

T/629.8
T511d



**PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL ECUADOR
SEDE AMBATO
SERÉIS MIS TESTIGOS**

ESCUELA DE INGENIERIA DE SISTEMAS

Tema:

**DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE
CONTROL DE ENTRADA Y SALIDA DE VEHÍCULOS DEL
PARQUEADERO DE LA PUCESA**

Disertación de Grado

**Previo a la obtención del título de Ingeniero en Sistemas y
Computación**

Autor:

MARIA ISABEL TIRADO FIALLOS



Director :

ING. GALO LÓPEZ

Ambato – Ecuador

Febrero - 2010

De ingreso:	005348
Valor:	\$80,00
Manejo:	Donación: <input checked="" type="checkbox"/> Compra: <input type="checkbox"/>
Fecha de factura:	
Fecha de ingreso:	07052010

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

HOJA DE APROBACIÓN

Tema:

DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE ENTRADA Y SALIDA DE VEHÍCULOS DEL PARQUEADERO DE LA PUCESA

Autor:

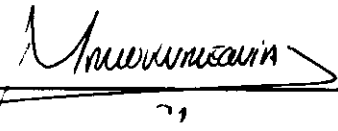
MARIA ISABEL TIRADO FIALLOS

Galo López, Ing.

f.  _____

DIRECTOR DE LA DISERTACION.

Marco Polo Silva, Ing.

f.  _____

CALIFICADOR.

Victor Chuncha, Ing.

f.  _____

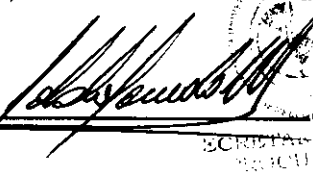
CALIFICADOR.

Santiago Acurio, Ing.

f.  _____

DIRECTOR ESCUELA DE SISTEMAS

Pablo Poveda, Ab.

f.  _____

SECRETARIO GENERAL PUCESA



SECRETARIA GENERAL
PUCESA

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, María Isabel Tirado Fiallos portador de la cédula de ciudadanía No. 180252267-0 declaro que los resultados obtenidos en la investigación que presento como informe final, previo la obtención del título de Ingeniero en Sistemas son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad legal y académica.



María Isabel Tirado Fiallos

CI. 180252267-0

AGRADECIMIENTO

Hay muchas personas a quienes deseo agradecer por ser parte de este logro pero primeramente deseo agradecerle a Dios, mi Señor, por estar siempre presente en mi vida y guiar mi camino día a día, sin Él nada de esto sería posible.

A mis padres, Jorge y Magdalena, por todo el amor que siempre me han dado, por su ejemplo de lucha y constancia y por todo el sacrificio que han hecho para que yo salga adelante. A mi hermano Jorge Luis por el apoyo incondicional y sobre todo por el ánimo que me ha dado para completar este trabajo. Gracias por ser mi familia.

A Israel, quien llegó a alegrar mi vida, por su ayuda, fuerza y empuje, por compartir conmigo todo lo que sabe.

A mi abuelita María y a mi tía Martha por el inmenso amor que siento de su parte.

A mi amigo Javier por colaborarme poniendo su tiempo y conocimientos desinteresadamente.

A mi director de tesis, Ing. Galo López por su guía y apoyo, así como también a mis profesores por compartir su conocimiento y experiencia semestre a semestre.

Quiero agradecer a mi primo Fernando Campaña quien, aun estando lejos, también ha puesto su granito de arena para que este proyecto se haga realidad.

DEDICATORIA

Este trabajo se lo dedico...

A Dios por ser mi fortaleza cuando más lo he necesitado...

A Dominic, con todo mi cariño aunque estés lejos...

A mis padres, familiares y amigos por creer en mí...

RESUMEN

En la presente disertación de grado se han puesto en práctica los conocimientos adquiridos de programación para la automatización del proceso de control de entrada y salida de vehículos de la Pontificia Universidad Católica del Ecuador – Sede Ambato.

El propósito principal es dar comienzo a que el control que se lleva del parqueadero de forma manual, se automatice.

El control se realizará mediante la lectura del código de barras del carnet de identificación de estudiantes, docentes y administrativos permitiendo así su ingreso o salida. Los usuarios del parqueadero deberán estar registrados en la base de datos cuya información será manejada por el software que ha sido desarrollado.

El software se ha desarrollado en el lenguaje de programación Visual Basic 6.0 del cual la universidad tiene licencia para su uso. La base de datos ha sido gestionada con MySql el cual es de código abierto permitiéndonos así administrar los datos sin necesidad de pagar por una licencia.

Mediante la combinación del software desarrollado, el circuito diseñado y los torniquetes que ya posee la universidad se podrá controlar el ingreso y salida de vehículos de las inmediaciones de la universidad. La finalidad es también facilitar el registro de los usuarios al inicio de cada semestre y durante el mismo obtener reportes de quienes están haciendo uso diario del parqueadero.

El resultado que se busca con esta tesis es mejorar el control del parqueadero de la universidad para que se convierta en un control fácil, moderno y de prevención de incidentes.

ABSTRACT

In this dissertation the acquired knowledge about programming has been put in practice to make automatic the process of controlling the entrance and exit of vehicles from the parking lot at the Pontificia Universidad Catolica del Ecuador – Sede Ambato.

The main purpose is to launch the start of an automated control of the parking lot which is manually managed.

The control will be made by reading the barcode on the ID card for students, faculty and administrative allowing their entry or exit. Users of the parking must be recorded in the database whose information will be handled by the software that has been developed.

The software has been developed in the programming language Visual Basic 6.0 for which the university has license. The database has been maintained with MySQL which is open source and allows us to manage data without paying for a license. By combining the developed software, designed circuit and the turnstiles which the university already has, the entry and exit of vehicles from the vicinity of the university can be controlled.

The aim of this dissertation is to facilitate the registration of users at the beginning of each semester and get reports from those who are making daily use of the parking lot.

The intended outcome of this thesis is to improve the control of the parking lot of the university to make it easy and modern and to prevent incidents.

Tabla de Contenidos

CAPÍTULO I	1
PROYECTO DE LA INVESTIGACIÓN	1
1.1. Introducción	1
1.2. Problema de investigación	1
1.2.1. Antecedentes	1
1.2.2. Significado del problema	3
1.2.3. Definición del problema	3
1.2.4. Planteamiento del tema	3
1.2.5. Delimitación del tema	3
1.3. Variables	4
1.3.1. Variables Independientes	4
1.3.2. Variables Dependientes	4
1.4. Objetivos	4
1.4.1. Objetivo General	4
1.4.1. Objetivos Específicos	4
1.5. Metodología de Trabajo	5
1.5.1. Investigación Explorativa	5
1.5.2. Investigación Descriptiva	5
1.5.3. Metodología de Programación	6
1.6. Justificación	6
CAPÍTULO II	7
MARCO TEÓRICO	7
2.1. Automatización	7
2.1.1. Inicios	8
2.1.2. Tipos de Automatización	9
2.1.3. Partes de la Automatización	11
2.2. Electrónica	12
2.2.1. Historia	13
2.2.2. Aplicación de la Electrónica	14
2.2.3. Circuitos Electrónicos	14

2.3. Visual Basic	17
2.3.1. Historia.....	17
2.3.2. Visual Basic 6.0	23
2.4. Código de Barras.....	33
2.4.1. Ventajas del Código de Barras.....	35
2.4.2. Estructura	36
2.4.3. Lectura del código de barras.....	36
2.5. Bases de Datos	38
2.5.1. Modelos de Bases de Datos	39
2.5.2. MySQL	41
2.3. Ingeniería de Software	44
CAPÍTULO III.....	46
ANÁLISIS Y DISEÑO DEL SISTEMA.....	46
3.1. ANÁLISIS ESTRUCTURADO DEL SISTEMA	46
3.1.1. Identificación de necesidades	46
3.1.2. Estudio de viabilidad.....	47
3.1.3. Análisis técnico y económico	48
3.1.4. Definición general del sistema.....	48
3.1.5. Análisis de requisitos del software	49
3.1.6. Diagramas de flujo de datos.....	51
3.1.7. Diagrama Entidad Relación	57
3.2. DISEÑO ARQUITECTÓNICO.....	58
3.2.1. DFD Nivel 2.....	58
3.2.2. Diseño Procedimental	61
3.2.3. Codificación del Sistema	69
3.2.4. Diseño del circuito electrónico	80
3.2.5. Elaboración del Circuito electrónico	83
3.2.7. Mantenimiento y Pruebas	88
CAPÍTULO IV.....	90
4.1. Conclusiones.....	90
4.2. Recomendaciones	91
Bibliografía	92
Consultada.....	92



Citada.....	92
Glosario.....	95
Anexo 1.....	98
Manual de usuario y de instalación.....	98
Anexo 2.....	120
Solicitud de Estacionamiento.....	120

Tabla de Gráficos

Figura 2.1. Automatización.....	7
Figura 2.2. Ejemplo de procesamiento electrónico de datos.....	10
Figura 2.3. Modelos de sensores.....	11
Figura 2.4. Ejemplo de interfaz gráfica.....	12
Figura 2.5. Circuito Electrónico.....	14
Figura 2.6. Entorno de desarrollo Visual Basic 1.0.....	19
Figura 2.7. Cuadro de propiedades de los controles de Visual Basic 2.0.....	20
Figura 2.8. Entorno de desarrollo de Visual Basic 3.0.....	21
Figura 2.9. Entorno de desarrollo de Visual Basic 4.0.....	22
Figura 2.10. Entorno de desarrollo de Visual Basic 5.0.....	23
Figura 2.11. Entorno de desarrollo de Visual Basic 6.0.....	24
Figura 2.12 Barra de menú de Visual Basic 6.0.....	25
Figura 2.13. Cuadro de controles de Visual Basic 6.0.....	26
Figura 2.14. Ventanas de proyecto y propiedades de Visual Basic 6.0.....	27
Figura 2.15. Editor de código de Visual Basic 6.0.....	28
Figura 2.16. Códigos de Barras.....	34
Figura 2.17. Ejemplo de código de barras 1D.....	34
Figura 2.18. Ejemplo de código de barras 2D.....	35
Figura 2.19. Estructura de un código de barras.....	36
Figura 2.20. Lectores de códigos de barras.....	37
Figura 2.21. Logo de MySQL.....	41
Figura 3.1. Ingreso de Usuario.....	64
Figura 3.2. Pantalla Principal.....	65
Figura 3.3. Control del Parqueadero.....	65
Figura 3.4. Usuarios del Sistema.....	66
Figura 3.5. Usuarios del Parqueadero.....	66
Figura 3.6. Búsqueda.....	67
Figura 3.7. Mensajes.....	67
Figura 3.8. Diagrama del Circuito Electrónico.....	80

Figura 3. 9. Circuito Electronico.....	80
Figura 3. 10. Pistas Para El Circuito Electronico.....	81
Figura 3. 11. Diagrama Software para diseño de Pistas	83
Figura 3. 12. Papel PnP.....	83
Figura 3.13. Cortar el diagrama de pistas	84
Figura 3. 14. Pistas en Papel PnP.....	84
Figura 3. 15 Limpieza de baquelita.....	84
Figura 3. 16. Fijar papel PnP en la baquelita.....	85
Figura 3. 17. Termotransferencia.....	85
Figura 3. 18. Remover Papel PnP de la baquelita.....	85
Figura 3. 19. Acido de Cloruro Férrico.....	86
Figura 3. 20. Pistas en la baquelita	86
Figura 3. 21. Limpieza de la baquelita.....	87
Figura 3. 22. Baquelita con pistas listas para perforar.....	87
Figura 3. 23. Circuito listo.....	87

Tablas

Tabla 2.1. Componentes de los circuitos electrónicos.....	17
Tabla 2.2. Tipos de datos de Visual Basic 6.0	30
Tabla 2.3. Controles de Visual Basic 6.0.....	33

Diagramas

Diag 1. Diagrama Entidad/Relación de base de datos relacional	40
Diag 2. Diseño lógico de base de datos jerárquica	40
Diag 3. Flujo de la información.....	49
Diag 4. DFD Nivel 0.....	51
Diag 5. DFD Nivel 1	52
Diag 6. DFD Nivel 2 Tipo de Usuario.....	53
Diag 7. DFD Nivel 2 Privilegios de Administrador	54
Diag 8. DFD Nivel 2 Privilegios de Operador.....	55
Diag 9. DFD Nivel 2 Selección acción a ejecutar	55
Diag 10. Entidad – Relación	57
Diag 11. Diseño Arquitectónico DFD Nivel 2.....	58

Diag 12. Diseño Arquitectonico Tipo de Usuario	59
Diag 13. Diseño Arquitectonico Privilegios Administrador.....	59
Diag 14. Diseño Aquitectónico Privilegios Operador	60
Diag 15. Diseño Arquitectonico Ejecutar Acción.....	60
Diag 16. Acceso al Sistema	61
Diag 17. Funcionamiento de el puerto paralelo en el sistema	62
Diag 18. Diagrama de Menú.....	68

CAPÍTULO I

PROYECTO DE LA INVESTIGACIÓN

1.1. Introducción

La automatización como su nombre lo indica, permite que los procesos y tareas se realicen de forma automática para lo cual debe tener una programación previa. Con la automatización se facilitan los trabajos haciendo que estos se realicen con mayor rapidez y en muchos casos con mayor eficacia que si lo realizaran las personas.

Los avances de la tecnología han permitido que se pueda contar con computadoras, sensores, controladores entre otros, para automatizar las tareas no solo en la industria sino también en la oficina y hasta en los hogares. Existen varias formas de automatizar entre las cuales se puede nombrar el control automático de procesos y el procesamiento electrónico de datos.

Con el propósito de aportar con algo al incremento de la seguridad en la universidad a través del presente proyecto, se busca hacer uso de la tecnología disponible en el medio para que la universidad pueda brindar un servicio aun mejor, de una forma actualizada. En este trabajo se pondrán en práctica los conocimientos adquiridos de programación, electrónica básica, administración de sistemas entre otros.

1.2. Problema de investigación

1.2.1. Antecedentes

La automatización es darle la habilidad a un conjunto de dispositivos tecnológicos de realizar las tareas que habitualmente las realizaba una o varias personas.

La automatización permite describir además a sistemas que no necesariamente son para la fabricación de un producto sino también para aparatos programados o que trabajen de forma automática independientemente de que una persona lo controle.

La automatización ha contribuido en el incremento de la producción, la reducción de costos, el incremento del tiempo libre de los trabajadores, la realización de trabajos que pueden ser de alto peligro para las personas de las empresas especialmente en países industrializados.

Se han generado varias polémicas debido a la automatización pues hay quienes han dicho que se puede llegar al punto de despedir al personal de las empresas para sustituirlas por una máquina, Otros afirman que en lugar de disminuir empleos, la automatización los incrementa y un claro ejemplo es la industria informática pues cada vez más empresas solicitan sus servicios ya sea para la fabricación, venta o mantenimiento de computadores y sistemas que automaticen los procesos.

En relación a la seguridad, la automatización permite implementar sistemas que controlen el ingreso o salida de un lugar sin que necesariamente una persona designada esté presente, esto puede ser posible con alarmas antirrobo, video vigilancia, etc.

Desde hace algún tiempo se ha buscado la manera de incrementar la seguridad, en la PUCESA por ejemplo, todo estudiante, docente o administrativo debe registrarse a inicio de semestre para obtener el sticker que le permite el ingreso al parqueadero, sin embargo no se tiene un control del ingreso y salida de los vehículos y además se da la posibilidad de que se presten los stickers y más de un automóvil ingresen con el mismo.

1.2.2. Significado del problema

Carencia de un proceso sistematizado para el control de entrada y salida de vehículos al parqueadero de la PUCESA

1.2.3. Definición del problema

¿Cómo puedo controlar de manera automática el torniquete de entrada al parqueadero?

¿Cómo funcionan los dispositivos eléctricos y electrónicos con una computadora?

¿Qué software permite desarrollar aplicaciones que controlen dispositivos externos?

¿Qué tipo de reportes son necesarios para obtener un control eficaz del parqueadero?

¿Cómo funcionan los sistemas de código de barra?

¿Qué dispositivos se necesitan para leer los códigos de barras?

1.2.4. Planteamiento del tema

“Desarrollo e Implementación de un Sistema de Control de Entrada y Salida de Vehículos del Parqueadero de la PUCESA”

1.2.5. Delimitación del tema

El sistema de control de entrada y salida de vehículos será implementado en el parqueadero de la PUCESA. El sistema permitirá que los estudiantes, docentes y administrativos utilicen su carnet de identificación para poder ingresar y salir. El sistema leerá el código de barra del carnet permitiendo o no el ingreso o salida al levantar de manera automática el torniquete. El sistema permitirá obtener reportes mensuales, semanales y diarios del número de vehículos que entran y salen, y de los nombres de los carnets que se leerán a través del sistema que tendrá una base de

datos donde se guardarán estos datos pudiendo el usuario solicitar los reportes. El sistema controlará el número de espacios disponibles dentro del parqueadero.

El tiempo de duración para el desarrollo de éste sistema será de siete meses a partir de la fecha de aprobación del tema por parte de las autoridades de la Escuela de Sistemas.

1.3. Variables

1.3.1. Variables Independientes

Número de vehículos que ingresan

Número de vehículos que salen

1.3.2. Variables Dependientes

Grado de satisfacción de los estudiantes, docentes y administrativos

Numero de espacios disponibles

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar e implementar un sistema de control de entrada y salida de vehículos del parqueadero de la PUCESA.

1.4.1. Objetivos Específicos

- Conocer cómo controlar los circuitos electrónicos por medio de una aplicación de software para levantar el torniquete de ingreso y salida de una forma automática.

- Incrementar la seguridad de la universidad por medio del control del ingreso y salida de los vehículos y con la obtención de reportes con número de vehículos que ingresan diario, semanal y mensual.
- Controlar el ingreso y salida de vehículos con la implementación de un lector de código de barras.
- Conocer el número de espacios disponibles en el parqueadero a través del control de vehículos que entran y salen del mismo.

1.5. Metodología de Trabajo

1.5.1. Investigación Explorativa

La investigación para esta disertación será explorativa pues será necesario integrarse en la PUCESA para conocer el proceso para obtener el permiso de ingreso y salida del parqueadero de la universidad. Es necesario conocer la estructura del área de ingreso y salida de vehículos y como se maneja en la actualidad para poder analizarlos y encontrar la mejora. Es indispensable conocer los dispositivos disponibles

1.5.2. Investigación Descriptiva

Como complemento a la investigación explorativa, se identificarán los elementos y características del ingreso y salida de vehículos del parqueadero, logrando así describir todo el proceso. Se hará una descripción de cuáles son las necesidades para que exista un mayor control. Igualmente necesario será identificar quienes serán los usuarios del sistema y que va a controlar.

1.5.3. Metodología de Programación

La metodología de programación será el modelo de prototipo pues ya se tienen identificados de una forma básica los objetivos de la aplicación. Además la parte física del ingreso y salida de vehículos ya existe por lo que se tendrá que hacer una adaptación del nuevo sistema a lo que ya se tiene disponible. El modelo de prototipos permite realizar cambios de acuerdo a las necesidades del usuario.

1.6. Justificación

Hacer uso de la tecnología y de los conocimientos adquiridos para incrementar la seguridad en la universidad, controlando el ingreso y salida de vehículos, para brindar un ambiente de mayor confianza a estudiantes, docentes y administrativos. La implementación de este sistema dará a la universidad un aspecto más actualizado.

Evitar que el parqueadero se llene con el ingreso de vehículos no autorizados o que ingresen dos o más vehículos con el mismo sticker. Evitar que personas no autorizadas saquen el vehículo del parqueadero. La necesidad de contar con reportes de los ingresos y salidas de los vehículos para fines estadísticos y para posibles mejoras en el servicio.

La tecnología para este sistema está disponible en el medio y no es de un costo muy elevado siendo posible su implementación.

CAPÍTULO II

MARCO TEÓRICO

2.1. Automatización

Automatización es una parte de la tecnología que busca la aplicación de sistemas mecánicos, electrónicos y de bases computacionales para operar y controlar las tareas y procesos. Sin embargo, la automatización no se trata únicamente de mecanizar los procesos sino de proveer a las personas mecanismos como ayuda en esfuerzo físico para depender menos del trabajo físico y mental del ser humano. Además, la automatización se usa también para sistemas de control y seguridad.

Objetivos de la automatización

- Minimizar el esfuerzo y tiempo de trabajo.
- Reducir la intervención del ser humano en los procesos.
- Evitar el error humano en los procesos.
- Aumentar seguridad.
- Mejorar la calidad del servicio o de los productos.



Figura 2.1. Automatización¹

¹ <http://alcorglobal.com.mx/productos-metrologic-honeywell/>

2.1.1. Inicios

Con la revolución industrial, se origina la automatización por los años 1750's. Este periodo de la historia tuvo lugar a mediados del siglo XVIII e inicios del siglo XIX primero en el Reino Unido y se expandió a otros lados de Europa dando lugar a las más grandes transformaciones de la historia tanto culturales, socioeconómicas y por supuesto tecnológicas. En esta etapa de la historia, la economía basada en la industria y la manufactura reemplazó a aquella basada en el trabajo manual. Es en esta época cuando se comenzaron a mecanizar las industrias textiles y a otras dando lugar a grandes incrementos en la producción. En el año 1801 la industria textil fue revolucionada con un telar automático manejado por tarjetas perforadas. Con las primeras máquinas se buscó suplantar un tipo de esfuerzo por otro que pudiera ser controlado por el ser humano como por ejemplo alzar artículos pesados con poleas. Luego las máquinas ya podían suplantar formas naturales de energía como el viento, el agua, etc.²

Hoy en día, la robótica industrial es una de las partes más notorias de la automatización. Este tipo de automatización, ha proporcionado ciertas ventajas como control de calidad, mayor eficiencia, repetitividad, integración de sistemas empresariales, incremento de productividad, etc. sin dejar de lado ciertas desventajas como es la necesidad de altas cantidades de dinero para implementarlo.

Hasta mitad el siglo XX, la automatización existía en pequeña escala, usando mecanismos simples, pero este concepto llegó a ser verdaderamente práctico con la evolución e implementación de la computadora y gracias a su flexibilidad permitió

² http://es.wikipedia.org/wiki/Revoluci%C3%B3n_industrial

controlar cualquier tarea añadiendo al concepto básico de automatización la posibilidad de realizar tareas repetitivas y especializadas, además de cálculos rápidos y almacenamiento de información, La computadora además permitió adicionar una interfaz entre el humano y la máquina para que éste la pueda manejar, ingresar datos o realizar algún cambio si es requerido.³

2.1.2. Tipos de Automatización

Existen diferentes formas de automatización de las cuales se deberá implementar la más adecuada de acuerdo a cada situación.

Los tipos de automatización son:

- **Automatización Fija.-** éste tipo de automatización emplea sistemas lógicos como por ejemplo compuertas lógicas. Como su nombre lo indica es relativamente inflexible a los diferentes cambios que se van dando aunque esto ha ido variando con la introducción de los Controladores Lógicos Programables (PLC)
- **Automatización Flexible.-** en éste tipo de automatización se pueden incluir los robots industriales que permiten trabajar y solucionar las variaciones en el diseño de los productos. Este tipo de automatización permite cambiar partes del programa o algo físico sin mayor pérdida de tiempo.
- **Procesamiento Electrónico de Datos.-** se relaciona con los sistemas de información y en general con los centros de cómputo. Hoy en día se suma a esto la obtención, el análisis y el registro de datos que se obtienen y se entregan a

³ <http://automatizacion2008.blogspot.com/>

través de computadores y sus respectivas interfaces. Para éste trabajo, es en parte el tipo de automatización que usará al recoger el número de cedula de los usuarios que ingresan al parqueadero a través de los lectores ópticos y los cuales serán registrados en la base de datos.

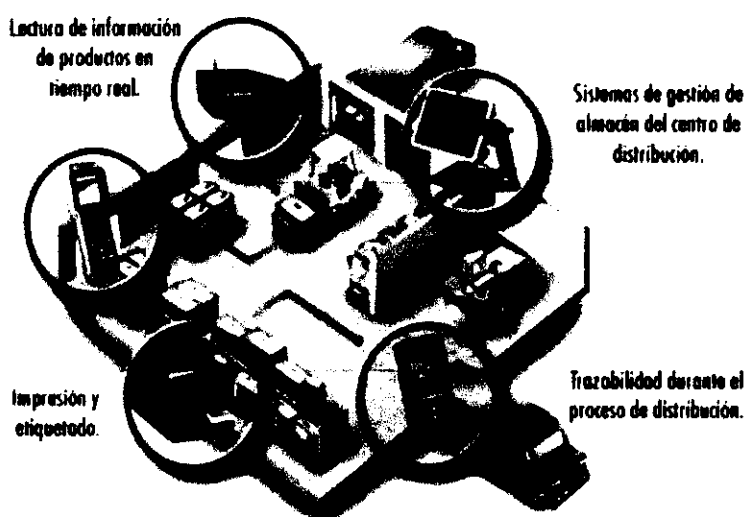


Figura 2.2. Ejemplo de procesamiento electrónico de datos⁴

- **Control Automático de Procesos.-** frecuentemente es usado para manejar procesos cuya característica principal son los diversos tipos de cambios que generalmente son cambios químicos o físicos, como por ejemplo los procesos de refinación del petróleo.
- **Control Numérico Computarizado.-** éste tipo se aplica en maquinas de herramientas de control numérico es decir aquellas que por ejemplo utilizan medidas exactas tales como fresadoras, tornos, maquinas de corte, etc.⁵

⁴ <http://www.sistemasintermec.com/empresa/>

2.1.3. Partes de la Automatización

Las partes de la automatización podrían variar de acuerdo al uso mismo pero en general son las siguientes:

2.1.3.1. Parte Operativa

La parte operativa es la parte tangible, lo que en el ser humano serian los músculos o articulaciones entre otros.

- **Proceso.-** son las operaciones que se hacen para obtener el producto
- **Sensores.-** captan la información precisa para la ejecución de los procesos, detecta temperaturas, posiciones, etc.



Figura 2.3. Modelos de sensores⁶

- **Interfaces.-** permite la comunicación entre la persona y el proceso, podría ser una interfaz gráfica de computadora, pulsadores, teclados, visualizadores, o como en este trabajo, los lectores ópticos.

⁵ <http://www.slideshare.net/wakuman/automatizacin>

⁶ <http://ees.wikispaces.com/transistores>

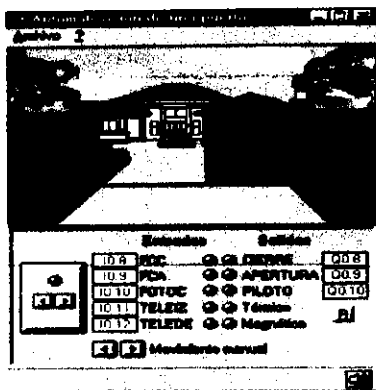


Figura 2.4. Ejemplo de interfaz gráfica⁷

- **Preaccionadores, Accionadores.-** son aquellos elementos que transforman las ordenes en acciones

2.1.3.2. Parte de Control

Si se lo compara el cuerpo humano, esta parte correspondería al cerebro que procesa los estímulos y da las órdenes a la parte operativa. Ésta parte lee variables y datos que recetan los sensores, transmite los estados de las variables, las analiza y compara para posteriormente enviar la orden adecuada a través de los accionadores para obtener para cumplir con el objetivo o producto preestablecidos.

2.2. Electrónica

La electrónica estudia los sistemas que funcionan en base a como se conduce y controla los electrones y otros elementos que se cargan de manera eléctrica en una unidad de tiempo para generar, transmitir, receptor y almacenar información. Esta información sufre cambios en sus características las mismas que se codifican y pueden ser amperaje, voltaje, frecuencia, etc. y cuando se alteran controladamente para ser conducidas en forma codificada dicha información recibe el nombre de

⁷ <http://www.autoware.com/spanish/gatesim.htm>

señales. La información podría ser por ejemplo una imagen, música o voz en un receptor de radio o números y datos en una computadora.

2.2.1. Historia

Se puede decir que los inicios de la electrónica se dió cuando John Ambrose Fleming creó el tubo de vacío propiciando un crecimiento rápido de la electrónica moderna. A partir de la invención de estos dispositivos, fue posible la manipulación de señales pudiendo por ejemplo amplificar las señales de radio y de sonido incluso superponer señales de sonido débiles a las ondas de radio. Luego para el año 1906 Lee De Forest inventó el triodo que básicamente es el mismo diodo pero con una rejilla adicional para variar la corriente. Este invento permitió fabricar los primeros televisores, amplificadores de sonido, y receptores de radio. A medida que pasaba el tiempo las válvulas de vacío se fueron perfeccionando y mejorando. Este desarrollo de tubos ayudó al fluido avance de la tecnología de comunicación radial y de las primeras computadoras.

En 1948, el transistor, aparecido de la mano de Bardeen y Brattain de la Bell Telephone permitió que se puedan hacer más pequeños los aparatos mejorando así su apariencia. A diferencia de los tubos de vacío, el transistor non funciona en vacío sino más bien en un estado sólido semiconductor que ya no necesita tantos voltios de tensión para trabajar. Este adelanto además benefició reduciendo costos, potencia y peso pero adicionando mayor eficacia.

En 1958 se desarrolló el primer circuito integrado, que constaba de seis transistores en un mismo chip. Para el año 1970 se desarrolló el primer microprocesador. Estos aparatos pueden integrar muchos transistores en un pequeño espacio dando paso a la

elaboración de circuitos electrónicos que hoy en día son parte de los ordenadores y demás equipos electrónicos.⁸

2.2.2. Aplicación de la Electrónica

La electrónica ayuda en una gran variedad de tareas siendo principalmente usada para controlar, procesar, distribuir información, además para convertir y distribuir la energía eléctrica. Los campos de uso principal son le electrónica de control, las telecomunicaciones y la electrónica de potencia.

2.2.3. Circuitos Electrónicos

Un circuito electrónico tiene varios componentes entre los cuales se puede nombrar las resistencias, inductancias, condensadores y otros que se conectan entre sí de forma eléctrica para generar, transportar o modificar las señales electrónicas.

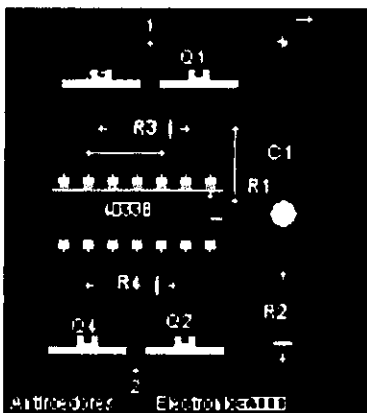


Figura 2.5. Circuito Electrónico⁹


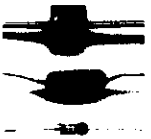




⁸ <http://www.monografias.com/trabajos5/electro/electro.shtml>





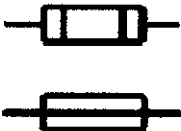
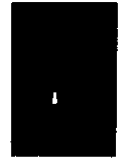
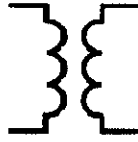





⁹ http://www.electronica2000.net/curso_elec/leccion70.htm

2.2.3.1. Componentes

Los componentes pueden ser de dos tipos, pasivos o activos. Los activos, cuando son expuestos a corriente o tensión, muestran amplificación (ganancia) o características de control como por ejemplo las baterías o pilas. Al contrario de los componentes activos, los componentes pasivos no tienen características de control o amplificación como por ejemplo las resistencias o los condensadores.

A continuación los componentes electrónicos más comunes:

Componente	Uso	Representación Simbólica	Imagen
Diodo	Rectificación de señales, regulación, multiplicador de tensión		
Relés	Apertura o cierre de circuitos mediante señales de control		
Resistencias	División de intensidad o tensión, limitación de intensidad		

Transistor	Amplificación, conmutación.		
DIAC	Control de potencia		
Fusible	Protección contra sobre-intensidades		
Transformador	Elevar o disminuir tensiones, intensidades, e impedancia aparente		
Interruptores	Apertura o cierre de circuitos manualmente		
Inductor	Adaptación de impedancias		

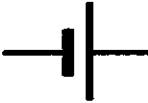

Pila	Generador de energía eléctrica		
------	--------------------------------	---	--

Tabla 2.1. Componentes de los circuitos electrónicos¹⁰

2.3. Visual Basic

Visual Basic es un lenguaje que permite desarrollar de manera visual, también se lo conoce como lenguaje de cuarta generación. Con un lenguaje de programación visual un gran número de tareas se las realiza sin escribir código sino con operaciones gráficas realizadas sobre la pantalla con el ratón.

Visual Basic constituye un entorno de desarrollo integrado o IDE que se conoce como un programa de aplicación que consta de un editor de código, un depurador, un compilador, y un constructor de interfaz gráfica.

2.3.1. Historia

2.3.1.1. Visual Basic 1.0

Para 1991 fue presentada la primera versión, Visual Basic 1.0, con la intención de hacer más simple la programación usando un ambiente de desarrollo completamente gráfico para hacer más fácil la creación de interfaces gráficas y también la programación en sí y su creador fue Alan Cooper. El equipo de Visual Basic 1.0, tuvo que eliminar algunas características de las muchas que tenían en mente para poder entregar el producto al mercado. Es así como ésta primera versión incluía solo algo más que la tecnología Embedded Basic que fue desarrollada en QuickBasic 4.0

¹⁰ <http://ees.wikispaces.com/transistores>

y un compilador simple diseñado para Windows 3.0, aunque nunca se utilizó para ello. Luego de 12 meses, se iniciaron mejoras la primera versión. Con el propósito de cubrir las exigencias del momento, Microsoft saca una herramienta de desarrollo llamado “Thunder”, nombre en clave.

Partiendo de un inicio poco favorable, se obtuvo un resultado difícil de entender: un gran incremento en el mercado de aplicaciones Windows y un gigantesco cambio en el desarrollo del software creando un impacto notorio en la industria de la informática. Cuando salió la primera versión, apenas un selecto grupo de personas podían desarrollar aplicaciones para Windows por eso Visual Basic constituyó el gran cambio en el desarrollo de software.

Visual Basic 1.0 se propagó rápidamente en la comunidad debido a sus capacidades e implementación. Después del boom inicial de este programa, un grupo de usuarios empezaron a crear componentes de Visual Basic o VBXs a partir de las bibliotecas de código que tenían con sus características, eventos y métodos. Luego el desarrollo de estos componentes que además eran reutilizables incrementó y pasó de un grupo especializado a toda una industria de vendedores de controles dando paso a que Visual Basic se convierta en un descubrimiento tecnológico.¹¹

¹¹ http://anylinda-12.blogspot.com/2007_09_01_archive.html

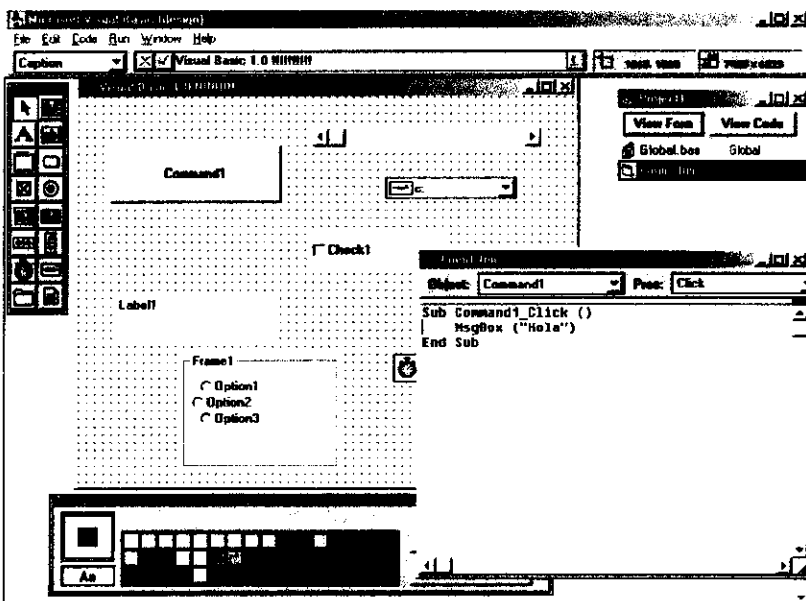


Figura 2.6. Entorno de desarrollo Visual Basic 1.012

2.3.1.2. Visual Basic 2.0

Pasando el tiempo y con el aumento de la demanda de Visual Basic era obvio que aquellos desarrolladores que lo utilizaban necesitaban un mejor y más capacitado Visual Basic. En 1992, Microsoft lanzó Visual Basic 2.0. Ésta nueva versión era mejorada y tenía mayor capacidad permitiendo crear aplicaciones de mayor tamaño y más sofisticadas. Tenía la capacidad de conectarse a bases de datos por medio de ODBC, incluía un ayuda para la depuración y la puesta a punto. La nueva versión contaba con nuevas herramientas como una ventana de propiedades, sintaxis del código en color y un soporte completo para MDIs (interfaz de múltiples documentos)

¹² <http://www.danielclemente.com/vb/>

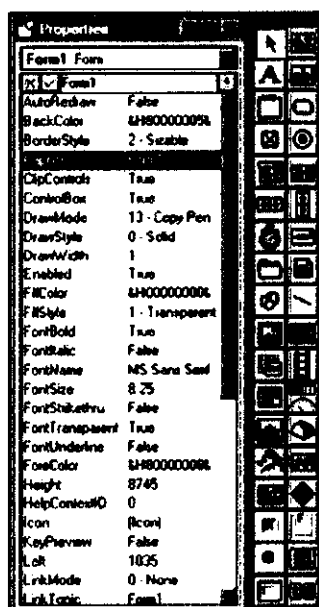


Figura 2.7. Cuadro de propiedades de los controles de Visual Basic 2013

2.3.1.3. Visual Basic 3.0

A medida que la adopción de Visual Basic se incrementaba en las corporaciones, también la necesidad de una herramienta que permita aplicaciones data-aware incrementaba. Luego de seis meses desde la salida de Visual Basic 2.0, aparece Visual Basic 3.0 con la solución a ésta necesidad al combinar el motor de base de datos Microsoft Access 1.1 con un conjunto de controles data-aware. Es así como los desarrolladores ya podían conectar a las bases de datos usando un diseñador visual en un ambiente cliente/servidor. Para complementar esto, los Data Access Objects u Objetos de Acceso a Datos, permitían el acceso a través de código a la base de datos. Visual Basic 3.0 amplió su capacidad de desarrollo al incluir los Crystal Reports que es un motor que en una variedad de formatos personalizables permite visualizar los datos extraídos de la base de datos.

¹³ http://personal.telefonica.terra.es/web/oscardmartinez/_articulos/articulo1-Historia-de-visual-basic.htm

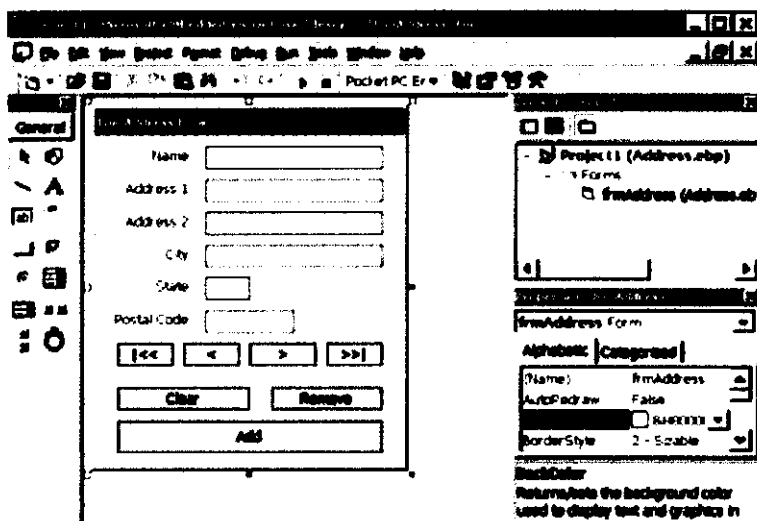


Figura 2.8. Entorno de desarrollo de Visual Basic 3.014

2.3.1.4. Visual Basic 4.0 - Herramienta más potente

En los años siguientes, se comenzó a trabajar en la programación en 32-bits siendo esto propiciado por la aparición de Windows 95 y Windows NT e hizo necesario que existan herramientas más potentes capaces de soportar la nueva arquitectura. Es así como se llegó a la conclusión de que migrar de un desarrollo en 16-bits a un desarrollo en 32-bits sería complejo pero haría que Visual Basic siga en el mercado por largo tiempo. En septiembre de 1995, fue anunciada la versión 4.0 de Visual Basic. Esta nueva versión incluía tanto la versión estándar como la empresarial. La versión empresarial tenía nuevos adelantos como control remoto de datos, automatización remota, y Microsoft SourceSafe para dirigir la configuración.¹⁵

¹⁴ <http://www.microsoft.com/Korea/MSDN/MSDNMAG/ISSUES/2001/cetools/default.aspx>

¹⁵ http://anylinda-12.blogspot.com/2007_09_01_archive.html

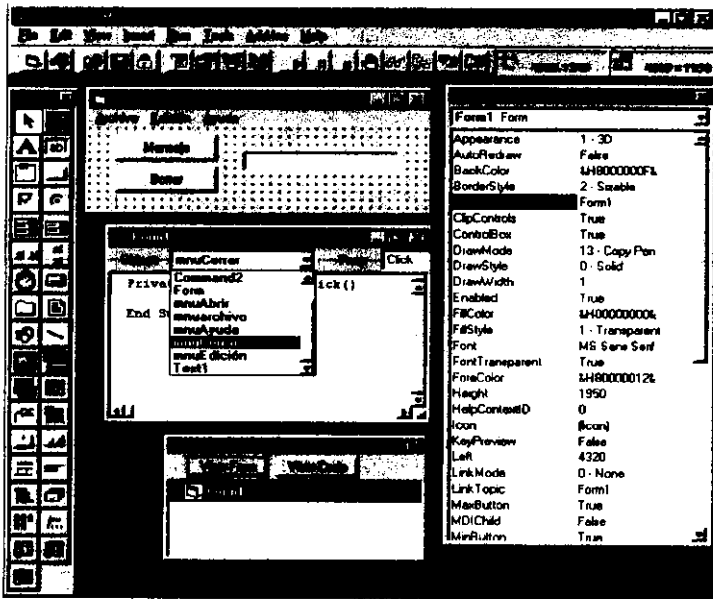


Figura 2.9. Entorno de desarrollo de Visual Basic 4.016

2.3.1.5. Visual Basic 5.0 y 6.0

Anunciadas en marzo de 1997 y en junio de 1998, las versiones 5.0 y 6.0 de Visual Basic dieron un paso sustancial para permitir a los desarrolladores programar en el nuevo campo que representaba el Internet. El compilador nativo incrementó el funcionamiento de manera gigantesca. El Diseñador de Clases Web (Webclass Designer) hizo más simple la programación de aplicaciones Web facilitando un modelo intuitivo para el servidor web. El diseñador de páginas, DHTML, permito crear aplicaciones para el navegador de internet de Microsoft el Explorer 4.0. También con estas versiones, los programadores podían construir con facilidad los controles ActiveX , destinados a internet, con el Control Creation Edition (Edición de Creación de Controles).¹⁷

¹⁶ <http://www.gui.uva.es/login/login/15/visual.html>

¹⁷ http://anylinda-12.blogspot.com/2007_09_01_archive.html

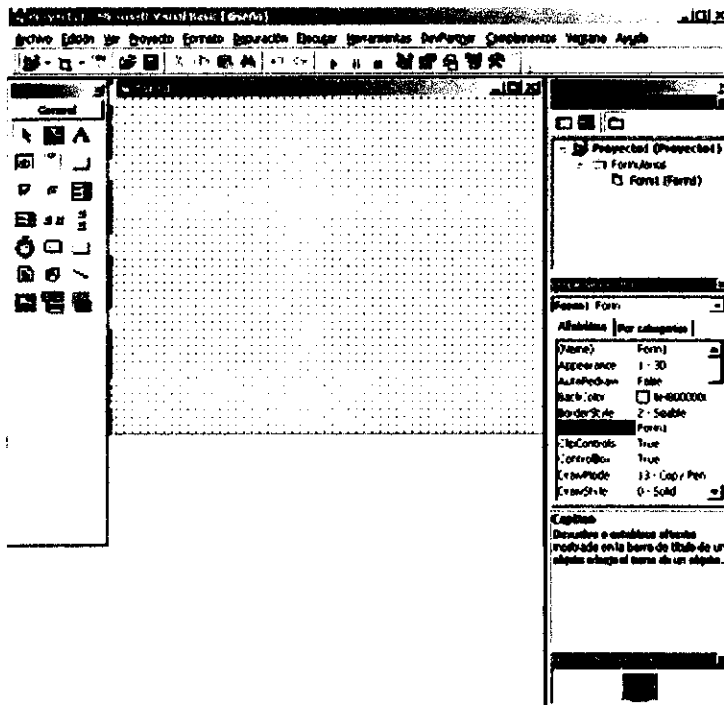


Figura 2.10. Entorno de desarrollo de Visual Basic 5.018

2.3.2. Visual Basic 6.0

Visual Basic 6.0 es una herramienta de programación para crear aplicaciones para Windows. Las aplicaciones pueden ser representaciones de movimientos mecánicos, funciones gráficas o matemáticas, simulación, etc. Visual Basic permite crear botones, ventanas, menús o cualquier elemento de Windows.

Visual Basic 6.0 es un programa basado en objetos y no orientado a objetos pues éste usa objetos que tienen propiedades y métodos pero no maneja lo que es polimorfismo y herencia que son características propias de los lenguajes orientados a objetos

¹⁸ http://anylinda-12.blogspot.com/2007_09_01_archive.html

2.3.2.1. Entorno de desarrollo

Al arrancar Visual Basic 6.0 se tiene en la pantalla una configuración como la que se muestra a continuación.

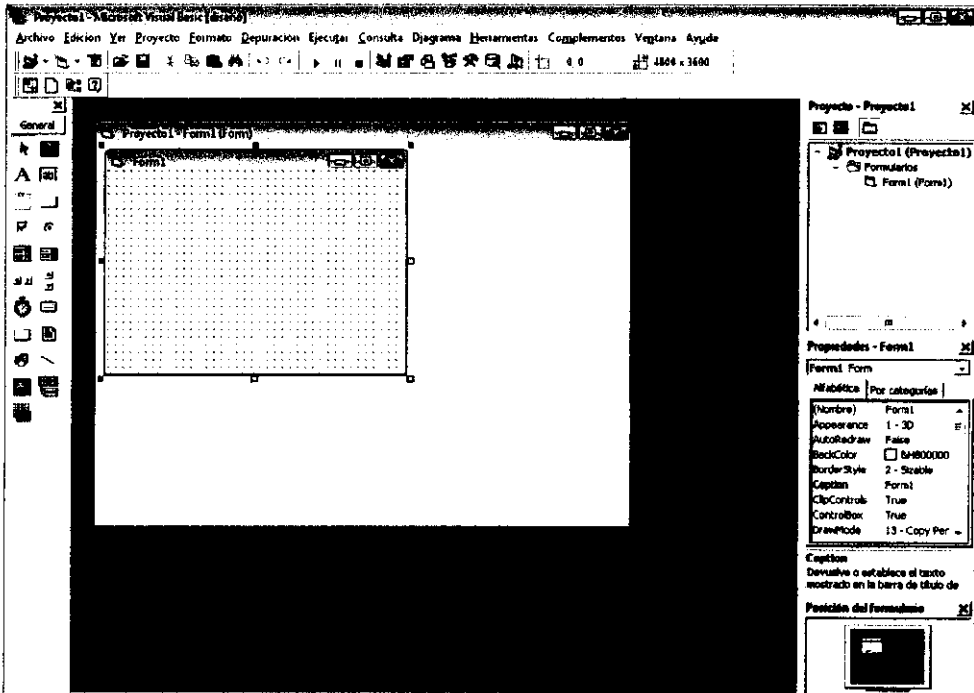


Figura 2.11. Entorno de desarrollo de Visual Basic 6.019

- **Barra de Menús y Barra de Herramientas**

La barra de menús de Visual Basic 6.0 es similar a otras aplicaciones de Windows. Debajo de la barra de menús se encuentran las barras de herramientas, con varios botones permitiendo acceder a las diferentes opciones de los menús. Visual Basic 6.0 tiene cuatro barras de herramientas: Depuración, Edición, Editor de formularios y Estandar. Por defecto sólo aparece la barra Standard, aunque en la Figura 2.2 se muestran las cuatro. Clicando con el botón derecho sobre cualquiera de las barras de herramientas aparece un menú contextual con el que se puede hacer aparecer y

¹⁹ Investigador

ocultar cualquiera de las barras. Al igual que en otras aplicaciones de Windows 95/98/NT, también pueden modificarse las barras añadiendo o eliminando botones (opción Customize). Son de gran importancia los botones que permiten arrancar o parar la ejecución de un proyecto, pasando de modo diseño a modo de ejecución y viceversa.

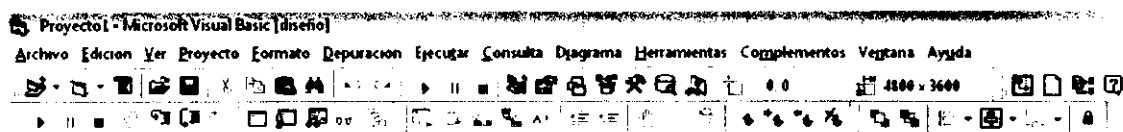


Figura 2. 12 Barra de menús de Visual Basic 6.020

- **Herramientas (toolbox)**

La caja de herramientas contiene los controles para poder diseñar la pantalla de la aplicación. Entre éstos controles están botones, etiquetas, cajas de texto, zonas gráficas, etc. Para poner un control en el formulario se hace clic en el icono deseado del cuadro de herramientas y se lo coloca en el formulario en la posición y con el tamaño deseado, haciendo clic y arrastrándolo con el ratón.



Figura 2.13. Cuadro de controles de Visual Basic 6.021

- **Ventana de Proyecto y Ventana de Propiedades**

Esta ventana permite acceder a los formularios así como a los módulos que hacen el proyecto. En ella se puede ver la parte gráfico de los formularios (botón View Object), y también se puede entrar a la parte en la que se edita el código que contienen (botón View Code).

La ventana de propiedades permite ver lo que todos los objetos de Visual Basic 6.0 tienen como son su nombre (Name), título (Caption), el texto (Text), tamaño, posición, color entre otros. Estas propiedades pueden ser cambiadas en la ventana de propiedades o directamente en el código.

²¹ Investigador

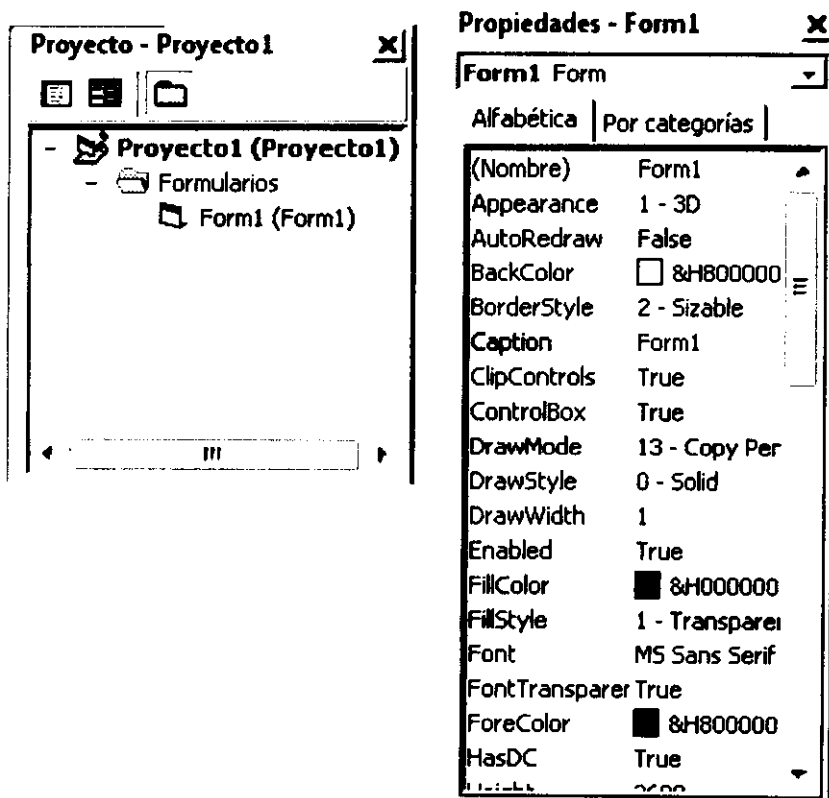


Figura 2.14. Ventanas de proyecto y propiedades de Visual Basic 6.022

- **Editor de Código (Code Editor)**

En el editor de código se escriben las sentencias del programa. Para abrir la ventana del editor de código se abre haciendo clic en el botón View Code de la Ventana del Proyecto, también eligiendo Code en el menú View o haciendo doble clic en el formulario o en cualquiera de sus controles. Cada formulario, cada módulo de clase y cada módulo estándar tienen su propia ventana de código. En el editor de código se puede notar que el programa usa un código de colores para distinguir los elementos del programa. Por ejemplo lo que escribe el programador aparece en letras negras y el lenguaje propio de Basic en color azul.

```

Project1 - Form1 (Código)
Command1 Click
Private Sub Command1_Click()
    Dim i, n, sum As Integer
    n = Val(Text1.Text)
    sum = 0
    For i = 1 To n - 1
        If (n Mod i = 0) Then
            sum = sum + 1
        End If
    Next i
    If sum = n Then
        Label1.Caption = "Es perfecto"
    Else
        Label1.Caption = " No es perfecto"
    End If
End Sub

Private Sub Command2_Click()
    Form6.Show
    Form1.Hide
End Sub

```

Figura 2.15. Editor de código de Visual Basic 6.023

2.3.2.2. Conceptos Básicos de Visual Basic 6.0

- **Proyectos y Módulos**

Un proyecto es el conjunto de los ficheros y modulos que se necesitan para que el programa que se realizó funcione. Toda la información que se refiere a aquellos ficheros son almacenados en uno del tipo *ProjectName.vbp*, la extensión *.vbp* hace referencia a proyecto de Visual Basic.

Los módulos por su parte son parte del proyecto y pueden ser de varios tipos:

*.*frm.*- asociados a un formulario y a toda la información referente a sus controles.

*.*bas.*- contienen únicamente código Basic.

*.*cls.*- definen agrupaciones de código y datos que se denominan clases.

- **Variables**

Una variable es una estructura que recibe un nombre único y contiene datos, mantiene estos datos hasta que se le asigne un nuevo valor o hasta que el programa termine.

Ámbito de las variables

El ámbito de las variables se refiere a la parte de aplicación donde se puede acceder a la variable, en Visual Basic estas pueden ser:

- **Dim:** Es una variable de ámbito local pudiendo estar dentro de un procedimiento o un formulario y es accesible solo desde este lugar.
- **Public:** A éstas variables se podrá acceder desde todos los formularios de la aplicación. Para conseguirlo se deberá declararlas en un módulo de código, no en la sección de declaración de cualquier formulario que tenga la aplicación. Para esto será necesario crear un módulo de código insertándolo en el proyecto.
- **Static:** Al declarar estas variables se le dice al programa que no se creen ni se destruyan al entrar o salir de los procedimientos donde se crearon sino que mantenga el valor mientras la aplicación este siendo ejecutada.

Tipos de datos

Como en otros lenguajes Visual Basic tiene varios tipos de datos como se muestra a continuación en la tabla

Tipo	Descripción	Carácter de declaración	Rango
Boolean	Binario		True o False
Byte	Entero corto		0 a 255
Integer	Entero (2 bytes)	%	-32768 a 32767
Long	Entero largo (4 bytes)	&	-2147483648 a 2147483647
Single	Real simple precisión (4 bytes)	!	-3.40E+38 a 3.40E+38
Double	Real doble precisión (8 bytes)	#	-3.40E+38 a 3.40E+38
Currency	Número con punto decimal fijo (8 bytes)	@	-9.22E+14 a 9.22E+14
String	Cadena de caracteres (4 bytes + 1 byte/car hasta 64 K)	\$	0 a 65500 caracteres.
Date	Fecha (8 bytes)		1 de enero de 100 a 31 de diciembre de 9999. Indica también la hora, desde 0:00:00 a 23:59:59.
Variant	Fecha/hora; números enteros, reales, o caracteres (16 bytes + 1 byte/car. En cadenas de caracteres)		F/h: como Date números: mismo rango que el tipo de valor almacenado

Tabla 2.2. Tipos de datos de Visual Basic 6.024

²⁴ http://anylinda-12.blogspot.com/2007_09_01_archive.html

- **Eventos**

Eventos Generales

Cuando se arranca una aplicación se producen varios: *Initialize, Load, Activate* y *Paint*. Al ocultar, cerrar o eliminar un formulario se producen otra serie de eventos: *Deactivate, QueryUnload, Unload* y *Terminat*.

El foco (focus), En todas las aplicaciones de Windows, siempre hay un único formulario, control o ventana que está listo para recibir entradas desde teclado en ese momento ese formulario, control, o ventana es el que dispone del “foco” (focus).

El evento *KeyDown* se produce cuando el usuario pulsa una tecla y el evento *KeyUp* cuando suelta la tecla. El evento *KeyPress* sucede cuando el usuario pulsa y suelta la tecla. Con el evento *KeyAscii* se conoce el código ASCII de la tecla presionada.

Eventos vinculados con el ratón

El evento *Click* se activa cuando el usuario pulsa y suelta rápidamente uno de los botones del ratón. El evento *DbClick* se activa al hacer clic dos veces seguidas.

El evento *MouseUp* sucede al soltar un botón que había sido pulsado El evento *MouseDown* al pulsar el botón del mouse. El evento *MouseMove* ocurre cuando se mueve el ratón sobre un formulario o sobre un control.


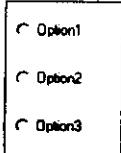
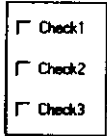
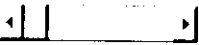
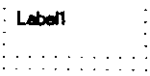

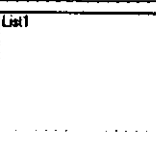
- **Propiedades comunes en varios controles**

A continuación algunas propiedades comunes en muchos controles:

- Name.- Nombre del objeto. Todos los objetos deben tener un nombre para poder referirse al momento de programar.
- Appearance.- Indica si un objeto tiene un aspecto plano (valor 0) o tridimensional (valor

- BackColor: Color de fondo de un objeto.
 - Caption: Es el título del objeto o el texto que aparece en el.
 - Enabled: Indica si el objeto es accesible y modificable o no.
 - Font: El tipo de letra del objeto.
 - ForeColor: Indica el color del texto y/o gráficos de un objeto.
 - Left y Top: Establecen la distancia horizontal y vertical entre el origen del objeto que lo contiene y el origen del control.
 - Visible: Si el objeto es visible o invisible.
- **Controles**

La siguiente tabla muestra algunos de los controles más usados en las aplicaciones de Visual Basic 6.0

Control	
Command Button (Botón de comando)	
Option Button (Botones de opción)	
Check Box (Botones de selección)	
Scroll Bars (Barras de desplazamiento)	
Labels (Etiquetas)	
Text Box (Cajas de texto)	
List Box (Listas)	

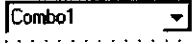
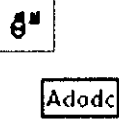
ComboBox (Cajas combinadas)	
Ado Data Control	

Tabla 2.3. Controles de Visual Basic 6.025

2.4. Código de Barras

El código de barras nació en los años 70 para identificar los productos comercializados en las tiendas. El código de barras es un código que se representa a través de un conjunto de líneas paralelas verticales de diferente grosor y distinto espaciado que en conjunto guardan una determinada información. Así con el código de barras se puede reconocer con rapidez un artículo o identificar una persona pudiendo además consultar las características asociadas al mismo. Hoy en día los códigos de barras y los lectores se han difundido en grandes proporciones a nivel mundial y son usados en la mayoría del sector productivo y administrativo.

Los códigos de barras tienen muchas aplicaciones como por ejemplo control de tiempo y asistencia, control de inventario, control de acceso, embarques y recibos, control de documentos, facturación, bibliotecas, etc.²⁶

²⁵ Investigador

²⁶ http://codigodebarras.com/tema.php?ID=como_nacio

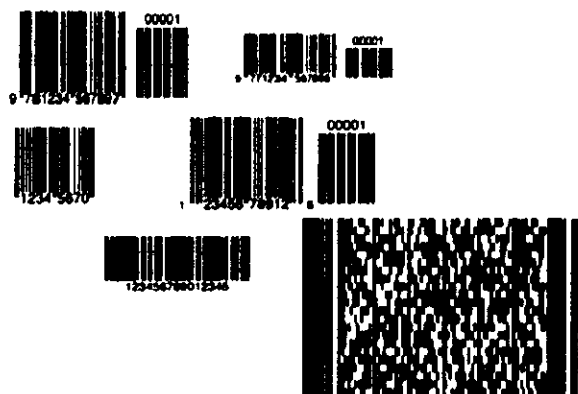


Figura 2.16. Códigos de Barras²⁷

El mapeo entre la información y el código que la representa se denomina simbología y se clasifica en:

Códigos de barras 1D: Almacenan códigos, ya sean números o letras, combinando líneas y espacios en diferentes medidas. Puede comenzar con una línea o con un espacio.²⁸



Figura 2.17. Ejemplo de código de barras 1D²⁹

Códigos de barras 2D: Los datos son almacenados tanto a lo alto como a lo ancho del símbolo; las barras en ésta simbología son múltiplos de una anchura determinada.

²⁷ <http://www.geocities.com/soho/cafe/8909/barcode.html>

²⁸ <http://www.monografias.com/trabajos11/yantucod/yantucod.shtml>

²⁹ http://codigodebarras.com/tema.php?ID=como_nacio

Con esta simbología se puede almacenar bastantes datos y es más difícil que sea alterada.³⁰



Figura 2.18. Ejemplo de código de barras 2D31

2.4.1. Ventajas del Código de Barras

El código de barras es un excelente sistema para gestionar datos con identificación automática, entre sus ventajas están:

- No hay retrasos notables desde que se lee la información hasta su uso gracias a la rápida captura de los datos.
- Los datos son exactos y hay mayor precisión en la información.
- Se mejora el control de las entradas y salidas.
- Bajo costo de impresión.
- Equipo de lectura fácil de conectar además de ser flexible y tener precios accesibles.
- Eliminación de errores comparado con los errores que se pueden producir al ingresar los datos de forma manual.³²

³⁰http://www.metrologiamexico.com/contenido1/informacion_tecnica/codigos_de_barras_de_dos_dimen.php

³¹ http://codigodebarras.com/tema.php?ID=como_nacio

³² <http://www.idautomatica.com/informacion-tecnica/codigo-de-barras.php>

2.4.2. Estructura

1.- **Quiet zone** es la zona que permite distinguir al código del resto de la información ya que es una zona libre de impresión.

2.- **Caracteres de inicio y terminación** barras y espacios específicos predefinidos que indican donde comienza y donde termina el código. Pueden ser iguales o diferentes una de la otra.

3.- **Caracteres de datos** son los números y letras particulares del código.

4.- **Checksum** sirve para la revisión matemática que valida los datos del código de barras. Algunos códigos de barras los tienen y otros no.³³

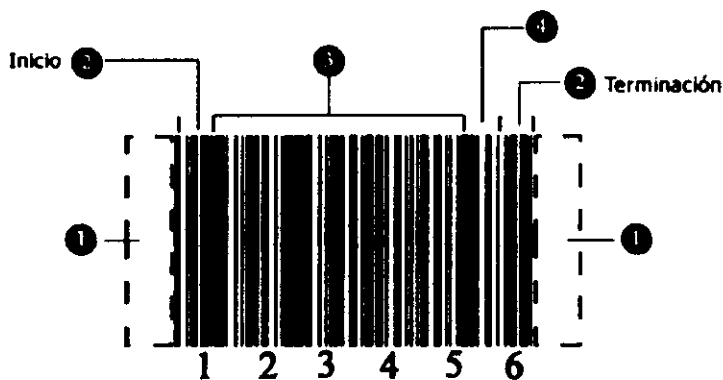


Figura 2.19. Estructura de un código de barras³⁴

2.4.3. Lectura del código de barras

El lector de código de barras decodifica la información por medio de la digitalización que proviene de una fuente de luz que se refleja en el código y luego se envía la información a la computadora como si se hubiese ingresado la información por el teclado.

³³ http://es.wikipedia.org/wiki/Codigo_de_barras

³⁴ http://es.wikipedia.org/wiki/Codigo_de_barras

En el proceso, el símbolo de código de barras se ilumina con una fuente de luz infrarrojo o visible, las barras oscuras absorben la luz y los espacios las reflejan nuevamente hacia un escáner.

El escáner transforma los cambios de luz en impulsos eléctricos que copian las barras y el modelo de espacio en el código de barras. Un decodificador usa algoritmos matemáticos para traducir estos impulsos eléctricos en un código binario y transmite el mensaje decodificado al computador. El decodificador puede estar integrado en el escáner o estar aparte.

Una vez que el código ha sido leído, será la aplicación de software la procese la información recibida. La aplicación recibe los datos enviados por el lector de código de barras y controla su flujo. La aplicación ayuda a que los datos se conviertan en información útil para el usuario ya que tiene el trabajo de clasificarla, procesarla, organizarla y guardarla.³⁵

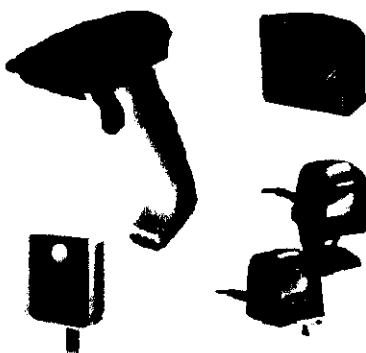


Figura 2.20. Lectores de códigos de barras³⁶

³⁵ <http://www.planetacurioso.com/2007/01/20/%C2%BFcomo-leer-el-codigo-de-barras/>

³⁶ http://www.tiendatpv.com/index.php?main_page=product_info&manufacturers_id=4&products_id=38&zenid=3da7f27534488b62fc154fa04415b28c

2.5. Bases de Datos

Una base de datos es un conjunto de datos organizados correlacionados y almacenados independientemente de los programas que los utilizan. La forma en cómo están organizados los datos da origen a la base de datos.

Una característica importante de las bases de datos es que se puede almacenar una gran cantidad de información logrando que ésta no sea redundante y que además se puedan hacer las consultas necesarias de acuerdo a los permisos que tenga el usuario para acceder a esa base de datos. Además es importante que una base de datos pueda ser actualizada, modificada y que se pueda aplicar el mantenimiento que la base requiera, un cambio en los datos no implica cambio en los programas y viceversa

En una base de datos las entidades de información están relacionadas a través de la organización y asociación. Un esquema que representa las definiciones de las relaciones entre las entidades de información define la arquitectura lógica, y como se configure el hardware residente define la arquitectura física de una base de datos. Estos dos aspectos deben satisfacer los requerimientos del usuario para poder obtener los informes o reportes.

Para crear una base de datos es necesario saber el flujo, estructura y la manera en la que se encuentran asociados los datos.

La creación de una base datos consta de dos partes:

Diseño Lógico: Es un modelo abstracto de la base de datos donde se detalla cómo debe ir agrupada la información de acuerdo a las necesidades del usuario.

Diseño Físico: Se especifica cómo queda la información ordenada en los dispositivos de almacenamiento en los que los datos serán guardados físicamente.

En el proceso de diseño se identifican elementos redundantes y los agrupamientos de los elementos de datos que se requieren para programas de aplicaciones específicos.

La manipulación de los datos se los hace a través de un sistema gestor de base de datos permitiendo el almacenamiento y posteriormente el acceso a los datos de forma rápida y estructurada.

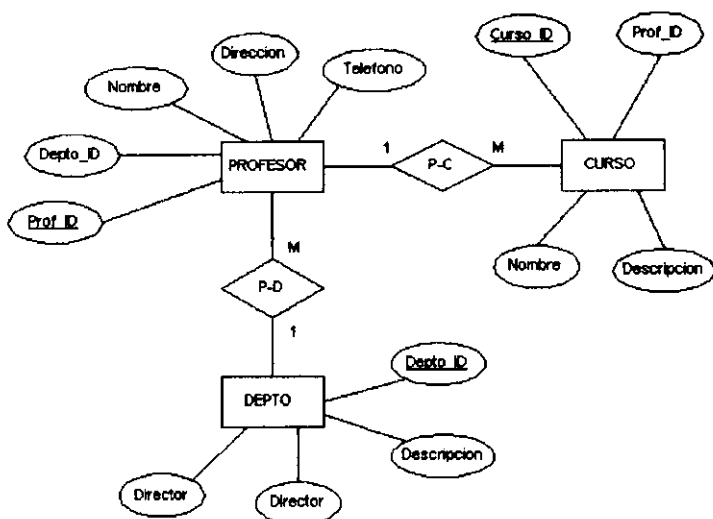
La información se almacena en tablas y cada tabla contiene uno o más registros. Cada registro puede contener uno o más campos y cada campo normalmente se relaciona con un tipo particular de información.

2.5.1. Modelos de Bases de Datos

2.5.1.1. Base de datos relacional

La idea fundamental de este modelo es el uso de relaciones. Los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas. Se puede considerar a una relación como un conjunto de tablas vinculadas entre sí que están compuestas por registros y campos. La información se recupera o almacena mediante consultas permitiendo así administrarla.

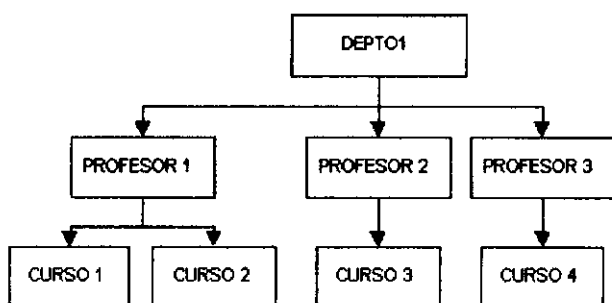
El lenguaje SQL (Structured Query Language) o Lenguaje Estructurado de Consultas es el más usado para crear las consultas de bases de datos relacionales. Los principales motores de sistemas de gestión de bases de datos son implementados en este estándar.



Diag 1. Diagrama Entidad/Relación de base de datos relacional³⁷

2.5.1.2. Base de datos Jerárquica

Estas bases de datos almacenan la información en una estructura jerárquica siendo los registros enlazados en forma de árbol donde un nodo padre puede tener varios nodos hijo. Una desventaja con este modelo es que es difícil controlar la redundancia de los datos, así como tampoco se puede garantizar una integridad referencial es decir se puede dar el caso de que un registro hijo esté relacionado a un registro padre inválido o inexistente.



Diag 2. Diseño lógico de base de datos jerárquica³⁸

³⁷ <http://elies.rediris.es/elies9/5-1.htm>

2.5.1.3. Base de datos de Red

A diferencia del anterior, este modelo permite que un nodo hijo tenga varios nodos padre que facilita el control de datos redundantes pero su administración es muy compleja.

2.5.2. MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. MySQL es un sistema para administrar bases de datos que pueden ser desde una simple lista de nombres a hasta una gran cantidad de información de una corporación. Para agregar, acceder a y procesar los datos guardados es necesario un administrador como MySQL Server.

MySQL es software de código abierto, esto quiere decir que cualquier persona puede usarlo y modificarlo sin necesidad de pagar. MySQL usa el GPL (GNU General Public License) para definir que se puede hacer y que no se puede hacer con el software en diferentes situaciones.

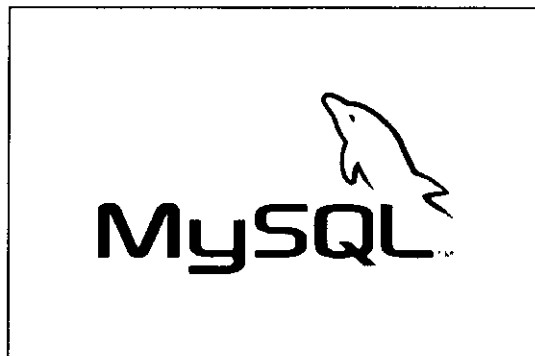


Figura 2.21. Logo de MySQL³⁹

³⁸ <http://elies.rediris.es/elies9/4-2-1.htm>

³⁹ <http://www.tecnologiadiaria.com/tag/mysql>

2.5.2.1. Características principales de MySQL

□ Interioridades y portabilidad

- Escrito en C y en C++
- Funciona en diferentes plataformas como GNU/Linux, Mac OS X, Solaris, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows.
- Uso completo de multi-threaded mediante threads del kernel. Pueden usarse fácilmente multiple CPUs si están disponibles.
- Proporciona sistemas de almacenamiento transaccionales y no transaccionales.
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible.
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (linkado) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.

□ Tipos de columnas

- Diversos tipos de columnas: enteros con/sin signo de 1, 2, 3, 4, y 8 bytes de longitud, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, y tipos espaciales OpenGIS
- Registros de longitud fija y longitud variable.

□ Sentencias y funciones

- Soporte completo para operadores y funciones en las cláusulas de consultas SELECT y WHERE.
- Soporte completo para las cláusulas SQL GROUP BY y ORDER BY. Soporte de funciones de agrupación (COUNT(), COUNT(DISTINCT ...), AVG(), STD(), SUM(), MAX(), MIN(), y GROUP_CONCAT()).
- Soporte para LEFT OUTER JOIN y RIGHT OUTER JOIN cumpliendo estándares de sintaxis SQL y ODBC.
- Soporte para alias en tablas y columnas como lo requiere el estándar SQL.
- DELETE, INSERT, REPLACE, y UPDATE devuelven el número de filas que han cambiado (han sido afectadas). Es posible devolver el número de filas que serían afectadas usando un flag al conectar con el servidor.
- El comando específico de MySQL SHOW puede usarse para obtener información acerca de la base de datos, el motor de base de datos, tablas e índices. El comando EXPLAIN puede usarse para determinar cómo el optimizador resuelve una consulta.
- Los nombres de funciones no colisionan con los nombres de tabla o columna. Por ejemplo, ABS es un nombre válido de columna. La única restricción es que para una llamada a una función, no se permiten espacios entre el nombre de función y el '(' a continuación.
- Puede mezclar tablas de distintas bases de datos en la misma consulta

□ Seguridad

- Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.

□ Escalabilidad y límites

- Soporte a grandes bases de datos. Se usa MySQL Server con bases de datos que contienen 50 millones de registros.
- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2). Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT.

□ Conectividad

- La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity).

□ Clientes y herramientas

- MySQL server tiene soporte para comandos SQL para revisar, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente **mysqlcheck**.
- Todos los programas MySQL pueden invocarse con las opciones --help o -? para obtener asistencia en línea.⁴⁰

2.3. Ingeniería de Software

La ingeniería de software es la rama de la ingeniería que permite la aplicación de técnicas y métodos para la producción de software con la finalidad de incrementar la calidad del software, controlar de principio a fin el proceso de desarrollo, proporcionar una base a los desarrolladores para que de manera eficiente construyan

⁴⁰ <http://www.tecnologiadiaria.com/tag/mysql>

software de alta calidad. La ingeniería del software ayuda además a crear una disciplina que controla que el proyecto de software sea culminado dentro del tiempo y costo estimados.

Para apoyar la realización de software, la ingeniería de software tiene varios paradigmas o modelos entre los cuales está el modelo de prototipos. Con este modelo tanto el cliente como el desarrollador definen los objetivos del software, identifican los requisitos y las áreas del esquema que deben ser mayormente definidos. La construcción del prototipo es planteada para un diseño rápido. El diseño rápido tiene aquellos aspectos que son más visibles para el usuario y ahí el cliente evalúa el software y comunica al desarrollador de los requisitos adicionales.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. ANÁLISIS ESTRUCTURADO DEL SISTEMA

3.1.1. Identificación de necesidades

La Pontificia Universidad Católica del Ecuador Sede Ambato está ubicada al sur de la ciudad y tiene un promedio de 700 alumnos de los cuales 30% aproximadamente utilizan el servicio de parqueadero de la universidad. Actualmente, los estudiantes, profesores y administrativos ingresan al parqueadero indicando al guardián asignado el sticker el cual les da el permiso para entrar. Para salir del parqueadero, el guardia simplemente acciona el interruptor que levanta el torniquete para que puedan ingresar o salir los autos.

Entre las necesidades que tiene la universidad es controlar de una forma automatizada el ingreso y salida de vehículos. Hay la necesidad de que evitar que el parqueadero se llene debido al ingreso de vehículos no autorizados. Existe además la necesidad de evitar que más de un vehículo ingrese con el mismo sticker. Es necesario controlar y evitar que personas no autorizadas retiren el vehículo del parqueadero. La administración necesita contar con reportes del ingreso u salida de vehículos para fines de estadística y para desarrollar eventuales mejoras en el servicio.

3.1.2. Estudio de viabilidad

3.1.2.1. Viabilidad económica

Este sistema se va a realizar sin ningún pago al proveedor pues se trata de una tesis de grado. Todos los gastos tanto de implementación y ciertos materiales serán cubiertos en su totalidad por la estudiante autora de la tesis

3.1.2.2. Viabilidad técnica

Se cuenta con las herramientas necesarias tanto de software como de hardware. Además se tiene acceso sin restricciones al parqueadero de la universidad para los estudios correspondientes. Se cuenta con el conocimiento de la parte técnica para la elaboración del sistema. Se cuenta además con la colaboración de la persona encargada del proceso de administración del parqueadero de la universidad, la Ing. María Elena Sandoval.

3.1.2.3. Viabilidad legal

El sistema se lo pueden realizar en varios lenguajes como Visual C++, Turbo C, C Sharp entre otros pero se eligió trabajar con el lenguaje Visual Basic 6.0 para el desarrollo del sistema para control de entrada y salida de vehículos pues la universidad cuenta con la licencia necesaria para poder desarrollar.

Las bases de datos serán desarrolladas por un sistema gestor de base de datos libre como es el administrador de bases de datos MySQL XAMPP, puesto que es un software fácil de manipular además de ser de código libre, es decir no se necesita adquirir una licencia para el mismo.

También todos los OCX y DLLs usados dentro del software serán de código libre para de esta forma poder usarlos sin tener que adquirir una licencia.

3.1.3. Análisis técnico y económico

Para el desarrollo de este sistema, se cuenta una persona es decir la estudiante que desarrollará el sistema, como aporte adicional se cuenta con un tutor que aportará con su conocimiento para el desarrollo del proyecto. Será necesario adquirir el equipo para el ingreso de la información al sistema como son dos lectores de código de barras o scanner, el material eléctrico y el material electrónico para la conexión entre el torniquete y el computador. El costo de equipos y servicios como será cubierto en su totalidad por la estudiante.

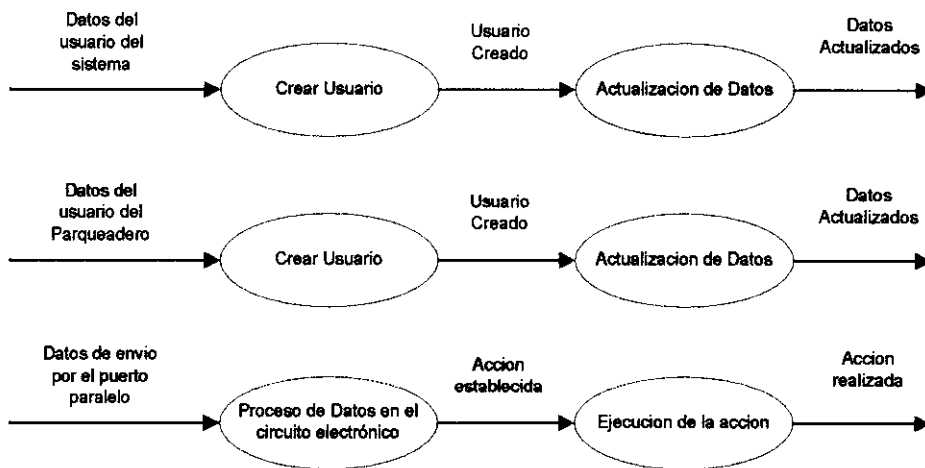
3.1.4. Definición general del sistema

El sistema permitirá ingresar a inicio de cada semestre los datos de los estudiantes, docentes y administrativos que tendrán acceso al parqueadero. El sistema se implementará en el parqueadero de la universidad para que estudiantes, docentes y administrativos usen su carnet de identificación para poder ingresar y salir con su vehículo. El sistema leerá la información contenida en el código de barras la misma que es el número de la cedula de cada persona, la comparará con la información de la base de datos y permitirá o no el ingreso o salida levantando de manera automática el torniquete. El sistema permitirá obtener reportes diarios, semanales y mensuales del número de vehículos que entran y salen y de los nombres de los carnets leídos por el sistema. El sistema mantendrá un control sobre el número de espacios disponibles dentro del parqueadero.

3.1.5. Análisis de requisitos del software

3.1.5.1. Ámbito de la información

3.1.5.1.1. Flujo de información



Diag 3. Flujo de la información⁴¹

3.1.5.1.2. Contenido de la información

Para el caso de los estudiantes, administrativos y docentes los datos son aquellos que se piden en la “Solicitud de Estacionamiento” y estos serán llamados Usuarios del Parqueadero, mientras que los usuarios del sistema serán los que tienen control del sistema en su totalidad o parte del mismo. La información se divide en dos partes, los datos personales y las características del vehículo para el caso de los usuarios del parqueadero. En los datos personales se solicita cédula de identidad, nombre, dirección, teléfono, celular, e-mail, escuela, nivel, horario, ya sea diurno o nocturno, y cargo. En las características del vehículo se solicita marca, modelo, color, año,

⁴¹ Investigador

placas del auto, además a nombre de quien está la matricula del auto. En el caso de los usuarios del sistema se requiere solo su usuario, clave y tipo para de esta forma establecer sus respectivos privilegios. Los usuarios del sistema serán de dos tipos Usuarios Administradores y Usuarios Operadores.

3.1.5.1.3. Estructura de la información

Los formularios que se van a manejar contienen los datos que especificaron en el contenido de la información. En estos formularios se van a llenar los diferentes datos que identifiquen al usuario, así como los datos del estudiante, docente o administrativo.

Datos de Usuarios del Sistema

Nombre (texto)

Password (texto)

Tipo (texto)

Datos personales del Usuario del Parqueadero

Cedula de identidad (texto)

Nombre (texto)

Dirección (texto)

Teléfono (texto)

Celular (texto)

E-mail (texto)

Escuela (texto)

Nivel (texto)

Horario (diurno o nocturno)

Cargo (texto)

Datos del vehículo del Usuario del Parqueadero

Cedula de identidad del usuario del parqueadero (texto)

Marca (texto)

Modelo (texto)

Color (texto)

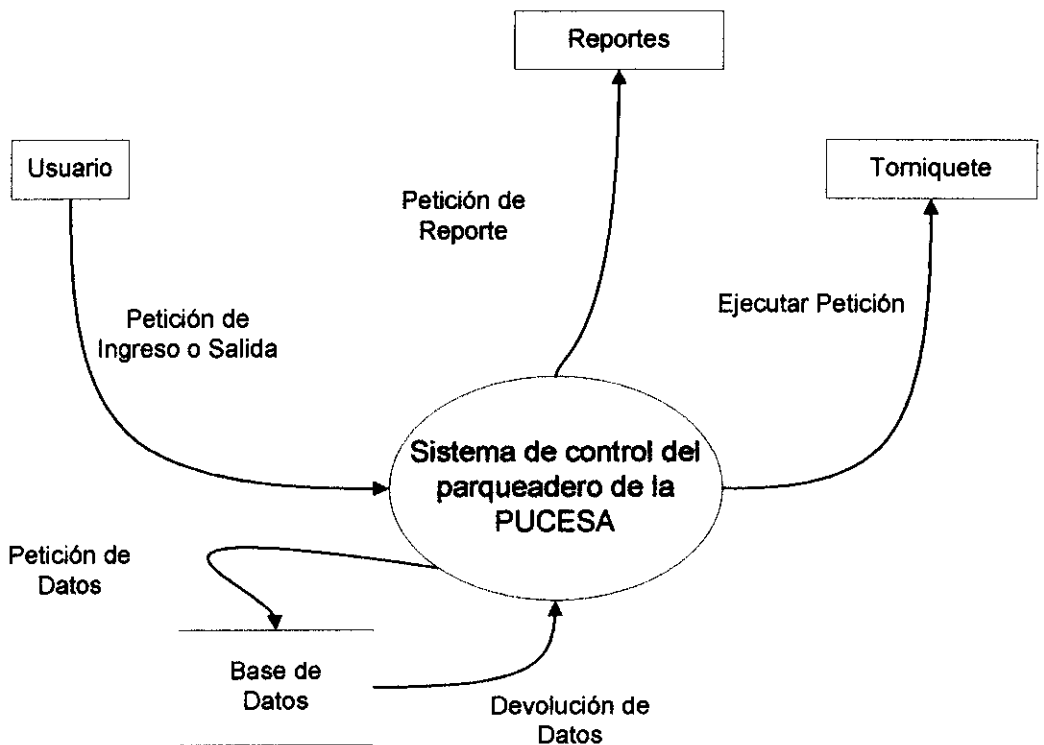
Año (texto)

placas del auto (texto)

Matricula (texto)

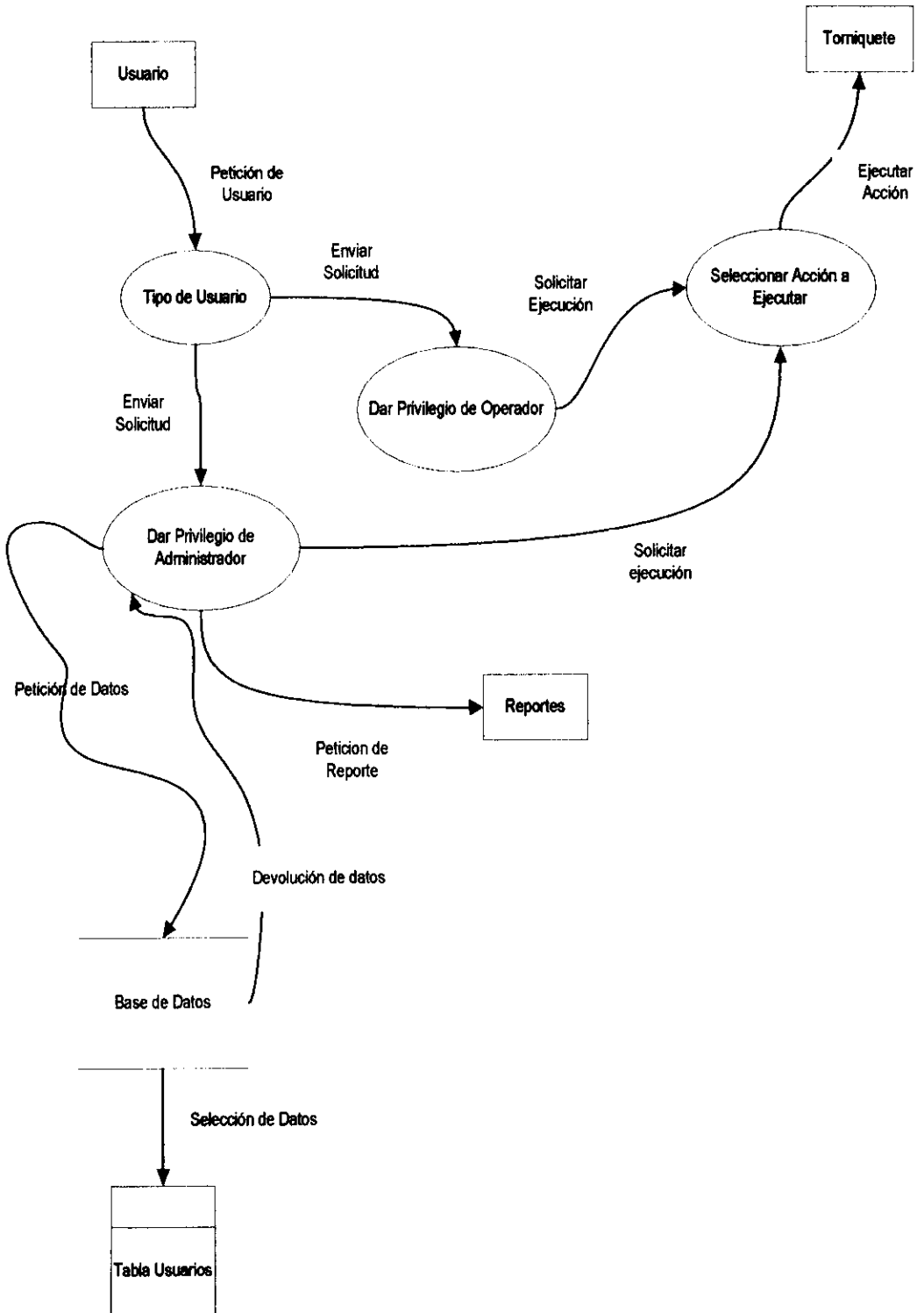
3.1.6. Diagramas de flujo de datos

DFD Nivel 0



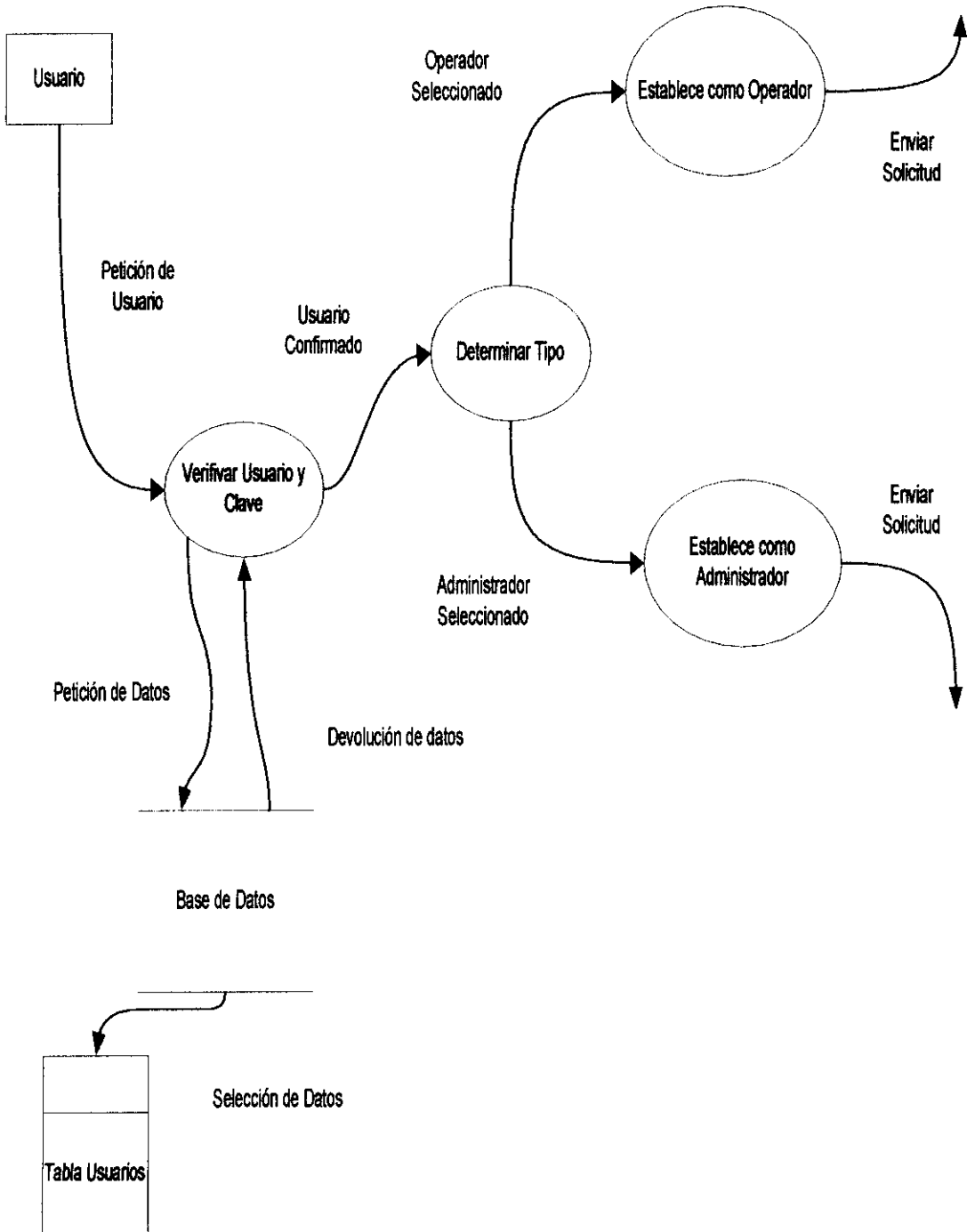
Diag 4. DFD Nivel 0 42

DFD Nivel 1



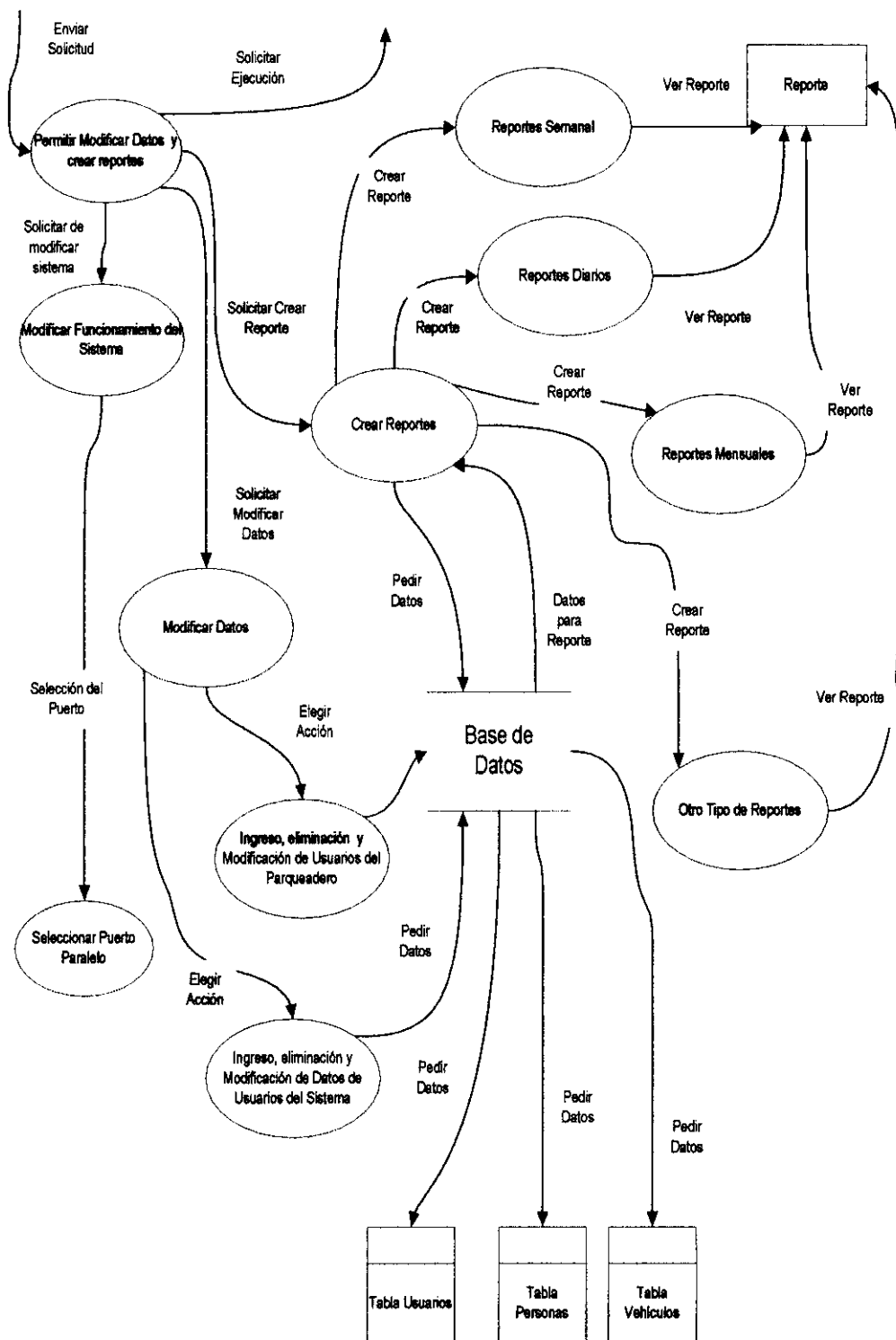
Diag 5. DFD Nivel 143

DFD Nivel 2 (Tipo de Usuario)



Diag 6. DFD Nivel 2 Tipo de Usuario44

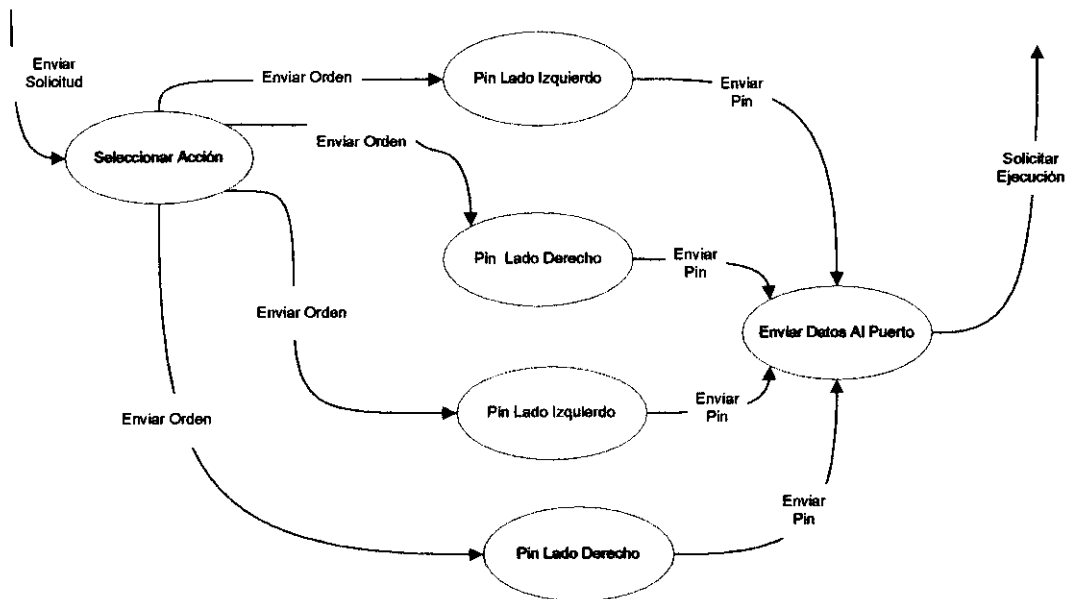
DFD Nivel 2 (Dar Privilegios de Administrador)



Diag 7. DFD Nivel 2 Privilegios de Administrador⁴⁵

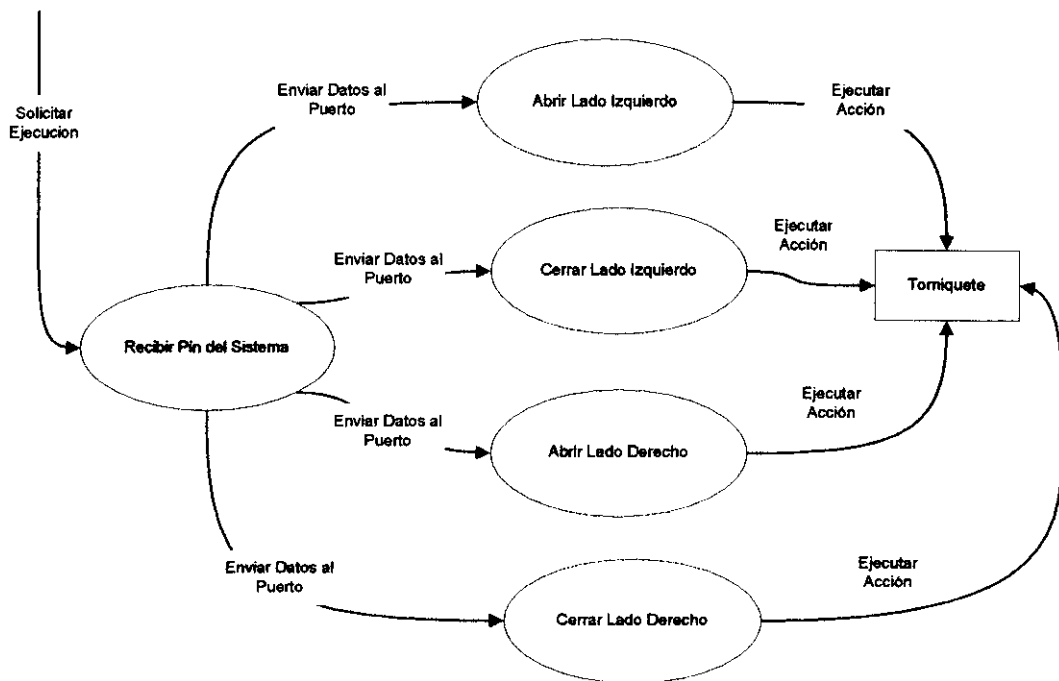
⁴⁵ Investigador

DFD Nivel 2 (Dar Privilegios de Operador)



Diag 8. DFD Nivel 2 Privilegios de Operador⁴⁶

DFD Nivel 2 (Seleccionar Acción a Ejecutar)



Diag 9. DFD Nivel 2 Selección acción a ejecutar⁴⁷

⁴⁶ Investigador

3.1.6.1. Diccionario de requisitos

➤ Nivel 0

Describe el funcionamiento principal del sistema, sus entradas, salidas y procesos.

➤ Nivel 1

En este nivel se describe los módulos con los cuales el sistema ira a trabajar los mismos que son detallados con sus respectivas entradas y salidas de datos.

➤ Nivel 2

- Tipo de Usuarios

Verifica el flujo de datos de los tipos de usuarios a usar el sistema dándoles o negándoles de esta forma ciertos privilegios dependiendo el tipo de usuario.

- Dar Privilegios de administrador

Permite visualizar las acciones y procesos que como administrador podrá ejecutarse en el sistema, como es el ingreso, eliminación y modificación de los registros de la base de datos, a demás permite controlar ciertas funciones del sistema que son amigables o flexibles, es decir se pueden configurar, como es el caso de seleccionar la dirección del puerto paralelo.

- Dar Privilegios de Operador

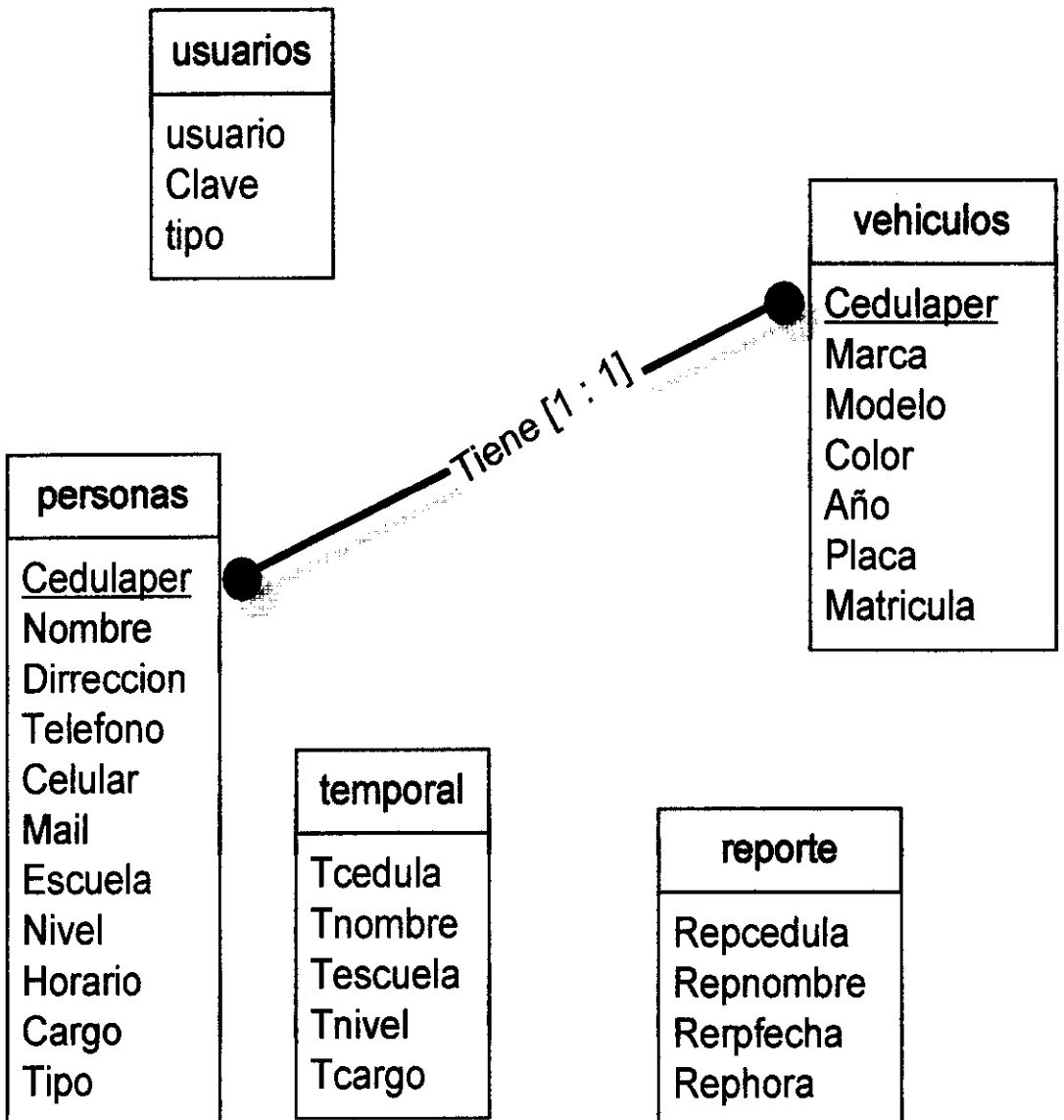
Permite observar las acciones y procesos que los usuarios operadores del sistema están en capacidad de usar, estas netamente son la apertura y cierre del torniquete de una forma manual y/o automática.

- Seleccionar Acción a Ejecutar

⁴⁷ Investigador

Da la muestra como correrán los datos al ser enviados al puerto paralelo para la apertura y cierre del torniquete dependiendo si es una petición de entrada o salida.

3.1.7. Diagrama Entidad Relación

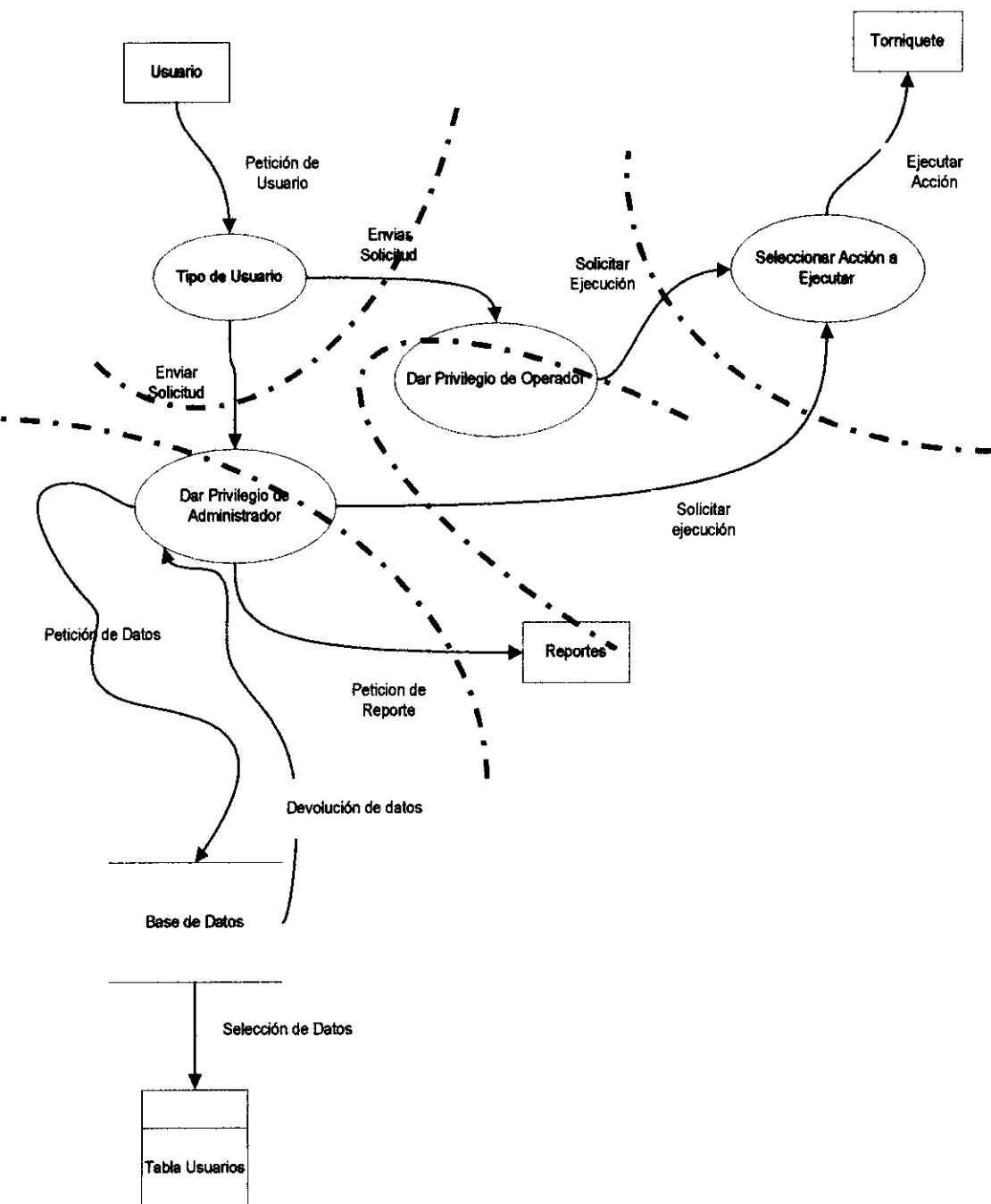


Diag 10. Entidad – Relación⁴⁸

⁴⁸ Investigador

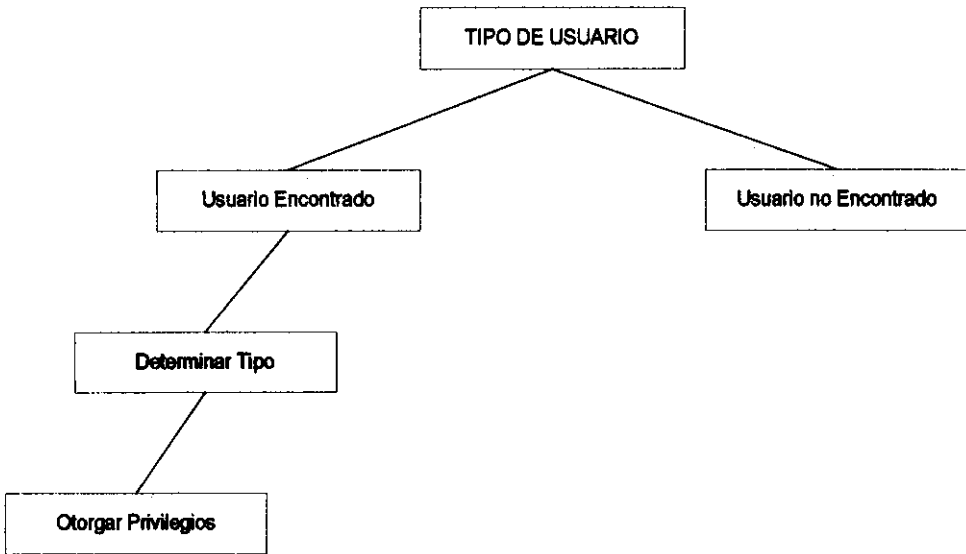
3.2. DISEÑO ARQUITECTÓNICO

3.2.1. DFD Nivel 2



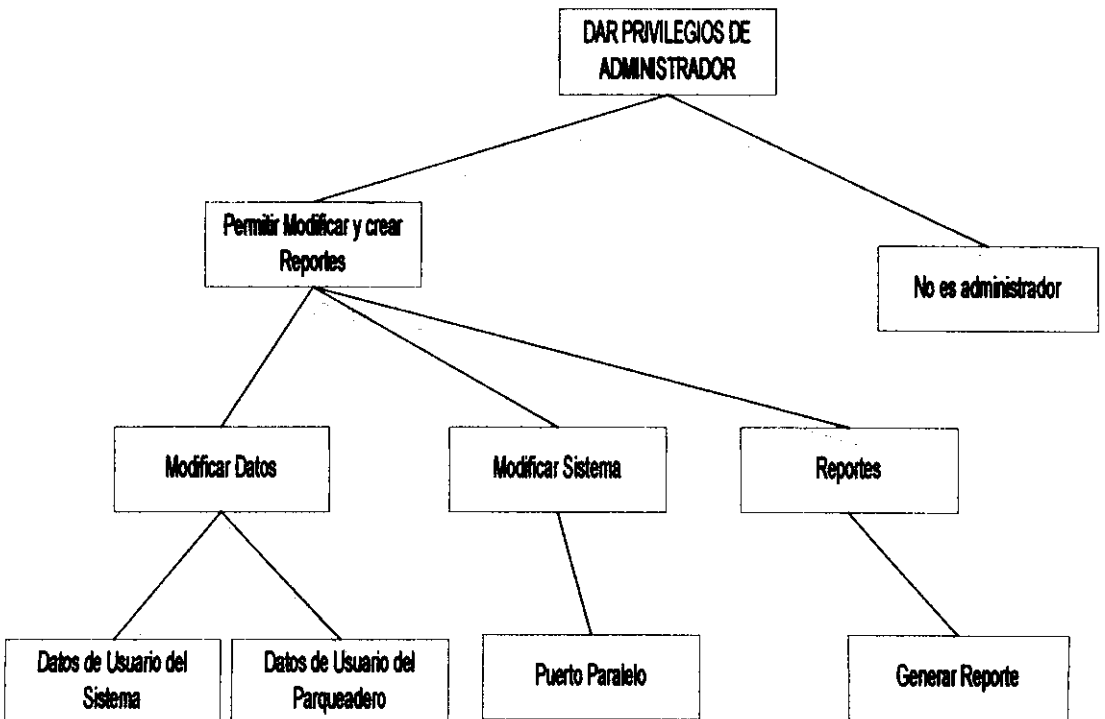
Diag 11. Diseño Arquitectónico DFD Nivel 249

3.2.1.1. Diseño Arquitectónico (Tipo de Usuario)



Diag 12. Diseño Arquitectónico Tipo de Usuario

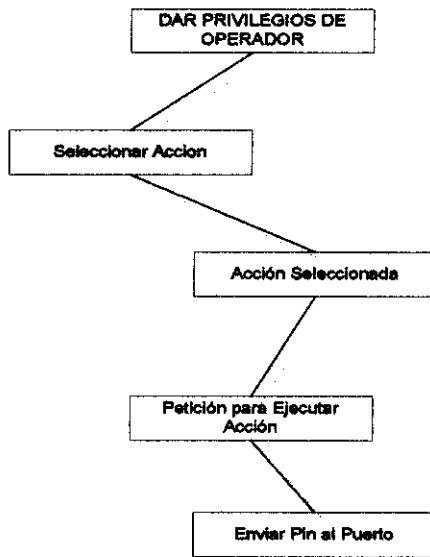
3.2.1.2. Diseño Arquitectónico (Dar Privilegios de Administrador)



Diag 13. Diseño Arquitectónico Privilegios Administrador⁵⁰

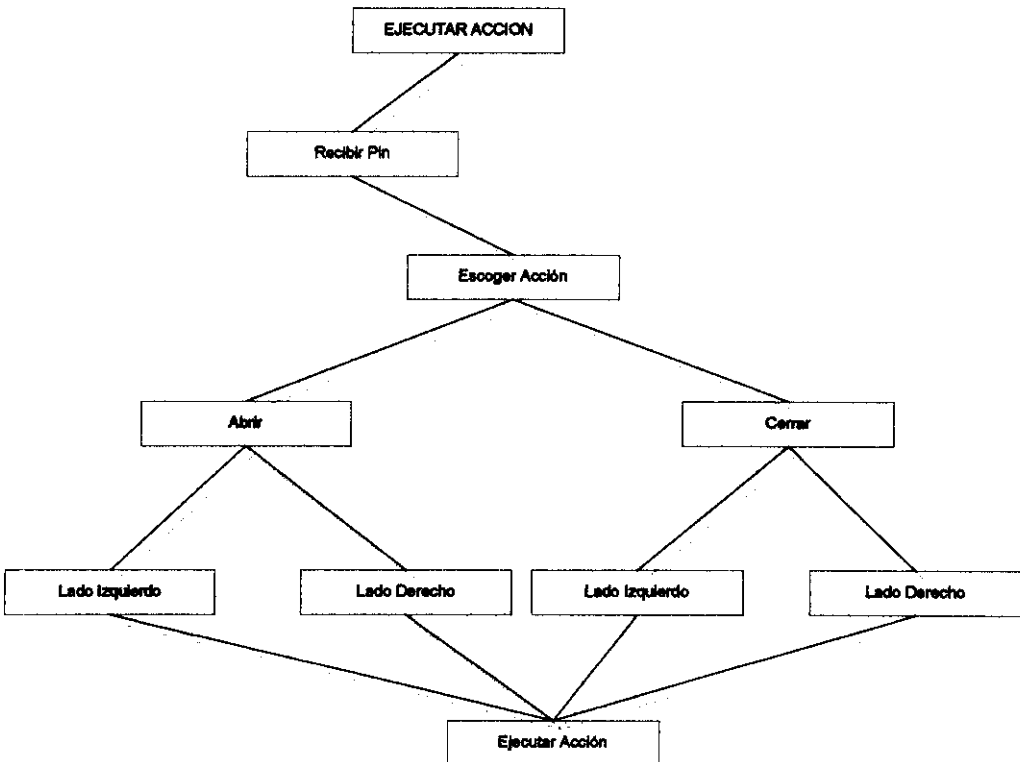
⁵⁰ Investigador

3.2.1.3. Diseño Arquitectónico (Dar Privilegios de Operador)



Diag 14. Diseño Arquitectónico Privilegios Operador⁵¹

3.2.1.4. Diseño Arquitectónico (Ejecutar Acción)



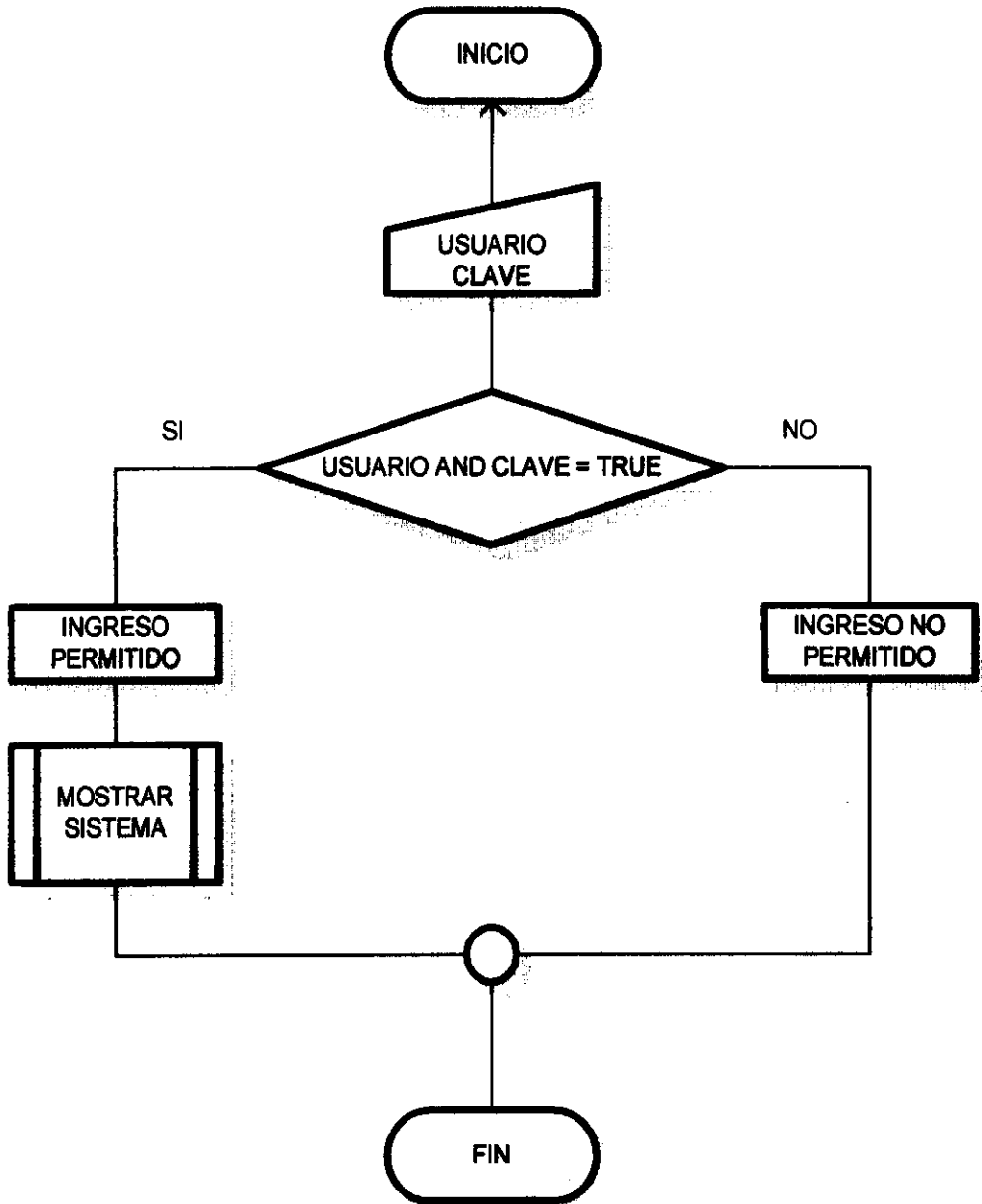
Diag 15. Diseño Arquitectonico Ejecutar Acción

⁵¹ Investigador

3.2.2. Diseño Procedimental

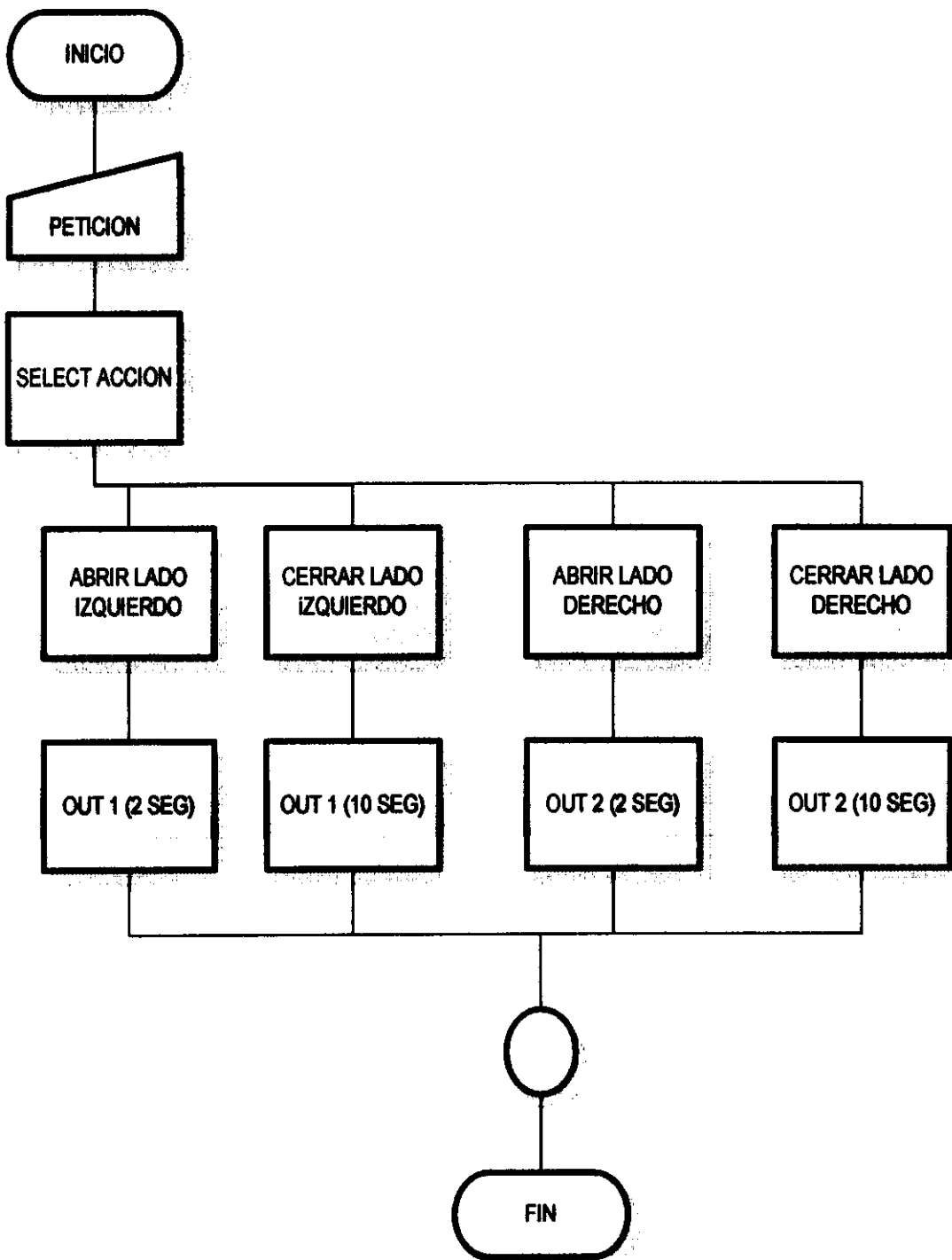
3.2.2.1. Diagramas de flujo

3.2.2.1.1. Diagrama de Acceso al Sistema



Diag 16. Acceso al Sistema⁵²

3.2.2.1.2. Diagrama de Funcionamiento del Puerto Paralelo en el Sistema



Diag 17. Funcionamiento de el puerto paralelo en el sistema53

3.2.2.2. Diseño De Interfaz

El sistema estará dotado de una interfaz grafica totalmente fácil de usar para dar así una comodidad al usuario, además de ser funcional y simple para de esta forma permitir al usuario un manejo seguro y rápido de la aplicación.

La aplicación contara con menús en la parte superior los mismos que desplegaran todas las opciones posibles dentro del manejo del sistema, dependiendo del tipo de usuario que este logeado el sistema permitirá o negara opciones de este menú.

La opción principal del sistema como es claro es el control del torniquete del parqueadero de la PUCESA, para esto la aplicación contara con un formulario en el cual se puede controlar el torniquete de una forma automática, es decir los usuarios del parqueadero podrán entrar y salir una vez al parqueadero introduciendo su cedula en el sistema, además también se lo puede controlar de forma manual, lo que permite que si en algún caso de excepción el auto que quiere entrar en el parqueadero no está registrado pero es necesario su ingreso un operador le puede permitir pasar o salir.

La aplicación además permite el manejo de los datos de los usuarios, como es la creación, eliminación, y modificación ya sean usuarios del parqueadero o del sistema.

Se mostraran también pantallas de mensajes y errores del sistema, y configuración del puerto paralelo.

Permite también el acceso a la ayuda del sistema la misma que permitirá obtener información sobre el uso del mismo, sus principales funciones y características.

3.2.2.2.1. Pantalla de ingreso

Esta pantalla permitirá el ingreso al sistema es donde se ingresara el usuario y la clave para tener acceso a las funciones del sistema, dependiendo del tipo de usuario se dará ciertos privilegios.

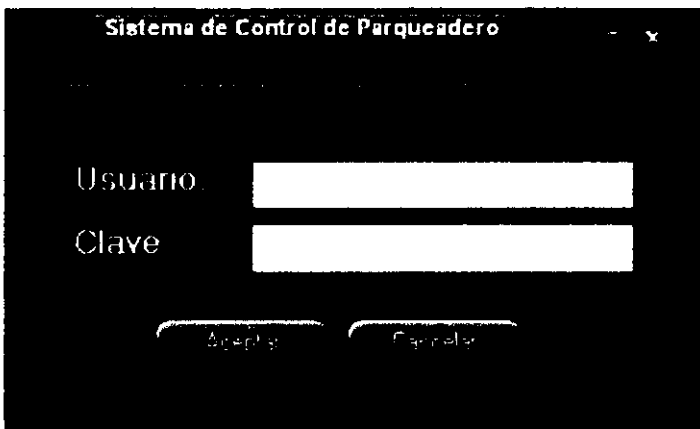


Figura 3.1. Ingreso de Usuario⁵⁴

3.2.2.2.2. Pantalla Principal del sistema

Esta mostrará los menús y sus opciones además dentro de esta se cargaran todas las ventanas del sistema, para permitir una forma ordenada de visualizar las pantallas.

⁵⁴ Investigador



Figura 3.2. Pantalla Principal⁵⁵

3.2.2.2.3. Pantalla de control parqueadero

Ventana en la cual se pueden controlar el parqueadero sea de forma manual o automática, esta es en donde se representa la función principal del sistema.



Figura 3.3. Control del Parqueadero⁵⁶

3.2.2.2.4. Pantalla de control de usuario del sistema

Esta ventana permite modificar, crear o eliminar usuarios del sistema, a esta solo tendrán acceso los usuarios administradores.



Figura 3.4. Usuarios del Sistema⁵⁷

3.2.2.2.5. Pantalla de control de usuario del parqueadero

Aquí se puede crear, modificar o eliminar usuarios que pueden usar el parqueadero es decir quiénes pueden o no entrar al parqueadero.

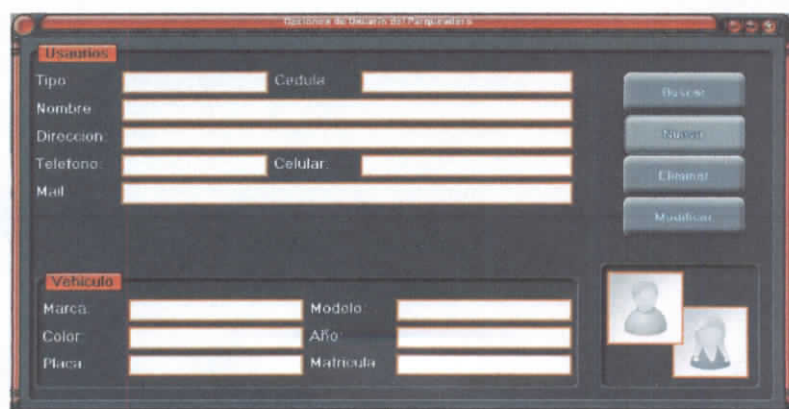


Figura 3.5. Usuarios del Parqueadero⁵⁸

⁵⁶ Investigador

⁵⁷ Investigador

⁵⁸ Investigador

3.2.2.2.6. Pantalla de Búsqueda

Esta ventana permite hacer búsquedas de los usuarios del parqueadero, ya sea por su cedula, nombre o placa de su vehículo.

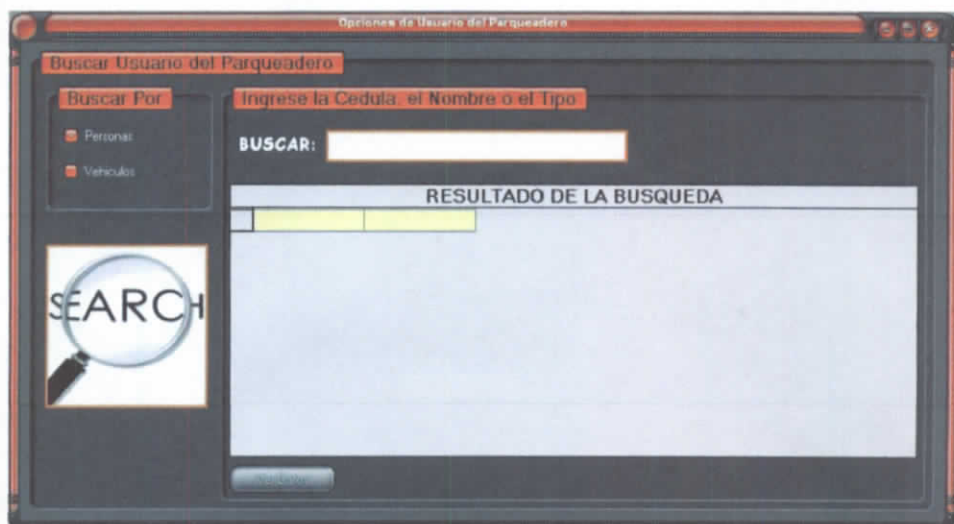


Figura 3.6. Búsqueda⁵⁹

3.2.2.2.7. Pantalla de Mensajes

Las pantallas que serán desplegadas cuando el sistema quiera emitir algún mensaje o alerta al usuario.

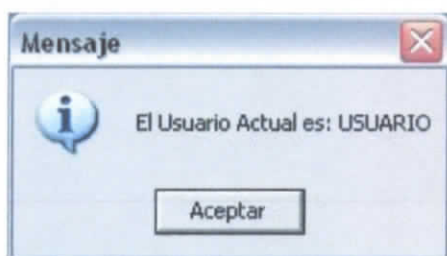
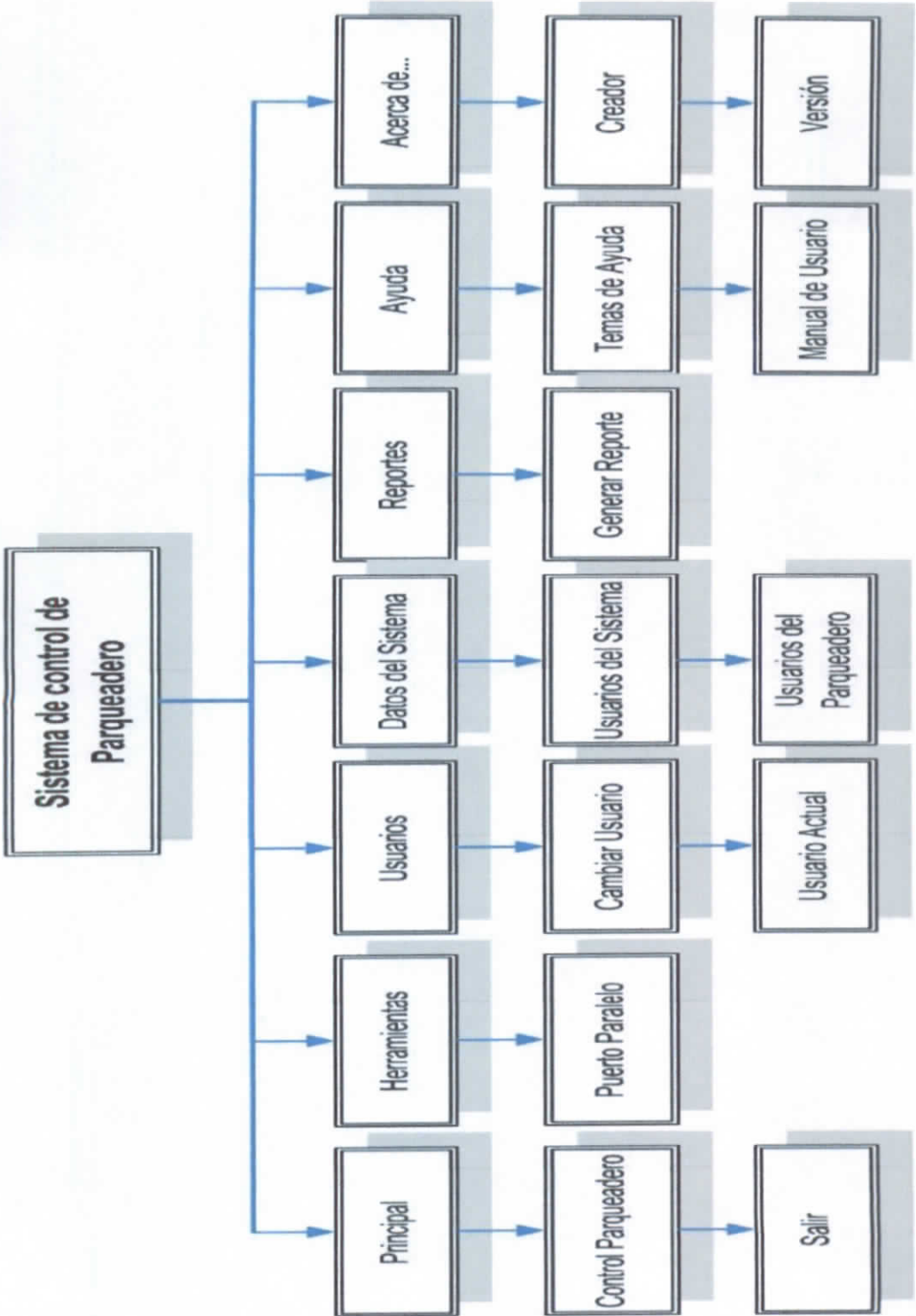


Figura 3.7. Mensajes⁶⁰

⁵⁹ Investigador

⁶⁰ Investigador

3.2.2.3. Diagrama de Menú



Diag 18. Diagrama de Menú⁶¹

⁶¹ Investigador

3.2.3. Codificación del Sistema

3.2.3.1. Código para conexión a la base de datos

```
Public mConn As ADODB.Connection
```

```
Public Sub Conectarbd()
```

```
    On Error GoTo controlerror
```

```
    Set mConn = New ADODB.Connection
```

```
    mConn.CommandTimeout = 40
```

```
    mConn.ConnectionTimeout = 40
```

```
    mConn.CursorLocation = adUseClient
```

```
    mConn.ConnectionString = "UID=root" & ";PWD=admin" &  
    ";DATABASE=parqueadero;" & "SERVER=localhost;" &  
    "DRIVER={MySQL ODBC 3.51 Driver};"
```

```
    mConn.Open
```

```
    Usuario = "root"
```

```
Exit Sub
```

```
controlerror:
```

```
    MsgBox "Compruebe nombre y contraseña" & frmLogin.txtPassword.Text,  
    vbCritical
```

```
End Sub
```

3.2.3.2. Código Control del Puerto Paralelo

Global dpuerto As String

Global valpuerto As Integer

Public Declare Function Inp Lib "inpout32.dll" _

Alias "Inp32" (ByVal PortAddress As Integer) As Integer

Public Declare Sub Out Lib "inpout32.dll" _

Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)

3.2.3.3. Funciones y Procedimientos

Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Public Declare Function PathIsDirectory Lib "shlwapi.dll" Alias "PathIsDirectoryA"
(ByVal pszPath As String) As Long

Public Declare Function ShellExecute Lib "shell32.dll" _

Alias "ShellExecuteA" (ByVal hwnd As Long, ByVal _

lpOperation As String, ByVal lpFile As String, ByVal _

lpParameters As String, ByVal lpDirectory As String, _

ByVal nShowCmd As Long) As Long

Public Const SW_SHOWNORMAL = 1

Public Sub selskin(ByVal queskin As String, ByVal formulario As Form)

fileexist (queskin)

If sifile = True Then

formulario.Skin1.LoadSkin App.Path & "\SKN\" & queskin & ".skn"

formulario.Skin1.ApplySkin formulario.hwnd

End If

End Sub

Public Sub fileexist(ByVal nombrefile As String)

On Error GoTo Fallo

x = GetAttr(App.Path & "\SKN\" & nombrefile & ".skn")

sifile = True

'MsgBox "El fichero existe." & sifile

Exit Sub

Fallo:

sifile = False

'MsgBox "El fichero no existe." & sifile

End Sub

Public Function RecordsetVacio(rs As Recordset) As Boolean

RecordsetVacio = ((rs.BOF = True) And (rs.EOF = True))

End Function

Public Sub Ordenargrid(ByVal ColIndex As Integer, rs As ADODB.Recordset,

DataGrid As DataGrid)

Dim strColName As String

Static bSortAsc As Boolean

Static strPrevCol As String

strColName = DataGrid.Columns(ColIndex).DataField

If strColName = strPrevCol Then

If bSortAsc Then

rs.Sort = strColName & " DESC"

bSortAsc = False

Else

```

        rs.Sort = strColName

        bSortAsc = True

    End If

Else

    rs.Sort = strColName

    bSortAsc = True

End If

strPrevCol = strColName

End Sub

Public Sub Aj_col_datagrid(un_DataGrid As DataGrid, ado As ADODB.Recordset,
Optional AccForHeaders As Boolean)

Dim TempCol() As Integer

Dim Nregistros As Integer, NCampos As Integer

Dim Fila As Long, Col As Long, width As Single

Dim maxWidth As Single, celdaText As String

Dim saveFont As StdFont, oldScaleMode As Integer

'Variables para la cantidad de registros y columnas

Nregistros = ado.RecordCount

NCampos = ado.Fields.Count

'Array para almacenar el ancho de cada columna

ReDim TempCol(NCampos)

' Si el número de registros es igual a 0 se sale

If Nregistros = 0 Then Exit Sub

' Guarda la fuente del DataGrid para luego reestablecerla

Set saveFont = un_DataGrid.Parent.Font

```

```

Set un_DataGrid.Parent.Font = un_DataGrid.Font
' Ajustar el ScaleMode en vbTwips para el formulario
oldScaleMode = un_DataGrid.Parent.ScaleMode
un_DataGrid.Parent.ScaleMode = vbTwips
'Mueve al Primer registro
ado.MoveFirst
maxWidth = 0
'Recorre las columnas
For Col = 0 To NCampos - 1
ado.MoveFirst
If AccForHeaders Then
'Almacena el Ancho del texto de la columna
maxWidth = un_DataGrid.Parent.TextWidth _
(un_DataGrid.Columns(Col).Text) + 200
End If
ado.MoveFirst
'Recorre los registros de esta columna
For Fila = 0 To Nregistros - 1
If NCampos = 1 Then
Else
celdaText = un_DataGrid.Columns(Col).Text
End If
'Almacena el Ancho del texto de la celda del Datagrid
width = un_DataGrid.Parent.TextWidth(celdaText) + 200
'Si el ancho de la celda es mayor se actualiza la variable maxWidth _

```

y se establece el ancho de la columna

```
If width > maxWidth Then
```

```
maxWidth = width
```

```
un_DataGrid.Columns(Col).width = maxWidth
```

```
End If
```

```
' Mueve el Ado al Siguiente registro
```

```
ado.MoveNext
```

```
Next Fila
```

```
'Almacena el ancho de la columna
```

```
TempCol(Col) = maxWidth
```

```
Next Col
```

```
'Recorre cada columna y le asigna el ancho
```

```
For Col = 0 To NCampos - 1
```

```
un_DataGrid.Columns(Col).width = TempCol(Col)
```

```
Next
```

```
'restablece la fuente del DataGrid y el scaleMode
```

```
Set un_DataGrid.Parent.Font = saveFont
```

```
un_DataGrid.Parent.ScaleMode = oldScaleMode
```

```
ado.MoveFirst
```

```
Erase TempCol
```

```
End Sub
```

```
Public Function BACKUPdb(ByVal DirOr As String, ByVal DirDes As String)
```

```
On Error GoTo errores
```

```
Dim fso, carpet
```

```
Set fso = CreateObject("Scripting.FileSystemObject")
```

```
'origen = "C:\\"
```

```
'destino = "D:\\"
```

```
Set carpeta = fso.copyfolder(DirOr, DirDes, True)
```

```
Exit Function
```

```
errores:
```

```
If Err.Number = 13 Then
```

```
Resume Next
```

```
End If
```

```
End Function
```

```
Public Sub OpenDirectory(Directory As String)
```

```
ShellExecute 0, "Open", Directory, vbNullString,
```

```
vbNullString, SW_SHOWNORMAL
```

```
End Sub
```

```
Public Sub OpenExplorer(Optional InitialDirectory As String)
```

```
ShellExecute 0, "Explore", InitialDirectory, _
```

```
vbNullString, vbNullString, SW_SHOWNORMAL
```

```
End Sub
```

'Función que comprueba si una dirección de email es válida

```
Public Function Validar_Email(ByVal Email As String) As Boolean
```

```
Dim i As Integer, il.en As Integer, caracter As String
```

```
Dim pos As Integer, bp As Boolean, iPos As Integer, iPos2 As Integer
```

```
On Local Error GoTo Err_Sub
```

```
Email = Trim$(Email)
```

```
If Email = vbNullString Then
```

```
Exit Function
```

End If

Email = LCase\$(Email)

iLen = Len(Email)

For i = 1 To iLen

 character = Mid(Email, i, 1)

 If (Not (character Like "[a-z]")) And (Not (character Like "[0-9]")) Then

 If InStr(1, "_-" & "." & "@", character) > 0 Then

 If bp = True Then

 Exit Function

 Else

 bp = True

 If i = 1 Or i = iLen Then

 Exit Function

 End If

 If character = "@" Then

 If iPos = 0 Then

 iPos = i

 Else

 Exit Function

 End If

 End If

 If character = "." Then

 iPos2 = i

 End If

 End If

Else

Exit Function

End If

Else

bp = False

End If

Next i

If iPos = 0 Or iPos2 = 0 Then

Exit Function

End If

If iPos2 < iPos Then

Exit Function

End If

Validar_Email = True

Exit Function

Err_Sub:

On Local Error Resume Next

Validar_Email = False

End Function

Public Function val_cedula(ByVal cedula As String) As Boolean

Dim tam As Integer

Dim spares, simpares, impares, decena, total, dijito As Integer

If Len(cedula) = 10 Then

""""sumar pares""""

For i = 2 To 8 Step 2

```
spares = spares + Val(Mid(cedula, i, 1))
```

```
Next i
```

```
""""""impares""""""
```

```
For i = 1 To 9 Step 2
```

```
impares = Val(Mid(cedula, i, 1)) * 2
```

```
If impares > 9 Then
```

```
    impares = impares - 9
```

```
End If
```

```
    simpares = simpares + impares
```

```
Next i
```

```
total = spares + simpares
```

```
decena = Val(total / 10)
```

```
decena = (decena * 10) + 10
```

```
dijito = Val(decena - total)
```

```
If dijito = Val(Mid(cedula, 10, 1)) Then
```

```
    val_cedula = True
```

```
Else
```

```
    val_cedula = False
```

```
End If
```

```
Else
```

```
    val_cedula = False
```

```
End If
```

```
End Function
```

3.2.3.4. Modulo para Abrir archivos CHM

Public Enum HH_COMMAND

HH_DISPLAY_TOPIC = &H0

HH_HELP_FINDER = &H0

HH_DISPLAY_TOC = &H1

HH_DISPLAY_INDEX = &H2

HH_DISPLAY_SEARCH = &H3

HH_SET_WIN_TYPE = &H4

HH_GET_WIN_TYPE = &H5

HH_GET_WIN_HANDLE = &H6

HH_GET_INFO_TYPES = &H7

HH_SET_INFO_TYPES = &H8

HH_SYNC = &H9

HH_ADD_NAV_UI = &HA

HH_ADD_BUTTON = &HB

HH_GETBROWSER_APP = &HC

HH_KEYWORD_LOOKUP = &HD

HH_DISPLAY_TEXT_POPUP = &HE

HH_HELP_CONTEXT = &HF

HH_TP_HELP_CONTEXTMENU

HH_TP_HELP_WM_HELP = &H11

HH_CLOSE_ALL = &H12

HH_ALINK_LOOKUP = &H13

End Enum

Public Declare Function HtmlHelp Lib "hhctrl.ocx" Alias "HtmlHelpA" _

(ByVal hwndCaller As Long, ByVal pszFile As String, _
 ByVal uCommand As HH_COMMAND, ByVal dwData As Long) As Long

3.2.4. Diseño del circuito electrónico

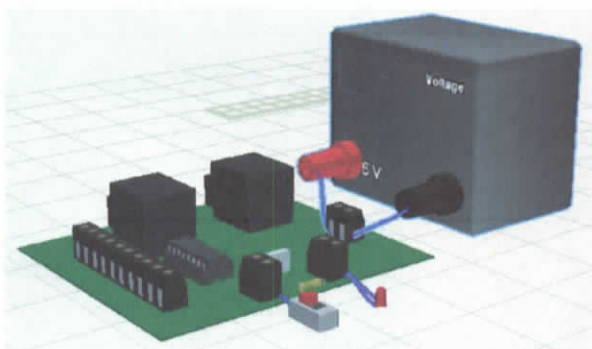


Figura 3. 8. Circuito Electrónico62

La imagen anterior muestra el diagrama del circuito ha elaborar para la conexión o interfaz entre el torniquete y el computador. Este circuito ayudará ha enviar la señal del puerto paralelo del computador al motor del torniquete, el mismo consta de varios componentes electrónicos los cuales son: resistencia, transistores, integrado, relés y borneras. Las siguientes imágenes muestran como deberá quedar terminado el circuito electrónico con todos sus componentes.

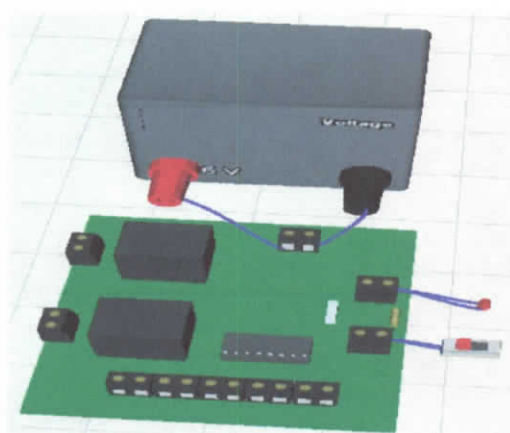


Figura 3. 9. Circuito Electrónico63

3.2.4.1. Diseño de las pistas del circuito

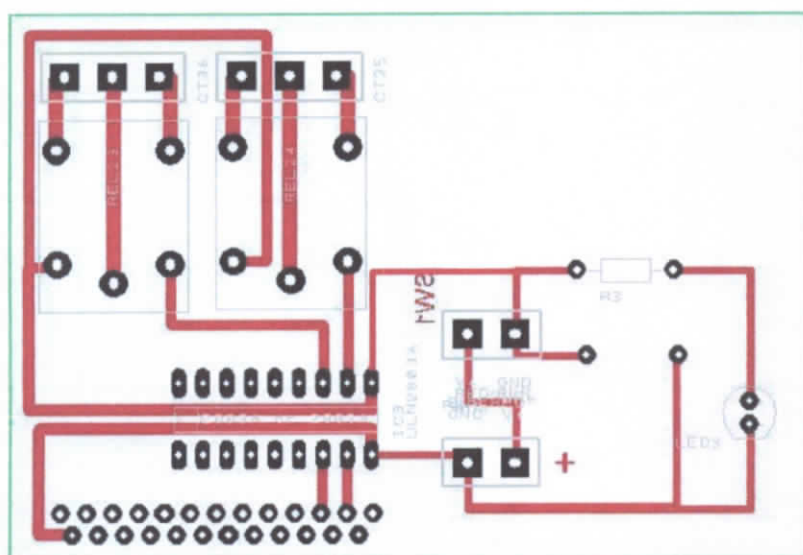


Figura 3. 10. Pistas Para El Circuito Electrónico⁶⁴

Para la elaboración de las pistas de la placa del circuito es necesario tener en cuenta el método a usar, en este caso se usará el método por Termotransferencia, este consta de pasar del papel fotográfico o papel transfer por medio de calor las pistas a la baquelita, para luego corroer el cobre sobrante de la placa.

Uno de los factores más importantes al momento de elaborar las pistas es comprobar toda su continuidad y calidad es decir que no existan rupturas o cortes de las pistas.

Tener en cuenta además donde irán ubicados correctamente los componentes electrónicos para así evitar problemas posteriores al momento del montaje del circuito.

Para la elaboración del diagrama de pistas se pueden usar varios programas de diseño electrónico, el que se ha usado en este caso es el Crocodile Real-PCB.

⁶⁴ Investigador

3.2.4.2. Componentes y Materiales a usar

- **Borneras:** Estas permiten conectar los cables del swith del torniquete al circuito.
- **Resistencia:** Evitara el paso de corriente en demasía ejerciendo una oposición a esta, esto ayudara a evitar que se quemen los componentes electrónicos del circuito.
- **Conector Ipt Macho:** Es el conector macho del puerto paralelo este permitirá que conectar el circuito al cable para la transmisión de datos desde el computador al circuito.
- **Sócalo:** es la ranura o espacio donde ira colocado el circuito integrado, este sócalo ira soldado a la baquelita y ayudará en el caso que el circuito integrado llegase a quemarse o dañarse, siendo asi fácil de remover y remplazar.
- **Moc 3011:** Establecerá la conexión entre la señal digital del computador para convertirse en una especie de swith y de esta forma permitir o negar la ejecución de apertura o cierre del torniquete.
- **Triac 2160:** Estos servirán para hacer las veces de interruptor en el circuito electrónico permitiendo el paso de corriente o no.
- **Baquelita:** Es una placa de pastico que no conduce electricidad o corriente pero en su parte inferior está dotada de una superficie de cobre las cual permite la elaboración de las pistas conductoras de corriente.
- **Acido cloruro férrico:** Con este acido se corroe la baquelita en la elaboración de las pistas de la placa.

- **Papel Transfer:** Ayuda en la impresión de las pistas para luego ser transferido por calor a la baquelita.
- **Estaño y Cautín:** Para soldar los componentes a la baquelita.

3.2.5. Elaboración del Circuito electrónico

Para elaborar el circuito electrónico se seguirán los siguientes pasos, los mismos que ayudaran a dejar la placa lista para soldar los componentes electrónicos.

1) Diseñar las pistas en un programa o software de diseño electrónico como puede ser el cocodrile, del circuito que se va a fabricar.



Figura 3. 11. Diagrama Software para diseño de Pistas⁶⁵

2) Imprimir las pistas en una impresora laser en un papel PnP o lo que es lo mismo en un papel transfer.

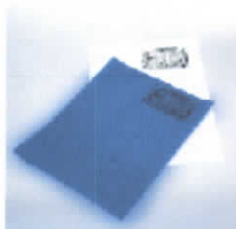


Figura 3. 12. Papel PnP⁶⁶

⁶⁵ <http://zerobots.net/viewtopic.php?f=9&t=9>

⁶⁶ <http://zerobots.net/viewtopic.php?f=9&t=9>

- 3) Recortar la zona de las pistas del circuito electrónico para luego pegarlo en la baquelita.



Figura 3.13. Cortar el diagrama de pistas⁶⁷

- 4) Dejar algunos centímetros al margen del circuito impreso para evitar daños posteriores en la transferencia a la baquelita.



Figura 3. 14. Pistas en Papel PnP⁶⁸

- 5) Limpiar cuidadosamente la baquelita en su lado de cobre tratando de dejar la menor impureza posible en la misma, esto se lo puede hacer con alcohol.



Figura 3. 15 Limpieza de baquelita⁶⁹

⁶⁷ <http://zerobots.net/viewtopic.php?f=9&t=9>

⁶⁸ <http://zerobots.net/viewtopic.php?f=9&t=9>

⁶⁹ <http://zerobots.net/viewtopic.php?f=9&t=9>

6) Pegar el papel PnP en la baquelita en su lado de cobre cuidadosamente.



Figura 3. 16. Fijar papel PnP en la baquelita⁷⁰

7) Pasar la plancha caliente sobre la baquelita de manera que se caliente uniformemente y de esta forma transferir las pistas del circuito impreso en el papel PnP a la superficie de cobre de la baquelita.



Figura 3. 17. Termotransferencia⁷¹

8) Dejar enfriar el papel y luego extraerlo de la baquelita para de esta forma tener ya las pistas del circuito a realizar en la superficie de cobre de la baquelita.



Figura 3. 18. Remover Papel PnP de la baquelita⁷²

⁷⁰ <http://zerobots.net/viewtopic.php?f=9&t=9>

⁷¹ <http://zerobots.net/viewtopic.php?f=9&t=9>

⁷² <http://zerobots.net/viewtopic.php?f=9&t=9>

9) Sumergir y agitar la baquelita en acido de cloruro férrico para de esta forma por medio de la corrosión eliminar el cobre innecesario de la placa.



Figura 3. 19. Acido de Cloruro Férrico⁷³

10) Después de la corrosión del cobre innecesario solo quedará el cobre de las pistas dibujadas para el circuito.



Figura 3. 20. Pistas en la baquelita⁷⁴

⁷³ <http://zerobots.net/viewtopic.php?f=9&t=9>

⁷⁴ <http://zerobots.net/viewtopic.php?f=9&t=9>

11) Secar la placa completamente y limpiarla con algún disolvente o alcohol.



Figura 3. 21. Limpieza de la baquelita⁷⁵

12) Revisar todas las pistas del circuito para de esta forma evitar que existan corto circuitos.



Figura 3. 22. Baquelita con pistas listas para perforar⁷⁶

13) Perforar los orificios de la placa donde iran colocado los componentes electrónicos de el circuito a realizar y colocarlos para posteriormente soldarlos y asi tener el circuito listo para su funcionamiento.

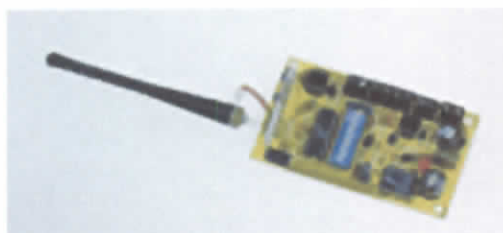


Figura 3. 23. Circuito listo⁷⁷

⁷⁵ <http://zerobots.net/viewtopic.php?f=9&t=9>

⁷⁶ <http://zerobots.net/viewtopic.php?f=9&t=9>

3.2.7. Mantenimiento y Pruebas

El sistema fue expuesto a una serie de pruebas para determinar su funcionalidad, entre las cuales se pueden detallar las siguientes:

Ingreso de datos: esto se lo efectuó de forma tradicional ingresando datos nuevos a la base de datos en los formularios correspondientes, siendo los datos a ingresar, datos validos, inválidos, erróneos, y condicionantes. Teniendo prueba satisfactoria del funcionamiento al receptar y guardar solo datos validos.

Modificación de Datos: se realizo escogiendo varios registros de la base de datos y cambiando la información de uno o todos sus campos desde los formularios correspondientes en el sistema teniendo resultados exitosos al momento de actualizar las tablas.

Verificación de Datos: se programo todos los controles necesarios para validar los datos desde el sistema, es decir los datos que serán destinados a la base de datos ya irán filtrados y con los tamaños y tipos adecuados para su almacenamiento.

Prueba de Envío de Pines al Puerto Paralelo: Para esta prueba se uso un protoboard con leds colocados de tal forma que cada uno reciba una señal enviada desde el puesto paralelo, por medio del Sistema de Control de parqueadero. Siendo exitosa la prueba el trabajar el sistema con los leds de las forma adecuada.

Búsquedas de Información: para esto se ingreso por medio del formulario de búsqueda distintos parámetros a buscar siendo los resultados satisfactorios al momento en que la información requerida fue mostrada de una manera correcta, y al

⁷⁷ <http://zerobots.net/viewtopic.php?f=9&t=9>

no encontrar ningún tipo de resultado de búsqueda si el parámetro de búsqueda es incorrecto o equivoco.

Mantenimiento del Sistema: este será efectuado una vez instalado el sistema en el parqueadero ha funcionar para de esta forma detectar o mejorar, o corregir algún error de sistema que se pueda encontrar ya en su uso real, para esto se tomara en cuenta los siguientes parámetros: instalaciones eléctricas, instalación del circuito, instalación del software, adhesión del circuito con el PC y el Torniquete a controlar, entre otros.

CAPÍTULO IV

4.1. Conclusiones

- La automatización de procesos manuales ayuda a la organización, control y eficiencia de los mismos, permitiendo así mejorar el estilo de vida.
- El sistema facilita a los usuarios del sistema interactuar con el parqueadero y éste puede ser usado por personas con conocimiento informático básico.
- Las aplicaciones de código libre son facilitaron el desarrollo del software ya que no hubo necesidad de adquirir licencias para ciertos procesos.
- Visual Basic permite desarrollar una gran cantidad de aplicaciones que pueden facilitar procesos cotidianos.
- El trabajo con el puerto paralelo es útil para el control de dispositivos electrónicos o eléctricos.
- MySQL brinda confianza y facilidad al momento de gestionar las bases de datos permitiendo múltiples operaciones con los registros, además de ser un sistema gestor multiusuario.
- Xampp ofrece la herramienta perfecta de código libre para administrar la base de datos.

4.2. Recomendaciones

- Definir las necesidades de software que el cliente requiere, para así poder plasmar la mejor idea del sistema siguiendo estos lineamientos.
- Tener en cuenta que el sistema será usado por distintos tipos de usuarios, que pueden ser o no de gran conocimiento informático, entonces es importante realizar la interfaz del sistema sumamente amigable para que esta se pueda adaptar a cualquier tipo de usuario.
- Interpretar la idea de automatización del proceso con el cliente para poder plasmar la misma de una forma concreta y mejorarlo.
- Usar un software de gestión de bases de datos que sea lo suficientemente potente y fácil de administrar.
- Analizar cuidadosamente el circuito electrónico a usar como intercomunicados entre el PC y los dispositivos eléctricos para evitar quemar o dañar el puerto paralelo.
- Seleccionar un computador con las características adecuadas para que el sistema funcione correctamente y no presente problemas de rendimiento.
- Tomar en cuenta el funcionamiento de los componentes electrónicos y el voltaje de corriente con el cual trabajan para evitar estropearlos con un sobre voltaje.

Bibliografía

Consultada

- Gary Cornel, Manual de visual basic 5, España: Osborne/Mc Graw-Hill, 1997
- Oficina de Asistencia al Extranjero en Administración Tributaria, Análisis y diseño de Sistemas Modulo III y IV, Mexico: Editorial Diana
- Equipo de Redactores de la editora de textos y papelería Editexpa, Informática Nuevo Milenio, Editorial Dimacor, 2001
- Juan Jose Castañeda León, Visual Basic 6.0 como debe ser, Perú: Megabyte, Abril,2002
- Prof. Ángel Zetina M. y Ing. Ángel Zetina C., Electronica Basica, Mexico: Limusa-Noriega Editores, 2003
- Marco Antonio Tiznado Santana, Visual Basic 6.0 Enter Plus, Bogota-Colombia: Mc Graw-Hill Interamericana S.A., Marzo, 2000

Citada

- http://es.wikipedia.org/wiki/Revoluci%C3%B3n_industrial
- <http://automatizacion2008.blogspot.com/>
- <http://www.slideshare.net/wakuman/automatizacin>
- <http://www.monografias.com/trabajos5/electro/electro.shtml>
- http://anylinda-12.blogspot.com/2007_09_01_archive.html
- http://anylinda-12.blogspot.com/2007_09_01_archive.html
- http://anylinda-12.blogspot.com/2007_09_01_archive.html
- http://codigodebarras.com/tema.php?ID=como_nacio
- <http://www.monografias.com/trabajos11/yantucod/yantucod.shtml>
- http://www.metrologicmexico.com/contenido1/informacion_tecnica/codigos_de_barras_de_dos_dimen.php
- <http://www.idautomatica.com/informacion-tecnica/codigo-de-barras.php>
- http://es.wikipedia.org/wiki/Codigo_de_barras

<http://www.planetacurioso.com/2007/01/20/%C2%BFcomo-leer-el-codigo-de-barras/>

<http://www.tecnologiadiaria.com/tag/mysql>

Internet

http://es.wikipedia.org/wiki/Puerto_paralelo Octubre, 2009

<http://www.angelfire.com/pa2/jcgr/tecnica/PP/pp.htm> Octubre, 2009

<http://www.globu.net/pp/> Octubre, 2009

<http://www.globu.net/pp/PP/Intro.htm> Octubre, 2009

<http://es.wikipedia.org/wiki/MySQL> Octubre, 2009

<http://dev.mysql.com/downloads/> Octubre, 2009

<http://mysql.conclase.net/> Octubre, 2009

<http://www.youtube.com/watch?v=2XLeb-7YGzA> Noviembre, 2009

<http://www.desarrolloweb.com/articulos/844.php> Noviembre, 2009

<http://www.sun.com/software/products/mysql/> Noviembre, 2009

http://perso.wanadoo.es/luis_ju/puerto/indexpp.html Noviembre, 2009

<http://www.desarrolloweb.com/articulos/306.php> Noviembre, 2009

<http://www.vb6.us/> Noviembre, 2009

<http://www.elguille.info/vb/default.aspx> Noviembre, 2009

<http://www.codigodebarras.org/> Noviembre, 2009

<http://www.inegi.gob.mx/inegi/contenidos/espanol/ciberhabitat/comercio/cbarras/index.html>

Diciembre, 2009

http://books.google.com.ec/books?id=T-pzjaQK1JgC&pg=PA106&lpg=PA106&dq=codigo+de+barras&source=bl&ots=QteIlrGpjE&sig=mJK4wcFMXvTESAxqU4jUwU5rU98&hl=es&ei=6kfWSdbWGYvltgeJnoXhDw&sa=X&oi=book_result&ct=result&resnum=1#PPA105,M1

Diciembre, 2009

http://es.wikipedia.org/wiki/Esc%C3%A1ner_de_c%C3%B3digo_de_barras

Diciembre, 2009

Glosario

- **Automatización:** Automatización es la tecnología que trata de la aplicación de sistemas mecánicos, electrónicos y de bases computacionales para operar y controlar un proceso manual o una producción.
- **Sensores:** Elemento de un instrumento de medida o de una cadena de medida que está directamente sometido a la acción del mensurando
- **Interfaz:** La interfaz de usuario (IU) es uno de los componentes más importantes de cualquier sistema computacional, pues funciona como el vínculo entre el humano y la máquina.
- **Electrónica:** La **electrónica** es la rama de la física, y fundamentalmente una especialización de la ingeniería, que estudia y emplea sistemas cuyo funcionamiento se basa en la conducción y el control del flujo microscópico de los electrones u otras partículas cargadas eléctricamente.
- **Programación:** Codificación de las órdenes y datos que permiten la creación de un programa o aplicación.
- **data-aware:** componente de datos.
- **Microsoft Access 1.1:** es un programa Sistema de gestión de base de datos relacional creado y modificado por Microsoft para uso personal de pequeñas organizaciones.
- **Depuración:** En programación, la depuración de programas es el proceso de identificar y corregir errores.
- **Código de Barras:** es un código basado en la representación mediante un conjunto de líneas paralelas verticales de distinto grosor y espaciado que en su conjunto contienen una determinada información.

- **Decodificador:** Un decodificador o descodificador es un circuito combinacional, cuya función es inversa a la del codificador, esto es, convierte un código binario de entrada de N bits de entrada y M líneas de salida.
- **MySQL:** es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.
- **XAMPP:** es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl.
- **OCX:** hace referencia a módulos que publican controles y funciones para ser utilizados en programas para Windows, incluyendo especialmente el navegador Internet Explorer. Típicamente, estas librerías se presentan en librerías de enlace dinámico (DLL) almacenadas con extensión.
- **DLLs:** Dynamic Link Library ("Biblioteca de vínculos dinámicos") es un archivo que contiene funciones que se pueden llamarse desde aplicaciones u otras Dlls.
- **Torniquete:** es la barrera metálica controlada manual o automáticamente que permite al ingreso o salida del un parqueadero.
- **DFD:** es una representación gráfica del "flujo" de datos a través de un sistema de información. Un diagrama de flujo de datos también se puede utilizar para la visualización de procesamiento de datos (diseño estructurado).
- **Puerto Paralelo:** es una interfaz entre una computadora y un periférico cuya principal característica es que los bits de datos viajan juntos enviando un byte completo o más a la vez. Es decir, se implementa un cable o una vía física para cada bit de datos formando un bus.

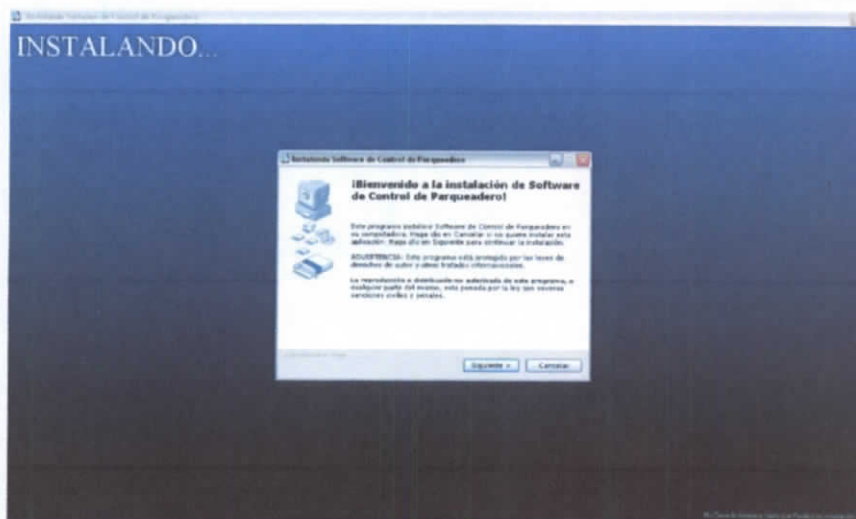
- **Dato:** es una representación simbólica (numérica, alfabética, algorítmica etc.), atributo o característica de una entidad. El dato no tiene valor semántico (sentido) en sí mismo, pero convenientemente tratado (procesado) se puede utilizar en la realización de cálculos o toma de decisiones. Es de empleo muy común en el ámbito informático.
- **Logeado:** Es iniciar una sección (Log In) habitualmente mediante un nombre de usuario y contraseña.
- **Privilegios:** es el honor o el permiso para realizar una actividad garantizada a una persona o usuario.

Anexo 1

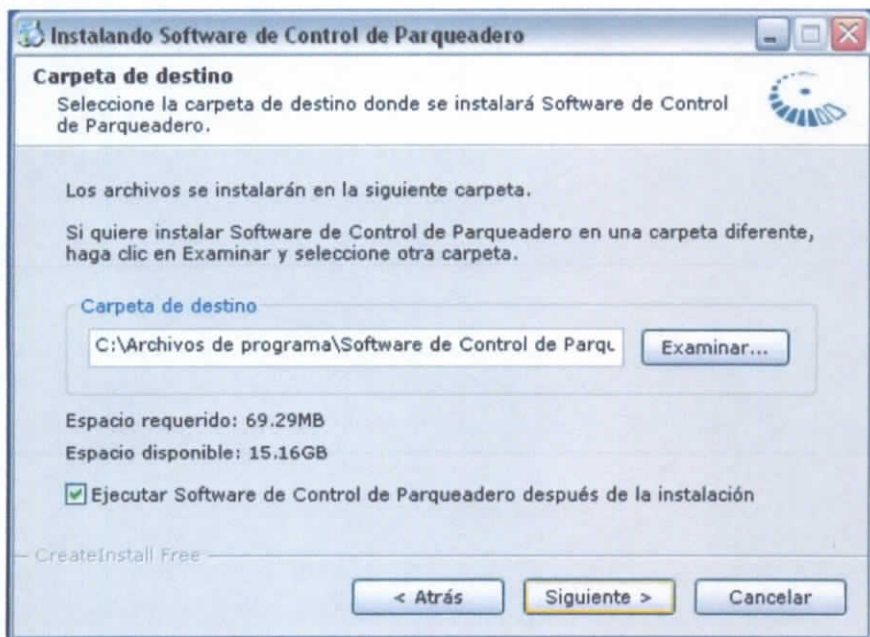
Manual de usuario y de instalación

Instalación del Software

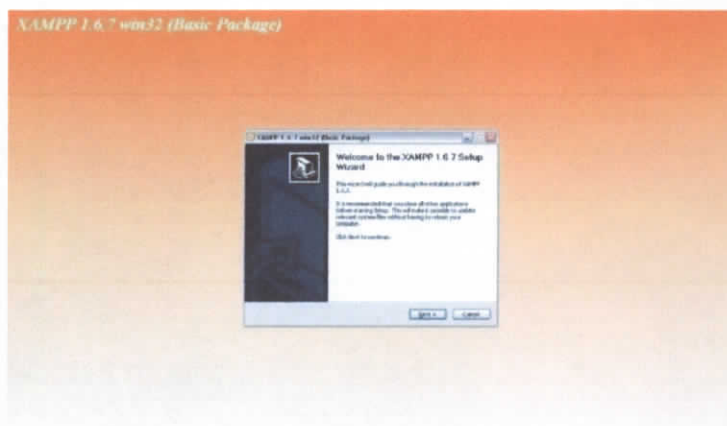
Para instalar el software es necesario únicamente dar doble clic en el icono del instalador el cual irá desplegando pantallas de lo que se está cargando en el computador incluyendo las aplicaciones externas que el sistema de control de parqueadero necesita para su correcto funcionamiento.



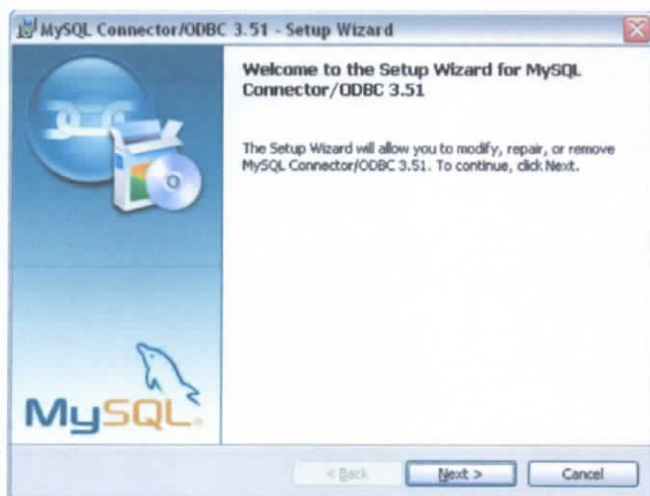
La imagen anterior representa la ventana principal del instalador del sistema de Control de Parqueadero. En ella solo dar siguiente.



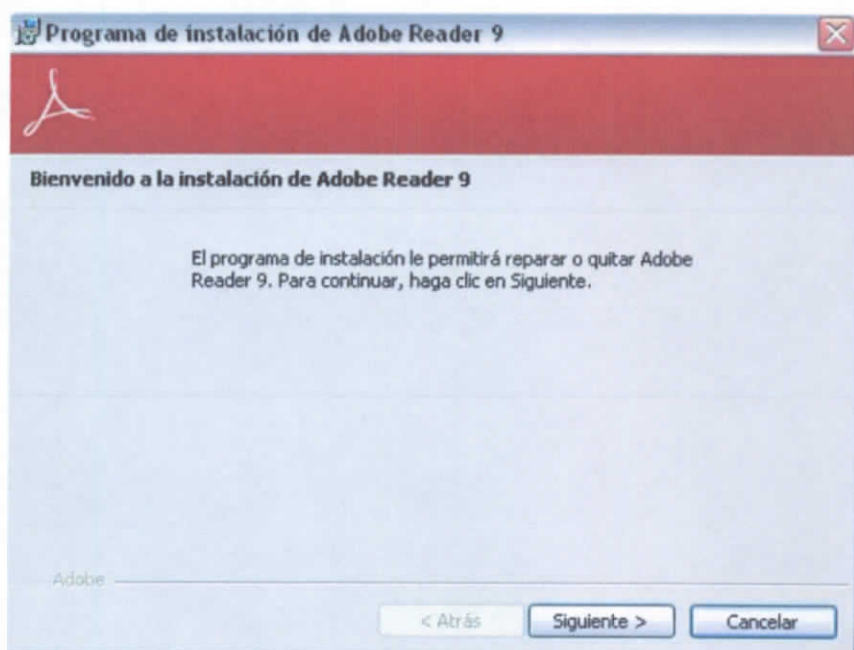
En la pantalla anterior se escoge la ruta donde se quiere que el sistema sea instalado, se recomienda dejarla por defecto para así evitar posibles errores de links y referencias.



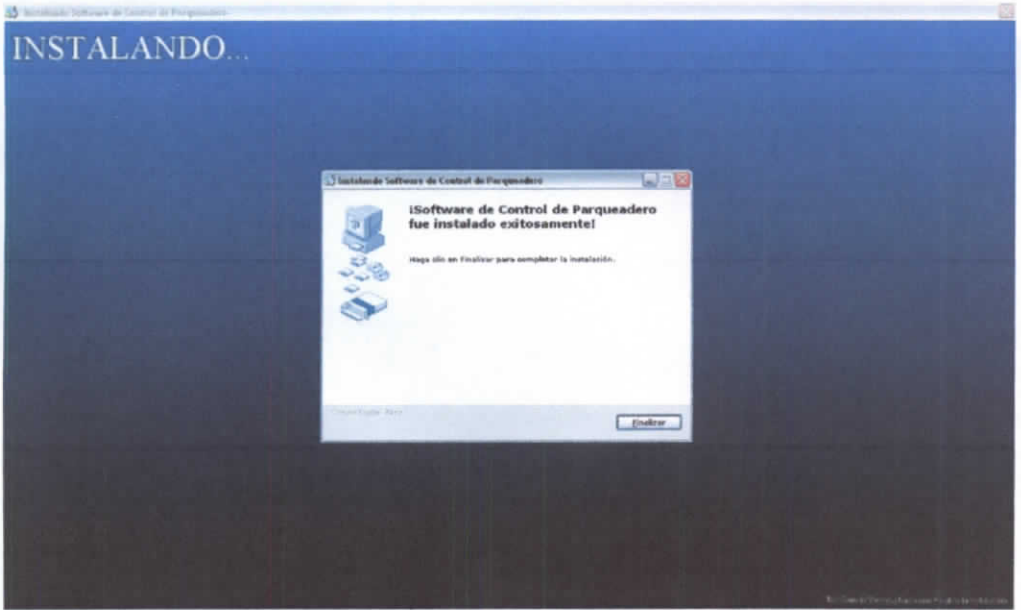
Seguidamente se desplegará la pantalla de instalación del servidor de bases de datos, en la cual no se cambiará ningún parámetro y dar clic solamente en los botones de siguiente hasta que finalice.



Esta pantalla es la aplicación de instalación del Driver MySQL, el cual sirve como enlace entre el servidor SQL y el Software del sistema. De la misma forma que en el servidor no se cambió ningún parámetro, solo dar siguiente hasta que finalice.



Finalmente pedirá instalar Acrobat Reader que es la aplicación que permitirá revisar los documentos PDF como los manuales del sistema.



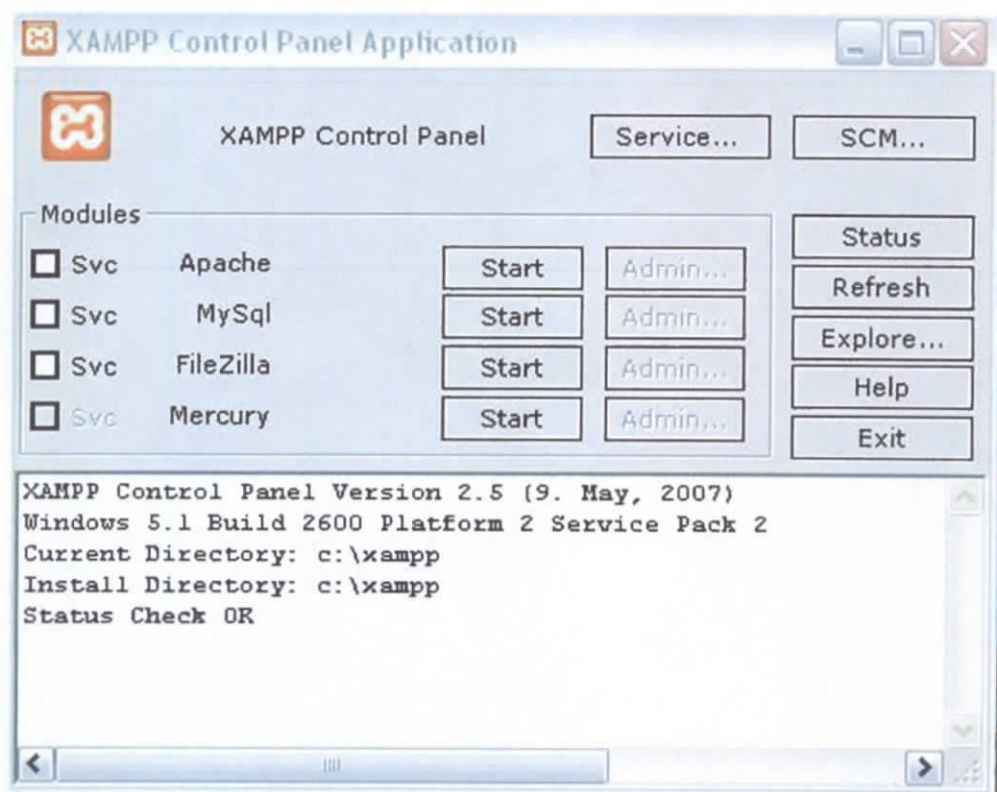
Una vez terminadas la instalación se podrá dar clic en finalizar y empezar a ocupar el sistema de control de parqueadero.

Iniciar Servidor

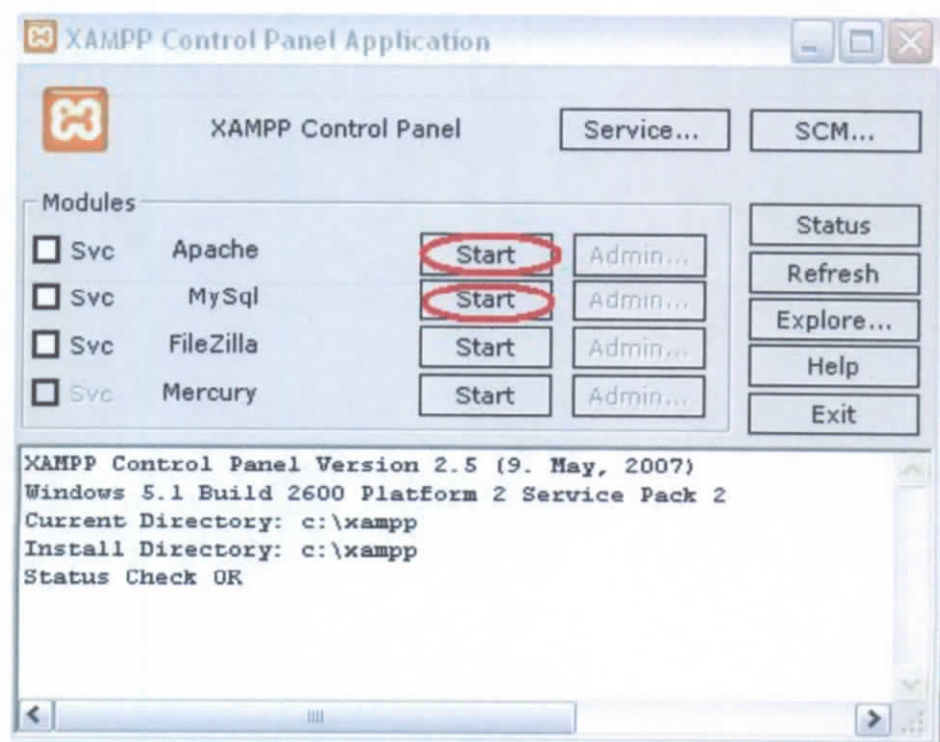
Para que el software de Control de Parqueadero funcione correctamente se debe tener instalado el servidor de base de Datos Xampp. Y una vez instalado es indispensable inicializarlo antes de ejecutar el software.

1. Abrir el Panel de Control del servidor. Desde el icono que por defecto la instalación lo crea en el Escritorio.

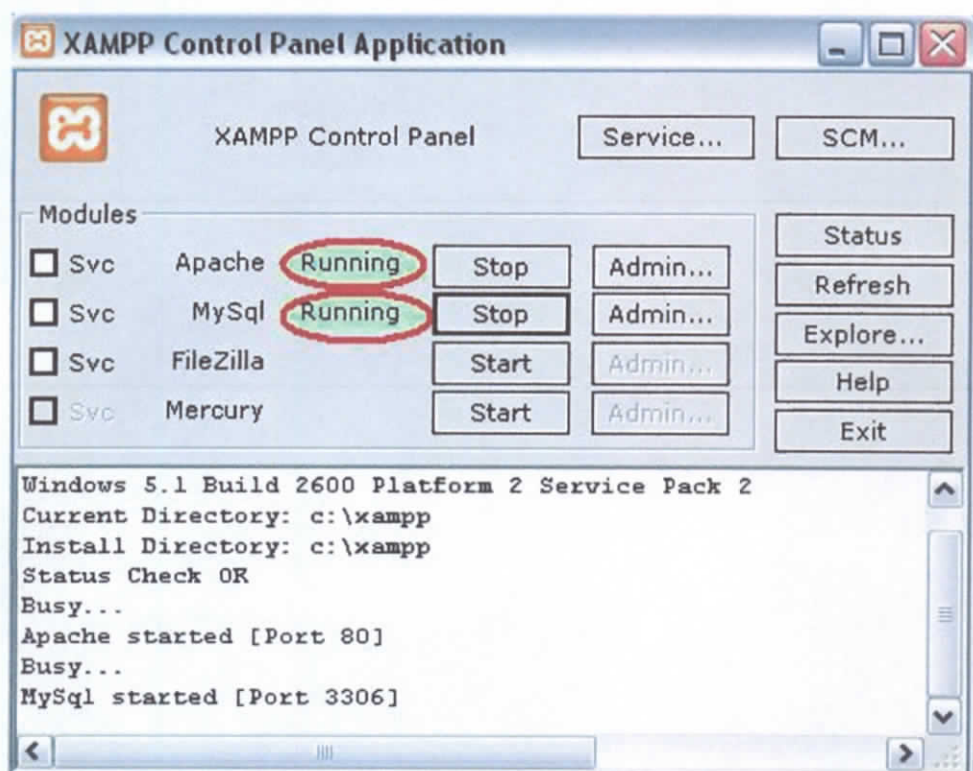




2. Ejecutar la inicialización de los servidores.



3. comprobar que se estén ejecutando correctamente.

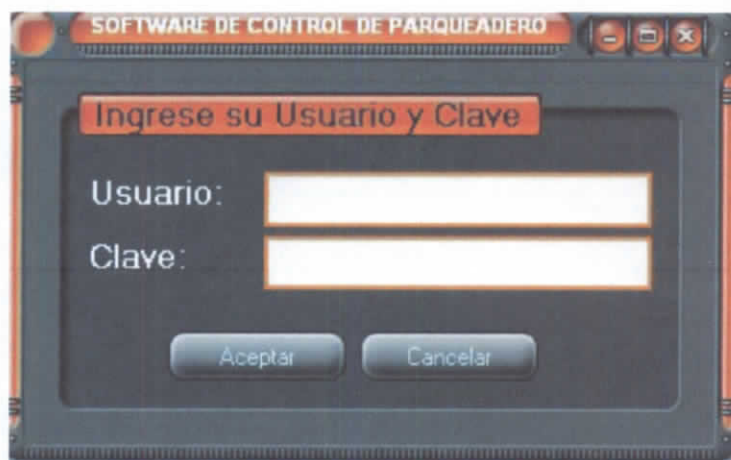


Esto ayudará a la hora de tener acceso a los datos del sistema. es sumamente importante que el servidor este inicializado previamente a la ejecución del software puesto que de no ser así, se provocará un error de conexión con el servidor.

Ingreso al Sistema

Para ingresar al Sistema es únicamente necesario dar doble clic en el icono del sistema y seguidamente aparecerá la ventana de ingreso al sistema donde es requerido un usuario y una clave.

Cada usuario tiene una clave que lo identificara como administrador u operador del sistema, dando o restringiendo privilegios dependiendo del Tipo de usuario.



Control del Torniquete

La ventana de Control del Torniquete es la ventana principal del sistema esta ayudará a abrir y cerrar el torniquete, es decir permitirá que los usuarios del Parqueadero puedan ingresar o salir con su vehículo del mismo.

En esta parte el sistema muestra algunas opciones como son El Control Automático, El Control Manual, y el Bloqueo, estas sirven como su mismo nombre lo expresa:

El Control Automático: Permitirá el ingreso o salida del parqueadero de los vehículos de una forma automática es decir que cada usuario puede ingresar su cedula por medio de los lectores de códigos para que el sistema le permita entrar o salir del estacionamiento.

El Control Manual: este ayudará a que el usuario del systema pueda permitir el ingreso o salida de algún vehículo.

Y Finalmente el Bloqueo: este ayuda a bloquear completamente el control del torniquete de entrada o salida.



En la Parte inferior de la ventana se muestra informacion inportante sobre el usuario que está entrando o saliendo del parqueadero.

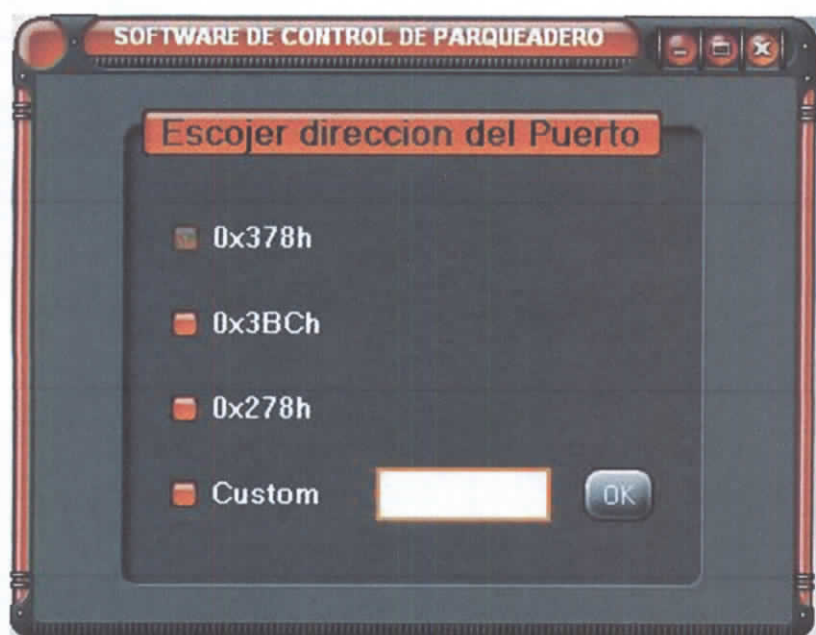
Herramientas



Puerto Paralelo

Las Opciones del Puerto Paralelo radican directamente en el control de la dirección del puerto, a la que el sistema envía los pines para poder de esta forma controlar el torniquete.

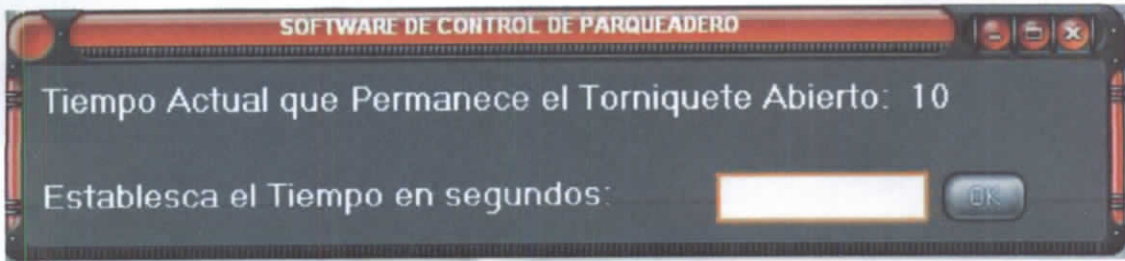
Existen tres tipos de direcciones estándar para el puerto paralelo que son las que el software presenta a su elección, además permite escribir una dirección personalizada en el caso que el equipo cuente con alguna otra dirección.



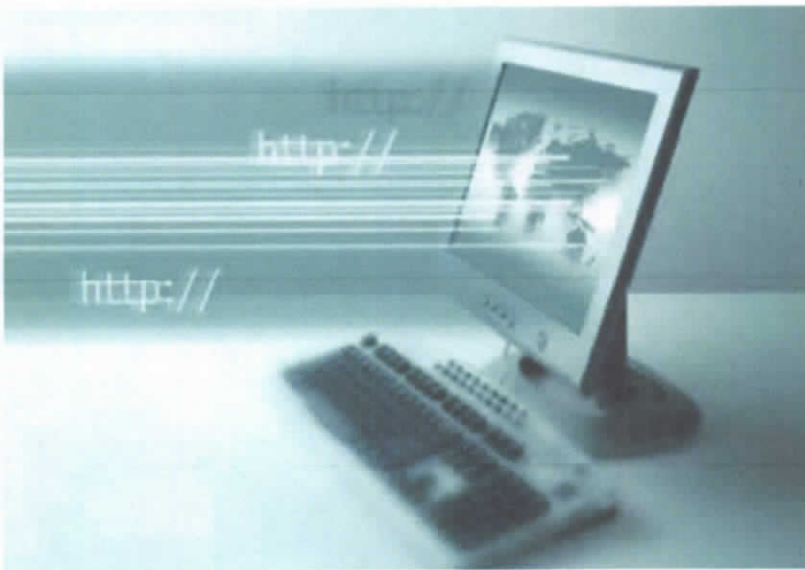
Torniquete

Las opciones del Torniquete se encuentran disponibles para elegir el tiempo que el torniquete permanece abierto mientras que un carro ingresa o sale del parqueadero.

Esto dependerá exclusivamente del usuario del sistema y de los usuarios del Parqueadero, ayudará a evitar accidentes que pueden ocurrir si el torniquete se cerrara demasiado rápido.

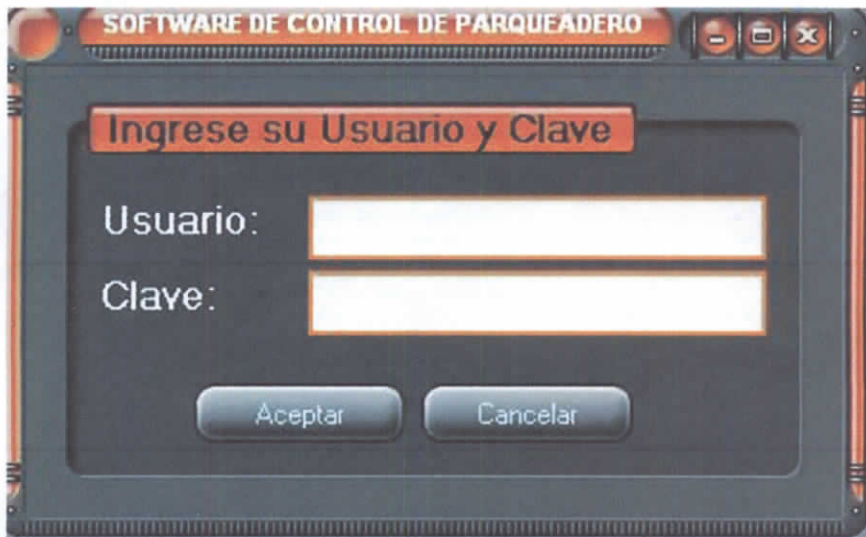


Nuevo Usuario



Usuario del Sistema

Permite que se cambie de usuario ya sea entre operador a un administrador o viceversa.



Usuario del Parqueadero

Esta es una ventana se muestra el usuario que ha iniciado sesión y con el que se está trabajando.



The image shows a software window titled "Opciones de Usuario del Sistema". It is divided into two main sections:

- Usuario:** This section contains a dropdown menu for selecting a user. Below it are three buttons: "Nuevo", "Modificar", and "Eliminar".
- Nuevo Usuario:** This section is for creating a new user. It includes:
 - A "Tipo:" dropdown menu currently set to "AD".
 - Three text input fields labeled "Nombre:", "Clave:", and "Confirme Clave:".
 - Three buttons at the bottom: "Guardar", "Aceptar", and "Cancelar".

Datos de Usuarios del Parqueadero

La ventana de usuarios del parqueadero permite crear, modificar o eliminar usuarios que tienen acceso al estacionamiento.

Opciones de Usuario del Parqueadero

Usuarios

Tipo: Cedula:

Nombre:

Direccion:

Telefono: Celular:

Mail:

Buscar

Nuevo

Eliminar

Modificar

Vehiculo

Marca: Modelo:

Color: Año:

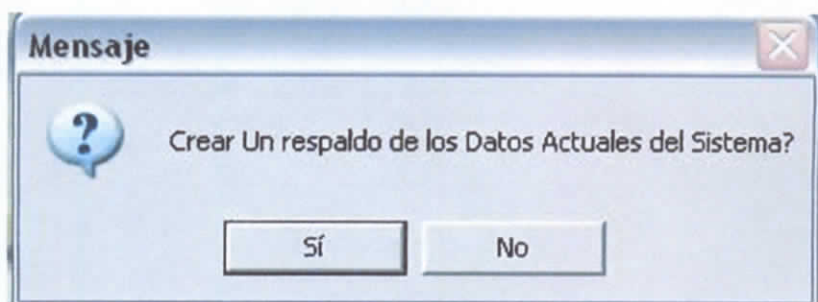
Placa: Matricula:

En la Parte Superior se muestra toda la información correspondiente a los datos personales del usuario, mientras que en la parte inferior permite visualizar la información del auto con que el usuario ha sido registrado.

Backups

EXPORTAR DATOS

Para exportar los datos del sistema o lo que quiere decir crear una copia de seguridad de la base de datos, hacer clic en el menú Datos del Sistema en la opción "Backups" y seguidamente en "Hacer un Respaldo de los Datos", esto desplegará una ventana de mensaje pidiendo la confirmación para crear el respaldo.



Si se hace clic en el botón "Si" esto creara una carpeta de la base de datos con todos los datos que esta contiene actualmente, en una ruta especificada ya en e sistema y con nombre de carpeta con la fecha actual.

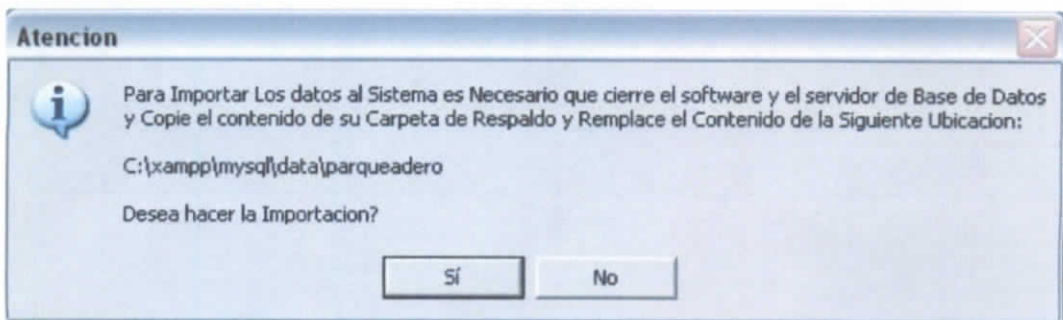
Si se desea acceder a esta carpeta de respaldos seguir la siguiente ruta por defecto:

C:\Archivos de programa\Software de Control de Parqueadero\BACKUP
DATABASE

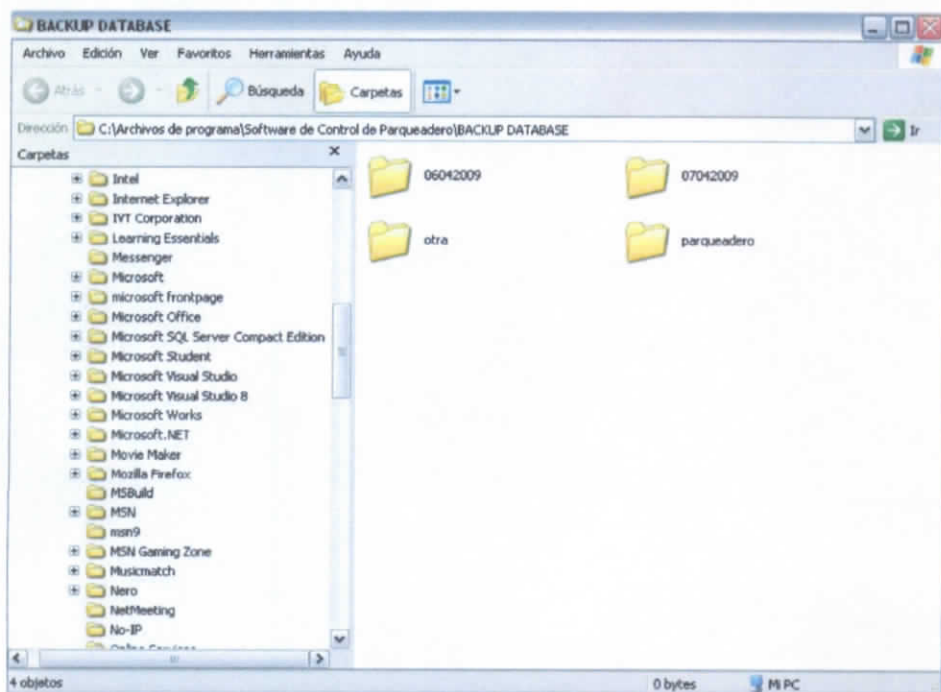
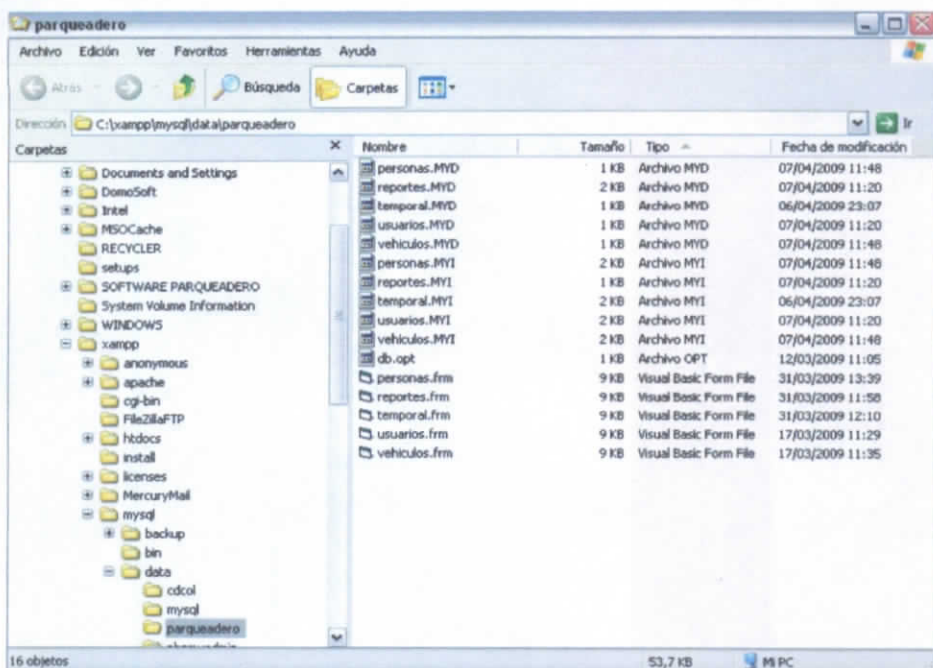
IMPORTAR DATOS

Para importar datos, o lo que quiere decir traer una base de datos anterior o nueva a funcionar dentro del sistema, es importante salir de la aplicación y copiar a la carpeta de respaldo en la ubicación adecuada del servidor de base de datos que se usa.

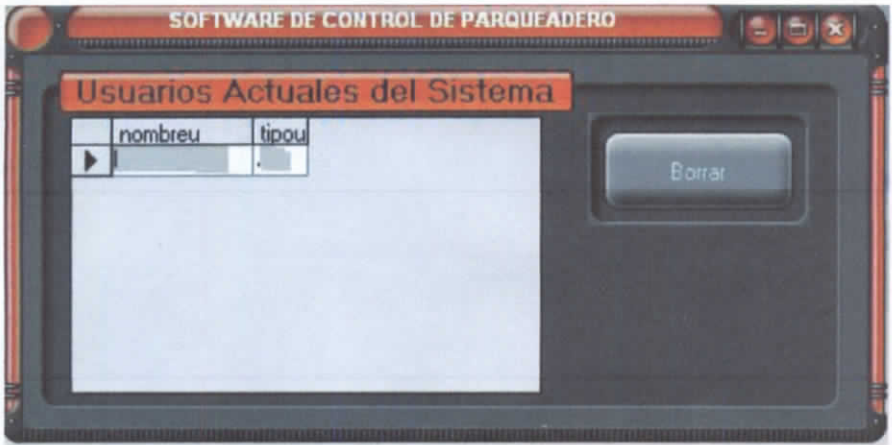
Desde el sistema se puede escoger la opción "Importar Datos" desde el menú "Datos del Sistema" en el submenú "Backups". Esto desplegará un mensaje el mismo que indica que se necesita cerrar la aplicación y copiar manualmente los datos, al hacer clic en el botón "Si" se cerrará la aplicación y se despliegan dos ventanas.



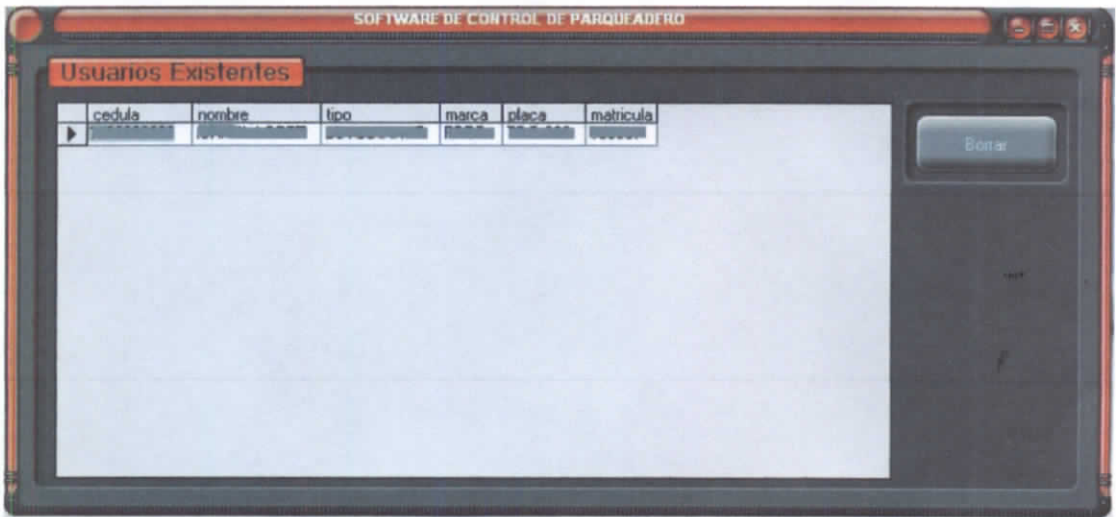
Las ventanas que se muestra automáticamente el software después de cerrarse son dos, la primera es la ruta donde están guardados los archivos de la base de datos, y la segunda es la ruta que por defecto se guardan las carpetas de respaldo.



Limpiar datos del Usuarios del Sistema:



Limpiar datos de Usuarios del Parqueadero:

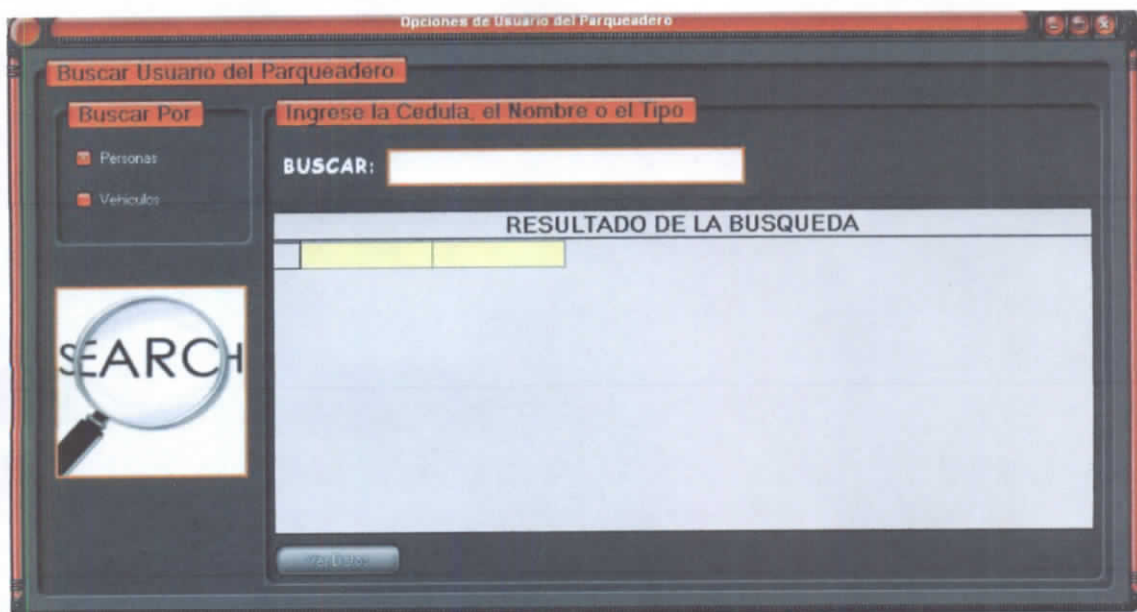


Buscar



Buscar Personas

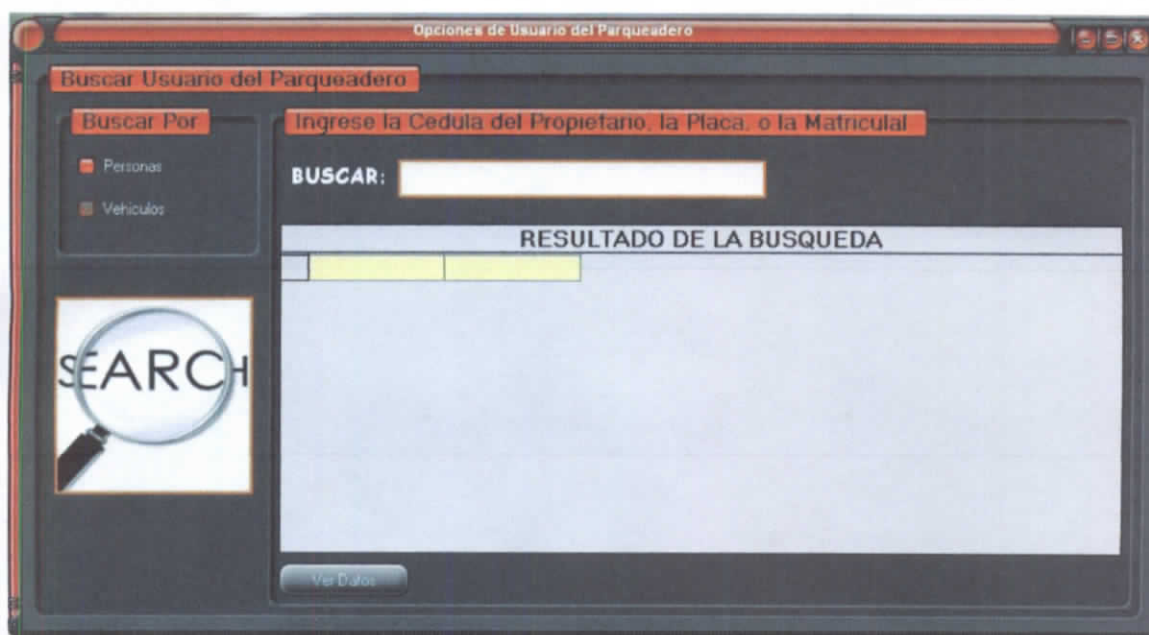
Las ventana de buscar personas permite hacer búsquedas por medio del nombre, la cedula, o el tipo de usuario que se desee encontrar o saber si esta registrado o también modificar sus datos.



El encontrar un usuario en una búsqueda se puede hacer doble clic sobre este registro y luego clic en "ver datos" para que se despliegue en una pantalla todos los datos de este usuario, permitiendo así una mejor visualización o a su vez la modificación de algún dato de este usuario.

Buscar Vehiculos

Las ventana de buscar vehículos permite hacer búsquedas por medio de la cedula del propietario, la placa o la matricula del vehículo ha encontrar.



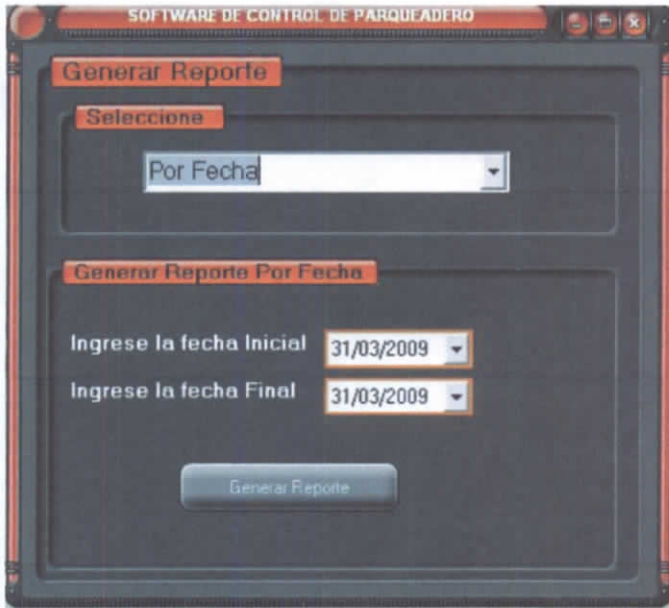
El encontrar un vehículo en una búsqueda se puede hacer doble clic sobre este registro y luego clic en "ver datos" para que se despliegue en una pantalla todos los datos de este usuario y vehículo asignado, permitiendo así una mejor visualización o a su vez la modificación de algún dato de este usuario o su vehículo.

Reportes



Generar Reportes

La ventana generar reportes ayuda a crear diversos tipos de reportes, basta con escoger el tipo de reporte que se desee crear y seguidamente desplegara la información del reporte creado, permitiendo así su visualización o su impresión.



SOFTWARE DE CONTROL DE PARQUEADERO

Generar Reporte

Seleccione

Por Fecha

Generar Reporte Por Fecha

Ingrese la fecha Inicial 31/03/2009

Ingrese la fecha Final 31/03/2009

Generar Reporte

Reporte de usuarios

Los reportes de Usuarios son creados con la finalidad de saber cuántos usuarios se tienen registrados en el sistema ya sean usuarios del sistema o usuarios del parqueadero.

SOFTWARE DE CONTROL DE PARQUEADERO

Zoom 100%

USUARIOS DEL PARQUEADERO

CEDULA	1803223923				
HOMBRE	MARIA ISABEL TRADO				
DIRECCION	LA CANTERA 12345				
TELEFONO	032944666	CELULAR	094483271	MAIL	isa01862@hotmail.com
ESCUELA	SISTEMAS	NIVEL	8	HORARIO	DIURNO
CARGO	-			TIPO	ESTUDIANTE

CEDULA	1803223924				
HOMBRE	JAVIER ESPIN				
DIRECCION	EL RECREO				
TELEFONO	032879865	CELULAR	094483265	MAIL	JAVI_ES7@HOTMAIL.COM
ESCUELA	-	NIVEL	-	HORARIO	-
CARGO	DIRECTOR DE ESCUELA			TIPO	ADMINISTRATIVO

Página: 1