

UNIVERSIDAD POLITECNICA SALESIANA

SEDE QUITO - CAMPUS SUR

**CARRERA DE INGENIERIA DE SISTEMAS
MENCION EN ROBOTICA E INTELIGENCIA ARTIFICIAL**

**“DISEÑO E IMPLEMENTACIÓN DEL HARDWARE Y SOFTWARE
DE UN PROTOTIPO DE REGISTRO LABORAL PARA EL GRUPO
SOCIAL FONDO POPULORUM PROGRESSIO (GSFEPP).”**

TESIS PREVIA A LA OBTENCION DEL TITULO DE INGENIERO DE SISTEMAS

GIOVANNY JAVIER MADRID PONCE

DIRECTOR: ING. RAFAEL JAYA.

Quito, Julio 2011

DECLARACIÓN

Yo Giovanni Javier Madrid Ponce, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedo mi derecho de propiedad intelectual correspondientes a este trabajo a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual por su reglamento y por la normatividad institucional vigente.

Giovanny Javier Madrid Ponce

AGRADECIMIENTOS

A mi papá y mamá por haberme inculcado siempre el valor del estudio, por apoyarme en todos los momentos y darme un ejemplo de superación para que me desarrolle como una persona de bien.

A la Universidad Politécnica Salesiana y todos los docentes que han contribuido en mi desarrollo profesional.

A mi tutor por el apoyo, paciencia, tiempo y conocimiento brindado.

A la Fundación Fondo Ecuatoriano Populorum Progressio y el personal que me colaboró para que esta tesis pueda ser planteada y desarrollada.

A mis familiares, compañeros y amigos por los ánimos y consejos que me dieron.

Antecedentes

El Grupo Social Fondo Ecuatoriano Populorum Progressio (GSFEPP) es una organización sin fines de lucro ONG que tiene como objetivo el impulsar la labor social en nuestro país, tiene su matriz en el sector la Floresta en las calles Mallorca N24-275 y Av. Coruña en la ciudad de Quito, en la cual laboran alrededor de unas 60 personas.

El sistema actual que usa el departamento de personal para el registro laboral del personal del GSFEPP se basa en un reloj tarjetero, el cual imprime en una tarjeta la hora de entrada, la hora de salida almorzar, la hora de entrada del almuerzo y la hora de salida del personal, cabe recalcar que por cada empleado hay una tarjeta mensual.

Para obtener información de las horas laboradas al mes, así como permisos, faltas y generación de reportes, es necesario ingresar los datos que se encuentran en las tarjetas de cada empleado a una hoja de Excel para hacer los debidos cálculos.

El proceso de digitación se hace diariamente para mantener actualizado el registro y al hacerlo de forma manual, es susceptible a errores a parte de dedicar un tiempo considerable en el proceso lo que conlleva a generar reportes con un mal cálculo de las horas laboradas, perjudicando al empleado o a la empresa.

Debido al crecimiento de personal que ha tenido la empresa, el proceso del Registro Laboral del Personal ha quedado obsoleto, trayendo al Departamento de Recursos Humanos una serie de problemas al rato de emitir reportes e informes.

Por lo anteriormente citado, el Departamento de Recursos Humanos como responsable de dicho proceso, ha visto indispensable y necesario la implementación de un sistema que este acorde a los requerimientos actuales de la empresa, viendo beneficioso y provechoso el desarrollo de esta tesis.

Resumen

Mediante el diseño e implementación de un Prototipo de Registro Laboral, se busca automatizar el proceso de ingreso manual de los datos, así como, el almacenamiento y el tratamiento de las horas laboradas por el personal, lo que permitirá de una manera eficiente y confiable realizar los cálculos y obtener reportes sin riesgo de errores, teniendo como beneficiarios al Departamento de Recursos Humanos así como a los empleados.

Esta tesis tiene como objetivo el demostrar que con un bajo costo se puede lograr dar solución algunos problemas que en el mercado serian mucho más costosos.

Mediante la construcción de un Hardware formado por un teclado, display LCD y mediante la comunicación serial RS232 se logra obtener la adquisición de datos.

Para el desarrollo del software se busco un gestor de BDD libre como MySql que integrado con la aplicación desarrollada en punto NET permite dar solución al almacenamiento digital de los registros diarios.

La mayor parte de la base teórica sobre los componentes usados ha sido vista en la Universidad en las materias de especialidad y se ha reforzado los temas con investigaciones pertinentes.

En las fases de diseño se presenta los circuitos y diagramas tanto de el hardware como software, así mismo luego de la implementación se presentan las pruebas en las cuales se trata de de seguir metodologías que permitan detectar los errores, también se realiza una comparación del costos beneficios para la empresa usando el sistema anterior y con el sistema actual.

En base a las conclusiones se presentan también las recomendaciones para mejorar el sistema en un futuro.

Presentación

El presente documento se halla dividido en cinco capítulos los cuales abarcan todo el desarrollo de la tesis propuesta y que se espera sirva como objeto de consulta para futuros proyectos similares.

Con la finalidad de dar una visión rápida de lo que se podrá encontrar en cada capítulo, se presenta un resumen de cada uno de estos.

Capítulo I

En este capítulo consta toda investigación que ha sido necesaria para actualizar el conocimiento más profundo de los temas relacionados con la tesis presentada. Gira en torno a tres ejes principales como los microcontroladores, la comunicación serial y las bases de datos.

Capítulo II

Se realiza el análisis de los requerimientos de la Fundación FEPP tanto del Hardware como del Software y demás aspectos que podrían servir para identificar las necesidades y sus posibles soluciones.

Capítulo III

Se presenta la fase de diseño tanto del hardware como del software, los diagramas que harán comprender la implementación del prototipo y su software.

Capítulo IV

Se documenta con imágenes las pruebas realizadas y sus resultados tanto del hardware como del software.

Capítulo V

Se dan las conclusiones de la tesis, las cuales redactarán las expectativas a las que se ha llegado y las recomendaciones que se pueden dar a futuro tanto a la fundación como a los lectores del presente documento.

CONTENIDO

DECLARACION.....	1
AGRADECIMIENTOS.....	2
ANTECEDENTES.....	3
RESUMEN.....	4
PRESENTACION.....	5
CONTENIDO.....	7
INDICE DE FIGURAS.....	12
INDICE DE TABLAS.....	14
INDICE DE CUADROS.....	15
CAPITULO 1 MARCO TEORICO	16
1.1 LOS MICROCONTROLADORES	16
1.1.1 INTRODUCCIÓN	16
1.1.2 DEFINICIONES.....	16
1.1.2.1 Controlador	16
1.1.2.2 Microcontrolador μ C.....	16
1.1.3 DIFERENCIA ENTRE MICROPROCESADOR Y MICROCONTROLADOR.	17
1.1.4 ARQUITECTURA	18
1.1.4.1 Arquitectura Von Neumann	18
1.1.4.2 Arquitectura Harvard.....	18
1.1.5 RECURSOS COMUNES A TODOS LOS MICROCONTROLADORES.	19
1.1.6 RECURSOS ESPECÍFICOS	21
1.1.7 FAMILIAS DE MICROCONTROLADORES.....	23
1.1.8 APLICACIONES DE LOS MICROCONTROLADORES.	25
1.1.9 EL MERCADO DE LOS MICROCONTROLADORES.	25
1.1.10 SELECCIÓN DEL MICROCONTROLADOR.....	25
1.1.11 MICROCONTROLADORES MÁS POPULARES.....	27
1.2 COMUNICACIÓN SERIAL	29
1.2.1 INTRODUCCIÓN	29
1.2.2 CONCEPTO	29
1.2.3 HISTORIA.....	29
1.2.4 TRANSMISIÓN MODULADA EN AMPLITUD	30

1.2.5	ESTÁNDARES CON FORMATO MARCA/ESPACIO	30
1.2.6	Consideraciones en la comunicación serie.....	30
1.2.7	VELOCIDAD DE TRANSMISIÓN.....	31
1.2.8	LA BASE DE RELOJ	31
1.2.9	TIPOS DE COMUNICACIÓN	32
1.2.10	MODOS DE TRANSMISIÓN.....	32
1.2.10.1	La transmisión asíncrona	33
1.2.10.1.1	Bit de inicio y bit de parada	33
1.2.10.1.2	Reglas de transmisión asíncrona.....	34
1.2.10.2	La transmisión síncrona	34
1.2.10.3	Detectar errores en la comunicación.....	36
1.2.10.4	Paridad par	36
1.2.10.5	Paridad impar	37
1.2.10.6	Método checksum	37
1.2.11	ALGUNOS EJEMPLOS DE TECNOLOGÍA DE COMUNICACIÓN SERIAL	38
1.2.12	COMUNICACIÓN CON EL MAX232 - CONVERSOR TTL-RS232	39
1.2.12.1	Conexión MAX 232 PIC y PC.....	40
1.2.12.2	El conector DB9 del PC.....	40
1.2.13	COMUNICACIÓN SERIAL USB	41
1.2.13.1	Introducción	41
1.2.13.2	Velocidad de transmisión.....	42
1.2.13.3	Conectores	43
1.2.13.4	Características	43
1.3	BASES DE DATOS	44
1.3.1	DEFINICIÓN	44
1.3.2	UTILIZACIÓN DE BDD.....	45
1.3.3	ADMINISTRACIÓN DE BASES DE DATOS.....	45
1.3.4	CARACTERÍSTICAS DE UN DBMS	46
1.3.5	LOS DIFERENTES MODELOS DE BASE DE DATOS	47
1.3.6	TIPOS DE BASE DE DATOS	50
1.3.6.1	Según la variabilidad de los datos almacenados	50
1.3.6.2	Según el contenido	50
1.3.7	VENTAJAS DE LAS BASES DE DATOS.....	51
1.3.8	DESVENTAJAS DE LAS BASES DE DATOS.....	54
1.3.9	PRINCIPALES DBMS :	55
1.3.10	OTROS DBMS.....	55

CAPITULO II	ANÁLISIS Y REQUERIMIENTOS DEL HARDWARE Y SOFTWARE	56
2.1	ANÁLISIS DE REQUERIMIENTOS DEL HARDWARE	56
2.1.1	INTRODUCCIÓN	56
2.1.2	PROPÓSITO	56
2.1.3	DEFINICIONES, ABREVIACIONES	56
2.1.4	PROCEDIMIENTO	56
2.1.5	IDENTIFICACIÓN Y SELECCIÓN DE LOS COMPONENTES DEL HARDWARE	60
2.1.5.1	FUNCIONES DEL HARDWARE	60
2.1.6	DEFINICIÓN DE LOS COMPONENTES SELECCIONADOS	62
2.1.6.1	TECLADO	62
2.1.6.2	DISPLAY LCD	63
2.1.6.3	MICROCONTROLADOR PIC 16F873A	63
2.1.6.4	CARACTERÍSTICAS DEL PIC 16F873A	64
2.1.6.5	ENCAPSULADO	64
2.1.6.6	DESCRIPCIÓN DE LOS PINES	65
2.2	ANÁLISIS Y REQUISITOS DEL SOFTWARE	66
2.2.1	SITUACIÓN ACTUAL DEL SISTEMA DE REGISTRO	66
2.2.2	ESPECIFICACIÓN DE REQUERIMIENTOS DEL SOFTWARE.	67
2.2.2.1	Funciones del software.	67
2.2.2.2	Políticas de la Fundación en cuanto a la asistencia.	67
2.2.2.3	Requisitos Funcionales.....	68
2.2.2.3.1	Conexión con la BDD	68
2.2.2.3.2	Ingreso al programa	69
2.2.2.3.3	Gestión de usuarios.....	69
2.2.2.3.4	Gestión de perfiles de usuarios.....	69
2.2.2.3.5	Gestión de horarios.....	70
2.2.2.3.6	Gestión de Departamentos	70
2.2.2.3.7	Gestión de Empleados.....	70
2.2.2.3.8	Gestión de Reportes.....	70
2.2.3	SELECCIÓN DEL SOFTWARE	71
2.2.3.1	Lenguajes de programación para el microcontrolador	71
2.2.3.2	Lenguaje Seleccionado para programación del microcontrolador	72
2.2.3.3	Lenguaje de desarrollo para la aplicación	72
2.2.3.4	Selección del entorno de desarrollo IDE	73
2.2.3.5	Selección del sistema gestor de Base de Datos	73

CAPITULO III	DISEÑO DEL HARDWARE Y SOFTWARE.....	75
3.1	DISEÑO DEL HARDWARE.....	75
3.1.1	DIAGRAMA DE BLOQUES DEL DISPOSITIVO	75
3.1.2	ESQUEMA DEL CIRCUITO EN PROTEUS	76
3.1.3	DIAGRAMA DEL CIRCUITO.....	77
3.1.4	DESCRIPCIÓN DEL DIAGRAMA DEL CIRCUITO.....	78
3.1.5	RUTEO DEL CIRCUITO.....	79
3.1.6	GENERACIÓN DEL PCB (PRINT CIRCUIT BOARD).....	80
3.1.7	IMPRESIÓN DEL CIRCUITO EN LA BAQUELITA POR TRANSFERENCIA TÉRMICA.....	82
3.1.8	PROGRAMACIÓN DEL PIC.....	83
3.1.8.1	Introducción	83
3.1.8.2	Diagrama de flujo del funcionamiento del Hardware.....	83
3.1.8.3	Protocolo de comunicación PC - PIC.	85
3.1.9	PRERREQUISITOS DE LA COMUNICACIÓN PC-PIC DEL PROTOTIPO.	86
3.1.10	SECUENCIA DE COMANDOS PARA LA COMUNICACIÓN	86
3.1.11	CÓDIGO DE PROGRAMACIÓN DEL PIC.....	87
3.1.12	DISPOSITIVO IMPLEMENTADO.....	90
3.2	DISEÑO DEL SOFTWARE	91
3.2.1	ARQUITECTURA DEL SISTEMA	91
3.2.1.1	Diagrama de clases	91
3.2.1.2	Diagrama de objetos.....	93
3.2.1.3	Diagrama de casos de usos	94
3.2.1.4	Actores.....	96
3.2.1.5	Diagramas de Secuencia	96
3.2.1.6	Descripción de los casos de Uso	102
3.2.2	MODELO DE LA BASE DE DATOS.....	111
3.2.2.1	Diccionario de datos	112
3.2.3	INTERFAZ DE USUARIOS.....	117
3.2.3.1	Interfaz de inicio de sesión.....	117
3.2.3.2	Interfaz de Conexión.....	118
3.2.3.3	Ventana Principal.....	118
3.2.4	MOTOR DE CONEXIÓN HARDWARE – BASE DE DATOS.....	119
3.2.5	ESTRUCTURA DE LA CLASE CONEXIÓN.....	120

CAPITULO IV	PRUEBAS DEL HARDWARE Y SOFTWARE	121
4.1	PRUEBAS DE HARDWARE.....	121
4.1.1	PRUEBAS EN PROTEUS.	121
4.1.1.1	Simulación de Encendido del dispositivo	121
4.1.1.2	Simulación de la estabilización del PIC	121
4.1.1.3	Simulación de la comunicación	122
4.1.1.4	Simulación de la comunicación	122
4.1.1.5	Recibiendo datos de Hora y Fecha (simulado)	123
4.1.1.6	Ingresando Código del empleado (simulado)	123
4.1.1.7	Recibiendo Datos del PC (simulado)	124
4.1.1.8	Simulación del menú de opciones	125
4.1.1.9	Simulación fin de la comunicación (simulado)	125
4.1.2	PRUEBAS EN EL HARDWARE.....	126
4.1.2.1	Encendido del dispositivo.	126
4.1.2.2	Estabilización.....	126
4.1.2.3	Esperando comunicación	127
4.1.2.4	Inicialización de la comunicación	127
4.1.2.5	Ingresando Código del empleado y Envío	128
4.1.2.6	Recepción de Datos desde el PC.....	128
4.1.2.7	Menú	129
4.1.2.8	Selección	129
4.2	PRUEBAS DEL SOFTWARE.	130
4.2.1	VALIDACIÓN DE INICIO DE SESIÓN	131
4.2.2	VALIDACIÓN DE CONEXIÓN CON LA BDD.....	131
4.2.3	VALIDACIÓN CREACIÓN DE USUARIO.	132
4.2.4	VALIDACIÓN CREACIÓN DE PERSONAL.	133
4.2.5	VALIDACIÓN DE PERMISOS, JUSTIFICACIONES Y VACACIONES.	133
4.2.6	PRUEBAS DE IMPRESIÓN DE REPORTES.	134
4.2.6.1	Reporte de Datos del personal	134
4.2.6.2	Reporte de horarios del personal	134
4.2.6.3	Reporte de horas laboradas por el personal.	135
4.3	ANÁLISIS COSTO – BENEFICIO	136
4.3.1	Análisis costo – beneficio para la empresa en moneda local.	136
4.3.2	Análisis Costo – Beneficio para la empresa en tiempo.	137

CAPITULO V	CONCLUSIONES Y RECOMENDACIONES	138
5.1	CONCLUSIONES.....	138
5.2	RECOMENDACIONES	139
GLOSARIO		140
DEFINICIONES, ACRONIMOS Y ABREVIACIONES.....		140
BIBLIOGRAFÍA		144
ANEXOS		146.
MANUAL DEL HARDWARE.....		162
MANUAL DEL SOFTWARE.....		166

INDICE DE FIGURAS

FIG.1.1. ESTRUCTURA DE UN SISTEMA ABIERTO	17
FIG.1.2. EL MICROCONTROLADOR ES UN SISTEMA CERRADO.	18
FIG.1.3 ARQUITECTURA VON NEUMANN	18
FIG.1.4 ARQUITECTURA HARVARD	19
FIG. 1.5 FORMATO SERIE MARCAR/ESPACIO	30
FIG. 1.6 BIT DE INICIO Y PARADA EN TRANSMISIÓN ASÍNCRONA.....	33
FIG. 1.7 TRANSMISIÓN SINCRÓNICA.....	35
FIG. 1.8 INSERCIÓN AUTOMÁTICA DE CARACTERES DE SINCRONISMO.	35
FIG. 1.9 MAX 232	39
FIG. 1.10 CONEXIÓN MAX 232.	40
FIG. 1.11 CONECTOR DB9 HEMBRA Y MACHO.	41
FIG.1.12 CONECTORES USB.....	43
FIG. 1.13 BASE DE DATOS CLIENTE – SERVIDOR	44
FIG. 1.14 ADMINISTRADOR DE BASE DE DATOS.....	46
FIG. 1.15 MODELO BD JERÁRQUICO.	48
FIG. 1.16 MODELO BD DE RED.	48
FIG. 1.17 MODELO BD RELACIONAL.	49
FIG. 1.18 MODELO ORIENTACIÓN A OBJETOS.....	49
FIG. 3.1 DIAGRAMA DE BLOQUES	75
FIG. 3.2 ESQUEMA DEL CIRCUITO EN ISIS	76
FIG. 3.3 DIAGRAMA DEL CIRCUITO	77

FIG. 3.4. ARRASTRANDO ELEMENTOS PARA EL RUTEO	80
FIG. 3.5 ESQUEMA DE LAS PISTAS EN FORMATO BMP	81
FIG. 3.6 ESQUEMA IMPRESO DE LOS ELEMENTOS.	81
FIG. 3.7 IMPRESIÓN EN LA BAQUELITA.	82
FIG. 3.8 DIAGRAMA DE FLUJO FUNCIONAMIENTO DEL HARDWARE	84
FIG. 3.10 DIAGRAMA DE CLASES	92
FIG. 3.11 DIAGRAMA DE OBJETOS	94
FIG. 3.12 DIAGRAMA CASOS DE USOS	95
FIG. 3.13 DIAGRAMA DE SECUENCIA INICIO SESIÓN	97
FIG. 3.14 DIAGRAMA DE SECUENCIA USUARIOS	97
FIG. 3.15 DIAGRAMA DE SECUENCIA PERFILES	98
FIG. 3.16 DIAGRAMA DE SECUENCIA DEPARTAMENTO	99
FIG. 3.17 DIAGRAMA DE SECUENCIA EMPLEADO	100
FIG. 3.18 DIAGRAMA DE SECUENCIA USUARIOS	100
FIG. 3.19 DIAGRAMA DE SECUENCIA SISTEMA	101
FIG. 3.20 DIAGRAMA DE SECUENCIA GESTIÓN FERIADO-VACACIONES.....	101
FIG.3.21 MODELO BASE DE DATOS.....	111
FIG. 3.22 INICIO DE SESIÓN	118
FIG. 3.23 CONFIGURACIÓN CONEXIÓN A BDD	118
FIG. 3.24 VENTANA PRINCIPAL	119
FIG. 4.1 ENCENDIDO DEL DISPOSITIVO EN PROTEUS	121
FIG. 4.2 SIMULACIÓN DE ESTABILIZACIÓN DEL PIC EN PROTEUS.....	121
FIG. 4.3 ESPERANDO COMUNICACIÓN.....	122
FIG. 4.4 INICIALIZACIÓN DE COMUNICACIÓN	122
FIG. 4.5 RECEPCIÓN DE DATOS DE FECHA Y HORA	123
FIG. 4.6 INGRESO DE CÓDIGO DE EMPLEADO.....	123
FIG. 4.7 ENVIANDO DATOS AL PC	124
FIG. 4.8 RECIBIENDO DATOS DEL PC.....	124
FIG. 4.9 MENÚ DE OPCIONES	125
FIG. 4.10 ENVIANDO CÓDIGO DE OPCIÓN SELECCIONADA.....	125
FIG. 4.11 FIN DE COMUNICACIÓN.....	125
FIG. 4.12 ENCENDIDO DE DISPOSITIVO	126
FIG. 4.13 ESTABILIZACIÓN DEL PIC	126
FIG. 4.14 ESPERANDO COMUNICACIÓN.....	127
FIG. 4.15 INICIALIZACIÓN DE LA COMUNICACIÓN	127
FIG. 4.16 INGRESO CÓDIGO Y ENVÍO	128
FIG. 4.17 RECEPCIÓN DE DATOS DESDE EL PIC	128

FIG. 4.18 MENÚ	129
FIG. 4.19 SELECCIÓN DE OPCIÓN.....	129
FIG. 4.20 VALIDACION O INICIO DE SESION	131
FIG. 4.21 VALIDACION CONEXIÓN CON BDD	131
FIG. 4.22 VALIDACION CONEXIÓN CON BDD	132
FIG. 4.23 VALIDACION CREACION DE USUARIO	132
FIG. 4.24 VALIDACION CREACION DE USUARIO	132
FIG. 4.25 VALIDACION CREACION DE USUARIO	133
FIG. 4.26 VALIDACION CREACION DE PERSONAL	133
FIG. 4.27 VALIDACION DE PERMISOS.....	133
FIG. 4.28 REPORTE DE DATOS PERSONAL	134
FIG. 4.29 REPORTE DE HORARIOS PERSONAL.....	134
FIG. 4.30 REPORTE DE REGISTROS PERSONAL	134

INDICE DE TABLAS

TABLA N°1.1 COMPONENTES DE UN MICROCONTROLADOR	21
TABLA N°1.2 RECURSOS DE LOS MICROCONTROLADORES.....	23
TABLA N°1.3 PARIDAD PAR.....	36
TABLA N°1.4 PARIDAD IMPAR.	37
TABLA N°1.5 TABLA DE LOS PINES Y SEÑAL.....	41
TABLA N° 1.6 PINES USB	43
TABLA N°3.1 SECUENCIA DE COMANDOS	87
TABLA N° 3.2 DESCRIPCIÓN DE CLASES	93
TABLA N° 3.3 ACTORES Y ACTIVIDADES.....	96
TABLA N° 3.4 CU1 INICIA SESIÓN.....	102
TABLA N° 3.5 CU2 GESTIÓN DE USUARIOS.....	104
TABLA N° 3.6 CU3 PERFIL DE USUARIOS.....	104
TABLA N° 3.7 CU4 GESTIONA HORARIOS.....	105
TABLA N° 3.8 CU5 GESTIONA DEPARTAMENTOS	106
TABLA N° 3.9 CU6 GESTIONA PERSONAL	107
TABLA N° 3.10 CU7 GESTIONA SISTEMA	108
TABLA N° 3.11 CU8 GESTIONA FALTAS Y PERMISOS	109
TABLA N° 3.12 CU9 GESTIONA VACACIONES, FERIADOS.....	109
TABLA N° 3.13 CU10 GESTIONA REPORTES.....	110
TABLA N° 3.14 CU14 REGISTRO DE HORAS DE ENTRADA Y SALIDA.	110
TABLA N° 3.15 TABLA DEPARTAMENTO	112

TABLA N° 3.16 TABLA EMPLEADO	113
TABLA N° 3.17 TABLA EMPRESA	114
TABLA N° 3.18 TABLA FERIADO	114
TABLA N° 3.19 TABLA PERFIL.....	115
TABLA N° 3.20 TABLA PERFIL HORARIO	115
TABLA N° 3.21 TABLA REGISTRO.....	116
TABLA N° 3.22 TABLA USUARIO	117
TABLA N° 3.23 CONEXIÓN.....	120
TABLA N° 4.1 COSTO – BENEFICIO SIN SISTEMA.....	137

INDICE DE CUADROS

CUADRO N° 1 FAMILIA DE MICROCONTROLADORES	24
CUADRO N° 2. NIVELES VOLTAJE PARA CADA ESTÁNDAR.	30

CAPITULO 1 MARCO TEORICO

1.1 LOS MICROCONTROLADORES

1.1.1 INTRODUCCIÓN

Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa y vida en general. Se pueden encontrar controlando el funcionamiento de ratones y teclados de los computadores, en teléfonos, en hornos microondas, televisores, en cualquier artefacto electrónico que podamos imaginar. Muchos aseguran que esto acaba de comenzar y el siglo XXI será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que se fabrican.

1.1.2 DEFINICIONES

1.1.2.1 Controlador

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los efectores que intentan llevar el valor de la temperatura dentro del rango estipulado.

1.1.2.2 Microcontrolador μC

En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador. Se dice entonces que **un microcontrolador es** un circuito integrado o chip programable que incluye en su interior las tres unidades funcionales de una computadora: unidad central de procesamiento, memoria y unidades de E/S (entrada/salida).

1.1.3 DIFERENCIA ENTRE MICROPROCESADOR Y MICROCONTROLADOR.

El microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (UCP), también llamada procesador de un computador.

Los pines de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, para permitir conectarle con la Memoria y los Módulos de E/S y configurar un computador implementado por varios circuitos integrados. Se dice que un microprocesador es un sistema abierto porque su configuración es variable de acuerdo con la aplicación a la que se destine, como se muestra la Figura 1.1

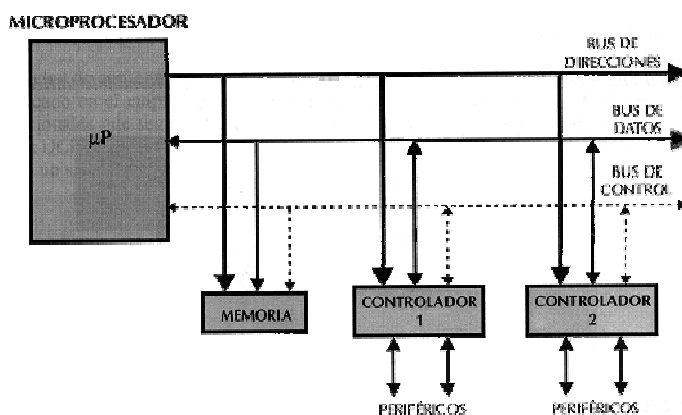


Fig 1.1. Estructura de un sistema abierto basado en un microprocesador. La disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación.
Fuente: www.monografias.com, Introducción a los microcontroladores

Si sólo se tuviera un modelo de microcontrolador, éste debería tener muy potenciados todos sus recursos para adaptarse a las exigencias de las diferentes aplicaciones. Esta potenciación supondría en muchos casos un despilfarro. En la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es posible seleccionar la capacidad de las memorias, el número de líneas de E/S, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar. La representación se observa en la Figura 1.2

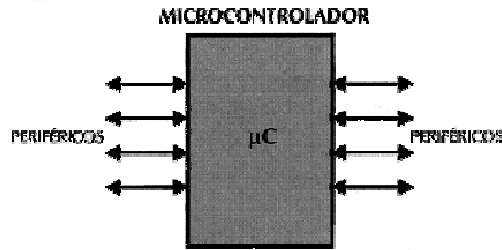


Fig 1.2. El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos.

Fuente: www.monografias.com, Introducción a los microcontroladores

1.1.4 ARQUITECTURA

La arquitectura de un Microcontrolador está basada en la arquitectura de una computadora. Hay dos arquitecturas conocidas: la clásica de Von Neumann, y la arquitectura Harvard.

1.1.4.1 Arquitectura Von Neumann

Dispone de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control) como se muestra en la Figura 1.3



Fig 1.3 Arquitectura Von Neumann

Fuente: www.monografias.com, Introducción a los microcontroladores

1.1.4.2 Arquitectura Harvard

Dispone de dos memorias independientes, una que contiene sólo instrucciones, y otra que contiene sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias. Esta arquitectura se observa en la Figura 1.4.

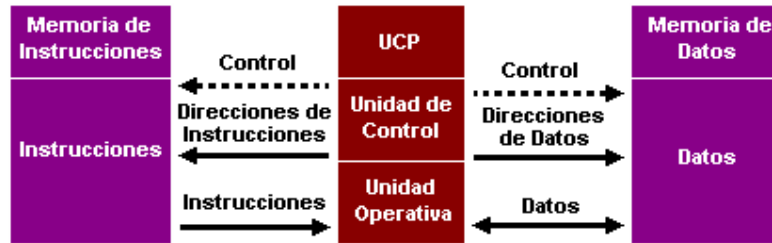


Fig. 1.4 Arquitectura Harvard

Fuente: www.monografias.com, Introducción a los microcontroladores

1.1.5 RECURSOS COMUNES A TODOS LOS MICROCONTROLADORES.

Al estar todos los microcontroladores integrados en un chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales Procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos. Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.

A continuación en Tabla N° 1.1, se detalla en forma resumida los componentes que normalmente dispone un microcontrolador.

COMPONENTE	DEFINICIÓN - UTILIDAD
Procesador UCP	Determina su característica a nivel de hardware como software.
Registros	Son un espacio de memoria muy reducido pero necesario para cualquier microprocesador.
Unidad de control	Aquí es donde se realizan las sumas, restas, y operaciones lógicas típicas del álgebra de Boole.
Buses	Son el medio de comunicación que utilizan los diferentes componentes del procesador para intercambiar información entre sí.
Unidad	Define las operaciones básicas que puede realizar el

Aritmético Lógica	procesador, que conjugadas y organizadas forman lo que conocemos como software
Conjunto de Instrucciones	Se Dispone de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución.
Cisc	El repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo.
Risc	En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y generalmente, se ejecutan en un ciclo.
Sisc	Microcontroladores destinados a aplicaciones muy concretas, o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista.
Memoria	En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.
RAM	En estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa.
ROM	Es una memoria no volátil de sólo de lectura cuyo contenido se graba durante la fabricación del chip.
OTP	Memoria no volátil de sólo lectura "programable una sola

	vez" por el usuario, es recomendable cuando es muy corto el ciclo de diseño del producto, o en prototipos y series muy pequeñas.
EPROM	Memoria que puede borrarse y grabarse muchas veces, los datos se borran sometiéndola a rayos ultra violeta.
EEPROM	Memoria de sólo lectura, programables y borrables eléctricamente, el número de veces que puede grabarse y borrarse una memoria EEPROM es finito.
FLASH	Memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña.
Puertas de Entrada y Salida	Son patitas que posee la cápsula que contiene un microcontrolador, su utilidad es soportar las líneas de E/S que comunican al computador interno con los periféricos exteriores.
Reloj principal	Los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema

Tabla N° 1.1 Componentes de un microcontrolador.

1.1.6 RECURSOS ESPECÍFICOS

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software.

En la Tabla N°1.2, se describe los principales recursos específicos que incorporan los microcontroladores:

RECURSOS	DEFINICION - FINALIDAD
Temporizadores o "Timers".	Se emplean para controlar periodos de tiempo y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).
Perro guardián o "Watchdog".	El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.
Protección ante fallo de alimentación o "Brownout".	Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo ("brownout").
Estado de reposo o de bajo consumo.	Los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos.
Conversor A/D (CAD)	Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patitas del circuito integrado.
Conversor D/A (CDA)	Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica, que saca al exterior por una de las patitas de la cápsula.
Comparador analógico.	Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por uno de los pines de la cápsula.
Modulador de	Son circuitos que proporcionan en su salida

anchura de impulsos o PWM.	impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.
Puertos de E/S digitales	Todos los microcontroladores destinan algunas de sus patitas a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertos.
Puertas de comunicación.	Le da al microcontrolador la posibilidad de comunicarse con otros dispositivos externos.
UART	Es el adaptador de comunicación serie asíncrona.
USART	E el adaptador de comunicación serie síncrona y asíncrona.
Puerta paralela esclava	Permite conectarse con los buses de otros microprocesadores.
USB	Bus serie para los PC.
Bus I2C	Es una interfaz serie de dos hilos desarrollado por Philips.
CAN	Permite la adaptación con redes de conexionado multiplexado.

Tabla N°1.2 Recursos de los microcontroladores

1.1.7 FAMILIAS DE MICROCONTROLADORES

Existen varias empresas dedicadas a la construcción y venta de microcontroladores cada empresa produce una variedad de modelos, cada una con diferentes características lo que permite a los usuarios escoger el microcontrolador que mejor se adapte a los requerimientos planteados.

En el cuadro N°1, se presenta un detalle básico de las Familias de microcontroladores y que cada empresa fabrica.

Empresa	8 bits	12 bits	14 bits	16 bits	32 bits	64 bits	Observaciones
Atmel	AVR (mega y tiny), 89Sxxxx familia similar 8051				SAM7 (ARM7TDMI), SAM3 (ARM Cortex-M3), SAM9 (ARM926)		
Freescale (antes Motorola)	68HC05, 68HC08, 68HC11, HCS08	x	x	68HC12, 68HCS12, 68HCSX12, 68HC16	683xx, PowerPC Architecture, ColdFire	x	x
Holtek	HT8						
Intel	MCS-48 (familia 8048) MCS51 (familia 8051) 8xC251	x	x	MCS96, MXS296	x	x	x
National Semiconductor	COP8	x	x	x	x	x	x
Microchip	Familia 10f2xx Familia 12Cxx Familia 12Fxx, 16Cxx y 16Fxx 18Cxx y 18Fxx			PIC24F, PIC24H y dsPIC30FXX, dsPIC33F con motor dsp integrado	PIC32	x	x
NXP (antes Philips)	80C51			XA	Cortex-M3, Cortex-M0, ARM7, ARM9		
Parallax							
Renesas (antes Hitachi, Mitsubishi y NEC)	78K, H8			H8S, 78K0R, R8C, R32C/M32C/M16C	RX, V850, SuperH, SH-Mobile, H8SX	x	x
STMicroelectronics	ST 62, ST 7						
Texas Instruments	TMS370, MSP430				C2000, Cortex-M3 (ARM), TMS570 (ARM)		
Zilog	Z8, Z86E02						
Silabs	C8051						

Cuadro N° 1 Familia de Microcontroladores

1.1.8 APLICACIONES DE LOS MICROCONTROLADORES.

Los microcontroladores están siendo empleados en multitud de sistemas presentes como juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de arranque del coche, etc. Y otras aplicaciones con las que seguramente no estamos familiarizados como instrumentación electrónica, control de sistemas en una nave espacial, etc. Una aplicación típica podría emplear varios microcontroladores para controlar pequeñas partes del sistema.

1.1.9 EL MERCADO DE LOS MICROCONTROLADORES.

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 ó 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer. La razón de esta tendencia es que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace absurdo emplear micros más potentes y consecuentemente más caros. Uno de los sectores que más tira del mercado del microcontrolador es el mercado automovilístico..

También los modernos microcontroladores de 32 bits van afianzando sus posiciones en el mercado, siendo las áreas de más interés el procesamiento de imágenes, las comunicaciones, las aplicaciones militares, los procesos industriales y el control de los dispositivos de almacenamiento masivo de datos.

1.1.10 SELECCIÓN DEL MICROCONTROLADOR

A la hora de escoger el microcontrolador a emplear en un diseño concreto hay que tener en cuenta multitud de factores, como la documentación y herramientas de desarrollo disponibles y su precio, la cantidad de fabricantes que lo producen y por supuesto las características del microcontrolador (tipo de memoria de programa, número de temporizadores, interrupciones, etc.)

En cuanto a la aplicación es necesario analizar los siguientes requisitos.

- **Procesamiento de datos:** puede ser necesario que el microcontrolador realice cálculos críticos en un tiempo limitado, si no es suficiente con un microcontrolador de 8 bits, puede ser necesario acudir a microcontroladores de 16 ó 32 bits.
- **Entrada Salida:** para determinar las necesidades de Entrada/Salida del sistema es conveniente dibujar un diagrama de bloques del mismo, de tal forma que sea sencillo identificar la cantidad y tipo de señales a controlar.
- **Consumo:** algunos productos que incorporan microcontroladores están alimentados con baterías y su funcionamiento puede ser tan vital como activar una alarma antirrobo. Lo más conveniente en un caso como éste puede ser que el microcontrolador esté en estado de bajo consumo pero que despierte ante la activación de una señal (una interrupción) y ejecute el programa adecuado para procesarla.
- **Memoria:** para detectar las necesidades de memoria de nuestra aplicación debemos separarla en memoria volátil (RAM), memoria no volátil (ROM, EPROM, etc.) y memoria no volátil modificable (EEPROM). Este último tipo de memoria puede ser útil para incluir información específica de la aplicación como un número de serie o parámetros de calibración.

El tipo de memoria a emplear vendrá determinado por el volumen de ventas previsto del producto: de menor a mayor volumen será conveniente emplear EPROM, OTP y ROM. En cuanto a la cantidad de memoria necesaria puede ser imprescindible realizar una versión preliminar, aunque sea en pseudo-código, de la aplicación y a partir de ella hacer una estimación de cuánta memoria volátil y no volátil es necesaria y si es conveniente disponer de memoria no volátil modificable.

- **Ancho de palabra:** hace referencia a una cadena finita de bits que son manejados como un conjunto por el microcontrolador, el tamaño o longitud de una

palabra hace referencia al número de bits contenidos en ella , para el diseño se debe seleccionar el microcontrolador de menor ancho de palabra que satisfaga los requerimientos de la aplicación. Usar un microcontrolador de 4 bits supondrá una reducción en los costes importante, mientras que uno de 8 bits puede ser el más adecuado si el ancho de los datos es de un byte. Los microcontroladores de 16 y 32 bits, debido a su elevado coste, deben reservarse para aplicaciones que requieran sus altas prestaciones (Entrada/Salida potente o espacio de direccionamiento muy elevado).

- **Diseño de la placa:** la selección de un microcontrolador concreto condicionará el diseño de la placa de circuitos. Debe tenerse en cuenta que quizá usar un microcontrolador barato encarezca el resto de componentes del diseño.

1.1.11 MICROCONTROLADORES MÁS POPULARES

8048 (Intel). Es el padre de los microcontroladores actuales, el primero de todos. Su precio, disponibilidad y herramientas de desarrollo hacen que todavía sea muy popular.

8051 (Intel y otros). Es sin duda el microcontrolador más popular. Fácil de programar, pero potente. Está bien documentado y posee cientos de variantes e incontables herramientas de desarrollo.

80186, 80188 y 80386 EX (Intel). Versiones en microcontrolador de los populares microprocesadores 8086 y 8088. Su principal ventaja es que permiten aprovechar las herramientas de desarrollo para PC.

68HC11 (Motorola y Toshiba). Es un microcontrolador de 8 bits potente y popular con gran cantidad de variantes.

683xx (Motorola). Surgido a partir de la popular familia 68k, a la que se incorporan algunos periféricos. Son microcontroladores de altísimas prestaciones.

PIC (MicroChip). Familia de microcontroladores que gana popularidad día a día. Fueron los primeros microcontroladores RISC.

AVRs(ATMEL).- es un microcontrolador desarrollado por la compañía ATMEL en 1996. Fue una de las primeras familias usadas en memorias flash para almacenamiento de información.

1.2 COMUNICACIÓN SERIAL

1.2.1 INTRODUCCIÓN

Cuando iniciamos una conversación, en primer lugar se llama la atención y luego se transmite el mensaje, una palabra cada vez. Cuando se termina de transmitir, se realiza una pausa para indicar que se ha concluido. Lo mismo se cumple con la lectura o la escritura, se comienza una oración con la letra mayúscula, y se lee o escribe una palabra cada vez, con intervalos de cierto período. Estas formas de comunicación humanas son serie.

1.2.2 CONCEPTO¹

La comunicación serial es el proceso de envío de datos de un bit por vez, secuencialmente, sobre un canal de comunicación o un bus de computadora. Contrasta con la comunicación paralela, donde todos los bits de cada símbolo (la más pequeña unidad de datos transmitida por vez) son enviados juntos.

1.2.3 HISTORIA

La historia de la comunicación serial se remonta a 1810 cuando el alemán Von Soemmerring realizó un experimento, utilizó 26 cables uno por cada letra del alfabeto y que estaban pegados al fondo de un acuario y al energizar estos cables se producían burbujas en el agua, de esta forma se podían enviar mensajes codificados por medio de burbujas. Los militares vieron este ingenio y se comenzó la carrera para el desarrollo de los sistemas de comunicación.

El señor Morse inventó un dispositivo que podía enviar sonidos cortos y largos que representaba los caracteres. En el código Morse el sonido corto se llama Espacios y el sonido largo Marcas, por lo tanto este es un código binario por lo que utiliza solo 2 valores posibles ya sea Marca o Espacio. En términos de computadoras se diría que trabaja con números 0 y 1 dependiendo de cada bit.

¹ Fuente: http://es.wikipedia.org/wiki/Puerto_serie

Por lo tanto, se puede decir que al hablar de código Morse se habla de una comunicación serial debido a que los bits se reciben uno de tras de otro.

1.2.4 TRANSMISIÓN MODULADA EN AMPLITUD

Dentro de las múltiples posibilidades existentes nos centraremos en la comunicación serie a través de la interpretación de dos niveles lógicos de tensión o corriente denominado formato marca/espacio.

El nivel lógico "1" representa un estado de tensión o corriente denominado marca, el nivel lógico "0" representa un estado de tensión o corriente denominado espacio. En la Figura 1.5 se puede observar el formato marca/espacio

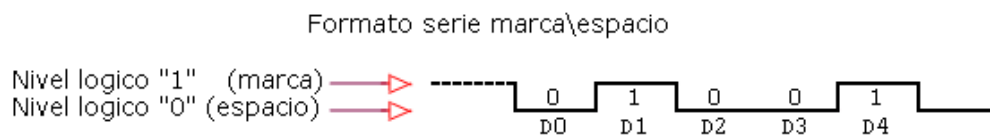


Fig. 1.5 Formato serie marcar/espacio

1.2.5 ESTÁNDARES CON FORMATO MARCA/ESPACIO

Existen varios estándares que usan el formato marca/espacio, los que interesan son: TTL, RS232

En el cuadro N°2 se detalla los niveles para cada estándar.

	Nivel Lógico "1" (Marca)	Nivel Lógico "0" (Espacio)
TTL	5V	0
RS 232C	-3V a -15V	+3V a +15V

Cuadro N° 2. Niveles voltaje para cada estándar.

1.2.6 Consideraciones en la comunicación serie

Cuando se transmite información a través de una línea serie es necesario utilizar un sistema de codificación que permita resolver los siguientes problemas:

Sincronización de bits: El receptor necesita saber donde comienza y donde termina cada bit en la señal recibida para efectuar el muestreo de la misma en el centro del intervalo de cada símbolo (bit para señales binarias).

Sincronización del carácter: La información serie se transmite por definición bit a bit, pero la misma tiene sentido en palabras o bytes.

Sincronización del mensaje: Es necesario conocer el inicio y fin de una cadena de caracteres por parte del receptor para, por ejemplo, detectar algún error en la comunicación de un mensaje.

1.2.7 VELOCIDAD DE TRANSMISIÓN

La velocidad de transmisión de datos es expresada en bits por segundo o baudios. El baudio es un concepto más general que bit por segundo. El primero queda definido como el número de estados de la señal por segundo, si sólo existe dos estados (que pueden ser representados por un bit, que identifica dos unidades de información) entonces baudio es equivalente a bit por segundo. Baudio y bit por segundo se diferencian cuando es necesario más de un bit para representar más de dos estados de la señal.

1.2.8 LA BASE DE RELOJ

Cuando se establece la comunicación es necesario implementar una base de tiempo que controle la velocidad. En un microcontrolador, se utilizaría la base de tiempos del reloj del sistema, si bien, en términos genéricos se utilizaría uno de los siguientes métodos:

- Mediante la división de la base de reloj del sistema. por ejemplo mediante un contador temporizador programable.
- A través de un oscilador TTL. Para cambiar frecuencia hay que cambiar el cristal.

- Generador de razón de baudios. Existen diferentes dispositivos especializados que generan diferentes frecuencias de reloj.

1.2.9 TIPOS DE COMUNICACIÓN

- Simplex
- Semi duplex (Half duplex)
- Totalmente duplex (Full duplex)

Simplex: En ella la comunicación serie usa una dirección y una línea de comunicación. Siempre existirá un transmisor y un receptor, no ambos.

Semi duplex: La comunicación serie se establece a través de una sola línea, pero en ambos sentidos. En un momento el transmisor enviará información y en otro recibirá, por lo que no se puede transferir información en ambos sentidos de forma simultánea.

Full duplex: Se utilizan dos líneas (una transmisora y otra receptora) y se transfiere información en ambos sentidos. La ventaja de este método es que se puede transmitir y recibir información de manera simultánea.

1.2.10 MODOS DE TRANSMISIÓN

Existen dos modos básicos para realizar la transmisión de datos y son:

- Modo asíncrono.
- Modo síncrono.

Las transmisiones asíncronas son aquellas en que los bits que constituyen el código de un carácter se emiten con la ayuda de impulsos suplementarios que permiten mantener en sincronismo los dos extremos.

En las transmisiones síncronas los caracteres se transmiten consecutivamente, no existiendo ni bit de inicio ni bit de parada entre los caracteres, estando dividida

la corriente de caracteres en bloques, enviándose una secuencia de sincronización al inicio de cada bloque.

1.2.10.1 La transmisión asíncrona

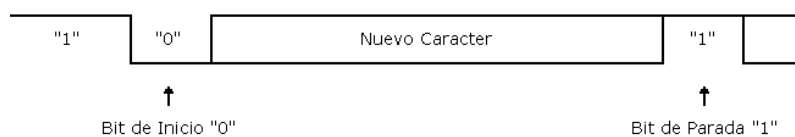
Cuando se opera en modo asíncrono no existe una línea de reloj común que establezca la duración de un bit y el carácter puede ser enviado en cualquier momento. Esto conlleva que cada dispositivo tiene su propio reloj y que previamente se ha acordado que ambos dispositivos transmitirán datos a la misma velocidad.

No obstante, en un sistema digital, un reloj es normalmente utilizado para sincronizar la transferencia de datos entre las diferentes partes del sistema. El reloj definirá el inicio y fin de cada unidad de información así como la velocidad de transmisión. Si no existe reloj común, algún modo debe ser utilizado para sincronizar el mensaje.

1.2.10.1.1 Bit de inicio y bit de parada

En la transmisión asíncrona un carácter a transmitir es encuadrado con un indicador de inicio y fin de carácter, de la misma forma que se separa una palabra con una letra mayúscula y un espacio en una oración. La forma estándar de encuadrar un carácter es a través de un bit de inicio y un bit de parada.

Durante el intervalo de tiempo en que no son transferidos caracteres, el canal debe poseer un "1" lógico. Al bit de parada se le asigna también un "1". Al bit de inicio del carácter a transmitir se le asigna un "0". Por todo lo anterior, un cambio de nivel de "1" a "0" lógico le indicará al receptor que un nuevo carácter será transmitido. En la figura 1.6, se observa el formato de transmisión asíncrona.



Formato de trasmision asincronica.

Fig. 1.6 Bit de inicio y parada en transmisión asíncrona.

1.2.10.1.2 Reglas de transmisión asíncrona

La transmisión asíncrona es la definida por la norma RS232, en la que se profundizará más adelante y que se basa en las siguientes reglas:

- Cuando no se envían datos por la línea, ésta se mantiene en estado alto (1).
- Cuando se desea transmitir un carácter, se envía primero un bit de inicio que pone la línea a estado bajo (0) durante el tiempo de un bit.
- Durante la transmisión, si la línea está a nivel bajo, se envía un 0 y si está a nivel alto se envía un 1.
- A continuación, se envían todos los bits del mensaje a transmitir con los intervalos que marca el reloj de transmisión. Por convenio se transmiten entre 5 y 8 bits.
- Se envía primero el bit menos significativo, siendo el más significativo el último en enviarse.
- A continuación del último bit del mensaje se envía el bit (o los bits) del final que hace que la línea se ponga a 1 por lo menos durante el tiempo mínimo de un bit. Estos bits pueden ser un bit de paridad para detectar errores y el bit o bits de stop, que indican el fin de la transmisión de un carácter.

Los datos codificados por esta regla, pueden ser recibidos siguiendo los pasos siguientes:

- Esperar la transición 1 a 0 en la señal recibida.
- Activar el reloj con una frecuencia igual a la del transmisor.
- Muestrear la señal recibida al ritmo de ese reloj para formar el mensaje.
- Leer un bit más de la línea y comprobar si es 1 para confirmar que no ha habido error en la sincronización.

1.2.10.2 La transmisión síncrona

Es un método más eficiente de comunicación en cuanto a velocidad de transmisión. Ello viene dado porque no existe ningún tipo de información adicional

entre los caracteres a ser transmitidos. En la figura N° 1.7, se observa el formato de envío de datos.

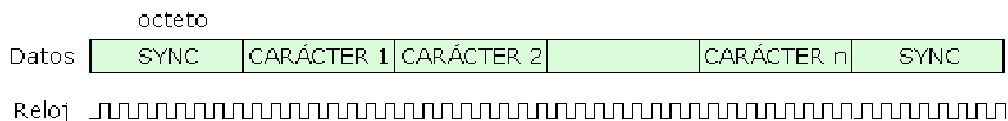


Fig 1.7 Transmisión síncronica

Cuando se transmite de manera síncrona lo primero que se envía es un octeto de sincronismo ("sync"). El octeto de sincronismo realiza la misma función que el bit de inicio en la transmisión asíncrona, indicando al receptor que va a ser enviado un mensaje. Este carácter, además, utiliza la señal local de reloj para determinar cuándo y con qué frecuencia será muestreada la señal, es decir, permite sincronizar los relojes de los dispositivos transmisor y receptor. La mayoría de los dispositivos de comunicación llevan a cabo una resincronización contra posibles desviaciones del reloj, cada uno o dos segundos, insertando para ello caracteres del tipo "sync" periódicamente dentro del mensaje.

Cuando se transmite de forma síncrona, es necesario mantener el sincronismo entre el transmisor y el receptor cuando no se envían caracteres, para ello son insertados caracteres de sincronismo de manera automática por el dispositivo que realiza la comunicación. En la figura 1.8, se observa el formato por inserción automática de caracteres.

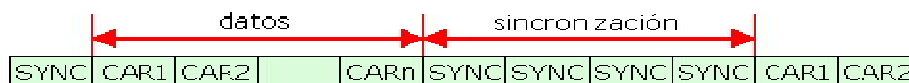


Fig. 1.8 Inserción automática de caracteres de sincronismo.

El receptor/transmisor síncrono debe indicar además cuándo el sincronismo ha sido logrado por parte del receptor.

1.2.10.3 Detectar errores en la comunicación

Cuando se escriben o se envían datos, pueden producirse errores, entre otras cosas, por ruidos inducidos en las líneas de transmisión de datos. Es por tanto necesario comprobar la integridad de los datos transmitidos mediante algún método que permita determinar si se ha producido un error.

En un caso típico, si al transmitirse un mensaje se determina que se ha producido un error, el receptor solicita de nuevo el mensaje al emisor.

Se pueden detectar errores de acuerdo a la forma de transmisión:

Transmisión asíncrona:

- Sobre escritura.
- Error de encuadre (framing).
- Transmisión síncrona:
- Paridad.
- Sobre escritura.

1.2.10.4 Paridad par

El bit de paridad será cero, cuando el número de bit "unos" que contienen los datos a transmitir sea un número par, y el bit de paridad será uno cuando los datos que se mandan contienen un número impar de unos, tal y como se muestra en la tabla N° 1.3.

Dato	Paridad
0000 0001	1
0101 0001	1
0101 0101	0
0000 0000	0

Tabla N°1.3. Paridad par.

La suma de los bits que son unos, contando datos y bit de paridad dará siempre como resultado un número par de unos.

1.2.10.5 Paridad impar

Dato	Paridad
0000 0001	0
0101 0001	0
0101 0101	1
0000 0000	1

Tabla N°1.4. Paridad Impar.

En el sistema de paridad impar, el número de unos (datos + paridad) siempre debe ser impar, como muestra la tabla N° 1.4.

1.2.10.6 Método checksum

El método checksum puede ser utilizado tanto en la transmisión síncrona como en la asíncrona. Se basa en la transmisión, al final del mensaje, de un byte (o bytes) cuyo valor sea el complemento a dos de la suma de todos los caracteres que han sido transmitidos en el mensaje. El receptor implementará una rutina que suma todos los bytes de datos recibidos y al resultado se le sumará el último byte (que posee la información en complemento a dos de la suma de los caracteres transmitidos) y si la recepción del mensaje ha sido correcta, el resultado debe ser cero.

Ejemplo:

Indicar el último carácter a transmitir cuando se implementa el método de checksum. Los datos a transmitir serán 40H, 35H y 0EH.

Se realiza la suma:

$$40H + 35H + 0EH = 83H$$

Ahora se determina el complemento a dos del resultado:

$$83H = 1000\ 0011$$

$$0111\ 1100$$

$$+ \quad 1$$

$$0111\ 1101 = 7DH$$

El último carácter a transmitir será 7DH.

1.2.11 ALGUNOS EJEMPLOS DE TECNOLOGÍA DE COMUNICACIÓN

SERIAL²

- El código morse en telegrafía
- RS-232 (de baja velocidad, implementado en puertos seriales)
- RS-423
- RS485
- USB (de velocidad media para conectar periféricos a la computadora)
- FireWire
- Ethernet
- Canal de Fibra (de alta velocidad para comunicar computadoras a dispositivos de almacenamiento masivo)
- Serial Attached SCSI
- Serial ATA
- PCI Express

² Fuente: <http://www.alegsa.com.ar/Dic/comunicacion%20serial.php>

1.2.12 COMUNICACIÓN CON EL MAX232 - CONVERSOR TTL-RS232

El **MAX232** es un circuito integrado que convierte los niveles de las líneas de un puerto serie RS232 a niveles TTL y viceversa³. Lo interesante es que sólo necesita una alimentación de 5V, ya que genera internamente algunas tensiones que son necesarias para el estándar RS232. Otros integrados que manejan las líneas RS232 requieren dos voltajes, +12V y -12V.

En la figura N° 1.9, se muestra representado el encapsulado MAX232 y el circuito interno.

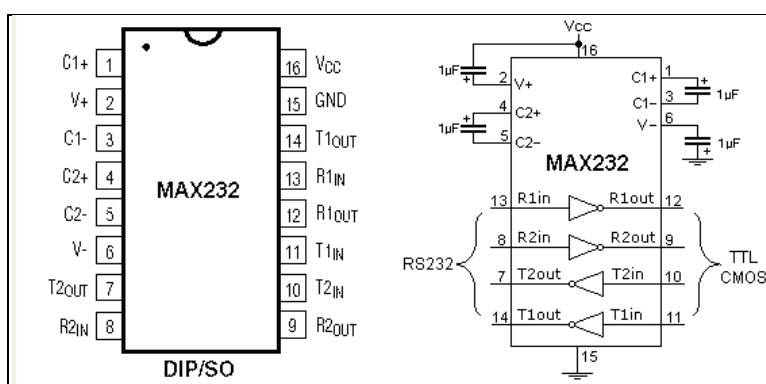


Fig. 1.9 MAX 232

Fuente: <http://perso.wanadoo.es/pictob/comserie.htm>

El MAX232 soluciona la conexión necesaria para lograr comunicación entre el puerto serie de una PC y cualquier otro circuito con funcionamiento en base a señales de nivel TTL/CMOS como son los microcontroladores.

El circuito integrado posee dos convertidores de nivel TTL a RS232 y otros dos que, a la inversa, convierten de RS232 a TTL.

Estos convertidores son suficientes para manejar las cuatro señales más utilizadas del puerto serie del PC, que son TX, RX, RTS y CTS.

TX es la señal de transmisión de datos, RX es la de recepción, y RTS y CTS se utilizan para establecer el protocolo para el envío y recepción de los datos.

³ **MAX232** es un circuito integrado que convierte los niveles de las líneas de un puerto serie RS232 a niveles TTL y viceversa

Fuente: <http://perso.wanadoo.es/pictob/comserie.htm>

1.2.12.1 Conexión MAX 232 PIC y PC

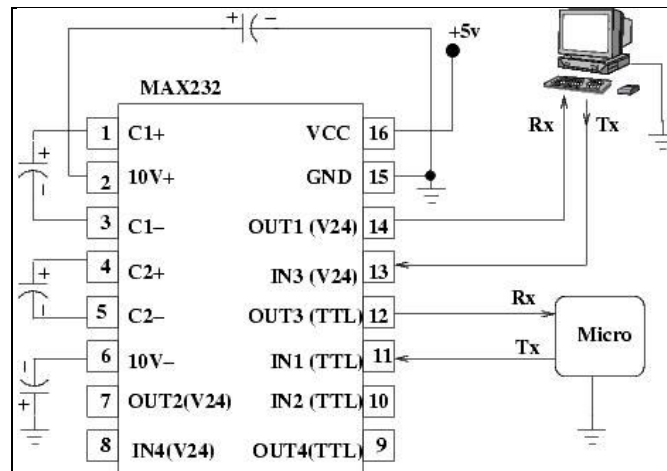


Fig. 1.10 Conexión MAX 232.

Fuente: <http://www.learobotics.com/proyectos/cuadernos/ct1/ct1.html>

El integrado MAX 232 permite al PIC transmitir a distancias hasta 15 m y con velocidades estándar de 9600 bps.

El puerto usado por una PC para la transmisión serial es llamado puerto COM o rs232 y tiene un conector DB9 macho por el cual el PC puede recibir las señales seriales de cualquier dispositivo. En la figura N° 1.10 se muestra la conexión de los pines del MAX232 con el PIC y con la PC.

1.2.12.2 El conector DB9 del PC

En los PCs hay **conectores DB9 macho**, de 9 pines, por el que se conectan los dispositivos al puerto serie. Los conectores hembra que se enchufan tienen una colocación de pines diferente, de manera que se conectan el pin 1 del macho con el pin 1 del hembra, el pin2 del macho con el pin 2 de la hembra, etc. En la figura N° 1.11, se muestra los conectores DB9 hembra y macho.

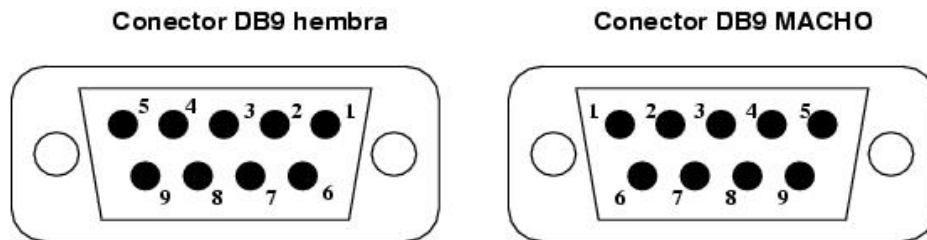


Fig 1.11 Conector DB9 hembra y macho.

Fuente: <http://www.learobotics.com/proyectos/cuadernos/ct1/ct1.html>

La información asociada a cada uno de los pines se describe en la tabla N° 5:

Número de pin	Señal
1	DCD (Data Carrier Detect)
2	RX
3	TX
4	DTR (Data Terminal Ready)
5	GND
6	DSR (Data Sheet Ready)
7	RTS (Request To Send)
8	CTS (Clear To Send)
9	RI (Ring Indicator)

Tabla N°1.5 Tabla de los pines y señal

Fuente: <http://www.learobotics.com/proyectos/cuadernos/ct1/ct1.html>

1.2.13 COMUNICACIÓN SERIAL USB

A continuación una breve descripción de la comunicación serial por USB, pero debido a que es un estándar que no cumple con los requerimientos del cliente en cuanto a la distancia que se puede transmitir, no se profundizará en su estudio.

1.2.13.1 Introducción

Universal Serial Bus es un bus estándar para la comunicación serial. Originalmente concebido como un reemplazo para conexiones en serie y paralelos de un PC para conectar los diferentes periféricos, y se ha convertido en el estándar de interfaz entre un ordenador y los periféricos.

El USB puede conectar varios tipos de dispositivos como pueden ser: mouse, teclados, escáneres, cámaras digitales, teléfonos móviles, reproductores multimedia, impresoras, discos duros externos entre otros ejemplos, tarjetas de sonido, sistemas de adquisición de datos y componentes de red. Para dispositivos multimedia como escáneres y cámaras digitales, el USB se ha convertido en el método estándar de conexión. Para impresoras, el USB ha crecido tanto en popularidad que ha desplazado a un segundo plano a los puertos paralelos porque el USB hace mucho más sencillo el poder agregar más de una impresora.

1.2.13.2 Velocidad de transmisión⁴

Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

- Baja velocidad (1.0): Tasa de transferencia de hasta 1,5 Mbps (192 KB/s). Utilizado en su mayor parte por dispositivos de interfaz humana (Human interface device, en inglés) como los teclados, los ratones (mouse), las cámaras web, etc.

- Velocidad completa (1.1): Tasa de transferencia de hasta 12 Mbps (1,5 MB/s) según este estándar, pero se dice en fuentes independientes que habría que realizar nuevamente las mediciones. Ésta fue la más rápida antes de la especificación USB 2.0, y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad. Estos dispositivos dividen el ancho de banda de la conexión USB entre ellos, basados en un algoritmo de impedancias LIFO.

- Alta velocidad (2.0): Tasa de transferencia de hasta 480 Mbps (60 MB/s) pero por lo general de hasta 125Mbps (16MB/s). Está presente casi en el 99% de los PC actuales. El cable USB 2.0 dispone de cuatro líneas, un par para datos, una de corriente y una de toma de tierra.

- Super alta velocidad (3.0): Tiene una tasa de transferencia de hasta 4.8 Gbps (600 MB/s). La velocidad del bus es diez veces más rápida que la del USB 2.0, debido a que han incluido 5 conectores extra, desechando el conector de fibra

⁴ Fuente : Wikipedia, Estandar USB

óptica propuesto inicialmente, y será compatible con los estándares anteriores. usa un cable de 9 hilos. En Octubre de 2009 la compañía taiwanesa ASUS lanzó la primera placa base que incluía puertos USB3, tras ella muchas otras le han seguido y se espera que en 2012 ya sea el estándar de facto.

1.2.13.3 Conectores

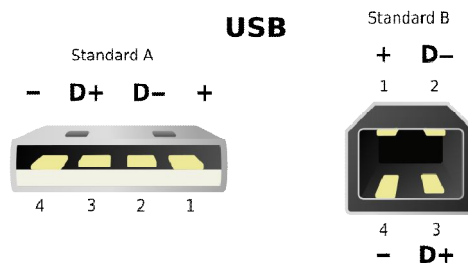


Fig.1.12 .Conectores USB

Pin	Nombre	Color del cable	Descripción
1	VCC	Rojo	+5v
2	D-	Blanco	Data -
3	D+	Verde	Data +
4	GND	Negro	Tierra

Tabla Nº 1.6 Pines USB

http://es.wikipedia.org/wiki/Universal_Serial_Bus

1.2.13.4 Características

- Distancia máxima de transmisión 5m.
- Conectable en caliente.
- Conexión de dispositivos externos.

Para la programación de PICs que permitan conexión USB, Microchip dispone de varias familias de microcontroladores así como también las librerías necesarias para la programación de las aplicaciones, todos estos recursos estan disponibles en la página web de Microchip.

1.3 BASES DE DATOS

1.3.1 DEFINICIÓN

Una base de datos es un “almacén” que permite guardar grandes cantidades de información de forma organizada para que luego se pueda encontrar y utilizar fácilmente.

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una **base de datos** se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Diferentes programas y diferentes usuarios deben utilizar estos datos. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red ya que se debe poder compartir esta información. De allí el término **base**. "Sistema de información" es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir datos que se han instalado.

En la Figura 1.12, se muestra una estructura básica de una BDD.

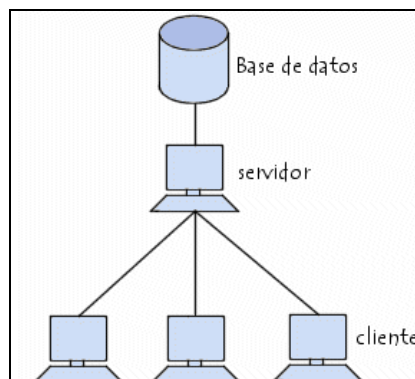


Fig 1.13 Base de datos Cliente – Servidor

Fuente: <http://es.kioskea.net/contents/bdd/bddintro.php3>

1.3.2 UTILIZACIÓN DE BDD

Una base de datos proporciona a los usuarios el acceso a datos, que pueden visualizar, ingresar o actualizar, en concordancia con los derechos de acceso que se les hayan otorgado. Se convierte más útil a medida que la cantidad de datos almacenados crece.

Una base de datos puede ser local es decir, que puede utilizarla sólo un usuario en un equipo, o puede ser distribuida, es decir que la información se almacena en equipos remotos y se puede acceder a ella a través de una red.

La principal ventaja de utilizar bases de datos es que múltiples usuarios pueden acceder a ellas al mismo tiempo.

1.3.3 ADMINISTRACIÓN DE BASES DE DATOS

La administración de bases de datos se realiza con un sistema llamado **DBMS** (Database Management System (Sistema de administración de bases de datos)).

Los DBMS son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

El DBMS es un conjunto de servicios (aplicaciones de software) para administrar bases de datos, que permite:

- un fácil acceso a los datos
- el acceso a la información por parte de múltiples usuarios
- la manipulación de los datos encontrados en la base de datos (insertar, eliminar, editar)

En la figura 1.13, se muestra la estructura de un DBMS

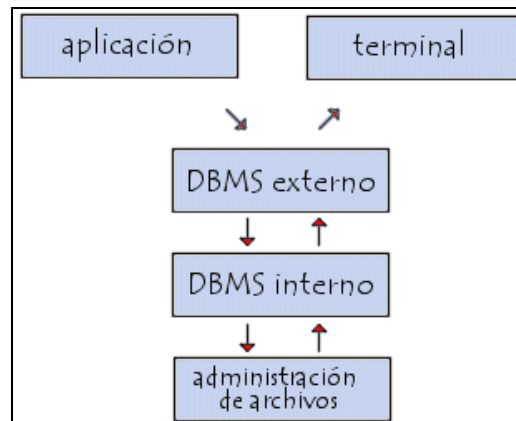


Fig 1.14 Administrador de Base de Datos.
Fuente: <http://es.kioskea.net/contents/bdd/bddintro.php3>

El DBMS puede dividirse en tres subsistemas:

- El sistema de administración de archivos:
para almacenar información en un medio físico, define cómo se organiza la información dentro de la base de datos.
- El DBMS interno:
para ubicar la información en orden, define cómo se almacenan los datos y los métodos de acceso.
- El DBMS externo:
representa la interfaz del usuario, define las vistas del usuario

1.3.4 CARACTERÍSTICAS DE UN DBMS

La arquitectura de tres niveles definida por el modelo ANSI/SPARC mantiene los datos y el procesamiento separados. Hablando en general, un DBMS debe tener las siguientes características:

- **Independencia física:** El nivel físico puede ser modificado independientemente del nivel conceptual. Esto significa que el usuario no puede ver todos los componentes de hardware de la base de datos, que es simplemente una estructura transparente para representar la información almacenada.

- **Independencia lógica:** El nivel conceptual debe modificarse sin alterar el nivel físico. En otras palabras, el administrador de la base de datos debe poder introducir mejoras sin afectar la experiencia de los usuarios.
- **Facilidad de uso:** Las personas que no estén familiarizadas con la base de datos deben poder describir su consulta sin hacer referencia a los componentes técnicos de la base de datos.
- **Acceso rápido:** El sistema debe poder responder a las consultas lo más rápido posible. Esto requiere algoritmos de búsqueda rápidos.
- **Administración centralizada:** El DBMS debe permitirle al administrador manipular los datos, agregar elementos y verificar su integridad de manera centralizada.
- **Redundancia controlada:** El DBMS debe poder evitar la redundancia de datos siempre que sea posible, tanto para minimizar los errores como para prevenir el desperdicio de memoria.
- **Verificación de integridad:** Los datos deben ser internamente coherentes y, cuando algunos elementos hacen referencia a otros, estos últimos deben estar presentes.
- **Uso compartido de datos:** El DBMS debe permitir que múltiples usuarios accedan simultáneamente a la base de datos.
- **Seguridad de los datos:** El DBMS debe poder administrar los derechos de acceso a los datos de cada usuario.

1.3.5 LOS DIFERENTES MODELOS DE BASE DE DATOS

Las bases de datos aparecieron a finales de la década de 1960, cuando surgió la necesidad de contar con un sistema de administración de información flexible.

Existen cinco modelos de DBMS, que se distinguen según cómo representan los datos almacenados:

- **El modelo jerárquico:** los datos se organizan jerárquicamente mediante un árbol invertido. Este modelo utiliza punteros para navegar por los datos almacenados. Fue el primer modelo DBMS, la estructura se representa en la Figura 1.14.

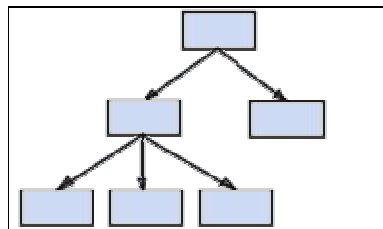


Fig. 1.15 Modelo BD Jerárquico.

Fuente: <http://es.kioskea.net/contents/bdd/bddintro.php3>

- **El modelo de red:** al igual que el modelo jerárquico, este modelo utiliza punteros hacia los datos almacenados. Sin embargo, no necesariamente utiliza una estructura de árbol invertido, como se muestra en la Figura 1.15.

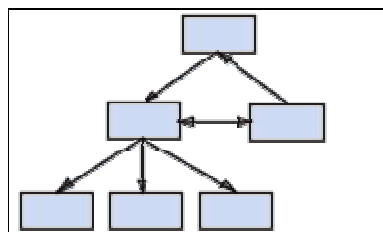


Fig. 1.16 Modelo BD de red.

Fuente: <http://es.kioskea.net/contents/bdd/bddintro.php3>

- **El modelo relacional** (RDBMS, *Relational database management system* [*Sistema de administración de bases de datos relacionales*]): los datos se almacenan en tablas de dos dimensiones (filas y columnas). Los datos se manipulan según la teoría relacional de matemáticas. Su representación se observa en la Figura 1.16

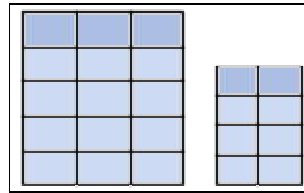


Fig 1.17 Modelo BD relacional.

Fuente: <http://es.kioskea.net/contents/bdd/bddintro.php3>

- **El modelo deductivo:** los datos se representan como una tabla, pero se manipulan mediante cálculos de predicados.
- **El modelo de orientación a objetos** (ODBMS, *object-oriented database management system* [*sistema de administración de bases de datos orientadas a objetos*]): los datos se almacenan como objetos, que son estructuras denominadas *clases* que muestran los datos que contienen. Los campos son instancias de estas clases. Su representación la se observa en la figura 1.17.

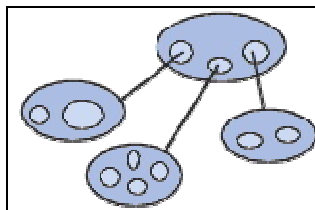


Fig 1.18 Modelo orientación a objetos.

Fuente: <http://es.kioskea.net/contents/bdd/bddintro.php3>

1.3.6 TIPOS DE BASE DE DATOS

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, o la utilidad de la misma:

1.3.6.1 Según la variabilidad de los datos almacenados

- **Bases de datos estáticas**

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

- **Bases de datos dinámicas**

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub.

1.3.6.2 Según el contenido

- **Bases de datos bibliográficas**

Solo contienen un subrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no, estaríamos en presencia de una base de datos a texto completo (o de fuentes primarias —ver más abajo). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

- **Bases de datos de texto completo**

Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

- **Directorios**

Un ejemplo son las guías telefónicas en formato electrónico.

- **Bases de datos o "bibliotecas" de información química o biológica**

Son bases de datos que almacenan diferentes tipos de información proveniente de la química, las ciencias de la vida o médicas. Se pueden considerar en varios subtipos:

- Las que almacenan secuencias de nucleótidos o proteínas.
- Las bases de datos de rutas metabólicas.
- Bases de datos de estructura, comprende los registros de datos experimentales sobre estructuras 3D de biomoléculas.
- Bases de datos clínicas.
- Bases de datos bibliográficas (biológicas, químicas, médicas y de otros campos): PubChem, Medline, EBSCOhost

1.3.7 VENTAJAS DE LAS BASES DE DATOS

- **Control sobre la redundancia de datos:**

Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos.

En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en

una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

- **Consistencia de datos:**

Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes.

- **Compartición de datos:**

En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

- **Mantenimiento de estándares:**

Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

- **Mejora en la integridad de datos:**

La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

- **Mejora en la seguridad:**

La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros.

- **Mejora en la accesibilidad a los datos:**

Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

- **Mejora en la productividad:**

El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación.

El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

- **Mejora en el mantenimiento:**

En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan.

Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.

Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias

a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

- **Aumento de la concurrencia:**

En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

- **Mejora en los servicios de copias de seguridad:**

Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

1.3.8 DESVENTAJAS DE LAS BASES DE DATOS

- **Complejidad:**

Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.

- **Coste del equipamiento adicional:**

Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina

más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara.

- **Vulnerable a los fallos:**

El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (Backup).

1.3.9 PRINCIPALES DBMS :

- **MySql:** es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.
- **PostgreSql y Oracle:** Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.
- **Access:** Es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa Access, el cual crea un archivo .mdb con la estructura ya explicada.
- **Microsoft SQL Server:** es una base de datos más potente que Access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones.

1.3.10 OTROS DBMS

- Borland Paradox
- Filemaker
- IBM DB2
- Interbase
- Microsoft FoxPro

CAPITULO II

ANÁLISIS Y REQUERIMIENTOS DEL HARDWARE Y SOFTWARE

2.1 ANÁLISIS DE REQUERIMIENTOS DEL HARDWARE

2.1.1 INTRODUCCIÓN

Para el análisis de los requerimientos del Hardware, es imprescindible conocer y tener claro la necesidad de la empresa, para automatizar el proceso de almacenamiento de los registros de horas laboradas de los empleados, con la automatización se suprimirá la tarea de tipiar los datos y posteriormente realizar los cálculos en Excel.

2.1.2 PROPÓSITO

El propósito es encontrar un hardware según los requerimientos para satisfacer las necesidades que el cliente tiene y solucionar el problema del almacenamiento de los datos.

2.1.3 DEFINICIONES, ABREVIACIONES

- Dispositivo: hardware a implementar.
- RH: Requerimiento del Hardware.
- BDD: Base de Datos.

2.1.4 PROCEDIMIENTO

Para presentar un detalle de los requerimientos que necesita cumplir el hardware se detallará una tabla con los siguientes puntos.

- **Número:** Identifica el requerimiento del hardware
- **Requisito:** es el requerimiento redactado.
- **Complejidad:** es el grado de complejidad del requerimiento.
Alta - Baja.
- **Prioridad:** Que tan importante es el requerimiento. Alto – Bajo
- **Descripción:** que es lo que se pretende con el requerimiento

- **Proceso:** detalla lo que debería hacer el hardware para cumplir el requerimiento.
- **Entrada:** Lo que espera el dispositivo para ejecutar y lograr el requisito.
- **Salida:** Lo que se espera como resultado del requisito.

A continuación desde la tabla 2.1 hasta la 2.6 se detallan los requerimientos del cliente.

Número :	RH1
Requisito :	El dispositivo permitirá el ingreso de datos que identifique al empleado.
Complejidad:	Baja
Prioridad:	Alta
Descripción :	El dispositivo permite al empleado ingresar sus datos para identificarse.
Proceso :	El dispositivo solicita al empleado ingresar los datos que le permita identificarse. Una vez que el empleado ha ingresado los datos, pulsa una tecla para enviar los datos a validar.
Entrada :	Código de identificación.
Salida:	Visualización de los datos ingresados.

Tabla N°2.1 Requisito RH1

Número :	RH2
Requisito :	Enviar a validar los datos
Complejidad:	Alta
Prioridad:	Alta
Descripción :	El dispositivo envía a una BDD a validar los datos ingresados.
Proceso :	El dispositivo envía los datos a un computador a través de una interfaz de comunicación donde se validan en una BDD.
Entrada :	Comando para enviar datos.
Salida:	Envío de los datos.

Tabla N°2.2 Requisito RH2

Número :	RH3
Requisito :	Permitir al empleado seleccionar la hora de registro.
Complejidad:	Baja
Prioridad:	Alta
Descripción :	El dispositivo debe presentar un menú al empleado.
Proceso :	El dispositivo solicita al empleado escoger una opción ya sea registrar una entrada, registrar una salida, o cancelar el registro.
Entrada :	Comando enviar datos + la opción escogida.
Salida:	El dispositivo envía la opción escogida a la BDD .

Tabla N° 2.3 Requisito RH3

Número :	RH4
Requisito :	Comunicación serial con PC
Complejidad:	Alta
Prioridad:	Alta
Descripción :	El dispositivo debe comunicarse permanentemente con un PC donde estará alojada una BDD para lo cual usará la comunicación serial RS232.La distancia entre el dispositivo y el PC es aproximadamente 13 m.
Proceso :	El dispositivo abre la comunicación con la BDD cuando el empleado manda a validar los datos.
Entrada :	Comando abrir comunicación.
Salida:	Comunicación establecida.

Tabla N°2.4 Requisito RH4

Número :	RH5
Requisito :	Dependencia del hardware de un software de conexión
Complejidad:	Alta
Prioridad:	Alta
Descripción :	El dispositivo depende de un software que sea el encargado de establecer la comunicación entre el PC – PIC teniendo la PC como maestro y el PIC como esclavo.
Proceso :	El dispositivo espera la inicialización del software en el PC para comenzar a entablar comunicación.
Entrada :	Comando de Inicialización de comunicación.
Salida:	Envío de datos de inicialización al PIC.

Tabla N°2.5 Requisito RH5

Número :	RH6
Requisito :	Mostrar hora y fecha.
Complejidad:	Alta
Prioridad:	Alta
Descripción :	El dispositivo debe mostrar la hora y fecha actual en el formato HH:mm:ss y dddd/dd/mmm/yyyy
Proceso :	El dispositivo luego de la inicialización del software en el PC, comienza a recibir la hora y fecha del sistema operativo.
Entrada :	Comando para mostrar fecha y hora
Salida:	Visualización de Hora y Fecha.

Tabla N°2.6 Requisito RH6

2.1.5 IDENTIFICACIÓN Y SELECCIÓN DE LOS COMPONENTES DEL HARDWARE

Luego de obtener los requerimientos y hacer un análisis se tiene la idea de que piezas conforman el hardware. Ahora se describirá las funciones del hardware y con cada función se mencionará los posibles componentes a elegir y su respectiva justificación.

2.1.5.1 Funciones del Hardware

- a) Permitir Ingreso de datos.
- b) Visualización de mensajes.
- c) Comunicación con la BDD.
- d) Procesamiento de datos.

a) Permitir ingreso de datos

Posibles componentes.

- Teclado.
- Biométrico.

Componente escogido

- Teclado.

Justificación

- Más económico.
- Conocimiento de su funcionamiento.
- Se ha realizado proyectos con matriz de teclados.
- Hay mucha información sobre su implementación.
- El cliente está de acuerdo.

Desventajas.

- El teclado debe interactuar con el microcontrolador para lograr su función.

- El microcontrolador debe ser programado para que interactúe con el teclado.

b) Visualización

Posibles componentes

- Matriz de Led.
- LCD.

Componente escogido

LCD.

Justificación

- Se necesita de pocas líneas programación para presentar los datos que en la matriz de led.
- Es más funcional que la matriz de led.
- Se requeriría de algunas matrices de led para desplegar la información, lo que aumentaría el costo.

c) Comunicación

Posible comunicación

- Serial.
- Paralelo.
- Wireless.

Comunicación escogida

- Comunicación Serial RS232.

Justificación

- La comunicación paralelo requiere más hardware para el envío y recepción de datos.
- La comunicación inalámbrica requiere de módulo de recepción y transmisión y también de hardware adicional lo que aumentaría el costo total.

- Se selecciona la **comunicación serial** debido a que es un estándar que cumple con los requisitos del cliente, en cuanto a la distancia nos permite alcanzar hasta los 15 m, mientras que otras comunicaciones como USB nos permite solo hasta 5 m.

d) Procesamiento de datos

Aunque no está muy explícito en los requisitos pero es necesario un microcontrolador (μC), ya que este es el corazón del hardware porque enlaza e interactúa a los componentes antes mencionados.

Posible microcontrolador

- PIC.
- AVR.

Microcontrolador Escogido

- PIC.

Justificación

- Pese a que los AVR hoy en día se han difundido mucho y son igual que funcionales que los PIC, la experiencia adquirida en proyectos anteriores con PIC es lo que prevalece para realizar el proyecto.

2.1.6 Definición de los componentes seleccionados

2.1.6.1 Teclado



Fig 2.1. Teclado Seleccionado

Un **teclado matricial** como el de la Figura 2.1 es un simple **arreglo de botones conectados en filas y columnas**, de modo que se pueden leer varios botones

con el mínimo número de pines requeridos. Un teclado matricial 4×4 solamente ocupa 4 líneas de un puerto para las filas y otras 4 líneas para las columnas, de este modo se pueden leer 16 teclas utilizando solamente 8 líneas de un microcontrolador. Si asumimos que todas las columnas y filas inicialmente están en alto (1 lógico), la pulsación de un botón se puede detectar al poner cada fila a en bajo (0 lógico) y checar cada columna en busca de un cero, si ninguna columna está en bajo entonces el 0 de las filas se recorre hacia la siguiente y así secuencialmente.⁵

2.1.6.2 Display LCD

Una **pantalla de cristal líquido** o **LCD** (acrónimo del inglés *liquid crystal display*) como se muestra en la Figura 2.2, es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora.⁶

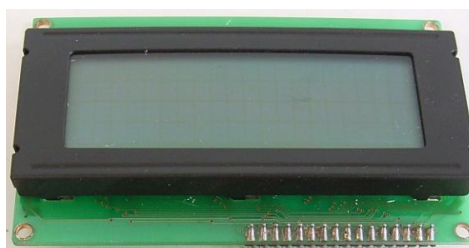


Fig 2.2 Display seleccionado.

2.1.6.3 Microcontrolador PIC 16f873A

Los **PIC** son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc⁷. y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instrument.

El nombre actual no es un acrónimo. En realidad, el nombre completo es **PICmicro**, aunque generalmente se utiliza como **Peripheral Interface Controller** (controlador de interfaz periférico).

⁵ Fuente <http://micropic.wordpress.com/2007/06/13/teclado-matricial-4x4/>

⁶ Fuente http://es.wikipedia.org/wiki/Pantalla_de_cristal_l%C3%ADquido

⁷ Microchip Technology Inc.- Industria creadora de los microcontroladores PIC

2.1.6.6 Descripción de los pines

En la tabla N°2.8 se detalla los pines del PIC 16f873A y una breve descripción de los mismos.

Pin	Tipo	Descripción
OSC1/CLKIN	E	Entrada de oscilador de cristal
OSC2/CLKOUT	S	Salida de oscilador de cristal
MCLR/VPP/THV	E/V	Reset o entrada de voltaje de programación
RA0/AN0	E/S	PUERTO A es un puerto bidireccional de E/S RA0 También puede ser entrada analógica 0
RA1/AN1 3 3	E/S	RA1 También puede ser entrada analógica 1
RA2/AN2/VREF	E/S	RA2 También puede ser entrada analógica 2 o referencia analógica de voltaje negativo
RA3/AN3/VREF +	E/S	RA3 También puede ser entrada analógica 3 o referencia analógica de voltaje positivo
RA4/T0CKI	E/S	RA4 Puede ser también la entrada de reloj del módulo Temporizador 0
RA5/SS/AN4	E/S	RA5 También puede ser entrada analógica 4 o esclavo en la selección de puertos serial síncrono
RB0/INT	E/S	PUERTO B es un puerto bidireccional de E/S RB0 También puede ser pin de interrupción externa
RB1, RB2	E/S	
RB3/PGM	E/S	RB3 Puede ser entrada de bajo voltaje de programación
RB4, RB5	E/S	
RB6/PGC	E/S	Reloj de programación serial
RB7/PGD	E/S	Dato serial de programación
RC0/T1OSO/T1CKI	E/S	PUERTO C es un puerto bidireccional de E/S RC0 Puede ser la salida del oscilador Timer1 o entrada de reloj de Timer1
RC1/T1OSI/CCP2	E/S ST	RC1 Puede ser la entrada del oscilador Timer1 o entrada de Captura2/salida de Captura2/PWM2
RC2/CCP1	E/S	RC2 entrada Captura1/salida Comparador1/salida PWM1
RC3/SCK/SCL	E/S	RC3 Puede ser entrada reloj de reloj de sincronía serial
RC4	E/S	RC4 puede ser entrada de dato SPI (en modo SPI)
RC5	E/S	RC4 puede ser salida de dato SPI (en modo SPI)
RC6/TX/CK	E/S	RC6 Puede ser pin transmisión USART
RC7/RX/DT	E/S	RC7 Puede ser pin de recepción USART
VSS	G	Referencia de tierra
VDD	V	Fuente positiva .

Tabla 2.8. Diagrama de pines PIC 16F873A.

2.2 ANÁLISIS Y REQUISITOS DEL SOFTWARE

En esta sección, se presentará primeramente un análisis general del estado actual del Registro Laboral que la fundación utiliza en la actualidad, para eso se recopilará información mediante una entrevista esperando conocer más acerca del hardware y software utilizado así como del proceso implicado.

Luego en un segundo punto se detallarán los requisitos que con el cliente se han analizado para en el futuro sirvan de guía para el desarrollo de la aplicación.

En un tercer punto se seleccionará y se describirá el software que se usara para el desarrollo de la aplicación así como de la Base de Datos a utilizar.

2.2.1 SITUACIÓN ACTUAL DEL SISTEMA DE REGISTRO

La mejor manera para saber la estructura y cómo funciona el registro laboral actual de la fundación, es en base a una entrevista, la cual puede ser vista en el apartado de ANEXOS (Entrevista) al final del documento, la misma nos ha dado un mejor criterio del proceso de operación que a diario se maneja y a la parte técnica que debe realizar los encargados de llevar el registro.

Se determinará cuáles serían los datos de entrada y la información de salida que el software necesitará para cumplir los objetivos.

En el apartado ANEXOS (Tarjeta - Reloj), se encuentra las imágenes de la tarjeta y el reloj tarjetero utilizado para el registro diario de empleados, también se muestra en formato en Excel que utilizan para el almacenamiento de los datos y en el cual se realizan los respectivos cálculos de las horas laboradas.

La recopilación de estos datos se utilizará para empezar a plantear los requerimientos que ayudarán a obtener un diseño que vaya acorde con las necesidades del cliente.

2.2.2 ESPECIFICACIÓN DE REQUERIMIENTOS DEL SOFTWARE.

El propósito es definir cuáles son los requerimientos que debe tener el software para que gestione el registro laboral de la fundación FEEP, para esto se necesitará la ayuda del encargado del registro laboral.

El producto Registro Laboral GSFEPP que se describirá es un programa que permitirá gestionar el registro de horas laboradas por los empleados.

Este sistema debe adaptarse al modo en que la actualidad se gestiona el sistema de registro, o sea, a las políticas de asistencia establecidas.

Además el programa debe ser capaz de emitir el reporte de horas laboradas por empleado hasta la fecha solicitada en cualquier momento, así como el reporte de faltas y feriados laborados.

2.2.2.1 Funciones del software.

Las funciones que debe realizar el software se puede clasificar en 2 bloques:

- **Conexión con la BDD.**

Permitir configurar el programa para que se conecte con la BDD.

- **Administración del sistema**

Los usuarios del sistema deberán gestionar perfiles de horario, departamentos, empleados, perfiles de usuario y reportes.

2.2.2.2 Políticas de la Fundación en cuanto a la asistencia.

- La tarjeta donde consta los registros de entrada y salida es el único documento que respalda la asistencia.
- Se debe cubrir las 8 horas diarias laboradas, 160 horas al mes.

- La entrada es a las 8:00, el cual debe ser registrado en la tarjeta en el casillero determinado para el horario de entrada de la mañana.
- Se tiene una salida al almuerzo de 45 min, y se debe registrar en la tarjeta en el respectivo casillero.
- La hora de salida es a las 16:30 la cual debe ser registrada en la tarjeta en el respectivo casillero.
- No hay horas extras ni reconocimiento económico por feriados o fines de semana laborados.
- Se considera una falta si no se halla registro en la tarjeta por lo menos en la hora de entrada en la mañana y salida en la tarde.
- Se justifica una falta presentando un certificado médico.
- Las faltas se descuentan de las vacaciones, o también de los feriados laborados.
- No hay límite de tiempo para timbrar la entrada, lo que quiere decir que no hay atrasos y por consiguiente no hay multas.

2.2.2.3 Requisitos Funcionales

2.2.2.3.1 Conexión con la BDD

Introducción: el sistema deberá permitir configurar la conexión a la BDD la primera vez que se ejecuta el programa, ingresando los datos requeridos para la conexión.

Entrada: aplastar ALT Gr + Enter, ingresar datos nombre de usuario de la BDD, contraseña y esquema de la BDD.

Proceso: el usuario ejecuta la aplicación por primera vez y antes de ingresar el Nick y clave deberá ingresar a configurar la BDD

Salida: conexión con la BDD exitosa.

2.2.2.3.2 *Ingreso al programa*

Introducción: el sistema deberá permitir ingresar el nombre de usuario y contraseña temporales predeterminados para la primera vez que ingrese al sistema.

Entrada: nombre de usuario temporal y clave temporal.

Proceso: el usuario digitará el nombre y clave, temporal. Luego deberá crear un usuario nuevo y eliminar el usuario temporal.

Salida: Ingreso al sistema.

2.2.2.3.3 *Gestión de usuarios*

Introducción: el sistema deberá permitir crear, seleccionar modificar, actualizar y eliminar usuarios.

Entrada: Nombre + Apellido + Nick+ Clave + perfil asignado.

Proceso: el usuario con perfil de administrador deberá crear nuevos usuarios, seleccionar y modificar sus datos si se requiere o eliminarlos.

Salida: Usuarios nuevos, actualización de datos o eliminación de usuarios.

2.2.2.3.4 *Gestión de perfiles de usuarios*

Introducción: el sistema deberá permitir crear, modificar, eliminar perfiles de usuarios.

Entrada: nombre de perfil + selección atributos.

Proceso: el usuario con privilegios de administrador deberá crear perfiles de usuarios y asignar los atributos para ese perfil.

Salida: perfil de usuario nuevo.

2.2.2.3.5 *Gestión de horarios*

Introducción: el sistema deberá permitir crear, modificar, eliminar perfiles de horario, los cuales serán asignado luego a los empleados.

Entrada: nombre de perfil de horario + hora de entrada + hora salida.

Proceso: el usuario con privilegios de administrador creará perfiles de horario según las políticas de la fundación para luego asignar a los empleados.

Salida: perfil de horario creado.

2.2.2.3.6 *Gestión de Departamentos*

Introducción: el sistema permitirá, modificar, eliminar departamentos para asignar a los empleados.

Entrada: nombre del departamento

Proceso: el usuario con privilegios de administrador ingresará el nombre del departamento a crear.

Salida: departamento creado.

2.2.2.3.7 *Gestión de Empleados*

Introducción: el sistema permitirá, crear, editar, actualizar, dar de baja, a los empleados; no se crearán empleados con el mismos id o código.

Entrada: Datos personales + código + datos opcionales.

Proceso: el usuario administrador ingresará los datos para crear al nuevo empleado.

Salida: empleado nuevo creado.

2.2.2.3.8 *Gestión de Reportes*

Introducción: permitirá ver un detalle de las horas laboradas, permisos, faltas, de los empleados, entre 2 fechas dadas e imprimir un reporte.

Entrada: Fecha 1 + Fecha 2

Proceso: se ingresará un rango de fechas de las cuales se requiere consultar el reporte de las horas laboradas por los empleados.

Salida: reporte de las horas laboradas por empleados.

2.2.3 SELECCIÓN DEL SOFTWARE

En esta sección se mencionará los posibles lenguajes tanto para la programación del microcontrolador como el de la aplicación y se seleccionará el que mejor se ajuste a los conocimientos por parte del desarrollador así como a la aplicación a desarrollar.

2.2.3.1 Lenguajes de programación para el microcontrolador

A continuación, en la tabla 2.9 se describe los lenguajes de programación para microcontroladores sus ventajas y desventajas.

Lenguaje	Ventajas	Desventajas
Basic	Es un lenguaje muy simple y con instrucciones fácilmente legibles, incluso por no expertos. Un compilador BASIC produce código que se ejecuta más rápido que un compilador de C.	Es muy complicado el manejo de interrupciones simultáneas. No optimiza el tamaño de memoria del programa del PIC. La mayoría de compiladores para este lenguaje pueden utilizarse únicamente bajo ambiente Windows.
C	Lenguaje de alto nivel más cercano a la máquina. Puedes construir rutinas matemáticas fácilmente. Puede ser de ayuda al combinarlo con Ensamblador sobre todo en la gama alta. Se pueden crear macros, para después simplificar el código en diferentes desarrollos.	Los programas al compilarlos pueden resultar un poco extensos y pesados por ello debe tenerse en cuenta la capacidad de memoria de programa del PIC a utilizar. Con este lenguaje tampoco se puede controlar del todo los tiempos y los registros bit a bit.

	Es aceptado por la empresa fabricante Microchip, incluso ellos tienen algunos compiladores C.	
Ensamblador	Lenguaje de bajo nivel Aprovechamiento eficiente de los recursos del PIC. Se pueden crear macros, para después simplificar el código en diferentes desarrollos. Se puede controlar los tiempos y los registros bit a bit. Excelente para manejar interrupciones simultáneas. Cuando se genera el archivo .hex éste es completamente optimizado.	Cuando no se tiene experiencia en programación puede tardarse en el desarrollo de alguna rutina en comparación con los otros lenguajes.

Tabla 2.9 Lenguajes de Programación para el microcontrolador.

2.2.3.2 Lenguaje Seleccionado para programación del microcontrolador

Se utilizará el **lenguaje Basic** para la programación del PIC, hay muchos ejemplos y se tiene experiencia en proyectos similares desarrollados con este lenguaje.

El entorno de desarrollo (IDE) será Microcode Studio con el compilador PICBasicPro y el programador ICProg, todos estos se pueden descargar desde el internet

2.2.3.3 Lenguaje de desarrollo para la aplicación

A continuación en la tabla 2.10, se describen las características del lenguaje estructurado y lenguaje orientado a objetos, las cuales darán la pauta para una futura elección del entorno de desarrollo.

Estructurado	Orientado a Objetos
• Los programas son más fáciles de entender, ya que pueden ser leídos de forma secuencial. • La estructura del programa es clara, puesto que las instrucciones están más ligadas o relacionadas entre sí. • Reducción del esfuerzo en las pruebas. El seguimiento de los fallos o errores del programa ("debugging") se facilita debido a la estructura más visible, por lo que los errores se pueden detectar y corregir más fácilmente. • Reducción de los costos de mantenimiento de los programas. • Programas más sencillos y más rápidos (ya que es más fácil su optimización). • Los bloques de código son auto explicativos, lo que facilita la documentación.	• Fomenta la reutilización y extensión del código. • Permite crear sistemas más complejos. • Relacionar el sistema al mundo real. • Facilita la creación de programas visuales. • Construcción de prototipos • Agiliza el desarrollo de software • Facilita el trabajo en equipo • Facilita el mantenimiento del software

Tabla 2.10 Características de los Lenguajes de comunicación

2.2.3.4 Selección del entorno de desarrollo IDE

Luego de analizar las características que ofrece los posibles lenguajes a utilizar, se ha optado por el entorno de desarrollo de Visual Basic .Net el cual está orientado a objetos y permitirá la reutilización de código desarrollado en aplicaciones similares, también se tendrá muy en cuenta aplicaciones de ejemplo desarrolladas en Visual Basic 6 sobre la comunicación serial.

2.2.3.5 Selección del sistema gestor de Base de Datos.

En la tabla 2.11 se detallan las características que se han tomado en cuenta para la selección de la base de datos.

Característica	MySQL	Microsoft SQL Server	ORACLE
Soporte de grandes volúmenes de datos	alta	alta	Alta
Multiplataforma	Si	Solo Windows	Si

Licencia	GPL o Propietario	Propietario	Propietario
Conocimiento por parte del desarrollador	mediano	bajo	Nulo
Documentación en Línea	alta	alta	Alta

Tabla 2.11 Características de la BDD

El sistema gestor de base de datos seleccionado será MySql por las características antes mencionadas así como por la experiencia que se tiene en proyectos anteriores.

Se utilizará la herramienta MySQL Administrator para la administración de la BDD y MySQL Query Browser para la ejecución de script y pruebas de sentencia SQL.

Estas herramientas incluidas MySQL están disponibles de forma gratuita en la web.

CAPITULO III

DISEÑO DEL HARDWARE Y SOFTWARE

3.1 DISEÑO DEL HARDWARE

Con los requerimientos y componentes del hardware citados en el capítulo II, se diseñará un circuito que permita la interconexión de todos los elementos que conforman el Hardware.

Para el diseño del circuito se usará una herramienta muy útil, se trata del paquete Proteus Profesional 7, el cual tiene programas que aparte de crear el circuito permiten hacer pruebas simuladas y verificar que el funcionamiento del circuito sea el adecuado.

3.1.1 DIAGRAMA DE BLOQUES DEL DISPOSITIVO

Para tener claro el prototipo, a continuación se muestra en la figura N° 3.1, el diagrama de bloques del hardware, el cual muestra las relaciones que existen entre los diferentes bloques.

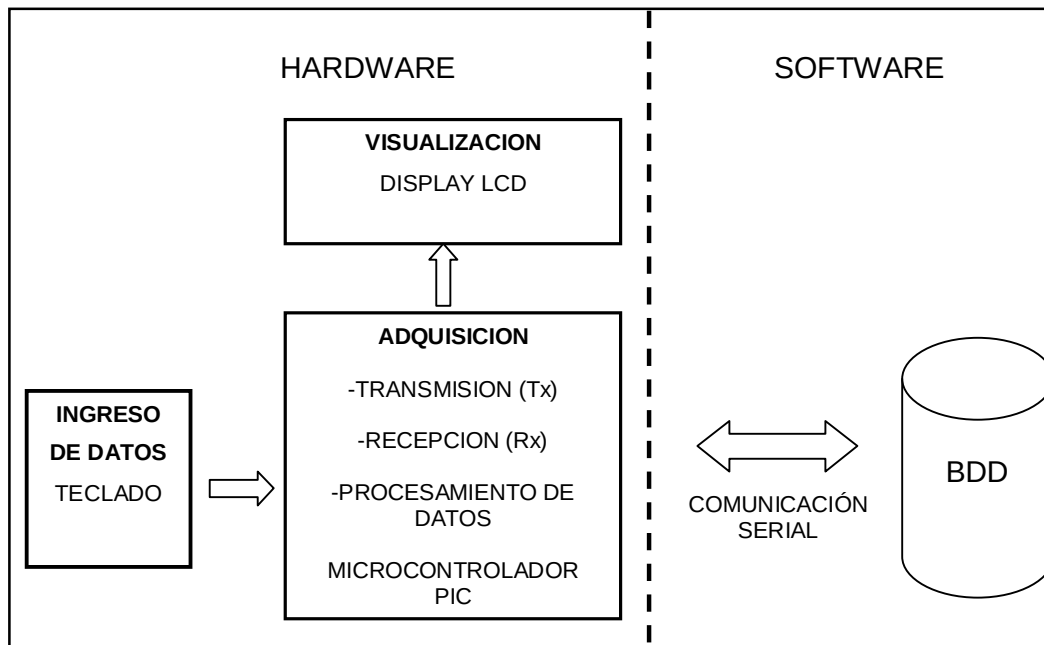


Fig. 3.1 Diagrama de Bloques

3.1.2 ESQUEMA DEL CIRCUITO EN PROTEUS

Para realizar el circuito se tomo de ejemplo varios diseños empleados en proyectos anteriores que fueron desarrollados en la carrera Universitaria.

A continuación, se muestra el esquema del circuito en el programa ISIS.

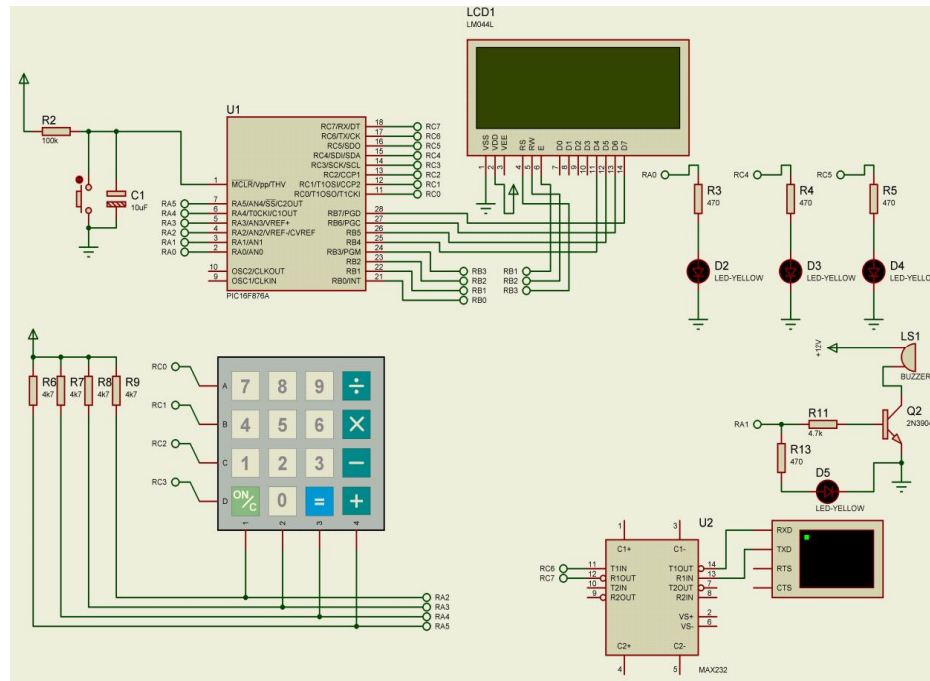


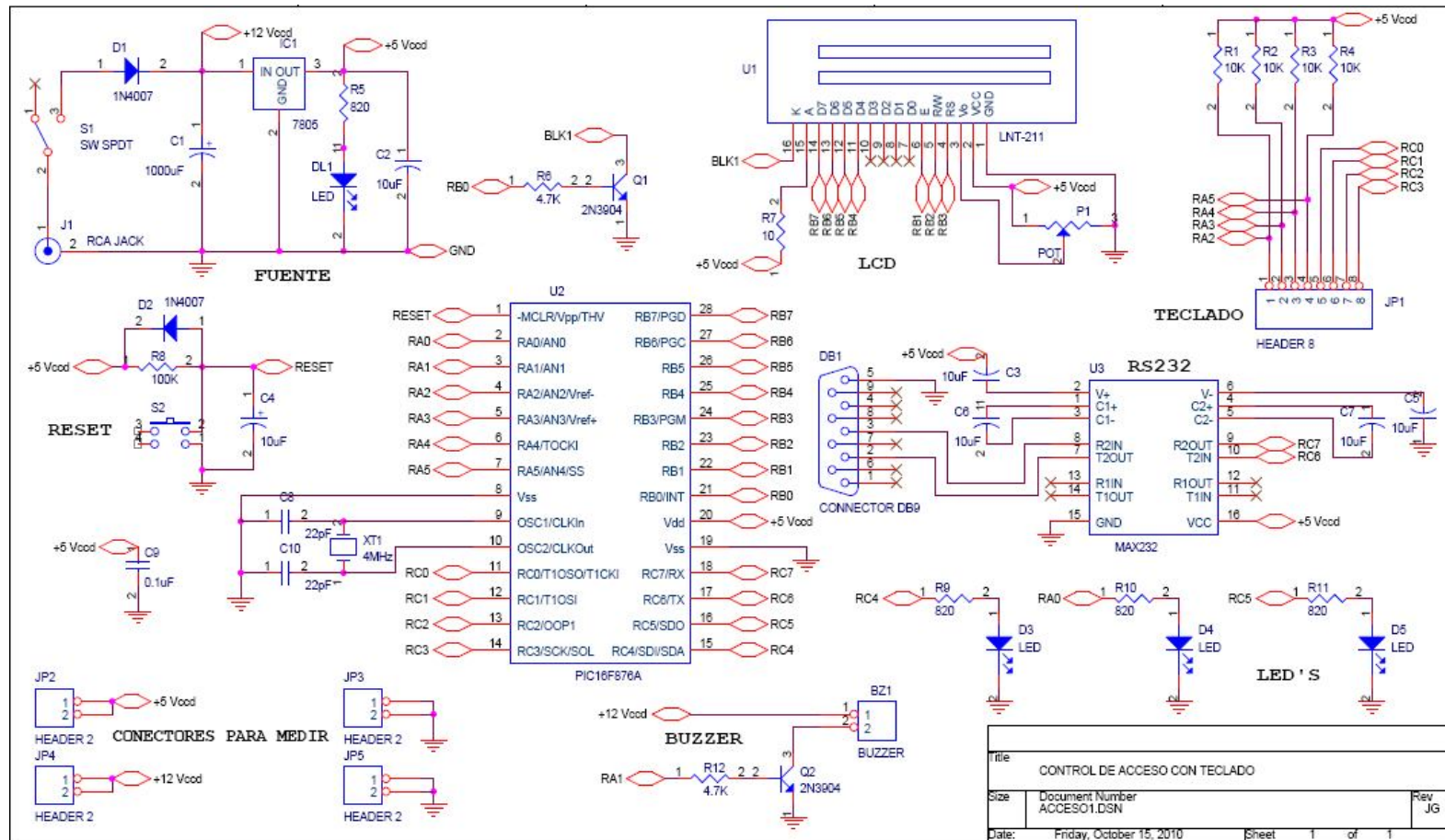
Fig. 3.2. Esquema del circuito en ISIS

En el esquema del circuito Figura 3.2 se observa los componentes que conforman el hardware así como las conexiones entre los mismos.

En la parte inferior derecha se encuentra un componente primordial en la comunicación serial se trata del MAX232 y una ventana de hyperterminal donde se visualizan los datos de la comunicación.

En la figura 3.3 se muestra el circuito general del hardware, está formado por los bloques principales antes mencionados y en conjunto con elementos extras como resistencias, capacitores, etc. que dan el funcionamiento deseado.

3.1.3 DIAGRAMA DEL CIRCUITO



3.1.4 DESCRIPCIÓN DEL DIAGRAMA DEL CIRCUITO

Para entender la descripción del diagrama circuito observar en la Figura 3.3 los nombres de los diferentes componentes que forman el circuito, los mismos que se detallan a continuación.

Fuente

Este bloque permite el ingreso del voltaje con un rango de 12 V – 10V para un óptimo funcionamiento, y tiene una salida de 5V con la cual trabajan los componentes a excepción de la bocina (Buzzer) que trabaja con una alimentación de 12V.

Reset

Este circuito conectado al pin 1 del PIC permite reinicializar el PIC cuando este se quede bloqueado.

Buzzer

La bocina, necesita una entrada de 12V y está conectado al pin RA1 del PIC.

LCD

Este componente solo usa el nible alto para la comunicación de datos con el PIC está conectado a los pines Rb4, Rb5, Rb6, Rb7 del PIC.

Es de 4 líneas por 20 columnas lo que permite mandar mensajes más largos, está conectado a un potenciómetro para regular la luminosidad del LCD.

Teclado

Es una matriz de 4*4, se conecta a 8 pines en 2 puertos diferentes del PIC del RC0 al RC3 las filas y del RA2 al RA5 las columnas.

MAX232

Este encapsulado permite que el PIC envíe datos hacia la PC a una distancia máxima de 15 m y también reciba los datos de la PC.

Está conectado a los pines RC7 para la recepción y al RC6 para la transmisión en el puerto USART del PIC.

Utiliza los pines 7 y 8 transmisión/recepción para conectarse al conector DB9 que servirá comunicarse con la PC y utiliza los pines 9 y 10 recepción/transmisión para conectarse al PIC.

Los pines 15 y 16 utilizan para conectarse a tierra y a 5V respectivamente.

LEDS

Conjunto de leds que sirven como indicador de acciones realizadas por el PIC.

Conectores para medir

Como su nombre lo indica son conectores que facilitan la medición del voltaje tanto el de entrada 12V y el de salida 5V que se encuentran en el bloque Fuente.

Microcontrolador PIC 16F873A

Es el encapsulado donde está almacenado el programa que gobierna el circuito por lo tanto, es uno de los componentes más importantes y del cual dependen los demás bloques.

Los pines 8 y 19 están conectados a tierra, mientras que en los pines 9 y 10 están conectados a un cristal de 4Mhz que da la frecuencia de reloj que utiliza el PIC.

3.1.5 RUTEO DEL CIRCUITO

En el paquete del Proteus también viene un módulo para realizar el ruteo del circuito accediendo desde la misma ventana del ISIS, dando clic en el icono ARES se abrirá una ventana como la figura 3.4 que permite importar el mismo circuito del ISIS y generar automáticamente el ruteo.

Es recomendable hacer el ruteo manualmente para colocar y ordenar los componentes de la mejor forma posible, es decir, teniendo una lógica de diseño, agrupando los elementos conforme según el circuito. El proceso es demoroso

porque se tiene que ir arrastrando elemento por elemento hasta tener todos los elementos en el diagrama, pero nos da un mejor resultado para que el ruteado salga de forma satisfactoria.

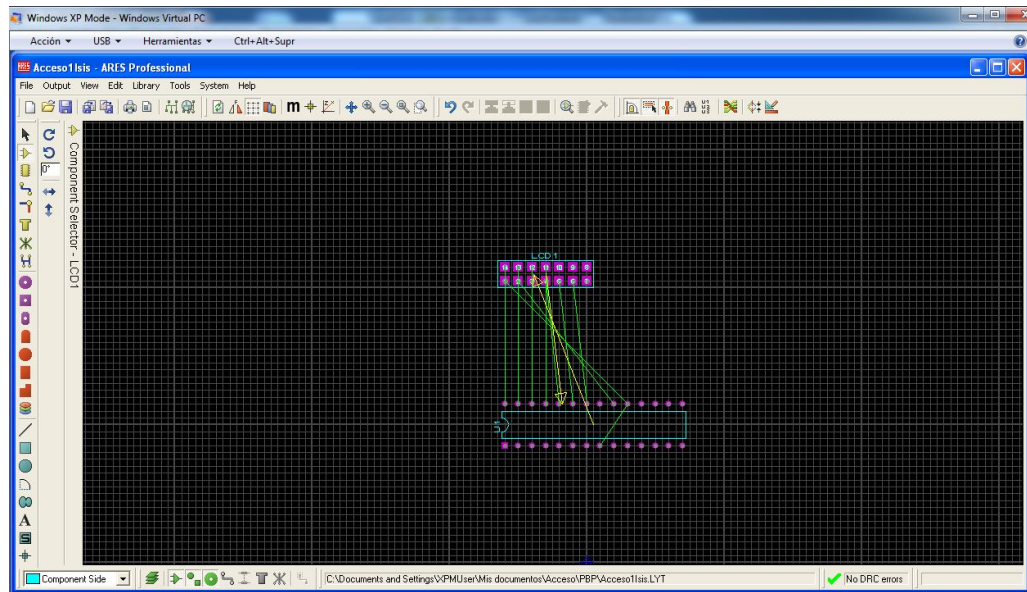


Fig. 3.4. Arrastrando elementos para el ruteo

3.1.6 GENERACIÓN DEL PCB (PRINT CIRCUIT BOARD)

Luego de elaborar el ruteado con nombres de las partes y demás, se procederá a generar las imágenes para impresión en la baquelita, esto se logra mandando a imprimir desde el mismo ARES o exportando las imágenes a un archivo BMP como el de la Figura 3.5 este proceso se llama generación del PCB (Print Circuit Board)

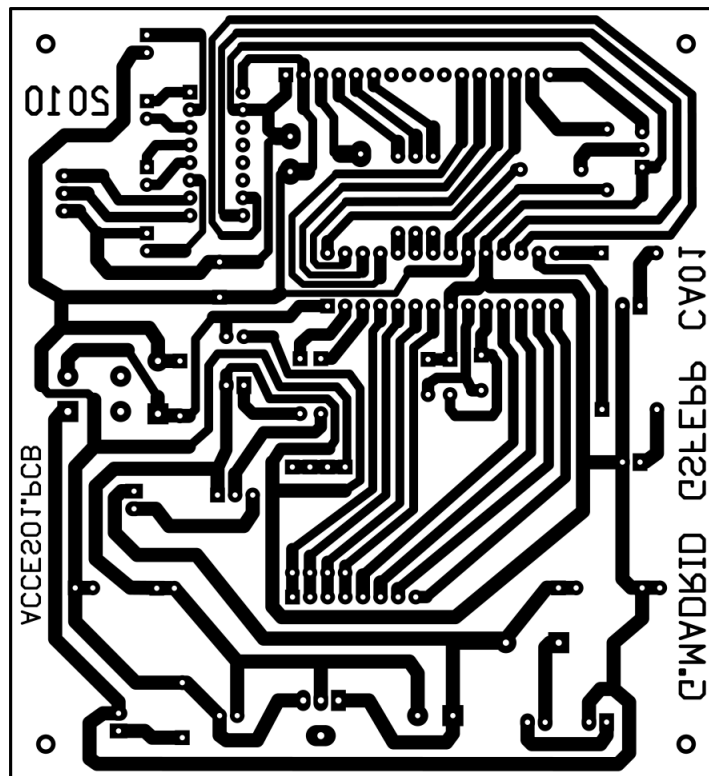


Fig. 3.5. Esquema de las pistas en formato BMP

Para no confundirse al momento de soldar se imprime la otra cara que es donde van los elementos esto dará una guía para saber la ubicación de los elementos con relación a las pistas como se muestra en la Figura 3.6

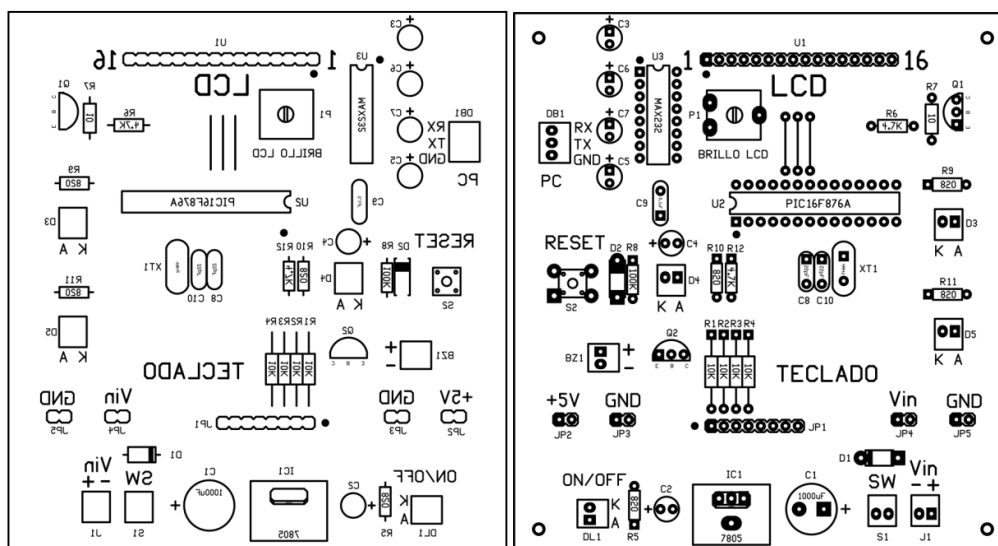


Fig.3.6. Esquema impreso de los elementos.

3.1.7 IMPRESIÓN DEL CIRCUITO EN LA BAQUELITA POR TRANSFERENCIA TÉRMICA.

Una vez diseñada las pistas se procederá a utilizar una impresora laser y un papel de transferencia térmica para imprimir el proyecto.

Para esto, se coloca el papel en la baquelita que para el proyecto será de 10*12(ancho * alto) y se procederá a pasar la plancha por encima del papel hasta lograr que el circuito quede impreso en la baquelita como se muestra en la Figura N° 3.7.

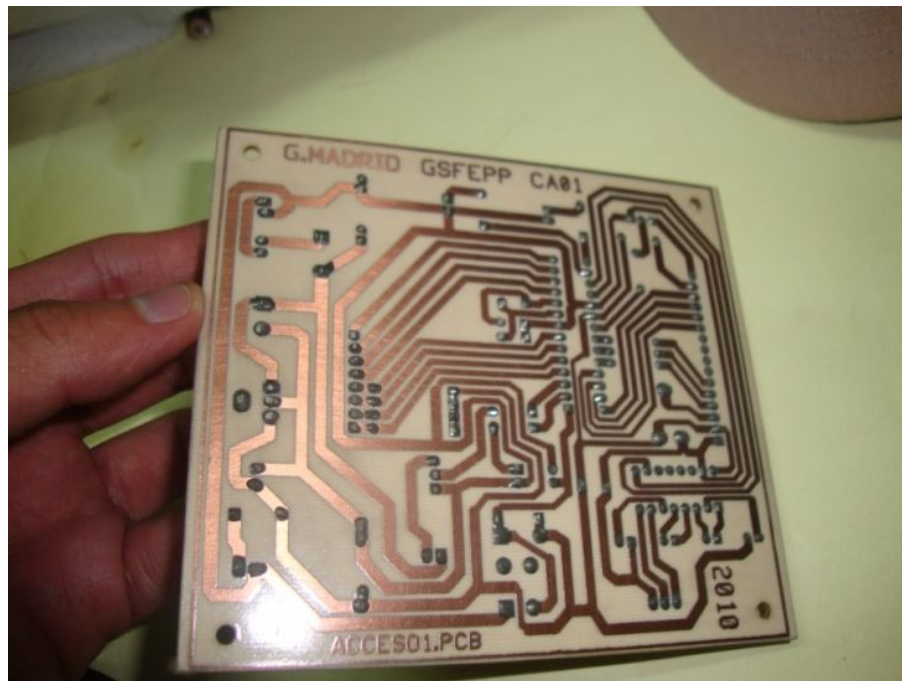


Fig. 3.7. Impresión en la Baquelita.

3.1.8 PROGRAMACIÓN DEL PIC

3.1.8.1 Introducción

Para la programación del PIC se utilizó la herramienta Microcode Studio, es una herramienta en lenguaje Basic por lo que la programación no es complicada y evita la dificultosa tarea de programar en lenguaje máquina.

Como el microcontrolador solo entiende lenguaje de bajo nivel, el Microcode tiene un traductor llamado PICBASIC PRO (PBP) que convierte el código Basic en código máquina, generando archivos .hex el cual es cargado en el PIC por medio de un quemador de PICs .

Este archivo .hex puede ser usado también en el PIC del PROTEUS para la simulación del circuito ya que como se mencionó anteriormente para que funcionen todos los componentes debe estar cargado un programa en el PIC para que este interactúe con los demás componentes.

Para la programación del PIC se tomó en cuenta los requerimientos del hardware mencionados en el capítulo II, los mismos que se obtuvieron de un análisis previo realizado conjuntamente con el cliente.

En el siguiente diagrama de flujo, se detalla el funcionamiento del dispositivo, el mismo que nos da una idea de que datos son los que se necesita ingresar, que operaciones realizará si ocurren determinadas condiciones y también las salidas que vendrían hacer los mensajes que visualiza el empleado así como los datos que se envían a la BDD.

3.1.8.2 Diagrama de flujo del funcionamiento del Hardware.

Para una mejor comprensión del proceso de funcionamiento, se detalla en la Figura 3.8 el flujo de secuencia que sigue el hardware.

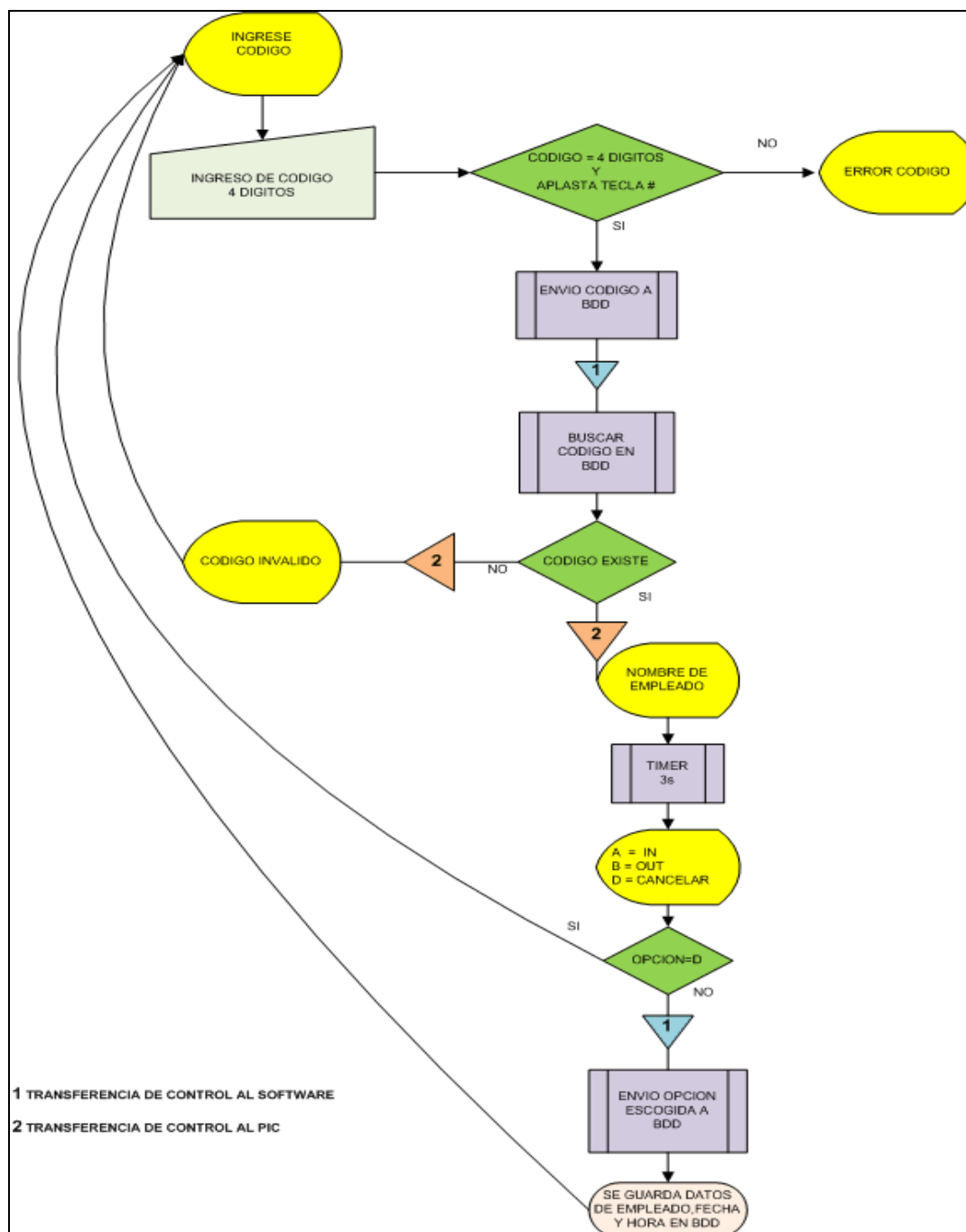


Fig. 3.8. Diagrama de Flujo funcionamiento del Hardware

3.1.8.3 Protocolo de comunicación PC - PIC.

Para establecer la comunicación del PC – PIC y viceversa se ha hecho uso del p rtico rs232 del PC y del m dulo de transmisi n serial USART en el PIC ambos a una velocidad de 9600 bps.

Para la programaci n de la comunicaci n se ha analizado lo investigado en cuanto a los modos de comunicaci n y tambi n tomado en cuenta c mo se comunican los seres.

Protocolo de comunicaci n de 2 personas.

Primero hay requisitos previos que se deben cumplir:

- Ambas personas tengan un medio de comunicaci n, puede ser el aire que lleve las ondas sonoras, cable telef nico, etc.
- Que ambas personas hablen el mismo lenguaje para que se puedan entender lo que dicen.
- Que ambas personas est n sincronizadas, es decir, que no hablen al mismo tiempo.

Luego que se cumplan estos requisitos m nimos, se analizar  c mo se comunican.

- La persona A manda un “Hola” como comando a la persona B a trav s del medio de comunicaci n.
- La persona B recibe el comando y responde con el mismo comando “Hola”, con esto la persona A sabe que el mensaje fue recibido por la persona B y se ha establecido la comunicaci n.
- La persona A manda otro comando “Como estas”, la persona B interpreta el comando y responde con otro comando “Bien gracias”

Lo anterior es un ejemplo claro y simple de c mo se establece la comunicaci n entre los seres humanos.

Ahora se seguirá el ejemplo anterior para modelar un protocolo de comunicación entre el PC y PIC del prototipo, que permita interpretar y luego pasar a código de programación, con el programa Microcode.

3.1.9 PRERREQUISITOS DE LA COMUNICACIÓN PC-PIC DEL PROTOTIPO.

- Medio de comunicación: Interfaz serial, puerto COM del PC y módulo de transmisión serial USART del PIC.
- Velocidad de comunicación: 9600 bps

3.1.10 SECUENCIA DE COMANDOS PARA LA COMUNICACIÓN

En la siguiente tabla se detalla la secuencia de operación que ayudará en la programación tanto del PIC, como del motor de conexión con la BDD.

PIC	PC(motor de conexión)
1) Espera carácter de inicialización de comunicación	
	2) Envía carácter "A"
3) Inicializa operaciones, despliega en pantalla mensaje "Ingrese Código"	
	4) Envía carácter "Z" cada segundo aproximadamente de aquí hasta que se detenga el software por alguna causa.
5) Recibe el carácter "Z" y se prepara para recibir datos de hora y fecha, cada segundo aproximadamente.	
	6) Envía carácter por carácter la hora y fecha del sistema operativo en unas milésimas de segundo aproximadamente.
7) Recibe carácter por carácter la hora, fecha y lo muestra en el LCD, hasta no detectar una interrupción de transmisión en el USART.	
8) Detecta que el usuario ingreso 4 dígitos y pulso la tecla #, entonces envía los 4 dígitos.	

	9) Detecta que en el buffer de entrada hay 4 dígitos y manda a buscar en la BDD los 4 dígitos. Valida los 4 dígitos y devuelve el nombre correspondiente al código digitado. Envía el carácter "B"
10) Detecta el carácter "B" y se prepara para recibir datos.	
	11) Envía el nombre del Empleado correspondiente al código ingresado carácter por carácter, debe completarse 20 espacios, si el nombre no llega a 20 caracteres se completará con espacios en blanco.
12) Recibe los caracteres y los despliega en el LCD.	
13) Detecta 20 caracteres, espera unos segundos y cambia a menú de opciones. A= in, B=out, C=cancelar.	
14) Envía código de opción seleccionada por el usuario en formato de 4 dígitos. 000E para registra una entrada. 000S para registrar una salida. Si opción es igual a "000C" regresa a pedir código.	
	15) Detecta código de 4 dígitos. Guarda en la BDD un registro de la hora y el código del empleado. Envía carácter C para terminar la comunicación
Este sería el fin del proceso luego se regresaría al tercer paso detallado en esta tabla y se repetiría el bucle indefinidamente.	

Tabla N° 3.1 Secuencia de Comandos

Con toda la documentación y análisis realizado se procedió con la programación del PIC y del motor de comunicación de la BDD.

3.1.11 CÓDIGO DE PROGRAMACIÓN DEL PIC

A continuación, se detalla las líneas principales de configuración de los dispositivos como teclado, LCD y del PIC.

'DEFINICIONES INICIALES fusibles de configuración

```
'@ DEVICE PIC16F873A, XT_OSC, WDT_OFF, PWRT_OFF, BOD_OFF, LVP_OFF
'@ DEVICE PIC16F873A, CPD_OFF, WRT_OFF, DEBUG_OFF, PROTECT_OFF
'@ DEVICE PIC16F873A
```

'DEFINICIONES INICIALES PARA CONFIGURAR EL HARDWARE DEL LCD*****

' Define el portico de Datos
DEFINE LCD_DREG PORTB

' Define el Bit de inicio de los Datos (0 or 4) si el bus es de 4-bit
DEFINE LCD_DBIT 4

' Define el p rtico en donde se encuentra el Bit E (Enable) del LCD
DEFINE LCD_EREG PORTB

' Define la posicion del bit en el portico del Bit E (Enable) del LCD
DEFINE LCD_EBIT 1

' Define el p rtico en donde se encuentra el Bit R/W (Read/Write) del LCD
DEFINE LCD_RWREG PORTB

' Define la posicion del bit en el portico del Bit R/W del LCD
DEFINE LCD_RWBIT 2

' Define el p rtico en donde se encuentra el Bit RS (Register Select) del LCD
DEFINE LCD_RSREG PORTB

' Define la posicion del bit en el portico del Bit RS (Register Select) del LCD
DEFINE LCD_RSBIT 3

' Define el tama o del bus de datos del LCD (4 or 8 bits)
DEFINE LCD_BITS 4

' Define el numero de lineas del LCD
DEFINE LCD_LINES 4

' Define el tiempo de retardo para el envio del comando en microsegundos (us)
DEFINE LCD_COMMANDUS 2000

' Define el tiempo de retardo para el envio del dato en microsegundos (us)
DEFINE LCD_DATAUS 50

' Definiciones del USART
DEFINE HSER_RCSTA 90h
DEFINE HSER_TXSTA 24h
DEFINE HSER_BAUD 9600
DEFINE HSER_CLROERR 1
INTCON = %11000000 ' Habilita Interrupciones, general y perif ricos
ON INTERRUPT GoTo serialin ' Declara manejador de la interrupci n
PIE1.5 = 1 ' Habilita Interrupci n del USART

'DEFINICIONES INICIALES PARA CONFIGURAR EL TECLADO *****

```
F1 VAR PORTC.0    'nombres para los pines de las filas
F2 VAR PORTC.1
F3 VAR PORTC.2
```

```
F4 VAR PORTC.3
```

```
C1 VAR PORTA.2 'nombres para los pines de las columnas
```

```
C2 VAR PORTA.3
```

```
C3 VAR PORTA.4
```

```
C4 VAR PORTA.5
```

```
DEFINE OSC 4 'definir oscilador externo de 4 MHZ
```

```
TRISA = %00111100 'Setea PORTA
```

```
TRISB = %00000000 'Setea PORTB
```

```
TRISC = %10000000 'Setea PORTC
```

```
ADCON1 = 7 'Portico A, como digitales
```

```
serialin:
```

```

    HSerin [char] ' Lee USART y almacena un carácter en CHAR
'
    pause 1
    select case char
        case "A" ' El pic está listo para enviar codigo de 4 digitos
            flag4=1
        case "B" ' El pic está listo para recibir datos y desplegarlos en el LCD.
            flag6=1
        case "C" ' El pic termina de recibir datos y retoma al estado de pedir el código
            flag8=1
        case "Z" ' El pic comienza a escribir datos fijos en LCD
            lcdout $fe,$9A
            HSerin [char]
            LCDOUT CHAR
            HSerin [char]
            LCDOUT CHAR,":"
            HSerin [char]
            LCDOUT CHAR
            HSerin [char]
            LCDOUT CHAR,":"
            HSerin [char]
            LCDOUT CHAR
            HSerin [char]
            LCDOUT CHAR
            HSerin [char]
            lcdout $fe,$D6
'
            LCDOUT CHAR
            HSerin [char]
            if char="1" then lcdout "Dom" ;mostrar día de la semana
            if char="2" then lcdout "Lun"
            if char="3" then lcdout "Mar"
            if char="4" then lcdout "Mie"
            if char="5" then lcdout "Jue"
            if char="6" then lcdout "Vie"
            if char="7" then lcdout "Sab"
            LCDOUT "/"
            HSerin [char]
            LCDOUT CHAR
            HSerin [char]
            LCDOUT CHAR, "/"
            HSerin [char]
            tempo1=char-$30
'
            LCDOUT CHAR
            HSerin [char]
            tempo2=char-$30

```

```

tempo3=(tempo1*10) +tempo2
if tempo3=1 then lcdout "Ene"      ;mostrar el mes
if tempo3=2 then lcdout "Feb"
if tempo3=3 then lcdout "Mar"
if tempo3=4 then lcdout "Abr"
if tempo3=5 then lcdout "May"
if tempo3=6 then lcdout "Jun"
if tempo3=7 then lcdout "Jul"
if tempo3=8 then lcdout "Ago"
if tempo3=9 then lcdout "Sep"
if tempo3=10 then lcdout "Oct"
if tempo3=11 then lcdout "Nov"
if tempo3=12 then lcdout "Dic"
LCDOUT "/20"
    HSerin [char]
LCDOUT CHAR
    HSerin [char]
LCDOUT CHAR," "
end select
'
    pause 1
Resume      ' Retorna de la interrupcion
enable     ' Habilita otra vez las interrupciones

```

3.1.12 DISPOSITIVO IMPLEMENTADO

Una vez fabricada la baquelita con el circuito impreso y grabado el programa en el PIC se prosiguió a ubicar los componentes en una caja plástica para proyectos y dando forma a la misma para que tenga estética y vaya de acuerdo a lo planeado. En la figura N° 3.9, se observa el dispositivo armado y listo para las pruebas.



Fig. 3.9. Dispositivo Implementado

3.2 DISEÑO DEL SOFTWARE

Luego de realizado el análisis de requerimientos se procederá a diseñar el software para lo cual, se utilizará varios diagramas para comprender lo que se desea realizar.

3.2.1 ARQUITECTURA DEL SISTEMA

El modelado del sistema nos da una visión general de cómo esta delineado el sistema dando un firme conocimiento de los tipos de objetos que se encuentran dentro del contexto del sistema.

3.2.1.1 Diagrama de clases

Es un diagrama estático que describe la estructura del sistema mostrando sus clases, atributos y las relaciones entre ellos.

Mediante este diagrama se describen los elementos que deben existir en el modelado de nuestro sistema.

En la figura 3.10 se muestra el diagrama de clases.

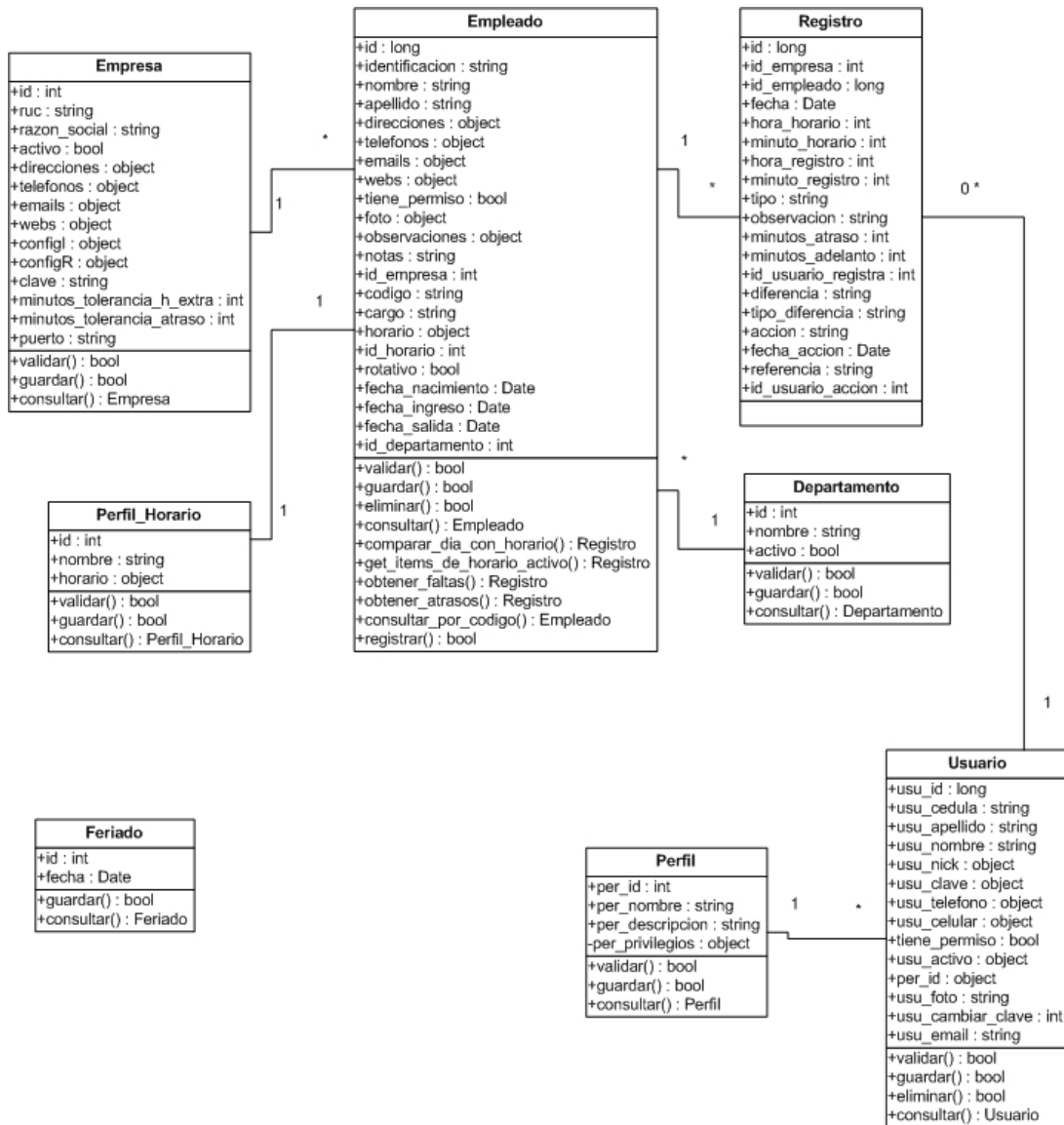


Fig. 3.10 Diagrama de Clases

En la tabla 3.14, se describen las clases más relevantes que conforman el sistema.

Clase	Descripción
Empresa	Se encuentran variables para presentar información como dirección, teléfono, email, etc. Además se encuentran una variable donde se ingresa el puerto de comunicación serial

Empleado	Se encuentran variables de información personal como identificación, nombre, apellido, teléfono, email, etc. Además se encuentra la variable código donde se ingresa el numero asignado para poder registrar las horas de entrada y salidas.
Usuario	Se encuentran variables para presentar nombre, apellido, clave, Nick, perfil de usuario, etc.
Registro	Contiene las variables para presentar la fecha, horas, minutos de registro de entrada o salida así como variables para operaciones de minutos de atraso o adelanto.
Departamento	Contiene variables para presentar nombre del departamento y su estado

Tabla N° 3.2 Descripción de Clases

3.2.1.2 Diagrama de objetos

Este diagrama de ámbito también estructural muestra las instancias específicas de las clases en forma resumida.

En la figura 3.11 se muestra el diagrama de objetos.

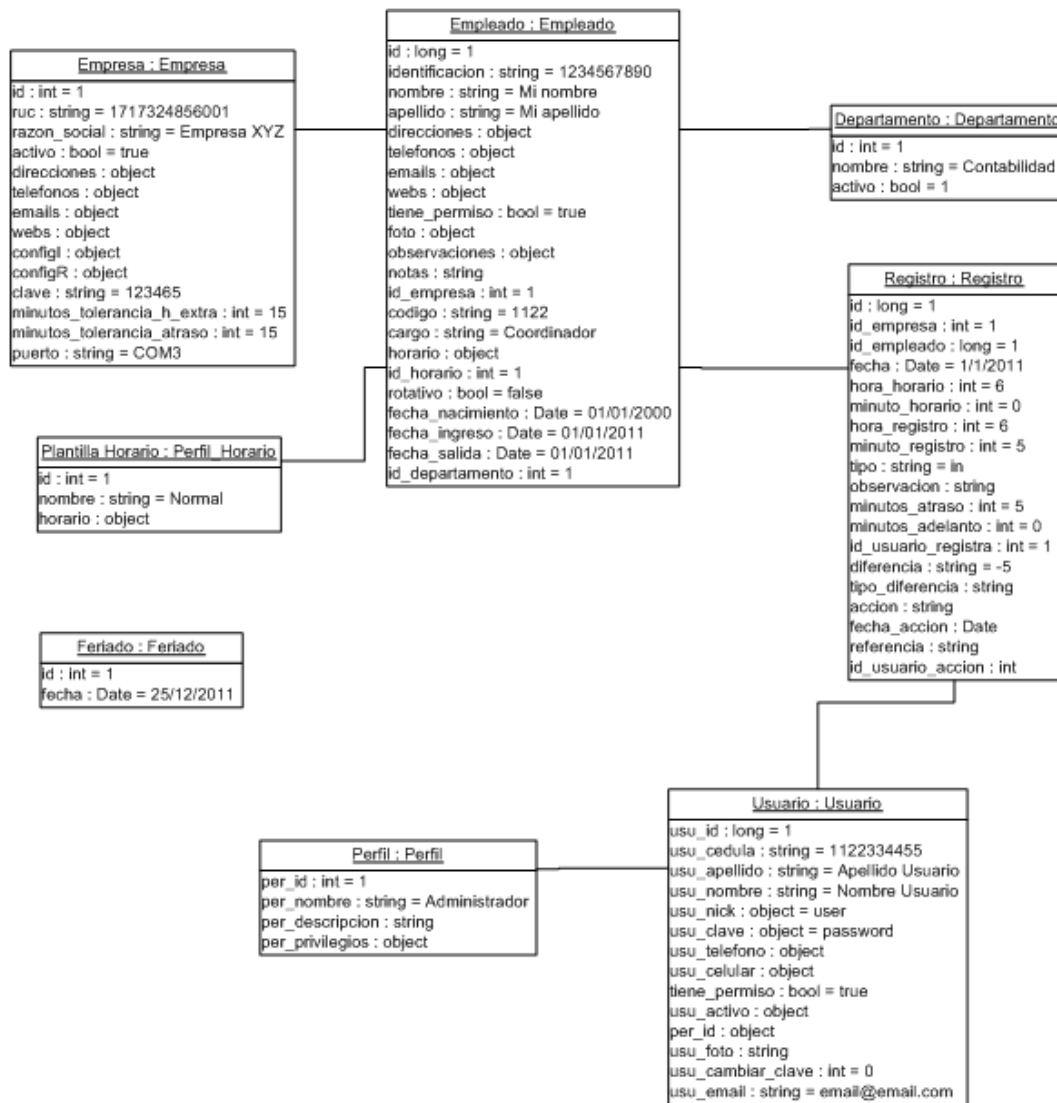


Fig. 3.11 Diagrama de Objetos

3.2.1.3 Diagrama de casos de usos

Mediante el diagrama de casos de usos se pretende capturar todas las acciones y comportamiento que los actores (usuarios), realizan en el sistema.

En la figura 3.12 se representa los casos de usos y los actores que intervienen en cada uno de ellos.

Luego en la tabla 3.2, se detallan los actores y se describe como cada uno interactúa con el sistema.

Desde la tabla 3.3 hasta la tabla 3.13, se describen cada uno de los casos de uso para una mejor comprensión de la funcionalidad de los mismos.

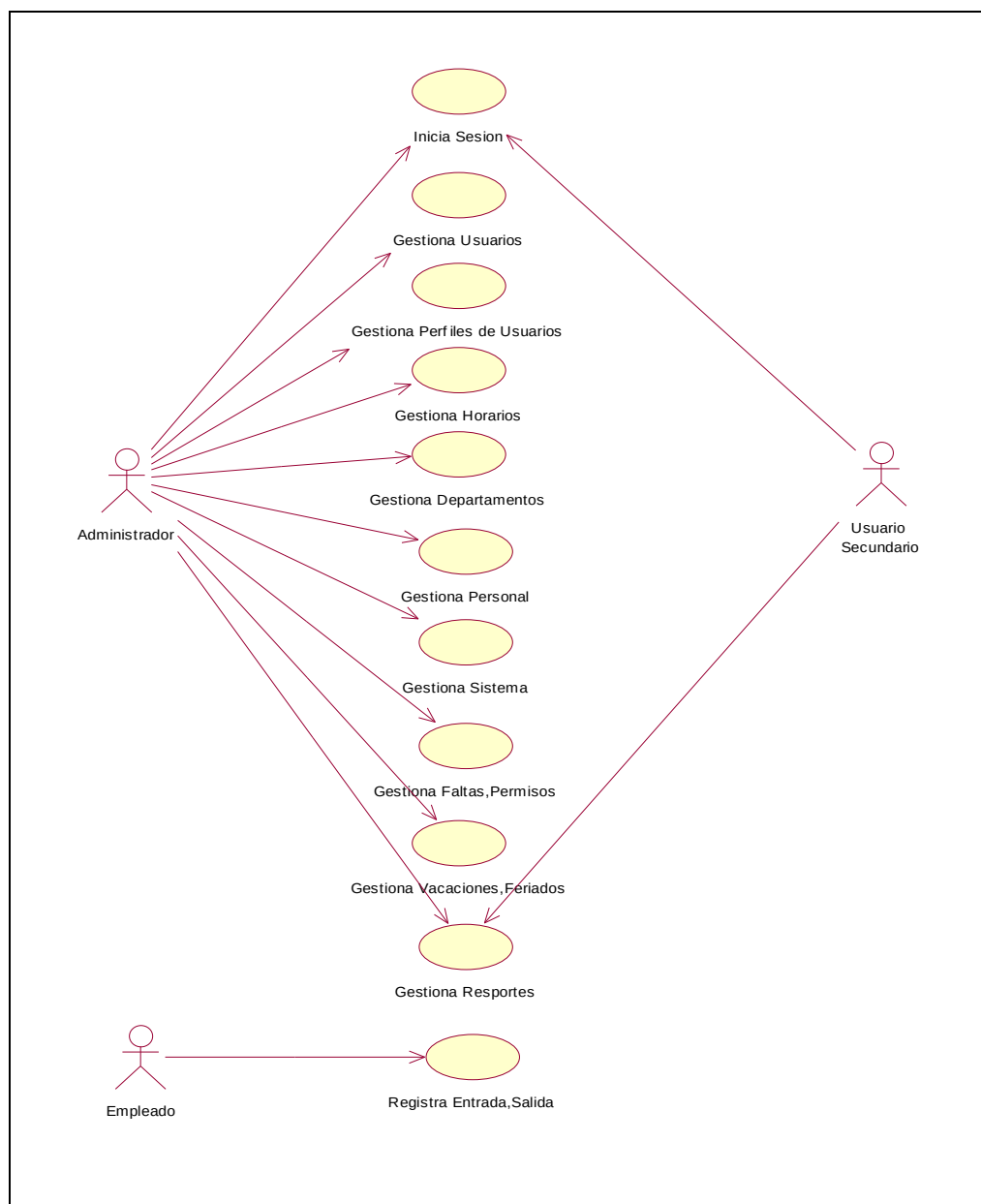


Fig. 3.12 Diagrama casos de usos

3.2.1.4 Actores

En la tabla 3.2 se muestran los actores que intervienen en el sistema y una descripción de los mismos.

ACTORES	DESCRIPCION
ADMINISTRADOR	Usuario que inicia la ejecución del sistema Encargado de crear, editar, modificar, actualizar usuarios. Asignar perfiles a los usuarios. Crear, editar modificar departamentos. Crear, editar modificar perfiles de horario. Gestionar Empleados. Imprimir Reportes. Tiene todos los privilegios sobre el sistema.
USUARIOS SECUNDARIOS	Encargado de: Gestionar empleados. Imprimir Reportes. Tiene privilegios limitados sobre el sistema.
EMPLEADO	Registra las horas de entrada y salida.

Tabla N° 3.3 Actores y Actividades

3.2.1.5 Diagramas de Secuencia

A continuación se presentan los diagramas de secuencias para los caso de uso pertinentes, esto permitirá tener una idea de las perspectiva cronológica de las iteraciones así como la secuencia de los mensajes entre el actor y el sistema dentro de un escenario en concreto.

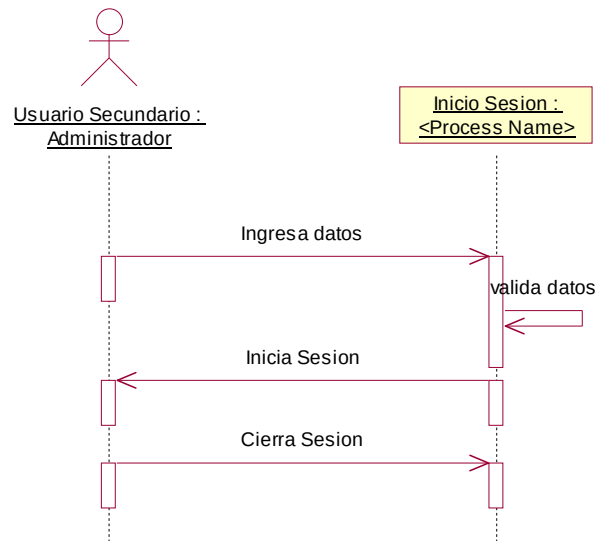


Fig. 3.13 Diagrama de secuencia Inicio Sesión

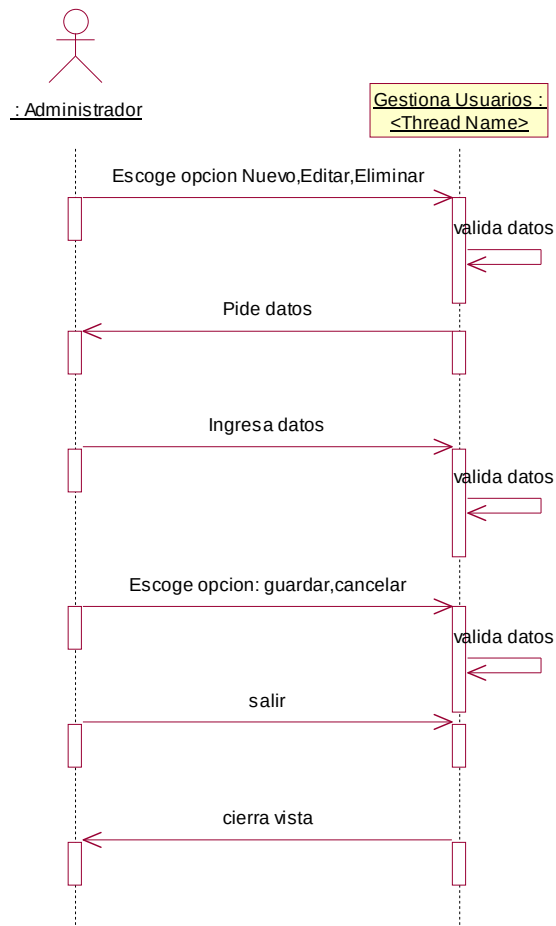


Fig. 3.14 Diagrama de secuencia Usuarios

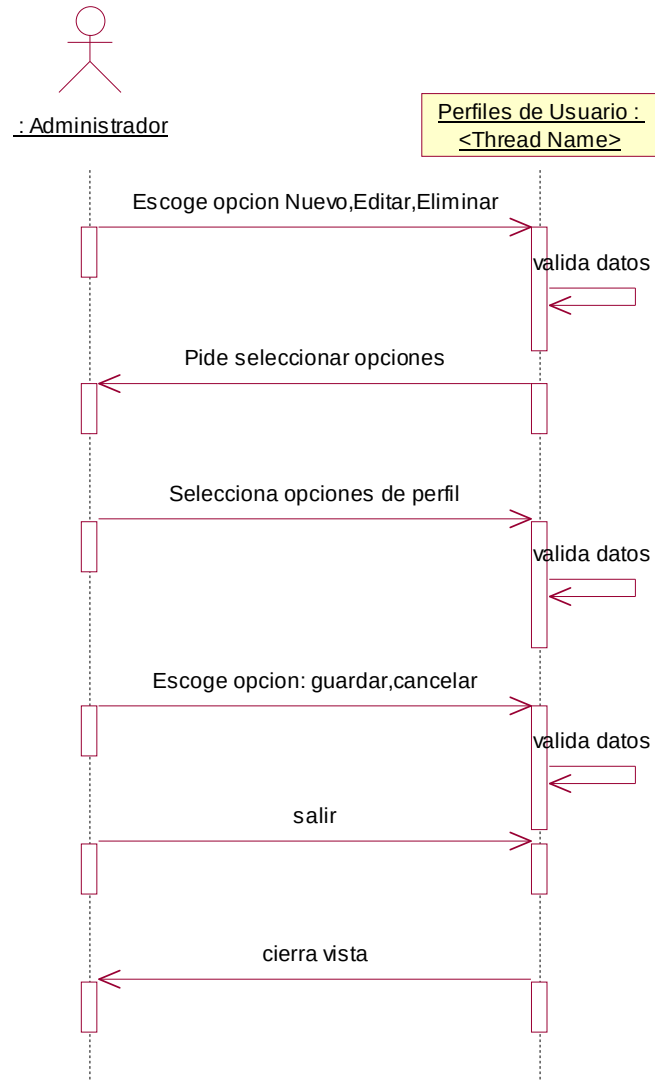


Fig. 3.15 Diagrama de secuencia Perfiles

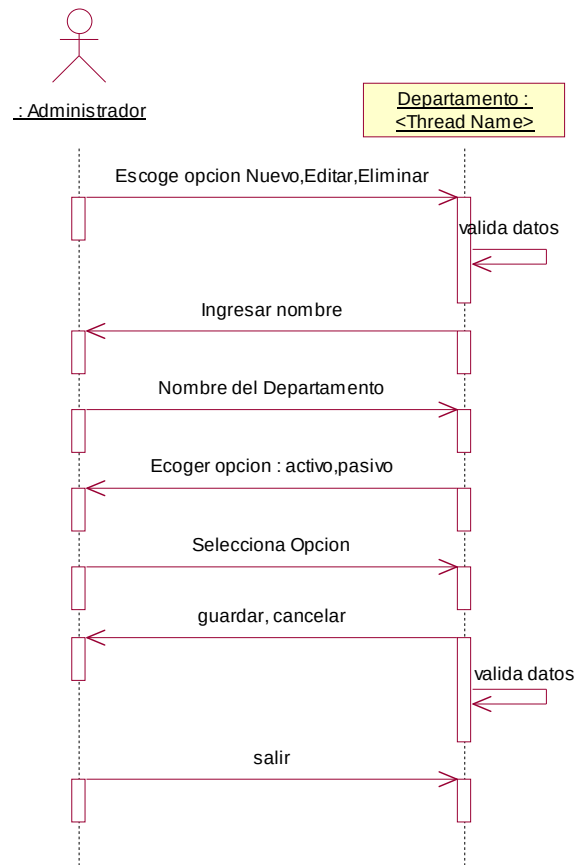


Fig. 3.16 Diagrama de secuencia Departamento

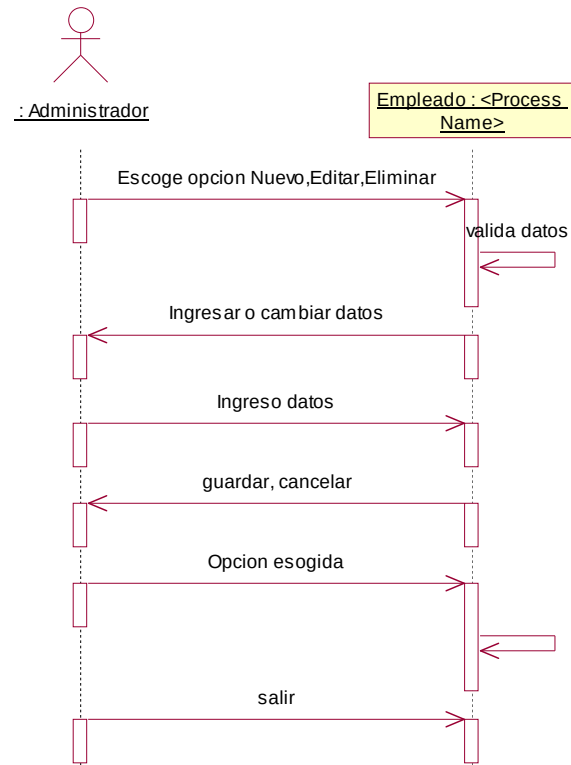


Fig. 3.17 Diagrama de secuencia Empleado

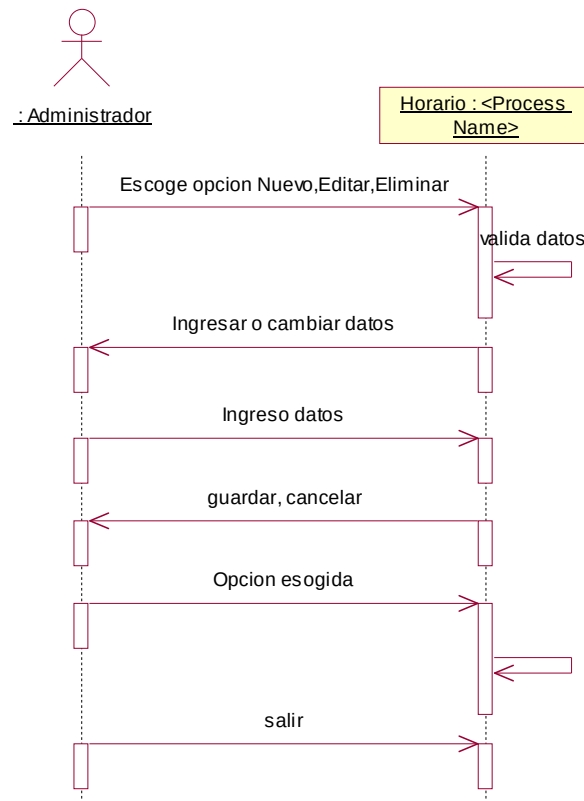


Fig. 3.18 Diagrama de secuencia Usuarios

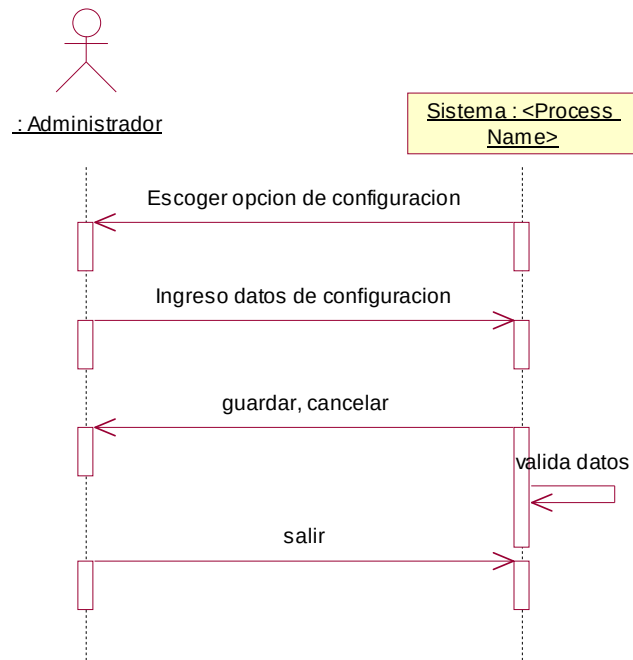


Fig. 3.19 Diagrama de secuencia Sistema

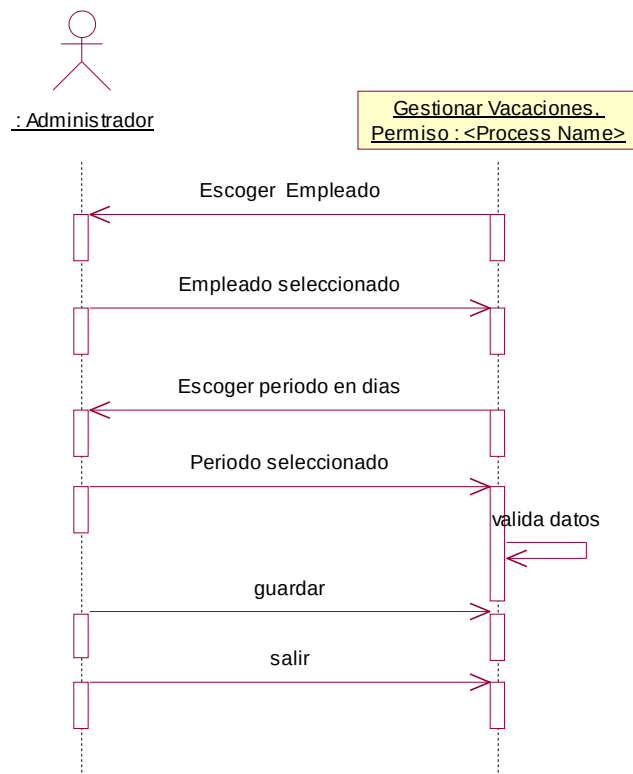


Fig. 3.20 Diagrama de secuencia Gestión Feriado-Vacaciones

3.2.1.6 Descripción de los casos de Uso

En esta sección Desde la tabla 3.4 hasta la tabla 3.14, se describirá cada caso de uso detallando el nombre del caso de uso, propósito, actores, precondiciones, poscondiciones, el flujo, los pasos y la acción que deberá realizar.

# Caso Uso	CU1	
Nombre	Inicia Sesión	
Propósito	Iniciar Sesión en el sistema	
Actores	Administrador, Usuario Secundario.	
Precondiciones	El administrador debe haber creado ya usuarios en el sistema.	
Poscondiciones	La ventana y menús están disponibles según los privilegios del usuario	
Flujo	Pasos	Acción
Principal	1	El usuario ejecuta la aplicación
	2	El sistema muestra la pantalla de ingreso de datos de nombre y clave para iniciar sesión
	3	El Administrador o el usuario secundario ingresan los datos de nombre y clave
	4	El sistema valida los datos. Si son validos los datos ingresados el sistema abre la ventana principal. Si no son validos los datos el sistema despliega un mensaje de error.

Tabla N° 3.4 CU1 Inicia Sesión

# Caso Uso	CU2	
Nombre	Gestión de Usuarios	
Propósito	El Administrador crea, modifica, actualiza, elimina usuarios.	
Actores	Administrador	
Precondiciones	El administrador debe haber creado un perfil de usuario.	
Poscondiciones	Usuario creado, modificado, eliminado.	
Flujo	Pasos	Acción
Principal	1	El sistema muestra opción al administrador

	2	Escenario Crear usuario: El Administrador ingresa: *Cedula Nombre Apellido *Nick *Password Teléfono Celular Email *Perfil
	3	El sistema valida que estén llenos todos los campos que están con * ya que son primordiales.
	4	Si están llenos el sistema guarda la información. Si no están llenos el sistema presenta un mensaje solicitando ingresar esos datos.
	5	Escenario Modificar. El administrador selecciona al usuario y la opción editar.
	6	El sistema despliega los datos del usuario.
	7	El Administrador modifica los datos y guarda.
	8	El sistema valida los datos.
	9	Si son correctos guarda los datos. Si no son correctos despliega un mensaje de error.
	10	Escenario Eliminar El Administrador selecciona al usuario y elige opción eliminar.

	11	El sistema elimina al usuario.
--	----	--------------------------------

Tabla N° 3.5 CU2 Gestión de Usuarios

# Caso Uso	CU3	
Nombre	Gestiona Perfiles de usuario	
Propósito	El Administrador crea, modifica, actualiza, elimina perfiles usuarios.	
Actores	Administrador	
Precondiciones	Para eliminar un perfil no debe haber usuarios asignados al perfil.	
Poscondiciones	Perfil creado, modificado, eliminado.	
Flujo	Pasos	Acción
Principal	1	El sistema muestra opción al administrador.
	2	Escenario Crear Perfil: El Administrador ingresa: Nombre del Perfil
	3	El sistema muestra menús y submenús que pueden ser asignados al perfil
	4	El Administrador selecciona.
	5	El sistema crea el nuevo perfil
	6	Escenario Modificar. El administrador selecciona la opción cambiar perfil.
	7	El sistema despliega los perfiles creados.
	8	El Administrador selecciona el perfil.
	9	El sistema muestra los menús y submenús.
	10	El administrador modifica el perfil.
	11	El sistema guarda los cambios.
	12	Escenario Eliminar El Administrador selecciona el perfil y elige opción eliminar.
	13	El sistema elimina al usuario.

Tabla N° 3.6 CU3 Perfil de Usuarios

# Caso Uso	CU4	
Nombre	Gestiona Horarios	
Propósito	El Administrador crea, modifica, actualiza, elimina perfiles horarios.	
Actores	Administrador	
Precondiciones	Para eliminar un perfil de horario no debe haber usuarios asignados al horario.	
Poscondiciones	Perfil de horario creado, modificado, eliminado.	
Flujo	Pasos	Acción
Principal	1	El sistema muestra opción al administrador.
	2	Escenario Crear Horario: El Administrador ingresa: Nombre del Horario Hora entrada Hora de salida Días de la semana.
	3	El sistema crea el perfil de horario.
	4	Escenario Modificar. El administrador selecciona el nombre del perfil de horario.
	5	El sistema despliega los datos del perfil de horario.
	6	El Administrador cambia el nombre, la hora de entrada, hora de salida o los días de la semana.
	7	El sistema guarda los cambios.
	8	Escenario Eliminar El Administrador selecciona el perfil y elige opción eliminar.
	9	El sistema elimina el perfil de horario.

Tabla N° 3.7 CU4 Gestiona Horarios

# Caso Uso	CU5
Nombre	Gestiona Departamentos
Propósito	El Administrador crea, activa, elimina departamentos.
Actores	Administrador
Precondiciones	Para eliminar un departamento no debe haber empleados asignados al departamento.

Poscondiciones	Departamento creado, modificado, eliminado.	
Flujo	Pasos	Acción
Principal	1	El sistema muestra opción al administrador.
	2	Escenario Crear Departamento: El Administrador ingresa: Nombre del Departamento. Selecciona si quiere activar departamento.
	3	El sistema crea y activa el departamento.
	4	Escenario Modificar. El administrador activa o desactiva al departamento.
	5	El sistema guarda los cambios.
	6	Escenario Eliminar El Administrador borra el departamento y guarda el cambio
	7	El sistema elimina el departamento.

Tabla N° 3.8 CU5 Gestiona Departamentos

# Caso Uso	CU6	
Nombre	Gestiona Personal	
Propósito	El Administrador crea, edita, elimina personal.	
Actores	Administrador Usuario Secundario	
Precondiciones		
Poscondiciones	Personal creado, modificado, eliminado.	
Flujo	Pasos	Acción
Principal	1	El sistema muestra opción al administrador.
	2	Escenario Crear Personal: El Administrador ingresa datos básicos: Nombre. Apellido. Cedula. Código. Fecha de nacimiento Fecha de ingreso El administrador ingresa datos opcionales Dirección, teléfono, email, observación.

	3	El sistema valida que este ingresado los datos básicos. Si el código y cedula no se repiten el sistema guarda los datos. Si se repiten el sistema muestra un mensaje de error.
	4	Escenario Modificar. El Administrador selecciona el Personal y el dato a modificar. Modifica y manda a guardar.
	5	El sistema valida los datos y guarda los cambios si es que no hay ningún error. Si en la validación hay algún error el sistema muestra un mensaje de error.
	6	Escenario Eliminar El Administrador selecciona el Personal y la opción eliminar.
	7	El sistema elimina el Personal seleccionado.

Tabla N° 3.9 CU6 Gestiona Personal

# Caso Uso	CU7	
Nombre	Gestiona Sistema	
Propósito	El Administrador escoge puerto de conexión, configura conexión a BDD, configura formato fecha y hora, asigna horario a personal, asigna departamento al personal.	
Actores	Administrador	
Precondiciones	Tener creado perfiles de horario y departamentos.	
Poscondiciones	Puerto de conexión con el hardware seleccionado, configuración de BDD realizada, configuración de formato fecha y hora realizada, departamento y horario asignados.	
Flujo	Pasos	Acción
Principal	1	El sistema muestra opción al administrador.
	2	Escenario seleccionar puerto: El sistema muestra los puertos Coque dispone el equipo para la comunicación serial.
	3	El Administrador escoge el puerto.
	4	El sistema guarda dato del puerto escogido.

	5	Escenario configurar conexión BDD. El administrador ingresa: Nombre de usuario de acceso a la BDD. Password de ingreso a la BDD. Nombre de la BDD a conectar.
	6	El sistema guarda los datos ingresados.
	7	Escenario configurar formato fecha y hora. El Administrador escoge el formato para representar los datos para fecha y hora.
	8	El sistema guarda los datos configurados.
	9	Escenario asignar horario y departamento a Personal. El Administrador selecciona al Personal y asigna horario y departamento.
	10	El sistema guarda la configuración.

Tabla N° 3.10 CU7 Gestiona Sistema

# Caso Uso	CU8	
Nombre	Gestiona Faltas y permisos	
Propósito	El Administrador justifica faltas, asigna permisos.	
Actores	Administrador	
Precondiciones		
Poscondiciones	Falta justificada, permiso concebido	
Flujo	Pasos	Acción
Principal	1	El sistema muestra opción al administrador.
	2	Escenario justificar falta: El Administrador ingresa: Periodo de fecha donde ocurrió la falta. Selecciona consultar
	3	El sistema muestra los días con faltas.
	4	El Administrador ingresa: Numero de referencia de documento de respaldo de la justificación. Observación. Selecciona justificar.
	5	El sistema muestra los días justificados.
	6	Escenario Asignar Permisos.

		El administrador ingresa: # Referencia. Observación. Periodo de fecha de permiso.
	7	El sistema asigna permisos.

Tabla N° 3.11 CU8 Gestiona Faltas y permisos

# Caso Uso	CU9	
Nombre	Gestiona Vacaciones, Feriados	
Propósito	El Administrador asigna vacaciones, y selecciona días feriados.	
Actores	Administrador	
Precondiciones		
Poscondiciones	Vacaciones de personal , días feriados en el año seleccionados	
Flujo	Pasos	Acción
Principal	1	El sistema muestra opción al administrador.
	2	Escenario asignar vacaciones: El Administrador selecciona personal Ingresa periodo de vacación escogido por personal
	3	El sistema guarda los datos.
	4	Escenario seleccionar días feriados: El sistema muestra un calendario.
	5	El administrador escoge los días feriados.
	6	El sistema guarda los datos.

Tabla N° 3.12 CU9 Gestiona Vacaciones, Feriados

# Caso Uso	CU10	
Nombre	Gestiona Reportes	
Propósito	El sistema emite un reporte de horas laboradas, horarios del personal, datos del personal.	
Actores	Administrador Usuario Secundario.	
Precondiciones	Tener un registro de horas laboradas.	
Poscondiciones	Reporte de horas laboradas, horarios del personal, datos del personal. Impresión de reportes.	

Flujo	Pasos	Acción
Principal	1	El sistema muestra opción al administrador.
	2	Escenario reporte horas laboradas: El Administrador ingresa periodo de fechas que quiere consultar.
	3	El sistema despliega un reporte de las horas laboradas.
	4	Escenario reporte horario personal: El sistema muestra un reporte del personal y el horario asignado.
	5	Escenario reporte datos básicos personal: El sistema muestra un reporte de los datos básicos del personal.

Tabla N° 3.13 CU10 Gestiona Reportes

# Caso Uso	CU11	
Nombre	Registra entrada, salida	
Propósito	Usar el hardware para registrar las horas de entrada y salidas que efectúa el personal durante el día laboral	
Actores	Empleado	
Precondiciones	Tener conectado el hardware y configurado el puerto de comunicación serial, estar creado el personal.	
Poscondiciones	Registro de horas de entrada y salida.	
Flujo	Pasos	Acción
Principal	1	El personal digita código
	2	El sistema valida código en la BDD.
	3	El sistema despliega nombre del Personal
	4	El personal valida su nombre. El personal escoge opción entrada o salida o cancelar.
	5	Si es entrada o salida el sistema guarda un registro de la hora y de la opción seleccionada. Si la opción es cancelar el sistema no guarda nada.

Tabla N° 3.14 CU14 Registro de horas de entrada y salida.

3.2.2 MODELO DE LA BASE DE DATOS.

En la figura 3.21, se encuentra representado el modelo físico de la Base de Datos la cual consta de 9 tablas que son registro, empleado, empresa, perfil, departamento, usuario, perfil horario, parámetros y feriado relacionadas entre si

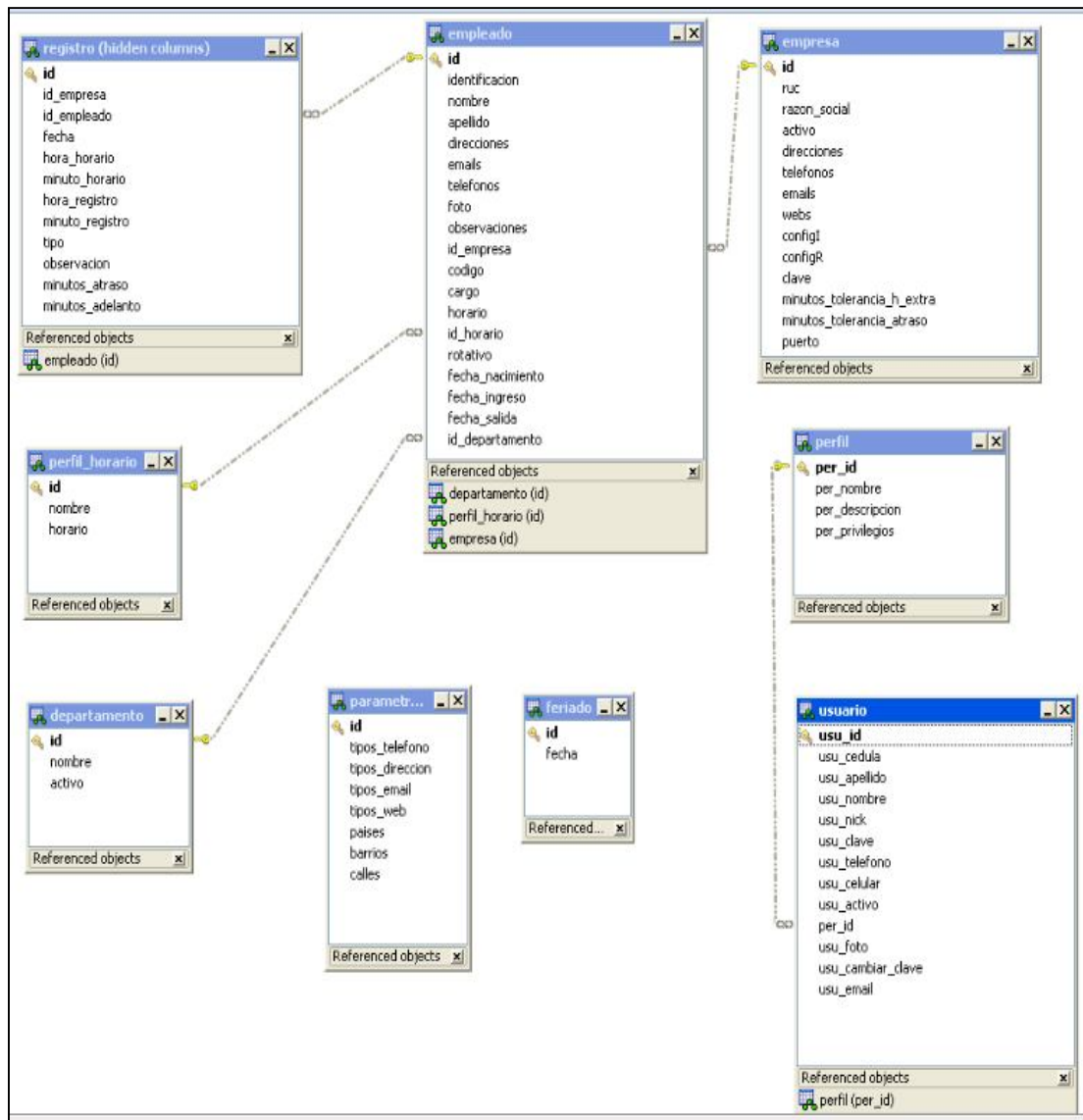


Fig.3.21 Modelo Base de Datos

Como se observa en la figura 3.21, la tabla empleado está relacionada con las tablas registro, departamento, perfil de horario y empresa mientras que la tabla perfil solo tiene relación con la tabla usuario, en el diccionario de datos podemos encontrar un detalle de cada tabla.

3.2.2.1 Diccionario de datos

A continuación, se detalla las tablas usadas con sus respectivos campos la descripción de los mismos, sus longitudes y si son o no campos requeridos.

TABLA	Departamento			
Descripción:	Contiene los datos de los departamentos existentes en la empresa			
NOMBRE	TIPO	DESCRIPCION	LONGITUD	REQUERIDO
<u>id</u>	int	Identificador único	11	S
nombre	string	Nombre del departamento	250	S
activo	bool	Indica si el objeto está o no visible al sistema		S

Tabla N 3.15 Tabla Departamento

TABLA	Empleado			
Descripción:	Contiene los datos de los empleados de la empresa			
NOMBRE	TIPO	DESCRIPCION	LONGITUD	REQUERIDO
<u>id</u>	long	Identificador único	11	S
identificacion	string	Número de identificación, p.ej.: cédula de identidad	25	S
nombre	string	Nombre del empleado	150	S
apellido	string	Apellido del empleado	150	S
direcciones	object	Listado de direcciones donde ubicar al empleado		N
emails	object	Listado de direcciones de correo		N
telefonos	object	Listado de teléfonos donde ubicar al empleado		N
foto	object	Foto del empleado		N

observaciones	object	Observaciones sobre el empleado		N
id_empresa	int	Identificador de la empresa a la que pertenece	11	S
codigo	string	Código de acceso al sistema para registrar asistencias	10	S
cargo	string	Cargo del empleado en la empresa	150	N
horario	object	Horario asignado al empleado. Solo si tiene horario personalizado		S
id_horario	int	Identificador del perfil de horario asignado	11	S
rotativo	bool	Indica si el empleado toma el horario personalizado o el horario del perfil		S
fecha_nacimiento	date	Fecha de nacimiento del empleado		S
fecha_ingreso	date	Fecha en la que ingresó a la empresa		N
fecha_salida	date	Fecha en la que sale de la empresa		N
id_departamento	int	Identificador del departamento al que pertenece el empleado	11	S

Tabla N 3.16 Tabla Empleado

TABLA	Empresa			
Descripción:	Contiene los datos de la empresa que utiliza el sistema			
NOMBRE	TIPO	DESCRIPCION	LONGITUD	REQUERIDO
<u>id</u>	int	Identificador único	11	S
ruc	string	Registro único de contribuyente de la empresa	15	S
razon_social	string	Razón social de la empresa	250	S
activo	bool	Indica si la empresa es visible al sistema		S

direcciones	object	Listado de direcciones donde ubicar a la empresa		N
telefonos	object	Listado de direcciones de correo		N
emails	object	Listado de teléfonos donde ubicar a la empresa		N
webs	object	Listado de direcciones web donde ubicar a la empresa		N
configl	object	Logo de la empresa para la aplicación		N
configR	object	Logo de la empresa para los reportes		N
clave	string	Clave de acceso para la empresa	200	S
minutos_tolerancia_h_extra	int	Número de minutos a partir de los cuales se cuentan las horas extras	3	S
minutos_tolerancia_atraso	int	Número de minutos a partir de los cuales se considera un atraso	3	S
puerto	string	Puerto de conexión del lector	10	S

Tabla N 3.17 Tabla Empresa

TABLA	Feriado			
Descripción:	Contiene las fechas en la que los días son feriados			
NOMBRE	TIPO	DESCRIPCION	LONGITUD	REQUERIDO
<u>id</u>	int	Identificador único	11	S
fecha	date	Fecha marcada como feriado		S

Tabla N 3.18 Tabla Feriado

TABLA	Perfil			
Descripción:	Contiene los datos de los perfiles de acceso al sistema			
NOMBRE	TIPO	DESCRIPCION	LONGITUD	REQUERIDO
<u>per_id</u>	int	Identificador único	11	S
per_nombre	string	Nombre del perfil	150	S
per_descripcion	string	Descripción del alcance del perfil	150	N

per_privilegios	object	Accesos del perfil en el sistema		S
------------------------	--------	----------------------------------	--	---

Tabla N 3.19 Tabla Perfil

TABLA	Perfil_Horario			
Descripción:	Contiene las plantillas de horas que se asignan a los empleados			
NOMBRE	TIPO	DESCRIPCION	LONGITUD	REQUERIDO
<u>id</u>	int	Identificador único	11	S
nombre	string	Nombre de la plantilla	150	S
horario	string	Horario, en días y horas		S

Tabla N 3.20 Tabla Perfil Horario

TABLA	Registro			
Descripción:	Contiene los datos de los registros de asistencia de los empleados			
NOMBRE	TIPO	DESCRIPCION	LONGITUD	REQUERIDO
<u>id</u>	int	Identificador único	11	S
id_empresa	string	Identificador de la empresa a la que pertenece	11	S
id_empleado	string	Identificador del empleado que realiza el registro	11	S
fecha	date	Fecha en la que se realiza el registro		S
hora_horario	int	Hora del horario asignado al empleado	2	N
minuto_horario	int	Minuto del horario asignado al empleado	2	N
hora_registro	int	Hora en la que se realiza el registro	2	S
minuto_registro	int	Minuto en la que se realiza el registro	2	S
tipo	string	Indica si es entrada o salida: in, out	20	S
observacion	string	Observacion sobre el registro	250	N

minutos_atraso	int	Minutos de retraso con respecto al horario del empleado	2	N
minutos_adelanto	int	Minutos de adelanto con respecto al horario del empleado	2	N
id_usuario_registra	int	Identificador del usuario que realiza el registro	11	N
diferencia	string	Diferencia, en palabras, de la hora de registro con la hora del horario	500	N
tipo_diferencia	string	Indica si el registro está en función de la entrada o salida del horario	500	N
accion	string	Indica el tipo de registro realizado: normal, justificacion, feriado, permiso, vacación, cancelado	100	S
fecha_accion	date	Fecha en la que se realiza un registro distinto a normal		N
referencia	string	Referencia sobre el tipo de registro, p.ej.: si es falta el número de documento anexo	350	N
id_usuario_accion	int	Identificador del usuario que realiza el registro si es distinto a normal	11	N

Tabla N 3.21 Tabla Registro

TABLA	Usuario			
Descripción:	Contiene los datos de los usuario que utilizan el sistema			
NOMBRE	TIPO	DESCRIPCION	LONGITUD	REQUERIDO
<u>usu_id</u>	int	Identificador único	11	S
usu_cedula	string	Número de identificación, p.ej.: cédula de identidad	50	S
usu_apellido	string	Apellido del empleado	100	S
usu_nombre	string	Nombre del empleado	100	S
usu_nick	string	Nombre de usuario con el que ingresa al sistema	30	S

usu_clave	string	Contraseña de acceso al sistema		S
usu_telefono	string	Número de teléfono del usuario	15	N
usu_celular	string	Número de celular del usuario	15	N
usu_activo	bool	Indica si el usuario es visible al sistema		S
per_id	int	Identificador del perfil de acceso asignado	11	S
usu_foto	object	Foto del usuario		N
usu_cambiar_clave	bool	Indica si el usuario debe cambiar la clave cuando inicie sesión		N
usu_email	string	Dirección de correo electrónico del usuario	250	N

Tabla N°3.22 Tabla Usuario

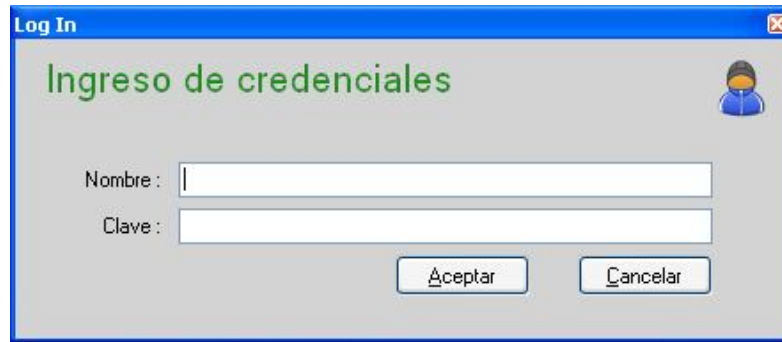
3.2.3 INTERFAZ DE USUARIOS.

A continuación, se detalla las ventanas principales de la aplicación, como en muchos menús se repite el mismo caso se tomará una muestra, más no se detallarán todas las interfaces.

Para ver todas las interfaces realizadas se puede direccionar a la sección manual de usuario del software al final del documento.

3.2.3.1 Interfaz de inicio de sesión

Es la primera ventana que aparece al ejecutar la aplicación. En esta se debe ingresar el nombre y clave para ingresar al sistema.

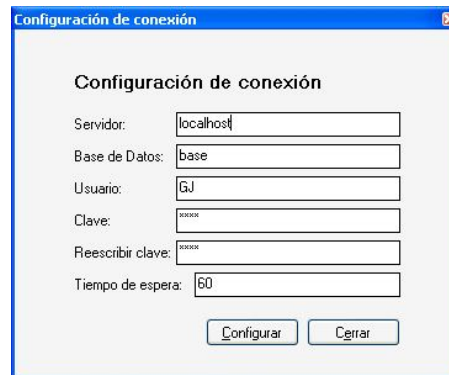


The image shows a 'Log In' window with a blue title bar. The main area has a light gray background. At the top left, the text 'Ingreso de credenciales' is displayed in green. To the right of this text is a small icon of a person. Below the text, there are two input fields: 'Nombre :' and 'Clave :'. At the bottom right, there are two buttons: 'Aceptar' and 'Cancelar'.

Figura 3.22 Inicio de sesión

3.2.3.2 Interfaz de Conexión

Esta interfaz permite configurar la conexión a la BDD.



The image shows a 'Configuración de conexión' window with a blue title bar. The main area has a light gray background. At the top, the text 'Configuración de conexión' is displayed in bold. Below this text, there are six input fields: 'Servidor:', 'Base de Datos:', 'Usuario:', 'Clave:', 'Reescribir clave:', and 'Tiempo de espera:'. At the bottom right, there are two buttons: 'Configurar' and 'Cerrar'.

Figura.3.23 Configuración conexión a BDD

3.2.3.3 Ventana Principal.

Esta interfaz es la ventana principal como muestra la figura 3.24, contiene el menú principal que se encuentra en la parte superior, mediante el cual se abren las vistas para gestionar usuarios, gestionar empleados, gestionar reportes, y administrar el sistema.

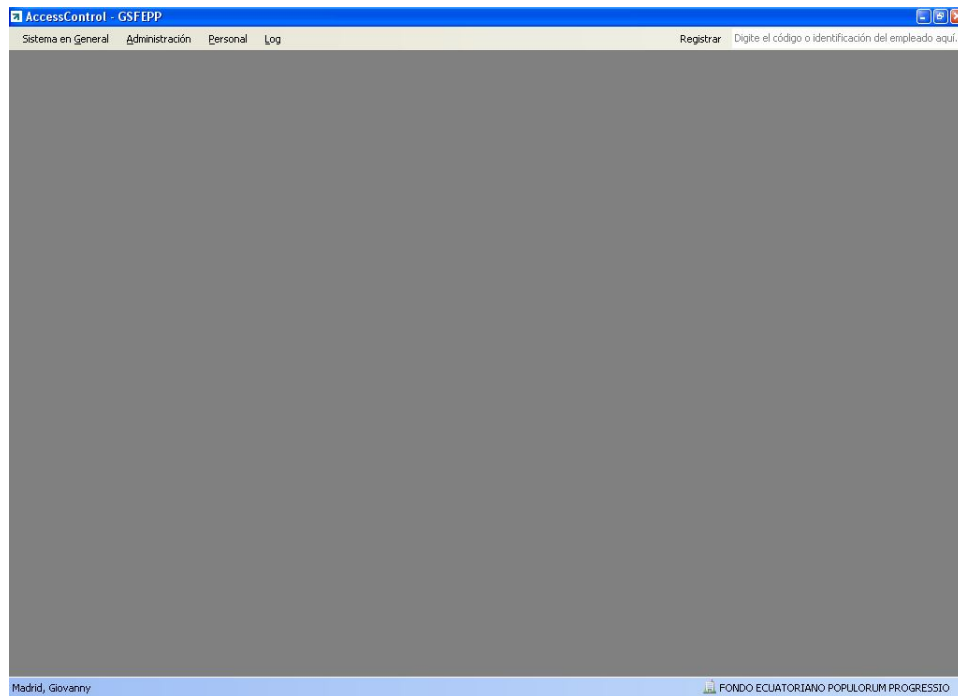


Figura.3.24 Ventana Principal

3.2.4 MOTOR DE CONEXIÓN HARDWARE – BASE DE DATOS.

El motor de conexión Hardware – Base de datos es un mini software que sirve de interfaz para la comunicación entre el hardware y la base de datos. Utiliza una librería que permite lo siguiente.

- Abrir puerto serial
- Cerrar el puerto serial

Para la implementación de la mini aplicación se ha tomado como referencia los comandos detallados en los requerimientos del hardware en el capítulo 2.

3.2.5 ESTRUCTURA DE LA CLASE CONEXIÓN

La clase conexión esta básicamente estructurada como muestra la figura 3.23.

Conexion
-codigo : String
-comando : Char
-sentencia : String
-nombre : String
+conectar()
+desconectar()
+validar()
+envio()
+recepcion()

Tabla N° 3.23 Conexión

Esta es la que permite la conexión entre el hardware y software por medio de la interfaz serial.

Además se encarga de interactuar con la BDD validando el código ingresado por el usuario para el registro de entrada o salida.

CAPITULO IV

PRUEBAS DEL HARDWARE Y SOFTWARE.

4.1 PRUEBAS DE HARDWARE.

4.1.1 PRUEBAS EN PROTEUS.

A continuación, se describen las pruebas realizadas en Proteus en modo simulado.

4.1.1.1 Simulación de Encendido del dispositivo

Al encender el dispositivo sale el mensaje predeterminado programado como se observa en el modulo LCD en la Figura 4.1.

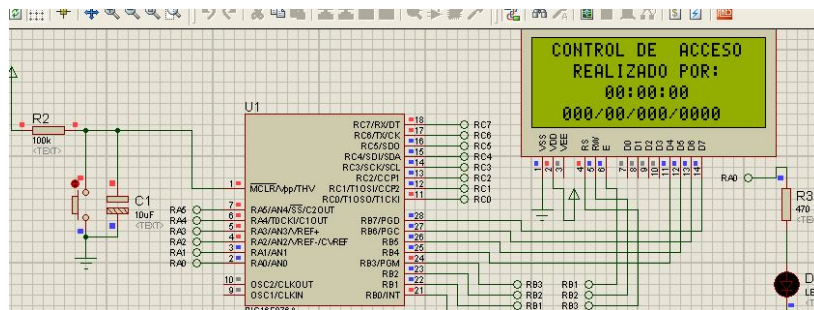


Fig. 4.1. Encendido del Dispositivo en Proteus

4.1.1.2 Simulación de la estabilización del PIC

Es el tiempo que se le da al PIC para que se estabilice, inicializa los formatos de hora, fecha y los despliega en el LCD como muestra la Figura 4.2.

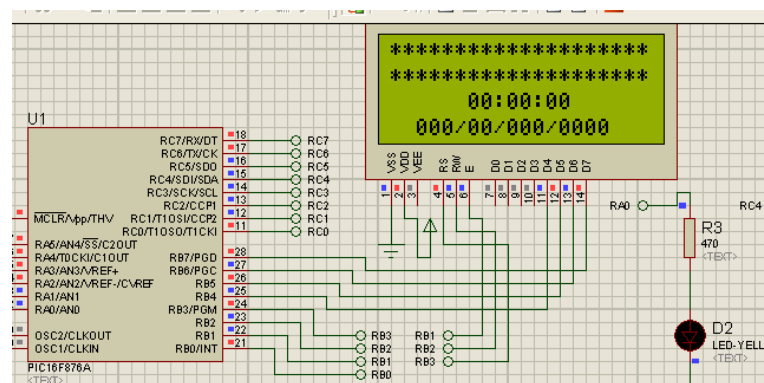


Fig.4.2 Simulación de estabilización del PIC en Proteus

4.1.1.3 Simulación de la comunicación

En este paso el PIC está esperando recibir el carácter de inicialización, enviado desde la PC al ejecutar el programa motor de conexión como muestra la figura 4.3.

Antes de este paso no se debe ejecutar el programa motor de conexión en el PC, porque se presentarán problemas de sincronización desfasándose todo el proceso de comunicación.

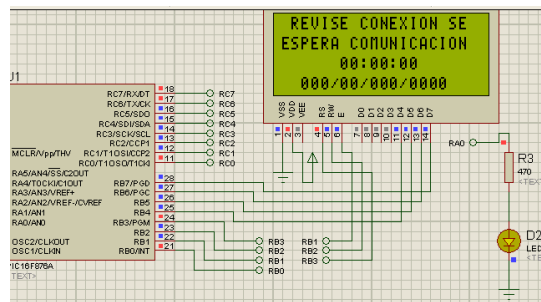


Fig.4.3 Esperando comunicación

4.1.1.4 Simulación de la comunicación

El PIC recibe el carácter “A” desde la PC y cambia el mensaje como se muestra en el modulo LCD en la Figura 4.4. La simulación de envió del carácter desde la PC se hace en el hyperterminal simulado por Proteus.

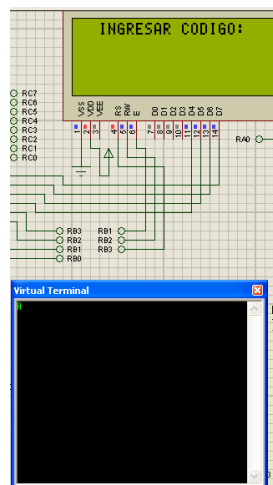


Fig.4.4 Inicialización de Comunicación

4.1.1.5 Recibiendo datos de Hora y Fecha (simulado)

El PIC recibe el carácter “Z” seguido de los caracteres de la hora y fecha del sistema operativo. En total son 15 caracteres recibidos uno seguido del otro hasta formar la siguiente cadena Z19375655261110. Como se muestra en la Figura 4.5 de la izquierda, están todos los datos de la cadena en el formato programado para visualizar la hora y fecha, a excepción del carácter de control “Z” como en la figura 4.5 de la derecha.

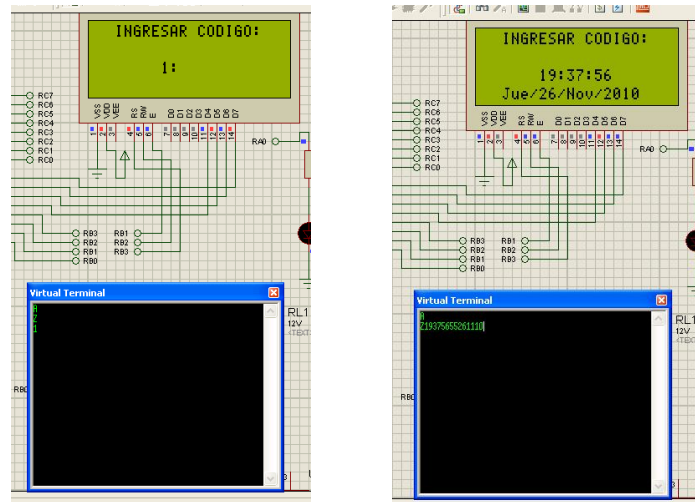


Fig.4.5 Recepción de datos de fecha y Hora

4.1.1.6 Ingresando Código del empleado (simulado)

El empleado ingresa un código de 4 dígitos, luego debe pulsar enter en el caso de Proteus el símbolo (=). Si el empleado pulsa el símbolo (=) antes de haber ingresado los 4 dígitos el LCD desplegará un mensaje de aviso que debe ingresar 4 dígitos, como el que se muestra en la Figura 4.6 de la derecha.

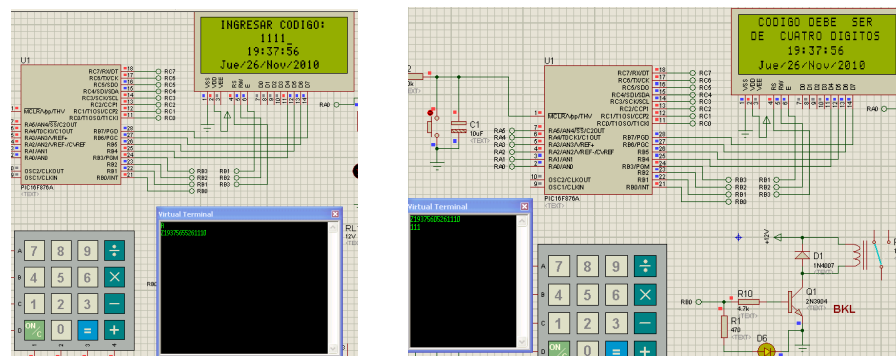


Fig.4.6 Ingreso de código de empleado

Si el usuario a digitado los 4 números de su código (1111) y pulsado la tecla enter (=), los datos serán enviados al PC y en el LCD se presentará el mensaje como se muestra en la Figura 4.7.

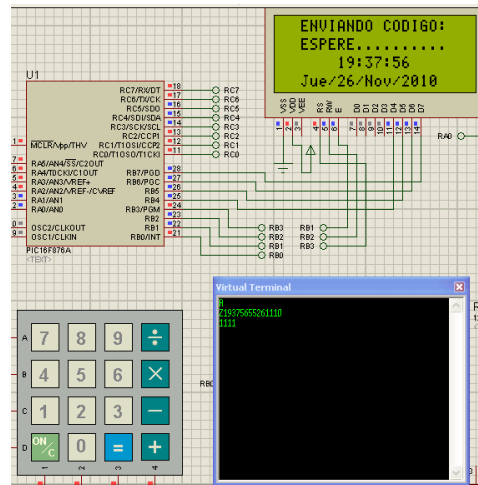


Fig.4.7 Enviando datos al PC

4.1.1.7 Recibiendo Datos del PC (simulado)

El PC envía el nombre del usuario correspondiente al código digitado en el siguiente formato BGIOVANNY MADRID PONC, primero el carácter de control “B” y luego los 20 caracteres siguientes como muestra el LCD en la Figura 4.8 de la izquierda. Si el código no se ha encontrado en la base de datos, en vez del nombre se enviará un mensaje de CODIGO INVALIDO como se muestra en la Figura 4.8 de la derecha.

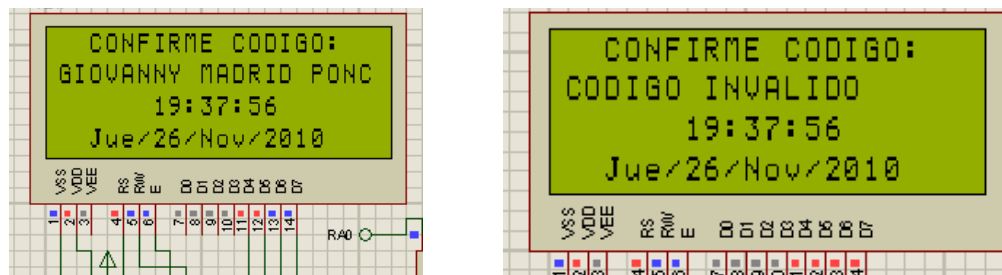


Fig.4.8 Recibiendo datos del PC

4.1.1.8 Simulación del menú de opciones

Luego de desplegar el nombre en el LCD, pasan aproximadamente 3 segundos y cambia al menú de opciones como muestra la Figura 4.9, donde el usuario debe seleccionar una opción para terminar con el registro.

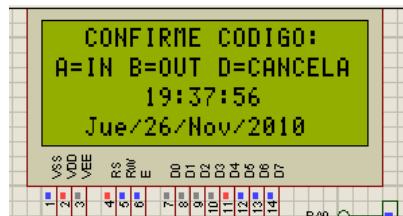


Fig.4.9 Menú de opciones

Inmediatamente que el usuario selecciona la opción el PIC manda el código seleccionado en este caso "000E" de entrada. Se hicieron las pruebas con las demás opciones obteniendo resultados satisfactorios.

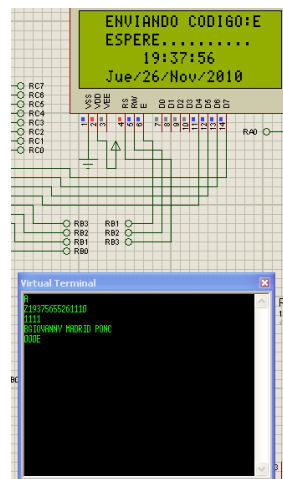


Fig.4.10 Enviando código de opción seleccionada

4.1.1.9 Simulación fin de la comunicación (simulado)

El PC recibe el código y manda el carácter "C" que indicando que terminada la comunicación, el PIC recibe el carácter y regresa al estado de pedir código.



Fig.4.11 Fin de comunicación

4.1.2 PRUEBAS EN EL HARDWARE.

Al realizar las pruebas con la conexión a la BDD se observó que se debía hacer algunos ajustes pero solo a nivel del software, ya que la simulación en Proteus fue de gran ayuda para descartar problemas del hardware.

Dichos problemas eran de sincronización pero a la final se logro sincronizar y que el dispositivo funcione en las condiciones deseadas.

A continuación, se muestra las imágenes resultantes de las pruebas finales con el hardware implementado y conectado a la BDD.

4.1.2.1 Encendido del dispositivo.



En la figura 4.12, se muestra el mensaje predeterminado programado al encender el dispositivo.

Fig.4.12 Encendido de dispositivo

4.1.2.2 Estabilización



En la figura 4.13, se muestra en el LCD el formato de la hora y fecha predeterminados.

Fig. 4.13 Estabilización del PIC

4.1.2.3 Esperando comunicación



Fig. 4.14 Esperando comunicación

En la figura 4.14, se muestra al dispositivo ya inicializado y listo para recibir la comunicación desde el PC.

En este momento el usuario ya puede ejecutar el programa motor de conexión en el PC para comenzar la comunicación.

4.1.2.4 Inicialización de la comunicación



Fig.4.15 Inicialización de la comunicación

En la figura 4.15, se muestra en el LCD los datos de la hora y fecha esto indica que el PC y el PIC han entablado comunicación y se están transfiriendo y actualizando los datos de la hora y fecha a cada momento.

4.1.2.5 Ingresando Código del empleado y Envío



En la figura 4.16, se muestra en el LCD el código digitado por el usuario y se dispone a digitar la tecla # para el envío de los datos hacia el PC.

Luego en la BDD se validará el código ingresado.

Fig.4.16 Ingreso Código y Envío

4.1.2.6 Recepción de Datos desde el PC



En la figura 4.17, se muestra el nombre usuario devuelto por la BDD correspondiente al código 1111 ingresado anteriormente.

El nombre se presenta en el LCD por 3 segundos aproximadamente y luego cambia al menú de opciones.

Fig. 4.17 Recepción de datos desde el PIC

4.1.2.7 Menú



Fig. 4.18 Menú

En la figura 4.18, se muestra el menú de opciones, el usuario debe seleccionar una de las presentadas para finalizar el proceso de registro.

4.1.2.8 Selección



Fig. 4.19. Selección de Opción

En la figura 4.19, se muestra al usuario seleccionando la opción OUT pulsando la tecla "B".

En el LCD se observa que se envía el carácter "S" para que el programa motor en el PC, sepa que es una salida y se registre en la BDD.

4.2 PRUEBAS DEL SOFTWARE.

Para realizar las pruebas del software se trató de seguir la metodología de Pruebas Orientada a Objetos para el Ciclo de Vida Completo FLOOT("Full Life-Cycle Object-Oriented Testing") dado que este ofrece una serie de técnicas para encontrar errores en el software.

Las técnicas que más semejanza tienen a lo realizado para encontrar los errores se describen a continuación:

- **Prueba de Caja-Negra**

Se verificó el sistema dando las entradas apropiadas y esperando que produzcan los resultados esperados.

- **Prueba de Valores-Frontera**

En este caso se daba una entrada errónea o inusual y el sistema debía ser capaz de manejarlas.

- **Prueba de Componente**

Consiste en validar que un componente funcione como está definido.

- **Prueba de Integración**

Consiste en realizar pruebas para verificar que un gran conjunto de partes del software funcionan juntas.

En este caso fue útil al realizar la integración del software de conexión de la BDD y el motor de la comunicación serial para verificar los errores.

- **Prueba de Interfaz de Usuario**

Para esta prueba fue necesaria la ayuda del usuario administrador ya que fue el encargado de probar las interfaces gráficas verificando que se cumplan los requerimientos definidos.

En la siguiente sección, se describe las pruebas realizadas al software con los posibles fallos que el usuario pueda tener al querer realizar una acción.

4.2.1 VALIDACIÓN DE INICIO DE SESIÓN

Si el usuario ingresa mal el nombre o clave se desplegará un mensaje de error.

Figura 4.20



Fig. 4.20

4.2.2 VALIDACIÓN DE CONEXIÓN CON LA BDD

- Si el usuario ingresa mal el nombre de la base datos al configurar la conexión le desplegará el siguiente mensaje de error. Figura N 4.21

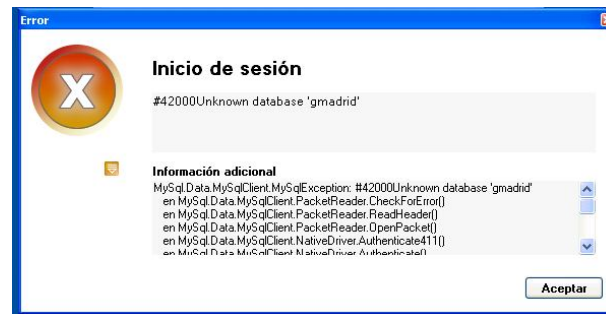


Fig. 4.21

- Si el usuario a ingresado mal el nombre o password para conectarse con la BDD se desplegará el siguiente mensaje de error. Figura N 4.22



Fig.4.22

4.2.3 VALIDACIÓN CREACIÓN DE USUARIO.

- Si el administrador al crear un usuario repite el número de identificación, Nick, clave se desplegará el siguiente mensaje de error. Figura N 4.23

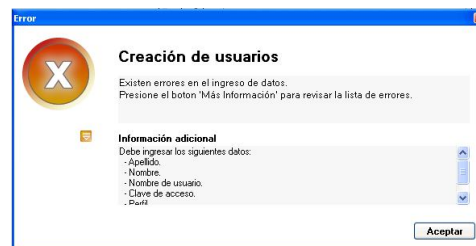


Fig.4.23

- Si el administrador se olvida de seleccionar el perfil para crear un usuario se desplegará el siguiente mensaje de error. Figura N 4.24

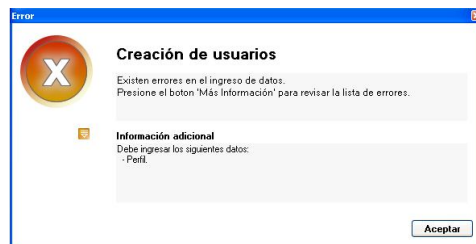


Fig.4.24

- Si el administrador confirma mal la clave del usuario a crear se desplegará el siguiente mensaje de error. Figura N 4.25.

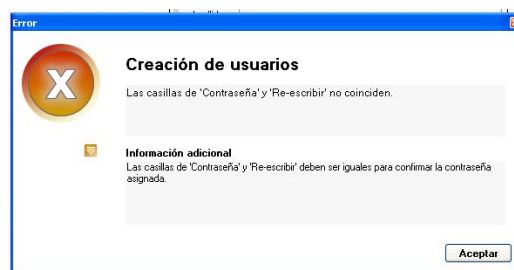


Fig.4.25

4.2.4 VALIDACIÓN CREACIÓN DE PERSONAL.

Si el usuario ingresa 2 veces el mismo código o el mismo número de cédula se desplegará el siguiente mensaje de error. Figura N 4.26



Fig.4.26

4.2.5 VALIDACIÓN DE PERMISOS, JUSTIFICACIONES Y VACACIONES.

Si el usuario se olvida de ingresar un número de referencia se desplegará el siguiente mensaje de error. Figura N 4.27



Fig.4.27

4.2.6 PRUEBAS DE IMPRESIÓN DE REPORTES.

A continuación, se muestran los reportes obtenidos como prueba de funcionamiento del software y cumplimiento de los objetivos propuestos.

4.2.6.1 Reporte de Datos del personal

En la figura 4.28, se muestra un listado del personal con los datos personales de cada uno.

FONDO ECUATORIANO POPULORUM PROGRESSIO

Mallorca N24-275 y Coruña
fepp@fepp.org.ec
2520408

Grupo Social fepp
FONDO ECUATORIANO POPULORUM PROGRESSIO

LISTADO DE EMPLEADOS

Id. : 171655337
Código : 1111
Nombre : MADRID, GIOVANNY
Cargo : Operaciones
Teléfono :
E-mail :
Dirección :

Id. : 1703797959
Código : 0101
Nombre : ANGULO, SAGRARIO
Cargo : Contadoras
Teléfono :

Nº total de páginas: 1 Factor de zoom: 100%

Fig.4.28

4.2.6.2 Reporte de horarios del personal

En la figura 4.29, se muestra los horarios establecidos para cada empleado.

FONDO ECUATORIANO POPULORUM PROGRESSIO

Mallorca N24-275 y Coruña
fepp@fepp.org.ec
2520408

Grupo Social fepp
FONDO ECUATORIANO POPULORUM PROGRESSIO

HORARIO DE EMPLEADOS

ANGULO, SAGRARIO
Contadoras

Hora	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo
08:00	Entrada > 08:00	Entrada > 08:00	Entrada > 08:00	Entrada > 08:00	Entrada > 08:00		
16:00	Salida > 16:30	Salida > 16:30	Salida > 16:30	Salida > 16:30	Salida > 16:30		

GUAÑA, ANDREA
Secretaria

Hora	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo
08:00	Entrada > 08:00	Entrada > 08:00	Entrada > 08:00	Entrada > 08:00	Entrada > 08:00		
16:00	Salida > 16:30	Salida > 16:30	Salida > 16:30	Salida > 16:30	Salida > 16:30		

MADRID, GIOVANNY
Operaciones

Fig.4.29

4.2.6.3 Reporte de horas laboradas por el personal.

En la figura 4.30, se muestra el reporte total de las horas laboradas, en este se encuentra agrupado las horas laboradas en días feriados, las faltas justificadas, el horario normal, y las vacaciones.

Mallorca N24-275 y Coruña
fepp@fepp.org.ec
2520408

Grupo Social fepp
FONDO ECUATORIANO POPULORUM PROGRESSIO

HORAS LABORADAS

Desde 2011-02-01 Hasta 2011-03-21

Madrid Giovanni

FERIADO

<i>Día In</i>	<i>Hora In</i>	<i>Día Out</i>	<i>Hora Out</i>	<i>Total</i>
07-Mar-2011	08:30	07-Mar-2011	14:30	6h, 0m

JUSTIFICACION_FALTA

<i>Día In</i>	<i>Hora In</i>	<i>Día Out</i>	<i>Hora Out</i>	<i>Total</i>
08-Mar-2011	08:00	08-Mar-2011	16:30	8h, 30m

NORMAL

<i>Día In</i>	<i>Hora In</i>	<i>Día Out</i>	<i>Hora Out</i>	<i>Total</i>
11-Mar-2011	08:22	11-Mar-2011	13:01	4h, 39m
11-Mar-2011	14:15	11-Mar-2011	17:16	3h, 1m
14-Mar-2011	08:50			0
14-Mar-2011	08:50	14-Mar-2011	13:15	4h, 25m
14-Mar-2011	14:00	14-Mar-2011	17:01	3h, 1m

VACACION

<i>Día In</i>	<i>Hora In</i>	<i>Día Out</i>	<i>Hora Out</i>	<i>Total</i>
09-Mar-2011	08:00	09-Mar-2011	16:30	8h, 30m
10-Mar-2011	08:00	10-Mar-2011	16:30	8h, 30m

Total : **46h, 36m**

Nº total de páginas: 1

Factor de zoom: 100%

Nº total de páginas: 1

Factor de zoom: 100%

Fig.4.30

4.3 ANÁLISIS COSTO – BENEFICIO.

El análisis de costo – beneficio permite ver de manera global los costos que ha tenido el proyecto y los beneficios que se han conseguido.

Como se indicó previamente en el inicio de esta tesis, la fundación dio el patrocinio de la misma, mas no se hizo cargo de los gastos por ser un proyecto prototipo.

Los gastos incurridos en el desarrollo de la tesis han sido solventados por parte del autor de la tesis por lo que también se tomó en cuenta el análisis para la construcción del hardware.

En los siguientes cuadros se muestra las relaciones costo – beneficio que ha criterio son las más relevantes.

4.3.1 Análisis costo – beneficio para la empresa en moneda local.

- Costo total del dispositivo \$40
- Inversión de la empresa \$0

Beneficio:

- Ahorro de tiempo en la generación de reportes
- Eliminación del proceso de digitación.
- Eliminación de errores.

Como se menciono anteriormente la empresa tiene un beneficio del 100% ya que la inversión fue \$0.

4.3.2 Análisis Costo – Beneficio para la empresa en tiempo.

Sin sistema		
Actividad	Tiempo (minutos /diarios)	% Error
Digitación Revisión	120	alta

Tabla N°4.1 Costo – beneficio sin sistema

Con sistema		
Actividad	Tiempo (minutos /diarios)	% Error
Consultar Datos	10	nulo

Tabla N°4.2 Costo – beneficio con sistema

Como se puede observar en la tabla N°4.1 se muestra el tiempo en lo que tarda una persona en digitar los datos para ser procesados y sacar un reporte, antes de implementar el sistema es de aproximadamente 120 minutos, así como la existencia de error en el proceso es alta.

En la tabla N°4.2 se muestra el tiempo con el sistema ya implementado, como se observa hay una disminución drástica del tiempo en el que una persona se demora en procesar los datos y obtener un reporte es de 10 minutos aproximadamente y al desaparecer el proceso de digitación el error es nulo.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- El diseño del prototipo se realizó tomando en cuenta el costo - beneficio, consiguiendo la implementación de un dispositivo que con la ayuda del software desarrollado, adquiera datos y al mismo tiempo interactúe con el usuario logrando registrar las horas laboradas en una BDD de una manera automática.
- Al realizar las pruebas en conjunto del hardware y software, surgió un problema de sincronización en la comunicación serial por lo que la presentación de los datos en el display se veía afectada. Se solucionó el problema incrementando el tiempo de espera para el envío de datos.
- Para guardar un registro de la fecha en la BDD se realizaron ajustes de formato en el software, ya que MySQL solo acepta el formato yyyy-mm-dd en un campo date.
- El cálculo de las horas laboradas depende de que los usuarios tengan emparejadas las horas de entradas y salidas en el registro.
- En la presentación e impresión del reporte de las horas laboradas, permisos, vacaciones, feriados, se demuestra el funcionamiento del hardware y software en conjunto y se da cumplimiento al objetivo primordial de automatizar el registro laboral de la fundación.

5.2 RECOMENDACIONES

- El prototipo mejoraría usando un RTC (real-time-clock) y con memorias que permitan el almacenamiento de registros de los usuarios, con lo que aseguraríamos que el prototipo funcione inclusive cuando no haya energía eléctrica.
- Una de las limitantes de trabajar con la transmisión serial RS232 es la distancia, se recomienda para proyectos futuros recalcar al cliente sobre la distancia de transmisión permitida.
- A futuro se debería implementar en el portal web de la fundación un módulo donde los empleados puedan consultar el estado de los registros y días laborados así como los días de vacaciones.
- Se puede migrar el hardware o adicionar un dispositivo biométrico, tiene su costo pero si uno de los requerimientos es la seguridad esta sería una de las soluciones a plantear.

GLOSARIO

DEFINICIONES, ACRONIMOS Y ABREVIACIONES.

Esta sección contiene las definiciones de términos, acrónimos y abreviaturas que tiene un significado técnico especial o que son únicos en cuanto al contenido.

ALU.- es un circuito digital que calcula operaciones aritméticas (como suma, resta, multiplicación, etc.) y operaciones lógicas (si, y, o, no), entre dos números.

ARES.- software que viene en el paquete Proteus para realizar el enrutamiento de los circuitos.

CMOS.- Tecnología de fabricación de procesadores que requiere un mínimo consumo eléctrico.

Codificación.- es un proceso a través del cual el emisor "prepara" el mensaje para que pueda ser comprendido por el sujeto. Para la preparación o codificación del mensaje se vale de sus conocimientos relacionados con las características del canal que va a emplear, las exigencias del público al que va a dirigirse, del lenguaje que va a utilizar, como de la cultura de quien lo va a recibir.

Decodificación.- es un proceso inverso de la codificación, en el cual el sujeto que recibe la información procede a "entender" el mensaje que ha recibido, lo descifra.

EEPROM o EAROM es una memoria de solo lectura reprogramable eléctricamente sin necesidad de extraerlas de la tarjeta del circuito.

ISIS.- software que viene en el paquete Proteus para el diseño y simulación de circuitos.

Microcontrolador.- es un circuito integrado o chip programable que incluye en su interior las tres unidades funcionales de una computadora: unidad central de procesamiento, memoria y unidades de E/S (entrada/salida).

Microprocesador.- es el cerebro del ordenador. Se encarga de realizar todas las operaciones de cálculo y de controlar lo que pasa en el ordenador recibiendo información y dando órdenes para que los demás elementos trabajen. Es el jefe del equipo y, a diferencia de otros jefes, es el que más trabaja.

MySql.- es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.

OTP.- Memoria no volátil de sólo lectura "programable una sola vez" por el usuario que contiene el microcontrolador.

Procesador.- es por decirlo de alguna manera, el cerebro del ordenador. Permite el procesamiento de información numérica, es decir, información ingresada en formato binario, así como la ejecución de instrucciones almacenadas en la memoria.

Registros.- En microprocesadores, un registro es una porción de memoria ultrarrápida, de poca capacidad e integrado al microprocesador, que permite almacenar y acceder datos usados frecuentemente. Se emplean para aumentar la velocidad de ejecución de los programas.

RISC.- Se trata de un tipo de procesador especialmente rápido que utiliza una tecnología del tipo pipeline muy desarrollada.

RS232.- norma creada por EIA, la cual define la interfaz mecánica, los pines, las señales y protocolos que debe cumplir la comunicación serial.

Transmisiones asíncronas.- son aquellas en que los bits que constituyen el código de un carácter se emiten con la ayuda de impulsos suplementarios que permiten mantener en sincronismo los dos extremos.

Transmisiones síncronas.- los caracteres se transmiten consecutivamente, no existiendo ni bit de inicio ni bit de parada entre los caracteres, estando dividida la corriente de caracteres en bloques, enviándose una secuencia de sincronización al inicio de cada bloque.

UART.- Es un chip cuya misión principal es convertir los datos recibidos del bus del PC en formato paralelo, a un formato serie que será utilizado en la transmisión hacia el exterior. También realiza el proceso contrario: transformar los datos serie recibidos del exterior en un formato paralelo entendible por el bus.

USART.- Se trata de un periférico para la transmisión de datos en formato serie, utilizando técnicas de transmisión sincrónica o asíncrona, según se configure el periférico.

ACRONIMOS Y ABREVIACIONES.

BDD: Base de Datos.

CAN: Controller Area Network.

CISC: Computadores de Conjunto de Instrucciones Complejo.

CMOS: Complementary Metal Oxide Semiconductor.

CPU: Central Processing Unit o Unidad Central de Procesamiento.

DBMS: Database Management System.

EIA: Electronics Industry Association

EPROM: Erasable Programmable Read Only Memory.

EEPROM: Electrical Erasable Programmable Read Only Memory.

FEPP: Fondo Ecuatoriano Populorum Progressio

IEEE: Institute of Electrical and Electronic engineers

LCD: Liquid Crystal Display.

ODBMS: Object-oriented Database Management System.

OTP: One Time Programmable.

PBP: Picbasic Pro.

PC: Computer Personal.

PCB: Print Circuit Board.

PIC: Peripheral Interface Controller.

RDBMS: Relational database management system.

RAM: Read Aleatory Memory.

ROM: Read Only Memory.

RISC: Computadores de Conjunto de Instrucciones Reducido.

SISC: Computadores de Conjunto de Instrucciones Específico.

USB: Universal Serial Bus.

Bibliografía

Microcontrolador, Wikipedia, Disponible en:

<<http://es.wikipedia.org/wiki/Microcontrolador>>, Fecha de consulta 15 de mayo de 2010.

Microcontrolador PIC, Wikipedia, Disponible en:

<http://es.wikipedia.org/wiki/Microcontrolador_PIC>, Fecha de consulta 15 de mayo de 2010.

Microcontrolador, Monografias, Disponible en:

<<http://www.monografias.com/trabajos12/microco/microco.shtml>>, Fecha de consulta 15 de mayo de 2010.

PicBasic PRO Compiler, TodoPic, Disponible en:

<http://www.todopic.com.ar/pbp_sp.html>, Fecha de consulta 15 de mayo de 2010.

Puerto Serie y Paralelo, Kioskea.net , Disponible en:

<<http://es.kioskea.net/contents/pc/serie.php3>>, Fecha de consulta 22 de mayo de 2010.

Definición comunicación Serial, Alegsa, Disponible en:

<<http://www.alegsa.com.ar/Dic/comunicacion%20serial.php>>, Fecha de consulta 22 de mayo de 2010.

Puerto Serie, Wikipedia, Disponible en:

<http://es.wikipedia.org/wiki/Puerto_serie>, Fecha de consulta 22 de mayo de 2010.

Comunicación Serial, Galaxi, Disponible en:

<<http://galaxi0.wordpress.com/el-puerto-serial/>>, Fecha de consulta 22 de mayo de 2010.

Puerto Serie, Debugmodeon, Disponible en:

<<http://es.debugmodeon.com/articulo/puerto-serie-uart-rs232-1-4>>, Fecha de consulta 27 de mayo de 2010.

Bases de Datos, maestrosdelweb, Disponible en:

<<http://www.maestrosdelweb.com/principiantes/que-son-las-bases-de-datos/>>, Fecha de consulta 27 de mayo de 2010.

BDD introducción, kioskea, Disponible en:

<<http://es.kioskea.net/contents/bdd/bddintro.php3>>, Fecha de consulta 27 de mayo de 2010.

Bases de Datos, Monografias, Disponible en:

<<http://www.monografias.com/trabajos11/basda/basda.shtml>>, Fecha de consulta 27 de mayo de 2010.

MySQL manual, MySQL, Disponible en:

<<http://www.mysql.com/>>, Fecha de consulta 27 de mayo de 2010.

Ingenieria de Requisitos, Wikipedia, Disponible en:

<http://es.wikipedia.org/wiki/Ingenieria_de_requisitos>, Fecha de consulta 27 de mayo de 2010.

SANTIAGO CORRALES, *Electrónica práctica con microcontroladores PIC, Programacion en lenguaje Basic, Electrónica y Computación Microchip*, 2006

A N E X O S

ANEXO 1

ENTREVISTA

Entrevista: E1	
Tipo: Técnica – Operacional	
Objetivos: Conocer el sistema actual de Registro Laboral y el modo como este opera.	
Responsable de Entrevista: Giovanni Madrid	
Persona Entrevistada: Geovanna Quishpe	Cargo: Coordinadora Departamento de Personal
Preguntas: 1) ¿Qué hardware usa para el registro Laboral? Un reloj tarjetero conectado a una fuente eléctrica. 2) ¿Qué datos nomas registra el reloj Tarjetero? La hora que se ve en el reloj 3) El reloj tarjetero es programable No se puede programar. 4) ¿Cómo se selecciona la hora de entrada o salida? El reloj tarjetero tiene un selector de varias posiciones el cual permite escogerla opción deseada. 5) ¿Qué opciones nomas tiene para escoger? • Entrada mañana, Salida mañana, • Entrada tarde, Salida tarde • Sobretiempo entrada, Sobretiempo salida 6) ¿Qué software tiene para almacenar los datos de las tarjetas?. Los datos son almacenados en una hoja de Excel. 7) El procesamiento de los datos en que software se lo hace. En Excel, hay una plantilla con formulas donde se va tipiendo los datos de las tarjetas.	

8) ¿Cuántos empleados utilizan el Reloj Tarjetero?

60 empleados

9) ¿Quién es el responsable de ingresar los datos al Excel?

Al ser una hoja de Excel en la que se maneja el registro, en cada departamento hay una responsable de mantener al día el Registro Laboral de su grupo.

10) ¿Cuántos departamentos usan el sistema actual

5 Departamentos.

11) ¿Qué tiempo les toma en ingresar los datos al Excel?

Dependiendo del número de personas que haya en cada departamento puede ir aproximadamente de 30 min a 1 hora.

12) ¿Cuáles son los horarios de la jornada laboral?

Entrada a las 8:00

Salida a la 16:30

Hora de almuerzo 45 min.

ANEXO 2

RELOJ TARJETERO



NOMBRE _____	No. _____
EMPRESA _____	AÑO _____ MES _____



PEDIDOS AL:

(t2) 396 - 5200

la competencia s.a.

KINGS POWER - NIPPO

MAÑANA		TARDE		SOBRETIEMPLO	
ENTRA	SALE	ENTRA	SALE	ENTRA	SALE

_____ Horas normales
Observaciones: _____

_____ Horas extras

Atrasos _____

ANEXO 3

REGISTRO EN EXCEL

19	13		L	10:38	14:29	14:55	18:38	8:00	7:34	-0:26					0:00	
20	14		MA	11:19	14:11	15:15	18:45	8:00	6:22	-1:38					0:00	
21	15		MI	14:43			19:15	8:00	4:32	-3:28					0:00	
22	16		J	10:13			19:07	8:00	8:54	0:54					0:00	
23	17		V	10:26	13:53	14:44	20:38	8:00	9:21	1:21					0:00	
24			S													
25			D													
26	18		L	10:24	14:08	14:57	18:12	8:00	6:59	-1:01					0:00	
27	19		MA	10:47	14:44	15:36	18:28	8:00	6:49	-1:11					0:00	
28	20		MI	10:39	14:34	15:00	18:19	8:00	7:14	-0:46					0:00	
29	21		J	10:34	14:31	14:50	18:17	8:00	7:24	-0:36					0:00	
30	22		V	9:46			20:54	8:00	11:08	3:08					0:00	
31			S												0:00	
Total # Horas								176:00:00	161:19:00	-14:41:00	0	0:00:00	0	0:00		
Total # Día								22:00	20:09	-1:50	0	0:00	0	0:00		

FER = Feriados

PER= Permisos.

VAC = Vacaciones.

ANEXO 4

DATA SHEET

PIC 16F873A

Pin Diagrams

28-Pin PDIP, SOIC, SSOP

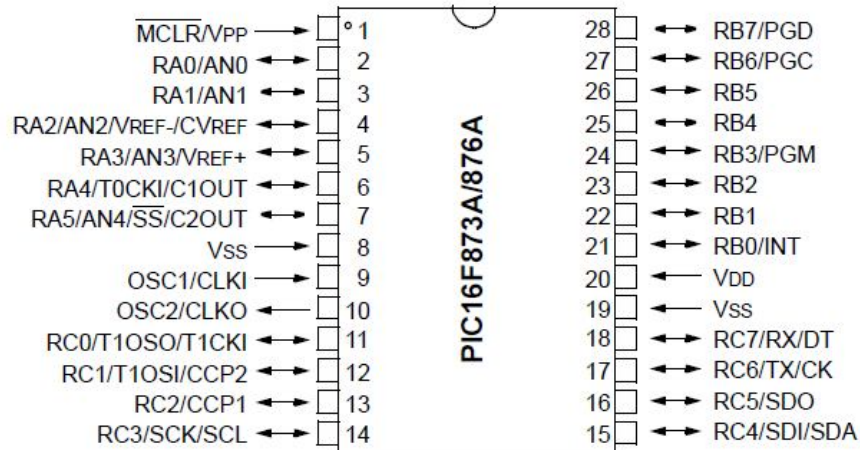


TABLE 1-1: PIC16F87XA DEVICE FEATURES

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

ANEXO 5

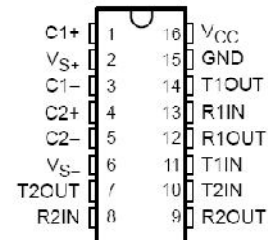
DATA SHEET

MAX 232

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SI I SC471 – FEBRUARY 1989 – REVISED OCTOBER 2002

- Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28
- Operate With Single 5-V Power Supply
- Operate Up to 120 kbit/s
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current . . . 8 mA Typical
- Designed to be Interchangeable With Maxim MAX232
- ESD Protection Exceeds JESD 22
 - 2000-V Human-Body Model (A114-A)
- Applications
 - TIA/EIA-232-F
 - Battery-Powered Systems
 - Terminals
 - Modems
 - Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)


description/ordering information

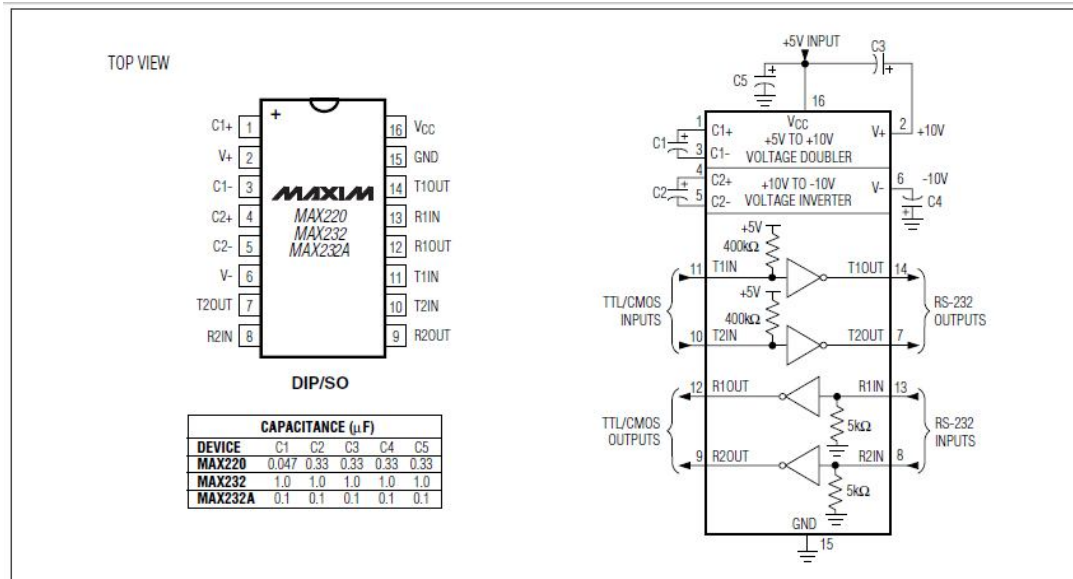
The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T _A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube	MAX232N	MAX232N
		Tube	MAX232D	MAX232
	SOIC (D)	Tape and reel	MAX232DR	
		Tube	MAX232DW	MAX232
		Tape and reel	MAX232DWR	
-40°C to 85°C	SOIC (NS)	Tape and reel	MAX232NSIR	MAX232
	PDIP (N)	Tube	MAX232IN	MAX232IN
		Tube	MAX232ID	MAX232I
	SOIC (D)	Tape and reel	MAX232IDR	
		Tube	MAX232IDW	MAX232I
		Tape and reel	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

PIN CONFIGURATION AND TYPICAL OPERATING CIRCUIT



www.datasheetcatalog.com

Datasheets for electronics components.

ANEXO 6

PRESUPUESTO

PRESUPUESTO

Items	Valor
Materiales Hardware	50
Suministros	10
Anillados	10
Impresiones	40
Otros	20
TOTAL	\$130

ANEXO 7

MANUAL

DEL

HARDWARE

INDICACIONES PREVIAS

Para el funcionamiento del hardware hay que seguir rigurosamente los siguientes pasos.

- Haber instalado el software GSFEPP.
- Constatar que esté conectado el prototipo al PC por medio de la Interfaz RS232.
- Constatar que el prototipo esté conectado a una fuente de poder.

NOTA IMPORTANTE

Debido a la configuración de emisor y receptor, el prototipo se debe encender antes de ejecutar la aplicación llamada **motor**

La aplicación **motor** se instala junto con la aplicación Registro Laboral al ejecutar el paquete de instalación GSFEPP.

El funcionamiento del prototipo depende de la aplicación motor.

Características Técnicas

Voltaje de entrada:	12 V
Amperaje:	500mA
Dimensiones (cm) (an*al*p)	10.5 * 19 * 6

Condiciones a tomar en cuenta.

- Cuando hay un corte de energía eléctrica el dispositivo necesita reiniciarse siguiendo el proceso de encendido.
- Cuando se desconecta el cable serial o se reinicia el servidor el Dispositivo pierde la configuración y necesita reiniciarse siguiendo el proceso de encendido.
- Si se desconecta o apaga el prototipo debe de reiniciarse siguiendo el proceso de encendido.

Proceso de encendido.

- Siempre iniciar con el dispositivo apagado.
- Encender el dispositivo
- Esperar hasta que salga mensaje de conexión
- Iniciar la aplicación en el PC.
- Verificar que se visualice la fecha, hora y mensaje de ingreso de código en el LCD.

Funcionamiento.

Luego de haber concluido el proceso de encendido el dispositivo está listo para su funcionamiento.

- 1) El personal debe ingresar el código de 4 dígitos dado por el administrador y presionar la tecla #.
- 2) El dispositivo devolverá su nombre en el LCD.
- 3) Esperar a que cambie al menú de opciones.
 - Pulsar la tecla A para registrar una entrada.
 - Pulsar la tecla B para registrar una salida.
 - Pulsar la tecla D para cancelar.

Con esto finalizaría el proceso de registro el Personal.

Casos particulares.

- Si el Personal desea borrar el código ingresado pulsar la tecla *.
- Si el Personal digita un código erróneo saldrá un mensaje de error.
- Si el Personal digita menos de los 4 números del código y pulsa #, saldrá un error pidiendo los 4 números.

Caso Especial.

Si se pierde la configuración del dispositivo por alguna causa antes mencionada, pese a mostrar la hora no se podrá ingresar el código.

Se debe reinicializar con el proceso de encendido nuevamente.