parzibyte's Recuperado de: [https://parzibyte.me/blog/2020/11/28/sistema-registro-asistencia-tarjetas-rfid/#Codigo del lector RFID](https://parzibyte.me/blog/2020/11/28/sistema-registro-asistencia-tarjetas-rfid/#Codigo_del_lector_RFID)

parzibyte. (29 NOV 2020). Base de Datos. 22 Enero 2022, de Parzibyte's blog Recuperado de: [https://parzibyte.me/blog/2020/11/28/sistema-registro-asistencia-tarjetas-rfid/#Base de datos](https://parzibyte.me/blog/2020/11/28/sistema-registro-asistencia-tarjetas-rfid/#Base_de_datos)

parzibyte. (29 NOV 2020). Leer RFID con NodeMCU ESP8266 y RC522. 29 Enero 2022, de parzibyte Recuperado de: <https://parzibyte.me/blog/2020/11/23/leer-rfid-nodemcu-esp8266-rc522/>

parzibyte. (29 NOV). Instalar librería . 29 Enero 2022, de parzibyte Recuperado de: [https://parzibyte.me/blog/2020/11/23/leer-rfid-nodemcu-esp8266-rc522/#Instalar libreria](https://parzibyte.me/blog/2020/11/23/leer-rfid-nodemcu-esp8266-rc522/#Instalar_libreria)