

## Indice

Introducción.....	3
Justificación.....	4
Objetivos.....	5
Caracterización del área en que se participó.....	6
Problemas a resolver.....	8
Alcances y limitaciones.....	8
CAPITULO 1	
Control On OFF, Microcontroladores y Actuadores.....	10
1.1 Sistema de Control Automáticos <sup>(1)</sup> .....	11
1.1.1 Sistemas de Lazo abierto <sup>(1)</sup> .....	12
1.1.2 Sistema de lazo cerrado <sup>(1)</sup> .....	12
1.1.3 Acción de control de dos posiciones o de encendido y apagado <sup>(2)</sup> .....	15
1.2 Actuadores <sup>(3)</sup> .....	15
1.2.3 Solenoide y Electroimán <sup>(3)</sup> .....	16
1.2.3.1 Funcionamiento e Instalación de un electroimán <sup>(3)</sup> .....	18
1.2.3.2 Fallas <sup>(3)</sup> .....	19
1.3 Microcontroladores <sup>(4)</sup> .....	20
1.4.1 Programación de Arduino <sup>(5)</sup> .....	25
1.4.2 Funciones básicas y operadores <sup>(5)</sup> .....	26
1.4.3 Estructuras de control Variables Constantes y Tipos de datos <sup>(5)</sup> .....	27
1.4.4 Bibliotecas en Arduino y creación de librerías <sup>(5)</sup> .....	29
1.4.4.1 Librería password, Librería de Cristal Líquido y Teclado Matricial <sup>(6)</sup> .....	31
1.4.5 Sensor fotoeléctrico y Botón de salida do not touch.....	32

## CAPÍTULO 2

Desarrollo del sistema de control de entrada.....	34
2.1 Diagrama de flujo.....	35
2.2 Procedimiento y descripción de las actividades realizadas.....	36
2.3 Realización del primer circuito impreso y diagrama de conexiones.....	38
2.4 Programación de la tarjeta Arduino.....	39

## CAPÍTULO 3

Pruebas y Resultados.....	54
---------------------------	----

## CAPÍTULO 4

Conclusiones.....	57
Bibliografía.....	59

## **Introducción**

Teléfonos de México, empresa dedicada al servicio de la telefonía, es una empresa internacional y de alto prestigio, por lo que todos sus procesos deben estar regulados.

Actualmente la matriz en la ciudad de Puebla, en el departamento de Centro de Operaciones y mantenimiento de larga distancia presenta el problema de que no se tiene un buen nivel de seguridad para registrar el acceso a dicho departamento.

Lo que quiere la empresa es poner un nivel de seguridad para tener un control fiable del ingreso de personas a dicho departamento. Por ello, Teléfonos de México propone el proyecto titulado “Control de acceso para áreas de alta seguridad” con lo cual piensa resolver el problema de inseguridad en el ingreso de personal no autorizado a dicho departamento.

En el presente trabajo se atiende el problema de la inseguridad a través de diseñar e implementar un control electrónico que identifique que personas ingresan al área de Centro de Operaciones y mantenimiento de la matriz de Teléfonos de México del estado de Puebla.

A continuación vamos a describir las actividades que se realizaron para llevar a cabo este trabajo, como parte de la residencia profesional. Para llevar a cabo dichas actividades, se organizaron por capítulos, los cuales se describen de manera breve a continuación.

Primeramente se está presentando el trabajo, es decir:

Introducción, Justificación, Objetivos, Alcances y Limitaciones entre otras.

En el Capítulo I se describe la teoría utilizada en la residencia profesional que trata sobre Control On OFF, Microcontroladores y Actuadores.

En el Capítulo II se plantean la propuesta para resolver el problema existente y a la vez se desarrolla la propuesta dada para resolver el problema.

En el Capítulo III presentamos las pruebas que se hicieron para comprobar la eficacia de la propuesta y los resultados obtenidos.

Finalmente en el Capítulo IV se presentan las conclusiones y recomendaciones

## **Justificación**

Telmex es una empresa de clase mundial y global. Como tal busca mejoras en sus diversos procesos como seguridad, atención al cliente, servicios de telefonía, Internet, entre otras.

En cuestiones de seguridad y en el Centro de Operaciones y Mantenimiento de Larga distancia, se adolece de esta, porque de acuerdo a las normas de calidad y seguridad de TELMEX, dicho centro debe de tener tres filtros o instancias de reten de personal.

Aunque existe un proyecto de seguridad para el centro de operaciones y mantenimiento de larga distancia, este sólo cuenta con dos filtros y es insuficiente, ya que se ha infiltrado personal ajeno a dicho centro. Por lo cual, Telmex, nos ha propuesto que los apoyemos con la realización del proyecto denominado “Control de acceso para áreas de alta seguridad”, con la intención de que se implementen los tres filtro.

El proyecto requiere que se realice un circuito electrónico, capaz de identificar contraseñas del personal autorizado. Si es la persona autorizada se abre una puerta, en caso contrario permanecerá cerrada la puerta. Además se debe hacer una aplicación Web, en donde se den de altas y bajas a los usuarios de dicho centro.

Como residente se participó en dicho proyecto y para la realización de la tarjeta se trabajará con un microcontrolador Arduino One, al cual, se le conectarán periféricos de entrada salida como Teclado Matricial, LCD, Relevador y otros. Además se trabajará con HTML para desarrollar la aplicación Web donde se dará de alta y baja a usuarios de dicho centro.

Como electrónico, este proyecto es relevante, no sólo por la aplicación de microcontroladores sino porque se va a resolver un problema real, además de desarrollar una página de alta y bajas de usuario.

## Objetivos

De lo anterior describiremos el objetivo principal del proyecto y luego pasaremos a los objetivos particulares

### Objetivo General

Diseñar, desarrollar e implantar un circuito electrónico que identifique contraseñas, y si la contraseña es la correcta abra una puerta de acceso y luego la cierre. Además desarrollar una aplicación Web para altas y bajas de usuarios del Centro de Operaciones y Mantenimiento de Larga Distancia de TELMEX zona centro.

### Objetivos Específicos

- Conocer el área en que se trabajar para poder diseñar la mejor solución al problema.
- Diseño del esquema eléctrico
- Desarrollo de la etapa de control
- Desarrollo de la etapa de acoplamiento
- Desarrollar un programa para la tarjeta arduino
- Elaboración de interfaz Web
- Instalación de un servidor centralizado
- Elaborar un manual técnico de conexiones y de uso

## **Caracterización del área en que se participó.**

TELÉFONOS DE MEXICO es una empresa líder a nivel mundial en área de las telecomunicaciones, en México ha mantenido una hegemonía bien constituida en todo México, actualmente cuenta con tecnología de punta en todo el país y es la principal empresa de telecomunicaciones acá en nuestro país. Cuenta con influencia en América Latina y Estados Unidos. Teléfonos de México ofrece servicios tales como telefonía móvil y fija, Internet, televisión por Internet, venta de equipos de comunicación.

Teléfonos de México como tal se creó con la privatización de la empresa federal de telefonía, y creció rápidamente auspiciada por el gobierno mexicano de ese entonces dirigido por el ex-presidente Carlos Salinas de Gortari, pero sus primeros pasos los dio cuando se unieron las compañías Ericsson y International Telephone and Telegraph Company en México.

A principios de los años 80 en México comenzaron a surgir algunas compañías que ofrecían servicios de telefonía celular. En 1993, Iusacell se había convertido en el líder tras comprar varios operadores regionales. Debido a que Telmex no tenía inversiones en este negocio, decidió entrar al mercado con la empresa Telcel que estaba en un lejano segundo lugar en el mercado nacional, ya que cuando Iusacell contaba con 3.000.000 de usuarios, Telcel tenía menos de 1.000.000.

Este escenario cambió en 1995, cuando México sufrió una de sus peores crisis económicas, ya que Iusacell decidió enfocarse en clientes de alto nivel (ejecutivos y empresas) con planes de renta mensual de alto costo, mientras que Telcel decidió enfocarse al sector de menor ingreso, impulsando los primeros planes de prepago, acaparando así un mayor número de clientes para convertirse en el líder de telefonía celular en el mercado mexicano.

Más tarde, la empresa de telefonía celular Telcel sería separada de la empresa Telmex para ofrecer un servicio por separado. En octubre del 2011, América Móvil, también controlada por el Grupo Carso, anuncia un proceso de fusión con Telmex, acción que se concreta en agosto del 2012.

A continuación presentamos la misión, visión y valores por los que se caracteriza o persigue Teléfonos de México.

**Misión.**

Ser un grupo líder de soluciones integrales de telecomunicaciones a nivel internacional, proporcionando a nuestros clientes servicios de gran valor, innovadores y de clase mundial, a través del desarrollo humano, y de la aplicación y administración de tecnología de punta.

**Visión.**

Consolidar el liderazgo de TELMEX en el mercado nacional, expandiendo su penetración de servicios de telecomunicaciones en todos los mercados posibles, para situarnos como una de las empresas de más rápido y mejor crecimiento a nivel mundial.

**Valores.**

Nuestros Valores apoyan nuestra Misión y sustentan tanto nuestros Principios Empresariales como nuestros Principios de Conducta.

Nuestros Valores son las cualidades que nos distinguen y nos orientan. Es necesario que nuestra labor cotidiana los tenga presentes siempre, y los lleve a la práctica.

Los Valores de nuestra cultura corporativa son:

- Trabajo
- Crecimiento
- Responsabilidad Social
- Austeridad

## **Problemas a resolver**

De acuerdo a la normativa interna el centro de Operaciones y Mantenimiento debe contar con tres filtros de entrada. Al momento solo existen dos.

Problemas de seguridad, como en muchas partes la seguridad viene siendo un tema fundamental en todo momento, equipo de esta área se ha extraviado por que no hay una manera de bloquear el paso a aquellas personas que sean ajenas a estas instalaciones.

## **Alcances y limitaciones**

### **Alcances**

Desarrollar el programa que sea capaz de validar Usuario y Contraseñas por medio de técnicas de identificación y autorización.

Contará con una memoria de alta capacidad para poder alojar hasta 100 000 Usuarios y Contraseñas.

Se instalará un botón inalámbrico que proporcionará la apertura de la puerta para accesos repentinos y salidas de emergencia.

El sistema debe ser escalable a todas las áreas de la empresa consideradas críticas exceptuando el área de atención al cliente.

### **Limitaciones**

La limitacion mas importante que hay en el trabajo es la falta de tiempo, como idea del proyecto se planeo diseñar y elaborar una interfaz web conectada a un servidor para que desde ahí se pudiera dar de alta y baja a los usuarios, y datos como graficas de asistencia, pero debido a la falta de tiempo no se pudo terminar esta parte del proyecto.



# **CAPITULO I**

## **Control On OFF, Microcontroladores y Actuadores**

En este capítulo se abordará la teoría que se necesitó para elaborar el proyecto, hablaremos de sistemas de control, centrandonos principalmente en los sistemas de control on off. Posteriormente pasaremos a hablar de los actuadores y finalizaremos describiendo la teoría concerniente a los microcontroladores

## 1.1 Sistema de Control Automáticos <sup>(1)</sup>

Cuando se piensa en un sistema de control se puede pensar en un sistema como en una caja negra que tiene una entrada y una salida. Se considera una caja negra debido a que en realidad no importa que tenga dentro, si no la relación entre la entrada y la salida. Este sistema es de control si la salida se controla de modo que pueda adoptar un valor o cambio en particular de alguna manera definida.

La palabra sistema se emplea para describir un conjunto de componentes que interactúan, alrededor de los cuales se dibuja una frontera imaginaria e modo que solo es de interés la interacción entre la entrada o las entradas y su salida o salidas, sin necesidad de estudiar a detalle la relación. Un sistema puede ser una estación de generación de energía o quizá un simple motor eléctrico. No importa que tan complejo sea un conjunto de componentes y sus interacciones dentro del sistema; se puede considerar que todos están dentro de una caja negra y solo tener en cuenta las entradas y salidas a dicha caja. La figura 1.1 muestra cómo es posible representar un sistema mediante una caja con las entradas y salidas del sistema indicadas como flechas, en las que la dirección de una flecha hace referencia ya sea a una entrada o a una salida



Figura 1.1 Sistema de caja negra

Existen dos clases comunes de sistemas de control, sistemas de lazo abierto y sistemas de lazo cerrado. En los sistemas de control de lazo abierto la salida se genera dependiendo de la entrada; mientras que en los sistemas de lazo cerrado la salida depende de las consideraciones y correcciones realizadas por la retroalimentación. Un sistema de lazo cerrado es llamado también sistema de control con realimentación. Los sistemas de control más modernos en ingeniería automatizan procesos en base a muchos parámetros y reciben el nombre de Controladores de Automatización Programables (PAC)

### **1.1.1 Sistemas de Lazo abierto <sup>(1)</sup>**

Los sistemas en los cuales la salida no tiene efecto sobre la acción de control se denominan sistemas de control en lazo abierto. En otras palabras, en un sistema de control en lazo abierto no se mide la salida ni se realimenta para compararla con la entrada. Un ejemplo práctico es una lavadora. El remojo, el lavado y el centrifugado en la lavadora operan con una base de tiempo. La máquina no mide la señal de salida, que es la limpieza de la ropa. En cualquier sistema de control en lazo abierto, la salida no se compara con la entrada de referencia. Así, a cada entrada de referencia le corresponde una condición de operación fija; como resultado de ello, la precisión del sistema depende de la calibración. Ante la presencia de perturbaciones, un sistema de control en lazo abierto no realiza la tarea deseada. En la práctica, el control en lazo abierto sólo se usa si se conoce la relación entre la entrada y la salida y si no hay perturbaciones internas ni externas. Es evidente que estos sistemas no son de control realimentado. Obsérvese que cualquier sistema de control que opere con una base de tiempo está en lazo abierto. Por ejemplo, el control de tráfico mediante señales operadas con una base de tiempo es otro ejemplo de control en lazo abierto.

### **1.1.2 Sistema de lazo cerrado <sup>(1)</sup>**

Un sistema que mantiene una relación determinada entre la salida y la entrada de referencia, comparándolas y usando la diferencia como medio de control, se denomina sistema de control de lazo cerrado o realimentado. Un ejemplo sería el sistema de control de temperatura de una habitación. Midiendo la

temperatura real y comparándola con la temperatura de referencia (temperatura deseada), el termostato activa o desactiva el equipo de calefacción o de enfriamiento para asegurar que la temperatura de la habitación se mantiene en un nivel confortable independientemente de las condiciones externas. Los sistemas de control realimentados no se limitan a la ingeniería, sino que también se encuentran en diversos campos ajenos a ella. Por ejemplo, el cuerpo humano es un sistema de control realimentado muy avanzado. Tanto la temperatura corporal como la presión sanguínea se conservan constantes mediante una realimentación fisiológica. De hecho, la realimentación realiza una función vital: hace que el cuerpo humano sea relativamente insensible a las perturbaciones externas, permitiendo que funcione de forma adecuada en un entorno cambiante. En la figura 1.2 se muestra los componentes de un sistema de lazo cerrado.

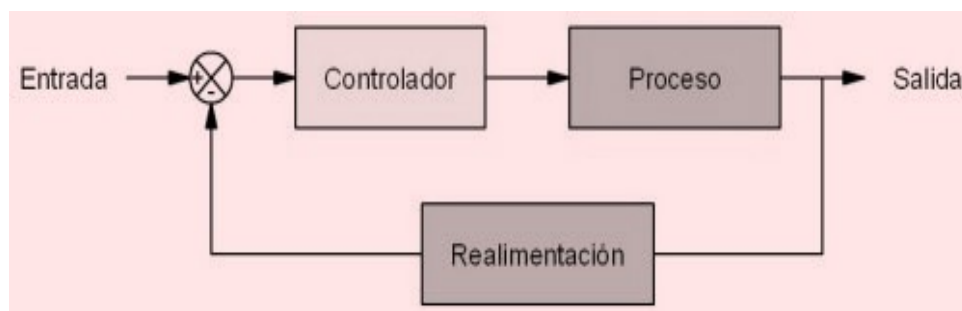


Figura 1.2 Sistema de Lazo cerrado

En la práctica, los términos control realimentado y control en lazo cerrado se usan indistintamente. En un sistema de control en lazo cerrado, se alimenta al controlador la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la propia señal de salida o una función de la señal de salida y sus derivadas y/o integrales), con el fin de reducir el error y llevar la salida del sistema a un valor deseado. El término control en lazo cerrado siempre implica el uso de una acción de control realimentado para reducir el error del sistema.

Una ventaja del sistema de control en lazo cerrado es que el uso de la realimentación vuelve la respuesta del sistema relativamente insensible a las perturbaciones externas y a las variaciones internas en los parámetros del

sistema. Es así posible usar componentes relativamente poco precisos y baratos para obtener el control adecuado de un proceso determinado, mientras que hacer eso es imposible en el caso de un sistema en lazo abierto. Desde el punto de vista de estabilidad, el sistema de control en lazo abierto es más fácil de desarrollar, porque la estabilidad del sistema no es un problema importante.

Por otra parte, la estabilidad es un gran problema en el sistema de control en lazo cerrado, que puede conducir a corregir en exceso errores que producen oscilaciones de amplitud constante o cambiante. Debe señalarse que, para los sistemas en los que se conocen con anticipación las entradas y en los cuales no hay perturbaciones, es aconsejable emplear un control en lazo abierto. Los sistemas de control en lazo cerrado sólo tienen ventajas cuando se presentan perturbaciones y/o variaciones impredecibles en los componentes del sistema. Obsérvese que la potencia nominal de salida determina en forma parcial el coste, peso y tamaño de un sistema de control. El número de componentes usados en un sistema de control en lazo cerrado es mayor que el que se emplea para un sistema de control equivalente en lazo abierto. Por tanto, el sistema de control en lazo cerrado suele tener costes y potencias más grandes. Para disminuir la potencia requerida de un sistema, se emplea un control en lazo abierto siempre que pueda aplicarse. Por lo general, una combinación adecuada de controles en lazo abierto y en lazo cerrado es menos costosa y ofrecerá un comportamiento satisfactorio del sistema global. En ciertas circunstancias (por ejemplo, si no hay perturbaciones o la salida es difícil de medir) pueden ser deseables los sistemas de control en lazo abierto.

Podemos apreciar en la figura 1.2 mostrada anteriormente que el Controlador o control es un elemento del sistema de lazo cerrado que tiene como entrada la señal de error y produce una salida que se convierte en la entrada del elemento correctivo o sistema. Los tipos de control que se puede emplear son los siguientes

1. De dos posiciones o controladores on-off
2. Controladores proporcionales
3. Controladores integrales
4. Controladores proporcionales-integrales

5. Controladores proporcionales-derivativos
6. Controladores proporcionales-integrales-derivativos

El control que al momento nos interesa es el control on off, debido a que fue el tipo de control utilizado en este proyecto, por lo cual el único tipo de control que vamos a definir es este.

### **1.1.3 Acción de control de dos posiciones o de encendido y apagado <sup>(2)</sup>**

En un sistema de control de dos posiciones, el elemento de actuación sólo tiene dos posiciones fijas, que, en muchos casos, son simplemente encendido y apagado. El control de dos posiciones o de encendido y apagado es relativamente simple y barato, razón por la cual su uso es extendido en sistemas de control tanto industriales como domésticos. Supóngase que la señal de salida del controlador es  $u(t)$  y que la señal de error es  $e(t)$ . En el control de dos posiciones, la señal  $u(t)$  permanece en un valor ya sea máximo o mínimo, dependiendo de si la señal de error es positiva o negativa. De este modo,

$$u(t) = \begin{cases} U_1, & \text{para } e(t) \geq 0 \\ U_2, & \text{para } e(t) < 0 \end{cases}$$

donde  $U_1$  y  $U_2$  son constantes. Por lo general, el valor mínimo de  $U_2$  es cero o  $-U_1$ . Es común que los controladores de dos posiciones sean dispositivos eléctricos, en cuyo caso se usa extensamente una válvula eléctrica operada por solenoides. Los controladores neumáticos proporcionales con ganancias muy altas funcionan como controladores de dos posiciones.

### **1.2 Actuadores <sup>(3)</sup>**

Un actuador es un dispositivo inherentemente mecánico cuya función es proporcionar fuerza para mover o “actuar” otro dispositivo mecánico. La fuerza que provoca el actuador proviene de tres fuentes posibles: Presión neumática, presión hidráulica, y fuerza motriz eléctrica (motor eléctrico o solenoide). Dependiendo de el origen de la fuerza el actuador se denomina “neumático”, “hidráulico” o “eléctrico”

### **1.2.1 Actuadores eléctricos <sup>(3)</sup>**

Los actuadores eléctricos son los mas extendidos y los que poseen un mayor campo de aplicación dada la facilidad de disponer energía eléctrica a través de las redes de distribución. Además son altamente versátiles debido a que utilizan cables eléctricos para transmitir señales de control y la electricidad por lo que practicamente no hay restricciones para la distancia entre el controlador y el actuador. De hecho hay actuadores que dependen de una etapa previa de accionamiento eléctrico, como son los neumáticos y los hidráulicos.

Los tipos de controladores eléctricos se clasifican de la siguiente manera.

1. Motores de corriente alterna
2. Motores de corriente continua
3. Motores a paso
4. Servomotores
5. Motor universal
6. Solenoide

El actuador que en proyecto utilizamos fue un electroimán y este tipo de actuadores cae en la clasificación de los solenoides.

### **1.2.3 Solenoide y Electroimán <sup>(3)</sup>**

Un solenoide es un alambre conductor devanado en forma de bobina helicoidal apretada y de muchas vueltas. Una corriente en este alambre produce un fuerte campo magnético dentro de la bobina como se muestra en la figura 1.3.

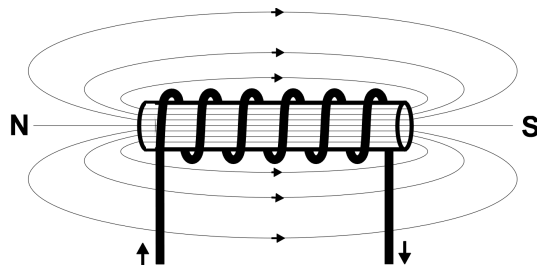


Figura 1.3 Solenoide

Un solenoide ideal es una solenoide muy largo (infinito) con espiras devanadas muy apretadamente para que la distribución de la corriente en la superficie sea uniforme. Cuando las vueltas están muy juntas entre si, se puede considerar como una vuelta circular y el campo magnético neto será la suma de los campos debido a todas las vueltas. El campo magnético de un solenoide es uniforme en su interior y nulo en su exterior. Un solenoide con un núcleo apropiado se convierte en un electroimán (un imán que produce su campo magnético mediante una corriente eléctrica).

Un electroimán (figura 1.4) es un tipo de imán En el que el campo magnético se produce mediante el flujo de una corriente eléctrica, desapareciendo en cuanto dicha corriente desaparece.

En 1819, el físico danés Hans Christian Orsted descubrió que una corriente eléctrica que circula por un conductor produce un efecto magnético que puede ser detectado con la ayuda de una brújula. Y en 1825 Joseph Henry inventó el electroimán en 1825.

El primer electroimán era un trozo de hierro con forma de herradura envuelto por una bobina enrollada sobre él. Henry envolvió los cables por los que hizo circular la corriente de una batería. Henry podía regular su electroimán, lo que supuso el principio del uso de la energía eléctrica en máquinas útiles y controlables, estableciendo los cimientos para las comunicaciones electrónicas a gran escala.





FIGURA 1.4 Electroimán

Pueden producirse campos magnéticos mucho más fuertes si se sitúa un núcleo de material paramagnético o ferromagnético (normalmente hierro dulce o ferrita, aunque también se utiliza el llamado acero eléctrico) dentro de la bobina. El núcleo concentra el campo magnético, que puede entonces ser mucho más fuerte que el de la propia bobina.

Los campos magnéticos generados por bobinas se orientan según la regla de la mano derecha. Además, dentro de la bobina se crean corrientes inducidas cuando ésta está sometida a un flujo variable. Estas corrientes son llamadas corrientes de Foucault y en general son indeseables, puesto que calientan el núcleo y provocan una pérdida de potencia de sí mismo.

#### **1.2.3.1 Funcionamiento e Instalación de un electroimán <sup>(3)</sup>**

El material del núcleo del imán (generalmente hierro) se compone de pequeñas regiones llamadas dominios magnéticos que actúan como pequeños imanes.

Antes que la corriente en el electroimán este activada, los dominios en el núcleo de hierro están en direcciones al azar, por lo que sus campos magnéticos pequeños se anulan entre sí, el hierro aún no tiene un campo magnético de gran escala. Cuando una corriente pasa a través del alambre envuelto alrededor de la plancha, su campo magnético penetra en el hierro, y hace que los dominios giren, alineándose en paralelo al campo magnético, por lo que sus campos magnéticos diminutos se añaden al campo del alambre, creando un campo magnético que se extiende en el espacio alrededor del imán. Cuanto mayor es la corriente que pasa a través de la bobina de alambre, más dominios son alineados, aumentando la intensidad del campo magnético.

Finalmente, todos los dominios estarán alineados, nuevos aumentos en la corriente sólo causan ligeros aumentos en el campo magnético: este fenómeno se denomina saturación. Cuando la corriente en la bobina está desactivada, la mayoría de los dominios pierden la alineación y vuelven a un estado aleatorio y así desaparece el campo. Sin embargo en algunos la alineación persiste, ya que los dominios tienen dificultades para perder su dirección de magnetización, dejando en el núcleo un imán permanente débil. Este fenómeno se llama histéresis y el campo magnético restante se llama magnetismo remanente. La magnetización residual del núcleo se puede eliminar por desmagnetización.

Las cerraduras magnéticas, requieren para su operación una alimentación de 12 V DC (11.5 a 14V) y una corriente entre 250 a 550 mA dependiendo de la referencia. Por tanto, la fuente que se utilice debe poder suministrar la corriente que consume el electroimán sin que se reduzca apreciablemente el voltaje. Una disminución del voltaje significa también una reducción proporcional en la fuerza de cierre.

Es necesario conectar los elementos siguiendo la polaridad indicada, pues de otra forma pueden resultar daños en los componentes y estos daños no están cubiertos por la garantía. Así mismo, siempre debe usarse el circuito anti remanente, el cual viene en una pequeña caja plástica negra, la cual siempre debe usarse el circuito anti remanente, el cual viene en una pequeña caja plástica negra, la cual siempre se suministra con el electroimán.

La conmutación del electroimán (por medio de pulsadores o relevos) siempre debe hacerse antes del circuito anti remanente para que este pueda operar correctamente.

Si la fuente de poder está distante de la cerradura habrá caída de voltaje y la fuerza del electroimán se verá reducida proporcionalmente a esta caída. Es importante que el instalador esté capacitado para calcular el calibre apropiado para evitar caídas notorias de voltaje. Para calcular la resistencia del cable, debe conocer la distancia de la fuente de poder a la cerradura y el calibre del cable. La siguiente escala muestra resistencia del cable para cada 100 metros.

### 1.2.3.2 Fallas <sup>(3)</sup>

Existen diversas causas por las que la instalación puede fallar y es necesario tener los conocimientos y la base para saber por donde empezar a atacar el problema y cuales son las mejores táctica para resolver el problema

La reducción en la fuerza de atracción se manifiesta con el hecho de que un pequeño empujón libera el electroimán de la placa. Para esto es necesario revisar las superficies del electroimán y la placa para asegurarse que no haya obstrucciones que impidan un perfecto alineamiento. Si el electroimán y la placa están correctos es muy probable que el problema esté en exceso de rigidez en el montaje de la placa. Literalmente la placa debe flotar sobre el anillo de caucho que da contra ella alrededor del tornillo central de montaje. En esa forma la atracción del electroimán propicia un correcto alineamiento. Para solucionar este problema libere un poco la presión en el tornillo central de montaje buscando que la placa flote libremente. Cabe también la posibilidad de que el voltaje esté demasiado bajo. Si sospecha esto último utilice el medidor correspondiente y verifique. En la figura 1.5 se muestra un esquema de conexión.

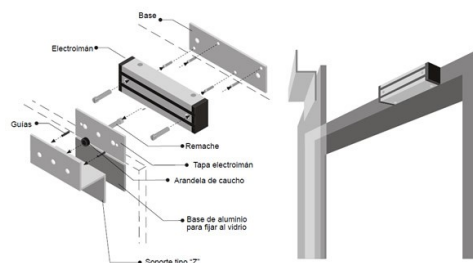


Figura 1.5 Colocación de un electroimán

### 1.3 Microcontroladores <sup>(4)</sup>

Un microcontrolador es un circuito integrado programable que contiene todos los componentes de una computador. Se emplea para controlar el funcionamiento de una tearea determinada y debido a su reducido tamaño, suele ir incorporado en el propio dispositivo al que gobierna. Esta última característica es la que le confiere la denominación de controlador incrustado (embedded cotroller)

El microcontrolador es un computador dedicado. En su memoria solo se reside un programa destinado a gobernar una aplicación determinada; sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles tienen como única finalidad atender sus requerimientos. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada. En pocas palabras podríamos definir un microcontrolador como

***Un microcontrolador es un computador completo, aunque de limitadas prestaciones, que está contenido en el chip de un circuito integrado y se destina a gobernar una sola tarea***

El número de productos que funcionan en base a uno o varios microcontroladores está aumentando de forma exponencial. No es aventurado decir que este siglo existen muy pocos dispositivos que funcionan sin un microcontrolador.

### **1.3.1 Arquitectura interna de un microcontrolador <sup>(4)</sup>**

Como ya se mencionó antes un microcontrolador posee todos los componentes de un computador, pero con unas características físicas que no pueden alterarse. Las partes principales de un microcontrolador son:

1. Procesador
2. Memoria no volátil para contener el programa
3. Memoria de lectura y escritura para guardar datos
4. Líneas de E/S para los controles periféricos
  - a) Comunicación en paralelo
  - b) Comunicación serie
  - c) Diversas puertas de comunicación (bus, I2C, USB etc.)
5. Recursos auxiliares
  - a) Circuitos de reloj
  - b) Temporizadores
  - c) Perro Guardian (Watchdog)

- d) Conversores AD y DA
- e) Comparadores analógicos
- f) Protección antes fallas de alimentación
- g) Estado de reposo en bajo consumo

Al día de hoy existe una gran variedad de fabricantes de microcontroladores, de lo cuales los mas importantes son los siguientes

1. Atmel
2. Freescale
3. Intel Corporation
4. National Semiconductor
5. Microchip Technology Inc.
6. Philips
7. Texas Instruments
8. Zilog
9. Picaxe

Muchas tarjetas electrónicas estan basadas en el uso de alguno de esto microcontroladores, tal es el caso de la tarjeta Arduino muy popular en estos momentos, su gran aceptación hoy en dia es debida a muchos factores, entre los cuales encontramos, su arquitectura y diseño Open Source, su gran cantidad de librerias disponibles y una comunidad de usuario creciente dia a dia. En el siguiente apartado hablaremos de la tarjeta Arduino, como se programa y su historia con mas detalle.

#### **1.4 Tarjeta Electrónica Arduino <sup>(5)</sup>**

Arduino es una tarjeta electrónica o plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios (figura 1.4).



Figura1.4 Arduino

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entradas/salidas. Los microcontroladores más usados son el Atmega128, Atmega328, Atmega1280, y Atmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa. Se programa en el ordenador para que la placa controle los componentes electrónicos.

Desde octubre de 2012, Arduino se utiliza también con microcontroladoras CortexM3 de ARM de 32 bits, que coexistirán con las más limitadas, pero también económicas AVR de 8 bits. ARM y AVR no son plataformas compatibles a nivel binario, pero se pueden programar con el mismo IDE de Arduino y hacerse programas que compilen sin cambios en las dos plataformas. Eso sí, las microcontroladoras CortexM3 usan 3,3V, a diferencia de la mayoría de las placas con AVR, que generalmente usan 5V. Sin embargo, ya anteriormente se lanzaron placas Arduino con Atmel AVR a 3,3V como la Arduino Fio y existen compatibles de Arduino Nano y Pro como Meduino en que se puede conmutar el voltaje.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash Processing Max/MSP Pure Data. Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino

(basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador.

También cuenta con su propio software que se puede descargar de su página oficial que ya incluye los drivers de todas las tarjetas disponibles lo que hace más fácil la carga de códigos desde el computador.

Arduino se inició en el año 2005 como un proyecto para estudiantes en el Instituto IVREA, en Ivrea (Italia). Posteriormente, Google colaboró en el desarrollo del Kit Android ADK (Accessory Development Kit), una placa Arduino capaz de comunicarse directamente con teléfonos celulares inteligentes bajo el sistema operativo Android para que el teléfono controle luces, motores y sensores conectados de Arduino. Para la producción en serie de la primera versión se tomó en cuenta que fuera multiplataforma. Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El modelo mas representativo a la fecha de hoy es el modelo conocido como Diecimila, los demás son una versión menor de este.

El modelo consta Diecimila de 14 entradas digitales configurables entrada y/o salidas que operan a 5 voltios. Cada contacto puede proporcionar o recibir como máximo 40 mA. Los contactos 3, 5, 6, 8, 10 y 11 pueden proporcionar una salida PWM (Pulse Width Modulation). Consta también de comunicación USB pero no soportaba como forma nativa del procesador. También tiene 6 entradas analógicas que proporcionan una resolución de 10 bits. Por defecto, aceptan de 0 voltios hasta 5 voltios, aunque es posible cambiar el nivel más alto, utilizando el contacto Aref y algún código de bajo nivel.

Las especificaciones para este modelo se muestran en la tabla 1.1

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB (ATmega2560) of which 4KB used by bootloader
SRAM	8 KB (ATmega328)
EEPROM	4 KB (ATmega328)
Clock Speed	16 MHz

Tabla 1.1 Especificaciones Arduino

A continuación se presentará la teoría de programación en la tarjeta Arduino, y se toma por hecho que el lector está familiarizado con programación básica, no siendo así es muy probable que al lector le cueste trabajo entender en cada parte de lo que se está hablando. Pero dejamos al lector la libre decisión de continuar con su lectura, al menos hasta que termine este apartado.

#### 1.4.1 Programación de Arduino <sup>(5)</sup>

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel Processing. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino, debido a que Arduino usa la transmisión serial de datos soportada por la mayoría de los lenguajes. Para los que no soportan el formato serie de forma nativa, es posible utilizar software intermediario que traduzca los mensajes enviados por ambas partes para permitir una comunicación fluida.

Algunos ejemplos son:

- 3DVIA Virtools: aplicaciones interactivas y de tiempo real.
- Adobe Director
- BlitzMax (con acceso restringido)
- C



- C++ (mediante libSerial o en Windows)
- C#
- Cocoa/Objective-C (para Mac OS X)
- Flash (mediante ActionScript)
- Gambas
- Isadora (Interactividad audiovisual en tiempo real)
- Instant Reality(X3D)
- Java
- Liberlab (software de medición y experimentación)
- Mathematica
- Matlab
- MaxMSP: Entorno gráfico de programación para aplicaciones musicales, de audio y multimedia
- Minibloq: Entorno gráfico de programación, corre también en las computadoras OLPC
- Perl
- Php
- Physical Etoys: Entorno gráfico de programación usado para proyectos de robótica educativa
- Processing
- Pure Data
- Python
- Ruby
- Scratch for Arduino (S4A): Entorno gráfico de programación, modificación del entorno para niños Scratch, del MIT)

- Squeak: Implementación libre de Smalltalk
- SuperCollider: Síntesis de audio en tiempo real
- VBScript
- Visual Basic .NET
- VVVV: Síntesis de vídeo en tiempo real

#### 1.4.2 Funciones básicas y operadores <sup>(5)</sup>

Arduino está basado en C y soporta todas las funciones del estándar C y algunas de C++. La programación de Arduino puede ser dividida en tres partes principales: estructura, variables (constantes y variables) y finalmente funciones. A continuación se presenta una lista con los operadores conocidos como ***Operadores de sintaxis adicionales***.

- Delimitadores: `;`, `{}`
- Comentarios: `//`, `/* */`
- Cabeceras: `#define`, `#include`
- Operadores aritméticos: `+`, `-`, `*`, `/`, `%`
- Asignación: `=`
- Operadores de comparación: `==`, `!=`, `<`, `>`, `<=`, `>=`
- Operadores Booleanos: `&&`, `||`, `!`
- Operadores de acceso a punteros: `*`, `&`
- Operadores de bits: `&`, `|`, `^`, `~`, `<<`, `>>`
- Operadores compuestos:
- Incremento y decremento de variables: `++`, `--`
- Asignación y operación: `+=`, `-=`, `*=`, `/=`, `&=`, `|=`

### 1.4.3 Estructuras de control Variables Constantes y Tipos de datos <sup>(5)</sup>

Se conoce como estructuras de control a aquellos segmentos de código que tienen la capacidad de cambiar el flujo del programa dependiendo el valor de las variables de entorno o contexto. A continuación presentamos nuevamente una lista con operadores de control.

- Condicionales: if, if...else, switch case
- Bucles: for, while, do... while
- Bifurcaciones y saltos: break, continue, return, goto

En cuanto al tratamiento de las variables también comparte un gran parecido con el lenguaje C.

HIGH/LOW: representan los niveles alto y bajo de las señales de entrada y salida. Los niveles de voltaje que se manejan son los mismos niveles que en los TTL.

Fase : Señal que representa al cero lógico. A diferencia de las señales HIGH/LOW, su nombre se escribe en letra minúscula.

True (verdadero): Señal cuya definición es más amplia que la de *false*. Cualquier número entero diferente de cero es "verdadero", según el álgebra de Booleana, como en el caso de -200, -1 o 1. Si es cero, es "falso".

El interprete de arduino reconoce como tipos de datos los siguientes: void, boolean, char, unsigned char, byte, int, unsigned int, word, long, unsigned long, float, double, string, array.

Además existen funciones que reciben como argumento una variable de cualquier tipo y devuelven una variable convertida en el tipo deseado: char(), byte(), int(), word(), long(), float()

Arduino es una tarjeta desarrollada principalmente para la interacción física con el entorno, por lo tanto cuenta con funciones que pueden activar directamente elementos físicos tales como actuadores

Dependiendo del tipo de dato se pueden clasificar en salidas digitales o analógicas.

## E/S DIGITAL

- pinMode(pin, modo)
- digitalWrite(pin, valor)
- int digitalRead(pin)

## E/S ANALÓGICA

- analogReference(tipo)
- int analogRead(pin)
- analogWrite(pin, valor)

## E/S AVANZADA

- shiftOut(dataPin, clockPin, bitOrder, valor)
- unsigned long pulseIn(pin, valor)

### 1.4.4 Bibliotecas en Arduino y creación de librerías <sup>(5)</sup>

Al igual que cualquier otro lenguaje de programación Arduino nos ofrece diversas librerías estándar para realizar nuestros proyectos que son las siguientes.

- EEPROM - Para leer y escribir en memorias "permanentes".
- Ethernet - Para conectar a Internet usando el *Ethernet Shield*
- Firmata - Para comunicarse con aplicaciones en la computadora usando un protocolo estándar *Serial*.
- LiquidCrystal - Para controlar Display de cristal líquido (*LCD*)
- Servo - Para controlar *servomotores*
- Software Serial - Para la comunicación serial de cualquier pin digital.

- Stepper - Para controlar motores paso a paso (*Stepper motors*)
- Wire - Interfaz de dos cables, o *Two Wire Interface (TWI/I2C)*, para enviar y recibir datos a través de una red de dispositivos y sensores.

Al ser arduino un software libre existen también las llamadas librerías contribuidas, con librerías abiertas donde los distintos programadores que trabajan con Arduino desarrollan y mejoran y que cualquiera puede usar. La siguiente es una lista de las librerías contribuidas.

Los usuarios de Arduino tienen la posibilidad de escribir sus propias bibliotecas. Ello permite disponer de código que puede reutilizarse en otros proyectos, mantener el código fuente principal separado de las bibliotecas y la organización de los programas construidos es más clara.

- Messenger - Para procesar mensajes de texto desde la computadora.
- NewSoftSerial - Versión mejorada de la librería *SoftwareSerial*.
- OneWire - Controla dispositivos (de *Dallas Semiconductor*) que usan el protocolo *One Wire*.
- PS2Keyboard - Lee caracteres de un teclado *PS2*.
- Simple Message System - Envía mensajes entre Arduino y la computadora.
- SSerial2Mobile - Envía mensajes de texto o emails usando un teléfono móvil (via comandos AT a través de software serial)
- Webduino - Librería de web server extendible (para usar con *Arduino Ethernet Shield*)
- X10 - Para enviar señales X10 a través de líneas de corriente AC.
- XBee - Para comunicaciones entre *XBees* en modo *API*.
- SerialControl - Para controlar remotamente otras Arduino a través de una conexión serial.

Ademas de estas existen muchísimas mas bibliotecas que se catalogan en el mundo de arduino como bibliotecas especiales.

## Sensores

- Capacitive Sensing - Convertir dos o más pins en sensores capacitivos.
- Debounce - Para lectura de inputs digitales con ruido (por ejemplo, botones).

## Displays y LEDs:

- Improved LCD library Arregla bugs de inicialización de LCD de la librería LCD oficial de Arduino.
- GLCD - Gráfica rutinas para *LCDs* basados en el chipset *KS0108* o equivalentes.
- LedControl - Para controlar matrices de LEDs o displays de siete segmentos con *MAX7221* o *MAX7219*.
- LedControl - Alternativa a la librería *Matrix* para controlar múltiples LEDs con chips *Maxim*.
- LedDisplay - Control para marquesina de LED *HCMS-29xx*.

## Generación de Frecuencias y Audio:

- Tone - Genera frecuencias de audio de onda cuadrada en el *background* de cualquier pin de un microcontrolador.

## Motores y PWM:

- TLC5940 - Controlador de *PWM* de 16 canales y 12 *bits*.

## Medición de Tiempo:

- DateTime - Librería para llevar registro de fecha y hora actual en el software.
- Metro - Útil para cronometrar acciones en intervalos regulares.
- MsTimer2 - Utiliza *timer 2 interrupt* para disparar una acción cada *N* mili segundos.

## Utilidades:

- TextString, también conocido como String - Maneja *strings*.

- PString - Liviana clase para imprimir en búfer.
- Streaming - Método para simplificar declaraciones de impresión.

#### **1.4.4.1 Librería password, Librería de Cristal Líquido y Teclado Matricial <sup>(6)</sup>**

La primera librería de la cual hablaremos sera la de password. De la pagina oficial de Arduino encontramos la siguiente descripción para la librería.

**“It is created to help simply the handling of passwords”**

Las funciones que podemos utilizar son las siguientes.

- Password password = Password(“Your password”) Crea la contraseña
- Is(password) Compara la contraseña
- Append(character) Agrega un carácter a la contraseña actual
- Reset() Reinicia la contraseña actual
- Evaluate() Evalúa la contraseña actual

La librería de cristal líquido nos permite desde una tarjeta de Arduino controlar un display LCD basado en HITACHI HD44780 o que tenga un compatible chipset, esta librería puede trabajar ya sea en modo de 4 bits u o 8 bits.

Las funciones que esta librería nos permite son las siguientes:

- LiquidCrystal() Crea una variable de tipo cristal liquido.
- Begin() Inicializa el LCD.
- Clear() Limpia el LCD
- home() Establece el cursos del LCD en las coordenadas iniciales
- setCursor () Se le pasa como parámetro las coordenadas a donde se quiere que se situé el cursor del LCD
- write() Escribe en el LCD

Pasemos a hablar de la librería del teclado matricial. El nombre original de esta librería es KeyPad y es una librería usada para el control de teclados

matriciales, en el código de esta librería viene previsto los rebotes que existen por el hecho de teclear y no es necesario utilizar un circuito anti rebote.

#### **1.4.5 Sensor fotoeléctrico y Botón de salida do not touch**

Este tipo de botones se pueden basar en cualquier de los dos siguientes principios para su funcionamiento. El primero de ellos utiliza un sensor fotoeléctrico y al momento que se interrumpe el paso de la luz en sensor se activa mandando un uno lógico como respuesta. El segundo principio en que se basan los botones de salida do not touch es la radiación infrarroja los conocidos como sensores pirolíticos. De ambos sensores, sensores de luz y sensores pirolíticos nos ocuparemos a continuación

Un sensor fotoeléctrico o foto-célula es un dispositivo electrónico que responde al cambio en la intensidad de la luz. Estos sensores requieren de un componente emisor que genera la luz, y un componente receptor que percibe la luz generada por el emisor. Todos los diferentes modos de censado se basan en este principio de funcionamiento. Están diseñados especialmente para la detección, clasificación y posicionado de objetos; la detección de formas, colores y diferencias de superficie, incluso bajo condiciones ambientales extremas.

Los sensores de luz se usan para detectar el nivel de luz y producir una señal de salida representativa respecto a la cantidad de luz detectada. Un sensor de luz incluye un transductor fotoeléctrico para convertir la luz a una señal eléctrica y puede incluir electrónica para condicionamiento de la señal, compensación y formateo de la señal de salida.

El sensor de luz más común es el LDR -Light Dependant Resistor o Resistor dependiente de la luz-.Un LDR es básicamente un resistor que cambia su resistencia cuando cambia la intensidad de la luz. Existen tres tipos de sensores fotoeléctricos, los sensores por barrera de luz, reflexión sobre espejo o reflexión sobre objetos.



# **CAPÍTULO II Desarrollo del sistema de control de entrada**

En el siguiente capítulo se describirá el desarrollo del trabajo, se empezará describiendo la razón por la cual se optó por dicha solución, y como se fue planteado la misma. Se desarrollará la solución en un diagrama de flujo, se presentará y analizará el circuito electrónico utilizado y también se describirá con detalle cada parte del programa codificado.

## 2.1 Diagrama de flujo

El primer paso para la realización de este proyecto fue la elaboración de un diagrama de flujo, el cual se dividirá en dos partes para una mejor comprensión, en la figura 2.1 mostramos el primero mostramos la inicialización del proyecto con la creación de una base de datos.

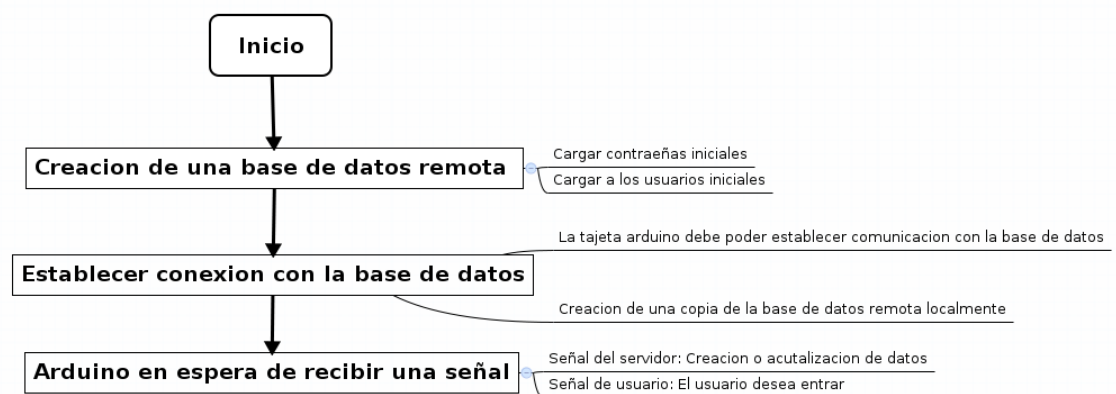


Figura 2.1 Diagrama de flujo

Como primer paso se creó una base de datos local, en donde se almacenan los datos de los usuarios, sus contraseñas y su número de usuario. Esta parte es muy importante ya que una pronta respuesta del sistema depende de una buena gestión de la información almacenada en la base de datos. Una vez que se tenga la base de datos creada y configurada correctamente pasamos a la comunicación de arduino con la misma, para que a través de la tarjeta arduino el usuario pueda ingresar sus credenciales de entrada.

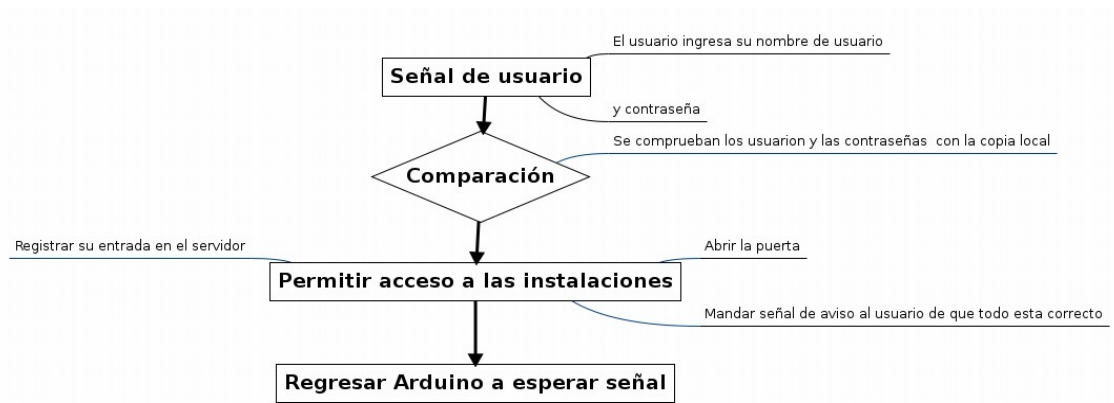


Figura 2.2 Diagrama de flujo

En la Figura 2.2 mostramos el segundo diagrama de flujo, una vez que se ha establecido comunicación entre arduino y la base de datos, lo siguiente que se debe de hacer es esperar a que el usuario ingrese su código de usuario y su código de contraseña y para cuando lo haga se debe comparar para ver si existe en la base de datos los registros que el usuario ingreso, de ahí se toma la decisión de permitir o negar la entrada a las instalaciones a la persona que ingreso los datos.

## 2.2 Procedimiento y descripción de las actividades realizadas.

Para resolver el problema se analizaron diferentes dispositivos que pudieran ser usados para resolver este problema.

En la figura 2.3 se exponen los dispositivos propuestos con sus ventajas y desventajas.

Tomando estas consideraciones se optó por utilizar la tarjeta programable “ARDUINO” por sus ventajas y pocas desventajas ya dichas, también es necesario mencionar que los asesores de TELMEX son los que escogieron el dispositivo a usar.

Para el primer desarrollo del proyecto se utilizaron diversas librerías que Arduino provee, las librerías que se utilizaron fueron las siguientes:

- LCD
- Password

- Teclado matricial

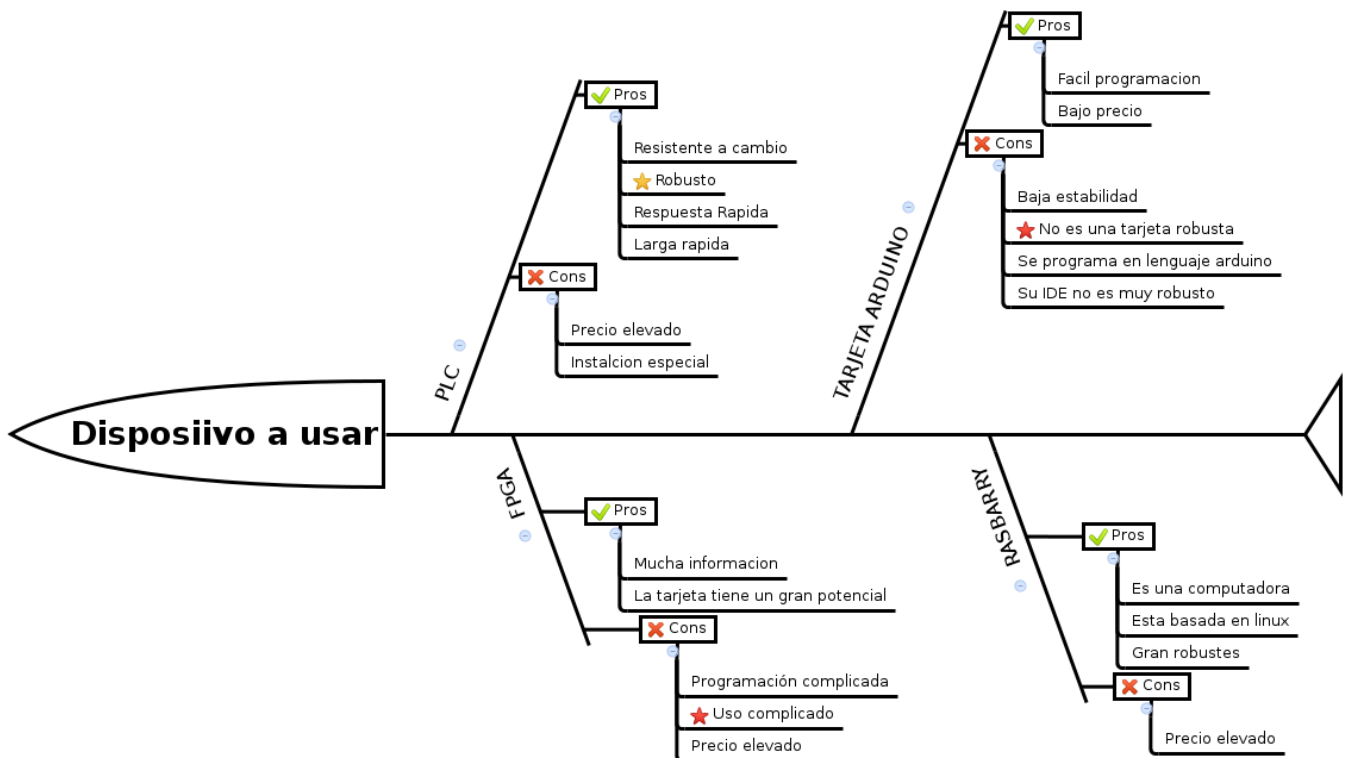


Figura 2.3 Dispositivos Propuestos

El primer prototipo fue hecho con la librería password las contraseñas se escribían directamente en el código del problema, nuestro principal objetivo con este primera aproximación era probar la instalación del circuito junto con el electroimán y e botón no touch, por lo cual escribir las contraseñas dentro del programa no representaba una seria limitación.

Los resultados de nuestra primera puesta en marcha fueron los siguientes. El primer programa estaba basado principalmente en el uso constante de la librería password y bien pronto nos dimos cuenta que esta librería es bastante deficiente.

El principal error de la librería es que cada contraseña crearse como un objeto de la clase o sea se tenia que crear una instancia del objeto para cada contraseña. Esto no presentaba ningún problema cuando las contraseñas eran solo unas cuantas a manejar, pero bien pronto nos dimos cuenta que al intentar crear mas de un limite de treinta contraseñas el programa se colgaba y dejaba

de responder, se tenía que ir destruyendo una por una las instancias de la clase después de haberles utilizado para compararlas y esta podría ser una posible solución a este problema, pero bien pronto surgió otro problema, no había manera de relacionar contraseñas con usuarios. La librería solo soporta el manejo de contraseñas pero no de usuarios.

Por lo anterior se optó por prescindir de usar esta librería.

### 2.3 Realización del primer circuito impreso y diagrama de conexiones

El sistema de conexiones es en sí algo bastante sencillo sin mucha complejidad, los dispositivos a conectar fueron una fuente de alimentación el electroimán un zumbador el botón no touch y todo esto a la placa arduino. Se diseñó una placa para que esta se montara justo encima de los pines de la tarjeta Arduino. El diseño se realizó con Eagle como se mencionó anteriormente.

El diseño del circuito impreso se muestra en la siguiente figura 2.4 y la placa hecha en la figura 2.5.

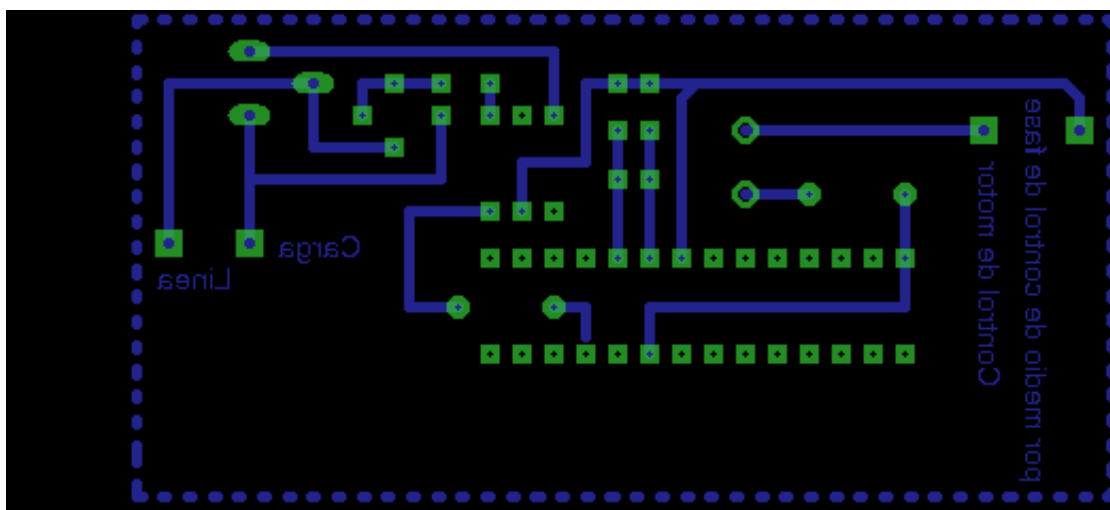


Figura 2.4 Diseño en Eagle de circuito PCB

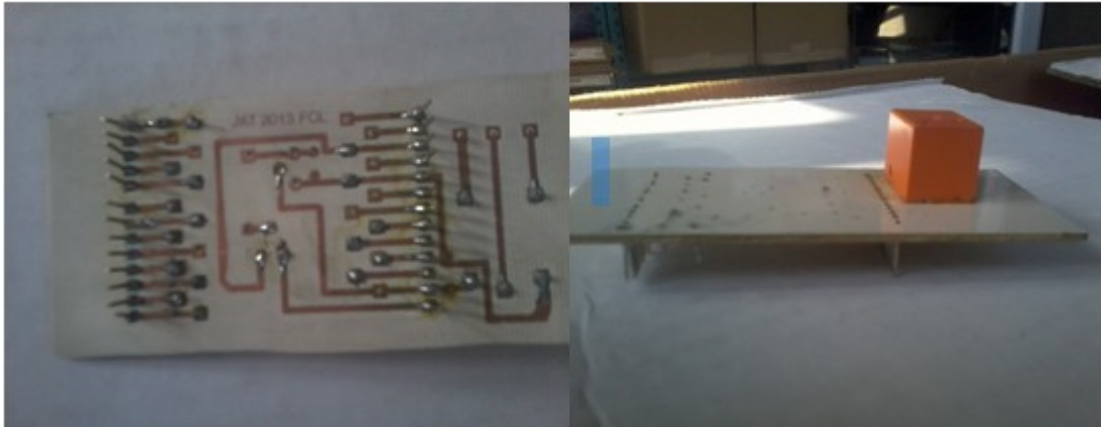


Figura 2.5 Circuito PCB

## 2.4 Programación de la tarjeta Arduino

Esta parte fue la que se llevo la mayor parte del tiempo y del esfuerzo, siendo esta parte el cerebro de la aplicación y en la que mas empeño dedicamos nos explayaremos un poco mas que en las anteriores partes. Mas fácil de concebir seria si presentamos el programa por bloques, el primer bloque sera dedicado al la programación relacionada con el teclado matricial y su conexión a la tarjeta Arduino. Siguiendo la misma filosofía de Arduino nosotros damos el código de nuestro programa libremente para que si alguien quiere hacer uso de el pueda ocuparlo sin problema y sin restricciones.

La idea de dar el código es la siguiente, si todos tuviéramos que reinventar la rueda jamas hubiéramos llegado a donde estamos. El conocimiento es de todos.

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

Como primer paso en cualquier programa es mandar a llamar todas las librerías que se vayan a necesitar. La primera librería que se llama es la que nos ayudara a leer crear y modificar datos de una memoria SD.

La segunda librería es para el uso del cristal liquido (LCD) y la tercera para el teclado.

Hasta acá son todas las librerías que se utilizaron, como se comentó anteriormente se desechó el uso de la librería password.

```
LiquidCrystal lcd(10, 9, A0, A1, A2, A3);
```

En la descripción de la librería encontramos que la manera correcta de utilizarla es definiendo primero los pines que se ocuparán para el manejo del LCD, en este caso utilizamos el LCD en modo de 4 bits.

```
lcd.clear();  
lcd.setCursor(0, 1);  
lcd.print("USUARIO:");  
lcd.setCursor(1, 1);
```

Después de definir las entradas y salidas pasamos el cursor a las coordenadas 0, 0, para luego desplegar en el LCD el mensaje inicial "USUARIO", para que el usuario sepa que tiene que ingresar su nombre de usuario, para no escribir en la misma línea del LCD perdiendo con esto algo de estética enviamos con la última línea el cursor a la posición 1, 0.

```
long lectura_teclado=0;  
long lectura_sd=0;  
int buttonState = 0;  
int aux50=0;  
char auxiliar;  
int contador_puntos;  
boolean auxiliar1;  
boolean auxiliar2;  
boolean aux_tiempo=0;
```

```
boolean aux110=0;

boolean aux120=0;

boolean aux100=0;
```

Como todo programa necesita auxiliares, acá definimos todos los auxiliares que se necesitaran, que nos servirán en la lectura del teclado, de la memoria SD, como contadores y como auxiliares en el workflow.

```
const byte ROWS = 4; //
const byte COLS = 3; //
// Define el Keymap
char keys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0',' '},
};
// Conectar keypad ROW0
//ROW1, ROW2 y ROW3 a los pines de Arduino.
byte rowPins[ROWS] = {5, 3, 2, 0};
byte colPins[COLS] = {8, 7, 6};
// Crear el Keypad
Keypad keypad = Keypad(
    makeKeymap(keys), rowPins,
    colPins, ROWS, COLS
);
```

El código anterior es necesario para controlar el teclado matricial. Definimos una matriz de 4X4 con cada uno de los caracteres correspondientes al teclado,



y luego se define en que pines de la tarjeta Arduino se conectara el teclado matricial.

Así Arduino reconocerá correctamente cuando se presione cada tecla y la tipificará de la forma adecuada. Por último creamos el objeto de la clase keypad.

```
int sal1 = A4;  
  
// pin A4 salida a rele  
  
int sal2 = A5;  
  
// pin A5 salida a rele}  
  
File myFile;  
  
int buttonState1 = 0;
```

Acá mostramos un poco mas de líneas de código, pero su explicación es sencilla, primero se definen las salidas para el control del relevador, luego de la biblioteca SD, del modelo File creamos un objeto llamado myFile, y seteamos nuestro primer auxiliar colocándolo a cero para iniciar así el main de nuestro programa con todas las configuraciones descritas.

Arduino toma como primer función a ejecutar lo que este contenido en la función setup, que como su nombre indica sirve para hacer las configuraciones correspondientes a nuestro proyecto.

```
void setup(){  
  
Serial.print (  
  
  "Initializing SD card..."  
  
);  
  
pinMode(53, OUTPUT);  
  
if (!SD.begin(53)) {  
  
  Serial.println(  
  
    "initialization failed!"
```

```

Serial.println(
    "initialization done."
);
keypad.addEventListener(
    keypadEvent
);
digitalWrite(sal1, LOW);
// activa relØ1 de la puerta
digitalWrite(
    sal2, Serial.print("Initializing SD")
pinMode(53, OUTPUT);
if (!SD.begin(53)) {
    Serial.println("initialization failed!");
    return;
}
reiniciar();
}

```

Aca como se puede apreciar se hace uso de esta funcion, la funcion setup, para empezar a trabajar con el SD inicializandolo y mandando el log a la computadora, y así si existiera algún problema seria fácil detectarlo, por último llamamos a la funcion reiniciar(), esta funcion será explicada mas adelante por ahora solo dire que devuelve todos los valores al estado inicial.

Arduino también cuenta con una función especial llamada loop la cual es la que repite Arduino una y otra vez como su nombre indica.

```

void loop () {
    keypad.getKey();
    buttonStatel = digitalRead(sal2);

```

```

    if (buttonState1 == HIGH) {
        digitalWrite(sal1, HIGH);
        // activa relØ1 de la
        // puerta
        delay(500);
        digitalWrite(sal1, LOW);
    }

    funcion_asteriscos()

```

Primero se añade la interrupcion para hacer caso a cualquier cambio de teclado y la llamada a la respectiva función,posteriormente tomando como ejemplo los workflow de otros programas masavanzados y con mayor potencia, se sigue a travez de flags el estado del programa, para asi saber que acción se debe de tomar en el siguiente paso.

Luego se checa o revisa el valor del primer auxiliar, este auxiliar nos ayuda a saber si se debe o no abrir la puerta.

```

void keypadEvent (KeypadEvent eKey) {
    switch (keypad.getState()) {
        case PRESSED:
            aux_tiempo=1;
            aux120=0;
            if(aux100==0) {
                lcd.print (eKey) PRESSED;
                aux100=1; PRESSED
            }
        else{
            if (aux120==0) {

```

```

        aux110=0;

        lcd.rightToLeft();

        lcd.print(' ');

        lcd.leftToRight();

        lcd.print('*');

        lcd.print(eKey);

    }

}

```

Esta es la función que hace el control del teclado matricial. Cuando se presione alguna tecla imprime en pantalla en el LCD el caracter presionado, y tambien se incia el siguiente proceso.

Se inicia un temporizador que contara hasta dos segundos y luego remplazara el caracter escrito por un asterisco, asi ocualtando a la vista de los demas la la tecla presionada, pero si el usuario tecla inmediadamente el cambio sera automatico tambien.

Se analiza la funcion con una sentencia switch se toman algunas decisiones, una de las funciones que mas trabajo nos costo y por los mismo es de las que mas orgullosos nos sentimos. La función en cuestión tiene como objetivo ocultar los caracteres, lo explicaremos merjor con un ejemplos.

Si presionamos una tecla, el valor de esta tecla se mostrará en el LCD y se tomaran dos posibles acciones.

- 1) Pasan dos segundos sin que el usuario vuelva a presionar el teclado  
Se remplazara el valor del LCD por un asterisco
- 2) El usuario digita otra tecla inmediateamente, se reemplaza lo que escribio el usuario sin esperar los dos segundos.

```

        if(eKey!=' ' && eKey!='*')

        lectura_teclado=(

```

```

(lectura_teclado*10)+eKey)-48;}

Serial.println(lectura_teclado);

switch (eKey){

case ' ': aux100=0; aux_tiempo=0;

guessPassword();

break;

case '*': lectura_teclado=0;

lectura_sd=0;

reiniciar();

}

}

```

En esta función es donde se lleva propiamente el flujo de trabajo aca podemos entender de una manera clara y rápida la manera en que se realiza la llamada a las funciones. Como primera función se llama la función usuario, cuando se haya caputarado el usuario independientemente si esta correco o no.

```

void guessPassword(){

if(aux50!=1){

funcion_usuario();

lcd.clear();

lcd.setCursor(0,0);

lcd.print("CLAVE:");

lcd.setCursor(1,1);

}

else{

funcion_clave();

funcion_comparar();

```

Después de que se hayan capturado tanto las contraseñas como los nombres de los usuarios, se procederá a hacer la comparación con la función `comparar`. La función `reiniciar` nos ayuda a volver a su estado inicial todas las variables y temporizadores que se ocuparon, para que en un nuevo ciclo del programa las funciones se ejecuten adecuadamente en el orden debido.

```
void reiniciar() {  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("USUARIO:");  
    lcd.setCursor(1, 1);  
    lectura_sd=0;  
    lectura_teclado=0;  
    contador_puntos=0;  
    aux50=0;  
    auxiliar1=false;  
    auxiliar2=false;  
    myFile.close();  
    aux110=0;  
    aux_tiempo=0;  
    aux100=0;  
}
```

La primera función que es llamada es la función `reiniciar` para que todos los valores iniciales sean de nuevo los mismos, después de haber ejecutado esta función se llama a la función `usuario`.

Esta función realiza la búsqueda comparando el valor concatenado en nuestra variable `usuario` con lo que encuentre almacenado en la memoria SD. La forma de hacer esta búsqueda es la siguiente. En la memoria SD existen

almacenados dos archivos, uno tiene la carga inicial del usuarios (usuarios.txt), y el otro contiene las contraseñas de estos usuarios respectivamente (contraseña.txt), cada contraseña y cada usuario están separados por una coma, siguiendo hasta acá la metodología de los archivos csv.

La siguiente función tiene como objetivo leer lo que haya tecleado el usuario y iniciar un flag llamado flag contraseña. La función crea una instancia de la clase myFile y a través de un método ya definido en esta función va leyendo uno por uno los caracteres de la memoria SD. Con el sentencia de programación while se recorre toda la cadena de caracteres existentes en la la memoria SD, cuando se encuentre una coma entonces se compara lo que hasta el momento se ha concatenado de la memoria SD con lo que el usuario tecleo, si no coincide se borra la variable usuario SD se continua con el recorrido, si por el contrario la comparación es correcta se pone el flag en uno y se termina la lectura con un break.

```
funcion_usuario() {  
    lectura_sd=0;  
    myFile = SD.open("usuario.txt");  
    aux50=1;  
    while(myFile.available()) {  
        auxiliar=myFile.read();  
        if (auxiliar!=',') {  
            lectura_sd= (auxiliar-48)+(lectura_sd*10);  
        }  
        else{  
            contador_puntos=contador_puntos+1;  
            If(  
                lectura_teclado==lectura_sd &&  
                lectura_teclado!=0) {  
                auxiliar1=true;  
            }  
        }  
    }
```

```

        myFile.close();

        aux50=1;

        lectura_teclado=0;

    }

```

En la función anterior podemos observar la manera en que se esta concatenando los caracteres, si usamos la opción concatenación que Arduino trae por defecto sera necesario que el tipo de datos sea String, pero al ser el tipo de dato String no es posible usar para compararlo un if ya que esta sentencia requiere forzosamente un operador int o flotante. Para el mejor manejo de memoria es necesario después de abrir la instancia para leer el archivo SD debemos cerrarlo de nuevo.

Así el método destructor de Arduino al ver que esta instancia ya no esta en uso procederá a desecharla ahorrando de esta manera memoria al momento de correr el programa.

```

void funcion_clave() {

    lectura_sd=0;

    myFile = SD.open("clave.txt");

    aux50=0;

    if(auxiliar1!=false){

        while(myFile.available()){

            auxiliar=myFile.read();

            if(auxiliar==' '){

                if (contador_puntos!=0){

                    contador_puntos=contador_puntos-1;

                }

                else{

```



```

        if(lectura_teclado==lectura_sd){

            auxiliar2=true;

            lectura_sd=0;

            myFile.close();

            return;

        }

        else{

            if(lectura_teclado==lectura_sd)

                auxiliar2=false;

            lectura_sd=0;

            myFile.close();

            return;

        }

    }

    else{

        if(contador_puntos==0){

            lectura_sd=auxiliar+(lectura_sd*10)-48;

        }

    }

}

myFile.close();

aux50=0;

```

Como ya se mencionó, cuando se entra en la función usuario hay un auxiliar que nos permite llevar la cuenta de los puntos que encuentre en cada comparación que tenga que realizar, dando así como resultado que al final de su comparación si esta fue exitosa tendrá un número determinado de puntos en el auxiliar.

Este auxiliar se emplea nuevamente en la función clave, por que a través de esa cuenta que se tiene de puntos, se llegará al lugar exacto de donde está la correspondiente contraseña del usuario, y al llegar acá se guardará este dato para posteriormente compararlo.

Para hacer un código más limpio y de mejor entendimiento se dividió el proceso en varias funciones. La función que realiza las comparaciones se muestra a continuación.

```
void funcion_comparar() {  
    if(auxiliar1==true && auxiliar2==true){  
        lcd.clear();  
        lcd.setCursor(0,0);  
        lcd.print("CORRECTO");  
        lcd.setCursor(0,1);  
        lcd.print("TELMEX-COMPUEBLA");  
        digitalWrite(sal1,HIGH);  
        delay(500);  
        digitalWrite(sal1, LOW);  
        delay(1000);  
        lcd.clear();  
        lcd.setCursor(0,1);  
        lcd.print("COMPUEBLA-TELMEX");  
    }
```

```

        lcd.setCursor(0, 0);
        lcd.print("Bienvenido - buen dia");
        delay(1000);

        for (int positionCounter = 0; positionCounter <
            22; positionCounter++) {
            lcd.scrollDisplayLeft();
            delay(150);
        }
        delay(3000);
        reiniciar();
    }

    else{
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("USUARIO O CLAVE INVALIDO");
        delay(1000);

        for (int positionCounter = 0; positionCounter <
            22; positionCounter++) {
            lcd.scrollDisplayLeft();
            delay(150);
        }
    }
}

```

Nosotros esperamos que el código del programa sea lo bastante limpio para que sobren las explicaciones. Acá se procede a comparar los valores obtenidos con anterioridad que son: la contraseña y el usuario teclados con la contraseña

y el usuario leídos de la memoria SD, si cualquiera de estas dos comparaciones falla el LCD mandará un mensaje de error y se tendrá que esperar 10 segundos para poder ingresar nuevamente las credenciales de usuario.

Por último mostramos la función que oculta los caracteres remplazándolos por asteriscos.

```
void funcion_asteriscos() {  
    if (aux_tiempo==1){  
        delay(10);  
        aux110=aux110+1;  
        if (aux110==156){  
            aux120=1;  
            aux_tiempo=0;  
            aux110=0;  
            lcd.rightToLeft();  
            lcd.print(' ');  
            lcd.leftToRight();  
            lcd.print('*');  
        }  
    }  
}
```

# **CAPÍTULO III**

## **Pruebas y Resultados**

Una vez que el sistema se instaló, aún tenían que hacerse pruebas para decidir que era confiable. Lo que se hizo fue lo siguiente. Durante un periodo de una semana se procedió de la siguiente manera, en horario de entrada o salida de personal en forma aleatoria, los errores a revisar son los siguientes.

1.- Como primer paso se contó el número de personas que entraron y salieron en ese horario.

2.- El usuario ingresaba correctamente su contraseña pero el sistema no la reconocía como contraseña válida, y por lo tanto el usuario tenía que volver a poner sus datos para poder ingresar.

3.- Cuando se tenía que reiniciar el sistema por que este había sufrido algún bloqueo.

4.- Cuando el usuario ingresaba correctamente su contraseña y número de usuario, el sistema lo registraba correctamente ya que encendía el led verde del sistema de reconocimiento, pero no se abría la puerta .

Durante el periodo que se realizaron las pruebas 173 veces los usuarios utilizaron el sistema, los resultados obtenidos se muestran en la siguiente gráfica



Gráfica 4.1 Resultado de prueba

Como se muestra en la gráfica podemos observar que el primer día se presentaron el mayor número de problemas, pero también ese día se identificaron las causas que en su mayoría fue una mala conexión eléctrica, una vez que se resolvió este problema en el segundo día de pruebas los errores disminuyeron.

El total de errores en todo el tiempo que duraron las pruebas fue de 42 y fueron 173 veces que los usuarios utilizaron el sistema , eso quiere decir que tuvimos un porcentaje de error del 24.27 %.

Todo sistema contruido por el ser humano es susceptible a presentar errores, pero se pueden ir detectando y mejorando, aunque claro nunca se tendrá un sistema perfecto.

# **CAPÍTULO IV**

## **Conclusiones**



Como objetivo principal se planteo desarrollar un control de acceso para el personal en las instalaciones del Centro de Operaciones y Mantenimiento de Larga distancia de TELMEX zona centro.

Este objetivo se cumplió satisfactoriamente ya que el sistema es capaz de impedir el acceso a las personas ajenas a las instalaciones, dejando solo paso a los que están debidamente calificados para estar dentro de esta área. Con esto se obtiene el control que se busca. Pasando ahora a los objetivos específicos como primer objetivo.

Es muy importante estar abierto a cualquier conocimiento sin importar que no forma parte explícitamente de nuestro trabajo principal, y también por que teniendo un mayor conocimiento del área donde se labora se puede llegar a cumplir los objetivos de una mejor manera. En el área donde estuvimos laborando se encargan de las comunicaciones y conmutaciones entre centrales troncales, estas comunicaciones ya desde hace tiempo ha empezado a cambiar su uso por fibra óptica.

A continuación se describen los objetivos específicos y como es que se cumplieron.

- Elaboración de un programa para Arduino.

La mayor parte de nuestros esfuerzos estuvieron canalizados a cumplir este objetivo, debido a que esta parte conforma la médula del sistema, al fin de cuentas es la que esta con relación constante con el usuario final. El programa desarrollado funciona correctamente cubriendo las expectativas generales.

- Instalación de servidor

La instalación del servidor se propuso el uso de servidores potentes como Ubuntu server o Centos, en vez de muy utilizado pero también muy problemático Windows server. Lamentablemente no se pudo realizar una implementación mayor, el mayor avance en este punto fue la instalación de OS dentro de un PC convencional.

- Elaboración de la interfaz web

Esta parte del proyecto debía ser el la la interfaz para la administración del sistema, se propusieron distintas formas para su realización como Django basado en Python o Drupal basado en PHP ya que estos framework poseen una amplia variedad de módulos y apoyo para crear proyectos de distintas maneras. La realización de esta parte no se llegó a completar por falta de tiempo principalmente.

El proyecto realizado funciona plenamente como un primer prototipo para aun existen diversidad de posibilidades por explotar y mejorar para darle una importante versatilidad.

Como parte importante en el proyecto se encontró el código libre, existen distintas y muy diversas librerías de código libre disponibles para nuestro uso que dan una gran potencia a todos nuestros trabajos. Con esto no quiero decir que uno como desarrollador sea incapaz de diseñar sus propias funciones o hardware, Arduino es una plataforma libre también, su uso se debió a que el ahorro de tiempo es inmenso dando la oportunidad al desarrollador ocupar ese tiempo ahorrado en la mejora de esas partes fundamentales del proyecto.

El sistema dará los siguientes beneficios al Centro de Operaciones y mantenimiento de TELMEX.

- Control de acceso a sus instalaciones.
- Seguridad al interior.
- Pauta para un sistema mejorado.

## **Bibliografía**

- [1]. Katsukito Ogata (2010), Ingeniería de Control Moderno, Madrid (España), Pearson.
- [2] W. Bolton (2001), Ingeniería de Control, México, Alfaomega.
- [3]. James T. Humphries, Leslie P. Sheets (2009), Electrónica Industrial, España, Paraninfo.
- [4] Angulo Usategui J. M., Angulo Martínez (1997), Microcontroladores PIC Diseño práctico de aplicaciones, Madrid (España), McGraw-Hill
- [5] Wikipedia La enciclopedia libre, <https://es.wikipedia.org/wiki/Arduino>,  
[Consulta: 14 de Junio del 2015]
- [6] Arduino Cc, <https://www.arduino.cc/en/Reference/HomePage>,  
[Consulta: 10 de Junio del 2015]