

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA
AUTENTICACIÓN, AUTORIZACIÓN, Y CONTEO (AAA) PARA EL
CONTROL DE ASISTENCIA DE LOS EMPLEADOS DE LA
ESCUELA POLITÉCNICA NACIONAL POR MEDIO DE HUELLAS
DACTILARES**

TOMO II

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y TELECOMUNICACIONES**

XAVIER EDUARDO ZURITA FRÍAS

DIRECTOR: ING. PABLO HIDALGO

Quito, Agosto 2004

ÍNDICE

TOMO I

ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS.....	X
RESUMEN.....	XI
PRESENTACIÓN.....	XII

CAPÍTULO 1

SISTEMAS DE AUTENTICACIÓN, AUTORIZACIÓN Y CONTEO BIOMÉTRICOS 1

1.1. GENERALIDADES DE SEGURIDAD.....	1
1.1.1. ¿Qué es seguridad?.....	1
1.1.1.1. La confidencialidad o Privacidad.....	1
1.1.1.2. La integridad.....	2
1.1.1.3. La disponibilidad.....	2
1.1.2. ¿Qué queremos proteger?.....	2
1.1.3. ¿De qué nos queremos proteger?.....	5
1.1.3.1. Personas.....	6
1.1.3.1.1. El personal de una empresa.....	8
1.1.3.1.2. Ex-empleados.....	9
1.1.3.1.3. <i>Crackers</i>	9
1.1.3.1.4. Curiosos.....	10
1.1.3.1.5. Terroristas.....	10
1.1.3.1.6. Intrusos remunerados.....	10
1.1.3.2. Amenazas lógicas.....	11
1.1.3.2.1. <i>Software</i> incorrecto.....	11
1.1.3.2.2. Herramientas de seguridad.....	12
1.1.3.2.3. Puertas traseras.....	12
1.1.3.2.4. Caballos de Troya.....	12
1.1.3.2.5. Bombas Lógicas.....	13
1.1.3.2.6. Canales cubiertos.....	14
1.1.3.2.7. Virus.....	14
1.1.3.2.8. Gusanos.....	15
1.1.3.2.9. Programas conejo o bacterias.....	16
1.1.3.3. Catástrofes.....	16
1.1.3.3.1. Incendios.....	16
1.1.3.3.2. Inundaciones.....	17
1.1.3.3.3. Accidentes Industriales.....	17
1.1.4. ¿Cómo nos podemos proteger?.....	17
1.1.5. Áreas de Administración de la Seguridad	19
1.1.5.1. Autenticación.....	19
1.1.5.2. Autorización.....	20
1.1.5.3. Auditoría.....	20

1.1.6. Seguridad Física y Lógica.....	21
1.1.6.1. Seguridad Física.....	21
1.1.6.2. Seguridad Lógica.....	22
1.1.7. Control de entrada al sistema.....	24
1.1.7.1. Mecanismos de autenticación.....	24
1.1.7.1.1. Claves informativas.....	25
1.1.7.1.2. Claves físicas.....	25
1.1.7.1.3. Claves biométricas.....	26
1.2. LA BIOMETRÍA APLICADA A LA SEGURIDAD.....	27
1.2.1. La Biometría Informática.....	28
1.2.2. Área de Acción.....	30
1.2.3. Una tecnología no aislada.....	32
1.2.4. La Autenticación Biométrica pilar de la seguridad.....	35
1.2.4.1. Funcionamiento de la Autenticación Biométrica.....	35
1.2.4.2. Identificación y Autenticación Biométrica.....	36
1.2.5. Sistema Biométrico.....	40
1.2.5.1. Arquitectura de un sistema biométrico para identificación personal.....	40
1.2.5.2. Características de un Sistema Biométrico.....	42
1.2.5.3. Partes del Sistema.....	44
1.2.6. Tipos de Identificadores biométricos.....	45
1.2.6.1. Características de un identificador biométrico.....	45
1.2.6.2. Terminología	46
1.2.6.3. Reconocimiento de Huellas Dactilares.....	47
1.2.6.3.1. Características del Autentificador	48
1.2.6.3.2. Sistema de Reconocimiento.....	52
1.2.6.3.3. Características del Sistema.....	52
1.2.6.4. Reconocimiento Facial.....	54
1.2.6.4.1. Etapas de detección.....	56
1.2.6.4.2. Características de una nueva tecnología.....	57
1.2.6.4.3. Características del Autentificador.....	58
1.2.6.4.4. Sistema de Reconocimiento y Métodos utilizados.....	58
a. Métodos basados en detección de características.....	59
b. Métodos basados en la imagen.....	60
b.1. Método de los <i>Eigenfaces</i>	60
b.2. Redes neuronales.....	61
1.2.6.4.5. Características del Sistema.....	62
1.2.6.5. Geometría de la mano.....	62
1.2.6.5.1. Características Biométricas de la mano y Métodos Utilizados.....	65

a. Proceso de extracción de características invariantes de la mano.....	67
a.1. Características invariantes basadas en la Geometría de la mano.....	68
a.2. Características invariantes basadas en la Disposición Natural de la mano.....	70
1.2.6.6. Reconocimiento de Iris y de la Retina.....	75
1.2.6.6.1. Reconocimiento de Iris.....	75
a. Características del Autenticador.....	76
b. Sistema de Reconocimiento.....	77
c. Características del Sistema.....	86
1.2.6.6.2. Reconocimiento de la Retina.....	87
a. Características del Sistema.....	88
1.2.6.7. Reconocimiento de la Voz.....	89
1.2.6.7.1. Características del Sistema.....	92
1.2.6.8. Reconocimiento de Firma.....	93
1.2.6.8.1. Características del Sistema.....	94
1.2.6.9. Otros Sistemas.....	95
1.2.7. Criterios para elegir una tecnología biométrica.....	95
1.2.8. Ventajas y desventajas de los Sistemas Analizados.....	96
1.2.9. La configuración por defecto.....	97

CAPÍTULO 2

<u>LA HUELLA DACTILAR, ESTUDIO Y ANÁLISIS COMO UN IDENTIFICADOR BIOMÉTRICO</u>	99
2.1. RAZONES E HISTORIA	99
2.1.1. ¿Porqué Identificar la Huella Dactilar?.....	99
2.1.2. Un poco de Historia.....	100
2.2. ERRORES Y EVALUACIÓN DE UN SISTEMA BIOMÉTRICO BASADO EN LA HUELLA DACTILAR	106
2.2.1. Terminología.....	106
2.2.1.1. <i>Template</i>	106
2.2.1.2. <i>Biometric Matching</i>	107
2.2.2. Evaluación del Sistema Biométrico.....	108
2.2.2.1. Errores en la verificación.....	109
2.2.2.2. Errores en la Identificación.....	114
2.3. PARTES DEL SISTEMA	115

2.3.1. <i>Hardware</i> para Adquisición de la Imagen.....	116
2.3.1.1. Tecnologías para la Adquisición.....	119
2.3.1.1.1. Tecnología Óptica.....	119
2.3.1.1.2. Tecnología del Silicio o Capacitivo.....	120
2.3.1.1.3. Tecnología de Ultrasonido.....	121
2.3.1.2. Tipos de Sensores.....	123
2.3.1.2.1. Sensor de Matriz Capacitivo.....	123
2.3.1.2.2. Sensor de Matriz de Antena.....	124
2.3.2. <i>Software</i>	128
2.3.2.1. Componentes de proceso de la imagen.....	129
2.3.2.1.1. Adquisición de la imagen.....	131
2.3.2.1.2. Mejoramiento de la Imagen.....	132
a. Normalización de la imagen.....	133
b. Cálculo del campo orientación y mapa del período.....	134
c. Región de la máscara.....	136
d. Filtrado.....	137
e. Simplificación o Adelgazamiento.....	137
f. Eliminación de imperfecciones.....	138
2.3.2.2. Componentes para la generación de la plantilla.....	138
2.3.2.2.1. Detección de los puntos singulares de una huella.....	138
2.3.2.2.2. Extracción de minucias.....	140
2.3.2.3. Emparejamiento o “ <i>matching</i> ”.....	141
2.3.2.3.1. Algoritmo de comparación de minucias.....	141
a. Formulación del problema.....	142
2.3.2.3.2. Emparejamiento de plantillas basadas en patrones.....	149
2.3.2.3.3. Métodos de la Extracción de las Característica: Basadas en Minucias vs. Basadas en Patrones - Comparación Técnica.....	150
a. Tamaño de la plantilla vs. velocidad de búsqueda y comparación.....	150
b. Sensibilidad a los cambios físicos.....	151
c. Seguridad y reproducción (<i>playback</i>).....	151
2.3.2.4. Almacenamiento de la huella.....	152
2.4. VENTAJAS Y DESVENTAJAS.....	154
2.4.1. Ventajas.....	154
2.4.1.1. Tecnología probada para altos niveles de exactitud.....	154
2.4.1.2. Gama de ambientes.....	154
2.4.1.3. Ergonómico, Facilidad de uso.....	155
2.4.1.4. Capacidad de almacenar múltiples huellas de una sola persona.....	155

2.4.2. Desventajas	156
2.4.2.1. El funcionamiento puede deteriorarse en un cierto plazo.....	156
2.4.2.2. Asociación con usos forenses y criminalísticos.....	156
2.4.2.3. Necesidad de despliegue.....	157
CAPÍTULO 3	
<u>DISEÑO, IMPLEMENTACIÓN Y EVALUACIÓN DEL PROTOTIPO</u>	158
3.1. REQUERIMIENTOS DEL SISTEMA DE CONTROL DE ASISTENCIA (SCA).....	158
3.1.1. El Control de Asistencia es una Política Laboral.....	158
3.1.2. Sistema de Control de Asistencia actual de la EPN.....	159
3.1.3. Requerimientos del Sistema.....	160
3.2. DISEÑO DEL SISTEMA.....	162
3.2.1. <i>Hardware</i> del SCA.....	162
3.2.1.1. Equipo de Red.....	162
3.2.1.2. Lector de Huellas.....	163
3.2.2. <i>Software</i> del SCA.....	164
3.2.2.1. Herramientas para el desarrollo de la aplicación.....	164
3.2.2.1.1. Sistemas Operativos.....	164
3.2.2.1.2. <i>Biologon 3.0</i>	165
a. Edición Cliente/Servidor.....	166
b. <i>BioShield</i>	166
3.2.2.1.3. <i>Visual 6.0</i>	167
3.2.2.1.4. SQL (<i>Structured Query Language</i>).....	167
3.2.2.2. Software del Servidor.....	168
3.2.2.2.1. Instalación de <i>Biologon 3.0</i> Cliente/Servidor.....	169
3.2.2.2.2. Código del Empleado y Cuentas de <i>Biologon 3.0</i>	170
3.2.2.2.3. Programación y Base de Datos.....	170
a. Tablas de la Base de Datos del SCA.....	170
b. Programación en <i>Visual 6.0</i>	173
b.1. Menú Archivo.....	175
b.2. Menú Administración.....	175
b.3. Menú Registro.....	184
b.4. Menú Consultas.....	195
b.5. Menú Reportes.....	195
3.2.2.3. <i>Software</i> del Terminal.....	196
3.3. EVALUACIÓN DEL PROTOTIPO.....	203

CAPÍTULO 4

<u>CONCLUSIONES Y RECOMENDACIONES</u>	210
4.1. CONCLUSIONES.....	210
4.2. RECOMENDACIONES.....	215
BIBLIOGRAFÍA.....	217

TOMO II

ANEXO A	Reconocimiento Facial - Métodos basados en detección de características
ANEXO B	Detección de los puntos singulares de una huella - Método de Poincaré
ANEXO C	Lector de huella dactilar <i>BioTouch</i> USB 200 de Identix - Características Técnicas
ANEXO D	Código fuente del <i>software</i> del Sistema de Control de Asistencia
ANEXO E	Manual de Instalación y Operación del Sistema de Control de Asistencia
ANEXO F	Presupuesto Referencial

ÍNDICE DE FIGURAS

CAPÍTULO 1

1. 1	Tipos de ataques a un sistema informático.....	5
1. 2	División básica de la Biometría.....	29
1. 3	Procedimiento para la obtención de un patrón biométrico basado en la huella dactilar.....	39
1. 4	Arquitectura de un sistema biométrico para identificación personal, aquí ejemplificado con huellas dactilares.....	41
1. 5	Relación entre FAR, FRR y ERR.....	47
1. 6	Huella dactilar.....	47
1. 7	Huella dactilar procesada.....	50
1. 8	Trazado del patrón de detalles.....	50
1. 9	Diagrama de bloques de un sistema reconocimiento de huellas dactilares.....	52
1. 10	Reconocimiento facial.....	55
1. 11	Extracción de parámetros geométricos de la cara.....	55
1. 12	Sombra geométrica de la mano.....	62
1. 13	Colocación de la mano en el dispositivo lector.....	64
1. 14	Ejemplo de obtención de medidas en los sistemas expuestos y posible variación de éstas al no respetar la colocación de la mano en los pivotes.....	65
1. 15	Prototipo para obtención de las características basado en la disposición natural de las manos.....	66
1. 16	(a) Imagen original en posición libre, (b) la imagen después de realizar la alineación junto con el sistema de referencia, (c) situación del sistema de referencia.....	67
1. 17	(a) Mano centrada en el nuevo sistema y posiciones de los valles y las crestas, (b) Medidas calculadas a partir de los puntos de los valles y las crestas.....	68
1. 18	(a) Proceso de segmentación de un dedo, (b) Salida del proceso de segmentación de la mano.....	69
1. 19	(a) Proceso de detección de las falanges; (b) Resto de medidas obtenidas.....	71
1. 20	Invarianza de la disposición de los dedos.....	71
1. 21	Sistemas de coordenadas propios y superposición de contornos con un sistema de origen común.....	72
1. 22	(a) Plantilla I_N , (b) Imagen de borde extendido E, (c) Función de módulo, (d) Función de fase del contorno.....	73
1. 23	Equipos lectores de la geométrica de la mano serie <i>Handke</i>	74
1. 24	Partes del ojo humano.....	75
1. 25	El iris del ojo humano.....	76
1. 26	Características visibles en un iris.....	77
1. 27	Preprocesado sobre la imagen adquirida.....	78
1. 28	Fronteras límbica y pupilar.....	79
1. 29	Imagen binaria, resultado de la aplicación del umbral.....	80
1. 30	Pasos para obtener la imagen binaria en iris miel, café o verde.....	81
1. 31	(a) Imagen binaria después de aplicarle el filtro de mediana, (b) pupila ubicada.....	81
1. 32	Pupila rellena.....	82
1. 33	Localización del centro de la pupila.....	82
1. 34	Modelo elíptico.....	83
1. 35	Radios límbico (rl) y pupilar (rp).....	83
1. 36	Área de análisis.....	84
1. 37	Coordenadas polares.....	84
1. 38	(a) Cinta de análisis, (b) Cinta contrastada.....	85
1. 39	Cinta interpolada.....	85
1. 40	Retina Humana.....	88
1. 41	Lector de retina de EyeDentify.....	89
1. 42	Imagen tridimensional del espectro de la voz.....	90
1. 43	Espectro de voz.....	90
1. 44	Firma.....	93

CAPÍTULO 2

2. 1	Plantillas biométricas versus datos biométricos identificables.....	107
2. 2	FMR y FNMR para un umbral dado t se grafican sobre las distribuciones genuinas y del impostor de la distribución del <i>score</i>	111
2. 3	Evaluación del sistemas FVC2002 para verificación de la huella digital (Maio et al., 2002b) utilizando la base de datos DB1: (a) Las distribuciones genuinas y del impostor a partir de 2800 pares genuinos y de 4950 pares del impostor, respectivamente; (b) FMR(t) y FNMR(t) se derivan de las distribuciones del <i>score</i> en a); (c) La curva de ROC se deriva de las curvas de FMR(t) y de FNMR(t) en b).....	112
2. 4	Ejemplo de las curvas de FMR(t) y de FNMR(t), donde se destacan los puntos que corresponden a EER, ZeroFNMR y a ZeroFMR.....	113
2. 5	Los puntos de funcionamiento típicos de diversos usos exhibidos en una curva ROC.....	113
2. 6	Dispositivos <i>finger-scan</i>	116
2. 7	Partes del sistema <i>finger-scan</i>	117
2. 8	Posibles escenarios esquemáticos de proceso y almacenamiento de los sistemas <i>finger-scan</i>	118
2. 9	Sensor capacitivo clásico.....	124
2. 10	Sensor de Matriz de Antena.....	125
2. 11	Scanner a disposición comercial.....	125
2. 12	(a) Biometrika FX2000, (b) Digital Persona UareU2000, (c) Identix DFR200, (d) Ethentica TactilSense T-FPM. (e) STMicroelectronics TouchChip TCS1AD, (f) Veridicom FPS110, (g) Atmel FingerChip AT77C101B, (h) Authentec AES4000.	128
2. 13	Diagramas de bloque de inscripción, verificación, e identificación.....	129
2. 14	(a) Región bien definida; (b) Región dañada recuperable (c); Región dañada no-recuperable.	131
2. 15	Típicas imágenes adquiridas por los sistemas <i>finger-scan</i>	131
2. 16	Diagrama general para el mejoramiento de la imagen de una huella.....	133
2. 17	(a) Campo de direcciones superpuesto a la huella; (b) Campo de direcciones de una huella.....	135
2. 18	(a) Variaciones de la huella (campo de variaciones) (b) Región importante- Blanco=huella, Negro=Ruido.....	136
2. 19	(a) Imagen binarizada; (b) Imagen adelgazada.....	138
2. 20	(a) Arco, (b) Lazo, (c) Espiral.....	139
2. 21	Puntos singulares en una huella dactilar.....	139
2. 22	(a) Primera extracción de minucias, (b) y (c) Proceso de eliminación de minucias falsas, (d) Patrón de minucias.....	140
2. 23	Ejc y minucias extraídas.....	141
2. 24	(a) Una minucia se caracteriza por la posición y la orientación; (b) Varios detalles señalados en una huella.....	143
2. 25	Las minucias de I mapeadas en las coordenadas de T para una alineación dada. Las minucias de T son denotadas por (o), mientras que las minucias de I son denotadas por (x).	147
2. 26	En este ejemplo, si m_1 fuera acoplado con m_2 (las minucias más cercanas), m_2 sería el único acoplamiento; sin embargo, apareando m_1 con m_1 , permite que m_2 sea acoplado con m_2 , así se maximiza la ecuación (2.10).....	148
2. 27	Dificultad en emparejar la huella digital. Las imágenes de la huella digital en (a) y (b) son diferentes a un ojo inexperto pero ellas son impresiones del mismo dedo. Las imágenes de la huella digital en (c) y (d) son iguales miradas con un ojo inexperto pero ellas son de diversos dedos.....	149
2. 28	Plantilla basada en patrones.....	150
2. 29	Imagen de una huella dactilar de 768x768 pixels.....	153

CAPÍTULO 3

3. 1	Reloj de fichar electromecánico.....	159
3. 2	Diagrama de bloques del SCA.....	161

3. 3	Esquema de red para el Sistema de Control de Asistencia.....	163
3. 4	Lector de huellas dactilares DFR-200 de Identix.....	164
3. 5	Diagrama de interacción de las tablas de la Base de Datos del SCA.....	171
3. 6	Pantalla de ingreso a la aplicación del Servidor.....	174
3. 7	Interfaz principal de la aplicación del Servidor.....	174
3. 8	Grilla programada para ordenar y desplegar los nombres de los empleados de la EPN.....	176
3. 9	Ventana para ingreso de los Datos Personales del empleado.....	177
3. 10	Grilla programada para ordenar y desplegar los horarios de los empleados de la EPN.....	178
3. 11	Ventana de definición de horarios, previa al ingreso de las horas de entrada y salida.....	178
3. 12	Definición de horarios.....	180
3. 13	Ingreso de las horas de entrada y salida.....	180
3. 14	Ventana para la asignación de horarios.....	181
3. 15	Ventana de selección del rango de fechas de atraso para una posible justificación.....	182
3. 16	Ventana de justificaciones.....	182
3. 17	Ventanas de justificación, (a) Para jornada única o diferenciada, (b) Para jornada diferenciada doble.....	183
3. 18	Pantalla de ingreso de vacaciones.....	184
3. 19	Proceso de Actualización del Registro de Asistencia.....	185
3. 20	Proceso de Toma de Decisiones para jornada diferenciada doble, opción: cuatros registros del empleado.....	190
3. 21	Proceso de Toma de Decisiones para jornada diferenciada doble, opción: tres registros del empleado.....	191
3. 22	Proceso de Toma de Decisiones para jornada diferenciada doble, opción: dos registros del empleado.....	192
3. 23	Proceso de Toma de Decisiones para jornada única o diferenciada.....	192
3. 24	Proceso de la subrutina "Evaluación General". Parte 1 de 2.....	193
3. 25	Proceso de la subrutina "Evaluación General". Parte 2 de 2.....	194
3. 26	Proceso de inscripción de un registro en el Terminal.....	198
3. 27	Interfaz de inscripción para los registros de asistencia.....	199
3. 28	Pantalla de captura del código del empleado.....	200
3. 29	Protección de la pantalla de captura del código del empleado utilizando el Banco de Contraseñas de Bioshield.....	201
3. 30	Pantallas de propiedades de <i>Bioshield</i>	202
3. 31	Pantalla de captura del código del empleado con las cajas de texto ocultas.....	202
3. 32	Lista de empleados para la evaluación del SCA.....	203
3. 33	Horario asignado al empleado FABIAN BAUTISTA.....	204
3. 34	Horario asignado al empleado LEONARDO CHUQUI.....	204
3. 35	Horario asignado al empleado FERNANDO IBARRA.....	205
3. 36	Horario asignado a la empleada ADRIANA MOLINA.....	205
3. 37	Horario asignado al empleado XAVIER ZURITA.....	206
3. 38	Asistencia del empleado FABIAN BAUTISTA.....	206
3. 39	Asistencia del empleado LEONARDO CHUQUI.....	207
3. 40	Asistencia del empleado FERNANDO IBARRA.....	207
3. 41	Asistencia de la empleada ADRIANA MOLINA.....	207
3. 42	Asistencia del empleado XAVIER ZURITA.....	208
3. 43	Reporte de minutos atrasados.....	208

ÍNDICE DE TABLAS

1.1	Productos biométricos basados en la Huella Dactilar.....	53
1.2	Productos biométricos basados en la Geometría de la mano.....	74
1.3	Productos biométricos basados en el iris.....	87
1.4	Producto biométrico basado en la Retina-EyeDentify.....	89
1.5	Producto biométrico basado en el reconocimiento de la voz.....	93
1.6	Rechazos y Aceptaciones equivocadas.....	98
2.1	Algunos <i>scanners</i> comerciales, agrupados por la tecnología, dentro de cada tecnología, se enumeran a las compañías en orden alfabético. La tabla presenta la resolución, el área de detección, y el número de <i>pixel</i>	127

ANEXO A

RECONOCIMIENTO FACIAL
basados en detección de características

RECONOCIMIENTO FACIAL

MÉTODOS BASADOS EN DETECCIÓN DE CARACTERÍSTICAS ^[46]

Análisis a Bajo Nivel

Son técnicas que trabajan a nivel de píxel. Hay diversas técnicas dentro de este apartado, las más relevantes son:

Detección de bordes.- Fue una de las primeras técnicas utilizadas. La idea se basa en analizar las líneas que componen los bordes de una cara y utilizarlas para detectar los rasgos faciales. Para llevar a cabo su propósito, el algoritmo sigue los siguientes pasos:

- Detectar los bordes de la imagen, utilizando para ello alguno de los operadores existentes, como el de *Marr-Hildreth*.
- Una vez obtenidos los bordes, se procede a efectuar un adelgazamiento a fin de obtener para cada borde una línea de un píxel de ancho que lo represente.
- Filtrado de componentes: El algoritmo se queda ahora sólo con las componentes que sean más susceptibles de formar parte de una cara. Por ejemplo, buscando líneas que en conjunto se asemejen a una elipse de determinadas proporciones de ancho y alto.
- Etiquetado. Una vez obtenidas dichas componentes, se etiquetan como lado derecho de la cara, lado izquierdo, etc.

Esta técnica detecta en forma correcta aproximadamente tres de cada cuatro caras, con un dos por ciento de falsos positivos. Este algoritmo puede trabajar (aunque con más dificultad) incluso si el fondo de la imagen no es uniforme. Si la cara no está de frente, es totalmente ineficiente.

Información de grises.- Trabaja sobre la idea de que rasgos faciales tales como las cejas, las pupilas y los labios aparecen como zonas más oscuras de la imagen

que las zonas que corresponden a las regiones faciales que los rodean. Este algoritmo se compone de las siguientes partes:

- Aumentar el contraste de la imagen. De esta forma se resalta aún más la diferencia de luminosidad entre las citadas partes de la cara.
- *Thresholding*: El algoritmo se queda sólo con las zonas de la imagen cuyo valor de gris supere un cierto umbral.
- Detección de caras mediante el uso de plantillas ponderadas. Hasta aquí se ha obtenido una imagen compuesta por multitud de manchas negras. Este paso trata de comparar la distribución de esas manchas con las manchas “tipo” de una cara, usando plantillas. Hay varias propuestas de plantillas, algunas se basan en detectar primero las zonas de los ojos y a partir de ahí intentar detectar el resto de componentes. Otros métodos tratan de buscar máximos locales, como la punta de la nariz.

Este algoritmo produce diferentes resultados en función del color de la piel del sujeto que aparece en la imagen. Para corregir este problema, existe una técnica similar aunque más potente que usa la información del color, dado que dos siluetas con la misma información de gris pueden ser completamente diferentes si se tiene en cuenta la información del color.

Análisis de rasgos

El análisis a bajo nivel puede dar información ambigua, por ejemplo, si aparecen en la imagen objetos que tengan un color similar al del modelo de color de piel utilizado. Los métodos que vienen a continuación se basan en la geometría de la cara para caracterizar y posteriormente verificar rasgos a fin de evitar dicha ambigüedad.

Búsqueda de rasgos.- Hay varias formas de llevar a cabo este algoritmo, siendo una de ellas la siguiente:

- Búsqueda de la parte superior de la cabeza. Se efectúa una hipótesis sobre lo que puede ser una posible línea del pelo en lo alto de la frente. Puede ser difícil si la persona tiene pelo cubriendo zonas de la frente.
- Búsqueda de los ojos. A partir de dicha línea se efectúa un barrido hacia abajo tratando de buscar zonas donde la densidad de gris aumente y disminuya bruscamente en el plano horizontal. Dichas zonas corresponden con las pupilas. Falla si el individuo usa gafas, si uno de los ojos no aparece por cualquier motivo, o evidentemente si la imagen está rotada.
- Uso de plantillas flexibles. La distancia entre la línea del pelo y el plano de los ojos se usa como medida de referencia para inicializar una plantilla flexible que cubre el resto de rasgos, como la nariz y la boca. La plantilla trata entonces de ajustarse a dichos rasgos.

Este método es capaz de detectar de forma correcta aproximadamente cuatro de cada cinco caras.

Análisis de constelaciones.- Los métodos vistos hasta ahora son muy rígidos, y sólo funcionan bajo condiciones muy específicas. Para subsanar este problema se ideó el método del análisis de constelaciones, que se basa en el uso de un modelo probabilístico que estudia la posición espacial de los rasgos faciales, intentando buscar patrones que se asemejen a una cara. Este método tiene un gran contenido matemático, y por eso no se lo detallará, aunque se debe decir que falla si se produce una rotación significativa de la cara del sujeto. Existen variantes de este algoritmo que llegan hasta el noventa por ciento de aciertos.

Modelos de silueta activa

Los modelos de silueta activa trabajan usando la información de los bordes y el brillo de la imagen, interactuando con los rasgos locales de la imagen de forma que poco a poco se van deformando y tomando la forma del rasgo.

Índice del método de Poincaré ^[2]

Un método práctico y elegante para la detección de los puntos *core* y delta es el basado en el índice de Poincaré, propuesto por Kawagoe y Tojo (1984). Sea G un campo vectorial y C una curva inmersa en G ; entonces se define $P_{G,C}$ como el índice de Poincaré y es definida como la rotación total de los vectores de G a lo largo de C , figura B.1.

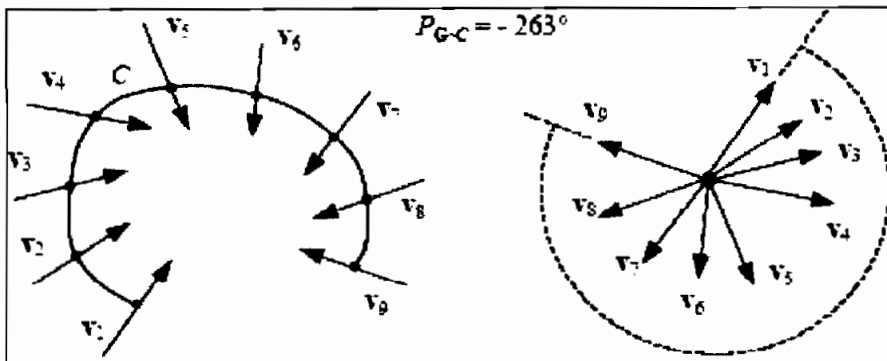


Figura B.1: Índice de Poincaré determinado sobre una curva C inmersa en un campo vectorial G . [1]

Entonces si G es el campo asociado a una imagen de orientaciones D de la huella digital y $[i, j]$ es la posición del elemento θ_{ij} en la imagen de orientaciones, entonces el índice de Poincaré $P_{G,C}(i, j)$ en $[i, j]$ se calcula de la siguiente manera:

- La curva C es una trayectoria cerrada definida como secuencia de algunos elementos de D , tal que $[i, j]$ es un punto interno;
- $P_{G,C}(i, j)$ es calculado algebraicamente sumando las diferencias de la orientación entre los elementos adyacentes de C . Sumar diferencias de orientación requiere que una dirección (entre los dos posibles) debe ser asociada en cada orientación. Una solución a este problema es seleccionar aleatoriamente la dirección del primer elemento y asignar la dirección más cercana a la del elemento anterior a cada elemento sucesivo. Es bien conocido y puede ser demostrado fácilmente que, en curvas cerradas, el

índice de Poincaré asume solamente uno de los valores discretos: 0° , $\pm 180^\circ$, y $\pm 360^\circ$. En el caso de singularidades de la huella digital:

$$P_{G,C}(i,j) = \begin{cases} 0^\circ & \text{Si } [i,j] \text{ no pertenece a ninguna región de singularidad} \\ 360^\circ & \text{Si } [i,j] \text{ pertenece a una región de singularidad tipo espiral} \\ 180^\circ & \text{Si } [i,j] \text{ pertenece a una región de singularidad tipo lazo} \\ -180^\circ & \text{Si } [i,j] \text{ pertenece a una región de singularidad tipo delta} \end{cases}$$

La figura B.2 muestra tres porciones de imágenes de orientación de la huella digital. La trayectoria C sería definida por la secuencia de los elementos d_k ($k = 0 \dots 7$) que rodea $[i, j]$. La dirección de los elementos d_k se elige como sigue: d_0 con dirección hacia arriba; d_k ($k = 1 \dots 7$) con direcciones de modo que el valor absoluto del ángulo entre d_k y d_{k-1} sea menor que o igual a 90° . El índice de Poincaré entonces se calcula:

$$P_{G,C}(i,j) = \sum_{k=0 \dots 7} \text{angle}(d_k, d_{(k+1) \bmod 8}) \quad (1)$$

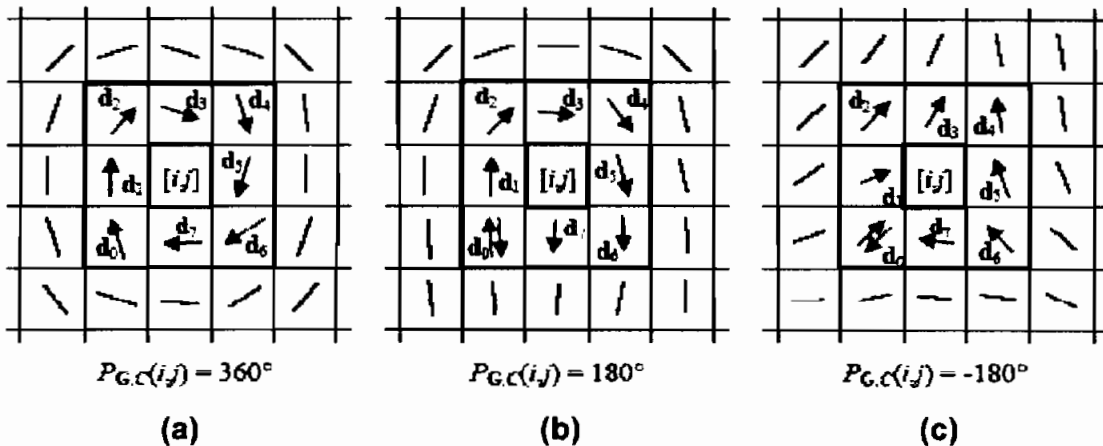


Figura B.2: En el ejemplo del cálculo del índice de Poincaré en los ocho elementos vecinos de los puntos que pertenecen a un arco, a un lazo, y a una singularidad delta, respectivamente.

ANEXO C

Identificador Dactilar *BioTouch* USB 200 de Identix
Características Técnicas

BioTouch™ USB 200

Lector de Huella Digital

El lector de huella digital BioTouch USB de Identix ofrece seguridad tanto en el desktop como en la red, mientras mantiene la confidencialidad y la privacidad del usuario. El Bio Touch de Identix es un periférico autónomo de fácil instalación que consistentemente produce una excelente calidad de imagen, crítica para las aplicaciones biométricas.

Cuando se trata de una prueba de identidad, no es fácil vencer a la huella digital. Las huellas digitales son únicas, fáciles de llevar y, a diferencia de las llaves, carnets y contraseñas, no se pueden olvidar, perder o robar y cuando se trata de leer huellas digitales, no es fácil vencer a los sistemas ópticos. Estos son rápidos, precisos y de bajo costo, además de carecer de la susceptibilidad de los sensores capacitivos a las descargas electrostáticas. El BioTouch de Identix es un sistema óptico que protege las computadoras personales y redes contra los accesos no autorizados. Con el simple toque del dedo, el BioTouch lee los atributos físicos de la huella para autenticar la identidad de la persona que solicita el acceso.

Tecnología: Una avanzada tecnología de cámaras CMOS de Motorola® mantiene la precisión aún cuando el dedo pueda estar mojado, seco o cortado.

Control de Calidad: El control en el BioTouch recibe de vuelta una interesante cantidad de señales de estatus en tiempo real como alineación vertical y horizontal, líneas o surcos interrumpidos y presión.

Seguridad y Privacidad del Usuario: Captura y almacena unos cuantos puntos de la huella y forma una plantilla de puntos conocidos como 'minutiae'. El resto de la imagen de la huella digital se descarta reduciendo el riesgo de que alguien utilice datos almacenados para replicar una huella digital.

Características y Beneficios

- La lectura de la imagen completa de la huella digital elimina problemas de colocación del dedo.
- La avanzada tecnología de cámara CMOS de Motorola proporciona una imagen de alta calidad y confiabilidad.
- El sistema integrado de control de calidad garantiza la captura de una buena imagen de la huella digital
- El interfaz USB soporta instalación plug-and-play
- El sensor óptico asegura la durabilidad e inmunidad a las descargas electrostáticas.

Especificaciones

Dimensiones: longitud 96 mm, altura 31 mm, ancho 60 mm, peso 150 gramos, negro

Sensor Biométrico: sensor óptico DFR-200, tamaño de la superficie de captura: 17 x 17 mm, resolución de cámara CMOS, resolución horizontal, resolución vertical, rotación del dedo de 360°

Interfaz: USB, cable negro de 6 pies; se conecta directamente a la PC, laptop o a un hub con energía autónoma; data rate hasta de 12 Mb/s.

Software: software BioLogon de Identix incluido.

Sistemas Operativos: Microsoft Windows® 95/98/NT/2000/ME/XP

Fuente de Poder: voltaje de entrada DC %V +/- 10% provisto vía puerto USB, modo de ahorro de 8 mA, consumo de energía de 340 mW con el LED encendido y 120 mW en stand-by

Otros: temperatura de operación entre 0 y 55 grados; LED color rojo; agencias que aprueban: FCC, CE, VCCI, Ctick.

identix

authenticating the wired and wireless world

ANEXO D

Código fuente del *software* del
Sistema de Control de Asistencia

Código fuente del SCA

SOFTWARE DEL SERVIDOR

Código cargado en la forma fmIngreso

Esta forma hace uso de los siguientes módulos: Módulo1, clsEmpleados, clsEmpleadosDat, clsUsuarios y clsUsuariosDat.

'Declaración de variables

```
Dim rs As New ADODB.Recordset
Dim con As New ADODB.Connection

Dim objEmpleadoDat As New clsEmpleadosDat
Dim objUsuarios As New clsUsuarios
```

'Subrutina del botón "Ingresar"; chequea si todos los campos, Login, Password y Servidor estén llenos.

```
Private Sub btnIngresar_Click()

    If txtLogin.Text <> "" And txtPassword.Text <> "" And txtServidor.Text <> "" Then
        If objEmpleadoDat.conectarBDD(txtServidor.Text) = True Then
            verificaUsuario
        End If
    Else
        MsgBox "Incluya la información en todos los campos.", vbInformation, Me.Caption
    End If

End Sub
```

'Subrutina "verificaUsuario", consulta la existencia en la Base de Datos del login y el password del usuario, caso contrario visualiza el mensaje "Usuario no definido"

```
Private Sub verificaUsuario()

    Dim strLog As String
    Dim strPas As String
    Dim strPer As String

    Set rs = objUsuarios.consultaUsuario(txtLogin.Text, txtPassword.Text)

    While Not rs.EOF
        strLog = rs!strLogin
```

```

        strPas = rs!strPassword
        strPer = rs!strPerfil
        rs.MoveNext

    Wend

rs.Close

If    strLog <> "" And strPas <> "" And strPer <> "" Then

    asignarPermisos (strPer)
    frmPrincipal.Show
    Unload Me

Else

    MsgBox "Usuario no definido.", vbCritical, Me.Caption
    Unload Me

End If

End Sub

```

'Subrutina "asignarPermisos" la cual chequea el perfil del usuario, ya sea administrador o usuario de consulta.

```

Private Sub asignarPermisos(strPerfil As String)

    If    strPerfil = "USUARIO" Then

        'Si es el perfil de USUARIO se deshabilita las opciones de Administración

        frmPrincipal.mnuAdministracion.Visible = False
        frmPrincipal.mnuConfiguracion.Visible = False
        frmPrincipal.brMenu.Visible = False

    End If

End Sub

```

'Subrutina que llama a la función "Alfabetico" la cual permite ingresar letras mayúsculas, si se ingresa minúsculas automáticamente las pasa a mayúsculas.

```

Private Sub txtServidor_KeyPress(KeyAscii As Integer)

    KeyAscii = Alfabetico(KeyAscii)

End Sub

```

Código cargado en la forma frmPrincipal

Esta forma hace uso de los siguientes módulos: Módulo1, clsEmpleados, clsEmpleadosDat, clsHorario y clsHorarioDat.

Subrutinas del menú principal, todas ellas responden al momento de pulsar cualquier menú o submenú especificado en la pantalla.

'Declaración de variables

```
Dim rs As New ADODB.Reco  
rdset
```

Menú Archivo

'Llama al submenú "Salir" del menú Archivo.

```
Private Sub mnuSalir_Click()  
  
    End  
  
End Sub
```

Menú Administración

'Llama al submenú "Empleados" del menú "Administración".

```
Private Sub mnuEmpleados_Click()  
  
    strMenu = "Empleados"  
    frmAdmEmpleados.Show  
  
End Sub
```

'Llamada al submenú "Administración" de Horarios perteneciente al submenú "Horarios".

```
Private Sub mnuAdmHorarios_Click()  
  
    strMenu = "Horarios"  
    frmAdmHorarios.Show  
  
End Sub
```

'Llamada al submenú "Asignación" perteneciente al submenú "Horarios".

```
Private Sub mnuAsignar_Click()  
  
    strMenu = "Asignar Horario"  
    frmAdmEmpleados.Caption = "Asignación de Horarios"  
    frmAdmEmpleados.Show  
  
End Sub
```

'Llama al submenú "Justificaciones" del menú "Administración".

```
Private Sub mnuJustificaciones_Click()  
  
    strMenu = "Justificacion"  
    frmAdmEmpleados.Caption = "Justificación"  
    frmAdmEmpleados.Show  
  
End Sub
```

'Llama al submenú "Vacaciones" del menú "Administración".

```
Private Sub mnuVacaciones_Click()  
  
    strMenu = "Vacaciones"  
    frmAdmEmpleados.Caption = "Vacaciones"  
    frmAdmEmpleados.Show  
  
End Sub
```

Menú Registro

'Llama al menú "Registro".

```
Private Sub mnuRegistro_Click()  
  
    frmActRegistro.Show  
  
End Sub
```

Menú Consultas

'Llamada al submenú "Datos Personales" del submenú "Empleados" del menú "Consultas".

```
Private Sub mnuConDatosPer_Click()  
  
    strMenu = "ConDatosEmp"  
    frmAdmEmpleados.Show  
  
End Sub
```

'Llamada al submenú "Horario" del submenú "Empleados" del menú "Consultas".

```
Private Sub mnuConHorarioAsig_Click()  
  
    strMenu = "ConHorarioAsig"  
    frmAdmEmpleados.Show  
  
End Sub
```


'Llama al submenú "Vacaciones" del submenú "Empleados" del menú "Consultas".

```
Private Sub mnuConVacaciones_Click()
```

```
    strMenu = "ConVacaciones"  
    frmAdmEmpleados.Show
```

```
End Sub
```

'Llamada al submenú "Horario" del submenú "Consultas".

```
Private Sub mnuConHorarios_Click()
```

```
    strMenu = "ConHorarios"  
    frmAdmHorarios.Show
```

```
End Sub
```

Menú Reportes

'Llama al reporte "Empleados" del menú "Reportes".

```
Private Sub mnuRepEmpl_Click()
```

```
    abreReporte "E:\Reporte\ListadoEmpleados.xls"
```

```
End Sub
```

'Llama al reporte "Empleados-Horario" del menú "Reportes".

```
Private Sub mnuEmpHor_Click()
```

```
    abreReporte "E:\Reporte\ListadoEmpHorario.xls"
```

```
End Sub
```

'Llama al reporte "Vacaciones" del menú "Reportes".

```
Private Sub mnuEmpVac_Click()
```

```
    abreReporte "E:\Reporte\ListadoEmpVacaciones.xls"
```

```
End Sub
```

'Llama al reporte "Horario" del menú "Reportes".

```
Private Sub mnuHorarios_Click()
```

```
    abreReporte "E:\Reporte\ListadoHorarios.xls"
```

```
End Sub
```

'Dependiendo del reporte escogido, abre el reporte en Excel.

```
Private Sub abreReporte(strReporte As String)

    Excel.Application.Visible = True
    Excel.Application.Workbooks.Open (strReporte)

End Sub
```

Código cargado en la forma frmAdmEmpleados

Esta forma hace uso de los siguientes módulos: Módulo1, clsEmpleados, clsEmpleadosDat, clsHorario y clsHorarioDat.

Subrutinas de los botones de la ventana principal de la grilla de empleados:

```
Dim rs As New ADODB.Recordset
Dim objEmpleados As New clsEmpleados
```

'Subrutina "Aceptar" aplicable a todos los botones "Aceptar" del menú "Reporte" que dependiendo del submenú seleccionado realiza la ejecución del mismo.

```
Private Sub btnAceptar_Click()

    If grdEmpleados.IsBlockSelected = False Then

        MsgBox "Debe seleccionar un empleado", vbInformation, Me.Caption
        Exit Sub

    End If

    If strMenu = "ConDatosEmp" Then

        frmConEmpleados.Show

    ElseIf strMenu = "ConHorarioAsig" Then

        frmConHorarioAsig.Show

    ElseIf strMenu = "ConVacaciones" Then

        frmConVacaciones.Show

    End If

End Sub
```

'Subrutina "Agregar" aplicable a todos los botones "Agregar" del menú "Administración" que dependiendo del submenú seleccionado realiza la ejecución del mismo.

```
Private Sub btnAgregar_Click()
```

```
    strAccion = "Agregar"
```

```
    'Bloque perteneciente al submenú "Vacaciones", escoge al empleado para el proceso de agregar vacaciones.
```

```
    If strMenu = "Vacaciones" Then
```

```
        If    grdEmpleados.IsBlockSelected = True Then
```

```
            frmVacaciones.Show
```

```
        Else
```

```
            'Si no se escoge un empleado en la grilla, se visualiza el siguiente mensaje:
```

```
                MsgBox "Debe seleccionar un empleado", vbInformation, Me.Caption
```

```
    End If
```

```
    'Bloque perteneciente al submenú "Asignar Horario", escoge al empleado para el proceso de asignar horario.
```

```
    ElseIf strMenu = "Asignar Horario" Then
```

```
        Dim strHor As String
```

```
        Set rs = objEmpleados.buscaHorarioEmp(strCodigoEmpleado)
```

```
        While Not rs.EOF
```

```
            strHor = rs!strCodigoHorario
```

```
            rs.MoveNext
```

```
        Wend
```

```
        rs.Close
```

```
        If    strHor = "" Then
```

```
            frmAsignarHorario.Show
```

```
        Else
```

```
            'Si el empleado ya tiene asignado un horario se visualiza:
```

```
                MsgBox "El empleado ya tiene un horario asignado.", vbInformation,
```

```
                Me.Caption
```

```
        End If
```

```
    'Bloque perteneciente al submenú "Empleados". escoge al empleado para el proceso de agregar empleado.
```

```
Elseif strMenu = "Empleados" Then
```

```
    frmEmpleados.Show
```

'Bloque perteneciente al submenú "Justificaciones", escoge al empleado para el proceso agregar justificaciones.

```
Elseif strMenu = "Justificacion" Then
```

```
    If grdEmpleados.IsBlockSelected = True Then  
        frmSelFechas.Show
```

```
Else
```

Si no se escoge un empleado en la grilla, se visualiza el siguiente mensaje:

```
    MsgBox "Debe seleccionar un empleado.", vbInformation, Me.Caption
```

```
End If
```

```
End If
```

```
End Sub
```

'Subrutina que invoca a la función "buscaEmpleado".

```
Private Sub btnBuscar_Click()
```

```
    buscaEmpleado
```

```
End Sub
```

'Subrutina "Eliminar" aplicable a todos los botones "Eliminar" del menú "Administración" que dependiendo del submenú seleccionado realiza la ejecución del mismo.

```
Private Sub btnEliminar_Click()
```

```
    Dim intP As Integer
```

```
    If    grdEmpleados.IsBlockSelected = False Then
```

```
        MsgBox "Debe seleccionar un empleado", vbInformation, Me.Caption  
        Exit Sub
```

```
    End If
```

'Bloque perteneciente al submenú "Empleados", escoge al empleado para el proceso de eliminación.

```
    If    strMenu = "Empleados" Then
```

```
        if    grdEmpleados.MaxRows <> 0 Then
```

```
intP = MsgBox("Está seguro de que desea eliminar el empleado?",
vbYesNo, Me.Caption)
```

```
If intP = vbYes Then
```

```
grdEmpleados.Row = grdEmpleados.ActiveRow
grdEmpleados.Col = 3
```

```
objEmpleados.eliminarEmpleado (grdEmpleados.Text)
```

```
Unload Me
frmAdmEmpleados.Show
```

```
End If
```

```
End If
```

'Bloque perteneciente al submenú "Asignar Horario", escoge al empleado para el proceso de eliminación.

```
Elsif strMenu = "Asignar Horario" Then
```

```
intP = MsgBox("Está seguro de que desea desasignar el horario?", vbYesNo,
Me.Caption)
```

```
If intP = vbYes Then
```

```
objEmpleados.desAsignarHorario (strCodigoEmpleado)
```

```
End If
```

'Bloque perteneciente al submenú "Vacaciones", escoge al empleado para el proceso de eliminación.

```
Elsif strMenu = "Vacaciones" Then
```

```
Dim strFechaS As String
Dim strFechaR As String
```

```
Set rs = objEmpleados.buscaUltVacaciones(strCodigoEmpleado)
```

```
While Not rs.EOF
```

```
strFechaS = rs!dtFechaSalida
strFechaR = rs!dtFechaRegreso
rs.MoveNext
```

```
Wend
```

```
rs.Close
```

```
intP = MsgBox("Está seguro de que desea eliminar las vacaciones de fechas:" & _
strFechaS & " y " & strFechaR & "?", vbYesNo, Me.Caption)
```

```
If intP = vbYes Then
```

```
objEmpleados.eliminarVacaciones (strCodigoEmpleado)
```

```
End If
```

```
End If
```

```
End Sub
```

'Subrutina "Modificar" aplicable a todos los botones "Modificar" del menú "Administración" que dependiendo del submenú seleccionado realiza la ejecución del mismo.

```
Private Sub btnModificar_Click()
```

```
    If grdEmpleados.IsBlockSelected = True Then
```

```
        strAccion = "Modificar"
```

```
        'Bloque perteneciente al submenú "Vacaciones", escoge al empleado para el proceso de modificación.
```

```
        If strMenu = "Vacaciones" Then
```

```
            Dim strFechaS As String
```

```
            Dim strFechaR As String
```

```
            Set rs = objEmpleados.buscaUltVacaciones(strCodigoEmpleado)
```

```
            While Not rs.EOF
```

```
                strFechaS = rs!dtFechaSalida
```

```
                strFechaR = rs!dtFechaRegreso
```

```
                rs.MoveNext
```

```
            Wend
```

```
            rs.Close
```

```
            If strFechaS = "" And strFechaR = "" Then
```

```
                MsgBox "El empleado no tiene vacaciones.", vbInformation, Me.Caption
```

```
            Exit Sub
```

```
        End If
```

```
        frmVacaciones.Show
```

```
        Unload Me
```

```
        'Bloque perteneciente al submenú "Asignar Horario", escoge al empleado para el proceso de modificación.
```

```
    Elseif strMenu = "Asignar Horario" Then
```

```
Dim strHor As String

Set rs = objEmpleados.buscaHorarioEmp(strCodigoEmpleado)

While Not rs.EOF
    strHor = rs!strCodigoHorario
    rs.MoveNext
Wend

rs.Close

If strHor <> "" Then

    frmAsignarHorario.Show
    Unload Me

Else

    MsgBox "El empleado no tiene asignado un horario.",
        vbInformation, Me.Caption

End If
```

'Bloque perteneciente al submenú "Empleados", escoge al empleado para el proceso de modificación.

```
Elseif strMenu = "Empleados" Then
```

```
    frmEmpleados.Show
    Unload Me
```

'Bloque perteneciente al submenú "Justificaciones", escoge al empleado para el proceso de modificación.

```
Elseif strMenu = "Justificacion" Then
```

```
    frmJustificaciones.Show
    Unload Me
```

```
End If
```

```
Else
```

```
    MsgBox "Debe seleccionar un empleado.", vbInformation, Me.Caption
```

```
End If
```

```
End Sub
```

'Subrutina que se invoca para "salir" del submenú que se encuentre ejecutando en ese momento

```
Private Sub btnSalir_Click()
```

```
    Unload Me
```

```
End Sub
```

'Subrutina que carga la ventana principal de los submenús que tengan que desplegar la grilla con los botones correspondientes

Private Sub Form_Load()

Dim objEmpleado As New clsEmpleados
Set rs = objEmpleado.listarEmpleados

listaEmpleados grdEmpleados, rs

'Subrutina que permite cambiar el nombre de la etiqueta: de ASIGNAR a AGREGAR y viceversa.

If strMenu = "SI" Then

btnAgregar.Caption = "&Asignar"

Else

btnAgregar.Caption = "&Agregar"

End If

'Subrutina que habilita y deshabilita los botones en la ventana principal dependiendo del submenú escogido.

If strMenu = "ConDatosEmp" Or strMenu = "ConHorarioAsig" Or _
strMenu = "ConVacaciones" Then

btnAgregar.Visible = False
btnModificar.Visible = False
btnEliminar.Visible = False
btnAceptar.Visible = True

Else

btnAgregar.Visible = True
btnModificar.Visible = True
btnEliminar.Visible = True
btnAceptar.Visible = False

End If

If strMenu = "Justificacion" Then

btnModificar.Visible = False
btnEliminar.Visible = False

End If

End Sub

'Subrutina que permite seleccionar un empleado de la grilla para su correspondiente proceso.

Private Sub grdEmpleados_Click(ByVal Col As Long, ByVal Row As Long)

grdEmpleados.Row = grdEmpleados.ActiveRow


```

grdEmpleados.Col = 3
strCodigoEmpleado = grdEmpleados.Text
grdEmpleados.Col = 1
strNombreEmpleado = grdEmpleados.Text
grdEmpleados.Col = 2
strNombreEmpleado = strNombreEmpleado & " " & grdEmpleados.Text

```

End Sub

'Subrutina buscaEmpleado que permite visualizar en la grilla el empleado escogido en la caja de texto "Buscar".

```
Private Sub buscaEmpleado()
```

```
    On Error GoTo sigue
```

```
    Set rs = objEmpleados.buscaEmpleadoAp(txtBuscar.Text)
```

```
    grdEmpleados.MaxRows = 0
```

```
    listaEmpleados grdEmpleados, rs
```

```
Exit Sub
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

'Subrutina que llama a la función "AlfabeticoConEspacio" la cual permite escribir solo letras del alfabeto con espacio para el ingreso de los Nombres de un empleado para su búsqueda en la grilla.

```
Private Sub txtBuscar_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = AlfabeticoConEspacio(KeyAscii)
```

```
End Sub
```

Código cargado en la forma fmrEmpleados

Esta forma hace uso de los siguientes módulos: Módulo1, clsEmpleados y clsEmpleadosDat.

'Declaración de variables.

```

Dim rs As New ADODB.Recordset
Dim objEmpleados As New clsEmpleados
Dim strSexo As String

```

'Subrutina del botón "Aceptar", evalúa que los campos obligatorios estén llenos correctamente tanto para la acción "Agregar" como "Modificar" para un empleado.

```
Private Sub btnAceptar_Click()

    If Len(txtCedula) <> 10 Then

        MsgBox "La cédula debe tener 10 dígitos.", vbInformation, Me.Caption
        Exit Sub

    End If

    If strAccion = "Agregar" Then

        If txtCodigo.Text <> "" And txtNombres.Text <> "" And txtApellidos.Text <> ""
            And txtCedula.Text <> "" Then

                objEmpleados.ingresarEmpleado txtCodigo.Text, txtNombres.Text _
                , txtApellidos.Text, txtDireccion.Text, txtTelefono1.Text, txtCedula.Text, _
                strSexo, dtFechaNac.Value, cmbEstadoCivil.Text, _
                dtFechaIngreso.Value, txtTelefono2.Text

                frmAdmEmpleados.Show
                Unload Me

            Else

                MsgBox "El código, nombres, apellidos y cédula no pueden ser vacios.",
                vbInformation, Me.Caption

            End If

        ElseIf strAccion = "Modificar" Then

            objEmpleados.modificarEmpleado txtCodigo.Text, txtNombres.Text,
            txtApellidos.Text, _ txtDireccion.Text, txtTelefono1.Text, txtCedula.Text, _
            strSexo, dtFechaNac.Value, cmbEstadoCivil.Text, dtFechaIngreso.Value,
            txtTelefono2.Text

            frmAdmEmpleados.Show
            Unload Me

        End If

    End Sub
```

'Subrutina del botón "Cancelar".

```
Private Sub btnCancelar_Click()

    Unload Me
    frmAdmEmpleados.Show

End Sub
```

'Subrutina "llenaEstadoCivil", llena la lista del combo "Estado Civil" con las opciones disponibles en la Tabla "EstadoCivil" de la Base de Datos del SCA.

```
Private Sub llenaEstadoCivil()  
    On Error GoTo sigue  
  
    strSql = "SELECT strDescripcion " & _  
            "FROM EstadoCivil "  
  
    rs.Open strSql, con, adOpenDynamic, adLockOptimistic  
  
    While Not rs.EOF  
        cmbEstadoCivil.AddItem rs!strDescripcion  
        rs.MoveNext  
    Wend  
  
    cmbEstadoCivil.Text = "SOLTERO"  
  
    rs.Close  
  
    Exit Sub  
sigue:  
    MsgBox Err.Description, vbCritical, Err.Source  
  
End Sub
```

'Subrutina que evita almacenar una fecha de ingreso del empleado mayor a la fecha actual.

```
Private Sub dtFechaIngreso_Change()  
    If dtFechaIngreso.Value > Date Then  
        MsgBox "La fecha de ingreso no puede ser mayor a la fecha actual",  
            vbInformation, Me.Caption  
        dtFechaIngreso.Value = Date  
    End If  
  
End Sub
```

'Subrutina que evita almacenar una fecha de nacimiento del empleado mayor a la fecha actual.

```
Private Sub dtFechaNac_Change()  
    If dtFechaNac.Value > Date Then  
        MsgBox "La fecha de nacimiento no puede ser mayor a la fecha actual",  
            vbInformation, Me.Caption  
        dtFechaNac.Value = Date  
    End If  
  
End Sub
```

```
End If
```

```
End Sub
```

'Subrutina que se activa cuando se carga la forma, setea el ingreso de las fechas con la fecha actual, el sexo en masculino y llama a la subrutina "llenaEstadoCivil" y "llena forma".

```
Private Sub Form_Load()
```

```
    dtFechaIngreso.Value = Date
    dtFechaNac.Value = Date
    llenaEstadoCivil
    strSexo = "M"
    llenaForma
```

```
End Sub
```

'Subrutina "llenaForma", cuando se selecciona la opción "Modificar" extrae los datos almacenados del empleado para llenar en los respectivos campos.

```
Private Sub llenaForma()
```

```
    On Error GoTo sigue
```

```
    Dim strSex As String
```

```
    If strAccion <> "Modificar" Then
```

```
        Exit Sub
```

```
    End If
```

```
    Set rs = objEmpleados.consultarEmpleado(strCodigoEmpleado)
```

```
    While Not rs.EOF
```

```
        txtCodigo.Text = rs!strCodigoEmpleado
        txtNombres.Text = rs!strNombres
        txtApellidos.Text = rs!strApellidos
        txtCedula.Text = rs!strCedula
        dtFechaNac.Value = rs!dtFechaNacimiento
        cmbEstadoCivil.Text = rs!strEstadoCivil
        txtDireccion.Text = rs!strDireccion
        txtTelefono1.Text = rs!strTelefono1
        txtTelefono2.Text = rs!strTelefono2
        dtFechaIngreso.Value = rs!dtIngreso
        strSex = rs!strSexo
```

```
        rs.MoveNext
```

```
    Wend
```

```
    If strSex = "F" Then
```

```
        optFemenino.Value = True
```

```
Elseif strSex = "M" Then
    optMasculino.Value = True
End If
rs.Close
Exit Sub
    sigue:
        MsgBox Err.Description, vbc, Err.Source
End Sub
```

'Subrutinas que setean el tipo de sexo del empleado.

```
Private Sub optFemenino_Click()
    If optFemenino.Value = True Then
        strSexo = "F"
    End If
End Sub
```

```
Private Sub optMasculino_Click()
    If optMasculino.Value = True Then
        strSexo = "M"
    End If
End Sub
```

'Subrutina que llama a la función "AlfabeticoConEspacio" la cual permite escribir solo letras del alfabeto con espacio para el ingreso de los Apellidos.

```
Private Sub txtApellidos_KeyPress(KeyAscii As Integer)
    KeyAscii = AlfabeticoConEspacio(KeyAscii)
End Sub
```

'Subrutina que llama a la función "Numerico" que solo permite escribir números para el ingreso de la cédula.

```
Private Sub txtCedula_KeyPress(KeyAscii As Integer)
    KeyAscii = Numerico(KeyAscii)
End Sub
```

'Subrutina que llama a la función "AlfabeticoConEspacioNum" la cual permite escribir letras del alfabeto y números con espacio exceptuando otros símbolos y caracteres para el ingreso de la Dirección.

```
Private Sub txtDireccion_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = AlfabeticoConEspacioNum(KeyAscii)
```

```
End Sub
```

'Subrutina que llama a la función "AlfabeticoConEspacio" la cual permite escribir solo letras del alfabeto con espacio para el ingreso de los Nombres.

```
Private Sub txtNombres_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = AlfabeticoConEspacio(KeyAscii)
```

```
End Sub
```

'Subrutina que llama a la función "Numerico" que solo permite escribir números para el ingreso del teléfono.

```
Private Sub txtTelefono1_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Numerico(KeyAscii)
```

```
End Sub
```

'Subrutina que llama a la función "Numerico" que solo permite escribir números para el ingreso del teléfono2.

```
Private Sub txtTelefono2_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Numerico(KeyAscii)
```

```
End Sub
```

Código cargado en la forma fmrAdmHorarios

Esta forma hace uso de los siguientes módulos: clsHorario y clsHorarioDat.

'Subrutinas del submenú "Horarios".

```
Dim rs As New ADODB.Recordset  
Dim objHorario As New clsHorarios
```

'Botón Aceptar.

```
Private Sub btnAceptar_Click()
```

```
    frmConHorario.Show
```

```
End Sub
```

'Botón Agregar.

```
Private Sub btnAgregar_Click()
```

```
    strAccion = "Agregar"  
    frmDefiHorarios.Show  
    Unload Me
```

```
End Sub
```

'Botón Eliminar.

```
Private Sub btnEliminar_Click()
```

```
    Dim intP As Integer
```

```
    intP = MsgBox("Está seguro de que desea eliminar el horario?", vbYesNo, Me.Caption)
```

```
    If intP = vbYes Then
```

```
        objHorario.elimnarHorario (strCodigoHorario)
```

```
        Unload Me  
        frmAdmHorarios.Show
```

```
    End If
```

```
End Sub
```

'Botón Modificar.

```
Private Sub btnModificar_Click()
```

```
    If grdHorarios.IsBlockSelected = True Then
```

```
        strAccion = "Modificar"  
        frmDefiHorarios.Show
```

```
    Else
```

```
        MsgBox "Debe seleccionar un horario.", vbInformation, Me.Caption
```

```
    End If
```

```
End Sub
```

'Botón Salir.

```
Private Sub btnSalir_Click()
```

```
    Unload Me
```

```
End Sub
```

'Subrutina que carga la interfaz de la grilla "Administración" con los botones correspondientes.

```
Private Sub Form_Load()
```

```
    grdHorarios.SortKeyOrder(1) = 1  
    llenaGrillaHorario
```

```
    If strMenu = "Horarios" Then
```

```
        btnAgregar.Visible = True  
        btnModificar.Visible = True  
        btnEliminar.Visible = True  
        btnAceptar.Visible = False
```

```
    ElseIf strMenu = "ConHorarios" Then
```

```
        btnAgregar.Visible = False  
        btnModificar.Visible = False  
        btnEliminar.Visible = False  
        btnAceptar.Visible = True
```

```
    End If
```

```
End Sub
```

'Subrutina que carga la grilla "Administración" con los horarios creados hasta la fecha.

```
Private Sub llenaGrillaHorario()
```

```
    On Error GoTo sigue
```

```
    Dim intContadorFilas As Integer
```

```
    Set rs = objHorario.listaHorarios
```

```
    grdHorarios.Col = -1  
    grdHorarios.Row = -1  
    grdHorarios.Lock = True  
    intContadorFilas = 1
```

```
    While Not rs.EOF
```

```
        grdHorarios.Col = 1  
        grdHorarios.MaxRows = intContadorFilas  
        grdHorarios.Row = intContadorFilas  
        grdHorarios.Text = rs!strCodigoHorario
```



```

        grdHorarios.Col = 2
        grdHorarios.Text = rs!strDescripcion

        intContadorFilas = intContadorFilas + 1
        rs.MoveNext

    Wend

    rs.Close

Exit Sub
    sigue:
        MsgBox Err.Description, vbCritical, Err.Source

End Sub

```

'Subrutina que permite escoger un Horario creado de la grilla "Administración" para un proceso escogido por los botones.

```

Private Sub grdHorarios_Click(ByVal Col As Long, ByVal Row As Long)

    grdHorarios.Row = grdHorarios.ActiveRow
    grdHorarios.Col = 1
    strCodigoHorario = grdHorarios.Text

End Sub

```

Código cargado en la forma frmAsignarHorarios

Esta forma hace uso de los siguientes módulos: Módulo1, clsEmpleados, clsEmpleadosDat, clsHorario y clsHorarioDat.

'Declaración de Variables.

```

    Dim objEmpleado As New clsEmpleados
    Dim objHorario As New clsHorarios
    Dim rs As New ADODB.Recordset

```

'Botón Modificar o Asignar dependiendo del proceso escogido.

```

Private Sub btnAceptar_Click()

    If strAccion = "Modificar" Then

        objEmpleado.desAsignarHorario strCodigoEmpleado

    End If

    asignarHorario

```

```

    frmAdmEmpleados.Caption = "Asignación de Horarios"
    frmAdmEmpleados.Show
    Unload Me

```

```
End Sub
```

'Botón Salir.

```
Private Sub btnSalir_Click()
```

```

    frmAdmEmpleados.Caption = "Asignación de Horarios"
    frmAdmEmpleados.Show
    Unload Me

```

```
End Sub
```

'Subrutina que carga la ventana para la asignación de un horario.

```
Private Sub Form_Load()
```

```

    llenarEmpleado
    llenaComboHorario

    If strAccion = "Modificar" Then
        marcarHorario
    End If

```

```
End Sub
```

'Subrutina "llenarEmpleado" que llena la ventana para la asignación de horario con los datos personales del empleado en los respectivas caja de texto.

```
Private Sub llenarEmpleado()
```

```
On Error GoTo sigue
```

```

    Set rs = objEmpleado.consultarEmpleado(strCodigoEmpleado)

    While Not rs.EOF

        txtCodigo.Text = rs!strCodigoEmpleado
        txtEmpleado.Text = rs!strApellidos & " " & rs!strNombres

        rs.MoveNext

    Wend

    rs.Close

```

```
Exit Sub
```

```

sigue:
    MsgBox Err.Description, vbCritical, Err.Source

```

```
End Sub
```

'Subrutina "llenaComboHorario" que llena el combo "Horarios" para que el usuario pueda escoger cuál de ellos se va a asignar al empleado.

```
Private Sub llenaComboHorario()
```

```
On Error GoTo sigue
```

```
    Set rs = objHorario.listaHorarios
```

```
    While Not rs.EOF
```

```
        cmbHorario.AddItem rs!strDescripcion & " - " & rs!strCodigoHorario
        rs.MoveNext
```

```
    Wend
```

```
    rs.Close
```

```
Exit Sub
```

```
sigue:
```

```
    MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

'Subrutina que busca el código del horario escogido para la asignación al respectivo empleado.

```
Private Function conseguirCodigo() As String
```

```
On Error GoTo sigue
```

```
    Dim strCodigo As String
```

```
    Dim intPos As Integer
```

```
    strCodigo = cmbHorario.Text
```

```
    intPos = InStrRev(strCodigo, " ")
```

```
    strCodigo = Mid(strCodigo, intPos + 1, Len(strCodigo))
```

```
    conseguirCodigo = strCodigo
```

```
Exit Function
```

```
sigue:
```

```
    MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

'Subrutina que enlaza el código del empleado con el código del horario asignado.

```
Private Sub asignarHorario()
```

```
On Error GoTo sigue
```

```
    Dim strHorario As String
```

```
    strHorario = conseguirCodigo
```

```
    objEmpleado.asignarHorario strCodigoEmpleado, strHorario
```

```
Exit Sub
    sigue:
        MsgBox Err.Description, vbCritical, Err.Source
End Sub
```

'Subrutina "marcarHorario" que captura del combo "Horarios" el horario escogido para la asignación.

```
Private Sub marcarHorario()
```

```
On Error GoTo sigue
```

```
    Dim intPos As Integer
    Dim intl As Integer
    Dim strCod As String

    Set rs = objEmpleado.buscaHorarioEmp(strCodigoEmpleado)

    For intl = 0 To cmbHorario.ListCount - 1

        intPos = InStrRev(cmbHorario.List(intl), " ")
        strCod = Mid(cmbHorario.List(intl), intPos + 1, Len(cmbHorario.List(intl)))

        If strCod = rs!strCodigoHorario Then

            cmbHorario.Text = cmbHorario.List(intl)

            Exit Sub

        End If

    Next
```

```
Exit Sub
    sigue:
        MsgBox Err.Description, vbCritical, Err.Source
End Sub
```

Código cargado en la forma frmDefiHorarios

Esta forma hace uso de los siguientes módulos: Módulo1, clsHorario y clsHorarioDat.

'Declaración de variables.

```
Dim objHorarios As New clsHorarios
Dim rs As New ADODB.Recordset
```

'Botón Cancelar.

```
Private Sub btnCancelar_Click()
```

```
    Unload Me
```

```
End Sub
```

'Subrutina "Ingresar" la cual llama a la forma frmHorarios siempre que exista al menos un día con horario.

```
Private Sub btnIngresar_Click()
```

```
    If cmbLunes.Text = "No ingresar horario" And _
       cmbMartes.Text = "No ingresar horario" And _
       cmbMiercoles.Text = "No ingresar horario" And _
       cmbJueves.Text = "No ingresar horario" And _
       cmbViernes.Text = "No ingresar horario" And _
       cmbSabado.Text = "No ingresar horario" Then
```

```
        MsgBox "Debe definir al menos un día.", vbInformation, Me.Caption
```

```
        Exit Sub
```

```
    End If
```

```
    If txtHorario.Text <> "" And txtDescripcion.Text <> "" Then
```

```
        frmHorarios.Show
```

```
    Else
```

```
        MsgBox "El código y la descripción no pueden estar vacíos.",
        vbInformation, Me.Caption
```

```
    End If
```

```
End Sub
```

'Subrutina que en la opción "Modificar" borra las variables para ingresar una jornada doble diferenciada para el día Lunes.

```
Private Sub chkDiferenciadoL_Click()
```

```
    If strAccion = "Modificar" Then
```

```
        frmHorarios.lstEntrada1L.Clear
```

```
        frmHorarios.lstEntrada2L.Clear
```

```
        frmHorarios.lstSalida1L.Clear
```

```
        frmHorarios.lstSalida2L.Clear
```

```
        Unload frmHorarios
```

```
    End If
```

```
End Sub
```

'Subrutina que en la opción "Modificar" borra las variables para ingresar una jornada doble diferenciada para el día Martes.

```
Private Sub chkDiferenciadoM_Click()  
  
    If      strAccion = "Modificar" Then  
  
        frmHorarios.LstEntrada1M.Clear  
        frmHorarios.LstEntrada2M.Clear  
        frmHorarios.LstSalida1M.Clear  
        frmHorarios.LstSalida2M.Clear  
        Unload frmHorarios  
  
    End If  
  
End Sub
```

'Subrutina que en la opción "Modificar" borra las variables para ingresar una jornada doble diferenciada para el día Miércoles.

```
Private Sub chkDiferenciadoMr_Click()  
  
    If      strAccion = "Modificar" Then  
  
        frmHorarios.LstEntrada1Mr.Clear  
        frmHorarios.LstEntrada2Mr.Clear  
        frmHorarios.LstSalida1Mr.Clear  
        frmHorarios.LstSalida2Mr.Clear  
        Unload frmHorarios  
  
    End If  
  
End Sub
```

'Subrutina que en la opción "Modificar" borra las variables para ingresar una jornada doble diferenciada para el día Jueves.

```
Private Sub chkDiferenciadoJ_Click()  
  
    If      strAccion = "Modificar" Then  
  
        frmHorarios.LstEntrada1J.Clear  
        frmHorarios.LstEntrada2J.Clear  
        frmHorarios.LstSalida1J.Clear  
        frmHorarios.LstSalida2J.Clear  
        Unload frmHorarios  
  
    End If  
  
End Sub
```

'Subrutina que en la opción "Modificar" borra las variables para ingresar una jornada doble diferenciada para el día Viernes.

```
Private Sub chkDiferenciadoV_Click()
```

```
    If strAccion = "Modificar" Then
        frmHorarios.LstEntrada1V.Clear
        frmHorarios.LstSalida1V.Clear
        frmHorarios.LstSalida2V.Clear
        frmHorarios.LstEntrada2V.Clear
        Unload frmHorarios
```

```
    End If
```

```
End Sub
```

'Subrutina que en la opción "Modificar" borra las variables para ingresar una jornada doble diferenciada para el día Sábado.

```
Private Sub chkDiferenciadoS_Click()
```

```
    If strAccion = "Modificar" Then
        frmHorarios.LstEntrada1S.Clear
        frmHorarios.LstEntrada2S.Clear
        frmHorarios.LstSalida1S.Clear
        frmHorarios.LstSalida2S.Clear
        Unload frmHorarios
```

```
    End If
```

```
End Sub
```

'Subrutina que no permite escribir o modificar el texto del combo de selección del tipo de jornada para el día Lunes.

```
Private Sub cmbLunes_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = 0
```

```
End Sub
```

'Subrutina que habilita las cajas de texto para ingresar las horas dependiendo del tipo de jornada escogida para el día Lunes.

```
Private Sub cmbLunes_Click()
```

'Si se escoge la opción "Jornada Diferenciada" se habilita la opción de ser doble, caso contrario se deshabilita.

```
    If cmbLunes.Text = "Jornada Diferenciada" Then
        chkDiferenciadoL.Enabled = True
```

```
    Else
```

```
chkDiferenciadoL.Enabled = False
```

```
End If
```

'Si la opción es modificar, borra las variables para ingresar otro tipo de horario.

```
If straccion = "Modificar" Then
```

```
frmHorarios.lstEntrada1L.Clear
frmHorarios.lstEntrada2L.Clear
frmHorarios.lstSalida1L.Clear
frmHorarios.lstSalida2L.Clear
Unload frmHorarios
```

```
End If
```

```
End Sub
```

'Subrutina que no permite escribir o modificar el texto del combo de selección del tipo de jornada para el día Martes.

```
Private Sub cmbMartes_KeyPress(KeyAscii As Integer)
```

```
KeyAscii = 0
```

```
End Sub
```

'Subrutina que habilita las cajas de texto para ingresar las horas dependiendo del tipo de jornada escogida para el día Martes.

```
Private Sub cmbMartes_Click()
```

'Si se escoge la opción "Jornada Diferenciada" se habilita la opción de ser doble, caso contrario se deshabilita.

```
If cmbMartes.Text = "Jornada Diferenciada" Then
```

```
chkDiferenciadoM.Enabled = True
```

```
Else
```

```
chkDiferenciadoM.Enabled = False
```

```
End If
```

'Si la opción es modificar, borra las variables para ingresar otro tipo de horario.

```
If strAccion = "Modificar" Then
```

```
frmHorarios.lstEntrada1M.Clear
frmHorarios.lstEntrada2M.Clear
frmHorarios.lstSalida1M.Clear
frmHorarios.lstSalida2M.Clear
```

Unload frmHorarios**End If****End Sub**

'Subrutina que no permite escribir o modificar el texto del combo de selección del tipo de jornada para el día Miércoles.

Private Sub cmbMiercoles_KeyPress(KeyAscii As Integer)**KeyAscii = 0****End Sub**

'Subrutina que habilita las cajas de texto para ingresar las horas dependiendo del tipo de jornada escogida para el día Miércoles.

Private Sub cmbMiercoles_Click()

'Si se escoge la opción "Jornada Diferenciada" se habilita la opción de ser doble, caso contrario se deshabilita.

If cmbMiercoles.Text = "Jornada Diferenciada" Then**chkDiferenciadoMr.Enabled = True****Else****chkDiferenciadoMr.Enabled = False****End If**

'Si la opción es modificar, borra las variables para ingresar otro tipo de horario.

If strAccion = "Modificar" Then**frmHorarios.LstEntrada1Mr.Clear****frmHorarios.IstEntrada2Mr.Clear****frmHorarios.LstSalida1Mr.Clear****frmHorarios.LstSalida2Mr.Clear****Unload frmHorarios****End If****End Sub**

'Subrutina que no permite escribir o modificar el texto del combo de selección del tipo de jornada para el día Jueves.

Private Sub cmbJueves_KeyPress(KeyAscii As Integer)**KeyAscii = 0**

End Sub

'Subrutina que habilita las cajas de texto para ingresar las horas dependiendo del tipo de jornada escogida para el día Jueves.

Private Sub cmbJueves_Click()

'Si se escoge la opción "Jornada Diferenciada" se habilita la opción de ser doble, caso contrario se deshabilita.

If cmbJueves.Text = "Jornada Diferenciada" Then

chkDiferenciadoJ.Enabled = True

Else

chkDiferenciadoJ.Enabled = False

End If

'Si la opción es modificar, borra las variables para ingresar otro tipo de horario.

If strAccion = "Modificar" Then

frmHorarios.LstEntrada1J.Clear

frmHorarios.LstEntrada2J.Clear

frmHorarios.LstSalida1J.Clear

frmHorarios.LstSalida2J.Clear

Unload frmHorarios

End If

End Sub

'Subrutina que no permite escribir o modificar el texto del combo de selección del tipo de jornada para el día Viernes.

Private Sub cmbViernes_KeyPress(KeyAscii As Integer)

KeyAscii = 0

End Sub

'Subrutina que habilita las cajas de texto para ingresar las horas dependiendo del tipo de jornada escogida para el día Viernes.

Private Sub cmbViernes_Click()

'Si se escoge la opción "Jornada Diferenciada" se habilita la opción de ser doble, caso contrario se deshabilita.

If cmbViernes.Text = "Jornada Diferenciada" Then

chkDiferenciadoV.Enabled = True

Else

 chkDiferenciadoV.Enabled = False

End If

'Si la opción es modificar, borra las variables para ingresar otro tipo de horario.

If strAccion = "Modificar" Then

 frmHorarios.LstEntrada1V.Clear
 frmHorarios.LstSalida1V.Clear
 frmHorarios.LstSalida2V.Clear
 frmHorarios.LstEntrada2V.Clear
 Unload frmHorarios

End If

End Sub

'Subrutina que no permite escribir o modificar el texto del combo de selección del tipo de jornada para el día Sábado.

Private Sub cmbSabado_KeyPress(KeyAscii As Integer)

 KeyAscii = 0

End Sub

'Subrutina que habilita las cajas de texto para ingresar las horas dependiendo del tipo de jornada escogida para el día Sábado.

Private Sub cmbSabado_Click()

 'Si se escoge la opción "Jornada Diferenciada" se habilita la opción de ser doble, caso contrario se deshabilita.

If cmbSabado.Text = "Jornada Diferenciada" Then

 chkDiferenciadoS.Enabled = True

Else

 chkDiferenciadoS.Enabled = False

End If

'Si la opción es modificar, borra las variables para ingresar otro tipo de horario

If strAccion = "Modificar" Then

 frmHorarios.LstEntrada1S.Clear
 frmHorarios.LstEntrada2S.Clear
 frmHorarios.LstSalida1S.Clear

```

frmHorarios.LstSalida2S.Clear
Unload frmHorarios

```

```
End If
```

```
End Sub
```

'Subrutina que en la opción "Modificar" llama a la subrutina "llenaPlan" la cual extrae y visualiza los datos anteriormente ingresados.

```
Private Sub Form_Load()
```

```

If strAccion = "Modificar" Then

    btnIngresar.Caption = "Modificar"
    llenaPlan

```

```
End If
```

```
Unload frmHorarios
```

```
End Sub
```

'Subrutina que extrae los horarios de la Base de Datos del SCA y los visualiza antes de su modificación.

```
Private Sub llenaPlan()
```

```
On Error GoTo sigue
```

```
Dim intPos As Integer
```

```
Set rs = objHorarios.consultaHorarios(strCodigoHorario)
```

```
While Not rs.EOF
```

```

txtHorario.Text = rs!strCodigoHorario
txtDescripcion.Text = rs!strDescripcion

```

```
'Día Lunes:
```

```

intPos = InStrRev(rs!strJornadaL, " ", Len(rs!strJornadaL))
cmbLunes.Text = Mid(rs!strJornadaL, 1, intPos - 1)

```

```
If Mid(rs!strJornadaL, intPos + 1, Len(rs!strJornadaL)) = 1 Then
```

```

    chkDiferenciadoL.Value = 1
    chkDiferenciadoL.Enabled = True

```

```
Elseif cmbLunes.Text = "Jornada Diferenciada" And Mid(rs!strJornadaL, intPos + 1, Len(rs!strJornadaL)) = 0 Then
```

```
    chkDiferenciadoL.Enabled = True
```

```
End If
```

'Día Martes:

```
intPos = InStrRev(rs!strJornadaM, " ", Len(rs!strJornadaM))
cmbMartes.Text = Mid(rs!strJornadaM, 1, intPos - 1)

If      Mid(rs!strJornadaM, intPos + 1, Len(rs!strJornadaM)) = 1 Then

    chkDiferenciadoM.Value = 1
    chkDiferenciadoM.Enabled = True

Elseif  cmbMartes.Text = "Jornada Diferenciada" And Mid(rs!strJornadaM, intPos + 1,
    Len(rs!strJornadaM)) = 0 Then

    chkDiferenciadoM.Enabled = True
End If
```

'Día Miércoles:

```
intPos = InStrRev(rs!strJornadaMr, " ", Len(rs!strJornadaMr))
cmbMiercoles.Text = Mid(rs!strJornadaMr, 1, intPos - 1)

If      Mid(rs!strJornadaMr, intPos + 1, Len(rs!strJornadaMr)) = 1 Then

    chkDiferenciadoMr.Value = 1
    chkDiferenciadoMr.Enabled = True

Elseif  cmbMiercoles.Text = "Jornada Diferenciada" And Mid(rs!strJornadaMr, intPos + 1,
    Len(rs!strJornadaMr)) = 0 Then

    chkDiferenciadoMr.Enabled = True
End If
```

'Día Jueves:

```
intPos = InStrRev(rs!strJornadaJ, " ", Len(rs!strJornadaJ))
cmbJueves.Text = Mid(rs!strJornadaJ, 1, intPos - 1)

If      Mid(rs!strJornadaJ, intPos + 1, Len(rs!strJornadaJ)) = 1 Then

    chkDiferenciadoJ.Value = 1
    chkDiferenciadoJ.Enabled = True

Elseif  cmbJueves.Text = "Jornada Diferenciada" And Mid(rs!strJornadaJ, intPos + 1,
    Len(rs!strJornadaJ)) = 0 Then

    chkDiferenciadoJ.Enabled = True
End If
```

'Día Viernes:

```
intPos = InStrRev(rs!strJornadaV, " ", Len(rs!strJornadaV))
cmbViernes.Text = Mid(rs!strJornadaV, 1, intPos - 1)
```

```
If Mid(rs!strJornadaV, intPos + 1, Len(rs!strJornadaV)) = 1 Then
    chkDiferenciadoV.Value = 1
    chkDiferenciadoV.Enabled = True

Elseif cmbViernes.Text = "Jornada Diferenciada" And Mid(rs!strJornadaV, intPos + 1,
    Len(rs!strJornadaV)) = 0 Then
    chkDiferenciadoV.Enabled = True

End If
```

Día Sábado:

```
intPos = InStrRev(rs!strJornadaS, " ", Len(rs!strJornadaS))
cmbSabado.Text = Mid(rs!strJornadaS, 1, intPos - 1)

If Mid(rs!strJornadaS, intPos + 1, Len(rs!strJornadaS)) = 1 Then
    chkDiferenciadoS.Value = 1
    chkDiferenciadoS.Enabled = True

Elseif cmbSabado.Text = "Jornada Diferenciada" And Mid(rs!strJornadaS, intPos + 1,
    Len(rs!strJornadaS)) = 0 Then
    chkDiferenciadoS.Enabled = True

End If
```

```
rs.MoveNext
```

```
Wend
```

```
Set rsHor = rs
rsHor.MoveFirst
```

```
blnVariable = True
```

```
Exit Sub
```

```
sigue:
    MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

Código cargado en la forma fmrHorarios

Esta forma hace uso de los siguientes módulos: clsHorario y clsHorarioDat.

```
Dim IstLista As ListBox
Dim IstLista2 As ListBox
Dim cmbCombo1 As ComboBox
```

```
Dim objHorario As New clsHorarios
```

'Subrutina del botón aceptar de la pantalla Horarios.

Private Sub btnAceptar_Click()

'Llama a la función "validar" y evalúa si es verdadero o falso. Si es verdadero, todos los datos han sido seleccionados o marcados, caso contrario falta información por seleccionar o ingresar.

If validar = False Then

MsgBox "Ingrese la información faltante.", vbInformation, Me.Caption

Exit Sub

End If

'Si la acción es "Modificar" llama a la función "eliminarHorario". Cuando se modifica los elementos de las Tablas de la Base de Datos del SCA es mejor borrar los antiguos y realizar una nueva operación de ingreso aunque no se modifique ninguno.

If strAccion = "Modificar" Then

eliminarHorario

End If

'Llama a la subrutina "ingresaHorario" y cierra las formas. Todo esto se ejecuta cuando todos los datos están ingresados y validados.

IngresarHorario
frmAdmHorarios.Show
Unload frmDefiHorarios
Unload Me

End Sub

'Función "validar" la cual llama a la función "valida" para cada día de la semana del Horario escogido, se asegura que todos los datos estén ingresados o seleccionados según el tipo de jornada escogido por día.

Private Function validar() As Boolean

On Error GoTo sigue

If valida(lstEntrada1L, LstSalida1L, lstEntrada2L, LstSalida2L,
frmDefiHorarios.cmbLunes, frmDefiHorarios.chkDiferenciadoL) = True And
_
valida(LstEntrada1M, LstSalida1M, lstEntrada2M, LstSalida2M,
frmDefiHorarios.cmbMartes, frmDefiHorarios.chkDiferenciadoM) = True
And _
valida(LstEntrada1Mr, LstSalida1Mr, lstEntrada2Mr, LstSalida2Mr,
frmDefiHorarios.cmbMiercoles, frmDefiHorarios.chkDiferenciadoMr) = True
And _

```

valida(LstEntrada1J, LstSalida1J, IstEntrada2J, LstSalida2J,
frmDefiHorarios.cmbJueves, frmDefiHorarios.chkDiferenciadoJ) = True
And _
valida(LstEntrada1V, LstSalida1V, IstEntrada2V, LstSalida2V,
frmDefiHorarios.cmbViernes, frmDefiHorarios.chkDiferenciadoV) = True
And _
valida(LstEntrada1S, LstSalida1S, IstEntrada2S, LstSalida2S,
frmDefiHorarios.cmbSabado, frmDefiHorarios.chkDiferenciadoS) = True
Then

```

```

validar = True

```

```

Else

```

```

validar = False

```

```

End If

```

```

Exit Function

```

```

sigue:

```

```

MsgBox Err.Description, vbCritical, Err.Source

```

```

End Function

```

'Función "valida" la cual chequea que el tipo de jornada y las horas ingresadas concuerden con el día escogido para el Horario, devuelve verdadero si todos están seleccionados y falso si al menos falta uno.

```

Private Function valida(IstEntrada1 As Object, IstSalida1 As Object, IstEntrada2 As Object,
IstSalida2 As Object, cmbDia As Object, chkCheck As Object) As Boolean

```

```

On Error GoTo sigue

```

```

Dim intl As Integer

```

```

Dim blnEn1J As Boolean

```

```

'Chequea el tipo de jornada escogida para ese día.

```

```

'¿Es jornada única?, entonces chequear que la entrada1 esté marcada.

```

```

If cmbDia.Text = "Jornada Única" Then

```

```

For intl = 0 To IstEntrada1.ListCount - 1

```

```

If IstEntrada1.Selected(intl) = True Then

```

```

blnEn1J = True

```

```

Exit For

```

```

End If

```

```

Next

```

```

End If

```



```
If cmbDia.Text = "Jornada Diferenciada" Then
```

```
Dim blnEn1 As Boolean  
Dim blnSal1 As Boolean  
Dim blnEn2 As Boolean  
Dim blnSal2 As Boolean
```

'¿Es jornada diferenciada doble?, entonces chequear que la entrada1, salida1, entrada2 y salida2 estén marcadas.

```
If chkCheck.Value = 1 Then
```

```
For intl = 0 To lstEntrada1.ListCount - 1
```

```
    If lstEntrada1.Selected(intl) = True Then
```

```
        blnEn1 = True  
        Exit For
```

```
    End If
```

```
Next
```

```
For intl = 0 To lstSalida1.ListCount - 1
```

```
    If lstSalida1.Selected(intl) = True Then
```

```
        blnSal1 = True  
        Exit For
```

```
    End If
```

```
Next
```

```
For intl = 0 To lstEntrada2.ListCount - 1
```

```
    If lstEntrada2.Selected(intl) = True Then
```

```
        blnEn2 = True  
        Exit For
```

```
    End If
```

```
Next
```

```
For intl = 0 To lstSalida2.ListCount - 1
```

```
    If lstSalida2.Selected(intl) = True Then
```

```
        blnSal2 = True  
        Exit For
```

```
    End If
```

```
Next
```

'¿Es jornada diferenciada simple?, chequear que entrada1 y salida2 estén marcadas.

```

Elseif chkCheck.Value = 0 Then

    Dim blnEn10 As Boolean
    Dim blnSal20 As Boolean

    For intl = 0 To lstEntrada1.ListCount - 1

        If    lstEntrada1.Selected(intl) = True Then

            blnEn10 = True
            Exit For

        End If

    Next

    For intl = 0 To lstSalida2.ListCount - 1

        If    lstSalida2.Selected(intl) = True Then

            blnSal20 = True
            Exit For

        End If

    Next

End If

End If

Dim blnNo As Boolean

'Si es "No ingresar horario", no chequee.

If    cmbDia.Text = "No ingresar horario" Then

    blnNo = True

End If

'Chequea si todos los valores que debía seleccionar para ese día estén marcados.

If    (blnEn1 = True And blnSal1 = True And blnEn2 = True And blnSal2 = True) Or _
      (blnEn10 = True And blnSal20 = True) Or blnEn1J = True Or blnNo = True Then

    valida = True

Else

    valida = False

End If

Exit Function

sigue:
MsgBox Err.Description, vbCritical, Err.Source

End Function

```

'Función que coloca "(vacío)" cuando no hay hora para ingresar. En todos los horarios se tiene cuatro variables a ser ingresadas: entrada1, salida1, entrada2 y salida2, si es jornada única no se ingresará salida1 y entrada2, el programa por defecto coloca "NULL" en los datos no ingresados, es mejor cambiar "NULL" por "(vacío)" para un mejor manejo de los datos.

Private Function devuelveValor(IstLista As Object) As String

```

If IstLista.ListCount = 0 Then
    devuelveValor = ""
Else
    devuelveValor = IstLista.List(IstLista.ListIndex)
End If

```

End Function

'Función que borra un horario para su modificación.

Private Sub eliminarHorario()

```

On Error GoTo sigue
    objHorario.elimnarHorario (strCodigoHorario)
Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

```

End Sub

'Función "ingresaHorario", prepara los datos para ser ingresados invocando a la función "devuelveValor" para que no se ingresen datos "NULL"

Private Sub ingresaHorario()

On Error GoTo sigue

```

objHorario.ingresarHorario _
    frmDefiHorarios.txtHorario.Text, frmDefiHorarios.txtDescripcion.Text, _
    devuelveValor(IstEntrada1L), devuelveValor(LstSalida1L), _
    devuelveValor(IstEntrada2L), devuelveValor(LstSalida2L), _
    devuelveValor(LstEntrada1M), devuelveValor(LstSalida1M), _
    devuelveValor(IstEntrada2M), devuelveValor(LstSalida2M), _
    devuelveValor(LstEntrada1Mr), devuelveValor(LstSalida1Mr), _
    devuelveValor(LstEntrada2Mr), devuelveValor(LstSalida2Mr), _
    devuelveValor(LstEntrada1J), devuelveValor(LstSalida1J), _
    devuelveValor(LstEntrada2J), devuelveValor(LstSalida2J), _
    devuelveValor(LstEntrada1V), devuelveValor(LstSalida1V), _
    devuelveValor(LstEntrada2V), devuelveValor(LstSalida2V), _
    devuelveValor(LstEntrada1S), devuelveValor(LstSalida1S), _
    devuelveValor(LstEntrada2S), devuelveValor(LstSalida2S), _

```

```

frmDefiHorarios.cmbLunes.Text & " " & frmDefiHorarios.chkDiferenciadoL.Value, _
frmDefiHorarios.cmbMartes.Text & " " & frmDefiHorarios.chkDiferenciadoM.Value, _
frmDefiHorarios.cmbMiercoles.Text & " " & frmDefiHorarios.chkDiferenciadoMr.Value, _
frmDefiHorarios.cmbJueves.Text & " " & frmDefiHorarios.chkDiferenciadoJ.Value, _
frmDefiHorarios.cmbViernes.Text & " " & frmDefiHorarios.chkDiferenciadoV.Value, _
frmDefiHorarios.cmbSabado.Text & " " & frmDefiHorarios.chkDiferenciadoS.Value

```

Exit Sub

sigue:

```
MsgBox Err.Description, vbCritical, Err.Source
```

End Sub

'Botón cancelar.

Private Sub btnCancelar_Click()

Unload Me

End Sub

'Función que llena las listas de los combos de manera consecutiva para que no se ingresen horas incoherentes, llena las horas en pasos de 30 minutos desde el valor del combo anterior, como es consecutivo si se altera la hora de un combo anterior, la función borrará las horas posteriores para ingresarlas de nuevo tomando como referencia la nueva hora del combo anterior; esto depende de las listas seteadas, utiliza las Subrutinas "LstEntrada1", "LstSalida1" o "LstEntrada2" dependiendo desde donde se debe reingresar los datos de manera consecutiva. Esto es para cada día que se desee ingresar.

Private Sub llenaListaEntrada1(intLimite As Integer)

On Error GoTo sigue

Dim intHora As Integer

Dim intMin As Integer

Dim intPos As Integer

Dim strMin As String

If lstLista.Selected(lstLista.ListIndex) = True Then

intPos = InStr(lstLista.List(lstLista.ListIndex), ":")

intHora = Mid(lstLista.List(lstLista.ListIndex), 1, intPos - 1)

intMin = Mid(lstLista.List(lstLista.ListIndex), intPos + 1, 5)

intHora = intHora + 1

If intMin = 0 Then

strMin = "00"

Else

```
        strMin = intMin
    End If
    While intHora < intLimite
        If Len(CStr(intHora)) = 1 Then
            lstLista2.AddItem "0" & intHora & ":" & strMin
        Else
            lstLista2.AddItem intHora & ":" & strMin
        End If
        intMin = intMin + 30
        If intMin = 60 Then
            intMin = 0
            intHora = intHora + 1
            strMin = "00"
        Else
            strMin = intMin
        End If
    Wend
    If Len(CStr(intLimite)) = 1 Then
        lstLista2.AddItem "0" & intLimite & ":00"
    Else
        lstLista2.AddItem intLimite & ":00"
    End If
End If
Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source
End Sub
```

'Subrutina "habilitar" que habilita los combos de las horas dependiendo del tipo de jornada.

Private Sub habilitar()

On Error GoTo sigue

 'Chequea si el día Lunes tiene jornada diferenciada doble.

```
If      frmDefiHorarios.cmbLunes.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoL.Value = 1 Then

    LstSalida1L.Enabled = True
    LstSalida1L.BackColor = vbWhite
    lstEntrada2L.Enabled = True
    lstEntrada2L.BackColor = vbWhite
    LstSalida2L.Enabled = True
    LstSalida2L.BackColor = vbWhite

End If

'Chequea si el día Martes tiene jornada diferenciada doble.

If      frmDefiHorarios.cmbMartes.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoM.Value = 1 Then

    LstSalida1M.Enabled = True
    LstSalida1M.BackColor = vbWhite
    lstEntrada2M.Enabled = True
    lstEntrada2M.BackColor = vbWhite
    LstSalida2M.Enabled = True
    LstSalida2M.BackColor = vbWhite

End If

'Chequea si el día Miércoles tiene jornada diferenciada doble.

If      frmDefiHorarios.cmbMiercoles.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoMr.Value = 1 Then

    LstSalida1Mr.Enabled = True
    LstSalida1Mr.BackColor = vbWhite
    lstEntrada2Mr.Enabled = True
    lstEntrada2Mr.BackColor = vbWhite
    LstSalida2Mr.Enabled = True
    LstSalida2Mr.BackColor = vbWhite

End If

'Chequea si el día Jueves tiene jornada diferenciada doble.

If      frmDefiHorarios.cmbJueves.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoJ.Value = 1 Then

    LstSalida1J.Enabled = True
    LstSalida1J.BackColor = vbWhite
    lstEntrada2J.Enabled = True
    lstEntrada2J.BackColor = vbWhite
    LstSalida2J.Enabled = True
    LstSalida2J.BackColor = vbWhite

End If

'Chequea si el día Viernes tiene jornada diferenciada doble.

If      frmDefiHorarios.cmbViernes.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoV.Value = 1 Then
```

```
LstSalida1V.Enabled = True
LstSalida1V.BackColor = vbWhite
lstEntrada2V.Enabled = True
lstEntrada2V.BackColor = vbWhite
LstSalida2V.Enabled = True
LstSalida2V.BackColor = vbWhite

End If

'Chequea si el día Sábado tiene jornada diferenciada doble.

If frmDefiHorarios.cmbSabado.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoS.Value = 1 Then

LstSalida1S.Enabled = True
LstSalida1S.BackColor = vbWhite
lstEntrada2S.Enabled = True
lstEntrada2S.BackColor = vbWhite
LstSalida2S.Enabled = True
LstSalida2S.BackColor = vbWhite

End If

'Chequea si el día Lunes tiene jornada diferenciada.

If frmDefiHorarios.cmbLunes.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoL.Value = 0 Then

LstSalida2L.Enabled = True
LstSalida2L.BackColor = vbWhite

End If

'Chequea si el día Martes tiene jornada diferenciada.

If frmDefiHorarios.cmbMartes.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoM.Value = 0 Then

LstSalida2M.Enabled = True
LstSalida2M.BackColor = vbWhite

End If

'Chequea si el día Miércoles tiene jornada diferenciada.
If frmDefiHorarios.cmbMiercoles.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoMr.Value = 0 Then

LstSalida2Mr.Enabled = True
LstSalida2Mr.BackColor = vbWhite

End If

'Chequea si el día Jueves tiene jornada diferenciada.

If frmDefiHorarios.cmbJueves.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoJ.Value = 0 Then

LstSalida2J.Enabled = True
```

```
LstSalida2J.BackColor = vbWhite

End If

'Chequea si el día Viernes tiene jornada diferenciada.

If frmDefiHorarios.cmbViernes.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoV.Value = 0 Then

    LstSalida2V.Enabled = True
    LstSalida2V.BackColor = vbWhite

End If

'Chequea si el día Sábado tiene jornada diferenciada.

If frmDefiHorarios.cmbSabado.Text = "Jornada Diferenciada" And _
frmDefiHorarios.chkDiferenciadoS.Value = 0 Then

    LstSalida2S.Enabled = True
    LstSalida2S.BackColor = vbWhite

End If

'Chequea si el día Lunes tiene la opción "No ingresar horario".

If frmDefiHorarios.cmbLunes.Text <> "No ingresar horario" Then

    lstEntrada1L.Enabled = True
    lstEntrada1L.BackColor = vbWhite
    Set cmbCombo1 = frmDefiHorarios.cmbLunes
    Set lstLista = lstEntrada1L
    llenarEntrada1

End If

'Chequea si el día Martes tiene la opción "No ingresar horario".

If frmDefiHorarios.cmbMartes.Text <> "No ingresar horario" Then

    LstEntrada1M.Enabled = True
    LstEntrada1M.BackColor = vbWhite
    Set cmbCombo1 = frmDefiHorarios.cmbMartes
    Set lstLista = LstEntrada1M
    llenarEntrada1

End If

'Chequea si el día Miércoles tiene la opción "No ingresar horario".

If frmDefiHorarios.cmbMiercoles.Text <> "No ingresar horario" Then

    LstEntrada1Mr.Enabled = True
    LstEntrada1Mr.BackColor = vbWhite
    Set cmbCombo1 = frmDefiHorarios.cmbMiercoles
    Set lstLista = LstEntrada1Mr
    llenarEntrada1

End If
```


'Chequea si el día Jueves tiene la opción "No ingresar horario".

```
If      frmDefiHorarios.cmbJueves.Text <> "No ingresar horario" Then

    LstEntrada1J.Enabled = True
    LstEntrada1J.BackColor = vbWhite
    Set cmbCombo1 = frmDefiHorarios.cmbJueves
    Set lstLista = LstEntrada1J
    llenarEntrada1
```

End If

'Chequea si el día Viernes tiene la opción "No ingresar horario".

```
If      frmDefiHorarios.cmbViernes.Text <> "No ingresar horario" Then

    LstEntrada1V.Enabled = True
    LstEntrada1V.BackColor = vbWhite
    Set cmbCombo1 = frmDefiHorarios.cmbViernes
    Set lstLista = LstEntrada1V
    llenarEntrada1
```

End If

'Chequea si el día Sábado tiene la opción "No ingresar horario".

```
If      frmDefiHorarios.cmbSabado.Text <> "No ingresar horario" Then

    LstEntrada1S.Enabled = True
    LstEntrada1S.BackColor = vbWhite
    Set cmbCombo1 = frmDefiHorarios.cmbSabado
    Set lstLista = LstEntrada1S
    llenarEntrada1
```

End If

Exit Sub

sigue:

```
MsgBox Err.Description, vbCritical, Err.Source
```

End Sub

'Subrutina "llenarEntrada1" que llena las listas de los combos de la primera entrada de un día dependiendo si es jornada única o diferenciada sea doble o no.

Private Sub llenarEntrada1()

On Error GoTo sigue

```
If      cmbCombo1.Text = "Jornada Única" Then

    lstLista.AddItem "07:00"
    lstLista.AddItem "07:30"
    lstLista.AddItem "08:00"
    lstLista.AddItem "08:30"
    lstLista.AddItem "09:00"
    lstLista.AddItem "09:30"
```

```

IstLista.AddItem "10:00"
IstLista.AddItem "10:30"
IstLista.AddItem "11:00"
IstLista.AddItem "11:30"

```

Else

```

IstLista.AddItem "07:00"
IstLista.AddItem "07:30"
IstLista.AddItem "08:00"
IstLista.AddItem "08:30"
IstLista.AddItem "09:00"
IstLista.AddItem "09:30"
IstLista.AddItem "10:00"
IstLista.AddItem "10:30"
IstLista.AddItem "11:00"
IstLista.AddItem "11:30"
IstLista.AddItem "12:00"
IstLista.AddItem "12:30"
IstLista.AddItem "13:00"
IstLista.AddItem "13:30"
IstLista.AddItem "14:00"
IstLista.AddItem "14:30"
IstLista.AddItem "15:00"
IstLista.AddItem "15:30"
IstLista.AddItem "16:00"
IstLista.AddItem "16:30"
IstLista.AddItem "17:00"

```

End If

Exit Sub

sigue:

```
MsgBox Err.Description, vbCritical, Err.Source
```

End Sub

'Subrutina que llama a la función "llenaPlan" cuando se escoge la opción "Modificar"

Private Sub Form_Load()

 habilitar , Llama a la subrutina "habilitar"

 If strAccion = "Modificar" And binVariable = True Then

 llenaPlan

 End If

End Sub

'Subrutina que calcula la hora de salida cuando se escoge la jornada única, suma ocho horas y media

Private Function calculaJUnica() As Boolean

 On Error GoTo sigue

```

Dim blnCalc As Boolean
Dim intLimite As Integer

If IstEntrada1L.Selected(IstEntrada1L.ListIndex) = True Then
    LstSalida1L.Clear
    LstSalida2L.Clear
    IstEntrada2L.Clear
    Set IstLista = IstEntrada1L
    Set cmbCombo1 = frmDefiHorarios.cmbLunes

    If frmDefiHorarios.cmbLunes.Text = "Jornada Única" Or _
        (frmDefiHorarios.cmbLunes.Text = "Jornada Diferenciada" And _
        frmDefiHorarios.chkDiferenciadoL.Value = 0) Then

        Set IstLista2 = LstSalida2L
        Set IstLista = IstEntrada1L
        blnCalc = calculaJUnica
        intLimite = 20

    ElseIf frmDefiHorarios.cmbLunes.Text = "Jornada Diferenciada" And _
        frmDefiHorarios.chkDiferenciadoL.Value = 1 Then

        Set IstLista2 = LstSalida1L
        intLimite = 18

    End If
End If
If blnCalc = False Then llenalistaEntrada1 (intLimite)

End Sub

```

'Subrutina "LstEntrada1" del día Martes, setea la lista de entrada1 para ser ingresada y borra las posteriores: salida1, entrada2, salida2 para un nuevo ingreso.

```

Private Sub LstEntrada1M_Click()

    Dim blnCalc As Boolean
    Dim intLimite As Integer

    If LstEntrada1M.Selected(LstEntrada1M.ListIndex) = True Then

        LstSalida1M.Clear
        LstSalida2M.Clear
        IstEntrada2M.Clear
        Set IstLista = LstEntrada1M
        Set cmbCombo1 = frmDefiHorarios.cmbMartes

        If frmDefiHorarios.cmbMartes.Text = "Jornada Única" Or _
            (frmDefiHorarios.cmbMartes.Text = "Jornada Diferenciada" And _
            frmDefiHorarios.chkDiferenciadoM.Value = 0) Then

            Set IstLista2 = LstSalida2M
            Set IstLista = LstEntrada1M
            blnCalc = calculaJUnica

```

```

        intLimite = 20

    Elseif frmDefiHorarios.cmbMartes.Text = "Jornada Diferenciada" And _
        frmDefiHorarios.chkDiferenciadoM.Value = 1 Then

        Set lstLista2 = LstSalida1M
        intLimite = 18

    End If
End If

If binCalc = False Then llenalistaEntrada1 (intLimite)

End Sub

```

'Subrutina "LstEntrada1" del día Miércoles, setea la lista de entrada1 para ser ingresada y borra las posteriores: salida1, entrada2, salida2 para un nuevo ingreso.

```

Private Sub LstEntrada1Mr_Click()

    Dim binCalc As Boolean
    Dim intLimite As Integer

    If LstEntrada1Mr.Selected(LstEntrada1Mr.ListIndex) = True Then

        LstSalida1Mr.Clear
        LstSalida2Mr.Clear
        lstEntrada2Mr.Clear
        Set lstLista = LstEntrada1Mr
        Set cmbCombo1 = frmDefiHorarios.cmbMiercoles

        If frmDefiHorarios.cmbMiercoles.Text = "Jornada Única" Or _
            (frmDefiHorarios.cmbMiercoles.Text = "Jornada Diferenciada" And _
            frmDefiHorarios.chkDiferenciadoMr.Value = 0) Then

            Set lstLista2 = LstSalida2Mr
            Set lstLista = LstEntrada1Mr
            binCalc = calculaJUnica
            intLimite = 20

        Elseif frmDefiHorarios.cmbMiercoles.Text = "Jornada Diferenciada" And _
            frmDefiHorarios.chkDiferenciadoMr.Value = 1 Then

            Set lstLista2 = LstSalida1Mr
            intLimite = 18

        End If

    End If

    If binCalc = False Then llenalistaEntrada1 (intLimite)

End Sub

```

'Subrutina "LstEntrada1" del día Jueves, setea la lista de entrada1 para ser ingresada y borra las posteriores: salida1, entrada2, salida2 para un nuevo ingreso.

```

Private Sub LstEntrada1J_Click()

    Dim blnCalc As Boolean
    Dim intLimite As Integer

    If LstEntrada1J.Selected(LstEntrada1J.ListIndex) = True Then

        LstSalida1J.Clear
        LstSalida2J.Clear
        LstEntrada2J.Clear
        Set lstLista = LstEntrada1J
        Set cmbCombo1 = frmDefiHorarios.cmbJueves

        If frmDefiHorarios.cmbJueves.Text = "Jornada Única" Or _
            (frmDefiHorarios.cmbJueves.Text = "Jornada Diferenciada" And _
            frmDefiHorarios.chkDiferenciadoJ.Value = 0) Then

            Set lstLista2 = LstSalida2J
            Set lstLista = LstEntrada1J
            blnCalc = calculaJUnica
            intLimite = 20

        ElseIf frmDefiHorarios.cmbJueves.Text = "Jornada Diferenciada" And _
            frmDefiHorarios.chkDiferenciadoJ.Value = 1 Then

            Set lstLista2 = LstSalida1J
            intLimite = 18

        End If

    End If

    If blnCalc = False Then llenaListaEntrada1 (intLimite)

End Sub

```

'Subrutina "LstEntrada1" del día Viernes, setea la lista de entrada1 para ser ingresada y borra las posteriores: salida1, entrada2, salida2 para un nuevo ingreso

```

Private Sub LstEntrada1V_Click()

    Dim blnCalc As Boolean
    Dim intLimite As Integer

    If LstEntrada1V.Selected(LstEntrada1V.ListIndex) = True Then

        LstSalida1V.Clear
        LstSalida2V.Clear
        LstEntrada2V.Clear
        Set lstLista = LstEntrada1V
        Set cmbCombo1 = frmDefiHorarios.cmbViernes

    End If

End Sub

```

```

If      frmDefiHorarios.cmbViernes.Text = "Jornada Única" Or _
      (frmDefiHorarios.cmbViernes.Text = "Jornada Diferenciada" And _
      frmDefiHorarios.chkDiferenciadoV.Value = 0) Then

      Set lstLista2 = LstSalida2V
      Set lstLista = LstEntrada1V
      blnCalc = calculaJUnica
      intLimite = 20

Elseif  frmDefiHorarios.cmbViernes.Text = "Jornada Diferenciada" And _
      frmDefiHorarios.chkDiferenciadoV.Value = 1 Then

      Set lstLista2 = LstSalida1V
      intLimite = 18

      End If
End If

      If blnCalc = False Then llenaListaEntrada1 (intLimite)

End Sub

```

'Subrutina "LstEntrada1" del día Sábado, setea la lista de entrada1 para ser ingresada y borra las posteriores: salida1, entrada2, salida2 para un nuevo ingreso..

```

Private Sub LstEntrada1S_Click()

      Dim blnCalc As Boolean
      Dim intLimite As Integer

      If      LstEntrada1S.Selected(LstEntrada1S.ListIndex) = True Then

      LstSalida1S.Clear
      LstSalida2S.Clear
      lstEntrada2S.Clear
      Set lstLista = LstEntrada1S
      Set cmbCombo1 = frmDefiHorarios.cmbSabado

      If      frmDefiHorarios.cmbSabado.Text = "Jornada Única" Or _
      (frmDefiHorarios.cmbSabado.Text = "Jornada Diferenciada" And _
      frmDefiHorarios.chkDiferenciadoS.Value = 0) Then

      Set lstLista2 = LstSalida2S
      Set lstLista = LstEntrada1S
      blnCalc = calculaJUnica
      intLimite = 20

      Elseif  frmDefiHorarios.cmbSabado.Text = "Jornada Diferenciada" And _
      frmDefiHorarios.chkDiferenciadoS.Value = 1 Then

      Set lstLista2 = LstSalida1S
      intLimite = 18

      End If
End If

```

```
If blnCalc = False Then llenaListaEntrada1 (intLimite)
```

```
End Sub
```

'Subrutina "LstEntrada2" del día Lunes, setea la lista de entrada2 para ser ingresada y borra la lista salida2 para ser ingresada nuevamente.

```
Private Sub lstEntrada2L_Click()
```

```
    If      lstEntrada2L.Selected(lstEntrada2L.ListIndex) = True Then
```

```
        LstSalida2L.Clear
        Set lstLista = lstEntrada2L
        Set lstLista2 = LstSalida2L
```

```
    End If
```

```
        llenaListaEntrada1 (20)
```

```
End Sub
```

'Subrutina "LstEntrada2" del día Martes, setea la lista de entrada2 para ser ingresada y borra la lista salida2 para ser ingresada nuevamente.

```
Private Sub lstEntrada2M_Click()
```

```
    If      lstEntrada2M.Selected(lstEntrada2M.ListIndex) = True Then
```

```
        LstSalida2M.Clear
        Set lstLista = lstEntrada2M
        Set lstLista2 = LstSalida2M
```

```
    End If
```

```
        llenaListaEntrada1 (20)
```

```
End Sub
```

'Subrutina "LstEntrada2" del día Miércoles, setea la lista de entrada2 para ser ingresada y borra la lista salida2 para ser ingresada nuevamente.

```
Private Sub lstEntrada2Mr_Click()
```

```
    If      lstEntrada2Mr.Selected(lstEntrada2Mr.ListIndex) = True Then
```

```
        LstSalida2Mr.Clear
        Set lstLista = lstEntrada2Mr
        Set lstLista2 = LstSalida2Mr
```

```
    End If
```

llenaListaEntrada1 (20)

End Sub

'Subrutina "LstEntrada1" del día Jueves, setea la lista de entrada1 para ser ingresada y borra las posteriores: salida1, entrada2, salida2 para un nuevo ingreso

Private Sub lstEntrada2J_Click()

If lstEntrada2J.Selected(lstEntrada2J.ListIndex) = True Then

LstSalida2J.Clear
Set lstLista = lstEntrada2J
Set lstLista2 = LstSalida2J

End If

llenaListaEntrada1 (20)

End Sub

'Subrutina "LstEntrada2" del día Viernes, setea la lista de entrada2 para ser ingresada y borra la lista salida2 para ser ingresada nuevamente.

Private Sub lstEntrada2V_Click()

If lstEntrada2V.Selected(lstEntrada2V.ListIndex) = True Then

LstSalida2V.Clear
Set lstLista = lstEntrada2V
Set lstLista2 = LstSalida2V

End If

llenaListaEntrada1 (20)

End Sub

'Subrutina "LstEntrada2" del día Sábado, setea la lista de entrada2 para ser ingresada y borra la lista salida2 para ser ingresada nuevamente.

Private Sub lstEntrada2S_Click()

If lstEntrada2S.Selected(lstEntrada2S.ListIndex) = True Then

LstSalida2S.Clear
Set lstLista = lstEntrada2S
Set lstLista2 = LstSalida2S

End If

llenaListaEntrada1 (20)

End Sub

'Subrutina "LstSalida1" del día Lunes, setea la lista de la salida1 para ser ingresada y borra la lista entrada2 y salida2 para ser ingresadas nuevamente.

Private Sub LstSalida1L_Click()

```

    If LstSalida1L.Selected(LstSalida1L.ListIndex) = True Then

        IstEntrada2L.Clear
        LstSalida2L.Clear
        Set IstLista = LstSalida1L

        If frmDefiHorarios.cmbLunes.Text = "Jornada Diferenciada" And _
           frmDefiHorarios.chkDiferenciadoL.Value = 1 Then

            Set IstLista2 = IstEntrada2L

        End If
    End If

    llenaListaEntrada1 (19)

```

End Sub

'Subrutina "LstSalida1" del día Martes, setea la lista de la salida1 para ser ingresada y borra la lista entrada2 y salida2 para ser ingresadas nuevamente.

Private Sub LstSalida1M_Click()

```

    If LstSalida1M.Selected(LstSalida1M.ListIndex) = True Then

        IstEntrada2M.Clear
        LstSalida2M.Clear
        Set IstLista = LstSalida1M

        If frmDefiHorarios.cmbMartes.Text = "Jornada Diferenciada" And _
           frmDefiHorarios.chkDiferenciadoM.Value = 1 Then

            Set IstLista2 = IstEntrada2M

        End If
    End If

    llenaListaEntrada1 (19)

```

End Sub

'Subrutina "LstSalida1" del día Miércoles, setea la lista de la salida1 para ser ingresada y borra la lista entrada2 y salida2 para ser ingresadas nuevamente.

Private Sub LstSalida1Mr_Click()

```

    If LstSalida1Mr.Selected(LstSalida1Mr.ListIndex) = True Then

```

```

    IstEntrada2Mr.Clear
    LstSalida2Mr.Clear
    Set IstLista = LstSalida1Mr

    If      frmDefiHorarios.cmbMiercoles.Text = "Jornada Diferenciada" And _
           frmDefiHorarios.chkDiferenciadoMr.Value = 1 Then

        Set IstLista2 = IstEntrada2Mr

    End If
End If

    llenaListaEntrada1 (19)

End Sub

```

'Subrutina "LstSalida1" del día Jueves, setea la lista de la salida1 para ser ingresada y borra la lista entrada2 y salida2 para ser ingresadas nuevamente.

```

Private Sub LstSalida1J_Click()

    If      LstSalida1J.Selected(LstSalida1J.ListIndex) = True Then

        IstEntrada2J.Clear
        LstSalida2J.Clear
        Set IstLista = LstSalida1J

        If      frmDefiHorarios.cmbJueves.Text = "Jornada Diferenciada" And _
               frmDefiHorarios.chkDiferenciadoJ.Value = 1 Then

            Set IstLista2 = IstEntrada2J

        End If

    End If

    llenaListaEntrada1 (19)

End Sub

```

'Subrutina "LstSalida1" del día Viernes, setea la lista de la salida1 para ser ingresada y borra la lista entrada2 y salida2 para ser ingresadas nuevamente.

```

Private Sub LstSalida1V_Click()

    If      LstSalida1V.Selected(LstSalida1V.ListIndex) = True Then

        IstEntrada2V.Clear
        LstSalida2V.Clear
        Set IstLista = LstSalida1V

        If      frmDefiHorarios.cmbViernes.Text = "Jornada Diferenciada" And _
               frmDefiHorarios.chkDiferenciadoV.Value = 1 Then

            Set IstLista2 = IstEntrada2V

        End If

    End If

End Sub

```

```

    End If
End If

    llenaListaEntrada1 (19)

```

```
End Sub
```

'Subrutina "LstSalida1" del día Sábado, setea la lista de la salida1 para ser ingresada y borra la lista entrada2 y salida2 para ser ingresadas nuevamente.

```
Private Sub LstSalida1S_Click()
```

```

    If    LstSalida1S.Selected(LstSalida1S.ListIndex) = True Then

        IstEntrada2S.Clear
        LstSalida2S.Clear
        Set IstLista = LstSalida1S

        If    frmDefiHorarios.cmbSabado.Text = "Jornada Diferenciada" And _
            frmDefiHorarios.chkDiferenciadoS.Value = 1 Then

            Set IstLista2 = IstEntrada2S

        End If
    End If

    llenaListaEntrada1 (19)

```

```
End Sub
```

'Subrutina "llenaPlan" la cual se invoca cuando se quiere modificar un horario. Extrae los datos almacenado en la Base de Datos del SCA del horario escogido.

```
Private Sub llenaPlan()
```

```

    On Error GoTo sigue

    rsHor.MoveFirst

    While Not rsHor.EOF

        'Lunes

        If    rsHor!strEntrada1L <> "" Then

            IstEntrada1L.AddItem rsHor!strEntrada1L, 0
            IstEntrada1L.Selected(0) = True

        End If

        If    rsHor!strSalida1L <> "" Then

            LstSalida1L.AddItem rsHor!strSalida1L, 0
            LstSalida1L.Selected(0) = True

        End If
    End If

```

```
If      rsHor!strEntrada2L <> "" Then
    lstEntrada2L.AddItem rsHor!strEntrada2L, 0
    lstEntrada2L.Selected(0) = True
End If

If      rsHor!strSalida2L <> "" Then
    LstSalida2L.AddItem rsHor!strSalida2L, 0
    LstSalida2L.Selected(0) = True
End If

'Martes

If      rsHor!strEntrada1M <> "" Then
    LstEntrada1M.AddItem rsHor!strEntrada1M, 0
    LstEntrada1M.Selected(0) = True
End If

If      rsHor!strSalida1M <> "" Then
    LstSalida1M.AddItem rsHor!strSalida1M, 0
    LstSalida1M.Selected(0) = True
End If

If      rsHor!strEntrada2M <> "" Then
    lstEntrada2M.AddItem rsHor!strEntrada2M, 0
    lstEntrada2M.Selected(0) = True
End If

If      rsHor!strSalida2M <> "" Then
    LstSalida2M.AddItem rsHor!strSalida2M, 0
    LstSalida2M.Selected(0) = True
End If

'Miércoles

If      rsHor!strEntrada1Mr <> "" Then
    LstEntrada1Mr.AddItem rsHor!strEntrada1Mr, 0
    LstEntrada1Mr.Selected(0) = True
End If

If      rsHor!strSalida1Mr <> "" Then
    LstSalida1Mr.AddItem rsHor!strSalida1Mr, 0
    LstSalida1Mr.Selected(0) = True
```

```
End If

If rsHor!strEntrada2Mr <> "" Then
    LstEntrada2Mr.AddItem rsHor!strEntrada2Mr, 0
    LstEntrada2Mr.Selected(0) = True
End If

If rsHor!strSalida2Mr <> "" Then
    LstSalida2Mr.AddItem rsHor!strSalida2Mr, 0
    LstSalida2Mr.Selected(0) = True
End If

'Jueves

If rsHor!strEntrada1J <> "" Then
    LstEntrada1J.AddItem rsHor!strEntrada1J, 0
    LstEntrada1J.Selected(0) = True
End If

If rsHor!strSalida1J <> "" Then
    LstSalida1J.AddItem rsHor!strSalida1J, 0
    LstSalida1J.Selected(0) = True
End If

If rsHor!strEntrada2J <> "" Then
    LstEntrada2J.AddItem rsHor!strEntrada2J, 0
    LstEntrada2J.Selected(0) = True
End If

If rsHor!strSalida2J <> "" Then
    LstSalida2J.AddItem rsHor!strSalida2J, 0
    LstSalida2J.Selected(0) = True
End If

'Viernes

If rsHor!strEntrada1V <> "" Then
    LstEntrada1V.AddItem rsHor!strEntrada1V, 0
    LstEntrada1V.Selected(0) = True
End If

If rsHor!strSalida1V <> "" Then
    LstSalida1V.AddItem rsHor!strSalida1V, 0
    LstSalida1V.Selected(0) = True
End If
```

```
If rsHor!strEntrada2V <> "" Then
    lstEntrada2V.AddItem rsHor!strEntrada2V, 0
    lstEntrada2V.Selected(0) = True
End If

If rsHor!strSalida2V <> "" Then
    LstSalida2V.AddItem rsHor!strSalida2J, 0
    LstSalida2V.Selected(0) = True
End If

'Sábado

If rsHor!strEntrada1S <> "" Then
    LstEntrada1S.AddItem rsHor!strEntrada1S, 0
    LstEntrada1S.Selected(0) = True
End If

If rsHor!strSalida1S <> "" Then
    LstSalida1S.AddItem rsHor!strSalida1S, 0
    LstSalida1S.Selected(0) = True
End If

If rsHor!strEntrada2S <> "" Then
    lstEntrada2S.AddItem rsHor!strEntrada2S, 0
    lstEntrada2S.Selected(0) = True
End If

If rsHor!strSalida2S <> "" Then
    LstSalida2S.AddItem rsHor!strSalida2S, 0
    LstSalida2S.Selected(0) = True
End If

rsHor.MoveNext

Wend

If rsHor.State = adStateOpen And blnVariable = False Then rsHor.Close

Exit Sub
    sigue:
        MsgBox Err.Description, vbCritical, Err.Source
End Sub
```

Código cargado en la forma fmrJustificaciones

Esta forma hace uso de los siguientes módulos: Módulo1, clsEmpleados, clsEmpleadosDat, clsHorario y clsHorarioDat.

'Declaración de Variables.

```
Dim rs As New ADODB.Recordset
Dim objEmpleado As New clsEmpleados
```

'Botón Salir.

```
Private Sub btnSalir_Click()
```

```
    Unload Me
```

```
End Sub
```

'Subrutina "llenaGrilla", la cual llena una grilla con todos los días de atraso del empleado entre la fechas escogidas junto con los minutos atrasados y justificados, si los hay, respectivamente.

```
Private Sub llenaGrilla()
```

```
    On Error GoTo sigue
```

```
    Dim intContadorFilas As Integer
    Dim intSum As Integer
```

```
    grdJustificar.Col = -1
    grdJustificar.Row = -1
    grdJustificar.Lock = True
    intContadorFilas = 1
```

```
    Set rs = objEmpleado.justificaciones(strCodigoEmpleado, strFechaInicio, strFechaFin)
```

```
    While Not rs.EOF
```

```
        grdJustificar.Col = 1
        grdJustificar.MaxRows = intContadorFilas
        grdJustificar.Row = intContadorFilas
        grdJustificar.Text = rs!dtDia
        grdJustificar.Col = 2
        grdJustificar.Text = rs!jornada
        grdJustificar.Col = 3
        grdJustificar.Text = rs!AEntJ1
        grdJustificar.Col = 4
        grdJustificar.Text = rs!ASalJ1
        grdJustificar.Col = 5
        grdJustificar.Text = rs!AEntJ2
        grdJustificar.Col = 6
        grdJustificar.Text = rs!ASalJ2
        grdJustificar.Col = 8
        grdJustificar.Text = rs!JE1
        grdJustificar.Col = 9
```

```

    grdJustificar.Text = rs!JS1
    grdJustificar.Col = 10
    grdJustificar.Text = rs!JE2
    grdJustificar.Col = 11
    grdJustificar.Text = rs!JS2
    grdJustificar.Col = 12
    grdJustificar.Text = rs!strMotivo1
    grdJustificar.Col = 13
    grdJustificar.Text = rs!strMotivo2

    intSum = rs!JE1 + rs!JS1 + rs!JE2 + rs!JS2

    If    intSum > 0 Then

        'Cambia el nombre del botón de "Justificar" a "Modificar" ya que existen minutos
        justificados anteriormente.

            grdJustificar.Col = 7
            grdJustificar.TypeButtonText = "Modificar"

        Else

            grdJustificar.Col = 7
            grdJustificar.TypeButtonText = "Justificar"

        End If

        intContadorFilas = intContadorFilas + 1
        rs.MoveNext

    Wend

    rs.Close

    Exit Sub
    sigue:
        MsgBox Err.Description, vbCritical, Err.Source

End Sub

```

'Subrutina que habilita únicamente la columna con los botones "Justificar" de la grilla de justificaciones, de esa manera se evita que puedan escribir o borrar los demás datos en las otras columnas.

```

Private Sub Form_Load()

    txtCodigo.Text = strCodigoEmpleado
    txtEmpleado.Text = strNombreEmpleado
    lblDe.Caption = lblDe.Caption & " " & strFechaInicio
    lblHasta.Caption = lblHasta.Caption & " " & strFechaFin

    llenaGrilla

    grdJustificar.Row = -1
    grdJustificar.Col = 7
    grdJustificar.Lock = False

```

End Sub

'Subrutina del botón "Justificar", evalúa el tipo de jornada que se quiere justificar para visulaizar una de las dos formas: frmJustJorUnica o frmJustJorDoble dependiendo de la jornada.

```
Private Sub grdJustificar_ButtonClicked (ByVal Col As Long, ByVal Row As Long, ByVal ButtonDown As Integer)
```

```
    grdJustificar.Col = Col  
    grdJustificar.Row = Row
```

```
    grdJustificar.Col = 1  
    strFechaJusti = grdJustificar.Text  
    grdJustificar.Col = 3  
    intJusEntrada1 = grdJustificar.Text  
    grdJustificar.Col = 4  
    intJusSalida1 = grdJustificar.Text  
    grdJustificar.Col = 5  
    intJusEntrada2 = grdJustificar.Text  
    grdJustificar.Col = 6  
    intJusSalida2 = grdJustificar.Text
```

```
    grdJustificar.Col = 8  
    intJusEnJust1 = grdJustificar.Text  
    grdJustificar.Col = 9  
    intJusSaJust1 = grdJustificar.Text  
    grdJustificar.Col = 10  
    intJusEnJust2 = grdJustificar.Text  
    grdJustificar.Col = 11  
    intJusSaJust2 = grdJustificar.Text  
    grdJustificar.Col = 12  
    strMotivo1 = grdJustificar.Text  
    grdJustificar.Col = 13  
    strMotivo2 = grdJustificar.Text
```

```
    grdJustificar.Col = 2  
    strJornadaEmp = grdJustificar.Text
```

```
    If    grdJustificar.Text = "Doble" Then  
        frmJustJorDoble.Show
```

```
    Else
```

```
        frmJustJorUnica.Show
```

```
    End If
```

```
    Unload Me
```

```
End Sub
```

Código cargado en la forma fmrJustJorUnica

Esta forma hace uso de los siguientes módulos: Módulo1, clsEmpleados, y clsEmpleadosDat.

'Declaración de variables.

```
Dim objEmpleados As New clsEmpleados
```

'Botón Aceptar, chequea si los datos pasaron las validaciones y de ser así ingresa los datos a la Base del SCA.

```
Private Sub btnAceptar_Click()
```

```
    If validaciones = True Then
```

```
        ingresaJustificacion
        Unload Me
        strMotivo1 = ""
        frmJustificaciones.Show
```

```
    End If
```

```
End Sub
```

'Subrutina que borra las justificaciones para ingresarlos nuevamente, no importa si no se hace ningún cambio.

```
Private Sub borraJustificacion()
```

```
    On Error GoTo sigue
```

```
    objEmpleados.borrarJustificaciones strCodigoEmpleado, CDate(strFechaJusti)
```

```
    Exit Sub
```

```
sigue:
    MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

'Subrutina "ingresaJustificacion" que permite almacenar los minutos justificados en el respectivo día de atraso, en caso de no haberlos, se ingresa "0".

```
Private Sub ingresaJustificacion()
```

```
    On Error GoTo sigue
```

```
    Dim intMinSa As Integer
```

```

Dim intMinEn As Integer

borraJustificacion

If OptJornada.Value = True Then

    objEmpleados.ingresaJustificaciones strCodigoEmpleado, CDate(strFechaJusti),
    intJusEntrada1, _
    intJusSalida1, intJusEntrada2, intJusSalida2, cmbMotivo.Text, ""

Else

    If txtMinutosEn1.Text = "" Then

        intMinEn = 0

    Else

        intMinEn = CInt(txtMinutosEn1.Text)

    End If

    If txtMinutosSa1.Text = "" Then

        intMinSa = 0

    Else

        intMinSa = CInt(txtMinutosSa1.Text)

    End If

    objEmpleados.ingresaJustificaciones strCodigoEmpleado, CDate(strFechaJusti),
    intMinEn, _
    intMinSa, 0, 0, cmbMotivo.Text, ""

End If

Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Sub

```

'Subrutina "validaciones", la cual chequea que los minutos justificados no sobrepasen a los atrasados y se ingrese un motivo de justificación.

```
Private Function validaciones() As Boolean
```

```
    On Error GoTo sigue
```

```
    Dim blnEn As Boolean
```

```
    Dim blnSa As Boolean
```

```
    validaciones = False
```

```
    If cmbMotivo.Text = "" Then
```

```
MsgBox "Escoja el motivo de la justificación", vbInformation, Me.Caption
validaciones = False
Exit Function

End If

If strJornadaEmp = "Única" Then
    If txtMinutosEn1.Text <> "" Then
        If Cint(txtMinutosEn1.Text) <= intJusEntrada1 Then
            blnEn = True
        Else
            blnEn = False
            MsgBox "Exceso de minutos justificados en la entrada.",
                vbInformation, Me.Caption
        End If
    End If
End If

If txtMinutosSa1.Text <> "" Then
    If Cint(txtMinutosSa1.Text) <= intJusSalida1 Then
        blnSa = True
    Else
        blnSa = False
        MsgBox "Exceso de minutos justificados en la salida.",
            vbInformation, Me.Caption
    End If
End If

If blnEn = True And blnSa = True Then
    validaciones = True
Else
    validaciones = False
End If

If OptJornada.Value = True Then
    validaciones = True
End If

Elseif strJornadaEmp = "Diferenciada" Then
    If txtMinutosEn1.Text <> "" Then
        If Cint(txtMinutosEn1.Text) <= intJusEntrada1 Then
```

```
        blnEn = True
    Else
        blnEn = False
        MsgBox "Exceso de minutos justificados en la entrada.",
        vbInformation, Me.Caption
    End If

    If    blnEn = True And blnSa = True Then
        validaciones = True
    Else
        validaciones = False
    End If
End If

If    txtMinutosSa1.Text <> "" Then
    If    CInt(txtMinutosSa1.Text) <= intJusSalida2 Then
        blnSa = True
    Else
        blnSa = False
        MsgBox "Exceso de minutos justificados en la salida.",
        vbInformation, Me.Caption
    End If

    If    blnEn = True And blnSa = True Then
        validaciones = True
    Else
        validaciones = False
    End If
End If

If    OptJornada.Value = True Then
    validaciones = True
End If

End If

Exit Function
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Function
```

'Botón Cancelar.

```
Private Sub btnCancelar_Click()  
  
    Unload Me  
    frmJustificaciones.Show  
  
End Sub
```

'Subrutina que impide borrar el texto de la lista del combo de "Seleccione el Motivo".

```
Private Sub cmbMotivo_KeyPress(KeyAscii As Integer)  
  
    KeyAscii = Alfabetico(KeyAscii)  
  
End Sub
```

'Subrutina que carga a la forma con los minutos justificados anteriormente.

```
Private Sub Form_Load()  
  
    txtMinutosEn1.Text = intJusEnJust1  
    txtMinutosSa1.Text = intJusSaJust1  
    cmbMotivo.Text = strMotivo1  
  
End Sub
```

'Subrutina que se activa cuando se marca "Jornada Completa" desactivando las demás entradas de la forma.

```
Private Sub OptJornada_Click()  
  
    If OptJornada.Value = True Then  
  
        txtMinutosEn1.Enabled = False  
        txtMinutosEn1.BackColor = &HC0C0C0  
        txtMinutosSa1.Enabled = False  
        txtMinutosSa1.BackColor = &HC0C0C0  
        txtMinutosEn1.Text = ""  
        txtMinutosSa1.Text = ""  
  
    End If  
  
End Sub
```

'Subrutina que se activa cuando se marca "Minutos Entrada" desactivando las demás entradas de la forma.

```
Private Sub optMinutosEn1_Click()  
  
    If optMinutosEn1.Value = True Then
```

```

        txtMinutosEn1.Enabled = True
        txtMinutosEn1.BackColor = &HFFFFFF
        txtMinutosSa1.Enabled = False
        txtMinutosSa1.BackColor = &HC0C0C0
    End If

```

End Sub

'Subrutina que se activa cuando se marca "Minutos Salida" desactivando las demás entradas de la forma.

```
Private Sub optMinutosSa1_Click()
```

```

    If    optMinutosSa1.Value = True Then

        txtMinutosSa1.Enabled = True
        txtMinutosSa1.BackColor = &HFFFFFF
        txtMinutosEn1.Enabled = False
        txtMinutosEn1.BackColor = &HC0C0C0
    End If

```

End Sub

'Subrutina que deshabilita la introducción de texto cuando se marca "Minutos Entrada"

```
Private Sub txtMinutosEn1_KeyPress(KeyAscii As Integer)
```

```

    KeyAscii = Numerico(KeyAscii)

```

End Sub

'Subrutina que deshabilita la introducción de texto cuando se marca "Minutos Salida"

```
Private Sub txtMinutosSa1_KeyPress(KeyAscii As Integer)
```

```

    KeyAscii = Numerico(KeyAscii)

```

End Sub

Código cargado en la forma fmrJustJorDoble

Esta forma hace uso de los siguientes módulos: Módulo1, clsEmpleados y clsEmpleadosDat.

'Declaración de variables.

Dim objEmpleados As New clsEmpleados

'Botón Aceptar, chequea si los datos pasaron las validaciones y de ser así ingresa los datos a la Base del SCA.

Private Sub btnAceptar_Click()

 If Frame1.Enabled = True And Frame2.Enabled = True Then

 If validaciones = True And validaciones2 = True Then

 ingresaJustificacion
 Unload Me
 strMotivo1 = ""
 strMotivo2 = ""
 frmJustificaciones.Show

 End If

 Elseif Frame1.Enabled = True And Frame2.Enabled = False Then

 If validaciones = True Then

 ingresaJustificacion
 Unload Me
 strMotivo1 = ""
 strMotivo2 = ""
 frmJustificaciones.Show

 End If

 Elseif Frame1.Enabled = False And Frame2.Enabled = True Then

 If validaciones2 = True Then

 ingresaJustificacion
 Unload Me
 strMotivo1 = ""
 strMotivo2 = ""
 frmJustificaciones.Show

 End If

 Elseif Frame1.Enabled = False And Frame2.Enabled = False Then

 If chkTodo.Value = 1 Then

 If cmbMotivoT.Text <> "" Then

 ingresaJustificacion
 Unload Me
 strMotivo1 = ""
 strMotivo2 = ""
 frmJustificaciones.Show

 Else


```

                MsgBox "Escoja el motivo de la justificación.", vbInformation,
                Me.Caption
    End If
End If
End If
End Sub

```

'Subrutina que borra las justificaciones para ingresarlos nuevamente, no importa si no se hace ningún cambio.

```

Private Sub borraJustificacion()
    On Error GoTo sigue
    objEmpleados.borrarJustificaciones strCodigoEmpleado, CDate(strFechaJusti)
Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source
End Sub

```

'Subrutina "ingresaJustificacion" que permite almacenar los minutos justificados en el respectivo día de atraso, en caso de no haberlos se ingresa "0".

```

Private Sub ingresaJustificacion()
    On Error GoTo sigue
    Dim intMinSa As Integer
    Dim intMinEn As Integer
    Dim intMinSa2 As Integer
    Dim intMinEn2 As Integer

    borraJustificacion

    If     chkTodo.Value = 1 Then

        objEmpleados.ingresaJustificaciones strCodigoEmpleado, CDate(strFechaJusti),
        intJusEntrada1, _
        intJusSalida1, intJusEntrada2, intJusSalida2, cmbMotivoT.Text, cmbMotivoT.Text

    ElseIf OptJornada.Value = True And optJornada2.Value = True Then

        objEmpleados.ingresaJustificaciones strCodigoEmpleado, CDate(strFechaJusti),
        intJusEntrada1, _
        intJusSalida1, intJusEntrada2, intJusSalida2, cmbMotivo1.Text, cmbMotivo2.Text

    ElseIf OptJornada.Value = True And optJornada2.Value = False Then

        If     txtMinutosEn2.Text = "" Then

```

```

        intMinEn2 = 0
    Else
        intMinEn2 = CInt(txtMinutosEn2.Text)
    End If

    If txtMinutosSa2.Text = "" Then
        intMinSa2 = 0
    Else
        intMinSa2 = CInt(txtMinutosSa2.Text)
    End If

    objEmpleados.ingresaJustificaciones strCodigoEmpleado,
    CDate(strFechaJusti), intJusEntrada1, _
    intJusSalida1, intMinEn2, intMinSa2, cmbMotivo1.Text, cmbMotivo2.Text

Elseif OptJornada.Value = False And optJornada2.Value = True Then

    If txtMinutosEn1.Text = "" Then
        intMinEn = 0
    Else
        intMinEn = CInt(txtMinutosEn1.Text)
    End If

    If txtMinutosSa1.Text = "" Then
        intMinSa = 0
    Else
        intMinSa = CInt(txtMinutosSa1.Text)
    End If

    objEmpleados.ingresaJustificaciones strCodigoEmpleado,
    CDate(strFechaJusti), intMinEn, _
    intMinSa, intJusEntrada2, intJusSalida2, cmbMotivo1.Text,
    cmbMotivo2.Text

Elseif OptJornada.Value = False And optJornada2.Value = False Then

    If txtMinutosEn1.Text = "" Then
        intMinEn = 0
    Else
        intMinEn = CInt(txtMinutosEn1.Text)
    End If

    If txtMinutosSa1.Text = "" Then
        intMinSa = 0
    Else
        intMinSa = CInt(txtMinutosSa1.Text)
    End If

```

```

    If      txtMinutosEn2.Text = "" Then
        intMinEn2 = 0
    Else
        intMinEn2 = CInt(txtMinutosEn2.Text)
    End If

    If      txtMinutosSa2.Text = "" Then
        intMinSa2 = 0
    Else
        intMinSa2 = CInt(txtMinutosSa2.Text)
    End If

    objEmpleados.ingresaJustificaciones strCodigoEmpleado,
    CDate(strFechaJusti), intMinEn, _
    intMinSa, intMinEn2, intMinSa2, cmbMotivo1.Text, cmbMotivo2.Text
End If

Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Sub

```

'Subrutina "validaciones"; valida la primera jornada, chequea que los minutos justificados no sobrepasen a los atrasados y se ingrese un motivo de justificación.

Private Function validaciones() As Boolean

```

    On Error GoTo sigue

    Dim blnEn As Boolean
    Dim blnSa As Boolean

    validaciones = False

    If      OptJornada.Value = False And optMinutosEn1.Value = False _
        And optMinutosSa1.Value = False Then

        Exit Function
    End If

    If      cmbMotivo1.Text = "" Then

        MsgBox "Escoja el motivo de la justificación en la Jornada 1", vbInformation,
        Me.Caption

        validaciones = False
        Exit Function
    End If

```

```

If      txtMinutosEn1.Text <> "" Then

    If      CInt(txtMinutosEn1.Text) <= intJusEntrada1 Then

        blnEn = True

    Else

        blnEn = False
        MsgBox "Exceso de minutos justificados en la entrada en la Jornada 1.",
        vbInformation, Me.Caption

    End If

End If

If      txtMinutosSa1.Text <> "" Then

    If      CInt(txtMinutosSa1.Text) <= intJusSalida1 Then

        blnSa = True

    Else

        blnSa = False

        MsgBox "Exceso de minutos justificados en la salida en la Jornada 1.",
        vbInformation, Me.Caption

    End If

End If

If      blnSa = True And blnEn = True Then

    validaciones = True

End If

If      OptJornada.Value = True Then

    validaciones = True

End If

Exit Function
sigue:
        MsgBox Err.Description, vbCritical, Err.Source

End Function

```

'Subrutina "validaciones2", valida la segunda jornada, chequea que los minutos justificados no sobrepasen a los atrasados y se ingrese un motivo de justificación.

```
Private Function validaciones2() As Boolean
```

```
    On Error GoTo sigue
```

```
    Dim blnEn As Boolean
    Dim blnSa As Boolean
```

```
    validaciones2 = False
```

```
If    optJornada2.Value = False And optMinutosEn2.Value = False _
      And optMinutosSa2.Value = False Then

      validaciones2 = True
      Exit Function
End If

If    cmbMotivo2.Text = "" Then

      MsgBox "Escoja el motivo de la justificación en la Jornada 2", vbInformation,
      Me.Caption
      validaciones2 = False
      Exit Function

End If

If    txtMinutosEn2.Text <> "" Then

      If    CInt(txtMinutosEn2.Text) <= intJusEntrada2 Then

              blnEn = True

      Else

              blnEn = False
              MsgBox "Exceso de minutos justificados en la entrada en la Jornada 2.",
              vbInformation, Me.Caption

      End If

End If

If    txtMinutosSa2.Text <> "" Then

      If    CInt(txtMinutosSa2.Text) <= intJusSalida2 Then

              blnSa = True

      Else

              blnSa = False
              MsgBox "Exceso de minutos justificados en la salida en la Jornada 2.",
              vbInformation, Me.Caption

      End If

End If

If    blnSa = True And blnEn = True Then

      validaciones2 = True
      End If

If    optJornada2.Value = True Then

      validaciones2 = True

End If

Exit Function
sigue:
      MsgBox Err.Description, vbCritical, Err.Source
```

End Function

'Botón Cancelar.

```
Private Sub btnCancelar_Click()
```

```
    Unload Me  
    frmJustificaciones.Show
```

```
End Sub
```

'Subrutina que se activa cuando se marca "Justificar Todo", la cual justifica todos los minutos atrasados en las jornadas que existan.

```
Private Sub chkTodo_Click()
```

```
    If    chkTodo.Value = 1 Then  
        Frame1.Enabled = False  
        Frame2.Enabled = False  
        txtMinutosEn1.Text = ""  
        txtMinutosSa1.Text = ""  
        txtMinutosEn2.Text = ""  
        txtMinutosSa2.Text = ""  
    Else  
        If    intJusEntrada1 = 0 And intJusSalida1 = 0 Then  
            Frame1.Enabled = False  
        Else  
            Frame1.Enabled = False  
        End If  
        If    intJusEntrada2 = 0 And intJusSalida2 = 0 Then  
            Frame2.Enabled = False  
        Else  
            Frame2.Enabled = False  
        End If  
    End If
```

```
End Sub
```

'Subrutina que impide borrar el texto de la lista del combo de "Seleccione el Motivo" de la primera jornada.

```
Private Sub cmbMotivo1_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Alfabetico(KeyAscii)
```

```
End Sub
```

'Subrutina que impide borrar el texto de la lista del combo de "Seleccione el Motivo" de la segunda jornada.

```
Private Sub cmbMotivo2_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Alfabetico(KeyAscii)
```

```
End Sub
```

'Subrutina que impide borrar el texto de la lista del combo de "Seleccione el Motivo" de "Justificar Todo".

```
Private Sub cmbMotivoT_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Alfabetico(KeyAscii)
```

```
End Sub
```

'Subrutina que carga a la forma con los minutos justificados anteriormente.

```
Private Sub Form_Load()
```

```
    If intJusEntrada1 = 0 And intJusSalida1 = 0 Then
```

```
        Frame1.Enabled = False
```

```
    End If
```

```
    If intJusEntrada2 = 0 And intJusSalida2 = 0 Then
```

```
        Frame2.Enabled = False
```

```
    End If
```

```
    btnMinutosEn1.Text = intJusEnJust1
```

```
    btnMinutosSa1.Text = intJusSaJust1
```

```
    btnMinutosEn2.Text = intJusEnJust2
```

```
    btnMinutosSa2.Text = intJusSaJust2
```

```
    If strMotivo1 = strMotivo2 Then
```

```
        cmbMotivoT.Text = strMotivo1
```

```
    ElseIf strMotivo1 <> "" And strMotivo2 = "" Then
```

```
        cmbMotivo1.Text = strMotivo1
```

```
    ElseIf strMotivo2 <> "" And strMotivo1 = "" Then
```

```
        cmbMotivo2.Text = strMotivo2
```

```
    End If
```

```
End Sub
```

'Subrutina que se activa cuando se marca "Jornada Completa" desactivando las demás entradas de la forma para la primera jornada.

```
Private Sub OptJornada_Click()
```

```
    If OptJornada.Value = True Then  
  
        txtMinutosEn1.Enabled = False  
        txtMinutosEn1.BackColor = &HC0C0C0  
        txtMinutosEn1.Text = ""  
        txtMinutosSa1.Enabled = False  
        txtMinutosSa1.BackColor = &HC0C0C0  
        txtMinutosSa1.Text = ""  
  
    End If
```

```
End Sub
```

'Subrutina que se activa cuando se marca "Jornada Completa" desactivando las demás entradas de la forma para la segunda jornada.

```
Private Sub optJornada2_Click()
```

```
    If optJornada2.Value = True Then  
  
        txtMinutosEn2.Enabled = False  
        txtMinutosEn2.BackColor = &HC0C0C0  
        txtMinutosEn2.Text = ""  
        txtMinutosSa2.Enabled = False  
        txtMinutosSa2.BackColor = &HC0C0C0  
        txtMinutosSa2.Text = ""  
  
    End If
```

```
End Sub
```

'Subrutina que se activa cuando se marca "Minutos Entrada" desactivando las demás entradas de la forma en la primera jornada.

```
Private Sub optMinutosEn1_Click()
```

```
    If optMinutosEn1.Value = True Then  
  
        txtMinutosEn1.Enabled = True  
        txtMinutosEn1.BackColor = &HFFFFFFF  
        txtMinutosSa1.Enabled = False  
        txtMinutosSa1.BackColor = &HC0C0C0  
  
    End If
```

```
End Sub
```

'Subrutina que se activa cuando se marca "Minutos Salida" desactivando las demás entradas de la forma en la primera jornada.

```
Private Sub optMinutosSa1_Click()  
    If    optMinutosSa1.Value = True Then  
        txtMinutosSa1.Enabled = True  
        txtMinutosSa1.BackColor = &HFFFFFF  
        txtMinutosEn1.Enabled = False  
        txtMinutosEn1.BackColor = &HC0C0C0  
    End If  
End Sub
```

'Subrutina que se activa cuando se marca "Minutos Entrada" desactivando las demás entradas de la forma en la segunda jornada.

```
Private Sub optMinutosEn2_Click()  
    If    optMinutosEn2.Value = True Then  
        txtMinutosEn2.Enabled = True  
        txtMinutosEn2.BackColor = &HFFFFFF  
        txtMinutosSa2.Enabled = False  
        txtMinutosSa2.BackColor = &HC0C0C0  
    End If  
End Sub
```

'Subrutina que se activa cuando se marca "Minutos Salida" desactivando las demás entradas de la forma en la segunda jornada.

```
Private Sub optMinutosSa2_Click()  
    If    optMinutosSa2.Value = True Then  
        txtMinutosSa2.Enabled = True  
        txtMinutosSa2.BackColor = &HFFFFFF  
        txtMinutosEn2.Enabled = False  
        txtMinutosEn2.BackColor = &HC0C0C0  
    End If  
End Sub
```

'Subrutina que deshabilita la introducción de texto cuando se marca "Minutos Entrada" en la primera jornada.

```
Private Sub txtMinutosEn1_KeyPress(KeyAscii As Integer)  
    KeyAscii = Numerico(KeyAscii)  
End Sub
```

'Subrutina que deshabilita la introducción de texto cuando se marca "Minutos Entrada" en la segunda jornada.

```
Private Sub txtMinutosEn2_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Numerico(KeyAscii)
```

```
End Sub
```

'Subrutina que deshabilita la introducción de texto cuando se marca "Minutos Salida" en la primera jornada.

```
Private Sub txtMinutosSa1_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Numerico(KeyAscii)
```

```
End Sub
```

'Subrutina que deshabilita la introducción de texto cuando se marca "Minutos Salida" en la segunda jornada.

```
Private Sub txtMinutosSa2_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = Numerico(KeyAscii)
```

```
End Sub
```

Código cargado en la forma fmrSelfFecha

Esta forma no hace uso de ningún módulo.

'Declaración de variables.

```
Dim rs As New ADODB.Recordset
Dim objEmpleado As New clsEmpleados
```

'Botón "Aceptar" de la forma, evalúa que las fechas concuerden con las fechas del registro.

```
Private Sub btnAceptar_Click()
```

```
    If Format(dtpHasta.Value, "yyyy-mm-dd") <= Mid(lblRegistro.Caption,
        Len(lblRegistro.Caption) - 9, Len(lblRegistro.Caption)) And _
        Format(dtpDesde.Value, "yyyy-mm-dd") < Format(dtpHasta.Value, "yyyy-mm-dd")
        Then
```

```
        strFechaInicio = dtpDesde.Value
        strFechaFin = dtpHasta.Value
```

```
frmJustificaciones.Show  
Unload Me
```

```
Else
```

```
MsgBox "Elija un rango menor o igual a la fecha de actualización del registro.",  
vbInformation, Me.Caption
```

```
End If
```

```
End Sub
```

'Botón "Salir".

```
Private Sub btnSalir_Click()
```

```
Unload Me
```

```
End Sub
```

'Subrutina que se activa cuando se carga la forma para invocar la subrutina "buscaFecha"

```
Private Sub Form_Load()
```

```
buscaFecha
```

```
End Sub
```

'Subrutina "buscaFecha" la cual busca la última fecha de actualización del registro.

```
Private Sub buscaFecha()
```

```
Dim strReg As String
```

```
Set rs = objEmpleado.buscaFechaAtraso
```

```
While Not rs.EOF
```

```
    If Not IsNull(rs.Fecha) Then strReg = rs.Fecha  
    rs.MoveNext
```

```
Wend
```

```
rs.Close
```

```
If strReg <> "" Then
```

```
    lblRegistro.Caption = lblRegistro.Caption & " " & strReg
```

```
Else
```

```
    lblRegistro.Caption = lblRegistro.Caption & " No existe registro"
```

```
End If
```

```
End Sub
```

Código cargado en la forma fmrVacaciones

Esta forma hace uso de los siguientes módulos: clsEmpleados y clsEmpleadosDat.

'Declaración de variables.

```
Dim objEmpleado As New clsEmpleados
Dim rs As New ADODB.Recordset
```

'Subrutina del botón "Aceptar", evalúa que las fechas ingresadas sean coherentes, tanto entre la fecha de salida y entrada como con las últimas vacaciones del empleado, si las hay, si todo esta bien ingresa las fechas en la Base de Datos del SCA.

```
Private Sub btnAceptar_Click()
```

```
    Dim dtFechaMax As Date
```

```
    If strAccion = "Agregar" Then
```

```
        Set rs = objEmpleado.buscaFechaRMayor(strCodigoEmpleado, True)
```

```
    Else
```

```
        Set rs = objEmpleado.buscaFechaRMayor(strCodigoEmpleado, False)
```

```
    End If
```

```
    While Not rs.EOF
```

```
        dtFechaMax = rs!FechaR
        rs.MoveNext
```

```
    Wend
```

```
    rs.Close
```

```
    If dtFechaRegreso.Value <= dtFechaSalida.Value Or dtFechaSalida.Value < Date _
        Or dtFechaSalida.Value < dtFechaMax Or dtFechaMax < Date Then
```

```
        MsgBox "Ingrese fechas coherentes.", vbInformation, Me.Caption
        Exit Sub
```

```
    End If
```

```
    If strAccion = "Agregar" Then
```

```
objEmpleado.ingresaVacaciones txtCodigo.Text, dtFechaSalida.Value,  
dtFechaRegreso.Value
```

```
Elseif strAccion = "Modificar" Then
```

```
objEmpleado.modificaVacaciones txtCodigo.Text, dtFechaSalida.Value,  
dtFechaRegreso.Value  
fmAdmEmpleados.Show
```

```
End If
```

```
Unload Me
```

```
End Sub
```

'Botón "Cancelar".

```
Private Sub btnCancelar_Click()
```

```
Unload Me  
fmAdmEmpleados.Show
```

```
End Sub
```

'Subrutina que se ejecuta cuando se carga la forma, setea por defecto las fechas de entrada y salida con la fecha presente. En caso de modificación o eliminación carga las ultimas vacaciones ingresadas para ese empleado.

```
Private Sub Form_Load()
```

```
dtFechaSalida.Value = Date  
dtFechaRegreso.Value = Date
```

```
llenaEmpleado
```

```
If strAccion = "Modificar" Then
```

```
Set rs = objEmpleado.buscaUltVacaciones(strCodigoEmpleado)
```

```
While Not rs.EOF
```

```
dtFechaSalida.Value = rs!dtFechaSalida  
dtFechaRegreso.Value = rs!dtFechaRegreso  
rs.MoveNext
```

```
Wend
```

```
rs.Close
```

```
End If
```

```
End Sub
```

'Subrutina "llenaEmpleado", carga los datos del empleado, Código y Nombres en la forma.

```
Private Sub llenaEmpleado()

    On Error GoTo sigue

    Set rs = objEmpleado.consultarEmpleado(strCodigoEmpleado)

    While Not rs.EOF

        txtCodigo.Text = rs!strCodigoEmpleado
        txtEmpleado.Text = rs!strApellidos & " " & rs!strNombres
        rs.MoveNext

    Wend

    rs.Close

    Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Sub
```

Código cargado en la forma ActRegistro

Esta forma hace uso de los siguientes módulos: Módulo1, clsEmpleados, clsEmpleadosDat, clsHorario y clsHorarioDat.

'Declaración de Variables.

```
Dim objEmpleado As New clsEmpleados

Dim rs As New ADODB.Recordset
Dim rs1 As New ADODB.Recordset
Dim rs2 As New ADODB.Recordset

Dim strSql As String
Dim strDia As String
Dim strEmp As String
Dim strFecha As String
Dim strP1 As String
Dim strP2 As String
Dim strP3 As String
Dim strP4 As String
Dim strE1 As String
Dim strS1 As String
Dim strE2 As String
Dim strS2 As String
Dim strEn1 As String
Dim strSa1 As String
Dim strEn2 As String
Dim strSa2 As String
Dim strJornada As String
```

```

Dim intReg As Integer
Dim blnUnificada As Boolean
Dim dtFecha As String

```

'Subrutina del botón "Aceptar", la cual verifica la fecha de la última actualización del registro, de tener que actualizarse llama a la subrutina "ActualizaRegistro".

```

Private Sub btnAceptar_Click()

    On Error GoTo sigue

    If    lblFecha.Caption = Format(DateAdd("d", -1, Date), "yyyy-mm-dd") Then

        MsgBox "El registro ya se actualizó el día de hoy.", vbInformation, Me.Caption
        Exit Sub

    End If

    ActualizaRegistro
    MsgBox "Proceso concluido con éxito.", vbInformation, Me.Caption
    Unload Me

Exit Sub
sigue:
    MsgBox Err.Number & "" & Err.Description, vcb, Err.Source

End Sub

```

'Subrutina que se ejecuta en el momento que se carga la forma, llama a la subrutina "buscaFecha" la cual devuelve la fecha de la última actualización del Registro.

```

Private Sub Form_Load()

    buscaFecha

End Sub

```

'Subrutina "buscaFecha", busca la fecha de la última actualización del Registro para visualizarla en la forma.

```

Private Sub buscaFecha()

    Dim strReg As String

    Set rs2 = objEmpleado.buscaFechaAtraso

    While Not rs2.EOF

        If    Not IsNull(rs2!Fecha) Then strReg = rs2!Fecha

        rs2.MoveNext

    Wend

```

```

rs2.Close

If strReg <> "" Then
    lblFecha.Caption = strReg
    dtFecha = strReg
Else
    lblFecha.Caption = "No existe registro"
    Set rs = objEmpleado.buscaFechasRegistro
    While Not rs.EOF
        dtFecha = rs!Fecha
        rs.MoveNext
    Wend
    rs.Close
End If
End Sub

```

'Subrutina "ActualizaRegistro", invoca a otras subrutinas para realizar la Evaluación de Asistencia de los empleados de la EPN.

```

Private Sub ActualizaRegistro()
    On Error GoTo sigue
    While Format(dtFecha, "yyyy-mm-dd") < Format(DateAdd("d", -1, Date), "yyyy-mm-dd")
        Set rs2 = objEmpleado.listarEmpleados
        While Not rs2.EOF
            strEmp = rs2!strCodigoEmpleado
            strFecha = Format(dtFecha, "yyyy-mm-dd")
            inicializarVar
            evaluar
            rs2.MoveNext
        Wend
        dtFecha = DateAdd("d", 1, dtFecha)
    rs2.Close
Wend
Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

```


End Sub

'Subrutina "inicializarVar", inicializa las variables que intervienen en el proceso de la Evaluación de Asistencia.

Private Sub inicializarVar()

On Error GoTo sigue

```

strSql = ""
strDia = ""
strP1 = ""
strP2 = ""
strP3 = ""
strP4 = ""
strE1 = ""
strS1 = ""
strE2 = ""
strS2 = ""
strEn1 = ""
strSa1 = ""
strEn2 = ""
strSa2 = ""
strJornada = ""
intReg = 0
blnUnificada = False

```

Exit Sub

sigue:

MsgBox Err.Description, vbCritical, Err.Source

End Sub

'Subrutina "evaluar", permite filtrar los días que el empleado no debió registrarse para que no vayan al proceso de Toma de Decisiones, además realiza un cambio de variables en el caso de jornada única o diferenciada para su evaluación, S1=S2, ya que en la base de datos se guarda por facilidad la salida como S2 y para el proceso se necesita que sea S1. Desde esta subrutina se llama al proceso de Toma de Decisiones según sea el caso.

Private Sub evaluar()

On Error GoTo sigue

```

strDia = buscaDia
intReg = buscaNumRegistros
buscaHorarioEmp

```

```

If strJornada = "No ingresar horario 0" Or _
strJornada = "" Then

```

Exit Sub

End If

```

If      intReg = 0 Or intReg = 1 Then

    strEn1 = ""
    strSa1 = ""
    strEn2 = ""
    strSa2 = ""

    If      strJornada <> "Jornada Diferenciada 1" Then
        strS1 = strS2
        blnUnificada = True
    End If

Else

    buscaRegistro

    If      strJornada = "Jornada Diferenciada 1" Then

        If      intReg = 2 Then

            evalua2Registros

        ElseIf  intReg = 3 Then

            evalua3Registros

        ElseIf  intReg = 4 Then

            evalua4Registros

        End If

    Else

        evaluaUnicaDif

    End If

End If

EvaluacionGeneral

Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Sub

```

'Función "buscaDia", busca todos los días del registro para la Evaluación de Asistencia, extrae un día a la vez.

Private Function buscaDia() As String

On Error GoTo sigue

buscaDia = DatePart("w", strFecha)

Exit Function

sigue:

MsgBox Err.Description, vbCritical, Err.Source

End Function

'Subrutina "buscaHorario", para el día que se encuentre evaluando busca el horario respectivo del empleado. La subrutina extrae todo el horario, por facilidad, para posteriormente escoger el día de la semana que se evaluará.

Private Sub buscaHorarioEmp()

On Error GoTo sigue

Dim intl As Integer

```
strSql = "SELECT strEntrada1L + ':00' strEntrada1L," & _
        "strSalida1L + ':00' strSalida1L," & _
        "strEntrada2L + ':00' strEntrada2L," & _
        "strSalida2L + ':00' strSalida2L," & _
        "strEntrada1M + ':00' strEntrada1M," & _
        "strSalida1M + ':00' strSalida1M," & _
        "strEntrada2M + ':00' strEntrada2M," & _
        "strSalida2M + ':00' strSalida2M," & _
        "strEntrada1Mr + ':00' strEntrada1Mr," & _
        "strSalida1Mr + ':00' strSalida1Mr," & _
        "strEntrada2Mr + ':00' strEntrada2Mr," & _
        "strSalida2Mr + ':00' strSalida2Mr," & _
        "strEntrada1J + ':00' strEntrada1J," & _
        "strSalida1J + ':00' strSalida1J," & _
        "strEntrada2J + ':00' strEntrada2J," & _
        "strSalida2J + ':00' strSalida2J," & _
        "strEntrada1V + ':00' strEntrada1V," & _
        "strSalida1V + ':00' strSalida1V," & _
        "strEntrada2V + ':00' strEntrada2V," & _
        "strSalida2V + ':00' strSalida2V," & _
        "strEntrada1S + ':00' strEntrada1S," & _
        "strSalida1S + ':00' strSalida1S," & _
        "strEntrada2S + ':00' strEntrada2S," & _
        "strSalida2S + ':00' strSalida2S,"

strSql = strSql & _
        "strJornadaL," & _
        "strJornadaM," & _
        "strJornadaMr," & _
        "strJornadaJ," & _
        "strJornadaV," & _
        "strJornadaS " & _
        "FROM HorarioEmpleado HE " & _
        "INNER JOIN Horarios H " & _
        "ON H.strCodigoHorario = HE.strCodigoHorario " & _
        "WHERE strcodigoEmpleado = " & strEmp & ""
```

rs1.Open strSql, con, adOpenDynamic, adLockOptimistic

While Not rs1.EOF

Select Case strDia

Case 2

strE1 = rs1!strEntrada1L
strS1 = rs1!strSalida1L
strE2 = rs1!strEntrada2L
strS2 = rs1!strSalida2L
strJornada = rs1!strJornadaL

Case 3

strE1 = rs1!strEntrada1M
strS1 = rs1!strSalida1M
strE2 = rs1!strEntrada2M
strS2 = rs1!strSalida2M
strJornada = rs1!strJornadaM

Case 4

strE1 = rs1!strEntrada1Mr
strS1 = rs1!strSalida1Mr
strE2 = rs1!strEntrada2Mr
strS2 = rs1!strSalida2Mr
strJornada = rs1!strJornadaMr

Case 5

strE1 = rs1!strEntrada1J
strS1 = rs1!strSalida1J
strE2 = rs1!strEntrada2J
strS2 = rs1!strSalida2J
strJornada = rs1!strJornadaJ

Case 6

strE1 = rs1!strEntrada1V
strS1 = rs1!strSalida1V
strE2 = rs1!strEntrada2V
strS2 = rs1!strSalida2V
strJornada = rs1!strJornadaV

Case 7

strE1 = rs1!strEntrada1S
strS1 = rs1!strSalida1S
strE2 = rs1!strEntrada2S
strS2 = rs1!strSalida2S
strJornada = rs1!strJornadaS

End Select

intl = intl + 1
rs1.MoveNext

Wend

rs1.Close

Exit Sub
sigue:

MsgBox Err.Description, vbCritical, Err.Source

End Sub

'Función "buscaNumRegistros", devuelve el número de registros que el empleado se registró en ese día de evaluación, dependiendo de la jornada y éste número de registros se llamará a la subrutina de Toma de Decisiones.

Private Function buscaNumRegistros() As Integer

On Error GoTo sigue

**strSql = "SELECT COUNT(*) Num " & _
"FROM Registro " & _
"WHERE CONVERT(VARCHAR,dtFecha,23) = " & strFecha & " " & _
"AND strCodigoEmpleado = " & strEmp & """**

rs.Open strSql, con, adOpenDynamic, adLockOptimistic

If Not rs.EOF And Not rs.BOF Then

buscaNumRegistros = rs!Num

rs.Close

Exit Function

sigue:

MsgBox Err.Description, vbCritical, Err.Source

End Function

'Subrutina "buscaRegistro", busca todos los registros que el empleado realizó en ese día de evaluación y setea las variables P1, P2, P3 y P4 equivalentes a Registro1, Registro2, Registro3 y Registro4 respectivamente.

Private Sub buscaRegistro()

On Error GoTo sigue

Dim intl As Integer

**strSql = "SELECT CONVERT(VARCHAR,dtFecha,108) dtFecha " & _
"FROM Registro " & _
"WHERE CONVERT(VARCHAR,dtFecha,23) = " & strFecha & " " & _
"AND strCodigoEmpleado = " & strEmp & " " & _
"ORDER BY CONVERT(VARCHAR,dtFecha,108) asc "**

rs.Open strSql, con, adOpenDynamic, adLockOptimistic

While Not rs.EOF

Select Case intl

Case 0

strP1 = rs!dtFecha

```

        Case 1
            strP2 = rs!dtFecha
        Case 2
            strP3 = rs!dtFecha
        Case 3
            strP4 = rs!dtFecha
    End Select

    intl = intl + 1
    rs.MoveNext

Wend

rs.Close

Exit Sub
    sigue:
        MsgBox Err.Description, vbCritical, Err.Source

End Sub

```

'Subrutina "evaluaUnicaDif", evalúa el caso de la Toma de Decisiones para la jornada única y diferenciada.

```

Private Sub evaluaUnicaDif()

    On Error GoTo sigue

    If    strP1 < strS2 And strP2 > strE1 Then

        strEn1 = strP1
        strSa1 = strP2

    Else

        strEn1 = ""
        strSa1 = ""

    End If

    'Cambio de variables para la "EvaluacionGeneral".

    If    strJornada <> "Jornada Diferenciada 1" Then

        strS1 = strS2
        blnUnificada = True

    End If

    Exit Sub
    sigue:
        MsgBox Err.Description, vbCritical, Err.Source

End Sub

```

'Subrutina "evalua2Registros", evalúa el caso de la Toma de Decisiones para la jornada diferenciada doble con dos registros del empleado.

```

Private Sub evalua2Registros()

    On Error GoTo sigue

    Dim intDifer As Integer
    Dim intHor As Integer

    If strP1 < strS1 And strP2 > strE1 Then

        intDifer = difFechas("00:30:00", strE2)
        intHor = difFechas("00:00:00", strP2)

        If intHor < intDifer Then

            strEn1 = strP1
            strSa1 = strP2
            strEn2 = ""
            strSa2 = ""

        Else

            strEn1 = ""
            strSa1 = ""
            strEn2 = ""
            strSa2 = ""

        End If

    Else

        strEn1 = ""
        strSa1 = ""

        If strP1 < strS2 And strP2 > strE2 Then

            intDifer = difFechas("00:30:00", strE2)
            intHor = difFechas("00:00:00", strP1)

            If intHor < intDifer Then

                strEn2 = ""
                strSa2 = ""

            Else

                strEn2 = strP1
                strSa2 = strP2

            End If

        Else

            strEn2 = ""
            strSa2 = ""

        End If

    End If

End Sub

```

```

Exit Sub
    sigue:
        MsgBox Err.Description, vbCritical, Err.Source

```

```
End Sub
```

'Subrutina "evalua3Registros", evalúa el caso de la Toma de Decisiones para la jornada diferenciada doble con tres registros del empleado.

```
Private Sub evalua3Registros()
```

```
    On Error GoTo sigue
```

```
    Dim intDifer As Integer
```

```
    Dim intHour As Integer
```

```
    If strP1 < strS1 And strP2 > strE1 Then
```

```
        intDifer = difFechas("00:30:00", strE2)
```

```
        intHour = difFechas("00:00:00", strP2)
```

```
        If intHour < intDifer Then
```

```
            strEn1 = strP1
```

```
            strSa1 = strP2
```

```
            strEn2 = ""
```

```
            strSa2 = ""
```

```
        Else
```

```
            If strP2 < strS2 And strP3 > strE2 Then
```

```
                strEn1 = ""
```

```
                strSa1 = ""
```

```
                strEn2 = strP2
```

```
                strSa2 = strP3
```

```
            Else
```

```
                strEn1 = ""
```

```
                strSa1 = ""
```

```
                strEn2 = ""
```

```
                strSa2 = ""
```

```
            End If
```

```
        End If
```

```
    Else
```

```
        strEn1 = ""
```

```
        strSa1 = ""
```

```
    If strP2 < strS2 And strP3 > strE2 Then
```

```
        intDifer = difFechas("00:30:00", strE2)
```

```
        intHour = difFechas("00:00:00", strP2)
```



```

        If      intHour < intDifer Then

            strEn2 = ""
            strSa2 = ""

        Else

            strEn2 = strP2
            strSa2 = strP3

        End If

    Else

        strEn2 = ""
        strSa2 = ""

    End If

End If

Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Sub

```

'Subrutina "evalua4Registros", evalúa el caso de la Toma de Decisiones para la jornada diferenciada doble con cuatro registros del empleado.

```
Private Sub evalua4Registros()
```

```
On Error GoTo sigue
```

```

    Dim intDifer As Integer
    Dim intHour As Integer

    If      strP1 < strS1 And strP2 > strE1 Then

        strEn1 = strP1
        strSa1 = strP2

    Else

        strEn1 = ""
        strSa1 = ""

    End If

    If      strP3 < strS2 And strP4 > strE2 Then

        intDifer = difFechas("00:30:00", strE2)
        intHour = difFechas("00:00:00", strP3)

        If      intHour < intDifer Then

            strEn2 = ""
            strSa2 = ""

```

```

        Else
            strEn2 = strP3
            strSa2 = strP4
        End If

    Else
        strEn2 = ""
        strSa2 = ""
    End If

Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Sub

```

'Subrutina "EvaluaciónGeneral", calcula los minutos atrasados en el caso de que el empleado los tenga.

```
Private Sub EvaluacionGeneral()
```

```
    On Error GoTo sigue
```

```

    Dim intDif As Integer
    Dim intAtE1 As Integer
    Dim intAtS1 As Integer
    Dim intAtE2 As Integer
    Dim intAtS2 As Integer

```

```
'Evalua entrada y salida 1.
```

```
If strEn1 = "" Then
```

```
    'ingresa atraso toda jornada 1.
```

```

    intDif = difFechas(strE1, strS1)
    intAtE1 = intDif
    intAtS1 = 0

```

```
Else
```

```
    intDif = difFechas(strE1, strEn1)
```

```
If intDif > 0 Then
```

```
    'ingresa atraso entrada1.
```

```
    intAtE1 = intDif
```

```
Else
```

```
    intAtE1 = 0
```

```
End If

intDif = difFechas(strS1, strSa1)

If intDif < 0 Then
    'ingresa atraso salida1.

    intAtS1 = intDif

Else

    intAtS1 = 0

End If

End If

'Evalua entrada2 y salida 2, en el caso de que sea jornada doble.

If strJornada = "Jornada Diferenciada 1" Then

    If strEn2 = "" Then

        'ingresa atraso toda jornada 2.

        intDif = difFechas(strE2, strS2)
        intAtE2 = intDif
        intAtS2 = 0

    Else

        intDif = difFechas(strE2, strEn2)

        If intDif > 0 Then

            'ingresa atraso entrada2.

            intAtE2 = intDif

        Else

            intAtE2 = 0

        End If

        intDif = difFechas(strS2, strSa2)

        If intDif < 0 Then

            'ingresa atraso salida2.

            intAtS2 = intDif

        Else

            intAtS2 = 0
```

```

                End If
            End If
        End If

        'Si no existe atraso alguno, no ingresar el día evaluado a la tabla de Atrasos.
        If    intAtE1 <> 0 Or intAtS1 <> 0 Or intAtE2 <> 0 Or intAtS2 <> 0 Then

            Dim intJor1 As Integer
            Dim intJor2 As Integer

            intJor1 = difFechas(strE1, strS1)

            If    blnUnificada = False Then intJor2 = difFechas(strE2, strS2)
                ingresaAtraso Abs(intAtE1), Abs(intAtS1), Abs(intAtE2), Abs(intAtS2),
                Abs(intJor1), Abs(intJor2)

        End If

        Exit Sub
        sigue:
            MsgBox Err.Description, vbCritical, Err.Source

    End Sub

```

'Función que permite realizar la resta de horas-minutos para la subrutina "EvaluacionGeneral"

```

Private Function difFechas(strFecha1 As String, strFecha2 As String) As Integer

    On Error GoTo sigue

    difFechas = DateDiff("n", strFecha1, strFecha2)

    Exit Function
        sigue:
            MsgBox Err.Description, vbCritical, Err.Source

End Function

```

'Subrutina "ingresaAtraso", permite ingresar los minutos atrasados en la Base de Datos del SCA en el caso que el empleado los haya tenido.

```

Private Sub ingresaAtraso(intMinEntJ1 As Integer, intMinSalJ1 As Integer, intMinEntJ2 As Integer,
intMinSalJ2 As Integer, intJor1 As Integer, intJor2 As Integer)

    On Error GoTo sigue

    strSql = "INSERT INTO Atrasos " & _
            "VALUES (" & strEmp & "," & _
            strFecha & "," & _
            intMinEntJ1 & "," & _

```

```

intMinSalJ1 & "," & _
intMinEntJ2 & "," & _
intMinSalJ2 & "," & _
intJor1 & "," & _
intJor2 & ")"

```

```
con.Execute strSql
```

```
Exit Sub
```

```
sigue:
```

```
MsgBox Err.Number & " " & Err.Description, vbCritical, Err.Source
```

```
End Sub
```

Código fuente cargado en las forma de consulta

Código cargado en la forma frmConEmpleados

Esta forma hace uso de los siguientes módulos: Módulo1, clsEmpleados y clsEmpleadosDat.

'Declaración de variables.

```

Dim objEmpleados As New clsEmpleados
Dim rs As New ADODB.Recordset

```

'Subrutina del botón "Aceptar", cierra la forma.

```
Private Sub btnAceptar_Click()
```

```
Unload Me
```

```
End Sub
```

'Subrutina que se activa cuando se carga la forma, llama a la subrutina "llenaForma".

```
Private Sub Form_Load()
```

```
llenaForma
```

```
End Sub
```

'Subrutina "llenaForma", extrae de la Base de Datos del SCA todos los datos personales del empleado escogido.

```
Private Sub llenaForma()
```

```

On Error GoTo sigue

Dim strSex As String

Set rs = objEmpleados.consultarEmpleado(strCodigoEmpleado)

While Not rs.EOF

    txtCodigo.Text = rs!strCodigoEmpleado
    txtNombres.Text = rs!strNombres
    txtApellidos.Text = rs!strApellidos
    txtCedula.Text = rs!strCedula
    txtFechaNac.Text = rs!dtFechaNacimiento
    txtEstadoCivil.Text = rs!strEstadoCivil
    txtDireccion.Text = rs!strDireccion
    txtTelefono1.Text = rs!strTelefono1
    txtTelefono2.Text = rs!strTelefono2
    txtFechaIng.Text = rs!dtIngreso
    strSex = rs!strSexo

    rs.MoveNext

Wend

If    strSex = "F" Then

    txtSexo.Text = "FEMENINO"

Elseif strSex = "M" Then

    txtSexo.Text = "MASCULINO"

End If

rs.Close

Exit Sub
sigue:
    MsgBox Err.Description, vbCr, Err.Source

End Sub

```

Código cargado en la forma fmrConHorarioAsig

Esta forma hace uso de los siguientes módulos: clsEmpleados, clsEmpleadosDat, clsHorario y clsHorarioDat.

'Declaración de variables.

```

Dim objEmpleados As New clsEmpleados
Dim rs As New ADODB.Recordset

```

'Subrutina del botón "Aceptar", cierra la forma.

```
Private Sub btnAceptar_Click()  
    Unload Me
```

```
End Sub
```

'Subrutina del botón "Salir", cierra la forma.

```
Private Sub btnSalir_Click()
```

```
    Unload Me
```

```
End Sub
```

'Subrutina que se activa cuando se carga la forma, llama a la subrutina "llenaPlan".

```
Private Sub Form_Load()
```

```
    llenaPlan
```

```
End Sub
```

'Subrutina "llenaPlan", extrae de la Base de Datos del SCA el horario asignado del empleado escogido, si el dato es "(vacío)" llena el espacio de color plomo.

```
Private Sub llenaPlan()
```

```
    On Error GoTo sigue
```

```
    Set rs = objEmpleados.buscaHorarioEmpCom(strCodigoEmpleado)
```

```
    While Not rs.EOF
```

```
        txtCodigo.Text = rs!strCodigoHorario  
        txtHorario.Text = rs!strDescripcion
```

```
        If rs!strEntrada1L <> "" Then
```

```
            txtEntrada1L.Text = rs!strEntrada1L
```

```
        Else
```

```
            txtEntrada1L.BackColor = &HE0E0E0 , 'Color plomo
```

```
        End If
```

```
        If rs!strSalida1L <> "" Then
```

```
            txtSalida1L.Text = rs!strSalida1L
```

```
        Else
```

```
            txtSalida1L.BackColor = &HE0E0E0
```

```
        End If
```

```
If      rs!strEntrada2L <> "" Then
    txtEntrada2L.Text = rs!strEntrada2L
Else
    txtEntrada2L.BackColor = &HE0E0E0
End If

If      rs!strSalida2L <> "" Then
    txtSalida2L.Text = rs!strSalida2L
Else
    txtEntrada2L.BackColor = &HE0E0E0
End If

If      rs!strEntrada1M <> "" Then
    txtEntrada1M.Text = rs!strEntrada1M
Else
    txtEntrada1M.BackColor = &HE0E0E0
End If

If      rs!strSalida1M <> "" Then
    txtSalida1M.Text = rs!strSalida1M
Else
    txtSalida1M.BackColor = &HE0E0E0
End If

If      rs!strEntrada2M <> "" Then
    txtEntrada2M.Text = rs!strEntrada2M
Else
    txtEntrada2M.BackColor = &HE0E0E0
End If

If      rs!strSalida2M <> "" Then
    txtSalida2M.Text = rs!strSalida2M
Else
    txtSalida2M.BackColor = &HE0E0E0
End If

If      rs!strEntrada1Mr <> "" Then
    txtEntrada1Mr.Text = rs!strEntrada1Mr
Else
    txtEntrada1Mr.BackColor = &HE0E0E0
End If

If      rs!strSalida1Mr <> "" Then
    txtSalida1Mr.Text = rs!strSalida1Mr
```



```
Else
    txtSalida1Mr.BackColor = &HE0E0E0
End If

If rs!strEntrada2Mr <> "" Then
    txtEntrada2Mr.Text = rs!strEntrada2Mr
Else
    txtEntrada2Mr.BackColor = &HE0E0E0
End If

If rs!strSalida2Mr <> "" Then
    txtSalida2Mr.Text = rs!strSalida2Mr
Else
    txtSalida2Mr.BackColor = &HE0E0E0
End If

If rs!strEntrada1J <> "" Then
    txtEntrada1J.Text = rs!strEntrada1J
Else
    txtEntrada1J.BackColor = &HE0E0E0
End If

If rs!strSalida1J <> "" Then
    txtSalida1J.Text = rs!strSalida1J
Else
    txtSalida1J.BackColor = &HE0E0E0
End If

If rs!strEntrada2J <> "" Then
    txtEntrada2J.Text = rs!strEntrada2J
Else
    txtEntrada2J.BackColor = &HE0E0E0
End If

If rs!strSalida2J <> "" Then
    txtSalida2J.Text = rs!strSalida2J
Else
    txtSalida2J.BackColor = &HE0E0E0
End If

If rs!strEntrada1V <> "" Then
    txtEntrada1V.Text = rs!strEntrada1V
Else
    txtEntrada1V.BackColor = &HE0E0E0
```

```
End If

If rs!strSalida1V <> "" Then
    txtSalida1V.Text = rs!strSalida1V
Else
    txtSalida1V.BackColor = &HE0E0E0
End If

If rs!strEntrada2V <> "" Then
    txtEntrada2V.Text = rs!strEntrada2V
Else
    txtEntrada2V.BackColor = &HE0E0E0
End If

If rs!strSalida2V <> "" Then
    txtSalida2V.Text = rs!strSalida2V
Else
    txtSalida2V.BackColor = &HE0E0E0
End If

If rs!strEntrada1S <> "" Then
    txtEntrada1S.Text = rs!strEntrada1S
Else
    txtEntrada1S.BackColor = &HE0E0E0
End If

If rs!strSalida1S <> "" Then
    txtSalida1S.Text = rs!strSalida1S
Else
    txtSalida1S.BackColor = &HE0E0E0
End If

If rs!strEntrada2S <> "" Then
    txtEntrada2S.Text = rs!strEntrada2S
Else
    txtEntrada2S.BackColor = &HE0E0E0
End If

If rs!strSalida2S <> "" Then
    txtSalida2S.Text = rs!strSalida2S
Else
    txtSalida2S.BackColor = &HE0E0E0
End If

rs.MoveNext
```

```
Wend

rs.Close

If      txtCodigo.Text = "" And txtDescripcion = "" Then
    MsgBox "El empleado no tiene horario asignado.", vbInformation, Me.Caption
    Exit Sub
End If

Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Me.Caption

End Sub
```

Código cargado en la forma frmConVacaciones

Esta forma hace uso de los siguientes módulos: clsEmpleados y clsEmpleadosDat.

'Declaración de variables.

```
Dim objEmpleado As New clsEmpleados
Dim rs As New ADODB.Recordset
```

'Subrutina del botón "Salir", cierra la forma.

```
Private Sub btnSalir_Click()
    Unload Me
End Sub
```

'Subrutina que se activa cuando se carga la forma, llama a la subrutina "llenaPlan"

```
Private Sub Form_Load()
    llenaPlan
End Sub
```

'Subrutina "llenaPlan", extrae de la Base de Datos del SCA todas las vacaciones asignadas al empleado para visualizarlas en la grilla respectiva.

```
Private Sub llenaPlan()
    On Error GoTo sigue
```

```

Dim intContadorFilas As Integer

grdVacaciones.Col = -1
grdVacaciones.Row = -1
grdVacaciones.Lock = True
intContadorFilas = 1

Set rs = objEmpleado.consultaVacaciones(strCodigoEmpleado)

While Not rs.EOF

    grdVacaciones.Col = 1
    grdVacaciones.MaxRows = intContadorFilas
    grdVacaciones.Row = intContadorFilas
    grdVacaciones.Text = rs!Mes
    grdVacaciones.Col = 2
    grdVacaciones.Text = rs!dtFechaSalida
    grdVacaciones.Col = 3
    grdVacaciones.Text = rs!dtFechaRegreso
    grdVacaciones.Col = 4
    grdVacaciones.Text = rs!NumDias

    intContadorFilas = intContadorFilas + 1
    rs.MoveNext

Wend

rs.Close

Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Sub

```

Código cargado en la forma fmrConHorario

Esta forma hace uso de los siguientes módulos: clsHorario y clsHorarioDat.

'Declaración de variables.

```

Dim objHorarios As New clsHorarios
Dim rs As New ADODB.Recordset

```

'Subrutina del botón "Aceptar", cierra la forma.

```

Private Sub btnAceptar_Click()

```

```

    Unload Me

```

```

End Sub

```

'Subrutina que se activa cuando se carga la forma, llama a la subrutina "llenaPlan".

```
Private Sub Form_Load()
```

```
    llenaPlan
```

```
End Sub
```

'Subrutina "llenaPlan", extrae de la Base de Datos del SCA el horario escogido, si el dato es "(vacío)" llena el espacio de color plomo.

```
Private Sub llenaPlan()
```

```
    On Error GoTo sigue
```

```
    Set rs = objHorarios.consultaHorarios(strCodigoHorario)
```

```
    While Not rs.EOF
```

```
        txtCodigo.Text = rs!strCodigoHorario  
        txtHorario.Text = rs!strDescripcion
```

```
        If rs!strEntrada1L <> "" Then
```

```
            txtEntrada1L.Text = rs!strEntrada1L
```

```
        Else
```

```
            txtEntrada1L.BackColor = &HE0E0E0
```

```
        End If
```

```
        If rs!strSalida1L <> "" Then
```

```
            txtSalida1L.Text = rs!strSalida1L
```

```
        Else
```

```
            txtSalida1L.BackColor = &HE0E0E0
```

```
        End If
```

```
        If rs!strEntrada2L <> "" Then
```

```
            txtEntrada2L.Text = rs!strEntrada2L
```

```
        Else
```

```
            txtEntrada2L.BackColor = &HE0E0E0
```

```
        End If
```

```
        If rs!strSalida2L <> "" Then
```

```
            txtSalida2L.Text = rs!strSalida2L
```

```
        Else
```

```
            txtEntrada2L.BackColor = &HE0E0E0
```

```
        End If
```

```
If      rs!strEntrada1M <> "" Then
    txtEntrada1M.Text = rs!strEntrada1M
Else
    txtEntrada1M.BackColor = &HE0E0E0
End If
If      rs!strSalida1M <> "" Then
    txtSalida1M.Text = rs!strSalida1M
Else
    txtSalida1M.BackColor = &HE0E0E0
End If
If      rs!strEntrada2M <> "" Then
    txtEntrada2M.Text = rs!strEntrada2M
Else
    txtEntrada2M.BackColor = &HE0E0E0
End If
If      rs!strSalida2M <> "" Then
    txtSalida2M.Text = rs!strSalida2M
Else
    txtSalida2M.BackColor = &HE0E0E0
End If
If      rs!strEntrada1Mr <> "" Then
    txtEntrada1Mr.Text = rs!strEntrada1Mr
Else
    txtEntrada1Mr.BackColor = &HE0E0E0
End If
If      rs!strSalida1Mr <> "" Then
    txtSalida1Mr.Text = rs!strSalida1Mr
Else
    txtSalida1Mr.BackColor = &HE0E0E0
End If
If      rs!strEntrada2Mr <> "" Then
    txtEntrada2Mr.Text = rs!strEntrada2Mr
Else
    txtEntrada2Mr.BackColor = &HE0E0E0
```

```
End If

If rs!strSalida2Mr <> "" Then
    txtSalida2Mr.Text = rs!strSalida2Mr
Else
    txtSalida2Mr.BackColor = &HE0E0E0
End If

If rs!strEntrada1J <> "" Then
    txtEntrada1J.Text = rs!strEntrada1J
Else
    txtEntrada1J.BackColor = &HE0E0E0
End If

If rs!strSalida1J <> "" Then
    txtSalida1J.Text = rs!strSalida1J
Else
    txtSalida1J.BackColor = &HE0E0E0
End If

If rs!strEntrada2J <> "" Then
    txtEntrada2J.Text = rs!strEntrada2J
Else
    txtEntrada2J.BackColor = &HE0E0E0
End If

If rs!strSalida2J <> "" Then
    txtSalida2J.Text = rs!strSalida2J
Else
    txtSalida2J.BackColor = &HE0E0E0
End If

If rs!strEntrada1V <> "" Then
    txtEntrada1V.Text = rs!strEntrada1V
Else
    txtEntrada1V.BackColor = &HE0E0E0
End If

If rs!strSalida1V <> "" Then
    txtSalida1V.Text = rs!strSalida1V
Else
    txtSalida1V.BackColor = &HE0E0E0
End If

If rs!strEntrada2V <> "" Then
```

```
        txtEntrada2V.Text = rs!strEntrada2V
    Else

        txtEntrada2V.BackColor = &HE0E0E0
    End If

    If    rs!strSalida2V <> "" Then

        txtSalida2V.Text = rs!strSalida2V
    Else

        txtSalida2V.BackColor = &HE0E0E0
    End If

    If    rs!strEntrada1S <> "" Then

        txtEntrada1S.Text = rs!strEntrada1S
    Else

        txtEntrada1S.BackColor = &HE0E0E0
    End If

    If    rs!strSalida1S <> "" Then

        txtSalida1S.Text = rs!strSalida1S
    Else

        txtSalida1S.BackColor = &HE0E0E0
    End If

    If    rs!strEntrada2S <> "" Then

        txtEntrada2S.Text = rs!strEntrada2S
    Else

        txtEntrada2S.BackColor = &HE0E0E0
    End If

    If    rs!strSalida2S <> "" Then

        txtSalida2S.Text = rs!strSalida2S
    Else

        txtSalida2S.BackColor = &HE0E0E0
    End If

    If    Mid(rs!strJornadaL, Len(rs!strJornadaL), Len(rs!strJornadaL)) = 1 Then

        txtJornadaL.Text = Mid(rs!strJornadaL, 1, Len(rs!strJornadaL) - 1) & "D"
    Else

        txtJornadaL.Text = Mid(rs!strJornadaL, 1, Len(rs!strJornadaL) - 1)
    End If

    If    Mid(rs!strJornadaM, Len(rs!strJornadaM), Len(rs!strJornadaM)) = 1 Then
```



```
        txtJornadaM.Text = Mid(rs!strJornadaM, 1, Len(rs!strJornadaM) - 1) & "D"
    Else
        txtJornadaM.Text = Mid(rs!strJornadaM, 1, Len(rs!strJornadaM) - 1)
    End If
    If Mid(rs!strJornadaMr, Len(rs!strJornadaMr), Len(rs!strJornadaMr)) = 1 Then
        txtJornadaMr.Text = Mid(rs!strJornadaMr, 1, Len(rs!strJornadaMr) - 1) &
        "D"
    Else
        txtJornadaMr.Text = Mid(rs!strJornadaMr, 1, Len(rs!strJornadaMr) - 1)
    End If
    If Mid(rs!strJornadaJ, Len(rs!strJornadaJ), Len(rs!strJornadaJ)) = 1 Then
        txtJornadaJ.Text = Mid(rs!strJornadaJ, 1, Len(rs!strJornadaJ) - 1) & "D"
    Else
        txtJornadaJ.Text = Mid(rs!strJornadaJ, 1, Len(rs!strJornadaJ) - 1)
    End If
    If Mid(rs!strJornadaV, Len(rs!strJornadaV), Len(rs!strJornadaV)) = 1 Then
        txtJornadaV.Text = Mid(rs!strJornadaV, 1, Len(rs!strJornadaV) - 1) & "D"
    Else
        txtJornadaV.Text = Mid(rs!strJornadaV, 1, Len(rs!strJornadaV) - 1)
    End If
    If Mid(rs!strJornadaS, Len(rs!strJornadaS), Len(rs!strJornadaS)) = 1 Then
        txtJornadaS.Text = Mid(rs!strJornadaS, 1, Len(rs!strJornadaS) - 1) & "D"
    Else
        txtJornadaS.Text = Mid(rs!strJornadaS, 1, Len(rs!strJornadaS) - 1)
    End If

    rs.MoveNext

Wend

rs.Close

Exit Sub
sigue:
```

MsgBox Err.Description, vbCritical, Err.Source

End Sub

Módulos

Module 1

'Declaración de variables públicas.

```

Public rsHor As New ADODB.Recordset
Public strSql As String
Public strMenu As String
Public strAccion As String
Public strCodigoEmpleado As String
Public strNombreEmpleado As String
Public strCodigoHorario As String
Public blnVariable As Boolean
Public strFechaInicio As String
Public strFechaFin As String
Public intJusEntrada1 As Integer
Public intJusSalida1 As Integer
Public intJusEntrada2 As Integer
Public intJusSalida2 As Integer
Public intJusEnJust1 As Integer
Public intJusSaJust1 As Integer
Public intJusEnJust2 As Integer
Public intJusSaJust2 As Integer
Public strJornadaEmp As String
Public strFechaJusti As String
Public strMotivo1 As String
Public strMotivo2 As String

Public Const strCon = "Provider=SQLOLEDB.1;Persist Security Info=false;User
ID=sa;Initial Catalog=EmpleadosEPN;Data Source=XAVIER;Connect Timeout=30"

Public con As New ADODB.Connection

```

'Función "Numerico", permite ingresar en una caja de texto solo números.

```

Public Function Numerico(KeyAscii As Integer) As Integer

    If (KeyAscii% <> 8) And ((KeyAscii% < 48) Or (KeyAscii% > 57)) Then

        KeyAscii% = 0
    End If

    Numerico = KeyAscii%

End Function

```

'Función "Alfabetico", permite ingresar en una caja de texto solo letras del alfabeto.

Public Function Alfabetico(KeyAscii As Integer) As Integer

```

If      (KeyAscii% <> 8) And ((KeyAscii% < 65) Or (KeyAscii% > 90)) And _
      ((KeyAscii% < 97) Or (KeyAscii% > 122)) Then

      KeyAscii% = 0

Else

      KeyAscii% = Asc(UCase(Chr(KeyAscii%)))

End If

Alfabetico = KeyAscii%

```

End Function

'Función "AlfabeticoConEspacio", permite ingresar en una caja de texto solo letras del alfabeto con espacio.

Public Function AlfabeticoConEspacio(KeyAscii As Integer) As Integer

```

If      (KeyAscii% <> 8) And (KeyAscii% <> 32) And ((KeyAscii% < 65) Or (KeyAscii% >
      90)) And _
      ((KeyAscii% < 97) Or (KeyAscii% > 122)) Then

      KeyAscii% = 0

Else

      KeyAscii% = Asc(UCase$(Chr$(KeyAscii%)))

End If

AlfabeticoConEspacio = KeyAscii%

```

End Function

'Función "AlfabeticoConEspacioNum", permite ingresar en una caja de texto letras, números y el espacio normal excluyendo símbolos y otros caracteres.

Public Function AlfabeticoConEspacioNum(KeyAscii As Integer) As Integer

```

      KeyAscii% = Asc(UCase$(Chr$(KeyAscii%)))

      AlfabeticoConEspacioNum = KeyAscii%

```

End Function

'Subrutina pública "listaEmpleados". Llena la grilla con la lista de los empleados creada en la Base de Datos del SCA.

Public Sub listaEmpleados(grdEmpleados As Object, rs As ADODB.Recordset)

```

      Dim intContadorFilas As Integer

```

```

grdEmpleados.Col = -1
grdEmpleados.Row = -1
grdEmpleados.Lock = True
intContadorFilas = 1

While Not rs.EOF

    grdEmpleados.Col = 1
    grdEmpleados.MaxRows = intContadorFilas
    grdEmpleados.Row = intContadorFilas
    grdEmpleados.Text = rs!strApellidos
    grdEmpleados.Col = 2
    grdEmpleados.Text = rs!strNombres
    grdEmpleados.Col = 3
    grdEmpleados.Text = rs!strCodigoEmpleado

    intContadorFilas = intContadorFilas + 1
    rs.MoveNext

Wend

rs.Close

End Sub

```

Módulos de Clase

Por jerarquía de programación por cada tipo de módulo de clase existirá un módulo de clase adicional que realice las interacciones propiamente dichas con la Base de Datos del SCA, estos módulos tendrán funciones con los mismos nombres, el primer módulo indicará la acción que realiza y el segundo, "Dat", indicará ¿cómo se realiza?.

Módulo clsEmpleados

'Declaración de variables.

```

Private strCodigoEmpleado As String
Private strNombres As String
Private strApellidos As String
Private strDireccion As String
Private strTelefono1 As String
Private strCedula As String
Private strSexo As String
Private dtFechaNacimiento As Date
Private strEstadoCivil As String
Private dtIngreso As Date
Private strTelefono2 As String

Dim rs As New ADODB.Recordset

```

'Declaración de los Datos Personales del empleado como variables propias de la clase.

Public Property Let codigoEmpleado(strCodigo As String)

strCodigoEmpleado = strCodigo

End Property

Public Property Let nombreEmpleado(strNombre As String)

strNombres = strNombre

End Property

Public Property Let ApellidosEmpleado(strApellido As String)

strApellidos = strApellido

End Property

Public Property Let direccionEmpleado(strDireccionE As String)

strDireccion = strDireccionE

End Property

Public Property Let telefono1Empleado(strTelefono1E As String)

strTelefono1 = strTelefono1E

End Property

Public Property Let cedulaEmpleado(strCedulaE As String)

strCedula = strCedulaE

End Property

Public Property Let sexoEmpleado(strSexoE As String)

strSexo = strSexoE

End Property

Public Property Let FechaNacEmpleado(dtFechaNacimientoE As Date)

dtFechaNacimiento = dtFechaNacimientoE

End Property

Public Property Let estadoEmpleado(strEstadoCivE As String)

 strEstadoCivE = strEstadoCivE

End Property

Public Property Let FechaIngEmpleado(dtIngresoE As Date)

 dtIngreso = strEstadoCivE

End Property

Public Property Let telefono2(strTelefono2E As String)

 strTelefono2 = strTelefono2E

End Property

'Función pública "listarEmpleados", extrae la lista de todos los empleados almacenados en la Base de Datos del SCA, se enlaza con el módulo clsEmpleadoDat para ingresar a la Base de Datos.

Public Function listarEmpleados() As ADODB.Recordset

 On Error GoTo sigue

 Set objEmpleadoDat = New clsEmpleadosDat
 Set listarEmpleados = objEmpleadoDat.listarEmpleados
 Set objEmpleado = Nothing

 Exit Function

 sigue:

 MsgBox Err.Description, vbCritical, Err.Source

End Function

'Función pública "ingresarEmpleado", ingresa los Datos Personales del empleado en la Base de Datos del SCA, se enlaza con el módulo clsEmpleadosDat para revisar que no se ingrese un mismo código para dos empleados e ingresar a la Base de Datos.

Public Sub ingresarEmpleado(strCodigo As String, strNombres As String, strApellidos As String, _ strDireccion As String, strTelefono1 As String, strCedula As String, _ strSexo As String, dtFechaNac As Date, strEstadoCiv As String, dtIngreso As Date, strTelefono2 As String)

 On Error GoTo sigue

 Set objEmpleadoDat = New clsEmpleadosDat

```
objEmpleadoDat.ingresarEmpleado strCodigo, strNombres, strApellidos, strDireccion,  
strTelefono1, strCedula, _  
strSexo, dtFechaNac, strEstadoCiv, dtIngreso, strTelefono2
```

```
Set objEmpleadoDat = Nothing
```

```
Exit Sub
```

```
sigue:
```

```
MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

'Función pública "eliminarEmpleado", elimina los datos personales del empleado de la Tabla "Empleados", se enlaza con el módulo clsEmpleadosDat para revisar que el empleado no tenga asignado datos de administración, en caso que los tenga no podrá eliminarse hasta que no haya eliminado primero las asignaciones.

```
Public Sub eliminarEmpleado(strCodigo As String)
```

```
On Error GoTo sigue
```

```
Set objEmpleadoDat = New clsEmpleadosDat  
objEmpleadoDat.eliminarEmpleado (strCodigo)
```

```
Set objEmpleadoDat = Nothing
```

```
Exit Sub
```

```
sigue:
```

```
MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

'Función pública "consultarEmpleado", extrae los datos personales del empleado de la Base de Datos del SCA para consulta, se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Function consultarEmpleado(strCodigo As String) As ADODB.Recordset
```

```
On Error GoTo sigue
```

```
Set objEmpleadoDat = New clsEmpleadosDat  
Set consultarEmpleado = objEmpleadoDat.consultarEmpleado(strCodigo)  
Set objEmpleadoDat = Nothing
```

```
Exit Function
```

```
sigue:
```

```
MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

'Función pública "modificarEmpleado", ingresa los datos modificados personales del empleado en la Tabla "Empleados", se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos los datos modificados.

```
Public Sub modificarEmpleado(strCodigo As String, strNombres As String, strApellidos As String, _ strDireccion As String, strTelefono1 As String, strCedula As String, _ strSexo As String, dtFechaNac As Date, strEstadoCiv As String, dtIngreso As Date, strTelefono2 As String)
```

```
On Error GoTo sigue
```

```
Set objEmpleadoDat = New clsEmpleadosDat
objEmpleadoDat.modificarEmpleado strCodigo, strNombres, strApellidos, _ strDireccion, strTelefono1, strCedula, _ strSexo, dtFechaNac, strEstadoCiv, dtIngreso, strTelefono2
```

```
Set objEmpleadoDat = Nothing
```

```
Exit Sub
```

```
sigue:
```

```
MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

'Función pública "buscaEmpleadoAp", permite buscar en la grilla un empleado específico por apellidos, se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos y realizar la búsqueda.

```
Public Function buscaEmpleadoAp(strApellidos As String) As ADODB.Recordset
```

```
On Error GoTo sigue
```

```
Set objEmpleadoDat = New clsEmpleadosDat
Set buscaEmpleadoAp = objEmpleadoDat.buscaEmpleadoAp(strApellidos)
Set objEmpleadoDat = Nothing
```

```
Exit Function
```

```
sigue:
```

```
MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

'Función pública "asignarHorario", permite ingresar una asignación de horario para un empleado en la Tabla "HorarioEmpleado", se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos la asignación realizada.

```
Public Sub asignarHorario(strEmpleado As String, strHorario As String)
```

```
On Error GoTo sigue
```

```
Set objEmpleadoDat = New clsEmpleadosDat
objEmpleadoDat.asignarHorario strEmpleado, strHorario
Set objEmpleadoDat = Nothing
```

```
Exit Sub
```

```
sigue:
```

```
MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```


'Función pública "buscaHorarioEmp", extrae el horario asignado de un empleado de la Tabla "Horarios", se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Function buscaHorarioEmp(strEmpleado As String) As ADODB.Recordset
```

```
    On Error GoTo sigue
```

```
    Set objEmpleadoDat = New clsEmpleadosDat
```

```
    Set buscaHorarioEmp = objEmpleadoDat.buscaHorarioEmp(strEmpleado)
```

```
    Set objEmpleadoDat = Nothing
```

```
    Exit Function
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

'Función pública "desAsignarHorario", elimina una asignación de horario para un empleado de la Tabla "HorarioEmpleado", se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Sub desAsignarHorario(strEmpleado As String)
```

```
    On Error GoTo sigue
```

```
    Set objEmpleadoDat = New clsEmpleadosDat
```

```
    objEmpleadoDat.desAsignarHorario (strEmpleado)
```

```
    Set objEmpleadoDat = Nothing
```

```
    Exit Sub
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

'Función pública "buscaHorarioEmpCom", extrae el horario asignado a un empleado de la Tabla "Hoarios" para consulta, se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos y verificar una asignación.

```
Public Function buscaHorarioEmpCom(strEmpleado As String) As ADODB.Recordset
```

```
    On Error GoTo sigue
```

```
    Set objEmpleadoDat = New clsEmpleadosDat
```

```
    Set buscaHorarioEmpCom = objEmpleadoDat.buscaHorarioEmpCom(strEmpleado)
```

```
    Set objEmpleadoDat = Nothing
```

```
    Exit Function
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

'Función pública "ingresaVacaciones", ingresa las vacaciones asignadas a un empleado en la Tabla "Vacaciones", se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Sub ingresaVacaciones(strEmpleado As String, dtFechaIn As Date, dtFechaFin As Date)
```

```
    On Error GoTo sigue
```

```
    Set objEmpleadoDat = New clsEmpleadosDat  
    objEmpleadoDat.ingresaVacaciones strEmpleado, dtFechaIn, dtFechaFin  
    Set objEmpleadoDat = Nothing
```

```
Exit Sub
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

'Función pública "buscaUltVacaciones", extrae las vacaciones asignadas a un empleado de la Tabla "Vacaciones" para consulta, se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Function buscaUltVacaciones(strEmpleado As String) As ADODB.Recordset
```

```
    On Error GoTo sigue
```

```
    Set objEmpleadoDat = New clsEmpleadosDat  
    Set buscaUltVacaciones = objEmpleadoDat.buscaUltVacaciones(strEmpleado)  
    Set objEmpleadoDat = Nothing
```

```
Exit Function
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

'Función pública "modificaVacaciones", ingresa los datos modificados de las vacaciones de un empleado en la Tabla "Vacaciones", se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos los datos modificados.

```
Public Sub modificaVacaciones(strEmpleado As String, dtFechaIn As Date, dtFechaFin As Date)
```

```
    On Error GoTo sigue
```

```
    Set objEmpleadoDat = New clsEmpleadosDat  
    objEmpleadoDat.modificaVacaciones strEmpleado, dtFechaIn, dtFechaFin  
    Set objEmpleadoDat = Nothing
```

```
Exit Sub
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

'Función pública "eliminarVacaciones", elimina las últimas vacaciones ingresadas de la Tabla "Vacaciones", se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Sub eliminarVacaciones(strEmpleado As String)

    On Error GoTo sigue

    Set objEmpleadoDat = New clsEmpleadosDat
    objEmpleadoDat.eliminarVacaciones strEmpleado
    Set objEmpleadoDat = Nothing

    Exit Sub
    sigue:
        MsgBox Err.Description, vbCritical, Err.Source

End Sub
```

'Función pública "consultaVacaciones", extrae los datos de todas las vacaciones ingresadas para un empleado desde la Tabla "Vacaciones", se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Function consultaVacaciones(strEmpleado As String) As ADODB.Recordset

    On Error GoTo sigue

    Set objEmpleadoDat = New clsEmpleadosDat
    Set consultaVacaciones = objEmpleadoDat.consultaVacaciones(strEmpleado)
    Set objEmpleadoDat = Nothing

    Exit Function
    sigue:
        MsgBox Err.Description, vbCritical, Err.Source

End Function
```

'Función pública "buscaFechaRMayor", busca la mayor fecha de regreso de vacaciones para prevenir que el usuario no superponga vacaciones existentes, se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Function buscaFechaRMayor(strEmpleado As String, binIn As Boolean) As ADODB.Recordset

    On Error GoTo sigue

    Set objEmpleadoDat = New clsEmpleadosDat
    Set buscaFechaRMayor = objEmpleadoDat.buscaFechaRMayor(strEmpleado, binIn)
    Set objEmpleadoDat = Nothing

    Exit Function
    sigue:
        MsgBox Err.Source, vbCritical, Err.Source

End Function
```

'Función pública "buscaFechaAtraso", busca la fecha del último atraso registro por el SCA, se ejecuta cada vez que se actualiza el Registro, se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Function buscaFechaAtraso() As ADODB.Recordset
```

```
    On Error GoTo sigue
```

```
    Set objEmpleadoDat = New clsEmpleadosDat
    Set buscaFechaAtraso = objEmpleadoDat.buscaFechaAtraso
    Set objEmpleadoDat = Nothing
```

```
    Exit Function
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

'Función pública "buscaFechasRegistro", busca la fecha del primer registro realizado por un empleado para tomarla como fecha de inicio para la "Evaluación de Asistencia", se ejecuta cuando el Registro se va actualizar por primera vez, se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Function buscaFechasRegistro() As ADODB.Recordset
```

```
    On Error GoTo sigue
```

```
    Set objEmpleadoDat = New clsEmpleadosDat
    Set buscaFechasRegistro = objEmpleadoDat.buscaFechasRegistro
    Set objEmpleadoDat = Nothing
```

```
    Exit Function
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

'Función pública "justificaciones", extrae los datos de todas las justificaciones ingresadas en su respectiva fecha, si las hay, para un empleado desde la Tabla "Justificaciones", se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Function justificaciones(strEmp As String, strFinicio As String, strFfin As String) As ADODB.Recordset
```

```
    On Error GoTo sigue
```

```
    Set objEmpleadoDat = New clsEmpleadosDat
    Set justificaciones = objEmpleadoDat.justificaciones(strEmp, strFinicio, strFfin)
    Set objEmpleadoDat = Nothing
```

```
    Exit Function
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

End Function

'Función pública "ingresaJustificaciones", ingresa las justificaciones asignadas a un empleado en su respectivo día y tipo de jornada en la Tabla "Justificaciones", se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Sub   ingresaJustificaciones(strEmp As String, dtFecha As Date, intEn1 As Integer, _
            intSa1 As Integer, intEn2 As Integer, intSa2 As Integer, _
            strMotivo1 As String, strMotivo2 As String)
```

```
    On Error GoTo sigue
```

```
    Set objEmpleadoDat = New clsEmpleadosDat
    objEmpleadoDat.ingresaJustificaciones strEmp, dtFecha, intEn1, intSa1, intEn2, intSa2,
    strMotivo1, strMotivo2
    Set objEmpleadoDat = Nothing
```

```
Exit Sub
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

'Función pública "borrarJustificaciones", cuando se realiza una modificación es más fácil borrar los datos existentes e ingresar todos los modificados o no, que tratar de ingresar solo los modificados, se enlaza con el módulo clsEmpleadosDat para ingresar a la Base de Datos.

```
Public Sub borrarJustificaciones(strEmp As String, dtFecha As Date)
```

```
    On Error GoTo sigue
```

```
    Set objEmpleadoDat = New clsEmpleadosDat
    objEmpleadoDat.borrarJustificaciones strEmp, dtFecha
    Set objEmpleadoDat = Nothing
```

```
Exit Sub
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

Módulo clsEmpleadosDat

'Declaración de variables.

```
Dim rs As New ADODB.Recordset
```

```
Private strCon As String
```

```
Dim strSql As String
```

'Conexión con el Servidor.

Public Function conectarBDD(strServidor) As Boolean

On Error GoTo sigue

```
strCon = "Provider=SQLOLEDB.1;Persist Security Info=false;User ID=sa;Initial
Catalog=EmpleadosEPN;Data Source=" & strServidor & ";Connect Timeout=30"
con.Open (strCon)
```

conectarBDD = True

Exit Function
sigue:

If Err.Number = -2147467259 Then

MsgBox "Servidor no encontrado, verifique el nombre o la conexión con
éste.", vbCritical, Err.Source

Else

MsgBox Err.Description, vbCritical, Err.Source

End If

conectarBDD = False

End Function

'Función pública "listarEmpleados".

Public Function listarEmpleados() As ADODB.Recordset

On Error GoTo sigue

```
strSql = "SELECT strApellidos," & _
        "strNombres," & _
        "strCodigoEmpleado " & _
"FROM Empleados " & _
"ORDER BY strApellidos," & _
        "strNombres "
```

rs.Open strSql, con, adOpenDynamic, adLockOptimistic

Set listarEmpleados = rs

Exit Function
sigue:

MsgBox Err.Description, vbCritical, Err.Source

End Function

'Función pública "ingresarEmpleado".

Public Sub ingresarEmpleado(strCodigo As String, strNombres As String, strApellidos As
String, _
strDireccion As String, strTelefono1 As String, strCedula As String, _

```
strSexo As String, dtFechaNac As Date, strEstadoCiv As String, dtIngreso As
Date, strTelefono2 As String)
```

```
On Error GoTo sigue
```

```
strSql = "INSERT INTO Empleados VALUES (" & strCodigo & "," & _
    strNombres & "," & _
    strApellidos & "," & _
    strDireccion & "," & _
    strTelefono1 & "," & _
    strCedula & "," & _
    strSexo & "," & _
    Format(dtFechaNac, "yyyy-mm-dd") & "," & _
    strEstadoCiv & "," & _
    Format(dtIngreso, "yyyy-mm-dd") & "," & _
    strTelefono2 & ",")"
```

```
con.Execute (strSql)
```

```
Exit Sub
```

```
sigue:
```

```
If Err.Number = -2147217873 Then
    MsgBox "El código de empleado ya existe.", vbInformation,
    Err.Source
Else
    MsgBox Err.Description & " " & Err.Number, vbCritical, Err.Source
End If
```

```
End Sub
```

'Función pública "eliminarEmpleado".

```
Public Sub eliminarEmpleado(strCodigo As String)
```

```
On Error GoTo sigue
```

```
strSql = "DELETE Empleados " & _
    "WHERE strCodigoEmpleado = " & strCodigo & """
```

```
con.Execute strSql
```

```
Exit Sub
```

```
sigue:
```

```
If Err.Number = -2147217873 Then
    MsgBox "No se puede eliminar el empleado, éste tiene datos
    asignados.", vbInformation, Err.Source
Else
    MsgBox Err.Description & " " & Err.Number, vbCritical, Err.Source
End If
```

```
End Sub
```

‘Función pública “consultarEmpleado”.

```
Public Function consultarEmpleado(strCodigo As String) As ADODB.Recordset
```

```
    On Error GoTo sigue
```

```
    strSql = "SELECT * " & _
            "FROM Empleados " & _
            "WHERE strCodigoEmpleado = " & strCodigo & """
```

```
    rs.Open strSql, con, adOpenDynamic, adLockOptimistic
```

```
    Set consultarEmpleado = rs
```

```
Exit Function
```

```
sigue:
```

```
    MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

‘Función pública “modificarEmpleado”.

```
Public Sub modificarEmpleado(strCodigo As String, strNombres As String, strApellidos As String, _
    strDireccion As String, strTelefono1 As String, strCedula As String, _
    strSexo As String, dtFechaNac As Date, strEstadoCiv As String, dtIngreso As Date, strTelefono2 As String)
```

```
    On Error GoTo sigue
```

```
    strSql = "UPDATE Empleados " & _
            "SET strNombres = " & strNombres & ", " & _
            "strApellidos = " & strApellidos & ", " & _
            "strDireccion = " & strDireccion & ", " & _
            "strTelefono1 = " & strTelefono1 & ", " & _
            "strCedula = " & strCedula & ", " & _
            "strSexo = " & strSexo & ", " & _
            "dtFechaNacimiento = " & Format(dtFechaNac, "yyyy-mm-dd") & ", " & _
            "strEstadoCivil = " & strEstadoCiv & ", " & _
            "dtIngreso = " & Format(dtIngreso, "yyyy-mm-dd") & ", " & _
            "strTelefono2 = " & strTelefono2 & "" & _
            "WHERE strCodigoEmpleado = " & strCodigo & """
```

```
    con.Execute strSql
```

```
Exit Sub
```

```
sigue:
```

```
    MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

‘Función pública “BuscaEmpleadoAp”.

```
Public Function buscaEmpleadoAp(strApellidos As String) As ADODB.Recordset
```

```
    On Error GoTo sigue
```



```

strSql = "SELECT strNombres," & _
         "strApellidos," & _
         "strCodigoEmpleado " & _
         "FROM Empleados " & _
         "WHERE strApellidos LIKE '" & strApellidos & "%'"

rs.Open strSql, con, adOpenDynamic, adLockOptimistic

Set buscaEmpleadoAp = rs

Exit Function
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Function

```

‘Función pública “asignarHorario”.

```

Public Sub asignarHorario(strEmpleado As String, strHorario As String)

    On Error GoTo sigue

    strSql = "INSERT INTO HorarioEmpleado " & _
            "VALUES('" & strEmpleado & "','" & _
            strHorario & "')"

    con.Execute strSql

Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Sub

```

‘Función pública “buscaHorarioEmp”.

```

Public Function buscaHorarioEmp(strEmpleado As String) As ADODB.Recordset

    On Error GoTo sigue

    strSql = "SELECT strCodigoHorario " & _
            "FROM HorarioEmpleado " & _
            "WHERE strCodigoEmpleado = '" & strEmpleado & "'"

    rs.Open strSql, con, adOpenDynamic, adLockOptimistic

    Set buscaHorarioEmp = rs

Exit Function
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Function

```

'Función pública "desAsignarHorario".

```
Public Sub desAsignarHorario(strEmpleado As String)

    On Error GoTo sigue

    strSql = "DELETE HorarioEmpleado " & _
            "WHERE strCodigoEmpleado = " & strEmpleado & ""

    con.Execute strSql

Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Sub
```

'Función pública "buscaHorarioEmpCom".

```
Public Function buscaHorarioEmpCom(strEmpleado As String) As ADODB.Recordset

    On Error GoTo sigue

    strSql = "SELECT * " & _
            "FROM HorarioEmpleado A " & _
            "INNER JOIN Horarios B " & _
            "ON B.strCodigoHorario = A.strCodigoHorario " & _
            "WHERE strCodigoEmpleado = " & strEmpleado & ""

    rs.Open strSql, con, adOpenDynamic, adLockOptimistic

    Set buscaHorarioEmpCom = rs

Exit Function
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Function
```

'Función pública "ingresaVacaciones".

```
Public Sub ingresaVacaciones(strEmpleado As String, dtFechaIn As Date, dtFechaFin As Date)

    On Error GoTo sigue

    strSql = "INSERT INTO Vacaciones " & _
            "VALUES (" & strEmpleado & ", " & _
            Format(dtFechaIn, "yyyy-mm-dd") & ", " & _
            Format(dtFechaFin, "yyyy-mm-dd") & ")"

    con.Execute strSql

Exit Sub
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

    If Err.Number = -2147217873 Then
```


End Sub

'Función pública "eliminarVacaciones".

Public Sub eliminarVacaciones(strEmpleado As String)

On Error GoTo sigue

```
strSql = "DELETE Vacaciones " & _
        "FROM Vacaciones " & _
        "WHERE strCodigoEmpleado = " & strEmpleado & " " & _
        "AND dtFechaRegreso = (SELECT MAX(dtFechaRegreso) " & _
        "FROM Vacaciones " & _
        "WHERE strCodigoEmpleado = " & strEmpleado & ")"
```

con.Execute strSql

Exit Sub

sigue:

MsgBox Err.Description, vbCritical, Err.Source

End Sub

'Función pública "consultaVacaciones".

Public Function consultaVacaciones(strEmpleado As String) As ADODB.Recordset

On Error GoTo sigue

```
strSql = "SELECT CONVERT(VARCHAR,DATEPART(yyyy,dtFechaSalida)) " & _
        "+ ' ' + DATENAME(month,dtFechaSalida) Mes," & _
        "dtFechaSalida," & _
        "dtFechaRegreso," & _
        "DATEDIFF(day,dtFechaSalida,dtFechaRegreso) NumDias " & _
        "FROM Vacaciones " & _
        "WHERE strCodigoEmpleado = " & strEmpleado & " " "
```

rs.Open strSql, con, adOpenDynamic, adLockOptimistic

Set consultaVacaciones = rs

Exit Function

sigue:

MsgBox Err.Description, vbCritical, Err.Source

End Function

'Función pública "buscaFechaRMayor".

Public Function buscaFechaRMayor(strEmpleado As String, blnIn As Boolean) As ADODB.Recordset

On Error GoTo sigue

If blnIn = True Then

```

strSql = "SELECT ISNULL(MAX(dtFechaRegreso),(SELECT
CONVERT(VARCHAR,DATEPART(yyyy,GETDATE()))+ '-' +
CONVERT(VARCHAR,DATEPART(mm,GETDATE()))+ '-' +
CONVERT(VARCHAR,DATEPART(dd,GETDATE())))) FechaR " & _
"FROM Vacaciones " & _
"WHERE strCodigoEmpleado = " & strEmpleado & " "

```

Else

```

strSql = "SELECT ISNULL(MAX(dtFechaRegreso),(SELECT
CONVERT(VARCHAR,DATEPART(yyyy,GETDATE()))+ '-' +
CONVERT(VARCHAR,DATEPART(mm,GETDATE()))+ '-' +
CONVERT(VARCHAR,DATEPART(dd,GETDATE())))) FechaR " & _
"FROM Vacaciones " & _
"WHERE strCodigoEmpleado = " & strEmpleado & " " & _
"AND dtFechaRegreso < (SELECT MAX(dtFechaRegreso) " & _
"FROM Vacaciones " & _
"WHERE strCodigoEmpleado = " & strEmpleado & " " ) "

```

End If

```
rs.Open strSql, con, adOpenDynamic, adLockOptimistic
```

```
Set buscaFechaRMayor = rs
```

Exit Function

sigue:

```
MsgBox Err.Description, vbCritical, Err.Source
```

End Function

‘Función pública “buscaFechaAtraso”.

```
Public Function buscaFechaAtraso() As ADODB.Recordset
```

On Error GoTo sigue

```
strSql = "SELECT CONVERT(VARCHAR,MAX(dtDia),23) Fecha " & _
"FROM Atrasos"
```

```
rs.Open strSql, con, adOpenDynamic, adLockOptimistic
```

```
Set buscaFechaAtraso = rs
```

Exit Function

sigue:

```
MsgBox Err.Description, vbCritical, Err.Source
```

End Function

‘Función pública “buscaFechaRegistro”.

```
Public Function buscaFechasRegistro() As ADODB.Recordset
```

On Error GoTo sigue

```
strSql = "SELECT DISTINCT MIN(CONVERT(VARCHAR,dtFecha,23)) Fecha " & _
        "FROM Registro "
```

```
rs.Open strSql, con, adOpenDynamic, adLockOptimistic
```

```
Set buscaFechasRegistro = rs
```

```
Exit Function
```

```
sigue:
```

```
MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

'Función pública "justificaciones".

```
Public Function justificaciones(strEmp As String, strFinicio As String, strFfin As String) As ADODB.Recordset
```

```
On Error GoTo sigue
```

```
strSql = "SELECT dtDia," & _
        "jornada = " & _
        "CASE DATEPART(dw,dtDia) " & _
        "WHEN 2 THEN " & _
        "CASE strJornadaL " & _
        "WHEN 'Jornada Única 0' THEN 'Única' " & _
        "WHEN 'Jornada Diferenciada 0' THEN 'Diferenciada' " & _
        "WHEN 'Jornada Diferenciada 1' THEN 'Doble' " & _
        "END " & _
        "WHEN 3 THEN " & _
        "CASE strJornadaM " & _
        "WHEN 'Jornada Única 0' THEN 'Única' " & _
        "WHEN 'Jornada Diferenciada 0' THEN 'Diferenciada' " & _
        "WHEN 'Jornada Diferenciada 1' THEN 'Doble' " & _
        "END " & _
        "WHEN 4 THEN " & _
        "CASE strJornadaMr " & _
        "WHEN 'Jornada Única 0' THEN 'Única' " & _
        "WHEN 'Jornada Diferenciada 0' THEN 'Diferenciada' " & _
        "WHEN 'Jornada Diferenciada 1' THEN 'Doble' " & _
        "END "
```

```
strSql = strSql & _
        "WHEN 5 THEN " & _
        "CASE strJornadaJ " & _
        "WHEN 'Jornada Única 0' THEN 'Única' " & _
        "WHEN 'Jornada Diferenciada 0' THEN 'Diferenciada' " & _
        "WHEN 'Jornada Diferenciada 1' THEN 'Doble' " & _
        "END " & _
        "WHEN 6 THEN " & _
        "CASE strJornadaV " & _
        "WHEN 'Jornada Única 0' THEN 'Única' " & _
        "WHEN 'Jornada Diferenciada 0' THEN 'Diferenciada' " & _
        "WHEN 'Jornada Diferenciada 1' THEN 'Doble' " & _
        "END " & _
        "WHEN 7 THEN " & _
        "CASE strJornadaS " & _
        "WHEN 'Jornada Única 0' THEN 'Única' " & _
```

```

        "WHEN 'Jornada Diferenciada 0' THEN 'Diferenciada' " & _
        "WHEN 'Jornada Diferenciada 1' THEN 'Doble' " & _
        "END " & _
        "END," & _
        "ISNULL(intMinEntJ1,0) AEntJ1," & _
        "ISNULL(intMinSalJ1,0) ASalJ1," & _
        "ISNULL(intMinEntJ2,0) AEntJ2," & _
        "ISNULL(intMinSalJ2,0) ASalJ2," & _
        "ISNULL(intE1,0) JE1,"

strSql = strSql & _
        "ISNULL(intS1,0) JS1," & _
        "ISNULL(intE2,0) JE2," & _
        "ISNULL(intS2,0) JS2," & _
        "ISNULL(strMotivo1,") strMotivo1," & _
        "ISNULL(strMotivo2,") strMotivo2 " & _
        "FROM Atrasos A " & _
        "LEFT OUTER JOIN HorarioEmpleado HE " & _
        "ON HE.strCodigoEmpleado = A.strCodigoEmpleado " & _
        "LEFT OUTER JOIN Horarios H " & _
        "ON H.strCodigoHorario = HE.strCodigoHorario " & _
        "LEFT OUTER JOIN Justificaciones J " & _
        "ON J.strCodigoEmpleado = HE.strCodigoEmpleado " & _
        "AND J.dtFecha = A.dtDia " & _
        "WHERE A.strCodigoEmpleado = " & strEmp & " " & _
        "AND dtDia BETWEEN " & Format(strFinicio, "yyyymmdd") & " AND " & _
        Format(strFin, "yyyymmdd") & " "

rs.Open strSql, con, adOpenDynamic, adLockOptimistic

Set justificaciones = rs

Exit Function
sigue:
    MsgBox Err.Description, vbCritical, Err.Source

End Function

```

'Función pública "ingresaJustificaciones".'

```

Public Sub    ingresaJustificaciones(strEmp As String, dtFecha As Date, intEn1 As Integer, _
            intSa1 As Integer, intEn2 As Integer, intSa2 As Integer, _
            strMotivo1 As String, strMotivo2 As String)

    On Error GoTo sigue

    strSql = "INSERT INTO Justificaciones VALUES (" & strEmp & _
            "," & Format(dtFecha, "yyyymmdd") & _
            "," & intEn1 & _
            "," & intSa1 & _
            "," & intEn2 & _
            "," & intSa2 & _
            "," & strMotivo1 & _
            "," & strMotivo2 & ")"

    con.Execute strSql

Exit Sub

```

```

sigue:
    MsgBox Err.Description, vbCritical, Err.Source

```

```
End Sub
```

'Función pública "borrarJustificaciones".

```
Public Sub borrarJustificaciones(strEmp As String, dtFecha As Date)
```

```
    On Error GoTo sigue
```

```
    strSql = "DELETE Justificaciones " & _
            "WHERE strCodigoEmpleado = " & strEmp & " " & _
            "AND dtFecha = " & Format(dtFecha, "yyyymmdd") & ""
```

```
    con.Execute strSql
```

```
Exit Sub
```

```
sigue:
    MsgBox Err.Description, vbCritical, Err.Source
```

```
End Sub
```

Módulo clsHorarios

'Declaración de variables.

```

Dim rs As New ADODB.Recordset
Dim objHorariosDat As clsHorarioDat

```

'Función pública "listaHorarios", extrae la lista de todos los horarios almacenados en la Tabla "Horarios", se enlaza con el módulo clsHorarioDat para ingresar a la Base de Datos.

```
Public Function listaHorarios() As ADODB.Recordset
```

```
    On Error GoTo sigue
```

```

Set objHorariosDat = New clsHorarioDat
Set listaHorarios = objHorariosDat.listaHorarios
Set objHorariosDat = Nothing

```

```
Exit Function
```

```
sigue:
    MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```


'Función pública "ingresarHorario", ingresa un horario creado por el usuario a la Tabla "Horario", se enlaza con el módulo clsHorario para ingresar a la Base de Datos.

```
Public Sub ingresarHorario(strCodigo As String, strDesc As String, _
    strEntrada1L As String, strSalida1L As String, strEntrada2L As String,
    strSalida2L As String, _
    strEntrada1M As String, strSalida1M As String, strEntrada2M As String,
    strSalida2M As String, _
    strEntrada1Mr As String, strSalida1Mr As String, strEntrada2Mr As String,
    strSalida2Mr As String, _
    strEntrada1J As String, strSalida1J As String, strEntrada2J As String,
    strSalida2J As String, _
    strEntrada1V As String, strSalida1V As String, strEntrada2V As String,
    strSalida2V As String, _
    strEntrada1S As String, strSalida1S As String, strEntrada2S As String,
    strSalida2S As String, _
    strJorL As String, strJorM As String, strJorMr As String, _
    strJorJ As String, strJorV As String, strJorS As String)
```

On Error GoTo sigue

```
Set objHorariosDat = New clsHorarioDat
objHorariosDat.ingresarHorario strCodigo, strDesc, _
    strEntrada1L, strSalida1L, strEntrada2L, strSalida2L, _
    strEntrada1M, strSalida1M, strEntrada2M, strSalida2M, _
    strEntrada1Mr, strSalida1Mr, strEntrada2Mr, strSalida2Mr, _
    strEntrada1J, strSalida1J, strEntrada2J, strSalida2J, _
    strEntrada1V, strSalida1V, strEntrada2V, strSalida2V, _
    strEntrada1S, strSalida1S, strEntrada2S, strSalida2S, _
    strJorL, strJorM, strJorMr, strJorJ, strJorV, strJorS
```

Set objHorariosDat = Nothing

Exit Sub

```
sigue:
    MsgBox Err.Description, vbCritical, Err.Source & " H "
```

End Sub

'Función pública "eliminarHorario", se ejecuta cada vez que se elimina un horario de la Tabla "Horarios", se enlaza con el módulo clsHorarioDat para ingresar a la Base de Datos.

```
Public Sub eliminarHorario(strCodigo As String)
```

On Error GoTo sigue

```
Set objHorariosDat = New clsHorarioDat
objHorariosDat.elimnarHorario (strCodigo)
Set objHorariosDat = Nothing
```

Exit Sub

```
sigue:
    MsgBox Err.Description, vbCritical, Err.Source
```

End Sub

'Función pública "consultaHorarios", extrae el horario escogido de la Tabla "Horarios" para su consulta, se enlaza con el módulo clsHorarioDat para ingresar a la Base de Datos.

```
Public Function consultaHorarios(strCodigo As String) As ADODB.Recordset
```

```
    On Error GoTo sigue
```

```
    Set objHorariosDat = New clsHorarioDat
```

```
    Set consultaHorarios = objHorariosDat.consultaHorario(strCodigo)
```

```
    Set objHorariosDat = Nothing
```

```
    Exit Function
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

Módulo clsHorariosDat

'Declaración de variables.

```
Dim rs As New ADODB.Recordset
```

```
Dim strSql As String
```

'Función pública "listaHorarios".

```
Public Function listaHorarios() As ADODB.Recordset
```

```
    On Error GoTo sigue
```

```
    strSql = "SELECT strCodigoHorario," & _
```

```
            "strDescripcion " & _
```

```
            "FROM Horarios " & _
```

```
            "ORDER BY strDescripcion "
```

```
    rs.Open strSql, con, adOpenDynamic, adLockOptimistic
```

```
    Set listaHorarios = rs
```

```
    Exit Function
```

```
    sigue:
```

```
        MsgBox Err.Description, vc, Err.Source
```

```
End Function
```

'Función pública "ingresarHorario".

```
Public Sub ingresarHorario(strCodigo As String, strDesc As String, _
```

```

strEntrada1L As String, strSalida1L As String, strEntrada2L As String, strSalida2L
As String, _
strEntrada1M As String, strSalida1M As String, strEntrada2M As String,
strSalida2M As String, _
strEntrada1Mr As String, strSalida1Mr As String, strEntrada2Mr As String,
strSalida2Mr As String, _
strEntrada1J As String, strSalida1J As String, strEntrada2J As String, strSalida2J
As String, _
strEntrada1V As String, strSalida1V As String, strEntrada2V As String, strSalida2V
As String, _
strEntrada1S As String, strSalida1S As String, strEntrada2S As String, strSalida2S
As String, _
strJorL As String, strJorM As String, strJorMr As String, _
strJorJ As String, strJorV As String, strJorS As String)

```

On Error GoTo sigue

```

strSql = "INSERT INTO Horarios " & _
"VALUES (" & strCodigo & "," & _
strDesc & "," & _
strEntrada1L & "," & _
strSalida1L & "," & _
strEntrada2L & "," & _
strSalida2L & "," & _
strEntrada1M & "," & _
strSalida1M & "," & _
strEntrada2M & "," & _
strSalida2M & "," & _
strEntrada1Mr & "," & _
strSalida1Mr & "," & _
strEntrada2Mr & "," & _
strSalida2Mr & "," & _
strEntrada1J & "," & _
strSalida1J & "," & _
strEntrada2J & "," & _
strSalida2J & "," & _
strEntrada1V & "," & _
strSalida1V & "," & _
strEntrada2V & "," & _
strSalida2V & "," & _
strSql = strSql &
strEntrada1S & "," & _
strSalida1S & "," & _
strEntrada2S & "," & _
strSalida2S & "," & _
strJorL & "," & _
strJorM & "," & _
strJorMr & "," & _
strJorJ & "," & _
strJorV & "," & _
strJorS & ")")

```

con.Execute strSql

Exit Sub

sigue:

```

If Err.Number = -2147217873 Then

```

```

        MsgBox "El código del horario ya existe.", vbInformation,
        Err.Source
    Else
        MsgBox Err.Description & " " & Err.Number, vbCritical, Err.Source
    End If
End Sub

```

'Función pública "eliminarHorario".

```

Public Sub eliminarHorario(strCodigo As String)
    On Error GoTo sigue
    strSql = "DELETE Horarios " & _
        "WHERE strCodigoHorario = " & strCodigo & " "
    con.Execute strSql

Exit Sub
sigue:
    If Err.Number = -2147217873 Then
        MsgBox "No se puede eliminar el horario, ya que se encuentra
        asignado.", vbInformation, Err.Source
    Else
        MsgBox Err.Description, vbCritical, Err.Source & " " & Err.Number
    End If
End Sub

```

'Función pública "consultaHorario".

```

Public Function consultaHorario(strCodigo As String) As ADODB.Recordset
    On Error GoTo sigue
    strSql = "SELECT * " & _
        "FROM Horarios " & _
        "WHERE strCodigoHorario = " & strCodigo & " "
    rs.Open strSql, con, adOpenDynamic, adLockOptimistic

    Set consultaHorario = rs

Exit Function
sigue:
    MsgBox Err.Description, vbCritical, Err.Source
End Function

```

Módulo clsUsuarios

'Declaración de variables.

```
Private strLogin As String
Private strPassword As String
Private strPerfil As String

Dim strSQL As String
Dim objUsuariosDat As clsUsuariosDat
```

'Declaración de los Datos Personales del empleado como variables propias de la clase.

```
Public Property Let login(strLog As String)
```

```
    strLogin = strLog
```

```
End Property
```

```
Public Property Let password(strPas As String)
```

```
    strPassword = strPas
```

```
End Property
```

```
Public Property Let perfil(strPer As String)
```

```
    strPerfil = strPer
```

```
End Property
```

'Función pública "consultaUsuario", busca la existencia del usuario que desea ingresar al SCA en la Tabla "Usuarios", se enlaza con el módulo clsUsuarioDat para ingresar a la Base de Datos.

```
Public Function consultaUsuario(strLogin As String, strPassword As String) As ADODB.Recordset
```

```
    On Error GoTo sigue
```

```
    Set objUsuariosDat = New clsUsuariosDat
    Set consultaUsuario = objUsuariosDat.consultaUsuario(strLogin, strPassword)
    Set objUsuariosDat = Nothing
```

```
    Exit Function
```

```
    sigue:
```

```
        MsgBox Err.Description, vbCritical, Err.Source
```

```
End Function
```

Módulo clsUsuariosDat

'Declaración de Variables.

```
Dim rs As New ADODB.Recordset
```

'Función pública "consultaUsuario".

```
Public Function consultaUsuario(strLogin As String, strPassword As String) As ADODB.Recordset
```

```
On Error GoTo sigue
```

```
strSql = "SELECT * " & _  
"FROM Usuarios " & _  
"WHERE strLogin = '" & strLogin & "' " & _  
"AND strPassword = '" & strPassword & "' "
```

```
rs.Open strSql, con, adOpenDynamic, adLockOptimistic
```

```
Set consultaUsuario = rs
```

```
Exit Function
```

```
sigue:
```

```
MsgBox Err.Description, vbCritical, Err.Source & " " & Err.Number
```

```
End Function
```

SOFTWARE DEL CLIENTE O TERMINAL

Código cargado en la forma fmrReloj

'Declaración de variables.

```
Dim rs As New ADODB.Recordset
```

'Subrutina del botón "Registrar", llama a la forma frmAcceso1n.

```
Private Sub btnRegistrar_Click()
```

```
frmAcceso1n.Show
```

```
End Sub
```

'Subrutina que se ejecuta en el momento que se carga la forma.

```
Private Sub Form_Load()
```

```
    'Abre la conexión con el Servidor.  
    con.Open (strCon)
```

```
    'Da formato a la fecha del reloj.  
    Label1.Caption = UCase(Format(Date, "ddd dd " & Chr(13) & " MMM-yyyy"))
```

```
End Sub
```

'Timer del Reloj.

```
Private Sub Timer1_Timer()
```

```
    If lblTime.Caption <> CStr(Time) Then  
        lblTime.Caption = Time  
    End If
```

```
End Sub
```

Código cargado en la forma frmAcceso1n

Utiliza el único módulo de este software: Module1.

'Declaración de variables.

```
    Dim rs As New ADODB.Recordset  
  
    Dim strHor As String  
    Dim strSql As String  
    Dim strReg As String  
    Dim strDe As String  
    Dim strNombre As String
```

'Subrutina del botón "Aceptar", es activado automáticamente por el software de Biologon 3.0: "BioShield", esta activación solo se produce en el momento que Biologon 3.0 autentique la huella dactilar del empleado.

```
Private Sub btnAceptar_Click()
```

```
    verificacionRegistro    'Llama a la subrutina "verificacionRegistro".
```

```
End Sub
```

'Subrutina "buscaEntrada", busca la primera entrada del empleado para ese día en su respectivo horario.

Private Sub buscaEntrada()

On Error GoTo sigue

```
strSql = "SELECT Entrada = " & _
        "CASE DATEPART(dw,GETDATE()) " & _
        "WHEN 2 THEN strEntrada1L " & _
        "WHEN 3 THEN strEntrada1M " & _
        "WHEN 4 THEN strEntrada1Mr " & _
        "WHEN 5 THEN strEntrada1J " & _
        "WHEN 6 THEN strEntrada1V " & _
        "WHEN 7 THEN strEntrada1S " & _
        "END " & _
        "FROM HorarioEmpleado HE " & _
        "INNER JOIN Horarios H " & _
        "ON H.strCodigoHorario = HE.strCodigoHorario " & _
        "WHERE HE.strCodigoEmpleado = " & txtCodigo.Text & "" "
```

rs.Open strSql, con, adOpenDynamic, adLockBatchOptimistic

While Not rs.EOF

```
strHor = rs!Entrada
rs.MoveNext
```

Wend

rs.Close

Exit Sub

sigue:

```
MsgBox Err.Description, vbInformation, Me.Caption
```

End Sub

'Subrutina "verificacionRegistro", realiza la validación del registro del empleado para ese día.

Private Sub verificacionRegistro()

On Error GoTo sigue

'Para la fecha actual extrae de la Base de Datos del SCA el tipo de jornada y el número de registros que debe ingresar el empleado.

```
strSql = "SELECT Total = " & _
        "CASE DATEPART(dw,GETDATE()) " & _
        "WHEN 2 THEN " & _
        "CASE strJornadaL " & _
        "WHEN 'Jornada Diferenciada 1' THEN 4 " & _
        "WHEN 'No ingresar horario 0' THEN 0 " & _
```



```

"ELSE 2 " & _
"END " & _
"WHEN 3 THEN " & _
    "CASE strJornadaM " & _
        "WHEN 'Jornada Diferenciada 1' THEN 4 " & _
        "WHEN 'No ingresar horario 0' THEN 0 " & _
        "ELSE 2 " & _
    "END " & _
"WHEN 4 THEN " & _
    "CASE strJornadaMr " & _
        "WHEN 'Jornada Diferenciada 1' THEN 4 "
strSql = strSql & _
        "WHEN 'No ingresar horario 0' THEN 0 " & _
        "ELSE 2 " & _
    "END " & _
"WHEN 5 THEN " & _
    "CASE strJornadaJ " & _
        "WHEN 'Jornada Diferenciada 1' THEN 4 " & _
        "WHEN 'No ingresar horario 0' THEN 0 " & _
        "ELSE 2 " & _
    "END "
strSql = strSql & _
"WHEN 6 THEN " & _
    "CASE strJornadaV " & _
        "WHEN 'Jornada Diferenciada 1' THEN 4 " & _
        "WHEN 'No ingresar horario 0' THEN 0 " & _
        "ELSE 2 " & _
    "END " & _
"WHEN 7 THEN " & _
    "CASE strJornadaS " & _
        "WHEN 'Jornada Diferenciada 1' THEN 4 " & _
        "WHEN 'No ingresar horario 0' THEN 0 " & _
        "ELSE 2 " & _
    "END " & _
"END," & _
"strApellidos + strNombres strNombres "
strSql = strSql & _
"FROM HorarioEmpleado HE " & _
"INNER JOIN Horarios H " & _
"ON H.strCodigoHorario = HE.strCodigoHorario " & _
"INNER JOIN Empleados E " & _
"ON E.strCodigoEmpleado = HE.strCodigoEmpleado " & _
"WHERE HE.strCodigoEmpleado = " & txtCodigo.Text & ""

```

rs.Open strSql, con, adOpenDynamic, adLockOptimistic

While Not rs.EOF

```

    strDe = rs!Total
    strNombre = rs!strNombres
    rs.MoveNext

```

Wend

rs.Close

'Para la fecha actual extrae de la Tabla "Registro" el número de veces que el empleado se ha registrado.

```
strSql = "SELECT COUNT(*) actual " & _
        "FROM Registro " & _
        "WHERE strCodigoEmpleado = " & txtCodigo.Text & " " & _
        "AND CONVERT(VARCHAR,dtFecha,23) = " & Format(Date, "yyyy-mm-dd") & " "
```

```
rs.Open strSql, con, adOpenDynamic, adLockOptimistic
```

```
While Not rs.EOF
```

```
    strReg = rs!actual
    rs.MoveNext
```

```
Wend
```

```
rs.Close
```

'Si el empleado no tenia que registrarse ese día, se le notifica.

```
If strDe = "0" Then
```

```
    MsgBox strNombre & " Este día no le corresponde registrarse.", vbInformation,
    Me.Caption
    Unload Me
    Exit Sub
```

```
End If
```

'Si tiene que registrarse, se evalúa cuantas veces se ha registrado en la fecha actual.

```
If strReg = "0" Then
```

```
    buscaEntrada 'Llama a la función "buscaEntrada"
```

'Si es la primera entrada, se verifica que el registro no sea antes de la media hora a la entrada.

```
If DateDiff("n", "00:30:00", strHor & ":00") > DateDiff("n", "00:00:00", Time)
Then
```

```
    MsgBox "No puede registrarse sino hasta una media hora antes de su
    entrada.", vbInformation, Me.Caption
    Unload Me
    Exit Sub
```

```
End If
```

```
End If
```

'Se verifica que el número de veces que ha registrado el empleado no sobrepase el número de registros que debe registrarse.

```
If strReg < strDe Then
```

```
    ingresaRegistro 'Llama a la función "ingresaRegistro".
    MsgBox Cint(strReg) + 1 & " de " & strDe & " registros", vbInformation, strNombre
    Unload Me
```

```
Else
```

ANEXO E

Manual de Instalación y Operación del
Sistema de Control de Asistencia

ÍNDICE

1. INSTALACIÓN Y CONFIGURACIÓN DE <i>BIOLOGON 3.0</i>	E-2
1.1. Descripción de producto.....	E-2
1.1.1 Funciones de <i>Biologon 3.0</i>	E-2
1.2. Instalación del <i>Hardware</i>	E-3
1.3. Instalación del <i>Software</i>	E-4
1.3.1 Configuración del lector de huellas dactilares	E-6
1.3.2 Instalación del <i>software</i> Cliente de <i>Biologon</i>	E-8
1.3.3 Instalación del <i>software</i> del Servidor.....	E-12
1.3.3.1 Extender el esquema de <i>Active Directory</i>	E-12
1.3.3.2 Tipo de controlador de <i>BioLogon</i>	E-13
1.4. Creación de cuentas biométricas de usuario de red	E-13
1.4.1 Adición de un nuevo usuario al servidor de <i>BioLogon</i>	E-14
1.5. Uso de <i>Bioshield</i>	E-18
1.5.1 Banco de contraseñas de <i>Bioshield</i>	E-19
2. MANUAL DE OPERACIÓN DEL SCA.....	E-23
2.1. Descripción de Sistema	E-23
2.2. Pantalla de ingreso al SCA	E-23
2.3. Manejo del Sistema	E-24
2.3.1 Menú Archivo.....	E-26
2.3.2 Menú Administración	E-26
2.3.2.1 Empleados	E-26
2.3.2.2 Horarios.....	E-29
2.3.2.2.1 Administración	E-29
2.3.2.2.2 Asignación.....	E-34
2.3.2.3 Justificaciones.....	E-36
2.3.2.4 Vacaciones.....	E-39
2.3.3 Menú Registro	E-41
2.3.4 Menú Consultas	E-41
2.3.5 Menú Reportes	E-42
2.3.6 Menú Usuarios	E-43

- Lector de huellas dactilares de puerto paralelo.
- Lector de huellas dactilares de tarjeta PC.
- Lector de huellas dactilares USB.

El periférico que se utilizará en el Sistema de Control de Asistencia (SCA) es el USB.

Instalación del lector de huellas dactilares USB:

- Conexión del lector de huellas dactilares USB al equipo. Inserte el lector de huellas dactilares USB al puerto USB del equipo.

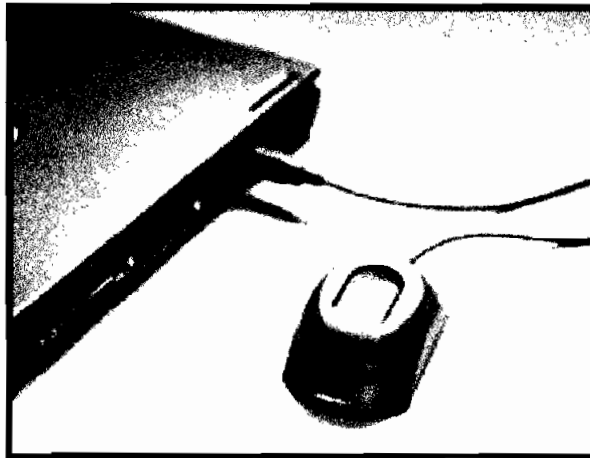


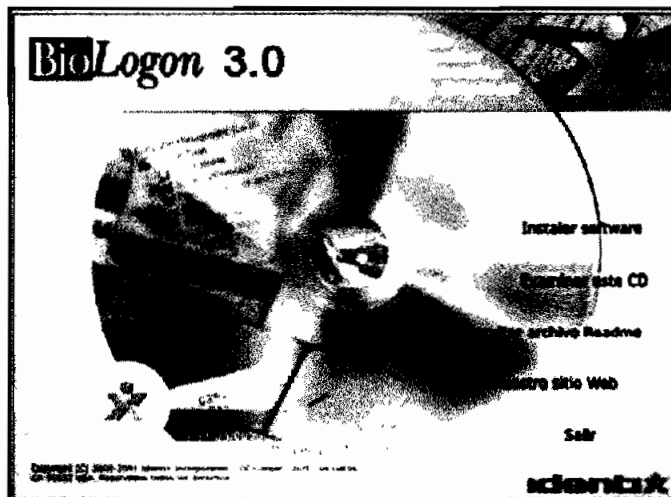
Figura E.1: Conexión del lector de huellas dactilares USB.

La instalación del lector de huellas dactilares USB no requiere más pasos.

1.3. Instalación del *Software*

A continuación podrá instalar el *software* del programa *BioLogon 3.0* y del lector de huellas dactilares en su sistema operativo de *Windows*.

1. Introduzca el CD del *software* de *BioLogon 3.0*. Tras una breve animación, aparecerá el cuadro de diálogo inicial (ver figura E.2 (a)). Si no se inicia el programa de instalación de forma automática, ejecute *setup.exe* desde la carpeta raíz del CD o el soporte que esté utilizando.



(a)



(b)

Figura E.2: (a) Cuadro de instalación de *BioLogon 3.0*; (b) Verificación de componentes.

2. Pulse **Instalar** para instalar los componentes de *BioLogon 3.0*.
3. Compruebe que va a instalar los componentes de *software* adecuados. En la instalación inicial, el sistema muestra dos marcas de verificación. Pulse cada una de ellas, si lo considera necesario (ver figura E.2 (b))
4. Pulse "**Instalar**" para proceder a la instalación de todos los productos seleccionados. Aparecerá el cuadro de diálogo del Asistente de instalación.

Sólo puede instalar el *software* si el programa de instalación no detecta ninguna copia previa en el disco. Si ha marcado la casilla de verificación *BioLogon* y el sistema detecta *software* de *BioLogon 3.0* en su PC, puede modificar, reparar o

quitar los componentes de *BioLogon* 3.0. Si ha marcado la casilla de verificación “Lector de huellas dactilares” y el sistema detecta un lector de huellas dactilares en su PC, puede modificar o quitar el *software*.

NOTA: Si no ha marcado ninguno de los dos productos y pulsa instalar, el programa de instalación finalizará.

1.3.1. Configuración del lector de huellas dactilares

Para configurar el lector de huellas dactilares:

1. Aparece el cuadro de diálogo Bienvenido. Pulse **Siguiente**. Siga las instrucciones y escriba la información necesaria en los cuadros de diálogo del Asistente.
2. Aparece el cuadro de diálogo Contrato de licencia de *software*:
 - a. Seleccione el botón de opción “Acepto los términos del contrato de licencia”. Pulse **Siguiente**.
3. Aparece el cuadro de diálogo Carpeta de destino:
 - a. Compruebe que la carpeta que se muestra es la ubicación en la que desea guardar el *software* del lector de huellas dactilares de *Biologon* 3.0. Si es así, pulse **Siguiente**; referirse a la figura E.3.
 - b. Si la carpeta que se muestra no es la que desea, pulse **Cambiar**. Aparece el cuadro de diálogo “Cambiar” la carpeta de destino actual. Seleccione la carpeta de destino. Pulse **Aceptar**. El cuadro de diálogo Carpeta de destino vuelve a aparecer. Pulse **Siguiente**.
4. Aparece el cuadro de diálogo Instalación de *Fingerprint Reader Components*:
 - a. Seleccione la interfaz del lector de huellas dactilares, en este caso DFR-200 (ver figura E.4).
 - b. Compruebe que ha seleccionado el lector de huellas dactilares adecuado. Pulse **Siguiente**.

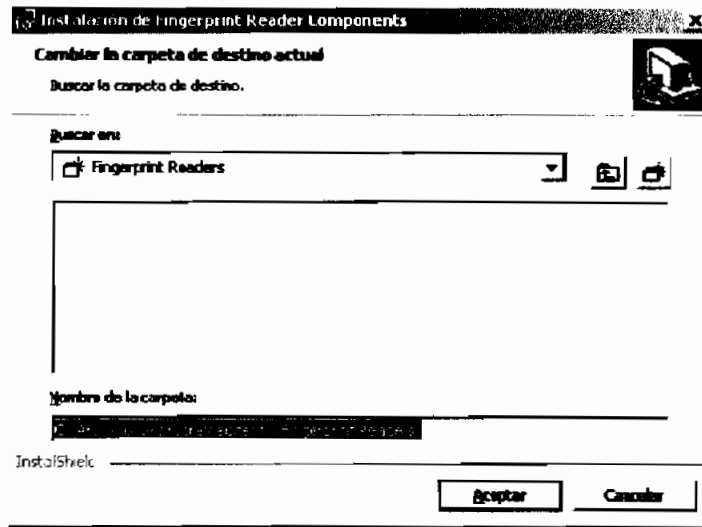


Figura E.3: Ubicación de la carpeta que guarda el *software* del lector

5. Aparece el cuadro de diálogo Preparado para instalar el programa. Pulse **Instalar** para dar comienzo al proceso de instalación.
 - a. Si desea revisar la configuración de la instalación o realizar algún cambio, pulse el botón **Atrás**.
 - b. También puede pulsar **Cancelar** para salir del Asistente.
 - c. Se instala el programa Componentes del lector de huellas dactilares. Este proceso puede durar varios minutos. El sistema muestra el progreso en la barra de estado. El sistema busca y ubica el lector de huellas dactilares (si está conectado).

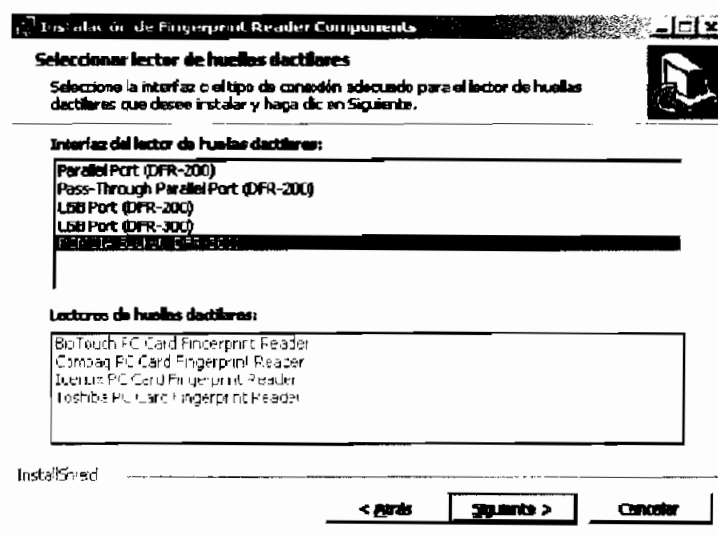


Figura E.4: Cuadro de selección del tipo de interfaz.

6. Siga las instrucciones del Asistente de nuevo hardware encontrado.
 - a. Pulse **Siguiente** para instalarlo y pulse **Finalizar**.
 - b. Pulse **Siguiente** para configurar el lector de huellas dactilares en ese momento.

7. Ajuste el lector de huellas dactilares.
 - a. Coloque el dedo sobre el lector de huellas dactilares.
 - b. Con el dedo quieto, pulse el botón **Ajustar**. El lector de huellas dactilares detecta su huella dactilar y muestra una lectura de la presencia de la misma.
 - c. Mantenga el dedo en el lector de huellas dactilares hasta que se resalte el botón **Siguiente**, y púselo.

8. Pulse **Finalizar** para salir del Asistente cuando haya instalado los Componentes del lector de huellas dactilares.
9. Ahora vaya a "Instalación de *BioLogon 3.0*"

1.3.2. Instalación del *software* Cliente de *Biologon*

Este *software* se instalará en los terminales del SCA.

1. Aparece el cuadro de diálogo Bienvenido. Pulse **Siguiente**. Siga las instrucciones y escriba la información necesaria en los cuadros de diálogo del Asistente.
2. Aparece el cuadro de diálogo Contrato de licencia de *software*:
 - a. Seleccione el botón de opción "Acepto los términos del contrato de licencia". Pulse **Siguiente**.
3. Aparece el cuadro de diálogo Información del cliente (ver figura E.5). Escriba lo siguiente:
 - i. Nombre de usuario
 - ii. Organización
 - iii. Clave del CD suministrada con el *software*
 - iv. Pulse **Siguiente**.

4. Aparece el cuadro de diálogo Carpeta destino (ver figura E.6):
 - a. Compruebe que la carpeta que se muestra es la ubicación en la que desea guardar el *software* de *BioLogon* 3.0. Si es así, pulse **Siguiente**.
 - b. Si la carpeta que se muestra no es la que desea, pulse **Cambiar**. Aparece el cuadro de diálogo Cambiar la carpeta de destino actual. Seleccione la carpeta de destino. Pulse **Aceptar**. El cuadro de diálogo Carpeta destino vuelve a aparecer. Pulse **Siguiente**.

5. Aparece la pantalla de la figura E.7, seleccione la opción Cliente de *Biologon* y pulse **Siguiente**.

BioLogon Security System 3

Información del cliente

Introduzca su información.

Nombre de usuario:
Identix User

Organización:
Identix

Clave del ID:
[] - [] - [] - []

InstallShield

< Atrás Siguiente > Cancelar

Figura E.5: Cuadro de diálogo Información del cliente.

6. Aparece el cuadro de diálogo Tipo de instalación (ver figura E.8). Seleccione la opción **Personalizada** para agregar la herramienta de *BioShield*; pulse **Siguiente**.

7. Aparece el cuadro de selección de funciones (ver figura E.9).
 - a. Significado de los símbolos:
 - i. Una X roja indica que la función no se instalará.

- ii. Un icono de un disco duro indica que sólo se instalará la función.
 - iii. Un icono de dos discos duros indica que se instalará las funciones y las subfunciones.
- b. Pulse el icono “*Biologon Applications*”, seleccione “*BioShield*” y coloque la opción de dos discos duros para que se instale todo este componente.

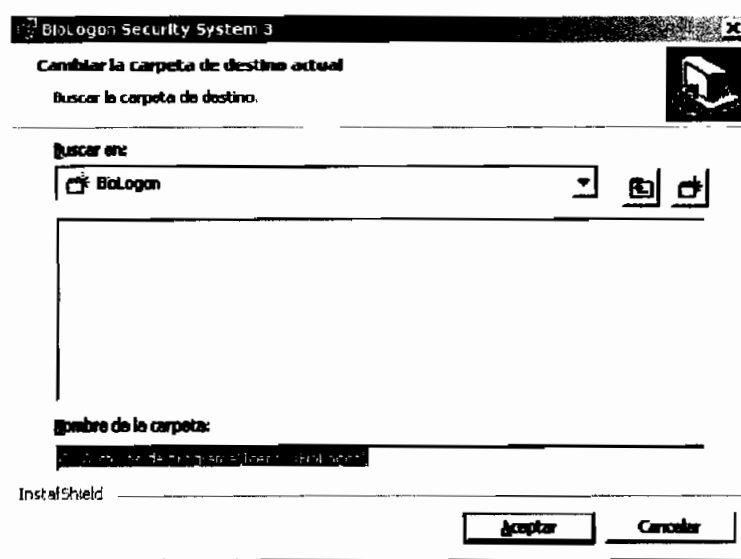


Figura E.6: Cuadro de diálogo Carpeta destino.

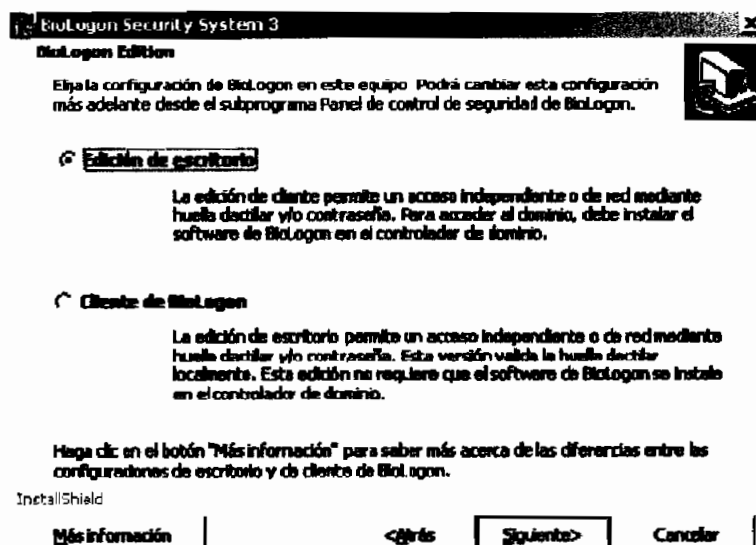


Figura E.7: Ventana de selección del tipo de *software* para PCs personales.

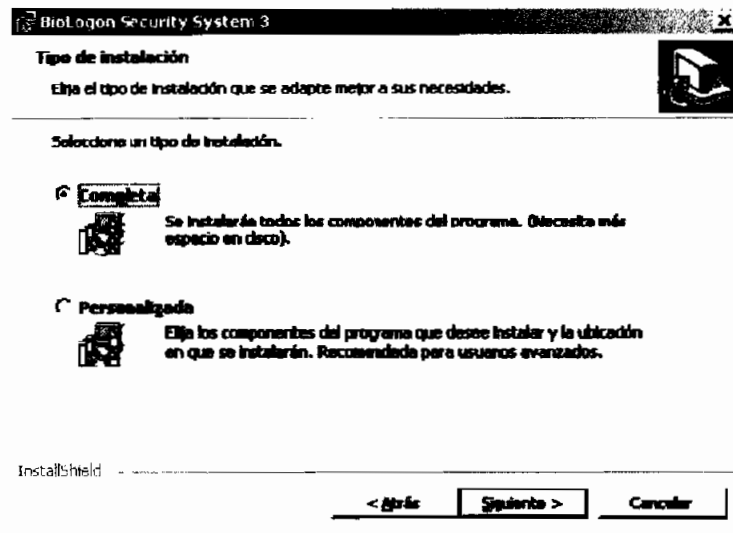


Figura E.8: Ventana de selección del tipo de instalación.

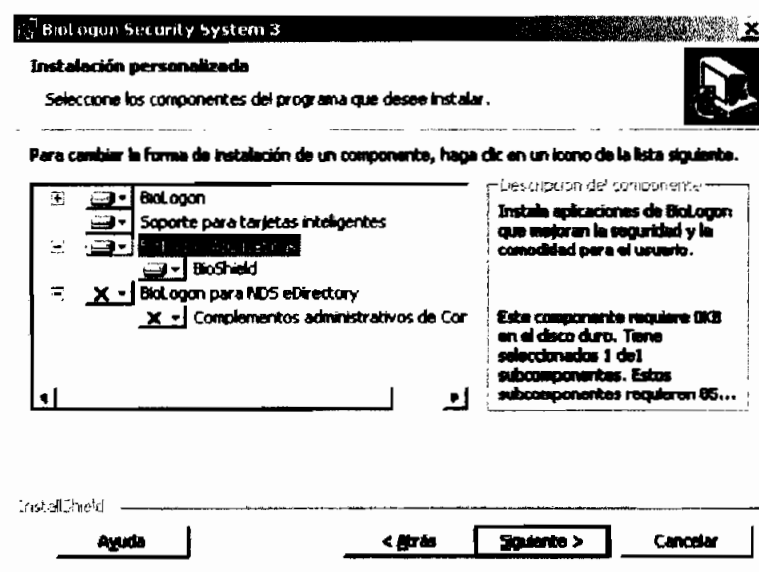


Figura E.9: Cuadro de selección de funciones.

8. Pulse **Siguiente**.
9. Pulse el botón **Instalar**. El sistema instala los archivos que haya seleccionado. Por favor, espere mientras el sistema instala *BioLogon 3.0*. Este proceso puede durar varios minutos. El sistema muestra el progreso en la barra de estado.
10. Pulse **Finalizar** para salir del asistente.
11. Pulse **Sí** para cerrar y reiniciar el equipo.

1.3.3. Instalación del *software* del Servidor

El programa de instalación de *BioLogon 3.0* detecta automáticamente la configuración del servidor. En esta instalación también deberá escoger la instalación Personalizada para agregar los componentes de *BioShield*.

Los siguientes cuadros de diálogo que afectan a la instalación son:

1.3.3.1. Extender el esquema de *Active Directory*

El cuadro de diálogo Extender el esquema de *Active Directory* aparece si *BioLogon 3.0* debe extender el esquema de *Active Directory* (ver figura E.10). Pulse **Siguiente** para extender el esquema de *Active Directory*.

El esquema de *Active Directory* define la estructura de la base de datos de *Active Directory*.

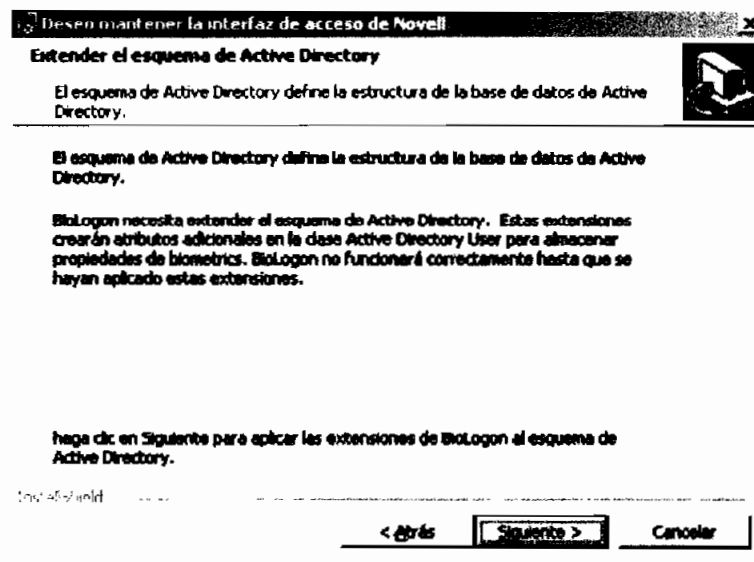


Figura E.10: Cuadro de diálogo para extender el esquema de *Active Directory*.

Estas extensiones agregarán atributos que almacenan propiedades de *Biometrics* en la clase *Active Directory User*. *BioLogon 3.0* no funcionará correctamente hasta que se hayan aplicado estas extensiones. El cuadro de diálogo Esquema de

Active Directory sólo aparece en la primera instalación del servidor de *BioLogon* 3.0 en el dominio.

1.3.3.2. Tipo de controlador de *BioLogon*

El programa de instalación de *BioLogon* 3.0 le preguntará si se trata del primer controlador de dominio que tendrá instalado el Sistema de seguridad de *BioLogon*. Seleccione la opción "Instalación del primer servidor de *Biologon* en el dominio" y pulse **Siguiente** y proceda a finalizar la instalación (ver figura E.11).

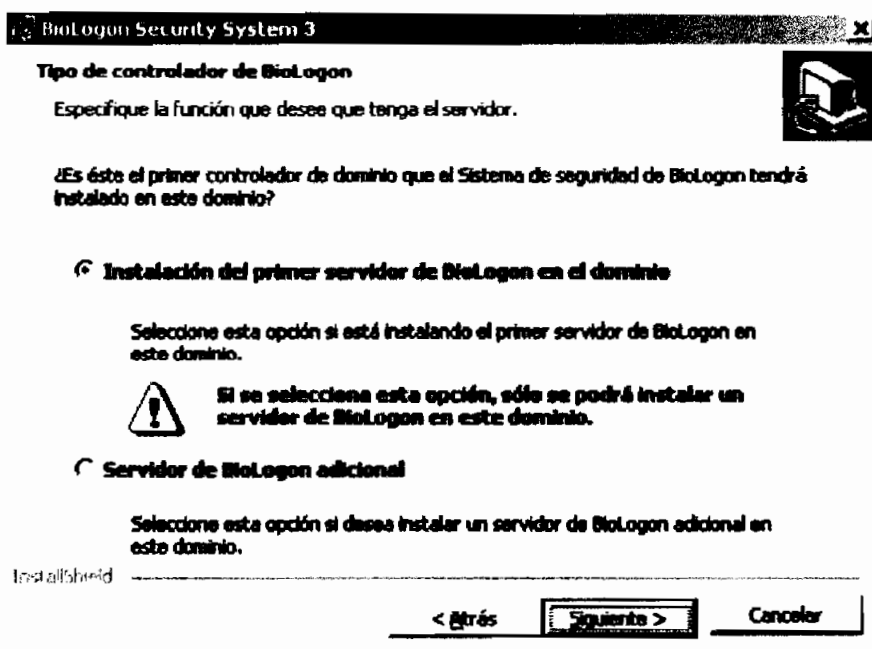


Figura E.11: Cuadro de selección del tipo de controlador de *Biologon*.

1.4. Creación de cuentas biométricas de usuario de red

Para agregar un empleado al dominio de *Biologon* se debe primero crear una cuenta de usuario de red con el respectivo *password* (código del empleado). Es posible que surjan restricciones, bien en el dominio o en el equipo local, que le impidan acceder al cuadro de diálogo **Validación** durante la creación de la cuenta de usuario. Si tiene problemas en este paso, compruebe primero que puede acceder al dominio desde la estación de trabajo. Al utilizar la edición *Cliente/Servidor* de *BioLogon*, las normativas de autenticación y la información biométrica se almacenan en los controladores de dominio de *Windows*, en la base

de datos del Administrador de cuentas de seguridad (SAM) o dentro de *Active Directory*.

1.4.1. Adición de un nuevo usuario al servidor de *BioLogon*

Para agregar un nuevo usuario de red para la autenticación biométrica de escritorio en sistemas operativos *Windows 2000 Server* o *Advanced Server*, haga lo siguiente:

1. Haga clic con el botón derecho del ratón en el icono:



Seguridad de *BioLogon*

2. Seleccione Gestor de usuarios en el menú emergente. El sistema muestra el cuadro de diálogo Usuarios y equipos de *Active Directory* (ver figura E.12).

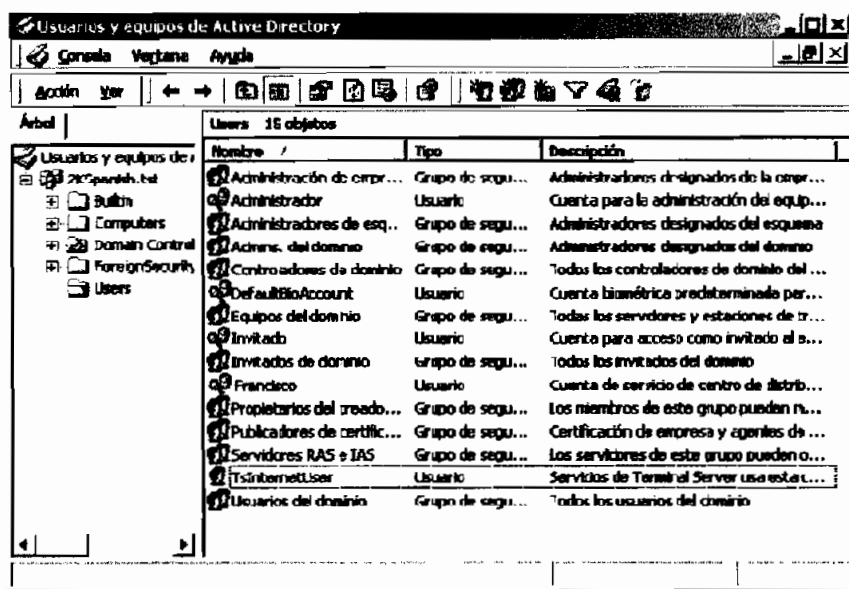
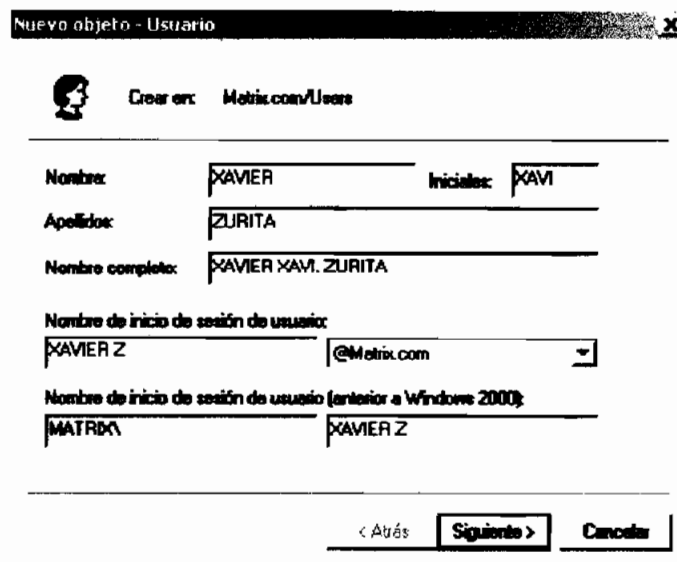


Figura E.12: Cuadro de diálogo Usuarios y equipos de *Active Directory*

3. Haga clic con el botón derecho del ratón en la carpeta *Users*. Seleccione Nuevo usuario. Aparece el cuadro de diálogo Nuevo Usuario (ver figura E.13). Escriba la siguiente información:

- o Nombre

- Iniciales
 - Apellidos
 - Nombre completo
 - Nombre de inicio de sesión de usuario (anterior a *Windows 2000*)
4. Pulse **Siguiente**. Aparecerá el siguiente cuadro de diálogo (ver figura E.14). Introduzca y confirme la contraseña de usuario que debe ser **IGUAL AL CÓDIGO DEL EMPLEADO** en el SCA. Marque **Registrar datos biométricos** para este usuario y pulse **Siguiente**.



Nuevo objeto - Usuario

Crear en: Metrix.com/Users

Nombre: XAVIER Iniciales: XAVI

Apellidos: ZURITA

Nombre completo: XAVIER XAVI. ZURITA

Nombre de inicio de sesión de usuario:
XAVIER Z @Metrix.com

Nombre de inicio de sesión de usuario (anterior a Windows 2000):
MATRBA XAVIER Z

< Atrás **Siguiente >** Cancelar

Figura E.13: Cuadro de diálogo Nuevo Usuario.

5. Aparece el cuadro de diálogo Asistente de inscripción de huellas digitales (ver figura E.15). Seleccione la casilla “Configurar el lector de huellas dactilares antes de la inscripción” para ajustar la configuración del lector. Esta opción le permite ajustar el lector de huellas dactilares antes de la inscripción, más adelante en este procedimiento. Pulse **Siguiente**.
6. Aparece el cuadro de diálogo Normativa de usuario (ver figura E.16). Seleccione el botón de opción Usuario de *Biometrics*. Seleccione la

normativa biométrica de usuario, en la lista desplegable, “Acceso con huella digital o contraseña”.

Nuevo objeto - Usuario

Crear en: Matrix.com/Users

Contraseña: _____

Confirmar contraseña: _____

El usuario debe cambiar la contraseña en el siguiente inicio de sesión

El usuario no puede cambiar la contraseña

La contraseña nunca caduca

Cuenta deshabilitada

Registrar datos biométricos para este usuario

< Atrás **Siguiente** > Cancelar

Figura E.14: Cuadro de diálogo de contraseña de usuario.

Asistente de inscripción de huellas digitales

Bienvenido a Asistente de inscripción de huellas digitales

Este asistente le ayudará a configurar una nueva cuenta de Biometric para el usuario.

Puede utilizar el asistente para hacer lo siguiente:

- Configurar la normativa de acceso del usuario que determina el modo en que el usuario accede al sistema.
- Inscribir las huellas digitales del usuario en la base de datos de Biometric o en la tarjeta inteligente (opcional).

Configurar el lector de huellas dactilares antes de la inscripción

Mostrar esta página la próxima vez que utilice el asistente

Cancelar < Atrás **Siguiente** > Enviar

Figura E.15: Cuadro de diálogo Asistente de inscripción de huellas digitales

7. Ajuste el lector de huellas dactilares (ver figura E.17). Coloque el dedo sobre el lector de huellas dactilares. Con el dedo quieto, pulse el botón Ajustar. El lector de huellas dactilares detecta su huella dactilar y muestra una lectura de la presencia de la misma. Mantenga el dedo en el lector de huellas dactilares hasta que se resalte el botón **Siguiente**. Este paso solo aparece en la primera vez que se inscribe una huella dactilar.

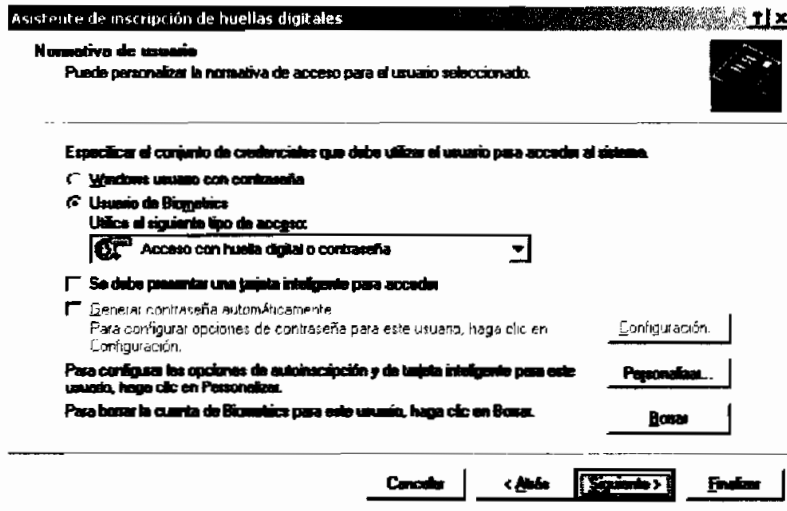


Figura E.16: Cuadro de diálogo Normativa de usuario.

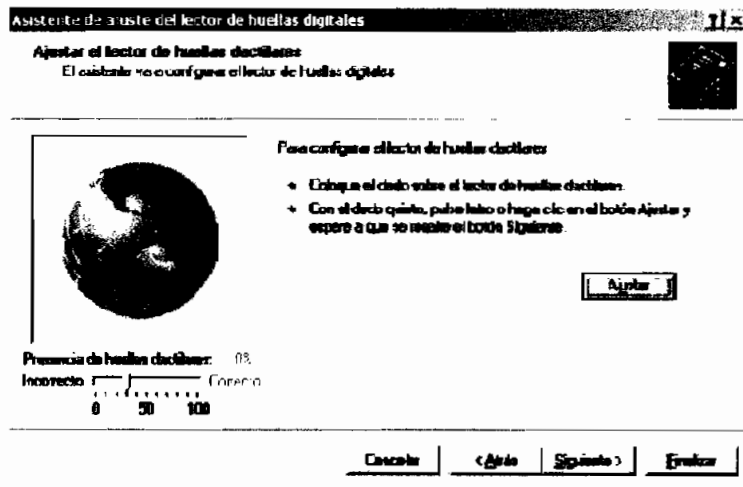


Figura E.17: Asistente de ajuste del lector de huellas dactilares.

8. Inscripción de la huella dactilar. Escoja el dedo que desea inscribir (ver figura E.18).
 - o Las marcas de las casillas indican que el dedo se ha inscrito.
 - o Se puede inscribir de un dedo.
9. Coloque y mantenga el dedo en la lente del lector de huellas dactilares y espere a que concluya la inscripción (ver figura E.19). Siga las instrucciones del cuadro de diálogo. El sistema le pedirá que coloque el

dedo en la posición correcta sobre el lector (en el centro del lector). Si la inscripción terminó con éxito pulse **Finalizar**.

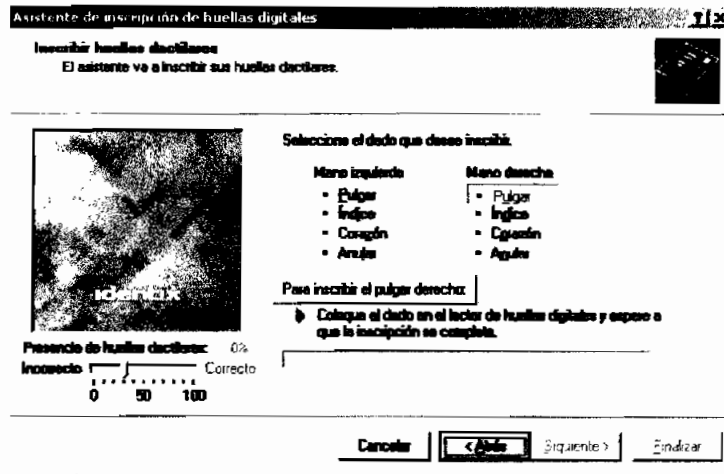


Figura E.18: Asistente de inscripción de huellas digitales.

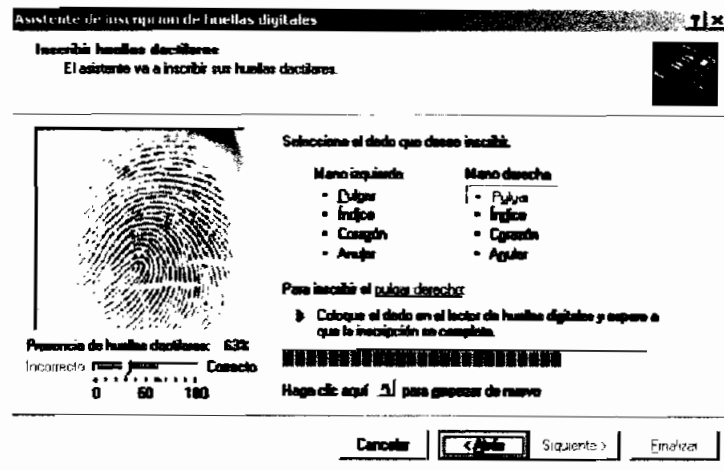




Figura E.19: Proceso de inscripción de la huella dactilar.

1.5. Uso de *Bioshield*

Esta opción se utilizará en cada Terminal del SCA.

Cuando se está ejecutando la aplicación *BioShield*, el icono de *BioShield* aparece en la bandeja del sistema, en la barra de tareas del escritorio. La caja de herramientas de *BioShield* está disponible cuando se está ejecutando la aplicación *BioShield* y debe activarla para verla en pantalla.

Para mostrar la caja de herramientas de *BioShield*:

1. Haga doble clic en el icono *BioShield*  situado en la bandeja del sistema del escritorio o bien
 - Acceda al cuadro de diálogo "Propiedades Seguridad de *BioLogon*", si no se muestra el icono.
 - a. Haga clic con el botón derecho del ratón en el icono  (Seguridad de *BioLogon*).
 - b. Seleccione *BioShield* en el menú emergente.

2. Seleccione Mostrar Caja de herramientas. Aparecerá la caja de herramientas de *BioShield*:

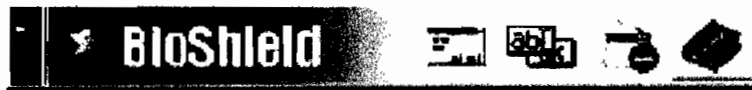


Figura E.20: Barra de herramientas del *Bioshield*.

1.5.1. Banco de contraseñas de *Bioshield*

La caja de herramientas de *BioShield* contiene cuatro botones situados a la derecha: Banco de Contraseñas, Aceleración de acciones, Protección de aplicaciones y la Ayuda del Usuario.

En este caso se utilizará el "Banco de Contraseñas"

El Banco de contraseñas le permite utilizar las huellas dactilares para iniciar una sesión de forma rápida en una aplicación, un cuadro de diálogo o una página Web.

Libera la información de autenticación de acceso de nombre de usuario y contraseña colocando el dedo en el lector de huellas dactilares.

Antes de acceder, debe introducir la información de autenticación en la base de datos del Banco de contraseñas de *BioShield*. Cuando intente acceder al cuadro

de diálogo de acceso, aparecerá el cuadro de diálogo de validación Banco de contraseñas de *BioShield*, en el caso de que haya registrado el cuadro de diálogo en la base de datos del Banco de contraseñas de *BioShield*. Cuando haya conseguido validar las credenciales de *Biometrics*, se enviará la información de acceso y podrá acceder al cuadro de la aplicación.

Para validar correctamente las credenciales utilizando *BioShield* debe:

- Enviar las credenciales de usuario válidas que requiera su normativa de usuario de *Biometrics*.
- Utilizar la misma cuenta de usuario que registró en las propiedades de *BioShield* para la operación de *BioShield* deseada.

Para registrar la aplicación del Terminal del SCA en la base de datos de *BioShield* haga lo siguiente:

1. Inicie la aplicación y pulse el botón “Registrar” de la pantalla principal (Ver figura E.21).
2. Pulse el primer icono de la barra de herramientas de *Bioshield* y arrástrelo a la ventana de la aplicación, en el momento que aparezca un recuadro rojo deje de pulsar. A continuación aparecerá la ventana del Banco de contraseñas; escoja la opción “Utilizar mis credenciales de *Windows*” y pulse **Aceptar**. (ver figura E.22).
3. Aparece la pantalla de la figura E.23. La opción “Interfaz de acceso biométrico” debe estar seleccionada, pulse el recuadro pequeño de la opción para deshabilitarla y pulse **Aplicar** y a continuación **Aceptar**.
4. La aplicación queda protegida y se podrá ingresar solamente con un permiso de acceso perteneciente al usuario que se encuentre en la sesión de *Windows* del Terminal. Ahora se debe ingresar los demás empleados. Haga clic derecho con el ratón en el icono de *bioshield* y escoja la opción propiedades (ver figura E.24).



Figura E.21: Interfaz de inscripción para los registros de asistencia

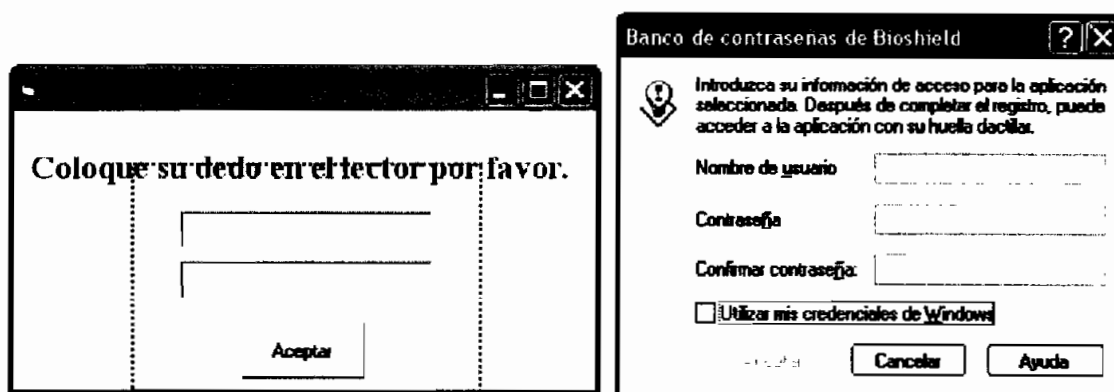


Figura E.22: Protección de la pantalla de captura del código del empleado utilizando el Banco de Contraseñas de *Bioshield*.

5. Aparece una ventana como la de la figura E.25, seleccione la opción **Copiar**. A continuación se desplegará una cuadro con todos los empleados ingresados en el servidor (todos los existentes en el dominio, ver figura E.26), seleccione uno y pulse **Agregar**. Esta tarea debe repetirla por cada empleado, de esa manera el Terminal queda listo para empezar a registrar la asistencia de los empleados de la EPN.

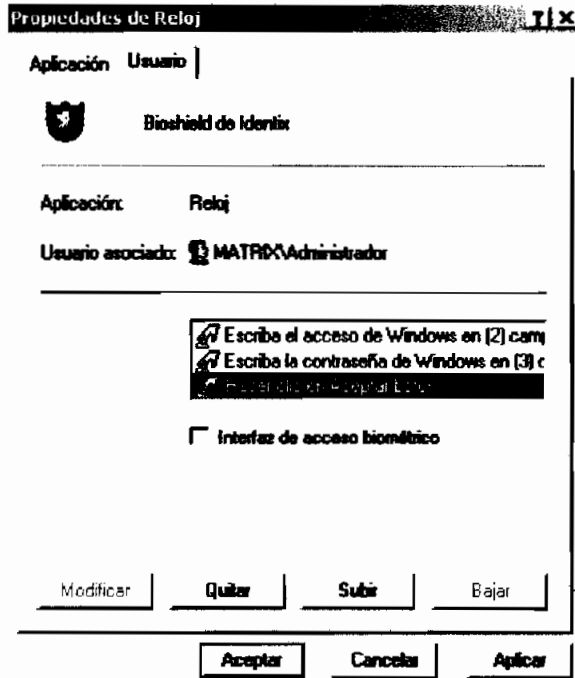


Figura E.23: Propiedades de la aplicación protegida.

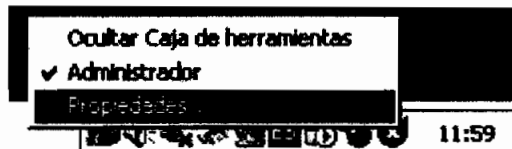


Figura E. 24: Acceso a la ventana de propiedades de *Bioshield*.

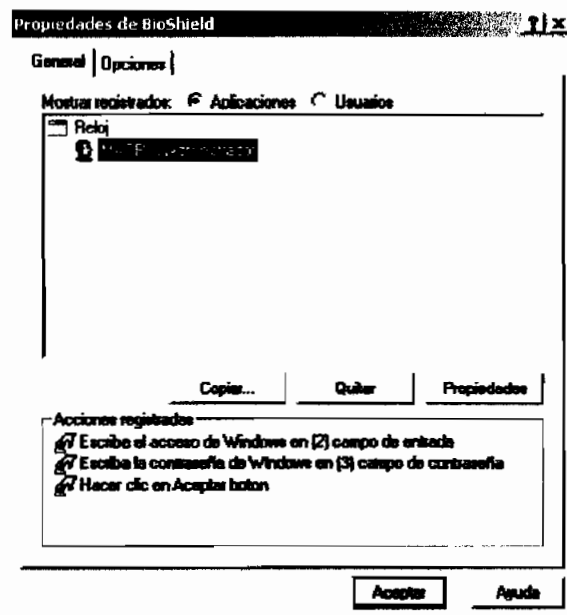


Figura E.25: Ventana de propiedades de *Bioshield*.

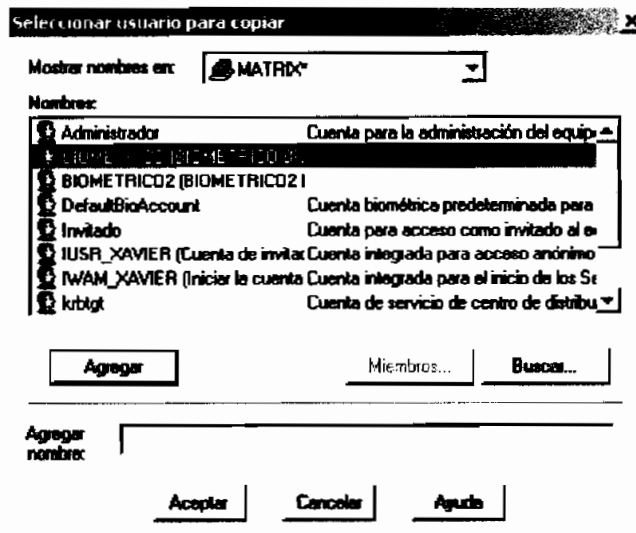


Figura E.26: Cuadro de selección de empleados.

2. MANUAL DE OPERACIÓN DEL SCA

El presente manual ha sido diseñado con el propósito de brindar al usuario una guía eficaz, confiable, oportuna y sencilla, para el óptimo manejo del Sistema de Control de Asistencia.

2.1. Descripción de Sistema

El Sistema de Control de Asistencia (SCA) permite llevar una administración total de la asistencia de los empleados de la Escuela Politécnica Nacional, así como también evaluar la misma mediante controles sencillos que el administrador del sistema debe ejecutar.

2.2. Pantalla de ingreso al SCA

El ingreso SCA se realiza a través de la pantalla de Ingreso al Sistema, mediante la cual el usuario debe ingresar la información correspondiente al *login*, *password* y servidor de la base de datos (ver figura E.27).

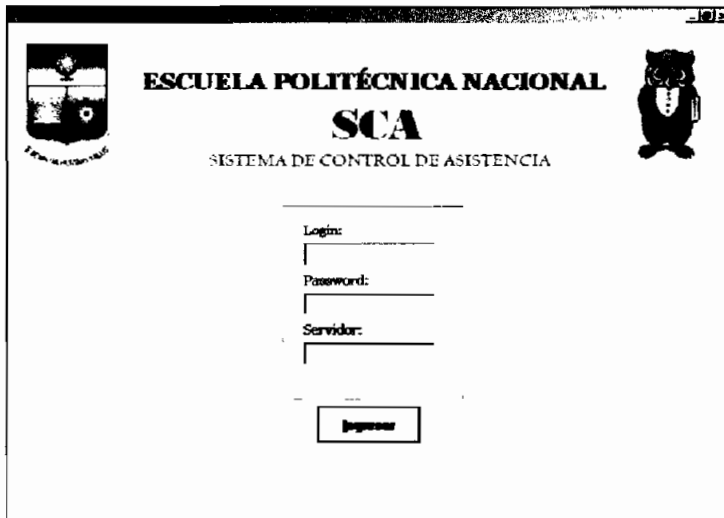


Figura E.27: Pantalla de ingreso al Sistema de Control de Asistencia.

Descripción de los campos de entrada:

Login: Campo obligatorio de tipo texto, de máximo 10 caracteres, que permite el ingreso del *login* del usuario.

Password: Campo obligatorio de tipo texto, de máximo 10 caracteres, que permite el ingreso del *password* del usuario.

Servidor: Campo obligatorio de tipo texto, de máximo 10 caracteres, que permite el ingreso del nombre del servidor (nombre del equipo).

Por defecto el *login* y el *password* a ser ingresado será **ADMINISTRADOR**, posteriormente el administrador del sistema podrá cambiar los campos de ingreso (ver menú Usuarios).

Una vez ingresado los datos de validación y nombre del servidor debe pulsar **Ingresar** para acceder al sistema.

2.3. Manejo del Sistema

El interfaz principal del SCA se aprecia en la figura E.28.

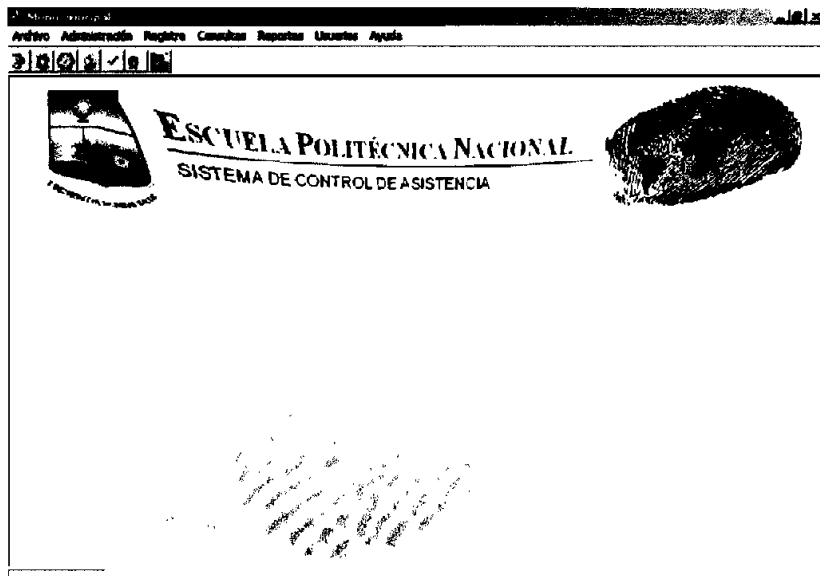


Figura E.28: Interfaz principal del SCA.

Existen dos formas para acceder a las distintas opciones con las que cuenta el sistema, las mismas que son: el Menú Principal y la Barra de Herramientas.

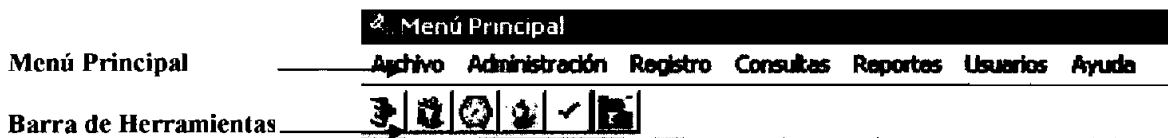








Figura E.29: Menú principal y barra de herramientas del SCA.

La Barra de Herramientas equivale a buscar y pulsar en el menú principal las siguientes opciones:

-  Salir.
-  Administración de empleados
-  Administración de horarios.
-  Asignar horarios.
-  Justificaciones.
-  Vacaciones.

Las opciones de la Barra de Herramientas las podrá utilizar una vez que haya comprendido el funcionamiento del sistema.

2.3.1. Menú Archivo

Cuenta con la opción **Salir** y su tarea como su nombre lo indica es cerrar el interfaz de usuario para regresar al entorno de *Windows*.

2.3.2. Menú Administración

Provee administración de los empleados, está dividida en los siguientes submenús:

- Empleados
- Horarios
- Justificaciones
- Vacaciones

2.3.2.1. Empleados

Tiene las funciones de: Agregar, Modificar y Eliminar un empleado. Despliega la grilla de la figura E.30, en la cual se tiene a todos los empleados registrados hasta ese momento; si no existen empleados registrados, aparecerá vacía.

	APELLIDOS	NOMBRES
1	BALTIMISTA	FABIAN
2	CHUQUI	LEONARDO
3	IBARRA	FERNANDO
4	MOLINA	ADRIANA
5	ZURITA	XAVIER

Figura E.30: Grilla programada para ordenar y desplegar los nombres de los empleados.

Opciones de la grilla:

Buscar.- Cuando se haya ingresado un número considerable de empleados esta opción ahorra tiempo en localizar a un empleado. Para su búsqueda se debe ingresar el nombre del empleado en el orden de Apellidos- Nombres.

Agregar.- Permite agregar un empleado a la Base de Datos del SCA. Al pulsar esta opción se despliega la pantalla de la figura E.31. En la pantalla se debe ingresar los siguientes campos:

- **Código.-** Permite ingresar el código de registro del empleado. Puede ingresar un código de hasta 10 caracteres sean números y/o letras.
- **Nombres.-** Permite ingresar los nombres del empleado. Sólo podrá ingresar letras.
- **Apellidos.-** Permite ingresar los apellidos del empleado. Sólo podrá ingresar letras.
- **Cédula.-** Permite ingresar el número de cédula del empleado. Sólo podrá ingresar números.
- **Fecha de nacimiento.-** Permite ingresar la fecha de nacimiento del empleado.
- **Sexo.-** Permite escoger el tipo de sexo del empleado.
- **Estado Civil.-** Permite escoger el estado civil del empleado; despliega las siguientes opciones: Casado, divorciado, soltero y viudo.
- **Dirección.-** Permite ingresar la dirección del empleado. Puede ingresar hasta 50 caracteres alfanuméricos.
- **Teléfono casa.-** Permite ingresar el número telefónico de la casa del empleado. Puede ingresar hasta 12 caracteres numéricos.
- **Extensión EPN.-** Permite ingresar el número de extensión del lugar u oficina de la EPN en la cual el empleado labora. Permite ingresar hasta 5 caracteres numéricos.
- **Fecha de ingreso.-** Permite ingresar la fecha en la cual el empleado ingresó a trabajar en la EPN.

Los cuatro primeros campos son obligatorios, los demás son opcionales. En el caso que no se ingresa los campos considerados obligatorios, el SCA no permitirá el ingreso de los datos personales del empleado en la Base de Datos del SCA.

Empleado

* Código: _____

* Nombre: _____

* Apellido: _____

* Cédula: _____

Fecha Nacimiento: 2004-06-14

Sexo: C F M

Estado Civil: SOLTERO

Dirección: _____

Teléfono Casa: _____

Estación EPN: _____

Fecha Ingreso: 2004-06-14

* Datos que se debe ingresar obligatoriamente.

Aceptar Cancelar

Figura E.31: Ventana para ingreso de los Datos Personales del empleado.

Modificar.- Permite modificar los datos personales de un empleado en la Base de Datos del SCA. Al pulsar esta opción se despliega la pantalla de la figura E.32 la misma que se desplegará con los datos ingresados hasta el momento. Todos los campos se pueden modificar excepto el del Código del empleado.

Empleado

* Código: EPN3

* Nombre: FABIAN

* Apellido: BAUTISTA

* Cédula: 1855669882

Fecha Nacimiento: 2004-09-27

Sexo: C F M

Estado Civil: SOLTERO

Dirección: ROSALES LOTE 28 Y TULIPANES

Teléfono Casa: 2321815

Estación EPN: 330

Fecha Ingreso: 2004-06-14

* Datos que se debe ingresar obligatoriamente.

Aceptar Cancelar

Figura E.32: Pantalla de modificación de los Datos Personales del empleado.

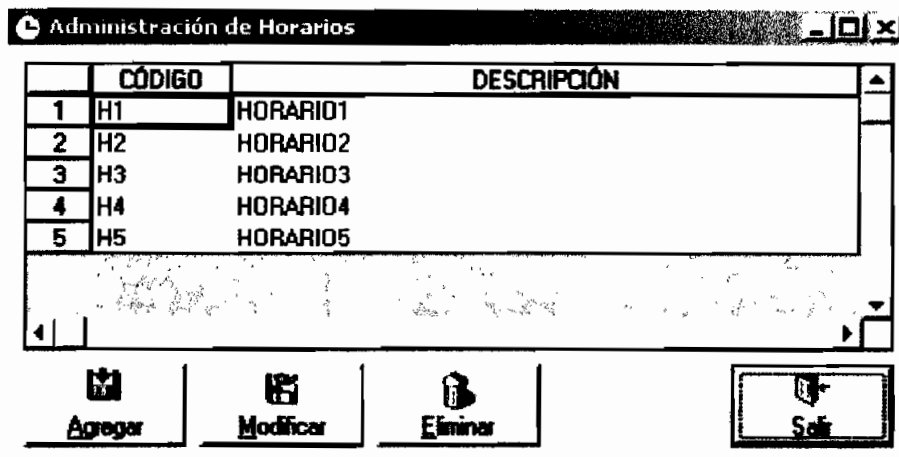
Eliminar.- Permite eliminar todos los datos personales de un empleado de la Base de Datos del SCA. Esta opción solo se aplica si el SCA no ha evaluado en ningún momento la asistencia del empleado, caso contrario no podrá ser eliminado.

2.3.2.2. Horarios

Controla todo lo referente a horarios y se divide en: Administración y Asignación:

2.3.2.2.1. Administración

Despliega una grilla con todos los horarios creados por el administrador del sistema, si no existe alguno aparece vacía, ver figura E.33. En este submenú se ingresará los horarios que se necesite para los empleados de la EPN.



	CÓDIGO	DESCRIPCIÓN
1	H1	HORARIO1
2	H2	HORARIO2
3	H3	HORARIO3
4	H4	HORARIO4
5	H5	HORARIO5

Figura E.33: Grilla programada para ordenar y desplegar los horarios de los empleados.

Opciones de la grilla:

Agregar.- Permite agregar un Horario a la Base de Datos del SCA. Al pulsar esta opción se despliega la pantalla de la figura E.34. En la pantalla se deben ingresar los siguientes campos:

- **Código.-** Campo que permite ingresar el código de horario a crearse. Puede ingresar un código de hasta 4 caracteres sean números y/o letras.

Código	Descripción
Lunes:	Jornada Única <input type="checkbox"/> Doble jornada
Martes:	Jornada Única <input type="checkbox"/> Doble jornada
Miércoles:	No ingresar horario <input type="checkbox"/> Doble jornada
Jueves:	Jornada Única <input type="checkbox"/> Doble jornada
Viernes:	Jornada Única <input type="checkbox"/> Doble jornada
Sábado:	Jornada Diferenciada <input checked="" type="checkbox"/> Doble jornada

Figura E.34: Ventana de definición de horarios.

- **Descripción.-** Campo que permite ingresar el nombre con el cual se identificara el horario. Permite ingresar hasta 20 caracteres alfanuméricos.
- **Tipo de Jornada.-** Este campo se debe seleccionar para cada uno de los días de la semana. Tiene las siguientes opciones:
 - **Jornada Única.-** Jornada normal de ocho horas, se debe agregar media hora de almuerzo. Este tipo de horario solo permite ingresar la hora de entrada; la hora de salida es calculada por el sistema sumando las ocho horas y media mencionadas.
 - **Jornada Diferenciada.-** Puede darse el caso que tenga que trabajar una jornada diferente a la normal de ocho horas, en este caso se ingresa la hora de entrada y salida del empleado. La mínima jornada que permite ingresar el sistema es de una hora.
 - **No ingresar horario.-** Al seleccionar esta opción le dice al SCA que ese día no se laborará.
 - Si a más de selecciona la opción de "Jornada Diferenciada" escoge la opción "Doble jornada" permitirá ingresar un horario que tenga doble jornada. Entre jornadas debe haber un mínimo de una hora. Se deberá ingresar la hora de entrada y salida de ambas jornadas.

Una vez seleccionado los tipos de jornada, se debe pulsar **Ingresar** si desea proseguir con el horario creado o **Cancelar** si desea cancelar el ingreso.

Si pulsa **Ingresar** se desplegará la pantalla de la figura E.35, en ella se debe proceder a ingresar las horas de entrada y salida, éstas se seleccionarán en los combos de sus respectivos días dependiendo del tipo de jornada escogida para ese día. Como ejemplo se ingresa un horario denominado "HORARIO 6" (ver figura E.35) con los siguientes tipos de jornadas:

- **Lunes** Jornada Diferenciada
- **Martes** Jornada Diferenciada Doble
- **Miércoles** No ingresar horario
- **Jueves** No ingresar horario
- **Viernes** Jornada Única
- **Sábado** No ingresar horario

Para ingresar las horas de entrada y/o salida solo debe desplegar el combo respectivo, el combo automáticamente le mostrará todas las opciones que dispone dependiendo del tipo de jornada y de lo que se haya ingresado en los otros combos. Si para un día en particular ya se ingresó las horas de la jornada y se desea cambiar alguna hora, todos los combos afectados por ese cambio se colocarán en blanco invitándolo a ingresar nuevamente esas horas. El resultado del ingreso de las horas se muestra en la figura E.36. Si alguna hora no ha sido ingresada el SCA le notificará y no permitirá el registro del horario.

Las opciones de la pantalla de ingreso de horario (figura E.35) son:

- **Aceptar.**- Esta opción permite almacenar el horario y finalizar el proceso.
- **Regresar.**- Si desea cambiar el tipo de jornada de algún día, esta opción permite regresar a la pantalla de definición de horarios.

✓ Ingreso de Horario

Ingresar Horario

	Entrada - Salida 1		Entrada - Salida 2	
Lunes:	07:00			
Martes:	07:00			
Miércoles:				
Jueves:				
Viernes:	07:00			
Sábado:				

Figura E.35: Pantalla de ingreso de las horas de entrada y/o salida del horario. Parte 1 de 2

✓ Ingreso de Horario

Ingresar Horario

	Entrada - Salida 1		Entrada - Salida 2	
Lunes:	07:00			15:00
Martes:	08:00	12:00	13:30	17:30
Miércoles:				
Jueves:				
Viernes:	08:30			17:30
Sábado:				

Figura E.36: Pantalla de ingreso de las horas de entrada y/o salida del horario. Parte 2 de 2.

Modificar.- Permite modificar los datos del horario seleccionado en la Base de Datos del SCA. Al pulsar esta opción se despliega la pantalla de definición de horarios la misma que se desplegará con los datos ingresados hasta el momento. Todos los campos se pueden modificar excepto el del Código del horario.

Siguiendo con el ejemplo anterior se modificará el HORARIO 6 el cual ya fue ingresado. Se va a modificar el tipo de jornada del día viernes (de jornada única a jornada única diferenciada), referirse a la figura E.37.

Una vez que se haya cambiado el tipo de jornada, pulse **Modificar**. Si el cambio solo es de horas y no de tipo de jornada, no cambie ningún tipo de jornada y pulse **Modificar**.

The screenshot shows a window titled "Definición de Horarios" with a table of days and their corresponding work schedules. Below the table are two buttons: "Modificar" and "Cancelar".

Código:	Descripción:
H6	HORARIO 6
Lunes:	Jornada Diferenciada <input type="checkbox"/> Doble jornada
Martes:	Jornada Diferenciada <input checked="" type="checkbox"/> Doble jornada
Miércoles:	No ingresar horario <input type="checkbox"/> Doble jornada
Jueves:	No ingresar horario <input type="checkbox"/> Doble jornada
Viernes:	Jornada Diferenciada <input checked="" type="checkbox"/> Doble jornada
Sábado:	No ingresar horario <input type="checkbox"/> Doble jornada

Modificar Cancelar

Figura E.37: Pantalla de modificación del HORARIO 6.

Una vez que se pulse Modificar se desplegará la pantalla de la figura E.38. Se deberá ingresar la jornada modificada y si desea se podrá modificar las horas de entrada y/o salida de los demás días, esto no implica que se esté modificando el tipo de jornada de ese día.

Eliminar.- Permite eliminar un horario de la Base de Datos del SCA. Esta opción solo es aplicable si el horario no ha sido asignado algún empleado; si desea eliminarlo deberá desasignar a todos los empleados con ese horario.

Ingreso de Horario

Ingresar Horario

	Entrada - Salida 1		Entrada - Salida 2	
Lunes:	07:00			18:00
Martes:	08:00	12:00	13:30	17:30
Miércoles:				
Jueves:				
Viernes:	08:30	09:30		
Sábado:				

Figura E.38: Pantalla de ingreso de las horas de entrada y/o salida del HORARIO 6

2.3.2.2.2. Asignación

Permite asignar un horario a un empleado seleccionado, al pulsar esta opción se despliega la grilla de la figura E.39.

Asignación de Horarios

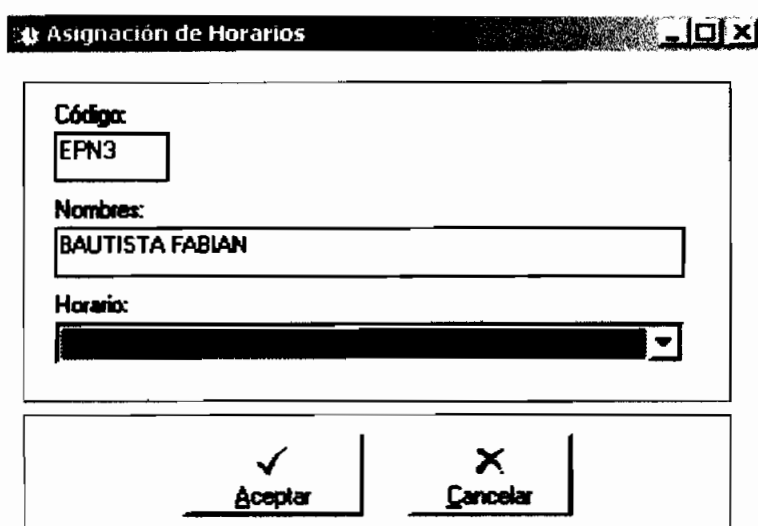
Buscar

	APELLIDOS	NOMBRES
1	BAUTISTA	FABIAN
2	CHUQUI	LEONARDO
3	IBARRA	FERNANDO
4	MOLINA	ADRIANA
5	ZURITA	XAVIER

Figura E.39: Grilla de asignación de horarios.

Opciones de la Grilla

Agregar.- Permite asignar un Horario de la Base de Datos del SCA a un empleado. Al pulsar esta opción se despliega la pantalla de la figura E.40. Para asignar el horario se debe desplegar el combo de la opción **Horario** de la pantalla de la figura E.40, en él se desplegará todos los horarios creados hasta ese momento (ver figura E.41) y se escogerá uno de esos.



Asignación de Horarios

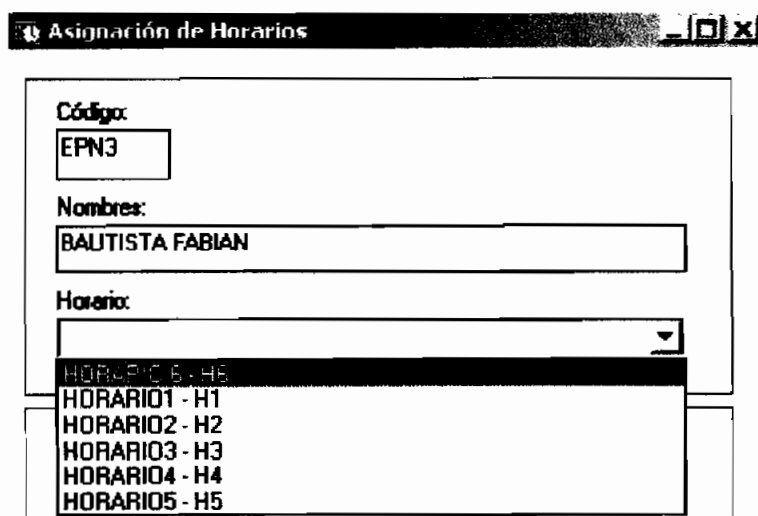
Código:
EPN3

Nombres:
BAUTISTA FABIAN

Horario:

✓ Aceptar ✗ Cancelar

Figura E.40: Ventana para la asignación de horarios.



Asignación de Horarios

Código:
EPN3

Nombres:
BAUTISTA FABIAN

Horario:

HORARIO 6 - H6
HORARIO1 - H1
HORARIO2 - H2
HORARIO3 - H3
HORARIO4 - H4
HORARIO5 - H5

Figura E.41: Combo Horario desplegado con los horarios creados en el SCA.

Las opciones de la ventana de asignación de horarios son:

- **Aceptar.**- Ingresa la asignación realizada a la Base de Datos del SCA.
- **Cancelar.**- Cancela la acción.

2.3.2.3. Justificaciones

Si un empleado ha faltado cierto día por un motivo justificado, éste puede hacer uso de las justificaciones para no permitir que esos minutos atrasados sean sancionados en su rol de pagos. El SCA consulta todos los días atrasados para registrarlos como posibles días a ser justificados, ya sea parte o total de los mismos.

Para que un día de atraso sea tomado como candidato a ser justificado, debe haber pasado por un proceso que evalúe la asistencia del empleado, este proceso se ejecuta con la "Actualización del Registro" (ver menú Registro).

Al momento de seleccionar el menú Justificación, se desplegará la grilla de la figura E.42 con una lista de los empleados de la EPN.

	APELLIDOS	NOMBRES
1	BAUTISTA	FABIAN
2	CHUQUI	LEONARDO
3	IBARRA	FERNANDO
4	MOLINA	ADRIANA
5	ZURITA	XAVIER

Figura E.42: Grilla de justificaciones.

Opciones de la Grilla:

Agregar.- Permite desplegar la pantalla de la figura E.43 en la cual se escogerá las fechas inicio y final de los días de atraso que se desea visualizar en la pantalla "Justificaciones". Solo se podrán seleccionar fechas menores o iguales a la fecha del Registro (consulte el menú Registro).

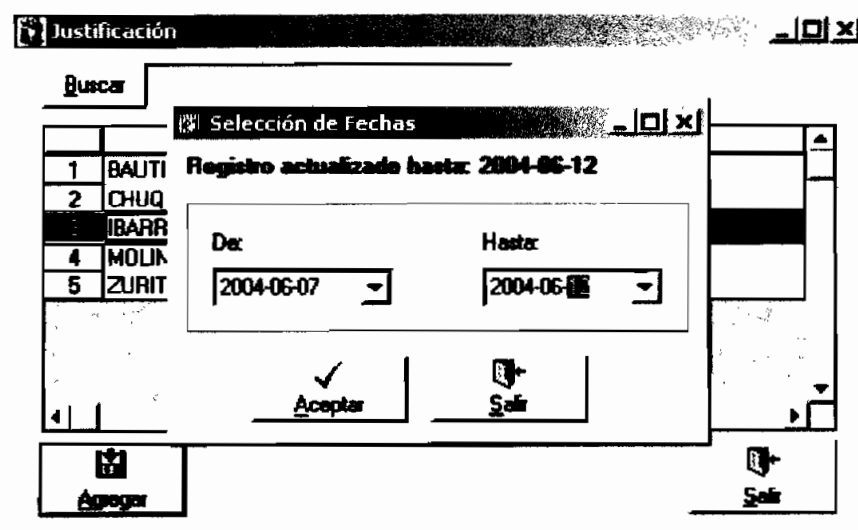


Figura E.43: Ventana de selección del rango de fechas de atraso para una posible justificación.

Una vez seleccionado el rango de fechas aparecerá la pantalla de la figura E.44 en la cual se visualizará los días comprendidos en ese rango que tengan al menos un atraso.

	FECHA	JORNADA	MINUTOS ATRASADOS				JUSTIFICAR	MINUTOS JUSTIFICADOS			
			E1	S1	E2	S2		E1	S1	E2	S2
1	07/06/2004	Doble	0	0	30	0	Justificar	0	0	0	0
2	08/06/2004	Doble	5	0	0	0	Justificar	0	0	0	0
3	09/06/2004	Doble	300	0	0	0	Justificar	0	0	0	0
4	10/06/2004	Doble	0	0	210	0	Justificar	0	0	0	0
5	11/06/2004	Doble	300	0	210	0	Justificar	0	0	0	0

Figura E.44: Pantalla de justificaciones.

Los días atrasados se presentan con la fecha, tipo de jornada y los minutos atrasados en la respectiva entrada o salida. Si se desea justificar algún atraso, se debe pulsar el botón “Justificar” en la fila del día correspondiente; dependiendo de la jornada se desplegará el tipo de ventana, jornada única o diferenciada tal como se indica en la figura E.45.

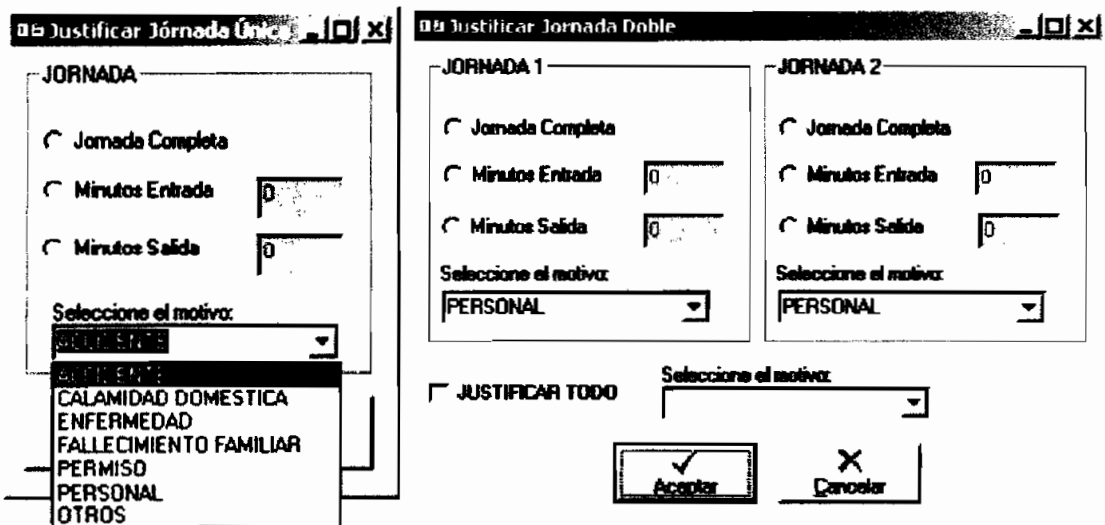


Figura E.45: Ventanas de justificación, (a) Para jornada única o diferenciada, (b) Para jornada diferenciada doble.

Las opciones que se presentan en las ventanas de justificación:

- **Jornada Completa.**- No importa cuantos minutos se haya atrasado en la entrada y/o salida, esta opción permite justificar todos esos minutos en la respectiva jornada.
- **Minutos Entrada.**- En el caso que los haya, permite ingresar el número de minutos que se desea justificar del atraso en la entrada de la respectiva jornada.
- **Minutos Salida.**- En el caso que los haya, permite ingresar el número de minutos que se desea justificar de los minutos sancionados por salir antes de la hora señalada en la respectiva jornada.
- **Selección del motivo.**- Cuando se justifique algún atraso se debe ingresar la respectiva justificación, ésta se selecciona en el combo “Selección del motivo”. Existen seis motivos de justificación: Accidente, Calamidad Doméstica, Enfermedad, Fallecimiento familiar, Permiso y

Personal, en caso de no estar definido el motivo en las opciones, se ingresará "Otros".

- **Justificar Todo.**- Esta opción está disponible en la ventana de justificación de la doble jornada y como su nombre lo indica permite justificar todos los atrasos de ese día seleccionado.

Estas pantallas tienen los respectivos filtros para no ingresar datos incoherentes en el día de atraso como por ejemplo, justificar más minutos de los atrasados, ingresar justificaciones en jornadas que no se atrasó, etc.

Una vez que se ha justificado se debe pulsar la opción **Aceptar** para ingresar la justificación o **Cancelar** para cancelar la acción. Si se desea modificar la justificación se debe pulsar el botón, que automáticamente se cambió de "Justificar" a "Modificar" una vez que se ingresó la justificación, de modificación en el respectivo día de atraso y se desplegará las mismas pantallas con los datos ingresados, los cuales se pueden modificar.

2.3.2.4. Vacaciones

Se despliega la grilla de la figura E.46 con la lista de empleados de la EPN.

Buscar

	APELLIDOS	NOMBRES
1	BAUTISTA	FABIAN
2	CHUQUI	LEONARDO
3	IBARRA	FERNANDO
4	MOLINA	ADRIANA
5	ZURITA	XAVIER

Aceptar Modificar Eliminar Salir

Figura E.46: Grillas para asignación de vacaciones.

Opciones de la Grilla:

Agregar.- Permite desplegar la pantalla de la figura E.47 en la cual se escogerá las fechas Salida y Entrada de las vacaciones. El sistema no admite que la fecha de salida sea menor que la fecha de regreso de las vacaciones.

Se puede ingresar varias vacaciones para que éstas se activen en el transcurso del año; si se tiene varias vacaciones, la opción de modificar o eliminar siempre será respecto a las últimas vacaciones ingresadas y si las vacaciones están transcurriendo o ya transcurrieron en fechas anteriores, ya no es posible eliminarlas o modificarlas.

The image shows a screenshot of a software application window titled "Vacaciones". On the left side, there is a search bar labeled "Bus" with a list of numbers 1 through 5. The main content area contains the following fields:

- Código:** A text box containing "EPN3".
- Nombres:** A text box containing "BAUTISTA FABIAN".
- Fecha Salida:** A date picker showing "2004-06-11".
- Fecha Entrada:** A date picker showing "2004-06-14".

At the bottom of the form, there are two buttons: "Aceptar" (with a checkmark icon) and "Cancelar" (with an 'X' icon).

Figura E.47: Pantalla de ingreso de vacaciones.

Modificar.- Permite modificar las últimas vacaciones ingresadas siempre y cuando las mismas no estén transcurriendo o hayan transcurrido en fechas anteriores.

Eliminar.- Permite eliminar las últimas vacaciones ingresadas siempre y cuando éstas no se encuentren transcurriendo o hayan transcurrido en fechas anteriores.

2.3.3. Menú Registro

Cuando se empiece a registrar la asistencia de los empleados se ingresará en la base de datos para cada día, el código y fecha-hora de sus registros, éstos se acumulan hasta que un proceso de decisiones evalúe la asistencia de los mismos. Este proceso lo realiza la opción “Menú Registro”.

El proceso ocurre desde la fecha de la última actualización hasta un día antes del día que se actualiza; primero se consulta la última actualización del registro, si ya se ha actualizado éste envía un mensaje de aviso.

Al pulsar el menú aparece la pantalla de la figura E.48 con la última fecha de actualización.

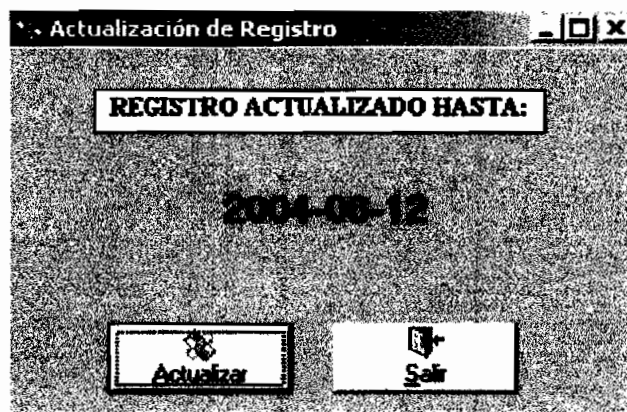


Figura E.48: Ventana de actualización del Registro.

Es importante que se actualice este proceso periódicamente ya que toda evaluación de asistencia y tareas de justificación, consultas y reportes estarán actualizadas hasta esa fecha.

2.3.4. Menú Consultas

Permite ingresar a la base de datos del SCA en modo sólo lectura, de una manera rápida, para poder consultar datos de interés, como son:

- Asistencia.

- Datos personales del empleado.
- Horario asignado al empleado.
- Vacaciones.
- Horarios existentes en el SCA.

2.3.5. Menú Reportes

Permite obtener los reportes del SCA, éstos son:

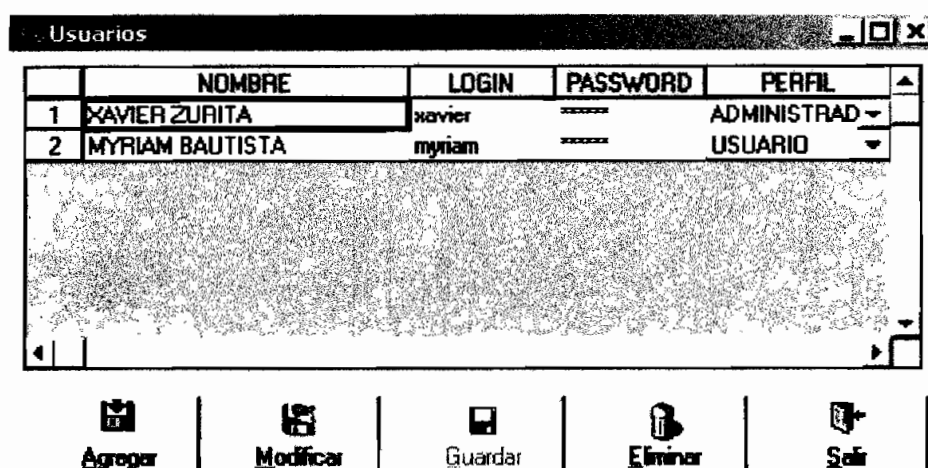
- Empleados.- Presenta una lista de los empleados ingresados en la base de datos del SCA con sus nombres, apellidos y código respectivo.
- Empleado Asistencia.- Este reporte permite desplegar en una hoja de Excel el registro de asistencia de un empleado seleccionado para un rango de fechas.
- Empleados Horarios.- Despliega una lista de los empleados con su horario asignado. Si algún empleado no se encuentra asignado a un horario, aparece el texto: NO ASIGNADO.
- Empleados Vacaciones.- Presenta todos los empleados que tengan vacaciones. En el caso de tener más de una asignación, el reporte presenta las vacaciones más recientes que tenga el empleado.
- Horario.- Este reporte permite escoger un horario en particular, desplegar una hoja de Excel con el horario respectivo y en otra todos los empleados asociados a ese horario.
- Minutos Atrasados.- El reporte más importante, es el producto neto del SCA. Se realiza para un mes determinado o para un rango de fechas y presenta el nombre del empleado y el total de minutos atrasados y justificados en ese mes.

Para obtener cada uno de ellos solo se debe pulsar el submenú respectivo y automáticamente se desplegará una hoja de Excel con un botón denominado "Generar" que al accionarlo calculará y presentará el reporte respectivo.

2.3.6. Menú Usuarios

Permite ingresar o eliminar usuarios del SCA, existirán dos tipos: Administrador y Usuario. El Administrador tendrá derecho a modificar y crear en el SCA, en cambio el Usuario solo tendrá derecho a consulta.

Al pulsar el menú Usuarios se desplegará la grilla de la figura E.49.



	NOMBRE	LOGIN	PASSWORD	PERFIL
1	XAVIER ZURITA	xavier	-----	ADMINISTRAD
2	MYRIAM BAUTISTA	myriam	-----	USUARIO

Figura E.49: Grilla de usuarios del SCA.

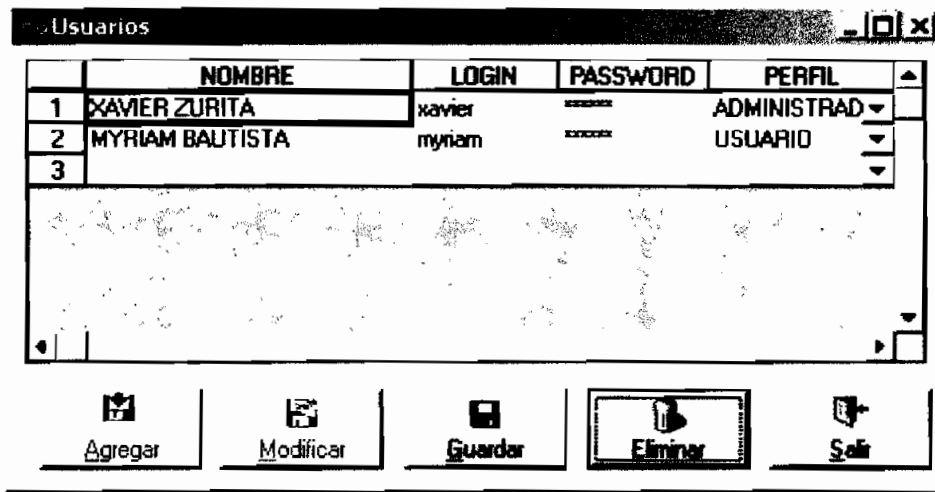
Opciones de la Grilla:

Agregar.- Permite agregar un nuevo usuario del SCA. No despliega ninguna pantalla sino que inserta una nueva fila en la grilla para ingresar en su respectiva columna: Nombre, *login*, *password* y perfil del usuario (Administrador o Usuario); ver figura E.50.

Una vez ingresada la información mencionada se debe presionar la opción Guardar para que el usuario sea añadido a la Base de Datos del SCA.

Modificar- Permite modificar los datos del usuario seleccionado. El único dato que no se puede modificar es el NOMBRE.

Eliminar.- Permite eliminar el usuario seleccionado. No puede eliminar el usuario que se encuentre activo en ese momento.



	NOMBRE	LOGIN	PASSWORD	PERFIL
1	XAVIER ZURITA	xavier	*****	ADMINISTRAD
2	MYRIAM BAUTISTA	myriam	*****	USUARIO
3				

Agregar Modificar Guardar Eliminar Salir

Figura E.50: Grilla de usuarios del SCA.

ANEXO F

Presupuesto Referencial

PRESUPUESTO REFERENCIAL

Los costos que se presentan a continuación dan una estimación aproximada del valor total de la implementación del SCA.

Cantidad	Artículo	Descripción	Valor Unitario (USD)	Valor Total (USD)
3	Computador Pentium IV.	<ul style="list-style-type: none"> • Microprocesador Intel Pentium IV de 1.8 GB. • 256 MB de memoria RAM. • 40 GB Disco Duro. • Tarjeta de red. • Floppy Disk de 3½. • CD-ROM. • Licencia de Windows XP. 	700	2100
1	Licencia para Windows Server 2002.	<ul style="list-style-type: none"> • Licencia para Microsoft Windows 2000 Server. • Permiso para cinco clientes (equipos). 	849	849
3	Lector de huellas dactilares.	<ul style="list-style-type: none"> • Serie DFR-200. • Interfaz USB. • Propietario de Identix. • Incluye el software Biologon 3.0 con cinco licencias. 	120	360
1	Licencia para Biologon 3.0 Cliente/Servidor.	<ul style="list-style-type: none"> • Licencia para un usuario de red. 	37	37
1	Cableado de interconexión.	<ul style="list-style-type: none"> • Cableado estructurado para interconexión de la red. • Incluye materiales e instalación. 	700	700
TOTAL				4046