

DISEÑO E IMPLEMENTACIÓN DE REGISTRO AUTOMATIZADO DE  
ASISTENCIA MEDIANTE UN SISTEMA RFID PARA EL COLEGIO MILITAR  
INOCENCIO CHINCHA DEL ESPINAL, TOLIMA, EN EL AÑO 2015

NELSON ANDRES CALDERON RAMIREZ  
RICHARD EDUARDO GARCIA VILLANUEVA

INSTITUCION DE EDUCACION SUPERIOR ITFIP  
FACULTAD DE INGENIERIA Y CIENCIAS AGROINDUSTRIALES  
PROGRAMA INGENIERIA ELECTRÓNICA  
ESPINAL – TOLIMA  
2015

DISEÑO E IMPLEMENTACIÓN DE REGISTRO AUTOMATIZADO DE  
ASISTENCIA MEDIANTE UN SISTEMA RFID PARA EL COLEGIO MILITAR  
INOCENCIO CHINCHA DEL ESPINAL, TOLIMA, EN EL AÑO 2015

NELSON ANDRES CALDERON RAMIREZ  
RICHARD EDUARDO GARCIA VILLANUEVA

Trabajo de grado en la modalidad de proyecto de investigación para alcanzar el  
título de ingeniería electrónica.

Director  
DIEGO MAURICIO LAGUNA  
Ingeniero electrónico

INSTITUCION DE EDUCACION SUPERIOR ITFIP  
FACULTAD DE INGENIERIA Y CIENCIAS AGROINDUSTRIALES  
PROGRAMA INGENIERIA ELECTRÓNICA  
ESPINAL – TOLIMA  
2015

Nota de Aceptación

---

---

---

---

Presidente del Jurado

---

Jurado

---

Jurado

Espinal, Tolima, 20 de enero de 2016

## **AGRADECIMIENTOS**

### **Richard:**

Esta tesis se la dedico a mi Dios quien supo guiarme por el buen camino, darme fuerzas para seguir adelante y no desmayar en los problemas q se me presentaban, enseñándome a encarar las adversidades sin perder nunca la dignidad ni desfallecer nunca en el intento.

A mi familia porque por ellos es que soy lo que soy.

A mis padres por su apoyo, consejos, comprensión, amor, ayuda en los momentos difíciles, y por ayudarme con los recursos necesarios para estudiar. Me han dado todo lo q soy como persona, mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para conseguir mis objetivos. Gracias por estar siempre presentes guiándome y acompañándome para poderme realizar.

### **Nelson:**

Ante todo doy gracias Dios, quien supo guiarme en cada etapa de mi vida, haciéndome fuerte en cada tropiezo, para poder seguir adelante y no desmayar ante los problemas y adversidades, por permitirme seguir en mi camino directo a mis sueños; doy gracias a mis padres que me dieron la oportunidad de crecer personalmente, que confiaron en mí y me apoyaron en todo aspecto, gracias a ellos son la persona que soy hoy en día, gracias por ser mi ejemplo a seguir.

Gracias a todos y cada uno de mis familiares que de una u otra forma hicieron parte de este gran logro, que en su momento me fortalecieron y me apoyaron con sus consejos, gracias a la vida porque me ha dado la oportunidad de demostrarme que cuando uno tienes sueños y ganas, las grandes cosas sin diminutivas frente al deseo de triunfar!!

A todos les debo este gran logro y los que de aquí en adelante comenzaré a construir.

Gracias

## Contenido

1. OBJETIVOS.....	10
2.1 OBJETIVO GENERAL.....	10
2.2 OBJETIVOS ESPECÍFICOS.....	10
2. PROBLEMA DE INVESTIGACIÓN .....	11
3.1 JUSTIFICACIÓN .....	12
4. ESTADO DEL ARTE .....	13
5. MARCO TEÓRICO.....	16
5.1.1. Las TIC`s y la Educación.....	16
5.1.2 ¿Qué es una Base de datos? .....	18
5.1.3Tecnologías de Auto identificación.....	19
5.1.4 Tecnología RFID.....	21
5.1.5 Sistemas RFID.....	23
5.1.6 Frecuencia de funcionamiento .....	24
5.1.7 La etiqueta RFID.....	24
5.2 MARCO LEGAL.....	26
5.4 METODOLOGÍA .....	26
5.4.1 Cronograma .....	26
5.4.2 Diseño Metodológico.....	27
5.4.3. Presupuesto .....	28
6. ELEMENTOS QUE SE UTILIZARON .....	28
6.1 SENSOR.....	28
6.2 Arduino Leonardo .....	29
6.1 Pantalla LCD 16x2.....	30
7. DESARROLLO DEL PROYECTO.....	30
7.1 EL COLEGIO OBJETO DE INTERVENCIÓN.....	30
7.2 Base de datos.....	31
7.3 Programa Arduino.....	31
7.3.1 Librería MFRC522.h.....	31
7.3.2 Librería MFRC522.cpp.....	34

7.3.3 Librería LiquidCrystal.h .....	42
7.3.4 Programa Principal.....	45
7.4 Programa en visual estudio.....	46
7.4.1 Conexión con la base de datos.....	46
7.4.2 Consulta a la base de datos .....	47
7.4.3 Variables.....	48
6.4.4 Clase de datos para la tabla usuarios de la base de datos.....	48
7.4.5 Clase de datos para la tabla registro de la base de datos.....	50
7.4.5 Clase para ingresar datos a la tabla usuarios de la base de datos.....	50
7.4.6 Clase para ingresar datos a la tabla registro de la base de datos.....	51
7.4.7 Form1.vs Registro Automatizado de Asistencia .....	52
7.4.7.1 Código del Form1.vs Registro Automatizado de Asistencia .....	52
7.4.8 Form2.vs Ingresar Usuario .....	55
7.4.8.1 Código del Form2.vs Ingresar Usuario .....	55
7.4.9 Form3.vs Registro .....	57
7.4.9.1 Código Form3.vs Registro .....	58
6.5 Manual de Usuario.....	59
8. CONCLUSIONES .....	62
9. BIBLIOGRAFIA.....	63

LISTADO DE TABLAS

Tabla 1: Cronograma..... 26  
Tabla 2: Presupuesto..... 28

<b>Figuras</b>	
Figura 1: Tipos de etiqueta .....	21
Figura 2: Sistema RFID .....	23
Figura 3: Etiqueta RFID.....	25
Figura 4: Modulo RFID RC-522 .....	28
Figura 5: Nombre de los pines del RC-522.....	29
Figura 6: Arduino Leonardo .....	29
Figura 7: Pantalla 16x2.....	30
Figura 8: Escudo Colegio Militar Inocencia Chinca .....	30
Figura 9: Pantallazo de la base de datos.....	31
Figura 10: pantallazo del form1.vs en visual.....	52
Figura 11: pantallazo Form.vs ejecutado en visual.....	52
Figura 12: Pantallazo de Form2.vs ingresar usuario en visual.....	55
Figura 13: Pantallazo de Form3.vs Registro en Visual .....	57
Figura 14: Pantallazo de Exportación de datos a WORD .....	57
Figura 15: Portada del Manual de usuario Pág. 1 .....	59
Figura 16: Manual Usuario Pág. 2 y 3 .....	60
Figura 17: Manual de Usuario Pág. 4 y 5 .....	60
Figura 18: Manual de Usuario Pág. 6 y 7 .....	61



## **INTRODUCCION**

La tecnología RFID, ha tenido mucho auge en los últimos años debido a la relativa reducción de precios en el mercado, al incremento en sus capacidades y a las ventajas que presenta frente a otras tecnologías de auto identificación RFID promete revolucionar la vida de las personas por sus diversas aplicaciones.

Al estar involucrada en la cadena de producción y distribución de las fábricas, generará grandes beneficios tales como: especificación de la línea de producción, verificación de la calidad de los productos, elaboración de inventaros automáticos, a partir de que éstos ingresan al almacén de las tiendas; conocimiento sobre el momento de abastecimiento y cobro automático del carrito al pasarlo frente a la caja.

Los beneficios también se incluyen dentro de las actividades cotidianas. El desarrollo de refrigeradores con tecnología RFID permitirá no sólo identificar el momento en que un producto está a punto de caducar sino también en caso de requerir su abastecimiento, se informará al usuario de esta condición. Las lavadoras, por ejemplo, identificarán el ciclo de lavado que le corresponde a determinado tipo de ropa.

Finalmente otra aplicación se encuentra en plantas recicladoras de basura ya que las máquinas emitirán informes acerca del material en que están hechos ciertos productos logrando así facilidad en su separación y agrupamiento.

En fin las aplicaciones de esta tecnología son prometedoras, no obstante, aún se encuentra en proceso de evolución.

# **1. OBJETIVOS**

## **2.1 OBJETIVO GENERAL**

Diseñar e Implementar un registro automatizado de asistencia mediante un sistema RFID para el Colegio Militar Inocencio Chincha de El Espinal

## **2.2 OBJETIVOS ESPECÍFICOS**

- Analizar el procedimiento del registro de asistencia manual de los docentes del colegio.
- Categorizar los diferentes métodos para realizar un registro de asistencia automatizado.
- Seleccionar dispositivos electrónicos para el diseño de un prototipo de lectura de datos.
- Desarrollar una interfaz entre hardware y software.
- Realizar toma de muestras del módulo.

## 2. PROBLEMA DE INVESTIGACIÓN

El sistema de control de acceso propuesto en esta investigación surge a partir de la idea de realizar un sistema barato; con tecnología de punta; que su tiempo de vida sea, al menos a mediano plazo; escalable; confiable; seguro y con interfaces amigables.

La tecnología utilizada fue RFID, debido a la seguridad que presenta actualmente, la adaptabilidad que posee para este proyecto, además de ser una innovación tecnológica que poco a poco va tomando fuerza en el sector industrial.

Una ventaja de estas tarjetas es que no necesitan contacto físico (como introducir la tarjeta en una ranura o esperar a que sea reconocido ópticamente); sólo con aproximarla a cierta distancia del lector, la tarjeta será validada. Esto proporciona una “firma virtual”, esto es, el sistema asigna identificadores únicos a cada tarjeta para validar en el sistema dos tipos posibles de ingreso al poseedor de la credencial, según las necesidades de seguridad: aceptado, para usuarios autorizados y, denegado, para usuarios que en determinado acceso no están autorizados para ingresar.

En el Colegio Militar Inocencio Chinca del municipio del Espinal, cuenta actualmente con un sistema manual de asistencia para los docentes; este opera con un formato impreso en donde cada educador debe registrarse a través de su firma. Dicho proceso se realiza con el fin de tener una información de apoyo en el momento de realizar la liquidación de la nómina mensual.

Cabe mencionar que este proceso está generando situaciones negativas dentro de la institución, porque en ocasiones los formatos se pierden o no se diligencian en su totalidad dejando el historial incompleto.

Otro inconveniente que se presenta con el registro manual es al momento de elaborar la nómina, porque hay que revisar el registro diario de todo el mes, para concluir cuál es el sueldo que devenga cada docente.

Desde el punto de vista para la confiabilidad de la información, no es recomendable llevar el control de los datos en esta clase de formato, ya que ésta se puede perder y peor aún se le puede dar un mal manejo, debido a que este proceso no es operable por medio de un sistema de información automatizado, el cual ha venido presentando inconsistencias en el manejo de los datos, donde están involucrados los docentes, personal de nómina y de administración.

### **3.1 JUSTIFICACIÓN**

La tecnología de RFID es un sistema de autoidentificación inalámbrico, el cual consiste de etiquetas que almacenan información y lectores que pueden leer a estas etiquetas a distancia. La tecnología RFID está siendo adoptada cada vez por más industrias debido a que su costo es cada vez menor y sus capacidades son mayores.

Esto permite genera grandes beneficios como incrementos en la productividad y administración principalmente en los sectores de cadenas de suministro, transporte, seguridad y control de inventarios.

Con la implementación de un registro automatizado de asistencia, lo que se busca es dejar atrás el método de registro de asistencia manual, por uno sistematizado y confiable, haciendo posible que la información quede guardada directamente en una base de datos de fácil acceso, permitiendo de esta manera realizar un trabajo más ágil y eficiente sobre el control laboral de los docentes.

De aquí nace la idea implementar un sistema mediante el uso de unas tarjetas MIFARE que llevará un código único que identifica a cada educador del ente educativo. La información será captada por el módulo de lectura RFID y almacenada de forma instantánea.

Este sistema de poner en funcionamiento el uso de estas tarjetas, pretende incluirse en el campo tecnológico educativo, innovando un proceso que es manual, especialmente en la elaboración de Nómina.

Este registro podrá ser consultado en el ordenador que contenga la base de datos. Por lo cual será mucho más rápida, eficiente y segura la búsqueda.

## 4. ESTADO DEL ARTE

RFID no es una tecnología nueva, sino que lleva existiendo desde 1940. Durante la Segunda Guerra Mundial, los militares estadounidenses utilizaban un sistema de identificación por radiofrecuencia para el reconocimiento e identificación a distancia de los aviones: "FriendorFoe" (amigo o enemigo). Acabada la guerra, 37 los científicos e ingenieros continuaron sus investigaciones sobre estos temas. En octubre de 1948, Harry Stockman publicó un artículo en los Proceedings of the IRE titulado "CommunicationsbyMeans of ReflectedPower", que se puede considerar como la investigación más cercana al nacimiento de la RFID.

Durante la década de los 50 se realizaron muchos estudios relacionados con la tecnología, principalmente orientados a crear sistemas seguros para su aplicación en minas de carbón, explotaciones petrolíferas, instalaciones nucleares, controles de acceso o sistemas antirrobo. Durante esta época se publicaron dos artículos importantes: "Applications of MicrowaveHomodyne", de F. L. Vernon, y "RadioTransmission Systems with Modulatable Passive Responders", de D. B. Harris.

En los años 60 se profundizó en el desarrollo de la teoría electromagnética y empezaron a aparecer las primeras pruebas de campo, como por ejemplo, la activación remota de dispositivos con batería, la comunicación por radar o los sistemas de identificación interrogación-respuesta. Aparecieron las primeras invenciones convocación comercial, como "RemotelyActivated Radio Frequency Powered Devices", de Robert Richardson, "Communicationby Radar Beams" de Otto Rittenback, "Passive Datavt tecnología de identificación por radiofrecuencia (RFID) Transmisión RechniquesUtilizing Radar Beams" de J. H. Vogelmann, y "Interrogator-Responder IdentificationSystem", de J. P. Vinding. Asimismo, comenzaron las primeras actividades comerciales. Se fundaron Sensormatic y Checkpoint, que junto con otras compañías, desarrollaron un equipo de vigilancia electrónica anti-intrusión denominado EAS (ElectronicArticleSurveillance). EAS fue el primer desarrollo de RFID y el que indiscutiblemente se ha venido utilizando más ampliamente. Fue el preludio de la explosión de esta tecnología.

Durante los años 70 desarrolladores, inventores, fabricantes, centros de investigación, empresas, instituciones académicas y administración realizaron un arduo trabajo de desarrollo de la tecnología, lo que redundó en notables avances, apareciendo las primeras aplicaciones de RFID. A pesar de ello, la tecnología se siguió utilizando de modo restringido y controlado. Grandes empresas como Raytheon, RCA y Fairchild empezaron a desarrollar tecnología de sistemas de identificación electrónica, y en 1978 ya se había desarrollado un transpondedor pasivo de microondas. A finales de esta década ya se había completado una buena parte de la investigación necesaria en electromagnetismo y electrónica para RFID,

y la investigación en otros de los componentes necesarios, las tecnologías de la información y las comunicaciones, estaba empezando a dar sus frutos, con la aparición del PC y de ARPANET.

En los años 80 aparecieron nuevas aplicaciones. Fue la década de la completa implementación de la tecnología RFID. Los principales intereses en Estados Unidos estuvieron orientados al transporte, al acceso de personal y, más débilmente, a la identificación de animales. En Europa sí cobró un especial interés el seguimiento de ganado con receptores de identificación por radiofrecuencia como alternativa al marcado. Más tarde también aparecieron los primeros peajes electrónicos. La primera aplicación para aduanas se realizó en 1987, en Noruega, y en 1989 en Dallas. Todos los sistemas eran propietarios, y no existía la interoperabilidad.

Ya en la década de los 90 se tomó conciencia de las enormes posibilidades que podía brindar la explotación de RFID y comenzaron a aparecer los primeros estándares. En Estados Unidos se siguió profundizando en la mejora de los peajes automáticos y la gestión de autopistas. Mientras tanto en Europa se implementaron aplicaciones RFID para controles de acceso, peajes y otras aplicaciones comerciales. En 1999, un consorcio de empresas fundó el Auto-ID Center en el MIT.

Y a partir del año 2000, empezó a quedar claro que el objetivo de desarrollo de etiquetas a 0,05 dólares podría alcanzarse, con lo que la RFID podía convertirse en una tecnología candidata a sustituir a los códigos de barras existentes. El año 2003 marcó un hito en el desarrollo de la tecnología RFID: Walmart y el Departamento de Defensa (DoD) estadounidense decidieron adherirse a la tecnología RFID. Les siguieron otros fabricantes, como Target, Procter & Gamble y Gillette. En 2003 el centro AutoID se convirtió en EPC global, creadora de estándares adoptados por Walmart y el DoD. La empresa Texas Instruments desarrolló diversas aplicaciones para el control del encendido del motor del vehículo, control de acceso de vehículos o pases de esquí.

Asimismo, numerosas empresas en Europa se introdujeron en el mercado, más aún tras detectar la potencial aplicación en la gestión de artículos. En año 2002 empezó a despuntar la tecnología NFC (Near Field Communication), tecnología que mejora las prestaciones de RFID gracias a que incluye en un único dispositivo, un emisor y un receptor RFID, y que puede insertarse en un dispositivo móvil, aportando a éste nuevas funcionalidades para un gran número de aplicaciones.

Asimismo, numerosas empresas en Europa se introdujeron en el mercado, más aún tras detectar la potencial aplicación en la gestión de artículos.

En Europa, el proyecto lanzado en 2005 por Correos (España), Q-RFID, liderado por AIDA Centre SL, ha contribuido a incorporar las últimas tecnologías de control por radiofrecuencia para permitir la trazabilidad de la correspondencia a lo largo de todo el proceso postal. Q-RFID ha resultado uno de los más importantes proyectos

de RFID de Europa, suponiendo una gran contribución al desarrollo e implantación de la tecnología. Aunque el proyecto ha finalizado en 2007, el éxito alcanzado garantiza la continuidad del mismo.

La tecnología RFID parece estar tecnológicamente madura, aunque se halla inmersa en una continua evolución y mejora de sus prestaciones, como evidencia el número cada vez mayor de patentes y publicaciones en este campo. Las etiquetas son cada vez más pequeñas y su capacidad de almacenamiento continúa en aumento, las antenas son más eficientes y potentes permitiendo alcanzar rangos de cobertura mayores, los algoritmos de seguridad son cada vez más robustos...y con ello van surgiendo nuevas aplicaciones innovadoras.

Ha habido diversos casos de éxito en la implantación de sistemas de RFID, especialmente en actividades relacionadas con la logística y la distribución, como los ejemplos de grandes distribuidores como Walmart y Metro. Igualmente, en muchos aeropuertos hay implantado un sistema RFID para la gestión de las maletas. También hay empresas cuyo personal está equipado con una tarjeta RFID para su identificación y gestión de zonas autorizadas o restringidas.

Últimamente parece que la sanidad está atrayendo gran interés por parte del mercado de RFID. Un informe reciente de IDTechEx prevé un crecimiento del mercado RFID en sanidad desde los 90 M\$ en 2006 a más de 1000 M\$ en 2011 y 2.1 B\$ en 2016. Esta previsión de crecimiento resulta, sin duda, espectacular.<sup>1</sup>

En el 2008 un grupo de jóvenes en México implementaron *SISTEMA DE INFORMACION Y CONTROL DE ACCESO BASADO EN TECNOLOGIA RFID*. 2009, el que consistía básicamente, en darle una tarjeta a cada usuario para controlar el ingreso de los mismos, también cada medio de transporte (automóvil, moto) tendría una tarjeta con los datos del vehículo como placa, número de chasis, propietario.<sup>2</sup>

---

<sup>1</sup>HERNÁNDEZ POLANCO, Enrique. Diseño e implementación de un sistema de asistencia basado en RFID. 2012..<http://cdigital.uv.mx/handle/123456789/32385>

<sup>2</sup> CERVANTES NAJERA, ALEJANDRO; HERNANDEZ REYES, PABLO; SANTIAGO JACOBO, MIRIAM. SISTEMA DE INFORMACION Y CONTROL DE ACCESO BASADO EN TECNOLOGIA RFID. 2009. Tesis Doctoral.

## 5. MARCO TEÓRICO

### 5.1.1. Las TIC`s y la Educación

Las TIC en estos tiempos que corren ya no son tema de discusión en si sirven o no, si son o no necesarias, pues ya son una realidad, los migrantes digitales son personas que no manejan las Tics de manera integrada a su vida desde la niñez; que son la muchos de los profesores, administrativos y padres de familia actualmente; tienen que asumir que los alumnos son verdaderos nativos digitales, que muchas veces lo pasan por delante en el uso de las tecnologías, ya que viven mediatizados por teléfonos móviles, PlayStation, internet...etc. Pues tal vez los docentes quizás no saben sacar ese currículum no formal que se aprende fuera del aula y utilizarlo en sus clases, es por qué los discursos educativos ya no los convencen, el modelo del Pizarrón o el Busto Parlante, ya no les interesa, en suerte esto es una especie en vías de extinción, porque el docente ya no es el *propietario de la información*, y menos ahora que el conocimiento está socializado en la red, algo que se tiene miedo a perder, tal vez porque se está enseñando con métodos de siglo anteriores. Si hoy disponemos de 1 ordenador por cada 3 ó 4 alumnos, el acceso a Internet es de casi el 90%, hay un gran interés por integrar las TIC en la actividad pedagógica, y el 60% de docentes, emplea contenidos multimedia.

Entonces... ¿Qué está pasando, cuando se dispone de más medios que nunca y solo hay quejas?, pues el problema de la eficacia en el trabajo en educación pasa por mejorar el *trabajo en equipo* en el centro educativo.

Hoy en día se propone una taxonomía conceptual entre las dos grandes tipologías de usuarios de las nuevas tecnologías: los nativos digitales y los migrantes digitales. Los primeros serían aquellos consumidores y próximos productores de casi todo lo que existe (y existirá), son jóvenes que están desplegando plenamente su aprendizaje en el entorno tecnológico cuya lengua materna es la de las computadoras, los videojuegos e Internet. Frente a ellos, los inmigrantes digitales serían aquellos nacidos con anterioridad a la era tecnológica pero que sienten fascinación e interés por la tecnología, habiendo experimentado un proceso de adaptación más lento y lineal.

Se concluye que mientras para los padres y educadores (generalmente migrantes digitales) el *conocimiento es el poder*, en el emergente mundo liderado por los nativos digitales *el poder está en compartir el conocimiento*. Evidentemente, los principios de colectivización y apertura al intercambio del conocimiento previamente expuesto se adoptan de manera mucho más cercana a los hábitos de los nativos digitales.



Hipertextual: No es lo mismo leer y escribir documentos con una estructura secuencial y contruidos a base de caracteres alfanuméricos y alguna imagen, que leer y escribir documentos con una estructura hipertextual que pueden combinar fácilmente los textos alfanuméricos con imágenes, vídeos y sonido. Además los documentos digitales tienen una dimensión de profundidad de la que carece el texto escrito ya que las palabras pueden actuar también como hipervínculos y pueden ir apareciendo progresivamente según las actuaciones del lector. Cortázar pretendía con su novela *Rayuela*: que se pudiera leer alterando las páginas para que en cada lectura surgiera una novela distinta. El proceso de lectura siempre es un proceso de construcción del significado. El texto es interpretado e interiorizado de distinta forma por lectores que tengan un bagaje cultural y un contexto diferentes e, incluso, por el mismo lector en momentos diferentes. Pero siempre el texto es el mismo. La lectura digital no, porque es interactiva.

Interactiva: En el caso de la lectura digital por Internet el lector construye su propio texto a partir de fragmentos a partir de los hipervínculos que pulsa. El lector colabora con el autor en la creación del texto, en el sentido de que cada enlace escogido determina lo que a continuación podrá ver en la pantalla.

Intertextual: La intertextualidad consiste en relacionar textos, proponiendo una relectura de ambos, original y nuevo. Consiste en el hecho de la presencia en un determinado texto de expresiones, temas y características estructurales procedentes de otros textos, y que han sido incorporados a una obra en forma de citas, alusiones, imitaciones, recreaciones paródicas... No hay nada en Internet que no se haya tratado en otras páginas y lugares. Su versión más pedestre es el *corta y pega*, una forma de intertextualidad que es simplemente plagio, que es condenable, en mi opinión, más por el hecho de no dar lugar a reflexión ni pensamiento alguno, que por el hecho de copiar. Porque cuando la cita, la parodia, la imitación, da lugar a un diálogo de ideas, a la construcción de un texto que se relaciona con otros, buscando nuevos significados; en ese caso, siempre es enriquecedora. Debemos pedir a nuestros alumnos que corten y peguen, pero solo cuando tenga un sentido.

Si bien es cierto, los espacios en las deben ser fundamentales para propiciar este hábito, solo que el entorno y los cambios actuales que se dan, hacen que los adolescente no hagan buen uso de estos tiempos dedicados no solo a una lectura comprensiva sino que se enriquezca el lenguaje a partir de elementos necesarios en su alfabetización tecnológica y virtual

Partiendo de esta habilidad, cada individuo teje un compendio de defintorios que se puede llamar alfabetización tecnológica que dan significados para construir pensamientos, partiendo de un concepto. Sin duda es el docente que debe buscar las diferentes estrategias y herramientas necesarias para hacer que sus estudiantes

se motiven, haciendo de la “transversalidad curricular del uso de las TIC, como lo plantea el Plan Nacional Decenal de Educación 2006-2016”, Desde el ejercicio de la docencia, las Tics deben representar una herramienta que posibilite un proceso interactivo de comunicación mediante estrategias que faciliten el fortalecimiento de la comprensión lectora, que posibilite la codificación de la información de un modo sistematizado a un proceso lecto- escritor coherente y que mediado por herramientas tecnológicas se ha capaz de comprender un texto, extraer el significado que el mismo texto le ofrece. Esto implica reconocer que el sentido del texto está en las palabras y oraciones que lo componen y que el papel del lector consiste en descubrirlo.

### 5.1.2 ¿Qué es una Base de datos?

Una base de datos, es una colección de archivos, la cual es controlada por un software denominado: Sistema de Administración de Datos, o DBMS por su sigla en inglés. La mayoría de bases de datos actuales son de tipo *relacional*, son llamadas así porque utilizan tablas relacionadas por un campo en común entre las mismas.<sup>3</sup>

Las bases de datos brindan la posibilidad de almacenar información de manera eficiente en un sitio web. Permite proporcionar información actualizada, facilita la búsqueda de información, optimiza los costos de mantenimiento, permite controlar el acceso a la misma y puede almacenar las preferencias de los usuarios.

#### Bases de Datos en MySQL

En sus inicios, MySQL por sus siglas en inglés (My Structured Query Language) o Lenguaje de Consulta Estructurado en español, era considerado solo una aplicación para sitios web; actualmente es una solución para el manejo de datos. Una de las ventajas más grandes de MySQL es que es un software de código abierto, lo que implica que cualquier persona puede acceder a su código fuente, lo que necesariamente conduce a que cualquier programador en el mundo puede contribuir a arreglar problemas que tenga, agregar elementos o realizar mejoras.

El MySQL, Es un sistema que gestiona bases de datos, ampliamente utilizado por su rendimiento y facilidad de manejo. Una de sus ventajas es su facilidad de uso y una rápida puesta en marcha. Está disponible para muchas plataformas.

Sus principales características son:

- ✓ Es una base de datos estable

---

<sup>3</sup> ALVARADO SÁNCHEZ, Jorge Alberto. Sistema de control de Acceso con RIFD. Trabajo de Grado Maestría en ciencias, Especialidad de Ingeniería Electrónica. México D. F.: Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. 2008.

- ✓ Los mensajes de error se pueden presentar en español y puede hacer ordenaciones utilizando la letra Ñ o con las palabras acentuadas: manejo de la tilde.
- ✓ Posee múltiples opciones de conectividad: TCP/IP; sockets tanto para NT como para UNIX.
- ✓ Puede gestionar hasta cinco millones de columnas, sesenta mil tablas y cincuenta millones de registros
- ✓ La forma de ser administrada es a través de *privilegiados* y *usuarios*
- ✓ Su versatilidad abarca su utilización incrustada en aplicaciones o bien como cliente-servidor.

### **5.1.3Tecnologías de Auto identificación**

Desde hace años están disponibles en el mercado distintas tecnologías para la identificación de productos, personas e incluso animales. En ese sentido, uno de los principales exponentes ha sido el código de barras, el cual, ha logrado penetrar prácticamente en todas las cadenas de distribución, almacenes y sistemas de control de acceso, por citar algunos ejemplos. Sin embargo, en los últimos 10 años, se ha dado un *boom* de nuevas tecnologías, o más bien de aquellas que ya existían, pero que hasta ahora pudieron entrar al mercado masivo. La razón principal son todas las ventajas tecnológicas que ofrecen frente a los esquemas tradicionales, aunado a la baja en los precios.

### **Comparación de Tecnologías de Autoidentificación**

Dentro del ámbito de la tecnología de identificación, aplicado al control de acceso, se pueden encontrar diversas tecnologías como: sistemas biométricos, tarjetas magnéticas, código de barras, RFID y memorias de contacto que se describen en los siguientes párrafos.

### **Acceso con Sistemas Biométricos**

Este tipo de identificación se realiza a través del análisis y/o medición de características físicas. Algunas de las técnicas biométricas que existen son:

- Reconocimiento de iris
- Reflexión retinal
- Geometría de la mano

- Geometría facial
- Termografía mano, facial
- Huellas dactilares
- Patrón de la voz

La identificación biométrica ofrece una ventaja significativa, dado que bajo este sistema, se identifica explícitamente a la persona, no así a alguna credencial u otro objeto.

La razón por la cual no es aplicable para ciertos problemas una tecnología de este tipo es porque no existen sistemas que ofrezcan una confiabilidad cercana al 100 por ciento.

La mayoría de los sistemas de este tipo tienen una eficiencia menor a lo deseable. Otra desventaja de este tipo de sistemas es que son más costosos.

#### Acceso con Tarjetas magnéticas

Estos sistemas se basan en la lectura de una banda magnética. Utilizan señales electromagnéticas para registrar y codificar información en una banda que puede ser leída por una máquina para identificación instantánea. La aplicación más difundida es la de las tarjetas de crédito.

Sus ventajas son proporcionar agilidad en el acceso, dar identificación única al poseedor, bajo costo, además de que no son fácilmente falsificables. Sin embargo, su uso continuo las deteriora físicamente como consecuencia de la fricción al momento de la lectura. Además si alguna tarjeta es acercada a alguna fuente electromagnética, relativamente fuerte, puede modificar la información que contiene, perdiendo con ello su utilidad.

#### **Acceso con Tarjetas de Código de Barras**

El código de barras se inventó hace más de 25 años y durante este tiempo, ha sido la tecnología más utilizada por los comercios para identificar los productos en venta. Este tipo de identificación se realiza codificando datos en una imagen formada por combinaciones de barras y espacios. Las imágenes son leídas por equipos especiales de lectura óptica a través de los cuales se pueden comunicar datos a la computadora.

Proporciona las mismas ventajas que las tarjetas magnéticas y no es necesario el contacto físico entre la tarjeta y el lector, no obstante debe de existir una línea de vista entre ellos. Este tipo de sistema es barato, sin embargo, estas tarjetas son fácilmente falsificables o alterables siendo esto una gran debilidad para un sistema estricto de control de acceso, por lo que esta desventaja es significativa para descartar el uso de tarjetas por código de barras para esta aplicación.

Se han inventado alrededor de 270 diferentes simbologías para soportar requerimientos específicos y aproximadamente 50 de éstos se utilizan ampliamente en la actualidad. Cada una de estas simbologías cae dentro de alguna de las siguientes tres categorías:

**Lineal.** Consiste en líneas verticales, de diferentes anchos, con espacios blancos que separan dos líneas adyacentes. El máximo número de caracteres que pueden ser codificados, mediante esta metodología, son 50.

**Dos dimensiones.** Esta simbología tiene la mayor capacidad de almacenamiento, el máximo número de caracteres que pueden ser codificados es de 3,750.

**Tres dimensiones (Bumpy).** Este tipo de código de barras es leído, utilizando el relieve de las barras, es decir, no depende del contraste entre barras oscuras y espacios, por lo tanto puede ser embebidos directamente en los productos como por ejemplo en llantas o en partes plásticas directamente desde el molde. La ventaja de estos códigos es que pueden ser utilizados en ambientes de uso rudo.



Figura 1: Tipos de etiqueta

#### 5.1.4 Tecnología RFID

En los últimos años se ha hecho muy popular la implementación de tecnología de identificación por radiofrecuencia RFID (Radio-frecuencia-identificación) en aplicaciones como control de acceso, transporte, logística, bibliotecas, fidelización, etc., ya que RFID permite el acceso a los datos de manera inalámbrica empleando

para ello tags de radiofrecuencia (en adelante llamados PICC por sus siglas en inglés Proximity Integrated Circuit Cards)<sup>4</sup>

La RFID es una tecnología de identificación automática similar a la del código de barras. Su particularidad consiste en utilizar una señal de radiofrecuencia en lugar de una óptica. Sus características son las siguientes:

- Posibilidad de modificar los datos. Depende del estándar que se utilice, aunque sí es posible. Por ejemplo, utilizando el estándar EPC, existen básicamente varias clases de etiquetas de sólo lectura, de una escritura y múltiples lecturas o de lectura escritura.
- Seguridad de los datos. En las últimas generaciones de dispositivos RFID es posible cifrar los datos, de forma que no puedan ser leídos con lectores RFID estándar.
- Cantidad de datos almacenados. Hasta 1 MB de información en los últimos prototipos.
- Costos. En descenso a medida que se aplican los últimos avances tecnológicos. El objetivo de hacer unos años de alcanzar los 0,05€ por etiqueta parece cada vez más cercano, aunque lógicamente depende del tipo de etiqueta.
- Estándares. Existen diferentes estándares universalmente aceptados, y relacionados con la banda de frecuencia utilizada, que como ya hemos visto, determina el tipo de sistema RFID.

Los dos estándares principales son el estándar EPC y el estándar ISO.

- Vida útil. Al no haber necesidad de contacto físico ni de baterías, la vida útil de las etiquetas pasivas es muy grande. Las etiquetas activas tienen limitada su vida útil a la duración de su batería.
- Tamaño. En general, desde el tamaño de un botón o un caramelo hasta el tamaño de un paquete de tabaco. No obstante, Hitachi ha anunciado recientemente su mu-chip, un chip RFID con tecnología de 2,4 GHz y un tamaño de 0,4 x 0,4 mm, con un espesor de 0,06 mm.

---

<sup>4</sup>MARÍN RAMÍREZ, Leonel. Arquitectura de seguridad en la implementación de soluciones que utilicen tarjetas inteligentes sin contacto. 2013.- <http://hdl.handle.net/10819/1394>

- Distancia de lectura. Las etiquetas pasivas tienen un alcance del orden del metro, y las activas pueden tener un alcance de decenas de metros. Además, para realizar la lectura o escritura no se necesita línea de visión directa.
- Número de elementos que se pueden leer simultáneamente. Un lector puede leer cientos de etiquetas de forma casi simultánea.
- Posibilidad de interferencias. En función de la frecuencia, los líquidos, madera o metales puede impedir la propagación de la señales.<sup>5</sup>

### 5.1.5 Sistemas RFID

El funcionamiento de los sistemas RFID es simple, y son necesarios dos elementos básicos: una etiqueta electrónica y un lector. Básicamente lo que ocurre es que cuando se enciende el lector, este empieza a emitir una señal a una frecuencia previamente establecida (típicamente 860-915 MHz o 13.56MHz para HF). Cualquier etiqueta que este asociado a ese lector y que se encuentre en la proximidad del lector, detectara la señal y usando la energía de la misma. Hará funcionar sus circuitos internos. Una vez que la etiqueta haya decodificado la señal como válida, enviara una respuesta al lector enviando la información que contiene en forma de señales de radio frecuencia.

En un sistema típico, las etiquetas están relacionados con objetos, y cada una contiene una memoria interna EEPROM, en el cual almacena información acerca de objeto, como un código único, o en algunos casos información más detallada.

El proceso de comunicación entre el lector y las etiquetas son controladas por un de los varios protocolos existentes, como el de la ISO 15693 e ISO 18000-3 para HF o el de la ISO 18000-6 y EPC para UHF.

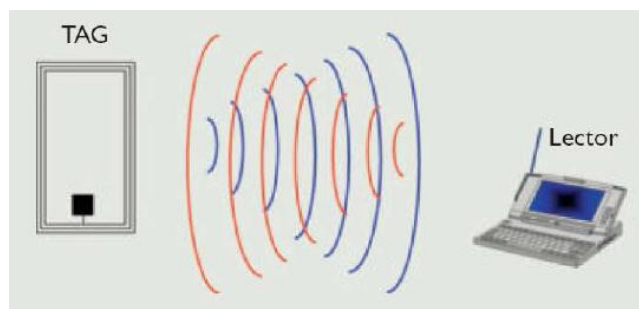


Figura 2: Sistema RFID

<sup>5</sup>HERNÁNDEZ POLANCO, Enrique. Diseño e implementación de un sistema de asistencia basado en RFID. 2012. -<http://cdigital.uv.mx/handle/123456789/32385>

Una vez que la información ha sido procesada se envía al sistema de control, que es el que gestiona y procesa la información, este sistema puede ir desde un Microcontrolador hasta un computador lo cual depende de la complejidad del sistema a controlar.

### **5.1.6 Frecuencia de funcionamiento**

Los sistemas RFID funcionan en distintas frecuencias dependiendo de la aplicación. Se las puede agrupar en cuatro rangos de frecuencias:

- Baja frecuencia LF (9 - 135KHz).- su principal ventaja es su aceptación en todo el mundo debido a que está ampliamente difundida. La distancia de lectura es inferior a 1.5 metros, por lo que las aplicaciones más habituales son la identificación de animales, inmovilización de vehículos, control de acceso, identificación de objetos
- Frecuencia Alta HF (13.56 MHz).- Esta frecuencia también está muy difundida. Normalmente se utiliza en aplicaciones tales como la trazabilidad de productos, movimientos de equipaje de avión o acceso a edificios.
- Frecuencia Ultra-alta UHF (433 MHz y 860-960 MHz).- los equipos que operan a estas frecuencias UHF no pueden ser utilizados de forma global porque no existen regulaciones globales para su uso y su aplicación depende de la legalidad del país. Este tipo de frecuencia se usa para aplicaciones de trazabilidad con etiqueta activas.
- Frecuencia de Microondas (2.5 GHz y 5.8 GHz).- Estas Frecuencias son más habituales para las etiquetas activas y no tienen el problema de la falta de regulaciones globales, además ofrecen largas distancias de lectura y altas velocidades de transmisión. Las etiquetas activas que operan en el rango de las microondas son usados para el seguimiento y trazabilidad de personas u objetos.<sup>6</sup>

### **5.1.7 La etiqueta RFID**

Una etiqueta RFID consta de tres elementos:

- Una antena
- Un circuito integrado
- Un elemento almacenador de energía

La antena permite realizar la comunicación entre la etiqueta y el lector. Su tamaño limita la distancia máxima a la que puede realizarse la lectura.

---

<sup>6</sup>GORDÓN DÍAZ, NathaliYessenia. Control de acceso en la entrada del Instituto Geofísico utilizando tecnología RFID. 2009.- <http://bibdigital.epn.edu.ec/handle/15000/1482>



El circuito integrado es un circuito mixto analógico-digital. La parte analógica se encarga de controlar la alimentación y la comunicación por radiofrecuencia. Por otro lado, la parte digital gestiona la información almacenada en la etiqueta. Por último, es necesario incluir un elemento para alimentar al circuito. En función del elemento usado existen dos tipos de etiquetas: las activas y las pasivas. En las primeras se usa una batería para alimentar el circuito. Aunque los fabricantes garantizan una duración de la batería de 10 años, su uso aumenta el tamaño de la etiqueta y la encarece. La principal ventaja es un mayor rango de lectura al poder emitirse la información con mayor potencia. Este tipo de etiquetas se usan, por ejemplo, en los peajes automáticos de autopistas.

En las etiquetas pasivas, el elemento almacenador de energía es un condensador, el cual se carga con la energía emitida por el lector y luego utiliza dicha energía para responder. Por ello, la potencia de emisión está limitada, por lo que la distancia entre el lector y la etiqueta no puede ser muy elevada. La ventaja obvia de este tipo de etiquetas es el ahorro de espacio, la duración prácticamente ilimitada de la etiqueta y su menor coste. Debido a esto, éstas son las etiquetas más usadas, aplicándose en campos tan diversos como la identificación de animales, llaves de contacto de automóviles, identificación de productos en cadenas de montaje, control de accesos, cronometraje de carreras, etc. Obviamente, los componentes mencionados han de protegerse del ambiente exterior, por lo que en función de la aplicación habrá de elegirse el encapsulado adecuado. El más sencillo es el que se ha mostrado en la Figura 2 consistente en una lámina de plástico. No obstante los fabricantes ofrecen innumerables encapsulados, incluso a medida del cliente. Por ejemplo existen etiquetas en formato tarjeta de crédito para control de accesos, encapsuladas en una ampolla de vidrio para identificar animales, etiquetas en forma de clavo para palets, encapsulados resistentes a altas temperaturas para etiquetar equipos que tengan que soportar condiciones adversas, etc.<sup>7</sup>

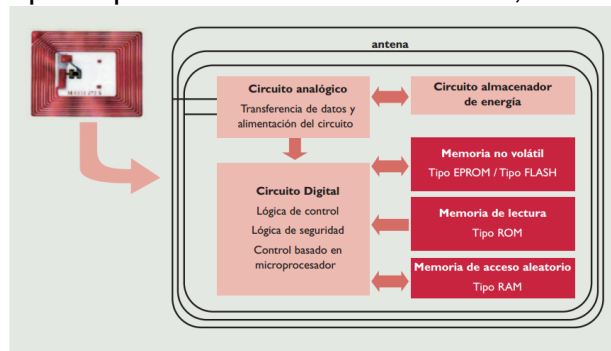


Figura 3: Etiqueta RFID

<sup>7</sup>FERNÁNDEZ, SadotAlexandres; FRÍAS, José Daniel Muñoz; GARCÍA, Carlos Rodríguez-Morcillo. RFID: La tecnología de identificación por radiofrecuencia. En *anales de mecánica y electricidad*. Asociación de Ingenieros del ICAI, 2006. p. 47-52

## 5.2 MARCO LEGAL

- ISO/IEC14443-1 Physical characteristics
- ISO/IEC14443-2 Radio Frequency Power and Signal Interface
- ISO/IEC14443-3 Initialization and anti-collision
- ISO/IEC14443-4 Transmission Protocol

## 5.4 METODOLOGÍA

Con el fin de alcanzar los objetivos generales y específicos será necesario realizar los siguientes pasos:

Contacto permanente entre los usuarios de la plataforma (Directivos, docentes, padres de familia, estudiantes) dando como resultado el seguimiento de las actividades realizadas por el alumno el principal beneficiario de la plataforma; gestión de contenidos y publicación de los mismos con el fin de que pueda ser visualizados o descargados por el usuario dependiendo de su rol en la plataforma; comunicación en tiempo real, para estos se crearan (foros, chat) dando como resultado la comunicación y un trabajo común entre los usuarios del sistema; asignación de permisos con el fin de controlar los accesos a los usuarios dependiendo de su labor en la plataforma.

### 5.4.1 Cronograma

		NOVIEMBRE				DICIEMBRE				ENERO.			
	ACTIVIDAD PROGRAMADA	1	2	3	4	1	2	3	4	1	2	3	4
1	Estado del arte												
2	Estudio del módulo RFID												
3	Análisis del módulo a utilizar												
4	Buscar información de pruebas realizadas con el módulo.												
5	Lectura del módulo utilizando Microcontrolador												
6	Creación de la Base de Datos												
7	Creación de programa de interfaz en el ordenador												
8	Enlace de la base de datos e interfaz												
9	Prueba general de la tarjeta												
10	Prueba de funcionamiento en diferentes entornos de trabajos												
11	Seguimiento al funcionamiento del dispositivo												
12	Entrega del dispositivo												

Tabla 1: Cronograma

## 5.4.2 Diseño Metodológico

Es muy importante realizar el análisis y la planificación cuando se empieza el desarrollo de un proyecto web, con el fin de conseguir que la plataforma funcione bien y que el proyecto sea un éxito, para esto se analizará el mercado, con el fin de ser más competitivos, investigar el sector y el medio en que se envuelve la empresa, de la mano del cliente se definirá: mapa web del sitio, fechas de entrega, las tecnologías que se van a implementar

Se realizará el diseño del prototipo según las características que especifique el cliente, de modo que sea usable y accesible, hasta que el cliente quede totalmente satisfecho no se dará por terminado el diseño, no importa cuántos cambios se le deba desarrollar.

- **Desarrollo del estado del arte:** describir las investigaciones más recientes y actuales sobre la tecnología RFDI como también mirar su historia y procedencia.
- **Desarrollo del marco teórico:** se orientará el estudio de cómo se puede prevenir errores que han cometido otros estudios referentes a la tecnología RFDI.
- **Pruebas de lectura y almacenamiento de datos:** se realizará un diseño de una tarjeta que transmitirá los datos tomados y la sincronizará con la base de datos. Luego de esto se realizarán pruebas para la detección de fallas que puedan llegar a pasar en el proceso de lectura del mismo.
- **Desarrollo de interfaz de usuario:** se procederá a crear la interfaz usuario para la visualización de los datos.

### Herramientas de recolección de información

Se utilizarán las siguientes herramientas de recolección de información:

*Revisión Bibliográfica y cibergráfica*

### Resultados esperados

- Automatizar el registro de asistencia manual.
- Reemplazar los archivos físicos por archivos digitales.
- Recolección de datos de forma precisa y sistematizada.
- interfaz humano maquina (HMI).

## Recursos utilizados

*Institucionales:* Biblioteca, laboratorio de electrónica, pasillos, salones de clase.

*Humanos:* Ing. Profesional en electrónica, estudiante.

*Tecnológicos:* Módulo RDFI, microcontroladores, software, PC, instrumentos de medición.

### 5.4.3. Presupuesto

	ITEM	COSTO
1	MICROCONTROLADOR	\$30.000
2	MÓDULO RFID	\$80.000
3	OTROS ELEMENTOS ELECTRÓNICOS	\$50.000
4	FABRICACION CIRCUITO IMPRESO	\$80.000
5	PAPELERIA	\$50.000
6	TRANSPORTE	\$30.000
7	OTROS GASTOS	\$80.000
	<b>TOTAL</b>	<b>\$ 400.000</b>

Tabla 2: Presupuesto

## 6. ELEMENTOS QUE SE UTILIZARON

### 6.1 SENSOR

El sensor que se utilizó es el RFID RC-522. Este sensor tiene una distancia de lectura de 60mm, un protocolo de comunicación SPI y su fabricante es Arduino.



Figura 4: Modulo RFID RC-522

Este sensor compone de 8 pines los cuales 4 son para comunicación, 2 para la alimentación y 1 de Reset.

RFID RC552
SDA
SCK
MOSI
MISO
IRQ
GND
RST
3.3 V

Figura 5: Nombre de los pines del RC-522

## 6.2 Arduino Leonardo

El Arduino Leonardo es una placa electrónica basada en el ATmega32u4. Cuenta con 20 pines digitales de entrada / salida (de los cuales 7 se pueden utilizar como salidas PWM y 12 entradas como analógicas), un 16 MHz oscilador de cristal, una conexión micro USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el Microcontrolador; simplemente conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA o la batería a CC para empezar.

El Leonardo difiere de todas las placas anteriores en que el ATmega32u4 ha incorporado en la comunicación USB, eliminando la necesidad de un procesador secundario. Esto permite que el Leonardo aparezca a un ordenador conectado como un ratón y el teclado, además de un virtual (CDC) puerto serie / COM.



Figura 6: Arduino Leonardo

### 6.1 Pantalla LCD 16x2

Un LCD 16x2 significa que puede mostrar 16 caracteres por línea y hay 2 tales líneas. En este LCD cada personaje se muestra en la matriz de píxeles 5x7. Este LCD tiene dos registros, es decir, del sistema y de datos.



Figura 7: Pantalla 16x2

## 7. DESARROLLO DEL PROYECTO

### 7.1 EL COLEGIO OBJETO DE INTERVENCIÓN



Figura 8: Escudo Colegio Militar Inocencia Chinca

El colegio militar Inocencia Chinca Es pinal fue fundado en el año 2012.

Su visión es El Colegio Militar Inocencia Chinca para el 2015se proyecta como el primer plantel educativo del departamento del Tolima

Su misión es el colegio militar Inocencia Chinca, forma hombres y mujeres útiles a la sociedad, íntegros, aptos ética, moral, intelectual y físicamente. Personas respetuosas de los derechos humanos, la constitución y la ley, al servicio de la patria y promotores de una cultura de protección del medio ambiente.

## 7.2 Base de datos

Se elaboró una base de datos en MYSQL Workbench 6.3 con dos tablas, la primera tabla es de datos personales del docente, y la segunda es para el registro.

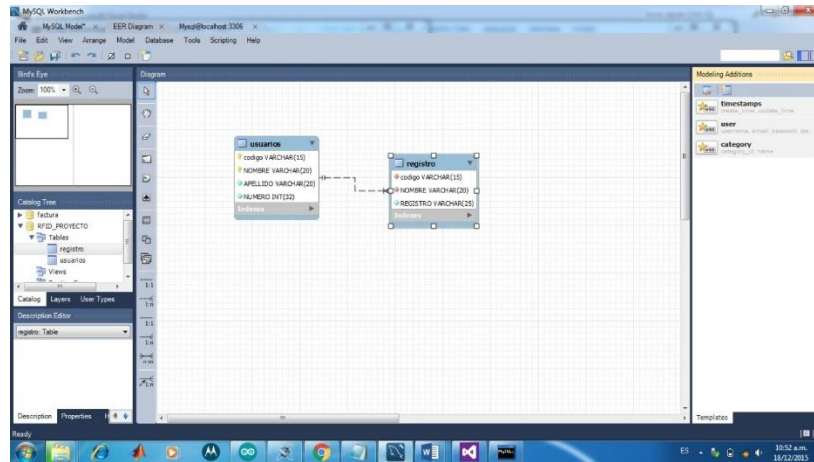


Figura 9: Pantallazo de la base de datos

## 7.3 Programa Arduino

Para la programación de Arduino se utilizó la plataforma Arduino-Nightly, la programación se divide en tres librerías y el programa principal.

Las librerías MFRC522.h y MFRC522.cpp deberán ser guardadas en la carpeta de librerías de Arduino-Nightly

### 7.3.1 Librería MFRC522.h

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

#define MAX_LEN      16    // Maximum length of an array.

// MF522 MFRC522 error codes.
#define MI_OK        0    // Everything A-OK.
#define MI_NOTAGERR  1    // No tag error
#define MI_ERR       2    // General error

// MF522 Command word
```

```

#define MFRC522_IDLE      0x00  // NO action; Cancel the current command
#define MFRC522_MEM       0x01  // Store 25 byte into the internal buffer.
#define MFRC522_GENID    0x02  // Generates a 10 byte random ID number.
#define MFRC522_CALCCRC  0x03  // CRC Calculate or selftest.
#define MFRC522_TRANSMIT 0x04  // Transmit data
#define MFRC522_NOCMDCH  0x07  // No command change.
#define MFRC522_RECEIVE  0x08  // Receive Data
#define MFRC522_TRANSCIVE 0x0C  // Transmit and receive data,
#define MFRC522_AUTHENT  0x0E  // Authentication Key
#define MFRC522_SOFTRESET 0x0F  // Reset

// Mifare_One tag command word
#define MF1_REQIDL      0x26  // find the antenna area does not enter hibernation
#define MF1_REQALL     0x52  // find all the tags antenna area
#define MF1_ANTICOLL   0x93  // anti-collision
#define MF1_SELECTTAG  0x93  // election tag
#define MF1_AUTHENT1A  0x60  // authentication key A
#define MF1_AUTHENT1B  0x61  // authentication key B
#define MF1_READ       0x30  // Read Block
#define MF1_WRITE      0xA0  // write block
#define MF1_DECREMENT  0xC0  // debit
#define MF1_INCREMENT  0xC1  // recharge
#define MF1_RESTORE    0xC2  // transfer block data to the buffer
#define MF1_TRANSFER   0xB0  // save the data in the buffer
#define MF1_HALT       0x50  // Sleep

//----- MFRC522 registers-----
//Page 0:Command and Status
#define Reserved00      0x00
#define CommandReg     0x01
#define CommIEncReg    0x02
#define DivIEncReg     0x03
#define CommIrqReg     0x04
#define DivIrqReg      0x05
#define ErrorReg       0x06
#define Status1Reg     0x07
#define Status2Reg     0x08
#define FIFODataReg    0x09
#define FIFOLevelReg   0x0A
#define WaterLevelReg  0x0B
#define ControlReg     0x0C
#define BitFramingReg  0x0D
#define CollReg        0x0E
#define Reserved01     0x0F
//Page 1:Command

```



```

#define Reserved10      0x10
#define ModeReg         0x11
#define TxModeReg       0x12
#define RxModeReg       0x13
#define TxControlReg    0x14
#define TxAutoReg       0x15
#define TxSelReg        0x16
#define RxSelReg        0x17
#define RxThresholdReg  0x18
#define DemodReg        0x19
#define Reserved11      0x1A
#define Reserved12      0x1B
#define MifareReg       0x1C
#define Reserved13      0x1D
#define Reserved14      0x1E
#define SerialSpeedReg  0x1F
//Page 2:CFG
#define Reserved20      0x20
#define CRCResultRegM   0x21
#define CRCResultRegL   0x22
#define Reserved21      0x23
#define ModWidthReg     0x24
#define Reserved22      0x25
#define RFCfgReg        0x26
#define GsNReg          0x27
#define CWGsPReg        0x28
#define ModGsPReg       0x29
#define TModeReg        0x2A
#define TPrescalerReg   0x2B
#define TReloadRegH     0x2C
#define TReloadRegL     0x2D
#define TCounterValueRegH 0x2E
#define TCounterValueRegL 0x2F
//Page 3:TestRegister
#define Reserved30      0x30
#define TestSel1Reg     0x31
#define TestSel2Reg     0x32
#define TestPinEnReg    0x33
#define TestPinValueReg 0x34
#define TestBusReg      0x35
#define AutoTestReg     0x36
#define VersionReg      0x37
#define AnalogTestReg   0x38
#define TestDAC1Reg     0x39
#define TestDAC2Reg     0x3A
#define TestADCReg      0x3B

```

```

#define Reserved31      0x3C
#define Reserved32      0x3D
#define Reserved33      0x3E
#define Reserved34      0x3F
//-----

class MFRC522 {
public:
    MFRC522(int sad, int reset);
    void writeToRegister(byte addr, byte val);
    byte readFromRegister(byte addr);
    void setBitMask(byte addr, byte mask);
    void clearBitMask(byte addr, byte mask);
    void begin();
    void reset();
    byte getFirmwareVersion();
    boolean digitalSelfTestPass();
    int commandTag(byte command, byte *data, int dlen, byte *result, int *rlen);
    int requestTag(byte mode, byte *type);
    int antiCollision(byte *serial);
    void calculateCRC(byte *data, int len, byte *result);
    byte selectTag(byte *serial);
    int authenticate(byte mode, byte block, byte *key, byte *serial);
    int readFromTag(byte blockAddr, byte *recvData);
    int writeToTag(byte blockAddr, byte *writeData);
    int haltTag();

private:
    int _sad, _reset;
};

```

### 7.3.2 Librería MFRC522.cpp

```

#include <Arduino.h>
#include <MFRC522.h>
MFRC522::MFRC522(int sad, int reset) {
    _sad = sad;
    pinMode(_sad, OUTPUT);    // Set digital as OUTPUT to connect it to the RFID /ENABLE pin
    digitalWrite(_sad, HIGH);

    _reset = reset;
    pinMode(_reset, OUTPUT);  // Set digital pin, Not Reset and Power-Down
    digitalWrite(_reset, HIGH);
}

```

```

void MFRC522::writeToRegister(byte addr, byte val) {
    digitalWrite(_sad, LOW);

    //Address format: 0XXXXXX0
    SPI.transfer((addr<<1)&0x7E);
    SPI.transfer(val);

    digitalWrite(_sad, HIGH);
}
byte MFRC522::readFromRegister(byte addr) {
    byte val;
    digitalWrite(_sad, LOW);
    SPI.transfer(((addr<<1)&0x7E) | 0x80);
    val =SPI.transfer(0x00);
    digitalWrite(_sad, HIGH);
    return val;
}

void MFRC522::setBitMask(byte addr, byte mask) {
    byte current;
    current = readFromRegister(addr);
    writeToRegister(addr, current | mask);
}
void MFRC522::clearBitMask(byte addr, byte mask) {
    byte current;
    current = readFromRegister(addr);
    writeToRegister(addr, current & (~mask));
}
void MFRC522::begin() {
    digitalWrite(_sad, HIGH);

    reset();

    //Timer: TPrescaler*TreloadVal/6.78MHz = 24ms
    writeToRegister(TModeReg, 0x8D);    // Tauto=1; f(Timer) = 6.78MHz/TPreScaler
    writeToRegister(TPrescalerReg, 0x3E); // TModeReg[3..0] + TPrescalerReg
    writeToRegister(TReloadRegL, 30);
    writeToRegister(TReloadRegH, 0);

    writeToRegister(TxAutoReg, 0x40);    // 100%ASK
    writeToRegister(ModeReg, 0x3D);     // CRC initial value 0x6363

    setBitMask(TxControlReg, 0x03);    // Turn antenna on.
}

```

```

void MFRC522::reset() {
    writeToRegister(CommandReg, MFRC522_SOFTRESET);
}

byte MFRC522::getFirmwareVersion() {
    byte response;
    response = readFromRegister(VersionReg);
    return response;
}

boolean MFRC522::digitalSelfTestPass() {
    int i;
    byte n;

    byte selfTestResultV1[] = {0x00, 0xC6, 0x37, 0xD5, 0x32, 0xB7, 0x57, 0x5C,
        0xC2, 0xD8, 0x7C, 0x4D, 0xD9, 0x70, 0xC7, 0x73,
        0x10, 0xE6, 0xD2, 0xAA, 0x5E, 0xA1, 0x3E, 0x5A,
        0x14, 0xAF, 0x30, 0x61, 0xC9, 0x70, 0xDB, 0x2E,
        0x64, 0x22, 0x72, 0xB5, 0xBD, 0x65, 0xF4, 0xEC,
        0x22, 0xBC, 0xD3, 0x72, 0x35, 0xCD, 0xAA, 0x41,
        0x1F, 0xA7, 0xF3, 0x53, 0x14, 0xDE, 0x7E, 0x02,
        0xD9, 0x0F, 0xB5, 0x5E, 0x25, 0x1D, 0x29, 0x79};
    byte selfTestResultV2[] = {0x00, 0xEB, 0x66, 0xBA, 0x57, 0xBF, 0x23, 0x95,
        0xD0, 0xE3, 0x0D, 0x3D, 0x27, 0x89, 0x5C, 0xDE,
        0x9D, 0x3B, 0xA7, 0x00, 0x21, 0x5B, 0x89, 0x82,
        0x51, 0x3A, 0xEB, 0x02, 0x0C, 0xA5, 0x00, 0x49,
        0x7C, 0x84, 0x4D, 0xB3, 0xCC, 0xD2, 0x1B, 0x81,
        0x5D, 0x48, 0x76, 0xD5, 0x71, 0x61, 0x21, 0xA9,
        0x86, 0x96, 0x83, 0x38, 0xCF, 0x9D, 0x5B, 0x6D,
        0xDC, 0x15, 0xBA, 0x3E, 0x7D, 0x95, 0x3B, 0x2F};

    byte *selfTestResult;
    switch(getFirmwareVersion()) {
        case 0x91 :
            selfTestResult = selfTestResultV1;
            break;
        case 0x92 :
            selfTestResult = selfTestResultV2;
            break;
        default:
            return false;
    }

    reset();
    writeToRegister(FIFODataReg, 0x00);
    writeToRegister(CommandReg, MFRC522_MEM);
    writeToRegister(AutoTestReg, 0x09);
    writeToRegister(FIFODataReg, 0x00);
}

```

```

writeToRegister(CommandReg, MFRC522_CALCCRC);

// Wait for the self test to complete.
i = 0xFF;
do {
    n = readFromRegister(DivIrqReg);
    i--;
} while ((i != 0) && !(n & 0x04));

for (i=0; i < 64; i++) {
    if (readFromRegister(FIFODataReg) != selfTestResult[i]) {
        Serial.println(i);
        return false;
    }
}
return true;
}

int MFRC522::commandTag(byte cmd, byte *data, int dlen, byte *result, int *rlen) {
    int status = MI_ERR;
    byte irqEn = 0x00;
    byte waitIRq = 0x00;
    byte lastBits, n;
    int i;

    switch (cmd) {
    case MFRC522_AUTHENT:
        irqEn = 0x12;
        waitIRq = 0x10;
        break;
    case MFRC522_TRANSCEIVE:
        irqEn = 0x77;
        waitIRq = 0x30;
        break;
    default:
        break;
    }

    writeToRegister(CommIEReg, irqEn|0x80); // interrupt request
    clearBitMask(CommIrqReg, 0x80); // Clear all interrupt requests bits.
    setBitMask(FIFOLevelReg, 0x80); // FlushBuffer=1, FIFO initialization.

    writeToRegister(CommandReg, MFRC522_IDLE); // No action, cancel the current command.

    // Write to FIFO
    for (i=0; i < dlen; i++) {
        writeToRegister(FIFODataReg, data[i]);
    }
}

```

```

}

// Execute the command.
writeToRegister(CommandReg, cmd);
if (cmd == MFRC522_TRANSCEIVE) {
    setBitMask(BitFramingReg, 0x80); // StartSend=1, transmission of data starts
}

// Waiting for the command to complete so we can receive data.
i = 25; // Max wait time is 25ms.
do {
    delay(1);
    // CommIrqReg[7..0]
    // Set1 TxIRq RxIRq IdleIRq HiAlerIRq LoAlertIRq ErrIRq TimerIRq
    n = readFromRegister(CommIrqReg);
    i--;
} while ((i!=0) && !(n&0x01) && !(n&waitIRq));

clearBitMask(BitFramingReg, 0x80); // StartSend=0

if (i != 0) { // Request did not time out.
    if(!(readFromRegister(ErrorReg) & 0x1B)) { // BufferOvfl Collerr CRCErr ProtocolErr
        status = MI_OK;
        if (n & irqEn & 0x01) {
            status = MI_NOTAGERR;
        }
    }

    if (cmd == MFRC522_TRANSCEIVE) {
        n = readFromRegister(FIFOLevelReg);
        lastBits = readFromRegister(ControlReg) & 0x07;
        if (lastBits) {
            *rlen = (n-1)*8 + lastBits;
        } else {
            *rlen = n*8;
        }
    }

    if (n == 0) {
        n = 1;
    }

    if (n > MAX_LEN) {
        n = MAX_LEN;
    }

    // Reading the recieved data from FIFO.
    for (i=0; i<n; i++) {

```

```

        result[i] = readFromRegister(FIFODataReg);
    }
} else {
    status = MI_ERR;
}
}
return status;
}

int MFRC522::requestTag(byte mode, byte *data) {
    int status, len;
    writeToRegister(BitFramingReg, 0x07); // TxLastBists = BitFramingReg[2..0]

    data[0] = mode;
    status = commandTag(MFRC522_TRANSCEIVE, data, 1, data, &len);

    if ((status != MI_OK) || (len != 0x10)) {
        status = MI_ERR;
    }

    return status;
}

int MFRC522::antiCollision(byte *serial) {
    int status, i, len;
    byte check = 0x00;

    writeToRegister(BitFramingReg, 0x00); // TxLastBits = BitFramingReg[2..0]

    serial[0] = MF1_ANTICOLL;
    serial[1] = 0x20;
    status = commandTag(MFRC522_TRANSCEIVE, serial, 2, serial, &len);
    len = len / 8; // len is in bits, and we want each byte.
    if (status == MI_OK) {
        // The checksum of the tag is the ^ of all the values.
        for (i = 0; i < len-1; i++) {
            check ^= serial[i];
        }
        // The checksum should be the same as the one provided from the
        // tag (serial[4]).
        if (check != serial[i]) {
            status = MI_ERR;
        }
    }
}

```

```

    return status;
}
void MFRC522::calculateCRC(byte *data, int len, byte *result) {
    int i;
    byte n;

    clearBitMask(DivIrqReg, 0x04); // CRCIrq = 0
    setBitMask(FIFOLevelReg, 0x80); // Clear the FIFO pointer

    //Writing data to the FIFO.
    for (i = 0; i < len; i++) {
        writeToRegister(FIFODataReg, data[i]);
    }
    writeToRegister(CommandReg, MFRC522_CALCCRC);

    // Wait for the CRC calculation to complete.
    i = 0xFF;
    do {
        n = readFromRegister(DivIrqReg);
        i--;
    } while ((i != 0) && !(n & 0x04)); //CRCIrq = 1

    // Read the result from the CRC calculation.
    result[0] = readFromRegister(CRCResultRegL);
    result[1] = readFromRegister(CRCResultRegM);
}
byte MFRC522::selectTag(byte *serial) {
    int i, status, len;
    byte sak;
    byte buffer[9];

    buffer[0] = MF1_SELECTTAG;
    buffer[1] = 0x70;
    for (i = 0; i < 5; i++) {
        buffer[i+2] = serial[i];
    }
    calculateCRC(buffer, 7, &buffer[7]);

    status = commandTag(MFRC522_TRANSCEIVE, buffer, 9, buffer, &len);

    if ((status == MI_OK) && (len == 0x18)) {
        sak = buffer[0];
    }
    else {
        sak = 0;
    }
}

```



```

return sak;
}

int MFRC522::authenticate(byte mode, byte block, byte *key, byte *serial) {
    int i, status, len;
    byte buffer[12];

    //Verify the command block address + sector + password + tag serial number
    buffer[0] = mode;    // 0th byte is the mode
    buffer[1] = block;   // 1st byte is the block to address.
    for (i = 0; i < 6; i++) { // 2nd to 7th byte is the authentication key.
        buffer[i+2] = key[i];
    }
    for (i = 0; i < 4; i++) { // 8th to 11th byte is the serial of the tag.
        buffer[i+8] = serial[i];
    }

    status = commandTag(MFRC522_AUTHENT, buffer, 12, buffer, &len);

    if ((status != MI_OK) || (!(readFromRegister(Status2Reg) & 0x08))) {
        status = MI_ERR;
    }

    return status;
}

int MFRC522::readFromTag(byte block, byte *result) {
    int status, len;

    result[0] = MF1_READ;
    result[1] = block;
    calculateCRC(result, 2, &result[2]);
    status = commandTag(MFRC522_TRANSCEIVE, result, 4, result, &len);

    if ((status != MI_OK) || (len != 0x90)) {
        status = MI_ERR;
    }

    return status;
}

int MFRC522::writeToTag(byte block, byte *data) {
    int status, i, len;
    byte buffer[18];

    buffer[0] = MF1_WRITE;
    buffer[1] = block;

```

```

calculateCRC(buffer, 2, &buffer[2]);
status = commandTag(MFRC522_TRANSCEIVE, buffer, 4, buffer, &len);

if ((status != MI_OK) || (len != 4) || ((buffer[0] & 0x0F) != 0x0A)) {
    status = MI_ERR;
}

if (status == MI_OK) {
    for (i = 0; i < 16; i++) {
        buffer[i] = data[i];
    }
    calculateCRC(buffer, 16, &buffer[16]);
    status = commandTag(MFRC522_TRANSCEIVE, buffer, 18, buffer, &len);

    if ((status != MI_OK) || (len != 4) || ((buffer[0] & 0x0F) != 0x0A)) {
        status = MI_ERR;
    }
}

return status;
}

int MFRC522::haltTag() {
    int status, len;
    byte buffer[4];

    buffer[0] = MF1_HALT;
    buffer[1] = 0;
    calculateCRC(buffer, 2, &buffer[2]);
    clearBitMask(Status2Reg, 0x08); // turn off encryption
    status = commandTag(MFRC522_TRANSCEIVE, buffer, 4, buffer, &len);

    return status;
}

```

### 7.3.3 Librería LiquidCrystal.h

```

#ifndef LiquidCrystal_h
#define LiquidCrystal_h

#include <inttypes.h>
#include "Print.h"

// commands

```

```

#define LCD_CLEARDISPLAY 0x01
#define LCD_RETURNHOME 0x02
#define LCD_ENTRYMODESET 0x04
#define LCD_DISPLAYCONTROL 0x08
#define LCD_CURSORSHIFT 0x10
#define LCD_FUNCTIONSET 0x20
#define LCD_SETCGRAMADDR 0x40
#define LCD_SETDRAMADDR 0x80

// flags for display entry mode
#define LCD_ENTRYRIGHT 0x00
#define LCD_ENTRYLEFT 0x02
#define LCD_ENTRYSHIFTINCREMENT 0x01
#define LCD_ENTRYSHIFTDECREMENT 0x00

// flags for display on/off control
#define LCD_DISPLAYON 0x04
#define LCD_DISPLAYOFF 0x00
#define LCD_CURSORON 0x02
#define LCD_CURSOROFF 0x00
#define LCD_BLINKON 0x01
#define LCD_BLINKOFF 0x00

// flags for display/cursor shift
#define LCD_DISPLAYMOVE 0x08
#define LCD_CURSORMOVE 0x00
#define LCD_MOVERIGHT 0x04
#define LCD_MOVELEFT 0x00

// flags for function set
#define LCD_8BITMODE 0x10
#define LCD_4BITMODE 0x00
#define LCD_2LINE 0x08
#define LCD_1LINE 0x00
#define LCD_5x10DOTS 0x04
#define LCD_5x8DOTS 0x00

class LiquidCrystal : public Print {
public:
    LiquidCrystal(uint8_t rs, uint8_t enable,
                 uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3,
                 uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7);
    LiquidCrystal(uint8_t rs, uint8_t rw, uint8_t enable,
                 uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3,
                 uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7);
    LiquidCrystal(uint8_t rs, uint8_t rw, uint8_t enable,

```

```

        uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3);
LiquidCrystal(uint8_t rs, uint8_t enable,
        uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3);

void init(uint8_t fourbitmode, uint8_t rs, uint8_t rw, uint8_t enable,
        uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3,
        uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7);

void begin(uint8_t cols, uint8_t rows, uint8_t charsize = LCD_5x8DOTS);

void clear();
void home();

void noDisplay();
void display();
void noBlink();
void blink();
void noCursor();
void cursor();
void scrollDisplayLeft();
void scrollDisplayRight();
void leftToRight();
void rightToLeft();
void autoscroll();
void noAutoscroll();

void setRowOffsets(int row1, int row2, int row3, int row4);
void createChar(uint8_t, uint8_t[]);
void setCursor(uint8_t, uint8_t);
virtual size_t write(uint8_t);
void command(uint8_t);

using Print::write;
private:
void send(uint8_t, uint8_t);
void write4bits(uint8_t);
void write8bits(uint8_t);
void pulseEnable();

uint8_t _rs_pin; // LOW: command. HIGH: character.
uint8_t _rw_pin; // LOW: write to LCD. HIGH: read from LCD.
uint8_t _enable_pin; // activated by a HIGH pulse.
uint8_t _data_pins[8];

uint8_t _displayfunction;
uint8_t _displaycontrol;

```

```

uint8_t _displaymode;

uint8_t _initialized;

uint8_t _numlines;
uint8_t _row_offsets[4];
};

```

### 7.3.4 Programa Principal

```

#include <MFRC522.h>
#include <SPI.h>
#define SAD 10
#define RST 5
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 2, 3, 4, 5);
MFRC522 nfc(SAD, RST);
int dato;
void setup() {
    SPI.begin();
    Serial.begin(9600);
    lcd.begin(16, 2);
    lcd.print("BIENVENIDO.");
    delay(1000);
    lcd.clear();
    nfc.begin();
}
byte keyA[6] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, };
byte keyB[6] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, };

void loop() {
    while(1){
        lcd.print("REGISTRATE");
        delay(2000);
        lcd.clear();
        byte status;
        byte data[MAX_LEN];
        byte serial[5];
        int i, j, pos;
        lcd.setCursor(0, 1);
        status = nfc.requestTag(MF1_REQIDL, data);
        if (status == MI_OK) {
            status = nfc.antiCollision(data);
            memcpy(serial, data, 5);
            for (int i = 0; i <3; i++) {
                Serial.print(serial[i]);
            }
        }
    }
}

```

```

    }
    Serial.println(serial[3]);
    }
    if (Serial.available()) {

    int c=Serial.read();
    if (c=='1') {
        lcd.print("REGISTRADO");
        delay(2000);}
    if (c=='2') {
        lcd.print("NO REGISTRADO");
        delay(2000);}
    }
    //delay(2000);
    lcd.clear(); nfc.haltTag(); //delay(2000);
    }}

```

## 7.4 Programa en visual estudio

### 7.4.1 Conexión con la base de datos

```

Imports MySql.Data
Imports MySql.Data.Types
Imports MySql.Data.MySqlClient

Module mod_conexion_global
    Public _cadena As String
    Public _conexion As New MySqlConnection
    Public _conexion1 As New MySqlConnection

    Public Function conexion_Global() As Boolean
        Dim estado As Boolean = True
        Try
            _cadena = ("server=localhost;user
id=root;password=1234;database=rfid_proyecto")
            '_cadena = ("server=192.168.43.17;user
id=invitado;password=123456;database=conteo")
            _conexion = New MySqlConnection(_cadena)
            _conexion1 = New MySqlConnection(_cadena)

            Catch ex As MySqlException
                MessageBox.Show(ex.Message)
                estado = False
            End Try
            Return estado

        End Function

    Public Sub cerrar()
        _conexion.Close()
        _conexion1.Close()

```

```
End Sub
```

```
End Module
```

## 7.4.2 Consulta a la base de datos

```
Imports MySql.Data
```

```
Imports MySql.Data.Types
```

```
Imports MySql.Data.MySqlClient
```

```
Module Mod_consulta
```

```
Private _adaptador As New MySqlDataAdapter
```

```
Public _dtsdatos As New DataSet
```

```
Public _dtvdatos As New DataView
```

```
Public _dtsdatos1 As New DataSet
```

```
Public _dtvdatos1 As New DataView
```

```
Public _dtsdatos2 As New DataSet
```

```
Public _dtvdatos2 As New DataView
```

```
Public Sub consulta_datos()
```

```
Try
```

```
    _dtsdatos.Reset()
```

```
    conexion_Global()
```

```
    _adaptador.SelectCommand = New MySqlCommand("select * from usuarios",  
_conexion)
```

```
    _adaptador.Fill(_dtsdatos)
```

```
    _dtvdatos.Table = _dtsdatos.Tables(0)
```

```
    _conexion.Open()
```

```
    _adaptador.SelectCommand.Connection = _conexion
```

```
    _adaptador.SelectCommand.ExecuteNonQuery()
```

```
Catch ex As MySqlException
```

```
    MessageBox.Show(ex.Message)
```

```
Finally
```

```
    cerrar()
```

```
End Try
```

```
End Sub
```

```
Public Sub consulta_datos1()
```

```
Try
```

```
    _dtsdatos1.Reset()
```

```
    conexion_Global()
```

```
    _adaptador.SelectCommand = New MySqlCommand("select * from usuarios  
where codigo='" + _serial + "';", _conexion)
```

```
    _adaptador.Fill(_dtsdatos1)
```

```
    _dtvdatos1.Table = _dtsdatos1.Tables(0)
```

```
    _conexion.Open()
```

```
    _adaptador.SelectCommand.Connection = _conexion
```

```
    _adaptador.SelectCommand.ExecuteNonQuery()
```

```
Catch ex As MySqlException
```

```
    MessageBox.Show(ex.Message)
```

```
Finally
```

```
    cerrar()
```

```
End Try
```

```

End Sub
Public Sub consulta_datos2()
    Try
        _dtsdatos2.Reset()
        conexion_Global()
        _adaptador.SelectCommand = New MySqlCommand("select * from registro
where nombre='" + _busdoc + "';", _conexion)
        _adaptador.Fill(_dtsdatos2)
        _dtvdatos2.Table = _dtsdatos2.Tables(0)
        _conexion.Open()
        _adaptador.SelectCommand.Connection = _conexion
        _adaptador.SelectCommand.ExecuteNonQuery()
    Catch ex As MySqlException
        MessageBox.Show(ex.Message)
    Finally
        cerrar()
    End Try
End Sub
End Module

```

### 7.4.3 Variables

```

Imports MySql.Data
Imports MySql.Data.Types
Imports MySql.Data.MySqlClient
Module Mod_var
    Public _serial As String
    Public _serial2 As String
    Public _comprobacion As String
    Public _fecha As String
    Public _intco As String
    Public _intnom As String
    Public _busdoc As String

    Public Sub _variable()
        conexion_Global()
        _conexion.Open()
        Dim consulta2 As String = ("select codigo from usuarios where codigo='" +
_serial + "';")
        Dim cmd As New MySqlCommand(consulta2, _conexion)
        Dim dr As MySqlDataReader = cmd.ExecuteReader
        If dr.Read() Then
            _comprobacion = dr("codigo").ToString()
        End If
        _conexion.Close()

    End Sub
End Module

```

### 6.4.4 Clase de datos para la tabla usuarios de la base de datos

```

Public Class Class_datos
    Private _codigo As String

```



```

Private _nombre As String
Private _apellido As String
Private _numero As Integer

Private _docente_iddocente As Integer

Public Property codigo() As String
    Get
        Return _codigo
    End Get
    Set(value As String)
        _codigo = value
    End Set
End Property

Public Property nombre() As String
    Get
        Return _nombre
    End Get
    Set(value As String)
        _nombre = value
    End Set
End Property

Public Property apellido() As String
    Get
        Return _apellido
    End Get
    Set(value As String)
        _apellido = value
    End Set
End Property

Public Property numero() As Integer
    Get
        Return _numero
    End Get
    Set(value As Integer)
        _numero = value
    End Set
End Property

Public Property docente_iddocente() As Integer
    Get
        Return _docente_iddocente
    End Get
    Set(value As Integer)
        _docente_iddocente = value
    End Set
End Property
End Class

```

## 7.4.5 Clase de datos para la tabla registro de la base de datos

```
Public Class Class_datos1

    Private _codigo1 As String
    Private _nombre1 As String
    Private _registro As String

    Public Property codigo1() As String
        Get
            Return _codigo1
        End Get
        Set(value As String)
            _codigo1 = value
        End Set
    End Property

    Public Property nombre1() As String
        Get
            Return _nombre1
        End Get
        Set(value As String)
            _nombre1 = value
        End Set
    End Property

    Public Property registro() As String
        Get
            Return _registro
        End Get
        Set(value As String)
            _registro = value
        End Set
    End Property

End Class
```

## 7.4.5 Clase para ingresar datos a la tabla usuarios de la base de datos

```
Imports MySql.Data
Imports MySql.Data.Types
Imports MySql.Data.MySqlClient

Public Class Class_int

    Private _adaptador As New MySqlDataAdapter

    Public Function insertardatos(ByVal datos As Class_datos) As Boolean
        Dim estado As Boolean = True
        Try
            conexion_Global()
        End Try
    End Function

End Class
```

```

        _adaptador.InsertCommand = New MySqlCommand("insert into usuarios
(codigo,nombre, apellido, numero) values (@codigo, @nombre, @apellido, @numero)",
_conexion)
        _adaptador.InsertCommand.Parameters.Add("@codigo", MySqlDbType.VarChar,
15).Value = datos.codigo
        _adaptador.InsertCommand.Parameters.Add("@nombre", MySqlDbType.VarChar,
20).Value = datos.nombre
        _adaptador.InsertCommand.Parameters.Add("@apellido",
MySqlDbType.VarChar, 20).Value = datos.apellido
        _adaptador.InsertCommand.Parameters.Add("@numero",
MySqlDbType.Int32).Value = datos.numero
        _conexion.Open()
        _adaptador.InsertCommand.Connection = _conexion
        _adaptador.InsertCommand.ExecuteNonQuery()
    Catch ex As MySqlException
        MessageBox.Show(ex.Message)
        estado = False
    Finally
        cerrar()
    End Try

```

#### 7.4.6 Clase para ingresar datos a la tabla registro de la base de datos

```

Imports MySql.Data
Imports MySql.Data.Types
Imports MySql.Data.MySqlClient
Public Class Class_int1
    Private _adaptador As New MySqlDataAdapter
    Public Function insertardatos1(ByVal datos1 As Class_datos1) As Boolean
        Dim estado As Boolean = True
        Try
            conexion_Global()
            _adaptador.InsertCommand = New MySqlCommand("insert into registro
(codigo,nombre, registro) values (@codigo, @nombre, @registro)", _conexion)
            _adaptador.InsertCommand.Parameters.Add("@codigo", MySqlDbType.VarChar,
15).Value = datos1.codigo1
            _adaptador.InsertCommand.Parameters.Add("@nombre", MySqlDbType.VarChar,
20).Value = datos1.nombre1
            _adaptador.InsertCommand.Parameters.Add("@registro",
MySqlDbType.VarChar, 20).Value = datos1.registro
            _adaptador.InsertCommand.Parameters.Add("@numero",
MySqlDbType.Int32).Value = datos1.numero
            _conexion.Open()
            _adaptador.InsertCommand.Connection = _conexion
            _adaptador.InsertCommand.ExecuteNonQuery()
        Catch ex As MySqlException
            MessageBox.Show(ex.Message)
            estado = False
        Finally
            cerrar()
        End Try
        Return estado
    End Function
End Class

```

### 7.4.7 Form1.vs Registro Automatizado de Asistencia

Este Form es el menú principal de la aplicación donde se hace la conexión serial con la placa lectora.



Figura 10: pantallazo del form1.vs en visual

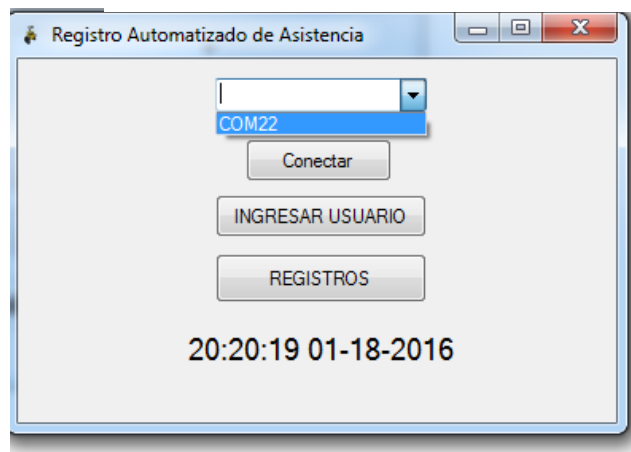


Figura 11: pantallazo Form.vs ejecutado en visual

#### 7.4.7.1 Código del Form1.vs Registro Automatizado de Asistencia

```
Imports MySql.Data
Imports MySql.Data.Types
Imports MySql.Data.MySqlClient
```

```
Public Class Form1
```

```
Dim bufferout As String
Dim bufferin As String
```

```
Private Delegate Sub Delegadoacceso(ByVal addtxt As String)
```

```
Private Sub accesoprincipal(ByVal textoform As String)
```

```
bufferin = textoform
TextBox1.Text = bufferin
```

```

TextBox4.Text = bufferin
_serial = TextBox1.Text.Replace(vbCrLf, " ")
_serial2 = TextBox4.Text

_variable()
TextBox3.Text = _comprobacion
consulta_datos1()
DataGridView1.DataSource = _dtvdatos1

If _serial = _comprobacion Then

    For Each fila As DataRowView In DataGridView1.DataSource
        'La columna 1 de grilla la paso al textbox1
        _intco = fila.Item(0).ToString
        'La columna 2 de grilla la paso al textbox2
        _intnom = fila.Item(1).ToString

    Next
    _variable()
    Dim conexion As New Class_int1
    Dim datos1 As New Class_datos1

    datos1.codigo1 = _intco
    datos1.nombre1 = _intnom
    datos1.registro = _fecha

    '_dtvdatos.Table = _dtsdatos.Tables(0)

    If conexion.insertardatos1(datos1) Then
        SerialPort1.Write("1")
        'MessageBox.Show("Datos Guardados")

    Else
        SerialPort1.Write("2")
        'MessageBox.Show("Datos No Guardados")

    End If
    consulta_datos1()
End If
If _serial <> _comprobacion Then
    SerialPort1.Write("2")
    'MessageBox.Show("Datos No Guardados")
End If
End Sub
'-----
'-----
Private Sub puertaaccesoint(ByVal entrada As String)
    Dim textoint() As Object = {entrada}
    Dim delegadoint As Delegadoacceso
    delegadoint = New Delegadoacceso(AddressOf accesoprincipal)
    MyBase.Invoke(delegadoint, textoint)
End Sub
'-----
'-----

```

```

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Timer1.Start()
    bufferin = ""
    bufferout = ""
    Button2.Enabled = False
    ComboBox1.Items.Clear()
    For Each puertosdisponibles As String In My.Computer.Ports.SerialPortNames
        ComboBox1.Items.Add(puertosdisponibles)
    Next

End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click

    If Button1.Text = "Conectar" Then

        Try
            With SerialPort1
                Dim ComPort As New System.IO.Ports.SerialPort()
                .PortName = ComboBox1.Text
                .BaudRate = 9600
                .DataBits = 8
                .Parity = IO.Ports.Parity.None
                .StopBits = IO.Ports.StopBits.One
                .DtrEnable = True
                .Open()

            End With
            Button1.Text = "Desconectar"
            Button2.Enabled = True

        Catch ex As Exception
            MsgBox(ex.Message, MsgBoxStyle.Critical)
        End Try
    ElseIf Button1.Text = "Desconectar" Then
        Button1.Text = "Conectar"
        Button2.Enabled = False
        SerialPort1.Close()
    End If

End Sub

Private Sub datorecibido(sender As Object, e As
IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived

    Dim datoint As String
    datoint = SerialPort1.ReadExisting
    puertaaccesoint(datoint)

End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
    SerialPort1.DiscardOutBuffer()

```

```

        bufferout = TextBox2.Text
        SerialPort1.Write(bufferout)
    End Sub
    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
        TextBox1.Text = " "
        bufferin = " "
        Form2.Show()

    End Sub
    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        Label1.Text = TimeString
        Label2.Text = DateString
        Label3.Text = Label1.Text + " " + Label2.Text
        _fecha = TimeString + " " + DateString

    End Sub
    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
Button4.Click
        Form3.Show()
    End Sub
End Class

```

## 7.4.8 Form2.vs Ingresar Usuario

Este Form2.vs es una ventana donde podemos agregar o eliminar usuarios.

	codigo	NOMBRE	APELLIDO	NUMERO
▶	11323522930	gregorio	perez	74748484
	1364223190	ANDREA	PORTELA	5366373
*				

Figura 12: Pantallazo de Form2.vs ingresar usuario en visual

### 7.4.8.1 Código del Form2.vs Ingresar Usuario

```

Imports MySql.Data
Imports MySql.Data.Types
Imports MySql.Data.MySqlClient
Public Class Form2

    Private Sub Form2_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        conexion_Global()
        consulta_datos()
        DataGridView1.DataSource = _dtvdatos
    End Sub

```

```

End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click

    Dim conexion As New Class_int
    Dim datos As New Class_datos
    datos.codigo = TextBox1.Text
    datos.nombre = TextBox2.Text
    datos.apellido = TextBox3.Text
    datos.numero = TextBox4.Text
    '_dtvdatos.Table = _dtsdatos.Tables(0)

    If conexion.insertardatos(datos) Then
        MessageBox.Show("Datos Guardados")
    Else
        MessageBox.Show("Datos No Guardados")
    End If
    consulta_datos()
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click

    Try
        If DataGridView1.Rows.Count > 0 Then
            If DataGridView1.SelectedRows.Count > 0 Then
                Dim intid As Integer =
DataGridview1.SelectedRows(0).Cells("codigo").Value
                If Not _conexion.State = ConnectionState.Open Then
                    _conexion.Open()
                End If
                Dim cmd As New MySqlCommand
                cmd.Connection = _conexion
                cmd.CommandText = "delete from usuarios where codigo=" & intid
                cmd.ExecuteNonQuery()
                MessageBox.Show("estas seguro", "datos irrecuperable",
MessageBoxButtons.OK, MessageBoxIcon.Warning)

                consulta_datos()
                _conexion.Open()
            End If

        End If

    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try

End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
    Me.Close()

```



End Sub

```
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles  
Button4.Click  
    TextBox1.Text = _serial  
End Sub  
End Class
```

## 7.4.9 Form3.vs Registro

Este Form3.vs es una ventana donde podemos buscar los registros de los usuarios y si es necesario tiene la opción de llevar los datos a WORD.

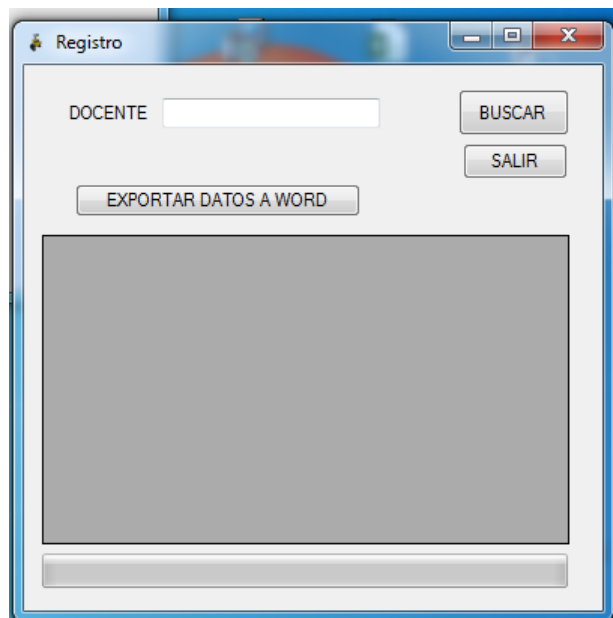


Figura 13: Pantallazo de Form3.vs Registro en Visual

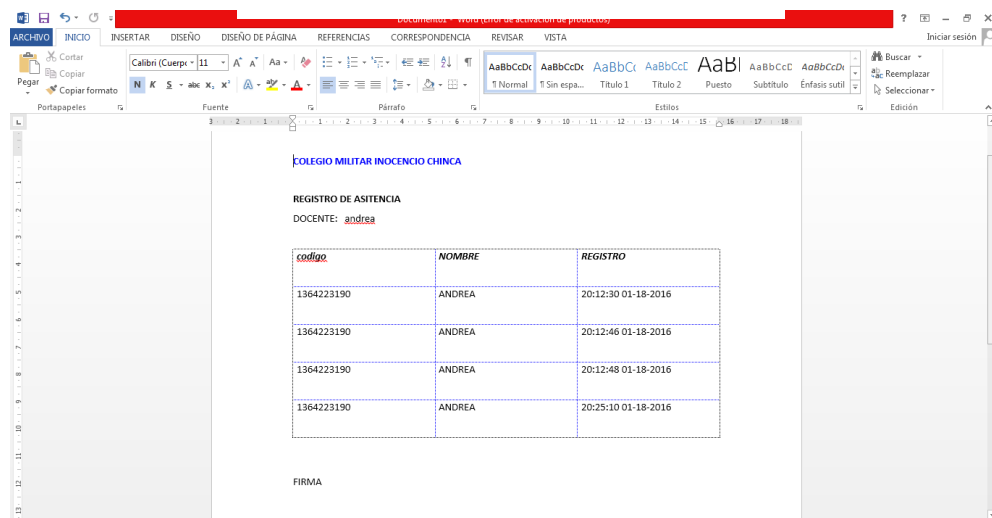


Figura 14: Pantallazo de Exportación de datos a WORD

### 7.4.9.1 Código Form3.vs Registro

```
Imports MySql.Data
Imports MySql.Data.Types
Imports MySql.Data.MySqlClient
Imports Microsoft.Office.Interop.Word
Public Class Form3

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        _busdoc = TextBox1.Text
        consulta_datos2()
        DataGridView1.DataSource = _dtvdatos2
    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        Me.Close()
    End Sub
    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
        Call GrillaAWord(DataGridView1)
    End Sub
    Public Function GrillaAWord(ByVal Grilla As DataGridView)
        Dim Word As Application
        Dim Doc As Document
        Dim Table As Table
        Dim Rng As Range
        Dim Prf1 As Paragraph
        Dim Prf2 As Paragraph
        Dim Prf3 As Paragraph

        Word = CreateObject("Word.Application")
        Word.Visible = False
        Doc = Word.Documents.Add

        Prf1 = Doc.Content.Paragraphs.Add
        Prf1.Range.Text = "COLEGIO MILITAR INOCENCIO CHINCA"
        Prf1.Range.Font.Color = WdColor.wdColorBlue
        Prf1.Range.Font.Bold = True
        Prf1.Format.SpaceAfter = 24
        Prf1.Range.InsertParagraphAfter()

        Prf2 = Doc.Content.Paragraphs.Add(Doc.Bookmarks.Item("\endofdoc").Range)
        Prf2.Range.Text = "REGISTRO DE ASITENCIA "
        Prf2.Range.Font.Color = WdColor.wdColorBlack
        Prf2.Format.SpaceAfter = 6
        Prf2.Range.InsertParagraphAfter()

        Prf3 = Doc.Content.Paragraphs.Add(Doc.Bookmarks.Item("\endofdoc").Range)
        Prf3.Range.Text = "DOCENTE: " & TextBox1.Text & " "
        Prf3.Range.Font.Bold = False
        Prf3.Format.SpaceAfter = 24
        Prf3.Range.InsertParagraphAfter()
        Dim NCol As Integer = Grilla.ColumnCount
        Dim NRow As Integer = Grilla.RowCount
    End Function
End Class
```

```

'alternativo
ProgressBar1.Maximum = NRow - 1
Table = Doc.Tables.Add(Doc.Bookmarks.Item("\endofdoc").Range,
Grilla.RowCount, Grilla.ColumnCount)
'Agregando Los Campos De La Grilla
For i As Integer = 1 To NCol
    Table.Cell(1, i).Range.Text = Grilla.Columns(i - 1).Name.ToString
Next
'Agregando Los Registros A La Tabla
For Fila As Integer = 0 To NRow - 2
    For Col As Integer = 0 To NCol - 1
        If Grilla.Rows(Fila).Cells(Col).Value IsNot DBNull.Value Then
            Table.Cell(Fila + 2, Col + 1).Range.Text =
Grilla.Rows(Fila).Cells(Col).Value
        End If
    Next
    'Incremento
    ProgressBar1.Increment(1)
Next
'Negrita y Kursiva Para Los Nombres De Los Campos
Table.Rows.Item(1).Range.Font.Bold = True
Table.Rows.Item(1).Range.Font.Italic = True
'Boder De La Tabla
Table.Borders.InsideLineStyle = WdLineStyle.wdLineStyleDot
Table.Borders.OutsideLineStyle = WdLineStyle.wdLineStyleDot
Table.Borders.InsideColor = WdColor.wdColorBlue
Rng = Doc.Bookmarks.Item("\endofdoc").Range
Rng.InsertParagraphAfter()
Rng.InsertAfter("FIRMA")
MessageBox.Show("Exportacion Completado.. ")
Word.Visible = True
End Function
End Class

```

## 6.5 Manual de Usuario



Figura 15: Portada del Manual de usuario Pág. 1

## Contenido

Introducción .....	3
Modulo Lector .....	4
Especificaciones .....	4
Software .....	4
Menu Principal .....	5
Aggrar Usuario .....	5
Borrar Usuario .....	6
Ver los registros .....	6

## Introducción

Este programa se a diseñado para su uso como aplicación de registro automatico de asistencia mediante un modulo lector que lleva la información a al misma



Figura 16: Manual Usuario Pág. 2 y 3

### Modulo lector



### ESPECIFICACIONES

Tensión de funcionamiento .....	5v
Velocidad de reloj .....	16MHz
RFID RC522 .....	3.3v
LCD 16x2 .....	5v
Protocolo de Comunicación .....	Serial

### Software



### Menu Principal

Primero que todo debemos conectar con el dispositivo para ello seleccionamos el puerto en el que se encuentra como se apresia en la siguiente imagen.



Luego de estar cocentado el dispositivo, solo es pasar las tarjetas Mifare por el modulo lector

### Agregar Usuario

Para agregar un usuario debemos hacer click en el boton al ingr-sar nos aparecera la siguiente ventana.




Figura 17: Manual de Usuario Pág. 4 y 5

Ingresamos los datos del usuario pedido, luego pasamos la tarjeta del usuario por le modulo lector y le damos click en el boton lector.



### Borrar Usuario

Seleccionamos en la parte de abajo el usuario a eliminar como se muestra en la imagen.

Luego de tenerla seleccionada damos click en el boto de eliminar. Para volver el menu principal damos click en el boton SALIR

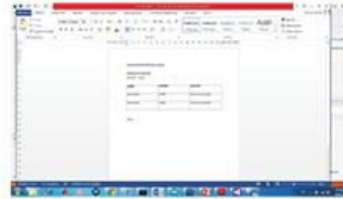
### Ver los Registros

Para ver los registros damos click en el boton registro y nos aparecera la seiguiente imagen.

6



Ingresamos el nombre del usuario a buscar, los registros aparecern en el cuadro de abajo y si se quiere expandir a work solo debemos hacer click en EXPORTAR DATOS A WORD



7

Figura 18: Manual de Usuario Pág. 6 y 7

## 8. CONCLUSIONES

- Se pudo observar que para tener una buena lectura la distancia óptima no debe superar los 5cm aunque las especificaciones técnicas de los fabricantes dicen 6cm. Dado a que a esta distancia no se tiene lectura de la tarjeta.
- El método de registro automatizado que se implemento es mucho más confiable y seguro que el registro manual, porque la información queda soportada en una base de datos minimizando el riesgo a que se pierda o se altere los datos.
- Para que haya una buena comunicación entre el módulo de lectura y el ordenador se debe especificar una misma frecuencia de funcionamiento ya sea 115200Hz o 9600Hz.

## 9. BIBLIOGRAFIA

- ALVARADO SÁNCHEZ, Jorge Alberto. Sistema de control de Acceso con RIFD. Trabajo de Grado Maestría en ciencias, Especialidad de Ingeniería Electrónica. México D. F.: Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. 2008. Pág 120.
- Arduino. [en línea] Arduino Leonardo. disponible en [www.arduino.cc/en/Main/ArduinoBoardLeonardo](http://www.arduino.cc/en/Main/ArduinoBoardLeonardo)
- Miguel Balboa,[en línea] Arduino RFID Library for MFRC522, disponible en [www.github.com/miguelbalboa/rfid](https://github.com/miguelbalboa/rfid)
- ProgsisdelAdvanced,[en línea] Comunicación Serial con Visual Basic 2013 " Parte I " - Serial Port Visual Basic 2013 - I , Disponible en <https://www.youtube.com/watch?v=AwaKnxzEYak>, 09 de DIC del 2015,