

**UNIVERSIDAD INTERNACIONAL DEL ECUADOR  
FACULTAD INGENIERÍA AUTOMOTRIZ**

**TESIS DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN MECÁNICA AUTOMOTRIZ**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE  
RECONOCIMIENTO POR VOZ PARA EL ACCIONAMIENTO DE  
ACCESORIOS EN UN AUTOMÓVIL.**

**ANDRÉS EDUARDO ESCOBAR SÁNCHEZ  
CARLOS ANDRÉS YUMISEVA CABRERA**

**DIRECTOR: ING. JUAN FERNANDO IÑIGUEZ**

**MARZO, 2012**

**Quito, Ecuador**

## CERTIFICACIÓN

Nosotros, ANDRÉS EDUARDO ESCOBAR SÁNCHEZ, CARLOS ANDRÉS YUMISEVA CABRERA, declaramos que somos autores exclusivos de la presente investigación y que ésta es original, auténtica y personal nuestra. Todos los efectos académicos y legales que se desprendan de la presente investigación serán de mi exclusiva responsabilidad.

.....  
ANDRÉS EDUARDO ESCOBAR SÁNCHEZ

CI: 160040257-0

.....  
CARLOS ANDRÉS YUMISEVA CABRERA

CI: 172270766-6

Yo, JUAN FERNANDO IÑIGUEZ, declaro que, en lo que yo personalmente conozco, los Señores ANDRÉS EDUARDO ESCOBAR SÁNCHEZ, CARLOS ANDRÉS YUMISEVA CABRERA, son los autores exclusivos de la presente investigación y que ésta es original, auténtica y personal suya.

.....  
ING. JUAN FERNANDO IÑIGUEZ

DIRECTOR DE TESIS

## **AGRADECIMIENTO**

Los resultados del desarrollado y culminación de este proyecto, se los agradezco a quienes han sido parte fundamental del mismo.

Al Ingeniero Juan Fernando Iñiguez ya que ha sabido darnos consejos y guiarnos, en los momentos de dudas, porque sin su ayuda habría sido muy complicado haber culminado con este proyecto.

A la Facultad de Ingeniería Automotriz y por ende a la Universidad Internacional Del Ecuador, ya que allí he recibido los conocimientos necesarios de parte de todos quienes han sido mis profesores, preparándome en lo académico para el desarrollo profesional y como persona de bien.

Andrés Escobar

## **DEDICATORIA**

Dedico este proyecto de grado a Dios y a mis Padres, quienes han sido el pilar fundamental en mi vida y mi etapa como estudiante.

A mis padres Eduardo Escobar y Fanny Sánchez, por su apoyo incondicional en todo momento, también han sido quienes me han inculcado valores para ser una persona de bien en la sociedad, además de haberme dado la educación y cuidado en los momentos de enfermedad. Es a ellos a quienes les debo lo que soy ahora.

Andrés Escobar

## **AGRADECIMIENTO**

Al Director de tesis Juan Fernando Iñiguez por guiarnos en el desarrollo de la misma y a la Facultad de Ingeniería Mecánica Automotriz de la UIDE.

Andrés Yumiseva

## **DEDICATORIA**

A mi familia; en especial a mis abuelos, padres y hermana por su cariño y apoyo incondicional.

Andrés Yumiseva

## ÍNDICE GENERAL

CAPÍTULO 1 .....	1
INTRODUCCIÓN .....	1
1.1. ANTECEDENTES .....	1
1.2. IMPORTANCIA .....	2
1.3. OBJETIVOS .....	3
1.3.1. Objetivos Específicos.....	3
1.4. JUSTIFICACIÓN .....	4
CAPÍTULO 2 .....	6
MARCO TEÓRICO.....	6
2.1. LA VOZ HUMANA Y EL RECONOCIMIENTO DE VOZ.....	6
2.1.1. Señal de voz .....	6
2.1.1.1. Aparato fonador.....	7
2.1.1.2. Formantes .....	8
2.1.2. Cualidades del Sonido .....	8
2.1.3. Clasificación de los Sonidos .....	9
2.1.3.1. Vocales y Consonantes .....	9
2.1.3.2. Oralidad y nasalidad.....	9
2.1.3.3. Tonalidad.....	10
2.1.4. ¿Qué es el reconocimiento de voz?.....	10
2.1.4.1. Software disponible para reconocimiento de voz .....	10
2.1.5. Procesamiento digital de señales .....	12
2.1.6. El procesamiento de la información y la electrónica de potencia.....	13
2.1.6.1. Conversión de señales de formato analógico a digital.....	14
2.2. ELECTRÓNICA DIGITAL.....	15
2.2.1. Puertas lógicas .....	17
2.2.1.1. Puerta NOT (NO).....	17

2.2.1.2.	Puerta OR (O).....	18
2.2.1.3.	Puerta AND (Y).....	18
2.2.1.4.	Puerta NOR (NO-O) .....	19
2.2.1.5.	Puerta NAND (NO-Y).....	19
2.2.1.6.	Puerta OREX (O-Exclusiva) .....	20
2.2.1.7.	Puerta NOR-EX .....	20
2.2.2.	Sistemas Digitales .....	21
2.2.2.1.	Desarrollo de sistemas digitales .....	22
2.3.	REPRESENTACIÓN BÁSICA DEL SISTEMA .....	24
2.4.	DISEÑO DEL BLOQUE DE COMUNICACIÓN CON EL ORDENADOR..	25
2.4.1.	La Norma RS-232 .....	25
2.4.2.	Puerto Serie RS-232.....	26
2.4.2.1.	Ventajas y desventajas del Puerto Serie RS-232 .....	28
2.4.3.	Cable Adaptador USB a Serial RS-232 DB-9 .....	29
2.4.4.	Velocidad de transmisión de datos .....	30
2.4.4.1.	El Baudio .....	30
2.4.5.	Señales.....	31
2.4.5.1.	DTE y DCE .....	32
2.4.5.2.	Nueve señales del RS-232 .....	33
2.4.6.	Niveles lógicos RS-232 .....	34
2.4.7.	Elementos eléctricos y electrónicos .....	35
2.4.8.	Conexión Puerto RS232, MAX232 y PIC16F877A .....	37
2.4.8.1.	Circuito Integrado .....	38
2.4.8.2.	MAX232 .....	39
2.4.8.3.	¿Qué es un Microcontrolador? .....	42
2.4.8.3.1.	Microcontrolador PIC 16F877A .....	43
2.4.8.4.	Capacitor .....	45

2.5.1.	ULN 2803.....	47
2.5.2.	Relé .....	48
2.5.3.	Diodo 1N4007 .....	48
2.6.	DISEÑO DEL BLOQUE DE ALIMENTACIÓN.....	49
2.6.1.	Regulador de voltaje 7805 .....	49
2.7.	SOFTWARE .....	50
2.7.1.	Programación.....	51
2.7.2.	Tipos de Software .....	52
2.7.2.1.	Diferencia entre el lenguaje Basic y el lenguaje ensamblador.....	54
CAPÍTULO 3 .....		56
DISEÑO, CONSTRUCCIÓN DE LA PLACA PCB Y SELECCIÓN DE ELEMENTOS .....		56
3.1.	DISEÑO DE LA PLACA PCB .....	56
3.1.1.	Desarrollo en ISIS.....	56
3.1.2.	Desarrollo en ARES .....	57
3.2.	CONSTRUCCIÓN DE LA PLACA PCB.....	58
3.2.1.	Fabricación del circuito .....	58
3.2.2.	Montaje y soldadura de elementos .....	63
3.2.2.1.	Soldadura .....	63
3.2.2.2.	Elementos adicionales.....	65
3.2.2.2.1.	Bornera de alimentación de 2 pines .....	65
3.2.2.2.2.	Borneras de salida de 2 pines .....	66
3.2.2.2.3.	Bases para Circuitos Integrados .....	66
3.3.	SELECCIÓN DE ELEMENTOS ELECTRÓNICOS .....	67
3.4.	SELECCIÓN DEL SOFTWARE .....	69
CAPÍTULO 4 .....		71
DISEÑO DEL SOFTWARE .....		71

4.1. TRABAJO DESARROLLADO CON EL RECONOCIMIENTO DE VOZ DE WINDOWS Y DRAGON NATURALLY SPEAKING 10.....	71
4.1.1. Proceso de activación (Reconocimiento de voz de Windows).....	71
4.1.2. Proceso de instalación de Dragon Naturally Speaking 10 .....	73
4.2. MICROCODE STUDIO .....	81
4.3. PIC - kit 2 .....	82
4.4. VISUAL BASIC VERSIÓN 6.0 Y VERSIÓN 2010 .....	83
4.4.1. Aplicaciones.....	85
4.4.2. Controles.....	86
4.4.3. Formularios / Hojas de diseño .....	86
4.4.4. Objetos y propiedades .....	87
4.4.5. Eventos y métodos .....	87
4.4.6. Proyecto.....	87
4.5. MICROSOFT NET FRAMEWORK.....	88
4.6. PROTEUS.....	88
4.6.1. ISIS (Generación plano eléctrico) .....	88
4.6.2. ARES (Generación circuito PCB).....	89
CAPÍTULO 5 .....	90
IMPLEMENTACIÓN DEL SISTEMA, PRUEBAS Y RESULTADOS.....	90
5.1. IMPLEMENTACIÓN DEL SISTEMA EN EL VEHÍCULO.....	90
5.2. PRUEBAS .....	95
5.2.1. Prueba de precisión de reconocimiento de voz ante ruido ambiental ...	96
5.2.2. Prueba de autonomía de batería del computador usando el sistema...	97
5.3. RESULTADOS.....	98
CONCLUSIONES.....	101
RECOMENDACIONES .....	102
BIBLIOGRAFÍA .....	103
ANEXOS .....	105

Anexo 1. Programación del PIC .....	105
Anexo 2. Programación software Visual Basic 6.0 .....	111
Anexo 3. Elaboración software Visual Basic 2010 .....	129
Anexo 4. Diagrama eléctrico conexión PC - PIC .....	143
Anexo 5. Diagrama de alimentación Placa PCB .....	144
Anexo 6. Diagrama eléctrico conexión PIC - Relés .....	145
Anexo 7. Diagrama eléctrico conexión Relés - Accesorios .....	146
Anexo 8. Diagramas eléctricos de los accesorios del vehículo .....	147

## ÍNDICE DE FIGURAS

Figura 2.1. Partes del Aparato Fonador .....	7
Figura 2.2. Las señales analógicas toman valores continuos de amplitudes. Las señales digitales toman unas pocas amplitudes discretas.....	14
Figura 2.3. Conversión de una señal analógica en un equivalente digital aproximado mediante muestreo. Cada valor de muestra viene representado por un código de 3 bits. Los convertidores reales utilizan palabras de código más largas. ....	15
Figura 2.4. Niveles lógicos .....	16
Figura 2.5. Funcionamiento de compuerta lógica.....	17
Figura 2.6. Compuerta NOT .....	17
Figura 2.7. Compuerta OR .....	18
Figura 2.8. Compuerta AND .....	18
Figura 2.9. Compuerta NOR.....	19
Figura 2.10. Compuerta NAND .....	19
Figura 2.11. Compuerta OREX .....	20
Figura 2.12. Compuerta NOR-EX.....	20
Figura 2.13. Pasos usuales para el desarrollo de sistemas digitales .....	22
Figura 2.14. Representación esquemática Del sistema. ....	24
Figura 2.15. Niveles de voltaje RS-232 y TTL .....	26
Figura 2.16. Conectores RS232: a) DB-25 macho-hembra. b) DB-9 macho - hembra.....	27
Figura 2.17. Patillaje del conector DB-9 .....	28
Figura 2.18. Cable Adaptador USB a DB9 .....	30
Figura 2.19. Niveles de tensión lógicos para RS-232.....	34
Figura 2.20. Conexión PC - Puerto RS-232 - MAX-232 y PIC16f877A .....	38
Figura 2.21. Niveles RS 232 y niveles TTL .....	40

Figura 2.22. Circuito integrado MAX232 .....	41
Figura 2.23. Configuración interna y operación típica del circuito integrado MAX232 .....	41
Figura 2.24: Microcontrolador PIC 16F877A .....	42
Figura 2.25. La arquitectura Harvard maneja la memoria de datos y la memoria de instrucciones por separado y con diferentes capacidades.....	43
Figura 2.26. Capacitores de 0.1 y 1uf .....	45
Figura 2.27. Diagrama de conexión PIC - ULN2803 - Relé.....	46
Figura 2.28. Descripción de pines del circuito integrado ULN2803 .....	47
Figura 2.29. Vista superior del relé y el diagrama eléctrico del mismo.....	47
Figura 2.30. Diodo.....	48
Figura 2.31. Configuración del patillaje del regulador 7805 .....	48
Figura 2.32. Esquema de bloques de la conexión de los reguladores 7805 .....	49
Figura 2.33. Pasos en la traducción del software.....	50
Figura 2.34. Cuadro de los niveles de programación .....	54
Figura 3.1. Plano eléctrico diseñado en ISIS .....	56
Figura 3.2. Vista superior de la placa PCB diseñada en ARES.....	57
Figura 3.3. Diagrama que se imprime en el papel foto sensible (vista inferior de la placa) .....	58
Figura 3.4. Vista superior e inferior de la placa PCB virgen .....	58
Figura 3.5. Vista inferior de la placa PCB terminada.....	60
Figura 3.6. Diagrama que se imprime en el papel foto sensible (vista superior de la placa) .....	61
Figura 3.7. Vista superior de la placa PCB terminada (ubicación de elementos) .	61
Figura 3.8. Soldadura de los materiales eléctricos en la placa PCB .....	62
Figura 3.9. Vista de la soldadura final de los elementos .....	63
Figura 3.10. Visualización 3D placa PCB.....	63

Figura 3.11 Vista Real placa PCB terminada .....	64
Figura 3.12. Vistas frontal y superior de la bornera de 2 pines .....	64
Figura 3.13. Vista frontal del grupo de borneras de salida de la placa PCB .....	65
Figura 3.14. Bases para circuitos integrados de 16, 18 y 40 pines respectivamente .....	65
Figura 4.1. Centro de accesibilidad de Windows.....	71
Figura 4.2. Opciones de reconocimiento de voz .....	71
Figura 4.3. Interfaz gráfica de reconocimiento de voz Windows .....	72
Figura 4.4. Tipo de instalación; Seleccionada Típica/Completa.....	73
Figura 4.5. Asistente para crear usuarios.....	73
Figura 4.6. Indicación de que debe conectar el micrófono .....	74
Figura 4.7. Calibración del volumen del micrófono.....	74
Figura 4.8. Comprobación de la calidad de sonido .....	75
Figura 4.9. Prueba de calidad de sonido completada .....	76
Figura 4.10. Asistente para entrenamiento de su voz .....	76
Figura 4.11. Elección de texto para lectura .....	77
Figura 4.12. Selección para leer el texto .....	77
Figura 4.13. Ventana para iniciar entrenamiento de voz .....	78
Figura 4.14. Programa se encuentra Grabando Su Voz .....	78
Figura 4.15. Adaptación del Programa a la grabación de voz del usuario .....	79
Figura 4.16. Configuración completada.....	79
Figura 4.17. Barra de Herramientas DragonBar.....	80
Figura 4.18. MicroCode Studio.....	81
Figura 4.19. PICkit 2.....	82
Figura 4.20. Visual Basic 6.0.....	83
Figura 4.21. Visual 2010 .....	83
Figura 5.1. Caja protectora PCB .....	89

Figura 5.2. Ubicación Placa PCB .....	90
Figura 5.3. Desmontaje Paneles Puertas .....	91
Figura 5.4. Desmontaje consola volante .....	91
Figura 5.5. Cableado de ventanas y seguros; Cableado de luces bajas, medias, altas y direccionales.....	92
Figura 5.6. Ejemplo utilización lámpara de prueba para identificar cables Luces direccionales .....	92
Figura 5.7. Cableado de los accesorios .....	93
Figura 5.8. Conexiones Accesorios - Placa PCB .....	94
Figura 5.9. Lista de accesorios controlados por voz en el vehículo .....	95
Figura 5.10. Medidor de batería – 100% disponible .....	97
Figura 5.11. Medidor de batería – 76% disponible .....	99
Figura 5.12. Medidor de batería – 11% disponible .....	99

## **ÍNDICE DE TABLAS**

Tabla 2.1. Descripción de señales del conector DB-9.....	32
Tabla 5.1. Precisión de Reconocimiento de voz Vs. Volumen del reproductor de sonido .....	97
Tabla 5.2. Autonomía de la batería del computador.....	98

## **SÍNTESIS**

Este proyecto tiene como finalidad brindar un soporte al usuario en la conducción del vehículo, controlando los accesorios mediante comandos de voz, los cuales serán procesados por una computadora a la cual se debe conectar un micrófono logrando activar el reconocimiento de voz de Windows o de Dragon Naturally Speaking 10.

Para lograr la comunicación entre computador y vehículo, fue necesario realizar el siguiente proceso: Desarrollo de ejecutables (software) en Visual Basic 6.0, el cual se encarga del envío de datos por el puerto serial; En Visual Basic 2010 se diseña una interfaz gráfica de gran calidad visual con botones que nombran a los accesorios intervenidos y son la ruta de ejecución de órdenes. Estos ejecutables trabajan en conjunto para lograr formar un solo software, que es una aplicación de fácil uso para el usuario, además son reconocidos por los programas de reconocimiento de voz inmediatamente, logrando cumplir la función de éste proyecto.

Una vez realizada la fase de diseño del software, el desarrollo del circuito en planos se lo realiza en el programa Proteus, que se subdivide, en Isis el cual permite crear el plano eléctrico del circuito, y Ares que permite diseñar la placa PCB como un circuito eficiente. El circuito es una interface de control que se comunica con el computador mediante un cable USB a serial, el cual sirve de enlace de transmisión de datos, logrando que el computador envíe las ordenes del usuario a la interface de control y que el circuito procese la información, controlando la activación o desactivación de los accesorios del vehículo.

## **SUMMARY**

This project aims to provide driver support with the control of accessories of the vehicle using voice commands, which will be processed by a computer with a microphone to activate voice recognition of Windows or Dragon Naturally Speaking 10.

To achieve the communication between computer and car, will be necessary to realize the following steps: Development of executables (software) in Visual Basic 6.0, which is responsible to send data through the serial port. In Visual Basic 2010 is designed a graphical interface of high visual quality with buttons that named the accessories and are the route of execution of orders. These executables work together to achieve a single software, which is an application easy to utilize for the user, and it are recognized by voice recognition software immediately, achieving to comply the function of this project.

When this phase of software design is completed, the circuit development was conducted in the Proteus program, which is subdivided in Isis, that allow to create the electrical circuit, and Ares that allow to design the plate of PCB with an efficient circuit. The circuit is a control interface that communicates with the computer using a cable USB to serial, which serves as a data transmission link, the computer will send the orders of the user to the control interface, and finally it processes the information with the control of the activation or deactivation of vehicle accessories.

# **CAPÍTULO 1**

## **INTRODUCCIÓN**

### **1.1. ANTECEDENTES**

El habla es una de las partes más importantes de la expresión del ser humano, es algo que nos diferencia del resto de seres vivos en planeta, ya que sin el habla el pensamiento mismo del hombre no sería posible. No se trata simplemente de un sistema para transmitir información, aunque sea claro una de sus funciones. Pero es por medio de los sonidos que se presenta la esencia del hombre.

El reconocimiento de voz o reconocimiento del habla es el proceso de conversión de un mensaje hablado en texto, que permite al usuario una comunicación con el computador o algunos otros dispositivos diseñados para el efecto.

En esta clase de comunicación, la identidad del interlocutor va a estar directamente relacionada con las características fisiológicas y de comportamiento del mismo (hábitos lingüísticos, entonación de frases, entre otras).

Con la tecnología de hoy en día, existen una gran variedad de situaciones en las cuales el volumen de utilización de sistemas de reconocimiento de voz están creciendo rápidamente, algunos ejemplos de ello son: servicios de telecomunicaciones, sistemas de control, la redacción de textos sin teclado, ayuda a discapacitados físicos, etc.

Este tipo de sistemas en el área automotriz han sido escasamente utilizados, es por esto que se vuelve interesante su implementación en el vehículo.

## **1.2.IMPORTANCIA**

El reconocimiento de voz, creado por Microsoft y Nuance permite controlar varias aplicaciones en una computadora con sistema operativo Windows.

Razón por la cual el desarrollo de nuevos sistemas automáticos de control, despierta el interés de crear futuras aplicaciones para el desarrollo tecnológico del país.

En éste caso, para generar el control de accesorios mediante comandos de voz en un automóvil, se desarrolla una interfaz gráfica (software) y una interface de control (hardware), que permite la comunicación entre computadora y automotor.

Por lo que este proyecto tiene como propósito principal, el beneficio de la sociedad y del campo automotriz, brindando comodidad y confort al conductor mientras maneja el vehículo con un software fácil de usar y de respuesta rápida, logrando disminuir el porcentaje de riesgo de sufrir un accidente de tránsito por distracción al manipular accesorios.

Por otra parte el sistema brinda independencia a pasajeros con discapacidad motriz en las extremidades de su cuerpo, siendo para estos últimos, un aporte tecnológico vital para el desarrollo en su diario vivir.

Dada la importancia del habla, el presente trabajo de graduación pretende crear una interacción entre una de las expresiones esenciales del hombre con la computadora, creando así un Sistema de Reconocimiento de Voz.

### **1.3.OBJETIVOS**

El objetivo principal del proyecto consiste en el diseño e implementación de un sistema de reconocimiento por voz para el accionamiento de accesorios en el automóvil, a través de comandos de voz, utilizando una PC con programas disponibles en el mercado, y otros desarrollados para este fin, además de la construcción de una interface de control PC-PIC. A continuación se exponen los objetivos específicos marcados para el presente proyecto:

#### **1.3.1. Objetivos Específicos**

- Diseñar y construir la interface de control PC-PIC.
- Utilizar Visual Basic 6.0 para el desarrollo de la programación que permita la comunicación PC-PIC.
- Programar y grabar el Microcontrolador (PIC) utilizando para ello MicroCode Studio y PICKIT2 respectivamente.
- Realizar el entrenamiento y reconocimiento de sonidos mediante el software Dragon Naturally Speaking 10 y Reconocimiento de voz perteneciente a Windows
- Elaborar un software-ejecutable (programa), utilizando Visual Basic 2010, para el control de los accesorios del vehículo.

## **1.4. JUSTIFICACIÓN**

El reconocimiento por voz, es un proceso complejo del cual se extrae información de una señal de voz.

Actualmente el uso de fonemas o palabras para el accionamiento de dispositivos viene mostrando varias alternativas y utilidades, para el comando de un proceso eléctrico o mecánico, sin embargo las dificultades, por lo regular se encuentran en las representaciones acústicas de la voz, para que por medio de estas se puedan controlar cargas eléctricas o electrónicas.

Así se plantea una alternativa y solución para el control de dispositivos por medio de órdenes por voz, brindando un esquema para el desarrollo de sistemas automáticos de control.

Este sistema permite una comunicación óptima e interactiva entre hombre y máquina, brindando facilidad de uso de los accesorios del vehículo, proporcionando seguridad y confort, siendo así su aplicación de gran utilidad para el sector automotriz, cuyos beneficios serían infinitos.

Esta aplicación en el automóvil beneficiará a la sociedad; tanto a personas discapacitadas, como a la disminución en un porcentaje alto, del riesgo de tener un accidente de tránsito a causa de la impericia del conductor.

Varias fuentes estadísticas señalan que gran parte de los accidentes de tránsito en las vías, se deben a la distracción del conductor de su objetivo principal de conducción mientras manipula accesorios del vehículo, por lo que es necesario el desarrollo e implementación de un sistema de control por comandos de voz,

para evitar dicha situación y brindar facilidades de ejecución de acciones en un tiempo mínimo.

El sistema podrá ser instalado en un futuro en vehículos del parque automotor ecuatoriano.

La realización y culminación de este proyecto es de gran factibilidad debido a que en nuestro país existe el software y los elementos electrónicos necesarios para este fin, acompañado del intelecto humano que con dedicación y empeño se pueden desarrollar este tipo de sistemas.

## **CAPÍTULO 2**

### **MARCO TEÓRICO**

#### **2.1.LA VOZ HUMANA Y EL RECONOCIMIENTO DE VOZ**

##### **2.1.1. Señal de voz**

La señal de voz, es una onda sonora o un conjunto de frecuencias, que estimulan el oído humano y al final son procesadas a través del cerebro.

Este tipo de señal acústica puede ser convertida por medio de un micrófono como una señal eléctrica y analizarla completamente mediante la utilización de un plano cartesiano.

La voz se produce a través del funcionamiento de cuatro elementos que generan el habla y se clasifican de la siguiente manera:

“En primer lugar, los pulmones suministran la columna de aire que, atravesando los bronquios y la tráquea, van a sonorizar las cuerdas vocales que se encuentran en la laringe.

Es en la laringe donde propiamente se produce la voz en su tono fundamental y sus armónicos; luego sufre una modificación en la caja de resonancia de la nariz, la boca y la garganta (naso-buco-faríngea), en la que se amplifica y se forma el timbre de voz.

Los órganos articuladores (labios, dientes, paladar duro, velo del paladar, mandíbula) van finalmente a moldear esa columna sonora transformándola en sonidos y articulaciones del habla; es decir, en fonemas, sílabas y palabras.”<sup>1</sup>

### 2.1.1.1. Aparato fonador

Son varios órganos que participan en la generación de sonidos o efectos sonoros, cuyo fin, es fomentar la comunicación entre personas a través del habla.

“Las emisiones sonoras del habla son producidas por el aparato fonador (Figura 2.1), compuesto por el sistema respiratorio, la laringe, las cuerdas vocales y la cavidad bucal”<sup>2</sup>

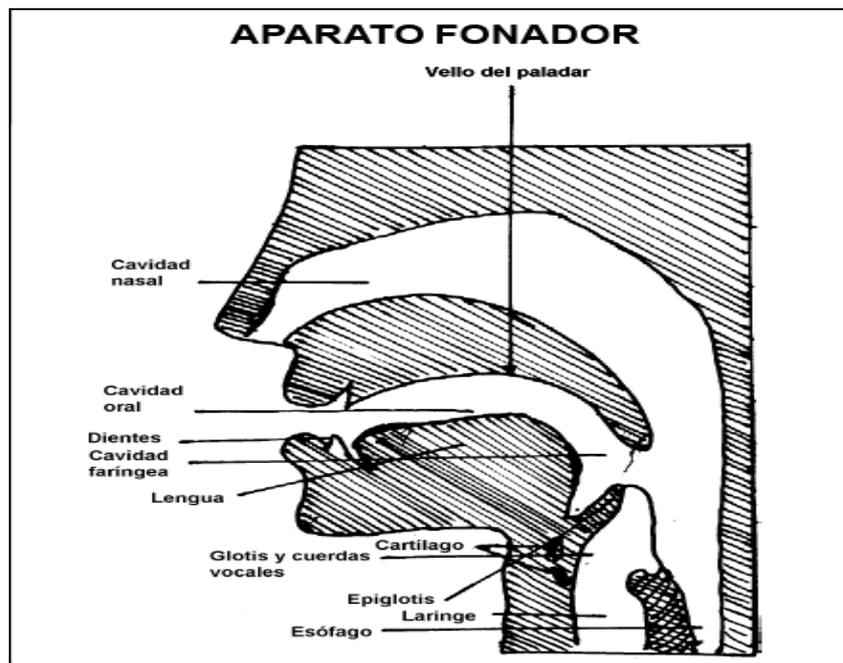


Figura 2.1. Partes del Aparato Fonador

Fuente: CASTAÑEDA, Pablo Félix. El lenguaje verbal del niño, Aparato fonador, pág.123

<sup>1</sup> CASTAÑEDA, Pablo Félix. El lenguaje verbal del niño, Explorando la producción sonora del habla. Págs.126-127

<sup>2</sup> CASTAÑEDA, Pablo Félix. El lenguaje verbal del niño, Aparato fonador, pág.123

### **2.1.1.2. Formantes**

Son producidas por la vibración de las cuerdas bucales y se las considera como una amplitud de onda, la cual puede tener frecuencia determinada. La frecuencia puede ser variada tanto como en tiempo y en intensidad, se la puede definir como una cantidad de energía sonora que tiene un sonido.

Las formantes permiten segmentar las características o patrones del habla humana, que son distintos e individuales por cada persona, por lo que han permitido crear avanzados sistemas electrónicos de control como el reconocimiento de voz.

### **2.1.2. Cualidades del Sonido**

Las cualidades del sonido son:

- “La altura es lo que produce la sensación de más agudo o más grave, resultando ésta del número de vibraciones por segundo. Cuanto mayor es el número de vibraciones por segundo, tanto más agudo es el sonido.
- En cambio, la intensidad depende no de la cantidad sino de la amplitud de las vibraciones por segundo. Cuanto más ancha es la vibración, más intenso será el sonido.
- Y el timbre depende de los armónicos, que son múltiplos del tono fundamental. La vibración provoca una onda sonora o tono fundamental y unos armónicos que filtrados en la cavidad bucal y en la nasal producen el timbre del sonido.

Estas características o cualidades de la voz humana, una vez que sale de los resonadores, son modificadas y moldeadas por los articuladores, transformándose en sonidos del habla: fonemas, sílabas, palabras.”<sup>3</sup>

### **2.1.3. Clasificación de los Sonidos**

Se los conoce como elementos fonéticos del lenguaje verbal, las palabras o fonemas pronunciados por una persona, se las conoce como unidades sonoras y tienen la siguiente clasificación:

#### **2.1.3.1. Vocales y Consonantes**

Son elementos del habla, que se clasifican según su pronunciación a través del aparato fonador.

Las vocales, se forman a través de la vibración de las cuerdas vocales con mínimas dificultades para pronunciar una palabra y las consonantes se forman con cierta dificultad mediante la constricción del tracto vocal para la salida del aire o palabras.

#### **2.1.3.2. Oralidad y nasalidad**

Las vocales orales (a, e, o) se producen mediante la abertura o semiabertura, entre la lengua y el paladar mientras las vocales nasales (i, u), se generan en la abertura entre la lengua y el estrechamiento con el paladar.

---

<sup>3</sup> CASTAÑEDA, Pablo Félix. El lenguaje verbal del niño, Calidad de la voz y sus diferencias entre las personas, págs.142, 143.

### **2.1.3.3. Tonalidad**

Es una variación en aumento o disminución en el tono de voz, al decir un fonema o palabra según la intensidad o sentido con el que se exprese una persona.

### **2.1.4. ¿Qué es el reconocimiento de voz?**

El reconocimiento de voz y el campo de la electrónica han evolucionado, su aplicación en la informática, especialmente en computadoras, permiten el control de varias aplicaciones mediante comandos de voz.

Se lo puede definir como un nuevo tipo de comunicación que se realiza a través elementos electrónicos como dispositivos móviles, computadores y máquinas. La aplicación de un sistema de reconocimiento por voz, permite mejorar y optimizar la comunicación entre hombre-máquina.

Este sistema permite al computador comparar la información obtenida de las características acústicas de la voz, con parámetros preestablecidos y grabados con anterioridad por el operador transformándolos de formato análogo a digital. La computadora interpreta las palabras o fonemas dictados, procesando de inmediato una orden y ejecutándola en una acción.

#### **2.1.4.1. Software disponible para reconocimiento de voz**

En el mercado existen disponibles programas para el reconocimiento de voz, estos son capaces de identificar palabras, llamadas también comandos, estas palabras son dictadas a través de un micrófono, luego de ello el programa compara los comandos con una base de datos que el usuario ha añadido con

anterioridad al programa que se esta utilizando; estos comandos requieren de entrenamiento de palabras que se guardan para un posterior uso. Los programas mas utilizados son los siguientes:

- Tyle.

Tyle es un producto de la empresa Tiny Thoughts LTD, la cual fue fundada en 2006. Dicha compañía ofrece aplicaciones de voz y video, además de otras aplicaciones fáciles de usar, integradas con el reconocimiento de comandos de voz en la computadora, las cuales están disponibles para la descarga en el sitio web: [www.tyle.com](http://www.tyle.com).

Tyle fue construido para entender cualquier idioma, por lo tanto, cualquier persona lo puede utilizar y además es gratis para la mayoría de sus propósitos, en nuestro mercado solo existe la versión gratuita del programa, el cual posee limitación de comandos, cabe recalcar no requiere de un entrenamiento previo de voz.

- Reconocimiento de voz perteneciente a Windows.

Es un software de reconocimiento de voz creado por Windows, esta disponible en versiones de Windows vista y siete. Viene incorporado en el sistema y necesita ser configurado para operar satisfactoriamente.

El programa permite al operador la ejecución de acciones como dictado y selección de ciertos botones de varios programas mediante comandos de voz que son reconocidos en un tiempo mínimo.

El tutorial de entrenamiento permite al programa asimilar la señal de voz proveniente del usuario para ejecutar el comando dictado por el mismo. El programa se mantiene activo en el escritorio para reconocer cualquier comando de voz, generando una respuesta rápida a la petición del usuario.

- Dragon Naturally Speaking 10.

Este software fue creado por la empresa Nuance Communications, el cual es el programa de reconocimiento de voz más vendido en el mundo.

Dragon Naturally Speaking 10, es casi dos veces más rápido que la versión anterior, ya que convierte la voz en texto automáticamente con una precisión de hasta el 99% y a una velocidad de hasta 160 palabras por minuto.

Con Dragon se pueden escribir documentos y correos electrónicos, navegar por Internet, e incluso controlar el ordenador con la voz, consiguiendo así un ahorro de tiempo y un aumento de la productividad, por sus múltiples funciones, velocidad y precisión este es el software utilizado en el presente proyecto.

Además en el mercado ecuatoriano existe la versión preferred del mismo, es decir sin restricciones ni límite de uso.

#### **2.1.5. Procesamiento digital de señales**

Una señal, es una magnitud eléctrica que se origina en relación al tiempo; representando una información, misma que permite la comunicación entre máquinas o elementos electrónicos que conforman un sistema.

El procesamiento digital de las señales, se genera cuando la fuente o emisor proporciona información, que será transformada a través de un transductor, el cual obtiene la información de la señal convirtiéndola en una onda de tensión; que es receptada por el sistema o microprocesador y a la salida del mismo según su aplicación se ejecutará una acción a través de otro transductor, hasta que la información llegue al usuario.

Este rol lo cumple en este proyecto un micrófono, que es un transductor electroacústico, el cual procesa las señales de voz, convirtiendo la energía sonora en energía eléctrica.

La información o señal acústica de la voz, es procesada por el sistema, microprocesador o computador, los cuales están específicamente diseñados para el procesamiento digital de señales, ya sea analógico a digital o viceversa para realizar diferentes funciones. Dicho sistema está ubicado en la tarjeta de sonido de la computadora, y se encarga de realizar la tarea anteriormente mencionada.

En el mercado existen software de reconocimiento de voz, que permiten realizar tareas tales como dictado, control de ventanas y programas, etc., a través de comandos de voz.

#### **2.1.6. El procesamiento de la información y la electrónica de potencia**

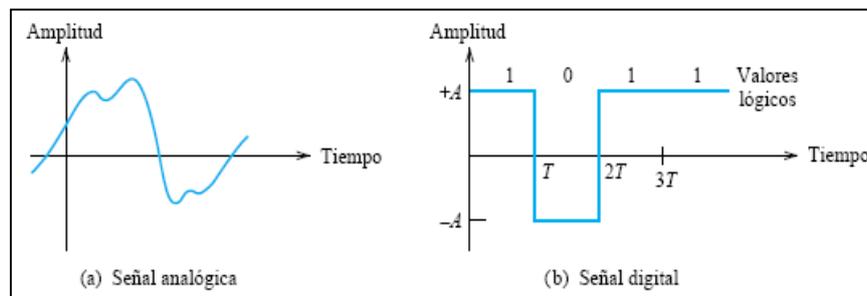
“Los sistemas electrónicos pertenecen a la siguiente clasificación:

Procesamiento digital de la señal, comunicación, electromedicina, instrumentación, control e informática. Estos sistemas se caracterizan por extraer, recopilar y procesar las señales de la información obtenida (Figura 2.2).

En estos sistemas se suministra energía a los dispositivos de salida, para tener un resultado, como escuchar música a través de un parlante, etc.

Además es importante crear eficiencia en los circuitos porque permiten asegurar la vida útil de la batería.

Existen varios sistemas que se concentran en la potencia de las señales en vez de la información obtenida.”<sup>4</sup>



**Figura 2.2. Las señales analógicas toman valores continuos de amplitudes. Las señales digitales toman unas pocas amplitudes discretas.**

Fuente: HAMBLEY, Alan. Electrical Engineering: Principles and Applications, pág. 3.

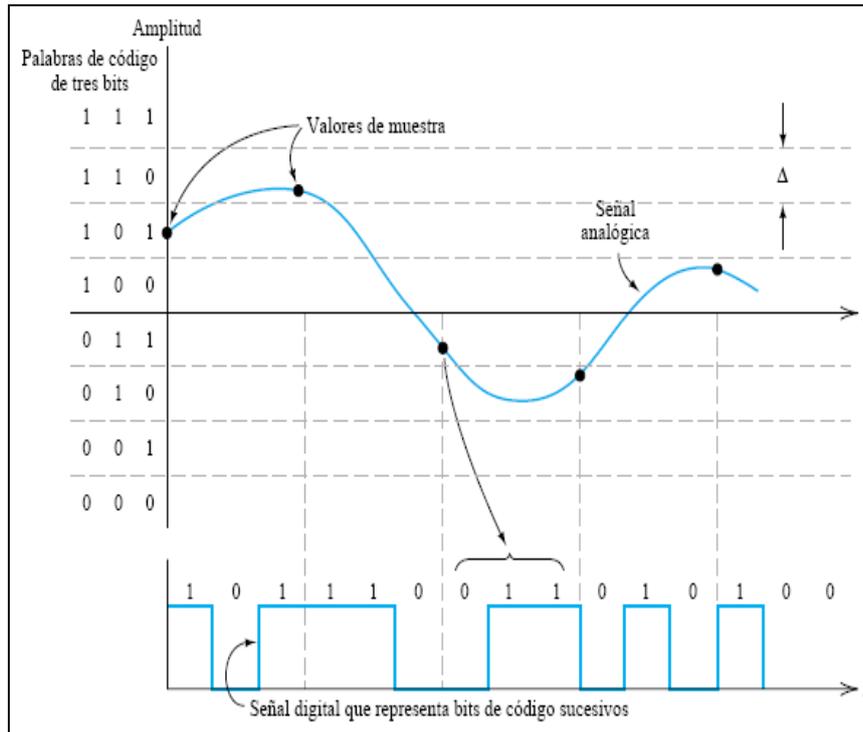
### 2.1.6.1. Conversión de señales de formato analógico a digital

“Las señales análogas se pueden transformar a formato digital, para hacerlo, se mide o se realiza un muestreo de la señal análoga en intervalos de tiempo periódicos (Figura 2.3). Cada valor está representado por un código de 3 bits que corresponden a la zona de amplitud obtenidas de la muestra, por consecuencia se obtiene varios códigos que representan una onda digital”<sup>5</sup>

---

<sup>4</sup> HAMBLEY, Alan. Electrónica, El procesamiento de la información y la electrónica de potencia, págs. 3 y 4

<sup>5</sup> HAMBLEY, Alan. Electrónica, Conversión de señales de formato analógico a digital, pág. 4



**Figura 2.3. Conversión de una señal analógica en un equivalente digital aproximado mediante muestreo. Cada valor de muestra viene representado por un código de 3 bits. Los convertidores reales utilizan palabras de código más largas.**

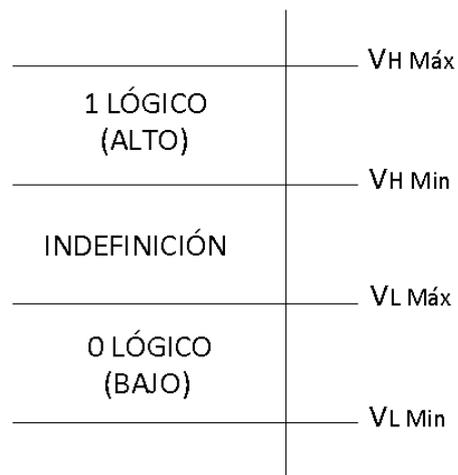
Fuente: HAMBLEY, Alan. Electrónica, pág. 5.

## 2.2. ELECTRÓNICA DIGITAL

“La electrónica digital utiliza variables discretas que solo pueden tomar dos valores (cero y uno, alto y bajo, verdadero y falso, etc.) denominados valores lógicos. Un sistema en el que las variables sólo pueden tomar dos valores se denomina sistema binario, y la electrónica digital se basa en el empleo de un sistema binario. Un dígito binario se denomina bit, y es la menor unidad de información disponible en un sistema digital. Estos dígitos pueden tomar dos valores, que denominaremos valores lógicos: cero y el uno lógicos.”<sup>6</sup>

<sup>6</sup> BLANCO VIEJO, Cecilio. Electrónica Digital, Introducción, págs. 1.

En los sistemas digitales se pueden emplear dos tipos de lógica: positiva y negativa. Se habla de lógica positiva cuando se considera que el cero lógico es el nivel de reposo y el uno lógico el nivel activo. Por el contrario, se habla de lógica negativa cuando se considera el nivel activo es el cero lógico y el de reposo el uno lógico.



**Figura 2.4. Niveles lógicos<sup>7</sup>**

“El cero y el uno lógicos pueden estar representados de distintas formas: dos valores de tensión, de corriente o un interruptor que abre y cierra.

Así se puede asignar el cero lógico una tensión de 0 voltios y al uno lógico una tensión de 5 voltios, o bien una corriente de 0mA al cero lógico y una corriente de 10mA al uno lógico, es decir, los valores de la variable lógica binaria están representados por unos valores determinados de una magnitud física.”<sup>8</sup>

<sup>7</sup> BLANCO VIEJO, Cecilio. Electrónica Digital, Introducción, pág. 2

<sup>8</sup> BLANCO VIEJO, Cecilio. Electrónica Digital, Introducción, pág. 2

## 2.2.1. Puertas lógicas

Existen muchas formas de implementar una función lógica, razón por la cual se utilizan las compuertas lógicas; éstas cumplen funciones simples, pero en conjunto conforman sistemas más complejos, por ejemplo, ciertos elementos electrónicos; en especial circuitos integrados.



Figura 2.5. Funcionamiento de compuerta lógica<sup>9</sup>

### 2.2.1.1. Puerta NOT (NO)

También conocida como puerta inversora. Es aquella en la que la función o salida tiene el valor inverso de la variable o entrada.

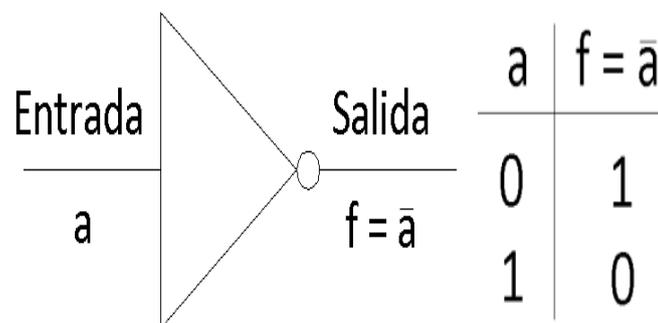


Figura 2.6. Compuerta NOT<sup>10</sup>

---

<sup>9</sup> LUQUE SACALUGA, David. Electrónica Digital y Microprogramable, Tipología, Función y Características de las puertas lógicas, pág.32.

<sup>10</sup> LUQUE SACALUGA, David. Electrónica Digital y Microprogramable, Tipología, Función y Características de las puertas lógicas, pág.32.

### 2.2.1.2. Puerta OR (O)

Es aquella que describe la suma lógica.

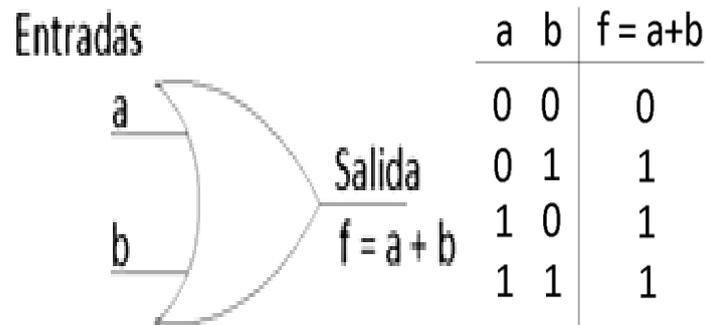


Figura 2.7. Compuerta OR<sup>11</sup>

### 2.2.1.3. Puerta AND (Y)

Es aquella que describe el producto lógico.

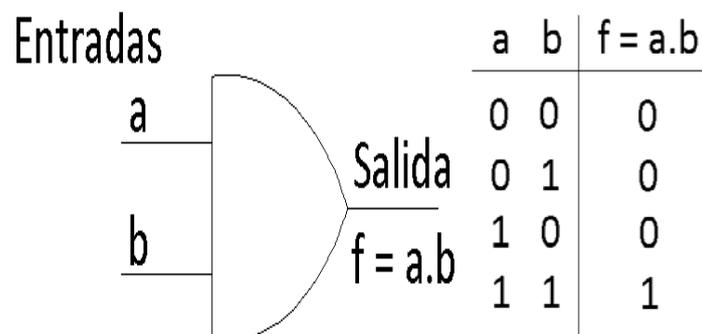


Figura 2.8. Compuerta AND<sup>12</sup>

---

<sup>11</sup> LUQUE SACALUGA, David. Electrónica Digital y Microprogramable, Tipología, Función y Características de las puertas lógicas, pág.33.

<sup>12</sup> LUQUE SACALUGA, David. Electrónica Digital y Microprogramable, Tipología, Función y Características de las puertas lógicas, pág.33.

#### 2.2.1.4. Puerta NOR (NO-O)

Es aquella en la que la salida es el complemento o inverso de la suma de las entradas.

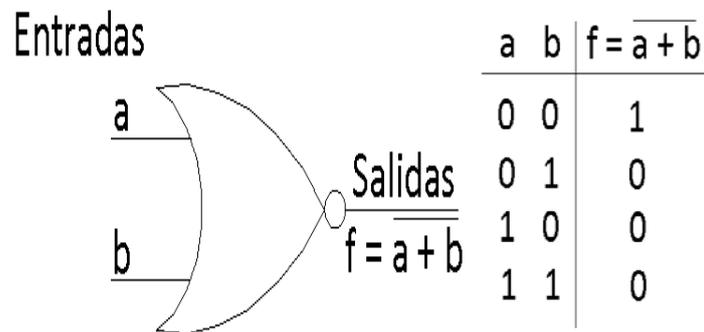


Figura 2.9. Compuerta NOR<sup>13</sup>

#### 2.2.1.5. Puerta NAND (NO-Y)

Es aquella en la que la salida es el complemento o inverso del producto de las entradas.

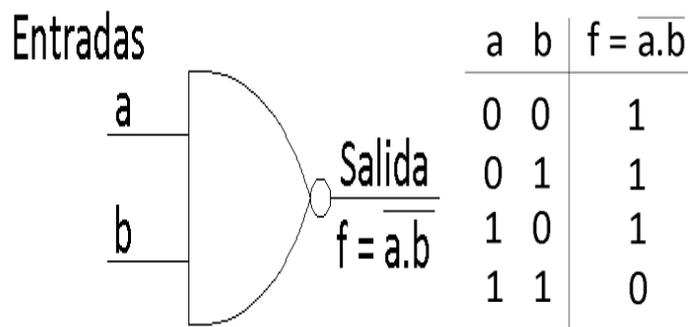


Figura 2.10. Compuerta NAND<sup>14</sup>

---

<sup>13</sup> LUQUE SACALUGA, David. Electrónica Digital y Microprogramable, Tipología, Función y Características de las puertas lógicas, pág.34.

<sup>14</sup> LUQUE SACALUGA, David. Electrónica Digital y Microprogramable, Tipología, Función y Características de las puertas lógicas, pág.34.

### 2.2.1.6. Puerta OREX (O-Exclusiva)

Es una de las puertas más importantes. Es aquella cuya salida toma el valor 1 cuando una de las variables de entrada toma el valor 1 y la otra toma, el valor 0.

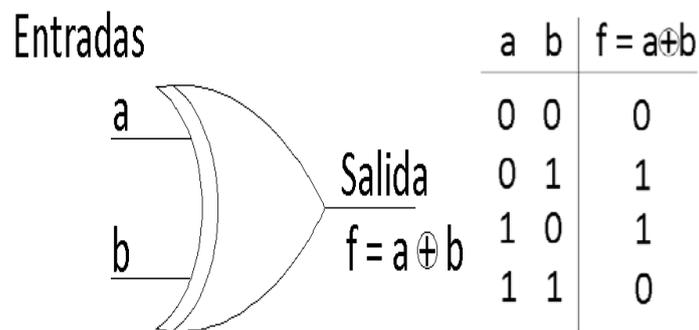


Figura 2.11. Compuerta OREX<sup>15</sup>

### 2.2.1.7. Puerta NOR-EX

Es simplemente la inversión de la puerta OREX.

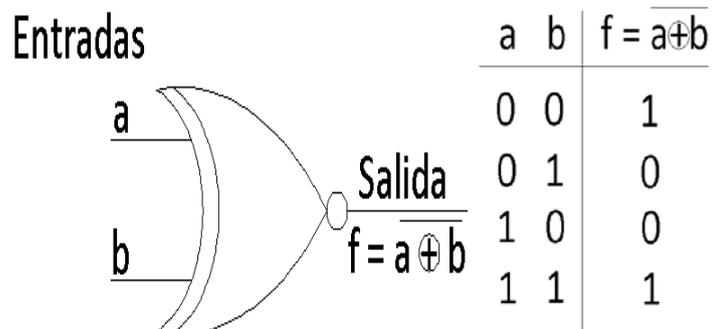


Figura 2.12. Compuerta NOR-EX<sup>16</sup>

---

<sup>15</sup> LUQUE SACALUGA, David. Electrónica Digital y Microprogramable, Tipología, Función y Características de las puertas lógicas, pág.34.

<sup>16</sup> LUQUE SACALUGA, David. Electrónica Digital y Microprogramable, Tipología, Función y Características de las puertas lógicas, pág.34.

### **2.2.2. Sistemas Digitales**

Un sistema digital puede definirse como un conjunto de componentes electrónicos interconectados que procesan información digital. La mayoría de componentes de los sistemas digitales son circuitos integrados. Los enlaces entre ellos son conductores eléctricos que transmiten la información digital.

Los sistemas digitales cumplen funciones complejas, desde unos pocos componentes hasta millones. La estructura de un sistema queda definida como el conjunto de sus componentes y conjunto de enlaces.

“En un sistema digital los componentes transmiten entre sí e intercambian con el exterior señales eléctricas, que varían en el tiempo y contienen información convenientemente codificada. El sistema completo (o cada componente interno) se puede contemplar como un dispositivo de procesamiento de información, que recibe un conjunto de señales de entrada procedentes de otros sistemas (o componentes), utilizadas para generar un conjunto de señales de salida que puede transmitir a otros sistemas (o componentes).”<sup>17</sup>

El comportamiento del sistema queda definido por las relaciones entre entradas y salidas. En un sistema es necesario realizar un análisis sobre la función que cumplirá incluyendo a su estructura y componentes, con lo que se debe verificar que éste sistema cumpla con el comportamiento y prestaciones deseadas además de tener bajo coste.

---

<sup>17</sup> LLORIS RUIZ, Antonio, PRIETO ESPINOSA, Alberto, PARRILLA ROURE, Luis. Sistemas Digitales, Representación de Sistemas Digitales, pág. 501, 502.

### 2.2.2.1. Desarrollo de sistemas digitales

“La metodología seguida en la ingeniería, particularizada para los sistemas digitales, se resumen en la figura 2.13. Se comienza con la concepción de un sistema comercializable y a satisfacción de quien lo idea. Posteriormente viene la fase de diseño, que pretende resolver el problema planteado. Con el diseño, a partir de la especificación del problema, se obtiene la descripción detallada necesaria para su implementación física”<sup>18</sup>.

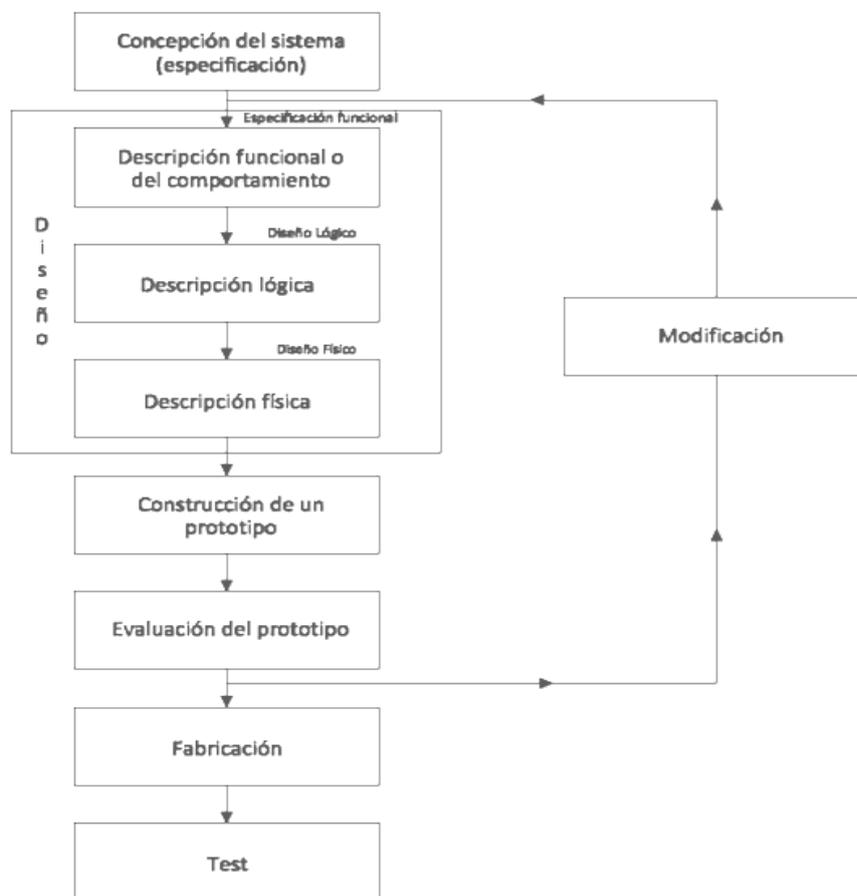


Figura 2.13. Pasos usuales para el desarrollo de sistemas digitales<sup>19</sup>

<sup>18</sup> LLORIS RUIZ, Antonio, PRIETO ESPINOSA, Alberto, PARRILLA ROURE, Luis. Sistemas Digitales, Metodología para el desarrollo de Sistemas Digitales, pág. 506.

<sup>19</sup> LLORIS RUIZ, Antonio, PRIETO ESPINOSA, Alberto, PARRILLA ROURE, Luis. Sistemas Digitales, Metodología para el desarrollo de Sistemas Digitales, pág. 507.

Normalmente se utiliza un diseño descendente, según los niveles de complejidad, con una serie de representaciones sucesivas del sistema como un conjunto de bloques funcionales y sus correspondientes conexiones.

El objetivo de la etapa de diseño suele ser la construcción de un prototipo, que es evaluado con la determinación experimental de prestaciones.

“Las medidas efectuadas se comparan con las especificaciones y, en función de estas comprobaciones, se decide si se debe o no modificar el diseño (a partir de alguna de sus etapas).

Una vez validado el diseño es usual construir prototipos de pre-producción, y posteriormente se pasa a la producción del sistema.

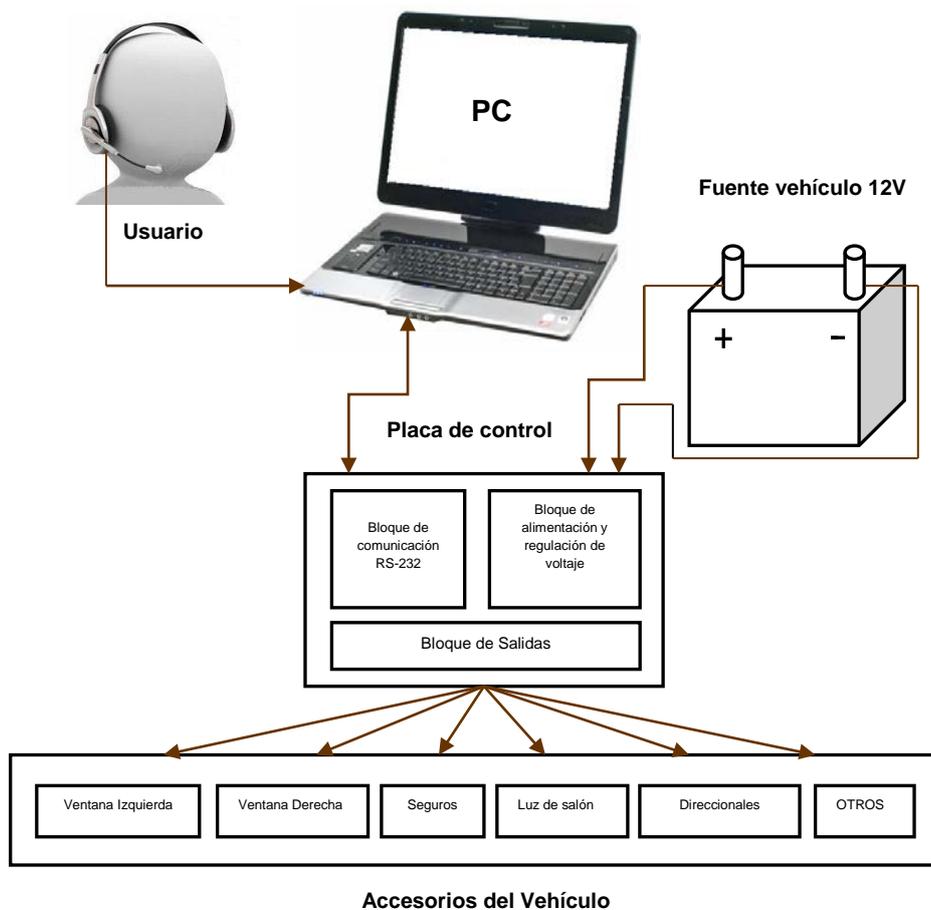
La etapa final del proceso es el test del sistema, con objeto de identificar cualquier posible defecto y asegurar la calidad del producto final.”<sup>20</sup>

---

<sup>20</sup> LLORIS RUIZ, Antonio, PRIETO ESPINOSA, Alberto, PARRILLA ROURE, Luis. Sistemas Digitales, Metodología para el desarrollo de Sistemas Digitales, pág. 508.

### 2.3. REPRESENTACIÓN BÁSICA DEL SISTEMA

El sistema está formado por una placa de control ó circuito PCB, la misma que está compuesta por el bloque de comunicación RS-232, el bloque de alimentación y regulación de voltaje, y finalmente un bloque de salidas; las cuales permiten controlar los múltiples accesorios en el vehículo, a través de una PC que adicionalmente utiliza un micrófono para que el programa de reconocimiento de voz interprete los comandos de voz y se realicen las acciones de control, a través de la interfaz gráfica creada para este propósito. A continuación se muestra lo descrito previamente en el siguiente esquema. (Figura 2.14).



**Figura 2.14. Representación esquemática Del sistema.**

Autor: Andrés E. & Andrés Y.

## **2.4. DISEÑO DEL BLOQUE DE COMUNICACIÓN CON EL ORDENADOR**

Algunos circuitos microprogramados, así como los microcontroladores se encargan de almacenar datos o comandos para realizar determinadas tareas y funciones previamente almacenadas en ellos, ya sea para llevar a cabo las mismas de manera autónoma, o si se requiere de la capacidad de intercambiar información con ordenadores, para ejecutar diversas actividades.

### **2.4.1. La Norma RS-232**

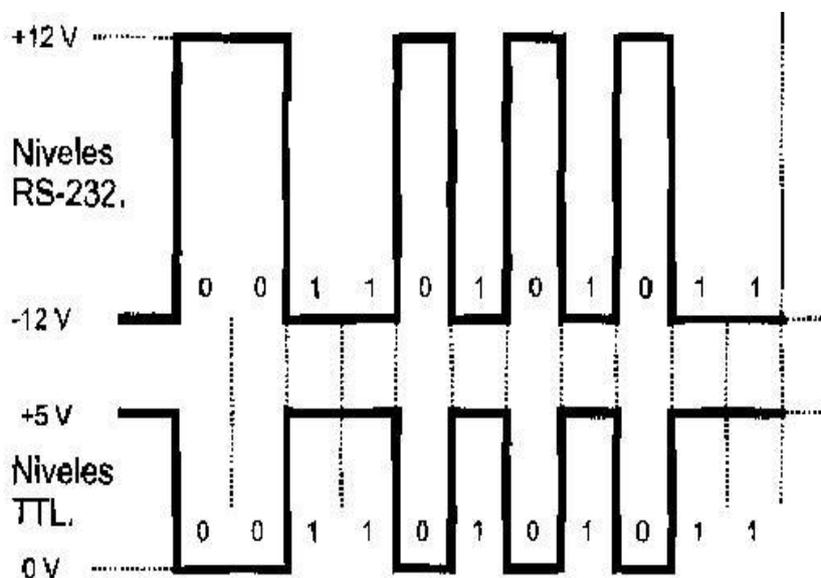
Ante la gran variedad de equipos, sistemas y protocolos que existen surgió la necesidad de un acuerdo que permitiera a los equipos de varios fabricantes comunicarse entre si. La EIA (Electronics Industry Association) elaboró la norma RS-232, la cual define la interface mecánica, los pines, las señales y los protocolos que debe cumplir la comunicación serial.

Todas las normas RS-232 cumplen con los siguientes niveles de voltaje:

- Un "1" lógico es un voltaje comprendido entre  $-5\text{v}$  y  $-15\text{v}$  en el transmisor y entre  $-3\text{v}$  y  $-25\text{v}$  en el receptor.
- Un "0" lógico es un voltaje comprendido entre  $+5\text{v}$  y  $+15\text{v}$  en el trasmisor y entre  $+3\text{v}$  y  $+25\text{v}$  en el receptor.

El envío de niveles lógicos (bits) a través de cables o líneas de transmisión necesita la conversión a voltajes apropiados. En los microcontroladores para representar un "0" lógico se trabaja con voltajes inferiores a  $0.8\text{V}$  y para un "1" lógico con voltajes mayores a  $2\text{V}$ . En general cuando se trabaja con familias

TTL y CMOS se asume que un “0” lógico es igual a cero Volts y un “1” lógico es igual a 5 Volts (Figura 2.15).



**Figura 2.15. Niveles de voltaje RS-232 y TTL.**

Fuente: PALACIOS, Enrique. Microcontrolador PIC 16F84. Pág. 308.

La importancia de conocer esta norma, radica en los niveles de voltaje que maneja el puerto serial del ordenador, ya que son diferentes a los que utilizan los microcontroladores y los demás circuitos integrados. Por lo tanto se necesita de una interface que haga posible la conversión de niveles de voltaje a los estándares manejados por los circuitos integrados TTL.

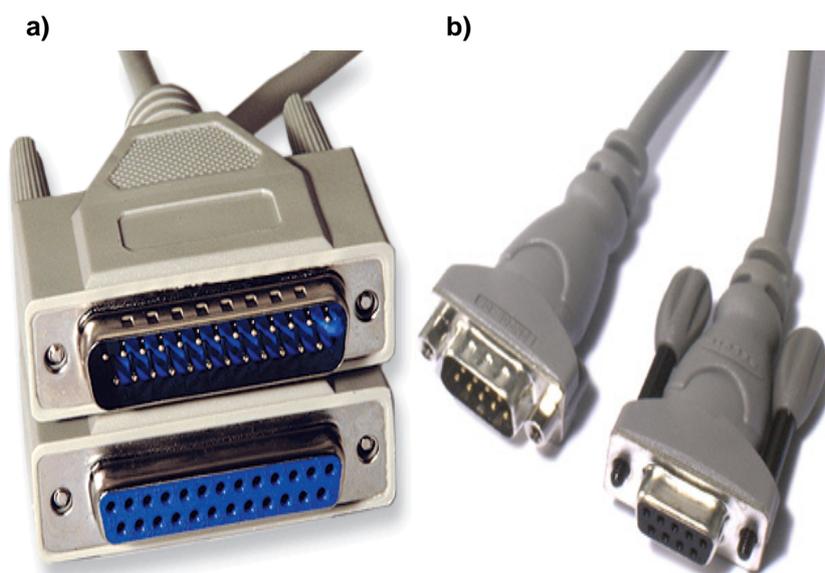
#### **2.4.2. Puerto Serie RS-232**

La forma más común y sencilla de comunicar cualquier dispositivo con un ordenador es a través de un puerto serie; Su uso es variado, permitiendo crear módulos de adquisición de datos, instrumentos de prueba y circuitos de control.

En un ordenador puede haber varios puertos series, normalmente denominados COM 1, COM 2, etc., lo cual puede variar entre un ordenador u otro.

Los puertos serie son accesibles mediante conectores. La norma RS-232 establece dos tipos de conectores llamados DB-25 (de 25 pines) y DB-9 (de 9 pines), estos pueden ser machos y hembras (Figura 2.16). En sus inicios se estableció para conectar un ordenador con un periférico, por lo que aparecen muchas patillas en los conectores DB-25 que en otro tipo de aplicaciones no se utilizan y en las que es más común utilizar el conector tipo DB-9, el cual utilizaremos para desarrollar nuestro proyecto.

Cada una de las patillas del conector RS-232 tiene una función especificada por la norma. Hay unos terminales por los que se transmiten y reciben datos y otros que controlan el establecimiento, flujo y cierre de la comunicación. La figura 2.17 describe cada una de las patillas del conector DB-9.

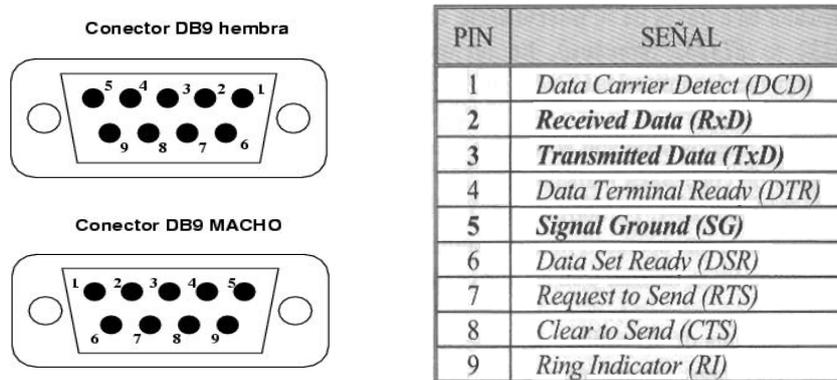


**Figura 2.16. Conectores RS232: a) DB-25 macho-hembra. b) DB-9 macho-hembra.**

Para comunicarse con un Microcontrolador es suficiente con tres líneas:

- Línea de transmisión (TxD), pin 3 (*Transmitted Data*).
- Línea de recepción (RxD), pin 2 (*Received Data*).
- Pin de masa (SG), pin 5 (*Signal Ground*).

Las especificaciones del puerto serie están contenidas en la norma RS-232 (ó *EIA-232 Standard*).



**Figura 2.17. Patillaje del conector DB-9**

Fuente: PALACIOS, Enrique. Microcontrolador PIC16F84. Pág. 306.

#### 2.4.2.1. Ventajas y desventajas del Puerto Serie RS-232

“El puerto serial maneja una comunicación entre dos dispositivos con una distancia máxima de 50 a 100 pies, dependiendo de la velocidad de transmisión de datos (bits/segundo.), y el tipo de cable.

Los equipos nuevos poseen interfaces de comunicación serial como los puertos USB, pero su desventaja es que no pueden realizar ciertas tareas que si realizan los Puertos RS-232.

Su ventaja principal es su uso con microcontroladores que lo convierten un puerto completamente útil versus a un puerto USB.

Como desventaja, el RS-232 no puede conectarse a más de dos dispositivos en una conexión, con un límite de transferencia de datos como máximo de 20000 (bits/segundo.), en caso de necesitar una mayor velocidad se debe utilizar, en este caso una interface RS-485 para lograr este fin.”<sup>21</sup>

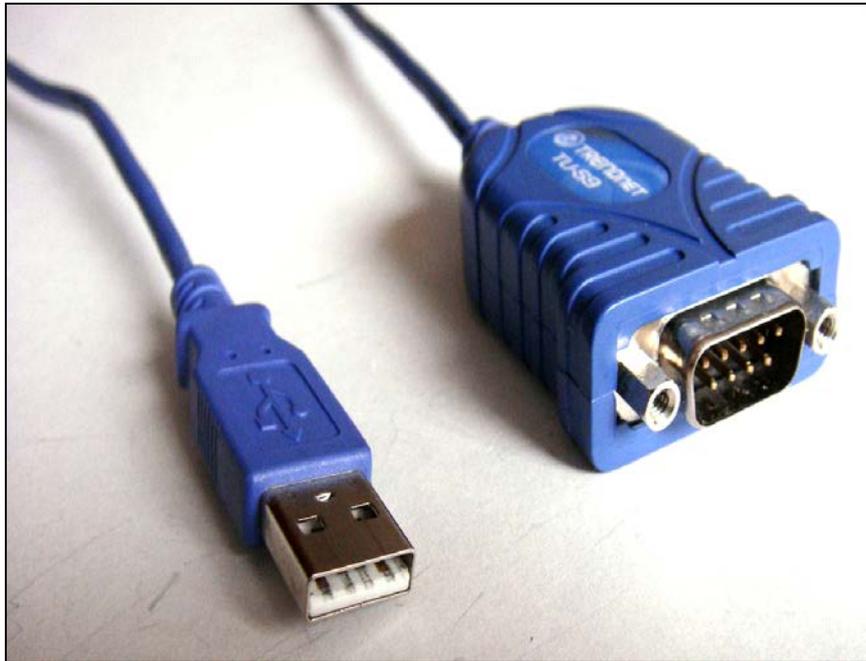
#### **2.4.3. Cable Adaptador USB a Serial RS-232 DB-9**

La comunicación serial en la mayoría de los sistemas operativos se realiza mediante periféricos de gran alcance, como el USB. Sin embargo existen varios productos de hardware que ofrecen aplicaciones mucho más especializadas que carecen de puerto USB y pueden estar compuestos de puertos RS-232 por sus facilidades de uso. Esta circunstancia es a menudo considerada como una falencia en los equipos de hardware.

Razón por la cual, los fabricantes de elementos electrónicos se vieron obligados a crear adaptadores para los dispositivos seriales con interfaz RS-232 mediante el uso de adaptadores USB (Figura 2.18).

---

<sup>21</sup> AXELSON, Jan. Serial port complete: programming and circuits for RS-232 and RS-485 links, About RS-232 Port, pág. 118.



**Figura 2.18. Cable Adaptador USB a DB9**

Autor: Andrés E. & Andrés Y.

#### **2.4.4. Velocidad de transmisión de datos**

Un dato importante a tener en cuenta en cualquier comunicación es la velocidad de transmisión, que es la cantidad de información enviada por la línea de transmisión en la unidad de tiempo.

##### **2.4.4.1. El Baudío**

Hay distintas unidades para expresar la velocidad de transmisión de datos, la más utilizada es el Baudio, que es proporcional a los Bits/segundo (bps), definidos como el número de bits de información enviados por segundo.

La velocidad a la que pueden trabajar los puertos COM de un ordenador está normalizada a 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 Baudios, etc.

Estos valores son demasiado pequeños para los estándares de hoy en día, pero suficientemente rápidos para multitud de aplicaciones.

Para que el PIC capte correctamente la señal de transmisión de datos enviada por el puerto serie se debe modificar este valor y fijarlo en 9600 bits/segundo o más.

El RS-232 puede transmitir datos en grupos de 5, 6, 7 u 8 bits a velocidades que pueden ser configuradas en la programación de visual basic 6.0 y del PIC.

Después de la transmisión de los datos, le sigue un bit opcional de paridad, éste indica si el número de bits transmitidos es par o impar ello permite detectar si existen fallas; y después 1 o 2 bits de stop.

La mayoría de las veces el protocolo utilizado, ya sea en la programación del PIC como en la programación de Visual Basic 6.0 para realizar los ejecutables, es "8, n, 1" (que significa 8 bits de datos sin paridad y con un bit de stop).

La comunicación de datos en un puerto serie RS232 se usa normalmente para efectuar comunicaciones asíncronas sin tiempo preestablecido para iniciarse. Los datos llegan en paquetes de información normalmente de 8 bits.

#### **2.4.5. Señales**

Para el óptimo funcionamiento de un puerto RS-232, se establecieron normas en las que se definen cuatro cosas: los nombres y funciones de las señales en el enlace, las características eléctricas de las señales, y los aspectos mecánicos como las asignaciones de pin de la interfaz, con una característica, de que a menudo no se utilizan los pines en su totalidad.

**Tabla 2.1. Descripción de señales del conector DB-9.**

DB9 (9 pin)	Señal	Fuente	Tipo	Descripción
1	CD	DCE	control	Carrier detect
2	RD	DCE	data	Received data
3	TD	DTE	data	Transmitted data
4	DTR	DTE	control	Data terminal ready
5	GND	-	-	Signal ground
6	DSR	DCE	control	Data set ready
7	RTS	DTE	control	Request to send
8	CTS	DCE	control	Clear to send
9	RI	DCE	control	Ring Indicator

Fuente: PALACIOS, Enrique. Microcontrolador PIC16F84. Pág. 309.

Autor: Andrés E. & Andrés Y.

#### **2.4.5.1. DTE y DCE**

“Estos nombres asignados a los pines de la interface, reflejan el propósito de la misma, en la que DTE y el DCE, siempre deben ser tomados en cuenta para una conexión y determina el tipo de señales de entrada o salida.

Todos los nombres de las señales son desde la perspectiva de los DTE. Por ejemplo, TD (transmitir datos) es una salida en un DTE y una entrada en DCE, mientras que RD (recibir datos) es una entrada en un DTE y una salida de un DCE.”<sup>22</sup>

---

<sup>22</sup> AXELSON, Jan. Serial port complete: programming and circuits for RS-232 and RS-485 links and networks, Signals, págs. 119,120.

#### **2.4.5.2. Nueve señales del RS-232**

“Las tres señales esenciales para la RS-232 de 2 vías de comunicación son los siguientes:

- **TD**, lleva los datos del DTE al DCE, también llamado TX y TXD.
- **RD**, lleva los datos desde el DCE al DTE, también llamado RX y RXD.
- **SG**, de tierra de señal, también llamado GND y SGND.

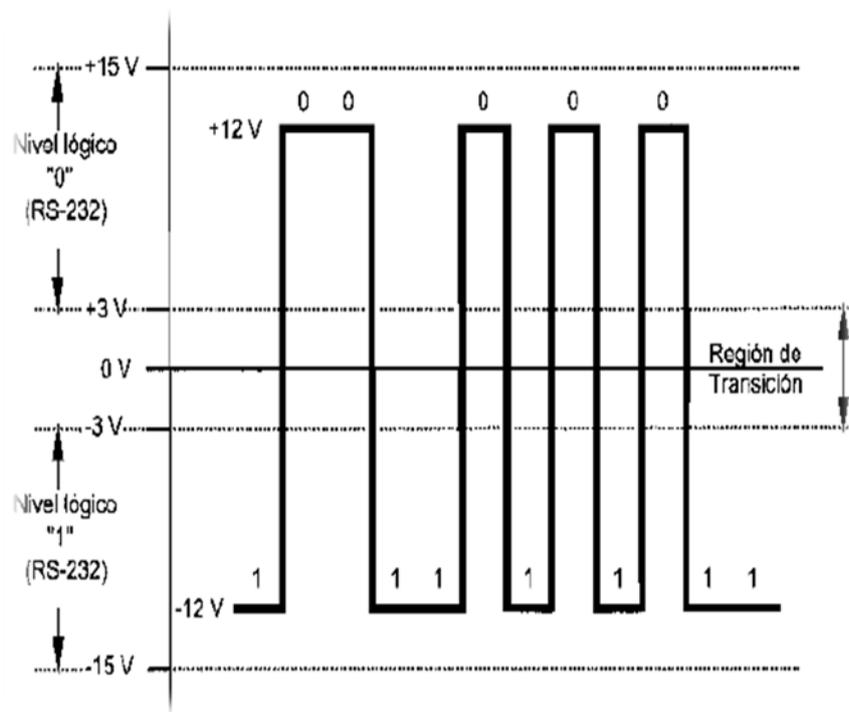
También existen dos pares de señales de intercambio: DTR / DSR y RTS / CTS, que se presentan en un enlace o conexión. En algunos enlaces, DTR y DSR se plantean en el inicio y sólo indican que el equipo está encendido y funcionando. RTS y CTS proporciona información adicional acerca de si el dispositivo está listo para recibir datos.

Las demás señales que son RI y CD, son señales de control opcional destinados a la comunicación sobre la disposición de un dispositivo, etc.”<sup>23</sup>

<sup>23</sup> AXELSON, Jan. Serial port complete: programming and circuits for RS-232 and RS-485 links and Networks, The PC's nine signals, págs.121, 122.

#### 2.4.6. Niveles lógicos RS-232

La figura 2.19 ilustra los requisitos en cuanto a niveles lógicos que debe cumplir una transmisión serie según la norma RS-232. A destacar:



**Figura 2.19. Niveles de tensión lógicos para RS-232**

Fuente: PALACIOS, Enrique. Microcontrolador PIC 16F84. Pág. 307.

Los datos se transmiten con lógica negativa, es decir, un voltaje positivo en la conexión representa un "0", mientras que un voltaje negativo representa un "1".

Para garantizar un “0” lógico una línea debe mantener un voltaje entre +3 y +15 [V]; del mismo modo para “1” lógico garantizado debe estar entre -3 y -15 [V].

Los voltajes más usados son +12 [V] para el “0” y -12 [V] para el “1”. Es importante conocer que cuando un puerto serie no está transmitiendo, este mantiene el terminal de transmisión a “1” lógico a -12 [V].

La banda muerta entre +3 [V] y -3 [V] es llamada como la región de transición donde los niveles lógicos no están definidos. Esto significa que cualquier valor entre +3 y -3 voltios puede interpretarse ambiguamente como “0” ó “1”.

#### **2.4.7. Elementos eléctricos y electrónicos**

Los componentes electrónicos son primordiales para crear circuitos por lo que en la actualidad la evolución de varios elementos eléctricos y electrónicos se deben a las investigaciones en busca de mejoras. En electricidad y electrónica, el movimiento de los electrones fomentan el funcionamiento de los circuitos. “Los equipos electrónicos son sistemas técnicos complejos contruidos con piezas diferentes conectadas entre sí para realizar unas funciones electrónicas concretas, tales como rectificación, amplificación, oscilación, conversión analógica/digital, etc.

Cada una de dichas piezas se designa generalmente con el nombre de componente electrónico, concepto que puede definirse como elemento eléctrico que realiza una función física simple por si solo, si es utilizado de manera adecuada. La noción de componente implica la idea de indivisibilidad, es decir, un componente no puede ser dividido en partes sin perder su función específica.

Pero el término componente se utiliza también para designar un conjunto funcional dentro de un sistema, así por ejemplo, un sintonizador es un componente de una cadena de audio o un disco duro es un componente de un ordenador, y ambos pueden descomponerse en puntos más simples.”<sup>24</sup>

---

<sup>24</sup> PRAT VIÑAS, Lluís, Circuitos y Dispositivos electrónicos, Componentes Electrónicos, pág. 403. A cada componente electrónico le corresponden propiedades eléctricas y mecánicas que permiten definir con precisión su comportamiento en determinadas condiciones, es lo que constituyen las especificaciones técnicas que son utilizadas como bases o normas que permiten garantizar las características de empleo.

Entre las especificaciones se encuentran las dimensiones geométricas, la forma de realizar la conexión eléctrica, las características eléctricas típicas y máximas de empleo, las condiciones ambientales extremas (temperatura, humedad, etc.).

“Según el tipo de función que desempeñan en los circuitos electrónicos, los componentes pueden clasificarse como se indica a continuación:

- Componentes activos, capaces de realizar funciones de control y amplificación de potencia, u otras más complejas. Son los transistores, tubos electrónicos y circuitos integrados.
- Componentes pasivos, que no pueden realizar funciones de control o amplificación, ni otras funciones electrónicas complejas. La tensión y la corriente presentes en ellos suelen estar relacionadas por una proporcionalidad, una derivación o una integración con respecto al tiempo.

Son las resistencias, condensadores, bobinas, transformadores, diodos, termistores.

- Transductores, capaces de transformar una energía en otra de diferente tipo. Por ejemplo, tubos de rayos catódicos, motores, altavoces, micrófonos, detectores, cápsulas grabadoras o reproductores, etc.
- Elementos de conmutación e interrupción, que incluyen conmutadores, interruptores y relés.
- Elementos de conexión, que incluyen hilos y cables de conexión, placas de circuitos impresos, conectores, zócalos, etc."<sup>25</sup>

#### **2.4.8. Conexión Puerto RS232, MAX232 y PIC16F877A**

Para conectar el puerto RS-232 con el PIC, y su posterior comunicación se necesita utilizar solo tres patillas que son:

- Pin de transmisión (TxD),
- Pin de recepción (RxD)
- Pin de masa (SG)

Estos pines permiten realizar la transmisión y recepción de datos, así como su conexión a tierra (SG).

El lenguaje de comunicación entre el puerto RS-232 difiere a la que el Microcontrolador posee razón por la cual es necesario utilizar una interface que traduzca las funciones a un mismo lenguaje. Esta tarea la cumple el chip MAX-

232. En la figura 2.20 se puede apreciar las líneas de conexión, entre el puerto RS-232, el circuito integrado MAX232 y el PIC16F877A; así como también las líneas de alimentación de los mismos (Ver Anexo 4).

<sup>25</sup> PRAT VIÑAS, Luís, Circuitos y Dispositivos electrónicos, Componentes Electrónicos, pág. 403.

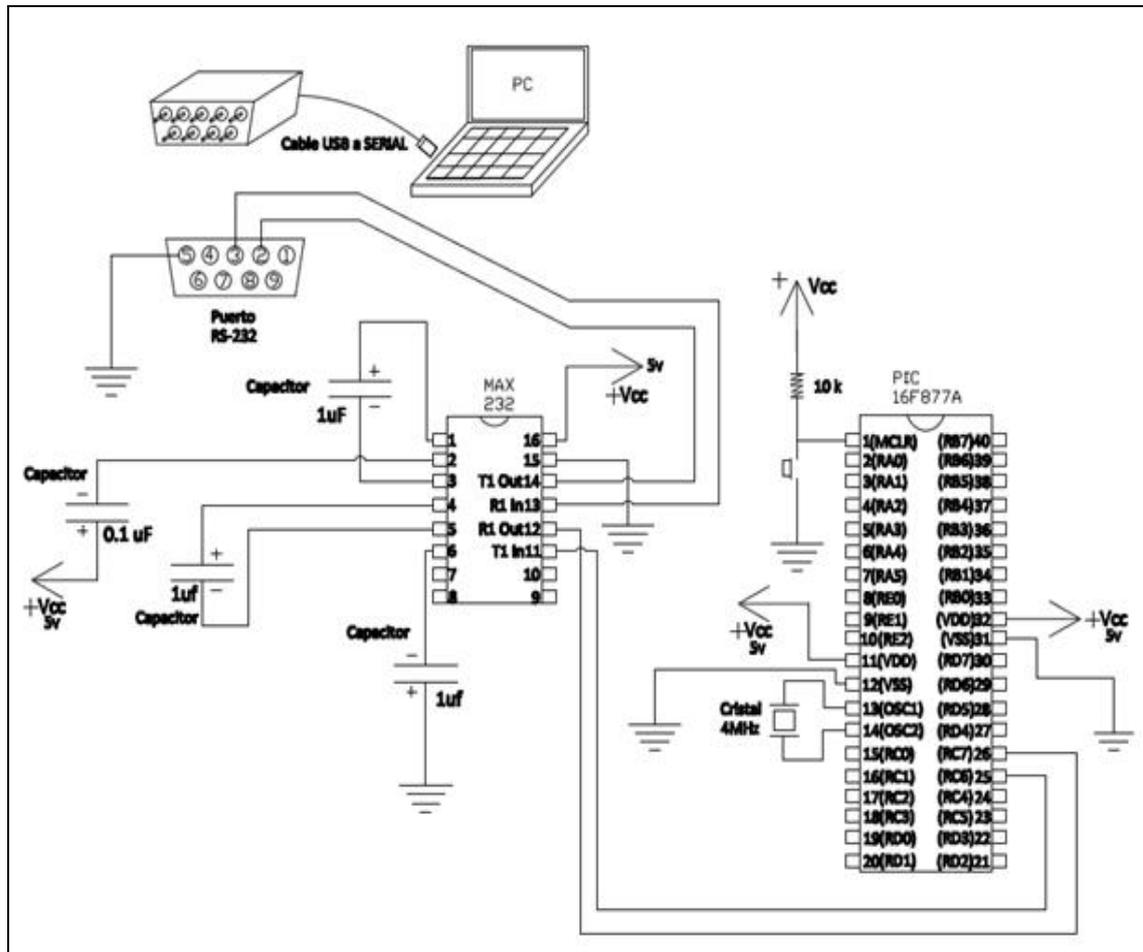


Figura 2.20. Conexión PC - Puerto RS-232 - MAX-232 y PIC16f877A

Autor: Andrés E. & Andrés Y.

### 2.4.8.1. Circuito Integrado

“Un circuito integrado es un circuito que agrupa transistores, diodos y resistencias fabricados en una sola pieza de material semiconductor (normalmente silicio) al que habitualmente se denomina chip.” El chip se encuentra dentro de un receptáculo plástico o cerámico que lo protege y que se denomina encapsulado, del que sobresalen una serie de patillas o pines metálicos.

---

<sup>26</sup> BLANCO VIEJO, Cecilio. Fundamentos de Electrónica Digital, Circuitos Integrados, pág. 18.

Estas patillas o pines están conectadas a los terminales del chip, son sus entradas y salidas, su conexión con el mundo exterior.

“Existen dos tipos de encapsulados básicos: de inserción y de montaje superficial. En los primeros, los pines se insertan en los agujeros hechos a tal efecto en la tarjeta de circuito impreso y se sueldan a las pistas por la cara opuesta. Los encapsulados de montaje superficial, una tecnología más moderna, permiten ahorrar espacio en la tarjeta de circuito impreso y evitan la realización de taladros. Las patillas de los circuitos integrados de montaje superficial se sueldan directamente a las pistas de una de las caras de la tarjeta de circuito impreso, la otra queda totalmente libre para añadir otros circuitos o pistas que puedan ser necesarios para el diseño del sistema.”<sup>27</sup>

#### **2.4.8.2. MAX232**

La comunicación con el PC se realiza a través de la interface serial RS-232, que para este caso se utiliza un conector de 9 pines (DB-9), y para lo cual se requiere del circuito integrado MAX232, el cual se encarga de cambiar los niveles RS-232 (cerca de +12 [V] y -12 [V]) propios de la interface, a voltajes lógicos TTL (0 [V] a

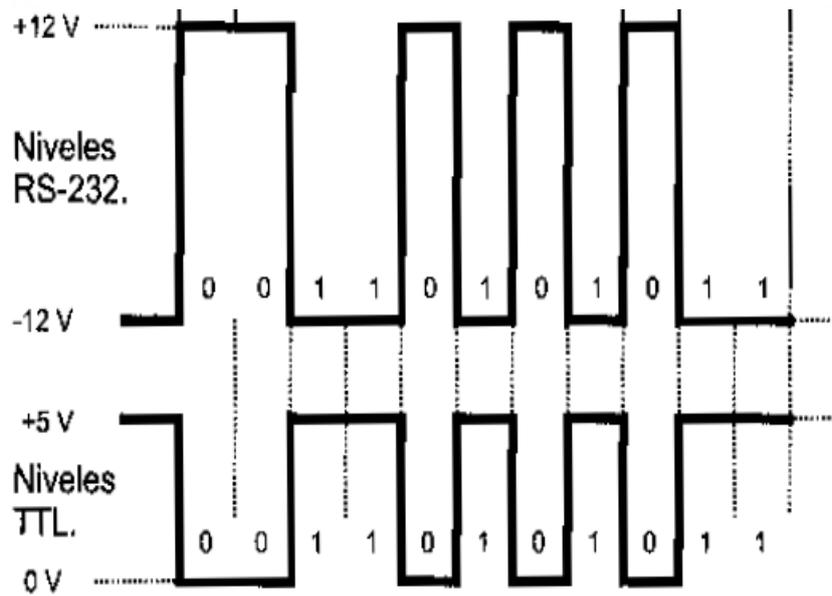
+5 [V]), que son requeridos por el PIC para su funcionamiento, y viceversa sin requerir nada más que una fuente de +5 [V].

---

<sup>27</sup> BLANCO VIEJO, Cecilio. Fundamentos de Electrónica Digital, Circuitos Integrados, pág. 18.

Es importante conocer, que el Microcontrolador trabaja con lógica positiva a diferencia del puerto RS-232 del ordenador que trabaja con lógica negativa. Por tanto la forma de la señal en el microcontrolador (niveles TTL) será inversa a los niveles RS-232 tal como se muestra en la figura 2.21.

El MAX232 cambia los niveles TTL a los del estándar RS-232 cuando se hace una transmisión, y cambia los niveles RS-232 a TTL cuando se tiene una recepción.



**Figura 2.21. Niveles RS 232 y niveles TTL**

Fuente: PALACIOS, Enrique. Microcontrolador PIC 16F84. Pág. 308.

En resumen, el MAX 232 (Figura 2.22), permite realizar una comunicación entre la computadora y el PIC, debido a que convierte los niveles RS232 (+ 12 voltios DC y -12 voltios DC) a niveles TTL (0 a 5 voltios DC), estos últimos necesarios para el correcto funcionamiento del PIC. En la figura 2.23 se muestra la estructura interna del MAX232.



**Figura 2.22. Circuito integrado MAX232.**

Autor: Andrés E. & Andrés Y.

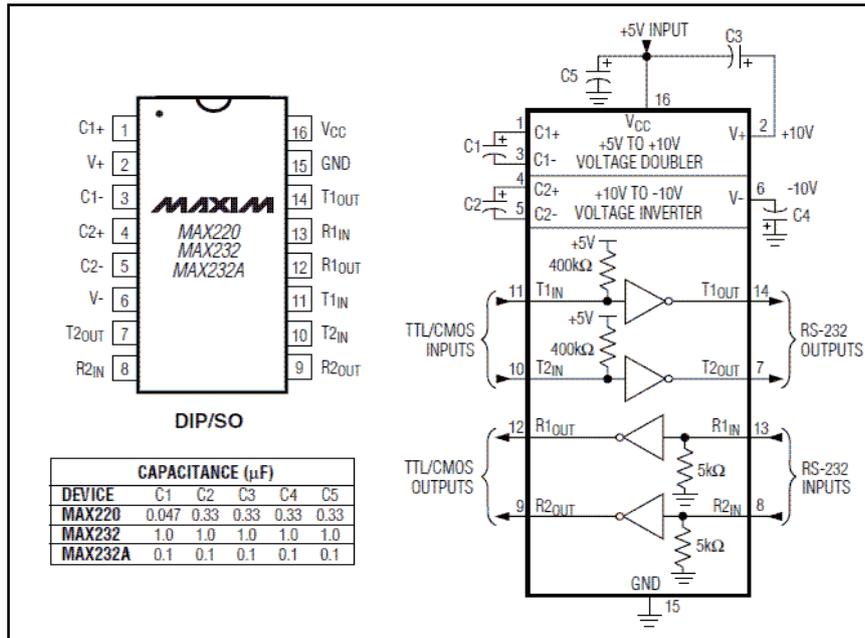


Figura 2.23. Configuración interna y operación típica del circuito integrado MAX232.

Fuente: Maxim Datasheet MAX220 – MAX249 Pág.17

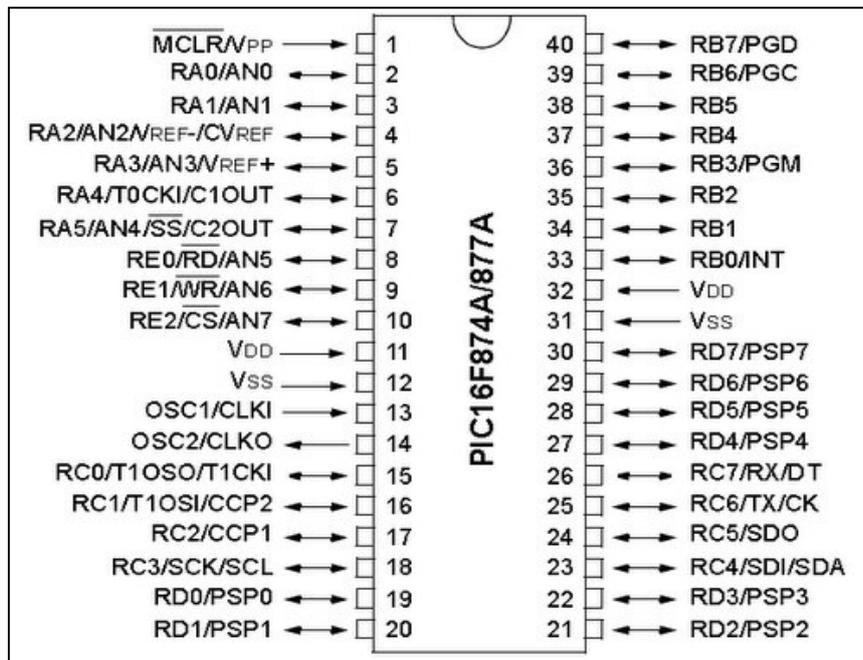
### 2.4.8.3. ¿Qué es un Microcontrolador?

“Un microcontrolador es un circuito integrado, en cuyo interior posee toda la arquitectura de un computador.”<sup>28</sup>

<sup>28</sup> REYES, Carlos. Microcontroladores PIC Programación en Basic, El Microcontrolador PIC, pág. 17.

“Un microcontrolador de fábrica, no realiza tarea alguna, este debe ser programado para que realice tareas simples y complejas. Un microcontrolador es capaz de realizar la tarea de muchos circuitos lógicos como compuertas lógicas, conversores A/D, D/A, temporizadores, decodificadores, etc., simplificando todo el diseño a una placa de reducido tamaño y pocos elementos.”<sup>29</sup>

“Los microcontroladores PIC (Peripheral Interface Controller), son fabricados por la empresa MICROCHIP Technology, con alrededor de 180 modelos, que se caracterizan por su velocidad bajo costo, bajo consumo de potencia y gran disponibilidad de herramientas para su programación.”<sup>30</sup>



**Figura 2.24: Microcontrolador PIC 16F877A**

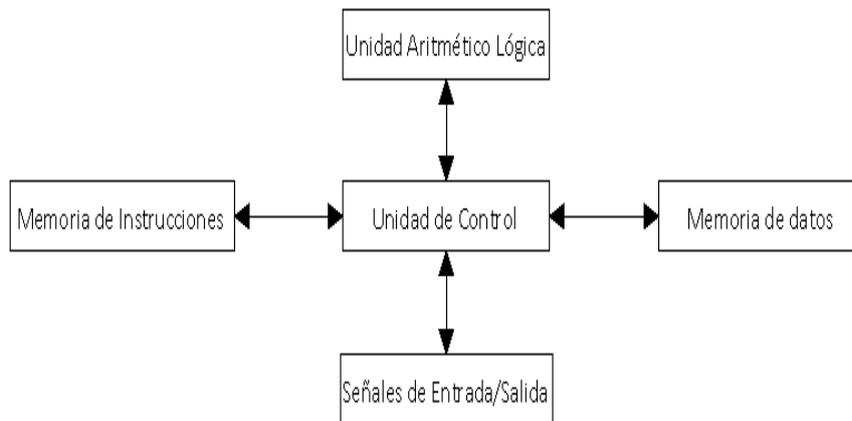
Fuente: Datasheet PIC 16F87XX

<sup>29</sup> REYES, Carlos. Microcontroladores PIC Programación en Basic, El Microcontrolador PIC, pág. 17.

<sup>30</sup> REYES, Carlos. Microcontroladores PIC Programación en Basic, El Microcontrolador PIC, pág. 18.

#### 2.4.8.3.1. Microcontrolador PIC 16F877A

PIC, perteneciente a la compañía Microchip, compuestos por 40 pines, que posee un procesador Harvard, consiguiendo mayor rendimiento en el procesamiento de instrucciones.



**Figura 2.25. La arquitectura Harvard maneja la memoria de datos y la memoria de instrucciones por separado y con diferentes capacidades<sup>31</sup>.**

- Memoria de programa

“Conocida como memoria de instrucciones, en donde se describe las órdenes para que el CPU las ejecute. El PIC 16F877A, tiene memoria de programa no volátil tipo FLASH, que se caracteriza por ser más rápido el proceso de escritura y borrado eléctrico, además dispone de mayor capacidad de almacenamiento.”<sup>32</sup>

---

<sup>31</sup> REYES, Carlos. Microcontroladores PIC Programación en Basic, El Microcontrolador PIC, pág. 19.

<sup>32</sup> REYES, Carlos. Microcontroladores PIC Programación en Basic, El Microcontrolador PIC, pág. 19.

- Memoria de datos

“El PIC, tiene dos tipos de memorias de datos, la RAM estática o SRAM (Random Access Memory) o memoria de acceso casual, que es un tipo de memoria volátil, es decir sus datos permanecen en la memoria mientras exista alimentación en el

dispositivo y es de vital importancia porque ahí residen dos tipos de datos, los registros de propósito general (GPR), en donde se almacenan variables y los registros especiales (SFR), que son los encargados de llevar el contador del programa, el conteo del temporizador, el estado de los puertos, la configuración de las interrupciones, etc.

El otro tipo de memoria, es una memoria auxiliar no volátil llamada EEPROM, esta memoria puede ser accedida por el usuario mediante programación, es muy útil para almacenar datos que el usuario necesita que se conserven aún sin alimentación, tal es el caso de la clave de una alarma, esta puede ser modificada, pero no debe perderse por un corte de energía.”<sup>33</sup>

#### **2.4.8.4. Capacitor**

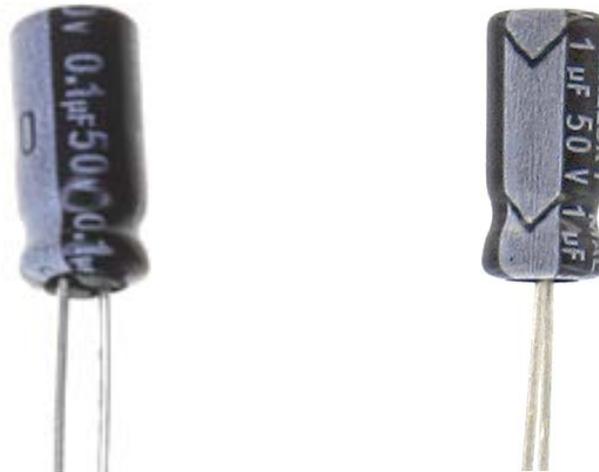
Los capacitores tienen la función de almacenar tensión y estabilizar el flujo de la misma cuando esta baja. Un capacitor o condensador, está formado por dos placas de metal que están polarizadas, una opuesta a la otra, separadas por un dieléctrico que está compuesto por un material aislante como plástico, papel, etc.

---

<sup>33</sup> REYES, Carlos. Microcontroladores PIC Programación en Basic, El Microcontrolador PIC, pág. 20.

Existen capacitores de cerámicos y electrolíticos, el dieléctrico en el caso de los capacitores electrolíticos es un ácido líquido llamado electrolito. Generalmente se los utilizan como filtros de alimentación de corriente, almacenar carga, regular el voltaje de salida y fluctuaciones de corriente de salida rectificadas.

Para el perfecto funcionamiento del MAX232 según las indicaciones en su datasheet (hoja de datos con especificaciones técnicas que provee el fabricante del componente), se requiere del uso de condensadores que son de 0.1uf y 1uf (Figura 2.26). En la figura 2.23 se detalla su conexión con el MAX232.



**Figura 2.26. Capacitores de 0.1 y 1uf**

Autor: Andrés E. & Andrés Y.

## **2.5. DISEÑO DEL BLOQUE DE CONTROL (PIC16F877A - ULN 2803 – RELE)**

Este bloque principalmente está compuesto por el PIC, dos ULN2803 y trece relés. Los pines de salidas que fueron utilizados del PIC, van desde el pin #21 hasta el pin #40, del lado derecho del PIC (Figura 2.27), lo cual permite una mayor facilidad para realizar las conexiones al momento de hacer el diseño de la placa PCB en el programa ARES.

Se ocuparon en el diseño dos circuitos integrados ULN2803, las salidas del primer CI están conectadas a 7 relés; y las salidas del segundo CI están conectadas a 6 relés; sumando un total de trece relés que serán controlados por el PIC. La razón

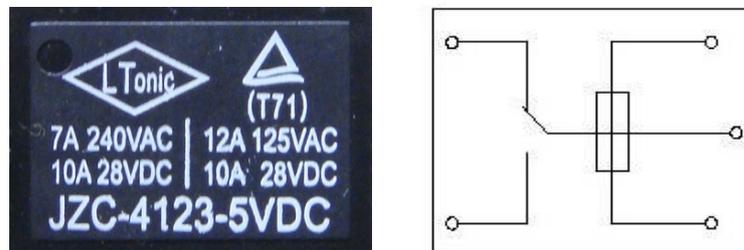


**Figura 2.28. Descripción de pines del circuito integrado ULN2803.**

Autor: Andrés E. & Andrés Y.

### **2.5.2. Relé**

En el desarrollo de la placa PCB, decidimos utilizar varios relés que son interruptores electromecánicos, que se activan con la señal recibida del PIC, facilitando una conexión segura y una rápida ejecución para activar o desactivar una carga, que en este caso serán los accesorios. (Figura 2.29).



**Figura 2.29. Vista superior del relé y el diagrama eléctrico del mismo.**

Autor: Andrés E. & Andrés Y.

### **2.5.3. Diodo 1N4007**

El diodo siempre se utiliza para proteger al circuito de la placa, es por eso que el diodo se conecta en paralelo al relé, para que no deje pasar la corriente de retorno liberada por la bobina cuando se desconecta, en caso de no existir este elemento el circuito integrado ULN 2803 se dañaría rápidamente porque la corriente de rebote es alta, aunque la tensión baja.



**Figura 2.30. Diodo.**

Autor: Andrés E. & Andrés Y.

## 2.6. DISEÑO DEL BLOQUE DE ALIMENTACIÓN

### 2.6.1. Regulador de voltaje 7805

En el desarrollo de la placa PCB, la aplicación de reguladores de voltaje 7805 permitió obtener a la salida de éstos, un voltaje de 5 voltios DC a partir de la fuente de 12 voltios DC de un vehículo; dicho voltaje es necesario para la operación y correcto funcionamiento de varios elementos electrónicos que conforman al circuito. Cabe recalcar que la intensidad máxima que soporta un regulador 7805 es de 1 amperio. A continuación se muestra la configuración del patillaje del regulador. (Figura 2.31)

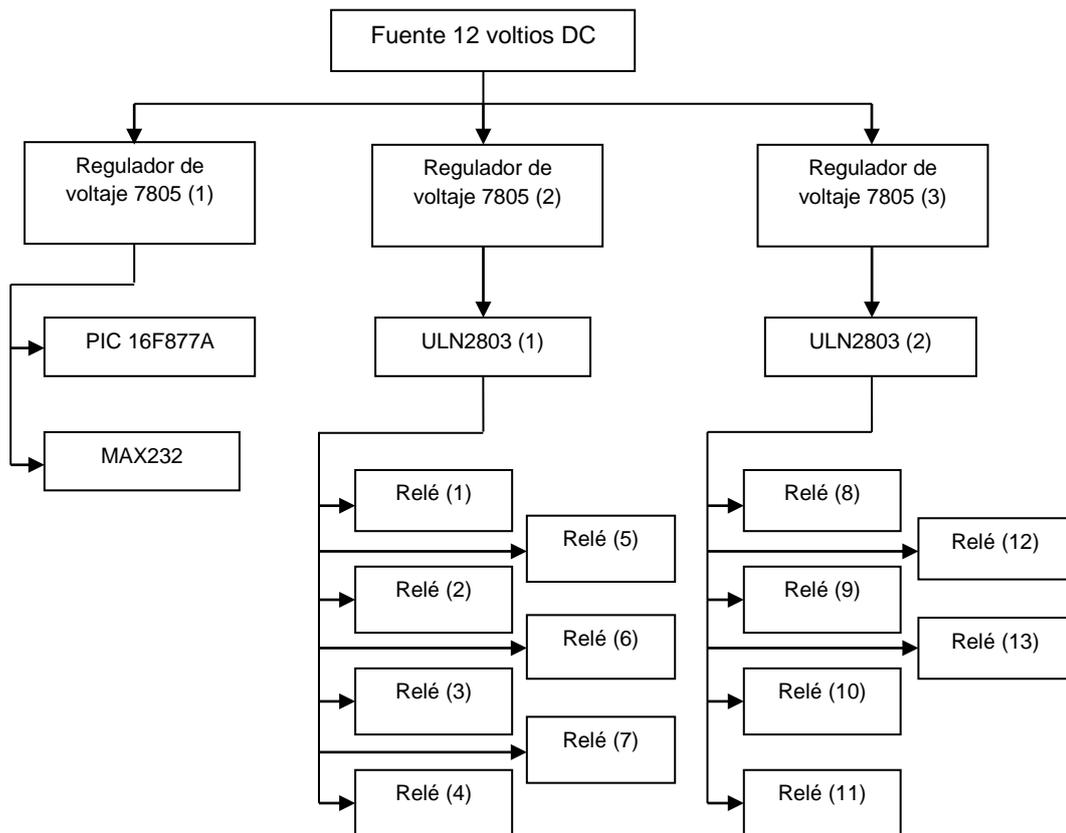


**Figura 2.31. Configuración del patillaje del regulador 7805**

Autor: Andrés E. & Andrés Y.

En este caso se utilizaron tres reguladores 7805 (Ver Anexo 5), y se diseñaron tres bloques que conforman la placa. Así como se muestra en la figura 2.32.

- El primer regulador está conectado al PIC 16F877A, junto con el MAX232.
- El segundo regulador está conectado al primer ULN 2803 y a 7 relés.
- El tercer regulador está conectado al segundo ULN 2803 y a 6 relés.



**Figura 2.32. Esquema de bloques de la conexión de los reguladores 7805.**

Autor: Andrés E. & Andrés Y.

## 2.7. SOFTWARE

“Es el conjunto de instrucciones (programas de computadora) que cuando se ejecutan, suministran la función y comportamiento deseado. Incluso se lo define como una estructura de datos que facilitan a los programas manipular adecuadamente la información. Generalmente se lo asocia como el grupo de documentos que describen la operación y uso de los programas.”<sup>34</sup>

### 2.7.1. Programación

El software de computadora es información que existe en dos formas básicas:

“Componentes no ejecutables en la máquina y componentes ejecutables en la máquina. La forma en la que el software se traduce en una forma ejecutable por la máquina se ilustra en la figura 2.33”<sup>35</sup>



Figura 2.33. Pasos en la traducción del software<sup>36</sup>

<sup>34</sup> PRESSMAN, Roger. Ingeniería del Software, Definición Software, pág. 6.

<sup>35</sup> PRESSMAN, Roger. Ingeniería del Software, Componentes del Software, pág. 9.

<sup>36</sup> PRESSMAN, Roger. Ingeniería del Software, Componentes del Software, pág. 9.

El diseño del software se traduce a un lenguaje que especifica la estructura de datos del software, los atributos procedimentales y los requerimientos

relacionados. Un traductor procesa este lenguaje convirtiendo el código en instrucciones de máquina ejecutables.

“Idealmente los humanos podrían comunicarse con las computadoras usando un lenguaje natural (por ejemplo, inglés, español, ruso). Desgraciadamente, los grandes vocabularios, gramáticas sofisticadas y nuestro uso del contexto para comprender, impiden la comunicación máquina - hombre mediante el lenguaje natural.”<sup>37</sup>

La investigación en procesamiento de información básica y reconocimiento del lenguaje natural ha puesto las bases para el uso del lenguaje natural como un medio de comunicación con la computadora.”

### **2.7.2. Tipos de Software**

“Todos los lenguajes de programación son lenguajes artificiales. Todos tienen un vocabulario limitado, una gramática definida explícitamente y reglas bien formadas de sintaxis y semántica. Estos atributos son esenciales para la traducción por la máquina. Las clases de lenguajes que son componentes del software se caracterizan como lenguajes a nivel máquina, lenguajes de alto nivel y lenguajes no procedimentales (bases de datos, etc.).”<sup>38</sup>

---

<sup>38</sup> PRESSMAN, Roger. Ingeniería del Software, Componentes del Software, pág. 10.

Los lenguajes a nivel máquina, representados por un extracto de un lenguaje ensamblador de un microprocesador, son una representación simbólica del conjunto de instrucciones de la CPU.

Cuando un buen realizador de software produce un programa mantenible y bien documentado, el lenguaje a nivel de máquina puede hacer un uso extremadamente eficiente de la memoria y optimizar la velocidad de ejecución del programa. Cuando un programa está mal diseñado y con poca documentación, el lenguaje máquina tiende a aumentar los problemas que se presentarán.

“Aunque el lenguaje a nivel de máquina suministra una atractiva velocidad de ejecución y características de memoria, tiene varias desventajas importantes:

- El tiempo de implementación se prolonga debido a la complejidad del lenguaje y al bajo nivel de abstracción en el que se representa la información.
- Los programas resultantes son difíciles de leer.
- La prueba y mantenimiento es difícil.

La productividad del software se deteriora seriamente cuando se utiliza un lenguaje a nivel de máquina.”<sup>39</sup>

Debido a las desventajas asociadas con su uso, es probable que estos lenguajes se utilicen muy poco durante la próxima década. Los lenguajes de alto nivel (Basic y otros), permiten al que desarrolla los programas y al programa ser independientes de la máquina.

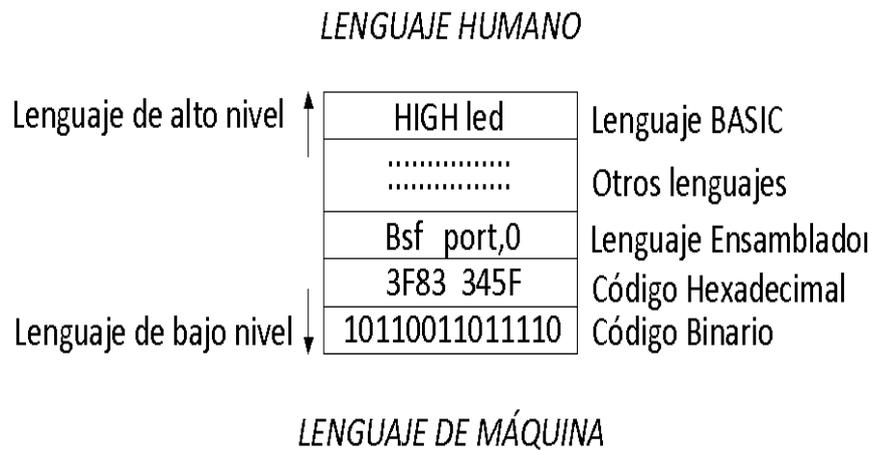
---

<sup>39</sup> PRESSMAN, Roger. Ingeniería del Software, Componentes del Software, pág. 10.

Cuando se utiliza un traductor más sofisticado, el vocabulario, gramática, sintaxis y semántica de un lenguaje de alto nivel pueden ser mucho más complejas que los lenguajes de nivel máquina como salida. La función del traductor del lenguaje de programación, esto es, a la transformación de un lenguaje en instrucciones máquina ejecutables. El más frecuente traductor de lenguajes de alto nivel es el compilador. Mediante la evaluación global de un programa, un compilador es capaz de optimizar el tamaño de la memoria y velocidad de ejecución de las instrucciones máquinas ejecutables que produce. Las instrucciones máquina ejecutables están en el fondo de la configuración del software. Un código especializado hexadecimal o de otro tipo se utiliza para representar la cadena binaria de bits que especifica los pasos del procesamiento de la CPU. El código máquina, los lenguajes ensambladores (nivel máquina) y lenguajes de programación de alto nivel son normalmente referenciados como las tres primeras generaciones de lenguajes de computadora. Con cualquiera de estos lenguajes, el programador debe preocuparse tanto de la especificación de la estructura de la información como de la de control del propio programa.

#### **2.7.2.1. Diferencia entre el lenguaje Basic y el lenguaje ensamblador**

Para poder entender la diferencia entre los dos lenguajes de programación, se debe tener en claro qué es un lenguaje de alto nivel y qué es un lenguaje de bajo nivel, a través del siguiente cuadro podemos ver los niveles de programación (Figura 2.34).



**Figura 2.34. Cuadro de los niveles de programación.<sup>40</sup>**

El lenguaje que más se acerca a los humanos es el de más alto nivel, el lenguaje más próximo al tipo de datos que entiende el microcontrolador es un bajo nivel.

La programación en lenguaje ensamblador, se diferencia del lenguaje de alto nivel como BASIC, el que más entendemos los humanos, se debe a que es mucho más largo y difícil de entender, además debe conocer las posiciones de las memoria que están disponibles para este PIC, así como también la arquitectura del PIC, también se deben realizar cálculos muy precisos para realizar ciertas funciones e incluso BASIC permite al programador tener un ahorro de tiempo considerable que al programar en Ensamblador.

---

<sup>40</sup> REYES, Carlos. Microcontroladores PIC Programación en Basic, Diferencia entre el Lenguaje BASIC y el ensamblador, págs. 33, 34.

## **CAPÍTULO 3**

### **DISEÑO, CONSTRUCCIÓN DE LA PLACA PCB Y SELECCIÓN DE ELEMENTOS**

#### **3.1. DISEÑO DE LA PLACA PCB**

Para desarrollar la placa PCB, se necesita un software especial en este caso PROTEUS, el cual se subdivide en ISIS, para el desarrollo del plano eléctrico principal para luego pasarlo a ARES, en el cual los elementos se conectan, pero es necesario especificar tamaños y realizar cambios para obtener una placa correctamente diseñada.

##### **3.1.1. Desarrollo en ISIS**

Como antes se mencionó el desarrollo del plano eléctrico (Figura 3.1), está compuesto por los elementos anteriormente señalados en el capítulo 2 como: PIC 16F877A, MAX-232, Conector para puerto serial DB9, ULN 2803, Relés, Capacitores y reguladores de voltaje. En este programa se conectan entre si cada uno de los elementos a utilizar con sus respectivos pines.

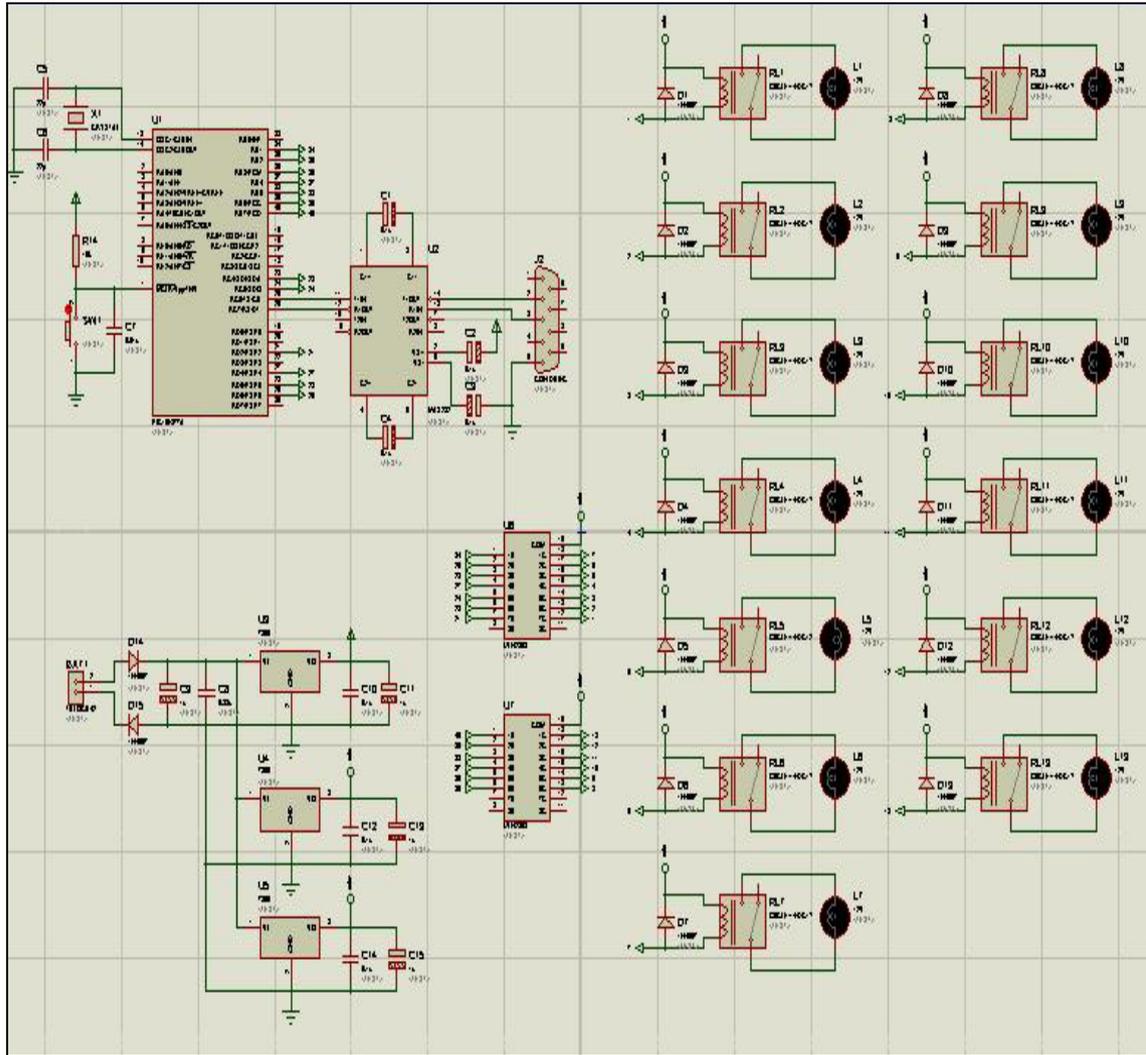


Figura 3.1. Plano eléctrico diseñado en ISIS

Autor: Andrés E. & Andrés Y.

### 3.1.2. Desarrollo en ARES

Para diseñar la placa PCB, se debe tener en cuenta el tamaño de las pistas para el paso de corriente, el tamaño de la placa e incluso se deben ordenar a los elementos haciendo que éstos estén conectados correctamente y que estén ubicados de una manera eficiente en la placa. Las dimensiones en las cuales se desarrollo el circuito PCB comprenden, de alto 18,5 cm y de largo 21,7 cm. (Figura 3.2).

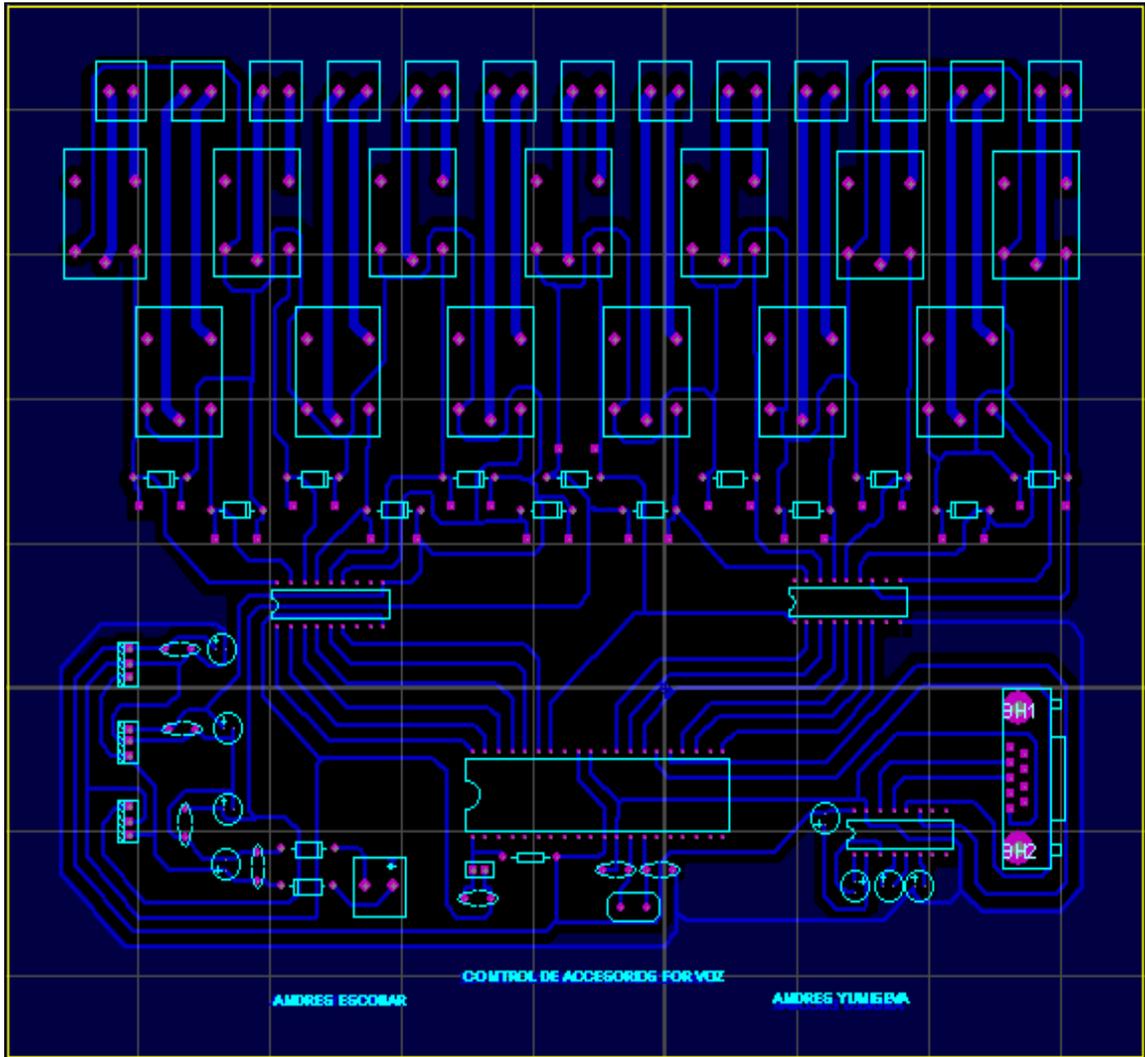


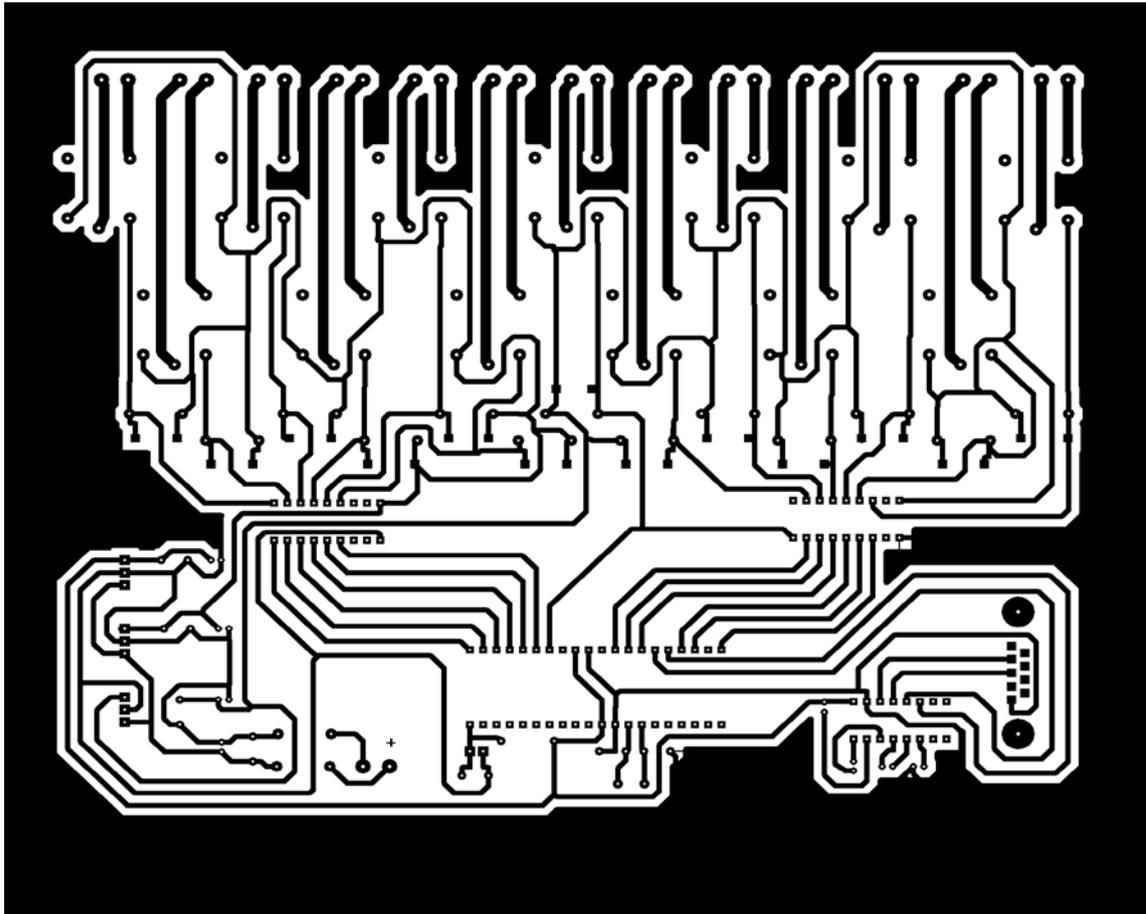
Figura 3.2. Vista superior de la placa PCB diseñada en ARES

Autor: Andrés E. & Andrés Y.

## 3.2. CONSTRUCCIÓN DE LA PLACA PCB

### 3.2.1. Fabricación del circuito

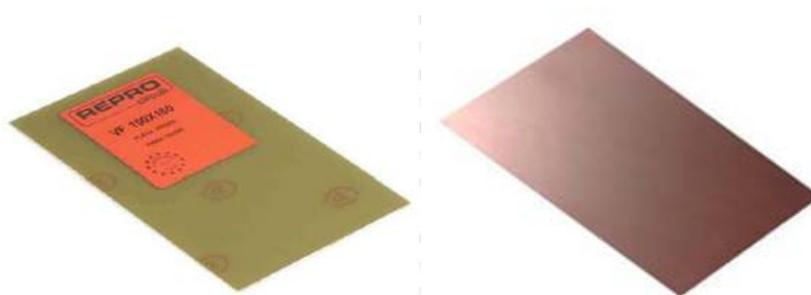
Para crear la placa PCB, se parte desde la impresión del circuito realizado en el programa ARES (Figura 3.3), el cual fue elegido ya que no es demasiado complicado de usar, además que está disponible en el mercado ecuatoriano.



**Figura 3.3. Diagrama que se imprime en el papel foto sensible (vista inferior de la placa).**

Autor: Andrés E. & Andrés Y.

El elemento principal para que se materialice el circuito PCB, es la placa de acrílico que tiene adherido una lámina de cobre en su parte inferior, la cual se la conoce como placa PCB virgen. (Figura 3.4).



**Figura 3.4. Vista superior e inferior de la placa PCB virgen.**

Autor: Andrés E. & Andrés Y.

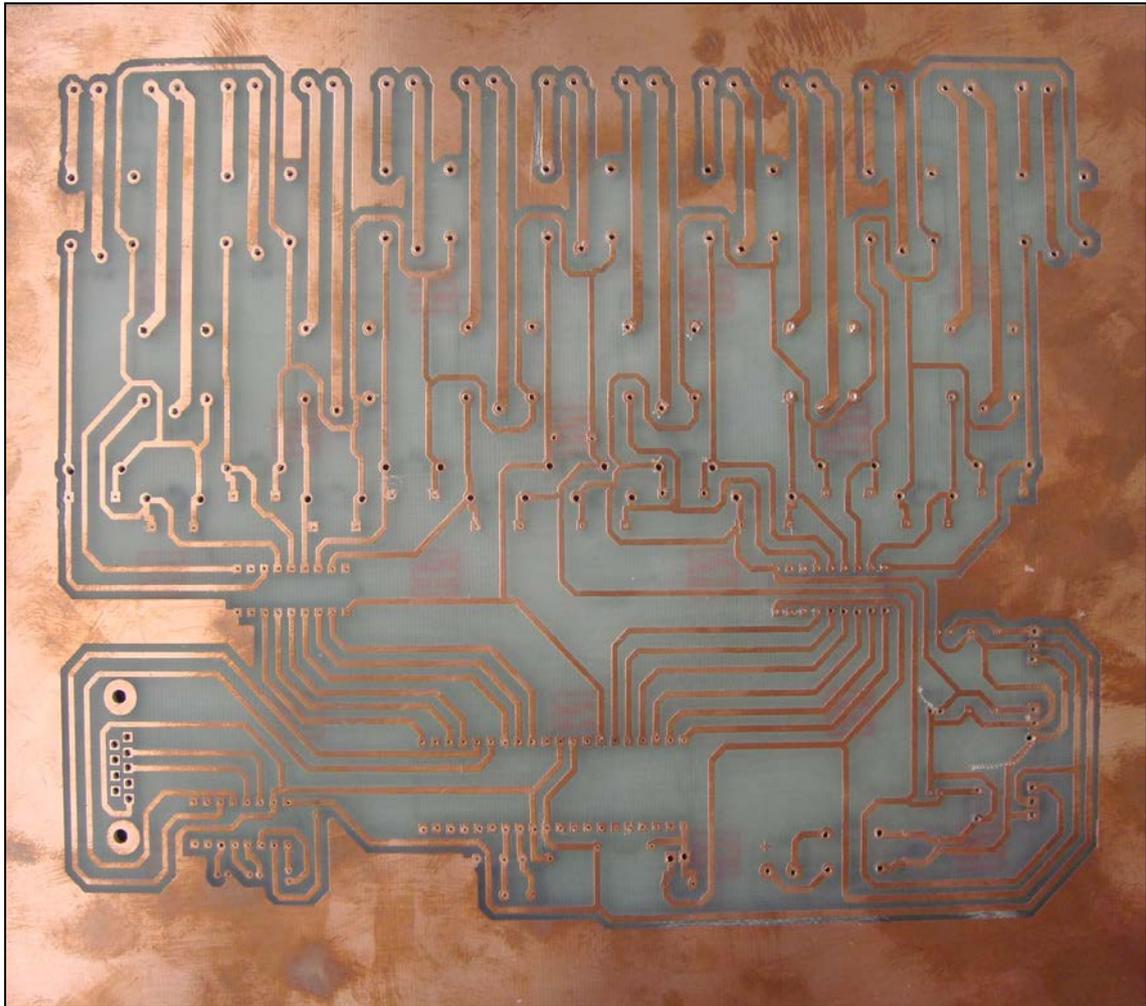
Para transferir el circuito a la placa, se utiliza papel foto transferible en donde se imprime el circuito (cabe recalcar que la impresora debe ser a láser o puede también ser una copiadora, porque funcionan con tóner).

Una vez sujeto el papel foto transferible con la placa, se humedece un trapo cubriendo a la misma y aplicando fuerza con una plancha caliente.

Se espera hasta que el circuito se transfiera a la placa, por lo que con cuidado se chequea que éste se haya transferido completamente.

Después se calienta agua con percloruro férrico, se sumerge a la placa en el recipiente donde esta la mezcla antes señalada, con lo que este ácido reacciona en las partes donde no esta el circuito en tóner, producto de la reacción el cobre que esta libre se disuelve en el líquido, desprendiéndose de la placa, y dejando libre al circuito.

Con una estopa de alambre se limpia las pistas que están cubiertas por el tóner, dejando al cobre sobrante sobre la placa, es decir el circuito principal. (Figura 3.5).



**Figura 3.5. Vista inferior de la placa PCB terminada.**

Autor: Andrés E. & Andrés Y.

Así como se puede imprimir con el programa ARES el circuito en papel de las pistas para la placa PCB, también es posible imprimir en el papel foto transferible, la ubicación de cada uno de los elementos electrónicos, también su polaridad para que al momento de soldar los materiales no haya errores (Figura 3.6).

Inclusive es posible escribir cualquier leyenda en la parte superior de la placa, en este caso se ha impreso los nombres de los autores del proyecto y el nombre principal de este último (Figura 3.7).

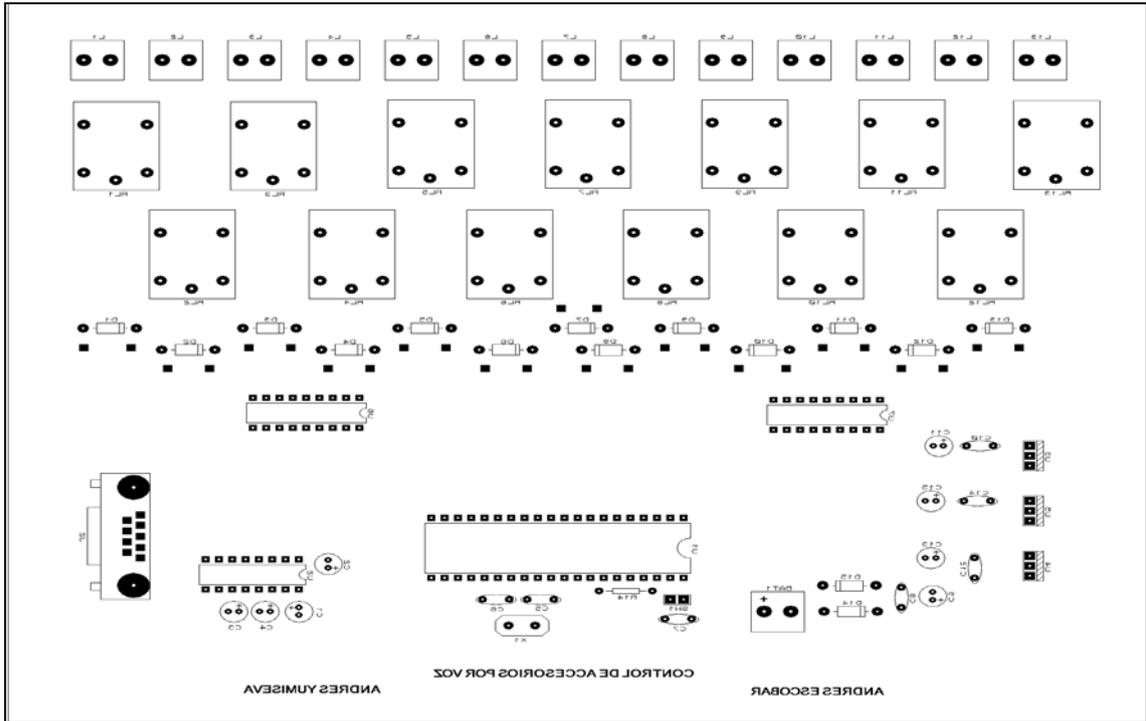


Figura 3.6. Diagrama que se imprime en el papel foto sensible (vista superior de la placa).

Autor: Andrés E. & Andrés Y.

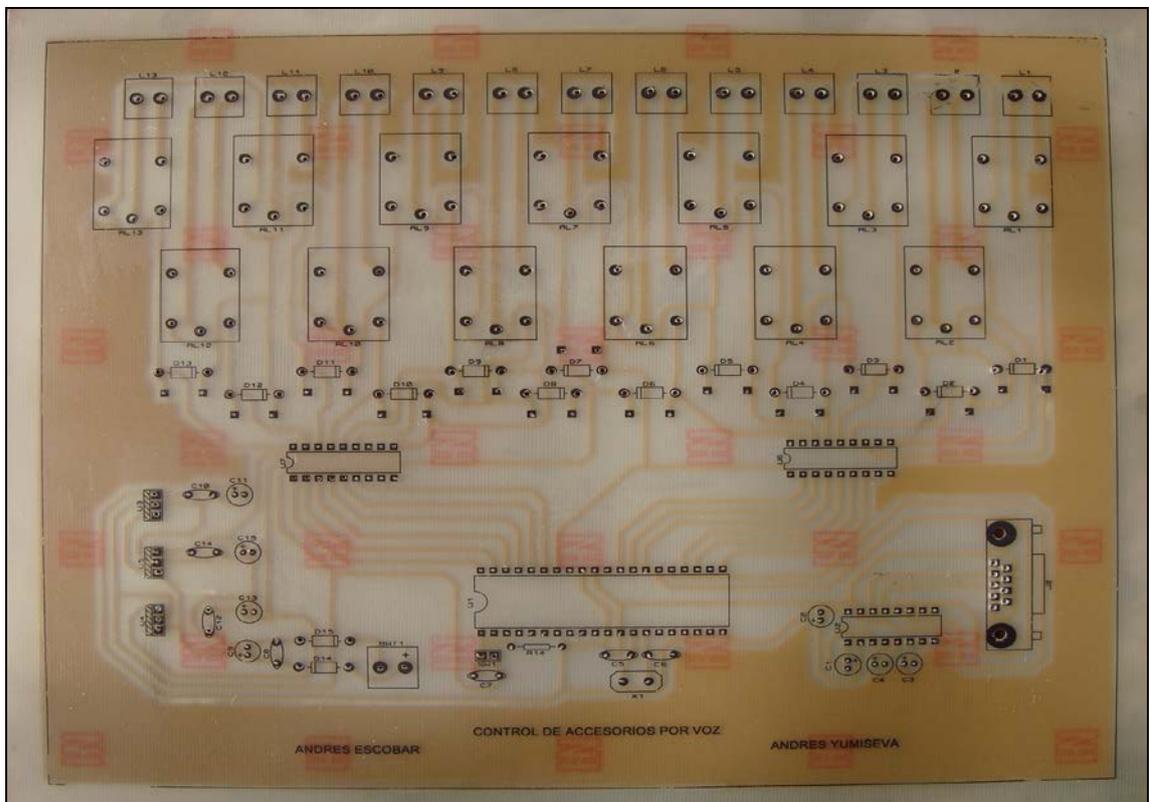


Figura 3.7. Vista superior de la placa PCB terminada (ubicación de elementos).

Autor: Andrés E. & Andrés Y.

### **3.2.2. Montaje y soldadura de elementos**

Para colocar los elementos sobre la placa, se debe realizar agujeros ayudados de un taladro con una broca de un milímetro, de arriba hacia abajo, ya que en la parte superior de la placa se indica exactamente la ubicación en donde hacer los agujeros para los pines de los elementos electrónicos.

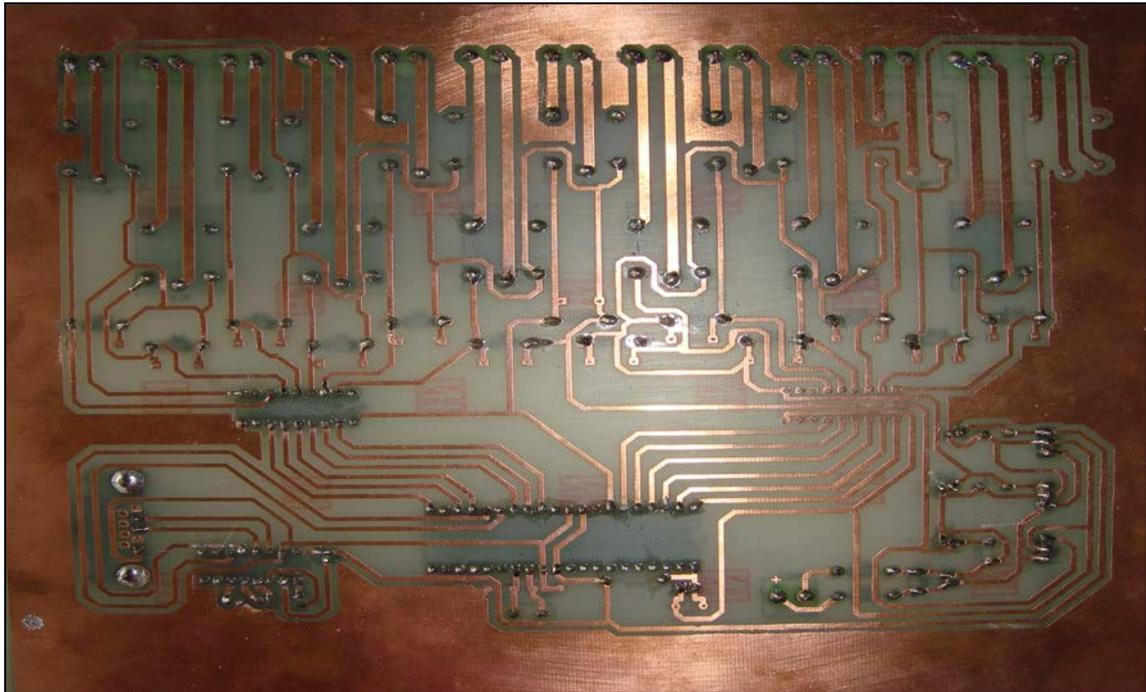
#### **3.2.2.1. Soldadura**

En una placa PCB, al soldar los elementos electrónicos que la componen, el uso del estaño es primordial (Figura 3.8), debido a que es un buen conductor para el flujo de corriente permitiendo una sujeción fiable y resistente, generando un circuito seguro y de larga vida útil (Figura 3.9), además el cautín debe estar en excelentes condiciones de funcionamiento para obtener una soldadura perfecta.



**Figura 3.8. Soldadura de los materiales eléctricos en la placa PCB.**

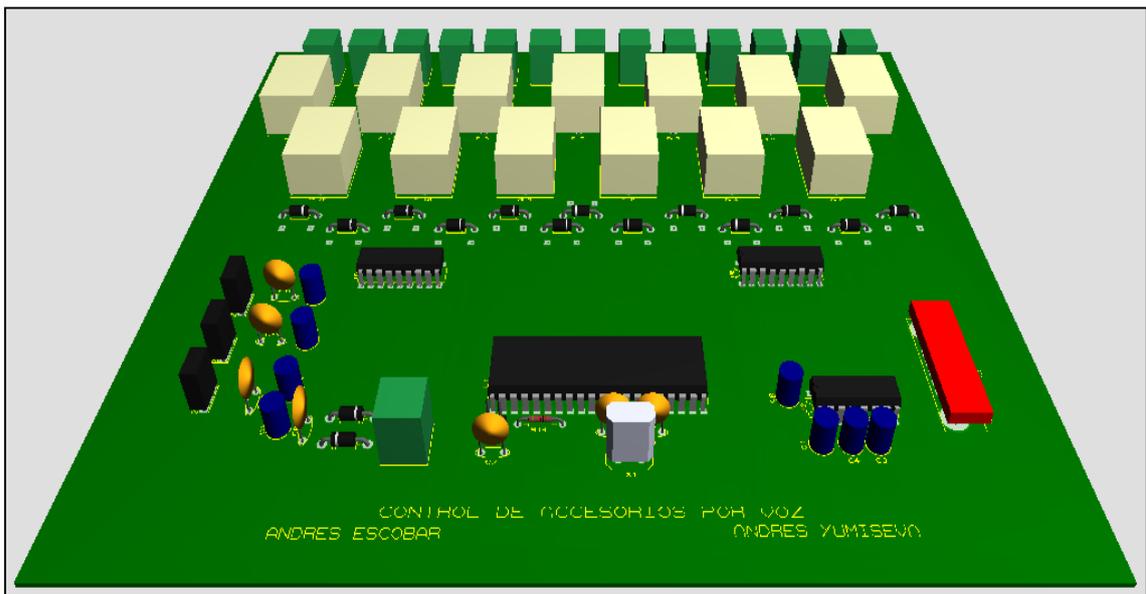
Autor: Andrés E. & Andrés Y.



**Figura 3.9. Vista de la soldadura final de los elementos.**

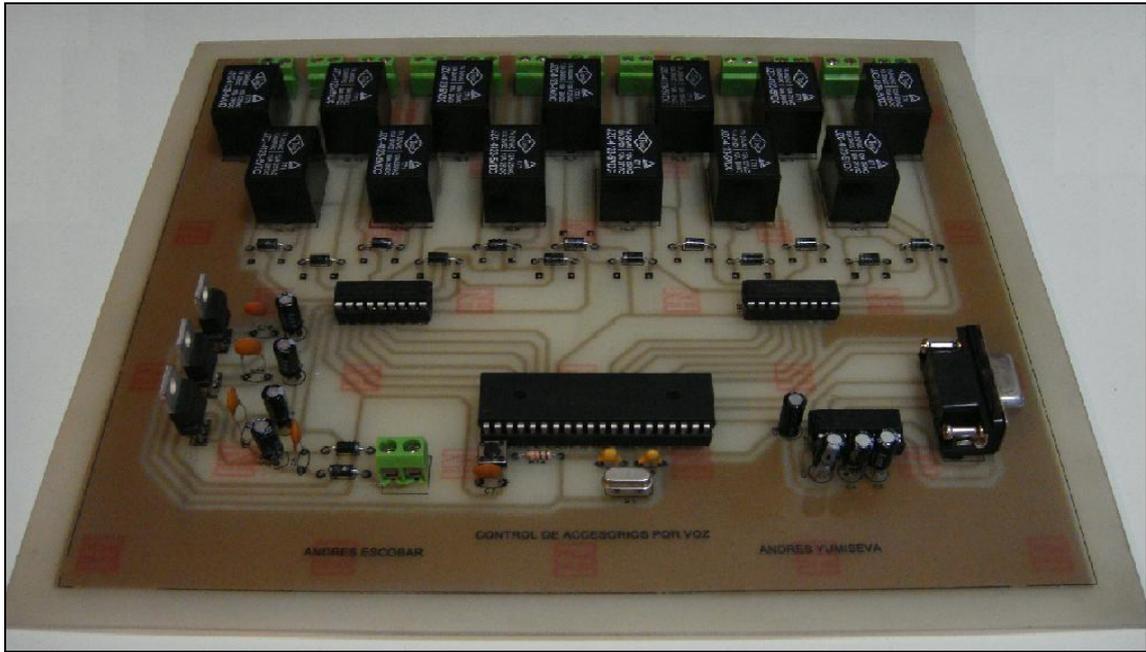
Autor: Andrés E. & Andrés Y.

La figura 3.10 muestra una vista en 3D de la placa PCB realizada en ARES, y en la figura 3.11 se puede ver la placa real y final del proyecto una vez realizadas todas las soldaduras.



**Figura 3.10. Visualización 3D placa PCB.**

Autor: Andrés E. & Andrés Y.



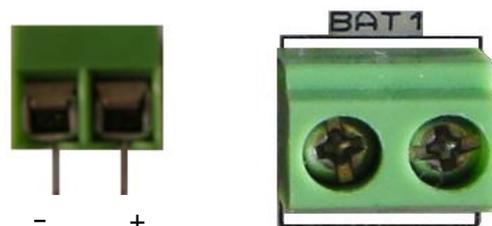
**Figura 3.11 Vista Real placa PCB terminada.**

Autor: Andrés E. & Andrés Y.

### 3.2.2.2. Elementos adicionales

#### 3.2.2.2.1. *Bornera de alimentación de 2 pines*

Constituye el punto de conexión entre la placa PCB y la fuente de alimentación con 12 voltios DC, tomados de la batería del automóvil, los cuales serán transformados por el regulador de voltaje 7805, a 5 voltios DC, este tipo de bornera de 2 pines es perfecta para este tipo de aplicaciones, ya que es de fácil instalación además de un tamaño reducido (Figura 3.12).



**Figura 3.12. Vistas frontal y superior de la bornera de 2 pines.**

Autor: Andrés E. & Andrés Y.

#### 3.2.2.2. *Borneras de salida de 2 pines*

Permiten una conexión sencilla de los accesorios a la interface de control (Figura 3.13), cada una de estas salidas está controlada por un relé que funciona con 5 voltios DC, pueden usarse con cables entre 22 y 14 AWG, la distancia entre pines es de 5mm, lo cual es ideal para el uso en circuitos impresos.

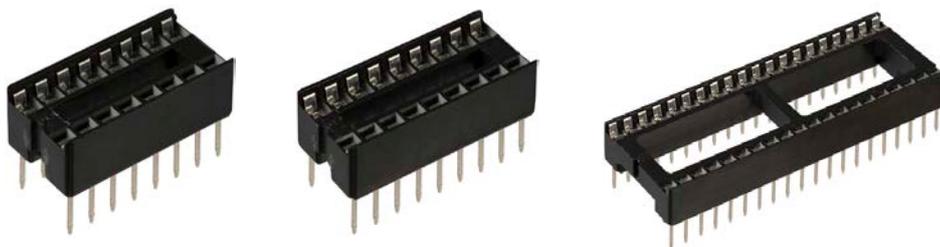


**Figura 3.13. Vista frontal del grupo de borneras de salida de la placa PCB.**

Autor: Andrés E. & Andrés Y.

#### 3.2.2.3. *Bases para Circuitos Integrados*

Este tipo de bases son de vital importancia en el proyecto, ya que ayudan para el montaje y desmontaje de los circuitos integrados en la placa de control, sin tener la necesidad de desoldarlos, en el caso de que se requiera su cambio por daño, o que sea necesario su reprogramación, esto último en el caso del PIC. La cantidad de pines o agujas puede variar, se utilizaron una base de 16 pines para el MAX232, dos bases de 18 pines para los circuitos integrados ULN2803 y una base de 40 pines para el PIC16F877A (Figura 3.14).



**Figura 3.14. Bases para circuitos integrados de 16, 18 y 40 pines respectivamente.**

Autor: Andrés E. & Andrés Y.

### **3.3. SELECCIÓN DE ELEMENTOS ELECTRÓNICOS**

Para diseñar el circuito final y el posterior desarrollo de la placa PCB, se utilizan varios componentes electrónicos, cuyo uso es justificado a continuación:

Para la comunicación entre una computadora y un PIC, es necesario el uso de un puerto serial RS-232 que permite una conexión fácil y sencilla, en éste caso para cumplir este fin se utiliza al MAX-232 que se encarga de la recepción de datos que se generan en el puerto serial RS-232 de una computadora para luego enviar éstos datos a un PIC, que posterior a la programación del mismo, logra efectuar las actividades grabadas con precisión.

La ventaja de utilizar un PIC de MICROCHIP, es su fácil programación, además se conoce los programas que permiten crear el software y grabar la información, para que realice el trabajo deseado en el circuito además existe un mayor material bibliográfico y fuentes de información variada excluyendo a los AVR de ATMEL, que poseen similares características, pero localmente su uso es menor que el PIC, debido a la poca información.

Cabe recalcar que ciertas computadoras no poseen puerto serial, incluyendo a las sofisticadas portátiles que poseen solo puertos USB, lo cual no representa un problema, debido a que en estos casos se puede utilizar un cable USB a serial, además de permitir realizar una comunicación optima en toda computadora con sistema operativo Windows de Microsoft.

En la fase de control; la cual se compone de dos ULN-2803, mismos que están compuestos por transistores pequeños, reemplazando a los transistores 2N3904 que ocupan un mayor espacio y tienen un costo mayor.

El uso del relé brinda una fácil conectividad con los ULN-2803, logrando funcionar como un electroimán que cierra el circuito de cualquier elemento conectado a sus extremos al recibir una señal proveniente del PIC, la cual se transforma en el ULN-2803, logrando cumplir la función deseada, que es poder activar accesorios del vehículo en un tiempo mínimo.

La alimentación se toma de la batería del auto, pero para alimentar a los componentes del circuito, se utiliza tres reguladores de voltaje 7805, convirtiendo la corriente de 12 voltios DC a 5 voltios DC, que alimentan a tres bloques comprendidos por:

- El primero, alimenta al PIC y al MAX-232;
- El Segundo bloque, alimenta a un ULN-2803 y a 7 Relés;
- El tercer bloque alimenta al segundo ULN-2803 y 6 Relés.

Los capacitores y diodos se utilizan para protección y buen funcionamiento de los dispositivos electrónicos en éste caso; para evitar que ocurran interferencias electromagnéticas y protección de los ULN-2803 (transistores) respectivamente.

También puede haber errores como utilizar elementos electrónicos con daños o fallas que pueden afectar el funcionamiento general de la placa. Tales como utilizar relés en mal estado o ya usados, e incluso utilizar un foco de doce voltios que sin revisarlo puede estar conectado internamente haciendo cortocircuito logrando así reiniciar el PIC, interrumpiendo las funciones que se estaban ejecutando.

En la selección de elementos, ciertos componentes electrónicos permiten optimizar y mejorar el diseño de la placa ahorrando tiempo y dinero.

Por ejemplo están los ULN-2803 que reemplazan a los transistores 2N3904, y aparte los PIC de MICROCHIP, que permiten realizar una programación ordenada y fiable frente a los AVR de ATMEL, que por supuesto se tiene poco conocimiento de sus especificaciones, debido a que prácticamente son menormente usados, porque no existe mucha información acerca de los mismos e incluso su quemador difiere al de los PIC.

### **3.4. SELECCIÓN DEL SOFTWARE**

El uso de VISUAL BASIC 6.0 y VISUAL STUDIO 2010, permite desarrollar ejecutables, que logran una óptima comunicación entre el computador y el circuito, con lo que se desarrollo un software de fácil uso para el usuario final, cabe recalcar que el uso de este software requiere una instalación personalizada en cada computadora. Incluso la ventaja de estos programas, es que sus ejecutables son reconocidos con facilidad por el sistema operativo de Windows, con una instalación previa de Net Framework, que es una plataforma que reconoce al ejecutable y permite su funcionamiento en el computador.

Los ejecutables diseñados en visual 6.0 controlan la comunicación de Visual Basic 6.0 con el PIC y el ejecutable principal que conforma la interfaz gráfica se diseña en Visual Studio 2010 y está compuesto por varios botones de selección que nombran a los accesorios intervenidos, los cuales son reconocidos con facilidad por el reconocimiento de voz de Windows y Dragon Naturally Speaking 10.

El uso de programas como MicroCode Studio (software con que se programa el PIC), permite generar el software del PIC, de manera rápida e incluso tiene un sistema de advertencia que ante cualquier error en la escritura de la programación del PIC, señala que debemos corregirlo, evitando una errónea compilación del mismo.

El PicKit2, es un software que se conecta al quemador de PIC, el cual tiene una opción en donde se carga el programa del PIC en lenguaje hexadecimal (.HEX), y se procede a quemar el PIC que se va a utilizar en el circuito, también ofrece opciones como la de chequeo de información interna e incluso la opción de borrado del PIC.

El reconocimiento de voz de Windows y Dragon Naturally Speaking 10, con una capacidad rápida para reconocer la voz del usuario, permite aplicar este nuevo tipo de tecnología en el automóvil, brindando un soporte a la conducción del mismo con la activación de accesorios mediante comandos de voz e incluso con ventajas de aplicación de este sistema varias áreas incluyendo a los sectores vulnerables de la sociedad como gente con discapacidad.

El software para realizar los planos eléctricos y la placa PCB, es PROTEUS; que se subdivide en ISIS y ARES respectivamente. ISIS, permite generar un plano eléctrico de fácil visualización y en ARES se realiza la placa PCB, en este último se logra modificar la posición de los elementos electrónicos a conveniencia del diseñador, y así una mejor ubicación de los mismos, diseñándola para que cumpla sus funciones correctamente.

## **CAPÍTULO 4**

### **DISEÑO DEL SOFTWARE**

Para el desarrollo del presente proyecto son necesarias cuatro herramientas de software, la primera es el programa Dragon Naturally Speaking 10 ó el Reconocimiento de Voz de Windows, los mismo que se encargan de realizar el reconocimiento de voz emitido por el usuario a través del micrófono y transformarlo en texto que a su vez es interpretado por una aplicación creada utilizando Visual Basic 6.0 y Visual Basic 2010, las cuales son la segunda herramienta, estas ultimas que realizan las acciones de control que con anterioridad han sido definidas; la tercera herramienta es MicroCode Studio que permite realizar la programación para el PIC, y finalmente la cuarta herramienta con la que se graba la información realizada en MicroCode Studio en el PIC.

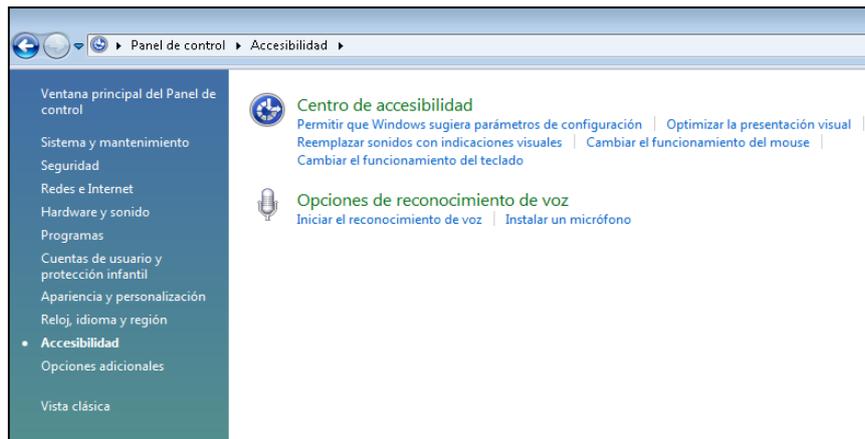
#### **4.1. TRABAJO DESARROLLADO CON EL RECONOCIMIENTO DE VOZ DE WINDOWS Y DRAGON NATURALLY SPEAKING 10.**

##### **4.1.1. Proceso de activación (Reconocimiento de voz de Windows)**

Para activar el reconocimiento de voz perteneciente a Windows, se debe hacerlo con ciertos pasos sencillos, que permiten controlar aplicaciones pertenecientes a Windows, como el reconocimiento de comandos de voz y dictado.

A continuación se describirán cada uno de los pasos que se deben seguir para activar éste software.

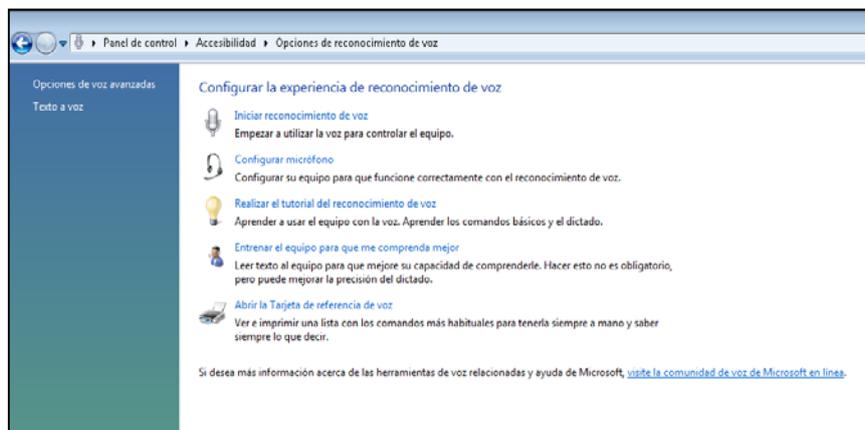
- Para activar el reconocimiento de voz, se ingresa a Panel de control y se hace doble click en Accesibilidad.
- En accesibilidad aparecen dos opciones: Centro de accesibilidad y Opciones de Reconocimiento de voz, se elige éste último. (Figura 4.1)



**Figura 4.1. Centro de accesibilidad de Windows**

Autor: Andrés E. & Andrés Y.

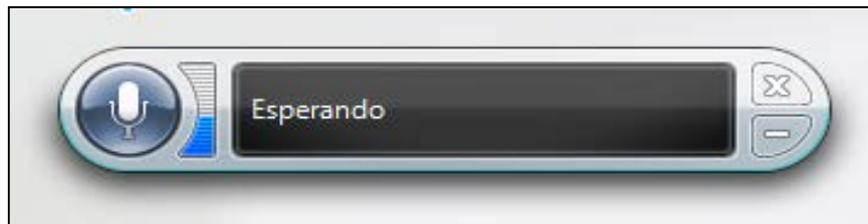
- Se visualizan diferentes opciones para instalar y configurar el reconocimiento de voz e incluso contiene un tutorial que permite mejorar el desempeño de este software de Windows (Figura 4.2).



**Figura 4.2. Opciones de reconocimiento de voz**

Autor: Andrés E. & Andrés Y.

- Aparece una interfaz gráfica del reconocimiento de voz de Windows (Figura 4.3), en donde se dictan los comandos de voz para realizar diversas funciones. (El programa funciona únicamente si se conecta el micrófono a la computadora, para captar la señal de voz proveniente del operador)



**Figura 4.3. Interfaz gráfica de reconocimiento de voz Windows**

Autor: Andrés E. & Andrés Y.

#### **4.1.2. Proceso de instalación de Dragon Naturally Speaking 10**

Este programa fue creado y se utiliza con el fin de reconocer los vocablos emitidos por el usuario, el entorno de trabajo es múltiple, se puede realizar dictado de documentos, comandar la computadora como por ejemplo, abriendo y cerrando ventanas de programas, ejecutar aplicaciones a través de rutas de acceso a las mismas; es así como Dragon Naturally Speaking en su versión en español, permite reconocer una gran cantidad de palabras con un entrenamiento básico, pero para una mayor precisión se puede realizar un entrenamiento del software más afinado según los requerimientos del usuario.

Para esta aplicación y su óptimo desempeño es necesario un *entrenamiento especial*, lo cual permite que el usuario realice una adaptación de su voz más dedicada y precisa con el software. A continuación se describen cada uno de los pasos que se deben seguir previo uso del software.

- Instalar Dragon Naturally Speaking 10, al insertar el disco de instalación automáticamente se ejecuta el programa que le guía en la instalación del mismo, se selecciona el tipo de instalación, la que se elige es la *Típica/Completa*, misma que puede durar alrededor de 45 minutos. (Figura 4.4).



**Figura 4.4. Tipo de instalación; Seleccionada Típica/Completa**

Autor: Andrés E. & Andrés Y.

- Una vez instalado el software en el computador, se escribe el nombre de usuario, se elige el idioma de su preferencia, también se escoge el tipo de entrenamiento; una vez hecho esto se selecciona con un click siguiente. (Figura 4.5).



**Figura 4.5. Asistente para crear usuarios.**

Autor: Andrés E. & Andrés Y.

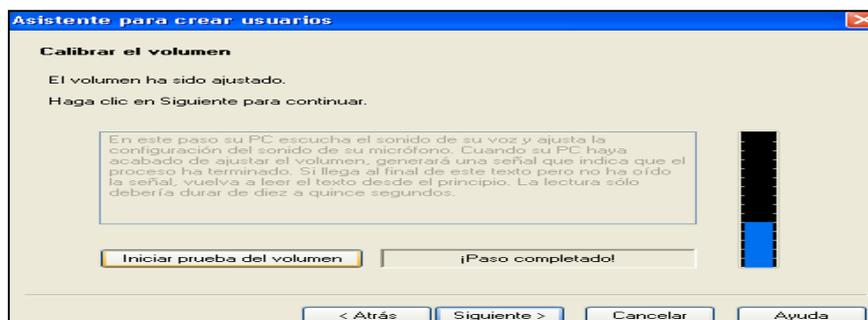
- La figura 4.6. muestra que debe conectar el micrófono en la PC. Una vez hecho esto dar click en siguiente.



**Figura 4.6. Indicación de que debe conectar el micrófono.**

Autor: Andrés E. & Andrés Y.

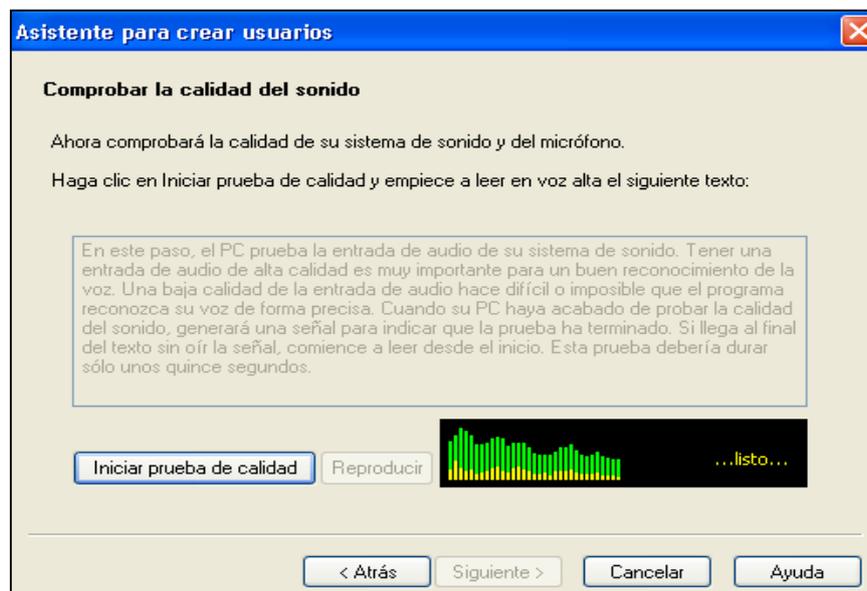
- En este paso se calibra el volumen, para ello se hace click en el botón *Iniciar prueba de volumen*, y leer el texto que se encuentra en el recuadro central de la ventana (Figura 4.7); la PC escucha el sonido de la voz y ajusta la configuración de sonido del micrófono. Cuando la PC acaba de ajustar el volumen, genera una señal que indica que el proceso ha terminado. Si se llega al final de este texto pero no se ha oído la señal, se vuelve a leer el texto desde el principio. La lectura sólo dura de diez a quince segundos. Pulsar en siguiente.



**Figura 4.7. Calibración del volumen del micrófono.**

Autor: Andrés E. & Andrés Y.

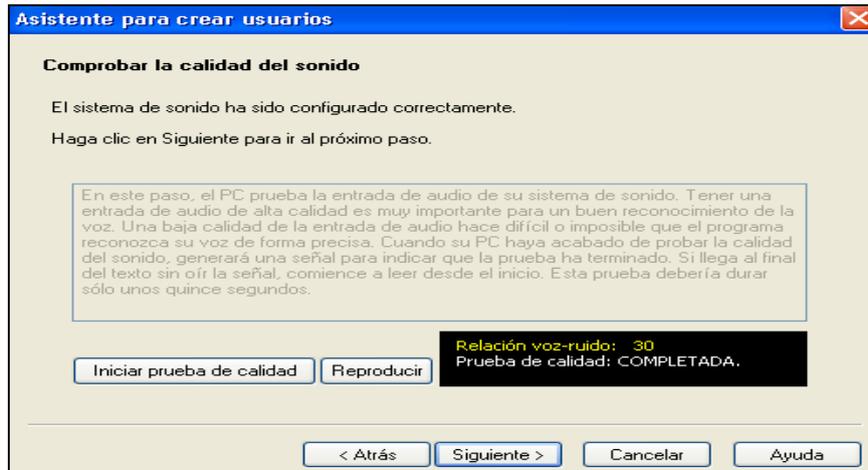
- Al pulsar en siguiente el asistente muestra una ventana en la que se comprueba la calidad del sonido, se da click en *Iniciar prueba de calidad* y se lee el texto que se encuentra en el recuadro central de la ventana (Figura 4.8). En este paso, el PC prueba la entrada de audio del sistema de sonido. Tener una entrada de audio de alta calidad es muy importante para un buen reconocimiento de la voz. Una baja calidad de la entrada de audio hace difícil o imposible que el programa reconozca la voz del usuario de forma precisa.



**Figura 4.8. Comprobación de la calidad de sonido.**

Autor: Andrés E. & Andrés Y.

- Cuando su PC acaba de probar la calidad del sonido, genera una señal para indicar que la prueba ha terminado y un mensaje indicando *Prueba de calidad completada* (Figura 4.9). Si se llega al final del texto sin oír la señal, se comienza a leer desde el inicio. Esta prueba dura sólo unos quince segundos; después se hace click en siguiente.



**Figura 4.9. Prueba de calidad de sonido completada.**

Autor: Andrés E. & Andrés Y.

- El asistente muestra inmediatamente una ventana para iniciar la etapa para entrenar el reconocimiento de la voz del usuario en el equipo. (Figura 4.10).



**Figura 4.10. Asistente para entrenamiento de su voz.**

Autor: Andrés E. & Andrés Y.

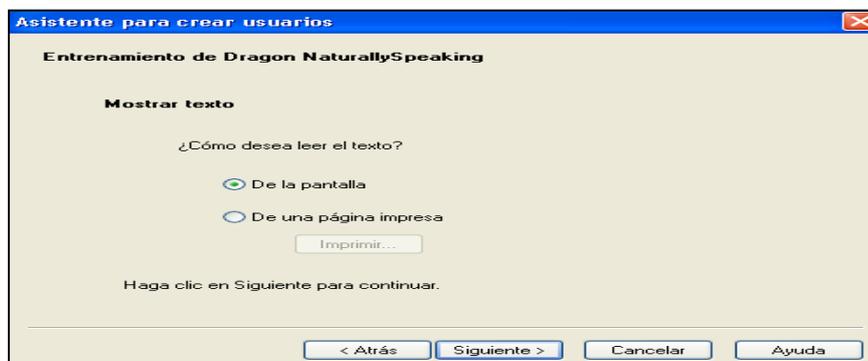
- Se escoge uno de los cuatro textos disponibles para su lectura (Figura 4.11), luego de ello seleccionar el botón de *siguiente*.



**Figura 4.11. Elección de texto para lectura.**

Autor: Andrés E. & Andrés Y.

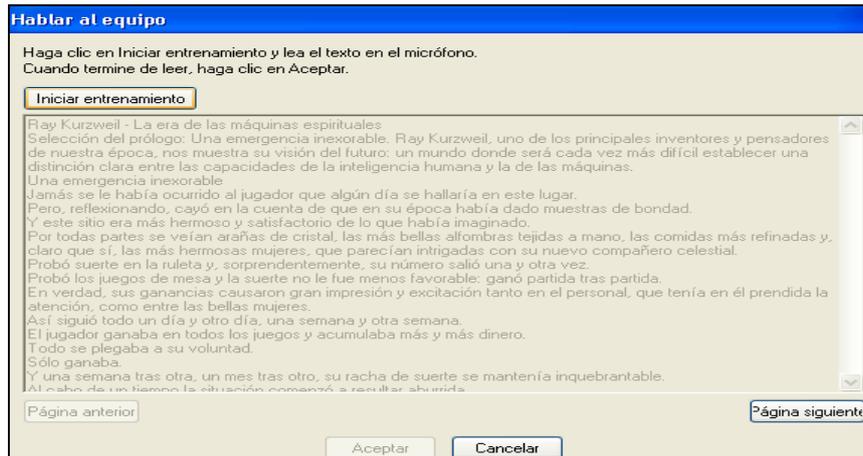
- A continuación en una ventana se tienen dos opciones para la lectura del texto (Figura 4.12), la primera es leerlo directamente de la pantalla de la PC; la segunda es leer el texto previamente seleccionado de una página impresa, si se selecciona esta última se imprime el texto de lectura. Hacer click en *siguiete*.



**Figura 4.12. Selección para leer el texto.**

Autor: Andrés E. & Andrés Y.

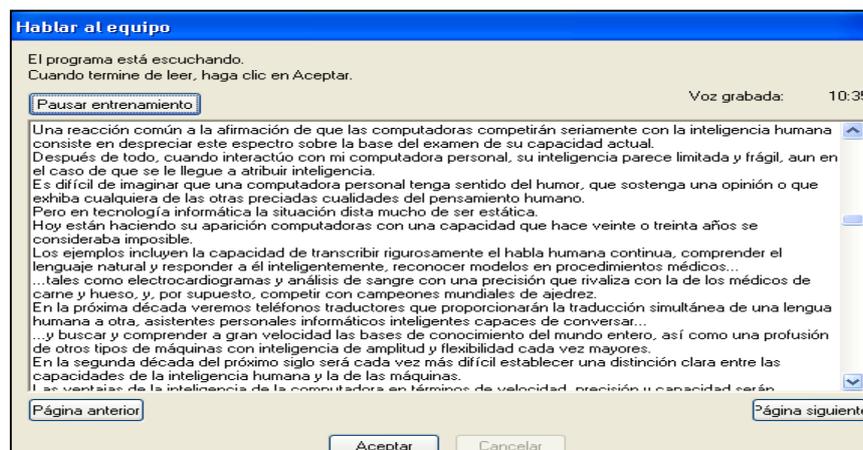
- Inmediatamente surge una ventana en la cual se debe leer el texto del recuadro, para empezar la prueba se elige el botón *iniciar entrenamiento* (Figura 4.13), esta etapa le permite al programa crear archivos con la voz del usuario, los cuales sirven de referencia para el reconocimiento de voz.



**Figura 4.13. Ventana para iniciar entrenamiento de voz.**

Autor: Andrés E. & Andrés Y.

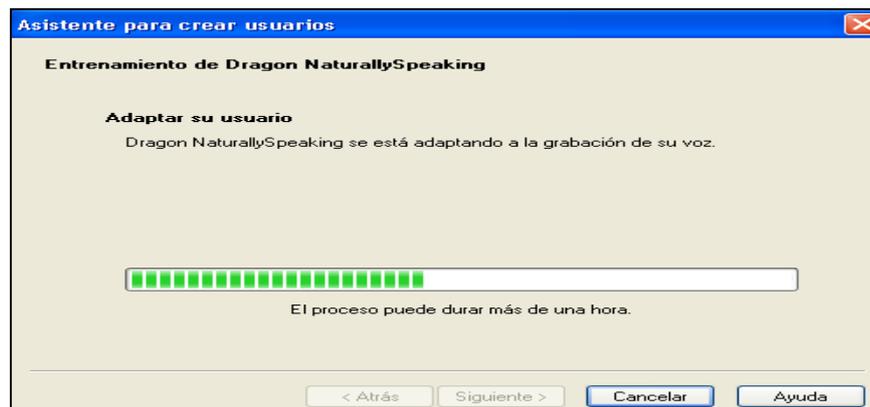
- Al iniciar el entrenamiento, el programa empieza a escuchar lo que el usuario está leyendo, una vez que se haya grabado archivos de voz suficientes el programa activará el botón de *Aceptar* que se encuentra en la parte inferior de la ventana, el cual se seleccionará para poder continuar (Figura 4.14); Es importante recordar que usted mientras lee el texto puede *pausar el entrenamiento* si lo desea. Este paso puede durar entre 10 a 15 minutos.



**Figura 4.14. Programa se encuentra Grabando Su Voz.**

Autor: Andrés E. & Andrés Y.

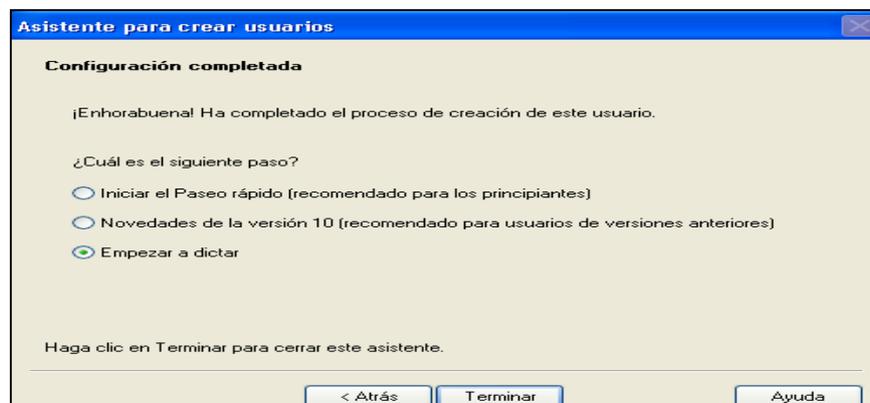
- La próxima etapa es la adaptación de Dragon Naturally Speaking a la grabación de la voz grabada del usuario, surge una ventana en la que se selecciona el botón *Iniciar adaptación*, inmediatamente el proceso de adaptación comienza (Figura 4.15), el cual dura aproximadamente unos 10 minutos.



**Figura 4.15. Adaptación del Programa a la grabación de voz del usuario.**

Autor: Andrés E. & Andrés Y.

- Una vez terminado el proceso de creación del nuevo usuario se muestra una ventana, indicándole que la configuración ha concluido (Figura 4.16), debe seleccionar *Empezar a dictar* y hacer click en *Terminar*.



**Figura 4.16. Configuración completada.**

Autor: Andrés E. & Andrés Y.

- El programa realiza el entrenamiento para el reconocimiento de voz, se abre la barra de herramientas de Dragon Naturally Speaking (Figura 4.17), en ella resalta el micrófono de color verde, lo cual indica que el micrófono está encendido y listo para empezar las acciones de reconocimiento.



**Figura 4.17. Barra de Herramientas DragonBar.**

Autor: Andrés E. & Andrés Y.

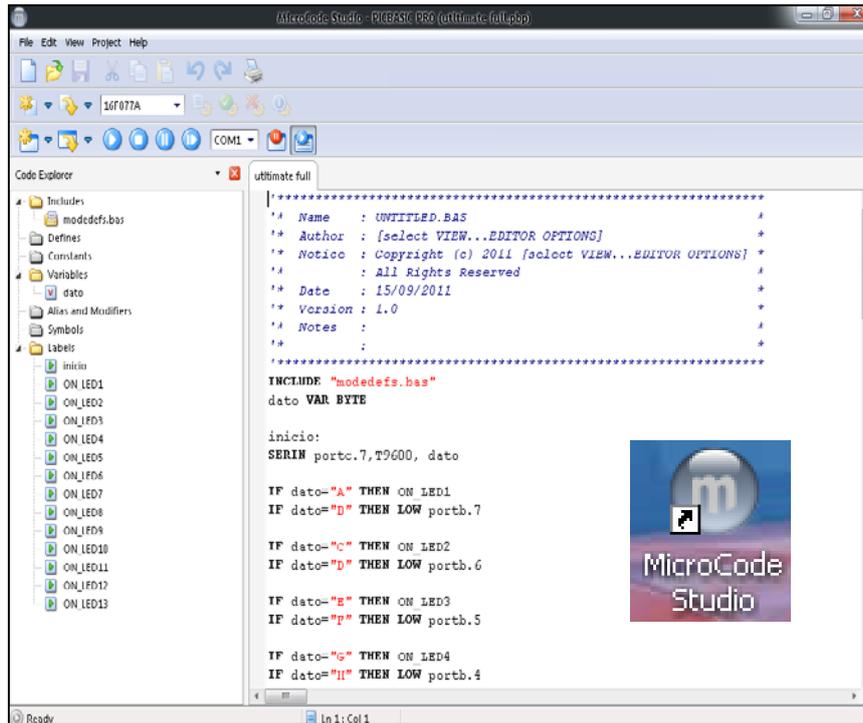
## 4.2.MICROCODE STUDIO

Es un entorno de desarrollo integrado, utilizado para la programación de microcontroladores.

El uso de este software es muy sencillo (Figura 4.18), debido a que permite escoger el modelo de microcontrolador, se puede diseñar el programa e incluso informa errores cometidos en la programación impidiendo la compilación del archivo creado.

Después de programar correctamente (Ver anexo 1. Programación del PIC), se compila el archivo, mediante las opciones que posee el programa, además que informa al usuario si el programa ocupa un espacio mayor a que el microcontrolador posee.

La compilación se trata de un sistema que permite guardar el archivo de programa en varias extensiones como mac, asm, y hex. Siendo hex la más importante porque son instrucciones en lenguaje hexadecimal, la cual es selecciona por el usuario para grabar la información en el microcontrolador.



**Figura 4.18. MicroCode Studio.**

Autor: Andrés E. & Andrés Y.

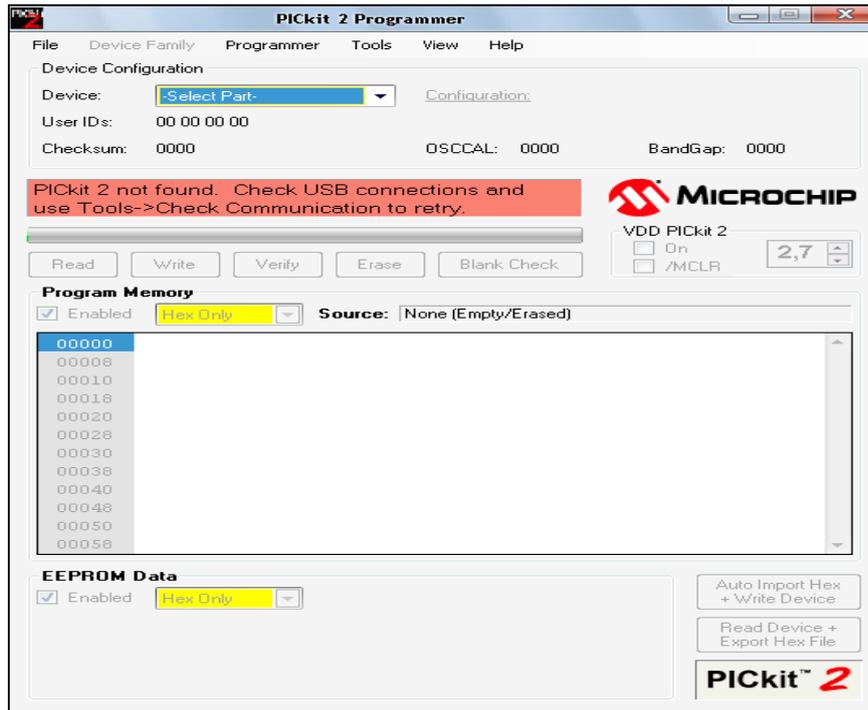
### 4.3. PIC - kit 2

Es una herramienta que permite grabar la información de un archivo compilado, en el microcontrolador especialmente si es de la familia Microchip. (Figura 4.19).

Funciona con hardware que posee un grabador de microcontroladores, en donde se coloca el chip que se desea programar.

El programa se encarga de la lectura del mismo, importando el archivo de extensión (.hex) que se hizo en MicroCode Studio, también permite borrar la programación anterior si fue utilizado previamente.

El programa tiene una opción en la que se bloquea el microcontrolador para no ser reprogramado e incluso ser copiada su información.



**Figura 4.19. PICkit 2.**

Autor: Andrés E. & Andrés Y.

#### **4.4. VISUAL BASIC VERSIÓN 6.0 Y VERSIÓN 2010**

Es un lenguaje de programación por sucesos o eventos, en donde se pueden desarrollar aplicaciones complejas con gran facilidad y en poco tiempo. La ventaja de las aplicaciones creadas, es que pueden ser utilizadas en Windows en lenguaje basic y programación orientada a objetos, dependiendo de las versiones, porque el sistema operativo y otras aplicaciones siempre se renuevan.

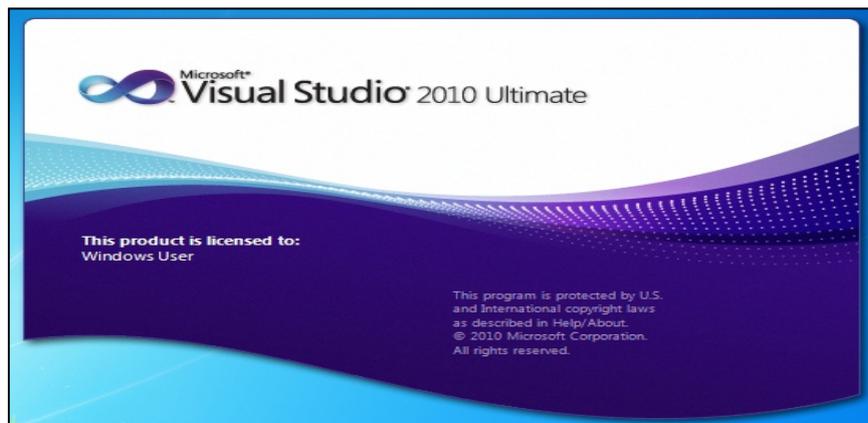
Visual, como su nombre lo indica, se refiere a una Interfaz Gráfica de Usuario, con lo que se evita realizar una programación complicada y permite la utilización de elementos prediseñados como ventanas, botones, barras de desplazamiento, etc., que pueden aplicarse en una hoja de diseño, al finalizar el usuario puede activar el modo de pre visualización del proyecto y realizar los cambios necesarios.

Para este proyecto se utilizan las versiones de Visual Basic 6.0 (Figura 4.20) y Visual 2010 (Figura 4.21).



**Figura 4.20. Visual Basic 6.0**

Autor: Andrés E. & Andrés Y.



**Figura 4.21. Visual 2010**

Autor: Andrés E. & Andrés Y.

En el proceso de diseño de la interfaz gráfica, se realizaron diferentes cambios y adecuaciones para obtener los resultados deseados.

El software (interno), fue creado mediante Visual Basic 6.0, con programas ejecutables que permiten realizar el control de acciones de comunicación entre computadora e interface de control (Placa PCB), a través del envío y recepción de

variables, que son en total 26. El diseño de estas variables está ligado a la programación (Ver anexo 2) ó control de acciones de la placa PCB.

Visual Basic 2010, permite crear el diseño de la interfaz gráfica que consta de 3 hojas de diseño las cuales según la programación (Ver anexo 3), se alternan para generar un programa seguro y accesible. Las hojas de diseño constan de lo siguiente:

- La primera hoja de diseño consta de un control de acceso con clave de seguridad, en caso de no ser correcta, el usuario no podrá acceder al programa.
- La segunda hoja de diseño, está compuesta por una barra de progreso que se carga para observar la tercera y última hoja de diseño.
- La tercera hoja de diseño está compuesta por 26 botones; éstos se dividen en dos bloques; 13 variables que permiten activar los accesorios intervenidos en el vehículo y 13 variables que permiten desactivar a los mismos.

La ventaja de Visual Basic 2010, tiene un sistema de corrección que evita generar ejecutables, si existen errores de programación.

#### **4.4.1. Aplicaciones**

Las aplicaciones principales de visual basic; son las de modo de diseño y modo de ejecución.

En el modo de diseño se realiza todo lo necesario para que el programa esté listo e incluso se pueden realizar modificaciones para mejorarlo.

En el modo de ejecución el programa funciona en cualquier computadora por lo que debe tener instalado en este caso para Visual Basic 6.0 y Visual Basic 2010, necesitará Microsoft .NET.Framework 2 y Microsoft .NET.Framework 4 correspondientemente. El programa ejecutable no puede ser modificado por los usuarios finales.

#### **4.4.2. Controles**

Cada uno de los elementos que se encuentran en Visual Basic, están como opciones disponibles como la caja de herramientas en las que se pueden escoger botones, barras de desplazamiento horizontales, gráficos, etc., que al final representan un programa diseñado directamente de Windows.

Cabe recalcar que para diseñar el programa cada opción de la caja de herramientas e incluso el formulario, tienen nombres y realizarán lo que el usuario desee dependiendo de las instrucciones generadas.

#### **4.4.3. Formularios / Hojas de diseño**

En Visual Basic, el formulario es una ventana, considerado como una fuente en donde se encuentran todos los elementos escogidos de la caja de herramientas por parte del diseñador de la aplicación.

Incluso un formulario puede estar diseñado para abrir varios formularios dependiendo de la aplicación.

#### **4.4.4. Objetos y propiedades**

Es el conjunto de datos y características que los elementos de la caja de herramientas de Visual Basic, que poseen ciertas funciones como tamaño, color, fuente, imagen dependiendo del diseño del programador. Las propiedades difieren según los objetos escogidos, en donde algunos tienen valores lógicos como True-False, valores numéricos propios del objeto que genera una aplicación más avanzada de los demás objetos.

#### **4.4.5. Eventos y métodos**

Es un efecto de acción-reacción, en donde se generan aplicaciones consecutivas del programa, que generalmente son respuestas a las peticiones que el usuario indica mediante el uso del mouse del computador, en el programa diseñado.

Los métodos están adheridos a cada función de la caja de herramientas y extras, permiten visualizar formatos preestablecidos por el programa para ejecutar una acción dependiendo de la programación.

Para pre-visualizar el proyecto antes de realizar el ejecutable, Visual Basic ofrece opciones para hacer esta función

#### **4.4.6. Proyecto**

El proyecto es el resultado final de la programación. Posteriormente puede ser modificado por el programador para obtener un resultado final óptimo. Cuando haya concluido este proceso de cambios finales, se extrae el archivo ejecutable, que funciona en sistemas operativos Windows, incluso sin instalar el programa creador Visual Basic.

## **4.5. MICROSOFT NET FRAMEWORK**

Es una plataforma de Windows, que permite a los usuarios y a todas las computadoras estandarizar el uso de aplicaciones creadas a través de programación en Visual Basic y otros lenguajes, permite tener interoperabilidad entre varios lenguajes y ofrecer servicios como creación de programas, páginas web, etc.

Se lo puede descargar de la página web de Windows, y para el óptimo funcionamiento de ejecutables de Visual Basic 6.0 y 2010, se descarga e instala Microsoft NET Framework 2 y Microsoft NET Framework 4 en el computador.

## **4.6. PROTEUS**

Software que permite la simulación de circuitos electrónicos con microcontroladores y por su facilidad de uso, se ha convertido como el más popular simulador software para microcontroladores PIC.

Es una herramienta de alto desempeño con unas capacidades graficas impresionantes.

### **4.6.1. ISIS (Generación plano eléctrico)**

El Programa ISIS, Intelligent Schematic Input System (*Sistema de Enrutado de Esquemas Inteligente*), permite diseñar el plano eléctrico del circuito que se desea realizar con componentes variados, como resistencias, microcontroladores, fuentes de señales y alimentación, etc.

Después de haber finalizado el circuito, Isis tiene un módulo de control que permite visualizar el funcionamiento del circuito en tiempo real e incluso hace observaciones si existe algún error en el diseño.

#### **4.6.2. ARES (Generación circuito PCB)**

ARES, o Advanced Routing and Editing Software (*Software de Edición y Ruteo Avanzado*); es la herramienta de enrutado, ubicación y edición de componentes, se utiliza para la fabricación de placas de circuito impreso, permitiendo editar generalmente, las capas superficial (Top Copper), y de soldadura (Bottom Copper).

## CAPÍTULO 5

### IMPLEMENTACIÓN DEL SISTEMA, PRUEBAS Y RESULTADOS

#### 5.1. IMPLEMENTACIÓN DEL SISTEMA EN EL VEHÍCULO

El sistema de control de accesorios por voz ha sido instalado en un vehículo

Nissan Pathfinder, para ello se realizaron los siguientes pasos:

- Primeramente se construye una caja protectora en donde irá la placa PCB, lo que permite evitar que cada uno de los elementos en la misma sufran golpes, además de dar un mejor aspecto visual; las características de la misma son: el largo es 27cm, de ancho tiene 21cm y su altura es 3.5cm. (Figura 5.1).



**Figura 5.1. Caja protectora PCB**

Autor: Andrés E. & Andrés Y.

- Se procede a la selección del lugar en donde irá el sistema, en este caso se lo ha ubicado en la parte inferior del asiento del piloto, es decir en el lado izquierdo, ya que en ese lugar es el más cercano para la ubicación de todo el cableado y se dispone del espacio necesario. (Figura 5.2).

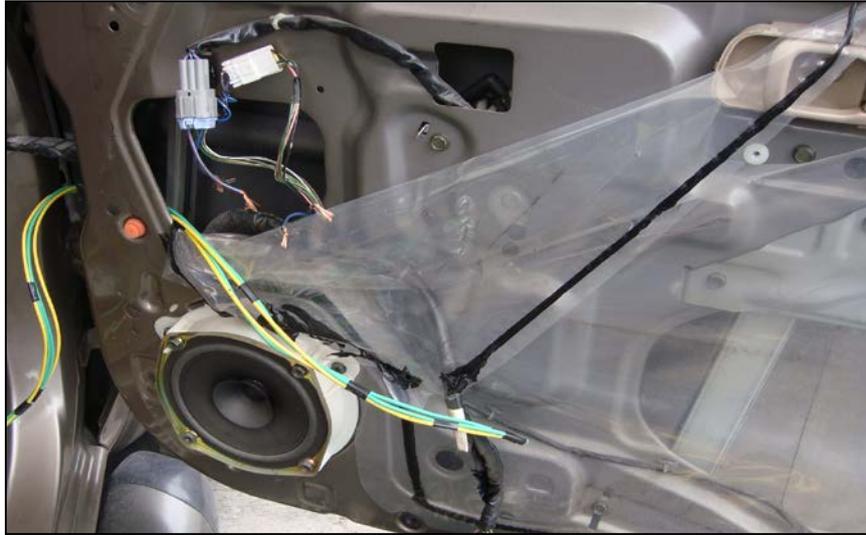


**Figura 5.2. Ubicación Placa PCB.**

Autor: Andrés E. & Andrés Y.

- Los accesorios que se controlan en este vehículo son las ventanas delanteras, Seguros, Luces bajas, Luces medias, Luces Altas, Luces de Salón frontales, Luces direccionales, y Luces intermitentes, para ello fue necesario el desmontaje de los paneles de las puertas derecha e izquierda (Figura 5.3), así como el desarme de la consola del volante (Figura 5.4).

Así como también se levanta la tapicería del piso y techo para pasar el cableado hacia debajo del asiento.



**Figura 5.3. Desmontaje Paneles Puertas.**

Autor: Andrés E. & Andrés Y.

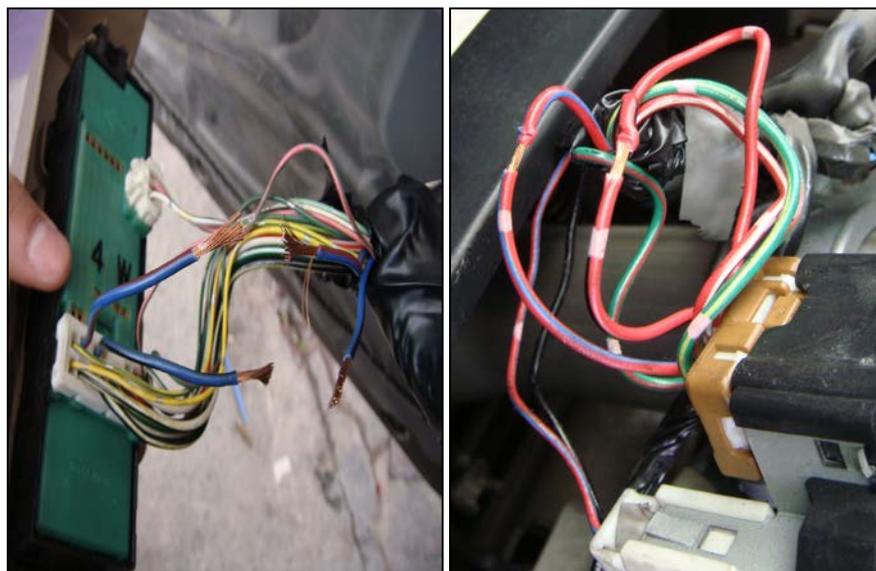


**Figura 5.4. Desmontaje consola volante.**

Autor: Andrés E. & Andrés Y.

- Se realizan las conexiones de los accesorios (Ver Anexo 7) (Figura 5.5), para que los accesorios anteriormente mencionados se puedan controlar de manera manual si el usuario lo desea, como también a través de comandos de voz; para detectar cada uno de los cables con los cuales funcionan los accesorios es necesario la ayuda de una lámpara de prueba,

la cual es muy práctica, misma que se enciende al pinchar los cables correctos que alimentan al accesorio (Figura 5.6), así como también se toma de base los diagramas eléctricos del vehículo (Ver Anexo 8).



**Figura 5.5. Cableado de ventanas y seguros; Cableado de luces bajas, medias, altas y direccionales.**

Autor: Andrés E. & Andrés Y.



**Figura 5.6. Ejemplo utilización lámpara de prueba para identificar cables Luces direccionales**

Autor: Andrés E. & Andrés Y.

- El cableado de todos los accesorios se lo llevo a un solo lugar que es debajo del asiento del piloto (Figura 5.7), el motivo de esta decisión es porque existe un espacio preciso para el alojamiento de la caja PCB, a la cual se conecta el cableado para el control de los accesorios, además de ser un lugar óptimo, ya que evita que la caja PCB este expuesta a golpes, y por esta razón sufra daños; otra característica importante de este lugar, es la cercanía y comodidad para la conexión del cable USB-Serial al computador personal del usuario, mismo que a su vez es conectado un micrófono, permitiendo que el sistema se localice muy próximo tanto al conductor como al pasajero, que son principalmente las personas que utilizan el sistema.



**Figura 5.7. Cableado de los accesorios.**

Autor: Andrés E. & Andrés Y.

- A continuación se realizaron las conexiones entre el cableado de los accesorios a la placa PCB (Figura 5.8). La placa PCB posee 13 borneras de salida de dos pines, cada una de ellas va conectada a un relé que se

encuentra normalmente abierto; a estas borneras se les conecta una de las líneas de alimentación del accesorio, ya sea positivo o negativo, actuando de la misma manera que un interruptor. (Ver Anexo 6)



**Figura 5.8. Conexiones Accesorios - Placa PCB.**

Autor: Andrés E. & Andrés Y.

## **5.2. PRUEBAS**

Para determinar la vulnerabilidad del proyecto frente al ruido y la duración de carga de batería del computador utilizado se realizaron 2 tipos de pruebas, el primero implementado para determinar la precisión que posee el software de reconocimiento de voz y como se ve afectado este, cuando el ruido ambiental en la cabina del vehículo aumenta, en esta prueba se incrementa el volumen del reproductor de sonido del vehículo hasta alcanzar el valor máximo posible; la segunda prueba se ha implementado con el fin de determinar la autonomía de carga de batería del computador, y por ende el tiempo del que se dispone para el uso del sistema de reconocimiento de voz sin recargar la batería del computador.

### 5.2.1. Prueba de precisión de reconocimiento de voz ante ruido ambiental

El porcentaje de precisión asignado al reconocimiento inicial del software corresponde a un 100%, esto ocurre cuando todas las acciones de control (Figura 5.9) emitidas por el usuario son totalmente reconocidas y ejecutadas correctamente, luego que se termina de pronunciar el comando de voz, con un ruido ambiental en la cabina del vehículo medido como 0%, es decir un silencio absoluto, hasta llegar al valor máximo admitido en el interior del automóvil con el 100% de ruido ambiental , lo cual significa el reproductor de sonido del vehículo con 100% de volumen.

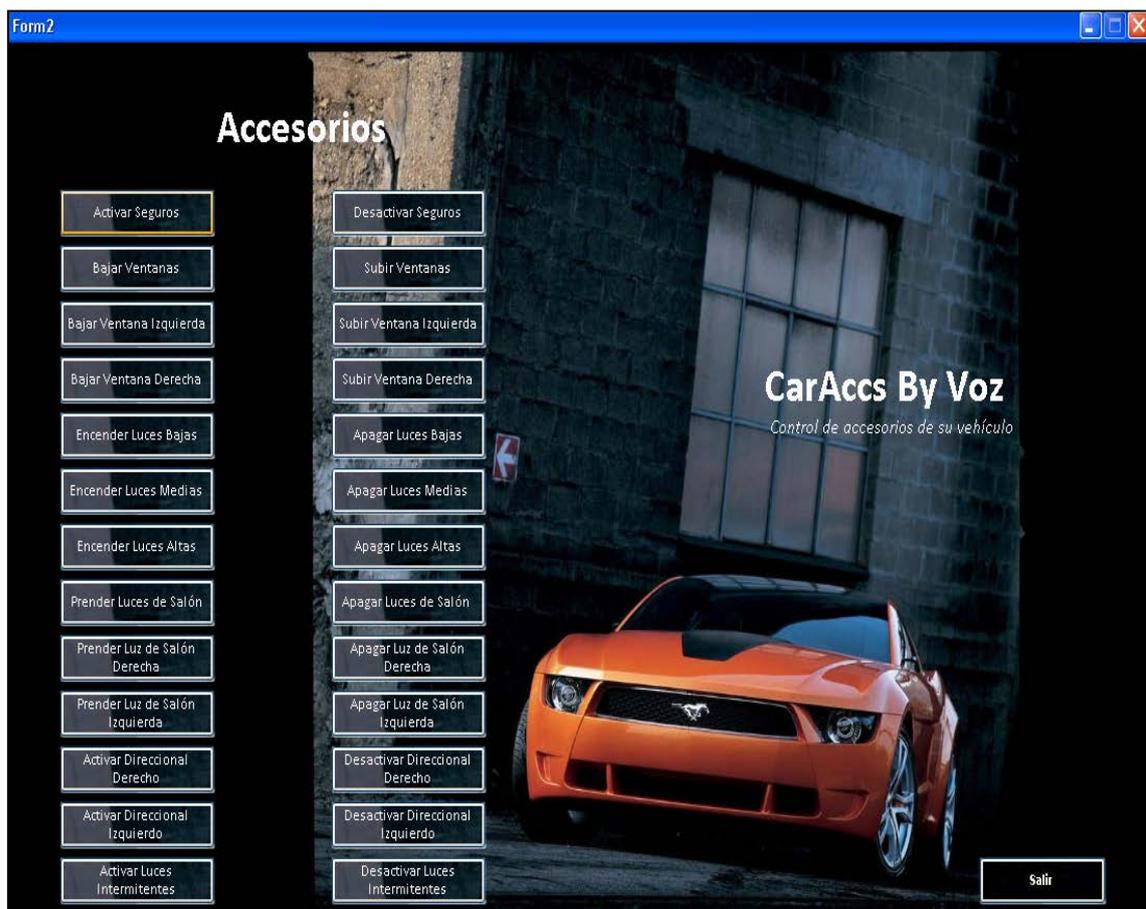


Figura 5.9. Lista de accesorios controlados por voz en el vehículo.

Autor: Andrés E. & Andrés Y.

Ante esta situación y para evitar fallas, estos programas poseen ciertas ventajas como control de activación y desactivación del micrófono o control de reinicio rápido que consiste en que si el comando de voz no concuerda con la acción que se necesita, el programa se reactiva al pronunciar el comando de voz indicado en el programa que está siendo usado por el usuario.

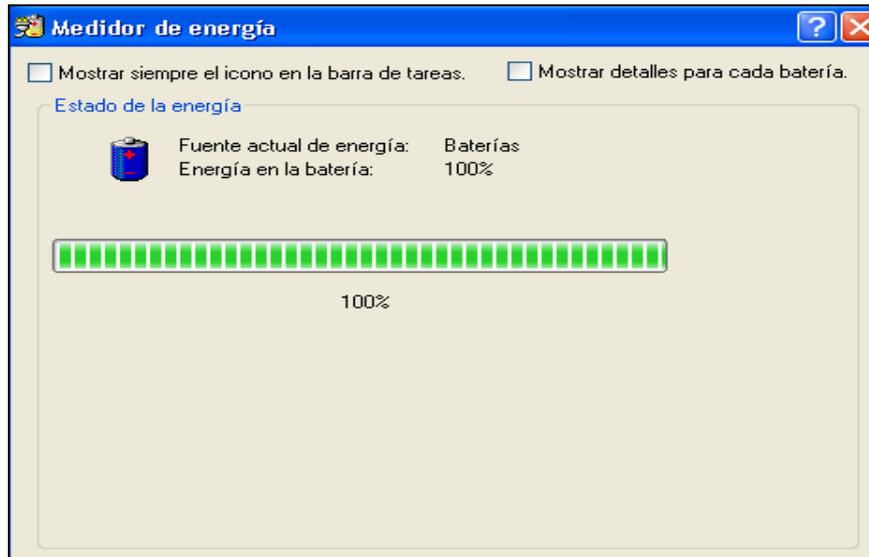
### **5.2.2. Prueba de autonomía de batería del computador usando el sistema**

Esta prueba se realiza con el objetivo de precisar cual es el tiempo de duración de la carga de la batería del computador, desde el 100% que es carga total (Figura 5.10), hasta la descarga total de la misma valorado en el 0%, usando el sistema de reconocimiento de voz en el vehículo.

En la primera parte de la prueba se realiza un recorrido de 45 minutos en ciudad, cuyo tiempo de inicio fue las 18H: 00, partiendo desde la Universidad Católica del Ecuador en Quito (Batería al 100%), hasta el aeropuerto.

En la segunda parte de la prueba se realiza un recorrido de 120 minutos, desde la avenida Eloy Alfaro y Gaspar de Villarroel en Quito (Batería al 68%), la hora de partida son las 14H: 00, y el lugar de llegada de este recorrido es el terminal terrestre en la ciudad de Ambato.

Durante toda la prueba el sistema de reconocimiento de voz se encuentra activo, y constantemente en uso.



**Figura 5.10. Medidor de batería – 100% disponible**

Autor: Andrés E. & Andrés Y.

### 5.3.RESULTADOS

Una vez realizadas las pruebas los resultados han sido resumidos y tabulados en las siguientes tablas:

**Tabla 5.1. Precisión de Reconocimiento de voz Vs. Volumen del reproductor de sonido.**

PRUEBA	VOLUMEN [%]	PRECISIÓN [%]
1	0	100
2	25	100
3	50	95
4	75	62
5	100	25

Autor: Andrés E. & Andrés Y.

Como se observa en la tabla 5.1 la precisión del reconocimiento de voz disminuye importantemente al 25%, cuando el volumen del reproductor de sonido del vehículo es incrementado a su valor máximo, aunque este es un nivel de volumen no usado en la vida cotidiana, razón por la cual no es preocupante que exista un bajo nivel de precisión en el reconocimiento para este proyecto.

El literal 1, 2 y 3 de pruebas, que se encuentra en la tabla 5.1, muestra claramente que este proyecto tiene una gran viabilidad, ya que el mismo tiene un altísimo porcentaje de precisión hasta con un 50% de volumen del reproductor, este valor de volumen es el máximo utilizado por el común de las personas al momento de encontrarse manejando el automotor o como pasajero.

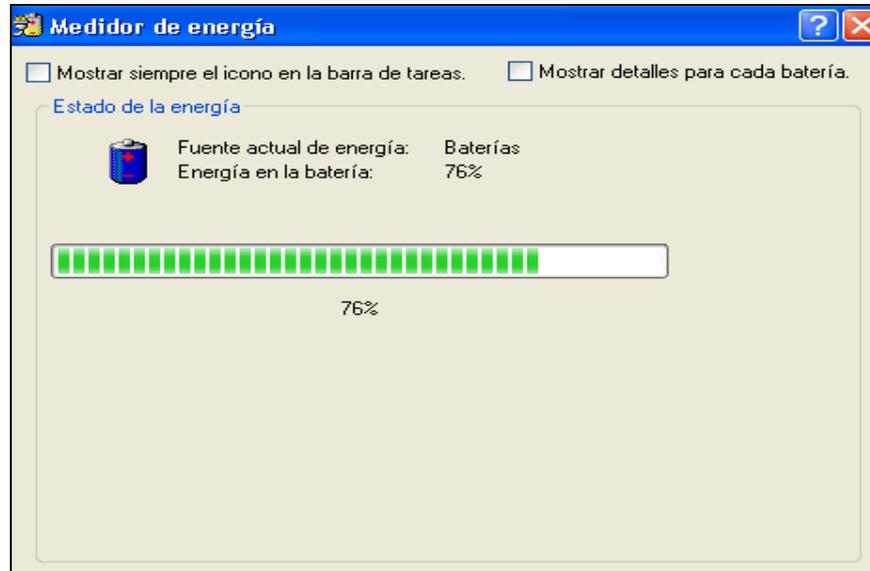
**Tabla 5.2. Autonomía de la batería del computador.**

PRUEBA	TIEMPO DE USO	DESCARGA DE BATERÍA [%]
1	45 Minutos	24
2	120 Minutos	65
TOTAL	165 Minutos	89

Autor: Andrés E. & Andrés Y.

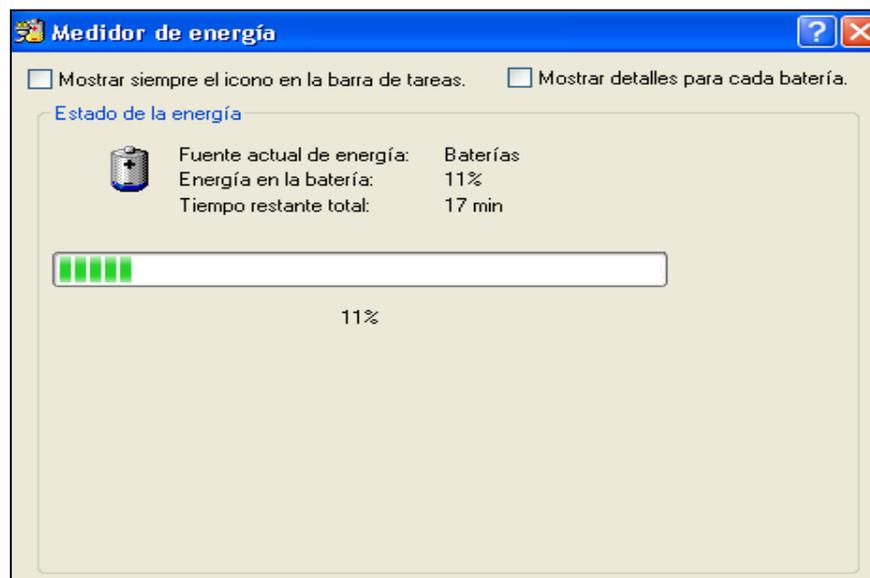
La tabla 5.2 muestra el porcentaje de descarga de la batería tras realizar cada uno de los recorridos anteriormente planificados; luego de finalizar el primer recorrido utilizando el sistema de reconocimiento de voz, se tiene un 24% menos del total de carga de la batería del computador (Figura 5.11), luego del segundo recorrido de esta prueba el porcentaje de batería descargada durante esta etapa es del 65%, sumando un total de 89% (Figura 5.12) de utilización de la energía de

la batería, tras 165 Minutos de actividad del sistema; así se concluye que la autonomía de la batería del computador esta alrededor de los 180 a 185 minutos.



**Figura 5.11. Medidor de batería – 76% disponible**

Autor: Andrés E. & Andrés Y.



**Figura 5.12. Medidor de batería – 11% disponible**

Autor: Andrés E. & Andrés Y.

## **CONCLUSIONES**

- Se ha construido e implementado satisfactoriamente un sistema innovador desarrollado en el país, capaz de controlar accesorios en un vehículo por comandos de voz, y que ofrece una verdadera comunicación directa y confiable entre el hombre y máquina.
- El proyecto proporciona al usuario un sistema que se caracteriza por sus variadas aplicaciones y fácil accesibilidad; mediante el uso de la interfaz gráfica diseñada exitosamente en visual basic 6.0 y 2010, que funciona bajo control del software de reconocimiento de voz (software comercial) caracterizándose por la capacidad inmensa para reconocer comandos de voz.
- La programación y grabación del microcontrolador realizada en MicroCode Studio y PICKit2 respectivamente, cumple con los requerimientos para la realización de tareas conjuntas en el vehículo, pudiendo controlar así más de un accesorio al mismo tiempo.
- Se ha comprobado que los programas utilizados son fiables, de rápido accionamiento, que trabajan en conjunto con la interface de control, cumpliendo el propósito de este proyecto. Incluso busca contribuir a la sociedad, brindando comodidad o asistencia para personas al activar o desactivar accesorios del vehículo mediante comandos de voz, especialmente a las personas que poseen discapacidad motriz y pueden beneficiarse de los programas de reconocimiento de voz.

## **RECOMENDACIONES**

- Ampliar el uso del reconocimiento de voz, y explotar el gran potencial que posee este tipo de software, ya que por sus beneficios y múltiples aplicaciones, se podría lograr un desarrollo significativo e importante en el área tecnológica del país.
- Buscar apoyo gubernamental para desarrollar proyectos en el campo del reconocimiento de voz para aporte social, que mejoren el estilo de vida de nuestra sociedad.
- Las características de la voz de una persona son parámetros variables por naturaleza de manera gradual, en donde la salud también juega un papel fundamental. En sistemas de reconocimiento de voz es recomendable brindar mantenimiento regular a la base de datos, es decir realizar un nuevo entrenamiento, el tiempo prudente para realizar esta actividad es cada dos meses, ya que esto permitirá que el sistema de reconocimiento de voz que se utilice funcione con gran precisión.
- Con la finalidad de que el sistema funcione correctamente es necesario que cada uno de los usuarios que usan el sistema de control de accesorios por voz, cree un usuario y realice el entrenamiento.

## **BIBLIOGRAFÍA**

- AXELSON, Jan. Serial Port Complete – Programming and circuits for RS-232 and Rs-485 Links and Networks, segunda edición, Estados Unidos, Editorial Lakeview Research, 2000.
- BLANCO VIEJO, Cecilio. Electrónica Digital, primera edición, España, Editorial de la Universidad de Oviedo, 2003.
- BLANCO VIEJO, Cecilio. Fundamentos de Electrónica Digital, segunda edición, España, Editorial Thomson Paraninfo, 2005.
- CASTAÑEDA, Pablo Félix. El lenguaje verbal del niño, primera edición, Perú, Universidad Nacional Mayor de San Marcos, 1999.
- HAMBLEY, Alan. Electrónica, segunda edición, España, Editorial Pearson, 2001.
- LUQUE SACALUGA, David. Electrónica Digital y Microprogramable, primera edición, México, Editorial Alfa Omega Ra-Ma, 2006.
- LLORIS RUIZ, Antonio, PRIETO ESPINOSA, Alberto, PARRILLA ROURE, Luis. Sistemas Digitales, primera edición, España, Editorial Edigrafos, 2003.
- NISSAN. Nissan Pathfinder Service Manual Electrical System, 2000.
- PALACIOS MUNICIO, Enrique, REMIRO DOMÍNGUEZ, Fernando y LÓPEZ PÉREZ, Lucas. Microcontrolador PIC16F84 – Desarrollo de proyectos, primera edición, México, Editorial Alfaomega, 2004.

- PRAT VIÑAS, Lluís, Circuitos y Dispositivos electrónicos, quinta edición, España, Editorial de la Universidad Politécnica de Cataluña, 1998.
- PRESSMAN, Roger. Ingeniería del Software, Un enfoque práctico, segunda edición, Estados Unidos, Editorial McGraw Hill, 1998.
- REYES, Carlos. Microcontroladores PIC Programación en Basic, segunda edición, Quito - Ecuador, Impreso por RISPERGRAF, 2006.
- Revista UControl, Electrónica en General PICS en Particular - Construye tus propios circuitos, número 1, Argentina, Enero 2008.

## ANEXOS

### Anexo 1. Programación del PIC

<b>Códigos</b>	<b>Descripción</b>
INCLUDE "modedefs.bas"	Código que activa modo de comunicación: Serial (Ordenador- PIC)
dato Var byte	Variable de almacenamiento
inicio:	Inicio de rutina
SERIN portc.7,T9600, dato	Esperar por dato serial, ejecutar acciones y establecer velocidad de transmisión de datos
If dato="A" then ON_LED1	Si dato "A" entonces ir a subrutina Activa relé 1 y se desactiva en 2 segundos (seguros)
If dato="B" THEN ON_LED2	Si dato "B" entonces ir a subrutina Activa relé 2 y se desactiva en 2 segundos (seguros)
If dato="C" THEN ON_LED35	Si dato "C" entonces ir a subrutina Activa relé 3-5 y baja las 2 ventanas
If dato="D" THEN ON_LED46	Si dato "D" entonces ir a subrutina Activa relé 4-6 y sube las 2 ventanas
If dato="E" THEN ON_LED3	Si dato "E" entonces ir a subrutina Activa relé 3 y se desactiva en 6 segundos (vent.sube1)
If dato="F" THEN ON_LED4	Si dato "F" entonces ir a subrutina Activa relé 4 y se desactiva en 6 segundos (vent.baja1)
If dato="G" then ON_LED5	Si dato "G" entonces ir a subrutina Activa relé 5 y se desactiva en 6 segundos (vent.sube1)

If dato="H" THEN ON_LED6	Si dato "H" entonces ir a subrutina Activa relé 6 y se desactiva en 6 segundos (vent.baja2)
If dato="I" then ON_LED7	Si dato "I" entonces ir a subrutina Activa relé 7 (activa luces bajas)
If dato="J" THEN HIGH portb.1	Si dato "J" entonces apagar portb.1 (Portb.1 tiene lógica inversa) Desactiva relé 7 (desactiva luces bajas)
If dato="K" THEN ON_LED8	Si dato "K" entonces ir a subrutina Prende luces medias y apaga las altas en 2 segundos
If dato="L" THEN LOW portd.6	Si dato "L" entonces apagar portd.6 Apaga luces medias
If dato="M" THEN ON_LED9	Si dato "M" entonces ir a subrutina Prende luces altas y apaga las medias en 2 segundos
If dato="N" THEN LOW portd.5	Si dato "N" entonces apagar portd.5 Apaga luces altas ( portd.5)
If dato="O" then ON_LED1011	Si dato "O" entonces ir a subrutina Prende luces de salón
If dato="P" THEN LOW_LED1110	Si dato "P" entonces ir a subrutina Apaga luces de salón
If dato="Q" THEN ON_LED10	Si dato "Q" entonces ir a subrutina Prende luz de salón izquierda
If dato="R" THEN LOW portd.4	Si dato "R" entonces apagar portd.4 Apaga luz de salón izquierda
If dato="S" then ON_LED11	Si dato "S" entonces ir a subrutina Prende luz de salón derecha
If dato="T" THEN LOW portc.5	Si dato "T" entonces apagar portc.5 Apaga luz de salón derecha
If dato="U" THEN ON_LED12	Si dato "U" entonces ir a subrutina Prende direccional izquierda

If dato="V" THEN LOW portc.5	Si dato "V" entonces ir a subrutina Apaga direccional izquierda
If dato="W" THEN ON_LED13	Si dato "W" entonces ir a subrutina Prende direccional derecha
If dato="X" THEN LOW portd.2	Si dato "X" entonces apagar portd.2 Apaga direccional derecha
If dato="Y" then ON_LED1213	Si dato "Y" entonces ir a subrutina Prende intermitentes
If dato="Z" THEN LOW_LED1312	Si dato "Z" entonces ir a subrutina Apaga intermitentes
GOTO INICIO	Ir a inicio
<b>ON_LED1:</b>	Nombre de la subrutina
HIGH portb.7	Prender portb.7
pause 2000	Pausar 2 segundos
LOW portb.7	Apagar portb.7
GOTO inicio	Ir a inicio
<b>ON_LED2:</b>	Nombre de la subrutina
HIGH portb.6	Prender portb.6
pause 2000	Pausar 2000
LOW portb.6	Apagar portb.6
GOTO inicio	Ir a inicio
<b>ON_LED35:</b>	Nombre de la subrutina
HIGH portb.5:high portb.3	Prender portb.5 / Prender portb.3
Pause 6000	Pausar 6 segundos
LOW portb.5:LOW portb.3	Apagar portb.5 / Apagar portb.3
GOTO inicio	Ir a inicio
<b>ON_LED46:</b>	Nombre de la subrutina
HIGH portb.4:high portb.2	Prender portb.4 / Prender portb.2

Pause 6000	Pausar 6 segundos
LOW portb.4:LOW portb.2	Apagar portb.4 / Apagar portb.2
GOTO inicio	Ir a inicio
<b>ON_LED3:</b>	Nombre de la subrutina
HIGH portb.5	Prender portb.5
Pause 6000	Pausar 6 segundos
LOW portb.5	Apagar portb.5
GOTO inicio	Ir a inicio
<b>ON_LED4:</b>	Nombre de la subrutina
HIGH portb.4	Prender portb.4
Pause 6000	Pausar 6 segundos
LOW portb.4	Apagar portb.4
GOTO inicio	Ir a inicio
<b>ON_LED5:</b>	Nombre de la subrutina
HIGH portb.3	Prender portb.3
Pause 6000	Pausar 6 segundos
LOW portb.3	Apagar portb.3
GOTO inicio	Ir a inicio
<b>ON_LED6:</b>	Nombre de la subrutina
HIGH portb.2	Prender portb.2
Pause 6000	Pausar 6 segundos
LOW portb.2	Apagar portb.2
GOTO inicio	Ir a inicio
<b>ON_LED7:</b>	Nombre de la subrutina
LOW portb.1	(Portb.1 funciona con lógica inversa) Prender portb.1

GOTO inicio	Ir a inicio
<b>ON_LED8:</b>	Nombre de la subrutina
HIGH portd.6	Prender portd.6
LOW portd.5	Apagar portd.5
GOTO inicio	Ir a inicio
<b>ON_LED9:</b>	Nombre de la subrutina
HIGH portd.5	Prender portd.5
LOW portd.6	Apagar portd.6
GOTO inicio	Ir a inicio
<b>ON_LED10:</b>	Nombre de la subrutina
HIGH portd.4	Prender portd.4
GOTO inicio	Ir a inicio
<b>ON_LED11:</b>	Nombre de la subrutina
HIGH portc.5	Prender portc.5
GOTO inicio	Ir a inicio
<b>ON_LED1011:</b>	Nombre de la subrutina
HIGH portd.4:high portc.5	Prender portd.4 / Prender portc.5
GOTO inicio	Ir a inicio
<b>LOW_LED1110:</b>	Nombre de la subrutina
LOW portd.4:LOW portc.5	Apagar portd.4 / Apagar portc.5
GOTO inicio	Ir a inicio
<b>ON_LED12:</b>	Nombre de la subrutina
HIGH portc.4	Prender portc.4
GOTO inicio	Ir a inicio
<b>ON_LED13:</b>	Nombre de la subrutina

HIGH portd.2	Prender portd.2
GOTO inicio	Ir a inicio
<b>ON_LED1213:</b>	Nombre de la subrutina
HIGH portc.4:high portd.2	Prender portc.4 / Prender portd.2
GOTO inicio	Ir a inicio
<b>LOW_LED1312:</b>	Nombre de la subrutina
LOW portc.4:LOW portd.2	Apagar portc.4 / Apagar portd.2
GOTO inicio	Ir a inicio
END	Finalizar rutina

## Anexo 2. Programación software Visual Basic 6.0

<b>Activar Seguros (1er Relé)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "A"	Envío de dato de salida ("A")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Desactivar Seguros ( 2do.Relé)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.

MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "B"	Envío de dato de salida ("B")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Activar Ventanas (Relé 3-5)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "C"	Envío de dato de salida ("C")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina

<b>Desactivar Ventanas (Relé 4-6))</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "D"	Envío de dato de salida ("D")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Activar Ventana Izquierda (Relé 3)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)

MSComm1.Output = "E"	Envío de dato de salida ("E")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Desactivar Ventana Izquierda (Relé 4)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "F"	Envío de dato de salida ("F")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina

<b>Activar Ventana Derecha (Relé 5)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "G"	Envío de dato de salida ("G")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Desactivar Ventana Derecha (Relé 6)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)

MSComm1.Output = "H"	Envío de dato de salida ("H")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Activar Luces Bajas ( Relé 7)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "I"	Envío de dato de salida ("I")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina

<b>Desactivar Luces Bajas (Relé 7)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "J"	Envío de dato de salida ("J")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Activar Luces Medias (Relé 8)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "K"	Envío de dato de salida ("K")

MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Desactivar Luces Medias (Relé 8)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "L"	Envío de dato de salida ("L")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Activar Luces Altas (Relé 9)</b>	

<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSCComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSCComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSCComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSCComm1.Output = "M"	Envío de dato de salida ("M")
MSCComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSCComm1.PortOpen = True Then MSCComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Desactivar Luces Altas (Relé 9)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSCComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSCComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSCComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSCComm1.Output = "N"	Envío de dato de salida ("N")
MSCComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)

End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Activar Luces de Salón (Relé 10-11)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "O"	Envío de dato de salida ("O")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Desactivar Luces de Salón (relé 10-11)</b>	

<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "P"	Envío de dato de salida ("P")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Activar Luz de Salón Derecha (Relé 10)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "Q"	Envío de dato de salida ("Q")

MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Desactivar Luz de Salón Derecha (Relé 10)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "R"	Envío de dato de salida ("R")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina

<b>Activar Luz de Salón Izquierda (Relé 11)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "S"	Envío de dato de salida ("S")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Desactivar Luz de Salón Izquierda (Relé 11)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.

MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "T"	Envío de dato de salida ("T")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Activar Direccional Derecha (Relé 12)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "U"	Envío de dato de salida ("U")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.

<b>End Sub</b>	Cierra la subrutina
<b>Desactivar Direccional Derecha (Relé 12)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "V"	Envío de dato de salida ("V")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Activar Direccional Izquierda (Relé 13)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).

MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "W"	Envío de dato de salida ("W")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Desactivar Direccional Izquierda (Relé 13)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "X"	Envío de dato de salida ("X")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then	Ordena al puerto cerrarse, en caso de

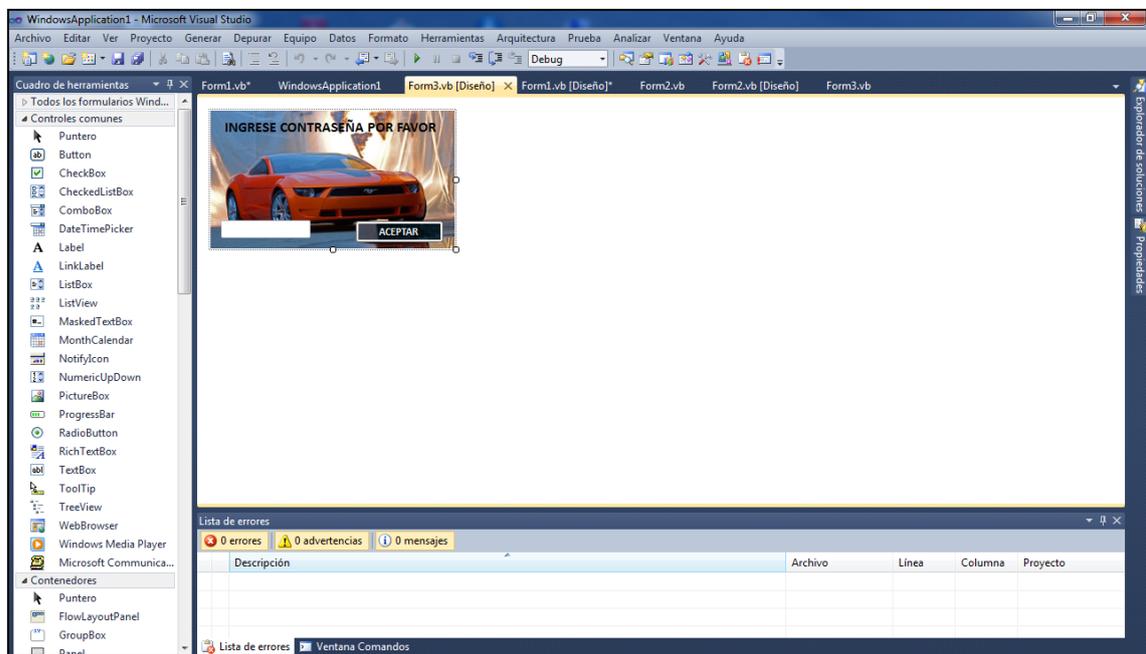
MSComm1.PortOpen = False	estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Activar Luces Intermitentes (Relé 12-13)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "Y"	Envío de dato de salida ("Y")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina
<b>Desactivar Luces Intermitentes (Relé 12-13)</b>	
<b>Private Sub Form_Load()</b>	Hoja de diseño de Visual Basic
MSComm1.CommPort = 16	Permite abrir el puerto serial (nro.16), esto después de haber instalado drivers, que permiten activar el cable

	USB a serial. (El puerto serial puede modificarse según la computadora).
MSComm1.Settings = "9600,n,8,1"	Establece la velocidad de transmisión de datos, etc.
MSComm1.PortOpen = True	Orden que activa el puerto (nro.16)
MSComm1.Output = "Z"	Envío de dato de salida ("Z")
MSComm1.PortOpen = False	Orden que desactiva el puerto (nro.16)
End	Cierra la aplicación.
<b>End Sub</b>	Cierra la Subrutina
<b>Private Sub Form_Unload(Cancel As Integer)</b>	Permite que la visualización de la ventana no pueda ser activada.
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False	Ordena al puerto cerrarse, en caso de estar activado, para realizar el envío de datos con el resto de variables.
<b>End Sub</b>	Cierra la subrutina

## Anexo 3. Elaboración software Visual Basic 2010

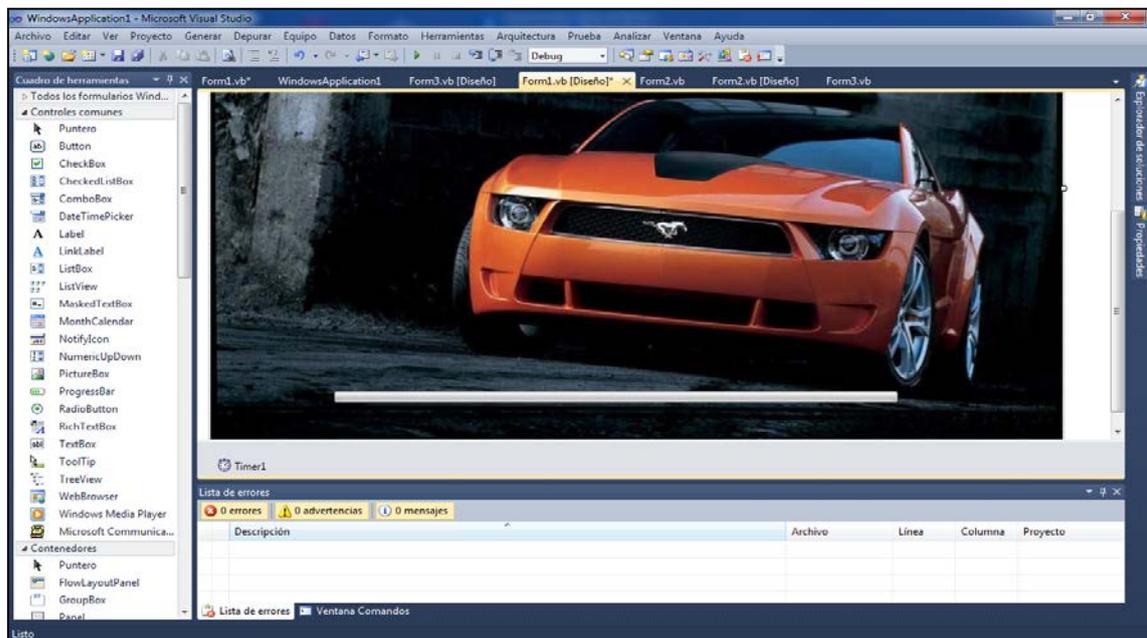
La elaboración de la interface gráfica se realiza con varios códigos e imágenes, que permiten obtener un software en versión ejecutable para el usuario final, que se complementa con los programas existentes en Windows y especialmente los que están hechos en Visual Basic 6.0.

### Interfaz grafica (Códigos)



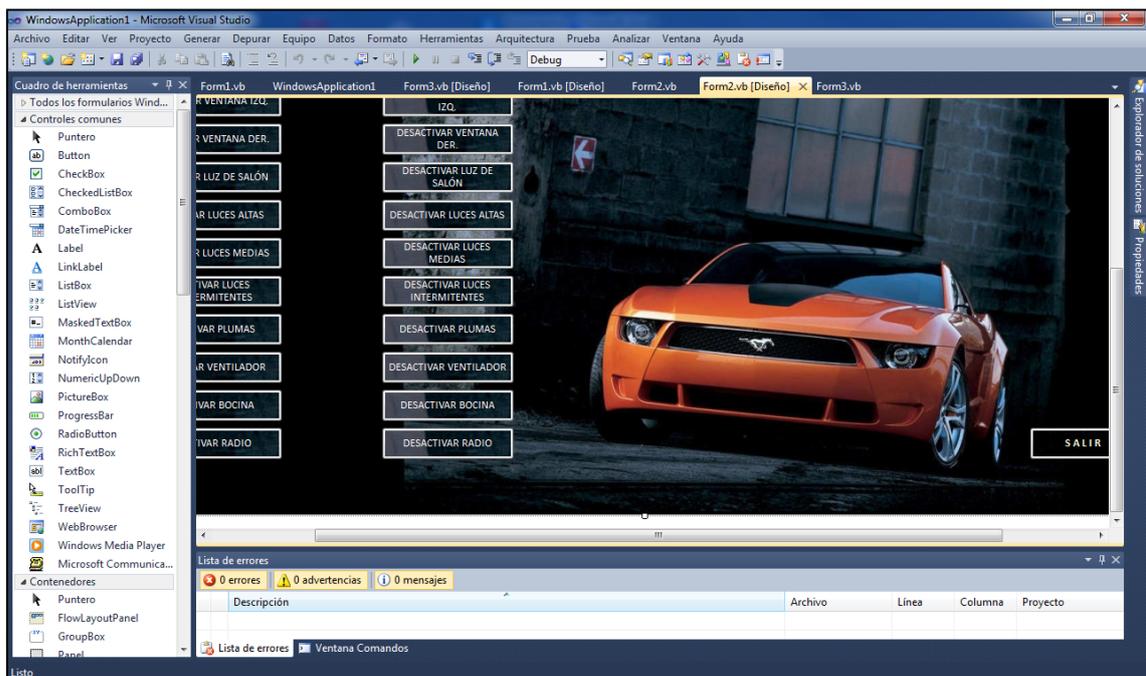
Elementos	Códigos
<b>Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load</b>	My.Computer.Audio.Play(My.Resources.ingrese_contraseña, AudioPlayMode.Background)
<b>Hoja de diseño (Nro.3)</b>	Permite reproducir sonidos; archivos de extensión (.wav), guardados previamente en el proyecto

<p><b>Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click</b></p>	<pre>If TextBox1.Text = "1342" Then     Form1.Show()     Me.Visible = False Else     My.Computer.Audio.Play(My.Resources.contraseña_incorrecta, AudioPlayMode.Background)     MsgBox("Contraseña Incorrecta", , "Advertencia") End If</pre>
<p><b>Boton 1(ACEPTAR)</b></p>	<p>Declara que si textbox1, es igual "1342", permita visualizar la hoja de diseño (Nro.1). Se cierra la hoja de diseño (Nro.3). En caso de ser incorrecta la contraseña, se reproduce sonido y aparecerá una ventana indicándole al usuario que debe ingresar la contraseña correcta.</p>



<p><b>Elementos</b></p>	<p><b>Códigos</b></p>
<p><b>Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)</b></p>	<pre>ProgressBar1.Increment(1)  If ProgressBar1.Value = ProgressBar1.Maximum Then</pre>

<b>Handles Timer1.Tick</b>	<pre> Form2.Show()  Me.Visible = False  Timer1.Enabled = False  ProgressBar1.Value = 1  End If </pre>
<b>Temporizador / ProgressBar</b>	<p>El temporizador permite activar la barra de progreso, en donde se indica que cuando llegue a su punto final se active la hoja de diseño (Nro.2), cerrando la Hoja de diseño (Nro.1) y el temporizador. El intervalo de tiempo de ejecución la barra de progreso, es de 1 segundo multiplicado por la distancia de la misma.</p>
<b>Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load</b>	<pre> My.Computer.Audio.Play(My.Resources.saludo1, AudioPlayMode.Background)  Timer1.Start() </pre>
<b>Hoja de diseño (Nro.1)</b>	<p>Cuando se activa la Hoja de diseño (Nro.1), se reproduce sonido y se activa el temporizador.</p>



Elementos	Códigos
<b>Private Sub Form2_Load (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load</b>	My.Computer.Audio.Play(My.Resources.SALUDO_2, AudioPlayMode.Background)
<b>Hoja de diseño (Nro.2)</b>	Reproducción de sonido (SALUDO_2), al iniciar hoja de diseño (Nro.2).
<b>Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click</b>	GetAttr("C:\Users\Hogar\Desktop\TESIS\LED1\V BASIC PRENDER 1\Prender 1.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\LED1\V BASIC PRENDER 1\Prender 1.exe")  My.Computer.Audio.Play(My.Resources.SEGUROS_ACTIVADOS, AudioPlayMode.Background)
<b>Botón 1 (Activar Seguros)</b>	Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender 1.exe), desarrollado en Visual Basic 6.0.  Reproducción de sonido.
<b>Private Sub Button28_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button28.Click</b>	GetAttr("C:\Users\Hogar\Desktop\TESIS\LED2\V BASIC PRENDER2\Prender2.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\LED2\V BASIC PRENDER2\Prender2.exe")  My.Computer.Audio.Play(My.Resources.SEGUROS_DESACTIVADOS, AudioPlayMode.Background)
<b>Botón 28 (Desactivar)</b>	Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender2.exe), desarrollado en Visual Basic 6.0.

<b>Seguros)</b>	Reproducción de sonido. (SEGUROS_DESACTIVADOS)
<b>Private Sub Button2_Click( ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click</b>	<p>GetAttr("C:\Users\Hogar\Desktop\TESIS\led3-5\prender led3-5\Prender3-5.exe")</p> <p>Shell("C:\Users\Hogar\Desktop\TESIS\led3-5\prender led3-5\Prender3-5.exe")</p> <p>My.Computer.Audio.Play(My.Resources.VENTANAS_ACTIVADAS, AudioPlayMode.Background)</p>
<b>Botón 2 (Activar Ventanas)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender3-5.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido. (VENTANAS_ACTIVADAS)</p>
<b>Private Sub Button11_Click( ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button11.Click</b>	<p>GetAttr("C:\Users\Hogar\Desktop\TESIS\led4-6\apagar led4-6\Apagar 4-6.exe")</p> <p>Shell("C:\Users\Hogar\Desktop\TESIS\led4-6\apagar led4-6\Apagar 4-6.exe")</p> <p>My.Computer.Audio.Play(My.Resources.VENTANAS_DESACTIVADAS, AudioPlayMode.Background)</p>
<b>Botón 11 (Desactivar Ventanas)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Apagar 4-6.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido. (VENTANAS_DESACTIVADAS)</p>
<b>Private Sub Button3_Click( ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click</b>	<p>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led3\VBasic Prender Led 3\Prender 3.exe")</p> <p>Shell("C:\Users\Hogar\Desktop\TESIS\Led3\VBasic Prender Led 3\Prender 3.exe")</p> <p>My.Computer.Audio.Play(My.Resources.VENTANA_IZQUIERDA_ACTIVADA, AudioPlayMode.Background)</p>

<b>Botón 3</b>  <b>(Activar Ventana Izquierda)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_3.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido.( VENTANA_IZQUIERDA_ACTIVADA)</p>
<b>Private Sub Button12_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button12.Click</b>	<p>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led4\VBasic Prender Led4\Prender_4.exe")</p> <p>Shell("C:\Users\Hogar\Desktop\TESIS\Led4\VBasic Prender Led4\Prender_4.exe")</p> <p>My.Computer.Audio.Play(My.Resources.VENTANA_IZQUIERDA_DESACTIVADA, AudioPlayMode.Background)</p>
<b>Botón 12</b>  <b>(Desactivar Ventana Izquierda)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_4.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido.</p> <p>(VENTANA_IZQUIERDA_DESACTIVADA)</p>
<b>Private Sub Button5_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click</b>	<p>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led5\Prender Led 5\Prender_5.exe")</p> <p>Shell("C:\Users\Hogar\Desktop\TESIS\Led5\Prender Led 5\Prender_5.exe")</p> <p>My.Computer.Audio.Play(My.Resources.VENTANA_DERECHA_ACTIVADA, AudioPlayMode.Background)</p>
<b>Botón 5</b>  <b>(Activar Ventana Derecha)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_5.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido.( VENTANA_DERECHA_ACTIVADA)</p>
<b>Private Sub Button13_Click (ByVal sender</b>	<p>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led6\Prender Led 6\Prender_6.exe")</p>

<b>As System.Object, ByVal e As System.EventArgs) Handles Button13.Click</b>	<pre>Shell("C:\Users\Hogar\Desktop\TESIS\Led6\Prender Led 6\Prender_6.exe")  My.Computer.Audio.Play(My.Resources.VENTANA_DERECHA_DESACTIVADA, AudioPlayMode.Background)</pre>
<b>Botón 13 (Desactivar Ventana Derecha)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_6.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido. (VENTANA_DERECHA_DESACTIVADA)</p>
<b>Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click</b>	<pre>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led7\Prender Led 7\Prender_7.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\Led7\Prender Led 7\Prender_7.exe")  My.Computer.Audio.Play(My.Resources.LUCES_BAJAS_ACTIVADAS, AudioPlayMode.Background)</pre>
<b>Botón 6 (Activar Luces Bajas)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_7.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido. (LUCES_BAJAS_ACTIVADAS)</p>
<b>Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button14.Click</b>	<pre>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led7\Apagar Led 7\Apagar_7.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\Led7\Apagar Led 7\Apagar_7.exe")  My.Computer.Audio.Play(My.Resources.LUCES_BAJAS_DESACTIVADAS, AudioPlayMode.Background)</pre>
<b>Botón 14 (Desactivar Luces Bajas)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Apagar_7.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido.( LUCES_BAJAS_DESACTIVADAS)</p>

<b>Private Sub Button7_Click( ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button7.Click</b>	<pre>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led8\Prender Led 8\Prender_8.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\Led8\Prender Led 8\Prender_8.exe")  My.Computer.Audio.Play(My.Resources.LUCES_MEDIAS_ACTIVADAS, AudioPlayMode.Background)</pre>
<b>Botón 7 (Activar Luces Medias)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_8.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido. (LUCES_MEDIAS_ACTIVADAS)</p>
<b>Private Sub Button15_Click( ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button15.Click</b>	<pre>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led8\Apagar Led 8\Apagar_8.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\Led8\Apagar Led 8\Apagar_8.exe")  My.Computer.Audio.Play(My.Resources.LUCES_MEDIAS_DESACTIVADAS, AudioPlayMode.Background)</pre>
<b>Botón 15 (Desactivar Luces Medias)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Apagar_8.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido. (LUCES_MEDIAS_DESACTIVADAS)</p>
<b>Private Sub Button8_Click( ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button8.Click</b>	<pre>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led9\Prender Led 9\Prender_9.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\Led9\Prender Led 9\Prender_9.exe")  My.Computer.Audio.Play(My.Resources.LUCES_ALTAS_ACTIVADAS, AudioPlayMode.Background)</pre>
<b>Botón 8 (Activar Luces</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_9.exe), desarrollado en Visual</p>

<b>Altas)</b>	Basic 6.0. Reproducción de sonido. (LUCES_ALTAS_ACTIVADAS)
<b>Private Sub Button16_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button16.Click</b>	GetAttr("C:\Users\Hogar\Desktop\TESIS\Led9\Apagar Led 9\Apagar_9.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\Led9\Apagar Led 9\Apagar_9.exe")  My.Computer.Audio.Play(My.Resources.LUCES_ALTAS_DESACTIVADAS, AudioPlayMode.Background)
<b>Botón 16 (Desactivar Luces altas)</b>	Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Apagar_9.exe), desarrollado en Visual Basic 6.0.  Reproducción de sonido. (LUCES_ALTAS_DESACTIVADAS)
<b>Private Sub Button9_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button9.Click</b>	GetAttr("C:\Users\Hogar\Desktop\TESIS\led10-11\Prender Led 10-11\Prender10-11.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\led10-11\Prender Led 10-11\Prender10-11.exe")  My.Computer.Audio.Play(My.Resources.LUCES_DE_SALON_ACTIVADAS, AudioPlayMode.Background)
<b>Botón 9 (Activar Luces de Salón)</b>	Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender10-11.exe), desarrollado en Visual Basic 6.0.  Reproducción de sonido. (LUCES_DE_SALON_ACTIVADAS)
<b>Private Sub Button17_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button17.Click</b>	GetAttr("C:\Users\Hogar\Desktop\TESIS\led10-11\Apagar Led 10-11\Apagar_10-11.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\led10-11\Apagar Led 10-11\Apagar_10-11.exe")  My.Computer.Audio.Play(My.Resources.LUCES_DE_SALON_DESACTIVADAS, AudioPlayMode.Background)

<b>Botón 17</b> <b>(Desactivar Luces de Salón)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Apagar_10-11.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido.  (LUCES_DE_SALON_DESACTIVADAS)</p>
<b>Private Sub Button20_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button20.Click</b>	<pre>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led10\Prender Led 10\Prender_10.exe") Shell("C:\Users\Hogar\Desktop\TESIS\Led10\Prender Led 10\Prender_10.exe") My.Computer.Audio.Play(My.Resources.LUZ_DE_SALÓN_DE RECHA_ACTIVADA, AudioPlayMode.Background)</pre>
<b>Botón 20</b> <b>(Activar Luz de Salón Derecha)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_10.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido.  (LUZ_DE_SALÓN_DERECHA_ACTIVADA)</p>
<b>Private Sub Button27_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button27.Click</b>	<pre>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led10\Apagar Led 10\Apagar_10.exe") Shell("C:\Users\Hogar\Desktop\TESIS\Led10\Apagar Led 10\Apagar_10.exe") My.Computer.Audio.Play(My.Resources.LUZ_DE_SALÓN_DE RECHA_DESACTIVADA, AudioPlayMode.Background)</pre>
<b>Botón 20</b> <b>(Desactivar Luz de Salón Derecha)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Apagar_10.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido.  (LUZ_DE_SALÓN_DERECHA_DESACTIVADA)</p>
<b>Private Sub Button19_Click (ByVal sender As</b>	<pre>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led11\Prender Led 11\Prender_11.exe") Shell("C:\Users\Hogar\Desktop\TESIS\Led11\Prender Led</pre>

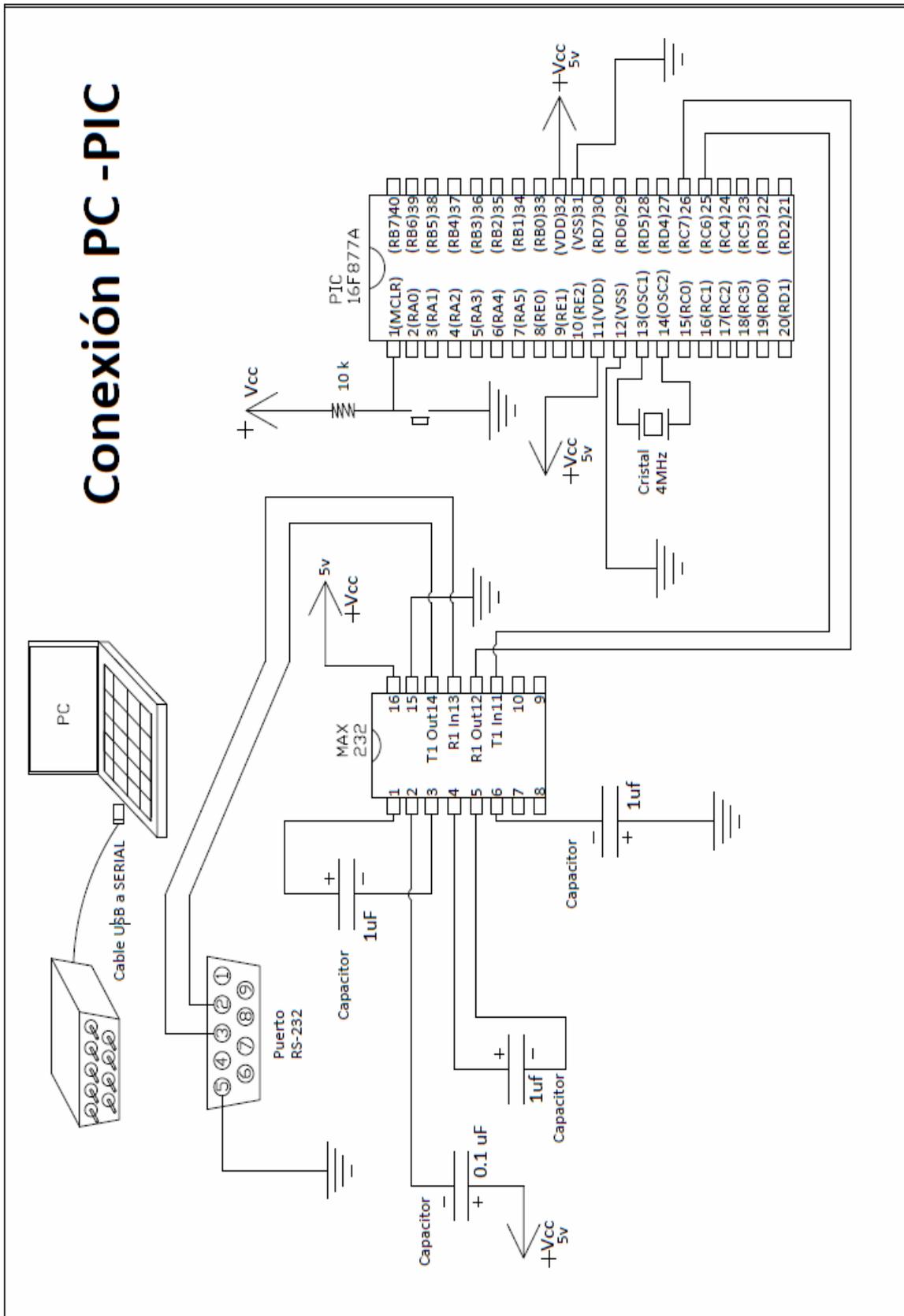
<b>System.Object, ByVal e As System.EventArgs) Handles Button19.Click</b>	<p>11\Prender_11.exe")</p> <p>My.Computer.Audio.Play(My.Resources.LUZ_DE_SALÓN_IZQUIERDA_ACTIVADA, AudioPlayMode.Background)</p>
<b>Botón 19 (Activar Luz de Salón Izquierda)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_11.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido.</p> <p>(LUZ_DE_SALÓN_IZQUIERDA_ACTIVADA)</p>
<b>Private Sub Button26_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button26.Click</b>	<p>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led11\Apagar Led 11\Apagar_11.exe")</p> <p>Shell("C:\Users\Hogar\Desktop\TESIS\Led11\Apagar Led 11\Apagar_11.exe")</p> <p>My.Computer.Audio.Play(My.Resources.LUZ_DE_SALÓN_IZQUIERDA_DESACTIVADA, AudioPlayMode.Background)</p>
<b>Botón 26 (Desactivar Luz de Salón Izquierda)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Apagar_11.exe), desarrollado en Visual Basic 6.0.</p> <p>Reproducción de sonido.</p> <p>(LUZ_DE_SALÓN_IZQUIERDA_DESACTIVADA)</p>
<b>Private Sub Button18_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button18.Click</b>	<p>GetAttr("C:\Users\Hogar\Desktop\TESIS\Led12\Prender Led 12\Prender_12.exe")</p> <p>Shell("C:\Users\Hogar\Desktop\TESIS\Led12\Prender Led 12\Prender_12.exe")</p> <p>My.Computer.Audio.Play(My.Resources.DIRECCIONAL_DERECHA_ACTIVADA, AudioPlayMode.Background)</p>
<b>Botón 18 (Activar Direccional)</b>	<p>Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_12.exe), desarrollado en Visual Basic 6.0.</p>

<b>Derecha)</b>	Reproducción de sonido. (DIRECCIONAL_DERECHA_ACTIVADA)
<b>Private Sub Button25_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button25.Click</b>	GetAttr("C:\Users\Hogar\Desktop\TESIS\Led12\Apagar Led 12\Apagar_12.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\Led12\Apagar Led 12\Apagar_12.exe")  My.Computer.Audio.Play(My.Resources.DIRECCIONAL_DERECHA_DESACTIVADA, AudioPlayMode.Background)
<b>Botón 25 (Desactivar Direccional Derecha)</b>	Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Apagar_12.exe), desarrollado en Visual Basic 6.0.  Reproducción de sonido. (DIRECCIONAL_DERECHA_DESACTIVADA)
<b>Private Sub Button22_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button22.Click</b>	GetAttr("C:\Users\Hogar\Desktop\TESIS\Led13\Prender Led 13\Prender_13.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\Led13\Prender Led 13\Prender_13.exe")  My.Computer.Audio.Play(My.Resources.DIRECCIONAL_IZQUIERDA_ACTIVADA, AudioPlayMode.Background)
<b>Botón 22 (Activar Direccional Izquierda)</b>	Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_13.exe), desarrollado en Visual Basic 6.0.  Reproducción de sonido. (DIRECCIONAL_IZQUIERDA_ACTIVADA)
<b>Private Sub Button24_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)</b>	GetAttr("C:\Users\Hogar\Desktop\TESIS\Led13\Apagar Led 13\Apagar_13.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\Led13\Apagar Led 13\Apagar_13.exe")  My.Computer.Audio.Play(My.Resources.DIRECCIONAL_IZQUIERDA_DESACTIVADA, AudioPlayMode.Background)

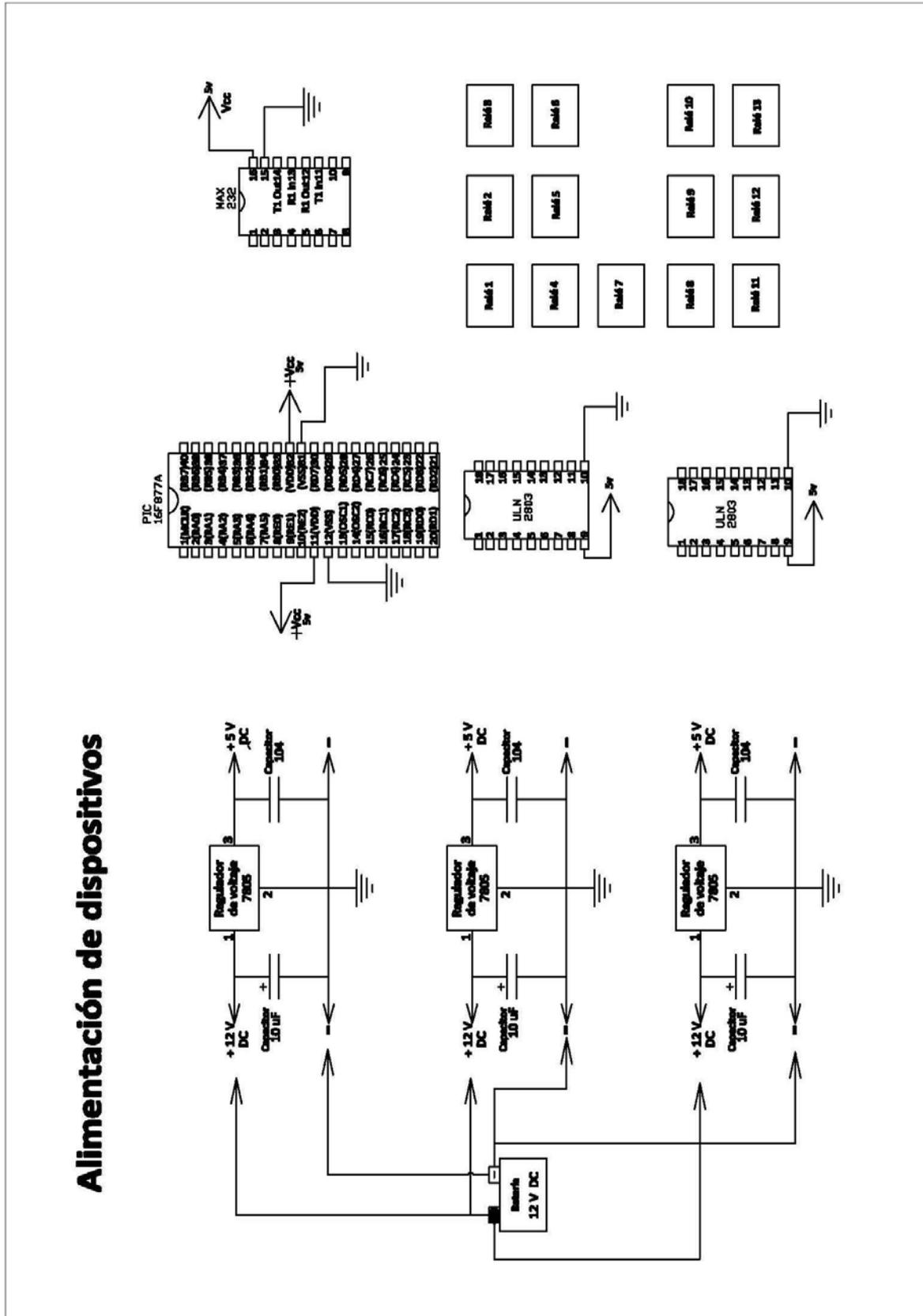
<b>rgs) Handles Button24.Click</b>	ERDA_DESACTIVADA, AudioPlayMode.Background)
<b>Botón 24 (Desactivar Direccional Izquierda)</b>	Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Apagar_12.exe), desarrollado en Visual Basic 6.0.  Reproducción de sonido.  (DIRECCIONAL_IZQUIERDA_DESACTIVADA)
<b>Private Sub Button21_Click (ByVal sender As System.Object, ByVal e As System.EventA rgs) Handles Button21.Click</b>	GetAttr("C:\Users\Hogar\Desktop\TESIS\Led12-13\Prender Led 12-13\Prender12-13.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\Led12-13\Prender Led 12-13\Prender12-13.exe")  My.Computer.Audio.Play(My.Resources.LUCES_INTERMITENTES_ACTIVADAS, AudioPlayMode.Background)
<b>Botón 22 (Activar Luces Intermitentes)</b>	Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_13.exe), desarrollado en Visual Basic 6.0.  Reproducción de sonido.  (LUCES_INTERMITENTES_ACTIVADAS)
<b>Private Sub Button23_Click (ByVal sender As System.Object, ByVal e As System.EventA rgs) Handles Button23.Click</b>	GetAttr("C:\Users\Hogar\Desktop\TESIS\Led12-13\Apagar Led 12-13\Apagar12-13.exe")  Shell("C:\Users\Hogar\Desktop\TESIS\Led12-13\Apagar Led 12-13\Apagar12-13.exe")  My.Computer.Audio.Play(My.Resources.LUCES_INTERMITENTES_DESACTIVADAS, AudioPlayMode.Background)
<b>Botón 23 (Activar Luces Intermitentes)</b>	Códigos que permiten tener la ruta de acceso y ubicación de archivo ejecutable (Prender_13.exe), desarrollado en Visual Basic 6.0.  Reproducción de sonido.

	(LUCES_INTERMITENTES_DESACTIVADAS)
<b>Private Sub Button4_Click( ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click</b>	End
<b>Botón 4 (Cerrar operaciones)</b>	Cierra la Hoja de diseño (Nro.2). Cierra la aplicación en general.

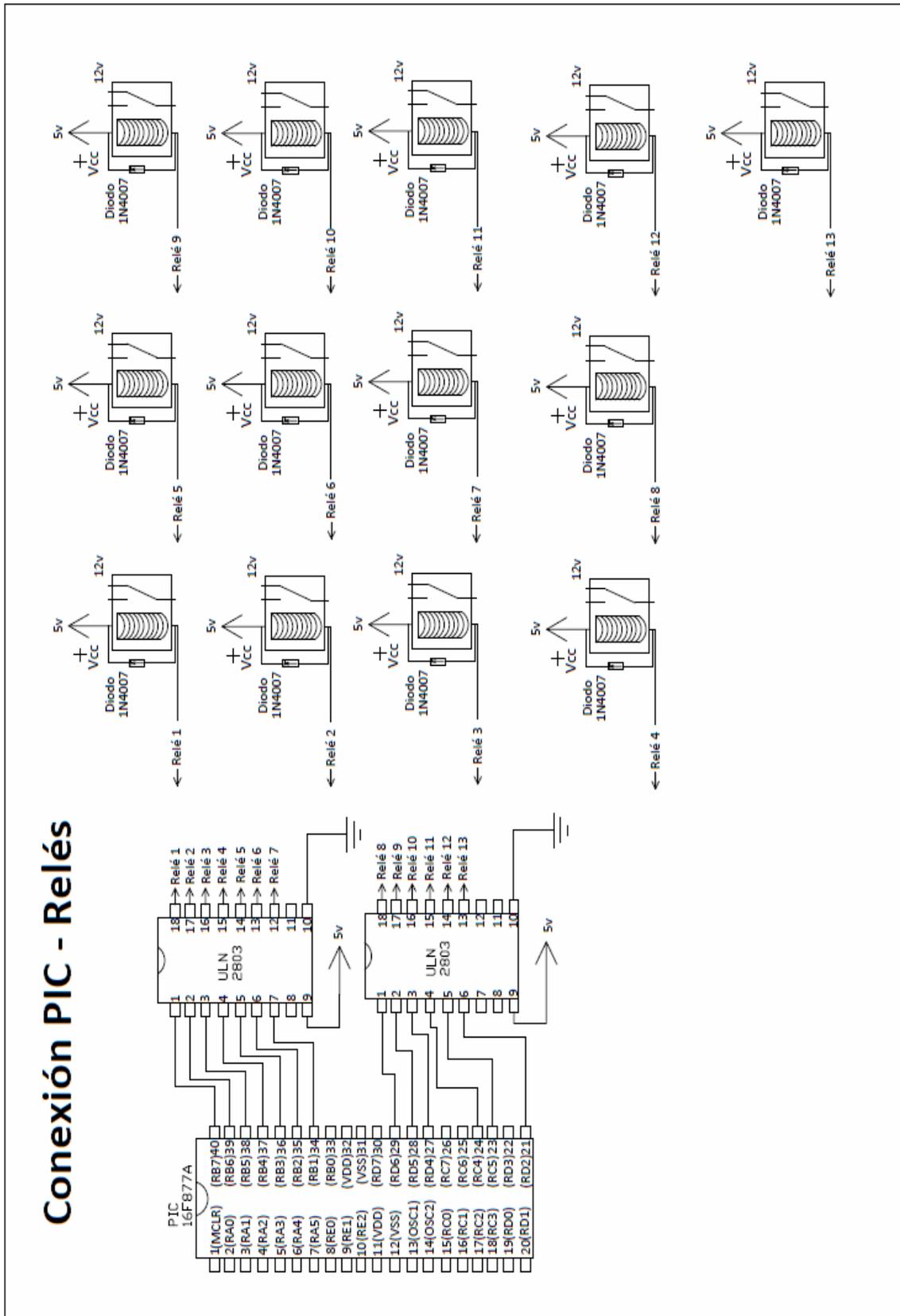
## Anexo 4. Diagrama eléctrico conexión PC - PIC



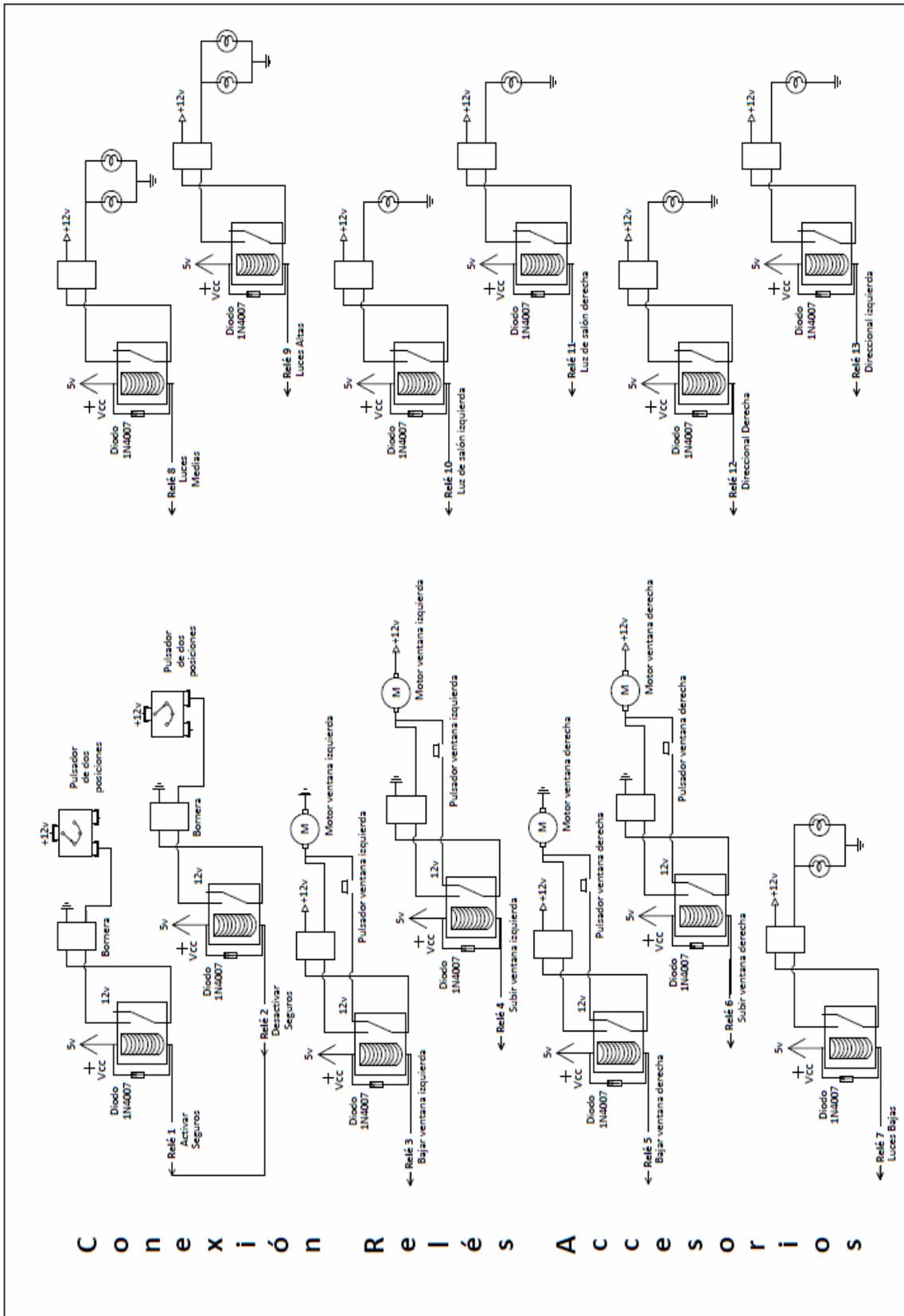
## Anexo 5. Diagrama de alimentación Placa PCB



## Anexo 6. Diagrama eléctrico conexión PIC - Relés



## Anexo 7. Diagrama eléctrico conexión Relés - Accesorios



# Anexo 8. Diagramas eléctricos de los accesorios del vehículo

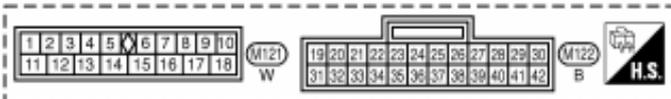
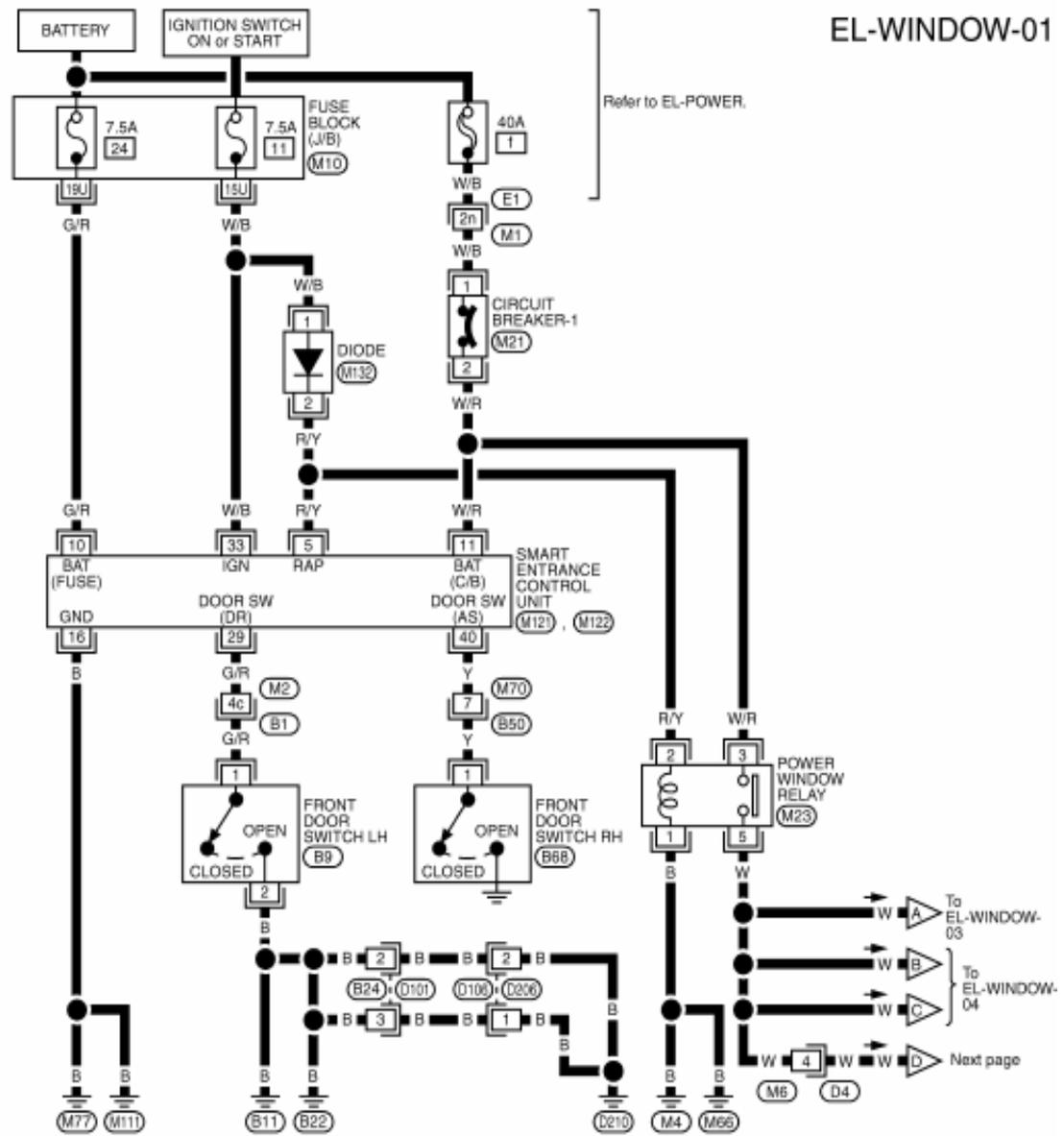
## POWER WINDOW

Wiring Diagram — WINDOW —

### Wiring Diagram — WINDOW —

RAEL0104

EL-WINDOW-01



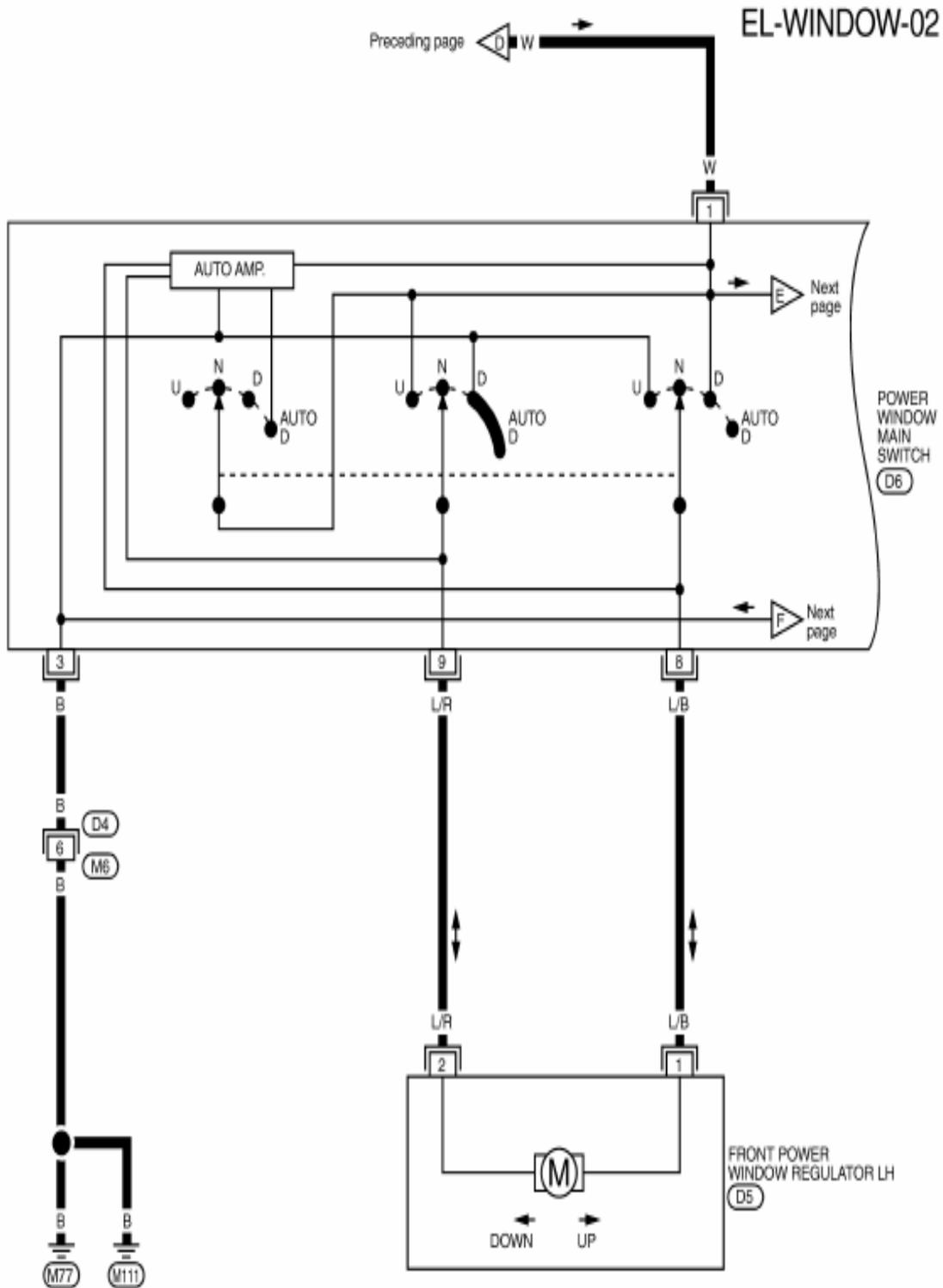
Refer to last page (Foldout page).

- M1 . E1
- M2 . B1
- M10

MFI 978.1

# POWER WINDOW

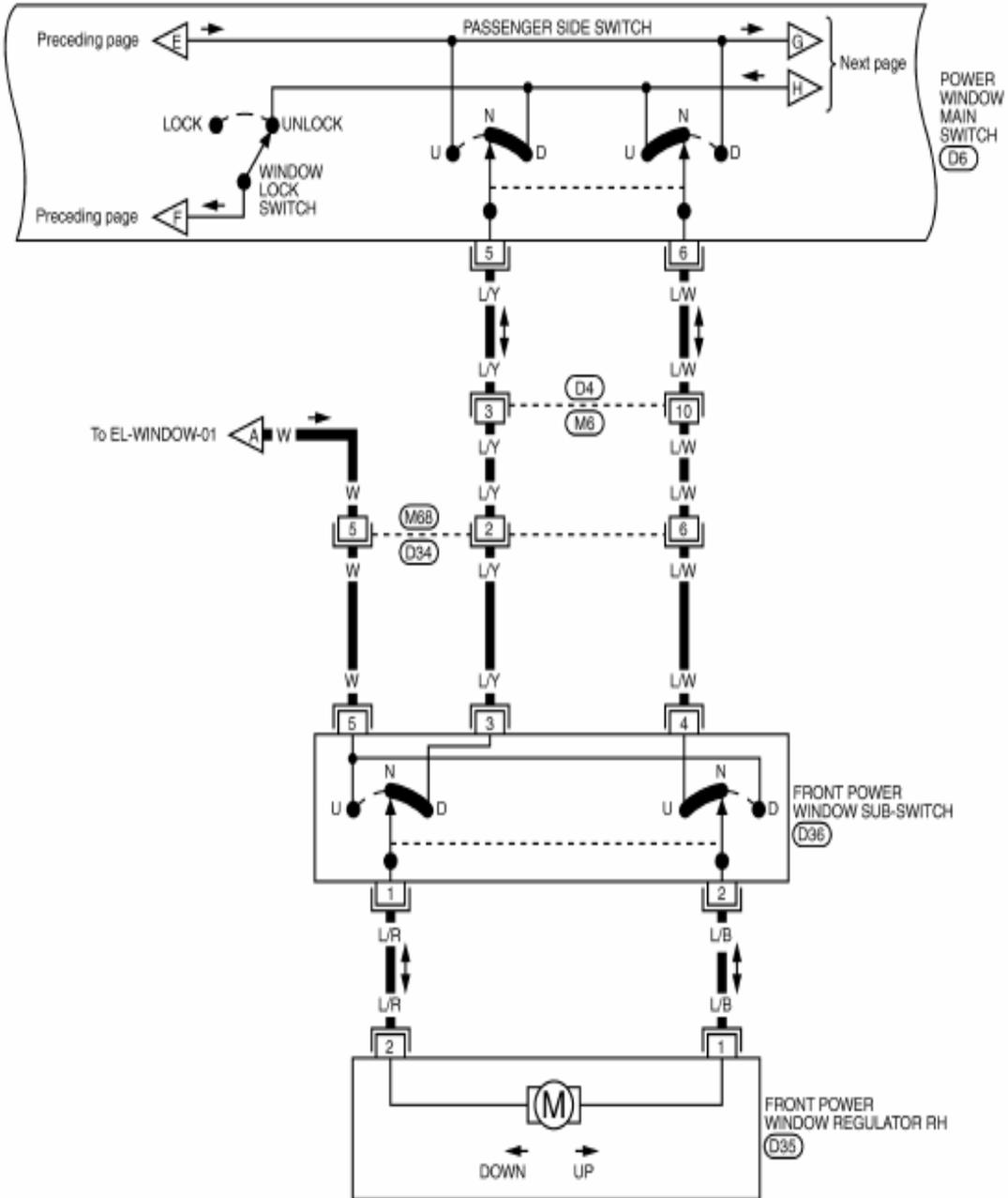
Wiring Diagram — WINDOW — (Cont'd)



# POWER WINDOW

Wiring Diagram — WINDOW — (Cont'd)

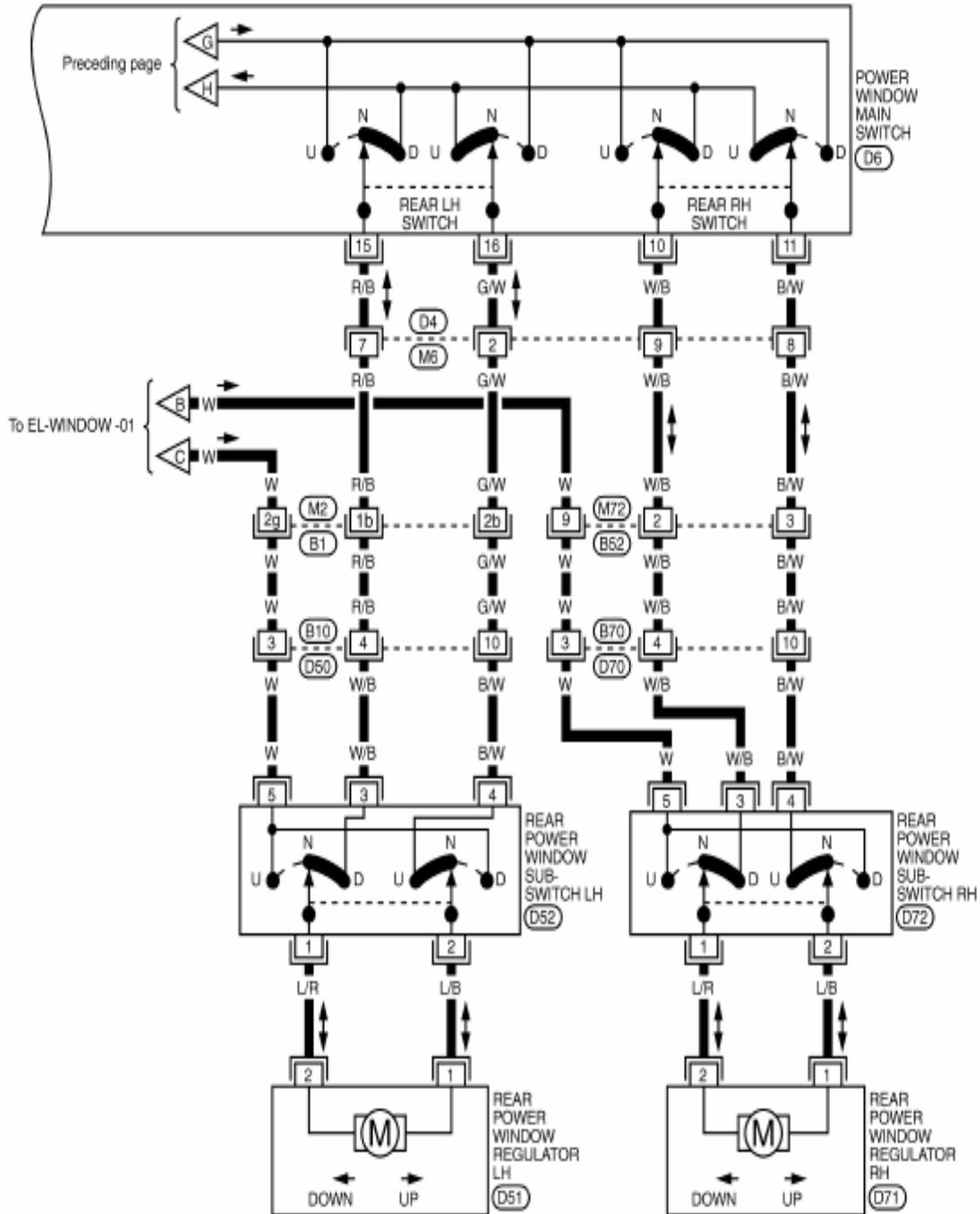
## EL-WINDOW-03



# POWER WINDOW

Wiring Diagram — WINDOW — (Cont'd)

## EL-WINDOW-04



Refer to last page (Foldout page).  
(M2), (B1)





# POWER DOOR LOCK

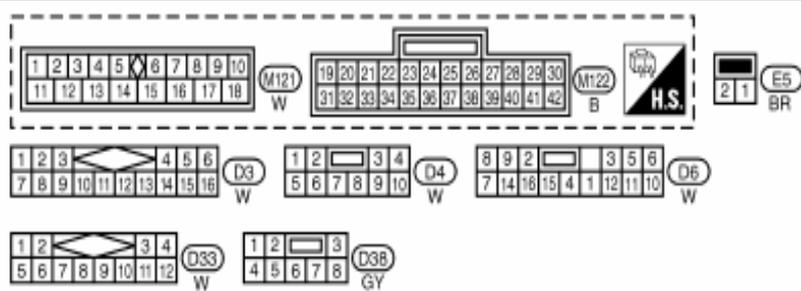
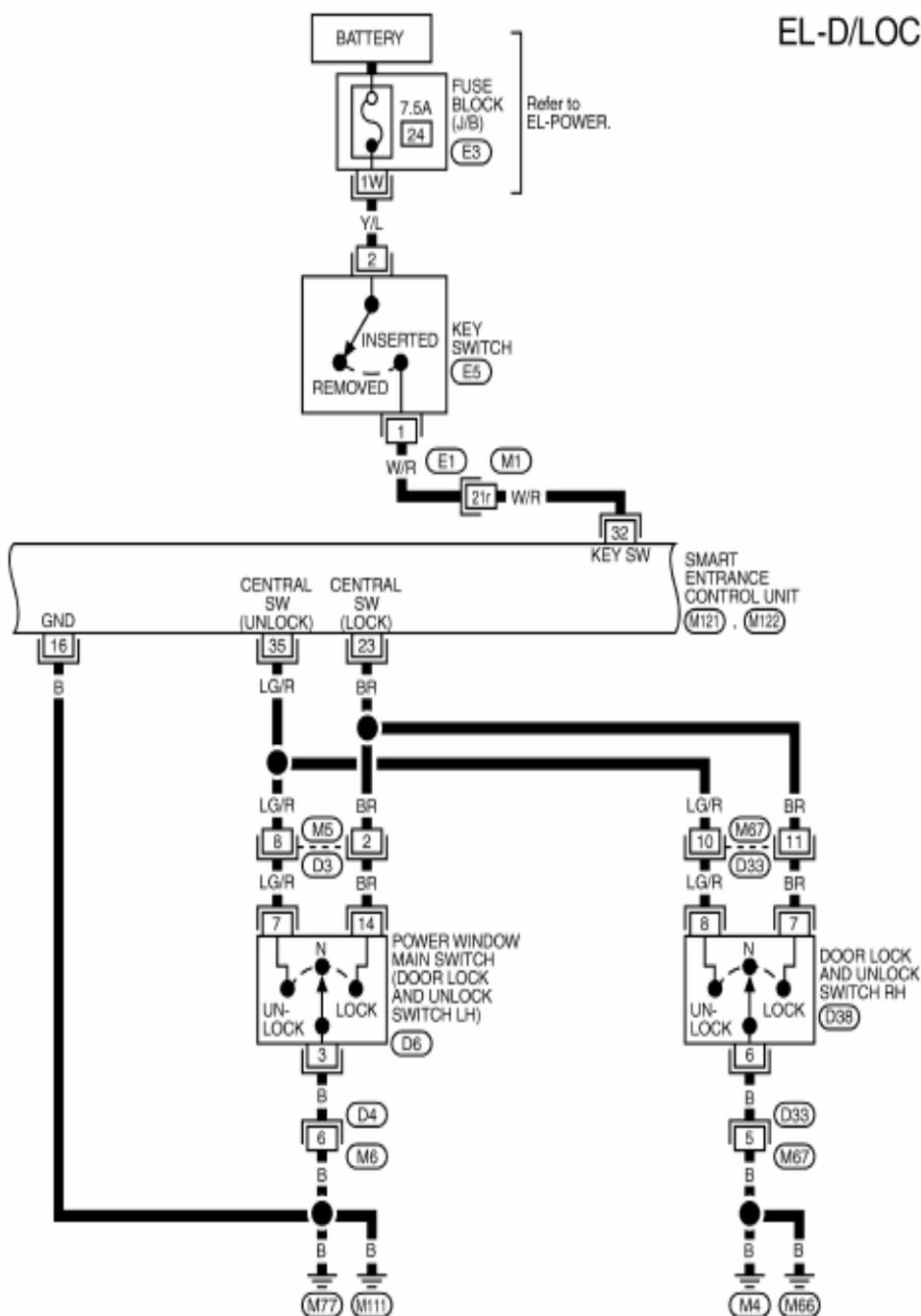


Wiring Diagram — D/LOCK — (Cont'd)

FIG. 2

NAE10109502

EL-D/LOCK-02



Refer to last page (Foldout page).





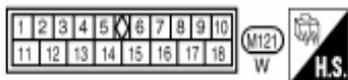
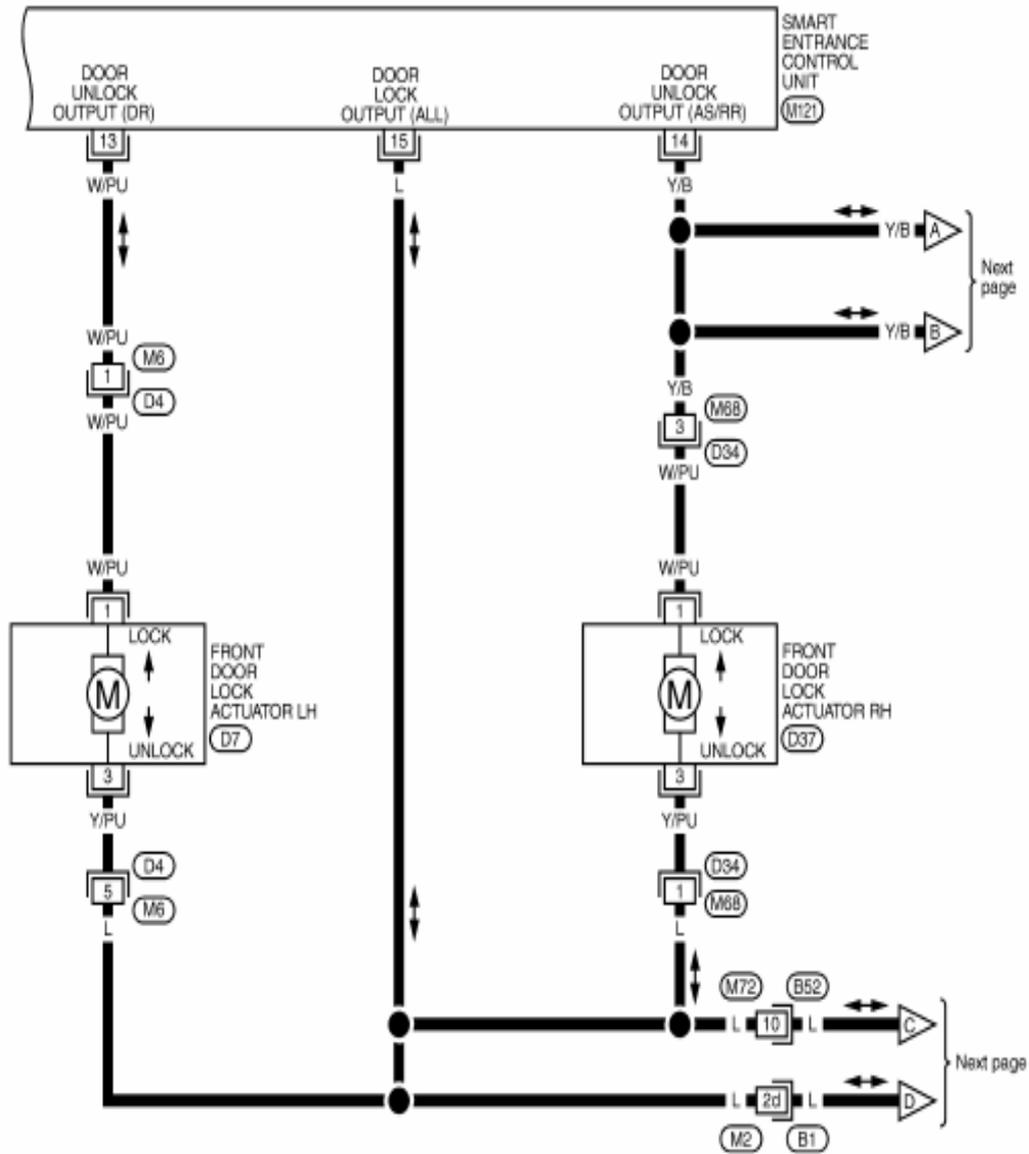
# POWER DOOR LOCK

Wiring Diagram — D/LOCK — (Cont'd)

FIG. 4

NAE10106504

EL-D/LOCK-04



Refer to last page (Foldout page).

M2, B1

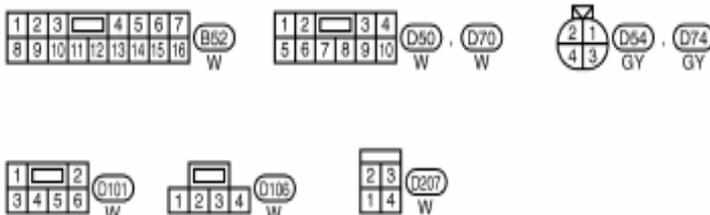
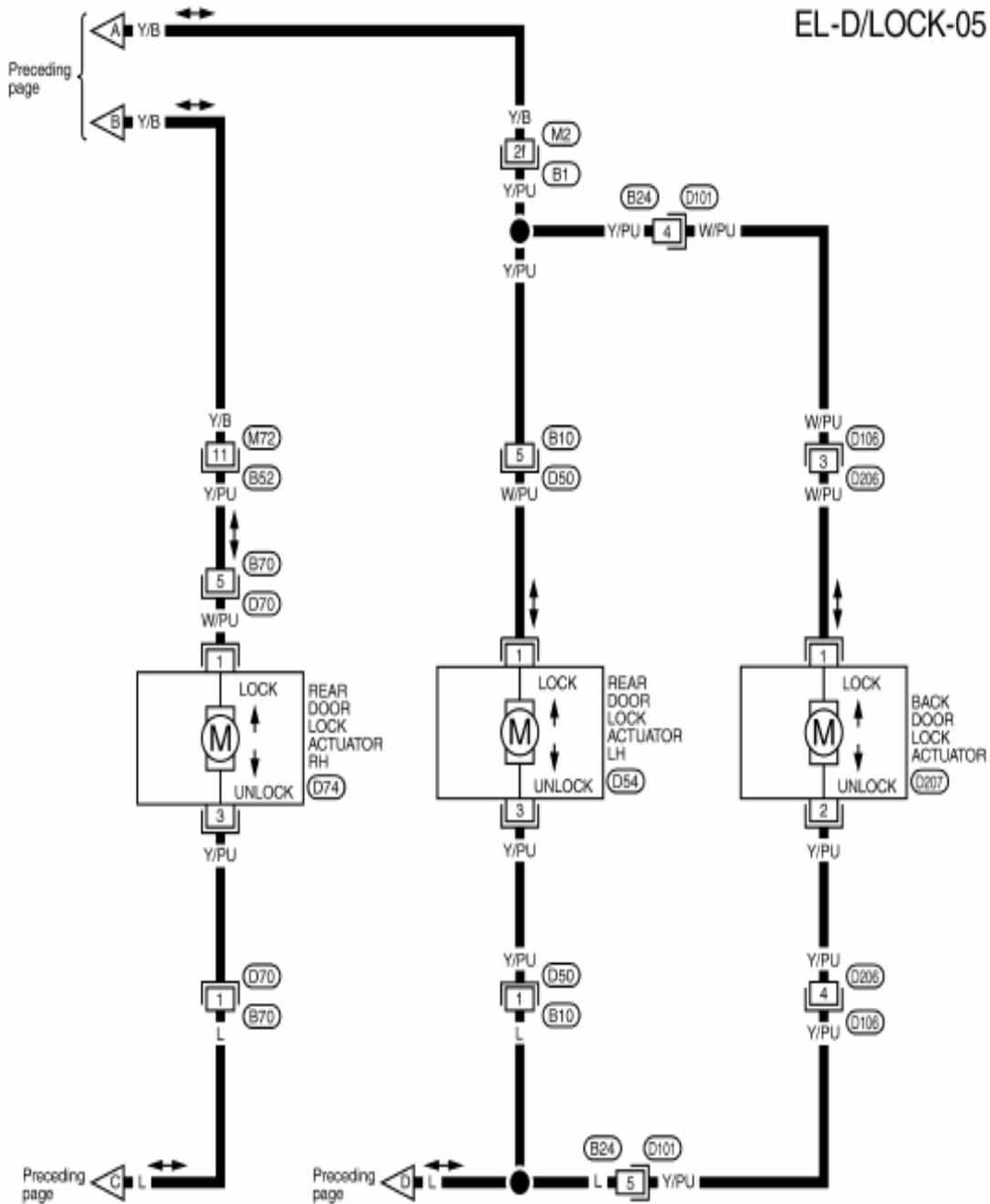
# POWER DOOR LOCK

Wiring Diagram — D/LOCK — (Cont'd)

FIG. 5

NA:EL0109205

EL-D/LOCK-05



Refer to last page (Foldout page).

(M2), (B1)