

**IMPLEMENTACION DE SISTEMA RFID PARA REGISTRO DE COMPONENTES  
DEL LABORATORIO DE ELECTRONICA DE LA UPB**

**CARLOS ALBERTO VELASCO GIL**

**OSCAR ANDRES LOPEZ ANAYA**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
ESCUELA DE INGENIERIAS Y ADMINISTRACION  
FACULTAD DE INGENIERIA ELECTRONICA  
BUCARAMANGA**

**2009**

**IMPLEMENTACION DE SISTEMA RFID PARA REGISTRO DE COMPONENTES  
DEL LABORATORIO DE ELECTRONICA DE LA UPB**

**CARLOS ALBERTO VELASCO GIL**

**OSCAR ANDRES LOPEZ ANAYA**

**Trabajo de Grado para optar al título de  
Ingeniero Electrónico**

**Director:**

**REYNALDO CLAROS**

**Ing. Electrónico**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
ESCUELA DE INGENIERIAS Y ADMINISTRACION  
FACULTAD DE INGENIERIA ELECTRONICA  
BUCARAMANGA**

**2009**

**Nota de aceptación**

---

---

---

---

Firma del jurado

---

Firma del jurado

Bucaramanga 23 de julio de 2009

A Dios por darme la oportunidad de existir y tener a mi lado todas las ayudas necesarias para poder crecer y formarme como persona de bien, a mis padres, mi más grande y valioso tesoro que sin su ayuda, ejemplo de pujanza, esfuerzo y dedicación no hubiese sido posible la culminación de mi carrera profesional, y con quienes estaré eternamente agradecido.

A mis dos hermanas quienes siempre sin dudarlo han estado a mi lado dándome el apoyo necesario para realizar y cumplir cada uno de los retos puestos en mi vida.

Gracias

Carlos Alberto Velasco Gil.

De antemano agradezco a Dios por permitirme culminar esta etapa de mi vida, a mi madre y a mi hermana por el apoyo incondicional que siempre me han brindado durante toda mi vida.

De la misma manera doy gracias al Ingeniero Reynaldo Claros por el apoyo y la orientación durante todo este proceso.

Gracias a Todos.

Oscar Andrés López Anaya

## **AGRADECIMIENTOS**

Los autores expresamos agradecimientos a:

Al Ingeniero Reynaldo Claros, director del proyecto, por su orientación y porque nos dio la libertad y confianza necesaria para la realización del proyecto.

A la universidad por todos los años de una formación profesional integra.

## INDICE DE CONTENIDO

	Pág.
1. INTRODUCCION.....	1
2. OBJETIVOS.....	2
2.1 Objetivo General.....	2
2.2 Objetivos específicos.....	2
3 MARCOTEORICO.....	3
3.1 RFID en sus inicios.....	3
3.2 RFID identificación por radiofrecuencia.....	4
3.2.1Clasificación de los Tags por su frecuencia operativa.....	6
3.2.2 Sistema anticolisión.....	8
3.2.3Conectividad.....	9
3.2.4 seguridad de sistema RFID.....	10
4. MARCO CONTEXTUAL.....	11
4.1 Aplicaciones.....	11
4.2 RFID VS CÓDIGO DE BARRAS.....	12
4.2.1 Ventajas del RFID.....	12
4.2.2 Comparación entre RFID y el código de barras.....	13
5. DISEÑO DEL SISTEMA UTILIZADO .....	14
5.1. Diseño de infraestructura.....	15
5.2 Arquitectura Implementada.....	17
6. HARDWARE.....	19

6.1 Descripción del sistema RFID utilizado.....	19
6.1.1 TRANSPONDEDOR (Tag).....	20
6.1.2 Lector de RFID.....	20
7. DISEÑO DEL SOFTWARE.....	21
7.1 Diseño de la Base de Datos.....	22
7.1.1 Diagrama entidad-relación para la base de datos.....	22
7.1.2. Plantilla o vista en visual.....	25
7.1.3. Diseño de la ventana laboratorio.....	28
7.1.4. Diseño de la ventana administrar.....	31
7.1.5 Diseño de la ventana consultar .....	34
7.1.6 Descripción del proceso de envío de datos.....	38
8. RESULTADOS.....	39
9. CONCLUSIONES.....	40
10. RECOMENDACIONES.....	42
11. BIBLIOGRAFIA.....	43
11.1 Tomados de Internet.....	43
11.2 Libros.....	44

## INDICE DE FIGURAS

	Pág.
Figura 1. funcionamiento de un sistema RFID .....	4
Figura 2. Figura de frecuencias de operación de los Tag.....	7
Figura 3. Diseño de Tags utilizados .....	15
Figura 4. Lector utilizado.....	16
Figura 5. Logotipo de Visual Basic.Net.....	17
Figura 6. Modelo de arquitectura.....	18
Figura 7. Imagen del Kit.....	20
Figura 8. Imagen del lector.....	21
Figura 9. Diagrama entidad-relación.....	23
Figura 10. Tabla estudiantes.....	24
Figura 11. Tabla instrumentos.....	24
Figura 12. Tabla carreras.....	25
Figura 13. Tabla Laboratorios.....	25
Figura 14. Ventana de contraseña.....	26
Figura 15. Código venta contraseña.....	26
Figura 16. Ventana principal.....	27
Figura 17. Código para la ventana principal.....	28
Figura 18. Ventana laboratorios I.....	28
Figura 19. Ventana laboratorios II.....	29



Figura 20. Descripción prestamos.....	30
Figura 21. Ventana descripción de los instrumentos prestados.....	30
Figura 22. Identificación del botón listo.....	31
Figura 23. Descripción del código de la ventana laboratorios.....	31
Figura 24. Ventana Administrar.....	32
Figura 25. GroupBox “Estudiantes” o GroupBox “Equipos”.....	32
Figura 26. Barra de opciones.....	33
Figura 27. Código ventana administrar.....	34
Figura 28. Ventana Consultar.....	34
Figura 29. Paleta Filtrar.....	35
Figura 30. Filtro por Código de Estudiante.....	35
Figura 31. Resultado después de filtrar.....	36
Figura 32. Filtro por Fecha, calendario “MonthCalendar” .....	36
Figura 33. Identificación de la opción préstamos” .....	37
Figura 34. Código Ventana Consultar.....	38

## INDICE DE TABLAS

Pág.

TABLA1. Características de los rangos de frecuencia.....	8
TABLA2. Comparación entre RFID y código de barras.....	14

## ÍNDICE DE ANEXOS

<b>ANEXO 1.</b> Programación con Visual Basic.Net.....	46
<b>ANEXO 2.</b> APSX RFID Reader/Writer Data Sheet V1.6.....	64

## **RESUMEN GENERAL DE TRABAJO DE GRADO**

**TITULO:** IMPLEMENTACION DE SISTEMA RFID PARA REGISTRO DE COMPONENTES DEL LABORATORIO DE ELECTRONICA DE LA UPB

**AUTORES:** CARLOS ALBERTO VELASCO GIL  
OSCAR ANDRES LOPEZ ANAYA

**FACULTAD:** Ingeniería Electrónica

**DIRECTOR:** REYNALDO CLAROS

### **RESUMEN**

La tecnología RFID es un sistema de comunicación donde se implementa la auto identificación de manera inalámbrica, está compuesto de etiquetas que almacenan información y de un sistema de lectura que permite la identificación de la información de estas etiquetas a diferentes distancias. Por sus grandes beneficios la tecnología RFID ha sido adoptada como nuevo medio de identificación en distintos sectores como control de inventarios, suministros y transportes.

Teniendo en cuenta las características del sistema de comunicación RFID en este proyecto de grado se realizara la implementación de este sistema en el laboratorio de la facultad de ingeniería electrónica así permitiendo una agilidad y calidad del servicio en el laboratorio en el momento de la solicitud del préstamo de los equipos. Con la ayuda de un software que contara con el acople necesario par manejar las etiquetas se podrá observar la base de datos que le permitirá al laboratorio tener un control en el registro de los equipos prestados y indicará el mantenimiento de los mismos.

**PALABRAS CLAVES:** RFID, software, hardware

## **GENERAL SUMMARY OF WORK OF DEGREE**

**TITLE:** RFID IMPLEMENTATION OF SYSTEM FOR RECORDING OF ELECTRONIC COMPONENTS LABORATORY OF UPB

**AUTHORS:** CARLOS ALBERTO VELASCO GIL  
OSCAR ANDRES LOPEZ ANAYA

**FACULTY:** Electronic Engineering

**DIRECTOR:** REYNALDO CLAROS

### **ABSTRACT**

RFID technology is a communication system that implements a self-wireless, consists of tags that store information and a reading system that enables the identification of information in these tags at different distances. Because of its great benefits RFID technology has been adopted as a new means of identification in various sectors such as inventory control, supplies and transportation.

Taking into account the characteristics of the RFID communication system in this project will undertake the degree of implementation of this system in the laboratory of the faculty of electrical engineering and enabling agility and service quality in the laboratory at the time of loan application of equipment. With the help of software without the need to attach labels to handle the torque you can see the database that will allow the laboratory to control the registration of the equipment provided and indicate the maintenance.

**KEY WORDS:** RFID, software, hardware

## 1. INTRODUCCION

Con el pasar del tiempo ha crecido la tendencia en el registro y control de personal y objetos en diversas instituciones, llevando esto la investigación y el desarrollo de tecnologías importantes ya sea para control interno o externo.

La investigación muestra que contar con un control de los instrumentos en las instituciones, de una manera mas personalizada puede llegar a mitigar problemas y brindar mejor eficiencia en el momento que se desee tener una clara base de datos de los equipos con los que cuenta en ese momento.

Para poder tener control de las funciones del Laboratorio de la Facultad de Ingeniería Electrónica en el préstamo y el inventario de los equipos con los que se cuenta en ese momento, es posible la implementación de un sistema RFID (identificación por radio frecuencia). Un sistema RFID se basa en Transpondedores (etiquetas o Tags) y lectores que realizan una comunicación entre si, permitiendo una seguridad y un registro del equipo que contenga el Transponedor (etiqueta o Tags).

Con la implementación de esta tecnología se observa un beneficio en la disminución de los materiales de registro como el papel. De esta forma, se optimizan costos y tiempo empleados en el préstamo de los materiales. Para conseguir esto, se implementa un software con una base de datos de los materiales con los que cuenta el laboratorio de la Facultad de Ingeniería Electrónica. El sistema de comunicación RFID que se implantará esta compuesto de una antena de recepción y un emisor que estará ubicado en los instrumentos destinados para el préstamo.

# **1 OBJETIVOS**

## **1.1 OBJETIVO GENERAL**

Implementar la tecnología RFID en la universidad para agilizar el proceso de préstamos de equipos en el laboratorio de la Facultad de Ingeniería Electrónica.

## **1.2 OBJETIVOS ESPECÍFICOS**

- Determinar cuál es el mejor sistema de comunicación RFID a usar en el laboratorio de la facultad de ingeniería electrónica.
- Diseñar un software de gestión que se integre con el hardware seleccionado para optimizar el trabajo realizado en el préstamo de los equipos.

## 2 MARCO TEORICO

### 2.1 RFID EN SUS INICIOS [1]

La tecnología RFID ha tenido diferentes cambios en su forma de utilizar y avances en el desarrollo para brindar mejor y mas calidad en sus funciones. Para tener un precedente de sus primeras funciones citaremos sus inicios a lo largo de la historia.

Iniciando con la de cada de los años 40 durante la segunda guerra mundial donde fue implantada la tecnología por parte de EE.UU. para conseguir un control de sus aviones y obteniendo así un reconocimiento a distancias de los suyos o de los enemigos.

Seguida de esta etapa donde fue utilizado en guerra se observa su funcionalidad en otros aspectos de la vida y es aplicada en el año 69 por el señor Mario Cardillo en cuatro áreas como transporte, medicina y seguridad.

Entrando en los años 70 se observa su utilización en el sector industrial, el cual maneja un sistema de seguridad que deseaba ser modificado y se aplica el RFID encontrándose que es manejado por este sector pues los costos de implementación de esta tecnología eran altos y era el único sector capaz de aplicarlo.

Con la llegada de los años 80 se observa la incorporación de la tecnología RFID en Europa encontrando su implementación en el área de la ganadería para el control e identificación de la posición de los animales. Tiempo después es utilizada en el sector automotriz.

Observando la funcionalidad que se le ha dado a lo largo de los años, en la década de los 90 se observa como la tecnología RFID empieza a tomar mayor peso en diferentes empresas, ya que los costos de esta tecnología son disminuidos por IBM realizando un prototipo que integra todo el circuito en un solo chip.



## 2.2 RFID IDENTIFICACION POR RADIOFRECUENCIA

RFID identificación por radiofrecuencia es una tecnología de identificación remota e inalámbrica esta compuesta por dos partes importantes (Figura1) [1]:

Lector o Readers: el encargado de leer la información enviada por los transpondedores y también pueden transmitir información a los mismos.

Transpondedor (Tag): emite una señal mediante ondas de radio que son leídas por el lector y se encuentran ubicados sobre el objeto a identificar.

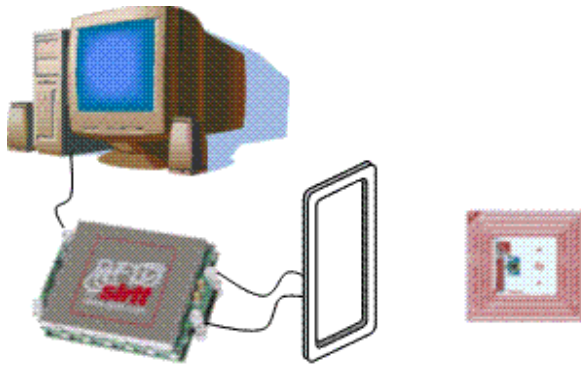


Figura1: funcionamiento de un sistema RFID<sup>1</sup>

Este sistema de comunicación permite la recepción y el envío de la información por una interfaz que puede variar dependiendo del modelo que se utilice, algunos pueden ser por medio de RS 232 o RS 485[1].

Cada sistema cuenta con un lector, el cual responde a una señal de entrada, en esta información se encuentra lo que es una base de datos que le permite tener el acople con al información recibida.

---

<sup>1</sup> Tomado de : [http://www.egomexico.com/tecnologia\\_rfid.htm](http://www.egomexico.com/tecnologia_rfid.htm)

En la base de datos se almacena la información del dispositivo a identificar y cada una de las características que se deseen tener en cuenta para la identificación al ser leído por el lector.

Los transpondedores son dispositivos que cuentan con unas características de diseño, que facilitan la implementación de esta tecnología en diferentes campos. Estos transpondedores están compuestos de un elemento de acople de radiofrecuencia conformado por una bobina o antena y un microchip.

En el mercado se encuentran transpondedores de dos tipos: activos o pasivos. Cada uno brinda la función que se observa en la tecnología RFID, mostrando la diferencia que hay en cada Tag dependiendo de su característica pasivo o activo.

[1]

- Los Tags pasivos: Estos Tags tiene como característica principal que no tiene la necesidad de una alimentación tal como una batería o alimentación de algún tipo. Debido a esta característica es el Tag mas usado en el mercado ya que es de fácil fabricación, y los costos para su comercialización son bajos permitiendo ser empleados en diferentes sectores. La mayoría de estos Tags la antena ha de estar diseñada para obtener la energía necesaria para funcionar a la vez que para transmitir la respuesta por backscatter. Estos Tags son activados con la señal que es enviada a través del lector, están compuestos de materiales duraderos y pueden tener tamaños y grosor de hasta casi menso q una hoja de papel.  
[1]

- Los Tags activos: Están diseñados para tener un mejor alcance en el envío y recepción de la información. A diferencia de los Tags pasivos estos Tags cuentan con su propia fuente de energía o fuente autónoma de energía que permite la mejor propagación de la onda hacia el lector. Por contar con su propia fuente de energía son capaces de enviar a mayores distancias la señal, sin importar el terreno en el cual se este realizando el envío ya que son capaces de soportar terrenos hasta de tipo húmedos. Algunos son diseñados con características que permiten tener una forma de advertencia del medio en el cual se esta trabajando. La energía que estos utilizan son diseñadas para ser usadas durante muchos años.[1]

- Los Tags semi-activos: Cuentan con una fuente de energía integrada permitiendo energizar el Tag para realizar su función. Para realizar la transmisión de datos este transponedor (etiqueta o Tag) utiliza la potencia

que es emitida por un lector. Una de las ventajas de estas etiquetas es que al no necesitar la señal del lector para energizarse pueden ser leídas a mayores distancias. Esto permite obtener lecturas de objetos que se encuentren en movimiento a altas velocidades.[1]

### **3.2.1 CLASIFICACIÓN DE LOS TAGS POR SU FRECUENCIA OPERATIVA [1]**

Las antenas o lectores manejan diferentes rangos de frecuencias para su transmisión dependiendo de su rango de operación. Las etiquetas de frecuencias bajas denominadas LF (Low frequency), estas se destacan ya que el voltaje inducido es proporcional a la frecuencia y se puede producir el voltaje necesario para alimentar el circuito dependiendo de la cantidad de espiras.

Las etiquetas de alta frecuencia HF (High frequency), son diseñadas con espiras de 5-7 vueltas lo que hace una reducción en su costo, a diferencia de los LF (Low Frequency) y permiten tener una propagación de decenas de centímetros.

Las etiquetas de tipo pasivo de frecuencia UHF (Ultra High Frequency) y de microondas suelen realizar su conexión por medio de antenas clásicas de dipolo. Como estas antenas dependen de la radiación que se realiza en el acople de sus ejes, se observa la calidad de la identificación de la etiqueta, la cual es identificada según la orientación que se le da a la etiqueta respecto a la antena.

Dependiendo de la aplicación para la cual se desean utilizar las etiquetas RFID, se encuentra útil tener en cuenta el rango en el cual se puede trabajar (Figura2).

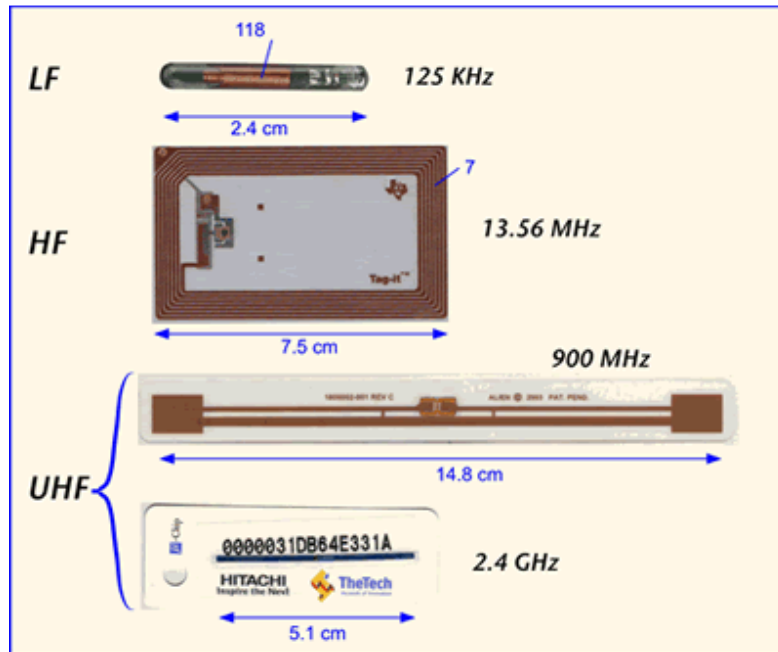


Figura 2. Figura de frecuencias de operación de los Tags. <sup>2</sup>

Las etiquetas de LF (Low frequency) son utilizadas en el control de accesos, identificación de animales, procesos de manufactura, etc. En las etiquetas HF (High Frequency) existen aplicaciones tales como identificación de pacientes (industria de la salud), control de accesos, bibliotecas, seguimiento de productos, trazabilidad, etc. Y para etiquetas UHF (Ultra High Frequency) y microondas se encuentra aplicaciones en centros logísticos, administración de activos, tracking de sistemas de inventario, parkings, industria farmacéutica, laboratorios, exposiciones, tracking de containeres, trazabilidad de ítems, etc. (Tabla1)

<sup>2</sup> Tomado de: [http://www.dipole.es/Tags\\_RFID/Tags\\_RFID.html](http://www.dipole.es/Tags_RFID/Tags_RFID.html)

<b>BANDA DE FRECUENCIA</b>	<b>CARACTERISTICAS</b>	<b>APLICACIONES TIPICAS</b>
<b>LF (BAJA FRECUENCIAS)</b>	Se caracteriza por lectura para distancias pequeñas y medianas, compatibilidad de Tags económicos y velocidad baja en su lectura.	Control de acceso Identificación de animales Control de existencias Control de automóviles
<b>HF (ALTA FRECUENCIA)</b>	Lecturas de grandes distancias alrededor de las decenas de centímetros, mas seguridad en la calidad de información.	Control de acceso Tarjetas inteligentes
<b>UHF Y MICROONDAS</b>	Lecturas medias y altas, mayor costo a comparación de la frecuencia baja y alta, mejor calidad de lectura alta.	Supervisión en sistemas ferroviarios y automotriz, acceso y control de peajes

TABLA1. Características de los rangos de frecuencia de las etiquetas RFID<sup>3</sup>

### 3.2.2 SISTEMA ANTICOLISIÓN [2]

Si se desea con seguir una efectividad en la comunicación de la tecnología RFID, al utilizarse muchas etiquetas a la vez es necesario implementar lo que se denomina un algoritmo anticolisión. Esto se realiza como medio de prevención debido a que un lector antes de emitir una señal de lectura no sabe cuantas etiquetas se encuentran a su alrededor; debido a esto debe existir un mecanismo que permita obtener estas lecturas, de lo contrario en el momento en que existan muchas etiquetas en el rango de lectura intentando contestar al mismo tiempo, podrían existir colisiones.

<sup>3</sup> Tomado de : [http://www.pwc.com/ve/spa/pdf/Advisory\\_200712.pdf](http://www.pwc.com/ve/spa/pdf/Advisory_200712.pdf)

El lector detecta la colisión y envía una señal para la transmisión de las etiquetas durante un tiempo. Después estas irán respondiendo cada una por separado por medio de un algoritmo bastante complejo.

### **2.2.3 CONECTIVIDAD [3]**

Para la implementación de un sistema RFID, se debe tener en cuenta la forma de conectividad de la red del sistema de comunicación. Los lectores de RFID utilizan la comunicación que puede ser de tipo serial, ya sea RS-232 o RS-485, Ethernet, Wireless 802.11 y USB. Aunque se ha observado en la actualidad que la mayoría de los fabricantes intentan habilitar Ethernet en sus lectores e inclusive conectividad wireless 802.11.

Los sistemas de comunicación usados son:

- RS-232. Permite la comunicación favorable en corto alcance; debido a esto, se observa que la velocidad en la comunicación es baja y trabaja entre 9600 bps a 115.2 kbps. Un limitante para este sistema, es el largo cable el cual no puede superar una distancia de más de 30 metros.
- RS-485. Es considerada como el reemplazo del RS-232, teniendo como característica permitir una velocidad mayor en el rango de 2.5 Mbps y conectividad en forma de bus, dejando la posibilidad de conectar dispositivos al mismo cable que puede tener una distancia hasta de 1.200 metros.
- Ethernet. Es de los sistemas, el considerado como el mas seguro por su protocolo TCP/IP, el cual está implementado en la mayoría de las empresas, por su velocidad en el envío de la información siendo esta una velocidad mucho mayor de la que se necesita en RFID.

- Wireless 802.11. Es implementada cuando se trabaja con lectores de RFID móviles, dando una mayor ventaja sobre los anteriores sistemas de conectividad mencionados, pues reduce costos por no ser necesario el cableado.
- USB. Como bien es sabido, la tecnología RFID está buscando siempre poder implementarse en cualquier empresa sin importar las características del sistema de comunicación que usen, por eso se encuentra también este sistema de comunicación con la posibilidad de ser adaptado a USB.

#### **3.2.4 SEGURIDAD DE SISTEMA RFID [4]**

Este tema es muy mencionado siempre que se habla de RFID. Cualquier empresa o entidad que tenga interés en implementar este sistema tiene que saber si existen o no factores de riesgo en esta tecnología. Este tema es una de las desventajas que tiene esta tecnología, al igual que los demás sistemas de seguridad, en algún momento pueden ser intervenidos, ya que si es ubicado un lector diferente al que se usa en la empresa por medio de un software, una persona con este sistema puede conocer las características que guarda el Transpondedor.

La forma de prevención o esquema de seguridad que recomiendan al utilizar esta tecnología pueden ser:

- La jaula de Faraday: Es así denominada una de las formas de seguridad que consiste en un contenedor desarrollado con una malla de metal la cual da como característica la no penetración de señales de radio; otra de las utilizadas es la desactivación de las etiquetas RFID antes de ser entregadas al consumidor, teniendo en cuenta que si se realiza este proceso ya no sería posible reutilizarlas. De igual forma, este método no viene siendo el más adecuado pues cuando se adquiere esta tecnología lo

usual es obtenerla para más de una función. Por último, otra forma es la interferencia activa la cual no permite que las etiquetas sean manipuladas por el consumidor, ya que el consumidor puede estar usando un sistema que emita señales de radio que podrían causar algún tipo de interferencia a los transpondedores (Tags) más cercanos.

## **4. MARCO CONTEXTUAL**

### **4.1 APLICACIONES [5]**

La tecnología RFID, se ha desarrollado buscando una seguridad y confiabilidad. Por su fácil manejo es manipulada y aplicada para todo tipo de equipos, elementos quirúrgicos, animales, vehículos, seres humanos, etc. Este sistema de autoidentificación ha permitido tener una nueva opción dejando atrás el código de barras que presenta limitaciones.

Con el pasar del tiempo se espera que la variación y disminución del costo de las etiquetas permita suprimir por completo al código de barras en el mejor de los casos.

Actualmente se encuentra en investigación la implementación de sistemas RFID en otros campos como la introducción de chips en seres humanos teniendo el control de sus características físicas, si en algún momento se necesita la visita a un medio se pueda contar con toda la historia clínica del paciente. Otro de los campos, es el control del equipaje en los aeropuertos y el ingreso o salida de pasajeros a diferentes países por medio de la implantación de tecnología RFID en pasaportes.

Entre la industria comercializadora de productos se ha observado la implementación de la tecnología RFID en tiendas de cadena como Wal-Mart



pionera en la implementación en sus productos, contando con un control de inventario en sus estantes y venta de productos.

Algunas otras aplicaciones más comunes en las cuales la tecnología de RFID brinda soluciones y calidad de servicio son:

Control de calidad, producción y distribución, localización y seguimiento de objetos, control de accesos, identificación de materiales, control de fechas de caducidad, detección de falsificaciones, almacenaje de datos, automatización de los procesos de fabricación, información al consumidor, reducción de tiempo y coste de fabricación, reducción de colas a la hora de pasar por caja e Inventario automático.

## **4.2 RFID VS CÓDIGO DE BARRAS**

### **4.2.1 VENTAJAS DEL RFID**

El sistema de comunicación RFID permite observar grandes ventajas sobre el único método de autoidentificación que se conocía hasta el momento como es el código de barras. Encontrando mejoras Como:

- La tecnología de RFID tiene la posibilidad de realizar una identificación simultánea de objetos, mientras que el código de barras solo permite la identificación de un objeto a la vez.[7]
- Las Transpondedores pueden permitir la escritura varias veces en ellos, mientras que el código de barras solo permite el almacenamiento de información una sola vez.

- Los Tags pueden tener una capacidad de almacenamiento que no se consigue con el código de barras.[7]
- Las falsificaciones de los Tags es obsoleta ya que es imposible de realizar, lo que si es posible con el código de barras al fotocopiarlo. El Transpondedor queda inutilizable si es fotocopiado.
- Las etiquetas RFID no necesitan un contacto visual directo con el lector permitiendo ganar tiempo en diferentes actividades. [7]
- El cuidado que se debe tener con los Tags RFID es mucho menor, pues están diseñados para soportar diferentes medios. Gracias a esto, es posible utilizarlo con más facilidad en campos como los supermercados, donde se encuentran en constante contacto de diferentes personas.[7]
- Las etiquetas RFID ofrecen mayor rapidez de lectura especialmente para las operaciones de alta velocidad, como la clasificación de empaques.

#### **4.2.2 COMPARACIÓN ENTRE RFID Y EL CÓDIGO DE BARRAS**

Cada tecnología de seguridad busca dar lo mejor en diseño, calidad en su material, brindar el mejor servicio y que el usuario adquiera una tranquilidad por contar con este tipo de sistemas.

Para tener claro cual es mejor o porque trae ventajas uno sobre otro, en un cuadro comparativo se observa los Pro y contra de dos sistemas de seguridad. (Tabla

TEGNOLOGIA RFID	CODIGO DE BARRAS
<ul style="list-style-type: none"> <li>• Mayor capacidad para guardar información.</li> <li>• No requiere una distancia directa con el lector.</li> <li>• Código único, determinado por el fabricante y por su nuevo adquisidor.</li> <li>• Adquisición de las características del producto de manera completa.</li> <li>• Resistente a diferentes medios climáticos.</li> <li>• Difícil falsificación por las características y sus materiales.</li> <li>• Precio en descenso por su demanda e implementación.</li> </ul>	<ul style="list-style-type: none"> <li>• Limitación en almacenamiento de información.</li> <li>• Para una adecuada lectura debe tener acercamiento directo al lector.</li> <li>• El código es igual, y se maneja la secuencia numérica.</li> <li>• Identificación de cada producto directa.</li> <li>• Limitantes a el medio en el que se disponga su utilización por daños en su material.</li> <li>• Fácil falsificación ya que puede sacarse una fotocopia del código.</li> <li>• Su costo o valor se considera insignificante.</li> </ul>

TABLA 2. Comparación entre RFID y código de barras<sup>4</sup>

## 5. DISEÑO DEL SISTEMA UTILIZADO

El desarrollo de este proyecto esta diseñado con el fin de cumplir con la utilización de la mayor parte de tecnología que una implementación de sistema RFID lo requiera.

<sup>4</sup> Tomado de: <http://www.ihg.net/java/X?cgi=lateral.rfid.VentajaVsBarras.pattern>

Teniendo en cuenta los problemas en el laboratorio que se pueden rectificar con la utilización de un sistema RFID, la estructura del proyecto esta compuesta de la siguiente forma.

## 5.1 DISEÑO DE INFRAESTRUCTURA

- Transpondedores o Tags: Cuentan con al capacidad de almacenar la información del objeto o infraestructura a trabajar, emitiendo esta información después de recibir la señal procedente del lector [4]. En este proyecto los utilizados son los de etiquetas, con una frecuencia de 13.56 MHz, protocolo ISO 15693 y lector/escritor.[8] (Figura3)

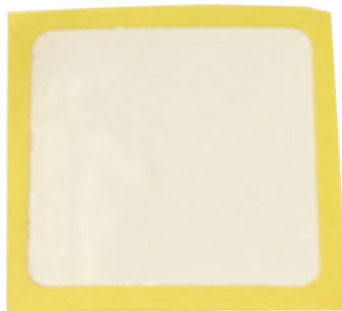


Figura 3. Diseño de Tags utilizados<sup>5</sup>

- Lector (RFID): [9] Genera un campo de radiofrecuencia, normalmente conmutando a una bobina de alta frecuencia. Las frecuencias usuales van desde 125 Khz hasta la banda ISM de 2.4 Ghz. El campo de radiofrecuencia genera una corriente eléctrica sobre la bobina de recepción del dispositivo. Esta señal es rectificadora y de esta manera se alimenta el

---

<sup>5</sup> Tomado de : <http://www.trossenrobotics.com/store/p/4162-Inventory-Label-Square-Version-38-x-38-mm-.aspx>

Transpondedor. Cuando la alimentación llega a ser suficiente para los Tags transmiten sus datos. El lector detecta los datos transmitidos por los Tags como una perturbación del propio nivel de la señal. (Figura4)



Figura 4. Lector utilizado<sup>6</sup>

- PC (computador): Este cuenta con el diseño de la base de datos que se implementa en el proyecto. Esta conectado directamente con el lector para realizar el proceso de comunicación. El utilizado en las pruebas es un laptop de uso personal, donde se instala el software de prueba, permitiendo visualizar la lectura y desarrollo del sistema RFI
- Software: En esta etapa del proyecto se diseña y se almacena la información, con la cual se cuenta para el diseño de cada plataforma a utilizar para el proyecto. Para el desarrollo de esta etapa se cuenta con la ayuda de Visual Basic.Net (Figura5). Así como también con el software con el cual trabaja el lector y los transpondedores.

---

<sup>6</sup>Tomada de: foto realizada por gestores del proyecto a la antena utilizada la

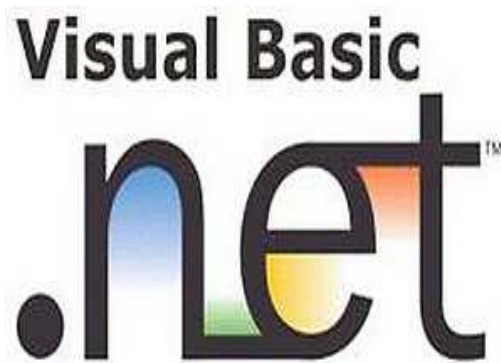


Figura 5. Logotipo de Visual Basic.net<sup>7</sup>

- Otros: Se citan los componentes del laboratorio que se utilizan para realizar las pruebas, así como los Laboratorios de la Facultad de Ingeniería Electrónica.

## 5.2 ARQUITECTURA IMPLEMENTADA

La principal característica de este proyecto, es dar una solución al manejo del préstamo y control de los instrumentos del Laboratorio de la Facultad de Ingeniería Electrónica.

La arquitectura implementada esta dividida de la siguiente manera:

- Conectividad: Un computador (PC) y un sistema de comunicación RFID.

---

<sup>7</sup> Tomado de:

[http://www.ceintec.com/curso\\_de\\_programador\\_de\\_visual\\_basic\\_net\\_presencial\\_en\\_bilbao\\_bilbo\\_vizcaya\\_bizkaia\\_3892315.jpg](http://www.ceintec.com/curso_de_programador_de_visual_basic_net_presencial_en_bilbao_bilbo_vizcaya_bizkaia_3892315.jpg)

- **Funcionamiento Básico:** Un sistema RFID, se compone de un lector y un Transpondedor (etiqueta o Tag) que responde al lector. El lector genera un campo de radiofrecuencia normalmente conmutando una bobina de alta frecuencia. Las frecuencias usuales van desde 125 Khz hasta la banda ISM de 2.4 Ghz. El campo de radiofrecuencia genera una corriente eléctrica sobre la bobina de recepción del Transpondedor. [10]

Cuando la alimentación llega a ser suficiente, el circuito transmite sus datos. El lector detecta los datos transmitidos por el Transpondedor como una perturbación del propio nivel de la señal. [10]

Seguido de la señal detectada por el lector, se obtiene un código de identificación contenido en cada Transpondedor y es enviado al computador. El control de acceso indica si este código tiene permiso para iniciar una actividad, de ser así le permite el acceso al préstamo de los instrumentos del laboratorio.

Simultáneamente ya se crea un registro en la base de datos del acceso y el préstamo del componente. (Figura6)

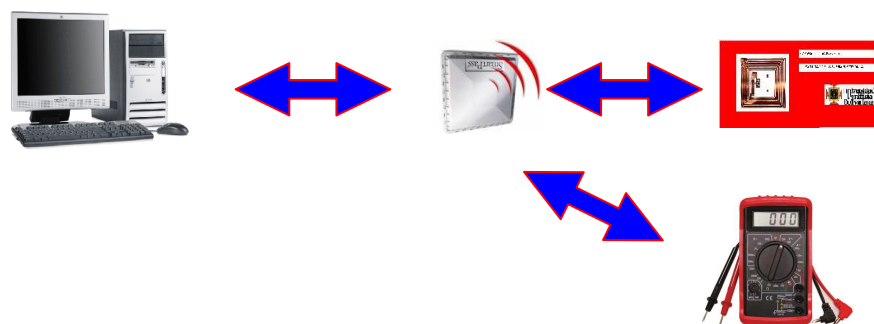


Figura 6. Modelo de arquitectura

## **6. HARDWARE**

El sistema de comunicación RFID que se utiliza para la implementación en el Laboratorio de la Facultad de Ingeniería Electrónica es un APSX RFID Kit (Icode 2) RW-210. [11]

### **6.1 DESCRIPCIÓN DEL SISTEMA RFID UTILIZADO**

El APSX RFID Kit (Icode 2) RW-210[11], cuenta con diferentes diseños de Transpondedores, que permite la posibilidad de tener opciones para realizar las pruebas y encontrar el Transpondedor a utilizar en los equipos del laboratorio.

El APSX RFID Kit (Icode 2) RW-210[11], esta diseñado para lectura / escritura en una frecuencia de 13.56MHz. Las etiquetas RFID en el Kit utilizan la norma ISO15693 (Figura 7). El Kit cuenta con un cable de comunicación de la antena al computador y un cable de alimentación de 6V DC.

Beneficios del Kit:

- Interfaz sencilla.
- No requiere de mantenimiento.
- Permite la programación de los transpondedores.
- Bajo consumo de energía adecuado para terminales autónomos.
- Reducción en el tiempo de transmisión en 125 kHz.





Figura 7. Imagen del kit <sup>8</sup>

### 6.1.1 TRANSPONDEDOR (TAG)

En cada prueba se observa por su tamaño y forma, el Transpondedor de etiqueta. El Inventory Label - Square Version (38 x 38 mm) [8], es un Tag que cuenta con la posibilidad de lectura y escritura, trabaja a una frecuencia de 13.56 MHz y maneja el protocolo ISO 16593. (Figura3)

### 6.1.2 LECTOR DE RFID

El lector o antena utilizada es el APSX RW-210 RFID Reader/Writer [9]. Esta antena trabaja con Transpondedores que operan con la ISO15693 y manejan una frecuencia de 13.56MHz. (Figura8)

Características del lector [9]:

- Una interfaz sencilla.
- Pequeño tamaño

<sup>8</sup> Tomado de: <http://www.trossenrobotics.com/store/p/4772-APSX-RFID-Kit-Icode-2-RW-210.aspx>

- Permite la programación al usuario de sus propias tarjetas
- Maneja un bajo consumo de energía adecuado para terminales autónomos
- Trabaja a una frecuencia de 13.56MHz
- Velocidad en el envío de datos de 19.200 bps
- Necesita usar un convertidor para conectar directamente al PC
- Un consumo de energía de 20 mA
- Cuenta con dos LED indicadores.



Figura 8. Imagen del lector <sup>9</sup>

## 7. DISEÑO DEL SOFTWARE

En esta etapa del proyecto se realiza el estudio de prototipos que puedan cumplir con el diseño a aplicar en el laboratorio, y que permita tener el acople con la tecnología RFID. Para este proceso se decide trabajar con Visual Basic y el software con el que trabajan los Tags.

### 7.1 DISEÑO DE LA BASE DE DATOS

---

<sup>9</sup> Tomado de: <http://apsx.com/RW210.aspx>

Para el desarrollo de la base de datos, se realiza diferentes esquemas que permiten contar con la comodidad y el diseño, supliendo las necesidades del Laboratorio de la Facultad de Ingeniería Electrónica.

### **7.1.1 DIAGRAMA ENTIDAD-RELACION PARA LA BASE DE DATOS.**

La base de datos, se diseña mediante un diagrama de bloques, basado en el software DBDesigner<sup>10</sup>, debido a que este es un completo sistema visual, que permite el desarrollo de aplicaciones para bases de datos que integra diseño, modelado, creación y mantenimiento en un ambiente simple y fácil de usar. DBDesigner permite crear tablas e interconectarlas entre si para compartir datos e incluso variables de la misma denominación entre tablas, como se muestra en la Figura9.

Se crean 5 tablas principales: Estudiante, Instrumentos, Carreras, Laboratorios y Registros, las cuales se entrelazan para obtener las tablas de relación.

El esquema que se utiliza es:

---

<sup>10</sup> DBDesigner: Es un sistema totalmente visual de diseño de bases de datos, que combina características y funciones profesionales con un diseño simple. (Tomado de: <http://www.taringa.net/posts/downloads/840850/DB-Designer---Dise%C3%B1o-de-Base-de-Datos.html>)

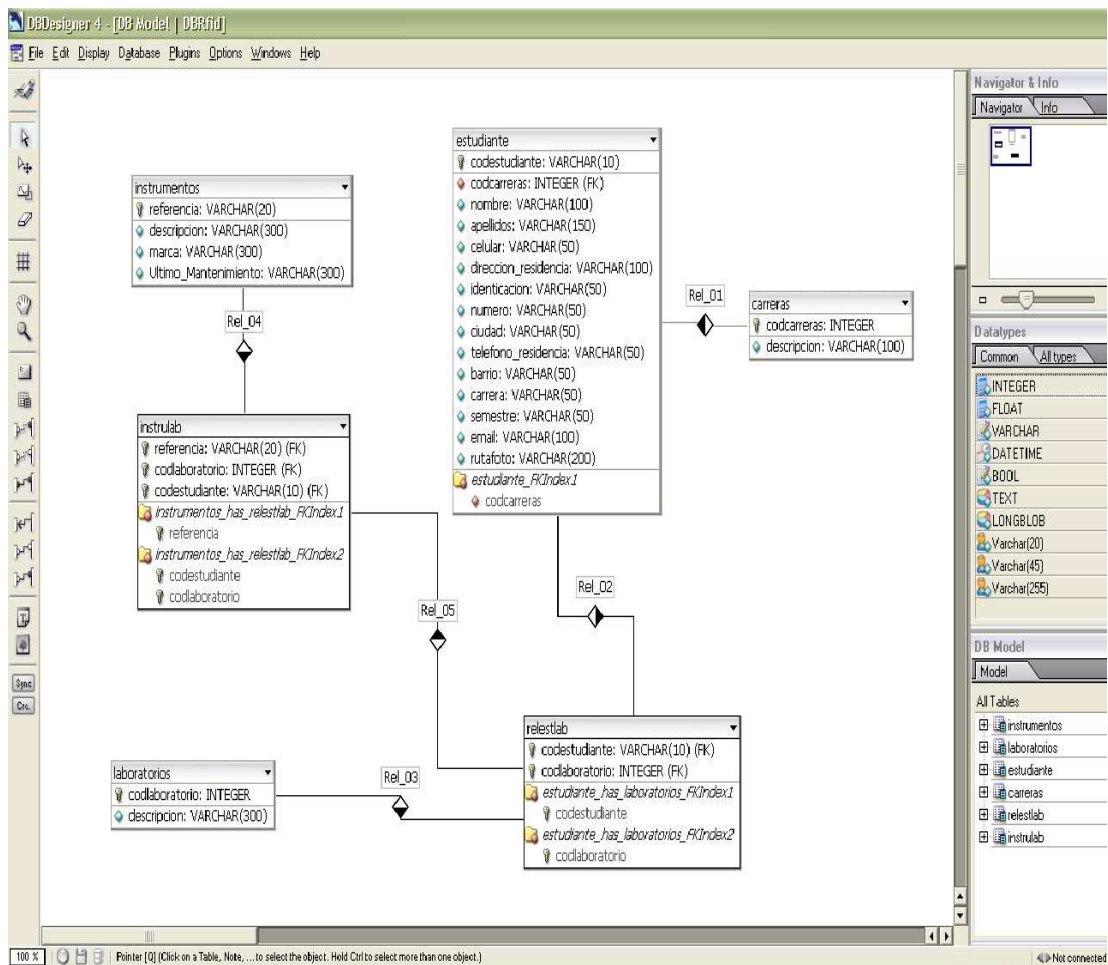


Figura 9. Diagrama entidad-relación

- Tabla Estudiante: En esta tabla se almacena los datos correspondientes al número de la dirección MAC (Media Access Control) o código interno del transpondedor asignado a cada estudiante.

Esta tabla cuenta con campos tales como: codestudiante, codcarreras, nombre, apellidos, celular, dirección\_residencia, identificación, número, ciudad, teléfono\_residencia, barrio, carrera, semestre, email y rutafoto. (Figura10)

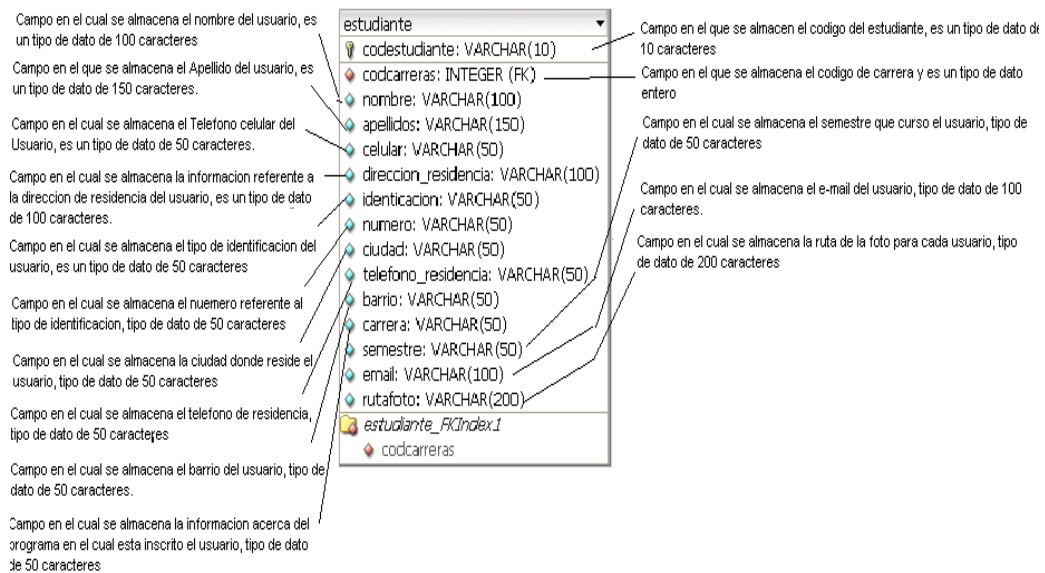


Figura 10. Tabla Estudiantes

- T  
 abla Instr

umentos: Guarda la información del instrumento, la cual se clasifica por la dirección MAC (Media Access Control) o código del transpondedor que se le asigna a cada instrumento. Esta tabla cuenta con los campos: Referencia, descripción, marca y último\_Mantenimiento (Figura11).

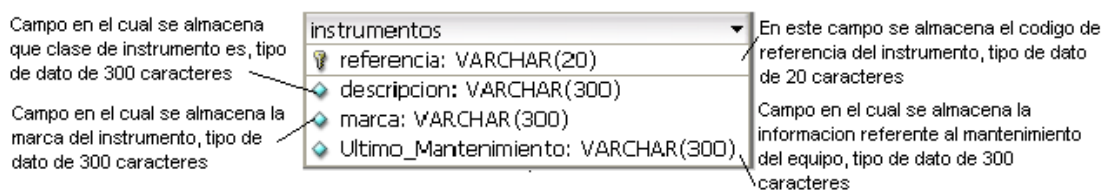


Figura 11. Tabla Instrumentos

- Tabla Carreras: Da la descripción de las facultades o carreras que utilizan el servicio del laboratorio ( Figura12).

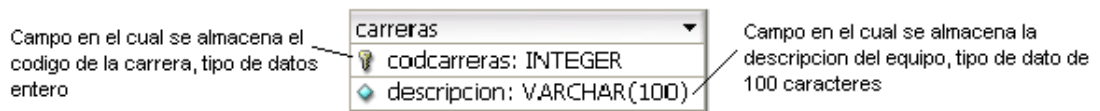


Figura 12. Tabla Carreras

- Tabla Laboratorios: Muestra todas las materias para las cuales se realizan los préstamos de los instrumentos, con su Código de Laboratorio y su descripción (Figura13).

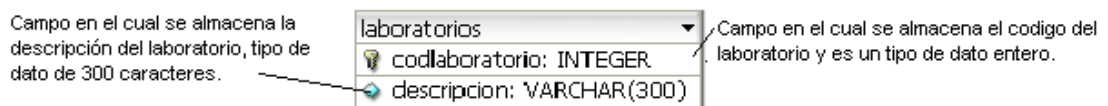


Figura 13. Tabla Laboratorios.

Una vez que se diseña el diagrama entidad-relación en diagrama de bloques, se genera el código para trabajar en el ambiente gráfico (diseño de ventanas), desarrolladas en los siguientes numerales.

### 7.1.2 PLANTILLA O VENTANA EN VISUAL

Una vez creada la base de datos, se busca una forma más práctica de mostrar un ambiente gráfico, que cumpla con las condiciones de diseño. Para conseguir este acople de la base de datos con un ambiente grafico se utiliza Visual Basic.Net.

Para el acceso a la plantilla o ventana principal, se requiere digitar una contraseña establecida por el usuario con el fin de proteger y brindar seguridad, evitando el acceso a personas no autorizadas. La ventana diseñada se observa en la figura14.



Figura 14. Ventana de contraseña

El código de esta ventana se diseña con un IF<sup>11</sup> (sentencia de programación), donde la contraseña que pide es “123456”; si es correcta esta contraseña, permite el acceso y deja visible la siguiente plantilla. Si es falsa la contraseña, se utiliza un Else<sup>12</sup> (sentencia de programación), con esta sentencia la condición cambia, y se observa un mensaje “Contraseña incorrecta” y “Error”.(Figura15)

```
Public Class Inicio
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        funcion()
    End Sub
    Private Sub funcion()
        If Me.TextBox1.Text = "123456" Then
            Principal.Visible = True
            Me.Visible = False
        Else
            MsgBox("Contraseña Incorrecta", MsgBoxStyle.Information, "ERROR")
        End If
    End Sub
    Private Sub TextBox1_KeyPress(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles
        If e.KeyChar = Chr(13) Then
            funcion()
        End If
    End Sub
    Private Sub Inicio_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        End Sub
End Class
```

Figura 15. Código Ventana contraseña

<sup>11</sup> IF: Tipo de estructura de selección empleada en la programación de algoritmos.(tomado de: <http://www.alegsa.com.ar/Dic/if.php>)

<sup>12</sup> ELSE: Extiende una sentencia if para ejecutar una sentencia en caso de que la expresión en la sentencia if se evalúe como FALSE. (Tomado de: <http://www.rlz.cl/php/control-structures.else.html>)

Una vez digitada la clave correcta por el usuario, se obtiene el acceso a la ventana principal (figura16). En esta ventana se diseña con 4 botones de acceso que tiene como funcionalidad la conexión a las demás ventanas: Laboratorio, Administrar, Consultar y Salir.



Figura 16. Ventana principal

El programa de la ventana principal, se inicia con una línea Private Sub<sup>13</sup> (sentencia de programación), que se crea para poder cerrar la ventana de la plataforma principal. La siguiente línea se inicia con un Private Sub, donde está el acceso a la tabla estudiantes que trabaja con el botón Laboratorio de la base de datos. Si la sentencia es verdadera permite observar las características guardadas de esta tabla, y si no existe un acceso no permite observar la información. Para los otros botones tales como Administrar, Consultar y Salir, se crea la misma rutina y se enlaza a la tabla correspondiente en la base de datos como se observa en la Figura17.

---

<sup>13</sup> Private Sub: es una combinación de instrucciones para definir un sub procedimiento.



```

Public Class Principal
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        Close()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Estudiantes.Visible = True
        Me.Visible = False
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Tablas.Visible = True
        Me.Visible = False
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
        Consultar.Visible = True
        Me.Visible = False
    End Sub

    Private Sub Principal_FormClosed(ByVal sender As System.Object, ByVal e As System.Windows.Forms.FormClosedEventArgs)
        Inicio.Close()
    End Sub

    Private Sub Principal_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    End Sub
End Class

```

Figura 17. Código para la Ventana principal

### 7.1.3 DISEÑO DE LA VENTANA LABORATORIO

La ventana Laboratorio cuenta con un botón principal que tiene la palabra “Conectar” como se observa en la figura 18. Una vez dado clic sobre este botón, el software hace un barrido de puertos para verificar en cuál de ellos se encuentra conectado el dispositivo RFID.

Seguido de la identificación del puerto donde se encuentra el dispositivo de RFID, el usuario pasa su identificación, la cual tiene un transpondedor. Si el transpondedor no es reconocido, una ventana le indica que no esta registrado este código de identificación.



Figura 18. Ventana Laboratorios I

Si es reconocido este código, tiene un acceso a la siguiente plantilla donde encuentra características del usuario, laboratorios al cual pertenece y la mesa en la que trabaja, como se observa en la Figura19.

La ventana esta conformada de la siguiente forma:

- GroupBox: Se utilizan para proporcionar un agrupamiento de cuadros de texto.
- ComboBox: Combinación de información en un cuadro desplegable

El GroupBox en esta platilla brinda la información nombre, apellido, ID y carrera del usuario (Figura19). Junto a esta información se crean dos ComboBOx, uno despliega una lista de los laboratorios y el otro una lista de las mesas.

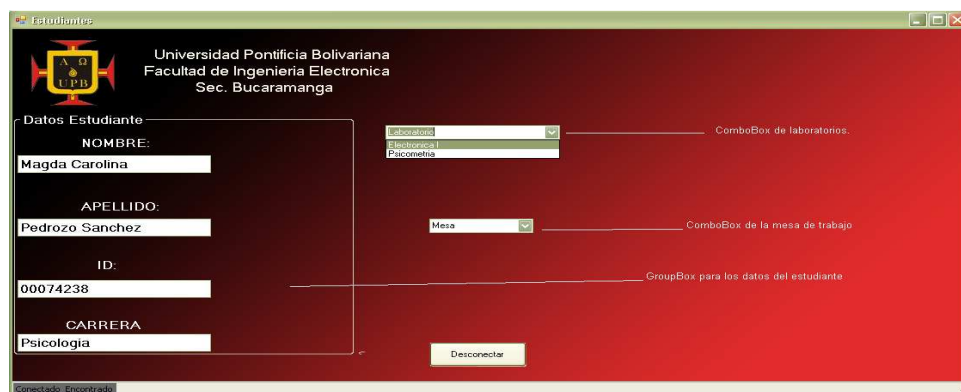


Figura 19. Ventana Laboratorios II

Una vez seleccionado el laboratorio y la mesa de trabajo, se activa un grupo de campos de texto llamado (GroupBox Instrumentos) (Figura20). Se observa actividad en el campo de texto, cuando es leído el instrumento.

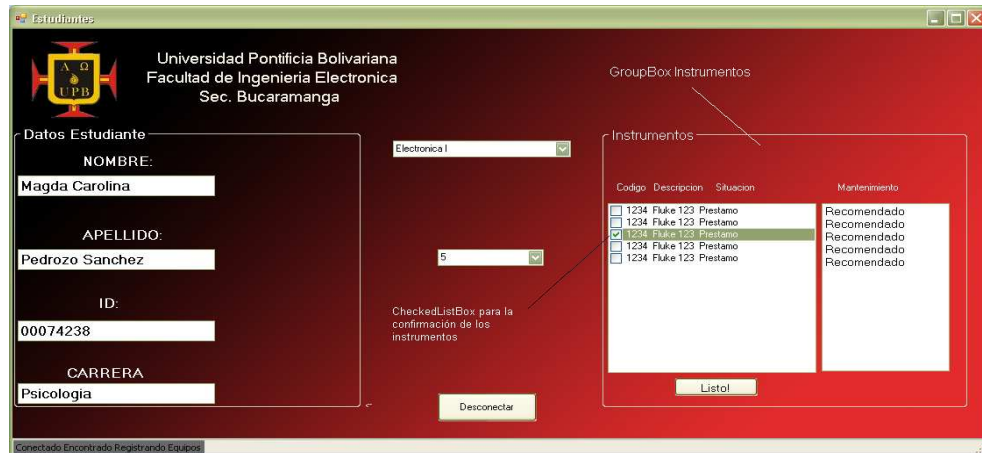


Figura 20. Descripción préstamo de instrumentos

La información que se observa es un código de inventario, nombre del instrumento, si se encuentra disponible y el mantenimiento del instrumento (recomendado o para mantenimiento), como lo muestra la Figura21.

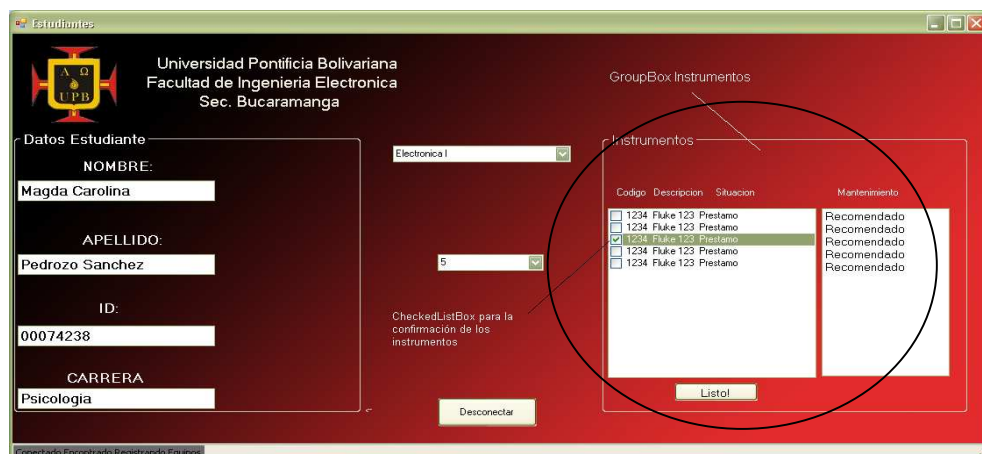


Figura21. Ventana descripción de los instrumentos prestados

El proceso de detección de los transpondedores (Tags), requiere de un Timer, de esta forma el dispositivo RFID, que esta en constante escaneo recibe los datos de entrada y permite observar la información en el campo de texto.

Identificado el instrumento, se visualiza un cuadro de confirmación (CheckedListBox), de esta forma se asegura el préstamo del instrumento a ese

usuario. Prestado el instrumento al usuario, se da clic en “Listo” y la información se guarda directamente en la base de datos. (Figura22)

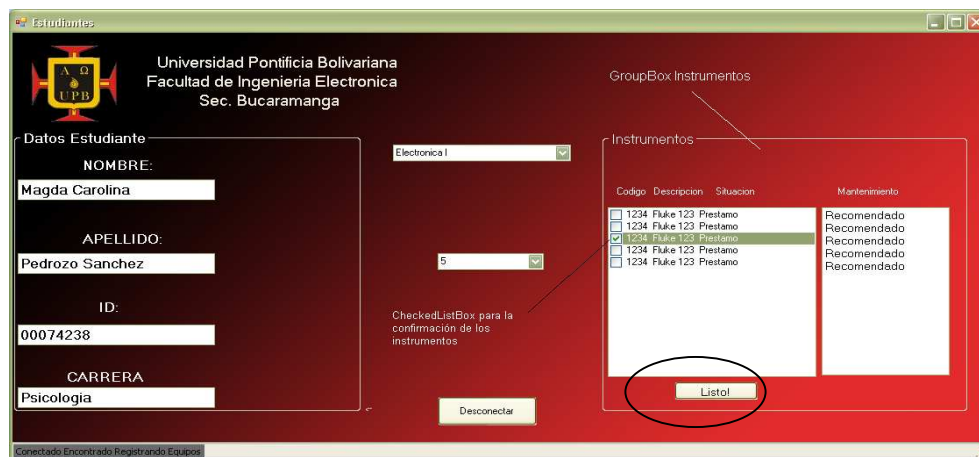


Figura22. Identificación del botón Listo

En el código se crean las variables publicas llamadas apsx (sentencias de programación), conectado, encontrado y reg, en las cuales se guardaran los registros al ser detectado una de las etiquetas, tal como se muestra en la figura 23.

```
Public Class Estudiantes
    Dim apsx As New APSXDLL.apsxrfid
    Dim uidbytes() As Byte
    Dim conectado As Boolean
    Dim encontrado As Boolean
    Dim reg As Integer

    Private Sub AlumnosBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Me.Validate()
        Me.AlumnosBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.BaseDatosDataSet)
    End Sub
End Class
```

Figura 23. Descripción del

código de la ventana Laboratorios.

#### 7.1.4 DISEÑO DE LA VENTANA ADMINISTRAR

La ventana administrar muestra los datos correspondientes a los estudiantes, equipos, registros, laboratorios y carreras. Esta ventana está compuesta por 5 campos de texto unidos en un GroupBox (Figura24). Cada campo brinda la información guardada en las tablas de la base de datos. En la parte inferior de

cada cuadro de texto, se implementa una barra de opciones que permite buscar los registros asignados a cada estudiante o a cada equipo (Figura26). La barra de opciones, se encuentra directamente conectada a la base de datos, lo cual permite agregar o eliminar datos sujetos a posibles cambios.



Figura 24. Ventana Administrar

En la figura 25, se observa en los GroupBox “Estudiantes” o GroupBox “Equipos” una actividad cuando se detecta el código de los Tags, asignados a cada estudiante o cada equipo. La detección de los Tags está unida con la información guardada en la base de datos. La información se muestra en los editores de texto o casillas de texto.

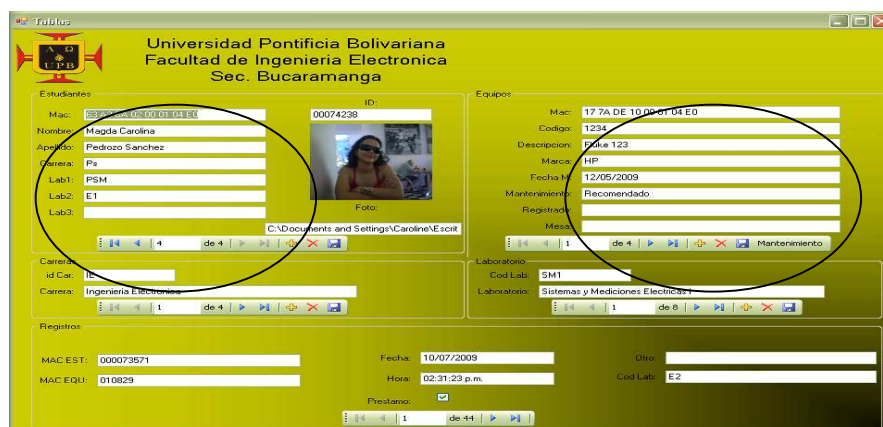





Figura 25. GroupBox “Estudiantes” o GroupBox “Equipos”

En la barra de opciones se tiene de forma numérica los registros incluidos en la base de datos. (Figura26))



Figura 26. Barra de opciones

La barra de opciones está compuesta por los siguientes botones:

-  Permite agregar nuevos registros en la base datos, borrando debidamente cada casilla de los editores de texto (Agregar Nuevo).
-  Permite eliminar los registros en la base datos, una vez estos se muestren en las casillas de los editores de texto (Eliminar).
-  Permite guardar la información digitada en las casillas de los editores de texto almacenándola en la base de datos ( Guardar).

El código de la ventana administrar se ilustra en la figura 27. Muestra cómo se procede con el registro de la fecha de mantenimiento en la ventana administrar.

Se crea una rutina enlazada directamente a la tabla “Equipos” de la base de datos. Este enlace se inicia con la variable  $i=1$ (descripción de programación), esta será comparada con dos variables, la variable “dosmeses” y la variable “unmes”, en las que se guardan los registros de mantenimiento.

Seguidamente, se crea una variable “fm” (sentencia de programación), esta variable se carga con la fecha actual en la que se ejecuta el programa. Luego se realiza una comparación; si fm es menor a dos meses, el mensaje mostrado es “Obligatorio”; si fm es menor a un mes, el mensaje mostrado es “recomendado”; si fm es mayor que un mes el mensaje mostrado es “OK”.

```

Private Sub ToolStripButton28_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton2
    Dim unmes As Date
    Dim dosmeses As Date
    Dim fm As Date

    dosmeses = DateAndTime.Today.AddMonths(-2)
    unmes = DateAndTime.Today.AddMonths(-1)

    Me.EquiposBindingSource.MoveFirst()

    For i = 1 To Me.EquiposBindingSource.Count

        If i > 1 Then
            Me.EquiposBindingSource.MoveNext()
        End If

        fm = Convert.ToDateTime(Me.FechaMTextBox.Text)

        If fm < dosmeses Then
            Me.MantenimientoTextBox.Text = "Obligatorio"
        ElseIf fm < unmes Then
            Me.MantenimientoTextBox.Text = "Recomendado"
        ElseIf fm > unmes Then
            Me.MantenimientoTextBox.Text = "Ok"

            Me.Validate()
            Me.EquiposBindingSource.EndEdit()
            Me.TableAdapterManager.UpdateAll(Me.BaseDatosDataSet)
        End If

    Next
    Me.EquiposBindingSource.MoveFirst()
End Sub

```

Figura 27. Código ventana Administrar.

## 7.1.5 DISEÑO DE LA VENTANA CONSULTAR

La ventana consultar, ilustra los registros correspondientes a los préstamos de los equipos, junto con la persona que solicita un instrumento, la fecha del préstamo, la hora y el laboratorio en el cual el usuario del servicio del laboratorio trabajó. (Figura28)

The screenshot shows a web application window titled 'Consultar'. It features a search interface with the following elements:

- Search Bar:** A dropdown menu with '1' selected, and a search button.
- Filters:** 'Filtrar por:' with options for 'ID', 'Codigo Equipo', and 'Apellido'. A 'Limpiar Filtros' button is also present.
- Form Fields:**
  - ID:** 000072552
  - Nombre:** Carlos Alberto
  - Apellido:** Velasco Gil
  - Carrera:** Ingeniería Electrónica
  - Codigo:** 010829
  - Descripcion:** Oscoscopio Digital
  - Marca:** FLUCKE 12-3
  - Laboratorio:** Sistemas Digitales II
  - Fecha:** 10/07/2009
  - Hora:** 03:06:59 p.m.
- Calendar:** A calendar for July 2009, with the date 23/07/2009 highlighted. A checkbox 'Incluir Esta Fecha' is checked.
- Buttons:** A 'Consultar' button is located at the bottom right.
- Form State:** The 'Préstamo' checkbox is checked, and the text 'Préstamo/Entrago' is displayed below it.

Figura 28. Ventana Consultar

Esta ventana Consultar, además cuenta con 3 botones los cuales tienen como imagen principal una pequeña lupa y son llamados “ToolStripButtons” (Figura 29). Estos botones filtran la información de manera que el operario consulte el registro de los préstamos mediante el código del Equipo, el ID o el Apellido del estudiante. La información es clasificada en celdas de texto. Esta información es direccionada de la base de datos.

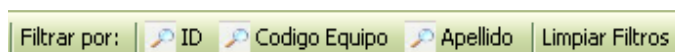


Figura 29. Paleta Filtrar

Para filtrar la información por el ID del estudiante, es necesario dar clic en “ID” de la paleta “Filtrar por”. Seguido se muestra un cuadro en donde se ingresa el código a buscar tal como se muestra en la figura 30.

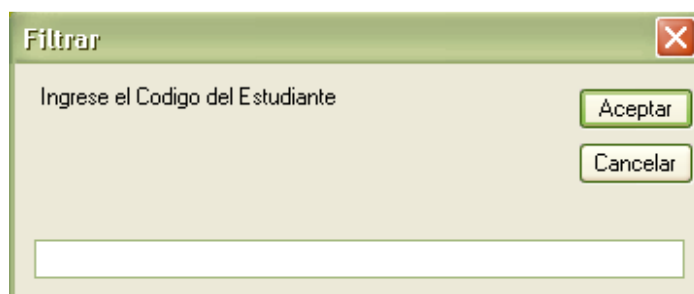


Figura 30. Filtro por Código de Estudiante

Una vez ingresa el ID, se observa en los cuadros de textos asignados la información requerida (Figura31). Esta información es la actividad que se genera en la ventana “laboratorios.” Este proceso se realiza igual cuando se filtra por el Apellido.





Figura 31. Resultado después de Filtrar

Para filtrar la información dependiendo de la fecha, se usa un calendario en forma gráfica llamado (MonthCalendar) (Figura 32). En el se selecciona la fecha del día que se desea ver la actividad, mediante una ventana pequeña llamada (CheckList); que tiene como texto "Incluir Esta Fecha".



Figura 32. Filtro por Fecha, calendario "MonthCalendar".

Al ser accionada la fecha deseada, solo se muestra los datos en los editores de texto. La información que se muestra es la correspondiente a la fecha seleccionada.

En la parte inferior de la plantilla “Consultar”, se observa un CheckBox llamado “Préstamos” (Figura33). El muestra la información en las celdas de texto, si el equipo se encuentra aun prestado o si este ya fue entregado.

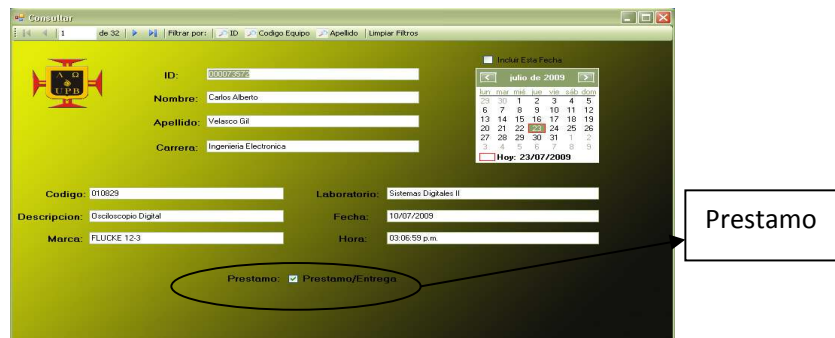


Figura 33. Identificación de la opción Préstamos

Para la inicialización de la ventana consultar se crea una subrutina “Private Sub Consultar” en la que se carga toda la información contenida en la tabla de la base de datos, posteriormente se hace visible la ventana principal para seleccionar el acceso hacia el filtrado de la información. (Figura34)

Una vez terminada la subrutina, se carga una variable llamada “filtro”, esta variable almacena la información digitada en el cuadro de texto. La variable “filtro” permite realizar una comparación. Se ingresa una subrutina condicional “If”, si está vacía, permite realizar la consulta de la información en la base de datos, borrando posteriormente el filtro. Si no está vacía, la información se filtra mediante el “NRC”; este se digita para luego salir de la subrutina.

```

Public Class Consultar
    Dim Fecha1 As String

    Private Sub Consultar_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla 'BaseDatosDataSet.Consultar' Puede moverla o quitarla según sea
        Me.ConsultarTableAdapter.Fill(Me.BaseDatosDataSet.Consultar)
    End Sub

    Private Sub Consultar_FormClosed(ByVal sender As System.Object, ByVal e As System.Windows.Forms.FormClosedEventArgs) Han
        Principal.Visible = True
    End Sub

    Private Sub ToolStripButton2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton2.
        Dim filtro As String
        filtro = InputBox("Ingrese el Código del Estudiante", "Filtrar", "", , )
        If filtro = "" Then
            Me.ConsultarBindingSource.RemoveFilter()
        Else
            Me.ConsultarBindingSource.Filter = "NRC='" + filtro + "'"
        End If
    End Sub
End Class

```

Figura 34. Código Ventana Consultar.

#### 7.1.6 DESCRIPCION DEL PROCESO DE ENVIO DE DATOS

La conexión de la antena y el computador forman una parte muy importante en el proyecto, esta etapa permite la comunicación y el envío de la información. Para visualizar este envío se realiza por el archivo comprimido llamado DLL, la cual contiene toda la información necesaria y comandos de programación que permiten la comunicación y envío de los datos uniéndolos con la base de datos realizada. Este DLL trabaja con una trama de datos, de esta trama se captura la parte necesaria, el código Mac que transmite los Tags. Este código es unido con la base de datos mediante el comando que describe la DLL para realizar su conexión.

## **8. RESULTADOS**

Del proyecto de grado “IMPLEMENTACION DE SISTEMA RFID PARA REGISTRO DE COMPONENTES DEL LABORATORIO DE ELECTRONICA DE LA UPB”, se obtuvo un diseño de software de fácil manejo mediante la aplicación de Visual Basic.Net, que cumple con las características necesarias para brindar una mayor comodidad al operador del laboratorio de Ingeniería Electrónica. Mediante el software obtendrá control en el préstamo e inventario de los elementos de laboratorio. De igual forma la implementación de la comunicación de RFID para hacer de este servicio una actividad más eficiente.

Este proyecto cuenta con diferentes etapas las cuales parten del software con las características de cada alumno, la carrera, los instrumentos que se prestarán, además con la información de su último mantenimiento. Por otra parte se consigue el objetivo de implementar la tecnología RFID.

## 9. CONCLUSIONES

Con este proyecto se logró realizar la implementación de la comunicación por radiofrecuencia, incursionando en un campo tan importante que le permite a la Universidad Pontificia Bolivariana obtener una contribución más para el crecimiento de la institución.

La principal herramienta en este proyecto fue el computador, ya que por medio de él se pudo realizar el diseño del software que cuenta con la estructura de la base de datos y la comunicación necesaria para trabajar con un sistema RFID.

Es importante resaltar que la implementación de nuevas tecnologías en la Universidad Pontificia Bolivariana permite que se encuentre a la vanguardia, para encontrar soluciones a los problemas de seguridad en el préstamo de instrumentos del laboratorio de la Facultad de Ingeniería Electrónica.

La aplicación de la tecnología RFID, facilita la recolección de datos enlazados a un puerto de comunicación como el rs232 o el puerto USB; consiguiendo que este puerto sea direccionado a la aplicación basada en software.

Para direccionar los datos obtenidos en un puerto de comunicaciones a cualquier aplicación en software, es necesario contar con un hardware que permita la comunicación bidireccional de los dispositivos, enlazándolos en un buffer y después llevarlos mediante sentencias a los campos deseados en el software.

El reconocimiento de datos que se realiza mediante la tecnología RFID, es mucho mas practico a la hora de identificar un Código-Tags, ya que la lectura de estos se hace a distancias las cuales dependen de la frecuencia de fabricación de los dispositivos de almacenamiento.

En los dispositivos RFID, las distancias de reconocimiento están ligadas directamente a la frecuencia de operación de los Tags, haciendo que este sistema sea mucho más versátil en el momento de decodificación de datos, ya que al hacer el radio de cubrimiento del dispositivo, se podrá obtener un mayor monitoreo de los Tags para aplicaciones industriales.

## **10. RECOMENDACIONES**

La utilización de comunicación RFID podría ser utilizada en la universidad para el ingreso de los estudiantes a la institución.

Utilizar una antena que maneje un rango de frecuencia más alto, de esta forma se podría implementar el sistema de comunicación RFID para realizar un inventario logrando el escaneo de todos los elementos que contengan los Tags RFID, mitigando el tiempo en su realización.

Obtener la implementación de un software en el cual se le pueda dar una mayor seguridad a los datos, y así obtener una mejor visión de las diferentes herramientas que pueden ser utilizadas para observar los beneficios de un sistema de comunicación RFID.

## 11. BIBLIOGRAFIA

### 11.1 TOMADOS DE INTERNET

- [1] <http://es.wikipedia.org/wiki/RFID>
- [2] <http://www.ihg.net/java/X?cgi=lateral.rfid.ComoFunciona.pattern>
- [3] Patrick J. Sweeney, RFID for Dummies, Wiley Publishing, Inc 2005
- [4] [http://es.wikitel.info/wiki/RFID\\_\(Radio\\_Frequency\\_Identification\)](http://es.wikitel.info/wiki/RFID_(Radio_Frequency_Identification))
- [5] <http://www.ecojoven.com/dos/03/RFID.html>
- [6] <http://www.pcdoctor.com.mx/Radio%20Formula/temas/RFID%20ETIQUETAS%20DEL%20FUTURO.htm>
- [7] <http://www.ihg.net/java/X?cgi=lateral.rfid.VentajaVsBarras.pattern>
- [8] <http://www.trossenrobotics.com/store/p/4162-Inventory-Label-Square-Version-38-x-38-mm-.aspx>
- [9] <http://www.trossenrobotics.com/store/p/4140-APSX-RW-210-RFID-Reader-Writer.aspx>
- [10] <http://www.ecojoven.com/dos/03/RFID.html>
- [11] <http://www.trossenrobotics.com/store/p/4772-APSX-RFID-Kit-Icode-2-RW-210.aspx>
- [12] <http://www.techtear.com/2007/02/27/rfid-el-nuevo-codigo-de-barras/>
- [13] [http://www.egomexico.com/tecnologia\\_rfid.htm](http://www.egomexico.com/tecnologia_rfid.htm)
- [14] [http://www.pwc.com/ve/spa/pdf/Advisory\\_200712.pdf](http://www.pwc.com/ve/spa/pdf/Advisory_200712.pdf)



## 11.2 LIBROS

[12] RFID A GUIDE TO RADIO FREQUENCY IDENTIFICATION, V. Daniel Hunt, Albert Puglia, Mike Puglia

[13] RFID HANDBOOK Applications, Technology, Security and Privacy, Edit by SYED AHSON MOHAMMAD ILYAS

# **ANEXOS**

## **ANEXO 2**

Programación con

# Visual Basic .NET

Francisco Charte

INCLUYE  
CD-ROM

con multitud  
de ejemplos  
listos para utilizar

Aprenda a usar formularios  
Windows y Web usando los  
diseñadores de Visual Basic .NET

Adquiera las bases necesarias  
para utilizar GDI+, ADO .NET,  
ASP .NET y los demás servicios  
de la plataforma .NET

Conozca el nuevo lenguaje Visual  
Basic .NET, y sus características de  
orientación a objetos y componentes

ANAYA  
UNIVERSIDAD

## Introducción a SQL

---

Microsoft SQL Server es un sistema de bases de datos relacional que, como todos los demás, utilizan el lenguaje SQL para facilitar al usuario la ejecución de sentencias tanto de manipulación como de selección de datos. Conocer este lenguaje es, por tanto, básico para trabajar no ya con SQL Server, sino con cualquier otro RDBMS y, en general, para poder desarrollar aplicaciones con tratamiento de bases de datos en Visual Basic .NET.

SQL es un lenguaje no procedural y orientado a datos. Esto significa que no tiene estructuras de control como las de los lenguajes de programación, ya que su finalidad es devolver conjuntos de datos, modificarlos y, en el más amplio sentido, manipular datos. Aunque SQL es un lenguaje estándar, ampliamente aceptado, existen algunas diferencias puntuales entre el dialecto de SQL que entiende cada RDBMS. Las construcciones básicas, no obstante, son suficientemente genéricas como para funcionar en prácticamente cualquier sistema.

En los puntos siguientes no se tratará de explicar todos los detalles del lenguaje SQL, sino más bien transmitir los conceptos necesarios para que, en el capítulo siguiente, pueda desenvolverse sin problemas al tratar los comandos de bases de datos de ADO.NET.

## El Analizador de consultas SQL

---

Para poder hacer pruebas con el lenguaje SQL necesitamos saber dónde podemos introducir nuestras sentencias SQL para poder ejecutarlas y que el servidor efectúe las operaciones que le indiquemos. En este caso la herramienta que necesitamos es el **Analizador de consultas SQL** de SQL Server.

Para abrir el **Analizador de consultas SQL** deberá desplegar el menú **Herramientas del Administrador corporativo**, seleccionando la opción que puede verse en la figura 20.3. No confunda ésta con la opción **Analizador SQL** que hay debajo y que no tiene nada que ver aunque el nombre sea similar. Se trata de una herramienta diferente cuya finalidad es examinar el flujo SQL mientras se trabaja con una aplicación.

Aunque las posibilidades de esta herramienta son muchas, para el fin que nos ocupa tan sólo tiene que saber cómo introducir las sentencias SQL y cómo ejecutarlas. Como se aprecia en la figura 20.4, correspondiente a la ventana del **Analizador de consultas SQL**, la mayor parte del área de trabajo está ocupada por una ventana en la que podemos introducir texto. Será aquí, por tanto, donde tecleemos las sentencias a ejecutar. Cuando procesemos las órdenes, pulsando el botón que puede ver destacado encima o mediante la tecla F5, aparecerá un panel en la parte inferior con mensajes y resultados.

En la parte izquierda de la ventana hay una página que puede usar para explorar los diversos objetos que puede utilizar. De esta forma no tiene por qué memorizar los nombres de tablas, columnas, etc. Incluso puede tomar cualquiera de esos objetos y arrastrarlo hasta la ventana de texto para insertar el identificador correspondiente.

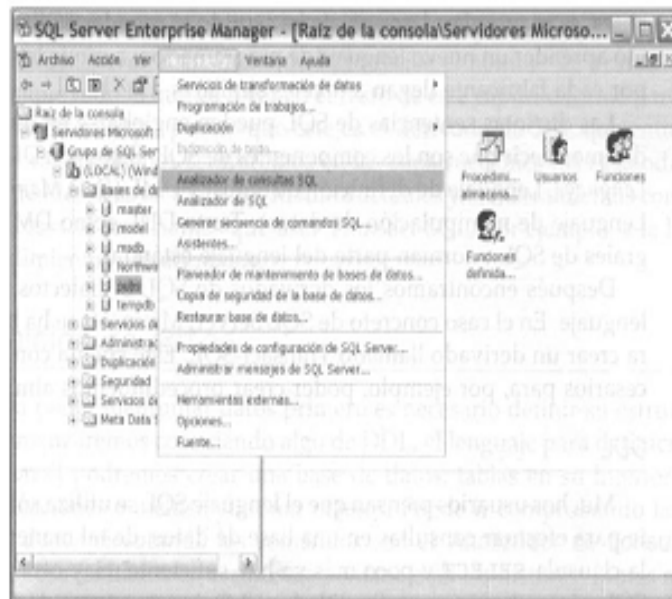


Figura 20.3. Opción para abrir el Analizador de consultas SQL

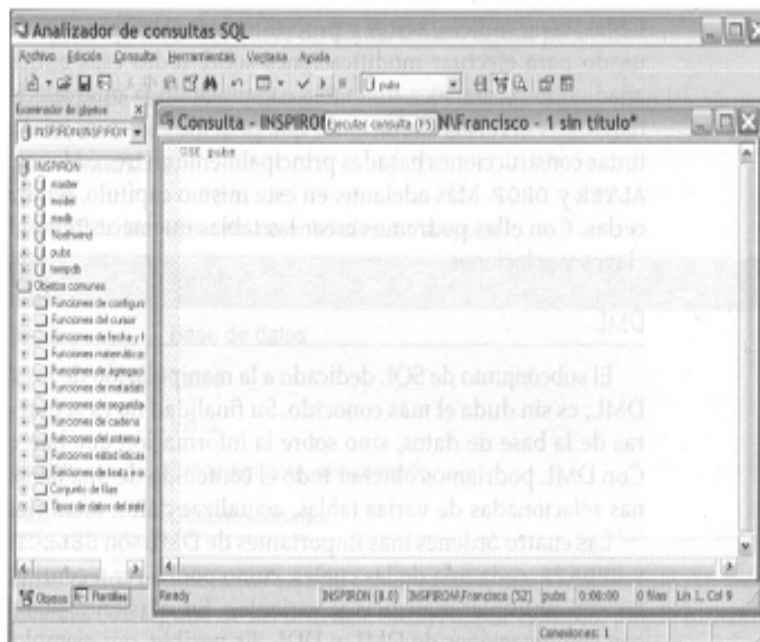


Figura 20.4. Aspecto del Analizador de consultas SQL

## Componentes y derivados de SQL

El lenguaje SQL (*Structured Query Language*, Lenguaje estructurado de consulta) cuenta con múltiples construcciones cuya finalidad, tal como se ha dicho

antes, es manipular datos en algún sentido. Aprender SQL no es tan difícil como aprender un nuevo lenguaje de programación, aunque los derivados de SQL por cada fabricante llegan a niveles similares.

Las distintas sentencias de SQL pueden englobarse en dos grupos, que podríamos decir que son los componentes de SQL. Éstos son DDL (*Data Definition Language*, Lenguaje de definición de datos) y DML (*Data Manipulation Language*, Lenguaje de manipulación de datos). Tanto DDL como DML son partes integrales de SQL, y forman parte del lenguaje estándar.

Después encontramos los derivados de SQL o dialectos basados en dicho lenguaje. En el caso concreto de SQL Server, Microsoft se ha basado en SQL para crear un derivado llamado Transact-SQL. Éste cuenta con los elementos necesarios para, por ejemplo, poder crear procedimientos almacenados.

---

## DDL

Muchos usuarios piensan que el lenguaje SQL se utiliza sólo y exclusivamente para efectuar consultas en una base de datos, de tal manera que conociendo la cláusula `SELECT` y poco más ya hay suficiente. Hay otra parte del lenguaje SQL, sin embargo, cuya finalidad es definir estructuras de datos. Es lo que se denomina Lenguaje de definición de datos o DDL.

Mediante este subconjunto de SQL podemos crear bases de datos, definir tablas, crear índices, vistas y procedimientos almacenados. También puede ser usado para efectuar modificaciones sobre todas esas entidades, por ejemplo añadiendo columnas a una tabla o cambiando la intercalación de una base de datos. Las diversas operaciones que podemos efectuar mediante DDL son distintas construcciones basadas principalmente en tres órdenes distintos: `CREATE`, `ALTER` y `DROP`. Más adelante, en este mismo capítulo, tendrá ocasión de conocerlas. Con ellas podremos crear las tablas que necesitemos, estableciendo sus claves y relaciones.

---

## DML

El subconjunto de SQL dedicado a la manipulación de los datos, denominado DML, es sin duda el más conocido. Su finalidad no es actuar sobre las estructuras de la base de datos, sino sobre la información que hay contenida en ellas. Con DML podríamos obtener todo el contenido de una tabla, combinar columnas relacionadas de varias tablas, actualizar datos, añadirlos o eliminarlos.

Las cuatro órdenes más importantes de DML son `SELECT`, `INSERT`, `DELETE` y `UPDATE`, cada una de las cuales, como podrá ver a continuación, puede contar con múltiples cláusulas y variantes. En ocasiones podemos encontrarnos con combinaciones de DML y DDL. Es posible, por ejemplo, utilizar una consulta para crear una nueva tabla. Estas construcciones en ocasiones tienen una sintaxis dependiente del RDBMS que esté utilizándose.

---

## Transact-SQL

Como se indicaba antes, cada sistema RDBMS suele usar un dialecto o derivado de SQL con características específicas que, en cierta manera, le permiten

aprovechar las singularidades de ese servidor en particular. En el caso de Microsoft SQL Server 2000, dicho dialecto recibe el nombre de Transact-SQL.

Aunque todo lo que va a ver en el resto de este capítulo forma parte del lenguaje Transact-SQL, puesto que éste es el derivado de SQL que entiende SQL Server, realmente nos centraremos en las construcciones más estándar, prácticamente idénticas en todos los RDBMS actuales y en herramientas como Microsoft Access. Sí tendríamos que usar Transact-SQL, por ejemplo, a la hora crear procedimientos almacenados.

## Definición de datos

Para poder manipular datos primero es necesario definir su estructura. Por ello comenzaremos conociendo algo de DDL, el lenguaje para definición de datos. Con él podremos crear una base de datos, tablas en su interior, efectuar modificaciones e incluso eliminar objetos. Puede ir comprobando las explicaciones dadas ejecutando los comandos con el **Analizador de consultas SQL**, viendo los resultados en esa misma ventana o en las distintas ramas mostradas en el **Administrador corporativo**.

Con DDL podremos crear elementos con la orden **CREATE**, modificarlos con la orden **ALTER** y eliminarlos mediante la orden **DROP**. Cada una de éstas tiene variantes para operar sobre bases de datos, tablas, vistas, procedimientos almacenados, etc.

Cualquiera de estas tres órdenes irá seguida de uno de los identificadores enumerados en la tabla 20.1. Con ellos se indica la entidad que va a crearse, modificarse o eliminarse.

**Tabla 20.1.** Identificación de entidades a manipular

Identificador	Entidad
DATABASE	Base de datos
TABLE	Tabla
VIEW	Vista
PROCEDURE	Procedimiento almacenado
TRIGGER	Desencadenador

Además de los enumerados en esta tabla, existen otros objetos que tan sólo son aplicables con alguna de las órdenes. **CREATE**, por ejemplo, puede utilizarse para crear un índice, pero éste no puede ser modificado con **ALTER**.

### Creación de una base de datos

La primera entidad que deberíamos crear es la propia base de datos, para lo cual usaremos la sentencia **CREATE DATABASE**. El único parámetro imprescindible es el nombre lógico de la base de datos, aunque opcionalmente pueden facilitarse muchos más.



Utilizando el Analizador de consultas SQL, la ejecución de la simple sentencia `CREATE DATABASE Pruebas` crearía una base de datos llamada `Pruebas` utilizando parámetros por defecto para los nombres de los archivos físicos, su tamaño, crecimiento y resto de propiedades. En la figura 20.5 puede ver el resultado obtenido tras ejecutar la sentencia. Puede utilizar el Administrador corporativo para comprobar que la base de datos ha sido creada.

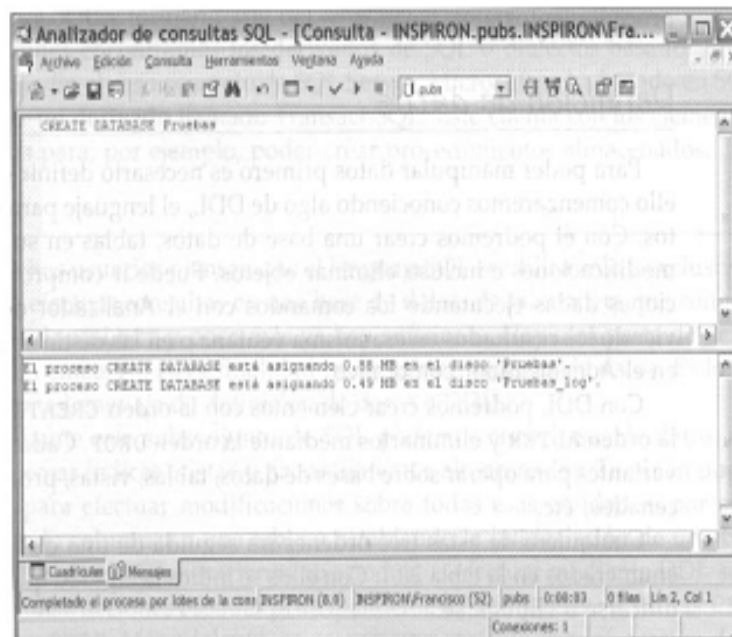


Figura 20.5. Podemos crear una nueva base de datos con una sentencia muy simple

En el momento de creación de la base de datos podemos especificar tanto el nombre del archivo de datos como el del archivo de registro de transacciones, pudiendo también establecer su tamaño inicial, tamaño máximo y el factor de crecimiento. Los parámetros para el archivo de datos se facilitarán, entre paréntesis, tras la cláusula `ON`, mientras que los del archivo de registro, que son equivalentes, lo harán detrás de las palabras `LOG ON`.

```
CREATE DATABASE nombre
ON (parámetros)
LOG ON (parámetros)
```

Los parámetros posibles, en ambos casos, son `NAME`, `FILENAME`, `SIZE`, `MAX-SIZE` y `FILEGROWTH`, facilitando el nombre lógico del archivo, su camino y nombre físico en disco, el tamaño inicial, el tamaño máximo y el crecimiento. La mayoría de estos parámetros son opcionales.

Observe la imagen mostrada en la figura 20.6. La primera sentencia que aparece, `DROP DATABASE Pruebas`, elimina la base de datos del servidor puesto que, de lo contrario, no podríamos volver a crearla. Debajo la creamos de nue-

vo estableciendo algunos de los parámetros del archivo de datos y de registro de transacciones.

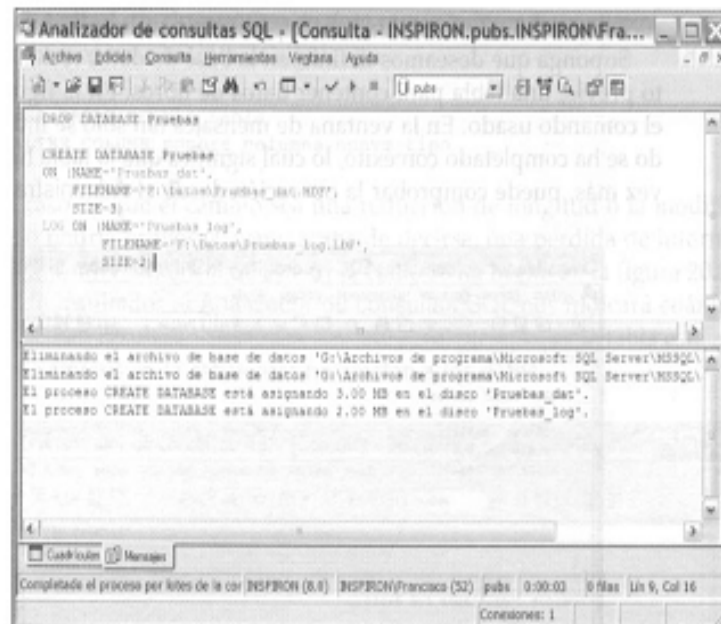


Figura 20.6. Al crear una base de datos podemos establecer los atributos de los archivos de datos y registro de transacciones

De forma similar, usando parámetros opcionales, podríamos establecer el orden de intercalación por defecto para la base de datos o definir un grupo de archivos de almacenamiento.

### Creación de tablas

Otra de las variantes de la orden `CREATE` es `CREATE TABLE` que, como es obvio, usaremos para definir tablas en el interior de una base de datos. El primer paso será seleccionar la base de datos en la que va a efectuarse la operación, para lo cual usaremos la sentencia `USE nombre_base`. A continuación ya podremos crear las tablas que necesitemos.

El parámetro más importante, y único imprescindible a la hora de crear una tabla aparte de su nombre, es la lista de columnas que deberá facilitarse entre paréntesis. Por cada columna debemos facilitar un nombre y un tipo. De forma adicional podríamos especificar si será la clave principal, cuenta con restricciones, hace referencia a una clave de otra tabla, etc. La sintaxis general, obviando todas esas posibilidades, es la siguiente:

```
CREATE TABLE nombre
(
  nombre_columna tipo,
  nombre_columna tipo ...
)
```

Lógicamente, para crear una tabla deberíamos conocer los tipos de datos que hay disponibles. Dos de los más habituales son `smallint` y `varchar`. El primero de ellos es un entero mientras que el segundo es una secuencia de caracteres, debiendo facilitarse entre paréntesis su longitud máxima.

Suponga que deseamos definir, en la base de datos recién creada en el punto previo, una tabla para contener datos de libros. En la figura 20.7 puede ver el comando usado. En la ventana de mensajes tan sólo se indica que el comando se ha completado con éxito, lo cual significa que la tabla ha sido creada. Una vez más, puede comprobar la operación desde el Administrador corporativo.

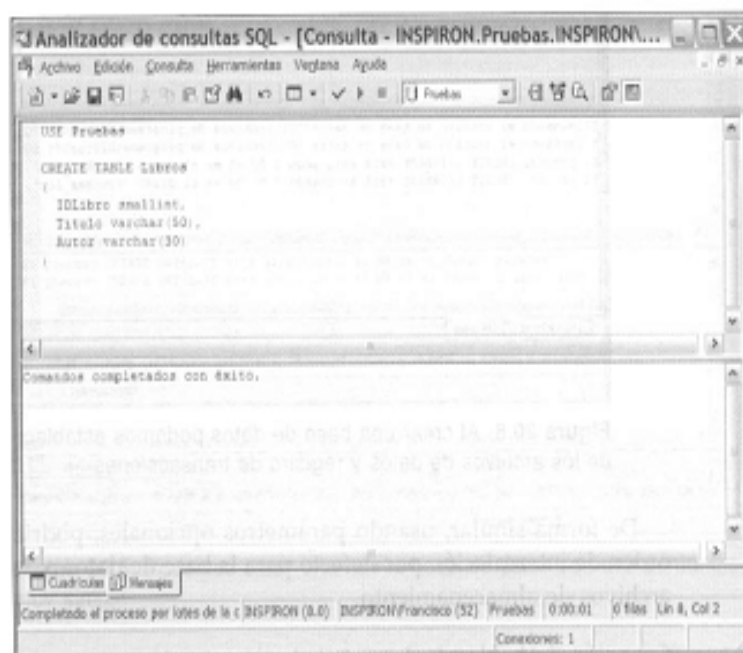


Figura 20.7. Seleccionamos la base de datos y creamos una tabla en ella

Utilizando opciones como `PRIMARY KEY`, `CHECK` y `DEFAULT` podríamos indicar qué columna actúa como clave principal, definir restricciones de datos o facilitar valores por defecto. Son algunas de las posibilidades existentes.

### Modificación de una tabla

Con una correcta planificación y un estudiado diseño las modificaciones a la estructura de una base de datos deberían ser, teóricamente, mínimas. Hay que tener en cuenta que este tipo de operaciones podrían, en determinados casos, provocar pérdidas de información, algo que no es admisible para una base de datos que esté utilizándose en un entorno de explotación.

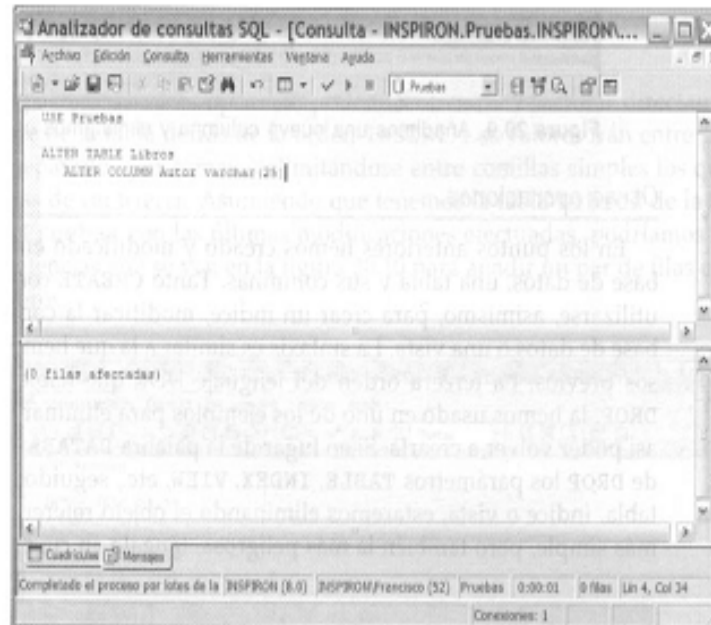
Para efectuar cambios en una tabla usaríamos la sentencia `ALTER TABLE` seguida del nombre de la tabla y las opciones que correspondan. Básicamente podemos efectuar tres operaciones distintas: añadir columnas, modificarlas o eliminarlas. También puede utilizarse esta construcción para modificar, elimi-

nar o añadir opciones de una columna, como pueden ser los valores por defecto o las restricciones.

Si deseamos cambiar una determinada columna, por ejemplo para modificar su longitud o incluso el tipo de dato que le habíamos asignado en un principio, debemos usar la sintaxis siguiente:

```
ALTER TABLE nombre_tabla
ALTER COLUMN nombre_columna nuevo_tipo
```

En caso de que el cambio sea una reducción de longitud o la modificación del tipo podría provocar, como acaba de decirse, una pérdida de información. Al ejecutar una sentencia de este tipo, como se ha hecho en la figura 20.8, en el panel de resultados el **Analizador de consultas SQL** nos indicará cuántas filas de datos han resultado afectadas. En este caso, puesto que la tabla está aún vacía, los cambios en la estructura no afectan a los datos.



**Figura 20.8.** Reducimos la longitud de una de las columnas de la tabla **Libros**

Las construcciones para añadir y eliminar columnas de la tabla son similares, como puede verse a continuación. En la figura 20.9 puede ver cómo se ha añadido una columna para el ISBN y se ha eliminado la columna que contendría el nombre del autor.

```
ALTER TABLE nombre_tabla
ADD nombre_columna tipo

ALTER TABLE nombre_tabla
DROP COLUMN nombre_columna
```

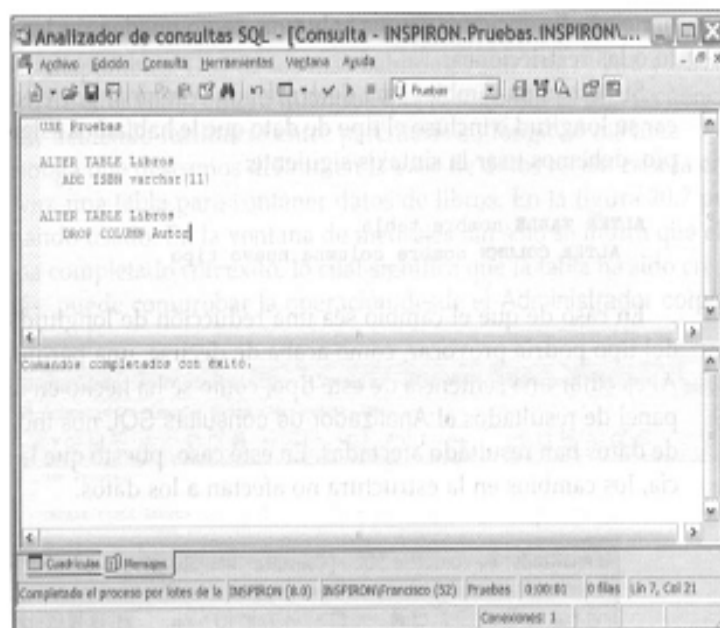


Figura 20.9. Añadimos una nueva columna y eliminamos una de las existentes

### Otras operaciones

En los puntos anteriores hemos creado y modificado entidades como una base de datos, una tabla y sus columnas. Tanto CREATE como ALTER pueden utilizarse, asimismo, para crear un índice, modificar la configuración de una base de datos o una vista. La sintaxis es similar a la que hemos conocido en pasos previos. La tercera orden del lenguaje DDL que habíamos mencionado, DROP, la hemos usado en uno de los ejemplos para eliminar la base de datos y así poder volver a crearla. Si en lugar de la palabra DATABASE ponemos detrás de DROP los parámetros TABLE, INDEX, VIEW, etc., seguidos del nombre de la tabla, índice o vista, estaremos eliminando el objeto referenciado. Es la orden más simple, pero también la más peligrosa, téngalo en cuenta.

#### Nota

Efectúe todas las pruebas que desee, con las distintas órdenes DDL que ha conocido, siempre sobre una base de datos de pruebas. No utilice las entregadas como ejemplo con SQL Server 2000, comopubs, y mucho menos las bases de datos de sistema que, comomaster, son imprescindibles para un correcto funcionamiento del servidor.

### Manipulación de datos

Como se indicaba anteriormente, las operaciones fundamentales de manipulación de datos son cuatro: selección, inserción, modificación y eliminación.

Estas cuatro operaciones se corresponden con las órdenes `SELECT`, `INSERT`, `UPDATE` y `DELETE`, respectivamente. Cada una de estas órdenes tiene una estructura básica, relativamente simple, que puede complementarse con diversas opciones.

A la hora de seleccionar datos, modificarlos o eliminarlos es posible, utilizando algunas de esas opciones, limitar la acción a un cierto conjunto de datos. Veamos primero cómo podemos efectuar las tareas generales y, después, entraremos en el estudio de algunas de esas opciones.

### Inserción de nuevas filas

Comenzaremos por la inserción de datos, ya que sin datos en las tablas no es posible efectuar operaciones de selección, actualización ni borrados. La orden a utilizar, por tanto, es `INSERT`, cuya sintaxis básica podemos ver a continuación.

```
INSERT INTO nombre_tabla  
VALUES (valor, valor ...)
```

La palabra `INTO` es opcional, podemos obviarla y facilitar directamente el nombre de la tabla detrás de la orden `INSERT`. Los valores irán entre paréntesis y separados por comas, delimitándose entre comillas simples los que sean cadenas de caracteres. Asumiendo que tenemos la tabla `Libros` de la base de datos `Pruebas` con las últimas modificaciones efectuadas, podríamos utilizar las sentencias que se ven en la figura 20.10 para añadir un par de filas con sendos libros.

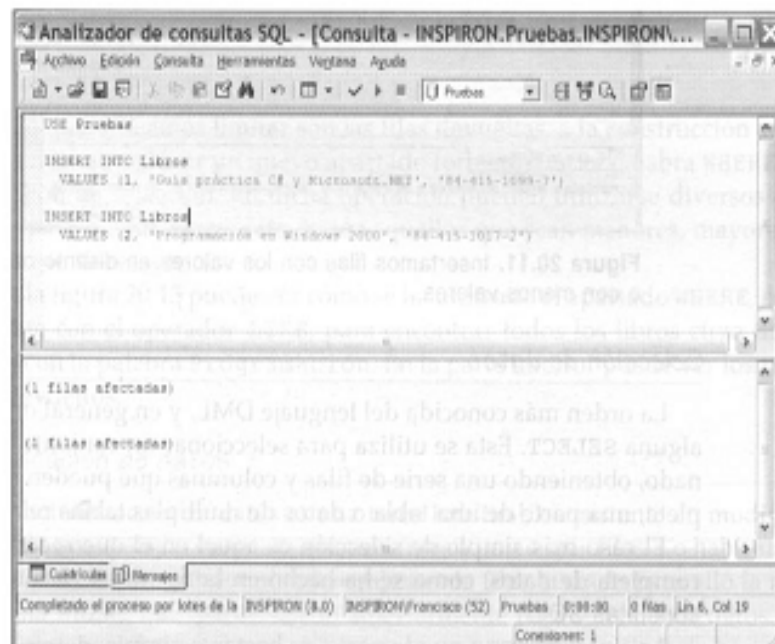
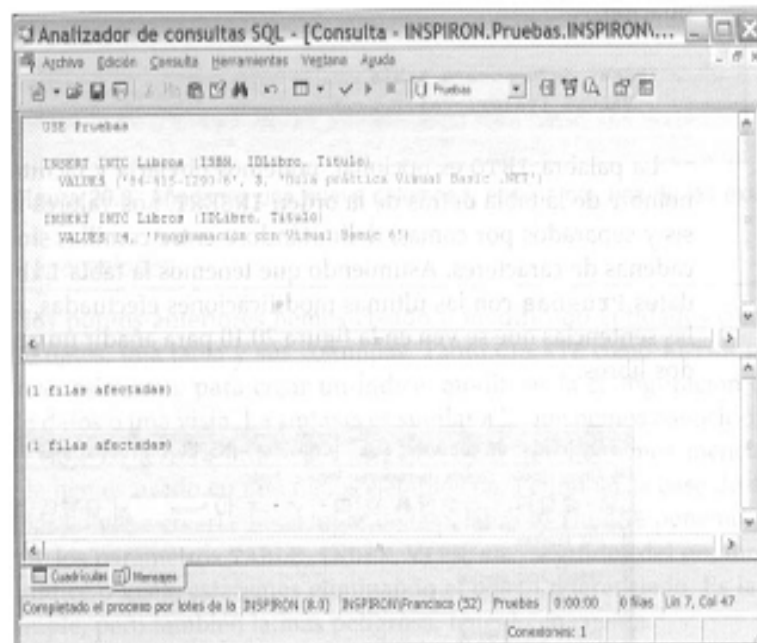


Figura 20.10. Insertamos dos filas de datos en la tabla

En este ejemplo se facilitan tantos valores como columnas existen en la tabla, respetando también el orden. Si no se entregan valores para todas las columnas, o el orden es distinto, entonces la sintaxis a utilizar sería la siguiente:

```
INSERT INTO nombre_tabla (nombre_columna, ...)  
VALUES (valor, ...)
```

La figura 20.11 presenta dos ejemplos en los cuales se utiliza esta sintaxis. En la primera inserción se facilitan los tres datos, pero en un orden diferente al que ocupan las columnas en la tabla. En el segundo caso se entregan valores tan sólo para dos de las tres columnas, por lo que la tercera tomará el valor NULL. Esto es posible porque al definir la tabla no hemos indicado que esta tabla no aceptase valores nulos.



**Figura 20.11.** Insertamos filas con los valores en distinto orden o con menos valores

### Selección de datos

La orden más conocida del lenguaje DML, y en general de SQL, es sin duda alguna `SELECT`. Ésta se utiliza para seleccionar un conjunto de datos determinado, obteniendo una serie de filas y columnas que pueden ser una tabla completa, una parte de una tabla o datos de múltiples tablas relacionadas entre sí.

El caso más simple de selección es aquel en el que recuperamos una tabla completa de datos, como se ha hecho en la figura 20.12 con el `Analizador de consultas SQL`.

La sintaxis, como puede verse, es bastante simple: detrás de `SELECT` disponemos un asterisco y tras la palabra `FROM` el nombre de la tabla.

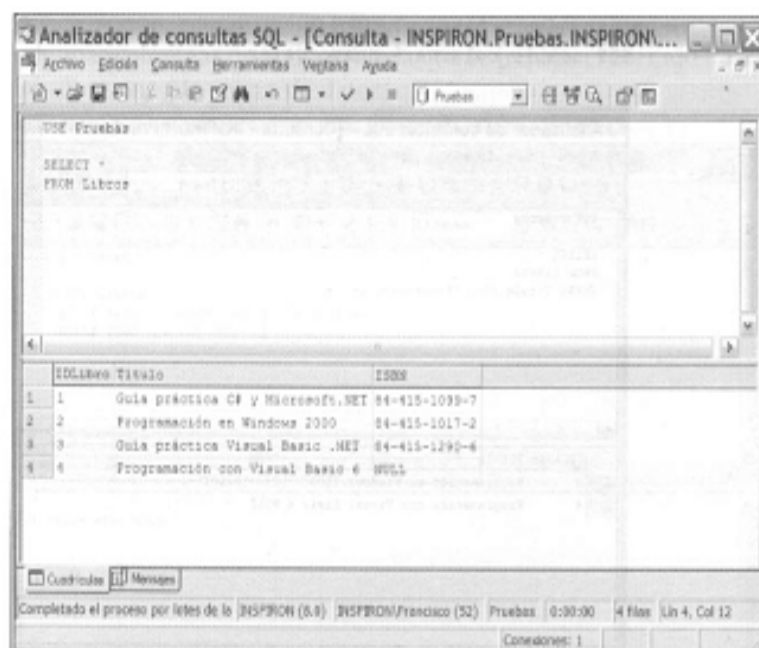


Figura 20.12. Seleccionamos todas las filas y columnas de la tabla Libros

Podemos limitar las columnas a recuperar sustituyendo el asterisco por los nombres de las columnas que nos interesen, separando unas de otras mediante comas. Con la sentencia siguiente, por ejemplo, obtendríamos todos los títulos e ISBNs de la tabla de libros, pero no sus códigos.

```
SELECT Titulo, ISBN
FROM Libros
```

Si lo que deseamos limitar son las filas devueltas, a la construcción anterior tendremos que añadir un nuevo apartado formado por la palabra **WHERE** y una expresión de selección. En dicha operación pueden utilizarse diversos operadores para buscar cierto dato, todos aquellos que sean menores, mayores, distintos, similares, etc.

En la figura 20.13 puede ver cómo se ha utilizado el apartado **WHERE**, conjuntamente con el operador **LIKE**, para encontrar todos los libros cuyo título se inicia con la palabra **Programación**. En la parte inferior puede ver los resultados devueltos.

#### Modificación de datos

La actualización de datos es una tarea habitual, consistente en modificar el valor de una o más columnas pertenecientes a una o más filas. Lo habitual es limitar la acción a un determinado conjunto de filas, usando para ello la misma cláusula **WHERE** que hemos usado anteriormente para seleccionar datos.

En este caso la orden que nos interesa es **UPDATE** y la sintaxis básica es la mostrada a continuación:



```

UPDATE nombre_tabla
SET nombre_columna = nuevo_valor

```

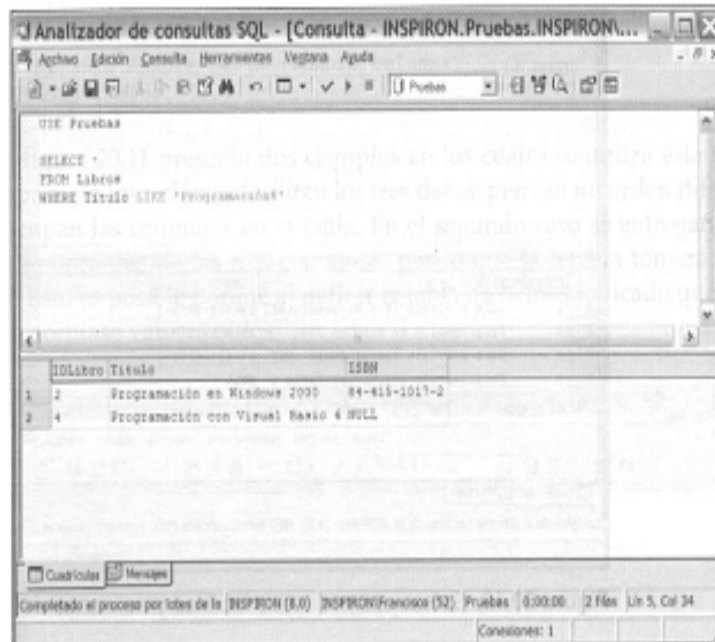


Figura 20.13. Seleccionamos algunas de las filas de la tabla Libros

Para facilitar el nuevo valor podemos usar un dato constante, como los que previamente hemos insertado en las filas de la tabla Libros, o una expresión. Podríamos, por ejemplo, sumar un determinado número de ejemplares a una hipotética tabla Almacen que ya tuviésemos creada.

Con la sintaxis anterior la actualización afectaría a todas las filas del conjunto de datos, sin excepción. Suponga que desea corregir el título de uno de los libros que ha introducido previamente, para lo cual pretende utilizar una sentencia como la siguiente:

```

UPDATE Libros
SET Titulo='Programación DNA en Windows 2000'

```

¿Qué conseguiría con esto? Perder los títulos de todos los libros que hay en la tabla y establecer el mismo para todos ellos. Para conseguir lo que quiere tiene que limitar la operación a la fila concreta sobre la que debe efectuarse el cambio, usando la cláusula WHERE y una expresión que identifique dicha fila de manera única. Podría, por ejemplo, comprobar el actual título o basarse en el ISBN, como se ha hecho en el ejemplo de la figura 20.14. A continuación puede ejecutar una consulta para verificar los cambios.

#### Eliminación de datos

La última operación que podemos efectuar, de las cuatro mencionadas con anterioridad, es el borrado de filas de una columna. A diferencia de lo que ocu-

re con la selección, actualización e inserción, casos en los que puede afectarse sólo a algunas columnas de la fila sobre la que se trabaja, el borrado tiene lugar siempre sobre una fila entera de datos.

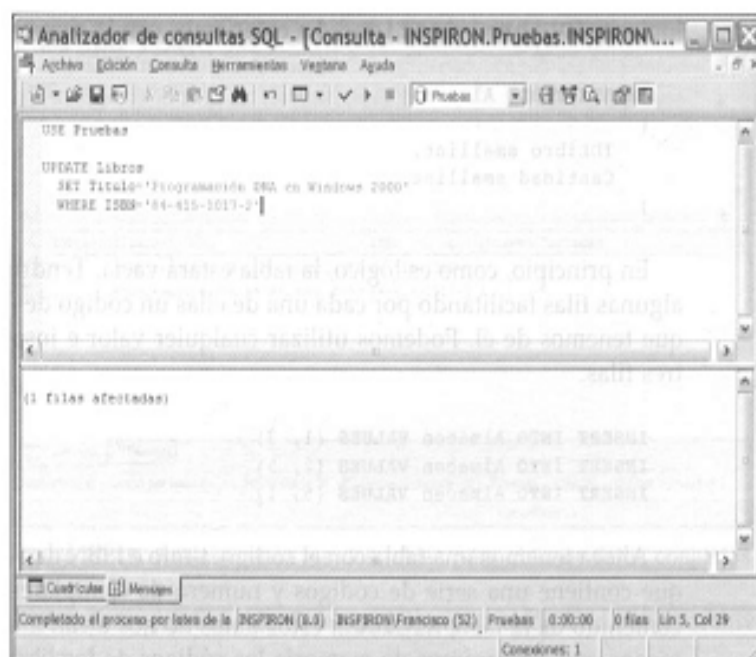


Figura 20.14. Cambiamos el título de uno de los libros

Podemos usar la orden `DELETE` tanto para borrar todo el contenido de una tabla, simplemente facilitando el nombre de ésta detrás, como para eliminar un determinado conjunto de filas. En este segundo caso deberemos seleccionar dicho conjunto usando la cláusula `WHERE`, como se ha hecho previamente con la actualización.

La sentencia para eliminar un libro concreto, por ejemplo facilitando su ISBN, podría ser la siguiente:

```
DELETE FROM Libros  
WHERE ISBN='84-415-1017-2'
```

Puede comprobar de inmediato, con una sentencia `SELECT`, el efecto causado sobre la tabla. Con ésta ya conoce las cuatro operaciones fundamentales, por lo que puede insertar datos, seleccionarlos, modificarlos y eliminarlos. En principio, es todo lo que necesita para trabajar con el Analizador de consultas SQL y hacer pruebas.

### Relaciones entre tablas

En este momento tenemos en nuestra base de datos de pruebas una sola tabla, por lo que difícilmente podemos relacionarla con ninguna otra. Para ver cómo podemos usar la orden `SELECT` para obtener datos de dos tablas tendremos

primero que crear otra, a continuación insertar datos en ella y, por último, ejecutar la consulta.

Supongamos que ahora queremos crear la tabla *Almacen* para contener el número de ejemplares que hay de cada título. Esta tabla tendría tan sólo dos columnas: *IDLibro* y *Cantidad*. La crearíamos con la siguiente sentencia SQL:

```
CREATE TABLE Almacen
(
  IDLibro smallint,
  Cantidad smallint
)
```

En principio, como es lógico, la tabla estará vacía. Tendríamos que insertar algunas filas facilitando por cada una de ellas un código de libro y la cantidad que tenemos de él. Podemos utilizar cualquier valor e insertar, por ejemplo, tres filas.

```
INSERT INTO Almacen VALUES (1, 3)
INSERT INTO Almacen VALUES (2, 5)
INSERT INTO Almacen VALUES (5, 1)
```

Ahora tenemos una tabla con el código, título e ISBN de varios libros, y otra que contiene una serie de códigos y número de ejemplares. Si obtenemos el contenido de la tabla *Almacen*, estos datos no nos dirán mucho por sí solos a menos que conozcamos de memoria los códigos de los libros. Es mucho más fácil combinar ambas tablas creando la oportuna relación que, en este caso, se basará en el código del libro. La sentencia sería la siguiente:

```
SELECT *
FROM Libros, Almacen
WHERE Libros.IdLibro = Almacen.IdLibro
```

Observe la expresión que sigue a la cláusula *WHERE*. En ella solicitamos aquellas filas de la combinación de tablas en que la columna *IdLibro* de la primera sea igual a la columna del mismo nombre en la segunda. En este caso ambas columnas tienen el mismo nombre, pero esto no es absolutamente necesario. El resultado, como puede ver en la figura 20.15, es que junto al título de cada libro ahora también vemos el número de ejemplares disponibles.

Si en lugar de *\** hubiésemos especificado el nombre de algunas columnas, tras la orden *SELECT*, el resultado podría ser más claro. Pruebe, por ejemplo, a solicitar tan sólo el título de cada libro y la cantidad disponible.

## Puntos clave

---

- Una base de datos es un recipiente lógico donde almacenar información, tanto datos como la estructura de esos datos y la lógica necesaria para tratarlos.

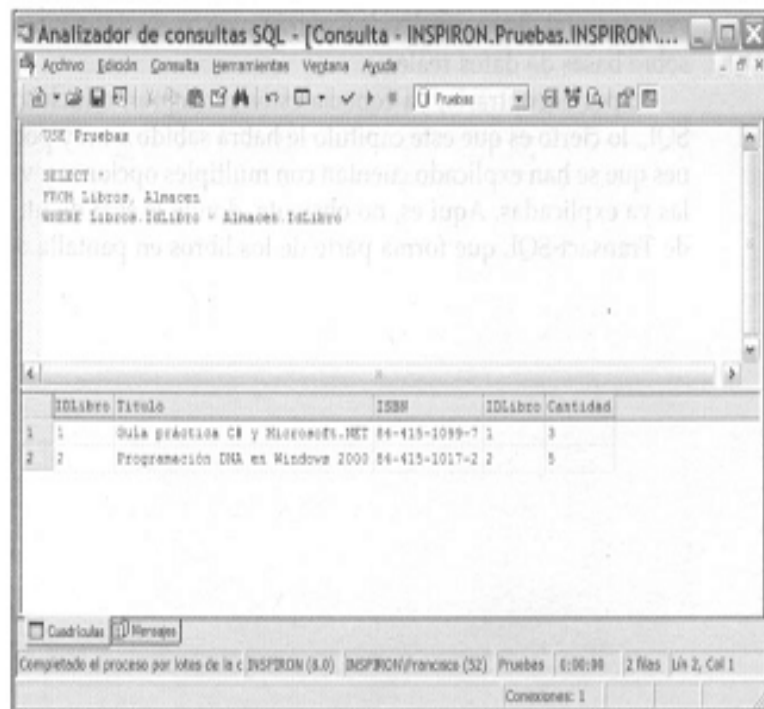


Figura 20.15. Relacionamos las dos tablas para obtener datos conjuntos

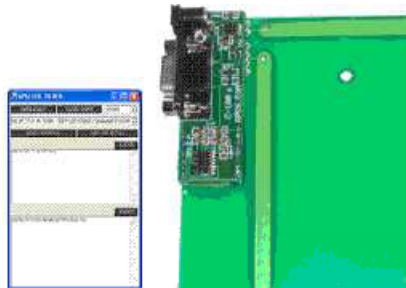
- Físicamente una base de datos está compuesta de uno o más archivos en disco.
- En la práctica nunca es necesario abrir esos archivos y trabajar directamente con ellos, porque las aplicaciones utilizan los RDBMS como intermediarios para la gestión de datos.
- Los elementos básicos de una base de datos son las tablas, conjuntos de filas de datos compuestas cada una de ellas de una serie de columnas.
- El acceso a los datos puede acelerarse definiendo los índices adecuados sobre las columnas de las tablas.
- Una base de datos también puede contener vistas, procedimientos almacenados y otros elementos de utilidad.
- Para trabajar con sistemas RDBMS es fundamental conocer el lenguaje SQL. Éste se compone de dos subconjuntos: DDL y DML, utilizándose el primero para manipular las estructuras de la base de datos y el segundo los datos propiamente dichos.

## **ANEXO 2**

# APSX

## RFID Reader/Writer Data Sheet

V 1.6



**APSX**

**Advanced Productivity Solutions, LLC.**

9435 Waterstone Blvd. Suite 140 Cincinnati, OH 45249

[info@apsx.com](mailto:info@apsx.com)

[www.apsx.com](http://www.apsx.com)

## FUNCTIONAL DESCRIPTION

### Hardware

- APSX RFID reader/Writers operate at 19200 baud rate.
- Built-in TTL. Need to use level converter to connect directly to PC (max232) or use APSX C-100 level converter.
- Read Range for RW310: up to 11" for credit card size tags, 5" for CD/DVD size labels, 4" for 24mm circular labels, and 6" for mini tags.
- Three different board dimensions: RW-310 (9" x 6" x 0.25") - RW-210 (4.5" x 2.7" x 0.25") – RW-110 (1 1/8" x 1 7/8" x 0.25")
- On-board antenna
- Power consumption 20mA at stand by, 80mA in fast mode (20 reads per sec.) under 5V
- Read and write any type of 13.56 MHz RFID tag (ISO15693)
- Allows user to program own cards
- Shorter transaction time than 125 kHz systems
- Programmable LEDs are available for visual feedback
- Mountable to any surfaces
- Operates on regulated 5VDC or 6-9 VDC when using with C-100

### Software

You can build your own .NET application with available DLL.

If you are not using VB.Net or do not want to use the DLL, you can use serial commands to directly talk to our devices.

Contents of the applications available at [www.apsx.com/support.aspx](http://www.apsx.com/support.aspx) page:

#### *APSX RFID Setup File (with .NET Framework)*

Installs APSX RFID TEST Application and .NET Framework if you do not have .NET Installed in your PC. Test application allows you experiment with all the features of the ISO15693 tag. It may take up to 5 minutes.

#### *APSX RFID Setup File (w/o .NET Framework)*

Installs only APSX RFID TEST Application. Test application allows you experiment with all the features of the ISO15693 tag.

#### *Help File (CHM)*

Provides useful information on key features of our readers, 13.56MHz memory organization, installation guide of Test Application, connection to PC, commands and source code.

#### *APSX Test Application Source Code*

It includes the whole package of codes, Visual Basic project files and DLL components.

### *RFID Direct Talk Application*

Includes two things: First, PDF data sheet which provides pin connection info, field descriptions, ISO15693 and APSX specific commands. Second, APSX serial talk software (serhex.exe)

### *Sample Project for VB6*

### **Software Installation (more detail available in Help File on support pages)**

1. Go to [www.apsx.com](http://www.apsx.com) support page
2. Download APSX TEST application to your PC
3. If you have .NET installed on your PC, you can download only APSX TEST application. Otherwise, you need to install the extended version with .NET Framework (dotnetfx.exe) in it. The extended version takes 5 minutes to install
4. When its installed click to APSX RFID TEST Application setup icon to start install. Dialog screen will appear. Click Next
5. Select you folder and user spec then click Next
6. Then click to confirm. Then system starts installing.
7. Click Close when you are done.

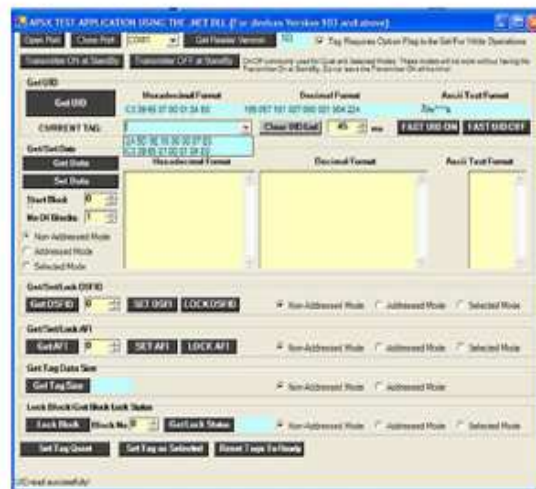
### **Connection to PC**

1. APSX RFID Readers come as TTL. However, if you use a TTL to RS232 converter (C-100 sold separately), you can connect directly to your PC. You will need a serial cable, standard 6V power adapter, APSX RFID reader, APSX TEST APPLICATION, 13.56MHz RFID tag to start testing
2. Connect the reader and PC with the serial cable
3. Connect the reader to the power adapter
4. Run the APSX TEST application
5. The main screen will appear

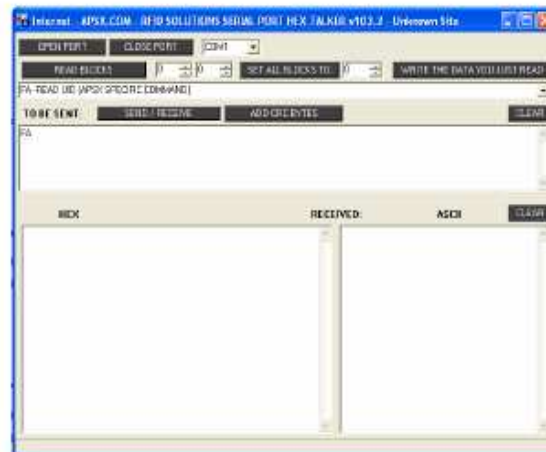


## SOFTWARE DESCRIPTION

### APSX RFID TEST Application Main Screen



### APSX Hex Talker Main Screen (serhex.exe)



**Field Descriptions**

Fields				
# of Bytes Sent	# of Bytes Expected	Flag	Command	CRC
8 bits	8 bits	8 bits		16 bits

**Example1:**

Read Command: 06 0C 27 01 00 2A 50

- 06: 6 bytes sent
- 0C: 12 bytes expected or needed
- 27: Flag (00100111 bin)
- 01: Inventory command (ISO 15693 standard)
- 00: Block number
- 2A 50: CRC16 (Low Byte & High Byte)

**Example2:**

Write Command: 0A 00 43 21 00 06 07 08 09 CD F6

- 0A: 10 bytes sent
- 00: No bytes expected
- 43: Flag
- 21: Write command
- 00: Block number
- 06 07 08 09: Four Bytes Data
- CD F6: CRC16 (Low Byte & High Byte)

## **Serial Talker Commands**

### *APSX Specific Commands*

#### **FF – Green Led On**

It turns the green led on. Whenever you need to use this function of the green led, you can simply use this command.

#### **FE –Green Led off**

It turns the green led off. Whenever you need to use this function of the green led, you can simply use this command.

#### **FD – Red Led On**

It turns the red led on. Whenever you need to use this function of the red led, you can simply use this command.

#### **FC – Red Led off**

It turns the red led off. Whenever you need to use this function of the red led, you can simply use this command.

#### **FB – Switch to Fast Mode**

Fast mode is the special mode that the reader tracks the proximity area continuously. When you need to read tags whenever they enter into that proximity area you can use this mode.

#### **FA – Read UID**

This allows you to read the unique ID of each tag. It gives you back UID of a specific tag.

#### **F7 – Transmitter Off**

#### **F8 – Transmitter On**

#### **F9 – Get Version Number**

## ISO 15693 Standard Commands

### Get UID

It's the same command as FA. For this command you need to put the variables by yourself. The number of bytes you sent and the number of bytes you need to read are up to you.

Example:

- Open port
    - Select Get UID command from the dropdown menu
- Command looks like this:  
06 0C 27 01 00 2A 50
- 06: 6 bytes sent  
0C: 12 bytes expected  
27: Flag (00100111 bin)  
01: Inventory command (ISO 15693 standard)  
00: Block number  
2A 50: CRC16 (Low & High)
- Place RFID Tag close enough to the reader
  - Press Send/Receive button
  - On the bottom screen you will see the UID of that specific tag

Response would be like this (12 Bytes):  
00:00:C3:27:D5:06:00:00:07:E0:FA:23:

### Write Block

This command writes on the desired block. Blocks have the range of "0-64". Each block has four bytes. Therefore four bytes are needed for each block.

Example:

- Open port
- Select Write Block command from the dropdown menu

Command looks like this:

0A 00 43 21 00 06 07 08 09 CD F6

0A: 10 bytes sent

00: No bytes expected

43: Flag

21: Write single block command

00: Block number

06 07 08 09: Four Bytes Data (one block)

CD F6: CRC

- Put RFID Tag close enough to the reader
- Press Send/Receive button

### Read Block

This command reads the desired block from the tag. Blocks have the range of "0-64". Each block has four bytes.

Example:

- Open port
- Select Get UID command from the dropdown menu  
Command looks like this: 06 07 03 20 00 9B 0A
  
- 06: 6 bytes sent (Not including this byte)
- 07: Seven bytes expected
- 03: Flag
- 20: Read single block command (ISO 15693 standard)
- 00: Block number
- 9B 0A: CRC (Low & High)
  
- Place RFID Tag close enough to the reader
- Press Send/Receive button
- On the bottom screen you will see the data of that specific tag

Response would be like this (7 Bytes):

00:06:07:08:09:E9:5B:

00: Block number  
06 07 08 09: Four bytes data  
E9 5B: CRC

**APSX RFID TEST Application Commands (more available at [www.apsx.com/support.aspx](http://www.apsx.com/support.aspx))**

When APSX RFID TEST application is running, hit OPEN PORT button by selecting your port such as COM1

**VERSION**

If you want to see the version number, click to GET READER VERSION.

**UID**

You can click to GetUID button to get UID number of a tag. You can see in three format including hexadecimal, decimal, Ascii. When you start reading different tags, they are stored in the combo box right under the hexadecimal UID box. You can also decide which tag to work on by selecting the tag as current from the combo box.

**FAST UID**

The other important feature is reading UIDs in fast mode. You can read UID non-stop by clicking FAST UID ON button once. The close this feature just click FAST UID OFF button.

**TAG DATA**

*Non-Address Mode*

Non-Addressed mode indicates that the tag in the range will be read or written without considering selected or addressed modes.

In order to get the data on the tag, just click the GET DATA button. There are hexadecimal, decimal and Ascii formats available. If you want to get some part of the data, you can select START BLOCK and NO OF BLOCKS indicators to specify the starting block and how many blocks you want to read. Maximum number of blocks to read is 12.

The same logic applies for data writing purposes. You need to type your data in the hexadecimal format box by leaving a space between each data. Then you click SET DATA button to write on the tag.

*Addressed Mode*

Addressed mode indicates that the tag addressed will be read or written no matter which tag is in the range or selected. Addressed tag can be selected by using the ADDRESSED TAG combo box next to GET UID button.

#### *Selected Mode*

Selected mode indicates that the tag selected will be read or written no matter which tag is in the range or addressed. A tag can be selected by using the SET TAG AS SELECTED button at the very bottom of the screen.

#### DSFID

DSFID can be read or written by using GET DSFID or SET DSFID buttons. The ID you want to set can be entered into the box before you hit the SET DSFID button. The other feature is to lock DSFID. When you lock it, it can not be changed again. The modes (Non-Addressed, Addressed, Selected) are applied by the same logic UID has.

#### AFI

AFI can be read or written by using GET AFI or SET AFI buttons. The ID you want to set can be entered into the box before you hit the SET AFI button. The other feature is to lock AFI. When you lock it, it can not be changed again. The modes (Non-Addressed, Addressed, Selected) are applied by the same logic UID has.

#### TAG SIZE

Tag size can be read by using GET TAG SIZE button. The modes (Non-Addressed, Addressed, Selected) are applied by the same logic UID has.

#### LOCK BLOCK

You can lock and get the lock status of any block. But when you lock it, there is no way to unlock it. Select the block you want to lock then hit LOCK BLOCK button. If you want to know the status of any block you want, you can select the block and then hit GET LOCK STATUS button.

#### TRANSMITTER ON/OFF

You need to leave the transmitter on for tags to stay in any mode they need to be powered and be in the range. Otherwise they will lose power and reset. You can use the buttons on the top part of the RFID Test Application. You can use F8(HEX) to turn on and F7(HEX) to turn off.



## Pin Definitions

In order to connect the reader to your system, you need to follow these guidelines for the wire connections:

- Pin 1: Ground
- Pin 2 & 3: Not Used
- Pin 4: 5V (Regulated)
- Pin 5: TX (Serial Transmit)
- Pin 6: RC (Serial Receive)
- Pin 7: Ground
- Pin 8: Not Used

## **Appendix**

### **Serial Talker Command Summary**

FF – Green LED on  
FE – Green LED off  
FD – Red LED on  
FC – RED LED off  
FB – Switch to Fast Mode  
FA – Read UID  
Get UID  
Read Block  
Write Block  
TRANSMITTER ON F8(HEX) / OFF F7(HEX)  
F9 – Get the Version Number  
And any other ISO 15693 command

### **APSX RFID TEST Application Command Summary**

VERSION  
UID  
FAST UID  
TAG DATA  
    *Non-Address Mode*  
    *Addressed Mode*  
    *Selected Mode*  
DSFID  
AFI  
TAG SIZE  
LOCK BLOCK

### **Product dimensions**

#### **APSX RW-110 13.56MHz RFID Reader Module Board**

Dimensions (W X L): 1 1/8" X 1 7/8"

#### **APSX RW-210 13.56MHz RFID Reader Module Board**

Dimensions (W X L): 2.70" X 4.50"

#### **APSX RW-310 13.56MHz RFID Reader Module Board**

Dimensions (W X L): 6.00" X 9.00"

