



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

**“SISTEMA PARA CLASIFICAR ASISTENCIA, BASADO EN
LÓGICA DIFUSA Y BASES DE DATOS RELACIONALES”.**

**TESIS PROFESIONAL
PARA OBTENER EL TÍTULO DE
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN.**

PRESENTA
Carlos Valdivia Carranco

ASESOR:
Dr. Pedro García Juárez.

COASESOR:
M.C. Hilario Salazar Martínez

Puebla, Pue

Junio 2008

Agradecimientos.

A Dios.

A mi Familia, conformada por mi esposa Andrea y mi hijo Carlitos, ya que ellos son mi fuente de energía y sin ellos no hubiera podido lograr esta meta.

A mis Padres, por su apoyo en todo lo que hago, muchas gracias.

A mis hermanas, por impulsarme día a día y sacarme canas verdes de vez en cuando.

A mi asesor el Dr. Pedro García, por su tiempo, críticas y consejos para que éste trabajo fuera realizado de la mejor manera posible.

A mi coasesor el M.C. Hilario Salazar por que cuando nadie quería ayudarme con este trabajo, él lo adoptó sin poner peros, gracias por el apoyo que me ha brindado.

Gracias a Todos.

Índice General

INTRODUCCIÓN.....	8
CAPÍTULO 1.	
CONCEPTOS FUNDAMENTALES DE LA LÓGICA DIFUSA Y LAS BASES DE DATOS. 10	
1.1 Breve Historia de la Lógica Difusa y sus Aplicaciones.....	10
1.2 Conceptos Fundamentales.	11
1.2.1 Conjuntos Difusos.	11
1.2.2 Propiedades de Conjuntos Difusos.	12
1.2.3 Operaciones con Conjuntos Difusos.	13
1.2.3.1 Complemento.....	13
1.2.3.2 Intersección.....	13
1.2.3.3 Unión.....	14
1.2.4 Funciones de Membresía.....	15
1.2.4.1 Función de Membresía Triangular.	15
1.2.4.2 Función de Membresía Trapezoidal.....	16
1.2.4.3 Función de Membresía Gamma.....	16
1.2.4.4 Función de Membresía Lambda.....	17
1.2.4.5 Función de Membresía S.....	18
1.2.4.6 Función de Membresía Z.....	19
1.2.4.7 Función de Membresía Gaussiana.	20
1.3 Breve Historia y sus Aplicaciones de las Bases de Datos.....	21
1.4 Conceptos Fundamentales de Bases de Datos Relacionales.	21
1.4.1 Base de Datos.....	21
1.4.2 Sistema Administrador de Bases de Datos.	22
1.4.2.1 Funcionamiento de un Sistema Administrador de Bases de Datos.....	24
1.4.3 Niveles de una base de Datos.	24
1.4.4 Proceso para el desarrollo de una base de datos.	26
1.4.5 Modelos conceptuales de datos.	27
1.4.5.1 Normalización del Modelo de Datos.....	29
1.4.5.1.1 Primera Forma Normal (1FN).....	29
1.4.5.2 Segunda Forma Normal (2FN).....	31
1.4.5.3 Tercera Forma Normal (3FN).	32
1.4.5.4 Cuarta Forma Normal (4FN) o de Boyce-Code (FNBC).	33
1.4.5.5 Quinta Forma Normal (5FN).....	34

CAPÍTULO 2.

DISEÑO DEL PROTOTIPO.....	35
2.1 Planteamiento del Problema.....	35
2.1.1 Diseño General Empleado – Institución.....	35
2.2 Diseño General del Sistema.....	36
2.2.1 Módulo Selección de Fecha.	39
2.2.2 Módulo Búsqueda de Actividades.	39
2.2.3 Módulo Búsqueda de Horarios.	40
2.2.4 Módulo Búsqueda de Registros.	40
2.2.5 Módulo de Evaluación.	41
2.2.6 Módulo de Análisis y Reportes.	41
2.3 Diseño General de las Bases de Datos Relacionales.....	42
2.3.1 Base de Datos Relacional: Control de Profesores.	42
2.3.2 Base de Datos Relacional: Control de Trabajo.	44
2.4 Ajuste de Funciones Difusas para el Sistema.	45
2.4.1 Funciones difusas para evaluar el Registro de Entrada:	45
2.4.1.1 Función Difusa de Asistencia para el registro de entrada.	45
2.4.1.2 Función Difusa de Retardo	46
2.4.1.3 Función Difusa de Falta o Inasistencia.	46
2.4.2 Funciones difusas para evaluar el Registro de Salida:	47
2.4.2.1 Función Difusa de Salida en Tiempo.	47
2.4.2.2 Función Difusa de Retiro Previo a la tolerancia	48
2.4.3 Funciones difusas para evaluar el Tiempo Efectivo Parcial:	49
2.4.3.1 Función Difusa de Tiempo Parcial Considerable.	49
2.4.3.2 Función Difusa de Tiempo Malo.....	50

CAPÍTULO 3.

IMPLEMENTACIÓN.....	51
3.1 Programación de los módulos.	51
3.1.1 Variables Generales de Control.	51
3.1.2 Módulo Selección de Fecha.	52
3.1.2.1 Pseudocódigo.	53
3.1.2.1.1 Actividades y Funciones a Realizar.	54
3.1.2.2 Código Fuente.	55
3.1.2.2.1 Creación de Variables.	55
3.1.2.2.2 Estructura del Módulo.	56
3.1.3 Módulo Búsqueda de Actividades.	56
3.1.3.1 Pseudocódigo.	57

3.1.3.1.1	Actividades y Funciones a Realizar.	57
3.1.3.2	Código Fuente.	58
3.1.3.2.1	Estructura del Módulo.	58
3.1.4	Módulo Búsqueda de Horarios.	58
3.1.4.1	Pseudocódigo.	58
3.1.4.1.1	Actividades y Funciones a Realizar.	59
3.1.4.2	Código Fuente.	59
3.1.4.2.1	Estructura del Módulo.	59
3.1.5	Módulo Búsqueda de Registros.	59
3.1.5.1	Pseudocódigo.	60
3.1.5.1.1	Actividades y Funciones a Realizar.	61
3.1.5.1.1.1	Estructura del módulo.	61
3.1.5.1.1.2	Procedimiento Checa_Entrada.	62
3.1.5.1.1.3	Procedimiento Checa_Entrada2.	62
3.1.5.1.1.4	Procedimiento Checa_Salida.	62
3.1.5.2	Código Fuente.	63
3.1.5.2.1	Estructura del Módulo.	63
3.1.5.2.2	Procedimiento Checa_Entrada.	64
3.1.5.2.3	Procedimiento Checa_Entrada2.	65
3.1.5.2.4	Procedimiento Checa_Salida.	66
3.1.6	Módulo de Evaluación.	66
3.1.6.1	Pseudocódigo.	66
3.1.6.1.1	Actividades y Funciones a Realizar.	67
3.1.6.1.1.1	Evaluando el registro de entrada.	68
3.1.6.1.1.2	Evaluando el registro de salida.	69
3.1.6.1.1.3	Evaluando el tiempo efectivo parcial.	69
3.1.6.2	Código Fuente.	69
3.1.6.2.1	Modo de acceso al módulo:	70
3.1.6.2.2	Estructura del Módulo.	70
3.1.6.2.3	Procedimientos Internos.	72
3.1.7	Módulo de Análisis y Reportes.	73
3.1.7.1	Pseudocódigo.	75
3.1.7.1.1	Actividades y Funciones a Realizar.	75
3.1.7.2	Código Fuente.	77
3.1.7.2.1	Al cargar el módulo:	78
3.1.7.2.2	Estructura del Módulo.	79
3.1.7.2.3	Llamado al Formulario Secundario Resumen.	81
3.1.7.2.4	Código Fuente del Formulario Resumen.	81

CAPÍTULO 4.

PRUEBAS Y RESULTADOS. 83

4.1 1ª Etapa: Migrando la información proporcionada por la institución. 83

4.2	2ª Etapa: Generando las Evaluaciones.....	85	
4.3	3ª Etapa: Muestra de Resultados.	86	
CAPÍTULO 5.			
CONCLUSIONES.....			91
5.1	Trabajos futuros.....	92	
ANEXO I.			
ASIGNACIÓN DE ETIQUETAS DIFUSAS.....			93
A1.1	Evaluando Funciones Difusas.	93	
A1.1.1	Funciones Difusas para el Registro de Entrada.....	93	
A1.1.1	Funciones Difusas para el Registro de Salida.....	94	
A1.1.1	Funciones Difusas para el Tiempo Efectivo Parcial.....	95	
A1.1	Etiquetando los Resultados.....	96	
A1.1.1	Etiquetas Difusas para el registro de entrada.....	96	
A1.1.2	Etiquetas Difusas para el registro de salida.....	97	
A1.1.2	Etiquetas Difusas para el tiempo efectivo parcial.....	97	
ANEXO II.			
CORRIMIENTO PASO A PASO.			98
A2.1	Módulo Selección de Fecha(s).	98	
A2.2	Módulo Búsqueda de Actividades.	99	
A2.3	Módulo Búsqueda de Horarios.....	100	
A2.4	Módulo Búsqueda de Registros.....	101	
A2.5	Módulo de Evaluación.....	103	
BIBLIOGRAFÍA.....			108

Índice de Diagramas y Figuras

Figura 1. 1: Función de Membresía Triangular.	15
Figura 1. 2: Función de Membresía Trapezoidal.	16
Figura 1. 3: Función de Membresía Gamma.	17
Figura 1. 4: Función de Membresía Gamma (Variación).	17
Figura 1. 5: Función de Membresía L.	18
Figura 1. 6: Función de Membresía S.	19
Figura 1. 7: Función de Membresía Z.	20
Figura 1. 8: Función de Membresía Guassiana (π).	20
Figura 1. 9: Esquema del funcionamiento de un DBMS.	24
Figura 1.10: Niveles de Abstracción.	25
Figura 1.11: Esquema del proceso para la creación de una base de datos.	27
Figura 1.12: Ejemplo de Diagrama Entidad – Relación.	29
Figura 1.13: Ejemplo del diagrama Entidad – Relación con la notación Barker.	29
Figura 1.14: Ejemplo de Tabla en Formalización CERO.	30
Figura 1.15: Ejemplo de Tabla en 1FN.	30
Figura 1.16: Ejemplo de la Tablas generadas por 2FN.	31
Figura 1.17: Ejemplo de las Tablas generadas por 3FN.	32
Figura 1.18: Ejemplos de las bases generadas por 4FN o FNBC.	34
Figura 2. 1: Figura laboral profesor – Institución.	36
Figura 2. 2: Diagrama General del Sistema.	36
Figura 2. 3: Esquema a Bloques.	37
Figura 2. 4: Esquema de Clases.	38
Figura 2. 5: Módulo Selección de Fecha.	39
Figura 2. 6: Módulo Búsqueda de Actividades.	39
Figura 2. 7: Módulo Búsqueda de Horarios.	40
Figura 2. 8: Módulo Búsqueda de Registros.	40
Figura 2. 9: Módulo de Evaluación.	41
Figura 2. 10: Módulo de Análisis y Reportes.	41
Figura 2. 11: Esquema Base de Datos Relacional Control de Profesores (Primera Parte).	42
Figura 2. 12: Esquema Base de Datos Relacional Control de Profesores (Segunda Parte).	43
Figura 2. 13: Esquema Base de Datos Relacional Control de Trabajo.	44
Figura 2. 14: Función Difusa de Asistencia para el Registro de Entrada.	45
Figura 2. 15: Función Difusa de Retardo para el Registro de Entrada.	46
Figura 2. 16: Función Difusa de Inasistencia para el Registro de Entrada.	47
Figura 2. 17: Función Difusa de Inasistencia para el Registro de Salida.	48
Figura 2. 18: Función Difusa de Retiro Previo para el Registro de Salida.	48

Figura 2. 19: Función Difusa de Tiempo Considerable para el Tiempo Efectivo Parcial	49
Figura 2. 20: Función Difusa de Tiempo Malo para el Tiempo Efectivo Parcial.....	50
Figura 3. 1: Pantalla de la Selección de Fecha(s) a Evaluar.....	53
Figura 3. 5: Pantalla de Reportes:	74
Figura 3. 6: Resultados mostrados por el módulo de reportes.....	76
Figura 3. 7: Desglose de resultados al dar doble click en un renglón deseado.....	77
Figura 4. 1: Ejemplo del archivo con los registros en Microsoft Excel.....	83
Figura 4. 2: : Ejemplo del archivo de texto con los registros.	84
Figura 4. 3: Formulario para importar los registros de los empleados al sistema.....	84
Figura 4. 4: Resultado de la importación de archivos proporcionados por la institución, en la tabla de los registros.	85
Figura 4. 5: Prueba gráfica del sistema, se muestra el funcionamiento de los módulos en el formulario.	86
Figura 4. 6: Se hace el llamado al Módulo Búsqueda de registros.....	86
Figura 4. 7: Reporte de Resultados sin Filtro.	87
Figura 4. 8: Reporte de Resultados con filtrado de Fechas.....	87
Figura 4. 9: Reporte de Resultados con Filtrado de Etiqueta Difusa de Entrada.....	88
Figura 4. 10: Reporte de Resultados con Filtrado de Etiqueta Difusa de Salida.	88
Figura 4. 11: Reporte de Resultados con Filtrado de Etiqueta Difusa de Tiempo Efectivo Parcial..	89
Figura 4. 12: Reporte de Resultados con Filtrado de la Clave del Empleado.....	89
Figura 4. 13: Archivo impreso directamente del sistema.	90
Figura A1. 1: Gráfica de Funciones Difusas para el registro de entrada	94
Figura A1. 2: Gráfica de Funciones Difusas para el registro de Salida	95
Figura A1. 3: Gráfica de Funciones Difusas para el tiempo efectivo parcial.....	96
Figura A2. 1: Acceso al módulo selección de Fecha(s).....	98
Figura A2. 2: Determinando día de la semana a evaluar y ejecución de la Consulta1.	99
Figura A2. 3: Efectuando Consulta2 para llamar a los siguientes módulos.	99
Figura A2. 4: Llamada interna al módulo búsqueda de actividades.	99
Figura A2. 5: Parámetro Clave_Empleado devuelto por el módulo.....	100
Figura A2. 6: Llamada interna al módulo búsqueda de horarios.....	100
Figura A2. 7: Parámetros de retorno del módulo son hora_entrada y hora_salida de la actividad.	100
Figura A2. 8: Accediendo al módulo búsqueda de registros.....	101
Figura A2. 9: Marcando rango para los registros y efectuar Consulta5.....	101
Figura A2. 10: Llamado al procedimiento Checa_Entrada.....	101

Figura A2. 11: Comprobando el valor del registro activo en el campo <Concepto>.	101
Figura A2. 12: Llamado al procedimiento Checa_Entrada2.....	102
Figura A2. 13: Comprobando valor en el campo activo.....	102
Figura A2. 14: Llamada al procedimiento Checa_Salida.....	102
Figura A2. 15: Comprobando valor en el campo activo.....	103
Figura A2. 16: Activando variable completo para validez que existen ambos registros.....	103
Figura A2. 17: Confirmando que haya encontrado ambos registros para llamar al siguiente módulo.	103
Figura A2. 18: Llamado al módulo de evaluaciones.....	103
Figura A2. 19: Accediendo al módulo de evaluación.....	104
Figura A2. 20: Evaluando en Funciones Difusas para el registro de entrada.....	104
Figura A2. 21: Evaluando en Funciones Difusas para el registro de salida.....	104
Figura A2. 22: Verificando tiempo efectivo parcial.....	105
Figura A2. 23: Llamada al procedimiento para evaluar el tiempo efectivo parcial.....	105
Figura A2. 24: Evaluando en Funciones Difusas para el Tiempo Efectivo Parcial.....	105
Figura A2. 25: Llamada al procedimiento para asignación de etiquetas difusas.....	105
Figura A2. 26: Asignando etiqueta difusa para el registro de entrada.....	106
Figura A2. 27: Asignando etiqueta difusa para el registro de salida.....	106
Figura A2. 28: Asignando Etiqueta Difusa para el tiempo efectivo parcial.....	106
Figura A2. 29: Llamada al procedimiento que llena las tablas con los resultados de la evaluación.	107

Introducción

Hoy en día la información es uno de los factores que más importancia tiene en el desarrollo de un país. Por este motivo, cualquier sociedad que pretenda no quedarse rezagada en el desarrollo tecnológico, debe estar al tanto de las técnicas que van surgiendo en el almacenamiento, transmisión y análisis de la información. Históricamente, las Bases de Datos han sido las herramientas diseñadas para llevar a cabo las tareas de almacenamiento y para proporcionar algunos de los mecanismos necesarios para el análisis de la información. El objetivo de una Base de Datos es el almacenar la información en forma conveniente, permitiendo su modificación de forma segura y facilitando el proceso de recuperación de aquella información que resulte necesaria.

Con respecto a la organización de los datos, los principales enfoques han sido el de Redes, el Jerárquico, el Relacional y en los últimos años el Orientado a Objetos y el de Bases de Datos Lógicas. En cuanto al tipo de los datos que pueden soportar las diferentes bases de datos, tradicionalmente han sido los de tipo texto, numéricos, de fechas, alfanuméricos por mencionar algunos.

La importancia de poder medir el rendimiento y el tiempo de cumplimiento en el trabajo de los empleados mediante normas de calidad, es una iniciativa que hoy en día es una necesidad para las instituciones educativas u otras organizaciones. Una gran ventaja que presentan éstas, es que en el manejo de su control de personal pueden utilizar las bases de datos, debido a que, éstas responden a la necesidad de manejar y coordinar el almacenamiento así como la recuperación de información, teniendo en cuenta los aspectos y problemas que esto implica.

A pesar del avance que tienen estos administradores de bases de datos, la mayoría no admiten el poder trabajar con conceptos básicos que las personas manejamos de forma cotidiana; un ejemplo muy común es cuando se intenta determinar si una persona llega temprano o llega tarde. Éstos conceptos de tiempo son muy inexactos y generan incertidumbre ya que habría que plantear rangos de tiempo y otros aspectos.

En el mundo real existen muchos conceptos no-perfectos, es decir, con conocimiento vago, impreciso, incierto, ambiguo, inexacto, o probabilístico por naturaleza. El razonamiento y pensamiento humano frecuentemente conlleva información de este tipo, probablemente originada de la inexactitud inherente de los conceptos humanos y del razonamiento basado en experiencias similares, pero no idénticas a experiencias anteriores. A estos tipos de problemas la lógica difusa intenta darles una solución, en base a un tipo de razonamiento aproximado.

La lógica difusa nace en 1965 a partir de la publicación del artículo "Fuzzy Sets", escrito por Lofti A. Zadeh para la revista Information and Control [27]. En contraste con la lógica convencional, que

utiliza conceptos absolutos para referirse a la realidad, la lógica difusa es una metodología que permite dar conclusiones a partir de información de entrada vaga, ambigua, con ruido o incompleta, siguiendo patrones de razonamiento similares a los del pensamiento humano.

En la actualidad, la lógica difusa es un campo de investigación muy importante, tanto por sus implicaciones matemáticas o teóricas como por sus aplicaciones prácticas. Algunas de las aportaciones de la lógica difusa están en: Teoría de Control, Inteligencia Artificial, Redes Neuronales por mencionar algunas. Prueba de esta importancia es el gran número de revistas internacionales de alto impacto (Fuzzy Sets and Systems, IEEE Transactions on Fuzzy Systems), congresos (FUZZ-IEEE, IPMU, EUSFLAT, ESTYLF...) y libros (First Course on Fuzzy Theory and Applications, 2005), (DESIGN OF ANALOG FUZZY LOGIC CONTROLLERS IN CMOS TECHNOLOGIES, 2003) dedicados al tema.

El objetivo principal de este trabajo es la creación de un sistema, que permita la clasificación de la asistencia del profesorado de una institución educativa, tomando como principales herramientas las bases de datos relacionales y la lógica difusa. La primera disciplina permitirá manipular información de forma rápida y flexible con respecto a los registros de entrada – salida y los horarios de las actividades; mientras que la segunda nos facilitará el manejo de conceptos de tiempo que son: llegar temprano, llegar en tiempo, llegar retrasado o no llegar y así poder clasificar la asistencia del personal mediante funciones adecuadas que permitan medir su grado de pertenencia.

Este documento está dividido en cinco capítulos. A continuación se describe brevemente de qué trata cada uno de ellos.

El Capítulo 1, “Conceptos Fundamentales de la Lógica Difusa y las Bases de Datos”, como su nombre lo indica se presentan los conceptos fundamentales de estas dos áreas, en las que se desarrolla el sistema, así como también sus principales aplicaciones. En el capítulo 2, se describe el análisis y diseño de los módulos que forman al sistema, siguiendo las técnicas de ingeniería de software, a este capítulo le llamamos “Diseño del Prototipo”. La descripción de los pasos que se realizaron para la implementación de los módulos, es presentado en el capítulo 3 así como también se presentan el pseudocódigo y el código fuente del sistema. La parte de pruebas y resultados, son presentados en el Capítulo 4. Y por último en el Capítulo 5, se dan las conclusiones finales y los posibles trabajos a futuro.

CAPÍTULO 1.

Conceptos Fundamentales de la Lógica Difusa y las Bases de Datos.

En este capítulo se presentan las dos áreas en las cuales se desarrolla el sistema: la lógica difusa y las bases de datos. En la primera parte se trata de los antecedentes que dan lugar a la teoría difusa, sus principales características, funciones y reglas que la rigen. En la segunda parte se hace un breve historial de las bases de datos, sus aplicaciones así como las normas de trabajo que tienen.

1.1 Breve Historia de la Lógica Difusa y sus Aplicaciones.

La lógica, en el sentido clásico, es considerada como un conjunto de reglas usadas para generar inferencias creíbles, basado en razonamiento exacto, es decir, una lógica dicotómica o binaria que admite dos posibles valores: verdadero o falso. En contraposición a la lógica dicotómica está la lógica multivaluada, que admite varios valores de verdad posibles.

La lógica difusa es una forma de lógica multivaluada y fue concebida por Lotfi A. Zadeh en el año de 1965, que disconforme con los conjuntos clásicos que sólo permiten dos opciones, la pertenencia o no de un elemento a dicho conjunto, la presentó como una forma de procesar información permitiendo pertenencia parciales a conjuntos, que en contraposición a los conjuntos clásicos los denominó Conjuntos Difusos [27]. Esta teoría permite representar el conocimiento común en un lenguaje matemático a través de la teoría de conjuntos difusos y funciones características asociadas a ellos, permitiendo el trabajo a la vez con datos numéricos y términos lingüísticos, logrando con esto tener flexibilidad y tolerancia con la imprecisión y la resolución de problemas no-lineales [4], [10].

En 1974, Assilian y Mamdani desarrollaron el primer controlador difuso, diseñado para una máquina de vapor, aunque la primera implementación real de un controlador de este tipo fue realizada en 1980 por F.L. Smidth & Co [13], [15], [26]. Otra aplicación se puede constatar en el controlador de la inyección química en plantas depuradoras de agua en Japón, creado por Fuji en 1983. Cuatro años después Hitachi hace lo propio para el control del tren-metro de Sendai. Cabe mencionar que los primeros controladores difusos comerciales fueron desarrollados por la empresa Morón [3].

El control difuso ha sido aplicado con mucho éxito en diversas ramas tecnológicas, por ejemplo: la metalurgia, controles de maniobras de aviones, sensores de imagen y sonido, reconocimiento de escritura, ascensores, aire acondicionado, sistemas de frenado ABS, electrodomésticos, mejoras en imágenes médicas, sistemas automáticos de regulación de la cantidad de anestesia, sistemas de concesión o denegación automática de créditos, entre otros. Estas son algunas de las

muchísimas aplicaciones de la lógica difusa que ya están funcionando en el campo de los llamados sistemas expertos.

Los primeros precursores en la construcción de reglas difusas mediante datos de entrenamiento son Takagi y Sugeno, teniendo una primera aproximación para ello (ver [26], [36], [37]). Otro de los factores que contribuye a seguir con la investigación en este campo, es el creciente interés en las redes neuronales y su similitud con los sistemas difusos, uno de los principales aportadores en las redes neuronales difusas es B. Kosko [1].

Actualmente se continúa investigando en áreas como el reconocimiento de patrones visuales o la identificación de segmentos de ADN, por mencionar dos ejemplos. Al parecer el futuro para poder abordar la inmensa cantidad de datos, la recuperación de información, controlar y gestionar la red, serán propias de las tecnologías borrosas.

1.2 Conceptos Fundamentales.

Muchos conceptos de la teoría clásica de conjuntos, se pueden hacer extensivos a los conjuntos difusos como son: unión, complemento, intersección por mencionar algunos. Sin embargo, otros son exclusivos de la teoría de conjuntos difusos.

Esta sección está dedicada en presentar algunos conceptos que van inmersos en la lógica difusa y que sirvieron como apoyo para entender tal teoría.

1.2.1 Conjuntos Difusos.

Recordemos que un conjunto A en el sentido clásico, puede definirse por extensión o por comprensión, es decir, listando los elementos que conforman al conjunto o especificando las propiedades de los elementos, para pertenecer a él.

Un ejemplo sería el siguiente. Sea $A = \{0, 2, 4, 6, 8\}$, observemos que este conjunto también puede representarse como

$$A = \{x \mid (x \text{ es par}) \wedge (0 \leq x \leq 9)\}.$$

Sea A un conjunto y $\varphi : A \rightarrow \{0,1\}$ definida por

$$\varphi_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases}$$

A tal función se le conoce como función característica de A ,

Observe que φ_A está íntimamente relacionada con la pertenencia o no pertenencia de los elementos de A .

En los conjuntos difusos, la función característica es conocida como función de membresía, la cual mide el grado de pertenencia al conjunto, tomando como referencia los valores entre 0 y 1, la verdad absoluta y la falsedad total. A tal función se le denota como $M_A : X \rightarrow [0,1]$

Un conjunto difuso A se representa, usualmente, como un conjunto de pares ordenados conformado por elementos x 's y su valor de pertenencia al conjunto $M_A(x)$, es decir

$$A = \{(x, M_A(x)) \mid x \in U\},$$

donde U es el universo de discurso.

1.2.2 Propiedades de Conjuntos Difusos.

Algunas de las propiedades o conceptos básicos más utilizados son: Igualdad, Punto de Cruce, Inclusión, Soporte, Núcleo entre otros. Los cuales presentamos a continuación.

Igualdad Conjuntos: Dos conjuntos difusos A y B sobre un universo U , son iguales si y sólo si sus funciones de membresía $M_A(x)$ y $M_B(x)$ son iguales, esto es $A=B \Leftrightarrow M_A(x)=M_B(x)$ para todo $x \in U$.

Punto de Cruce: Sea $x \in U$, A un conjunto difuso con función de membresía $M_A(x)$. Se dice que x es un punto de cruce para A , si $M_A(x) = 0.5$ y se denota como $\text{cruce}(x)$

Subconjunto: Sean A y B dos conjuntos difusos con funciones de membresía $M_A(x)$ y $M_B(x)$ respectivamente, se dice que A es subconjunto de B si y sólo si $M_A(x) \leq M_B(x)$ para todo $x \in U$.

Soporte: Sea A un conjunto difuso con función de membresía $M_A(x)$, se define el soporte de A como el conjunto clásico B , tal que si $x \in B$; hacen que $M_A(x) > 0$ y se denota $\text{soporte}(A)$.

Conjunto Vacío: Un conjunto difuso A es vacío si y sólo si su soporte no contiene elementos.

Conjunto Unitario: Un conjunto difuso A es unitario o singleton si y sólo si su soporte consta de un sólo elemento.

Núcleo: Sean B un conjunto, en sentido clásico, y A un conjunto difuso con función de membresía $M_A(x)$. B es el núcleo de A si contiene a todos los elementos de U tales que $M_A(x) = 1$ y se denota como $\text{núcleo}(A) = \{x \in U \mid M_A(x) = 1\}$.

Existen otros conceptos como altura de un conjunto, conjunto normalizado, alfa-corte que no son de nuestro interés y, que se pueden consultar en [14].

1.2.3 Operaciones con Conjuntos Difusos.

Al igual que la teoría clásica de conjuntos, se pueden definir operaciones como complemento, intersección y unión.

1.2.3.1 Complemento.

Sea A un conjunto difuso con función de pertenencia $M_A(x)$, sobre un universo de discurso U .

Sea $c : [0,1] \times [0,1] \rightarrow [0,1]$ una función tal que $\forall \alpha, \beta \in [0,1]$:

- | | | |
|-----|--|---------------------------------|
| C1: | $c(1)=0$ y $c(0)=1$ | Concordancia con el caso nítido |
| C2: | $\forall \alpha, \beta \in [0,1], \alpha > \beta \Rightarrow c(\alpha) < c(\beta)$ | Estrictamente decreciente |
| C3: | $\forall \alpha \in [0,1], c(c(\alpha)) = \alpha$ | Involución |

Se define el complemento $M_{\bar{A}}(x) = \{(x, M_{\bar{A}}(x)) \mid x \in U\}$, donde $M_{\bar{A}}(x)$ está dada por

$$M_{\bar{A}}(x) = c(M_A(x))$$

Se hace la distinción que el concepto de complemento satisface únicamente C1, C2 y C3, sin embargo cuando se cumplen adicionalmente

- | | | |
|----|----------------------------|------------------|
| C4 | c , una función continua | Función continua |
|----|----------------------------|------------------|

El concepto de complemento se denomina complemento fuerte [13], [14], [19], [20].

1.2.3.2 Intersección

Sean A y B dos conjuntos difusos con funciones de membresías $M_A(x)$ y $M_B(x)$

respectivamente, sobre un universo de discurso U . Sea $i : [0,1] \times [0,1] \rightarrow [0,1]$ una función tal que

$\forall \alpha, \beta, \gamma \in [0,1]$:

- | | | |
|-----|---|---------------------------------|
| I1: | $i(0,1) = i(0,0) = i(1,0) = 0; i(1,1) = 1$ | Concordancia con el caso nítido |
| I2: | $i(\alpha, \beta) = i(\beta, \alpha)$ | Conmutatividad |
| I3: | $i(\alpha, i(\beta, \gamma)) = i(i(\alpha, \beta), \gamma)$ | Asociatividad |
| I4: | $i(\alpha, 1) = \alpha$ | Identidad |
| I5 | $Si \alpha \leq \alpha' \wedge \beta \leq \beta'$, entonces $i(\alpha, \beta) \leq i(\alpha', \beta')$ | Monotonía |

Se define la intersección $A \cap B = \{(x, M_{A \cap B}(x)) \mid x \in U\}$, donde $M_{A \cap B}(x)$ está dada por $M_{A \cap B}(x) = i(M_A(x), M_B(x))$.

Cabe mencionar que las funciones que satisfagan I1, I2, ..., I5 se les llama normas triangulares o T-normas [13], [14], [19].

1.2.3.3 Unión.

Sean A y B dos conjuntos difusos sobre un universo de discurso U , con funciones de pertenencia $M_A(x)$ y $M_B(x)$ respectivamente. Sea f una función con dominio $[0,1] \times [0,1]$ y codominio $[0,1]$ tal que $\forall \alpha, \beta, \gamma \in [0,1]$:

U1:	$f(0,1) = f(1,1) = f(1,0) = 1; f(0,0) = 0$	Concordancia con el caso nítido
U2:	$f(\alpha, \beta) = f(\beta, \alpha)$	Conmutatividad
U3:	$f(\alpha, f(\beta, \gamma)) = f(f(\alpha, \beta), \gamma)$	Asociatividad
U4:	$f(\alpha, 0) = \alpha$	Identidad $A \cup \emptyset = A$
U5:	$Si \alpha \leq \alpha', \beta \leq \beta', entonces f(\alpha, \beta) \leq f(\alpha', \beta')$	Monotonía
U6:	$f(\alpha, \beta) = c(i(c(\alpha), c(\beta)))$ $i(\alpha, \beta) = c(f(c(\alpha), c(\beta)))$	Leyes de De Morgan

Se define la unión $A \cup B = \{(x, M_{A \cup B}(x)) \mid x \in U\}$, donde $M_{A \cup B}(x)$ está dada por

$$M_{A \cup B}(x) = f(M_A(x), M_B(x)), x \in U$$

Cabe mencionar que las funciones que satisfagan U1, U2, ..., U6 se les llama conormas triangulares o T-conorma [13], [14], [19].

Dentro de las funciones T-norma y T-conorma existen otros operadores que complementan esta sección. Los más importantes son: Operadores Idempotentes, Arquimedianos y Acotados.

Operadores Idempotentes: Dentro de estos operadores tenemos el Máximo para la Unión y el Mínimo para la Intersección. El máximo se puede expresar de la forma $f_{\max}(\alpha, \beta) = \max(\alpha, \beta)$; mientras que el mínimo puede expresarse de la forma $i_{\min}(\alpha, \beta) = \min(\alpha, \beta)$. Estos dos operadores satisfacen las propiedades de idempotencia, distributiva y son estrictamente crecientes.

Operadores Arquimedianos: Dentro de estos operadores tenemos la Suma Probabilística para la Unión representada por la ecuación $f^*(\alpha, \beta) = \alpha + \beta - \alpha * \beta$ y el Producto para la Intersección expresado por la ecuación $i - (\alpha, \beta = \alpha * \beta)$. Estos dos operadores no satisfacen las propiedades de idempotencia y la distributiva.

Operadores Acotados: Dentro de estos operadores tenemos la ecuación dada por $\min(1, \alpha + \beta)$ para la Unión y la ecuación $\max(0, \alpha + \beta - 1)$ para la Intersección. Estos operadores no satisfacen las propiedades de idempotencia, ni la distributiva ni la de absorción pero, satisfacen la conmutativa, asociativa y la de identidad.

1.2.4 Funciones de Membresía.

Recordemos que una función de membresía, es aquella que mide el grado de pertenencia de los elementos a un conjunto A y es denotada por $M_A(x)$. Este tipo de funciones son lo que caracterizan a la lógica difusa. Existen diversas funciones de membresía cuyo grado de convergencia a la toma de decisiones correctas son tolerantes a la imprecisión con que pueda definirse estas funciones. A continuación mostraremos algunas de las funciones difusas más comunes.

1.2.4.1 Función de Membresía Triangular.

Una función de membresía triangular se construye a partir de tres parámetros. Sean a, b, m tres números reales tales que $a < m < b$. Donde a es el límite inferior, b el límite superior y m es el valor modal. La ecuación que describe a la función de membresía triangular es:

$$\Delta(x, a, b, m) = \begin{cases} 0 & x \leq a \text{ o } b \leq x \\ \frac{x-a}{m-a} & a < x < m \\ \frac{b-x}{b-m} & m < x < b \\ 1 & x = m \end{cases}$$

También puede ser representada de la siguiente forma:

$$\Delta(x, a, b, m) = \max\left\{\min\left\{\frac{x-a}{m-a}, \frac{b-x}{b-m}\right\}, 0\right\}$$

La figura 1.1 muestra la gráfica de tal función.

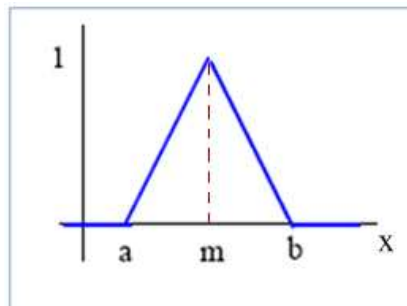


Figura 1. 1: Función de Membresía Triangular.

1.2.4.2 Función de Membresía Trapezoidal.

Una función de membresía trapezoidal se construye a partir de cuatro parámetros. Sean a, b, c, d números reales tales que $a < b < c < d$, donde a es el límite inferior, d el límite superior, b y c son los límites de su soporte inferior y superior respectivamente. La ecuación siguiente describe la función trapezoidal.

$$\text{trapezio}(x, a, b, c, d) = \begin{cases} 0 & x \leq a \text{ o } x \geq d \\ \frac{x-a}{b-a} & a < x \leq b \\ 1 & b < x \leq c \\ \frac{d-x}{d-c} & c < x \leq d \end{cases}$$

La función anterior puede ser representada por:

$$\text{trapezio}(x, a, b, c, d) = \max\left\{\min\left\{\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right\}, 0\right\}$$

La siguiente figura muestra el gráfico de una función trapezoidal.

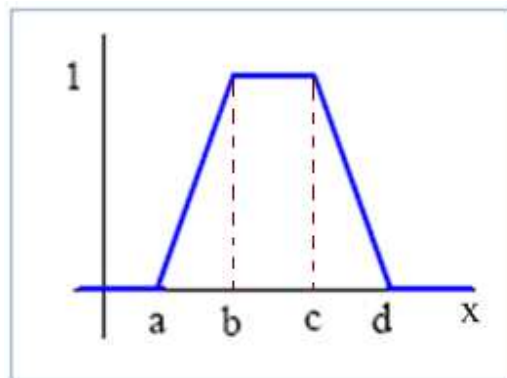


Figura 1. 2: Función de Membresía Trapezoidal.

1.2.4.3 Función de Membresía Gamma.

Para la construcción de una función de membresía del tipo gamma se requieren dos parámetros. Sean a, k ; donde a partir de a la función toma valores mayores o iguales a cero y k es un número mayor a 0. Las ecuaciones que describen a este tipo de funciones son

$$1 \quad \Gamma(x, a, k) = \begin{cases} 0 & x \leq a \\ 1 - e^{-k(x-a)^2} & x > a \end{cases}$$

$$2 \quad \Gamma(x, a, k) = \begin{cases} 0 & x \leq a \\ \frac{k(x-a)^2}{1 + k(x-a)^2} & x > a \end{cases}$$

Las funciones de tipo gamma, se caracterizan por un rápido crecimiento a partir de a dependiendo del valor de K , es decir, conforme el valor de k aumenta, el crecimiento de la función aumenta. Cabe mencionar que la función tiene una asíntota horizontal en 1.

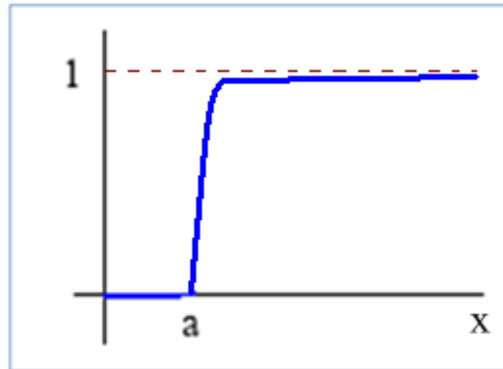


Figura 1. 3: Función de Membresía Gamma.

En [10], [14] y [19] presentan la modificación de este tipo de funciones para poder alcanzar el valor de 1. La ecuación a la que hacen referencia para su representación está dada por:

$$\Gamma(x, a, m) = \begin{cases} 0 & x \leq a \\ \frac{x - a}{m - a} & a < x < m \\ 1 & x \geq m \end{cases}$$

Donde a es el límite inferior y m es el valor modal, tal que $a < m$. La gráfica asociada a esta función es mostrada en la siguiente figura.

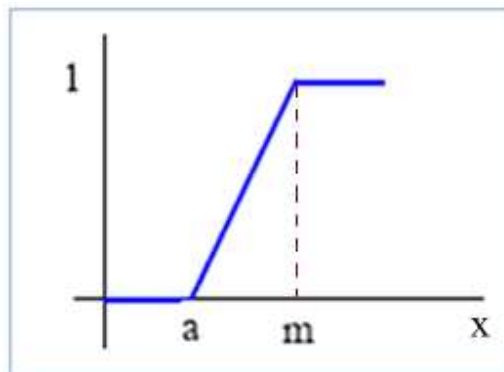


Figura 1. 4: Función de Membresía Gamma (Variación).

1.2.4.4 Función de Membresía Lambda.

Las funciones de membresía lambda usualmente son las que se conocen como funciones de tipo L . Para la construcción de este tipo de funciones se requieren dos parámetros. Sean a , m dos números reales tales que $m < a$, donde a es el límite superior y m es el valor modal. La ecuación que representa a las funciones de tipo L es:

$$\mathbf{L}(x, a, m) = \begin{cases} \mathbf{0} & x \geq a \\ \frac{a-x}{a-m} & m < x < a \\ \mathbf{1} & x \leq m \end{cases}$$

La siguiente figura muestra el gráfico de tales funciones.

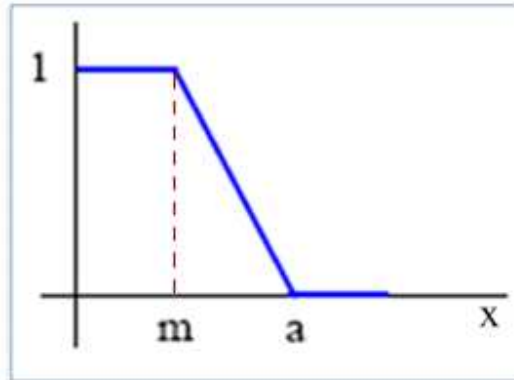


Figura 1. 5: Función de Membresía L.

Observe que las funciones de tipo L son reflexiones de las funciones de tipo Gamma, es decir, $\mathbf{L}(x, a, m) = \mathbf{1} - \mathbf{\Gamma}(x, a, m)$.

Nota: Las funciones de tipo *Gamma* y de tipo L son utilizadas principalmente para representar valores lingüísticos extremos tales como: Bebé o Anciano. Las funciones Triangulares y Trapezoidales (TR) son utilizadas para representar valores intermedios como son: Joven, de mediana Edad, Maduro. Su principal diferencia de estas dos funciones reside en que la función TR implica un margen de tolerancia alrededor del valor que se toma como más representativo del valor lingüístico asociado al conjunto difuso, mientras que la otra no tiene un margen de tolerancia, es decir es más cerrada en ese aspecto.

Existen otro tipo de funciones de membresía, que no son completamente lineales en trozos, como la Función de Membresía S, Función de Membresía Z y Función de Membresía Gaussiana que a continuación se detallan.

1.2.4.5 Función de Membresía S.

Para la construcción de una función de membresía S se requieren tres parámetros. Sean a , m , b números reales tales que $a < m < b$, donde a es el límite inferior, b el límite superior y m es el punto de inflexión, comúnmente $m=(a+b/2)$. La ecuación que representa a este tipo de funciones es:

$$S(x,a,b,m) = \begin{cases} 0 & x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2 & a < x \leq m \\ 1 - 2\left(\frac{x-b}{b-a}\right)^2 & m < x < b \\ 1 & b \leq x \end{cases}$$

La figura 1.6 muestra el gráfico de la función de membresía de tipo S.

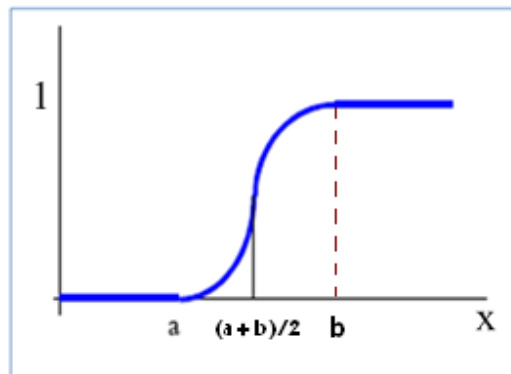


Figura 1. 6: Función de Membresía S.

1.2.4.6 Función de Membresía Z.

Las funciones de membresía del tipo Z son construidas a partir de tres parámetros. Sean a , m , b tres números reales tales que $a < m < b$, donde a es el límite superior, b el límite inferior y m es el punto de inflexión, el cual comúnmente su valor es $(a+b)/2$. La ecuación que representa a las funciones de tipo Z es de la forma:

$$Z(x,a,b,m) = \begin{cases} 1 & x \leq a \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2 & a < x \leq m \\ 2\left(\frac{x-b}{b-a}\right)^2 & m < x < b \\ 0 & b \leq x \end{cases}$$

La siguiente figura muestra el gráfico de este tipo de funciones.

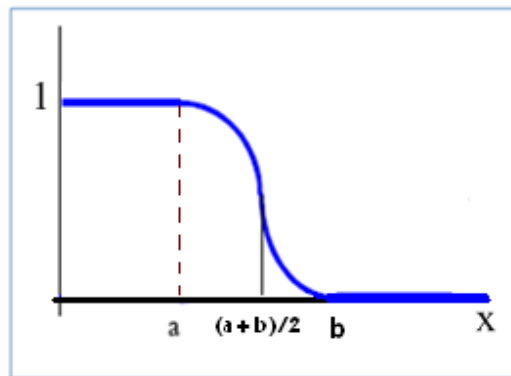


Figura 1. 7: Función de Membresía Z.

Observe que las funciones de tipo Z son reflexiones de las funciones de tipo S, es decir $Z(x,a,b,m) = 1 - S(x,a,b,m)$.

1.2.4.7 Función de Membresía Gaussiana.

Las funciones de membresía del tipo Gaussianas son construidas a partir de dos parámetros, sean m , k donde m es el punto medio de la función y k es un valor mayor a 0. La ecuación que representa a las funciones del tipo gaussianas es:

$$\pi(x, m, k) = e^{-k(x-m)}$$

La figura 1.8 muestra la gráfica de las funciones tipo gaussianas.

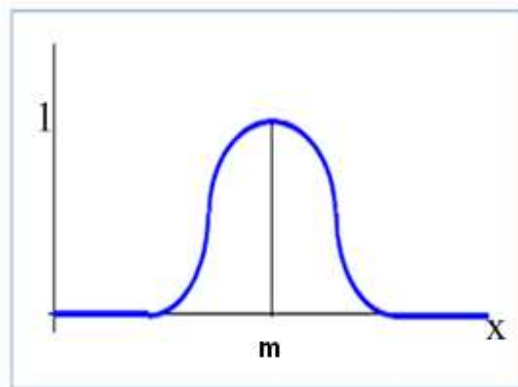


Figura 1. 8: Función de Membresía Gaussiana (π).

En [10], [14] y [19] dan otra representación para la ecuación de este tipo de funciones, la ecuación a la cual hacen referencia para su representación está dada por:

$$\pi(x, m, k) = \begin{cases} S(x) & x \leq m \\ Z(x) & x > m \end{cases}$$

1.3 Breve Historia y sus Aplicaciones de las Bases de Datos.

Una de las desventajas de los primeros programas, es que no tenían la opción de guardar información en memoria secundaria, debido a que tenían una sola actividad específica. Además que la tecnología no tenía los elementos necesarios para el almacenamiento de la información tanto procesada como la proporcionada. Conforme al avance de la tecnología, se va resolviendo el problema del almacenamiento de la información, con lo cual surge un nuevo reto, la administración de dicha información.

Los sistemas de archivos son la primera herramienta que da solución a esta problemática, donde los datos se almacenan de manera permanente para sobrevivir a los programas que los usan, a esta característica se le conoce como la persistencia de datos.

Aunque los sistemas de archivos representaron un gran avance, éstos se enfrentaron a nuevos problemas, como es la redundancia de los datos, es decir que un mismo dato podría estar repetido en un mismo archivo o en otros. Otro problema fue el no poder manipular tal información entre archivos distintos, por lo que la información no podía relacionarse de forma directa, a este tipo de problema se le conoce como inflexibilidad de la información. El problema más grande que presentaron los sistemas de archivos, se dio al cambiar la estructura de los datos que por cada cambio de estructura necesitó una reingeniería en los programas de control, propiciando con esto un alto costo en tiempo y en mano de obra.

El comienzo de las bases de datos se debe al proyecto APOLO a principios de los 70's, la empresa encargada del proyecto fue la NAA (North American Aviation), desarrollando un software denominado GUAM (General Update Access Method) que se basaba en la idea de unir varias piezas pequeñas para formar una pieza más grande. La compañía IBM (International Business Machines) se incorpora a este proyecto a mediados de los años sesenta, formando lo que ahora se conoce como IMS (Information Management System), restringiéndolo al manejo de jerarquías de registros con la finalidad de poder permitir el uso de dispositivos de almacenamiento serie (cintas magnéticas), ya que era un requisito del mercado por aquella época [9], [18].

El Dr. E.F. Codd de la compañía de IBM en 1970 en el afán de simplificar la estructura en las Bases de Datos, generó el modelo relacional, dando lugar a los Sistemas Administradores de Bases de Datos; estos sistemas están basados en 12 reglas publicadas por él en 1986 [5], [6], [7], [8].

1.4 Conceptos Fundamentales de Bases de Datos Relacionales.

La cantidad de información que debe manejar una organización es cada vez mayor. Para ello, deben existir métodos eficientes tanto para el almacenamiento rápido como para la consulta ágil. La herramienta tecnológica que actualmente es más utilizada para ésta es la de Bases de Datos.

1.4.1 Base de Datos.

La tecnología de bases de datos proporciona diversos medios para que las organizaciones tengan mayor operatividad con los datos y como consecuencia una información más confiable para la toma de decisiones e incrementar su productividad.

Definición (Base de Datos). Una base de datos es una colección de datos almacenada de una manera permanente, que puede ser compartida y usada con variados propósitos por múltiples usuarios.

Los usuarios de una base de datos se pueden clasificar en tres categorías:

Usuario Final: Es el usuario que interactúa con la base de datos, por lo general, a través, de las aplicaciones.

Usuario Especialista: Es el usuario que diseña y programa las aplicaciones para los usuarios finales.

Usuario Administrador: Es el encargado de administrar la base de datos, es llamada Data Base Administrator (DBA).

1.4.2 Sistema Administrador de Bases de Datos.

Un sistema Administrador de Bases de Datos (DBMS o Data Base Management System) es el software que sirve de intermediario entre el usuario y la base de Datos y tiene las siguientes características.

- a. Interactuar en forma transparente con el manejador de archivos del sistema operativo para la actualización, almacenamiento y recuperación de los datos.

El usuario, a excepción del DBA, no debería preocuparse por las estructuras internas o por los procedimientos usados para manipularlos.

- b. Optimizar la búsqueda de la información.
- c. Ofrecer un catálogo fácil de manejar por el usuario.
- d. Manejo de transacciones, esto para que se realicen todas o ninguna de las actualizaciones.

Tomemos en cuenta que una transacción es una secuencia de pasos para cumplir una tarea, según el punto de vista del usuario final.

- e. Controlar el acceso concurrente o simultáneo a los mismos datos para que no existan conflictos por las peticiones de los usuarios o aplicaciones.
- f. Servicios de recuperación ante fallas.
- g. Seguridad para impedir el acceso a los datos a los usuarios no autorizados.
- h. Implementar mecanismos para preservar la integridad de los datos mediante la especificación.
- i. Servicios para independizar datos.

Se puede apreciar que estos tipos de sistemas deben realizar muchas tareas bastante complejas y de ahí su tamaño y costo.

Uno de los objetivos de los sistemas de bases de datos es tener la posibilidad de usar los datos de nuevas maneras sin generar una reacción en cadena de modificaciones difíciles sobre los otros programas existentes. Así también lograr la independencia de datos, lo cual consiste en que todos los programas estén más aislados de los efectos de reorganizar los datos.

Un sistema de base de datos tiene ventajas y desventajas en su manejo. A continuación mencionamos algunas de estas situaciones.

Dentro de las ventajas, están:

- | | |
|--|--|
| Economía de Escala: | La concentración de aplicaciones en una sola localidad puede reducir costos, entre ellos tener menor cantidad de personas especializadas, menor gasto en software, etc. |
| Facilidad para el análisis: | Se puede obtener mayor información de la misma cantidad de datos. |
| Datos y programas compartidos: | La reutilización de los mismos datos y programas, permiten minimizar o controlar la redundancia. |
| Manejo de Estándares: | Incentiva la adopción de estándares. |
| Consistencia de los datos: | Está dada por el control o eliminación de la redundancia. |
| Integridad: | Se deben definir reglas que deben cumplir los datos, en la base de datos. |
| Seguridad: | Se pueden especificar niveles de acceso con una granularidad más fina, según los perfiles de los usuarios. |
| Flexibilidad: | El uso de lenguajes de cuarta generación hacen más fácil la construcción de los programas por parte de los usuarios finales. |
| Mayor productividad de los programadores: | Las nuevas aplicaciones pueden desarrollarse en la mitad del tiempo, o menos, que con los sistemas de archivos tradicionales debido al uso de lenguajes de tercera generación. |
| Mantenimiento y reingeniería: | Se puede cambiar la estructura de los datos sin cambiar los programas que los usan. |

Las principales desventajas de este tipo de sistema son:

- | | |
|----------------|---|
| Tamaño: | Un Sistema Administrador de Bases de Datos es un gran |
|----------------|---|

conjunto de programas.

Susceptibilidad a las fallas Más cantidad de información almacenada en un solo lugar.

Recuperación a las fallas: La recuperación en un sistema administrador de bases de datos interactivo y multiusuario puede ser muy compleja.

1.4.2.1 Funcionamiento de un Sistema Administrador de Bases de Datos.

El funcionamiento de estos administradores, se resume en la siguiente figura.

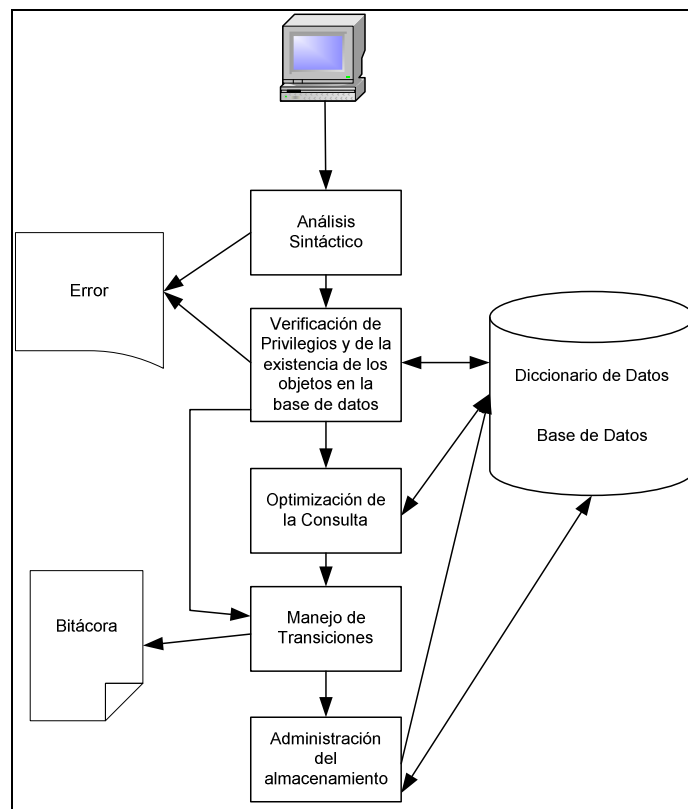


Figura 1. 9: Esquema del funcionamiento de un DBMS

En la figura anterior se observa el funcionamiento general de los DBMS, que en primera instancia analizarán las sintaxis de las solicitudes y los permisos que tienen los usuarios en la base de datos para poder efectuar las consultas solicitadas por éstos. Cabe mencionar que cada una de las transacciones efectuadas, se almacena en un historial en forma de bitácora para tener un control de seguridad.

1.4.3 Niveles de una base de Datos.

La manera en que percibimos las bases de datos, según seamos usuarios finales, especialistas o administradores, corresponde con un nivel de abstracción. La ANSI/SPARC (American National Standard Institute - Standards Planning and Requirements Committee) ha propuesto una

arquitectura considerando los niveles de abstracción: Nivel de Visión, Nivel Conceptual y Nivel Físico.

Nivel de Visión o Externo: Este nivel es el más cercano a los usuarios y se ocupa de la forma de cómo los usuarios individuales perciben una base de datos.

Nivel Conceptual: Es el nivel mediador entre el nivel físico y el de visión, se ocupa de cuáles son los datos reales almacenados en la base de datos y de las relaciones existentes entre ellos.

Nivel Físico o interno: Es el nivel más cercano a la máquina, es decir, es el que se ocupa de la forma de cómo se almacenan los datos físicamente en la memoria secundaria.

Estos tres niveles deben presentar tres características: Independencia de datos, independencia física e independencia lógica.

Independencia de Datos: Es una característica de las bases de datos que permiten modificaciones en la definición de un esquema sin afectar, en la medida de lo posible, la reescritura del esquema inmediatamente superior.

Independencia Física: Es una característica de las bases de datos que permiten modificaciones en el esquema físico sin afectar cambios en el esquema lógico.

Independencia Lógica: Es una característica de las bases de datos que permiten modificaciones en el esquema lógico sin afectar cambios en el nivel de visión. Este tipo de independencia es más difícil de lograr que la independencia física.

En la figura 1.10 se muestran de forma gráfica a estos tres niveles.

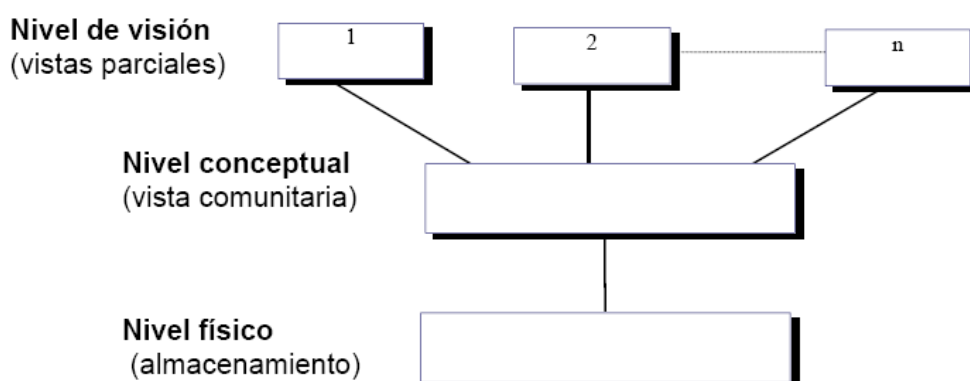


Figura 1.10: Niveles de Abstracción.

1.4.4 Proceso para el desarrollo de una base de datos.

Para la creación de una base de datos, se requieren de ciertas técnicas, las cuales permitan transformar los requerimientos para el manejo de información almacenada, tomando en cuenta el diseño conceptual, lógico y físico.

Diseño Conceptual: Es la parte del desarrollo de una base de datos en donde se especifican los requerimientos y el resultado que se espera. Su propósito es describir el contenido de información de la base de datos, más que las estructuras de almacenamiento que se necesitarán para manejar la información. Es una descripción de alto nivel que es completamente independiente del software de DBMS que se use, incluso si se pensara en implementar con archivos tradicionales y con algún lenguaje de programación convencional.

Diseño Lógico: Es donde se da una descripción de la estructura de la base de datos que puede procesar el software DBMS. El modelo lógico más usado actualmente, es el modelo relacional que ha sido enriquecido con los modelos orientados a objetos. El modelo lógico no depende del DBMS en particular, sino del modelo de datos usado éste.

Diseño Físico: Aquí se describe cómo está almacenada la base de datos en la memoria secundaria así como también las estructuras de almacenamiento y los métodos usados para tener un acceso efectivo a los datos.

Los esquemas lógicos y físicos se expresan haciendo uso del lenguaje de definición del DBMS elegido.

El diseño general del proceso para la creación de una base de datos se muestra en la siguiente figura.

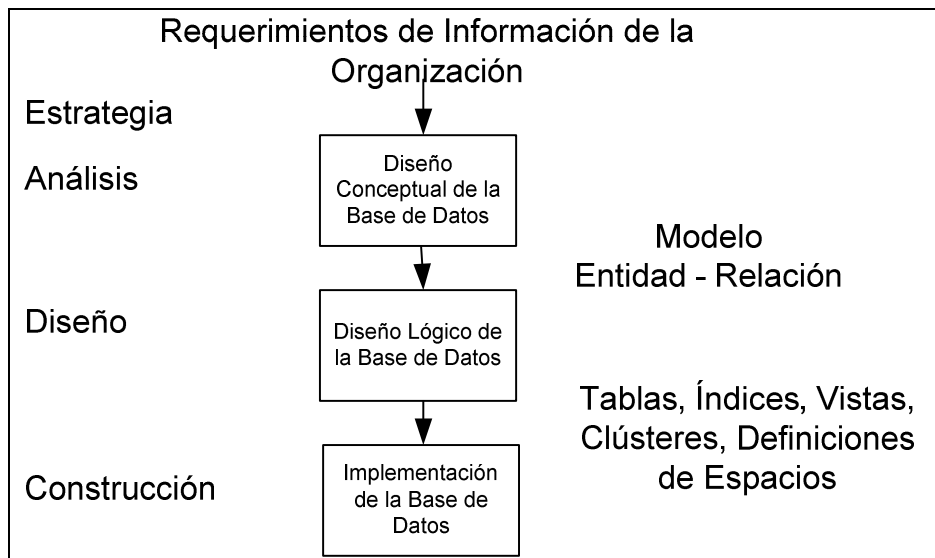


Figura 1.11: Esquema del proceso para la creación de una base de datos.

1.4.5 Modelos conceptuales de datos.

Un modelo de datos es considerado cómo una serie de conceptos que se utilizan para describir un conjunto de datos y de las operaciones para la manipulación de éstos. Cuando los datos describen a una realidad se llamará modelo de datos conceptuales.

En el modelado conceptual se identifican las propiedades estructurales sobre los objetos, sus atributos y relaciones, así como las propiedades dinámicas como son las operaciones sobre los objetos. Además, de identificar ciertas restricciones de integridad de un dominio de aplicación.

Estos modelos están caracterizados por su expresividad, simplicidad, minimalidad y formalidad.

Expresividad. Los modelos más ricos en conceptos son más expresivos. La mayoría de los modelos conceptuales hacen uso frecuente de la abstracción de generalización lo que permite la representación de una gran variedad de restricciones de integridad.

Simplicidad. Un modelo conceptual debe ser simple o fácil de entender por los diseñadores y por los usuarios finales. Esta propiedad y la expresividad son objetivos en conflicto; si un modelo es semánticamente rico, es probable que no sea simple.

Minimalidad. Esta característica se consigue si ningún concepto presente puede expresarse por otros conceptos. Esta característica persigue que el modelo abstraiga lo esencial de la realidad, buscando la parsimonia.

Formalidad. Todos los conceptos deben tener una interpretación única, precisa y bien definida.

Los modelos en la ingeniería de software, suelen describirse mediante representaciones lingüísticas y gráficas. El éxito de un modelo de datos depende en gran medida de su representación gráfica que debe ser lo más completa posible y con facilidad de lectura.

El modelo Entidad – Relación (E-R) es el más usado para el modelado conceptual de bases de datos. Fue propuesto por Peter Chen en 1976. En 1998, la ANSI seleccionó el modelo E-R como el modelo estándar para los sistemas de diccionarios de recursos de información (IRDS, Information Resource Dictionary Systems) [18], [36], [37]. Posteriormente, el modelo fue extendido por otros autores, como Richard Barker en la metodología Case Method, para adicionar una mayor y mejor semántica.

Los conceptos básicos de un modelo Entidad - Relación son Entidad, Relación y Atributo que se describen a continuación.

Entidad: Una entidad es una etiqueta que representa a un grupo de objetos con una misma característica, llamados instancias. Las entidades se simbolizan en un diagrama E-R mediante un rectángulo o caja donde en su interior se le asigna un nombre. Ejemplos de entidades habituales en los sistemas de información son: facturas, personas, empleados, etc.

Relación: Una relación describe la interdependencia entre entidades, cuya representación en un diagrama E-R es con un rombo etiquetado en su interior por un verbo, éste debe unirse mediante líneas con las entidades que relaciona.

Atributo: Los atributos son propiedades relevantes propias de una entidad y/o relación. Es representado en un diagrama E-R mediante un círculo o elipse etiquetado mediante un nombre en su interior. Cuando un atributo es identificativo de la entidad se suele subrayar dicha etiqueta.

El modelo Entidad – Relación es un modelo efectivo para recopilar y documentar los requerimientos en un formato claro y preciso. Esto es, un marco de especificación formal, fácil de entender por los usuarios finales, agilizando la comunicación entre éstos y los usuarios especialistas.

En la Figura 1.12 podemos ver un ejemplo del Modelo Entidad – Relación, mientras que en la figura 1.13 con la notación Barker.

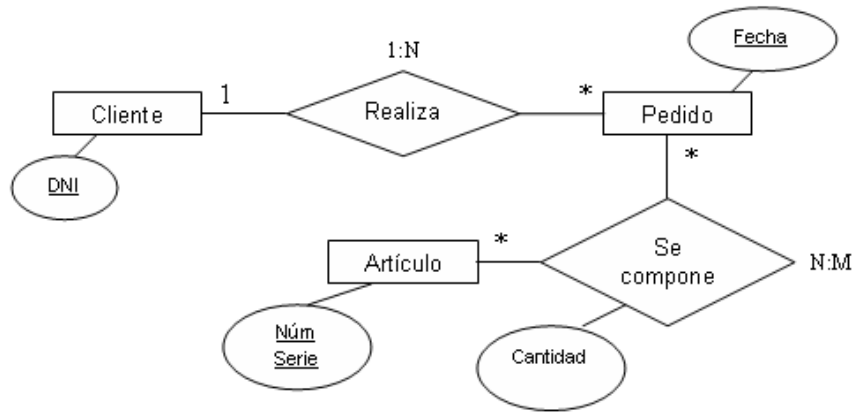


Figura 1.12: Ejemplo de Diagrama Entidad – Relación

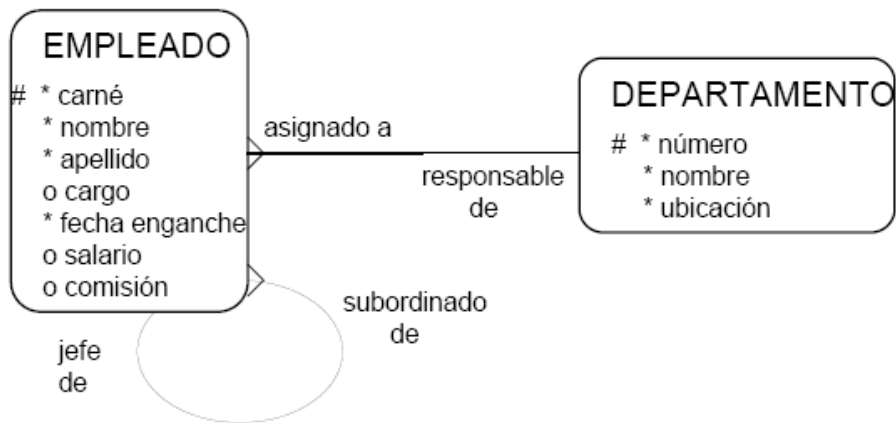


Figura 1.13: Ejemplo del diagrama Entidad – Relación con la notación Barker.

1.4.5.1 Normalización del Modelo de Datos.

La normalización es el proceso de organizar bases de datos, donde se incluye la creación de tablas y las relaciones entre ellas. Estas relaciones se establecen mediante reglas con la finalidad de proteger los datos así como para hacer más flexible la eliminación de la redundancia y las dependencias incoherentes de las tablas recién creadas o de las existentes en la base de datos.

Básicamente las reglas progresivas para normalizar una base de datos son cinco. A cada una de ellas se le denomina "forma normal". A continuación se describirán cada una de ellas.

1.4.5.1.1 Primera Forma Normal (1FN).

Una base de datos está en primera forma normal (1FN), si los valores para cada atributo de la relación son atómicos o indivisibles, es decir, que cada atributo sólo puede pertenecer a un dominio y que tiene un valor único para cada fila.

Cabe mencionar que 1FN se definió para prohibir los atributos multivaluados o compuestos así como sus combinaciones.

Para alcanzar la 1FN se sugiere seguir los siguientes pasos:

- 1º. Eliminar los grupos repetidos de las tablas individuales.
- 2º. Crear una tabla independiente para cada conjunto de datos relacionados.
- 3º. Identificar cada conjunto de datos relacionados con una clave principal.

La Figura 1.14 representa una tabla sin formalización mientras que la figura 1.15 representa a ésta misma en su 1FN.

Empleados							
APaterno	AMaterno	Nombre(s)	Curr	ANac	HInf	Teléfono(1)	Teléfono(2)
Domínguez	Chávez	José Alfonso	Si	Si	Si	22.22.75.81.71	4.62.13.91,
Flores	Méndez	Karina	Si	Si	Si	2221.03.29.94	2.20.71.41,
Portillo	Zepeda	Claudia Elena	Si	Si	Si	3.95.62.27,	

Figura 1.14: Ejemplo de Tabla en Formalización CERO.

Empleados							
EmplID	APaterno	AMaterno	Nombre(s)	Curr	ANac	HInf	Teléfono
1	Domínguez	Chávez	José Alfonso	Si	Si	Si	22.22.75.81.71
1	Domínguez	Chávez	José Alfonso	Si	Si	Si	4.62.13.91,
2	Flores	Méndez	Karina	Si	Si	Si	2221.03.29.94
2	Flores	Méndez	Karina	Si	Si	Si	2.20.71.41,
3	Portillo	Zepeda	Claudia Elena	Si	Si	Si	3.95.62.27,

Figura 1.15: Ejemplo de Tabla en 1FN

Observemos que en la figura 1.14 se tiene una tabla con dos campos para almacenar *Teléfonos*, al utilizar la 1FN se reduce a un solo campo *Teléfono* y se crea un nuevo campo llamado *EmplID* que va a fungir como clave principal en la distinción de los *Teléfonos*, esto es mostrado en la figura 1.15.

1.4.5.2 Segunda Forma Normal (2FN).

Se dice que una base de datos está en segunda forma normal (2FN), si está en 1FN y todos los atributos secundarios reducidos por 1FN, no sólo dependerán de una clave primaria sino también de claves compuestas que se obtienen de los atributos no reducidos por 1FN..

Para llevar a una base de datos a su 2FN se sugiere seguir los siguientes pasos:

- a. Realizar 1FN.
- b. Crear tablas independientes para conjuntos de valores que se apliquen a varios registros.
- c. Relacionar estas tablas con una clave externa.

Los registros no deben depender de nada que no sea una clave principal de una tabla, una clave compuesta si es necesario.

Observe que en la figura 1.15, los atributos que no fueron reducidos cuando se llevó a su 1FN son *APaterno, AMaterno, Nombre(s), Curr, ANac, HInf* causan duplicidad.

Al efectuar 2FN en la tabla 1FN, se elimina la redundancia y se les reasigna una clave principal reduciendo la tabla *Empleados*. Los atributos normalizados por 1FN (*Teléfono*) crean una nueva tabla con dependencia al identificador *REEMPLID* así como su propia clave principal (*TELID*), ver figura 1.16.

Empleados						
EmplID	APaterno	AMaterno	Nombre(s)	Curr	ANac	HInf
1	Domínguez	Chávez	José Alfonso	Si	Si	Si
2	Flores	Méndez	Karina	Si	Si	Si
3	Portillo	Zepeda	Claudia Elena	Si	Si	Si

Teléfonos		
TelId	RelEmplID	Teléfono
1	1	22.22.75.81.71
2	1	4.62.13.91,
3	2	2221.03.29.94
4	2	2.20.71.41,
5	3	3.95.62.27,

Figura 1.16: Ejemplo de la Tablas generadas por 2FN

1.4.5.3 Tercera Forma Normal (3FN).

Una tabla está en tercera forma normal (3FN) si todos los atributos de la relación dependen funcionalmente de la clave principal.

Para llevar una tabla a su 3FN se tienen que eliminar aquellos campos que son información secundaria, creando para ellos tablas con dicha información.

En la figura 1.16 los atributos de la tabla *EMPLEADOS* son: *EMPLID*, *APATERNO*, *AMATERNO*, *NOMBRE(S)*, *CURR*, *ANAC* Y *HINF*. Al aplicar 3FN en dicha tabla se realiza la separación de los atributos *CURR*, *ANAC* Y *HINF* en una nueva tabla llamada *DOCUMENTOS*, quedando los campos de *APATERNO*, *AMATERNO* Y *NOMBRE(S)* como atributo dependientes directamente de la clave principal. Ver figura 1.17.

Empleados			
EmplID	APaterno	AMaterno	Nombre(s)
1	Domínguez	Chávez	José Alfonso
2	Flores	Méndez	Karina
3	Portillo	Zepeda	Claudia Elena

Documentos				
IdDoc	RelEmplID	Curr	ANac	HInf
1	1	Si	Si	Si
2	2	Si	Si	Si
3	3	Si	Si	Si

Teléfonos		
TelId	RelEmplID	Teléfono
1	1	22.22.75.81.71
2	1	4.62.13.91,
3	2	2221.03.29.94
4	2	2.20.71.41,
5	3	3.95.62.27,

Figura 1.17: Ejemplo de las Tablas generadas por 3FN.

Cabe aclarar que pasar a la tercera forma normal, el acceso a la información secundaria no siempre es de manera práctica, debido a que se tendrían que efectuar dos accesos en lugar de uno.

1.4.5.4 Cuarta Forma Normal (4FN) o de Boyce-Code (FNBC).

La cuarta forma normal (4FN) consiste en quitar la redundancia en la información almacenada en algunos atributos, creando para ello tablas que solamente contengan llaves externas.

Para que una tabla “X” alcance la 4FN, se debe verificar que en los atributos no contengan duplicidad de información, en caso de tenerlas se crean dos tablas en donde a una de ellas se le asigne la información de los atributos duplicados asignándoles una clave primaria, la segunda tabla será una modificación de la tabla “X”, donde el atributo que tenía duplicidad se cambiará por su respectivo identificador. Ver figura 1.18.

Empl_Nivel										
RelEmpId	Grado	Est	Const	Egr	Especialidad	Escuela	Cert	Aexam	Tit	Ced
1	<u>Lic</u>	No	No	Si	ING. EN INSTRUMENTACIÓN ELECTRÓNICA	U. VERACRUZANA	No	Si	Si	Si
1	<u>Maes</u>	No	No	Si	EN CS. ING. ELECTRÓNICA	IPN	Si	Si	No	No
2	<u>Lic</u>	No	No	Si	LIC EN INFORMATICA	ITP	Si	Si	Si	Si
2	<u>Maes</u>	No	No	Si	ADMON DE PEQUEÑAS Y MED EMPRESAS	BUAP	No	No	Si	Si
3	<u>Lic</u>	No	No	Si	LIC. EN INFORMÁTICA	ITP	Si	Si	No	Si
4	<u>Lic</u>	No	No	Si	LIC. EN ELECTRÓNICA	BUAP	Si	Si	Si	Si
4	<u>Maes</u>	No	No	Si	ESPECIALIDAD EN ÓPTICA	INAOE	Si	Si	Si	Si
4	<u>Doct</u>	No	Si	No	ESPECIALIDAD EN ÓPTICA	INAOE	No	Si	Si	Si

Grados	
NivID	Grado
1	TSU
2	Licenciatura
3	Maestría
4	Doctorado

Empl_Nivel										
RelEmplId	IdGrad	Est	Const	Egr	Especialidad	Escuela	Cert	Aexam	Tit	Ced
1	<u>2</u>	No	No	Si	ING. EN INSTRUMENTACIÓN ELECTRÓNICA	U. VERACRUZANA	No	Si	Si	Si
1	<u>3</u>	No	No	Si	EN CS. ING. ELECTRÓNICA	IPN	Si	Si	No	No
2	<u>2</u>	No	No	Si	LIC EN INFORMATICA	ITP	Si	Si	Si	Si
2	<u>3</u>	No	No	Si	ADMON DE PEQUEÑAS Y MED EMPRESAS	BUAP	No	No	Si	Si
3	<u>2</u>	No	No	Si	LIC. EN INFORMÁTICA	ITP	Si	Si	No	Si
4	<u>2</u>	No	No	Si	LIC. EN ELECTRÓNICA	BUAP	Si	Si	Si	Si
4	<u>3</u>	No	No	Si	ESPECIALIDAD EN ÓPTICA	INAOE	Si	Si	Si	Si
4	<u>4</u>	No	Si	No	ESPECIALIDAD EN ÓPTICA	INAOE	No	Si	Si	Si

Figura 1.18: Ejemplos de las bases generadas por 4FN o FNBC.

1.4.5.5 Quinta Forma Normal (5FN).

Existe una quinta forma normal (5FN), la cual consiste en reconstruir la tabla original partiendo de las tablas normalizadas. Esto se puede considerar como un proceso inverso, el cual es recomendado para garantizar que no se ha cometido algún error en los procesos de normalizaciones anteriores.

Cabe mencionar que no siempre se lleva a la práctica esta 5FN y más cuando se trabaja con extensas estructuras de datos.

Aunque son posibles los cinco niveles de normalización, la tercera forma normal se considera el máximo nivel necesario para la mayor parte de las aplicaciones.

CAPÍTULO 2.

Diseño del Prototipo.

El objetivo principal de esta sección es el diseño del prototipo del sistema así como el las bases de datos relacionales que se desarrollaron y de las funciones difusas generadas para el sistema. En primer lugar se hace una presentación de la forma en que labora la institución para que con esto, se pueda comenzar el desarrollo del prototipo del sistema. En segundo lugar, con la utilización de UML (Lenguaje Unificado de Modelado), se presenta el diseño y la descripción general de los módulos, que hacen que funcione el sistema para concluir con el bosquejo de las bases de datos a utilizar. Por último se presentan las funciones difusas que se utilizarán para la evaluación de los registros.

2.1 Planteamiento del Problema.

Dentro de las instituciones existen diversas formas de llevar el control de jornada laboral o el tiempo de trabajo de sus empleados, algunas de éstas son:

- Registrarse en bitácoras.
- Registrar su entrada y salida en tarjetas de asistencia.
- Registrar su asistencia con tarjetas de proximidad o con código de barras.
- Registrarse mediante equipos biométricos.

Supongamos que cierta institución educativa tiene como norma interna, que cada uno de sus profesores, deba registrar su entrada y salida de forma digital por cada una de sus asignaturas. Esto nos lleva a considerar que un profesor además de poder tener asignadas más de una materia, cada una de estas asignaturas puede tener una duración de una o más horas consecutivas o por sesión; lo cual nos permite visualizar a que la cantidad de registros diarios que maneja la institución, está ligada conjuntamente con el número de profesores que laboren en ella y la cantidad de materias que cada uno tenga asignado así como la duración de estas asignaturas.

Con esto se pretende evaluar los registros de los profesores, para así poder determinar un grado del cumplimiento que tienen en sus actividades diarias.

2.1.1 Diseño General Empleado – Institución.

El diseño general de la forma en que los profesores deben trabajar con la institución es el que se presenta en la siguiente figura:

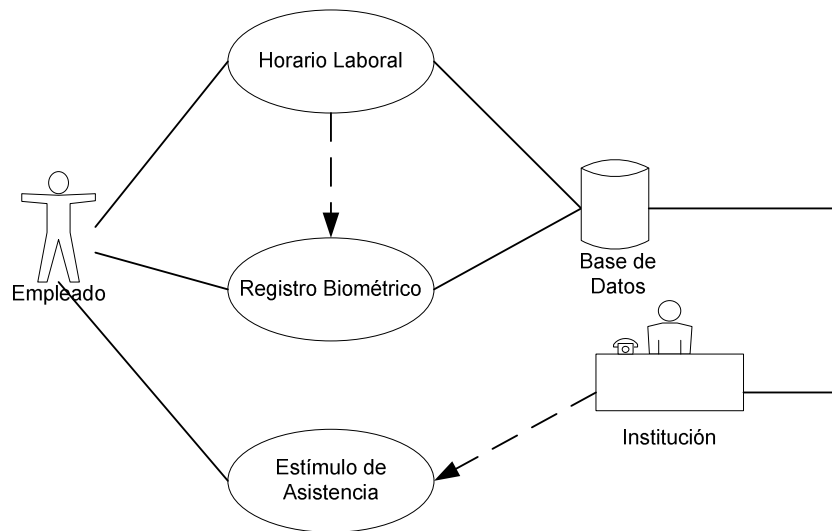


Figura 2. 1: Figura laboral profesor – Institución.

Como se puede ver en la figura 2.1, los registros que efectúe el empleado están en relación con las actividades de trabajo y, para poder ser acreedor a un estímulo de asistencia debe ser evaluado por la institución.

2.2 Diseño General del Sistema.

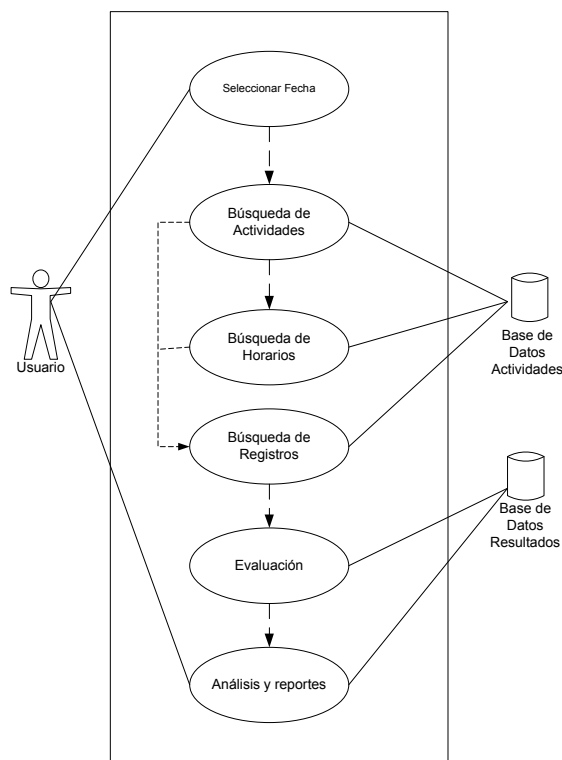


Figura 2. 2: Diagrama General del Sistema

La figura 2.2 muestra que el sistema consta de seis módulos principales para su funcionamiento; mientras que en la figura 2.3 se presenta el esquema a bloques del sistema, en el cual se puede ver la forma en que se relacionan dichos módulos.

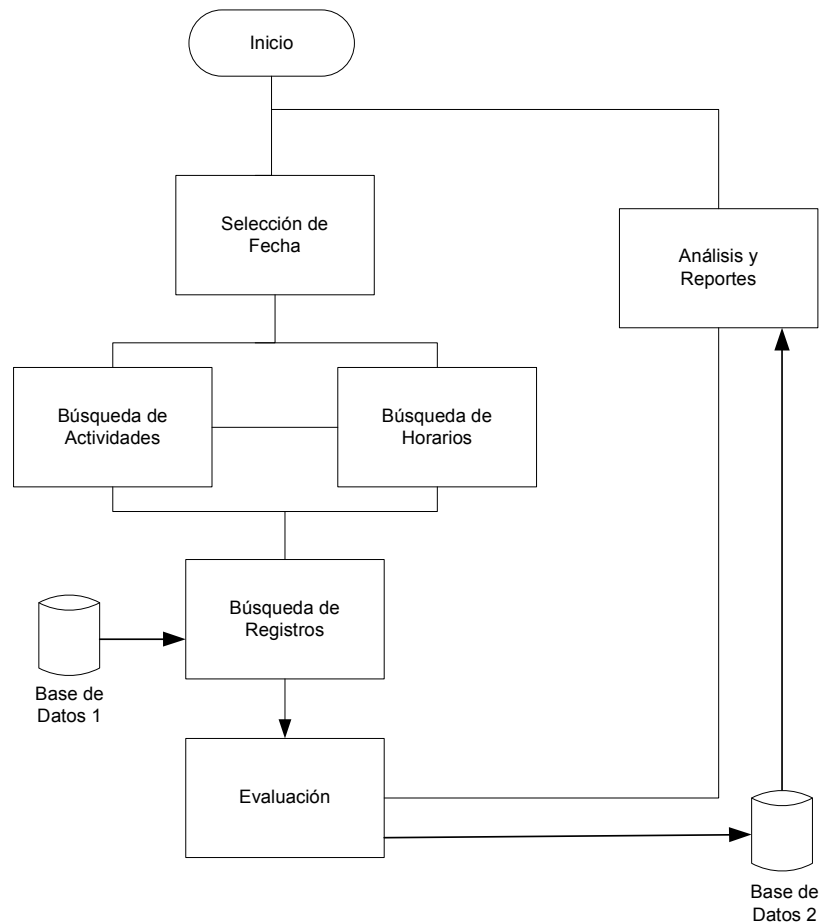


Figura 2. 3: Esquema a Bloques

De inicio, se entrará a la selección de la fecha, una vez que el usuario proporcione dicha información se procederá a la búsqueda de las actividades y de los horarios que se desarrollen en ese día. Así mismo se localizarán los registros correspondientes de entrada y salida de las actividades, para concluir con la evaluación de estos registros en las funciones difusas correspondientes.

La evaluación de registros se efectuará forzosamente una vez obtenidos los registros correspondientes a las actividades y horarios que se desarrollen en dicho día.

El análisis y los reportes se podrán ver después de haber iniciado, de evaluaciones previamente guardadas o después de haber evaluado un día seleccionado.

Los módulos presentados en la figura 2.3 así como la relación que tiene uno con otro también pueden ser representados desde un Esquema de Clases, el cual también nos muestra sus variables dependientes, independientes y procesos internos que lleguen a utilizar, ver figura 2.4.

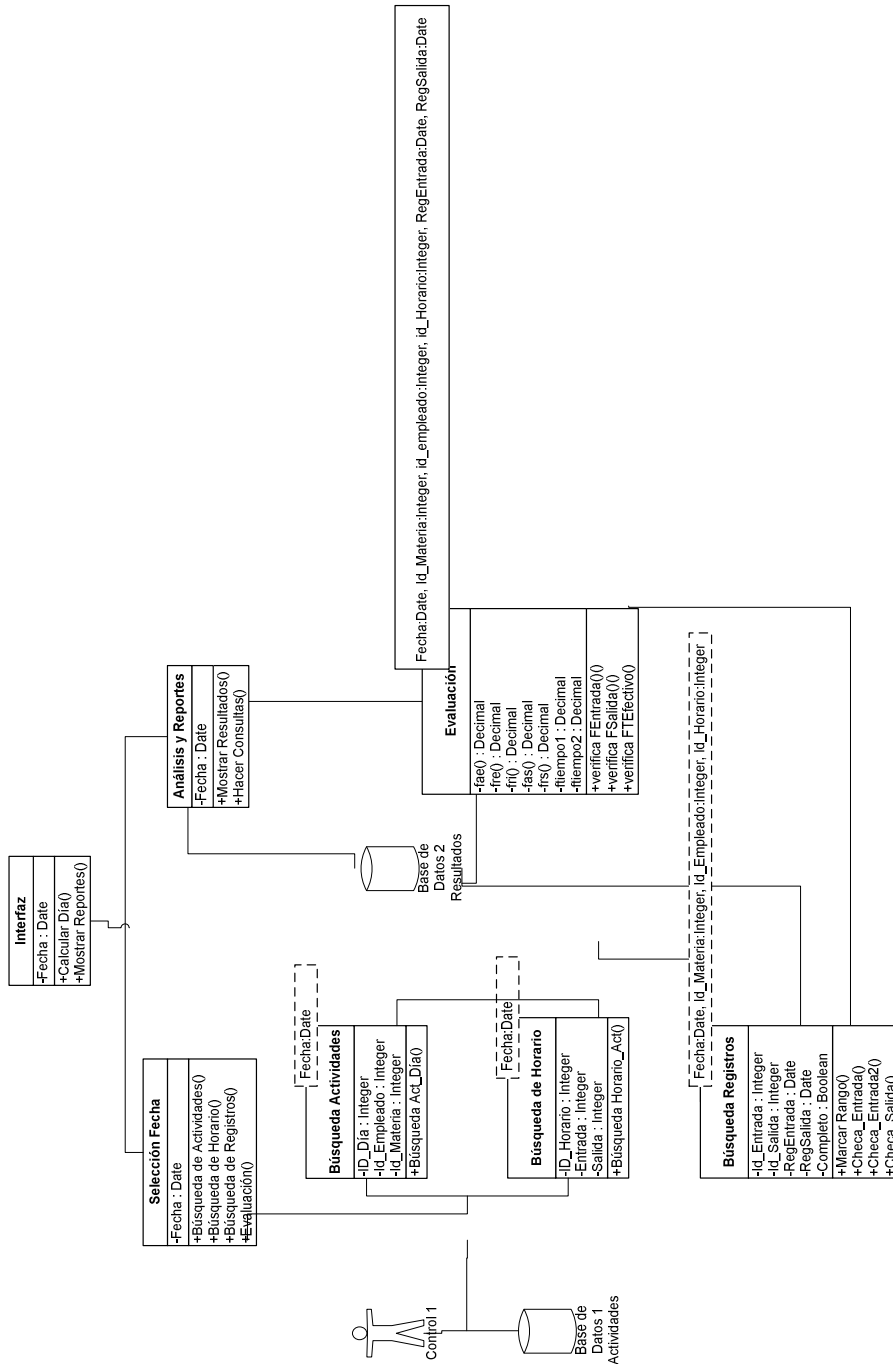


Figura 2. 4: Esquema de Clases

Los seis módulos principales se describen a continuación:

2.2.1 Módulo Selección de Fecha.

Este módulo proporcionará condiciones iniciales al sistema, así como una interfaz gráfica para que el usuario pueda seleccionar un día a evaluar. El esquema general de este módulo se puede observar en la figura 2.5.

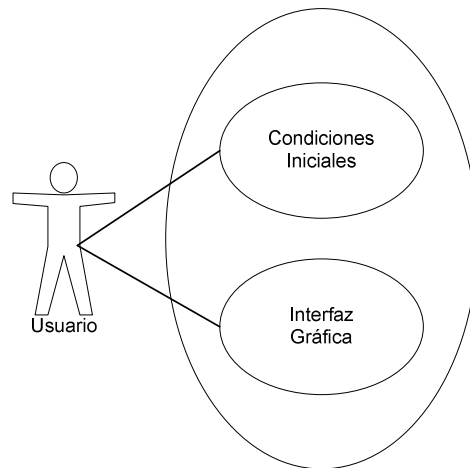


Figura 2. 5: Módulo Selección de Fecha.

La función principal del módulo es proporcionar el parámetro del día del cual se evaluarán los registros en las funciones difusas correspondientes.

2.2.2 Módulo Búsqueda de Actividades.

Es el encargado de buscar cada una de las actividades que se lleven a cabo en el día seleccionado por el usuario. Dentro de las funciones internas deberá obtener un parámetro que se utilizará en el módulo de evaluaciones, dicho parámetro es la clave del empleado.

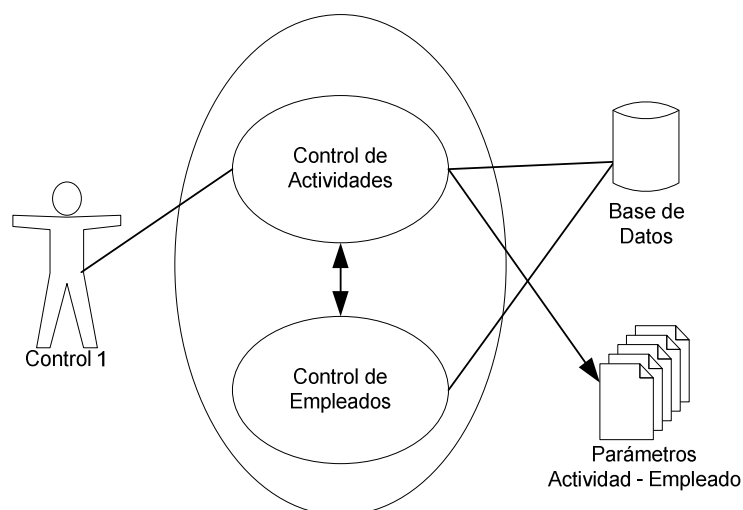


Figura 2. 6: Módulo Búsqueda de Actividades

Observe que en la figura 2.6, la función principal del módulo es obtener las actividades y los empleados que laboren en esa fecha, los cuales serán utilizados como parámetros en los módulos subsecuentes.

2.2.3 Módulo Búsqueda de Horarios.

La búsqueda de los horarios comienza una vez que se tienen a los empleados y a las actividades que se desarrollan en la fecha solicitada. En la figura 2.7 se presenta el esquema de trabajo del módulo búsqueda de horarios.

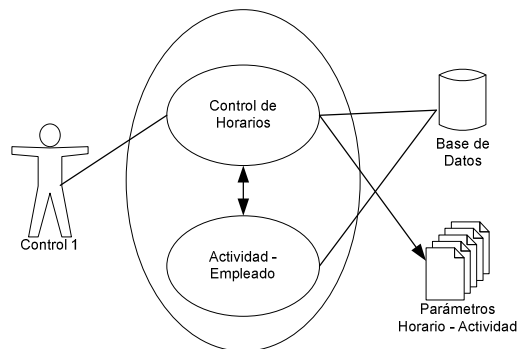


Figura 2. 7: Módulo Búsqueda de Horarios.

Su finalidad es poder obtener los horarios de estas actividades para poder delimitar a los posibles registros en el siguiente módulo.

2.2.4 Módulo Búsqueda de Registros.

Este módulo recibe los parámetros de la Fecha, la Clave del Empleado, la Actividad a desarrollar y el Horario de esta última. La figura 2.8 muestra el funcionamiento del módulo búsqueda de registros, en donde todos estos parámetros de entrada, le permitirán determinar cuales registros de todo el conjunto existente en la base de datos, son los correspondientes para la entrada y salida de las labores.

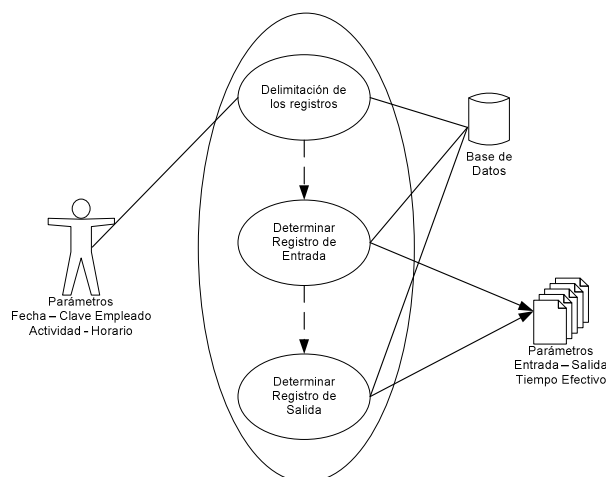


Figura 2. 8: Módulo Búsqueda de Registros.

2.2.5 Módulo de Evaluación.

Los parámetros de entrada para este módulo son el registro de entrada, el registro de salida y el tiempo efectivo parcial que haya cubierto el empleado en sus actividades. Estos tres datos serán evaluados en sus funciones difusas correspondientes para determinar los resultados del día. este funcionamiento es presentado en la figura siguiente.

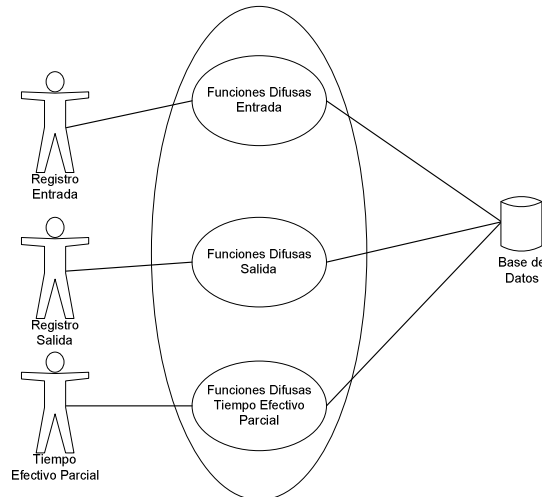


Figura 2. 9: Módulo de Evaluación.

Todos los resultados son resguardados en una base de datos para el análisis y realización de Reportes.

2.2.6 Módulo de Análisis y Reportes.

El módulo de análisis y reportes es el encargado de dar un análisis de la información generada y permitirá al usuario generar los reportes correspondientes. Los reportes generados podrán filtrarse por fechas, etiquetas difusas y clave del empleado. El esquema general de este módulo es presentado en la figura a continuación.

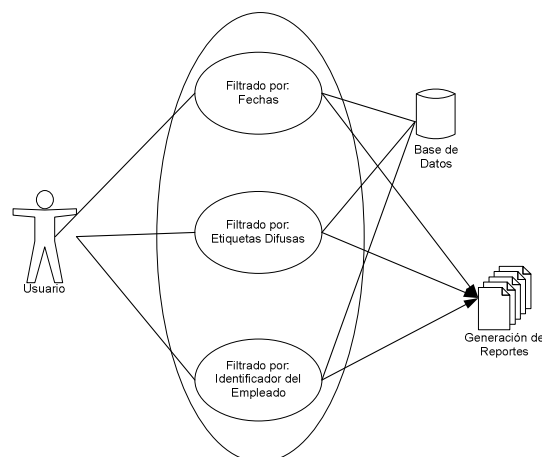


Figura 2. 10: Módulo de Análisis y Reportes.

2.3 Diseño General de las Bases de Datos Relacionales.

Como se pudo observar en la sección anterior, el sistema utilizará dos bases de datos para trabajar: Control de Profesores y Control de Trabajo, que se describen a continuación.

2.3.1 Base de Datos Relacional: Control de Profesores.

Esta primera base de datos relacional contiene información que la institución utiliza para control del profesorado como son: los datos generales del empleado, las carreras en la que labora, las actividades que realiza, el horario en el que realiza sus jornadas, etc.

El diseño de esta base de datos relacional se puede representar con los siguientes esquemas:

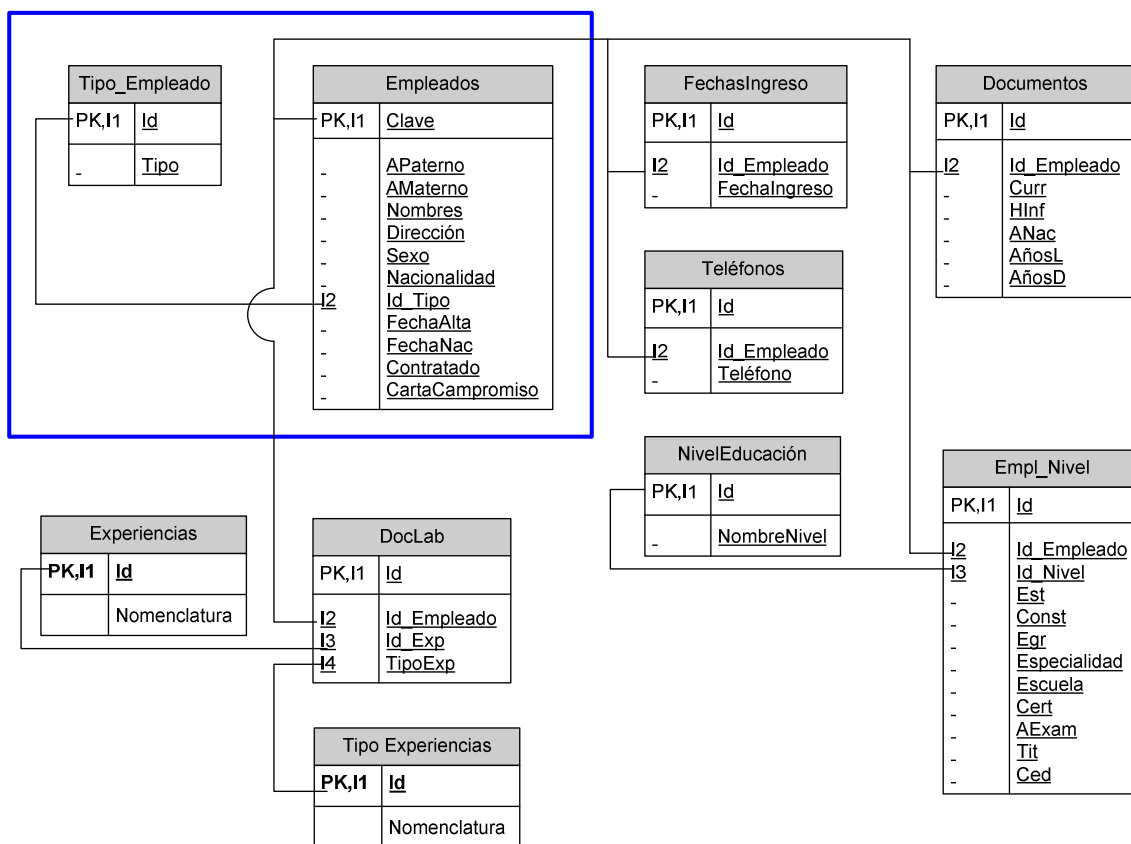


Figura 2. 11: Esquema Base de Datos Relacional Control de Profesores (Primera Parte)

La figura 2.11 muestra la primera parte de la base de datos, esta parte es solamente la información general de los empleados de la institución. El Recuadro señala las tablas de unión con la figura 2.12.

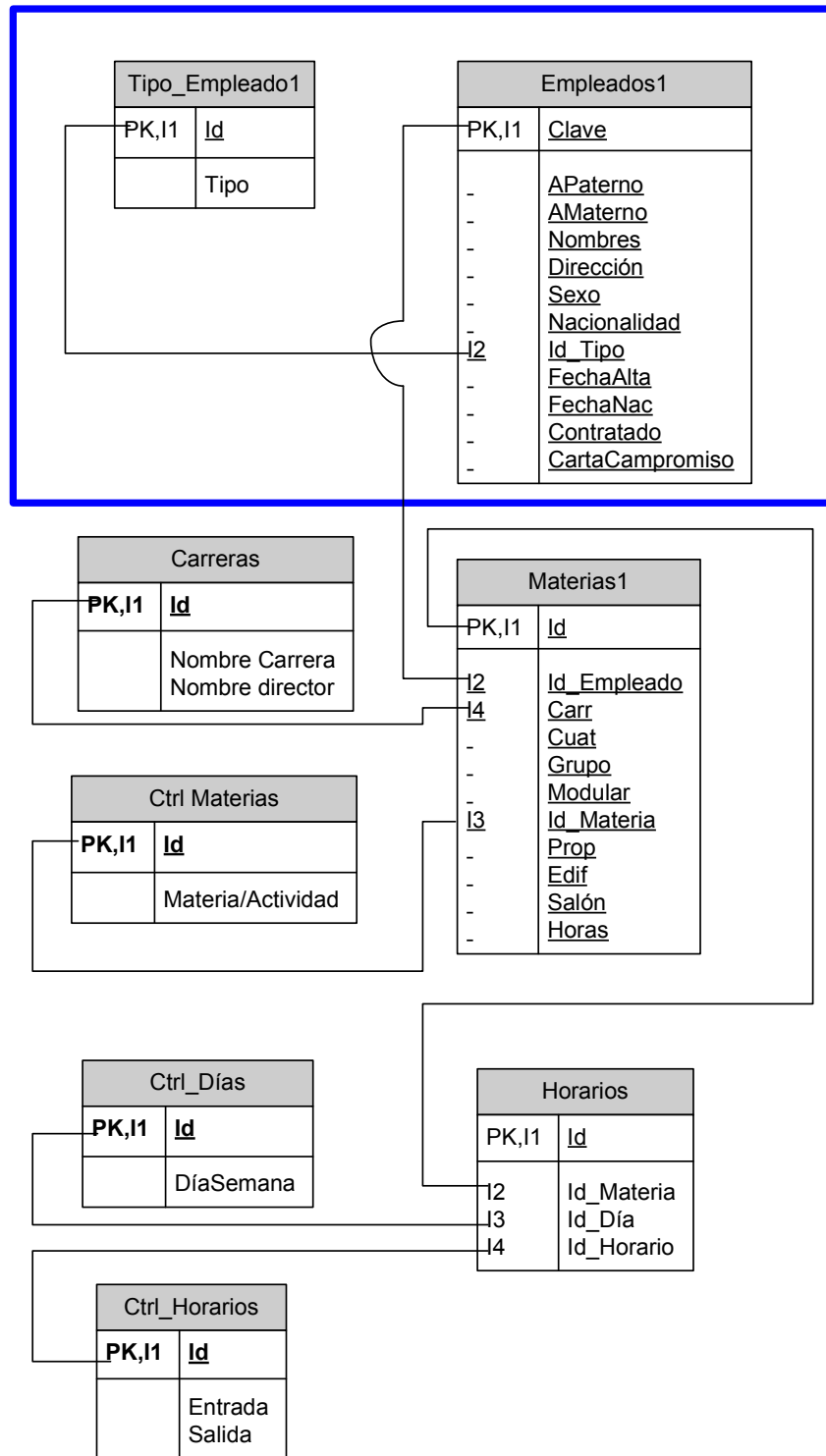


Figura 2. 12: Esquema Base de Datos Relacional Control de Profesores (Segunda Parte)

En la figura 2.12 se muestra el complemento de la primera base de datos que utilizará el sistema. Como se puede apreciar en esta figura, la Tabla llamada Horarios es la que permitirá trabajar a los dos módulos de búsqueda, el de actividades y el de horarios.

2.3.2 Base de Datos Relacional: Control de Trabajo.

En esta base de datos, se almacenarán los registros biométricos de los empleados así como las evaluaciones que se les efectúen a dichos registros.

El esquema general de esta base de datos relacional se muestra en la siguiente figura.

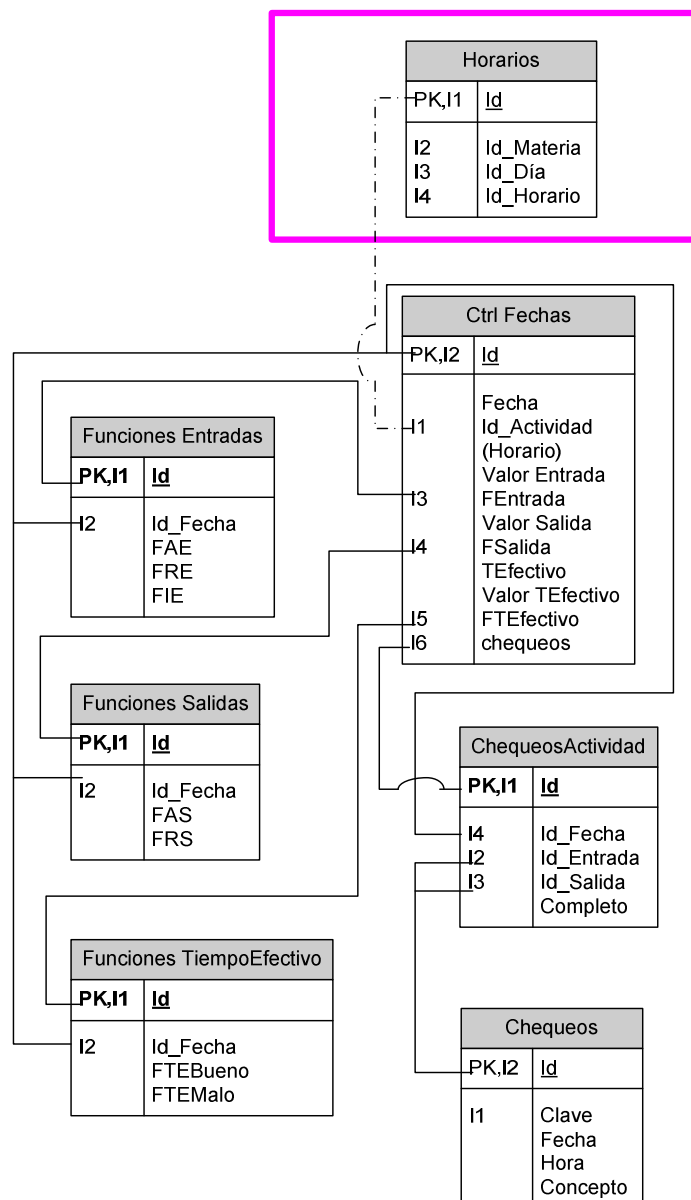


Figura 2. 13: Esquema Base de Datos Relacional Control de Trabajo

Como se puede observar en la figura 2.13, esta base de datos tiene relación con la Tabla Horarios de la base de datos Control de Profesores.

Los resultados de las evaluaciones a los registros de entrada, salida y tiempo efectivo se almacenarán en sus correspondientes tablas.

Las relaciones entre las tablas <Funciones Entradas>, <Funciones Salidas>, <Funciones TiempoEfectivo> y <ChequesActividad> con la Tabla llamada <Ctrl Fechas> es de 1 a 1.

En el siguiente epígrafe se describirán las funciones difusas que utiliza el sistema.

2.4 Ajuste de Funciones Difusas para el Sistema.

Se plantean tres categorías para las funciones difusas en base a los lineamientos internos de la institución, para la aplicación del sistema, estas categorías permiten una clasificación de asistencia del personal académico basándose en los registros de entrada y salida en cada uno de ellos.

2.4.1 Funciones difusas para evaluar el Registro de Entrada:

En esta sección, hablaremos de tres funciones para la evaluación del registro de entrada, estas son: para evaluar en el rango establecido por la institución como asistencia, asistencia con retardo e inasistencia.

2.4.1.1 Función Difusa de Asistencia para el registro de entrada.

La función difusa de asistencia que se considera es una del tipo L, cuyo dominio es $[a-10, \infty)$ donde a es la hora de inicio de su actividad. Tal función está definida como:

$$L(x, a, m) = \begin{cases} 1 & x \leq a \\ \frac{m-x}{m-a} & a < x < m \\ 0 & x \geq m \end{cases}$$

Cuya gráfica se describe la figura 2.14.

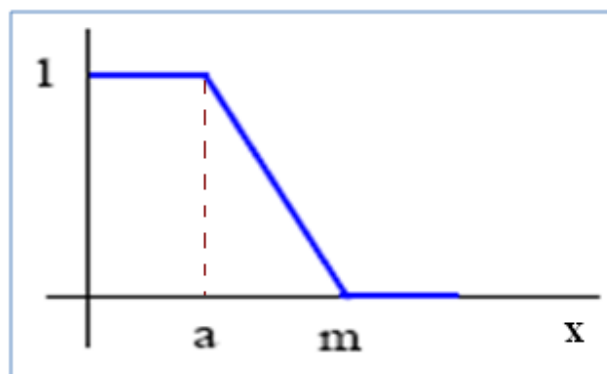


Figura 2. 14: Función Difusa de Asistencia para el Registro de Entrada.

La justificación que se da para la utilización de esta función es porque el profesor puede checar unos minutos antes de la hora en que empieza la sesión hasta el tiempo de tolerancia, que son 10 minutos después.

Cabe aclarar que el personal puede checar con un retardo es por eso que a continuación construiremos una función difusa para evaluar tal retardo.

2.4.1.2 Función Difusa de Retardo

La función difusa que medirá el retardo es de tipo trapezoidal, cuyo dominio está determinado por el intervalo $(-\infty, \infty)$ y está definida por:

$$\text{trapezio}(x, b, m, c, d) = \begin{cases} 0 & x \leq b \quad \text{o} \quad x \geq d \\ \frac{x-b}{m-b} & b < x < m \\ 1 & m \leq x \leq c \\ \frac{d-x}{d-c} & c < x < d \end{cases}$$

La figura 2.14 muestra la gráfica de la función anterior.

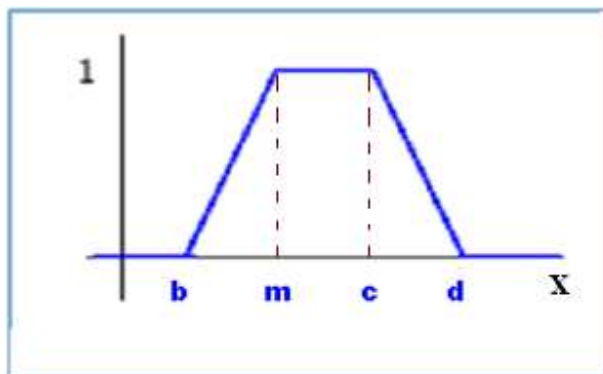


Figura 2. 15: Función Difusa de Retardo para el Registro de Entrada.

Observe que esta función tiene valores significativos entre el intervalo (b, d) es decir valores distintos de 0, donde se mide el grado de retardo. Cabe aclarar que en el intervalo $[m, c]$ es donde, en la práctica, se le permite al profesor checar con su entrada con retardo. Después de esto se le considera como falta. Sin embargo se utiliza una tercera función difusa que mide la inasistencia.

2.4.1.3 Función Difusa de Falta o Inasistencia.

Esta función de inasistencia es de tipo gamma, cuyo dominio esta determinado por el intervalo $(-\infty, \infty)$ y su regla de definición es:

$$\Gamma(x, c, d) = \begin{cases} 0 & x \leq c \\ \frac{x-c}{d-c} & c < x < d \\ 1 & d \leq x \end{cases}$$

Y su gráfica es representada en la figura 2.16.

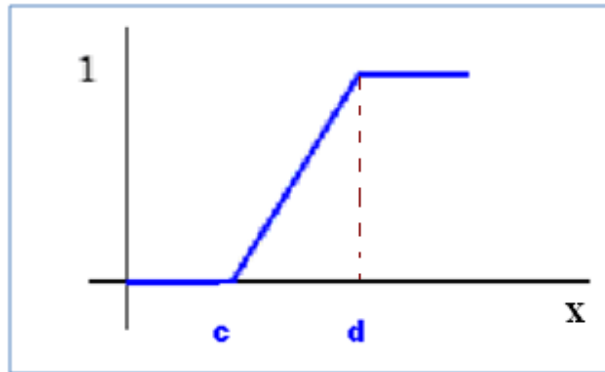


Figura 2. 16: Función Difusa de Inasistencia para el Registro de Entrada.

Esta función tendrá valores significativos entre el intervalo (c,d) , estos valores nos permiten determinar el grado en que el profesor llega después del tiempo establecido para el retardo.

A continuación se presentan las funciones difusas que evalúan el registro de salida.

2.4.2 Funciones difusas para evaluar el Registro de Salida:

Siguiendo la misma tonalidad que en el registro de entrada, aquí se presentan dos funciones que evaluarán la salida en tiempo y la salida en retiro previo a la tolerancia establecida por la institución.

2.4.2.1 Función Difusa de Salida en Tiempo.

La función difusa que se considera es una del tipo gamma, cuyo dominio es $(-\infty, p+10]$ donde p es la hora en que debe culminar su actividad. Tal función está definida como:

$$\Gamma(x, p, q) = \begin{cases} 0 & x \leq q \\ \frac{x-q}{p-q} & q < x < p \\ 1 & p \leq x \end{cases}$$

Y su gráfica es representada en la figura 2.17.

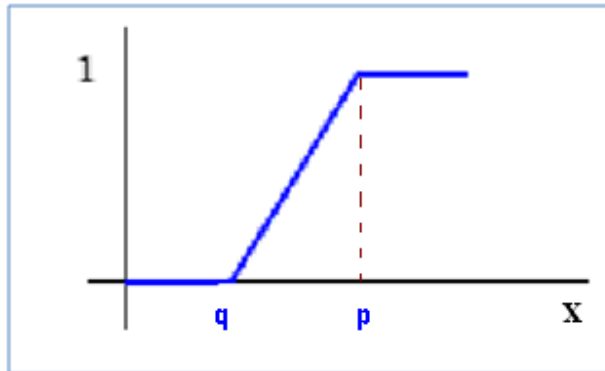


Figura 2. 17: Función Difusa de Inasistencia para el Registro de Salida.

La justificación que se da para la utilización de esta función es porque el profesor puede checar tomando 10 minutos de tolerancia previos a su hora de salida hasta unos minutos después de esta.

Cabe aclarar que el personal puede checar antes de ese tiempo de tolerancia, por esa razón a continuación construiremos una función difusa para evaluar ese registro previo.

2.4.2.2 Función Difusa de Retiro Previo a la tolerancia

La función difusa que medirá el registro previo a la tolerancia de salida es de tipo L, cuyo dominio está determinado por el intervalo $(-\infty, \infty)$ y está definida por:

$$L(x, p, q) = \begin{cases} 1 & x \leq q \\ \frac{p-x}{p-q} & q < x < p \\ 0 & p \leq x \end{cases}$$

La figura 2.18 es la gráfica de la función anterior.

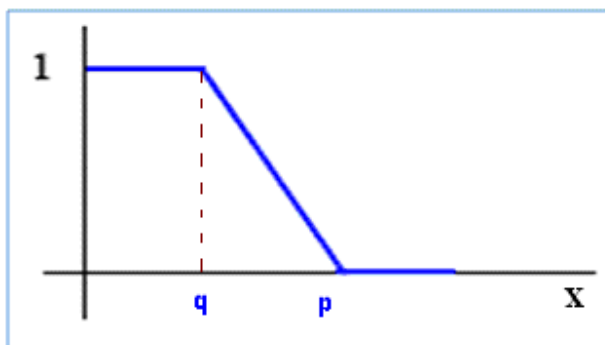


Figura 2. 18: Función Difusa de Retiro Previo para el Registro de Salida.

Observe que esta función tiene valores significativos entre el intervalo $(-\infty, p)$ es decir valores distintos de 0, donde se mide el grado en que se retira el profesor antes del tiempo establecido como tolerancia.

Una vez descritas las funciones utilizadas para la evaluación de los registros de entrada y salida, se realizará lo propio para el tiempo efectivo parcial.

2.4.3 Funciones difusas para evaluar el Tiempo Efectivo Parcial:

En esta sección presentaremos dos funciones que evaluarán el tiempo parcial de las actividades de los empleados, estas funciones nos determinarán en que grado cumple con un tiempo establecido como considerable y en que grado no.

2.4.3.1 Función Difusa de Tiempo Parcial Considerable.

Esta función difusa de tiempo efectivo es de tipo sigma, cuyo dominio está determinado por el intervalo $(-\infty, \infty)$ y está definida por:

$$S(x, a, b, m) = \begin{cases} 0 & x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2 & a < x \leq m \\ 1 - 2\left(\frac{x-b}{b-a}\right)^2 & m < x < b \\ 1 & b \leq x \end{cases}$$

La figura 2.19 representa la gráfica de la función anterior.

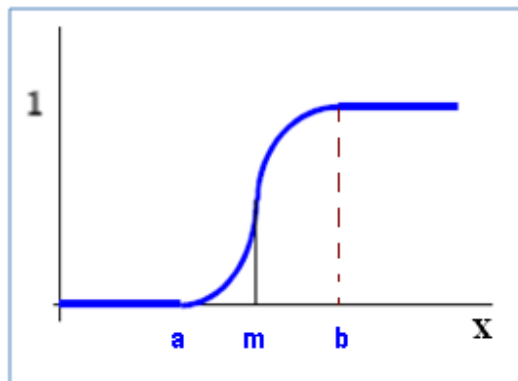


Figura 2. 19: Función Difusa de Tiempo Considerable para el Tiempo Efectivo Parcial

Observe que los valores significativos se encuentran en el intervalo (a, ∞) , donde después de a se comienza a medir la jornada parcial de trabajo y, cabe recalcar que después de b el profesor está cumpliendo con un tiempo establecido por la institución.

2.4.3.2 Función Difusa de Tiempo Malo.

Esta función difusa es de tipo Z, esto es, la función inversa de la función anteriormente vista. Su dominio está determinado por el intervalo $(-\infty, \infty)$ y está definida por:

$$Z(x, a, b, m) = \begin{cases} 1 & x \leq a \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2 & a < x \leq m \\ 2\left(\frac{x-b}{b-a}\right)^2 & m < x < b \\ 0 & b \leq x \end{cases}$$

La siguiente figura presenta la gráfica de tal función.

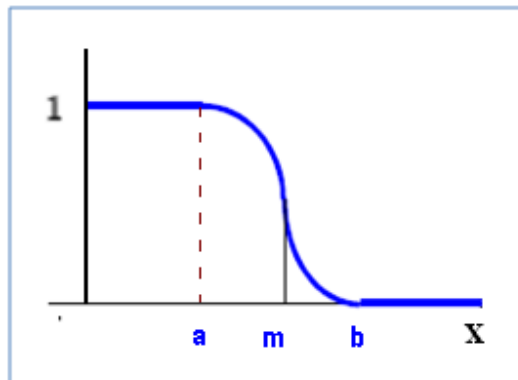


Figura 2. 20: Función Difusa de Tiempo Malo para el Tiempo Efectivo Parcial.

Esta función proporciona valores significativos en el intervalo $(-\infty, b)$ que nos permiten determinar el grado en que el profesor se retira antes del tiempo marcado como considerable para su actividad.

Ya esquematizados los módulos, las clases del sistema, las bases de datos relacionales y las funciones difusas a utilizar en este proyecto, se presenta la implementación del sistema.

CAPÍTULO 3.

Implementación.

En este capítulo se presenta la implementación del sistema, el cual abarca tanto el pseudocódigo como el código fuente de los módulos que lo conforman así como sus procedimientos que estos contienen. El sistema fue desarrollado en un equipo Pentium® 4 a 2.60 GHz con un sistema operativo Windows XP, el lenguaje de programación utilizado fue Microsoft Visual Basic 6 con SP6; para la elaboración de diagramas del diseño fue empleado Microsoft Visio y para la creación de gráficas de las funciones difusas se utilizó MathLab 6.5.

3.1 Programación de los módulos.

Para la creación de los módulos se programó en Visual Basic, por lo que se utilizaron algunos componentes de este lenguaje para el diseño, librerías para la conexión con las bases de datos y se crearon nuevos tipos de variables para sus procesos.

3.1.1 Variables Generales de Control.

Dentro de las variables que se utilizaron como control interno en diversos de los módulos, tenemos a la variable PROFESOR, esta variable es de un tipo personalizado llamado PROFE, el cual está conformado por varias variables. Esta variable almacena la clave del profesor, los registros de una actividad a evaluar y, los resultados de esas evaluaciones. El tipo PROFE se describe a continuación.

Variable	Tipo de la Variable	Descripción
Clave	Entera	Almacena el identificador del empleado.
Reg_entrada	Hora	Almacena el registro correspondiente a la entrada de la actividad.
Id_Entrada	Entera	Almacena el identificador del registro de entrada.
Reg_salida	Hora	Almacena el registro correspondiente a la salida de la actividad.

Variable	Tipo de la Variable	Descripción
Id_salida	Entera	Almacena el identificador del registro de salida.
Completo	Booleana	Bandera que permite identificar cuando existen los dos registros para la actividad.
ValorEntrada, ValorSalida, ValorFTEfectivo	Etiqueta	Etiquetas que almacenarán los resultados de las evaluaciones.
fAE, fRE, fIE	Real	Almacenan los resultados de las evaluaciones en las funciones difusas para el registro de entrada.
fAS, fRS	Real	Almacenan los resultados de las evaluaciones en las funciones difusas para el registro de salida.
Fbueno, Fmalo	Real	Almacenan los resultados de las evaluaciones en las funciones difusas para el tiempo efectivo parcial.
Tiempoefectivo	Hora	Almacena el tiempo efectivo parcial de la actividad.

Una vez presentadas las variables comunes entre los módulos del sistema se describe la implementación de cada uno de sus módulos:

3.1.2 Módulo Selección de Fecha.

Este módulo le permite al usuario seleccionar el día a evaluar; también el usuario podrá seleccionar la base de datos del control de profesores que desee utilizar así como la base de datos Control de Trabajo en la que desee guardar los resultados de la evaluación. La interfaz gráfica de este módulo se presenta en la figura 3.1.

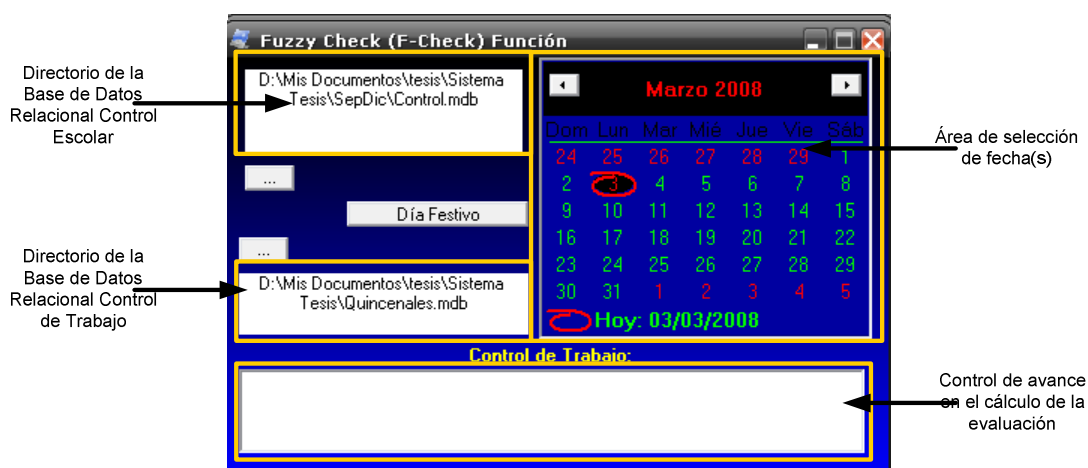


Figura 3. 1: Pantalla de la Selección de Fecha(s) a Evaluar.

Como se puede ver en la figura anterior, la interfaz gráfica es sencilla, permitiendo que el usuario trabaje dando un click en el día a evaluar o mantenga sostenido el click seleccionando un rango de fechas. Una vez que usuario ha hecho su selección, la secuencia de iteración entre los módulos internos aparecerá en el área de control de Trabajo en forma de lista.

Para el área de selección de fecha(s) se utiliza un componente de visual basic llamado MothView. De este componente se emplean los comandos DayOfWeek, SelStart, SelEnd y Value.

- a. DayOfWeek: Nos presenta que día de la semana es el correspondiente a la fecha seleccionada. Este comando nos proporciona el acceso a la tabla Ctrl_Días de la Base de Datos de Control de Profesores, presentada en la figura 2.11. Esto se logra ya que proporciona un valor numérico que va de 1 a 7 que son equivalentes a los días de la semana en orden de domingo a sábado.
- b. SelStart: Este comando nos indica el día en que comienza la evaluación del rango de fechas, proporcionado por el usuario.
- c. SelEnd: Nos indica el día en que termina la evaluación del rango de fechas, proporcionado por el usuario.
- d. Value: Proporciona la fecha de referencia.

Utilizando estos cuatro comandos, procedemos a presentar el pseudocódigo y el código del módulo.

3.1.2.1 Pseudocódigo.

A continuación se describen las variables de control del módulo en la tabla 3.1 así como las actividades y funciones que realiza.

Variable	Tipo de la Variable	Descripción
FechaINICIO	Fecha	Almacena la fecha de inicio de la evaluación.
FechaFINAL	Fecha	Almacena la fecha de término de la evaluación.
DíaSEMANA	Fecha	Almacena el día de la semana correspondiente a la fecha de referencia.
ClaveEmpleado	Entera	Almacena el identificador del empleado.
Entrada	Entera	Almacena la hora de entrada de la actividad.
Salida	Entera	Almacena la hora de salida de la actividad.

Tabla 3. 1: Tabla Variables de Control para el Módulo Selección de Fecha.

3.1.2.1.1 Actividades y Funciones a Realizar.

Las actividades y funciones a realizar en este módulo son las siguientes:

Al seleccionar un rango de fechas, hay que hacer la asignación de la fecha de inicio de la evaluación solicitada por el usuario a la variable FechaINICIO así como su valor correspondiente para la variable FechaFINAL.

Teniendo estos dos parámetros, se tiene que recorrer ese rango de fechas, verificando a que día de la semana corresponden, para esto se hace una selección de casos que se describe en la tabla siguiente.

Caso:	Actividad a Realizar:
Caso 1:	Asigna a DíaSEMANA el identificador correspondiente a la fecha, que es domingo.
Caso 2:	Asigna a DíaSEMANA el identificador correspondiente a la fecha, que es lunes.
...	...
Caso 7:	Asigna a DíaSEMANA el identificador correspondiente a la fecha, que es sábado.

Tabla 3. 2: Selección de Casos para el comando MonthView.DayOfWeek.

Una vez teniendo el día correspondiente a la semana, se requiere hacer una consulta en la tabla CTRL_DIAS para determinar cual es el identificador interno correspondiente a ese día. Esta consulta es identificada como CONSULTA1 que se describe de la siguiente forma:

- Seleccionar <ID> de CTRL_DIAS donde <diasemana> = DiaSEMANA.

Donde los campos de la tabla están identificados entre < > y, la referencia del filtro la proporciona la variable DíaSEMANA.

Teniendo valores en esta CONSULTA1, se efectúa una siguiente consulta en la tabla HORARIOS, con el fin de determinar que actividades y horarios se tienen en ese día. Esta consulta es identificada como CONSULTA2 que se describe a continuación.

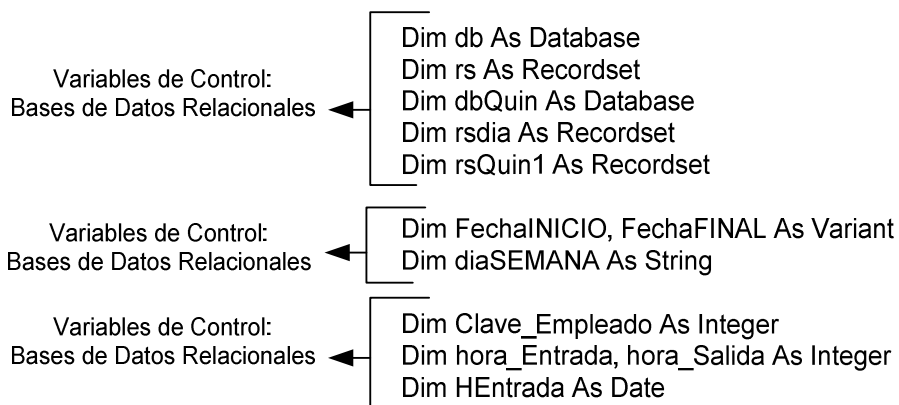
- Seleccionar TODO de HORARIOS donde <Id_día> esté en los resultados de la CONSULTA1.

Como se puede observar, se filtra exclusivamente los resultados que coinciden con la CONSULTA1. Teniendo estos resultados, se procede de la siguiente manera: Se inicializa la variable ClaveEmpleado, después se realiza la búsqueda de las actividades correspondientes al día para continuar con la habilitación de las variables Entrada y Salida, una vez que se han inicializado estas variables se efectúa la búsqueda de los horarios correspondientes a las actividades, seguido de la búsqueda de los registros de entrada – salida de estos horarios y, se concluye con la evaluación de los registros obtenidos.

3.1.2.2 Código Fuente.

Una que se describió el funcionamiento del módulo de selección de fechas, se presenta el código fuente correspondiente a la creación de las variables y de la estructura del módulo.

3.1.2.2.1 Creación de Variables.



3.1.2.2.2 Estructura del Módulo.

```

While FechaINICIO <= FechaFINAL
  Set rsQuin1 = dbQuin.OpenRecordset("Select * From Ctrl_Fechas where str(Fecha) = " & FechaINICIO & "'")
  If rsQuin1.AbsolutePosition <> -1 Then
    If MsgBox("Desea Eliminar los datos existentes del Día: " & MonthView1.Value, vbYesNo, "Borrar Datos") = vbYes Then
      Do Until rsQuin1.EOF
        rsQuin1.Delete
        rsQuin1.MoveNext
      Loop
    End If
  End If

  MonthView1.Value = FechaINICIO
  If MsgBox("Desea Calcular el día: " & MonthView1.Value, vbYesNo, "Calcular Fecha?") = vbNo Then
    GoTo nocalcula
  End If

  List1.AddItem ("Evaluación del día: " & MonthView1.Value)
  Select Case MonthView1.DayOfWeek
    Case 1: diaSEMANA = "DOMINGO"
    Case 2: diaSEMANA = "LUNES"
    Case 3: diaSEMANA = "MARTES"
    Case 4: diaSEMANA = "MIERCOLES"
    Case 5: diaSEMANA = "JUEVES"
    Case 6: diaSEMANA = "VIERNES"
    Case 7: diaSEMANA = "SÁBADO"
  End Select

  Set rsdia = db.OpenRecordset("Select * from Ctrl_Dias where Ucase(DiaSemana) = " & diaSEMANA & "'")
  List1.AddItem ("Determinando día de la Semana: " & diaSEMANA)
  If rsdia.AbsolutePosition <> -1 Then
    Set rs = db.OpenRecordset("Select * From Horarios where Id_Día = " & rsdia.Fields("Id") & " order by Id_Horario")
    List1.AddItem ("Búsqueda de Actividades/Horarios del día")
    Do Until rs.EOF
      Clave_Empleado = 0
      Call Busqueda_de_Actividades(db, rs.Fields("Id_Materia"), Clave_Empleado) 'Regresa Clave Empleado
      hora_Entrada = 0
      hora_Salida = 0
      Call Busqueda_de_Horario(db, rs.Fields("Id_Horario"), hora_Entrada, hora_Salida)
      Call Busqueda_de_Registros(dbQuin, FechaINICIO, Clave_Empleado, hora_Entrada, hora_Salida)
      If profee.completo = True Then
        HEntrada = hora_Entrada & ":00"
        Call Evalua(profee, HEntrada, hora_Salida - hora_Entrada, 0)
        Call Funciones(profee, hora_Salida - hora_Entrada)
      End If
      Call llena_tablas(dbQuin, rsQuin1, rs.Fields("Id_Materia"), FechaINICIO)
    rs.MoveNext
  Loop
  rs.Close
  End If
  rsdia.Close
nocalcula:
FechaINICIO = FechaINICIO + 1
Wend

```

3.1.3 Módulo Búsqueda de Actividades.

Este módulo permite obtener las actividades desarrolladas en el día a evaluar. Cabe mencionar que este módulo es dependiente y secuencial del módulo Selección de fechas cuyo fin es proporcionar un parámetro base, este parámetro es el identificador del empleado.

A continuación se procede a presentar el pseudocódigo y el código del módulo.

3.1.3.1 Pseudocódigo.

Para el funcionamiento de este módulo se describen en la tabla 3.3 las variables de entrada y, en la tabla 3.4 las variables de salida que se utilizan.

Variable	Tipo de la Variable	Descripción
Control Escolar	Base de Datos Relacional	Almacena la referencia de la base de datos relacional "Control de Profesores"
Id_Materia	Entera	Almacena el identificador de la actividad que se desarrolla. Este valor es proporcionado por la CONSULTA2, ver sección 3.1.2.1.1

Tabla 3. 3: Variables de Entrada del Módulo Búsqueda de Actividades.

Variable	Tipo de la Variable	Descripción
ClaveEmpleado	Entera	Almacena el identificador del empleado.

Tabla 3. 4: Variables de Salida del Módulo Búsqueda de Actividades.

3.1.3.1.1 Actividades y Funciones a Realizar.

Los procedimientos a realizar en este módulo para cada una de las actividades que se lleven a cabo en el día seleccionado por el usuario, son los siguientes:

Se requiere hacer una consulta en la tabla MATERIAS, para determinar cual es el identificador del empleado que efectúa dicha actividad. Esta consulta es identificada como CONSULTA3 que se describe de la siguiente forma:

- Seleccionar <ID_EMPLEADO> de Materias donde <id> esté en los resultados de *Consulta2* <id_Materia>.

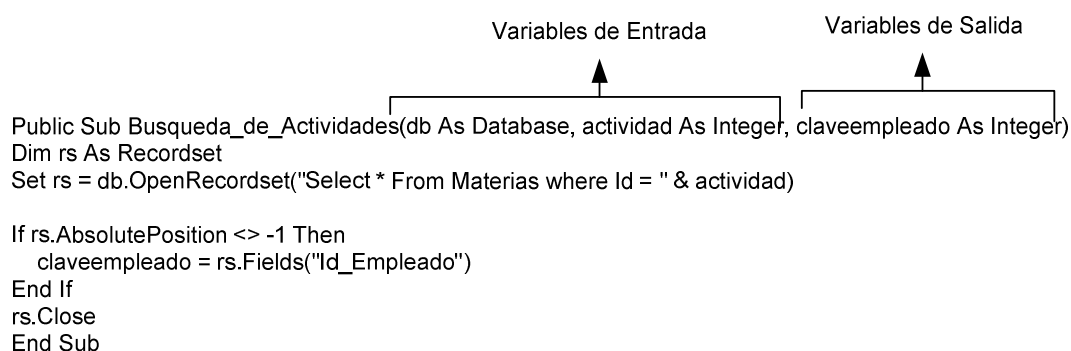
Donde los campos de la tabla están identificados entre < >, y la referencia del filtro la proporciona la variable Id_Materia proporcionada por la CONSULTA2.

Una vez efectuada la CONSULTA3, se verifica que no esté vacía, en caso de ser verdadero se asigna el identificador del empleado a la variable de salida ClaveEmpleado.

3.1.3.2 Código Fuente.

Una vez que se describió el funcionamiento del módulo de búsqueda de actividades, se presenta el código fuente correspondiente a la estructura del módulo.

3.1.3.2.1 Estructura del Módulo.



3.1.4 Módulo Búsqueda de Horarios.

Este módulo tiene como función obtener los horarios de las actividades del día seleccionado, estos horarios delimitarán a los posibles registros en el siguiente módulo.

A continuación se procede a presentar el pseudocódigo y el código del módulo.

3.1.4.1 Pseudocódigo.

La tabla 3.5 presenta las variables de entrada que requiere este módulo, mientras que la tabla 3.6 presenta las variables de salida del mismo.

Variable	Tipo de la Variable	Descripción
Control Escolar	Base de Datos Relacional	Almacena la referencia de la base de datos relacional "Control de Profesores"
Id_Horario	Entera	Almacena el identificador del horario de la actividad que se desarrolla. Este valor es proporcionado por la CONSULTA2, ver sección 3.1.2.1.1

Tabla 3. 5: Variables de Entrada para el Módulo Búsqueda de Horarios

Variable	Tipo de la Variable	Descripción
Entrada	Entera	Almacena la hora de entrada de la actividad.
Salida	Entera	Almacena la hora de salida de la actividad.

Tabla 3. 6: Variables de Salida del Módulo Búsqueda de Horarios.

3.1.4.1.1 Actividades y Funciones a Realizar.

Teniendo las variables de entrada como referencia, se requiere hacer una consulta en la tabla CTRL_HORARIOS para determinar cual es la entrada y salida en que se efectúa dicha actividad. Esta consulta es identificada como CONSULTA4 que se describe de la siguiente forma:

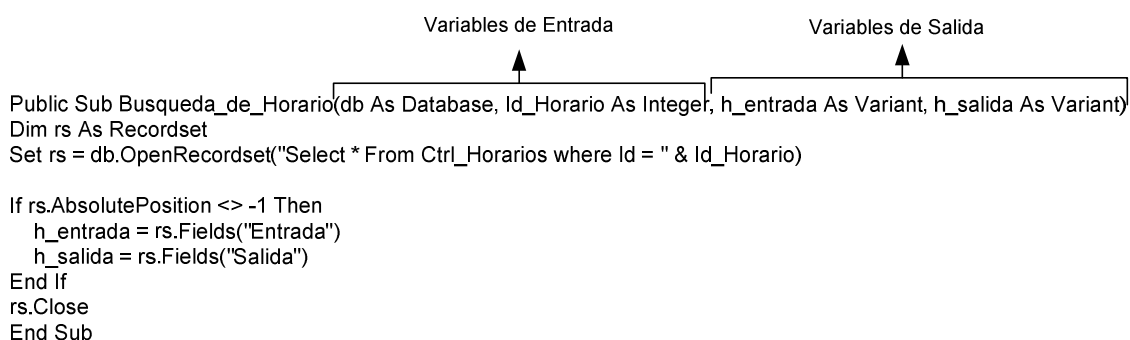
- Seleccionar <ENTRADA>, <SALIDA> de Ctrl_Horarios donde <id> esté en los resultados de **Consulta2** *<id_Horario>*.

Donde los campos de la tabla están identificados entre < >, y la referencia del filtro la proporciona la variable Id_Horario proporcionada por la CONSULTA2. Teniendo elementos en la CONSULTA4 se asigna la hora de entrada y salida del horario a las variables Entrada y Salida respectivamente.

3.1.4.2 Código Fuente.

Una descriptas las actividades y las funciones que desarrolla el módulo búsqueda de horarios, se presenta el código fuente de la estructura principal del módulo.

3.1.4.2.1 Estructura del Módulo.



3.1.5 Módulo Búsqueda de Registros.

Este módulo tiene la función de localizar cuales son los registros de todo el conjunto existente en la base de datos, correspondientes para la entrada y salida de las labores del día seleccionado.

A continuación se procede a presentar el pseudocódigo y el código del módulo.

3.1.5.1 Pseudocódigo.

En la tablas 3.7, 3.8 y 3.9 se presentan las variables de entrada, de salida y de control respectivamente, para el funcionamiento de este módulo.

Variable	Tipo de la Variable	Descripción
Control de Trabajo	Base de Datos Relacional	Almacena la referencia de la base de datos relacional "Control de Trabajo"
Fecha_Referencia	Fecha	Almacena la fecha seleccionada por el usuario. Este valor es proporcionado por MonthView.Value.
ClaveEmpleado	Entera	Almacena el identificador del empleado. Este valor es proporcionado por la CONSULTA3, ver sección 3.1.3.1.1
Entrada	Entera	Almacena la hora de entrada del horario de la actividad. Este valor es proporcionado por la CONSULTA4, ver sección 3.1.4.1.1.
Salida	Entera	Almacena la hora de salida del horario de la actividad. Este valor es proporcionado por la CONSULTA4, ver sección 3.1.4.1.1.

Tabla 3. 7: Variables de Entrada del Módulo Búsqueda de Registros.

Variable	Tipo de la Variable	Descripción
RegEntrada	Hora	Almacena el registro correspondiente a la entrada de la actividad.
IdEntrada	Entera	Almacena el identificador del registro de entrada.
RegSalida	Hora	Almacena el registro correspondiente a la salida de la actividad.
IdSalida	Entera	Almacena el identificador del registro de salida.

Tabla 3. 8: Variables de Salida Generadas por el Módulo Búsqueda de Registros.

Variable	Tipo de la Variable	Descripción
límiteAntes	Hora	Almacena la cota inferior del rango para tomar los registros como válidos, de la actividad.
límiteDespues	Hora	Almacena la cota superior del rango para tomar los registros como válidos, de la actividad.
Completo	Booleana	Bandera que permite identificar cuando existen los dos registros para la actividad.
Profesor	PROFE	Variable conformada por varias variables. Almacena la información necesaria para la evaluación de los registros y sus resultados.

Tabla 3. 9: Variables de Control empleadas por el Módulo Búsqueda de Registros.

3.1.5.1.1 Actividades y Funciones a Realizar.

Teniendo las variables utilizadas en el módulo de búsqueda de registros, se presentan los procedimientos que efectúa este módulo para cada una de las actividades del día.

3.1.5.1.1.1 Estructura del módulo.

Se marca el rango del conjunto de registros, para ello hay que asignar los límites inferior y superior e inicializar la variable Profesor. Después de inicializar la variable Profesor, se requiere hacer una consulta en la tabla CHEQUEOS para determinar cuales son los posibles registros de entrada y salida en que se efectúa dicha actividad. Esta consulta es identificada como CONSULTA5 que se describe de la siguiente forma:

- Seleccionar TODO de Chequeos donde <Fecha> = Fecha_Referencia y <Clave> = ClaveEmpleado y <Hora> esté entre (límiteAntes, limiteDespues).

Donde los campos de la tabla están identificados entre < >, y la referencia del filtro la proporciona las variables Fecha_Referencia, ClaveEmpleado, límiteAntes y límiteDespues.

Si la CONSULTA5 contiene elementos, se realiza el llamado al procedimiento CHECA_ENTRADA, tomando como referencia a la CONSULTA5. En caso contrario, es decir, que la CONSULTA5 no contenga elementos, se les asigna a las variables Id_Entrada e Id_Salida el valor de 0, esto marca que no existe identificador para los registros de entrada – salida, así mismo se le asigna a la variable Completo el valor de falso para marcar que no se tienen estos dos valores.

3.1.5.1.1.2 Procedimiento Checa_Entrada.

Como se puede observar en la sección 3.1.5.1.1.1, se hace un llamado a este procedimiento, el cual verifica si alguno de los registros puede ser considerado como la entrada de la actividad, este procedimiento trabaja de la siguiente forma.

Si dentro de la CONSULTA5 el registro activo contiene en el campo Concepto alguno de los marcados como valor de entrada, se les asigna a las variables RegEntrada el valor de la hora del registro y a la variable IdEntrada el identificador del mismo. Después se recorre al siguiente registro de la CONSULTA5 para posteriormente hacer el llamado al procedimiento CHECA_ENTRADA2 tomando como referencia a esta consulta.

En caso que la comparación efectuada no sea verdadera, habrá que recorrerse al siguiente registro de la CONSULTA5 y así hacer el llamado nuevamente al mismo procedimiento Checa_Entrada. Cabe aclarar que en caso que se recorra toda la consulta sin encontrar un posible valor para el registro de entrada, se limpia la variable IdSalida y se le asigna a la variable Completo el valor de falso para determinar que no se encontraron los registros correspondientes a la actividad.

3.1.5.1.1.3 Procedimiento Checa_Entrada2.

En el punto anterior se hace el llamado a este procedimiento, el cual verifica lo se describe a continuación.

Si dentro de la CONSULTA5 el registro activo contiene en el campo Concepto alguno de los conceptos marcados como salida, se hace el llamado al procedimiento CHECA_SALIDA tomando como referencia dicha consulta.

En caso que esta comparación sea falsa, habrá que hacer el llamado nuevamente al procedimiento CHECA_ENTRADA, aclarando que, si se llega al final de la consulta sin obtener algún valor para la entrada de la actividad, se limpiará la variable IdSalida y se le asignará a la variable Completo el valor de falso y así se determina que no se encontraron los registros correspondientes a la actividad.

3.1.5.1.1.4 Procedimiento Checa_Salida.

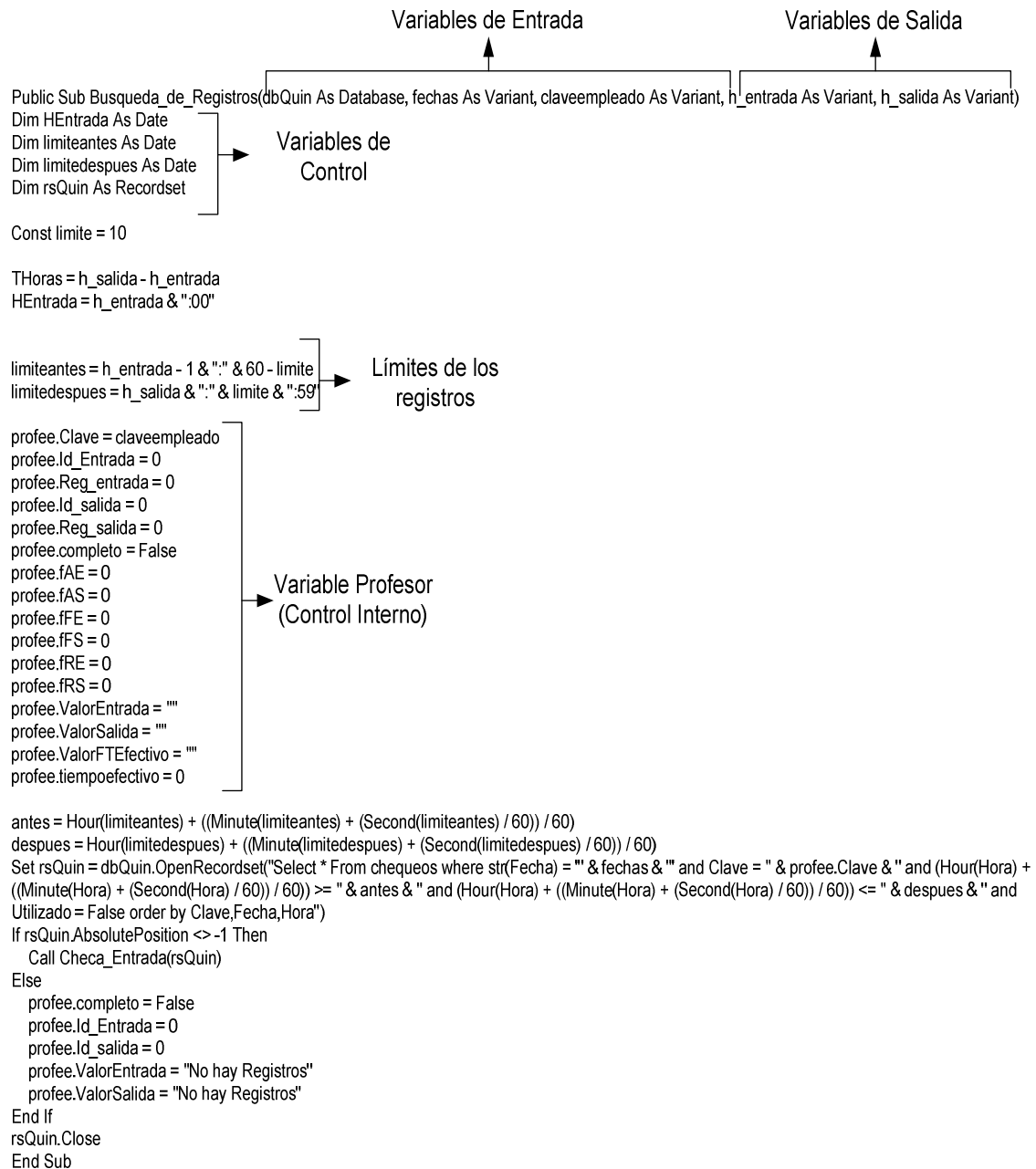
Como Checa_Entrada2 hace el llamado a Checa_Salida, su función principal es encontrar el registro de salida de la actividad, para ello efectúa lo siguiente.

Si dentro de la CONSULTA5, el registro activo contiene en el campo Concepto alguno de los marcados como salida, se cierra el ciclo asignando a las variables RegSalida e IdSalida el valor de la hora de registro y su identificador respectivamente. Así mismo, se activa la variable Completo a verdadero para indicar que están localizados tanto la entrada como la salida de la actividad. Cabe aclarar que si en el recorrido de la CONSULTA5 se llega al final de ésta sin encontrar el registro de salida, se limpiará la variable IdSalida y se le asignará a la variable Completo el valor de falso para determinar que no se encontraron los registros correspondientes a la actividad.

3.1.5.2 Código Fuente.

Una vez que se ha descrito el funcionamiento del módulo de búsqueda de registros, se presenta a continuación el código fuente de la estructura del módulo y sus procedimientos internos que lo conforman.

3.1.5.2.1 Estructura del Módulo.



3.1.5.2.2 Procedimiento Checa_Entrada.

```
Public Sub Checa_Entrada(rst As Recordset)
If (UCase(rst.Fields("Concepto")) = "IN_ENTRY") Or (UCase(rst.Fields("Concepto")) = "VERIFY_OK") Then
    profee.Reg_entrada = rst.Fields("Hora")
    profee.Id_Entrada = rst.Fields("Id")
    rst.MoveNext
    Call Checa_Entrada2(rst)
Else
If (UCase(rst.Fields("Concepto")) = "OUT_ENTRY") Then
    profee.Reg_entrada = rst.Fields("Hora")
    profee.Id_Entrada = rst.Fields("Id")
    rst.MoveNext
If rst.EOF Then
    rst.MovePrevious
    profee.Id_salida = 0
    profee.completo = False
    profee.ValorEntrada = "CHECÓ MAL"
Else
    rst.MoveNext
If rst.EOF = False Then
    rst.MovePrevious
    profee.Id_salida = 0
    Call Checa_Entrada(rst)
Else
    rst.MovePrevious
If UCase(rst.Fields("Concepto")) = "OUT_ENTRY" Then
    profee.Reg_entrada = rst.Fields("Hora")
    profee.Id_Entrada = rst.Fields("Id")
    profee.completo = False
    profee.ValorEntrada = "CHECÓ MAL"
Else
    profee.completo = False
    profee.ValorEntrada = "CHECÓ MAL"
End If
End If
End If
End If
End Sub
```

3.1.5.2.3 Procedimiento Checa_Entrada2.

```

Public Sub Checa_Entrada2(rst As Recordset)
If rst.EOF Then
    profee.completo = False
    profee.ValorEntrada = "CHECÓ MAL"
    Exit Sub
End If

If (UCase(rst.Fields("Concepto")) = "OUT_ENTRY") Then
    Min = Minute(profee.Reg_entrada) + (Second(profee.Reg_entrada) / 60)
    u = Hour(profee.Reg_entrada) + (Min / 60)
    Min = Minute(rst.Fields("Hora")) + (Second(rst.Fields("Hora")) / 60)
    v = Hour(rst.Fields("Hora")) + (Min / 60)
    If (v - u) < (20 / 60) Then
        rst.MoveNext
        If rst.EOF Then
            rst.MovePrevious
            Call Checa_Salida(rst)
        Else
            Call Checa_Entrada(rst)
        End If
    Else
        Call Checa_Salida(rst)
    End If
Else
    If (UCase(rst.Fields("Concepto")) = "IN_ENTRY") Then
        Min = Minute(profee.Reg_entrada) + (Second(profee.Reg_entrada) / 60)
        u = Hour(profee.Reg_entrada) + (Min / 60)
        Min = Minute(rst.Fields("Hora")) + (Second(rst.Fields("Hora")) / 60)
        v = Hour(rst.Fields("Hora")) + (Min / 60)
        If (v - u) < (30 / 60) Then
            Call Checa_Entrada(rst)
        Else
            profee.Reg_salida = 0
            profee.completo = False
            profee.ValorEntrada = "CHECÓ MAL"
        End If
    Else
        If (UCase(rst.Fields("Concepto")) = "VERIFY_OK") Then
            Min = Minute(profee.Reg_entrada) + (Second(profee.Reg_entrada) / 60)
            u = Hour(profee.Reg_entrada) + (Min / 60)
            Min = Minute(rst.Fields("Hora")) + (Second(rst.Fields("Hora")) / 60)
            v = Hour(rst.Fields("Hora")) + (Min / 60)
            If (v - u) < (20 / 60) Then
                Call Checa_Entrada(rst)
            Else
                Call Checa_Salida(rst)
            End If
        End If
    End If
End If
End Sub

```

3.1.5.2.4 Procedimiento Checa_Salida.

```
Public Sub Checa_Salida(rst As Recordset)
If rst.EOF Then
    profee.completo = False
    profee.ValorEntrada = "CHECÓ MAL"
    Exit Sub
End If
```

```
If (UCase(rst.Fields("Concepto")) = "OUT_ENTRY") Or (UCase(rst.Fields("Concepto")) = "VERIFY_OK") Then
    profee.Reg_salida = rst.Fields("Hora")
    profee.Id_salida = rst.Fields("Id")
    profee.completo = True
End If
End Sub
```

3.1.6 Módulo de Evaluación.

Este módulo se encargará de evaluar los registros del empleado en las funciones difusas creadas. A continuación se procede a presentar el pseudocódigo y el código del módulo.

3.1.6.1 Pseudocódigo.

En las siguientes tablas se presentan las variables de entrada, las variables de salida y las variables de control que utiliza este módulo.

Variable	Tipo de la Variable	Descripción
Entrada	Entera	Almacena la hora de entrada de la actividad. Este valor es proporcionado por la CONSULTA4, véase sección 3.1.4.1.3.
Salida	Entera	Almacena la hora de Salida de la actividad. Este valor es proporcionado por la CONSULTA4, véase sección 3.1.4.1.3.
RegEntrada	Hora	Almacena el registro de entrada de la actividad. Este valor el proporcionado por el procedimiento CHECA_ENTRADA del módulo anterior, véase sección 3.1.5.1.4.2.
RegSalida	Hora	Almacena el registro de salida de la actividad. Este valor el proporcionado por el procedimiento CHECA_SALIDA del módulo anterior, véase sección 3.1.5.1.4.3.
Completo	Booleana	Bandera que permite identificar cuando existen los dos registros para la actividad. Este valor es proporcionado por el módulo anterior, véase sección 3.1.5.1.4.

Tabla 3. 10: Variables de Entrada del Módulo de Evaluación.

Variable	Tipo de la Variable	Descripción
fAE, fRE, fIE	Real	Almacenan los resultados de las evaluaciones en las funciones difusas para el registro de entrada.
fAS, fRS	Real	Almacenan los resultados de las evaluaciones en las funciones difusas para el registro de salida.
Fbueno, Fmalo	Real	Almacenan los resultados de las evaluaciones en las funciones difusas para el tiempo efectivo parcial.
Tiempoefectivo	Hora	Almacena el tiempo efectivo parcial de la actividad.

Tabla 3. 11: Variables de Salida del Módulo de Evaluación.

Variable	Tipo de la Variable	Descripción
Profesor	PROFE	Variable conformada por varias variables. Almacena la información necesaria para la evaluación de los registros y sus resultados.

Tabla 3. 12: Variables de Control Utilizadas por el Módulo de Evaluación.

3.1.6.1.1 Actividades y Funciones a Realizar.

Teniendo como referencia las variables de entrada, hacen que este módulo opere siempre y cuando la variable COMPLETO sea VERDADERA y los procedimientos a efectuar son:

- a. Convertir la variable RegEntrada en un valor de tipo real para poder evaluarlo en las funciones difusas para el registro de entrada.
- b. Convertir la variable RegSalida en un valor de tipo real y así evaluarlo en las funciones difusas para el registro de salida.
- c. Calcular el tiempo efectivo para evaluarlo en las funciones difusas para el tiempo efectivo parcial.

Para poder calcular el tiempo efectivo, se tiene que comparar los registros de entrada y salida con la hora de entrada y salida marcada por el horario, esta situación se puede explicar de la siguiente forma:

Si el valor almacenado en la variable RegEntrada es menor o igual que la Entrada marcada por el horario, se le asigna a la variable $TIEMPO_{ENTRADA}$ la entrada marcada por el horario, en caso contrario, es decir, que el valor de la variable RegEntrada sea mayor a la Entrada marcada por el horario, se le asigna a la variable $TIEMPO_{ENTRADA}$ el valor del registro de entrada marcado por el usuario.

De la misma forma que se compara el registro de entrada con la entrada marcada por el horario, se compara el valor de la variable RegSalida si es menor o igual que la Salida marcada por el horario, se le asigna a la variable $TIEMPO_{SALIDA}$ el valor del registro de salida marcado por el usuario. En caso contrario, que el registro de salida sea mayor que la hora de salida marcada por el horario, se le asigna a la variable $TIEMPO_{SALIDA}$ la hora de salida marcada por el horario establecido.

Por último se le asigna a la variable $TIEMPO_{EFECTIVO}$ el resultado del $TIEMPO_{SALIDA}$ menos el $TIEMPO_{ENTRADA}$.

Una vez efectuados la conversión de los registros y el cálculo del tiempo efectivo, lo siguiente es evaluar estos datos en las funciones difusas correspondientes.

- a. Evaluar el valor almacenado en la variable RegEntrada en las funciones Difusas para el registro de entrada.
- b. Evaluar el valor almacenado en la variable RegSalida en las funciones Difusas para el registro de Salida.
- c. Evaluar el valor almacenado en la variable TiempoEfectivo en las funciones Difusas para el tiempo efectivo parcial.

3.1.6.1.1.1 Evaluando el registro de entrada

Como se pudo observar en la sección 2.4.1 se hace una presentación de las tres funciones difusas consideradas para evaluar el registro de entrada (asistencia, asistencia con retardo e inasistencia) así como los parámetros que utilizan cada una. Los valores para estos parámetros quedan definidos a continuación.

- x Es el registro de entrada generado por el profesor.
- a Es la hora de entrada de la actividad marcada por el horario.
- b Es el valor de a más 8 minutos.

- m Es el valor almacenado en \underline{a} más 11 minutos.
- c Es sumarle 18 minutos al valor contenido en \underline{a} .
- d Resulta de sumarle 21 minutos al valor de \underline{a} .

Dentro del código fuente, esta asignación queda representada en la páginas 70 mientras que las funciones difusas se muestran en la página 71.

3.1.6.1.1.2 Evaluando el registro de salida.

Al igual que en la sección 3.1.6.1.1.1, se presentan los valores de evaluación para las dos funciones difusas para el registro de salida presentadas en 2.4.2 (Salida en tiempo y retiro previo a la tolerancia).

- x Es el registro de salida generado por el profesor.
- p Es la hora de salida de la actividad marcada por el horario.
- q Es el valor de \underline{p} menos 11 minutos.

La asignación de estas variables y sus funciones difusas quedan presentadas la página 71 de la sección 3.1.6.2.2.

3.1.6.1.1.3 Evaluando el tiempo efectivo parcial.

Recordando que en la sección 2.4.3 se presentan las dos funciones difusas para evaluar el tiempo efectivo parcial (Considerable y Malo), en esta sección se marcan los valores utilizados para los parámetros.

- x Es el tiempo efectivo parcial generado por los registros del profesor.
- a Es el tiempo estipulado por el horario para la actividad menos 15 minutos.
- b Es el tiempo estipulado por el horario para la actividad menos 10 minutos.

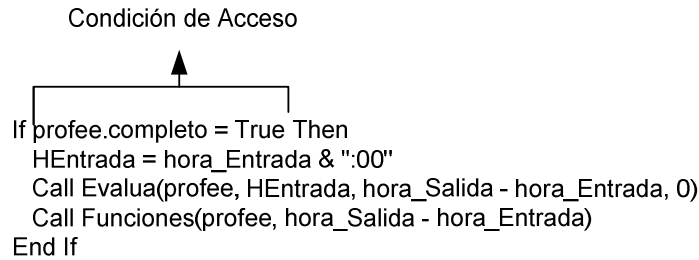
La presentación de estas dos funciones en el código fuente se aprecia en la página 71.

La última acción que efectúa en este módulo es la asignación de Etiquetas Difusas en base a las Funciones Difusas Evaluadas. La asignación de estas etiquetas se presenta en el Anexo 1 así como un ejemplo de la forma de evaluar.

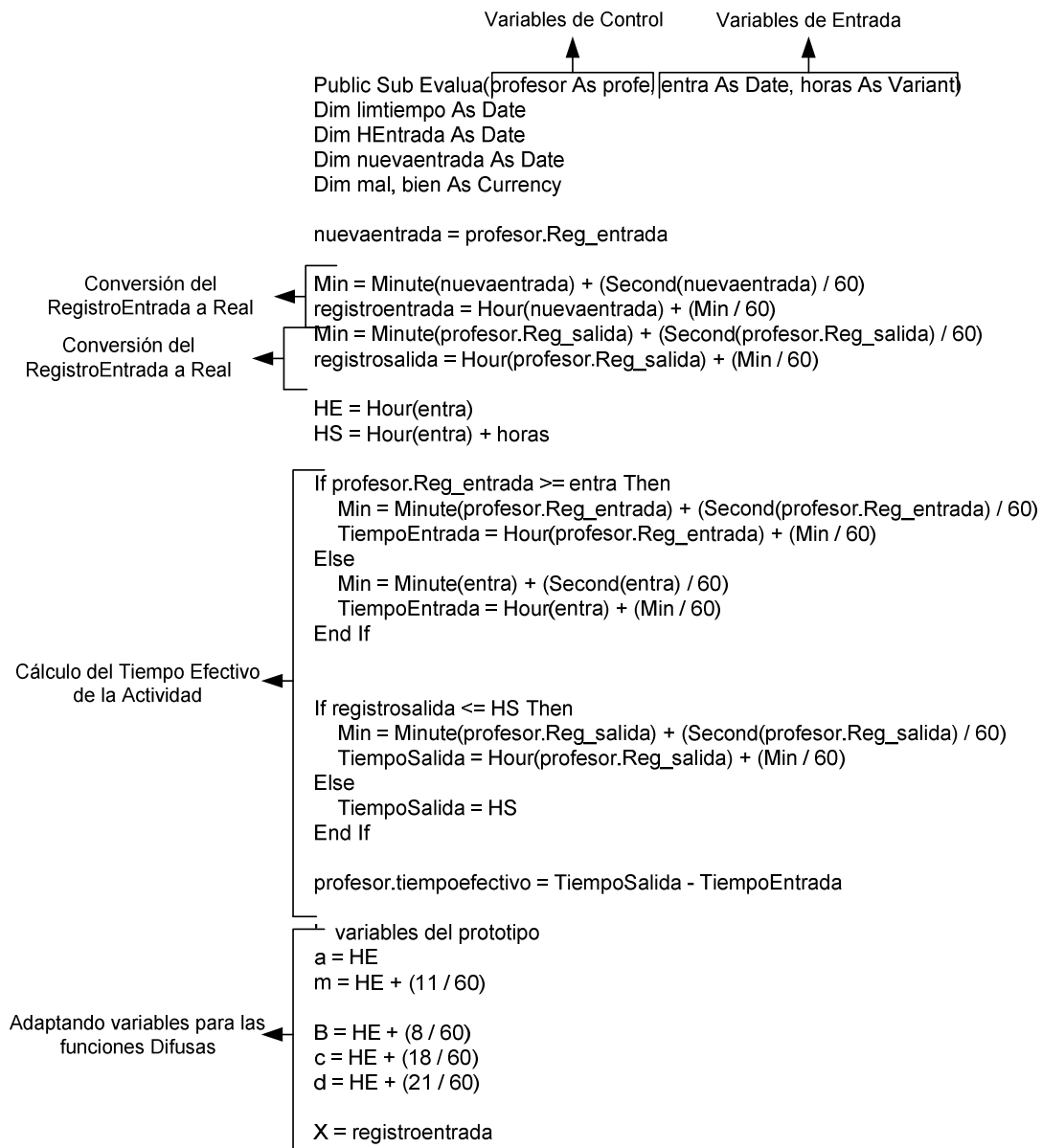
3.1.6.2 Código Fuente.

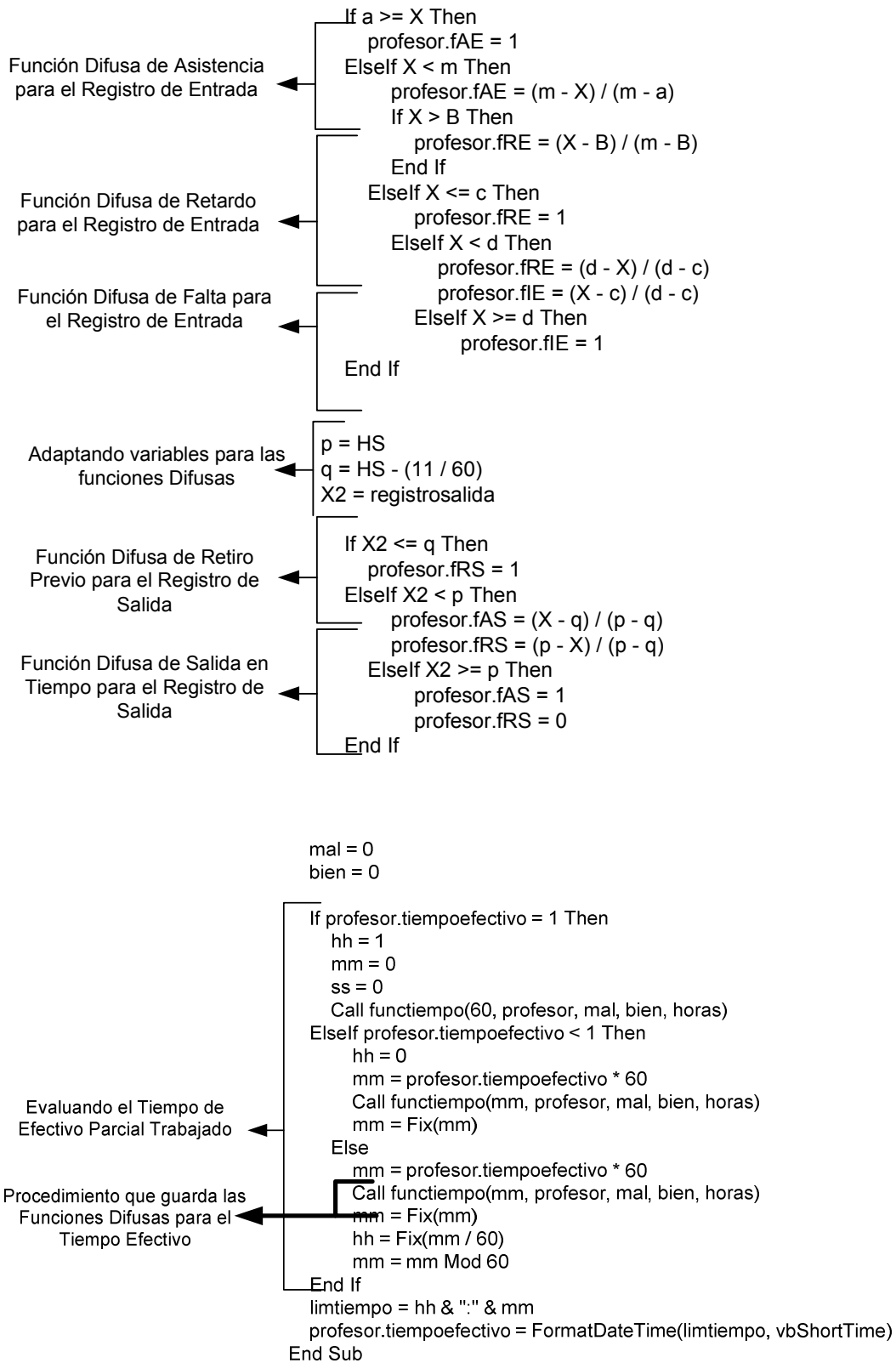
Una descritas las actividades y las funciones que desarrolla el módulo de evaluación, se presenta el código fuente de la estructura principal y del modo en como se accede a este módulo.

3.1.6.2.1 Modo de acceso al módulo:



3.1.6.2.2 Estructura del Módulo.





3.1.6.2.3 Procedimientos Internos.

Public Sub functiempo(tiempo As Variant, profesor As profe, malo As Variant, bueno As Variant, horas As Variant)

Dim refcompleto As Variant

Dim refbueno As Variant

Dim refcorte As Variant

Dim refmalo As Variant

refcompleto = horas * 60 '1:60

refbueno = refcompleto - 10 '1:50

refcorte = refcompleto - 12.5 '1:47.5

refmalo = refcompleto - 15 '1:45

If tiempo <= refmalo Then

malo = 1

Elseif tiempo <= refcorte Then

malo = 1 - (2 * (((tiempo - refmalo) / (refbueno - refmalo)) ^ 2))

Elseif tiempo < refbueno Then

malo = 2 * (((tiempo - refbueno) / (refbueno - refmalo)) ^ 2)

Else

malo = 0

End If

Función Difusa de
Tiempo Efectivo
Bueno o Considerable

If tiempo <= refmalo Then

bueno = 0

Elseif tiempo <= refcorte Then

bueno = 2 * (((tiempo - refmalo) / (refbueno - refmalo)) ^ 2)

Elseif tiempo < refbueno Then

bueno = 1 - (2 * (((tiempo - refbueno) / (refbueno - refmalo)) ^ 2))

Else

bueno = 1

End If

profesor.Fbueno = bueno

profesor.Fmalo = malo

Función Difusa de
Tiempo Efectivo malo

If malo = 1 Then

profesor.ValorFTEfectivo = "Tiempo Malo"

Elseif malo > bueno Then

profesor.ValorFTEfectivo = "Tiempo Parcialmente Malo"

Elseif bueno = 1 Then

profesor.ValorFTEfectivo = "Tiempo Bueno"

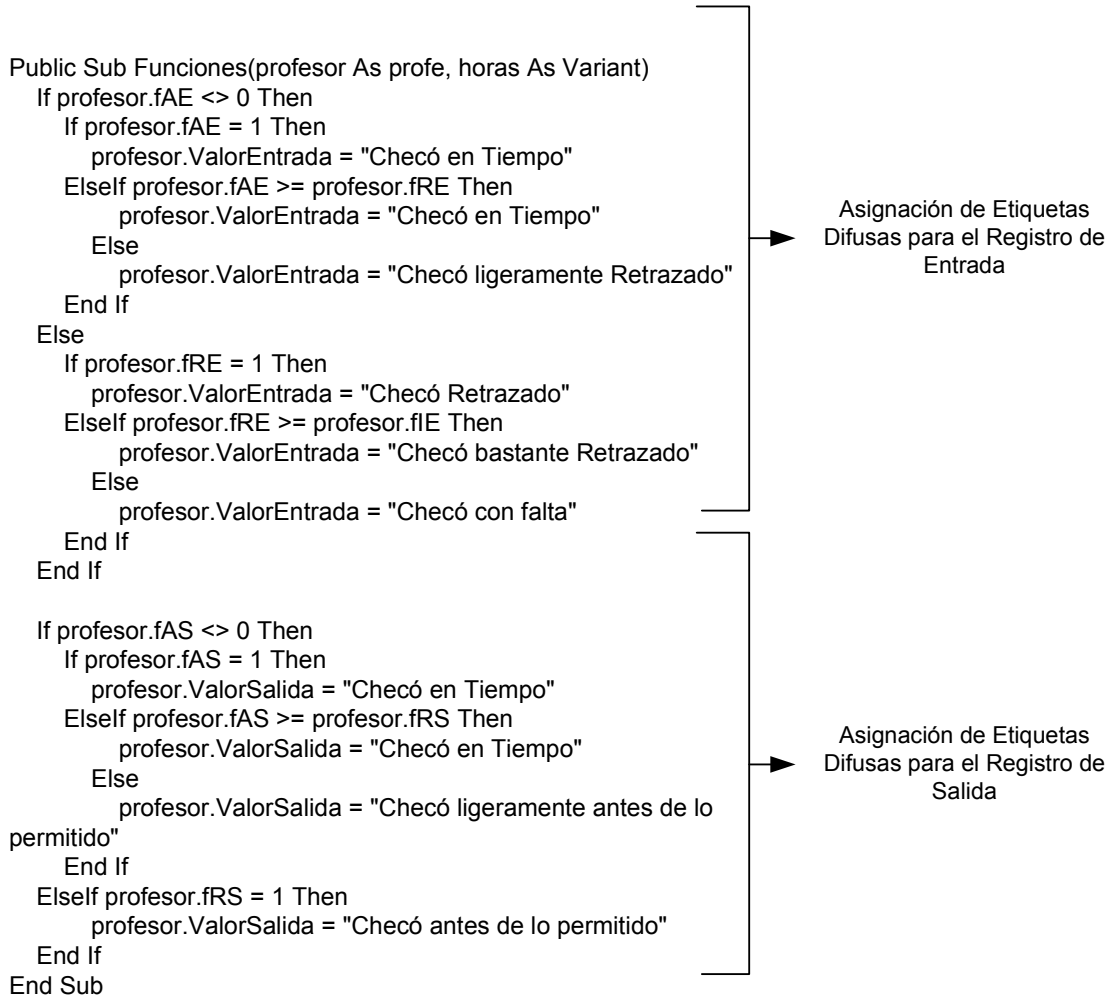
Else

profesor.ValorFTEfectivo = "Tiempo Parcialmente Bueno"

End If

End Sub

Asignación de
Etiquetas Difusas para
el Tiempo Efectivo



3.1.7 Módulo de Análisis y Reportes.

Este módulo le permite al usuario mostrar los resultados de las evaluaciones efectuadas; también el usuario podrá seleccionar la base de datos del control de profesores que desee utilizar así como la base de datos Control de Trabajo en la que están resguardados los resultados de las evaluaciones. A los resultados se les puede hacer filtrados por fechas, clave de empleado y variables difusas. La interfaz gráfica de este módulo se presenta en la siguiente figura.

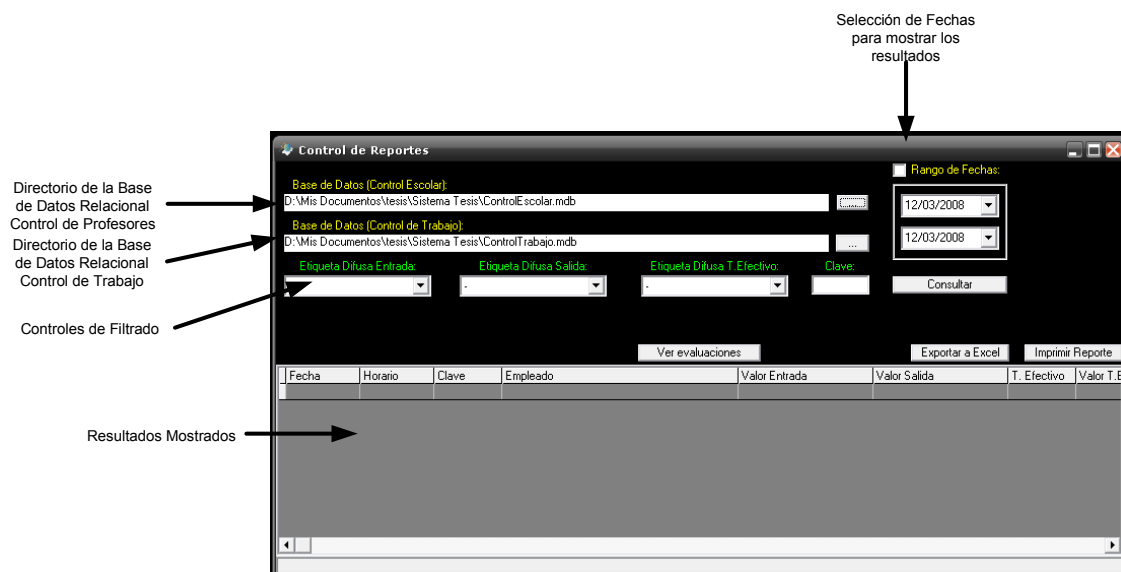


Figura 3. 2: Pantalla de Reportes:

Como se puede observar en la figura anterior, el usuario únicamente necesita seleccionar el rango de los días que desee visualizar y, en caso necesario, las etiquetas difusas y/o la clave del empleado a filtrar. Esta pantalla muestra únicamente los valores difusos de la evaluación, en caso que el usuario requiera ver los valores de las funciones difusas y sus registros, necesitará dar doble click sobre el Control MSFLEXGrid.

Para el área de selección de fechas se utiliza un componente de visual basic llamado DtPicker. De este componente se emplea el comando Value, el cual proporciona la fecha de referencia.

Para el mostrado de resultados se utiliza el MsFlexGrid, del cual se emplean los siguientes comandos:

- Rows: El total de renglones que contiene el componente.
- Cols: El total de columnas que contiene el componente.
- ColWidth: Establece u obtiene el ancho de una columna.
- RowHeight: Establece u obtiene el alto de una renglón.
- TextMatrix: Establece u obtiene el texto de la matriz marcada por (Renglón, Columna).
- Clear: Limpia el componente.

Las actividades y las funciones que efectúa el módulo son las siguientes:

3.1.7.1 Pseudocódigo.

Para el funcionamiento de este módulo se utilizan las siguientes variables de entrada y de control descritas en las tablas 3.13 y 3.14 respectivamente:

Variable	Tipo de la Variable	Descripción
Control Escolar	Base de Datos Relacional	Almacena la referencia de la base de datos relacional "Control de Profesores"
Control de Trabajo	Base de Datos Relacional	Almacena la referencia de la base de datos relacional "Control de Trabajo"

Tabla 3. 13: Variables de Entrada para el Módulo de Reportes.

Variable	Tipo de la Variable	Descripción
FechaINICIO	Fecha	Almacena la fecha de inicio que se requiere en el reporte.
FechaTérmino	Fecha	Almacena la fecha de término que se requiere en el reporte.
ClaveEmpleado	Entera	Almacena el identificador del empleado que se requiere en el reporte.

Tabla 3. 14: Variables de Control del Módulo de Reportes.

3.1.7.1.1 Actividades y Funciones a Realizar.

Teniendo las variables de entrada como referencia, se verifica que estén proporcionadas las dos bases de datos Relacionales (Control de Profesores y Control de Trabajo). En caso de faltar alguna de estas no ejecuta ninguna acción y, sale del procedimiento.

Teniendo las variables de control de fechas para efectuar el reporte correspondiente, se verifica si el reporte es por rango de fechas o, con todas las fechas para hacer una consulta a la tabla CTRL_FECHAS, a ésta la identificamos con el nombre de CONSULTA6:

- A. En caso de ser TODAS las fechas:

Selecciona TODO de Ctrl_Fechas ordenado por <Fecha>, <Id>.

- B. En caso de ser un rango de fechas:

Seleccionar TODO de Ctrl_Fechas donde <Fecha> este en el rango de (DTPICKER1.value,DTPICKER2.value) ordenados por <Fecha>, <id>.

Donde los campos de la tabla están identificados entre < >, y la referencia del filtro la proporciona las variables FechaINICIO y FechaTérmino, como se puede observar estas dos variables se asocian a los componentes DTPICKER1 y DTPICKER2 respectivamente.

Teniendo la CONSULTA6 se verifica si existe algún filtrado para las etiquetas difusas, en caso de requerir dicho filtrado, éste efectúa.

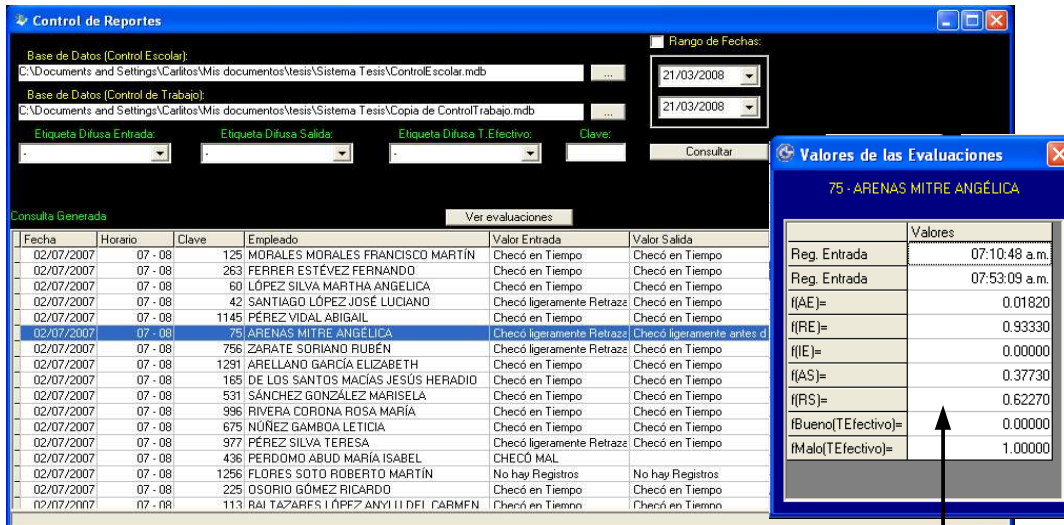
Reutilizando la CONSULTA6 se verifica si requiere un filtrado por el identificador de algún empleado, en caso de requerirlo se efectúa dicho filtrado.

Cabe aclarar que para mostrar los resultados, se asigna la CONSULTA6 al MsFlexGrid para ser mostrada de la siguiente manera:

Fecha	Horario	Clave	Empleado	Valor Entrada	Valor Salida	T. Efectivo	Valor T. Efectivo
02/07/2007	07 - 08	125	MORALES MORALES FRANCISCO MARTÍN	Checó en Tiempo	Checó en Tiempo	00:55	Tiempo Bueno
02/07/2007	07 - 08	263	FERRER ESTÉVEZ FERNANDO	Checó en Tiempo	Checó en Tiempo	00:51	Tiempo Bueno
02/07/2007	07 - 08	60	LÓPEZ SILVA MARTHA ANGELICA	Checó en Tiempo	Checó en Tiempo	00:59	Tiempo Bueno
02/07/2007	07 - 08	42	SANTIAGO LÓPEZ JOSÉ LUCIANO	Checó ligeramente Retraza	Checó en Tiempo	00:51	Tiempo Bueno
02/07/2007	07 - 08	1145	PÉREZ VIDAL ABIGAIL	Checó en Tiempo	Checó en Tiempo	00:53	Tiempo Bueno
02/07/2007	07 - 08	75	ARENAS MITRE ANGÉLICA	Checó ligeramente Retraza	Checó ligeramente antes d	00:42	Tiempo Malo
02/07/2007	07 - 08	756	ZARATE SORIANO RUBÉN	Checó ligeramente Retraza	Checó en Tiempo	00:50	Tiempo Bueno
02/07/2007	07 - 08	1291	ARELLANO GARCÍA ELIZABETH	Checó en Tiempo	Checó en Tiempo	00:52	Tiempo Bueno
02/07/2007	07 - 08	165	DE LOS SANTOS MACÍAS JESÚS HERADIO	Checó en Tiempo	Checó en Tiempo	01:00	Tiempo Bueno
02/07/2007	07 - 08	531	SÁNCHEZ GONZÁLEZ MARISELA	Checó en Tiempo	Checó en Tiempo	01:00	Tiempo Bueno
02/07/2007	07 - 08	996	RIVERA CORONA ROSA MARÍA	Checó en Tiempo	Checó en Tiempo	00:58	Tiempo Bueno
02/07/2007	07 - 08	675	NÚÑEZ GAMBOA LETICIA	Checó en Tiempo	Checó en Tiempo	00:52	Tiempo Bueno
02/07/2007	07 - 08	977	PÉREZ SILVA TERESA	Checó ligeramente Retraza	Checó en Tiempo	00:51	Tiempo Bueno
02/07/2007	07 - 08	436	PERDOMO ABUD MARÍA ISABEL	CHECÓ MAL		0	
02/07/2007	07 - 08	1256	FLORES SOTO ROBERTO MARTÍN	No hay Registros	No hay Registros	0	
02/07/2007	07 - 08	225	OSORIO GÓMEZ RICARDO	Checó en Tiempo	Checó en Tiempo	00:56	Tiempo Bueno
02/07/2007	07 - 08	113	RAI TÁZARFS I LÓPEZ ANYILU DFI CARMEN	Checó en Tiempo	Checó en Tiempo	00:56	Tiempo Bueno

Figura 3. 3: Resultados mostrados por el módulo de reportes.

Recuerde que los resultados de las evaluaciones difusas no son presentados, si éstos fueran requeridos deberá darse doble click en el renglón deseado, esta acción presenta una nueva pantalla que se muestra a continuación:



Resultados de las evaluaciones en las funciones difusas

Figura 3. 4: Desglose de resultados al dar doble click en un renglón deseado.

Para lograr que se muestre este nuevo formulario llamado resumen, es necesario realizar los siguientes procedimientos:

- Para obtener los resultados de las evaluaciones en las funciones difusas del registro entrada, se hace el llamado de la CONSULTA7.
- Si se quiere obtener los resultados de las evaluaciones en las funciones difusas para el registro de salida, se utiliza la CONSULTA8.
- Si la información que se requiere es sobre el tiempo efectivo parcial, se hace el llamado de la CONSULTA9.
- Para visualizar los registros de entrada – salida de la actividad, se efectúa la CONSULTA10.

Por último, para ser mostradas estas consultas en el componente MsFlexGrid, se hace la asignación correspondiente.

3.1.7.2 Código Fuente.

Una vez que se ha descrito el funcionamiento y las actividades que se realizan en el módulo de reportes, se presenta el código fuente de las variables de control, de la estructura del módulo y del formulario llamado resumen.

3.1.7.2.1 Al cargar el módulo:

Referencia a las Bases de
Datos Relacionales

```
Label2 = App.Path & "ControlTrabajo.mdb"  
Label1 = App.Path & "ControlEscolar.mdb"
```

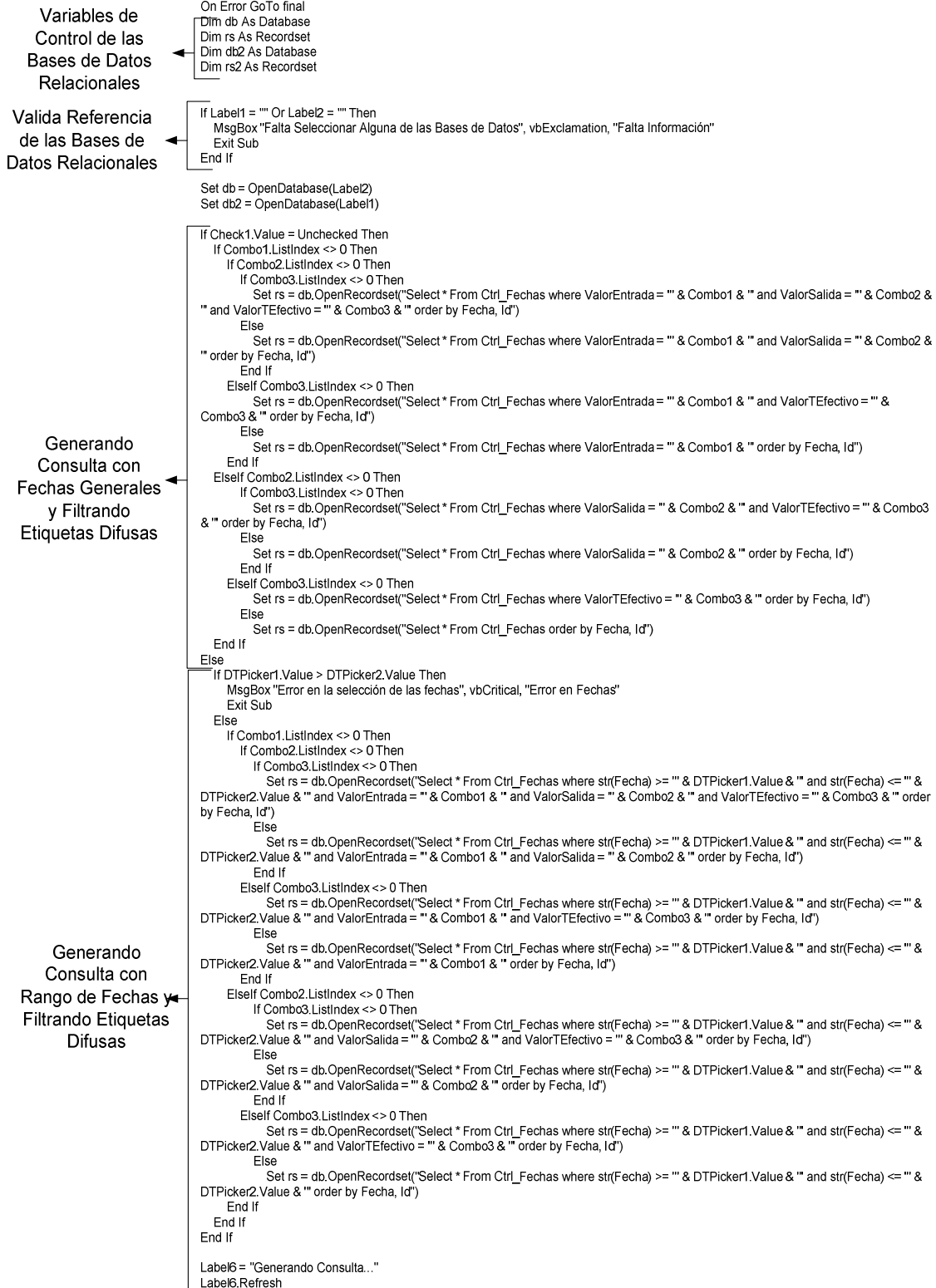
Inicializa MsFlexGris

```
MSFlex.ColWidth(0) = 100  
MSFlex.ColWidth(1) = 0  
MSFlex.ColWidth(2) = 1100  
MSFlex.ColWidth(3) = 0  
MSFlex.ColWidth(4) = 1100  
MSFlex.ColWidth(5) = 1000  
MSFlex.ColWidth(6) = 3500  
MSFlex.ColWidth(7) = 2000  
MSFlex.ColWidth(8) = 0  
MSFlex.ColWidth(9) = 2000  
MSFlex.ColWidth(10) = 0  
MSFlex.ColWidth(11) = 1000  
MSFlex.ColWidth(12) = 1500  
MSFlex.ColWidth(13) = 0  
MSFlex.ColWidth(14) = 0  
  
MSFlex.TextMatrix(0, 1) = "Id"  
MSFlex.TextMatrix(0, 2) = "Fecha"  
MSFlex.TextMatrix(0, 3) = "IdActividad"  
MSFlex.TextMatrix(0, 4) = "Horario"  
MSFlex.TextMatrix(0, 5) = "Clave"  
MSFlex.TextMatrix(0, 6) = "Empleado"  
MSFlex.TextMatrix(0, 7) = "Valor Entrada"  
MSFlex.TextMatrix(0, 8) = "IdEntrada"  
MSFlex.TextMatrix(0, 9) = "Valor Salida"  
MSFlex.TextMatrix(0, 10) = "IdSalida"  
MSFlex.TextMatrix(0, 11) = "T. Efectivo"  
MSFlex.TextMatrix(0, 12) = "Valor T.Efectivo"  
MSFlex.TextMatrix(0, 13) = "IdTEfectivo"  
MSFlex.TextMatrix(0, 14) = "IdChequesos"
```

Asigna la fecha actual a las
DTPicker

```
DTPicker1.Value = Date  
DTPicker2.Value = Date
```

3.1.7.2.2 Estructura del Módulo.



Inicializando
MsFlexGrid

```

Label6 = "Generando Consulta..."
Label6.Refresh
MSFlex.Clear
MSFlex.Rows = 2
MSFlex.TextMatrix(0, 1) = "Id"
MSFlex.TextMatrix(0, 2) = "Fecha"
MSFlex.TextMatrix(0, 3) = "IdActividad"
MSFlex.TextMatrix(0, 4) = "Horario"
MSFlex.TextMatrix(0, 5) = "Clave"
MSFlex.TextMatrix(0, 6) = "Empleado"
MSFlex.TextMatrix(0, 7) = "Valor Entrada"
MSFlex.TextMatrix(0, 8) = "IdEntrada"
MSFlex.TextMatrix(0, 9) = "Valor Salida"
MSFlex.TextMatrix(0, 10) = "IdSalida"
MSFlex.TextMatrix(0, 11) = "T. Efectivo"
MSFlex.TextMatrix(0, 12) = "Valor T.Efectivo"
MSFlex.TextMatrix(0, 13) = "IdT.Efectivo"
MSFlex.TextMatrix(0, 14) = "IdChequeos"
Progreso.Min = 0
If rs.AbsolutePosition <> -1 Then
    rs.MoveLast
    Progreso.Max = rs.AbsolutePosition
    rs.MoveFirst
End If
I = 0

```

Llenado del
MsFlexGrid con la
Consulta

```

Do Until rs.EOF
    MSFlex.TextMatrix(MSFlex.Rows - 1, 1) = rs.Fields("Id")
    MSFlex.TextMatrix(MSFlex.Rows - 1, 2) = rs.Fields("Fecha")
    MSFlex.TextMatrix(MSFlex.Rows - 1, 3) = rs.Fields("Id_Actividad")
    Set rs2 = db2.OpenRecordset("Select * From Horarios where Id = " & rs.Fields("Id_Actividad")")
    Id_Horario = 0
    id_Actividad = 0
    If rs2.AbsolutePosition <> -1 Then
        Id_Horario = rs2.Fields("Id_Horario")
        id_Actividad = rs2.Fields("Id_Materia")
    End If
    rs2.Close

    Horario = ""
    Set rs2 = db2.OpenRecordset("Select * From Ctrl_Horarios where Id = " & Id_Horario)
    If rs2.AbsolutePosition <> -1 Then
        Horario = Left$("00", 2 - Len(rs2.Fields("Entrada"))) & rs2.Fields("Entrada") & " - " & Left$("00", 2 - Len(rs2.Fields("Salida"))) &
rs2.Fields("Salida")
    End If
    rs2.Close
    MSFlex.TextMatrix(MSFlex.Rows - 1, 4) = Horario

    Set rs2 = db2.OpenRecordset("Select * From Materias where Id = " & id_Actividad)
    Clave = 0
    If rs2.AbsolutePosition <> -1 Then
        Clave = rs2.Fields("Id_Empleado")
    End If
    rs2.Close
    MSFlex.TextMatrix(MSFlex.Rows - 1, 5) = Clave
    a = ""
    Set rs2 = db2.OpenRecordset("Select * From Empleados where Clave = " & Clave)
    If rs2.AbsolutePosition <> -1 Then
        a = rs2.Fields("APaterno") & " " & rs2.Fields("AMaterno") & " " & rs2.Fields("Nombres")
    End If
    rs2.Close
    MSFlex.TextMatrix(MSFlex.Rows - 1, 6) = a

    MSFlex.TextMatrix(MSFlex.Rows - 1, 7) = rs.Fields("ValorEntrada")
    MSFlex.TextMatrix(MSFlex.Rows - 1, 8) = rs.Fields("FEntrada")
    MSFlex.TextMatrix(MSFlex.Rows - 1, 9) = rs.Fields("ValorSalida")
    MSFlex.TextMatrix(MSFlex.Rows - 1, 10) = rs.Fields("FSalida")
    MSFlex.TextMatrix(MSFlex.Rows - 1, 11) = rs.Fields("TEFECT")
    MSFlex.TextMatrix(MSFlex.Rows - 1, 12) = rs.Fields("ValorTEfectivo")
    MSFlex.TextMatrix(MSFlex.Rows - 1, 13) = rs.Fields("FTEfectivo")
    MSFlex.TextMatrix(MSFlex.Rows - 1, 14) = rs.Fields("Chequeos")
    MSFlex.Rows = MSFlex.Rows + 1
    Progreso.Value = I
    I = I + 1
rs.MoveNext
Loop
Progreso.Value = 0
db.Close
Set db = Nothing

```

```

Label6 = "Filtrando Clave..."
Label6.Refresh
Dim k As Integer
Progreso.Max = MSFlex.Rows
inicial = 0
k = 1
If IsNumeric(Text1) = True Then
  While k < MSFlex.Rows - 1
    If MSFlex.TextMatrix(k, 5) <> Text1 Then
      MSFlex.RemoveItem k
    Else
      k = k + 1
    End If
    Progreso.Value = inicial
    inicial = inicial + 1
  Wend
  Progreso.Value = 0
End If
Label6 = "Consulta Generada"
Exit Sub

final:
MsgBox "Error al Ejecutar", vbCritical, "Ocurrió un error"

```

Filtrado de la Clave del Empleado

3.1.7.2.3 Llamado al Formulario Secundario Resumen.

```

Dim resu As New Resumen
If MSFlex.RowSel <> 0 And MSFlex.RowSel <> MSFlex.Rows - 1 Then
  resu.Show
  resu.Label1 = MSFlex.TextMatrix(MSFlex.RowSel, 5) & " - " & MSFlex.TextMatrix(MSFlex.RowSel, 6)
  Call resu.Llenado(Label2, MSFlex.TextMatrix(MSFlex.RowSel, 8), MSFlex.TextMatrix(MSFlex.RowSel, 10),
  MSFlex.TextMatrix(MSFlex.RowSel, 14), MSFlex.TextMatrix(MSFlex.RowSel, 13))
End If

```

3.1.7.2.4 Código Fuente del Formulario Resumen.

```

Public Sub Llenado(txt As String, Id_Entrada As Variant, Id_salida As Variant, Id_chequeos As Variant, Id_Efectivo As Variant)
Dim db As Database
Dim rs As Recordset
Set db = OpenDatabase(txt)
Set rs = db.OpenRecordset("Select * From FuncionesEntradas where Id = " & Id_Entrada)
If rs.AbsolutePosition <> -1 Then
  MSFlex.TextMatrix(3, 1) = FormatNumber(rs.Fields("FAE"), 5)
  MSFlex.TextMatrix(4, 1) = FormatNumber(rs.Fields("FRE"), 5)
  MSFlex.TextMatrix(5, 1) = FormatNumber(rs.Fields("FIE"), 5)
End If
rs.Close
Set rs = db.OpenRecordset("Select * From FuncionesSalidas where Id = " & Id_salida)
If rs.AbsolutePosition <> -1 Then
  MSFlex.TextMatrix(6, 1) = FormatNumber(rs.Fields("FAS"), 5)
  MSFlex.TextMatrix(7, 1) = FormatNumber(rs.Fields("FRS"), 5)
End If
rs.Close

```

```
Set rs = db.OpenRecordset("Select * From Funciones_TEfectivo where Id = " & Id_Efectivo)
If rs.AbsolutePosition <> -1 Then
    MSFlex.TextMatrix(8, 1) = FormatNumber(rs.Fields("FTEBueno"), 5)
    MSFlex.TextMatrix(9, 1) = FormatNumber(rs.Fields("FTEMalo"), 5)
End If
rs.Close
```

```
Set rs = db.OpenRecordset("Select * From ChequeosActividad where Id = " & Id_chequeos)
If rs.AbsolutePosition <> -1 Then
    I1 = rs.Fields("Id_Entrada")
    I2 = rs.Fields("Id_Salida")
End If
rs.Close
```

```
Set rs = db.OpenRecordset("Select * From Chequeos where Id = " & I1)
If rs.AbsolutePosition <> -1 Then
    MSFlex.TextMatrix(1, 1) = rs.Fields("Hora")
End If
rs.Close
```

```
Set rs = db.OpenRecordset("Select * From Chequeos where Id = " & I2)
If rs.AbsolutePosition <> -1 Then
    MSFlex.TextMatrix(2, 1) = rs.Fields("Hora")
End If
rs.Close
db.Close
Set db = Nothing
End Sub
```

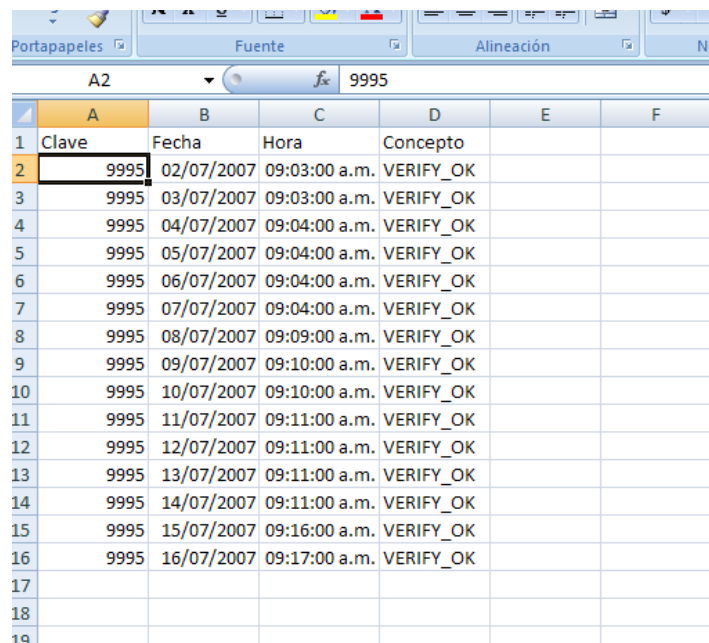
CAPÍTULO 4.

Pruebas y Resultados.

En este capítulo se muestran las pruebas efectuadas, de forma interna así como las de forma externa del sistema y los resultados obtenidos, mediante algunas figuras que describen el funcionamiento de dicho sistema. En primer lugar presentamos las pruebas efectuadas en la importación de los registros de los trabajadores, mostrando el funcionamiento de cada uno de los módulos y los resultados generados en el formulario de reportes.

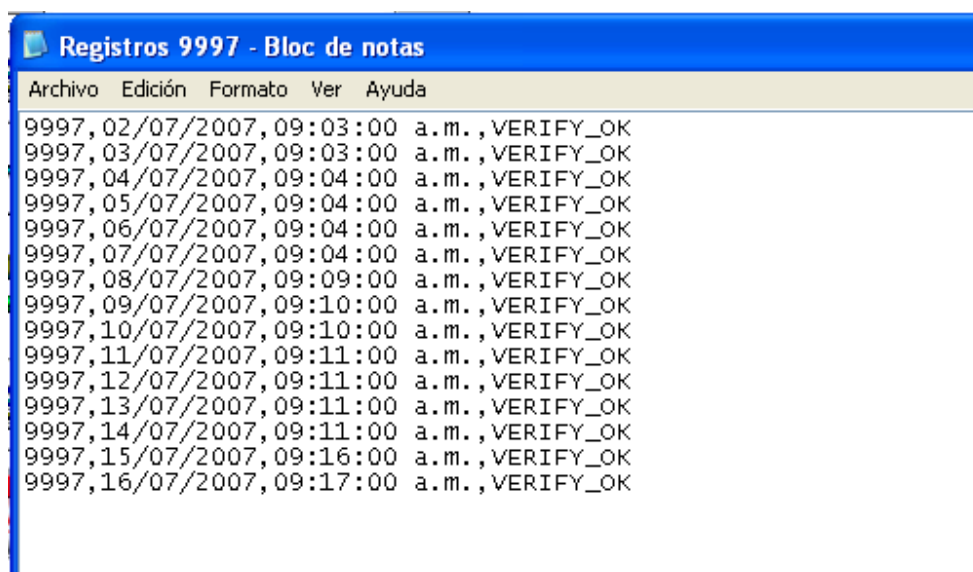
4.1 1ª Etapa: Migrando la información proporcionada por la institución.

Dentro de las pruebas que se efectuaron en el sistema, está como primera instancia la etapa de poder migrar los registros proporcionados por la institución al sistema. Cabe mencionar que este punto no había sido tratado en los capítulos anteriores. La institución nos proporciona únicamente dos tipos de información, la primera son los horarios de los profesores y el segundo son sus registros de las actividades. Este último, se nos puede proporcionar de dos formas, en un archivo de Microsoft Excel o en un archivo de texto. Los ejemplos de estos archivos son presentados en las figuras 4.1 y 4.2 respectivamente.



	A	B	C	D	E	F
1	Clave	Fecha	Hora	Concepto		
2	9995	02/07/2007	09:03:00 a.m.	VERIFY_OK		
3	9995	03/07/2007	09:03:00 a.m.	VERIFY_OK		
4	9995	04/07/2007	09:04:00 a.m.	VERIFY_OK		
5	9995	05/07/2007	09:04:00 a.m.	VERIFY_OK		
6	9995	06/07/2007	09:04:00 a.m.	VERIFY_OK		
7	9995	07/07/2007	09:04:00 a.m.	VERIFY_OK		
8	9995	08/07/2007	09:09:00 a.m.	VERIFY_OK		
9	9995	09/07/2007	09:10:00 a.m.	VERIFY_OK		
10	9995	10/07/2007	09:10:00 a.m.	VERIFY_OK		
11	9995	11/07/2007	09:11:00 a.m.	VERIFY_OK		
12	9995	12/07/2007	09:11:00 a.m.	VERIFY_OK		
13	9995	13/07/2007	09:11:00 a.m.	VERIFY_OK		
14	9995	14/07/2007	09:11:00 a.m.	VERIFY_OK		
15	9995	15/07/2007	09:16:00 a.m.	VERIFY_OK		
16	9995	16/07/2007	09:17:00 a.m.	VERIFY_OK		
17						
18						
19						

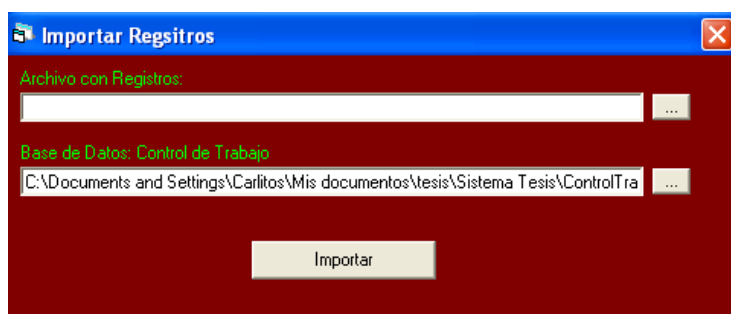
Figura 4. 1: Ejemplo del archivo con los registros en Microsoft Excel.



Archivo	Edición	Formato	Ver	Ayuda
9997,02/07/2007,09:03:00	a.m.	VERIFY_OK		
9997,03/07/2007,09:03:00	a.m.	VERIFY_OK		
9997,04/07/2007,09:04:00	a.m.	VERIFY_OK		
9997,05/07/2007,09:04:00	a.m.	VERIFY_OK		
9997,06/07/2007,09:04:00	a.m.	VERIFY_OK		
9997,07/07/2007,09:04:00	a.m.	VERIFY_OK		
9997,08/07/2007,09:09:00	a.m.	VERIFY_OK		
9997,09/07/2007,09:10:00	a.m.	VERIFY_OK		
9997,10/07/2007,09:10:00	a.m.	VERIFY_OK		
9997,11/07/2007,09:11:00	a.m.	VERIFY_OK		
9997,12/07/2007,09:11:00	a.m.	VERIFY_OK		
9997,13/07/2007,09:11:00	a.m.	VERIFY_OK		
9997,14/07/2007,09:11:00	a.m.	VERIFY_OK		
9997,15/07/2007,09:16:00	a.m.	VERIFY_OK		
9997,16/07/2007,09:17:00	a.m.	VERIFY_OK		

Figura 4. 2.: Ejemplo del archivo de texto con los registros.

La siguiente figura muestra el formulario encargado de la importación de los registros proporcionados por la institución.



Importar Registros

Archivo con Registros:

Base de Datos: Control de Trabajo

C:\Documents and Settings\Carlitos\Mis documentos\Tesis\Sistema Tesis\ControlTra

Importar

Figura 4. 3: Formulario para importar los registros de los empleados al sistema.

Una vez importados los registros se obtiene una nueva base de datos, que permite al usuario generar nuevas evaluaciones correspondientes a esta información mediante la siguiente etapa del sistema. La nueva base de datos se muestra en la figura 4.4.

Id	Clave	Fecha	Hora	Concepto
10629	9999	15/07/2007	09:16	VERIFY_OK
10630	9999	16/07/2007	09:17	VERIFY_OK
10631	9995	02/07/2007	09:03	VERIFY_OK
10632	9995	03/07/2007	09:03	VERIFY_OK
10633	9995	04/07/2007	09:04	VERIFY_OK
10634	9995	05/07/2007	09:04	VERIFY_OK
10635	9995	06/07/2007	09:04	VERIFY_OK
10636	9995	07/07/2007	09:04	VERIFY_OK
10637	9995	08/07/2007	09:09	VERIFY_OK
10638	9995	09/07/2007	09:10	VERIFY_OK
10639	9995	10/07/2007	09:10	VERIFY_OK
10640	9995	11/07/2007	09:11	VERIFY_OK
10641	9995	12/07/2007	09:11	VERIFY_OK
10642	9995	13/07/2007	09:11	VERIFY_OK
10643	9995	14/07/2007	09:11	VERIFY_OK
10644	9995	15/07/2007	09:16	VERIFY_OK
10645	9995	16/07/2007	09:17	VERIFY_OK
* (Nuevo)				

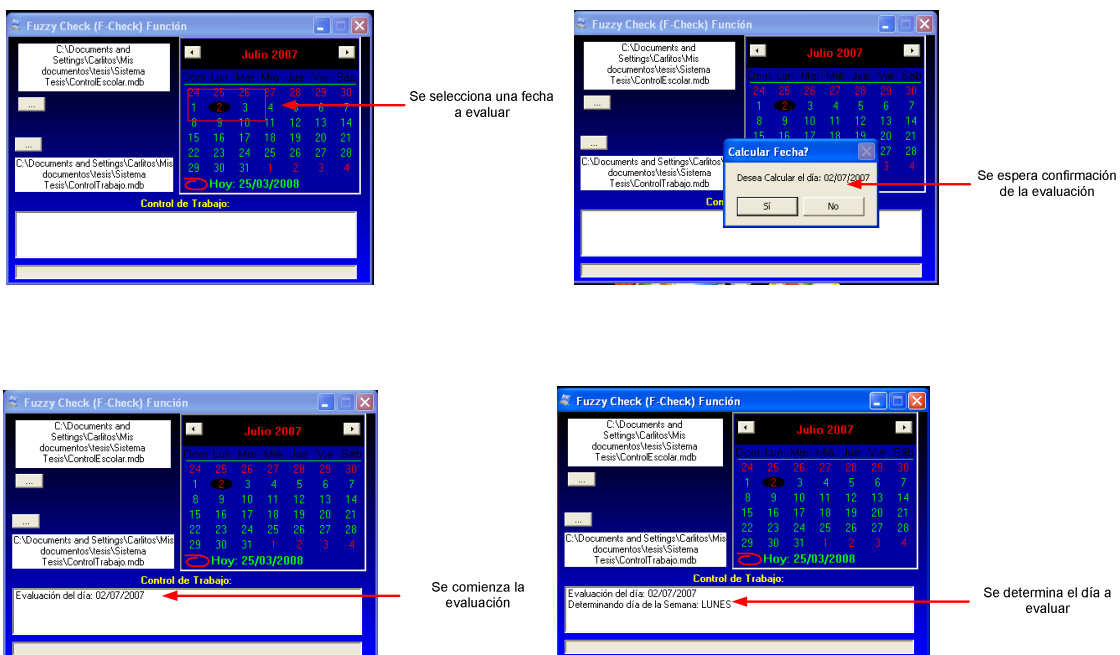
Registro: 10645 de 10645 Sin filtro Buscar

Figura 4. 4: Resultado de la importación de archivos proporcionados por la institución, en la tabla de los registros.

4.2 2ª Etapa: Generando las Evaluaciones.

Las pruebas efectuadas en esta etapa, están conformadas por la parte gráfica y por la parte lógica.

En la figura 4.5 se muestra el corrimiento del programa en su parte gráfica, en la cual se puede observar que el acceso a los módulos es de manera interna y, que el usuario no interviene en los cálculos efectuados.



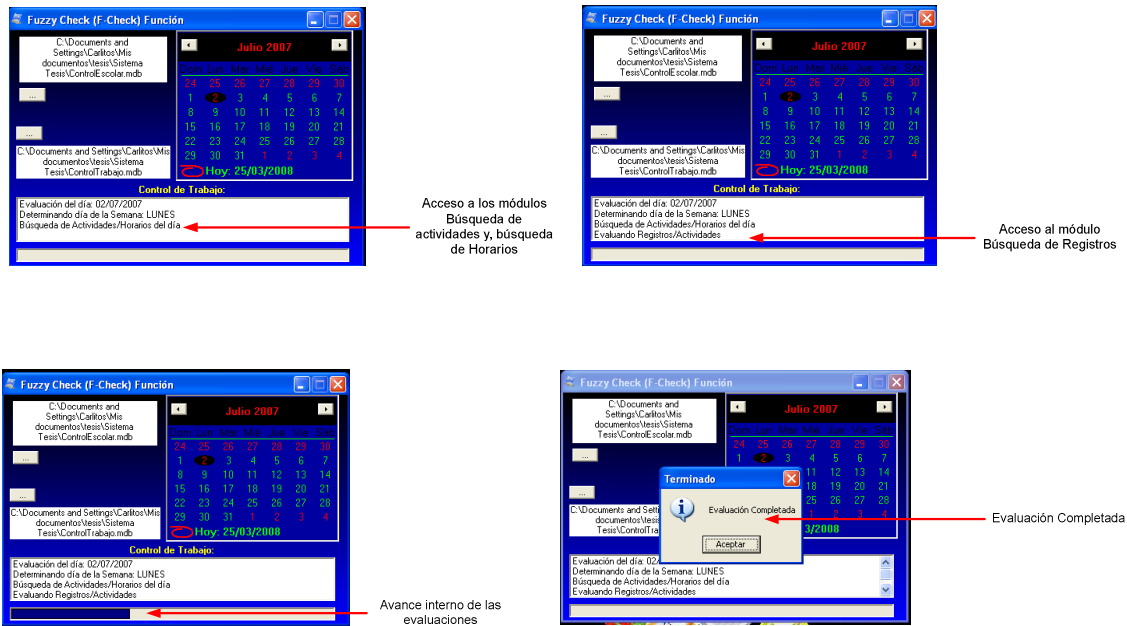


Figura 4. 5: Prueba gráfica del sistema, se muestra el funcionamiento de los módulos en el formulario.

De igual forma, se puede observar que en el corrimiento del programa, se accesa a los módulos de forma interna, un ejemplo de tal acceso lo muestra la figura 4.6, en la cual se pueden visualizar valores de entrada, que en módulos anteriores fueron obtenidos.

```

Do Until rs.EOF
  Clave_Empleado = 0
  Call Busqueda_de_Actividades(db, rs.Fields("Id_Materia"), Clave_Empleado) 'Regresa Clave Empleado
  hora_Entrada = 0
  hora_Salida = 0
  Call Busqueda_de_Horario(db, rs.Fields("Id_Horario"), hora_Entrada, hora_Salida)
  Call Busqueda_de_Registros(dbQuin, FechaINICIO, Clave_Empleado, hora_Entrada, hora_Salida)
  If profee.completo = True Then
    HEntrada = hora_Entrada & ":00"
    Call Evalua(profee, HEntrada, hora_Salida - hora_Entrada)
    Call Funciones(profee, hora_Salida - hora_Entrada)
  End If
  Call llena_tablas(dbQuin, rsQuin1, rs.Fields("Id"), FechaINICIO)
  Progreso.Value = Progreso.Value + 1
  rs.MoveNext
Loop
    
```

Figura 4. 6: Se hace el llamado al Módulo Búsqueda de registros.

Los resultados obtenidos de estos procesos, se pueden visualizar en la siguiente etapa.

4.3 3ª Etapa: Muestra de Resultados.

Esta última etapa del sistema genera los resultados en el formulario de reportes. Cabe mencionar que se pueden efectuar filtrados de la información, dependiendo las necesidades del usuario. Los resultados generados se presentan en las siguientes figuras.

Control de Reportes

Base de Datos (Control Escolar): C:\Documents and Settings\Carlitos\Mis documentos\tesis\Sistema Tesis\ControlEscolar.mdb

Base de Datos (Control de Trabajo): C:\Documents and Settings\Carlitos\Mis documentos\tesis\Sistema Tesis\ControlTrabajo.mdb

Rango de Fechas: 28/03/2008 - 28/03/2008

Etiqueta Difusa Entrada: - Etiqueta Difusa Salida: - Etiqueta Difusa T.Efectivo: - Clave: -

Consulta Generada: Ver evaluaciones Exportar a Excel Imprimir Reporte

Fecha	Horario	Clave	Empleado	Valor Entrada	Valor Salida	T. Efectivo	Valor T.Efectivo
02/07/2007	07 - 08	125	MORALES MORALES FRANCISCO MARTÍN	Checó en Tiempo	Checó en Tiempo	00:55	Tiempo Bueno
02/07/2007	07 - 08	263	FERRER ESTÉVEZ FERNANDO	Checó en Tiempo	Checó en Tiempo	00:51	Tiempo Bueno
02/07/2007	07 - 08	60	LÓPEZ SILVA MARTHA ANGELICA	Checó en Tiempo	Checó en Tiempo	00:59	Tiempo Bueno
02/07/2007	07 - 08	42	SANTIAGO LÓPEZ JOSÉ LUCIANO	Checó ligeramente Retraza	Checó en Tiempo	00:51	Tiempo Bueno
02/07/2007	07 - 08	1145	PÉREZ VIDAL ABIGAIL	Checó en Tiempo	Checó en Tiempo	00:53	Tiempo Bueno
02/07/2007	07 - 08	75	ARENAS MITRE ANGÉLICA	Checó ligeramente Retraza	Checó ligeramente antes d	00:42	Tiempo Malo
02/07/2007	07 - 08	756	ZARATE SORIANO RUBÉN	Checó ligeramente Retraza	Checó en Tiempo	00:50	Tiempo Bueno
02/07/2007	07 - 08	1291	ARELLANO GARCÍA ELIZABETH	Checó en Tiempo	Checó en Tiempo	00:52	Tiempo Bueno
02/07/2007	07 - 08	165	DE LOS SANTOS MACÍAS JESÚS HERADIO	Checó en Tiempo	Checó en Tiempo	01:00	Tiempo Bueno
02/07/2007	07 - 08	531	SÁNCHEZ GONZÁLEZ MARISELA	Checó en Tiempo	Checó en Tiempo	01:00	Tiempo Bueno
02/07/2007	07 - 08	996	RIVERA CORONA ROSA MARÍA	Checó en Tiempo	Checó en Tiempo	00:58	Tiempo Bueno
02/07/2007	07 - 08	675	NÚÑEZ GAMBDA LETICIA	Checó en Tiempo	Checó en Tiempo	00:52	Tiempo Bueno
02/07/2007	07 - 08	977	PÉREZ SILVA TERESA	Checó ligeramente Retraza	Checó en Tiempo	00:51	Tiempo Bueno
02/07/2007	07 - 08	436	PERDOMO ABUD MARÍA ISABEL	CHECÓ MAL		0	
02/07/2007	07 - 08	1256	FLORES SOTO ROBERTO MARTÍN	No hay Registros	No hay Registros	0	
02/07/2007	07 - 08	225	OSORIO GÓMEZ RICARDO	Checó en Tiempo	Checó en Tiempo	00:56	Tiempo Bueno
02/07/2007	07 - 08	113	RAI TAZARES LÓPEZ ANMILU DEL CARMEN	Checó en Tiempo	Checó en Tiempo	00:56	Tiempo Bueno

Figura 4. 7: Reporte de Resultados sin Filtro.

Control de Reportes

Base de Datos (Control Escolar): C:\Documents and Settings\Carlitos\Mis documentos\tesis\Sistema Tesis\ControlEscolar.mdb

Base de Datos (Control de Trabajo): C:\Documents and Settings\Carlitos\Mis documentos\tesis\Sistema Tesis\ControlTrabajo.mdb

Rango de Fechas: 09/07/2007 - 10/07/2007

Etiqueta Difusa Entrada: - Etiqueta Difusa Salida: - Etiqueta Difusa T.Efectivo: - Clave: -

Consulta Generada: Ver evaluaciones Exportar a Excel Imprimir Reporte

Fecha	Horario	Clave	Empleado	Valor Entrada	Valor Salida	T. Efectivo	Valor T.Efectivo
09/07/2007	07 - 08	125	MORALES MORALES FRANCISCO MARTÍN	Checó en Tiempo	Checó en Tiempo	01:00	Tiempo Bueno
09/07/2007	07 - 08	263	FERRER ESTÉVEZ FERNANDO	Checó en Tiempo	Checó en Tiempo	00:53	Tiempo Bueno
09/07/2007	07 - 08	60	LÓPEZ SILVA MARTHA ANGELICA	Checó en Tiempo	Checó en Tiempo	00:55	Tiempo Bueno
09/07/2007	07 - 08	42	SANTIAGO LÓPEZ JOSÉ LUCIANO	Checó en Tiempo	Checó en Tiempo	00:56	Tiempo Bueno
09/07/2007	07 - 08	1145	PÉREZ VIDAL ABIGAIL	Checó en Tiempo	Checó en Tiempo	00:53	Tiempo Bueno
09/07/2007	07 - 08	75	ARENAS MITRE ANGÉLICA	CHECÓ MAL		0	
09/07/2007	07 - 08	756	ZARATE SORIANO RUBÉN	Checó en Tiempo	Checó en Tiempo	00:59	Tiempo Bueno
09/07/2007	07 - 08	1291	ARELLANO GARCÍA ELIZABETH	Checó en Tiempo	Checó en Tiempo	00:53	Tiempo Bueno
09/07/2007	07 - 08	165	DE LOS SANTOS MACÍAS JESÚS HERADIO	Checó en Tiempo	Checó en Tiempo	00:57	Tiempo Bueno
09/07/2007	07 - 08	531	SÁNCHEZ GONZÁLEZ MARISELA	Checó en Tiempo	Checó en Tiempo	01:00	Tiempo Bueno
09/07/2007	07 - 08	996	RIVERA CORONA ROSA MARÍA	Checó en Tiempo	Checó en Tiempo	00:52	Tiempo Bueno
09/07/2007	07 - 08	675	NÚÑEZ GAMBDA LETICIA	Checó en Tiempo	Checó en Tiempo	00:52	Tiempo Bueno
09/07/2007	07 - 08	977	PÉREZ SILVA TERESA	Checó en Tiempo	Checó en Tiempo	01:00	Tiempo Bueno
09/07/2007	07 - 08	436	PERDOMO ABUD MARÍA ISABEL	CHECÓ MAL		0	
09/07/2007	07 - 08	1256	FLORES SOTO ROBERTO MARTÍN	Checó en Tiempo	Checó en Tiempo	00:53	Tiempo Bueno
09/07/2007	07 - 08	225	OSORIO GÓMEZ RICARDO	Checó en Tiempo	Checó en Tiempo	00:53	Tiempo Bueno
09/07/2007	07 - 08	113	RAI TAZARES LÓPEZ ANMILU DEL CARMEN	Checó en Tiempo	Checó en Tiempo	00:56	Tiempo Bueno

Figura 4. 8: Reporte de Resultados con filtrado de Fechas.

Control de Reportes

Base de Datos (Control Escolar): C:\Documents and Settings\Carlitos\Mis documentos\tesis\Sistema Tesis\ControlEscolar.mdb

Base de Datos (Control de Trabajo): C:\Documents and Settings\Carlitos\Mis documentos\tesis\Sistema Tesis\ControlTrabajo.mdb

Rango de Fechas: 09/07/2007 - 10/07/2007

Etiqueta Difusa Entrada: **Checó ligeramente Retraz** | Etiqueta Difusa Salida: - | Etiqueta Difusa T.Efectivo: - | Clave: -

Consulta Generada: Ver evaluaciones | Exportar a Excel | Imprimir Reporte

Fecha	Horario	Clave	Empleado	Valor Entrada	Valor Salida	T. Efectivo	Valor T.Efectivo
09/07/2007	07 - 09	886	ROSALES SANCHEZ VICTOR GABRIEL	Checó ligeramente Retraz	Checó ligeramente antes d	01:42	Tiempo Malo
09/07/2007	09 - 11	978	RAMIREZ CASCO ANA MARÍA	Checó ligeramente Retraz	Checó en Tiempo	01:47	Tiempo Parcialment
09/07/2007	13 - 14	1280	IBARGUEN FLORES BLANCA LILIA	Checó ligeramente Retraz	Checó en Tiempo	00:50	Tiempo Bueno
09/07/2007	15 - 16	914	MONDRAGÓN BOBADILLA MARTÍN ALFREDO	Checó ligeramente Retraz	Checó en Tiempo	00:50	Tiempo Bueno
09/07/2007	18 - 20	579	FLORES ROSAS IGNACIO CUAUHTÉMOC	Checó ligeramente Retraz	Checó en Tiempo	01:51	Tiempo Bueno
09/07/2007	19 - 21	116	LORANCA SANTOS ALFREDO	Checó ligeramente Retraz	Checó en Tiempo	01:49	Tiempo Parcialment
09/07/2007	07 - 15	1212	TORRES GONZÁLEZ ARELI	Checó ligeramente Retraz	Checó en Tiempo	07:50	Tiempo Bueno
09/07/2007	08 - 14	840	JUÁREZ FONSECA VERÓNICA	Checó ligeramente Retraz	Checó en Tiempo	05:50	Tiempo Bueno
09/07/2007	09 - 13	788	ROJAS HERNÁNDEZ MARIO	Checó ligeramente Retraz	Checó en Tiempo	03:48	Tiempo Parcialment
09/07/2007	11 - 16	247	RODRÍGUEZ GUTIÉRREZ ANA LUCIA	Checó ligeramente Retraz	Checó en Tiempo	04:50	Tiempo Bueno
09/07/2007	15 - 19	90	HERNÁNDEZ CALDERÓN MINERVA ALEJANI	Checó ligeramente Retraz	Checó en Tiempo	03:51	Tiempo Bueno
09/07/2007	16 - 21	131	TORRES ESPINOSA MARÍA VERÓNICA	Checó ligeramente Retraz	Checó en Tiempo	04:51	Tiempo Bueno
09/07/2007	08 - 17	834	ÁVILA MARTÍNEZ WENDY	Checó ligeramente Retraz	Checó en Tiempo	08:50	Tiempo Bueno
09/07/2007	09 - 18	822	Nieto Sosa Ma. Isabel	Checó ligeramente Retraz	Checó en Tiempo	08:49	Tiempo Parcialment
10/07/2007	07 - 08	977	PÉREZ SILVA TERESA	Checó ligeramente Retraz	Checó en Tiempo	00:51	Tiempo Bueno
10/07/2007	07 - 08	1256	FLORES SOTO ROBERTO MARTÍN	Checó ligeramente Retraz	Checó en Tiempo	00:51	Tiempo Bueno
10/07/2007	07 - 09	975	OCOTILA MIÑÓZ ALMA DELIA	Checó ligeramente Retraz	Checó en Tiempo	01:50	Tiempo Bueno

Figura 4. 9: Reporte de Resultados con Filtrado de Etiqueta Difusa de Entrada.

Control de Reportes

Base de Datos (Control Escolar): C:\Documents and Settings\Carlitos\Mis documentos\tesis\Sistema Tesis\ControlEscolar.mdb

Base de Datos (Control de Trabajo): C:\Documents and Settings\Carlitos\Mis documentos\tesis\Sistema Tesis\ControlTrabajo.mdb

Rango de Fechas: 09/07/2007 - 10/07/2007

Etiqueta Difusa Entrada: - | **Etiqueta Difusa Salida: Checó ligeramente antes** | Etiqueta Difusa T.Efectivo: - | Clave: -

Consulta Generada: Ver evaluaciones | Exportar a Excel | Imprimir Reporte

Fecha	Horario	Clave	Empleado	Valor Entrada	Valor Salida	T. Efectivo	Valor T.Efectivo
09/07/2007	07 - 09	886	ROSALES SANCHEZ VICTOR GABRIEL	Checó ligeramente Retraz	Checó ligeramente antes d	01:42	Tiempo Malo
10/07/2007	08 - 10	240	GÓMEZ DÍAZ HAYDEE	Checó ligeramente Retraz	Checó ligeramente antes d	01:43	Tiempo Malo
10/07/2007	09 - 11	507	ROSAS LABASTIDA JULIETA	Checó ligeramente Retraz	Checó ligeramente antes d	01:40	Tiempo Malo
10/07/2007	14 - 15	708	OSORIO SOLÍS PORFIRIO Ángel	Checó ligeramente Retraz	Checó ligeramente antes d	00:42	Tiempo Malo
10/07/2007	16 - 18	1116	CORONA HILARIO FABIÁN ARTURO	Checó ligeramente Retraz	Checó ligeramente antes d	01:44	Tiempo Malo

Figura 4. 10: Reporte de Resultados con Filtrado de Etiqueta Difusa de Salida.

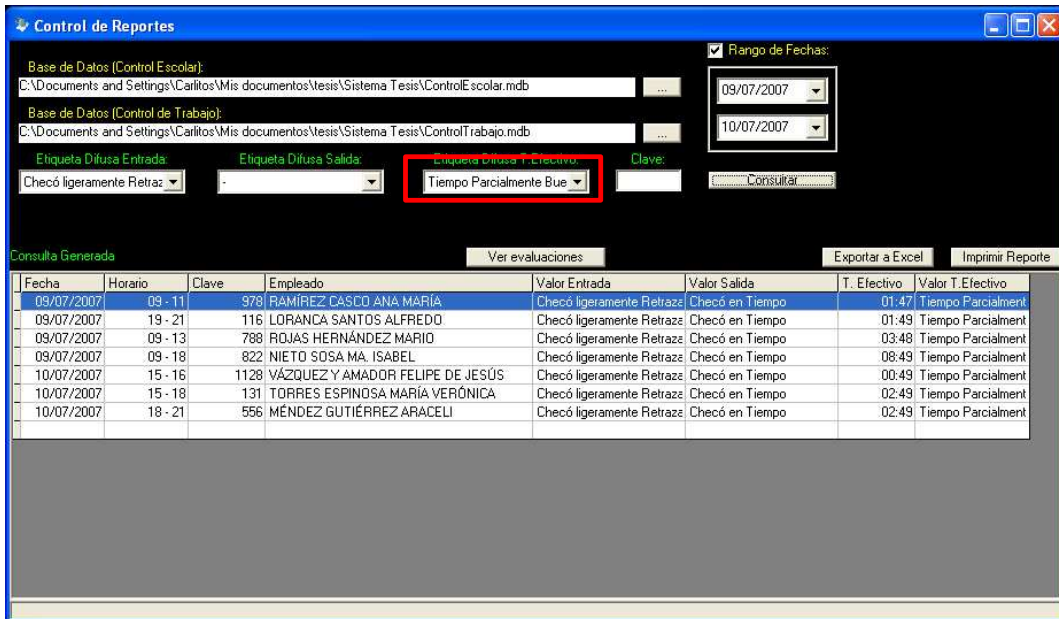


Figura 4. 11: Reporte de Resultados con Filtrado de Etiqueta Difusa de Tiempo Efectivo Parcial.

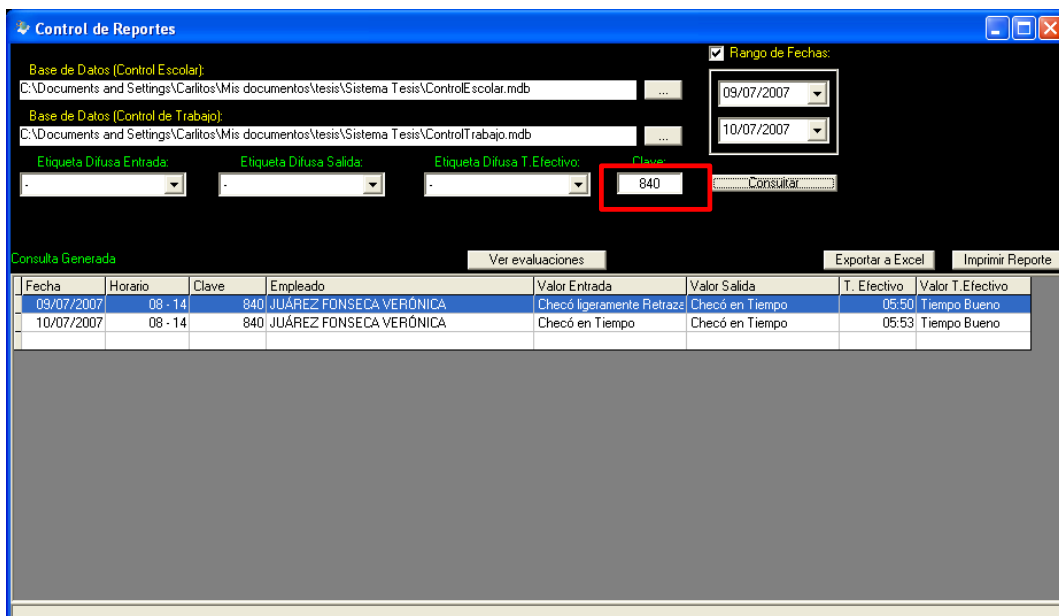


Figura 4. 12: Reporte de Resultados con Filtrado de la Clave del Empleado.

Por último, mencionaremos que tales resultados pueden ser exportados para una presentación formal de un reporte, al programa Microsoft Excel o, si lo prefiere el usuario imprimir directamente del sistema. El archivo impreso se muestra en la figura 4.13.

REPORTE EVALUACIONES
12/03/2008

Fecha	Horario	Clave	Empleado	Valor Entrada	Valor Salida	T. Efectivo	Valor T.Efectivo
09/07/2007	10 - 12	58	JUÁREZ SÁNCHEZ JOSÉ ELEUTERIO	Checó Retrazado	Checó en Tiempo	01:43	Tiempo Malo
10/07/2007	10 - 12	58	JUÁREZ SÁNCHEZ JOSÉ ELEUTERIO	Checó Retrazado	Checó Temprano	01:28	Tiempo Malo
11/07/2007	11 - 13	58	JUÁREZ SÁNCHEZ JOSÉ ELEUTERIO	Checó en Tiempo	Checó Demasiado Tempr	01:37	Tiempo Malo

Pag. 1 de 1

Figura 4. 13: Archivo impreso directamente del sistema.

Para ver el corrimiento paso a paso y el acceso a los módulos correspondientes consultar el Anexo 2.

CAPÍTULO 5.

Conclusiones.

Como parte final de este documento, se presentan las conclusiones generales, donde el propósito de este apartado es agrupar y resaltar las contribuciones más importantes de nuestro proyecto.

Se ha creado un sistema que permite clasificar la asistencia del profesorado de una institución educativa, tomando como elementos de entrada los horarios de actividades y los registros de cada uno de los empleados, logrando la separación de estos registros por cada una de las actividades y evaluarlos en las funciones difusas correspondientes, permitiendo así, ampliar el criterio de evaluación con respecto a la asistencia.

El problema más serio surgió en el análisis de los elementos que la institución nos proporciona, este problema fue el determinar con cierta exactitud que registros de los empleados correspondían a cuales de sus actividades, esto se debía a que existen varios empleados que registran: dos veces sus entradas y salidas o, una entrada seguido de una salida y nuevamente una entrada o, tres entradas seguido de dos salidas y nuevamente registra una entrada en un lapso de un minuto. Para dar solución a este problema, fue necesario el desarrollo de un algoritmo que permitió la depuración de estos registros y garantizar cual era el registro correspondiente a la entrada de la actividad, para subsecuentemente encontrar el registro de salida.

Por otra parte se implementó una aplicación que permitiera exportar los registros proporcionados por la institución al sistema.

No cabe duda que los conocimientos adquiridos permitieron tener una visión clara del problema a resolver, las estructuras de datos jugaron un papel básico en la solución del problema, sobretodo en el planteamiento del análisis de expresiones. Cabe mencionar que en la documentación de este trabajo, se utilizó el Lenguaje Unificado de Modelado (UML), lo que permitió comprender muchos de los aspectos que se deben tomar en cuenta durante el ciclo de vida de un software.

Los objetivos del proyecto fueron alcanzados, ya que se logró clasificar la asistencia utilizando como principales herramientas la lógica difusa y las bases de datos relacionales. La primera herramienta generó ese grado de pertenencia de la asistencia mientras que, la segunda herramienta permitió almacenar la información necesaria para esas evaluaciones y los resultados que se generaron.

5.1 Trabajos futuros.

Dentro de los trabajos a futuro, se puede ampliar el sistema a que no solamente maneje horarios cerrados, sino que también acepte medias horas con el fin de poder utilizar este software con criterios de evaluación similares.

Se puede llegar a migrar este proyecto a otros lenguajes de programación como pueden ser Delphi, C++, C#, en plataforma .NET por mencionar algunas, esta migración puede lograr que las consultas se optimicen, las bases de datos puedan ser cambiadas a otros tipos, para lograr un funcionamiento más rápido en ese aspecto, aunque el manejo de las variables de tipo hora cambian bastante de lenguaje a lenguaje.

Muchas veces vemos como imposible la realización de un proyecto, sobre todo cuando se empieza, sin embargo, al alcanzar y lograr ciertos objetivos, éstos influyen en que los siguientes sean más sencillos de lograr.

Anexo I.

Asignación de etiquetas difusas.

En esta parte se presenta la forma en que se etiquetan los resultados en las funciones difusas y la forma en que se tomó dicha evaluación.

A1.1 Evaluando Funciones Difusas.

Como se puede observar en el capítulo 2 - sección 2.4 y en el capítulo 3 - sección 3.1.6.1.1, se hace la presentación de las funciones difusas a utilizar y los valores de los parámetros de ellas.

Para el mejor entendimiento del trabajo con las funciones difusas consideremos el siguiente ejemplo:

Tengamos un maestro que tenga actividades en un horario de 7:00 a.m. a las 9:00 a.m. y los registros de entrada y salida de la actividad en que checó sean 7:10:00 a.m. y 8:52:00 a.m. respectivamente. Tomando esta información, los parámetros a utilizar quedarían representados en las tablas A1.1, A1.2 y A1.3.

A1.1.1 Funciones Difusas para el Registro de Entrada.

Parámetros para Evaluar Registro de Entrada	Valor
x	07:10:00 a.m.
a	07:00 a.m.
m	07:11 a.m.
b	07:08 a.m.
c	07:18 a.m.
d	07:21 a.m.

Tabla A1. 1: Parámetros de Funciones Difusas Entrada.

La gráfica de las funciones difusas para el registro de entrada con los valores de la tabla A1.1, queda presentada en la figura A1.1.

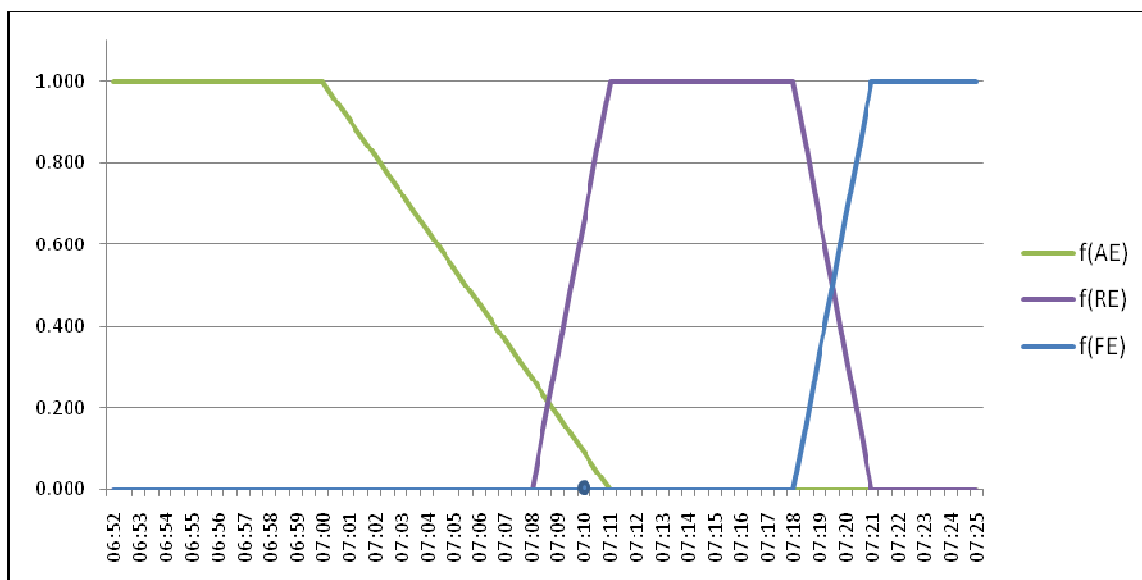


Figura A1. 1: Gráfica de Funciones Difusas para el registro de entrada

Como se puede observar en la figura A1.1, el registro de entrada x se localiza en la intersección entre la función de asistencia y la retardo, en estos casos se evalúa en ambas funciones para determinar a cuál de estas pertenece en mayor grado, en este caso pertenece en mayor grado a la función de retardo. La evaluación en las funciones difusas queda de la siguiente forma.

$$FAE(x) = \frac{m - x}{m - a} = \frac{(07:11) - (07:10)}{(07:11) - (07:10)} = 0.90909$$

$$FRE(x) = \frac{x - b}{m - b} = \frac{(07:10) - (07:08)}{(07:11) - (07:08)} = 0.181818$$

A1.1.1 Funciones Difusas para el Registro de Salida.

Parámetros para Evaluar Registro de Salida	Valor
x	08:52:00 a.m.
p	09:00 a.m.
q	08:49 a.m.

Tabla A1. 2: Parámetros de Funciones Difusas Salida.

Tomando los valores de los parámetros de la tabla A1.2, la gráfica de las funciones difusas correspondientes queda representada en la siguiente figura.

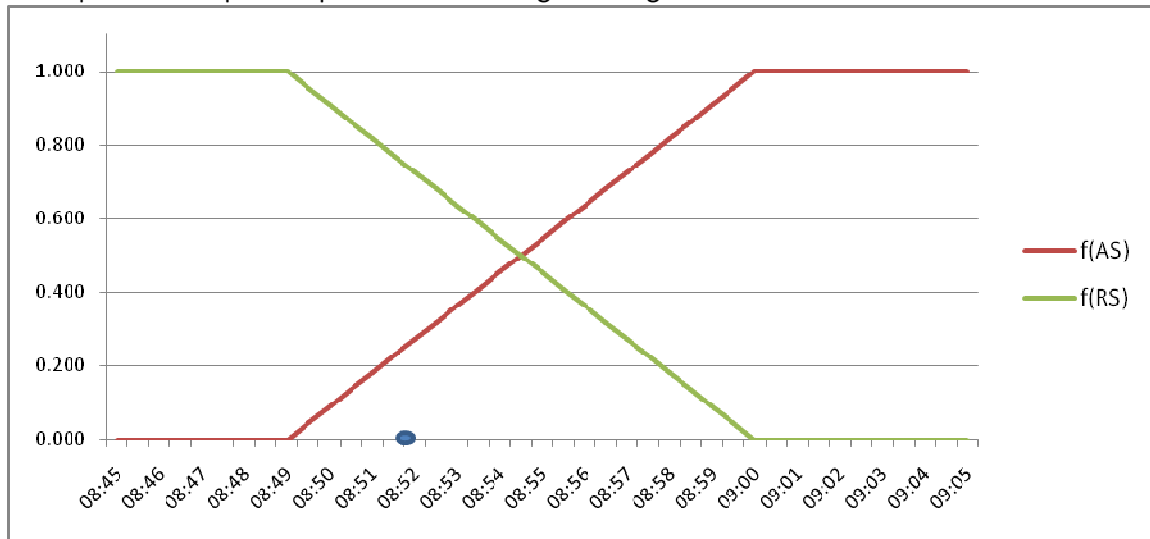


Figura A1. 2: Gráfica de Funciones Difusas para el registro de Salida

Como se puede observar en la figura anterior, el registro de salida x es localizado en la intersección de las dos funciones, en estos casos se procede de igual forma que en el punto anterior, es decir, evaluar x en ambas funciones para determinar a cuál de estas pertenece en mayor grado, en este caso pertenece en mayor grado a la función de retiro previo a la tolerancia. La evaluación en las funciones difusas queda de la siguiente forma.

$$FAS(x) = \frac{x - q}{p - q} = \frac{(08 : 52) - (08 : 49)}{(09 : 00) - (08 : 49)} = 0.272727$$

$$FRS(x) = \frac{p - x}{p - q} = \frac{(09 : 00) - (08 : 52)}{(09 : 00) - (08 : 49)} = 0.727272$$

A1.1.1 Funciones Difusas para el Tiempo Efectivo Parcial

Parámetros para Evaluar Tiempo Efectivo Parcial	Valor
x	01:42
a	01:45
b	01:50

Tabla A1. 3: Parámetros de Funciones Difusas Tiempo Parcial Efectivo.

Siguiendo de igual forma que el punto anterior, la gráfica de las funciones difusas para el tiempo efectivo parcial utilizando la tabla A1.3, queda mostrada en la figura A1.3.

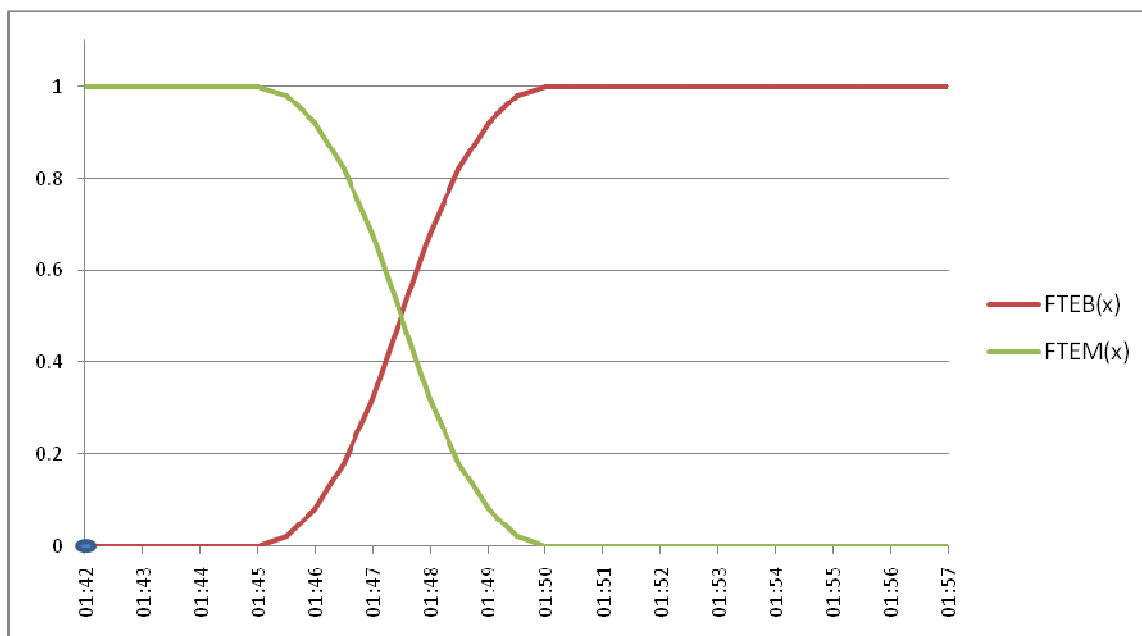


Figura A1. 3: Gráfica de Funciones Difusas para el tiempo efectivo parcial.

Como se puede observar en la figura A1.3, el registro de salida x tiene un valor de 1 en la función de tiempo malo y, un valor de 0 en la función de tiempo considerable.

A1.1 Etiquetando los Resultados.

Para asignar las etiquetas difusas correspondientes, se utilizan los siguientes criterios.

A1.1.1 Etiquetas Difusas para el registro de entrada.

Para etiquetar el registro de entrada es: si la función de asistencia es igual a 1, la etiqueta que se asigna es "Checó en Tiempo". Si en su lugar, la función de retardo es la que alcanza el valor de 1, la etiqueta que se asigna es "Checó Retrasado". Por el contrario, si la función que alcanza el valor de 1 es la función de inasistencia, la etiqueta correspondiente es "Checó con falta". Cabe mencionar que estas etiquetas son únicamente cuando sus funciones alcanzan el valor máximo.

Existen otras 2 etiquetas difusas que se asignan, éstas están en relación con las dos intersecciones que presentan nuestras gráficas, estas etiquetas son "Checó ligeramente Retrasado" (intersección Funciones Asistencia y Retardo) y "Checó bastante Retrasado" (intersección Funciones Retardo e Inasistencia). Estas etiquetas resultan de ampliar el criterio de evaluación para obtener mayor información acerca del registro de entrada.

A1.1.2 Etiquetas Difusas para el registro de salida.

Las etiquetas que se utilizan para el registro de salida son: "*Checó en Tiempo*" si la función difusa salida en tiempo es 1 y, "*Checó antes de lo permitido*" si la función salida antes de la tolerancia es 1.

De igual forma que en el criterio para el registro de entrada, existe una tercera etiqueta difusa que resulta de la intersección entre ambas funciones, dicha etiqueta es "*Checó ligeramente antes de lo permitido*".

A1.1.2 Etiquetas Difusas para el tiempo efectivo parcial.

Para etiquetar el tiempo efectivo parcial existen cuatro etiquetas, dos etiquetas para los valores máximos de las funciones ("*Tiempo bueno*" si considerable es 1, "*Tiempo malo*" si malo es 1) y, a diferencia de los casos anteriores, en la situación de la intersección, se amplía a dos etiquetas difusas dependiendo a cuál de estas dos funciones pertenece más ("*Tiempo Parcialmente Bueno*" si pertenece más a la función Considerable y, "*Tiempo Parcialmente Malo*" de lo contrario).

De esta forma es como se etiquetan dentro del sistema, cabe aclarar que estas etiquetas permiten efectuar los filtrados en la parte de reportes.

Anexo II.

Corrimiento Paso a Paso.

En esta parte se presentan imágenes del funcionamiento interno para mostrar el acceso a los módulos del sistema y sus procedimientos internos.

A2.1 Módulo Selección de Fecha(s).

Al dar click sobre el área de la selección de fechas el acceso al módulo es presentado en la figura A2.1.

```
⇒ Private Sub MonthView1_DateClick(ByVal DateClicked As Date)
  Dim db As Database
  Dim rs As Recordset

  Dim dbQuin As Database
  Dim rsdia As Recordset

  Dim rsQuin1 As Recordset
  Dim FechaINICIO, FechaFINAL As Variant
  Dim diaSEMANA As String

  Dim Clave_Empleado As Integer
  Dim hora_Entrada, hora_Salida As Integer
  Dim HEntrada As Date

  ⇒ FechaINICIO = MonthView1.SelStart
  FechaFINAL = MonthView1.SelEnd
  MonthView1.SelEnd = 02/07/2007

  Set db = OpenDatabase(Label2)
```

Figura A2. 1: Acceso al módulo selección de Fecha(s).

La figura A2.2 presenta la selección de casos para el comando MonthView.DayOfWeek presentada en la tabla 3.2 página 54 así como la ejecución de la Consulta1 (véase 3.1.2.1.1).

```

MonthView1.Value = FechaINICIO
If MsgBox("Desea Calcular el día: " & MonthView1.Value, vbYesNo, "Calcular Fecha?")
    GoTo nocalcula
End If

List1.AddItem ("Evaluación del día: " & MonthView1.Value)
Select Case MonthView1.DayOfWeek
    Case 1: diaSEMANA = "DOMINGO"
    Case 2: diaSEMANA = "LUNES"
    Case 3: diaSEMANA = "MARTES"
    Case 4: diaSEMANA = "MIERCOLES"
    Case 5: diaSEMANA = "JUEVES"
    Case 6: diaSEMANA = "VIERNES"
    Case 7: diaSEMANA = "SÁBADO"
End Select
List1.AddItem ("Determinando día de la Semana: " & diaSEMANA)
Set rsdia = db.OpenRecordset("Select * from Ctrl1_Dias where Ucase(DiaSemana) = '" & diaSEMANA &
If rsdia.AbsolutePosition <> -1 Then

```

Figura A2. 2: Determinando día de la semana a evaluar y ejecución de la Consulta1.

Una vez ejecutada la Consulta1, se ejecuta la Consulta2 como se muestra en la figura A2.3 para proceder a acceder a los módulos siguientes.

```

Set rs = db.OpenRecordset("Select * From Horarios where Id_Dia = " & rsdia.Fields("Id") & " order by Id_Horario")
List1.AddItem ("Búsqueda de Actividades/Horarios del día")
Progreso.Min = 0
If rs.AbsolutePosition <> -1 Then
    rs.MoveLast
    Progreso.Max = rs.AbsolutePosition + 1
    rs.MoveFirst
End If
Progreso.Value = 0

```

Figura A2. 3: Efectuando Consulta2 para llamar a los siguientes módulos.

A2.2 Módulo Búsqueda de Actividades.

Para que este módulo trabaje necesita el identificador de la actividad proporcionado por la Consulta2 (*Id_Materia*). La figura A2.4 muestra el acceso al módulo para efectuar Consulta3 (referencia en 3.1.3.1.1) y poder retornar el identificador del empleado que lleve a cabo la actividad (*Clave_Empleado*), ver figura A2.5.

```

Do Until rs.EOF
    Clave_Empleado = 0
    Call Busqueda_de_Actividades(db, rs.Fields("Id_Materia"), Clave_Empleado) 'Regresa Clave Empleado
    hora_Entrada = 0
    hora_Salida = 0
    Call Busqueda_de_Horario(db, rs.Fields("Id_Horario"), hora_Entrada, hora_Salida)
    Call Busqueda_de_Registros(dbQuin, FechaINICIO, Clave_Empleado, hora_Entrada, hora_Salida)
    If profee.completo = True Then
        HEntrada = hora_Entrada & ":00"
        Call Evalua(profee, HEntrada, hora_Salida - hora_Entrada)
        Call Funciones(profee, hora_Salida - hora_Entrada)
    End If
    Call llena_tablas(dbQuin, rsQuin1, rs.Fields("Id"), FechaINICIO)
    Progreso.Value = Progreso.Value + 1
rs.MoveNext
Loop

```

Figura A2. 4: Llamada interna al módulo búsqueda de actividades.

```

Do Until rs.EOF
  Clave_Empleado = 0
  Call Busqueda_de_Actividades(db, rs.Fields("Id_Materia"), Clave_Empleado) 'Regresa Clave Empleado
  hora_Entrada = 0
  hora_Salida = 0
  Call Busqueda_de_Horario(db, rs.Fields("Id_Horario"), hora_Entrada, hora_Salida)
  Call Busqueda_de_Registros(dbQuin, FechaINICIO, Clave_Empleado, hora_Entrada, hora_Salida)
  If profee.completo = True Then
    HEntrada = hora_Entrada & ":00"
    Call Evalua(profee, HEntrada, hora_Salida - hora_Entrada)
    Call Funciones(profee, hora_Salida - hora_Entrada)
  End If
  Call llena_tablas(dbQuin, rsQuin1, rs.Fields("Id"), FechaINICIO)
  Progreso.Value = Progreso.Value + 1
rs.MoveNext
Loop

```

Figura A2. 5: Parámetro Clave_Empleado devuelto por el módulo.

A2.3 Módulo Búsqueda de Horarios.

El acceso a este módulo requiere el identificador del horario de la actividad proporcionado por la Consulta2 (*Id_Horario*), este proceso se puede ver en la figura A2.6.

```

Do Until rs.EOF
  Clave_Empleado = 0
  Call Busqueda_de_Actividades(db, rs.Fields("Id_Materia"), Clave_Empleado) 'Regresa Clave Empleado
  hora_Entrada = 0
  hora_Salida = 0
  Call Busqueda_de_Horario(db, rs.Fields("Id_Horario"), hora_Entrada, hora_Salida)
  Call Busqueda_de_Registros(dbQuin, FechaINICIO, Clave_[hora_Entrada=0]ra_[hora_Salida=0]ora_Salida)
  If profee.completo = True Then
    HEntrada = hora_Entrada & ":00"
    Call Evalua(profee, HEntrada, hora_Salida - hora_Entrada)
    Call Funciones(profee, hora_Salida - hora_Entrada)
  End If
  Call llena_tablas(dbQuin, rsQuin1, rs.Fields("Id"), FechaINICIO)
  Progreso.Value = Progreso.Value + 1
rs.MoveNext
Loop

```

Figura A2. 6: Llamada interna al módulo búsqueda de horarios.

El módulo efectúa la Consulta4 (sección 3.1.4.1.1) para regresar la hora de inicio y finalización de la actividad (*hora_Entrada* y *hora_Salida*) como se muestra en la figura A2.7.

```

Do Until rs.EOF
  Clave_Empleado = 0
  Call Busqueda_de_Actividades(db, rs.Fields("Id_Materia"), Clave_Empleado) 'Regresa Clave Empleado
  hora_Entrada = 0
  hora_Salida = 0
  Call Busqueda_de_Horario(db, rs.Fields("Id_Horario"), hora_Entrada, hora_Salida)
  Call Busqueda_de_Registros(dbQuin, FechaINICIO, Clave_[hora_Entrada=7]ra_[hora_Salida=8]ora_Salida)
  If profee.completo = True Then
    HEntrada = hora_Entrada & ":00"
    Call Evalua(profee, HEntrada, hora_Salida - hora_Entrada)
    Call Funciones(profee, hora_Salida - hora_Entrada)
  End If
  Call llena_tablas(dbQuin, rsQuin1, rs.Fields("Id"), FechaINICIO)
  Progreso.Value = Progreso.Value + 1
rs.MoveNext
Loop

```

Figura A2. 7: Parámetros de retorno del módulo son hora_entrada y hora_salida de la actividad.

A2.4 Módulo Búsqueda de Registros.

En la figura A2.8, se puede apreciar el llamado al módulo búsqueda de registros, el cual para efectuar la Consulta5 (véase 3.1.5.1.1) tiene que delimitar el rango de los registros, esto se presenta en la figura A2.9.

```
Public Sub Busqueda_de_Registros(dbQuin As Database, fechas As Variant, claveempleado As Variant, h_entrada As
Dim HEntrada As Date
Dim limiteantes As Date
Dim limitedespués As Date
Dim rsQuin As Recordset

Const limite = 10

THoras = h_salida - h_entrada
HEntada = h_entrada & ":00"
```

Figura A2. 8: Accediendo al módulo búsqueda de registros.

```
profec.Clave = claveempleado
profec.Id_Entrada = 0
profec.Reg_entrada = 0
profec.Id_salida = 0
profec.Reg_salida = 0
profec.completo = False
profec.fAE = 0
profec.fAS = 0
profec.fIE = 0
profec.fRE = 0
profec.fRS = 0
profec.ValorEntrada = ""
profec.ValorSalida = ""
profec.ValorFTEfectivo = ""
profec.tiempoefectivo = 0
profec.Fbueno = 0
profec.Fmalo = 0

antes = Hour(limiteantes) + ((Minute(limiteantes) + (Second(limiteantes) / 60)) / 60)
después = Hour(limitedespués) + ((Minute(limitedespués) + (Second(limitedespués) / 60)) / 60)

Set rsQuin = dbQuin.OpenRecordset("Select * From chequeos where str(Fecha) = '" & fechas & "' and Clave = '" & profec
```

Figura A2. 9: Marcando rango para los registros y efectuar Consulta5.

Si la Consulta5 contiene elementos, se hace el llamado al procedimiento Checa_Entrada (sección 3.1.5.1.1.2) como se muestra en las figuras A2.10 y A2.11.

```
If rsQuin.AbsolutePosition <> -1 Then
    Call Checa_Entrada(rsQuin)
Else
    profec.completo = False
```

Figura A2. 10: Llamado al procedimiento Checa_Entrada.

```
Public Sub Checa_Entrada(rst As Recordset)
If (UCase(rst.Fields("Concepto")) = "IN_ENTRY") Or (UCase(rst.Fields("Concepto")) = "VERIFY_OK") Then
    rst.Fields("Concepto") = "IN_ENTRY"
    profec.Id_Entrada = rst.Fields("Id")
    rst.MoveNext
    Call Checa_Entrada2(rst)
Else
    If (UCase(rst.Fields("Concepto")) = "OUT_ENTRY") Then
        profec.Reg_entrada = rst.Fields("Hora")
        profec.Id_Entrada = rst.Fields("Id")
```

Figura A2. 11: Comprobando el valor del registro activo en el campo <Concepto>.

Si existe alguna posible entrada de la actividad, el procedimiento Checa_Entrada efectúa el llamado al procedimiento Checa_Entrada2 (sección 3.1.5.1.1.3), ver figura A2.12.

```

Public Sub Checa_Entrada(rst As Recordset)
If (UCase(rst.Fields("Concepto")) = "IN_ENTRY") Or (UCase(rst.Fields("Concepto")) = "VERIFY_OK") Then
profee.Reg_entrada = rst.Fields("Hora")
profee.Id_Entrada = rst.Fields("Id")
rst.MoveNext
⇒ Call Checa_Entrada2(rst)
Else

```

Figura A2. 12: Llamado al procedimiento Checa_Entrada2.

La labor de este procedimiento es determinar si el siguiente registro es el registro de salida de la actividad para proceder al llamado del procedimiento Checa_Salida (sección 3.1.5.1.1.4), estas actividades se presentan en las figuras A2.13 y A2.14.

```

Public Sub Checa_Entrada2(rst As Recordset)
If rst.EOF Then
profee.completo = False
profee.ValorEntrada = "CHECÓ MAL"
Exit Sub
End If
⇒ If (UCase(rst.Fields("Concepto")) = "OUT_ENTRY") Then
rst.Fields("Concepto")="OUT_ENTRY" entrada) + (Second(profee.Reg_entrada) / 60)
u = Hour(profee.Reg_entrada) + (Min / 60)
Min = Minute(rst.Fields("Hora")) + (Second(rst.Fields("Hora")) / 60)

```

Figura A2. 13: Comprobando valor en el campo activo.

```

If (UCase(rst.Fields("Concepto")) = "OUT_ENTRY") Then
Min = Minute(profee.Reg_entrada) + (Second(profee.Reg_entrada) / 60)
u = Hour(profee.Reg_entrada) + (Min / 60)
Min = Minute(rst.Fields("Hora")) + (Second(rst.Fields("Hora")) / 60)
v = Hour(rst.Fields("Hora")) + (Min / 60)
If (v - u) < (20 / 60) Then
rst.MoveNext
If rst.EOF Then
rst.MovePrevious
Call Checa_Salida(rst)
Else
Call Checa_Entrada(rst)
End If
Else
⇒ Call Checa_Salida(rst)
End If
Else
If (UCase(rst.Fields("Concepto")) = "IN_ENTRY") Then
Min = Minute(profee.Reg_entrada) + (Second(profee.Reg_entrada) / 60)
u = Hour(profee.Reg_entrada) + (Min / 60)

```

Figura A2. 14: Llamada al procedimiento Checa_Salida.

Si es que los dos registros son localizados por estos procedimientos se completa el ciclo de búsqueda, el cual se representa en las figuras A2.15 y A2.16.

```

Public Sub Checa_Salida(rst As Recordset)
If rst.EOF Then
    profee.completo = False
    profee.ValorEntrada = "CHECÓ MAL"
    Exit Sub
End If
> If (UCase(rst.Fields("Concepto")) = "OUT_ENTRY") Or (UCase(rst.Fields("Concepto")) = "VERIFY_OK") Then
    rst.Fields("Concepto") = "OUT_ENTRY"
    profee.Id_salida = rst.Fields("Id")
    profee.completo = True
End If
End Sub
    
```

Figura A2. 15: Comprobando valor en el campo activo.

```

If (UCase(rst.Fields("Concepto")) = "OUT_ENTRY") Or (UCase(rst.Fields("Concepto")) = "VERIFY_OK") Then
    profee.Reg_salida = rst.Fields("Hora")
    profee.Id_salida = rst.Fields("Id")
    profee.completo = True
End If
End Sub
    
```

Figura A2. 16: Activando variable completo para validad que existen ambos registros.

Cabe recordar que para acceder al siguiente módulo (Evaluación) es necesario que se hayan localizados los dos registros para la actividad como se muestra en la siguiente figura.

```

Do Until rs.EOF
    Clave_Empleado = 0
    Call Busqueda_de_Actividades(db, rs.Fields("Id_Materia"), Clave_Empleado) 'Regresa Clave Empleado
    hora_Entrada = 0
    hora_Salida = 0
    Call Busqueda_de_Horario(db, rs.Fields("Id_Horario"), hora_Entrada, hora_Salida)
    Call Busqueda_de_Registros(dbQuin, FechaINICIO, Clave_Empleado, hora_Entrada, hora_Salida)
    > If profee.completo = True Then
        profee.completo = Verdadero
        HEntrada = hora_Entrada & ":00"
        Call Evalua(profee, HEntrada, hora_Salida - hora_Entrada)
        Call Funciones(profee, hora_Salida - hora_Entrada)
    End If
    Call llena_tablas(dbQuin, rsQuini, rs.Fields("Id"), FechaINICIO)
    Progreso.Value = Progreso.Value + 1
    rs.MoveNext
Loop
    
```

Figura A2. 17: Confirmando que haya encontrado ambos registros para llamar al siguiente módulo.

A2.5 Módulo de Evaluación.

El acceso al módulo de evaluaciones es presentado en las figuras A2.18 y A2.19.

```

Do Until rs.EOF
    Clave_Empleado = 0
    Call Busqueda_de_Actividades(db, rs.Fields("Id_Materia"), Clave_Empleado) 'Regresa Clave Empleado
    hora_Entrada = 0
    hora_Salida = 0
    Call Busqueda_de_Horario(db, rs.Fields("Id_Horario"), hora_Entrada, hora_Salida)
    Call Busqueda_de_Registros(dbQuin, FechaINICIO, Clave_Empleado, hora_Entrada, hora_Salida)
    If profee.completo = True Then
        HEntrada = hora_Entrada & ":00"
        > Call Evalua(profee, HEntrada, hora_Salida - hora_Entrada)
        Call Funciones(profee, hora_Salida - hora_Entrada)
    End If
    Call llena_tablas(dbQuin, rsQuini, rs.Fields("Id"), FechaINICIO)
    Progreso.Value = Progreso.Value + 1
    rs.MoveNext
Loop
    
```

Figura A2. 18: Llamado al módulo de evaluaciones.

```

⇒ Public Sub Evalua(profesor As profe, entra As Date, horas As Variant)
  Dim limtiempo As Date
  Dim HEntrada As Date
  Dim nuevaentrada As Date
  Dim mal, bien As Currency

  nuevaentrada = profesor.Reg_entrada

  Min = Minute(nuevaentrada) + (Second(nuevaentrada) / 60)
  u = Hour(nuevaentrada) + (Min / 60)
  Min = Minute(profesor.Reg_salida) + (Second(profesor.Reg_salida) / 60)
  v = Hour(profesor.Reg_salida) + (Min / 60)

```

Figura A2. 19: Accediendo al módulo de evaluación.

Sus funciones principales del módulo de evaluación son: evaluar los registros en las funciones difusas para el registro de entrada (figura A2.20), para el registro de salida (figura A2.21) y el tiempo efectivo parcial (figuras A2.22, A2.23 y A2.24).

```

  If a >= X Then
    profesor.fAE = 1
  ElseIf X < m Then
⇒      profesor.fAE = (m - X) / (m - a)
    If X > B Then
      profesor.fRE = (X - B) / (m - B)
    End If
    ElseIf X <= c Then
      profesor.fRE = 1
    ElseIf X < d Then
      profesor.fRE = (d - X) / (d - c)
      profesor.fIE = (X - c) / (d - c)
    ElseIf X >= d Then
      profesor.fIE = 1
    End If

```

Figura A2. 20: Evaluando en Funciones Difusas para el registro de entrada.

```

  p = HS
  q = HS - (11 / 60)
  X2 = registrosalida
  If X2 <= q Then
    profesor.fRS = 1
  ElseIf X2 < p Then
⇒      profesor.fAS = (X - q) / (p - q)
      profesor.fRS = (p - X) / (p - q)
  ElseIf X2 >= p Then
    profesor.fAS = 1
    profesor.fRS = 0
  End If
  mal = 0
  bien = 0

```

Figura A2. 21: Evaluando en Funciones Difusas para el registro de salida.

```

If profesor.tiempoefectivo = 1 Then
    hh = 1
    mm = 0
    ss = 0
    Call functiempo(60, profesor, mal, bien, horas)
ElseIf profesor.tiempoefectivo < 1 Then
    profesor.tiempoefectivo = 0.916666666666667
    mm = profesor.tiempoefectivo * 60
    Call functiempo(mm, profesor, mal, bien, horas)
    mm = Fix(mm)
Else
    mm = profesor.tiempoefectivo * 60

```

Figura A2. 22: Verificando tiempo efectivo parcial.

```

Call functiempo(60, profesor, mal, bien, horas)
ElseIf profesor.tiempoefectivo < 1 Then
    hh = 0
    mm = profesor.tiempoefectivo * 60
    Call functiempo(mm, profesor, mal, bien, horas)
    mm = Fix(mm)
Else

```

Figura A2. 23: Llamada al procedimiento para evaluar el tiempo efectivo parcial.

```

If tiempo <= refmalo Then
    malo = 1
ElseIf tiempo <= refcorte Then
    malo = 1 - (2 * ((tiempo - refmalo) / (refbueno - refmalo)) ^ 2)
ElseIf tiempo < refbueno Then
    tiempo = 55
    refbueno = 50
    malo = 2 * (refbueno - tiempo) / (refbueno - refmalo) ^ 2
Else
    malo = 0
End If

```

Figura A2. 24: Evaluando en Funciones Difusas para el Tiempo Efectivo Parcial.

Una vez que se ha evaluado en las funciones difusas correspondientes, se les tienen que asignar sus etiquetas difusas, estas asignaciones se presentan en las figuras A2.25, A2.26, A2.27 y A2.28.

```

Do Until rs.EOF
    Clave_Empleado = 0
    Call Busqueda_de_Actividades(db, rs.Fields("Id_Materia"), Clave_Empleado) 'Regresa Clave Empleado
    hora_Entrada = 0
    hora_Salida = 0
    Call Busqueda_de_Horario(db, rs.Fields("Id_Horario"), hora_Entrada, hora_Salida)
    Call Busqueda_de_Registros(dbQuin, FechaINICIO, Clave_Empleado, hora_Entrada, hora_Salida)
    If profee.completo = True Then
        HEntrada = hora_Entrada & ":00"
        Call Evalua(profee, HEntrada, hora_Salida - hora_Entrada)
        Call Funciones(profee, hora_Salida - hora_Entrada)
    End If
    Call llena_tablas(dbQuin, rsQuin1, rs.Fields("Id"), FechaINICIO)
    Progreso.Value = Progreso.Value + 1
    rs.MoveNext
Loop

```

Figura A2. 25: Llamada al procedimiento para asignación de etiquetas difusas.

```

Public Sub Funciones(profesor As profe, horas As Variant)
  If profesor.fAE <> 0 Then
    If profesor.fAE = 1 Then
      profesor.ValorEntrada = "Checó en Tiempo"
    ElseIf profesor.fAE >= profesor.fRE Then
      profesor.ValorEntrada = "Checó en Tiempo"
    Else
      profesor.ValorEntrada = "Checó ligeramente Retrazado"
    End If
  Else
    If profesor.fRE = 1 Then
      profesor.ValorEntrada = "Checó Retrazado"
    ElseIf profesor.fRE >= profesor.fIE Then
      profesor.ValorEntrada = "Checó bastante Retrazado"
    Else
      profesor.ValorEntrada = "Checó con falta"
    End If
  End If
End Sub

```

Figura A2. 26: Asignando etiqueta difusa para el registro de entrada.

```

If profesor.fAS <> 0 Then
  If profesor.fAS = 1 Then
    profesor.ValorSalida = "Checó en Tiempo"
  ElseIf profesor.fAS >= profesor.fRS Then
    profesor.ValorSalida = "Checó en Tiempo"
  Else
    profesor.ValorSalida = "Checó ligeramente antes de lo permitido"
  End If
ElseIf profesor.fRS = 1 Then
  profesor.ValorSalida = "Checó antes de lo permitido"
End If

```

Figura A2. 27: Asignando etiqueta difusa para el registro de salida.

```

If malo = 1 Then
  profesor.ValorFTEfectivo = "Tiempo Malo"
ElseIf malo > bueno Then
  profesor.ValorFTEfectivo = "Tiempo Parcialmente Malo"
ElseIf bueno = 1 Then
  profesor.ValorFTEfectivo = "Tiempo Bueno"
Else
  profesor.ValorFTEfectivo = "Tiempo Parcialmente Bueno"
End If

```

Figura A2. 28: Asignando Etiqueta Difusa para el tiempo efectivo parcial.

Teniendo las evaluaciones y las etiquetas difusas, se procede a hacer el llenado de las tablas en la base de datos Control de Trabajo, este procedimiento se puede ver en la siguiente figura.

```

Do Until rs.EOF
    Clave_Empleado = 0
    Call Busqueda_de_Actividades(db, rs.Fields("Id_Materia"), Clave_Empleado) 'Regresa Clave Empleado
    hora_Entrada = 0
    hora_Salida = 0
    Call Busqueda_de_Horario(db, rs.Fields("Id_Horario"), hora_Entrada, hora_Salida)
    Call Busqueda_de_Registros(dbQuin, FechaINICIO, Clave_Empleado, hora_Entrada, hora_Salida)
    If profee.completo = True Then
        HEntrada = hora_Entrada & ":00"
        Call Evalua(profee, HEntrada, hora_Salida - hora_Entrada)
        Call Funciones(profee, hora_Salida - hora_Entrada)
    End If
    Call llena_tablas(dbQuin, rsQuin1, rs.Fields("Id"), FechaINICIO)
    Progreso.Value = Progreso.Value + 1
rs.MoveNext
Loop

```

Figura A2. 29: Llamada al procedimiento que llena las tablas con los resultados de la evaluación.

Bibliografía

- [1]. Babuska R, *Fuzzy and Neural Control*, Delft University of Technology, the Netherlands, 2001
- [2]. B.P. Buckles, F.E. Petry, "A Fuzzy Representation of Data for Relational Databases". *Fuzzy Sets and Systems*, 7, pp. 213-226, 1982.
- [3]. Dualibe Carlos, Verleysen Michel, Jespers P.G.A. "Design of Analog Fuzzy Logic Controllers in CMOS Technologies", Kluwer Academic Publishers 2003.
- [4]. D'Negri Carlos Eduardo, De Vito Eduardo Luis, "Introducción al Razonamiento Aproximado: Lógica Difusa", *Revista de Medicina Respiratoria*, 2006 No 4: 126 – 136.
- [5]. E. F. Codd, "Extending the Data Base Relational Model to Capture More Meaning". 1979
<http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/c/Codd:E= F=.html> -- 2005
- [6]. E. F. Codd, "Missing Information (Applicable and Inapplicable) in Relational Databases.", 1986.
<http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/c/Codd:E= F=.html> -- 2005
- [7]. E. F. Codd, "More Commentary on Missing Information in Relational Databases (Applicable and Inapplicable Information)", 1987.
<http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/c/Codd:E= F=.html> -- 2005
- [8]. E. F. Codd, "The Relational Model for Database Management", 1990.
<http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/c/Codd:E= F=.html> -- 2005
- [9]. C. J. DATE, "A Formal Definition of the relational Model", Association for Computing Machinery, 1982
- [10]. Galindo G.J. "Conjuntos y Sistemas Difusos", Depto. De Lenguas y Cs. De la Computación, Universidad de Málaga, 2003.
- [11]. Galindo G.J.: "Tratamiento de la imprecisión en Bases de Datos Relacionales: Extensión del modelo y adaptación de SGDB actuales", Universidad de Granada, España, 1999.
- [12]. Gómez Salas Fernando, Yu Liu Wen, "Sistemas Difusos Jerárquico para el modelado y Control", Instituto Politécnico Nacional, 2005.
- [13]. Klir George J., Yuan Bo, "Fuzzy Sets and Fuzzy Logic", Prentice Hall, 1995.
- [14]. Kwang H. Lee "First Course on Fuzzy Theory and Applications", Springer-Verlag Berlin Heidelberg 2005
- [15]. J.M. Medina, O. Pons, M.A. Vila, "GEFRED. A Generalized Model of Fuzzy Relational Databases". *Information Sciences*, 76(1-2),87-109, 1994.
- [16]. J.M. Medina, "Bases de Datos Relacionales Difusas. Modelo Teórico y Aspectos de su Implementación". Ph. Doctoral Thesis , University of Granada (Spain), 1994 (www.decsai.ugr.es).

-
- [17]. H. Prade, C. Testemale, "Fuzzy Relational Databases: Representational issues and Reduction Using Similarity Measures". J.Am. Soc. Information Sciences 38(2), pp. 118-126, 1987.
- [18]. H. Prade, C. Testemale, "Generalizing Database Relational Algebra for the Treatment of Incomplete/Uncertain Information and Vague Queries". Informat. Sciences 34, pp. 115-143, 1984.
- [19]. Von Raesfeld P.A. "Aplicación de mapas cognitivos difusos a VRML en un ambiente virtual", UDLAP, Mayo 2000, Capítulo 2: Referencia: http://www.pue.udlap.mx/~tesis/lis/von_r_pa/capitulo2.pdf --- 2004 .
- [20]. B. Schweizer. A. Sklar. "Probabilistic Metric Spaces", North-Holland, 1983.
- [21]. Trillas E. "Sobre Funciones de Negación en la Teoría de Conjuntos Difusos", Stochastica. Vol. 3, n 1,47-59 (1979).
- [22]. V. Tahani, "A ConceptualFramework for Fuzzy Query Processing--A Step toward Very Intelligent Database Systems", 1977
- [23]. M. Umamo, "Freedom-O: A Fuzzy Database System". In "Fuzzy Information and Decision Processes". Eds. M. Gupta , E. Sanchez. North-Holand , Amsterdam, Pub. Comp., pp. 339-347, 1982.
- [24]. M. Umamo, S. Fukami, "Fuzzy Relational Algebra for Possibility-Distribution -Fuzzy-Relational Model of Fuzzy Data". Journal of Intelligent Information Systems, 3, pp. 7-28, 1994.
- [25]. Valdivia C. Carlos – García J. Pedro, *Propuesta de una Base Relacional Difusa para el Control de Asistencia en una Institución Educativa*, Facultad de Cs. de la Computación – BUAP, Noviembre 2004
- [26]. Xue-Gong Zhang, "Takagi-Sugeno Fuzzy Logic versus Mamdani Fuzzy Logic", University of The West of England, 2001.
- [27]. Lotfi A. Zadeh, "Fuzzy Sets", Departament of Electrical Engineering and Electronics Research Laboratory, University of California – Berkeley 1965
- [28]. Lotfi A. Zadeh, "Probability Theory And Fuzzy Logic", Diciembre 2002
- [29]. Lotfi A. Zadeh, "From Computing with Numbers to Computing with Words—From Manipulation of Measurements to Manipulation of Perceptions", Enero 1999
- [30]. Lotfi A. Zadeh, "From Search Engines to Question-Answering Systems, the need for new tools", 2003
- [31]. Lotfi A. Zadeh, "Fuzzy Logic = Computing with Words", Mayo 1996
- [32]. Lotfi A. Zadeh, "Probability Theory and Fuzzy Logic", Diciembre 2002
- [33]. Lotfi A. Zadeh, "Soft Computing and Fuzzy Logic" , Noviembre 1994
- [34]. Lotfi A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic", Febrero 1997
- [35]. Lotfi A. Zadeh, "The Role of Fuzzy Logic and Soft Computing in the Conception and Design of Intelligent Systems", 2001.
- [36]. M. Zemankova, A. Kandel, "Fuzzy Relational Databases - A Key to Expert Systems". Köln, Germany, TÜV Rheinland, 1984.
- [37]. M. Zemankova, A. Kandel, "Implementing Imprecision in Information Systems". Information Sciences, 37, pp. 107-141, 1985.