Universidad Nacional Autónoma de México

Facultad de Ingeniería

TESIS

"SISTEMA DE ADMINISTRACIÓN ELECTRÓNICA APLICADA AL LABORATORIO ABIERTO" (DE LA FACULTAD DE INGENIERÍA)

QUE PARA OBTENER EL TÍTULO DE INGENIERO EN COMPUTACIÓN

PRESENTA

Víctor Pérez Hernández

Asesor: Ing. Vicente Flores Olvera

Ciudad Universitaria, agosto de 2009

DEDICATORIA

Este trabajo, que el dia de hoy culmino, constituye un gran esfuerzo para lograr cerrar un ciclo muy importante en mi vida; esto no hubiese sido posible si no hubiese tenido el apoyo de tantas personas que contribuyeron para el logro de ésta meta.

Primeramente, doy gracias a Dios por concederme tener vida, por permitirme conocerle y porque sin él, nada hubiese sido posible.

Aprecio de manera especial la asesoría y apoyo que recibí por parte del Ing. Vicente Flores Olvera y de la ing. Socorro Guevara por dedicarme algo de su tiempo, conocimientos y experiencia para lograr concretar éste trabajo. Sólo me resta decirles que estoy en deuda con ellos.

Este trabajo, lo dejo como legado y ejemplo de esfuerzo para mi pequeña Yolotzin para que si algún día lo llega a leer, le sirva de inspiración para que se esfuerce en ser cada día mejor y que vea que no todo está perdido y que mientras haya vida, aún se puede hacer mucho en la medida en que así lo vaya viviendo. Esta es una de las mejores herencias que te puedo dejar hija mia.

Al Doctor Roberto Miranda y a la Doctora Irma Saldivar, no tengo manera de pagarles lo mucho que me han ayudado y que gracias a ustedes, he vuelto a vivir; sé que dios los puso en mi camino y les estaré eternamente agradecido.

A mi incondicicional madre, quien me dió la vida, quien ha estado conmigo como una confidente; como mi complemento, quien ha vivido conmigo momentos de tristeza, momentos de lágrimas, de gloria: he aquí la culminación de lo que tanto soñé y que tanto te ofrecí; sé madre que sólo estamos viviendo momentos pasajeros, creo y estoy convencido que vendrán tiempos mejores y, si tengo que partir antes que tú, recuérdame con lo mejor de mí, no de lo malo.

A todos los demás, que puedo contar con los dedos de mis manos, les doy infinitas gracias pues con su granito de arena han aportado mucho al logro de este trabajo y en la medida de lo posible, les retribuiré lo tanto que me han ayudado.

Para la Universidad, miles de gracias por permitirme haberme preparado y adquirido los conocimientos y destrezas que me han ayudado en todo éste tiempo…

ÍNDICE DE CONTENIDO

OBJETIVOS **5**

OBJETIVO GENERAL 5
OBJETIVOS ESPECÍFICOS 5

INTRODUCCIÓN **8**

CAPÍTULO I: EL LABORATORIO ABIERTO **10**

I.1 ESTRUCTURA Y COMPOSICIÓN DEL LABORATORIO ABIERTO 10
I.1.1 Laboratorio abierto de la Facultad de Ingeniería 11
I.2 DESCRIPCIÓN FUNCIONAL SOBRE LOS SERVICIOS PROPORCIONADOS POR EL LABORATORIO ABIERTO DE LA FACULTAD DE INGENIERÍA 14
I.2.1 Servicios bridados por el Laboratorio Abierto 15
I.2.2 Facultades de los Administradores del Laboratorio 15
I.2.3 Procedimiento para otorgar servicios a los usuarios 15
I.2.4 Demanda de los servicios del Laboratorio 16
I.3 ANÁLISIS DE LOS PRINCIPALES PROBLEMAS A RESOLVER 17

CAPÍTULO II: SOLUCIÓN A LA PROBLEMÁTICA PRESENTADA **21**

II.1 ANTECEDENTES DE LA SOLUCIÓN 21
II.2 ESTRUCTURA DE LA SOLUCIÓN 23
II.3 SOFTWARE DE DESARROLLO 24

CAPÍTULO III: CONCEPTOS GENERALES **27**

III.1 SOFTWARE 27
III.1.1 Lenguajes de programación 28
III.1.2 Bases de Datos 29
III.1.3 Modelaje de las Bases de Datos 33
III.2 HARDWARE 36
III.2.1 Puerto serial 37
III.2.1.1 Niveles lógicos RS232 38
III.2.1.2 Formato de un byte 39
III.2.2 Puerto paralelo 40
III.2.3 Puerto USB 43
III.2.3.1 INTERFAZ FÍSICA (ELÉCTRICA) 45
III.2.3.2 TRANSMISIÓN EN EL USB 46
III.2.3.2.1 TRANSMISIÓN ASINCRÓNICA 46
III.2.3.2.2 TRANSMISIÓN SINCRÓNICA 47
III.2.3.2.3 TRANSMISIÓN ISOCRONICAS 47
III.2.3.2.4 TRANSMISIÓN BULK 48
III.2.3.2.5 TRANSMISIONES DE CONTROL 48
III.2.3.2.6 TRANSMISIONES DE INTERRUPCIÓN 48
III.3 *INTERFAZ ELECTRÓNICA* 48
III.3.1 Registros (latches) 48
III.3.2 Decodificadores 52
III.3.3 Triacs 53
III.3.3.1 TRIAC con acoplado óptico (opto coupler TRIAC) 54
III.3.4 Optoacopladores 54
III.3.4.1 Tipos de Optoacopladores 54
III.3.4.2 Funcionamiento 55

III.3.4.3 Encapsulados 56
III.3.5 Micros controladores 56
III.3.6 Convertidor de señal de niveles RS232 a TTL 57

CAPÍTULO IV: DESARROLLO DEL SISTEMA 59

IV.1 ELECCIÓN DEL PUERTO DE COMUNICACIONES 59
IV.2 DESARROLLO DE LA BASE DE DATOS BAJO SQL SERVER EXPRESS 61
IV.2.1 Entidad tbDump 63
IV.2.2 Entidad tbEspera 63
IV.2.3 Entidad tbInventario 64
IV.2.4 Entidad tbInvTemp 65
IV.2.5 Entidad tbModulos 66
IV.2.6 Entidad tbMovimientos 69
IV.2.7 Entidad tbMovsModulos 70
IV.2.8 Entidad tbPréstamos 71
IV.2.9 Entidad tbSanciones 72
IV.2.10 Entidad tbSeguridad 74
IV.2.11 Entidad tbSysParms 75
IV.2.12 Entidad tbSysTemp 75
IV.2.13 Entidad tbTransacción 76
IV.2.14 Entidad tbUsuarios 77
IV.3 DESCRIPCIÓN DEL SOFTWARE 79
IV.3.1 CLASE clsGeneral 86
IV.3.2 CLASE clsInterfase 86
IV.3.3 CLASE clsModulo 87
IV.3.4 CLASE clsParametros 88
IV.3.5 CLASE clsSeguridad 89
IV.3.6 CLASE clsTransacción 89
IV.3.7 CLASE frmAcercaDe 90
IV.3.8 CLASE frm_Administración_ActivarCuenta 91
IV.3.9 CLASE frm_Administración_ArchivoDeTransacciones 92
IV.3.10 CLASE frm_Administración_BloquearCuenta 93
IV.3.11 CLASE frm_Administración_BorrarCuenta 93
IV.3.12 CLASE frm_Administración_CambiarContraseña 94
IV.3.13 CLASE frm_Administración_ContraseñaNueva 94
IV.3.14 CLASE frm_Administración_CrearCuentaNueva 95
IV.3.15 CLASE frm_Administración_Cuentas 95
IV.3.16 CLASE frm_Administración_NombreNuevoDeUsuario 97
IV.3.17 CLASE frmAdministraciónPanel 97
IV.3.18 CLASE frm_Administración_UsuarioNuevoRápido 99
IV.3.19 CLASE frmBienvenida 99
IV.3.20 CLASE frmBox 100
IV.3.21 CLASE frmConfiguraciónPorPrimeraVez 100
IV.3.22 CLASE frmConfigurar 101
IV.3.23 CLASE frmConsulta_ActualizaciónDatosXOperador 102
IV.3.24 CLASE frmConsulta_ListaDeEspera 103
IV.3.25 CLASE frmConsulta_MaterialesXUsuario 103
IV.3.26 CLASE frmConsulta_MovsXMódulo 104
IV.3.27 CLASE frmConsulta_MovsXOperadorMóduloYSemestre 105
IV.3.28 CLASE frmConsulta_MovsXSemestre 105

IV.3.29 CLASE frmConsulta_MovsXUsuario *106*
IV.3.30 CLASE frmConsulta_MovsXMóduloYSemestre *107*
IV.3.31 CLASE frmConsulta_PréstamosXSemestre *107*
IV.3.32 CLASE frmConsultaDeSancionesAplicadasPorOperadorDelSistema *108*
IV.3.33 CLASE frmConsulta_SancionesxUsuario *109*
IV.3.34 CLASE frmConsultaDeUsuariosCreadosPorOperadorDelSistema *109*
IV.3.35 CLASE frmDespedida *110*
IV.3.36 CLASE frmEspera *111*
IV.3.37 CLASE frmEstadística_MaterialesPrestadosXUsuario *111*
IV.3.38 CLASE frmEstadísticas_MaterialesPrestados *113*
IV.3.39 CLASE frmEstadísticasDeSancionesAplicadasPorUsuario *114*
IV.3.40 CLASE frmEstadísticas_ServicioPorUsuario *115*
IV.3.41 CLASE frmEstadísticas_ListaEspera *116*
IV.3.42 CLASE frmEstadísticas_UsoDeEquiposPorUsuarios *117*
IV.3.43 CLASE frmEstadísticas_UsoDeEquipos *118*
IV.3.44 CLASE frmInicioDeSesión *119*
IV.3.45 CLASE frmInventario_Edición *119*
IV.3.46 CLASE frmInventario_Módulos *120*
IV.3.47 CLASE frm_Inventario_ConsultaDeLosMovimientos *120*
IV.3.48 CLASE frmInventario_Acción *121*
IV.3.49 CLASE frmListadoDeUsuarios *122*
IV.3.50 CLASE frmMateriales_Entrega *122*
IV.3.51 CLASE frmMateriales_Préstamos *123*
IV.3.52 CLASE frmPregunta *123*
IV.3.53 CLASE frmPrincipal *124*
IV.3.54 CLASE frmSanciones *130*
IV.3.55 CLASE frmUsuarios_Actualizar *131*
IV.3.56 CLASE frmUsuarios_Agregar *132*
IV.3.57 CLASE frm_Usurarios_SelecciónDeUsuario *133*
IV.3.58 CLASE frmUsuarios_Confirmar *133*
IV.3.59 MODULO modFunciones *134*
IV.3.60 MODULO modInicio *134*
IV.3.61 MODULO modInventario *135*
IV.3.62 MODULO modModulos *135*
IV.3.63 MODULO modPanel *136*
IV.3.64 MODULO modSanciones *136*
IV.3.65 MODULO modSeguridad *136*
IV.3.66 MODULO modUsuarios *137*

CAPÍTULO V: INTERFAZ ELECTRÓNICA 138

V.1 ELECCIÓN DE COMPONENTES 138
V.2 ANTECEDENTES 141
V.3 PROTOCOLO DE COMUNICACIÓN CON LA *INTERFAZ ELECTRÓNICA* 143
V.3.1 Comandos implementados en la interfaz electrónica *146*
V.3.2 Respuesta de la interfaz electrónica *146*
V.4 MODELO CONCEPTUAL DE LA *INTERFAZ ELECTRÓNICA* 147
V.5 DESCRIPCIÓN GENERAL DE BLOQUES DE MÓDULOS 149
V.6 CONTROL DE LA *INTERFAZ* 151
V.7 DIAGRAMAS DE FLUJO DEL SOFTWARE EL MICRO CONTROLADOR 153
V.8 LISTADO DEL CÓDIGO FUENTE DEL PROGRAMA DEL PIC16F84A 157

V.9 TARJETAS DE LA *INTERFAZ ELECTRÓNICA* 159
V.9.1 Tarjeta principal *160*
V.9.1.1 Lista de materiales 161
V.9.2 Tarjeta de bloque *162*
V.9.2.1 Lista de materiales 163
V.9.3 Materiales complementarios *163*
V.10 CONFIGURACIÓN DEL CABLE DE CONEXIÓN ENTRE LA COMPUTADORA Y LA *INTERFAZ ELECTRÓNICA* 164
V.11 PROTOTIPO DE PRUEBA 165

CONCLUSIONES 168

AMPLIACIONES, CONTINUIDAD Y MEJORAS DEL SISTEMA 171

MANTENIMIENTO AL SISTEMA 173

APÉNDICE A: TARJETAS PCB 174

TARJETA PRINCIPAL *174*
PCB DE LA TARJETA DE BLOQUE *175*

APÉNDICE B: PRUEBAS, MEDICIONES E IMPLEMENTACIÓN 176

PRUEBAS DE IMPLEMENTACIÓN 180

ANEXO: INSTRUCTIVO DE USO DEL SISTEMA 185

Menús de Opciones *187*
Menú Archivo 187
Menú Consultas 188
Menú Estadísticas 189
Menú Usuarios 190
Menú Administración 191
Menú Inventario 193
Opción Ayuda 193
Opción Acerca de 194
Barra de identificación y búsqueda *195*
Barra de título *197*
Reloj *197*
Indicador de conexión de la interfaz electrónica *197*
Botones de estado *198*
Panel de Control *198*
Iconos de los módulos del Panel de Control 198
Menús contextuales del Panel de Control 203
Menú contextual de un módulo libre 204
Menú contextual de un módulo forzado no asignado 208
Menú contextual de un módulo ocupado 209
Menú contextual de un módulo forzado y ocupado 215
Menú contextual del área del Panel de Control donde no se encuentran los módulos 215
Paneles complementarios *217*
Barra indicadora del uso del panel *220*
Consideraciones de la operación del Sistema con la Interfaz Electrónica *221*
Pérdida de comunicación con la *Interfaz Electrónica* 221
Icono en la barra de tareas de Windows *223*
Derechos por categoría de Operador *228*

BIBLIOGRAFÍA 230

OBJETIVOS

Objetivo general

Desarrollar un programa de cómputo que sea estructurado de manera conjunta con una interfaz electrónica que permitan brindar a los administradores (operadores) del Laboratorio Abierto de La Facultad de Ingeniería una herramienta que coadyuve en las tareas cotidianas que implican el uso óptimo, eficiente y equitativo de los bienes con que cuenta el Laboratorio para brindar a los usuarios los servicios que en él se brindan de manera ágil, equitativa y oportuna para sus tareas, proyectos o trabajos de investigación que les sean encomendadas por parte de los profesores de las distintas asignaturas del área electrónica que son consideradas en su plan de estudios y en consecuencia, mejorar la administración del Laboratorio y sus recursos.

Objetivos específicos

La tarea del software será la de registrar en una *Base de Datos* los detalles de los movimientos que detallen el uso que los usuarios hacen de las instalaciones y de los servicios que en el Laboratorio se proporcionan.

El software estará provisto de una interfaz "amistosa" e intuitiva para su operación con la finalidad de proporcionar a los operadores del mismo una manera ágil para el apoyo en las tareas cotidianas de la administración del Laboratorio.

Adicionalmente, serán incluidos distintos menús de opciones y contextuales mediante los cuales los operadores tendrán al alcance de un solo clic del mouse del equipo de cómputo, la gama de opciones que se consideraran en la elaboración de este proyecto.

El sistema tendrá un mecanismo de ajuste de parámetros relacionados a la cantidad de usuarios máxima por módulo, tiempo en horas que se les brindará el servicio a los usuarios y tiempo en minutos previos de aviso al operador cuando el tiempo de servicio en los módulos se encuentre próximo a vencer, entre otros parámetros asociados al uso del sistema.

Se incluirá un mecanismo de sanciones a los usuarios que infrinjan el reglamento interno del Laboratorio; su aplicación estará sujeta al juicio del administrador en turno.

Los operadores del Sistema se clasificarán en tres tipos: Súper-administrador, administrador y operador normal. Cada una de estas categorías tendrá privilegios de acceso a las distintas opciones del Sistema acorde a la categoría a la que pertenezcan.

Se incluirán distintas opciones de consulta sobre los movimientos registrados por usuario, operador, módulos, lista de espera y de los usuarios que se encuentran dentro de las instalaciones. Así mismo, se incluirán distintas opciones de estadísticas en las que se proporcione a los operadores una manera de cuantificar el uso de los recursos del Laboratorio, permitiendo exportar los datos generados con la finalidad de poder hacer el análisis cuantitativo de los movimientos en programas como Excel.

El Sistema se construirá sobre el principio de registro de transacciones detalladas sobre cada una de las operaciones registradas con la finalidad de supervisar la operación del Sistema a todos los niveles. Este registro permitirá dar seguimiento a cada una de las

acciones que son realizadas dentro de la operación cotidiana de la administración del Laboratorio y sus recursos.

Se proveerá de un conjunto de herramientas administrativas en las que se pretende que los operadores puedan dar mantenimiento al Sistema para garantizar su correcta operación de manera ágil y segura, así como el control de las cuentas de operadores, realizar un proceso de ajuste de parámetros mediante el cual se podrán incorporar al Sistema nuevas agrupaciones del tipo de bienes del Laboratorio, carreras, ubicación de los bienes del Laboratorio, entre otros. También se incluirán opciones enfocadas a preservar la integridad de la información de los operadores, creación de registro de transacciones para su análisis y seguimiento posterior, administración del panel de módulos de manera rápida y concisa y opciones para regular la operación de la *interfaz electrónica*.

Se incluirán opciones que estarán enfocadas al registro de los usuarios del Laboratorio mediante las que se podrán agregar nuevos usuarios que no se encuentren registrados previamente, actualizar sus datos básicos de manera manual o automática cada 6 meses en el momento en que el usuario requiera de los servicios del Laboratorio y se determine que la información registrada tiene una antigüedad de por lo menos 6 meses. Adicionalmente, se podrá exportar el catálogo de los usuarios registrados en el Sistema a un formato que pueda ser manipulado externamente desde otra aplicación; y, el proceso inverso que será el de importar masivamente el catalogo de usuarios del Laboratorio para mantener actualizada la *Base de Datos* de los mismos o para realizar un respaldo de la misma.

Para complementar el conjunto de herramientas, se dotará al Sistema de un mecanismo de registro y control de los bienes del laboratorio; desde aquí se podrá registrar en la *Base de Datos* a los bienes con que cuenta el Laboratorio para poderlos poner a disposición de los usuarios del Laboratorio en calidad de préstamo, manteniendo un estatus de las condiciones generales de los bienes, su ubicación física, su número de inventario, número de serie en caso de contar con éstos y, para propósito del Sistema, les será asignado un número único para su control de manera independiente de los datos ya indicados. Para complementar esta característica, se incluirá una opción dedicada al estado que guardan los equipos e instrumentos de los módulos, con la finalidad de reflejar su estatus en el panel de control del Sistema en el que se indicará que alguno de los equipos o instrumentos se encuentra en reparación o se encuentra con un estatus de descompuesto. Se incluirán también una opción de exportación de los datos correspondientes al inventario en conjunto para su uso externo del Sistema y de manera inversa para incorporar masivamente al inventario de los bienes del Laboratorio según un formato que será definido en este trabajo.

Como es natural, se incluirá un manual de ayuda en el que se explicaran los detalles de todas las opciones incluidas en el Sistema de manera que pueda ser consultado en cualquier momento desde el Sistema o desde el grupo de programas en donde se instalará el Sistema desde el menú de opciones del Sistema Operativo Windows.

El Sistema no estará necesariamente en una posición maximizada o visible siempre sobre todo en los momentos de pocas acciones a registrar, motivo por el cual se incluirá la característica de permanecer minimizado en la barra de tareas del Sistema Operativo en el que se incluirá un ícono que simbolice al Sistema y se pueda acceder a las opciones más comunes a partir un menú contextual para simplificar las tareas que son objeto de este trabajo.

Se diseñará una *interfaz electrónica* que será controlada por el software cuya tarea esencial será la de fungir como un switch electrónico que permitirá controlar el flujo eléctrico hacia los módulos de trabajo para reflejar las condiciones de operación sobre los módulos de manera

particular. Con esto se pretende reducir el consumo eléctrico en momentos en que los equipos no se encuentren ocupados con la finalidad de coadyuvar a concientizar a los usuarios sobre el grave problema que enfrentamos en cuestión del calentamiento global y en segundo término, aumentar la vida útil de los equipos con que cuenta este Laboratorio.

En síntesis, la finalidad en conjunto del ***Sistema de Administración Electrónica Aplicado al Laboratorio Abierto (de la Facultad de Ingeniería)*** estará enfocado a la organización y administración de los bienes del Laboratorio y el uso que se haga de ellos por parte de los usuarios y de la manera en que son administrados por los operadores del mismo en un esfuerzo organicista de la operación de los equipos e instalaciones como tal.

INTRODUCCIÓN

La finalidad de este trabajo es la de ofrecer una solución a la situación que en la actualidad guarda la administración y uso del Laboratorio Abierto de Electrónica de la Facultad de Ingeniería (en lo sucesivo: *Laboratorio*); situación que en la actualidad se ha venido manifestando sobre el estado que guarda el equipo, mobiliario y herramientas con que éste cuenta.

El registro de los movimientos se han mantenido por mucho tiempo de manera manual, sin mayor control que el de colocar en un casillero de cartón a las credenciales de los alumnos (en lo sucesivo *usuarios*) que hacen uso del Laboratorio asociado al número de módulo que se les asigna al momento en que solicitan el uso de las instalaciones o de los servicios que se brindan en el Laboratorio como apoyo y complemento a las tareas y proyectos que coadyuvan al aprendizaje de lo que en la parte teórica de las asignaturas les es enseñado por parte de los profesores responsables de la impartición de cátedra en las distintas áreas que implican a la electrónica de hoy en día.

La situación antes mencionado ha implicado en ocasiones el extravío de materiales, descomposturas del equipo, inconformidad por parte algunos de los usuarios por la subutilización de los recursos y el tiempo de espera de que éstos tienen que invertir para poder acceder a las instalaciones en momentos críticos en los que el Laboratorio se encuentra a su máxima capacidad y que por dicha situación no les es posible tener acceso en el momento en que requieren de los servicios del Laboratorio.

De esta situación, a lo largo de éste trabajo se irá describiendo a detalle las situaciones aquí expresadas y, ofreciendo una solución que en conjunto se ha llamado **SAEALABFI** (Sistema de Administración Electrónica Aplicada al Laboratorio Abierto de la Facultad de Ingeniería y que simplemente lo nombraré así: **SAEALABFI** o Sistema). En esta solución se han recabado las opiniones y necesidades de quienes son encargados de administrar los recursos y uso del Laboratorio en conjunto, así como los puntos clave o medulares que del análisis del problema se han identificado.

Durante la elaboración de éste trabajo, se realizaron tres cambios importantes al **SAEALABFI**, teniendo una primer versión para su prueba, observación y ajuste de los productos que ofrece éste por allá del mes de agosto de 2008, versión que fue bautizado como la versión 1.0.0000.0; mientras que a finales del mes de noviembre del mismo año, se desarrolló una segunda versión de liberación denominada versión 1.1.0000.0, en ésta liberación se incluyen los cambios correspondientes a las observaciones y peticiones realizadas por las personas encargadas de la administración del Laboratorio, así como la optimización en la ejecución del mismo y la inclusión de nuevas funciones; estas dos versiones sólo consideran la parte operativa del **SAEALABFI** y aún no se consideraron los detalles de la implementación del protocolo de comunicación y control de la *interfaz electrónica* ya que se suscitaron diversas situaciones técnicas que ocasionaron el estancamiento en su elaboración durante la etapa de prototipo de la misma.

Después de pruebas en la elaboración y operación del prototipo de la *interfaz electrónica* que se encargará de realizar las funciones de alimentación de energía eléctrica a los módulos del Laboratorio mediante las señales que el **SAEALABFI** le envía por uno de los puertos seriales (se tomó la decisión de emplear el puerto serial por las razones que más adelante se utilizaron para justificar), se ha liberado la versión 1.2.0000.0 en la que ya se incluye el envío de señales y recepción de respuestas de la *interfaz electrónica*, así como el constante chequeo de que ésta se encuentra conectada y alimentada eléctricamente para asegurar el

correcto funcionamiento del Sistema en su conjunto.

El presente trabajo se encuentra acompañado de dos discos en formato CD cuyo propósito es el de entregar en el disco 1 al **SAEALABFI** como instalador del mismo; la instalación está provista de todos los elementos necesarios para que se pueda personalizar dicha instalación, además de que incluye una versión electrónica del manual de operación del **SAEALABFI** que al término de este trabajo está conformada por 276 páginas en las que se detallan las distintas características del Sistema y que en el anexo de este trabajo se resume como un instructivo de la aplicación. En el disco 2 se incluyen los códigos fuente del Sistema, el código del programa que se emplea para implementar la comunicación entre la PC y la *interfaz electrónica* mediante el uso de un micro-controlador (**PIC16F84A**), así como los diseños de las etapas de la *interfaz electrónica* en formatos PCB y los materiales utilizados para la culminación de este proyecto.

CAPÍTULO I: EL LABORATORIO ABIERTO

Es momento de comenzar con la descripción de todos los elementos que componen a la sustancia que es el origen de este trabajo, para ello, en este Capítulo se describe la estructura interna tal y como se encuentra constituido y distribuido el Laboratorio Abierto y como complemento a esta descripción, se detalla la problemática que ha venido presentándose en él, así como una propuesta de solución que es tema de esta tesis.

I.1 Estructura y composición del Laboratorio Abierto

La División de Ingeniería Eléctrica y Electrónica (**DIEE**) cuenta con diversos Laboratorios y equipos especializados con la finalidad de impartir en ellos las distintas asignaturas relacionadas con la electrónica; en ellos se hacen prácticas específicas de cada área en particular.

Destaca de entre los Laboratorios el Laboratorio Abierto de Electrónica que se encuentra ubicado en las instalaciones del anexo de Ingeniería dentro del edificio Luis G. Valdés Vallejo, primer nivel, ala Norte. Ha sido creado para proporcionar a los usuarios, servicios de asesoría y apoyo en las labores de construcción y prueba de prototipos que les son solicitados por parte de los profesores con quienes se encuentran cursando alguna de las asignaturas impartidas en la Facultad de Ingeniería que implican a la electrónica analógica, digital y de potencia entre otras y sin importar qué carrera esté cursando cada usuario de los servicios del Laboratorio, estos servicios están encaminados a que los usuarios realicen todo tipo de actividades que no son consideradas ni tampoco guiadas o supervisadas por los profesores que les imparten el aspecto teórico y práctico en particular de cada una de las asignaturas antes referidas. En la fotografía I-1, se muestra la fachada del edificio Luis G. Valdés Vallejo.

Fotografía I-1. Fachada del edificio Luis G. Valdés Vallejo

I.1.1 Laboratorio abierto de la Facultad de Ingeniería

El Laboratorio Abierto tiene una distribución de agrupaciones de equipos especializados de instrumentación para ofrecer a los usuarios los distintos servicios que se les son proporcionados. En la Ilustración I-1 se muestra la distribución del Laboratorio Abierto.

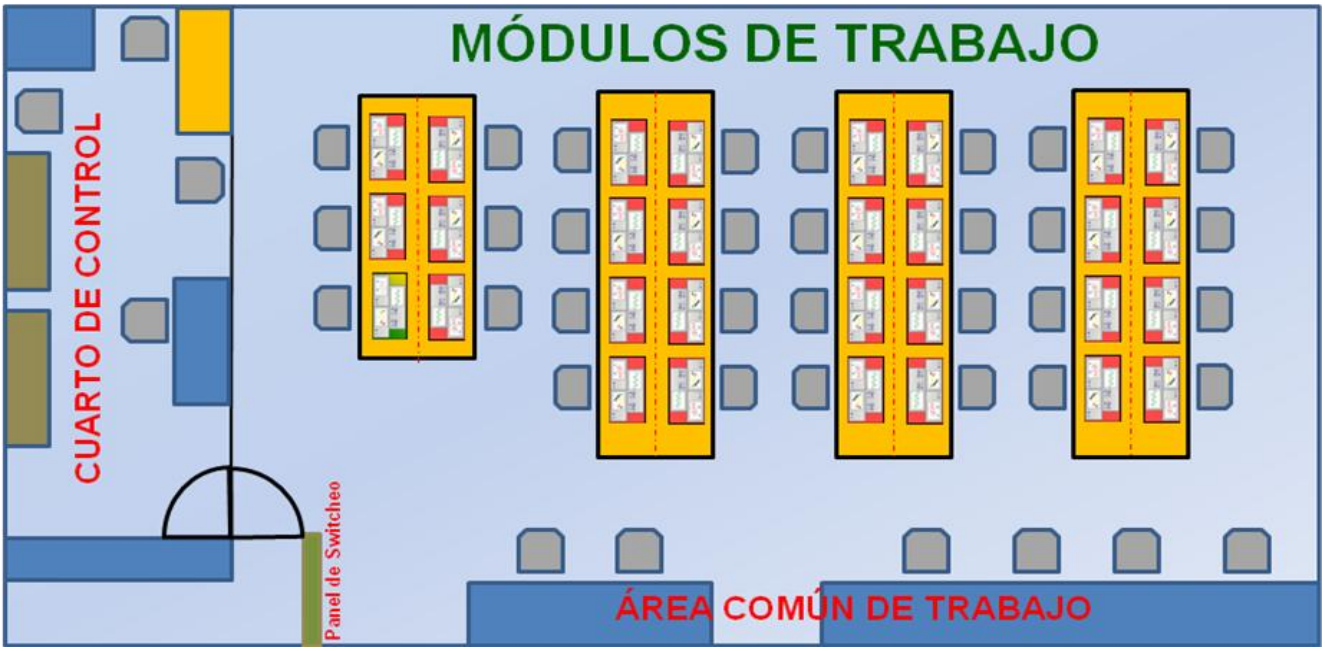


Ilustración I-1. Distribución del Laboratorio Abierto

En la Ilustración anterior resaltan cuatro áreas que se describen a continuación:

- ***Cuarto de Control***. Es aquí donde se encuentran los encargados del Laboratorio para tener un panorama de las instalaciones, así como para tener el control del acceso de los usuarios. Dentro de ésta área se encuentran unos anaqueles y un archivero en los que se resguardan los distintos manuales impresos de los dispositivos electrónicos utilizados con mayor frecuencia. Junto a la puerta de acceso al cuarto de control, se encuentra una pequeña barra en la que se encuentran organizadas las distintas herramientas con que cuenta el Laboratorio para ser prestadas a los usuarios. Se cuenta con una mesa en la que los Administradores o responsables del Laboratorio pueden contener sus documentos, la caja contenedora de credenciales de los usuarios y para colocar sobre ella un equipo de cómputo portátil y en la cual se colocará el equipo de cómputo que será utilizado en la implementación de la solución aquí propuesta. Las fotografías I-2 e I-3 muestran algunos de los componentes del cuarto de control.

Fotografía I-2. Anaqueles con documentos

Fotografía I-3. Contenedor de manuales

- ***Panel de switcheo***. Éste es un panel de conexiones que se encuentra empotrado en la pared que se encuentra al lado derecho del acceso al Laboratorio; desde este panel se puede controlar el suministro eléctrico de manera independiente a cada uno de los módulos de trabajo del Laboratorio. Es necesario mencionar que el suministro se encuentra de manera permanente a no ser que los Administradores del Laboratorio lo cancelen mediante una pastilla o switch. El aspecto que tiene dicho panel se muestra en la fotografía I-4.

- ***Área común de trabajo***. Esta área está conformada por 5 mesas de propósito general, en éstas, los usuarios pueden colocar sus pertenencias, elaborar sus tarjetas electrónicas (fenólicas) e incluso se cuenta con un recipiente donde los usuarios pueden colocar los residuos de cloruro férrico utilizado para su elaboración. Al fondo de las instalaciones (lado derecho de la Ilustración I-1), se encuentra un taladro y al frente de las instalaciones, se cuenta con dos pizarrones como material de apoyo para los casos en que los profesores reservan el uso del Laboratorio para citar allí a sus alumnos con el propósito de hacerles alguna explicación, revisión o evaluación de las actividades propias y complementarias de la asignatura que les imparte.

Fotografía I-4. Panel de switcheo

- ***Módulos de trabajo***. Esta es el área principal del Laboratorio y es aquí en donde se encuentran distribuidos los equipos de instrumentación que se les proporcionan en calidad de préstamo interno a los usuarios que solicitan los servicios del Laboratorio. El área está conformada por un total de 30 módulos de los cuales el primero está instalado para ofrecer a los usuarios apoyo en tareas de programación de memorias, micro-controladores y otros dispositivos electrónicos; un taladro de mesa y, un medidor de inductancia. Mientras que los otros 29 módulos son de propósito similar y se encuentran equipados con: una fuente de alimentación eléctrica; un multímetro digital, un generador de funciones y un osciloscopio de doble canal. Todos los equipos se encuentran colocados sobre una mesa perpendicular (según la perspectiva mostrada en la Ilustración I-1) agrupados en 4 conjuntos por lado, el primero contiene 3 módulos por lado por lo que en éste se encuentran 6 módulos; los otros 3 conjuntos se encuentran en mesas duales en las que se encuentran colocados 4 módulos por lado, haciendo un total de 8 módulos. De esta manera, se tienen distribuidos a los 30 módulos, de manera complementaria, cada módulo se encuentra identificado con una banderilla metálica con el número de módulo al que corresponden los equipos identificados sobre las mesas de

trabajo. Para que la estancia de los usuarios sea lo más confortablemente posible, se cuenta con bancos de trabajo suficientes. En la fotografía I-5 se muestra el aspecto del módulo de trabajo 1, mientras que en la fotografía I-6, se muestra el aspecto de los restantes 29 módulos.

Fotografía I-5. Aspecto del módulo de trabajo 0

Fotografía I-6. Aspecto de los módulos 1 a 29

I.2 Descripción funcional sobre los servicios proporcionados por el Laboratorio Abierto de la Facultad de Ingeniería

Los Administradores del Laboratorio Abierto de la Facultad de Ingeniería se encuentran Facultados para ofrecer a los usuarios apoyo en la elaboración de sus prototipos y/o tareas, pero solo en forma de asesoría, de revisión y observación sobre el avance de las tareas propias de cada usuario o equipo de usuarios, siempre y cuando los usuarios sepan con

claridad lo que van a realizar y teóricamente, como deberán resolver lo que se les ha encargado.

El Laboratorio Abierto abre las instalaciones para brindar sus servicios a los usuarios en un horario de 9:00 hrs a 19:30 hrs de lunes a viernes, este horario se ha distribuido en dos turnos entre los Administradores del Laboratorio: el primer turno está establecido de 9:00 hrs a 14:00 hrs y el segundo turno de 14:00 hrs a 19:30 hrs.

Algunos profesores encargan a los Administradores revisar prácticas o proyectos en cuyo caso se acude al módulo en que se encuentran trabajando los usuarios para verificar que se haya cumplido con el objetivo de la práctica o trabajo y, al final se coloca un sello de revisión en el documento de la práctica para constatar que se ha cumplido con el objetivo y corresponderá entonces al profesor dar seguimiento a las actividades complementarias de sus asignaturas.

I.2.1 Servicios bridados por el Laboratorio Abierto

Los Administradores del Laboratorio Abierto, además de las funciones de supervisión, control y distribución del acceso a las instalaciones otorgan los siguientes servicios:

- Evaluación de tareas (cuando son Facultados para éste propósito)
- Préstamo de herramientas
- Préstamo de manuales
- Asignación de módulos
- Asesoría general y específica sobre dispositivos electrónicos
- Asesoría general y específica sobre conceptos y conocimientos relacionados con los antecedentes necesarios para poder realizar la elaboración de tareas y/o proyectos

I.2.2 Facultades de los Administradores del Laboratorio

Los Administradores además de tener las obligaciones ya indicadas en los párrafos anteriores, tienen la Facultad de observar y hacer que los usuarios acaten los lineamientos generales del uso de las instalaciones. De esto se puede desprender que a juicio personal puede aplicar sanciones a los usuarios y solicitarles si es necesario que salga del Laboratorio por infringir las normas en las que está basada la operación del Laboratorio, así como la cancelación del suministro de energía eléctrica al módulo que tiene asignado el usuario infractor.

I.2.3 Procedimiento para otorgar servicios a los usuarios

Cuando un usuario o grupo de usuarios acuden a solicitar los servicios del Laboratorio, primeramente se pide de manera oral y personal al Administrador en turno del Laboratorio le asigne algún módulo de trabajo para elaborar las pruebas de medición, instrumentación y prueba de los prototipos por los cuales solicitan los servicios del Laboratorio.

Una vez realizada la petición de servicio, el Administrador verifica la disponibilidad de módulos y si hay alguno libre, ofrecerá al usuario o grupo de usuarios alguno de los módulos disponibles, considerando las condiciones que guardan los instrumentos de dicho módulo. En otras palabras, se verifica si alguno de los instrumentos se encuentra descompuesto y aún así los usuarios pueden trabajar con el resto del equipo, se les hará la asignación de dicho módulo.

Al serles asignado algún módulo a los usuarios, se les solicitará que deje en depósito una identificación, que generalmente es la credencial que les acredita como alumnos de la Facultad de Ingeniería y por consiguiente de la Universidad.

Al recibir las identificaciones de los usuarios, el Administrador las colocará en un contenedor de cartón en el que se ha construido un casillero para cada uno de los módulos, la apariencia que muestra el contenedor se observa en la Fotografía I-7 y como podrá percibir, en cada uno de los casilleros se anexa además de las identificaciones, trozos de cartón que contienen leyendas alusivas a la situación que guardan los módulos cuando se reporta la descompostura en alguno de los instrumentos, cuando le es prestado algún material, manual o herramienta a cualquiera de los integrantes del equipo al que le ha sido asignado el módulo de trabajo.

Fotografía I-7. Contenedor de identificaciones

En el caso contrario, cuando un usuario o grupo de usuarios termina de utilizar algún material, manual o herramienta y cuando han dejado de utilizar el módulo de trabajo acuden con el Administrador del Laboratorio les devolverá el material o para solicitar les sean devueltas las identificaciones que han dejado en depósito al momento de ingresar a las instalaciones del Laboratorio.

I.2.4 Demanda de los servicios del Laboratorio

Los servicios que se brindan en las instalaciones del Laboratorio no siempre muestran la misma demanda y carga de trabajo; para los Administradores del mismo. Al principio de los semestres, como es de suponer se presenta una baja o nula demanda.

A medida que transcurre el tiempo que comprende el semestre en curso, la demanda de servicio

muestra momentos de carga como es el caso en que se aproximan los periodos de evaluación e incluso en ellos mismos, al pasar el momento de las evaluaciones, la demanda tiende a disminuir hasta que alcanza un momento de uso con tendencia a estabilizarse y en seguida comienza a aumentar al aproximarse el siguiente periodo de evaluación. Esto sucede para los tres periodos de evaluación en los que se dividen los semestres.

La mayor intensidad de la demanda de servicio se muestra cuando se ha llegado el tercer periodo de evaluación y cuando éste termina, la demanda se estabiliza y en un periodo breve se dispara para afrontar las evaluaciones finales.

Al concluir las evaluaciones finales, la demanda se dispara hasta registrarse momentos de demanda nulos y que coinciden con el fin del semestre.

El comportamiento descrito en estos párrafos, se puede apreciar con mayor claridad en la Ilustración I-2.

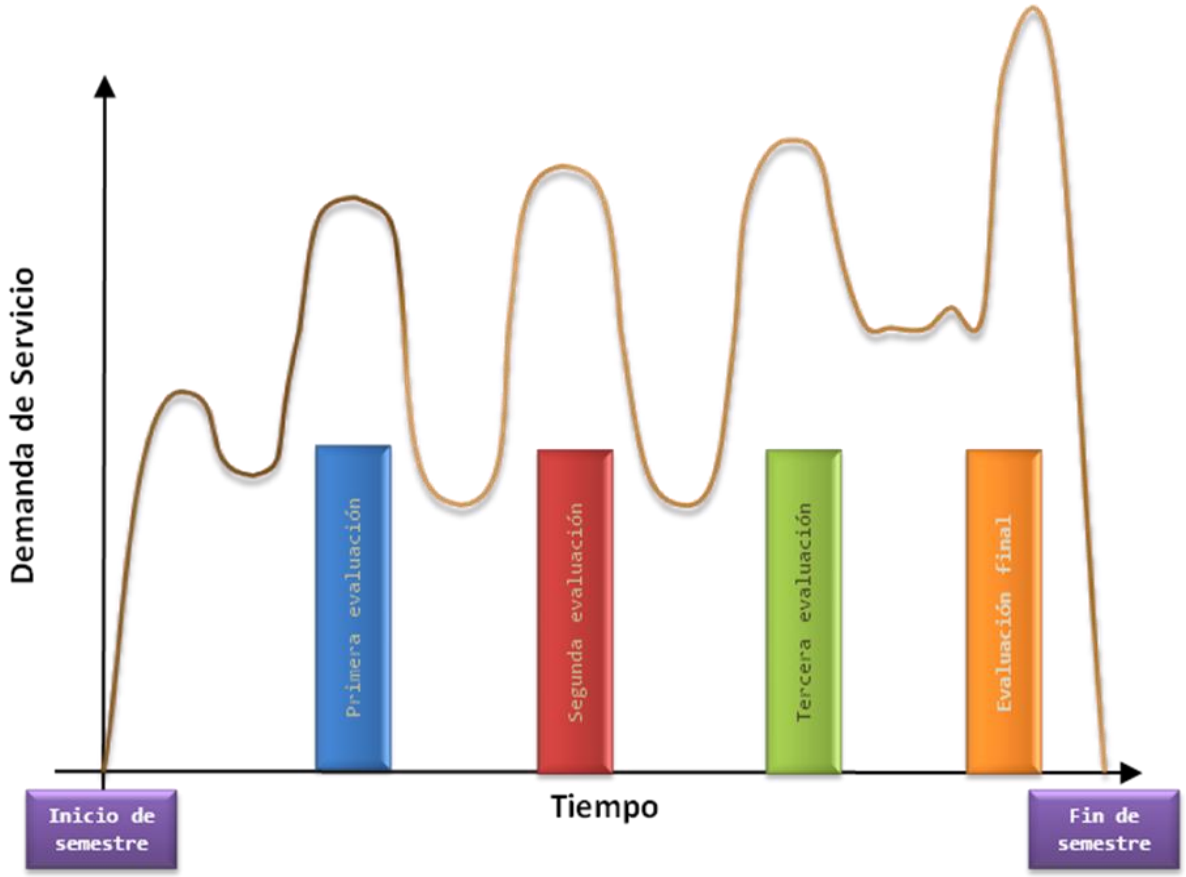


Ilustración I-2. Diagrama de demanda de servicio durante un semestre

I.3 Análisis de los principales problemas a resolver

Como ya ha sido mencionado con anterioridad, la finalidad de esta propuesta es la de ofrecer una solución a las distintas situaciones que se han venido manifestando dentro de las instalaciones del Laboratorio Abierto de la Facultad de Ingeniería.

Aunado a lo ya indicado, resulta importante considerar la situación actual que se vive día a día en nuestro país y en el planeta en general, con esto me refiero a la necesidad de reducir

la emisión de gases a la atmósfera que de manera acelerada han incrementado el efecto invernadero. Se preguntaran por que se dice esto en el contexto de un tema que aparentemente no tiene nada que ver con la problemática global, pero el correcto uso de las instalaciones del Laboratorio permitirá ahorrar energía eléctrica y con ello de manera implícita se tiene la posibilidad de contribuir con nuestro "granito de arena" para frenar la intensidad del fenómeno ya citado.

Lo primero que hay que considerar dentro de la problemática que se presenta en el Laboratorio Abierto, es que todo el control y registro de acceso y uso de los distintos materiales y equipos del Laboratorio se controla de manera manual como ya se ha indicado en el apartado I.2.3 Procedimiento para otorgar servicios a los usuarios. Esta situación en ocasiones no permite saber con certeza lo que cada usuario se encuentra realizando en las instalaciones del Laboratorio ya que por un lado, los Administradores del Laboratorio se encuentran apoyando a los usuarios, recibiendo a los usuarios que se acercan a solicitar el servicio; se tiene que considerar la situación que guardan los distintos equipos de los módulos, además de atender las distintas situaciones que implica la labor que realizan día a día.

Esta situación ha implicado que a través del tiempo en que el Laboratorio tiene ofreciendo sus servicios, se han presentado extravíos de materiales y descomposturas de los equipos. Aunado a esto, cuando se acercan los momentos de evaluaciones parciales (I.2.4. Demanda de los servicios del Laboratorio), la demanda se incrementa y ocasiona que la capacidad de servicio del Laboratorio se vea superada por la cantidad de módulos instalados en las instalaciones y por consecuencia, la cantidad de usuarios que en ese momento son atendidos.

Cuando la demanda de los servicios excede a la capacidad de las instalaciones del Laboratorio, se han suscitado inconformidades por parte de los usuarios que al no ser atendidos han presentado quejas por observar que algunos módulos se encuentran subutilizados pues los usuarios que están asignados a los módulos en ocasiones salen de las instalaciones o no se encuentran en el lugar que se les asignó y los querellantes reclaman la atención ante la observación de la situación que se ha indicado.

Como consecuencia de la situación aquí expresada, la supervisión de la operación de los equipos y de las instalaciones en general, no ha sido suficiente para evitar daños en los equipos aunque la **DIEE** cuenta con prestadores de servicio social a quienes le es encomendada la tarea de reparación de los instrumentos dañados.

Otra situación que se ha observado actualmente con mayor frecuencia es el uso de la tecnología móvil y se ha convertido en algo "normal"; se ha presentado la situación en la que los usuarios introducen a las instalaciones del Laboratorio equipos de cómputo tipo lap-top; por sí solo, esto no representa ningún problema aparente, pero en el momento en que los usuarios utilizan a los instrumentos de los módulos para depositar o acomodar allí a su computadora y para escuchar música a un volumen que ya no es para uso "personal", entonces sí se convierte en un problema porque lo que de manera implícita se pretende realizar es la mejora de los servicios y de la vida útil de los instrumentos y equipos con que está dotado el Laboratorio Abierto de Electrónica. En la fotografía I-8 se aprecia el uso de equipos portátiles de cómputo dentro de las instalaciones del Laboratorio Abierto.

Los alumnos no observan las medidas que se han establecido para el uso de los servicios del Laboratorio; muchos de ellos entran y salen de las instalaciones casi de manera habitual, en ocasiones se concentran varios usuarios en un solo módulo de trabajo, superando a los 2, 3 o hasta 4 que son los máximos permitidos por la naturaleza y estructura propia de las instalaciones.

En ocasiones, hay alumnos que se encuentran ocupando las instalaciones realizando actividades ajenas a los servicios que se ofrecen en el Laboratorio, motivo por lo cual ha llegado a suscitarse desconfianza o descontento entre los usuarios del mismo así como de quienes administran el Laboratorio por no poder ofrecerse el servicio de manera equitativa y oportuna a todos los usuarios.

Fotografía I-8. Usuarios con computadoras portátiles en las instalaciones del Laboratorio

Lo anterior aparentemente no presenta mayor problema cuando hay disponibilidad de equipos para ofrecerlos a los solicitantes, pero esto no es posible cuando la demanda de atención y servicio aumenta como ya ha sido indicado en el apartado I.2.4, en los momentos de evaluación parcial o final por parte de los profesores de las asignaturas correspondientes.

En resumen, es necesario que de manera conjunta a la implementación de la solución que se propone, se haga de conocimiento general a los usuarios los lineamientos o normas que se deben de considerar al encontrarse dentro de las instalaciones. Además de concentrar en una Base de Datos la información básica los usuarios del Laboratorio para poder tener un control más preciso sobre el uso que se va realizando de las instalaciones, así como de los materiales y herramientas con que se cuenta. Como consecuencia, se tendrá un mejor control para el acceso de los usuarios a las instalaciones, se observarán con precisión los tiempos máximos de uso de los módulos, se controlará la cantidad máxima de integrantes en los módulos, se podrá tener una lista de espera cuando la capacidad máxima de las instalaciones esté ocupada en su totalidad y quienes se encuentren en esta lista serán atendidos de manera automática y selectiva en la medida en que se vayan desocupando los módulos de trabajo y lo más importante es que se llevará un control riguroso sobre el uso de la energía eléctrica y bienes del Laboratorio.

Además, como resultado de estas situaciones que más adelante se detallarán, se ha considerado tener un inventario de todos los manuales, herramientas y equipos con los que cuenta el Laboratorio, para poder ubicarlos en el momento preciso que se requiera y, en conjunto en un Sistema de cómputo se concentrará y registrará toda la información que de todas las situaciones aquí manifestadas se han analizado.

Otro punto importante a considerar es que se deben de considerar medidas de sanciones a los usuarios al no cumplir con los nuevos lineamientos que se deberán de considerar al implementar esta solución que aquí se expone.

El Sistema como resultado de todos los registros de movimientos, el Sistema estará en la posibilidad de ofrecer a los Administradores del Laboratorio productos y/o servicios de información para el análisis y control de la operación ordinaria y extraordinaria del Laboratorio Abierto, así como para poder imponer sanciones a los usuarios infractores de las normas de operación del Laboratorio.

CAPÍTULO II: Solución a la problemática presentada

La propuesta que aquí se plantea, es una propuesta integral en la que se ha considerado en conjunto a la parte lógica y física que en conjunto otorgarán a los Administradores del Laboratorio las herramientas necesarias para poder realizar de manera más eficaz, organizada y sobre todo, con la certeza de que las situaciones que se han manifestado en el Capítulo anterior no se repitan ni en la más mínima frecuencia con que se ha manifestado hasta el día de hoy.

II.1 Antecedentes de la solución

Como antecedentes a esta propuesta que también se llama insumos, se tiene claro que lo que se va a implementar tiene que ver con la manera en que se ha operado la Administración del Laboratorio y partiendo de ello, en esta propuesta se consideran los tiempos de servicio que, con el apoyo mostrado en la Ilustración I-2. Diagrama de demanda de servicio durante el semestre, se ha de considerar que la duración del servicio tiene que tener la posibilidad de ser ajustada durante el transcurso del semestre a medida en que éste transcurre, por lo que se considera que un tiempo máximo de servicio (deseable) en los momentos de máxima demanda debería de estar en unas 2 horas como máximo, con prorrogas configurables en incrementos de 15 minutos a juicio y consideración del Administrador del Laboratorio.

Otro punto importante que se debe considerar, es el número máximo de usuarios que se encuentran asignados a un módulo de trabajo; lo ideal sería que fuesen máximo 2 debido a que en las instalaciones no se cuenta con suficientes bancos en los que puedan sentarse más usuarios y, a que el Laboratorio se vería abarrotado por gran cantidad de usuarios que se encuentren haciendo uso de los servicios de las instalaciones. Esto no se debe considerar como una regla, pues también he considerado que en los momentos de máxima demanda, los usuarios generalmente acuden en equipos de trabajo para presentar algún proyecto, prototipo o tarea a los profesores que les imparten la teoría por la cual les es solicitado dicho encargo y por consiguiente, el Sistema deberá tener la posibilidad de recibir a más alumnos por módulo y que en este caso se están considerando a 4 como límite para ser tomados en cuenta dentro de la configuración, aunque éste caso sería una excepción.

Se está considerando que el Sistema, administre el suministro de energía eléctrica a cada módulo de manera independiente y esto lo irá haciendo en la medida en que se registren los movimientos de los usuarios dentro de las instalaciones y observando en todo momento las condiciones o parámetros anteriores, es decir: tiempo máximo de servicio por usuario o grupo de usuarios y la cantidad de usuarios por cada módulo.

Dentro de los tiempos de servicio, el Sistema cancelará el suministro eléctrico cuando el o los usuarios de un módulo no hayan solicitado una prorroga de tiempo de servicio o hayan acudido al cuarto de control a recoger las identificaciones que han dejado como depósito por el uso de las instalaciones.

El Sistema estará dotado de un mecanismo en el que se administren los bienes del Laboratorio mediante la implementación de un inventario de manuales, herramientas e instrumentos que se encuentra vinculado con el uso que los usuarios hacen de ellos con la finalidad de registrar los movimientos que se hagan de manera particular e individual de cada uno de los bienes.

Cada uno de estos bienes contará con un control de estatus que guardan, así como la ubicación en que éstos se encuentran debido a que en ocasiones algunos de los bienes se encontrarán en

reparación, algunos en calidad de préstamo interno a otros usuarios y otros más, en calidad de préstamo a otros usuarios o Laboratorios de la **DIEE**.

En la solución aquí planteada, se ha considerado una ventana principal en forma de panel en el que se representan de manera gráfica y simbólica a cada uno de los módulos instalados sujetos a ser ofrecidos a los usuarios como parte de los servicios que el Laboratorio otorga. Complementariamente se ha considerado tener 3 paneles auxiliares que operaran en las condiciones de trabajo dentro de la situación que impera en el Laboratorio. Dichos paneles se indican en seguida, así como su significado:

- **Panel de módulos próximos a vencer el tiempo de servicio**. En este panel como su nombre lo indica, se colocarán avisos a los Administradores del Laboratorio con anticipación de 5 minutos mostrando los módulos que se encuentran próximos a terminar el tiempo de servicio, desde este panel se ofrece a los Administradores la posibilidades de prolongar el tiempo de servicio en múltiplos de 15 minutos, es decir 15, 30, 45 ó 60 minutos más y, a criterio de los Administradores así como también quedará a responsabilidad de los mismos, ofrecer prorrogas posteriores bajo este mecanismo. Si no se atienden los avisos en este panel, llegado el término de tiempo límite de servicio, se procederá a la liberación del módulo y en consecuencia se cortará el suministro de energía eléctrica a dicho módulo.

- **Panel de materiales por entregar**. En este panel se colocará a los usuarios que teniendo bajo resguardo cualquiera de los bienes del Laboratorio, han terminado de utilizar el módulo que les haya sido asignado o, una vez terminado el tiempo de servicio no se le hizo una prórroga en el tiempo de servicio. Para que el usuario se pueda quitar de este panel, será necesario que éste se acerque al cuarto de control a realizar la entrega del bien o bienes que les fueron prestados y en ese momento se le regresará la credencial que deje como garantía para que los Administradores lo puedan quitar de la lista de este panel.

- **Panel de la lista de espera**. En este panel los Administradores tendrán la capacidad de poner en lista de espera a los usuarios que acudan a solicitar los servicios del Laboratorio cuando no haya módulos libres para que les pueda ser asignados algún módulo y serán atendidos cuando se desocupe algún módulo mediante el termino de tiempo de servicio o mediante la solicitud expresa del o de los ocupantes de los módulos. Cuando ocurra alguna de estas dos situaciones de manera automática será asignado el primer usuario que se encuentre en este panel motivo por el cual se quitará de este panel. Si el usuario solicitante desiste de esperar turno, los Administradores podrán quitarlo de la lista en este panel mediante las opciones que se le incorporaran para dicho propósito.

Además de estas características que se han incorporado a la solución de la problemática en el Laboratorio Abierto, se debe considerar que el Sistema en conjunto se encontrará conformado por dos partes: la parte lógica que corresponde con el software propiamente dicho y la parte física que será la *interfaz electrónica* que se encontrará conectada a un equipo de cómputo para realizar las tareas de administración de suministro de energía eléctrica de los módulos que se encuentran en el Laboratorio Abierto de Electrónica.

El equipo de cómputo en que se contendrá esta solución deberá encontrarse ubicada dentro del cuarto de control, mientras que la *interfaz electrónica* deberá ser ubicada cercanamente al panel de switcheo para que se puedan realizar las conexiones necesarias para el propósito de éste *interfaz*.

Partiendo de lo anterior, se ha considerado de manera adicional que la distancia física entre el equipo de cómputo y la *interfaz electrónica* es del orden de 7.5 metros y es importante considerarlo pues de esto depende la justificación del medio que se ha seleccionado para poder realizar la comunicación entre ambos equipos y que se detalla en el Capítulo dedicado al diseño del Software y de la *interfaz electrónica*.

II.2 Estructura de la solución

La solución que se ha propuesto a la problemática expuesta en el Capítulo I, se muestra en la Ilustración II-1. Estructura de la implementación de solución. En dicha Ilustración se observan las distintas etapas que dan sentido al conjunto de elementos que conforman esta solución.

A continuación se indican las etapas y la función de cada una de ellas:

- **Equipo de cómputo y software de implementación**. Sin duda alguna, este es el conjunto de componentes que serán necesarios y de suma importancia para la implementación de este proyecto pues de este conjunto depende en gran medida las capacidades de operación y registro de todos los movimientos y acciones que se tendrán en cuenta para el apoyo en las tareas de administración de las instalaciones del Laboratorio. El equipo que se utilice para este propósito deberá contar con una configuración mínima que permita adaptarse a las necesidades de crecimiento de la información que contendrá el Sistema en conjunto, asociada a cada uno de los usuarios que para tal propósito deberán registrarse. En el Capítulo dedicado a la estructura del Software, serán detalladas las características específicas que deberá tener este equipo.

  Dentro del equipo de cómputo se tendrán las herramientas necesarias para poder realizar esta tarea fundamental. Con estas herramientas se incluye al Sistema de Administración de Base de Datos que se empleará para poder integrar la solución en su conjunto.

- **Protocolo de conexión**. Este es el mecanismo mediante el cual el Software estará en condiciones de enviar señales eléctricas a la *interfaz electrónica* para controlar el suministro de energía electrónicas a los módulos del Laboratorio. La estructura que mantiene este protocolo se detalla en el Capítulo dedicado a la descripción del Software y a la *Interfaz electrónica*.

- ***Interfaz electrónica***. Es el prototipo conformado por distintos componentes electrónicos digitales y analógicos que se han escogido para conformar las tareas de ejecución de acciones (comandos) recibidos a través de un medio (conexión) entre un puerto de comunicación del equipo de cómputo y el conjunto de componentes electrónicos que se han seleccionado para este proyecto. En el Capítulo dedicado a la descripción de la *interfaz electrónica*, se mostrarán los detalles y conformación de este componente que servirá de puente entre el equipo de cómputo (mediante el Software) y las conexiones de suministro eléctrico en el panel de switcheo.

- **Panel de switcheo**. Este componente representa el medio esencial en las tareas de suministro de energía eléctrica a los instrumentos y equipos vinculados a cada uno de los módulos de trabajo en el Laboratorio. En la actualidad, este panel ya se encuentra construido y empotrado en una de las paredes del Laboratorio (concretamente en la pared que se encuentra a la derecha de la puerta de acceso a las instalaciones). Está formado por una pastilla-interruptor para cada uno de los módulos de manera independiente. Cada

interruptor está identificado mediante un número que corresponde al módulo ubicado en las instalaciones del Laboratorio.

Módulos en el Laboratorio. Esta sección corresponde a los distintos módulos instalados en el Laboratorio, cuya configuración y ubicación ya ha sido descrita en el apartado I.1.1. Laboratorio abierto de la Facultad de Ingeniería.

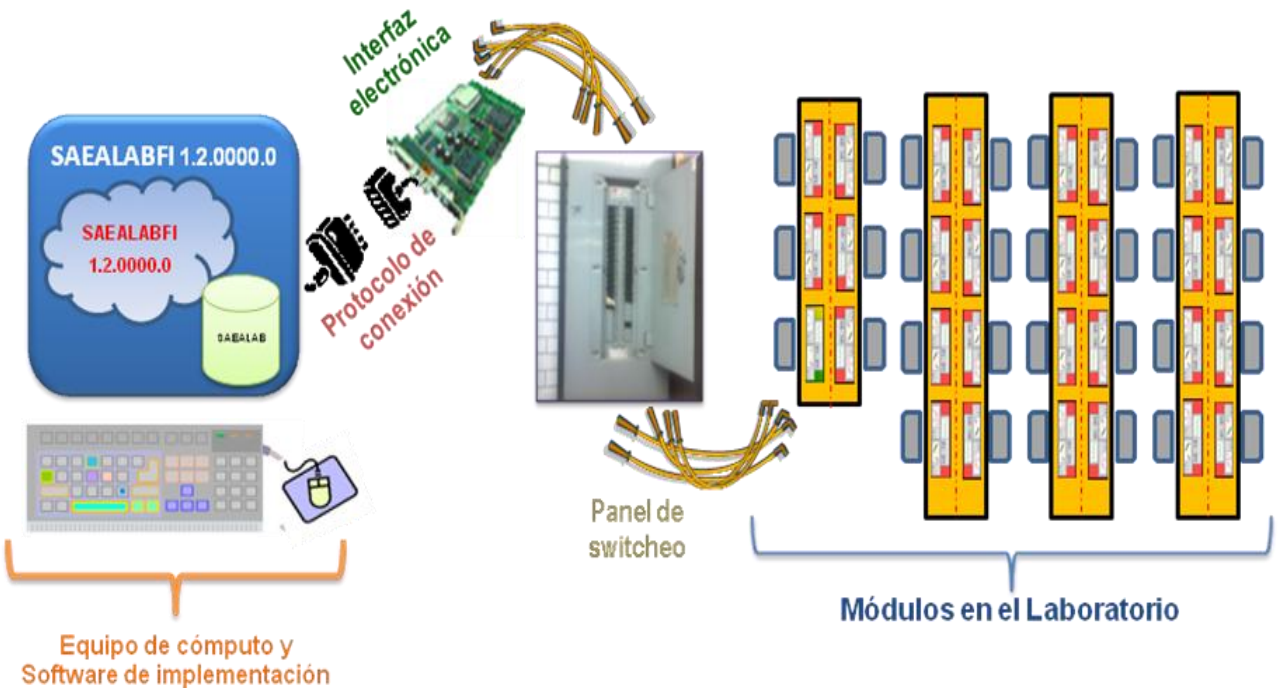


Ilustración II-1. Estructura de la implementación de solución

II.3 Software de desarrollo

El software de desarrollo está constituido por todas aquéllas herramientas comerciales de carácter público ya sea de distribución libre o de distribución bajo esquemas de licenciamiento con los que hoy en día se dispone para la realización de todas aquéllas aplicaciones de características particulares a las necesidades de los desarrolladores para instituciones públicas o privadas.

El software de desarrollo se encuentra enfocado a las distintas plataformas de cómputo y está vinculado a las características específicas de las aplicaciones de estos tiempos.

Por una parte, se encuentra el software que ha sido diseñado para la creación de aplicaciones de escritorio y, con el auge de las aplicaciones para intranets e internet, el software de tipo distribuido.

Los lenguajes de programación casi siempre se encuentran desarrollados para trabajar en forma conjunta con Sistemas de Bases de Datos y Sistemas operativos con la finalidad de aprovechar al máximo las características con que están construidos dichos productos y se encuentran estrechamente enfocados a las características físicas del Hardware de las distintas generaciones de cómputo.

Los productos más utilizados se resumen en la siguiente tabla:

Sistema Operativo	Lenguaje de Programación	Sistema de Administración de Bases de Datos	Tipo de aplicaciones
Microsoft Windows 95, 98, 98 SE, ME, 2000, XP	Visual Studio 4, 5 ó 6, compuesto por Visual Basic, Visual C++, Visual J++, ASP	Microsoft Access, Microsoft MSDE, Microsoft SQL Server 7, Microsoft SQL Server 2000 y Visual FoxPro	Escritorio y distribuidas
Microsoft Windows XP	Visual Studio .NET, .NET 2003 y .NET 2005, compuesto por Visual Basic .NET, Visual C++ ,NET, Visual C# ,NET, ASPX, Ediciones .NET Express	Microsoft Access, Microsoft MSDE, Microsoft SQL Server 7, Microsoft SQL Server 2000, Microsoft SQL Server 2005, Microsoft SQL Server 2005 Express y Visual FoxPro	Escritorio y distribuidas
Microsoft Windows Vista	Visual Studio .NET 2008, compuesto por Visual Basic .NET 2008, Visual C++ .NET 2008, Visual C# .NET 2008	Microsoft Access, Microsoft MSDE, Microsoft SQL Server 7, Microsoft SQL Server 2000, Microsoft SQL Server 2005, Microsoft SQL Server 2005 Express, Microsoft SQL Server 2008 y Visual FoxPro	Escritorio y distribuidas
Microsoft XP, Microsoft Vista, Linux	PHP	MySql, PostGre SQL	Distribuidas y de escritorio reducidas
Microsoft Windows 2000, Microsoft Windows XP, Microsoft, Unix, Solaris	Java	Oracle	Distribuidas y de escritorio reducidas

Por las características y magnitud de este proyecto, se ha determinado que el tipo de aplicación debe ser de escritorio por lo cual no habrá manera de accesarse remotamente ya sea mediante una intranet o desde internet, por lo que la búsqueda de software se acota al primer tipo de software de desarrollo.

La siguiente consideración a tomar es el costo de estas herramientas y debido a que son altos, se ha optado por elegir software de costo nulo o de libre distribución. De los productos que se han mostrado en la tabla anterior, los lenguajes que cubren esta característica son: Visual Basic .NET 2005 Express, Visual C++ .NET 2005 Express, Visual C# .NET 2005 Express, PHP, y algunas ediciones de Java; mientras que las bases de dato de costo nulo o distribución libre son: Microsoft SQL Server 2005 Express, una edición de MySQL y PostGre SQL.

De este grupo de productos, se ha escogido al lenguaje Microsoft Visual Basic .NET 2005

Express y la Base de Datos Microsoft SQL Server .NET 2005 Express por tratarse de productos de libre distribución, por contarse con conocimientos en esta plataforma y por tratarse de productos de fácil entrenamiento y que en su entorno de desarrollo integran una característica denominada IntelliSense que provee al desarrollador una manera intuitiva de desarrollo en la medida en que se va elaborando las aplicaciones ofrece tips mediante "globos" sobre la estructura de la sintaxis de los componentes del lenguaje, ayuda global y particular de todas las opciones que ofrece la plataforma de desarrollo.

Además de lo anterior y por tratarse de herramientas que permiten elaborar aplicaciones en el entorno gráfico del Sistema Operativo Microsoft Windows XP y posteriores, ofrece mecanismos para acceder de manera fácil, rápida y eficiente a todos los componentes del Sistema Operativo.

Las versiones .NET de los productos de Microsoft se encuentran sustentadas y basadas en un nuevo estilo de generación de aplicaciones en las que se agrupan todas las características que en versiones anteriores de entornos de programación resultaban muy laboriosas, se requería de muchos conocimientos en relación a la manera en que los llamados objetos de modelo de complemento **COM** requerían.

La "nueva" plataforma se llama .NET FrameWork y en su estructura agrupa de manera estratégica a todas esas funciones y tareas que anteriormente eran arduas y difíciles de considerar y más aún de utilizar.

Cada versión del .NET FrameWork está ligada a una versión distinta de la plataforma de desarrollo; así pues, la versión 1.0 estuvo vinculada a Microsoft Visual Studio .NET, la versión 1.1 con Microsoft Visual Studio .NET 2003, la versión 2.0 con Microsoft Visual Studio .NET 2005 y las versiones 3.0 a 3.5 con la versión de Microsoft Visual Studio .NET 2008.

Las versiones 1.0, 1.1 y 2.0 se encuentran estrictamente desarrolladas para las versiones del Sistema Operativo Windows 2000 y XP aunque también son soportadas por todas las ediciones de Windows Vista y, las versiones 3.0 a 3.5 son exclusivas del Sistema Operativo Windows Vista y posteriores.

Por los criterios de selección aquí indicados, el programa de implementación de la solución propuesta deberá incluir en las herramientas de distribución, a la versión 2.0 del .NET FrameWork.

CAPÍTULO III: Conceptos generales

Este Capítulo, está enfocado a proveer los elementos generales que sirven de apoyo para la elaboración de la solución ofrecida en este trabajo. Por una parte se agrupan los que sirven de apoyo a la parte lógica del Sistema en conjunto en el subtema denominado Software, debiendo entender que no hace referencia en especial a toda la filosofía que este concepto considera; de manera similar, el subtema de Hardware es utilizado para aportar los conceptos que en particular describen a los medios de comunicación básicos con que cuentan los equipos de cómputo actual con el propósito de aportar elementos que permitirán la justificación del medio o puerto por el cual se realizará el enlace físico entre el equipo de cómputo y la *interfaz electrónica*; finalmente, en el tercer que se encuentra enfocado a los elementos electrónicos que se utilizan de *interfaz electrónica*, con la finalidad de dar apoyo a la decisión en la seleccionar de los componentes particulares que la componen.

Así pues, en conjunto todos estos conceptos sirven para preparar el entorno de desarrollo de la solución propuesta para la problemática que ya ha sido descrita en el Capítulo I.

III.1 Software

Aquí toca abordar de manera general los distintos elementos que dan origen a la elaboración propia de las soluciones manifestadas en las aplicaciones que serán utilizadas como base en los equipos de cómputo que aportan el elemento sustancial en la solución de la problemática que se guarda en los distintos sectores sociales de estos tiempos.

Las herramientas de software de desarrollo han evolucionado de manera tal que al inicio de la era de las computadoras, surgieron los medios por los cuales el ser humano tuviese que indicarle mediante expresiones simbólicas estructuradas de manera tal que los elementos electrónicos que componen a un equipo de cómputo pudiesen “entender” lo que de ellos se esperaba utilizando el medio por el cual se describe lo que se quiere obtener; este medio lo conforman los lenguajes de programación, mismas que han ido evolucionando en la medida en que las generaciones de cómputo han cambiado, mejorando y haciéndose cada vez más complejas para realizar tareas y operaciones que se han diversificado en todos los ámbitos de la vida cotidiana del ser humano de estos días. De esto se puede decir que resulta iluso pensar que no hay nada que no salga del alcance del ámbito de acción de estas máquinas “inteligentes”; desde el punto de vista que observemos el entorno actual, veremos que no existe nada que salga del alcance de la aplicación que se ha dado a estos equipos.

En nuestro hogar, muchos poseemos un equipo de cómputo tal y como se conoce con equipos complementarios llamados periféricos como una impresora, un teclado, un mouse, un monitor, etc.; pero los componentes de cómputo también los podemos localizar dentro de nuestro televisor, lavadoras, refrigeradores, reproductores de video en disco (DVD), equipos de sonido y la lista es interminable pues también los podemos encontrar fuera de nuestro entorno de vida cotidiana.

Fuera de casa los podemos encontrar en bancos al solicitar servicios financieros en cajeros automáticos, Sistemas de transporte, comunicaciones telefónica y la lista es interminable pues la tecnología se ha convertido en parte importante de nuestra vida tal y como la concebimos.

Toda la tecnología que ha sido mencionada y la que no, también se encuentra diseñada de manera tal que realice tareas específicas y para ello se requiere de los medios necesarios

para que puedan cumplir con el objetivo y misión que les ha sido enfocado. Todo esto es posible mediante el uso de los lenguajes de programación que en seguida son descritos de manera sucinta para aportar los elementos básicos y así comenzar a expresar el conjunto de herramientas que dan solución a la problemática que aquí han sido identificados.

III.1.1 Lenguajes de programación

Ya se ha expresado en párrafos anteriores que los lenguajes de programación son los medios que se utilizan para indicar a las computadoras de todas las magnitudes, tamaños, formas y marcas, las tareas que se quiere que desempeñen con la finalidad de obtener en forma óptima, rápida y precisa, los resultados deseados y por supuesto, esperados.

Estos lenguajes han sido ideados con la finalidad de establecer un medio de comunicación entre el ser humano y los componentes electrónicos de los equipos de cómputo, utilizando los elementos que conforman a un lenguaje de programación que por naturaleza propia ha sido dotado de reglas y estructuras similares a la sintaxis y semántica de los idiomas de los seres humanos para poder estructurar mediante un “programa” o aplicación a una solución particular o global.

Los lenguajes de programación han evolucionado demasiado y cada vez más se han robustecido con características y complementos que permiten al desarrollador o programador ser mas productivos en el sentido de que no se tendrá la necesidad de volver a escribir código que pueda resultar redundante y en lugar de ello, utilizar agrupaciones de funciones, código y aplicaciones que las distintas versiones de los lenguajes van aglutinando en la medida que han evolucionado.

Algunos lenguajes se enfocan para resolver problemas de un tipo de soluciones específicas y otros son de propósito genera. Ejemplo de esto, han sido los lenguajes como el Fortran, que se utilizó para aplicaciones de tipo científico (matemático); el lenguaje ensamblador que se compone por un conjunto de mnemónicos (abreviaciones) de palabras del idioma inglés que se encuentran más cercanos a la manera en que los CPU están diseñados y cómo funcionan a nivel electrónico.

Han surgido distintos lenguajes de programación a lo largo de la existencia de los equipos de cómputo y dentro de estos se han encontrado dos grandes grupos de generaciones clasificados por la manera en que se han ido comportando en la manera en que han sido diseñados. Estos lenguajes son los estructurados, procedimentales y los lenguajes orientados a objetos, y con el auge de la globalización de los caminos de la información en internet, los lenguajes de desarrollo distribuido.

Los lenguajes estructurados fueron el primer salto que existió entre los lenguajes de bajo nivel (en particular el ensamblador) hacia los lenguajes que ya consideraban a estructuras particulares de los lenguajes en que se pueden realizar tareas de cálculo de manera eficiente; operaciones repetitivas y estructuras de control con características especiales.

Los lenguajes procedimentales, incorporaron una manera complementaria de ofrecer mecanismos de solución orientados a no repetir código innecesario al incluir los conceptos análogos a funciones y relaciones matemáticas (funciones y procedimientos) y hasta este tipo de lenguajes, las aplicaciones se ejecutaban de manera secuencial tomando el control en el flujo de ejecución del programa hasta que la aplicación diseñada bajo esta característica no terminase de realizar la tarea que fue “programada”.

En los lenguajes orientados a objetos, se incluye el gran salto en la evolución de las herramientas de desarrollo, debido a que en sus estructuras se incluyen mecanismos a través de los cuales, el desarrollador tiene la posibilidad de realizar un análisis más profundo de la situación que quiere resolver para conducirlo hacia una forma más abstracta de la forma en que se encuentran estructurados los elementos (objetos) de la vida real y poderlos así representar mediante elementos del un lenguaje dentro de una aplicación de solución a una problemática previamente analizada para posteriormente ser solucionada con este nivel de abstracción que ofrece este tipo de lenguajes de programación.

En el último grupo de lenguajes, los orientados al desarrollo de aplicaciones distribuidas en internet e intranets utilizando los protocolos TCP/IP, los lenguajes de programación se encuentran clasificado en dos grupos: los de páginas estáticas y los de páginas dinámicas. El primer tipo, corresponde al estándar HTTP (HyperText Protocol) que consiste en la utilización de etiquetas bien definidas y delimitadas por marcadores de posición que interpretan el contenido que encierran para mostrarlos a los usuarios mediante un explorador de internet (vg. Internet Explorer, Mozilla, Netscape, entre otros). En el caso de las aplicaciones dinámicas, los lenguajes de programación utilizan un mecanismo mediante el cual el navegador de internet puede enviar y recibir información en formato HTML a una aplicación de este tipo hospedada en un servidor de internet con la finalidad de ofrecer una manera de intercambio de información, basándose en las propias características del lenguaje y de la tecnología implementada en el protocolo TCP/IP.

Todos estos tipos de lenguajes de programación han ido evolucionando de manera paralela a la evolución de los Sistemas Operativos; para el caso de los lenguajes estructurados y procedimentales, surgieron y tuvieron auge en los entornos de consola de comandos de texto (como el caso de DOS, en sus distintas variantes y versiones, UNIX y en los orígenes de LINUX entre otros Sistemas Operativos). El salto de los lenguajes estructurados y procedimentales hacia los lenguajes orientados a objetos tuvieron su auge al surgir los Sistemas Operativos de ambiente gráfico como lo fue Microsoft Windows en sus versiones 3.0, 3.11, 95, 98, 98 SE, ME, 2000, XP, 2003 y ahora Vista. Por el lado de la competencia, LINUX, Mac OS, HP-UNIX y otros de dimensiones más grandes y para entornos de aplicaciones WAN o LAN de dimensiones importantes.

La ventaja que nos ofrecen los lenguajes orientados a objetos con respecto a los estructurados y procedimentales es que con ellos se pueden realizar aplicaciones de multiproceso y en tiempo real, además de que ofrecen el mecanismo de acceso directo a las características técnicas y operativas de los entornos gráficos.

III.1.2 Bases de Datos

Hablar de los lenguajes de programación no se puede hacer de manera aislada, pues en su origen y en conjunto con los equipos de cómputo, su tarea primordial fue la de concentrar información de manera masiva, cuyas aplicaciones principales fueron las de la elaboración de censos económicos y población, de índole científica (investigación) y de tipo militar.

De manera general, se puede definir a una Base de Datos como cualquier conjunto de datos organizados para su almacenamiento en la memoria (secundaria) de una computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar. La información se organiza en campos y registros. Un campo se refiere a un tipo o atributo de información, y un registro, a toda la información sobre un individuo. Por ejemplo, en una Base de Datos que almacene información de tipo agenda, un campo será el NOMBRE, otros los APELLIDOS, otro la

DIRECCIÓN..., mientras que un registro viene a ser como la ficha en la que se recogen todos los valores de los distintos campos para un individuo, esto es, su nombre, apellidos, dirección... Los datos pueden aparecer en forma de texto, números, gráficos, sonido o vídeo. Normalmente las Bases de Datos presentan la posibilidad de consultar datos, bien los de un registro o los de una serie de registros que cumplan una condición. También es frecuente que se puedan ordenar los datos o realizar operaciones sencillas, aunque para cálculos más elaborados haya que importar los datos en una hoja de cálculo u otro software para análisis y de cálculo estadístico, entre otras posibilidades. Para facilitar la introducción de los datos en la base se suelen utilizar formularios (elaborados en lenguajes de programación de alto nivel enfocados a objetos); también se pueden elaborar e imprimir informes sobre los datos almacenados.

Desde su aparición en la década de 1950, estas aplicaciones se han hecho imprescindibles para las sociedades en sus distintos ámbitos de acción y pensamiento.

Con el concepto de Base de Datos, apareció también el de Sistema de Administración de Base de Datos (SABD) y que para el caso del cómputo personal, el primer Sistema de administración de Bases de Datos tiene su origen en 1980; era el dBase II, desarrollado por el ingeniero estadounidense Wayne Ratliff. Desde entonces, su evolución ha seguido paralela a la que ha experimentado el *software* de desarrollo, como se ha indicado en el apartado anterior, y hoy existen desde Bases de Datos para una utilización personal hasta Bases de Datos corporativas, soportadas por grandes Sistemas informáticos.

Los Sistemas de administración de Bases de Datos también han evolucionado en su estructura y conceptos que han ido englobando, dicha evolución se ha resumido en cinco modelos de Base de Datos principales:

- Modelo jerárquico
- Red
- Relacional
- Base de Datos deductivas
- Orientadas a objetos o de objetos persistentes

El análisis de las características de cada uno de estos modelos queda fuera del alcance de este trabajo, pero si es importante concentrar la atención en el modelo relacional pues este es el que más se utiliza y ha evolucionado en todo el universo de productos orientados a este fin y la versión Microsoft SQL Server 2005 Express corresponde a esta categoría y como ya se dijo en el Capítulo II, es el que será empleado.

Una Base de Datos de Relacional se encuentra fundamentada en el modelo entidad-relación de la teoría de conjuntos y consta de una colección de tablas (llamadas entidades) en las que se almacena un conjunto específico de datos estructurados. Una tabla contiene una colección de filas, también denominadas tuplas o registros, y columnas, también denominadas atributos. Cada columna de la tabla se ha diseñado para almacenar un determinado tipo de información; por ejemplo, fechas, nombres, importes en moneda o números.

Las tablas contienen diversos tipos de controles, como restricciones, desencadenadores, valores predeterminados y tipos de datos de usuario personalizados, que garantizan la validez de los datos. Se pueden agregar restricciones de integridad referencial a las tablas con el fin de asegurar la coherencia de los datos interrelacionados que se encuentran en tablas distintas. Las tablas pueden presentar índices, similares a los de los libros, que permiten localizar las filas rápidamente. Una Base de Datos también puede incluir procedimientos que utilicen código de programación (que en SQL Server se denominan procedimientos almacenados)

para realizar operaciones con los datos de la Base de Datos. Entre estas operaciones se incluye la creación de vistas que proporcionen acceso personalizado a los datos de la tabla o una función definida por el usuario que realice cálculos complejos en un subconjunto de filas.

Entre las exigencias que se reclaman de las Bases de Datos, resalta la necesidad de tener una integridad de la información ya que esta garantiza la calidad de los datos de la Base de Datos. Por ejemplo, si se especifica para un empleado el valor de identificador de **123**, la Base de Datos no debe permitir que ningún otro empleado tenga el mismo valor de identificador. Si tiene una columna **CategoríaDeEmpleado** para la que se prevean valores entre **1** y **5**, la Base de Datos no debe aceptar valores fuera de ese intervalo. Si en la tabla hay una columna **NúmeroDeDepartamento** en la que se almacena el número de departamento del empleado, la Base de Datos sólo debe permitir valores que correspondan a los números de departamento de la empresa.

Dos pasos importantes en el diseño de las tablas son la identificación de valores válidos para una columna y la determinación de cómo forzar la integridad de los datos en la columna. La integridad de datos pertenece a una de las siguientes categorías:

- **Integridad de entidad**. La integridad de entidad define una fila como entidad única para una tabla determinada. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones únicas o restricciones de llave primaria.

- **Integridad de dominio.** La integridad de dominio viene dada por la validez de las entradas para una columna determinada. Puede exigir la integridad de dominio para restringir el tipo mediante tipos de datos, el formato mediante reglas y restricciones que fuerzan el chequeo de datos, o el intervalo de valores posibles mediante restricciones de llaves foreñas, restricciones de chequeo de tipo, definiciones de valore por omisión, definiciones en las que no se aceptan valores nulos y reglas.

- **Integridad referencial**. en las relaciones entre claves externas y claves principales o entre claves externas y claves exclusivas, mediante restricciones de llaves foreñas y de chequeo de condiciones. La integridad referencial garantiza que los valores de clave sean coherentes en las distintas tablas. Para conseguir esa coherencia, es preciso que no haya referencias a valores inexistentes y que, si cambia el valor de una clave, todas las referencias a ella se cambien en consecuencia en toda la Base de Datos.

  Cuando se exija la integridad referencial, se impedirá:

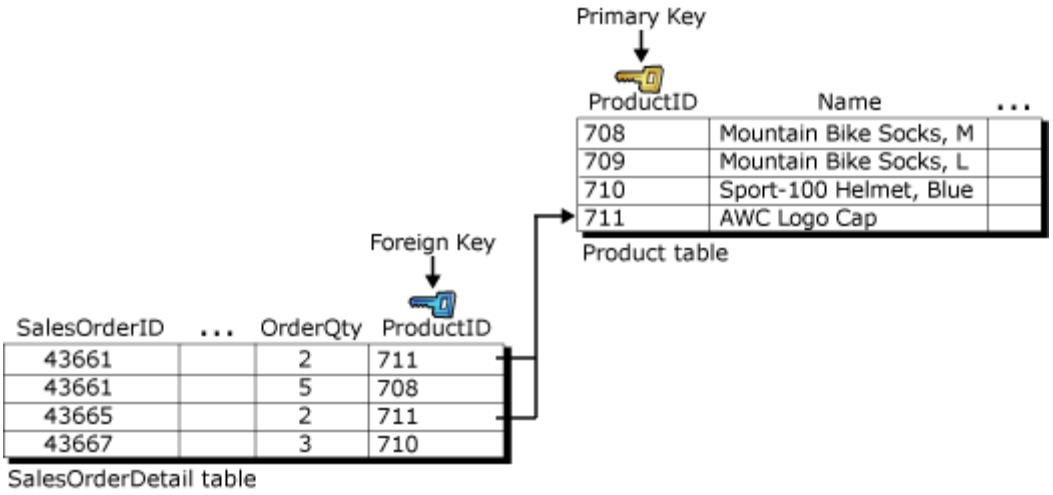
  - Agregar o cambiar filas en una tabla relacionada si no hay ninguna fila asociada en la tabla principal.

  - Cambiar valores en una tabla principal que crea filas huérfanas en una tabla relacionada.

  - Eliminar filas de una tabla principal cuando hay filas relacionadas coincidentes.

  Por ejemplo, en las tablas **Sales.SalesOrderDetail** y **Production.Product** de la Base de Datos **AdventureWorks**, la integridad referencial se basa en la relación entre la clave externa (**ProductID**) de la tabla **Sales.SalesOrderDetail** y la clave principal (**ProductID**) de la tabla **Production.Product**. Esta relación garantiza que un pedido de ventas no

pueda nunca hacer referencia a un producto que no existe en la tabla **Production.Product**.



- **Integridad definida por el usuario.** La integridad definida por el usuario permite definir reglas de empresa específicas que no pertenecen a ninguna otra categoría de integridad. Todas las categorías de integridad admiten la integridad definida por el usuario. Esto incluye todas las restricciones de nivel de columna y nivel de tabla que se definen en la creación de las tablas, procedimientos almacenados y desencadenadores.

Así, las llaves resultan ser un elemento importante mediante el cual se logra que la información estructurada en el modelo entidad-relación (relacional) garantice la unicidad e independencia física y lo más importante, la seguridad en la estructura de su entorno.

Los SABD de hoy en día se encuentran montados en motores de Bases de Datos que emplean un lenguaje interprete de scripts (pequeños programas) en un lenguaje denominado **SQL** (Structured Query Language) o también conocido como "Lenguaje Estructurado de Consultas", se utiliza para consultar, actualizar y manejar información de Bases de Datos relacionales. Se deriva de un proyecto de investigación de IBM, que creó el 'lenguaje estructurado de consulta en inglés' (SEQUEL) a mediados de la década de 1970. El SQL es un estándar aceptado en productos de Bases de Datos, que fue utilizado comercialmente por primera vez por la empresa Oracle en 1979. A pesar de que no se trata de un lenguaje de programación como puedan serlo C o Pascal, puede utilizarse en el diseño de consultas interactivas y puede incluirse en una aplicación como un conjunto de instrucciones de manejo de datos (sentencias). Por ejemplo, basándose en soportes externos al propio lenguaje y aún al propio programa ya compilado (como es un servidor o un motor SQL que haga de puente entre la aplicación, el Sistema Operativo y las Bases de Datos), lenguajes como Visual Basic o C/C++ pueden incluir sentencias SQL en su código.

Aunque en el mercado se pueden encontrar diversas variantes (dialectos) del lenguaje SQL, todos tienen un mínimo común basado en el estándar ANSI SQL de 1986, con extensiones propietarias. Desde 1991 se fueron realizando progresos para disponer de un estándar cada vez más completo, conocido como SAG SQL, propiciado por el denominado *SQL Access Group*.

El SQL estándar cuenta con elementos destinados a la *definición, modificación, control, protección de los datos* y *acceso a Bases de Datos distribuidas en una red*, haciendo factible que varios usuarios puedan consultar la misma Base de Datos de forma simultánea. Debido a su simplicidad, que proviene de ser sintáctica y gramaticalmente cercano al lenguaje natural, pueden utilizarlo tanto los usuarios técnicos como los que no lo son; esto, unido a su

potencia, lo ha hecho muy popular.

Los elementos que conforman al estándar SQL se encuentran clasificados en tres grandes grupos a saber:

- **Lenguaje de definición de datos (DDL).** Este es un subconjunto del lenguaje que se utiliza para crear y administrar Bases de Datos y objetos de Base de Datos, tales como tablas, procedimientos almacenados, funciones definidas por el usuario, desencadenadores, vistas, defaults, índices, reglas y estadísticas.

- **Lenguaje de manipulación de datos (DML).** Este es un subconjunto del lenguaje que es más utilizado. Básicamente se utiliza para recuperar, insertar, modificar y eliminar información de la Base de Datos. Estas cuatro operaciones se realizan a través de comandos que componen a este sub lenguaje de manera respectiva con: SELECT, INSERT, UPDATE y DELETE.

- **Lenguaje de control de datos (DCL).** Este es un subconjunto del lenguaje que se utiliza para administrar la seguridad en la Base de Datos. Especialmente, se utiliza para establecer permisos sobre los objetos de la Base de Datos y las sentencias. En general, después de que se crea una Base de Datos y los objetos de la Base de Datos a través del **DDL**, se estará preparado para establecer permisos utilizando las sentencias provistas por el **DCL**, las sentencias que componen a este subconjunto del lenguaje son GRANT, DENY y REVOKE.

III.1.3 Modelaje de las Bases de Datos

El diseño de una Base de Datos requiere un conocimiento profundo de las características físicas del modelo que se desea modelar, así como de las características y conceptos de Base de Datos que se desean utilizar para representar esas características. Hay que asegurarse de diseñar con precisión la Base de Datos que se utilizará para modelar el entorno porque el cambio del diseño de la Base de Datos una vez implementada puede requerir mucho tiempo. Por otra parte, una Base de Datos bien diseñada ofrece un mejor rendimiento.

El primer paso al crear una Base de Datos consiste en elaborar un plan que sirva de guía para la implementación de la Base de Datos y, al mismo tiempo, como especificación funcional después de su implementación. La complejidad y los detalles de diseño de una Base de Datos dependen de la complejidad y el tamaño de la aplicación de Bases de Datos, así como de los usuarios.

La naturaleza y complejidad de una aplicación de Bases de Datos, así como el proceso de diseño, pueden variar significativamente. Una Base de Datos puede ser relativamente sencilla y estar diseñada para que la utilice una sola persona, o bien ser grande y compleja, y estar diseñada para procesar, por ejemplo, las transacciones bancarias de miles de clientes. En el primer caso, el diseño de la Base de Datos puede consistir en poco más que unas anotaciones en un papel. En el segundo caso, el diseño puede ser un documento formal de cientos de páginas que contenga todos y cada uno de los posibles detalles relacionados con la Base de Datos.

Al diseñar la Base de Datos, independientemente de su tamaño y complejidad, se debe llevar a cabo los siguientes pasos básicos:

- **Recopilar la información**. Antes de crear una Base de Datos, debe conocer con detalle el cometido previsto de la Base de Datos. Si la Base de Datos va a reemplazar a un Sistema de información manual o en papel, dicho Sistema proporcionará la mayor parte de la información que se necesita. Se debería consultar a todas las personas involucradas en el Sistema para determinar lo que hacen y qué necesitan de la Base de Datos. También es importante identificar qué es lo que desean que haga el nuevo Sistema, así como los problemas, las limitaciones y los cuellos de botella del Sistema existente. Recopilar copias de las instrucciones del cliente, listas de inventarios, informes de administración y, en general, de todos aquellos documentos que formen parte del Sistema existente, porque servirán para diseñar la Base de Datos y las interfaces.

- **Identificar los objetos.** Durante el proceso de recopilación de información, se deberá identificar los objetos o las entidades más importantes que vayan a ser administrados por la Base de Datos. El objeto puede ser tangible (una persona o un producto, por ejemplo) o bien intangible (como una transacción empresarial, un departamento de una compañía o un período de liquidación de nóminas). Normalmente hay unos cuantos objetos principales; una vez identificados, se evidencian los elementos relacionados. Cada elemento diferenciado de la Base de Datos debe tener su tabla correspondiente.

- **Crear modelos de los objetos**. Cuando se identifican los objetos del Sistema, se tendrán que registrar de una manera que represente visualmente el Sistema. Puede utilizar su modelo de Base de Datos como referencia durante la implementación de la misma.

  Con este propósito, los programadores de Bases de Datos utilizan herramientas que varían en complejidad técnica: desde papel y bolígrafo hasta procesadores de texto y hojas de cálculo e, incluso, programas de software creados específicamente para la elaboración de modelos de datos destinados al diseño de Bases de Datos. Sea cual sea la herramienta que se decida utilizar, es importante que se mantenga actualizada.

- **Identificar los tipos de información para cada objeto**. Una vez identificados los objetos principales de la Base de Datos como candidatos para las tablas, el siguiente paso es identificar los tipos de información que deben almacenarse para cada objeto. Estos tipos son las columnas de la tabla del objeto. Las columnas de una tabla de Base de Datos contienen algunos tipos de información comunes:

  - *Columnas de datos sin procesar*. Estas columnas almacenan información tangible, como por ejemplo nombres, determinada por un origen externo a la Base de Datos.

  - *Columnas de categorías*. Estas columnas clasifican o agrupan los datos y almacenan una selección limitada de datos, tales como verdadero o falso; casado o soltero; presidente, director o responsable de equipo; etc.

  - *Columnas de identificadores*. Estas columnas proporcionan un mecanismo para identificar cada elemento almacenado en la tabla. Estas columnas suelen incluir un Id. o un número en el nombre (por ejemplo, **IdDeEmpleado**, **númeroDeFactura** y **IdDeEditor**. La columna del identificador es el componente principal para los usuarios y las funciones internas de proceso de la Base de Datos para el acceso a una fila de datos de la tabla. Algunas veces el objeto tiene una forma tangible de Id. utilizada en la tabla (por ejemplo, un número de seguro social), aunque en la mayoría de los casos se puede definir la tabla para poder crear un Id. confiable y artificial para la fila.

  - *Columnas relacionales o diferenciales*. Estas columnas establecen un vínculo entre la información de una tabla y la información relacionada que se encuentra en otra

tabla. Por ejemplo, una tabla que realiza el seguimiento de transacciones comerciales puede tener un vínculo con una tabla **clientes**, de modo que pueda asociarse toda la información del cliente a la transacción comercial.

- **Identificar las relaciones entre los objetos**. Uno de los puntos fuertes de una Base de Datos relacional es la posibilidad de relacionar o asociar información acerca de varios elementos de la Base de Datos. Pueden almacenarse por separado tipos aislados de información, pero el motor de Base de Datos puede combinar los datos cuando sea necesario. Para identificar la relación entre los objetos durante el proceso de diseño se necesita llevar a cabo un examen de las tablas con el fin de determinar sus relaciones lógicas y agregar columnas relacionales que establezcan un vínculo entre las tablas.

Una vez que se han identificado las características de todos los componentes de la Base de Datos, conviene ahora establecer un proceso de optimización de información con la finalidad de evitar utilizar campos que se repitan en distintas tablas con la finalidad de establecer alguna tabla que contenga esos datos y establecerlos como fijos y entonces así se podrá utilizar códigos en lugar de textos directos.

Todo este proceso se logra mediante un proceso denominado Normalización que se describe enseguida: El diseño lógico de la Base de Datos, que incluye las tablas y sus relaciones, es la clave de una Base de Datos relacional optimizada. Un buen diseño lógico de la Base de Datos puede ser la base de un rendimiento óptimo de la aplicación y de la Base de Datos. Un diseño lógico deficiente puede comprometer el rendimiento de todo el Sistema.

La normalización de un diseño lógico de la Base de Datos implica la utilización de métodos formales para separar los datos en varias tablas relacionadas. Una característica de una Base de Datos normalizada es la existencia de varias tablas pequeñas con menos columnas. En las Bases de Datos no normalizadas, existen menos tablas más amplias con más columnas.

Por lo general, una normalización razonable permite mejorar el rendimiento. Cuando se dispone de índices útiles, la normalización ofrece diversas ventajas, entre las que se incluyen:

- Mayor rapidez en la ordenación y en la creación de índices.
- Un número mayor de índices agrupados.
- Índices más estrechos y compactos.
- Menor número de índices por tabla. De esta forma, se mejora el rendimiento de las instrucciones INSERT, UPDATE y DELETE.
- Menor número de valores NULL y menos oportunidades para generar incoherencias. De esta forma, aumenta el rendimiento.

A medida que progresa la normalización, también aumenta el número y la complejidad de las combinaciones necesarias para recuperar los datos. Un número muy elevado de combinaciones relacionales complejas entre demasiadas tablas puede afectar al rendimiento. Una normalización razonable suele incluir un número reducido de consultas que se ejecutan con regularidad y utilizan combinaciones en las que intervienen más de cuatro tablas.

A veces, el diseño lógico de la Base de Datos es fijo y su remodelación resulta inviable. No obstante, incluso en estos casos puede normalizarse de forma selectiva una tabla de gran tamaño para crear tablas más pequeñas. Si el acceso a la Base de Datos se realiza mediante

procedimientos almacenados, este cambio del esquema puede llevarse a cabo sin que afecte a las aplicaciones. De lo contrario, puede crearse una vista que oculte el cambio de esquema a las aplicaciones.

Una vez que se han reunido todos los elementos, estructuras, tipos de datos definirán al comportamiento de los atributos de las entidades, los índices y demás elementos que componen al Sistema en general; además de aplicársele el proceso de normalización. El desarrollador se encuentra en condiciones de empezar a escribir los scripts de SQL que permita crear el esquema de la estructura ya definida.

Para lograr esto, dentro de los entornos de los SABD más comunes en estos días, se debe proceder a escribir el código de definición de datos mediante las sentencias o comandos CREATE DATABASE, CREATE TABLE, CREATE VIEW, CREATE INDEX, CREATE TRIGGER y, con la depuración de la estructura de la Base de Datos en conjunto se puede hacer uso de las sentencias ALTER DATABASE, ALTER TABLE, ALTER VIEW, ALTER INDEX y ALTER TRIGGER.

La manera en que operan los comandos aquí indicados queda fuera del alcance de este trabajo, pero a manera de referencia puede consultar los libros en pantalla de Microsoft SQL Server, direcciones ***ms-help://MS.SQLCC.v9/MS.SQLSVR.v9.es/tsqlref9/html/1e068443-b9ea-486a-804f-ce7b6e048e8b.htm*** y ***ms-help://MS.SQLCC.v9/MS.SQLSVR.v9.es/tsqlref9/html/29ddac46-7a0f-4151-bd94-75c1908c89f8.htm***.

III.2 Hardware

En el apartado anterior, se enfatizó un poco a encaminar la atención a los elementos básicos de Software que implican a los lenguajes y Bases de Datos que se utilizan con mayor frecuencia por tratarse de estándares que se han observando desde hace ya algún tiempo; ahora, toca turno a encaminar la atención hacia los elementos físicos con los cuales se encuentran provistos los equipos de cómputo de estos tiempos para poder comunicarse el entorno interno de dichos equipos con entornos externos que corresponden con los equipos complementarios o periféricos que el mundo del cómputo personal, servidores, mainframes y equipos de gran escala utilizan para que éstos últimos puedan reconocer a esos equipos complementarios.

En este apartado se hace referencia a los tres tipos de puertos que se han estandarizado para dicho propósito, aunque la tendencia es a que estos tipos de puertos se encuentren conducidos a que sólo uno de ellos (USB), sea el que predomine.

Durante la elaboración de este Sistema, se hizo una investigación respecto de las configuraciones que se están estandarizado en los equipos nuevos y, se pudo observar que ya muy pocos equipos ofrecen como soporte de escalabilidad de puertos, ya que sólo incluyen puertos USB y algunos más incluyen puertos IEEE 1394a.

Esta tendencia no representa mayor problema, pues en el mercado ya se encuentran dispositivos de conversión de puertos USB a puertos seriales y de USB a puertos paralelos, aunque heredan consigo los problemas aparentes que el estándar USB trae consigo.

Así pues, aquí se muestra la estructura y características de estos tres tipos de puertos que servirán como antecedente para poder justificar el puerto seleccionado para la comunicación entre el Sistema y la *interfaz electrónica*.

III.2.1 Puerto serial

La forma más común y sencilla de comunicar cualquier dispositivo con una computadora, es a través de un puerto serie, que es compatible con el denominado estándar RS-232 (o EIA232 Standard). En una computadora puede haber varios puertos serie dependiendo de su configuración y de la generación de tecnología a la que pertenece, normalmente los puertos se denominan con COM1, COM2, etc.

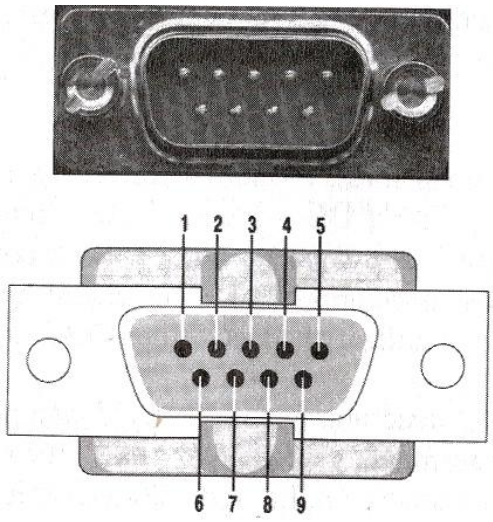
Los puertos serie son accesibles mediante conectores de la norma RS232 establece en dos tipos de conectores llamados DB-25 (de 25 pines) y DB-9 (de 9 pines), machos y hembras. La norma RS232 se estableció para conectar una computadora con un modem, por lo que aparecen muchas terminales en los conectores DB-25 que en otro tipo de aplicaciones no se utilizan y en las que es más común utilizar el conector tipo DB-9.

Cada una de las terminales del conector RS232 tiene una función especificada por la norma. Hay unas terminales por las que se transmiten y reciben datos y otros que controlan el establecimiento, flujo y cierre de la comunicación. En el Esquema III-1, se muestra la distribución de las terminales del conector DB-9.

Para comunicarse con algún dispositivo externo, en la mayoría de los casos son suficientes tres líneas del DB-9:

- Línea de transmisión (TxD), pin 3 (Transmitted Data)
- Línea de recepción (RxD), pin 2 (Received Data)
- Terminal de tierra, pin 5 (Signal Ground)

Las especificaciones del puerto serie están contenidas en la norma RS232 (o EIA232 Standard).



PIN	SEÑAL
1	Data Carrier Detect (DCD)
2	Received Data (RxD)
3	Transmitted Data (TxD)
4	Data Terminal Ready (DTR)
5	Signal Ground (SG)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Ring Indicator (RI)

Esquema 1. Distribución de las terminales del puerto serial

Un dato importante a tener en cuenta en cualquier comunicación es la velocidad de transmisión, que es la cantidad de información enviada por la línea de transmisión en la unidad de tiempo. Hay distintas unidades para expresar esta medida, la más utilizada es el

Baudio, que es proporcional a los bits por segundo (bps), definidos como el número de bits de información enviados por segundo.

La velocidad a la que pueden trabajar los puertos COM (serie) de una computadora está normalizada a 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800 y 921600 baudios. Los valores de 75 a 9600 baudios son demasiado pequeños para los estándares de la actualidad, pero suficientemente rápidos para multitud de aplicaciones.

III.2.1.1 Niveles lógicos RS232

En la Ilustración III-1, se muestran los requisitos en cuanto a niveles lógicos que se debe cumplir una transmisión serie según la norma RS232. A destacar:

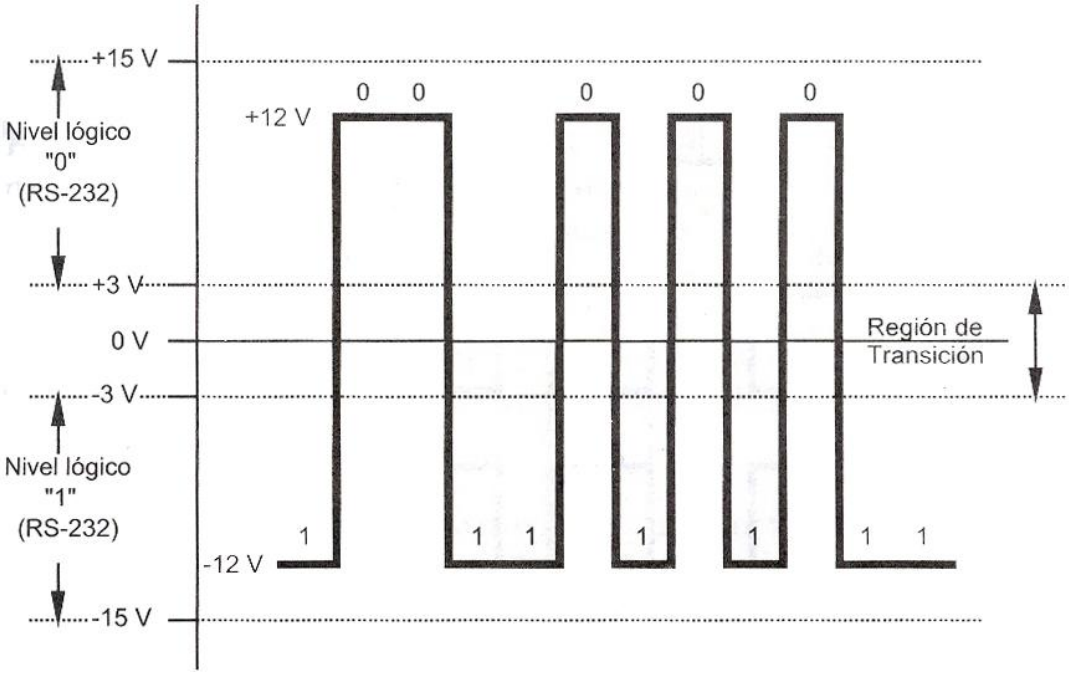


Ilustración III-1. Niveles lógicos para el RS232

- Los datos se transmiten con lógica negativa, es decir, un voltaje positivo en la conexión representa un “0”, mientras que un voltaje negativo representa un “1”.
- Para garantizar un “0” lógico, una línea debe mantener un voltaje entre +3 y +15 volts.
- Del mismo modo, para “1” lógico garantizado debe estar entre -3 y -15 volts.
- Los voltajes más usados son +12 volts para el “0” y -12 volts para el “1”.
- Es importante resaltar que cuando un puerto serie no está transmitiendo, mantiene la terminal de transmisión a “1” lógico a -12 volts, normalmente.
- La banda muerta entre +3 volts y -3 volts se conoce como la región de transmisión donde los niveles lógicos no están definidos. Esto significa que cualquier valor en esta banda puede interpretarse ambiguamente como “0” ó “1”.

Si se aumenta la velocidad de transmisión, las señales de datos se vuelven susceptibles de pérdidas de voltaje causadas por la capacitancia, resistencia e inductancia del cable. Estas pérdidas son conocidas como efectos e alta frecuencia y aumentan con la longitud del cable.

Estos valores de voltaje proporcionan un amplio margen de seguridad que es de gran utilidad cuando los cables deben pasar por zonas cercanas a elementos que generan interferencias eléctricas: motores, transformadores, equipos de comunicación, etc. Estos elementos, unidos a la longitud de cable, pueden hacer disminuir la señal hasta en varios volts, sin que afecte adversamente al nivel lógico de entrada.

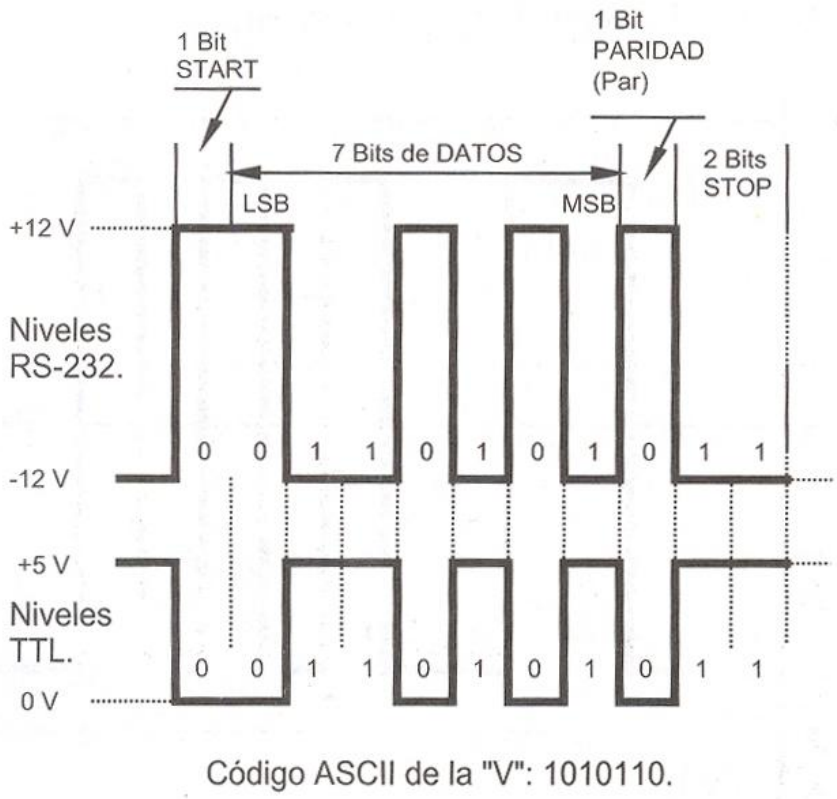


Ilustración III-2. Ejemplo de envío de un byte según la norma RS232

III.2.1.2 Formato de un byte

La comunicación de datos en un puerto serie RS232 se usa normalmente para efectuar comunicaciones asíncronas sin tiempo preestablecido para iniciarse. Los datos llegan en paquetes de información normalmente de 8 bits. Algunos equipos envían carácter por carácter, otros guardan muchos caracteres en la memoria y cuando les toca enviarlos los envían uno tras otro.

El protocolo establecido por la norma RS232 envía la información estructurada en 4 partes, mostradas en la Ilustración III-2 con un ejemplo:

- **Bit de inicio o arranque.** Es un paso de -12 volts a +12 volts, es decir, de un "1" a un "0" lógico en la lógica negativa de la norma RS232. Cuando el receptor detecta el bit de inicio sabe que la transmisión ha comenzado y, a partir de entonces, debe leer las señales de la línea a distancias concretas de tiempo en función de la velocidad fijada por el emisor y el receptor.

- **Bits de datos.** Los bits de datos son enviados al receptor después del bit de inicio. El bit menos significativo LSB (Least Significant Bit) es transmitido primero y el más significativo MSB (Most Significant Bit), el último. Un carácter de datos suele consistir en 7 u 8 bits. En el ejemplo de la Ilustración III-2., se trabaja con 7 bits.

- **Bit de paridad**. Dependiendo de la configuración de la transmisión, un bit de paridad puede ser enviado después de los bits de datos. En aplicaciones sencillas no suele utilizarse. Con este bit se pueden descubrir errores en la transmisión. Se puede dar paridad par o impar. En la paridad par, por ejemplo, la palabra de datos a transmitir se completa con el bit de paridad de manera que el número de bits "1" enviados sea par, como en el ejemplo de la Ilustración III-2, donde la información del byte de datos más paridad es "01010110" (cuatro "1").

- **Bit de parada**. **L**a línea queda a -12 volts después del último bit enviado, es decir, queda a "1" en la lógica negativa de la norma RS232. Indica la finalización de la transmisión de una palabra de datos. El protocolo de transmisión de datos permite 1, 1.5 ó 2 bits de parada.

En el ejemplo de la figura anterior se envía un bit de inicio, una palabra de 7 bits (101 0110), que corresponde a la letra "V" en código ASCII, un bit de paridad par y luego dos bits de parada.

III.2.2 Puerto paralelo

El Puerto paralelo es una herramienta simple y barata para la construcción de distintos periféricos y de proyectos controlados mediante éste dispositivo. La simplicidad y facilidad de la programación lo ha hecho popular en el mundo de la jovialidad y de la industria del cómputo personal sobre todo. Por excelencia, este tipo de puertos ha sido utilizado para la conexión entre equipos de cómputo personal e impresoras, scanners y una gran variedad de dispositivos aunque en la actualidad se ha visto desplazado por la incorporación de otros mecanismos de comunicación entre computadoras y periféricos, aunque por compatibilidad, se sigue manteniendo en la actualidad.

La sencillez de la programación de estos puertos obedece fundamentalmente a la manera en que son administrados los dispositivos de comunicación a niveles físicos y electrónicos, aunque a partir de los Sistemas operativos que tienen su origen en la plataforma de Windows NT, es decir, Windows 2000, Windows XP y Windows Vista, se considera el concepto de "recursos administrados", lo que implica, que el acceso a este tipo de puertos no puede hacerse de manera directa y para poderlo hacer, se requiere de una versión de software de desarrollo que considere las características de acceso a dichos recursos de esta categoría.

Por esta situación ya se distribuye en internet un conjunto de bibliotecas dinámicas (DLL) que están compuestas por todas las funciones necesarias para realizar las funciones de comunicación directa entre el equipo de cómputo y los puertos sin tener que preocuparse por el mecanismo de administración de recursos.

Como ya se ha indicado, el uso principal de los puertos paralelos es el de conectar impresoras a la computadora y se encuentra especialmente diseñado para este propósito, es por ello que a menudo se le conoce como puerto de impresoras o puerto Centronics (este nombre deriva de una compañía popular fabricante de impresoras llamada "Centronics", quien desarrolló algunos estándares para este tipo de puertos). El puerto se puede observar en la parte posterior del equipo de cómputo, tiene la forma de un conector hembra de 25 pines llamado DB25.

Como ya se ha dicho también, todos los equipos de cómputo cuentan con al menos un puerto de este tipo y en los casos en que ya no exista alguno de éstos, existen tarjetas de expansión

en las que se implementan uno o dos puertos conectados al equipo de computo mediante una tarjeta de tipo PCI o PCI Express.

Los puertos paralelos tienen distintos tipos de operación física, mismos que han sido definidos en el estándar IEEE 1284, y han sido publicados en 1994. Estos modos de transferencia de datos mediante un puerto paralelo son:

- Modo de compatibilidad
- Modo Nibble
- Modo byte
- EPP
- ECP

La relativa facilidad de programación de estos puertos, así como su rapidez comparada con los puertos seriales también muestra desventajas: se requiere de una línea (cable) para cada uno de los bits que representan al byte se que se está transmitiendo y de manera adicional se requieren de varias líneas más para implementar el proceso de control y estatus de la operación del puerto. Los puertos paralelos implementan directamente los niveles lógicos TTL, motivo por el cual no se pueden utilizar en distancias relativamente grandes.

La conexión será en el puerto de la computadora mediante un conector DB-25 hembra y en el otro extremo, un conector Centronics de 36 pines, la correspondencia entre los cables de los conectores se resume en la Ilustración III-3.

Las líneas del conector DB-25 se encuentran divididas en 3 grupos, estas son:

- Líneas de datos (bus de datos)

- Líneas de control

- Líneas de estatus

Como su nombre lo indica, los datos se transfieren sobre las líneas de datos, las líneas de control se utilizan para controlar al periférico y por supuesto, el periférico retorna las señales de control a la computadora a través de las líneas de estatus. Estas líneas se encuentran conectadas a los registros internos de la computadora, estos registros se conocen como puertos que físicamente son localidades de memoria que sirven para realizar el enlace físico con las líneas del puerto mismo y sus periféricos conectados. Las direcciones de los puertos de memoria en donde se encuentran las señales asociadas a los puertos paralelos se resumen en la tabla que se muestra después del esquema de distribución de terminales para este tipo de conectores.

No. de pin DB-25	No. de pin Centronics 36	Función
1	1	Strobe (habilitación)
2 a 9	2 a 9	Líneas de datos
10	10	Reconocimiento
11	11	Ocupado
12	12	Sin papel
13	13	Selección
14	14	Auto alimentación
15	15, 32	Error
16	16, 31	Reinicio
17	17, 36	Selección de entrada
18 a 25	18 a 30, 33	Tierra
-	34, 35	No utilizados

Tabla III-1. Correspondencia de señales entre conectores DB-25 y Centronics

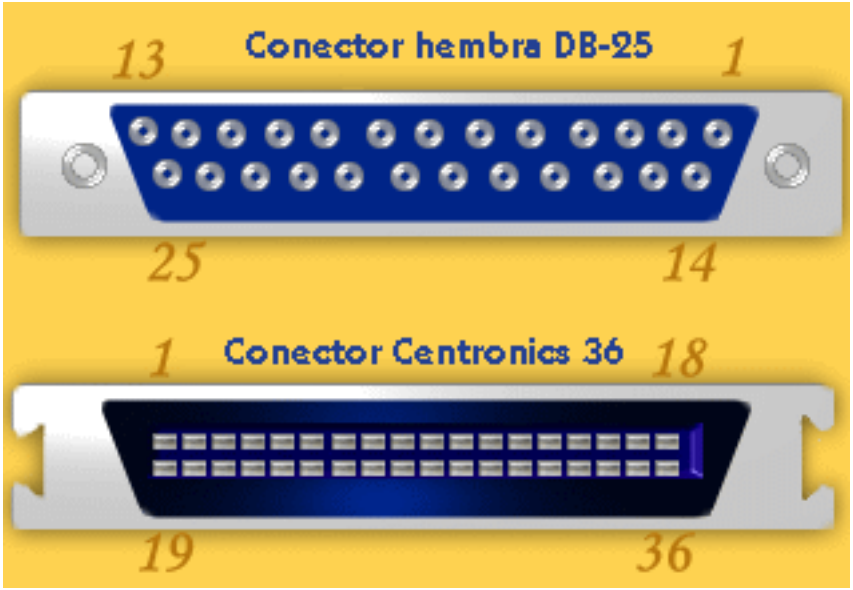


Ilustración III-3. Conectores y sus funciones

Registro	LPT1	LPT2
Registro de datos	0x378	0x278
Registro de estatus	0x379	0x279
Registro de control	0x37a	0x27a

Mientras que los detalles de las señales del puerto paralelo se muestran en la Ilustración III-4.

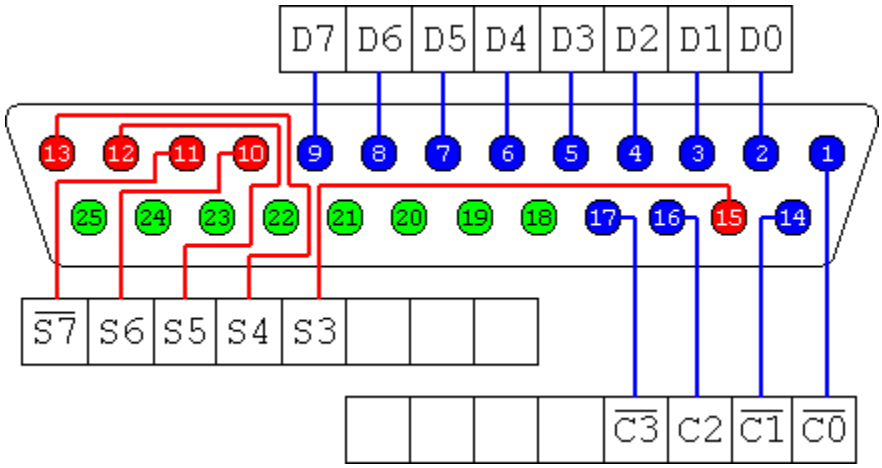


Ilustración III-4. Distribución de las señales del puerto paralelo

Finalmente, la distribución de las señales así como el tipo correspondencia con las señales de los puertos se resume en la tabla que se muestra a continuación:

No. de pin	Nombre de la señal	Dirección	Registro/bit	Invertido
1	nStrobe	Salida	C0	Si
2	Dato 0	Entrada/Salida	D0	No
3	Dato 1	Entrada/Salida	D1	No
4	Dato 2	Entrada/Salida	D2	No
5	Dato 3	Entrada/Salida	D3	No
6	Dato 4	Entrada/Salida	D4	No
7	Dato 5	Entrada/Salida	D5	No
8	Dato 6	Entrada/Salida	D6	No
9	Dato 7	Entrada/Salida	D7	No
10	nAck	Entrada	S6	No
11	Ocupado	Entrada	S7	Si
12	Sin papel	Entrada	S5	No
13	Selección	Entrada	S4	No
14	Alimentación de línea	Salida	C1	Si
15	nError	Entrada	S3	No
16	Inicializar	Salida	C2	No
17	nSelect	Salida	C3	Si
18-25	Tierra	-	-	-

Tabla III-2. Nombre de señales, pines y su correspondencia con los puertos de la computadora

III.2.3 Puerto USB

El surgimiento de los puertos y dispositivos USB (Universal Serial Bus) se ha convertido en el estándar que por excelencia ha desplazado a casi todos los mecanismos de interconexión

entre los equipos de cómputo y los periféricos que han sido desarrollados para las tareas y funciones complementarias en lo que se refiere al cómputo de hoy en día.

Ha sido surgido para ofrecer una manera de interconexión fácil y rápida sin que el usuario de los equipos y periféricos tenga grandes conocimientos a cerca de su operación, así como un mecanismo en el que se pueda mantener actualizado a un equipo con los distintos periféricos que día a día se han ido desarrollando. El uso de esta tecnología nos permite conectar y desconectar dispositivos externos, sin la necesidad de tener que reiniciar al Sistema o tener que hacer grandes tareas de configuración. En síntesis, su objetivo es de facilitar todo sin necesidad de que sepamos más que lo elemental de esta tecnología que promete mucho para las siguientes generaciones de cómputo.

La tecnología USB se trata de un Sistema de comunicación entre dispositivos electrónicos-informáticos que sólo transmite una unidad de información a la vez. El bus USB puede trabajar en dos modos, a baja velocidad (1.5 Mbps, para dispositivos como teclados, ratones, que no barajan grandes cantidades de información) y a alta velocidad (12 Mbps, para dispositivos como unidades de CDROM, bocinas, módems RTC e ISDN, etcétera). En cuanto a la comodidad, el bus USB se compacta en un cable de cuatro hilos, dos para datos, dos para alimentación. Esto supone un gran ahorro, tanto de espacio como de material. De acuerdo a estos parámetros, una de las principales ventajas que se obtiene de USB es precisamente su diseño.

El USB organiza el bus en una estructura de árbol descendente, con múltiples dispositivos conectados a un mismo bus, en la que unos elementos especiales, llamados *hubs o concentradores*, enrutan las señales en su camino desde un dispositivo al host o viceversa. Primero está el controlador del bus. Esta es la *interfaz* entre el bus USB y el bus de la computadora. De él cuelgan los dispositivos USB. Los *hubs*, como son un dispositivo USB más, también cuentan. A un hub se puede conectar uno o más dispositivos, que a su vez pueden ser otros *hubs*, así tenemos varios dispositivos conectados a un sólo controlador; como máximo alrededor de 126.

Así como ha avanzado la tecnología en los distintos periféricos y elementos electrónicos, mejorando ya sea su resolución, sonido, etc. También los dispositivos de conexión han tenido que mejorar a la par de esto; es por eso que en sus primeras versiones del USB (versión 1.0), fue diseñado para conectar periféricos como: módems, ratones, teclados, monitores, equipos estereofónicos, lectores de CD de baja velocidad a 4x o 6x, unidades de disquete, digitalizadores de imagen de baja resolución (scanner), teléfonos, conexiones ISDN, impresoras, unidades para almacenamiento en cinta, etc. En resumen toda clase de dispositivos existentes y los que vayan a crearse aprovechando las ventajas USB; la única condición, es que el dispositivo no requiera de rangos de transmisión superiores a los 12 Mbps, (también con la opción de transmisiones a 1.5 Mbps para dispositivos de baja velocidad, entre ellos los mouse) esto significa que las tarjetas de video, tarjetas de red a 100 Mbps y controladoras de discos duros particularmente, seguirían siendo tarjetas conectadas al interior de la PC. Dadas estas velocidades el Universal serial bus, es capaz de soportar hasta 127 dispositivos conectados directamente a la PC o mediante Host USB y/o combinación de periféricos, y el resto se irán conectando entre si de forma encadenada o bien empleando Hub USB.

Las modificaciones que se han hecho al estándar USB se encuentran en la velocidad de transmisión, con la cual se plantea que en estos momentos los 12 Mbps, serian revisados para dar paso a una velocidad 20 y 30 veces mayor que esta, pudiendo así ofrecer compatibilidad con las aplicaciones de usuario más exigentes sin incrementar el costo o la complejidad con respecto al objetivo anterior.

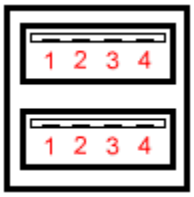
A principio de 1999, el Grupo Promotor de USB 2.0, compuesto por Compaq, HP, Intel, Lucent, Microsoft, NEC y Philips, anunció un estimado de que la velocidad de USB 2.0 sería de 120 a 240 mega bits por segundo (Mbs), o 10 a 20 veces más rápida que la de USB 1.1. Este incremento más reciente en el objetivo de velocidad, es elevada ahora a 360 a 480 Mbs, o 30 a 40 veces más rápida que la de USB 1.1. La velocidad objetivo revisada y significativamente más alta es el resultado de análisis realizados por el Grupo Promotor de USB 2.0 que concluyen que la velocidad se puede incrementar sin costo o complejidad adicionales con respecto al estimado anterior.

"Este logro en USB 2.0 impulsará aún más el gran momento por el que pasa USB en aplicaciones de usuario aún más exigentes, como la creación de imágenes y los juegos interactivos, y ofrecerá una ruta de actualización efectiva para los dispositivos periféricos USB de hoy", señaló Pat Gelsinger, vicepresidente y gerente general del Grupo de Productos de Escritorio de Intel.

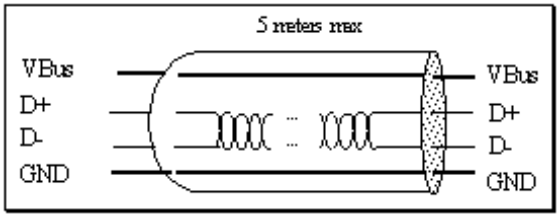
III.2.3.1 INTERFAZ FÍSICA (ELÉCTRICA)

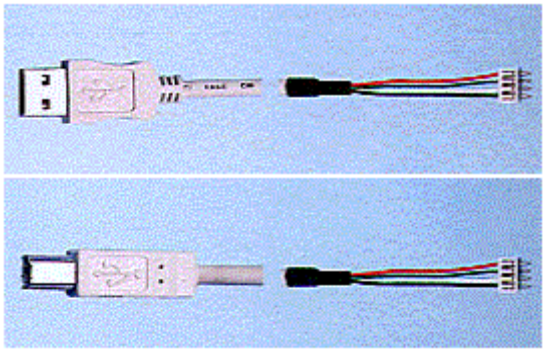
El *interfaz* de Bus Universal en Serie (USB) se identifica con este icono que se encuentra en la parte posterior de la computadora:

Los pines del conector se identifican en la siguiente tabla:

Conector	Pin	Señal
	1	+5V
	2	Datos -
	3	Datos +
	4	A Tierra

El Bus Serial Universal transfiere señales de información y energía eléctrica a través de 4 cables, cuya disposición se muestra en las siguientes figuras.





Por su parte las señales se mueven sobre dos cables y entre segmentos comprendidos entre un par de dispositivos USB, con rangos de velocidad de 12Mbps o 1.5Mbps, para transmisiones de alta y baja velocidad respectivamente. Ambos modos de transmisión son controlados automáticamente por medio de los dispositivos USB de manera transparente al usuario. Es importante notar que siempre ha sido un serio problema manejar velocidades diferentes de transmisión de datos por un mismo cable, y esto no sería posible sin que todos los dispositivos estén preparados para tal efecto.

Los pulsos de reloj o sincronismo son transmitidos en la misma señal de forma codificada bajo el esquema NRZI (Non Return To Zero Invert), uno de los más interesantes Sistemas de codificación de información que quedan fuera del alcance de este trabajo por ser muy amplio y complejo para el propósito de este proyecto.

Los otros dos cables VBus y GND tienen la misión de llevar suministro eléctrico a los dispositivos, con una potencia de +5V para VBus. **Los cables USB permiten una distancia que va de los pocos centímetros a varios metros, más específicamente 5 metros de distancia máxima entre un dispositivo USB y el siguiente**.

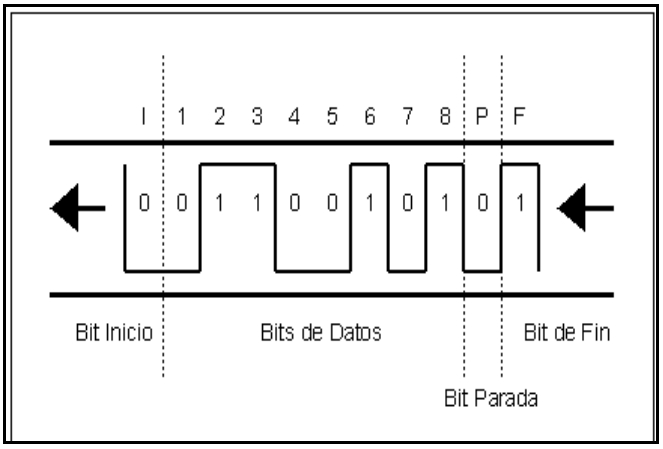
III.2.3.2 TRANSMISIÓN EN EL USB

III.2.3.2.1 TRANSMISIÓN ASINCRÓNICA

Las distintas formas de transmisión de datos a distancia siempre fueron seriales, ya que el desfase de tiempos ocasionada por la transmisión paralela en distancias grandes impide pensar en esta última como apta para cubrir longitudes mayores a algunos pocos metros.

Sobre ello, la transmisión serial ha topado con el problema de que la información generada en el transmisor sea recuperada en la misma forma en el receptor, para lo cual es necesario ajustar adecuadamente un sincronismo entre ambos extremos de la comunicación. Para ello, tanto el receptor como el transmisor deben disponer de relojes que funcionen a la misma frecuencia y posibilite una transmisión exitosa. Como respuesta a este problema surgió la transmisión asincrónica, empleada masivamente años atrás para la comunicación entre los equipos servidores conocidos como hosts y sus terminales.

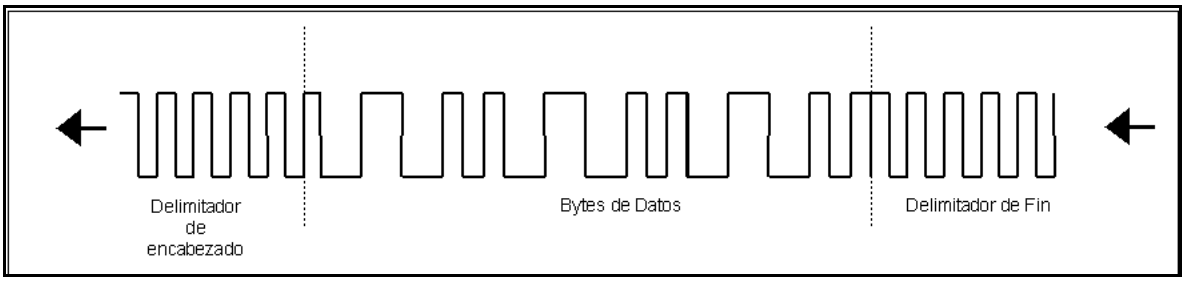
En este modelo cabe entender que ambos equipos poseen relojes funcionando a la misma frecuencia, por lo cual, cuando uno de ellos desea transmitir, prepara un grupo de bits encabezados por un BIT conocido como de arranque, un conjunto de 7 u 8 bits de datos, un BIT de paridad (para control de errores), y uno o dos bits de parada. El primero de los bits enviados anuncia al receptor la llegada de los siguientes, y la recepción de los mismos es efectuada. El receptor conocer perfectamente cuántos bits le llegarán, y da por recibida la información cuando verifica la llegada de los bits de parada. El esquema de los datos se muestra en la Figura siguiente.



Se denomina transmisión asincrónica no porque no exista ningún tipo de sincronismo, sino porque el sincronismo no se halla en la señal misma, más bien son los equipos mismos los que poseen relojes o clocks que posibilitan la sincronización. La sincronía o asincronía siempre se comprende a partir de la señal, no de los equipos de transmisión o recepción.

III.2.3.2.2 TRANSMISIÓN SINCRÓNICA

En este tipo de transmisión, el sincronismo viaja en la misma señal, de esta forma la transmisión puede alcanzar distancias mucho mayores como también un mejor aprovechamiento de canal. En la transmisión asincrónica, los grupos de datos están compuestos por generalmente 10 bits, de los cuales 4 son de control. Evidentemente el rendimiento no es el mejor. En cambio, en la transmisión sincrónica, los grupos de datos o paquetes están compuestos por 128 bytes, 1024 bytes o más, dependiendo de la calidad del canal de comunicaciones.



Las transmisiones sincrónicas ocupan en la actualidad gran parte del mundo de las comunicaciones seriales, especialmente las que emplean el canal telefónico.

III.2.3.2.3 TRANSMISIÓN ISOCRONICAS

Inicialmente vale la pena aclarar el origen de este término, ISO (algún) CRONOS (tiempo). La transmisión isocrónica ha sido desarrollada especialmente para satisfacer las demandas de la transmisión multimedial por redes, esto es integrar dentro de una misma transmisión, información de voz, video, texto e imágenes. La transmisión isocrónica es una forma de transmisión de datos en la cual los caracteres individuales están solamente separados por un número entero de intervalos, medidos a partir de la duración de los bits. Contrasta con la transmisión asincrónica en la cual los caracteres pueden estar separados por intervalos aleatorios. La transferencia isocrónica provee comunicación continua y periódica entre el host y el dispositivo, con el fin de mover información relevante a un cierto momento. La

transmisión isocrónica se encarga de mover información relevante a algún tipo de transmisión, particularmente audio y video.

III.2.3.2.4 TRANSMISIÓN BULK

La transmisión Bulk, es una comunicación no periódica, explosiva típicamente empleada por transferencias que requieren usar todo el ancho de banda disponible o en su defecto son demoradas hasta que el ancho de banda completo esté disponible. Esto implica particularmente movimientos de imágenes o video, donde se requiere de gran potencial de transferencia en poco tiempo.

USB permite dos tipos más de transferencias de datos:

III.2.3.2.5 TRANSMISIONES DE CONTROL

Es un tipo de comunicación exclusivamente entre el host y el dispositivo que permite configurar este último, sus paquetes de datos son de 8, 16, 32 o 64 bytes, dependiendo de la velocidad del dispositivo que se pretende controlar.

III.2.3.2.6 TRANSMISIONES DE INTERRUPCIÓN

Este tipo de comunicación está disponible para aquellos dispositivos que demandan mover muy poca información y poco frecuentemente. Tiene la particularidad de ser unidireccional, es decir del dispositivo al host, notificando de algún evento o solicitando alguna información. Su paquete de datos tiene las mismas dimensiones que el de las transmisiones de control.

III.3 *Interfaz electrónica*

En este apartado son abordados los conceptos básicos que implican la descripción y funcionamiento de los dispositivos electrónicos que son utilizados en la construcción de la *interfaz electrónica*, sin entrar al detalle de los fenómenos electrónicos que han dado origen a estos, así como su funcionamiento y ámbito de aplicación.

III.3.1 Registros (latches)

Los registros son un conjunto de 1 o varios latches o Flip-Flops asociados que permiten almacenar temporalmente una palabra o grupo de *n* bits.

Hay dos clases de registros típicos sincrónicos:

- El registro de entras y salidas en paralelo.
- El registro de desplazamiento

Ambos son ampliamente utilizados en el ámbito de los equipos de cómputo y en Sistemas de transmisión de datos.

El registro universal combina ambas modalidades de registro y presenta entradas para seleccionar una u otra.

Un registro “de paralelo a paralelo” (PIPO) permite transferir, de manera simultáneamente *n* bits hacia o desde el mismo.

Como ejemplo, en la siguiente ilustración se muestra la representación de un registro del tipo mencionado para 8 bits correspondiente a circuito integrado 74273.

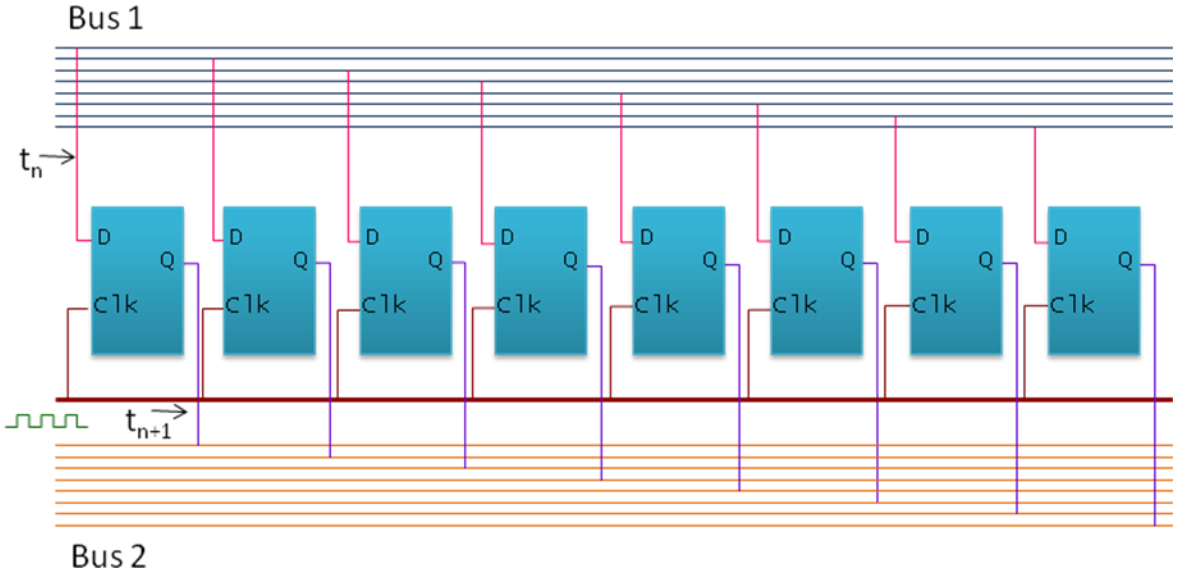


Ilustración III-5. Diagrama de un flip-flop (74273)

Consta de 8 latches tipo D con sus habilitaciones de reloj conectadas entre sí a una misma señal de reloj, así se puede cambiar al mismo tiempo todos los latches con cada pulso del reloj. De igual manera tienen todas sus conexiones de clear unidas hasta en cero de todos los latches al mismo tiempo.

Cada una de las entradas se encuentra conectada a su línea correspondiente del Bus 1 y cada salida a cada una de las correspondientes del Bus 2.

Los datos de entrada deben estar presentes en las líneas del Bus 1 y en consecuencia en las entradas de los 8 latches cuando Clk=0, en el momento en que el reloj cambia de estado Clk=1, se introducen los datos de cada latch de manera de quedar la palabra almacenada en el registro.

Luego el reloj pasa nuevamente al estado nulo y ya no importa que entonces cambien las entradas del Bus 1; la palabra quedará almacenada por el resto del período en el registro. Este proceso se suele llamar “carga” del registro.

Una vez cargados todos los latches, la palabra estará disponible en el Bus 2 durante el lapso en que Ck=0.

Ahora bien, un registro de desplazamiento es un dispositivo en el que se puede almacenar una cierta cantidad de bits en forma ordenada, pero, a diferencia del anterior, todos los bits ingresan por un mismo flip-flop y van pasando al siguiente (derecha a izquierda o de izquierda a derecha según se decida de antemano) con cada pulso de Ck, de manera que para ingresar una palabra de 8 bits hace falta que transcurran 8 pulsos de reloj.

Su operación se basa en una agrupación de memorias de un bit tipo flip-flop como ya se ha

dicho de tipo maestro-esclavo; todos los flip-flop están conectados al mismo reloj de habilitación de entrada de datos, en tanto que la salida de un flip-flop está conectada a cada entrada del siguiente.

Como ejemplo veamos un registro de desplazamiento (corrimiento) a la derecha hecho con flip-flop de tipo D, el chip 74164 y su secuencia de ingresos de datos.

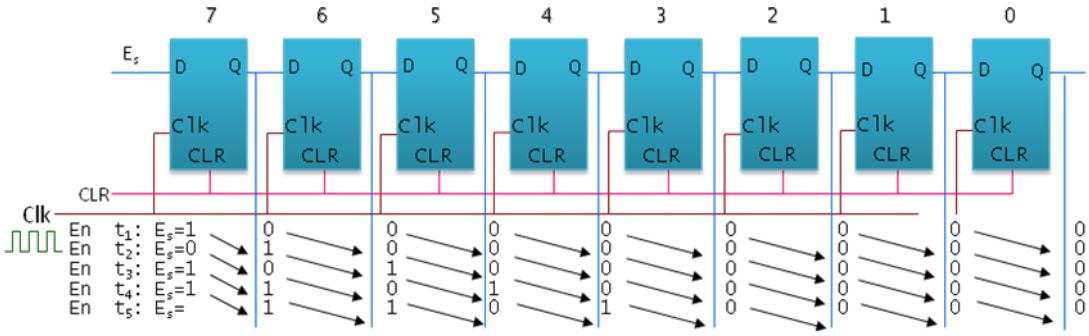


Ilustración III-6. Esquema de un registro de corrimiento a la derecha (74164)

Cuando el Clk=1 el latch maestro de cada flip-flop recibe la información del latch esclavo anterior (salvo el primer que lo hace de la entrada Es) de este lapso sólo pueden modificarse los latch maestro y la información se carga en este bloque. En el instante en que pasa Clk=0 se carga el esclavo de cada flip-flop, quedando entonces disponible para la salida Qi, como veremos la información pasa a estar disponible luego de un flanco descendiente.

En la gráfica de la Ilustración III-7 se muestra una secuencia de pulsos en los que, se muestran las salidas Q7, Q6, Q5, para los períodos t1, t2, t3 y t4, cuando se está ingresando el número 11101. Analicemos pulso a pulso lo que ocurre en el registro de desplazamiento.

Durante el período cero se colocan todas las salidas en cero (Qi=0), utilizando para ello las entradas Cr y, antes que comience t1, la entrada Es pasa a 1 o se adopta el valor del dígito de menor orden de magnitud.

Período t1: comienza cuando el reloj pasa a uno; el latch maestro del D7 adopta el valor de la entrada, (Q7=Es=1) en tanto que su salida no cambia Q7=0; recién en el flanco descendente del reloj pasa el valor de la entrada a la salida del flip-flop correspondiente (Q7=D7=1); a partir de este instante la entrada puede cambiar su valor para ser tomado por el latch maestro. El resultado es que al finalizar el pulso t1 las salidas valen Q7=1, Q6=0, Q5=0, Q4=0.

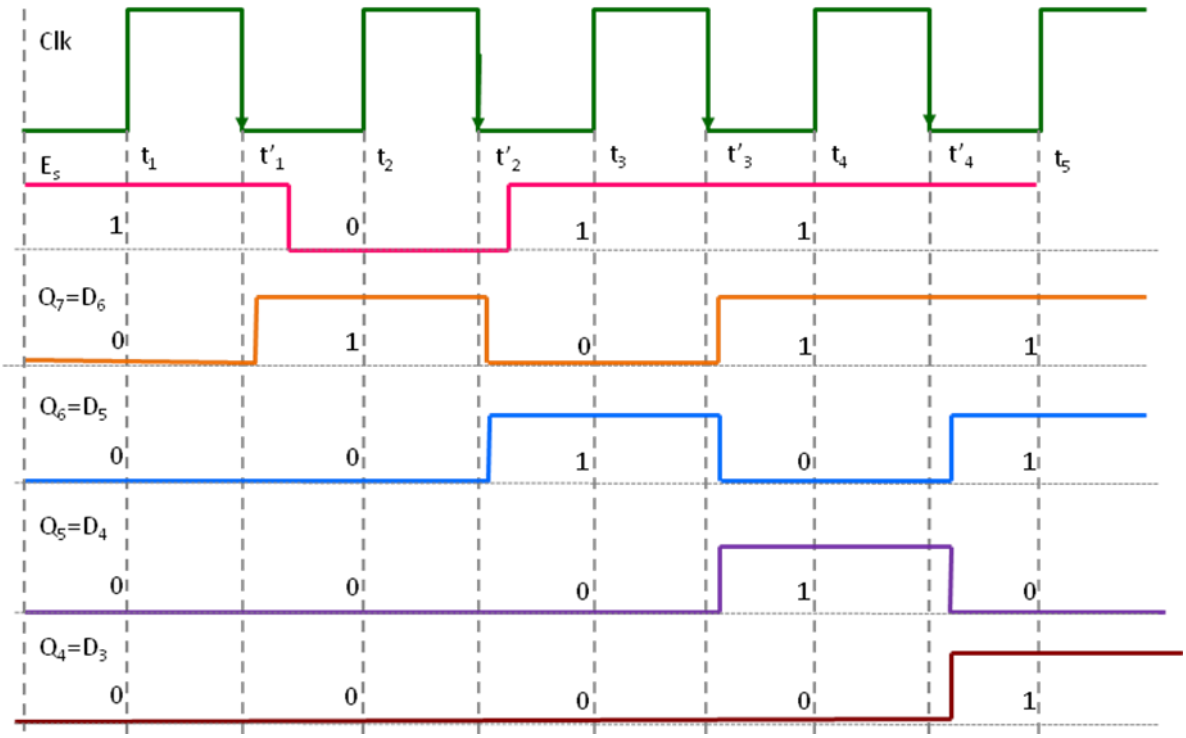


Ilustración III-7. Diagrama de tiempos del ejemplo de latch de corrimiento

En la primer parte del período t2, (cuando el reloj pasa al estado 1) el latch maestro del D7 adopta el valor de la entrada (D7=Es=0), en tanto que el latch maestro del D6 adopta el valor del latch esclavo del D7 (D6=Q7=1). En la segunda parte de éste período recién quedará disponible la información en las salidas Qi, y la entrada Es podrá tomar el valor del dígito de tercer orden de magnitud, así al terminar el pulso t2, las salidas valen Q7=0, Q6=1, Q5=0 y Q4=0.

Se podría completar el gráfico mostrando las salidas Qi de los latches maestros de cada flip-flop para ver que cambian de estado durante el flanco ascendente de pulso.

Siguiendo con el razonamiento, resulta evidente que al concluir el cuarto pulso t4 las salidas valen Q7=1, Q6=1, Q5=0 y Q4=1.

Si observamos cómo varía el vector formado por las salidas al cabo de cada pulso Q7, Q6, Q5, Q4, Q3, Q2, Q1 y Q0 tenemos que:

t_i	Q7Q6Q5Q4Q3Q2Q1Q0
t_0	00000000
t_1	10000000
t_2	01000000
t_3	10100000
t_4	11010000

Se ve la forma en que van variando los valores del registro que hay un desplazamiento de la información hacia la derecha por cada pulso lo que da origen al nombre del registro.

Es de hacer notar que en la presente explicación se usó como flanco de cambio en las salidas del flanco decreciente por una cuestión de comodidad ya que en realidad el planco de cambio en las salidas de este registro de desplazamiento es el creciente.

Existen distintas variantes de este tipo de dispositivos tanto para flanco de subida, de bajada, convertidores serie-paralelo, paralelo-serie, registros programables entre otros.

III.3.2 Decodificadores

Un decodificador es un circuito lógico que convierte un código binario de entrada de **n** (cualquier entero) bits en **m** (entero que es menor o igual a 2^n) líneas de salida de manera tal que cada una de estas líneas sólo sea activada para una posible combinación de entradas. En la ilustración III-5, se muestra el diagrama general del decodificador con n entradas y **m** salidas. Ya que cada una de las **"n"** entradas puede ser 0 o bien 1, hay 2^n posibles combinaciones o códigos de entrada. Para cada una de estas combinaciones de entrad sólo una de las **m** salidas será activa (ALTA); todas las otras son BAJAS. Muchos decodificadores están diseñados para producir salidas activas en BAJO, donde totalmente la salida seleccionada es BAJA, en tanto que todas las otras son ALTAS. Esto siempre lo indica la presencia de pequeños círculos en las líneas de salida del diagrama del decodificador en particular.

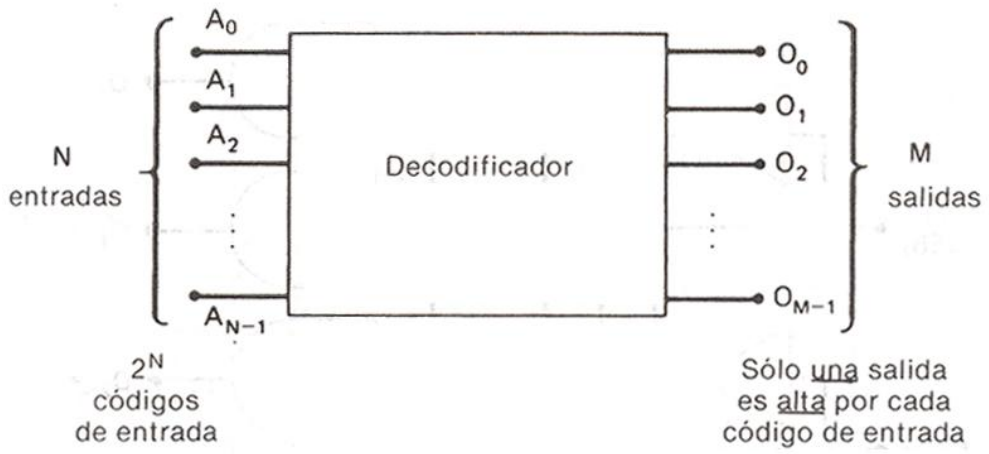


Ilustración III-8. Diagrama general del decodificador

Algunos decodificadores no utilizan los 2^n posibles códigos de entrada, sino sólo unos cuantos. Por ejemplo, un decodificador de BCD a deciman tiene un código de entrada de cuatro bits y 10 líneas de salida que corresponden a los 10 grupos de código BCD que van del 0000 hasta el 1001. Los decodificadores de este tipo a menudo están diseñados de forma tal que si cualquiera de los códigos que no se utilizan, se aplican a la entrada, ninguna de las salidas se activa.

Algunos decodificadores tienen una o más entradas de habilitación que se utilizan para controlar la operación del decodificador, por lo que el dispositivo será habilitado sólo si la señal de habilitación se encuentra activa dependiendo de las características del componente, así pues, hay algunos decodificadores que se activan con señal en alta o en baja. Esta característica ha de ser considerada al momento de seleccionar los dispositivos a utilizar en los proyectos particulares de cada proyecto.

Los decodificadores se emplean todas las veces que es necesario activar una salida o grupo de éstas cuando se presenta una combinación específica de niveles de entrada. Estos niveles son proporcionados con frecuencia por las salidas de un contador o registro. Cuando las entradas del decodificador provienen de un contador al que llegan pulsos de manera continua, las salidas del decodificador se activan en forma secuencial y se pueden emplear para temporizar o poner señales en secuencia para apagar y encender dispositivos en instantes específicos.

Los decodificadores se emplean mucho en el Sistema de memoria de una computadora, donde responden a la dirección proveniente de la unidad central (CPU) para activar la localidad de memoria indicada por la dirección deseada.

Otra gran área de aplicación para estos dispositivos es la conversión de datos binarios hacia una forma más adecuada para su presentación en dispositivos de visualización decimal.

III.3.3 Triacs

Los triacs son dispositivos semiconductores bidireccionales con tres terminales; o sea, puede conducir la corriente en ambos sentidos. Las terminales ánodo y cátodo se han bautizado con el nombre de MT1 y MT2 que son la abreviatura de Main Terminal (Terminal Principal) 1 y 2 respectivamente.

El circuito equivalente para un triac se puede formar con dos SCR en paralelo, pero con sus polaridades invertidas (Ilustración III-9). Cuando se aplica el pulso de activación en la terminal compuerta, no importa la polaridad aplicada a las terminales MT; la razón es que uno de los SCR se encontrará polarizado directamente por lo que éste conducirá.

El SCR1 se encuentra polarizado en forma inversa y el SCR2 en forma directa cuando se aplica el pulso a la compuerta G, solamente este ultimo conducirá. Si se invierte la polaridad de alimentación y se aplica el pulso de disparo nuevamente en la compuerta G, sólo el SCR1 conducirá.

El efecto total del dispositivo es el de permitir el paso de la corriente eléctrica, independiente de la polaridad de la tensión aplicada en las terminales MT.

Los parámetros a considerar cuando se elige un TRIAC, son iguales a los utilizados con los SCR; la única diferencia es que el VRRM o voltaje inverso no existe en el caso de los TRIACs, debido a que no importa la polaridad de sus extremos.

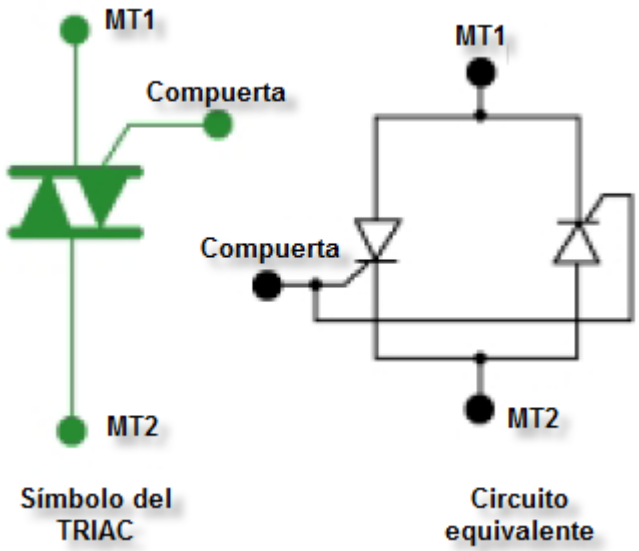


Ilustración III-9. Diagrama de un TRIAC y su equivalente con SRC

III.3.3.1 TRIAC con acoplado óptico (opto coupler TRIAC)

Los TRIACs acoplados ópticamente combinan un diodo emisor de luz (LED) con un TRIAC foto-detector (foto-TRIAC) dentro de un mismo encapsulado opaco con un esquema mostrado en la Ilustración III-10. Al no existir conexión eléctrica entre la entrada y la salida, el acoplado es unidireccional (LED a foto-TRIAC) y permite un aislamiento eléctrico entre ambos dispositivos hasta 700 volts (typ). Además, algunos foto-TRIACs incluyen un circuito de detección de paso por cero que permite sincronizar señales de la red eléctrica con señales de control del LED para ajustar el ángulo de conducción.

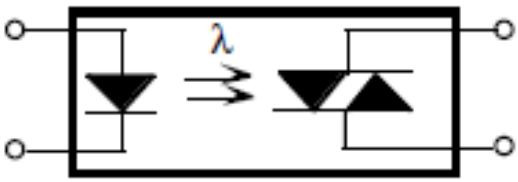


Ilustración III-10. Esquema del foto-TRIAC

III.3.4 Optoacopladores

También se denominan dispositivos de acoplamiento óptico. Emplean un haz de radiación luminosa para pasar señales de un circuito a otro sin conexión eléctrica. Fundamentalmente este dispositivo está formado por una fuente emisora de luz, y un fotosensor de silicio, que se adapta a la sensibilidad espectral del emisor luminoso.

Un opto acoplador combina un dispositivo semiconductor formado por un fotoemisor, un fotorreceptor y entre ambos hay un camino por donde se transmite la luz. Todos estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP.

III.3.4.1 Tipos de Optoacopladores

Existen varios tipos de optoacopladores cuya diferencia entre sí depende de los dispositivos

de salida que se inserten en el componente. Según esto tenemos los siguientes tipos:

- Fototransistor o lineal, conmuta una variación de corriente de entrada en una variación de tensión de salida. Se compone de un opto acoplador con una etapa de salida formada por un transistor BJT. Se utiliza en acoplamientos de líneas telefónicas, periféricos, audio, etc.
- Optotiristor: Diseñado para aplicaciones donde sea preciso un aislamiento entre una señal lógica y la red.
- Optotriac: Se compone de un opto acoplador con una etapa de salida formada por un triac. Al igual que el optotiristor, se utiliza para aislar una circuitería de baja tensión a la red.
- Fototriac de paso por cero: Opto acoplador en cuya etapa de salida se encuentra un triac de cruce por cero. El circuito interno de cruce por cero conmuta al triac sólo en los cruce por cero de la corriente alterna.

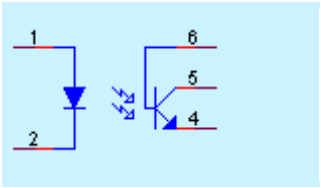
En general pueden sustituir a relés ya que tienen una velocidad de conmutación mayor, así como, la ausencia de rebotes.

III.3.4.2 Funcionamiento

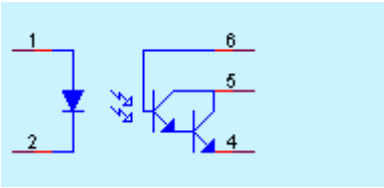
La señal de entrada es aplicada al fotoemisor y la salida es tomada del fotorreceptor. Los optoacopladores son capaces de convertir una señal eléctrica en una señal luminosa modulada y volver a convertirla en una señal eléctrica. La gran ventaja de un opto acoplador reside en el aislamiento eléctrico que puede establecerse entre los circuitos de entrada y salida.

Los foto emisores que se emplean en los optoacopladores de potencia son diodos que emiten rayos infrarrojos (IRED) y los foto receptores pueden ser tiristores o transistores.

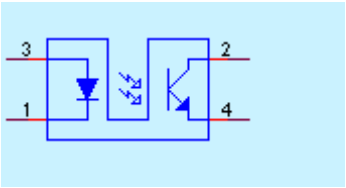
Cuando aparece una tensión sobre los terminales del diodo IRED, este emite un haz de rayos infrarrojo que transmite a través de una pequeña guía-ondas de plástico o cristal hacia el fotorreceptor. La energía luminosa que incide sobre el fotorreceptor hace que este genere una tensión eléctrica a su salida. Este responde a las señales de entrada, que podrían ser pulsos de tensión.



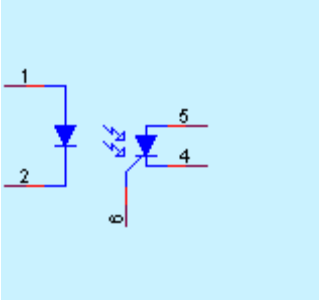
Símbolo del opto transistor



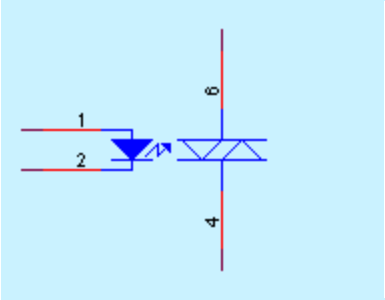
Símbolo de un opto transistor en configuración Darlington



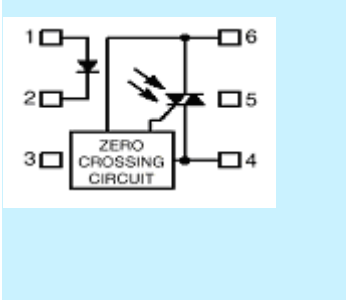
Símbolo de un opto transistor de encapsulado ranurado



Símbolo del Optotiristor



Símbolo Optotriac



Fototriac de cruce por cero

Tabla III-3. Símbolos representativos de los optoacopladores

III.3.4.3 Encapsulados

El encapsulado varía en función del tipo de opto acoplador y de su aplicación, así como del número de unidades que se encuentren en su interior. Normalmente, los pines del elemento emisor están a un lado de la cápsula y los del sensor en el lado opuesto.

Existen unos encapsulados diferentes en los que, físicamente se puede interrumpir el haz luminoso (usados para control de posición, nº de revoluciones, cerraduras...). De esta forma el encapsulado presenta una ranura entre el emisor y el receptor. Se les denomina de cápsula ranurada o fotocélulas de herradura.

Aspecto de un encapsulado DIL de 6 patillas

Encapsulados DIP-8 y DIP-14

Tabla III-4. Encapsulados para optoacopladores

III.3.5 Micros controladores

Los micro controladores son dispositivos electrónicos de un nivel alto de integración cuya finalidad es la de ofrecer un medio para solucionar tareas de diversa índole en los tiempos actuales. Estos dispositivos se distinguen por la potencialidad que ofrece la versión ampliada de estos que son los microprocesadores, con la diferencia de que además de incluir los componentes lógicos como lo son la unidad aritmético lógica, los registros de operación, los medios de control del programa (mediante el contador de programa o PC), de las líneas de interrupción y de toda la gama de características con que un microprocesador es utilizado para la realización de tareas complejas y rutinarias en los equipos de cómputo contemporáneas, los micro controladores incluyen consigo, algunas con características más reducidas que los microprocesadores; incluyen sus propios periféricos y memoria física integrada en el encapsulado del dispositivo.

En pocas palabras, el micro controlador es una pequeña computadora completa, claro está, sin los dispositivos complementarios o periféricos como los conocemos en nuestros equipos de uso

cotidiano.

Los micro-controladores han surgido con la necesidad de realizar tareas de control, de instrumentación y de todas aquéllas tareas que impliquen la implementación e instrumentación de Sistemas complejos de la vida cotidiana.

Los micro controladores han encontrado un campo fértil de aplicación en Sistemas como los video juegos, en los equipos de esparcimiento con que se cuenta en un hogar de estos tiempos como lo son los televisores, equipos de audio, juguetes de los niños y en una interminable lista de productos que han hecho de nuestras vidas una manera nueva de "vivir" y convivir con los demás y con nuestro entorno mismo.

Han tenido auge en la industria a gran escala pues no solamente hemos sido impactados dentro de nuestros hogares sino también fuera de ellos por doquiera que miremos, encontraremos algún aparato o instrumento en el que se hace uso de ellos.

Mas allá de estos enfoques de la apreciación y utilización de los micro controladores, se puede encontrar también su aplicación en la medicina, en la robótica e infinidad de temas que son los que regirán en un corto y mediano plazo la vida y convivencia del ser humano como individuo y sociedad.

En el mercado, se encuentran distintas empresas que han desarrollado este tipo de componentes, cada día les han dotado de más características y capacidades, y se pueden contar con la mano las empresas que llevan la batuta en este nicho de mercado: por una parte se encuentra Microchip, Motorola, Intel y Dallas semiconductor, entre otros.

El uso de estos productos se ha optimizado de tal manera, que resulta más fácil programar a los micro controladores que a los mismos microprocesadores debido a que utilizan una tecnología denominada **RISC** (Reduced Instruction Set Code) que implica el uso de un conjunto de instrucciones reducido, de fácil entendimiento y apoyado en localidades de memoria interna en la que se guardan los valores de las variables implícitas de las operaciones propias del micro controlador.

Utilizan una tecnología denominada Harvard en cuanto a los buses internos de datos y dirección ya que ésta permite tener la independencia física entre ambos buses y por consiguiente se emplea un bus para direcciones y uno para datos a diferencia de otros productos que emplean un mismo canal para ambos buses.

En este proyecto se utiliza un micro controlador de Microchip de la familia de rango medio ya que sus tareas y operación son justificadas en el Capítulo dedicado a la explicación de la *interfaz electrónica* en conjunto; apoyada la decisión de usar este componente por su costo y rapidez de asimilación en su estructura y operación.

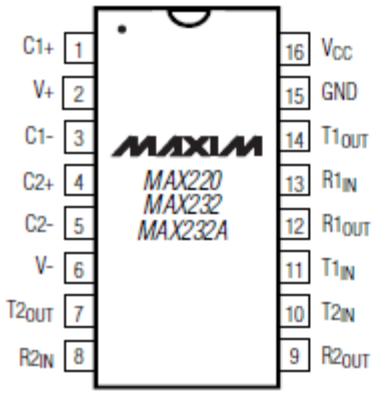
III.3.6 Convertidor de señal de niveles RS232 a TTL

Adelantándose un poco al criterio de selección del puerto de comunicación que se empleará para la construcción de la *interfaz electrónica*, aquí se mencionará la manera en que se puede emplear a las señales que son propias de la norma RS232 para los puertos seriales, como ya ha sido descrito en el apartado III.2.1. Puerto Serial.

Como ya se ha indicado, el puerto serie utiliza un mecanismo de codificación denominado NRZ (Non Return to Zero) que emplea una polarización inversa al dato que se representa,

apoyándose en niveles de voltaje que van del orden de -12 a 12 volts y como la lógica de la *interfaz* emplea niveles TTL, es necesario hacer la conversión de estos niveles. En el mercado existen varios dispositivos que se encargan de realizar esta tarea, pero se ha elegido el dispositivo MAX-232 que encapsula dos pares de conversión RS232-TTL y dos TTL-RS232.

Los detalles de este dispositivo se pueden encontrar en las hojas de especificaciones a los que se hace referencia en la descripción de la *interfaz electrónica*. Mientras, en la siguiente ilustración se muestra la distribución y configuración del dispositivo.



DIP/SO

CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

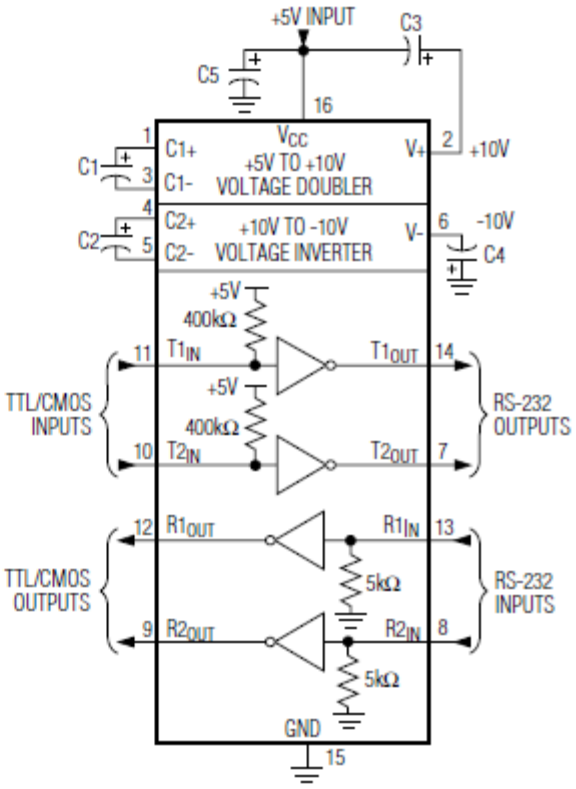


Ilustración III-11. Apariencia del dispositivo MAX-232

CAPÍTULO IV: DESARROLLO DEL SISTEMA

Ha llegado el momento de presentar el detalle de la manera en que se encuentra construido el Sistema en conjunto: software e *interfaz electrónica*. A lo largo del Capítulo se aborda en forma descriptiva sin caer demasiado en detalle la manera en que se ha estructurado la Base de Datos que será el punto central donde se almacenará la información que se irá generando con forme los usuarios vayan haciendo uso de las instalaciones del Laboratorio, así como de los servicios que brinda a quienes así lo solicitan.

Primeramente se hace el estudio de la estructura general de la creación de las Base de Datos y en seguida se describe esquemáticamente y mediante el script que permite al usuario crear la estructura de las distintas entidades y atributos con que están diseñadas, permitiendo así tener un panorama general de lo que será la sustancia del Sistema.

Una vez que se haya hecho la descripción de la Base de Datos, tocará el turno a la descripción del software, en donde serán proporcionados los elementos básicos necesarios para que se pueda entender la manera en que se ha estructurado a la solución y, para poder abordar la documentación gráfica de todos los componentes del Software. Por espacio no serán incluidos los códigos fuente, pues de hacerlo así, se tendría que escribir este trabajo en tomos y la intención es que sea lo más analíticamente posible su comprensión, basándome en los diagramas de clase. Si se tiene la inquietud de ver la estructura del Software, se podrá consultar en cualquier momento desde el CD 2 que acompaña a este trabajo.

Como siguiente tema, será descrito el protocolo de comunicación entre el software y la *interfaz electrónica* para que al final de este Capítulo se dedique la atención al modelaje, diseño y construcción del prototipo según ha sido propuesto y considerando los componentes que han sido empleados y probado para este fin.

IV.1 Elección del puerto de comunicaciones

En el Capítulo anterior en la parte dedicada a los conceptos generales del Hardware, fueron incluidos los conceptos generales en relación a cada uno de los tres tipos de puertos de comunicación que en la actualidad son utilizados para la conexión de los equipos de cómputo con los distintos dispositivos que se han diseñado y producido en serie para los proyectos actuales en cuanto a las aplicaciones que se les ha dado a las computadoras de manera general y para el propósito de este trabajo, las personales.

Cada uno de los puertos ofrece características únicas pero para decidir sobre uno en particular implica el análisis de los requerimientos que ya se expresaron en el Capítulo II y, aunado a ello es necesario considerar los costos que esto implica.

También se ha dicho que los equipos de cómputo de la actualidad vienen reflejando la tendencia a desaparecer a los puertos tradicionales: serial y paralelo, para dar entrada por completo a la tecnología basada en los puertos USB y, haciéndole competencia vienen los puertos IEEE 1394a.

Por la naturaleza y características del caso particular de este proyecto, primeramente se debe considerar el factor primordial: la distancia entre el equipo de cómputo y la *interfaz electrónica* y si se observa que esta distancia en promedio es de 7.5 metros para este proyecto; el puerto paralelo y USB quedan de entrada fuera de esta consideración pues los niveles de las señales que utilizan son TTL es decir de alrededor de 0 a 5 volts y por la

distancia y condiciones del entorno (balastras y en consecuencia campos electromagnéticos básicamente) podría existir problemas para utilizarlos.

Otra situación implica la tarea de programar los puertos, pues como ya se indicó en el Capítulo III, en el caso del puerto paralelo, el Sistema Operativo lo controla bajo el concepto de "Recursos Administrados", esto quiere decir que el acceso a los recursos asociados a este tipo de puertos de computadora se encuentran protegidos y no pueden ser accedidos directamente por una aplicación de este tipo, para ello se necesitaría contar con una versión del compilador que permitiese programar con derechos de administración del Sistema (no confundir con los derechos y privilegios de los tipos de cuenta y usuarios del entorno de Windows XP, 2000 y posterior) motivo por el cual se requeriría de una licencia cuyo costo no es nada barato (del orden del $20 mil pesos aproximadamente). Durante la búsqueda de las herramientas y componentes que conforman este proyecto, pudo encontrase con que en internet ya circula una biblioteca de funciones (DLL) que fue diseñada por alguien quien tiene acceso al compilador que permite hacer esta tarea, fue probada y funcionó pero en realidad, no se sabe lo que el código contenga en realidad y como es de saber, actualmente mucho del código malintencionado o virus se propaga bajo este tipo de situaciones. Como complemento a esta situación, la distancia sería el segundo inconveniente para no utilizar este puerto, además de la cantidad de cables que se requieren para hacer uso del puerto paralelo utilizando un conector DB-25 como ya se ha indicado.

El caso de los puertos USB, requieren también el uso de una licencia del software de desarrollo, pues se necesita acceder a los mecanismos internos del Sistema Operativo mediante los cuales se captan los paquetes de datos que son enviados desde y hacia los distintos dispositivos y esta tarea tiene su implementación en la estructura operativa del Sistema Operativo. Además, una vez la distancia vuelve a ser la limitante.

Si surgiera el caso en que el equipo de computo que se emplee para este proyecto tuviera las características tecnológicas recientes, se puede presentar la situación de que éste ya no cuente con puertos serie ni paralelo, pero en el mercado de componentes o dispositivos de cómputo ya se venden dispositivos de conversión de USB a puerto serial o de USB a puerto paralelo. No es momento de cantar victoria, pues también heredan las limitantes propias del alcance que puede existir entre el puerto USB y el extremo del puerto que emula (de 1.8m como máximo).

Por lo anterior, una vez más no sería rentable utilizar este tipo de convertidores.

En el mismo proceso de investigación de componentes y disponibilidad de los mismos para poderlos utilizar en este proyecto, se encontró que hay a la venta tarjetas de expansión de puertos seriales o paralelos montados en tarjetas tipo PCI o PCI express, motivo por el cual el problema de disponibilidad de al menos un puerto serial queda resuelto.

Por lo tanto, el puerto serie o serial es el más adecuado por las razones expuestas, además de que no se requieren tasas altas de transmisión entre el Software y la *interfaz electrónica* y que el puerto serial permite configurar y adaptarse a las necesidades. Las demás características y formas de operar de este puerto se pueden revisar en el Capítulo III, en el apartado III.2.1. Puerto serial.

IV.2 Desarrollo de la Base de Datos bajo SQL Server Express

En el Capítulo II, han sido indicadas las variedades de productos que existen en el mercado (por lo menos los más comunes) para el desarrollo de aplicaciones de escritorio y distribuidas, así como las características y restricciones que estas guardan respecto de los costos de licencias y uso tanto de software de desarrollo como de Bases de Datos.

Ha sido elegido Microsoft SQL Server 2005 Express, por ser una aplicación que cubre estas características y porque ha sido diseñado para trabajar estrechamente con las herramientas de desarrollo de Visual Studio .NET 2005 (para del proyecto, a Visual Basic .NET 2005 Express) e inter-construidas bajo el esquema de la tecnología FrameWork 2.0, como también se puede revisar en el apartado II.3 Software de desarrollo y el apartado III.1 Software.

En la Ilustración IV.1 que se encuentra en la siguiente página se presenta el esquema general del modelo entidad-relación de la Base de Datos cuyo nombre es **SAEALAB**. Para una descripción más detallada de la sintaxis de los comandos utilizados para la creación de la Base de Datos y de las entidades de ella, por favor revise los libros en pantalla de Microsoft SQL Server con las siguientes direcciones:

- ***ms-help://MS.SQLCC.v9/MS.SQLSVR.v9.es/tsqlref9/html/1e068443-b9ea-486a-804f-ce7b6e048e8b.htm***

- ***ms-help://MS.SQLCC.v9/MS.SQLSVR.v9.es/tsqlref9/html/29ddac46-7a0f-4151-bd94-75c1908c89f8.htm.***

El script que permite la creación de la Base de Datos en blanco dentro del Sistema de Administración de Bases de Datos Microsoft SQL Server 2005 Express se muestra en las siguientes líneas:

```
USE [master]
GO
CREATE DATABASE [SAEALAB] ON  PRIMARY
( NAME = N'SAEALAB', FILENAME = N'c:\Archivos de programa\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\SAEALAB.mdf' , SIZE = 4800KB , MAXSIZE = UNLIMITED, FILEGROWTH =
1024KB )
 LOG ON
( NAME = N'SAEALAB_log', FILENAME = N'c:\Archivos de programa\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\SAEALAB_log.LDF' , SIZE = 560KB , MAXSIZE = 2048GB , FILEGROWTH =
10%)
 COLLATE Modern_Spanish_CI_AS
GO
EXEC dbo.sp_dbcmptlevel @dbname=N'SAEALAB', @new_cmptlevel=90
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [SAEALAB].[dbo].[sp_fulltext_database] @action = 'enable'
end

GO
```

Código 1. Creación de la Base de Datos SAEALAB

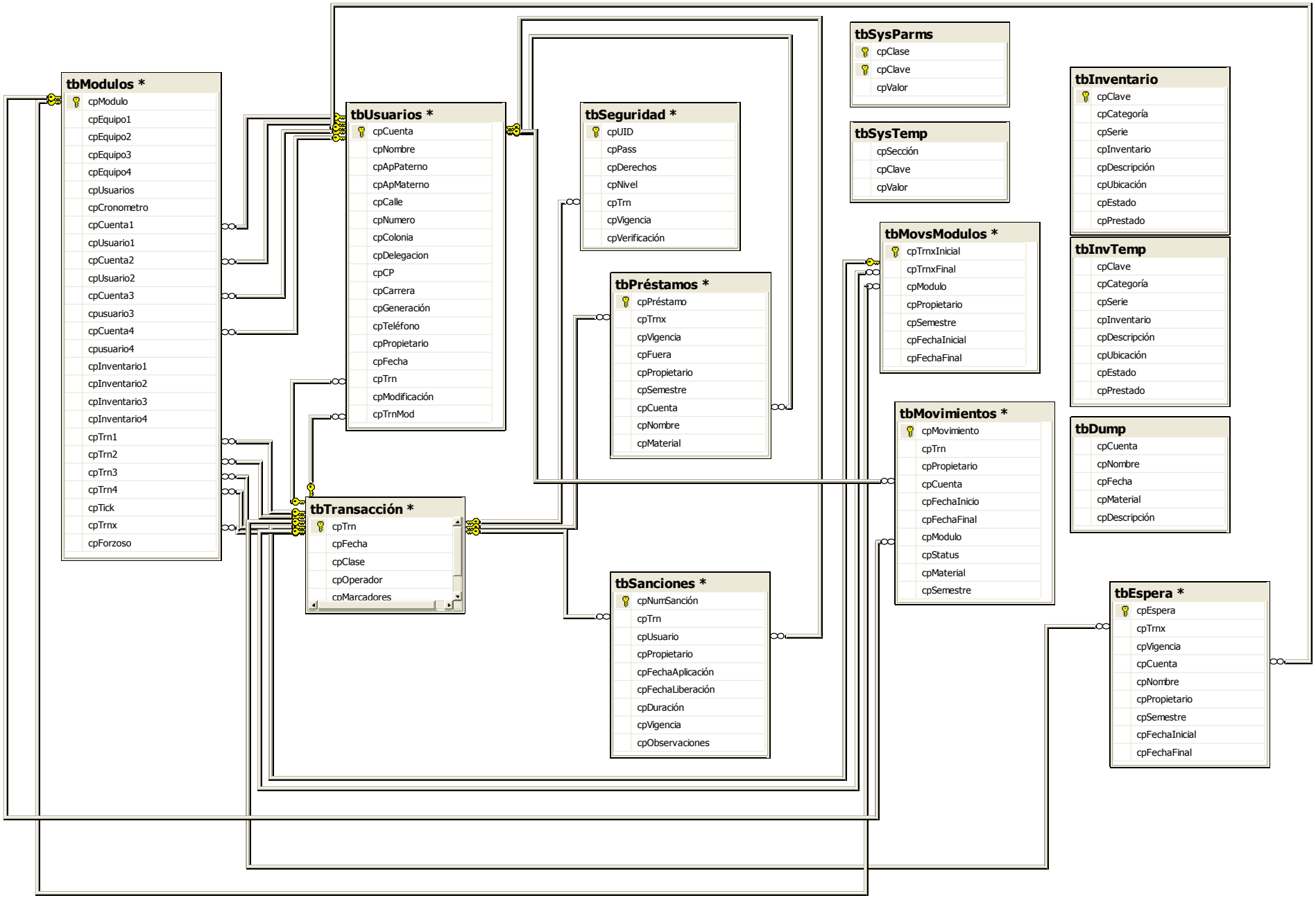


Ilustración IV-1. Modelo Entidad-Relación de la Base de Datos SAEALAB

Antes de comenzar con la descripción, se requiere precisar la nomenclatura utilizada para describir los elementos que componen a los elementos de la Base de Datos. La nomenclatura es la llamada húngara que consiste en anteponer al nombre o descriptor letras que designan el tipo de elemento al que corresponde; esta notación se ocupa con frecuencia en los lenguajes de programación como una regla no escrita para poder identificar fácilmente el tipo de variable o componente al que se refiere el identificador. Para el caso de las entidades o tablas se antepondrá las letras **"tb"** y para el caso de atributos o campos, las letras **"cp"**.

IV.2.1 Entidad tbDump

Esta es una Entidad cuyo propósito es el de guardar información temporal de los materiales prestados por número de cuenta de los usuarios y se utiliza para el procesamiento de las consultas y estadísticas por usuario. Los atributos de la entidad son: cpCuenta, cpNombre, cpFecha, cpMaterial y cpDescripción. Su significado es intuitivo al nombre que se le ha asignado, observando que cpNombre contiene el nombre y apellidos juntos. La estructura y definición de la entidad se muestra a continuación.

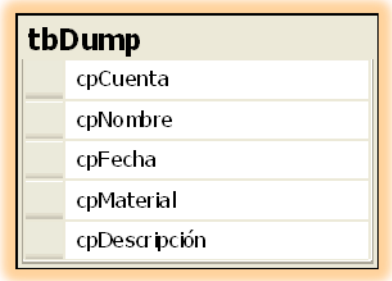


Ilustración IV-2. Estructura de la entidad tbDump

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[tbDump](
       [cpCuenta]          [nchar](9)     COLLATE Modern_Spanish_CI_AS NULL,
       [cpNombre]          [nchar](60)  COLLATE Modern_Spanish_CI_AS NULL,
       [cpFecha]           [datetime] NULL,
       [cpMaterial] [nchar](5)    COLLATE Modern_Spanish_CI_AS NULL,
       [cpDescripción]     [nchar](255) COLLATE Modern_Spanish_CI_AS NULL
) ON [PRIMARY]
```

Código 2. Creación de la Entidad tbDump

IV.2.2 Entidad tbEspera

La finalidad de esta Entidad es la de registrar la información correspondiente a todos los usuarios que se encuentran en la lista de espera de que se les proporcione los servicios solicitados al *Operador del Sistema* o Administrador del Laboratorio en turno. Los atributos que la forman son: cpEspera cuyo contenido es un valor numérico único y ascendente que el SABD asignará automáticamente en incrementos de 1, cada vez que se agregue un usuario nuevo a la lista de espera y sirve como índice para establecer relaciones entre las distintas Entidades de la Base de Datos; cpTrnx que es un valor numérico correspondiente a una transacción registrada en la Entidad tbTransacción; cpVigencia que es un valor que indica si el registro (tupla) correspondiente al número de espera cpEspera ya ha sido atendido o no;

cpCuenta corresponde al número de cuenta del usuario al que se está atendiendo; cpNombre que es el nombre completo del usuario; cpPropietario que es el código de identificación del *Operador del Sistema* quien le está atendiendo; cpSemestre es el Semestre en el que se ha registrado el movimiento; cpFechaInicial que es un valor de fecha y hora que corresponde al momento de inicio en que se colocó en lista de espera al usuario y, cpFechaFinal que como en el caso anterior mantiene una fecha y hora que corresponde al momento en que se le quitó de la lista de espera al usuario.

Ilustración IV-3. Estructura de la entidad tbEspera

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbEspera](
       [cpEspera] [bigint] IDENTITY(1,1) NOT NULL,
       [cpTrnx] [bigint] NOT NULL CONSTRAINT [DF__tbEspera__cpTrnx__489AC854]  DEFAULT ((0)),
       [cpVigencia] [int] NOT NULL,
       [cpCuenta] [varchar](9) COLLATE Modern_Spanish_CI_AS NOT NULL CONSTRAINT
[DF__tbEspera__cpCuen__498EEC8D]  DEFAULT (''),
       [cpNombre] [varchar](45) COLLATE Modern_Spanish_CI_AS NOT NULL CONSTRAINT
[DF__tbEspera__cpNomb__4A8310C6]  DEFAULT (''),
       [cpPropietario] [varchar](25) COLLATE Modern_Spanish_CI_AS NOT NULL CONSTRAINT
[DF_tbEspera_cpPropietario]  DEFAULT (''),
       [cpSemestre] [varchar](6) COLLATE Modern_Spanish_CI_AS NOT NULL CONSTRAINT
[DF_tbEspera_cpSemestre]  DEFAULT (''),
       [cpFechaInicial] [datetime] NULL,
       [cpFechaFinal] [datetime] NULL,
 CONSTRAINT [PK_tbEspera] PRIMARY KEY CLUSTERED
(
       [cpEspera] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

Código 3. Creación de la Entidad tbEspera

IV.2.3 Entidad tbInventario

Como advierte el nombre de esta Entidad, su propósito es el de contener la información de la descripción de los distintos bienes que conforman el Inventario General de los bienes del Laboratorio Abierto de la Facultad de Ingeniería. Su estructura está formada por los

siguientes atributos: cpClave que es un campo numérico además de único y con la característica de servir de índice para la finalidad de establecer relaciones entre las distintas entidades de la Base de Datos; cpCategoría que es un atributo de tipo numérico y que sirve para catalogar los bienes principalmente en: Manuales, Herramientas e Instrumentos; cpSerie que corresponde con el número de serie del bien que describe la tupla; cpInventario que corresponde con el número de inventario que tiene asignado el bien en caso de que posea uno; cpDescripción que corresponde con una breve descripción del bien asociado al cpClave; cpUbicación que corresponde con un código numérico que define la ubicación física del bien, para ver las ubicaciones definidas y cómo definir más revise el apartado del archivo de instalación **SAEALABFI**.INI del manual de referencia; cpEstado que sirve para describir el estado físico que guarda el bien al que corresponde la tupla; y, cpPrestado cuyo tipo es numérico que indica si este bien se encuentra o no prestado.

Ilustración IV-4. Estructura de la entidad tbInventario

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbInventario](
	[cpClave] [int] IDENTITY(1,1) NOT NULL,
	[cpCategoría] [int] NULL CONSTRAINT [DF_tbInventario_cpClase]  DEFAULT ((0)),
	[cpSerie] [varchar](12) COLLATE Modern_Spanish_CI_AS NULL,
	[cpInventario] [varchar](12) COLLATE Modern_Spanish_CI_AS NULL,
	[cpDescripción] [varchar](255) COLLATE Modern_Spanish_CI_AS NULL,
	[cpUbicación] [int] NULL,
	[cpEstado] [varchar](30) COLLATE Modern_Spanish_CI_AS NULL,
	[cpPrestado] [int] NULL CONSTRAINT [DF_tbInventario_cpPrestado]  DEFAULT ((0)),
 CONSTRAINT [PK__tbInventario__1BFD2C07] PRIMARY KEY CLUSTERED
(
	[cpClave] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
```

Código 4. Creación de la Entidad tbInventario

IV.2.4 Entidad tbInvTemp

La finalidad de esta Entidad es la de apoyar las tareas de instalación de la información del Inventario y, para las tareas de administración del Inventario General. Su estructura está formada por los siguientes atributos: cpClave cuya finalidad es como en el caso anterior de contener la clave única que el Sistema asignará a cada uno de los bienes del Inventario de

manera única y por consiguiente, irrepetible; cpCategoría contiene un valor numérico que establece una manera de identificación del bien cuyas categorías principales son: Manuales, Herramientas e Instrumentos; cpSerie contiene el número de serie del bien, si es que posee alguno, sino el valor será cero; cpInventario, aquí al igual que en el caso anterior se mantiene el número de serie del bien si es que lo posee y si no es así queda con valor de cero; cpDescripción, contiene una cadena que describe al bien; cpUbicación contiene un valor numérico codificado que indica la ubicación física del bien; cpEstado contiene una leyenda que indica el estado físico que guarda el bien; y, cpPrestado que contiene un valor numérico que indica si el bien se encuentra con el estatus de prestado o no.

Ilustración IV-5. Estructura de la entidad tbInvTemp

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbInvTemp](
      [cpClave] [int] NOT NULL,
      [cpCategoría] [int] NULL,
      [cpSerie] [varchar](12) COLLATE Modern_Spanish_CI_AS NULL,
      [cpInventario] [varchar](12) COLLATE Modern_Spanish_CI_AS NULL,
      [cpDescripción] [varchar](255) COLLATE Modern_Spanish_CI_AS NULL,
      [cpUbicación] [int] NULL,
      [cpEstado] [varchar](30) COLLATE Modern_Spanish_CI_AS NULL,
      [cpPrestado] [int] NULL
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

Código 5. Creación de la Entidad tbInvTemp

IV.2.5 Entidad tbModulos

Esta entidad es sin duda la piedra fundamental del Sistema de Administración Electrónica (**SAEALABFI**) ya que en cada una de las tuplas se guarda la información de la situación que guarda cada uno de los módulos instalados y configurados en el Laboratorio. Al ser instalado el **SAEALABFI**, esta entidad estará vacía y, cuando se ingrese por vez primera al Sistema, se le solicitará al operador que configure los parámetros de inicio del Sistema y dentro de éstos se encuentra el número de módulos instalados en la *interfaz electrónica*.

De esta manera, habrá una tupla para cada módulo y podrá ser modificada mediante las opciones

de configuración y cambio de parámetros del Sistema.

Se debe aclarar que el módulo 0 del Sistema, se utiliza para servicios únicos para los usuarios, es decir, se encuentra en éste conectado el grabador de memorias y microcrontroladores, taladro de banco y medidor de inductancia. Para los módulos 1 al *n*, se encuentran asociados un osciloscopio, un multímetro digital, una fuente de poder y un generador de funciones.

Esta entidad está formada por los siguientes atributos: cpModulo que representa de manera numérica al módulo al que corresponde la tupla; cpEquipo1, cpEquipo2, cpEquipo3 y cpEquipo4 se utilizan para guardar la situación o estado que guarda cada uno de los instrumentos y herramientas indicadas anteriormente, de tal suerte que para el caso del módulo 0, los atributos corresponderán en orden a: Grabador de memorias, medidor de inductancia y taladro de banco, el atributo cpEquipo4 no se utiliza, para el caso de los módulos 1 a n, los atributos corresponden en orden a: Fuente, Generador de funciones, Multímetro y osciloscopio; el atributo cpUsuarios es de tipo numérico y sirve para guardar la cantidad de usuarios que se encuentra en ese momento en el módulo correspondiente a la tupla, este valor se actualizará cada vez que se agregue un usuario nuevo al módulo pero sin rebasar la cantidad máxima que ha sido configurada en los parámetros de inicio y configuración posterior; el atributo cpCronometro es de tipo numérico y lo utiliza el Sistema para guardar la cantidad de tiempo que se le ha ofrecido al o a los usuario(s) del módulo; los atributos cpCuenta1, cpUsuario1, cpCuenta2, cpUsuario2, cpCuenta3, cpUsuario3, cpCuenta4 y cpUsuario4 se utilizan para almacenar la información básica sobre los usuarios que se encuentran utilizando el módulo que corresponde a la tupla en uso, los atributos cpCuenta$_n$ almacenan el número de cuenta del usuario y, cpUsuario***n*** guarda el nombre del usuario asociado a su respectivo atributo cpCuenta, se ha tomado como límite hasta 4 pares cpCuenta-cpUsuario pues por reglamento no se admitirán más de 4 usuarios por módulo y la cantidad máxima que se podrá asociar al módulo dependerá de los valores establecidos en la opción de configuración inicial o subsecuente dentro de las opciones que han sido creadas para este fin dentro del Sistema; los atributos cpInventario1, cpInventario2, cpInventario3 y cpInventario4 son de tipo cadena y sirven para almacenar en forma de lista de códigos de inventario separados por comas a los bienes que se han prestado a los usuarios que de manera respectiva se asocian al número del atributo cpInventario con su respectivo par cpCuenta-cpUsuario; los atributos cpTrn1, cpTrn2, cpTrn3 y cpTrn4 son de tipo numérico y almacenan un número de transacción único que le ha sido asignado a cada una de las ternas de atributos cpCuenta-cpUsuario-cpInventario a partir de un control centralizado del detalle de las distintas operaciones internas que se pueden realizar dentro de la operación propia del **SAEALABFI**, al momento de liberación del **SAEALABFI** versión 1.2.0000.0, se han clasificado 72 tipos de transacciones distintas mismas que se pueden consultar con un poco más de detalle en el manual de referencia del **SAEALABFI**; el atributo cpTick es de tipo numérico y se utiliza para almacenar en él, la cantidad de minutos que restan para brindarle servicio al o a los ocupante(s) del módulo y por consiguiente el Sistema lo irá actualizando de acuerdo a un cronómetro central de servicio para los módulos; el atributo cpTrnx es de tipo numérico y se utiliza para almacenar un número de transacción que se genera cuando se ha prolongado el tiempo de servicio al módulo cuando éste ya haya sido agregado al panel de módulos próximos a vencer el tiempo de servicio en cualquier otro caso contendrá el valor de cero; finalmente, el atributo cpForzoso es de tipo numérico y es utilizado para indicar al Sistema que este módulo ha sido programado para que se encienda de manera automática cada vez que se inicialice el Sistema hasta que se cambie esta situación (estado).

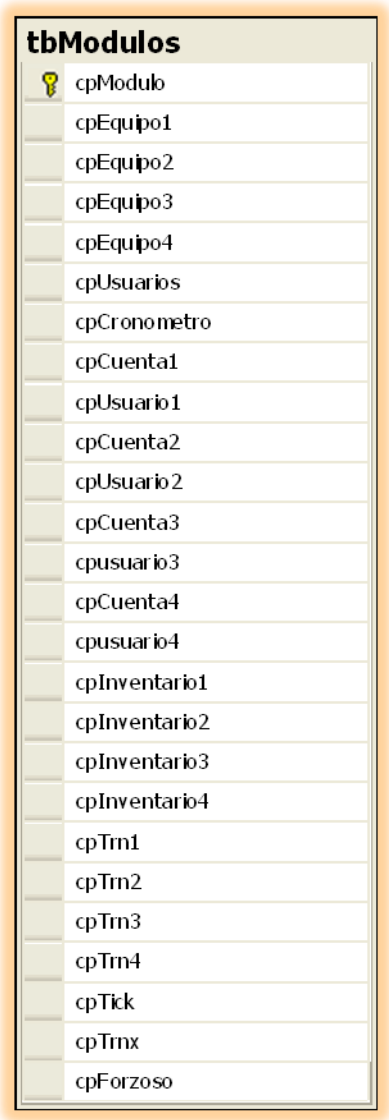


Ilustración IV-6. Estructura de la entidad tbModulos

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbModulos](
	[cpModulo] [int] NOT NULL,
	[cpEquipo1] [varchar](50) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpEqu__7F60ED59]  DEFAULT ('''),
	[cpEquipo2] [varchar](50) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpEqu__00551192]  DEFAULT ('''),
	[cpEquipo3] [varchar](50) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpEqu__014935CB]  DEFAULT ('''),
	[cpEquipo4] [varchar](50) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpEqu__023D5A04]  DEFAULT ('''),
	[cpUsuarios] [int] NOT NULL CONSTRAINT [DF__tbModulos__cpUsu__03317E3D]  DEFAULT ((0)),
	[cpCronometro] [datetime] NULL,
	[cpCuenta1] [varchar](9) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpCue__0519C6AF]  DEFAULT ('''),
	[cpUsuario1] [varchar](45) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpUsu__060DEAE8]  DEFAULT ('''),
	[cpCuenta2] [varchar](9) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpCue__07020F21]  DEFAULT ('''),
```

```
	[cpUsuario2] [varchar](45) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpUsu__07F6335A]  DEFAULT ('''),
	[cpCuenta3] [varchar](9) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpCue__08EA5793]  DEFAULT ('''),
	[cpusuario3] [varchar](45) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpusu__09DE7BCC]  DEFAULT ('''),
	[cpCuenta4] [varchar](9) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpCue__0AD2A005]  DEFAULT ('''),
	[cpusuario4] [varchar](45) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpusu__0BC6C43E]  DEFAULT ('''),
	[cpInventario1] [varchar](64) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpInv__0CBAE877]  DEFAULT ('''),
	[cpInventario2] [varchar](64) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpInv__0EA330E9]  DEFAULT ((0)),
	[cpInventario3] [varchar](64) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpInv__108B795B]  DEFAULT ((0)),
	[cpInventario4] [varchar](64) COLLATE Modern_Spanish_CI_AS NULL CONSTRAINT
[DF__tbModulos__cpInv__1273C1CD]  DEFAULT ((0)),
	[cpTrn1] [bigint] NULL CONSTRAINT [DF__tbModulos__cpTrn__145C0A3F]  DEFAULT ((0)),
	[cpTrn2] [bigint] NULL CONSTRAINT [DF__tbModulos__cpTrn__15502E78]  DEFAULT ((0)),
	[cpTrn3] [bigint] NULL CONSTRAINT [DF__tbModulos__cpTrn__164452B1]  DEFAULT ((0)),
	[cpTrn4] [bigint] NULL CONSTRAINT [DF__tbModulos__cpTrn__173876EA]  DEFAULT ((0)),
	[cpTick] [int] NULL,
	[cpTrnx] [bigint] NOT NULL CONSTRAINT [DF_tbModulos_cpTrnx]  DEFAULT ((0)),
	[cpForzoso] [int] NOT NULL CONSTRAINT [DF_tbModulos_cpForzoso]  DEFAULT ((0)),
 CONSTRAINT [PK__tbModulos__7E6CC920] PRIMARY KEY CLUSTERED
(
	[cpModulo] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

Código 6. Creación de la Entidad tbModulos

IV.2.6 Entidad tbMovimientos

La finalidad de esta entidad es la de registrar todos los movimientos que se realizan dentro del Sistema entre todos los paneles y los mismos módulos. Está compuesta por los siguientes atributos: primeramente se tiene al atributo cpMovimiento que es un valor numérico que se actualiza y autoincrementa con el propósito de tener movimientos de código de identificación único ya que utiliza una llave principal que permitirá establecer relaciones entre las distintas entidades de la Base de Datos; cpTrn es un atributo que contiene un número de transacción que se utiliza para registrar el tipo de movimiento que corresponde a la tupla en cuestión; cpPropietario es un atributo que registra el nombre de identificación del *Operador del Sistema* quien registró la operación o movimiento; cpCucuenta del usuario sobre el cual se está registrando el movimiento; cpFechaInicio es un atributo que registra la fecha y hora en que comenzó el movimiento asociado a la tupla; cpFechaFinal es un atributo que de manera similar al anterior registra una fecha y hora pero en este caso asociado al instante en que finalizó el movimiento que se ha registrado; cpModulo es un atributo de tipo numérico que contiene el número de módulo sobre el que se está registrando el movimiento; cpStatus es un atributo que indica si el movimiento se encuentra activo o no; cpMaterial es un atributo de tipo cadena Cuya finalidad es la de registrar una lista de materiales que pudiera tener en carácter de préstamo el usuario indicado en cpCuenta, la lista contiene en forma de lista delimitada por comas los distintos códigos de inventario que corresponde a los materiales; y finalmente, el atributo cpSemestre contiene el número del Semestre en el cual se está registrando el movimiento correspondiente a la tupla.

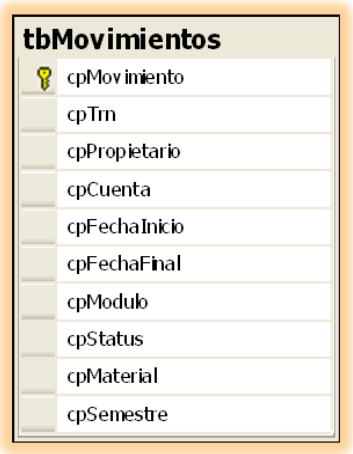


Ilustración IV-7. Estructura de la entidad tbMovimientos

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbMovimientos](
       [cpMovimiento] [bigint] IDENTITY(1,1) NOT NULL,
       [cpTrn] [int] NOT NULL,
       [cpPropietario] [varchar](20) COLLATE Modern_Spanish_CI_AS NULL,
       [cpCuenta] [varchar](9) COLLATE Modern_Spanish_CI_AS NULL,
       [cpFechaInicio] [datetime] NULL,
       [cpFechaFinal] [datetime] NULL,
       [cpModulo] [int] NULL,
       [cpStatus] [int] NULL,
       [cpMaterial] [varchar](126) COLLATE Modern_Spanish_CI_AS NULL,
       [cpSemestre] [varchar](6) COLLATE Modern_Spanish_CI_AS NULL,
 CONSTRAINT [PK__tbMovimientos__1DE57479] PRIMARY KEY CLUSTERED
(      [cpMovimiento] ASC )WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
```

Código 7. Creación de la Entidad tbMovimientos

IV.2.7 Entidad tbMovsModulos

La finalidad de esta entidad es la de registrar los movimientos que se registran dentro de la entidad Módulos y por consiguiente, en las operaciones que exclusivamente se registran en los servicios proporcionados y asociados a cualquiera de los módulos. Esta entidad se encuentra conformada por los siguientes atributos: cpTrnxInicial que representa un valor numérico de identificación de transacción único asociado a las tuplas de esta entidad; cpTrnxFinal es similar al atributo anterior con la diferencia en que se usa para registrar el número de transacción que se genera al terminar el registro del movimiento asociado a los módulos; cpModulo es un atributo que se utiliza para guardar el modulo sobre el cual se ha registrado el movimiento; cpPropietario es una cadena que guarda el nombre del operador quien se encuentra administrando al Sistema en el momento de registrar el movimiento; el atributo cpSemestre corresponde al número de semestre en el que se está registrando el movimiento; cpFechaInicial corresponde a la fecha y hora en que comenzó el registro del movimiento y el atributo cpFechaFinal es la hora y fecha en que se ha terminado de registrar el movimiento.

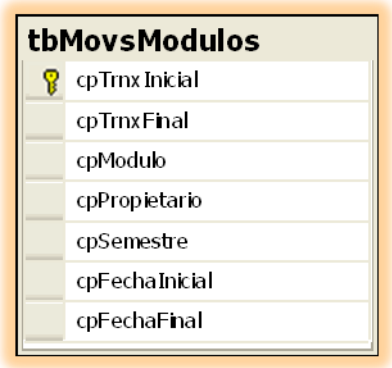


Ilustración IV-8. Estructura de la entidad tbMovsModulos

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbMovsModulos](
       [cpTrnxInicial] [bigint] NOT NULL CONSTRAINT [DF__MovsModul__cpTrn__40058253]  DEFAULT
((0)),
       [cpTrnxFinal] [bigint] NOT NULL CONSTRAINT [DF_tbMovsModulos_cpTrnxFinal]  DEFAULT
((0)),
       [cpModulo] [int] NULL CONSTRAINT [DF__MovsModul__cpMod__40F9A68C]  DEFAULT ((0)),
       [cpPropietario] [varchar](25) COLLATE Modern_Spanish_CI_AS NULL,
       [cpSemestre] [varchar](6) COLLATE Modern_Spanish_CI_AS NULL,
       [cpFechaInicial] [datetime] NULL,
       [cpFechaFinal] [datetime] NULL,
 CONSTRAINT [PK__MovsModulos__3F115E1A] PRIMARY KEY CLUSTERED
(
       [cpTrnxInicial] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

Código 8. Creación de la Entidad tbMovsModulos

IV.2.8 Entidad tbPréstamos

Esta entidad es la encargada de registrar el detalle de todos los préstamos de materiales, herramientas e instrumentos registrados en el inventario general a los usuarios de los servicios del Laboratorio Abierto. Esta conformado por los siguientes atributos: cpPréstamo que es un valor numérico único que se incrementa automáticamente en 1 y sirve como índice con la finalidad de generar registros únicos y que sirven para realizar vínculos o relaciones entre las entidades de la Base de Datos; el atributo cpTrnx es un atributo que se utiliza para registrar el número de transacción asociado al registro de préstamo de la tupla; el atributo cpVigencia se utiliza para indicar si el préstamo que describe la tupla se encuentra vigente o no; la entidad cpFuera se utiliza para indicar si el bien aún no ha sido entregado y por consiguiente se encuentra en el panel de materiales prestados; el atributo cpPropietario es el nombre o alias del operador o Administrador del Sistema quien realizó el préstamo que se registra en las tuplas de esta entidad; cpSemestre guarda el número de semestre en el que se registró el movimiento de préstamo; cpCuenta almacena el número de cuenta del usuario al que se le hizo el préstamo que con esta entidad se registra; cpNombre almacena el nombre completo asociado al atributo cpCuenta y cpMaterial es una lista

delimitada por comas de los números de inventario de cada uno de los bienes que se les ha prestado.

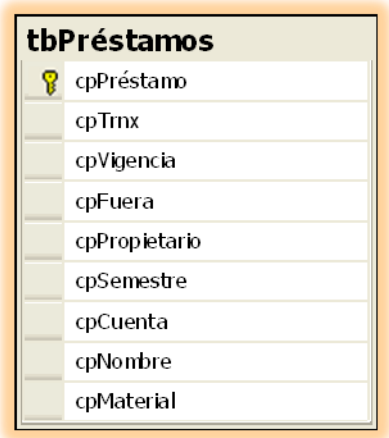


Ilustración IV-9. Estructura de la entidad tbPréstamos

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbPréstamos](
       [cpPréstamo] [bigint] IDENTITY(1,1) NOT NULL,
       [cpTrnx] [bigint] NOT NULL CONSTRAINT [DF__tbPréstam__cpTrn__4F47C5E3]  DEFAULT ((0)),
       [cpVigencia] [int] NOT NULL CONSTRAINT [DF_tbPréstamos_cpVigencia]  DEFAULT ((0)),
       [cpFuera] [int] NOT NULL CONSTRAINT [DF_tbPréstamos_cpFuera]  DEFAULT ((0)),
       [cpPropietario] [varchar](25) COLLATE Modern_Spanish_CI_AS NOT NULL CONSTRAINT
[DF__tbPréstam__cpPro__503BEA1C]  DEFAULT ('),
       [cpSemestre] [varchar](6) COLLATE Modern_Spanish_CI_AS NOT NULL CONSTRAINT
[DF__tbPréstam__cpSem__51300E55]  DEFAULT ('),
       [cpCuenta] [varchar](9) COLLATE Modern_Spanish_CI_AS NOT NULL CONSTRAINT
[DF__tbPréstam__cpCue__5224328E]  DEFAULT ('),
       [cpNombre] [varchar](50) COLLATE Modern_Spanish_CI_AS NOT NULL,
       [cpMaterial] [varchar](64) COLLATE Modern_Spanish_CI_AS NOT NULL CONSTRAINT
[DF__tbPréstam__cpInv__531856C7]  DEFAULT ((0)),
 CONSTRAINT [PK__tbPréstamos__4E53A1AA] PRIMARY KEY CLUSTERED
(      [cpPréstamo] ASC) WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]) ON
[PRIMARY]

GO
SET ANSI_PADDING OFF
```

Código 9. Creación de la Entidad tbPréstamos

IV.2.9 Entidad tbSanciones

Esta Entidad es la encargada de registrar las sanciones que se aplican a los usuarios que infringen el reglamento interno del Laboratorio y como ya se ha indicado con anterioridad, queda a criterio del Administrador del Sistema la duración y vigencia de las sanciones. Esta entidad está conformada por los siguientes atributos: cpNumSanción es un valor numérico único auto incrementable y sirve como llave principal de la entidad para establecer relaciones de ésta con las demás que conforman a la Base de Datos; el atributo cpTrn es de tipo numérico y sirve para guardar el número de transacción asociada a la operación de sanciones; el atributo cpUsuario contiene el número de cuenta del usuario al cual se está asociando la sanción para

la tupla que corresponde a la entidad que aquí se describe; el atributo cpPropietario contiene el nombre o alias del *Operador del Sistema* quien registró la sanción que corresponde a la tupla; el atributo cpFechaAplicación es de tipo fecha-hora y que registra la fecha a partir de cuando comenzará a aplicarse la restricción de servicio al usuario; el atributo cpFechaLiberación es de tipo fecha-hora y registra la fecha en la que la sanción dejará de tener vigencia; el atributo cpDuración es de tipo numérico y se utiliza para indicar la cantidad de días que dura la sanción descrita por la tupla; el atributo cpVigencia se utiliza para indicar al Sistema si el período establecido del rango de fechas y de duración de la sanción aún se encuentra vigente o no; y, finalmente, el atributo cpObservaciones es de tipo cadena en el que se le ofrece al *Operador del Sistema* detallar o anotar alguna observación en particular para la sanción que se está registrando mediante la tupla que corresponde en este caso.

Ilustración IV-10. Estructura de la entidad tbSanciones

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbSanciones](
       [cpNumSanción] [bigint] IDENTITY(1,1) NOT NULL,
       [cpTrn] [bigint] NOT NULL,
       [cpUsuario] [varchar](9) COLLATE Modern_Spanish_CI_AS NOT NULL,
       [cpPropietario] [varchar](25) COLLATE Modern_Spanish_CI_AS NOT NULL,
       [cpFechaAplicación] [datetime] NOT NULL,
       [cpFechaLiberación] [datetime] NOT NULL,
       [cpDuración] [int] NOT NULL,
       [cpVigencia] [smallint] NOT NULL,
       [cpObservaciones] [varchar](1024) COLLATE Modern_Spanish_CI_AS NULL,
 CONSTRAINT [PK__Sanciones__4CA06362] PRIMARY KEY CLUSTERED
(
       [cpNumSanción] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
```

Código 10. Creación de la Entidad tbSanciones

IV.2.10 Entidad tbSeguridad

La finalidad de esta entidad es la de registrar la información básica y esencial de cada uno de los operadores del Sistema que se encuentran registrados para poder hacer uso del **SAEALABFI.** Su contenido está protegido por algoritmos de encriptación y de código de redundancia cíclica (CRC), su estructura es la siguiente: atributo cpUID es de tipo alfanumérico y mantiene de manera comprimida y en formato hexadecimal el nombre de los usuarios registrados en el Sistema, por tratarse de usuarios únicos se utiliza a este atributo como la llave principal de la entidad para que no haya duplicidad de cuentas; el atributo cpPass es de tipo alfanumérico y contiene el código de redundancia cíclica en hexadecimal del password asociado a la cuenta de esta tupla; el atributo cpDerechos es de tipo alfanumérico y la finalidad es la de contener en forma hexadecimal la encriptación de una cadena que contiene los derechos a los que tiene acceso el usuario que corresponde a esta tupla y aunque se encuentra implementado en la entidad, aún no ha sido implementado en el Sistema por lo que en futuras liberaciones podría incluirse esta característica; el atributo cpNivel también es de tipo alfanumérico y guarda en formato hexadecimal la encriptación del tipo de usuario al que corresponde el usuario que describe la tupla; el atributo cpTrn es de tipo numérico y contiene el número de transacción que se registró en el Sistema en el momento en que se creó la cuenta; el atributo cpVigencia es de tipo alfanumérico en formato hexadecimal y contiene el estatus de vigencia de las cuentas de los operadores del Sistema y finalmente, el atributo cpVerificación es de tipo alfanumérico cuyo contenido se encuentra en hexadecimal de la encriptación de una copia de toda la tupa que el Sistema utiliza para verificar que la información no ha sido alterada externamente y si es así, intentar recuperar la información original de las cuentas de los operadores del Sistema.

Ilustración IV-11. Estructura de la entidad tbSeguridad

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbSeguridad](
       [cpUID] [varchar](64) COLLATE Modern_Spanish_CI_AS NOT NULL,
       [cpPass] [varchar](32) COLLATE Modern_Spanish_CI_AS NOT NULL,
       [cpDerechos] [varchar](64) COLLATE Modern_Spanish_CI_AS NOT NULL,
       [cpNivel] [varchar](32) COLLATE Modern_Spanish_CI_AS NOT NULL,
       [cpTrn] [bigint] NOT NULL,
       [cpVigencia] [varchar](32) COLLATE Modern_Spanish_CI_AS NULL,
       [cpVerificación] [varchar](512) COLLATE Modern_Spanish_CI_AS NULL,
 CONSTRAINT [PK__tbSeguridad__403A8C7D] PRIMARY KEY CLUSTERED ([cpUID] ASC) WITH (PAD_INDEX  =
OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
```

Código 11. Creación de la Entidad tbSeguridad

IV.2.11 Entidad tbSysParms

La finalidad de esta entidad es la de mantener un diccionario de datos en el cual se guardan los parámetros esenciales de la operación del Sistema dentro de estos datos se encuentran las horas por sesión, el número de módulos instalados, las descripciones de las transacciones y muchos valores más, por ser de suma importancia para el Sistema, el contenido de cada uno de los atributos se encuentra en formato hexadecimal aplicado al resultado de un proceso de encriptación de los valores que corresponden a la tupa. Los atributos de esta entidad son: cpClave su formato es de tipo alfanumérico y se utiliza para establecer el grupo de parámetros según corresponda como por ejemplo "Parámetros de la ventana principal del Sistema"; el atributo cpClave es de tipo alfanumérico y se utiliza para establecer una clave única dentro de la Clase establecida en el atributo cpClase, en conjunto el atributo cpClase y el atributo cpValor se utilizan para formar una llave de acceso primaria que hacen irrepetible el par Clase-Valor; y finalmente, se tiene al atributo cpValor que es de tipo alfanumérico y se utiliza para guardar aquí el valor que corresponde a la combinación conjunta de Clase-Valor.

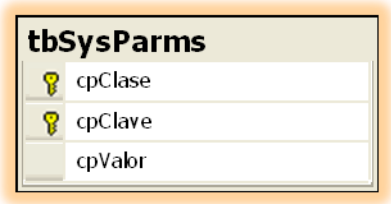


Ilustración IV-12. Estructura de la entidad tbSysParms

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbSysParms](
       [cpClase] [varchar](37)   COLLATE Modern_Spanish_CI_AS NOT NULL,
       [cpClave] [varchar](50)   COLLATE Modern_Spanish_CI_AS NOT NULL,
       [cpValor] [varchar](255)  COLLATE Modern_Spanish_CI_AS NULL,
 CONSTRAINT [PK_tbSysParms] PRIMARY KEY CLUSTERED ([cpClase] ASC,  [cpClave] ASC) WITH
(PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
```

Código 12. Creación de la Entidad tbSysParms

IV.2.12 Entidad tbSysTemp

La finalidad de esta entidad es la servir como auxiliar en el proceso de creación de la entidad tbSysParms durante el proceso de instalación y configuración por primera vez del Sistema. Los atributos que conforman a esta entidad, son: cpSección, cpClave y cpValor y corresponden en orden a los atributos que ya se han descrito para el caso de la entidad tbSysParms mostrado en la entidad anterior.

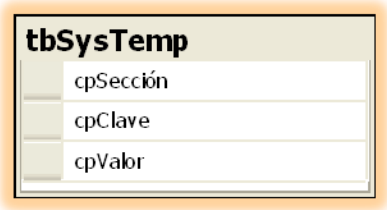


Ilustración IV-13. Estructura de la entidad tbSysTemp

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbSysTemp](
      [cpSección] [varchar](64) COLLATE Modern_Spanish_CI_AS NULL,
      [cpClave] [varchar](64) COLLATE Modern_Spanish_CI_AS NULL,
      [cpValor] [varchar](128) COLLATE Modern_Spanish_CI_AS NULL) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
```

Código 13. Creación de la Entidad tbSysTemp

IV.2.13 Entidad tbTransacción

Esta entidad es la encargada de registrar el pormenor de de todas las operaciones intrínsecas que se realizan durante la operación y uso del **SAEALABFI** y como ya se ha indicado, al momento de la liberación de la versión 1.2.0000.0, se han clasificado 72 distintas y se pueden consultar para mayor detalle en el Apéndice B del manual de referencia del **SAEALABFI.** Los atributos que conforman a esta entidad son: cpTrn es de tipo numérico cuyo valor es único, se autoincrementa y es la llave principal de la entidad con la finalidad de que garantizar que no haya transacción alguna que se pueda duplicar además de servir para establecer las relaciones entre las distintas entidades de la Base de Datos; el atributo cpFecha almacena la fecha y hora en que se ha registrado el inicio o termino de la transacción según corresponda al caso; el atributo cpClase se utiliza para establecer el tipo de transacción que corresponde a la tupla (cualquiera de las 72 transacciones definidas); el atributo cpOperador contiene el nombre o alias del *Operador del Sistema* quien durante la operación del mismo va registrando los distintos movimientos intrínsecos del **SAEALABFI**; y finalmente, el atributo cpMarcadores es de tipo alfanumérico cuyo contenido es el de establecer aquí los parámetros asociados a cada clase o tipo de transacción. Esta entidad se encuentra administrada por la clase-objeto; clsTransacción, misma que se encarga de que los valores correctos de los marcadores se establezcan para el tipo de transacción que se registra en las tuplas de la entidad.

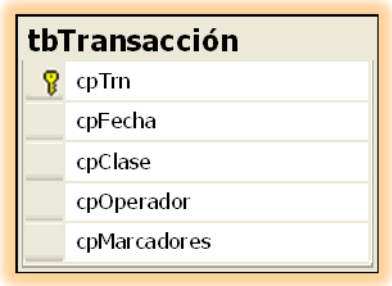


Ilustración IV-14. Estructura de la entidad tbTransacción

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbTransacción](
             [cpTrn] [bigint] IDENTITY(1,1) NOT NULL,
             [cpFecha] [datetime] NULL,
             [cpClase] [int] NULL,
             [cpOperador] [varchar](12) COLLATE Modern_Spanish_CI_AS NULL,
             [cpMarcadores] [varchar](64) COLLATE Modern_Spanish_CI_AS NULL,
 CONSTRAINT [PK_tbTransacción] PRIMARY KEY CLUSTERED
(      [cpTrn] ASC )WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY] ) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
```

Código 14. Creación de la Entidad tbTransacción

IV.2.14 Entidad tbUsuarios

La finalidad de esta entidad es la de registrar la información de cada uno de los usuarios que tienen derecho a solicitar los servicios que brinda el Laboratorio Abierto de la Facultad de Ingeniería, y como ya se ha indicado es necesario ser registrado primeramente para que se pueda dar atención oportuna y equitativa a los usuarios del Laboratorio. Los atributos de esta entidad son: cpCuenta su contenido es numérico y se usa para guardar el número de cuenta del usuario, por tratarse de el distintivo principal entre los usuarios se ha establecido como llave principal para asegurar que no haya duplicidad de tuplas y para que se pueda relacionar esta entidad con las demás que conforman a la Base de Datos; el atributo cpNombre es de tipo alfanumérico y se usa para guardar el nombre del usuario descrito por la tupla pero sin apellidos; el atributo cpApPaterno es de tipo alfanumérico y se utiliza para guardar en él, el apellido paterno del usuario; el atributo cpApMaterno es similar al anterior pero se utiliza para guardar el apellido materno; el atributo cpCalle es de tipo alfanumérico y se utiliza para guardar el nombre de la calle correspondiente al domicilio del usuario; el atributo cpNumero es de tipo alfanumerico y se utiliza para guardar el número del domicilio del usuario, se ha elegido alfanumerico para poder guardar los domicilios cuya distinción es Manzana y Lote; el atributo cpColonia es de tipo alfanumérico y se utiliza para guardar la colonia del domicilio del usuario; el atributo cpDelegación es de tipo alfanumérico y se utiliza para guardar la delegación o municipio en la que se encuentra ubicado el domicilio del usuario; el atributo cpCP es de tipo numérico y se utiliza para guardar el código postal del domicilio; el atributo cpCarrera es de tipo alfanumérico y se utiliza para registrar la carrera que estudia el usuario; el atributo cpGeneración es de tipo alfanumérico y se utiliza para guardar la Generación a la que pertenece el usuario o la fecha de expedición de la credencial con la que se identifica; el atributo cpTeléfono es de tipo alfanumérico y es

utilizado para almacenar el número de teléfono del usuario: el atributo cpPropietario es de tipo alfanumérico y se utiliza para almacenar el nombre o alias del operador que registró la información del usuario; el atributo cpFecha es de tipo fecha-hora y se utiliza para registrar la fecha y hora en que se registró al usuario en el Sistema; el atributo cpTrn es de tipo numérico y se utiliza para registrar la transacción en la que se generó la tupla del usuario; el atributo cpModificación es de tipo fecha-hora y se utiliza para registrar la última fecha de modificación de la información del usuario, cuando se crea la tupla, la fecha de modificación será la misma que la de creación y finalmente, el atributo cpTrnMod es de tipo numérico y corresponde al número de transacción asociado al proceso de actualización de la información del usuario.

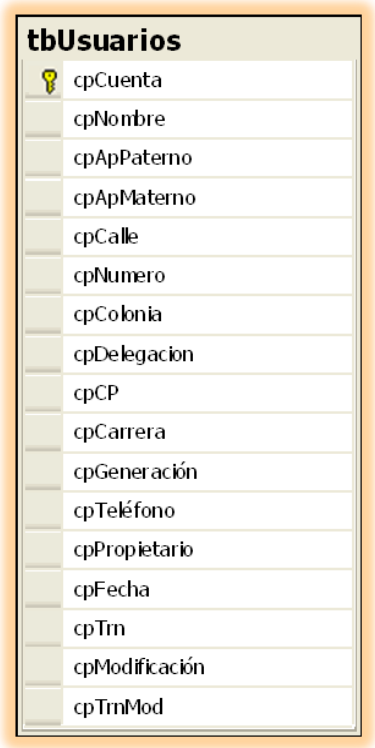


Ilustración IV-15. Estructura de la entidad tbUsuarios

```
USE [SAEALAB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbUsuarios](
	[cpCuenta] [varchar](9) COLLATE Modern_Spanish_CI_AS NOT NULL,
	[cpNombre] [varchar](25) COLLATE Modern_Spanish_CI_AS NULL,
	[cpApPaterno] [varchar](20) COLLATE Modern_Spanish_CI_AS NULL,
	[cpApMaterno] [varchar](20) COLLATE Modern_Spanish_CI_AS NULL,
	[cpCalle] [varchar](30) COLLATE Modern_Spanish_CI_AS NULL,
	[cpNumero] [varchar](10) COLLATE Modern_Spanish_CI_AS NULL,
	[cpColonia] [varchar](30) COLLATE Modern_Spanish_CI_AS NULL,
	[cpDelegacion] [varchar](30) COLLATE Modern_Spanish_CI_AS NULL,
	[cpCP] [varchar](6) COLLATE Modern_Spanish_CI_AS NULL,
	[cpCarrera] [varchar](20) COLLATE Modern_Spanish_CI_AS NULL,
	[cpGeneración] [varchar](20) COLLATE Modern_Spanish_CI_AS NULL,
	[cpTeléfono] [varchar](20) COLLATE Modern_Spanish_CI_AS NULL,
	[cpPropietario] [varchar](20) COLLATE Modern_Spanish_CI_AS NULL,
	[cpFecha] [datetime] NULL,
	[cpTrn] [bigint] NULL,
	[cpModificación] [datetime] NOT NULL,
	[cpTrnMod] [bigint] NULL,
 CONSTRAINT [PK__tbUsuarios__1920BF5C] PRIMARY KEY CLUSTERED
```

```
(      [cpCuenta] ASC)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY],
 CONSTRAINT [UQ__tbUsuarios__1A14E395] UNIQUE NONCLUSTERED
(      [cpCuenta] ASC)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

SET ANSI_PADDING OFF
```

Código 15. Creación de la Entidad tbUsuarios

IV.3 Descripción del Software

Ahora toca el turno de echar un vistazo a la estructura del desarrollo del Software de aplicación para este proyecto, para ello se describirá la manera general en que está orientada dicha estructura y, con el apoyo de los conceptos específicos para este tema se describirá de breve gráfica y rápida a cada uno de los componentes del proyecto.

Primero se comienza por describir los componentes principales del **SAEALABFI**. Este Sistema está construido en un ambiente 100% gráfico, aprovechando al máximo los elementos y componentes que el entorno gráfico (GDI) del Sistema Operativo Windows 2000, XP y posteriores proporcionan para sacar el máximo provecho y rendimiento de las aplicaciones que son desarrolladas para este entorno.

Es necesario considerar que dentro de la operatividad de Windows, se emplea en todo momento una tecnología de multiproceso, basándose en las características que los microprocesadores de estos tiempos ofrecen. De estas características técnicas y operativas de los elementos de Hardware y Software, el multiproceso se encuentra basado en un núcleo o *kernel* que permite dar seguimiento y ejecución a las aplicaciones de todos los niveles dentro de los equipos personales, que es el caso que se describe.

Dentro de toda la gama de aplicaciones que se ejecutan en primero y segundo plano, se pueden encontrar las aplicaciones orientadas al control del hardware y que comúnmente se denominan drivers. Por otra parte, existen aplicaciones que se encuentran ejecutándose en un esquema denominado **Servicios** que son aplicaciones que apoyan a otras aplicaciones que no requieren de la presencia gráfica y que ayudan en el proceso de interconexión, establecimiento de protocolos e incluso para la inter-conectividad entre una aplicación cliente y una aplicación servidor, o más comúnmente en aplicaciones distribuidas por internet.

El Sistema Operativo se encuentra diseñado para que administre las operaciones o aplicaciones que se ejecutan utilizando los recursos de hardware con que está configurado el equipo de cómputo asignando prioridades a las aplicaciones según un nivel de confianza con que se administran cada uno de los recursos.

Complementariamente, el Sistema Operativo está construido para que se pueda comunicar con las distintas aplicaciones de las dos jerarquías mencionadas y de las aplicaciones que son visibles y palpables por los usuarios de los equipos de cómputo utilizando un entorno, orientado a eventos y mediante el software de desarrollo se puede acceder a las características del Sistema Operativo empleando una tecnología de programación que se ha denominado **"Orientada a Objetos"** (OOP).

La operación orientada a eventos, es una tecnología en la cual todos los sucesos que ocurren en el hardware o software se pueden enviar a cada una de las aplicaciones con la finalidad de que éstas determine sí dicho evento ha ocurrido en alguno de los componentes de las

aplicaciones en particular.

Un ejemplo de este concepto es el hecho de que cuando se mueve el ratón sobre el escritorio o el área de trabajo de alguna aplicación, el Sistema Operativo se encuentra recibiendo constantemente los mensajes que el dispositivo envía al Sistema Operativo que mediante protocolos ya establecidos las aplicaciones puedan reconocer la procedencia de la ocurrencia de dichos eventos y de esta manera se encuentre en capacidad de ejecutar el código necesario para responder a dicho evento.

Otros ejemplos de este concepto son el uso del teclado, cuando se oprime una tecla cualquiera, se están enviando mensajes de eventos al Sistema Operativo. Cuando se encuentra rellenando un formulario o cuadro de dialogo de manera particular en una aplicación, se puede determinar una gran variedad de eventos que se encuentran inter-construidos con los componentes del Sistema Operativo.

Los Sistemas Operativos de Microsoft Windows han cobrado popularidad primeramente por una estrategia de mercado fuerte, sin embargo en la actualidad existen distintas distribuciones "gratuitas" del Sistema Operativo Linux que de manera análoga ofrece una gama de aplicaciones de distinta índole y para un entorno gráfico que son gratuitas, el único inconveniente, es que no todos los usuarios de los equipos tienen conocimientos y habilidades en la forma de operar con éste Sistema Operativo.

Windows ha tomado su nombre por la analogía que tiene con una ventana y esta analogía es un ejemplo claro de la abstracción que de un problema u objeto real y palpable, se puede extraer para construir y modelar dicha situación. Esta idea es la que la Programación Orientada a Objetos retoma.

La Programación Orientada a Objetos es una técnica mediante la cual se puede definir a partir de un objeto real, un modelo conceptual de ese objeto, estructurándolo en las distintas herramientas de los lenguajes de programación en donde se aplican estos conceptos. El modelo del objeto en conjunto se realiza mediante la creación de ***Clases***, estas clases se distinguen por poseer tres tipos de componentes: **Propiedades, Métodos** y **Eventos**.

Las **Propiedades**, son aquéllas características propias que definen la estructura y la forma de los objetos, por ejemplo, el color del objeto, el tamaño, tipo de material, el peso, textura, etc.

Los **Métodos** corresponden a las acciones que se realizan ante alguna situación, un ejemplo de esto podría ser un método que se utiliza para simular la mordida de un "perro" o "levantar una grúa", etc.

Los **Eventos** son un tipo de métodos especiales que se encargan de responder a las eventualidades del Hardware o del Software que han sido establecidos para el comportamiento en particular de alguno de los componentes de la clase. Un ejemplo de esta situación puede ser el movimiento del mouse, un mensaje de algún cronómetro, el aviso de llegada de datos desde algún puerto de comunicación, etc.

El modelo conceptualizado de la clase es la base para la construcción de estructuras reutilizables y que sólo son el "plano" de lo que representa. Para que este plano sea palpable, hay que construirlo, dicha construcción es el objeto como tal ajustado a las especificaciones que se han definido en el plano. De manera simple, podemos decir que el objeto es una variable de tipo clase.

Se ha dicho que Windows está basado en esta tecnología y por ello todos los componentes que constituyen al Sistema Operativo son objetos. De estos objetos derivan otros que se han utilizado para construir objetos de tipo específico pero que en esencia conservan las propiedades de los objetos generales. A estos objetos se les llaman **Controles**, estos son los componentes básicos de los que se puede echar mano para la construcción de las aplicaciones personalizadas de los usuarios.

En las generaciones de los compiladores de estos tiempos, existe una gran variedad de **controles** con la finalidad de hacer más rápido y ágil el proceso de desarrollo de Sistemas con la finalidad de incrementar la productividad. Estos controles son: el apuntador de Windows, un botón, una caja de verificación, una caja de lista seleccionable, una caja combo, un selector de fechas, una etiqueta, una etiqueta enlazable tipo hipervínculo, una caja de lista, una vista de lista, una caja de texto enmascarable, un calendario por mes, una icono de notificación, una lista desplazable numérica, una caja de imágenes, una barra de progreso (avance), un botón de radio, una caja de texto enriquecido, una caja de texto normal, un herramienta para tips rápidos, una vista de árbol, un navegador de internet, distintos tipos contenedores de controles básicos, controles para el manejo de menús, controles para el manejo y acceso a datos, controles para acceso al directorio de disco de los equipos de cómputo, manejo de errores, manejo de eventos, componentes para la implementación de Sistemas multi-hebra, proveedor de ayudas, contenedor de imágenes, puerto serial, cronómetros, procesos del Sistema, controles para la impresión, controles para activar el cuadro de dialogo de selección de color, control para la navegación de archivos, control para el cuadro de dialogo de selección de fonts (fuentes), cuadro de dialogo para la apertura de archivos, cuadro de dialogo para guardar archivos y un control para el diseño básico de informes.

Todos los controles se ofrecen como herramientas desde las cuales se puede basar y hacer uso para construir aplicaciones más complejas y también se pueden utilizar para construir controles personalizados.

La Programación Orientada a Objetos se caracteriza por observar tres características básicas: **el polimorfismo**, **la herencia** y **el encapsulamiento**. Estas características permiten que para el primer caso, se pueda utilizar una estructura similar entre distintas clases sin que sean alteradas permitiendo una independencia entre todas ellas; la herencia es una característica en la que las clases se pueden utilizar como base para construir otras, trayendo consigo a toda su estructura y comportamiento ante eventos o métodos específicos y, el encapsulamiento es la característica mediante las clases ofrecen una medida de protección a sus componentes conduciendo a que el acceso a sus miembros de la clase sean accesados solo a través de sus propiedades o métodos.

Dicho lo anterior, es necesario recordar que en los modelos de desarrollo de aplicaciones que se habían desarrollado hasta los Sistemas Operativo Windows 98, 98 SE, NT O ME estaban basados en un modelo de tipo “apartamento” o “servidor” que para poder acceder a las características propias del núcleo del Sistema Operativo, se requería tener un gran conocimiento de la estructura del Sistema y del MFC (Microsoft Fundation Class) que es el conjunto de clases que constituyen el acceso directo a los recursos del Sistema Operativo y se podían accesar mediante la API (Application Programmable Interfaz).

El lenguaje que se encontraba construido para estar a la par con estos conceptos era el Visual C++, su estructura es algo compleja de entender y aunado a que se requiere conocer a fondo la estructura del Sistema Operativo y la ubicación de las clases, implicaba una productividad reducida aumentando los costos de desarrollo y por consiguiente, el tiempo de mantenimiento y ajuste de las aplicaciones.

En otros lenguajes como el Visual Basic, se podía realizar el acceso a estas características mediante la API, pero no se podía realizar un aprovechamiento óptimo de esta pues sólo se mantenía acceso a los métodos como si se tratase de una función o procedimiento común.

Concomitantemente con esta situación, las versiones de Visual Basic para estas generaciones de entornos ya consideraba la Programación Orientada a Objetos pero su extensibilidad no podía ser considerada como la conocemos en la actualidad; ya poseía la capacidad de crear clases pero éstas no consideraban los detalles de la Programación Orientada a Objetos, el control de excepciones era pobre y el encapsulamiento resultaba pobre.

Con este modelo de aplicaciones y con el inicio del Sistema Operativo, se crearon las bibliotecas de enlace dinámico (DLL) en las que se incluyen todos los componentes necesarios reutilizables para las aplicaciones. Dentro de estas bibliotecas, existe por igual la capacidad de incluir objetos gráficos, de sonido, de cadenas, de definición de diálogos, de funciones y subrutinas para todo tipo de situaciones. Se dicen de enlace dinámico porque al ejecutarse una aplicación, el núcleo del Sistema Operativo verifica la procedencia y relación que existe entre la aplicación y otras bibliotecas o entre bibliotecas para que de ellas se extraigan los elementos que requiere para operar de acuerdo a lo previsto.

Otro elemento fundamental en estos entornos es el mecanismo de acceso a Bases de Datos o aplicaciones que permiten el uso de datos con estructura tabular basada en el modelo entidad-relación como ya se ha descrito en el Capítulo III. Los mecanismos de acceso a dichas fuentes de datos se realizaba mediante una tecnología denominada ADO (Access Data Object) que se basó en un modelo ODBC (Open Data Base Conectivity) que servía para que cada fabricante de software pudiera implementar sus productos siguiendo esta norma para asegurar que las aplicaciones que hicieran uso de estos productos pudieran ser utilizados en el ambiente gráfico de Microsoft Windows.

Ahora bien, en los entornos de desarrollo de Visual .NET en cualquiera de sus versiones, las cosas han cambiado. El mecanismo de la Programación Orientada a Objetos se ha enriquecido y mejorado, dotando al lenguaje de programación Visual Basic .NET con todos los elementos de la OOP (Programación Orientada a Objetos), manejando de manera eficaz a los eventos y un sinfín de características que se consideran en los lenguajes que tradicionalmente era exclusivo de estos (Visual C++, por ejemplo). Las tareas que anteriormente se requerían programar conociendo a fondo la estructura del Sistema Operativo, ahora se han concentrado en una colección de clases y objetos accesibles para todas las aplicaciones que lo requieran sin tener que centrarse en las tareas de diseño y desarrollo. Como ejemplo de esto, consideremos la programación multihebra, la programación de servicios y una gran gama de clases que ayudan a explotar al máximo a la *interfaz* gráfica.

Todas estas clases se han agrupado en un concepto llamado **espacios de nombres** que ahora es la estructura en donde se agrupan las distintas clases orientadas a un fin determinado, además de estas clases se incluyen los conceptos de Interfaz, de herencia de clases, de clases primarias y de todos los componentes que el desarrollador necesita para el desempeño de sus tareas.

Todos estos componentes se agrupan en bibliotecas que aparentemente son las mismas que en las generaciones anteriores de DLL, que ahora se llaman **ensamblados**, los desarrolladores ahora cuentan también con la posibilidad de estructurar de esta manera a las clases propias.

Los lenguajes de programación de las generaciones .NET han sido concebidos para poder construir aplicaciones de tipo distribuidas a través de internet o intranets o de la manera tradicional, en aplicaciones de escritorio. Se basan en ese conjunto de clases que se han

clasificado en espacios de nombres incluidos en la tecnología intermedia denominada .NET FrameWork.

El .NET FrameWork incluye en su estructura un mecanismo que sirve de puente entre las aplicaciones de los desarrolladores y la API del Windows. Considera además una forma nueva de administrar el acceso a Bases de Datos utilizando el concepto de ADO.NET, posee tres componentes de especial atención: MSIL, CLR y ASPX. El MSIL o Microsoft Intermediate Language, constituye un esfuerzo de implementación y compatibilidad entre las aplicaciones que se desarrollan bajo este esquema y que tiene como finalidad de que cualquiera de los lenguajes que se han construido bajo el entorno de FrameWork generen código objeto siguiendo las especificaciones del MSIL que a fin de cuentas es un código que al ser ejecutado en realidad se encuentra en proceso de interpretación por la plataforma .NET Framework. El CLR o Common Language Runtime es la estructura que se encarga de definir los estándares de compatibilidad de los distintos tipos de clases de tipo primitivas que serán de iguales características entre los distintos lenguajes por ejemplo, el tipo de datos **long** en Visual Basic, tiene su equivalente con la misma estructura, propiedades y métodos que **int** en Visual C#. ASPX o Active Server Page eXtended es el mecanismo que Microsoft ha desarrollado para que mediante sus productos de desarrollo se puedan construir aplicaciones dinámicas de manera distribuidas entre equipos en una intranet o en internet.

Considerando los conceptos aquí descritos se comenzara la descripción general de la estructura de la aplicación que se ha desarrollado como la solución a la problemática encontrada en el Laboratorio Abierto de la Facultad de Ingeniería.

Primero hay que considerar que esta aplicación está basada en 66 archivos de código de los cuales 6 corresponden a código de clases propias que son utilizadas en todo momento por los componentes de la aplicación; 52 archivos corresponden con clases de formularios o cuadros de dialogo mediante los cuales se realizan todas las tareas específicas y visibles para los Operadores del Sistema que a partir de este Capítulo hará referencia a los Administradores del Laboratorio; y, finalmente 8 archivos de módulo cuya finalidad es la de agrupar métodos orientados a apoyar a las distintas tareas del Sistema.

El código fuente de toda la aplicación se encuentra distribuida en **19,684** líneas que si se quisiera incluir en este trabajo, estaríamos en la necesidad de incluir alrededor de 390 páginas más y este no es el caso. En cambio, se ofrece una tabla de dependencias de todas las clases incluyendo las de formulario y módulos para poder ver la relación que guardan todas ellas, un esquema general de los componentes generales de la aplicación y después de ello se abordan a cada una de las clases y módulos, indicando el nombre del archivo de código y el número de líneas que lo componen.

Después de esto aparecerá un cuadro o diagrama de clase en la que se muestra en forma de lista dos grupos de componentes. Estos se encuentran bajo el concepto de **Campos** y **Métodos**. Los campos incluyen a todas las propiedades y variables locales dentro de la clase a la que pertenece, algunos componentes aparecen con el nombre en un color gris, esto significa que dichos componentes son protegidos y que por consiguiente no son accesibles de manera directa desde fuera de la clase-objeto, los componentes que no aparecen en gris son elementos que se pueden acceder de manera directa mediante una calificación de objeto-método, objeto-propiedad u objeto-campo.

En las secciones de **Métodos** aparecen de manera similar al caso anterior los identificadores de métodos que se encuentran listados en dos tonos, los que aparecen en tono gris son aquéllos que son transparentes o protegidos y que generalmente constituyen los métodos

asociados a los objetos que son creados con el uso de los controles del Sistema; los que aparecen en color negro son aquéllos métodos que son creados con las estructuras de control y definición de clases-controles de los usuarios y diseñadores de software.

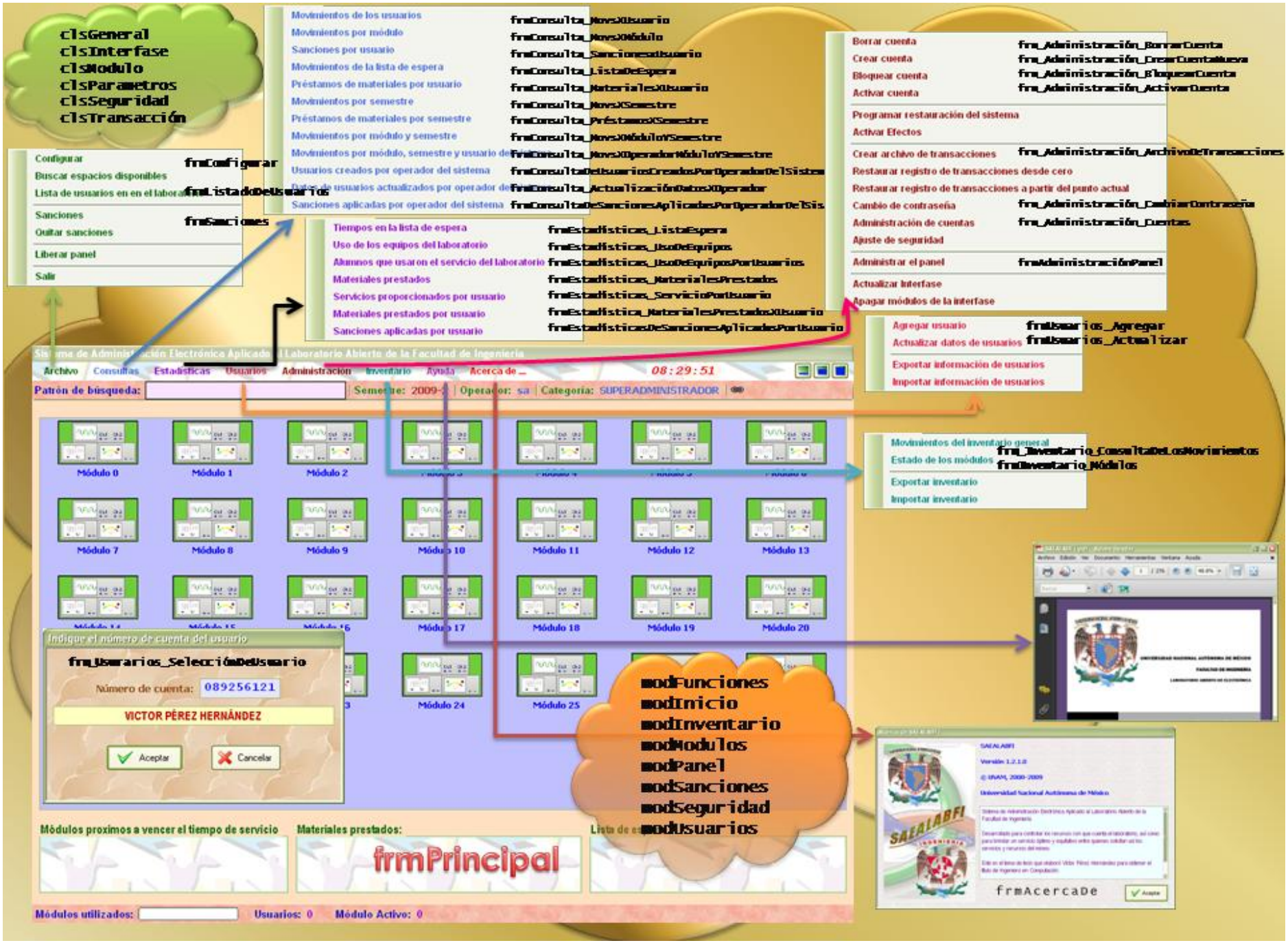
En esta sección destacan aquéllos métodos que corresponden a los eventos que como ya se ha dicho, son acciones que suceden y surgen ante un evento específico. Estos métodos de evento se pueden distinguir rápidamente porque en el identificador del objeto aparece seguido por un guión bajo y el nombre del evento. Un caso de este tipo podría ser **frmAcercaDe_Load** u **OKButton_Click** entre muchos ejemplos de estos.

Los eventos siempre están asociados a la forma intrínseca de implementación de los controles de Windows que ya se indicó, pero el diseñador o programador de soluciones cuenta con la amplia libertad de implementar sus propios eventos en sus aplicaciones, controles o clases que de manera particular desempeñen tareas que no han sido implementadas o que partiendo de las ya construidas, se puede apoyar en el concepto de herencia para mejorarlas o ampliarlas.

Como se verá en cada descripción de las clases y módulos, al igual que en la descripción de las estructuras de la Base de Datos y de sus entidades y atributos, también se hace uso de la notación húngara para describir en forma genérica el tipo al que corresponde cada uno de los componentes de la aplicación.

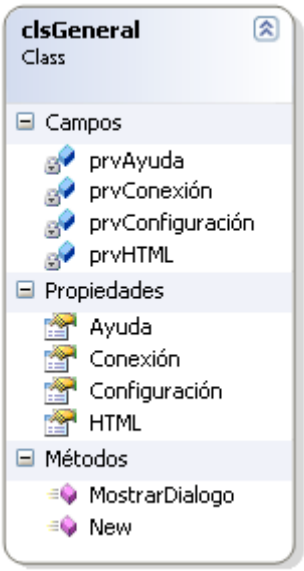
Así, los identificadores de clase comienzan con **cls**, los de formularios con **frm** y los de módulos con **mod**. En los campos ya sea como variables o como propiedades, se utiliza esta notación para indicar el tipo de valor que contiene o el alcance dentro de las clases; un ejemplo de esto es el uso de **int** para identificar a campos de tipo entero, **bol** para campos de tipo booleano, **str** para campos de tipo cadena, **lng** para campos de tipo long, etc. En cuanto al alcance de los campos se emplea **prv** para alcance privado o de alcance acotado al entorno de la clase que pertenece y de manera similar así se podrán interpretar cada uno de los componentes que se describen en cada una de las definiciones de las clases y módulos de este proyecto.

En las siguientes páginas se incluye la tabla de dependencias entre clases y el diagrama general de la aplicación; después de estas se muestra la descripción de cada clase y módulo siguiendo el criterio ya indicado.


clsGeneral
clsInterfase
clsModulo
clsParametros
clsSeguridad
clsTransacción
Movimientos de los usuarios fmConsulta_MovsXUsuario
Movimientos por módulo fmConsulta_MovsXModulo
Sanciones por usuario fmConsulta_SancionesXUsuario
Movimientos de la lista de espera fmConsulta_ListaDeEspera
Préstamos de materiales por usuario fmConsulta_MaterialesXUsuario
Movimientos por semestre fmConsulta_MovsXSemestre
Préstamos de materiales por semestre fmConsulta_PrestamosXSemestre
Movimientos por módulo y semestre fmConsulta_MovsXModuloYSemestre
Usuarios creados por operador del sistema
Sanciones aplicadas por operador del sistema
Configurar fmConfigurar
Buscar espacios disponibles
Lista de usuarios en en el laborat fmListadoDeUsuarios
Sanciones fmSanciones
Quitar sanciones
Liberar panel
Salir
Tiempos en la lista de espera fmEstadisticas_ListaEspera
Uso de los equipos del laboratorio fmEstadisticas_UsoDeEquipos
Alumnos que usaron el servicio del laboratorio fmEstadisticas_UsoDeEquiposPorUsuarios
Materiales prestados fmEstadisticas_MaterialesPrestados
Servicios proporcionados por usuario fmEstadisticas_ServicioPorUsuario
Materiales prestados por usuario fmEstadistica_MaterialesPrestadosXUsuario
Sanciones aplicadas por usuario fmEstadisticasDeSancionesAplicadasPorUsuario
Borrar cuenta fmAdministración_BorrarCuenta
Crear cuenta fmAdministración_CrearCuentaNueva
Bloquear cuenta fmAdministración_BloquearCuenta
Activar cuenta fmAdministración_ActivarCuenta
Programar restauración del sistema
Activar Efectos
Crear archivo de transacciones fmAdministración_ArchivoDeTransacciones
Restaurar registro de transacciones desde cero
Restaurar registro de transacciones a partir del punto actual
Cambio de contraseña fmAdministración_CambiarContraseña
Administración de cuentas fmAdministración_Cuentas
Ajuste de seguridad
Administrar el panel fmAdministraciónPanel
Actualizar interfase
Apagar módulos de la interfase
Agregar usuario fmUsuarios_Agregar
Actualizar datos de usuarios fmUsuarios_Actualizar
Exportar información de usuarios
Importar información de usuarios
Archivo Consultas Estadísticas Usuarios Administración Inventario Ayuda Acerca de ...
08:29:51
Patrón de búsqueda:
Semestre: 2009-2 Operador: sa Categoría: SUPERADMINISTRADOR
Módulo 0 Módulo 1 Módulo 2 Módulo 7 Módulo 8 Módulo 9 Módulo 10 Módulo 11 Módulo 12 Módulo 13 Módulo 17 Módulo 18 Módulo 19 Módulo 20 Módulo 24 Módulo 25
Movimientos del inventario general
Estado de los módulos fmInventario_ConsultaDeLosMovimientos fmInventario_Módulos
Exportar inventario
Importar inventario
Indique el número de cuenta del usuario
fmUsuarios_SelecciónDeUsuario
Número de cuenta: 089256121
VICTOR PÉREZ HERNÁNDEZ
Aceptar
Cancelar
modFunciones
modInicio
modInventario
modModulos
modPanel
modSanciones
modSeguridad
modUsuarios
Módulos proximos a vencer el tiempo de servicio
Materiales prestados:
frmPrincipal
Módulos utilizados:
Usuarios: 0
Módulo Activo: 0
SAEALABFI
Versión 1.2.1.0
© UNAM, 2000-2009
Universidad Nacional Autónoma de México
frmAcercaDe
Aceptar

IV.3.1 CLASE clsGeneral

Líneas	Archivo
63	clsGeneral.vb



Esta clase es de propósito general como lo sugiere su nombre, por lo que provee los enlaces a los componentes básicos del Sistema, dichos enlaces se pueden realizar mediante el uso de las propiedades de la clase.

Campos: *prvAyuda, prvConexión, prvConfiguración y prvHTML.*

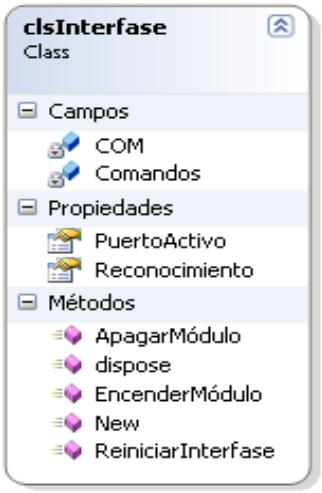
propiedades: *Ayuda, Conexión, Configuración y HTML.*

Métodos: *MostrarDialogo y New.*

Vínculos: *clsParámetros, modFunciones, clsModulo, clsParametros, clsTransacción, frm_Administración_ActivarCuenta, frm_Administración_ArchivoDeTransacciones, frm_Administración_BloquearCuenta, frm_Administración_BorrarCuenta, frm_Administración_CambiarContraseña, frm_Administración_CrearCuentaNueva, frm_Administración_Cuentas, frmConfiguraciónPorPrimeraVez, frmConsulta_ActualizaciónDatosXOperador, frmConsulta_ListaDeEspera, frmConsulta_MaterialesXUsuario, frmConsulta_MovsXMódulo, frmConsulta_MovsXOperadorMóduloYSemestre, frmConsulta_MovsXSemestre, frmConsulta_MovsXUsuario, frmConsulta_MovsXMóduloYSemestre, frmConsulta_PréstamosXSemestre, frmConsultaDeSancionesAplicadasPorOperadorDel*Sistema*, frmConsulta_SancionesxUsuario, frmConsultaDeUsuariosCreadosPorOperadorDel*Sistema*, frmEstadística_MaterialesPrestadosXUsuario, frmEstadísticas_MaterialesPrestados, frmEstadísticasDeSancionesAplicadasPorUsuario, frmEstadísticas_ServicioPorUsuario, frmEstadísticas_ListaEspera, frmEstadísticas_UsoDeEquiposPorUsuarios, frmEstadísticas_UsoDeEquipos, frmInicioDeSesión, frm_Inventario_ConsultaDeLosMovimientos, frmListadoDeUsuarios, frmMateriales_Entrega, frmMateriales_Préstamos, frmPrincipal, frmSanciones, frmUsuarios_Actualizar, frmUsuarios_Agregar, frm_Usurarios_SelecciónDeUsuario, modInicio, modInventario, modModulos, modPanel, modSanciones, modSeguridad, modUsuarios.*

IV.3.2 CLASE clsInterfase

Líneas	Archivo
99	clsIntrfase.vb



Esta clase es la encargada de proveer las herramientas básicas y necesarias para que el Sistema pueda establecer conexión mediante el protocolo de comunicación con la *interfaz electrónica* del **SAEALABFI**.

Campos: *COM y Comandos.*

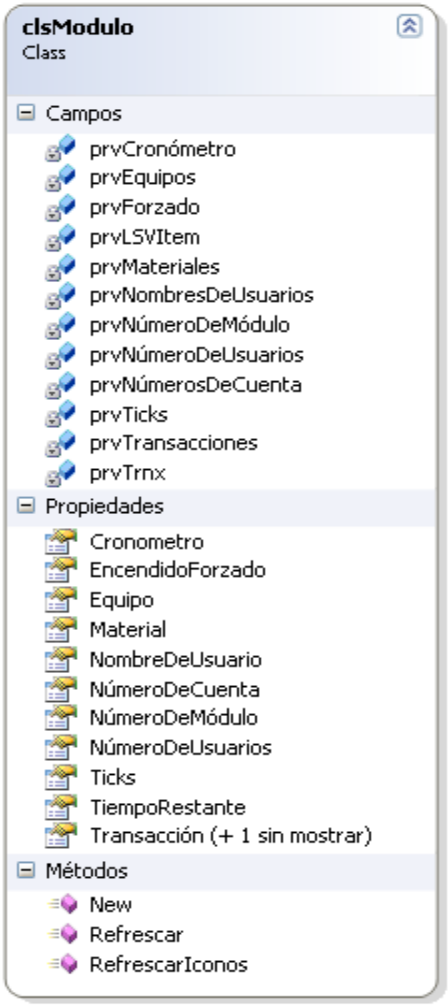
Propiedades: *PuertoActivo y Reconocimiento.*

Métodos: *ApagarMódulo, dispose, EncenderMódulo, New y ReiniciarInterfaz.*

Vinculos: *frmEspera, frmPrincipal, modFunciones, modInicio, modInventario y modPanel.*

IV.3.3 CLASE clsModulo

Líneas	Archivo
503	clsModulo.vb



Esta clase es el punto medular de las operaciones de cada uno de los módulos del Sistema; está provista de las propiedades y métodos que permiten la realización de todas las operaciones de control de tiempos, de registro de usuarios en el módulo que representa, los materiales prestados y el estado general en que se encuentra cada componente del módulo al que representa.

Campos: prvCronómetro, prvEquipos, prvForzado, prvLSVItem, prvMateriales, prvNombresDeUsuarios, prvNúmero, prvMódulo, prvNúmeroDeUsuarios, prvNúmeroDeCuenta, prvTicks, prvTransacciones, prvTrnx.

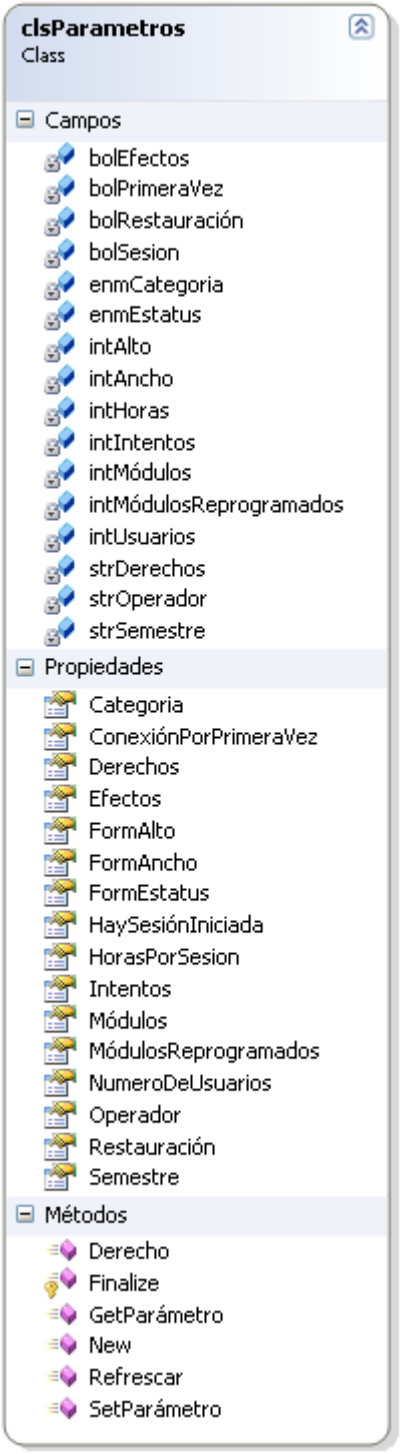
Propiedades: *Cronometro, EncendidoForzado, Equipo, Material, NombreDeUsuario, NúmeroDeCuenta, NúmeroDeMódulo, NúmeroDeUsuarios, Ticks, TiempoRestante y Transacción.*

Métodos: *New, Refrescar, RefrescarIconos.*

Vinculos: *frmAdministraciónPanel, frmConfiguraciónPorPrimeraVez, frmConsulta_MovsXMódulo, frmInventario_Módulos, frmListadoDeUsuarios, frmMateriales_Préstamos, frmPrincipal, frmSanciones, modInicio, modInventario, modModulos y modPanel.*

IV.3.4 CLASE clsParametros

Líneas	Archivo
597	clsParametros.vb



Esta clase proporciona todos los valores de las variables globales del Sistema y que les han sido nombradas como parámetros, dentro de ellos se encuentra el estatus de inicio de primera vez, la utilización o no de los efectos visuales en el Sistema, el proceso de restauración, el inicio de sesión y con ello el nombre y categoría del *Operador del Sistema*, la posición de la ventana principal, el número de horas por módulo, el número de usuarios por módulo, la reprogramación del número de módulos, eventualmente, la personalización de derechos y el semestre que se está cursando.

Campos: *bolEfectos, bolPrimeraVez, bolRestauración, bolSesión, enmCategoría, enmEstatus, intAlto, intAncho, intHoras, intIntentos, intMódulos, intMódulosReprogramados, intUsuarios, strDerechos, strOperador, strSemestre.*

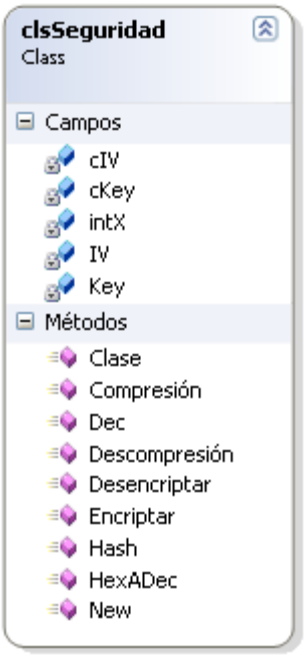
Propiedades: *Categoría, ConexiónPorPrimeraVez, Derechos, Efectos, FormAlto, FormAncho, FormEstatus, HaySesiónIniciada, HorasPorSesión, Intentos, Módulos, MódulosReprogramados, NúmeroDeUsuarios, Operador, Restauración y Semestre.*

Métodos: *Derecho, Finalize, GetParámetro, New, Refrescar y SetParámetro.*

Vinculos: *clsGeneral, clsMódulo, clsTransacción, frm_Administración_ActivarCuenta, frm_Administración_BloquearCuenta, frm_Administración_CambiarContraseña, frm_Administración_CrearCuentaNueva, frmAdministraciónPanel, frmConfiguraciónPorPrimeraVez, frmInicioDeSesión, frmInventario_Módulos, frm_Inventario_ConsultaDeLosMovimientos, frmListadoDeUsuarios, frmMateriales_Préstamos, frmPrincipal, frmSanciones, frmUsuarios_Agregar, modFunciones, modInicio, modInventario, modModulos, modPanel, modUsuarios.*

IV.3.5 CLASE clsSeguridad

Líneas	Archivo
188	clsSeguridad.vb



Esta clase proporciona al Sistema las funciones básicas de encriptación, de compresión, de cálculo del Código de Redundancia Cíclica (CRC), así como la conversión de una cadena a su equivalente hexadecimal de cada uno de los caracteres que la conforman.

Campos: *cIV, cKey, intX, IV yKey.*

Métodos: Clase, Compresión, Dec, Descompresión, Desencriptar, Encriptar, Hash, HexADec y New.

Vínculos: clsParametros, clsTransacción, frm_Administración_ActivarCuenta, frm_Administración_ArchivoDeTransacciones, frm_Administración_BloquearCuenta, frm_Administración_BorrarCuenta, frm_Administración_CambiarContraseña, frm_Administración_CrearCuentaNueva, frm_Administración_Cuentas, frmConfiguraciónPorPrimeraVez, frmInicioDeSesión, frmPrincipal, frmUsuarios_Actualizar, frmUsuarios_Agregar, modInicio, modInventario y modSeguridad.

IV.3.6 CLASE clsTransacción

Líneas	Archivo
561	clsTransacción.vb

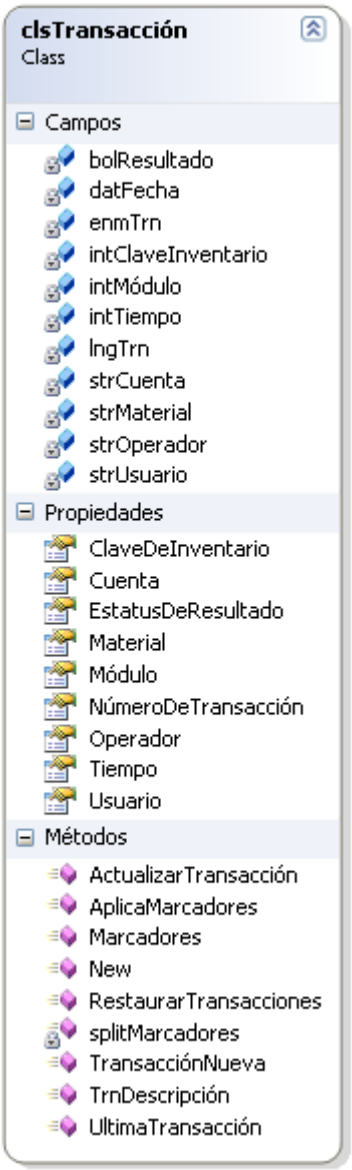
La finalidad de esta clase es la de permitir al Sistema guardar la información relacionada con cada una de las operaciones internas que se registran durante la operación cotidiana del Sistema, al momento de la liberación de esta versión del **SAEALABFI**, se han clasificado un total de 72 transacciones distintas y serán de utilidad en las tareas de administración y seguimiento para el uso que se le da al Sistema y para poder identificar posibles fallas para correcciones posteriores.

Campos: *bolResultado, datFecha, enmTrn, intClaveInventario, intMódulo, intTiempo, lngTrn, strCuenta, strMaterial, strOperador* y *strUsuario.*

Propiedades: *ClaveDeInventario, Cuenta, EstatusDeResultado, Material, Módulo, NúmeroDeTransacción, Operador, Tiempo* y *Usuario.*

Métodos: *ActualizarTransacción, AplicarMarcadores, Marcadores, New, RestaurarTransacciones, SplitMarcadores, TransacciónNueva, trnDescripción* y *UltimaTransacción.*

Vínculos: *frm_Administración_ArchivoDeTransacciones, frm_Administración_CambiarContraseña, frm_Administración_CrearCuentaNueva, frm_Administración_Cuentas, frmAdministraciónPanel, frmInicioDeSesión, frm_Inventario_ConsultaDeLosMovimientos, frmMateriales_Préstamos, frmPrincipal, frmSanciones, frmUsuarios_Actualizar, frmUsuarios_Agregar, modInicio, modInventario, modModulos, modPanel, modSeguridad y modUsuarios,*



IV.3.7 CLASE frmAcercaDe

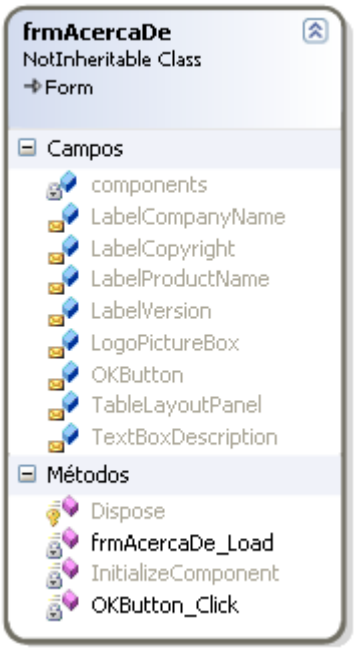
Líneas	Archivo
24	frmAcercaDe.vb

Esta clase representa el formulario **frmAcercaDe** cuya finalidad es la de presentar al *Operador del Sistema* la descripción básica de la información del Sistema, dentro de la que destacan el nombre de la aplicación, la versión, derechos y el logotipo.

Campos: *components, LabelCompanyName, LabelCopyright, LabelProductName, LabelVersion, LogoPictureBox, OKButton, TableLayoutPanel, TextBoxDescription.*

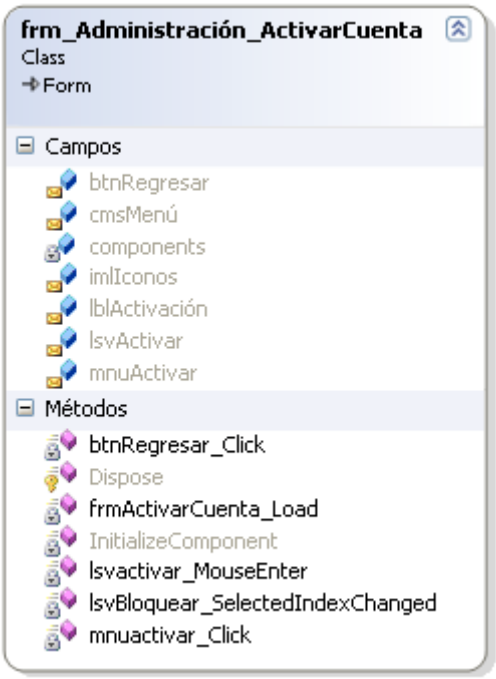
Métodos: *Dispose, frmAcercaDe_Load, IntializeComponent* y *OKButton_Click.*

Vínculos: *frmPrincipal.*



IV.3.8 CLASE frm_Administración_ActivarCuenta

Líneas	Archivo
96	frmAdministración-ActivarCuenta.vb



Esta clase representa al formulario "Activación de cuentas", está considerado dentro de las tareas administrativas del Sistema cuya finalidad es la de poder bloquear o desbloquear a las cuentas de operadores del Sistema para poder tener seguimiento de las actividades que realizan cada uno de ellos. Las tareas administrativas las puede realizar el súper-Administrador y los Administradores en jerarquía descendente; es decir, el súper-Administrador podrá realizar este tipo de operaciones con cualquier tipo de cuentas, mientras que los Administradores sólo sobre los usuarios y como restricción los usuarios normales no tienen acceso a esta opción.

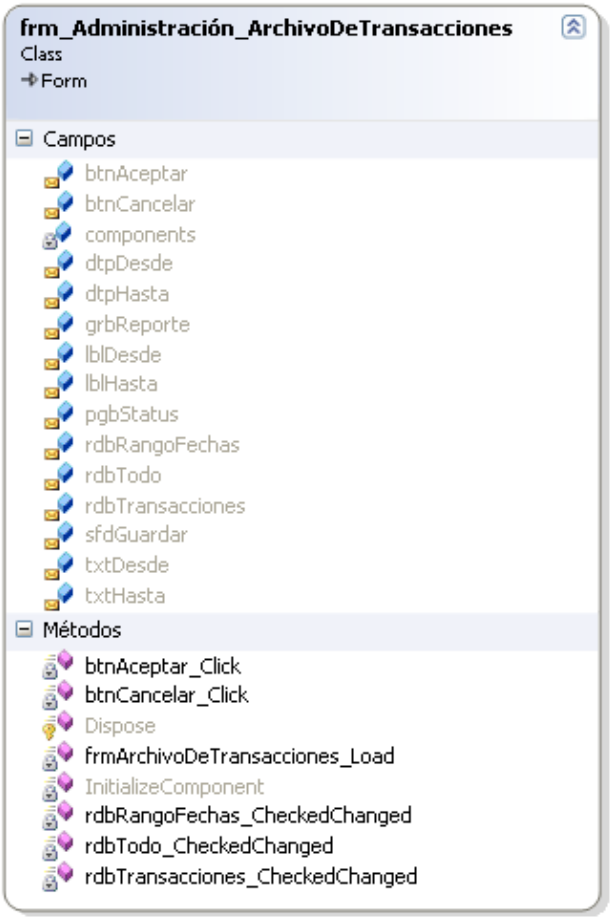
Campos: *btnRegresar, cmsMenú, components, imlIconos, lblActivación, lsvActivar, mnuActivar.*

Métodos: *btnRegresar_Click, Dispose, frmActivarCuenta_Load, InitializeComponent, lsvActivar_MouseEnter, lsvBloquear_SelectedIndexChanged* y *mnuactivar_click.*

Vínculos: *frmPrincipal.*

IV.3.9 CLASE frm_Administración_ArchivoDeTransacciones

Líneas	Archivo
212	frmAdministración-ArchivoDeTransacciones.vb



Esta clase representa al formulario frm_Administracin_ArchivoDeTransacciones cuya finalidad es la de ofrecer al Administrador apoyo para poder crear un archivo de texto plano en donde se plasmen el detalle de todas las transacciones que se han generado durante la operación normal del Sistema permitiéndole al operador elegir entre tres posibilidades: todas las transacciones, por rango de fechas y por rango en el número de transacciones.

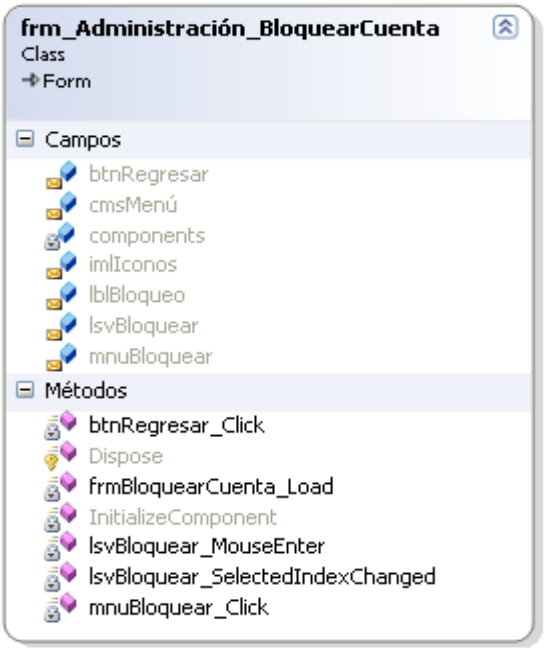
Campos: *btnAceptar, btnCancelar, components, dtpDesde, dtpHasta, grbReporte, lblDesde, lblHasta, pgbStatus, rdbRangoFechas, rdbTodo, rdbTransacciones, sfdGuardar, txtDesde, txtHasta.*

Métodos: *btnAceptar_Click, btnCancelar_Click, Dispose, frmArchivoDeTransacciones_Load, InitializeComponent, rdgRangeFechas_CheckedChanged, rdbTodo_CheckedChanged, rdbTransacciones_CheckedChanged.*

Vínculos: *frmPrincipal.*

IV.3.10 CLASE frm_Administración_BloquearCuenta

Líneas	Archivo
98	frmAdministración-BloquearCuenta.vb



La finalidad de esta clase es la de representar al formulario frm_Administración_BloquearCuenta, cuyo fin es el de apoyar al operador en las tareas administrativas del Sistema en relación con el bloqueo de cuentas, por lo que su uso es la opuesta a la descrita en la clase frm_Administracción_ActivarCuenta.

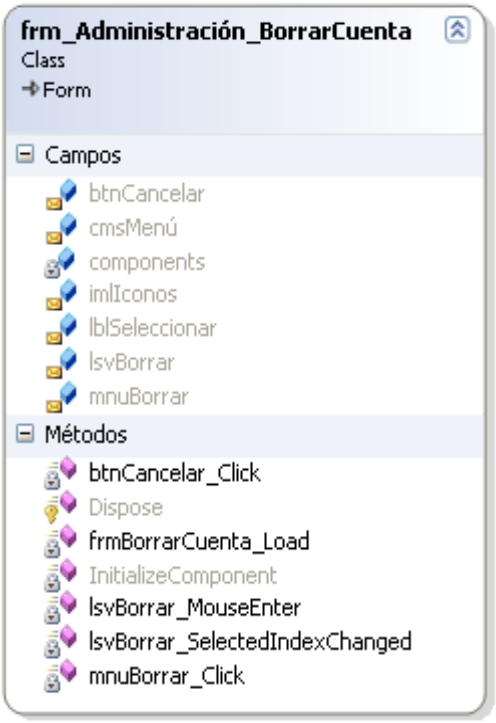
Campos: *btnRegresar, cmsMenú, components, imlIconos, lblBloqueo, lsvBloquear y mnuBloquear.*

Métodos: *btnRegresar_Click, Dispose, frmBloquearCuenta_Load, InitializeComponent, lsvBloquear_MouseEnter, lsvBloquear_SelectedIndexChanged y mnuBloquear_Click.*

Vínculos: *frmPrincipal.*

IV.3.11 CLASE frm_Administración_BorrarCuenta

Líneas	Archivo
84	frmAdministración-BorrarCuenta.vb



La finalidad de esta clase es la de dar apoyo al *Operador del Sistema* en las tareas administrativas relacionadas con las cuentas de operadores; primordialmente se enfoca a eliminar cuentas que han sido creadas con anterioridad. Esta clase al ser utilizada como formulario sólo lo pueden hacer el súper-Administrador y los Administradores.

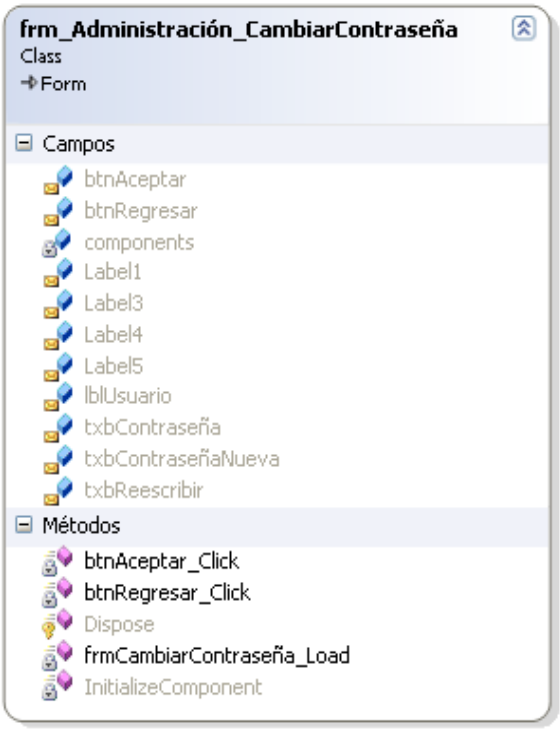
Campos: *btnCancelar, cmsMenú, components, imlIconos, lblSeleccionar, lsvBorrar y mnuBorrar.*

Métodos: *btnCancelar_Click, Dispose, frmBorrarCuenta_Load, InitializeComponent, lsvBorrar_MouseEnter, lsvBorrar_SelectedIndexChanged, mnuBorrar_Click.*

Vínculos: clsGeneral, clsParametros, clsSeguridad, modFunciones, modSeguridad y frmPrincipal.

IV.3.12 CLASE frm_Administración_CambiarContraseña

Líneas	Archivo
107	frmAdministración-CambiarContraseña.vb



Mediante esta clase se representa la opción de "Cambiar contraseña" a las cuentas de los operadores del Sistema, forma parte de las opciones que han sido escritas para las tareas de apoyo en la administración de los recursos del Laboratorio.

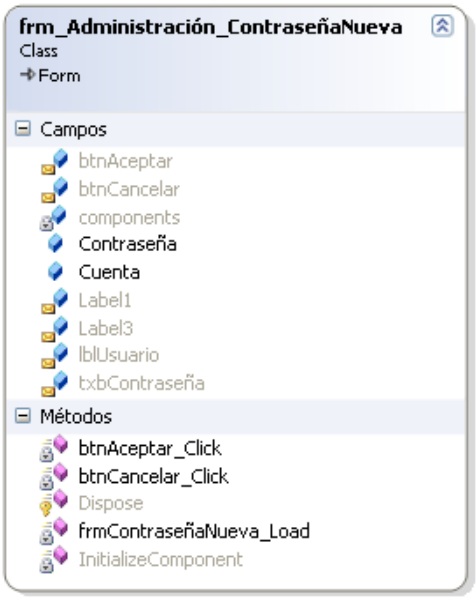
Campos: *btnAceptar, btnRegresar, components, Label1, Label3, Label4, Label5, lblUsuario, txbContraseñe, txbContraseñaNueva y txbReescribir.*

Métodos: *btnAceptar_Click, btnRegresar_Click, Dispose, frmCambiarContraseña_Load e InitializeComponent.*

Vínculos: *clsGeneral, clsParametros, clsSeguridad, clsTransacción, modFunciones, modSeguirdad y frmPrincipal.*

IV.3.13 CLASE frm_Administración_ContraseñaNueva

Líneas	Archivo
18	frmAdministración-ContraseñaNueva.vb



De igual manera que las clases anteriores, ésta clase sirve de apoyo complementario en las funciones de administración de cuentas de operadores del Sistema y su finalidad es la de apoyar en el proceso de selección y asignación de contraseñas a operadores que ya se encuentran registrados en el Sistema. La función de esta clase se invoca desde la clase frm_Administración_Cuentas y sólo es accesada por el súper-Administrador.

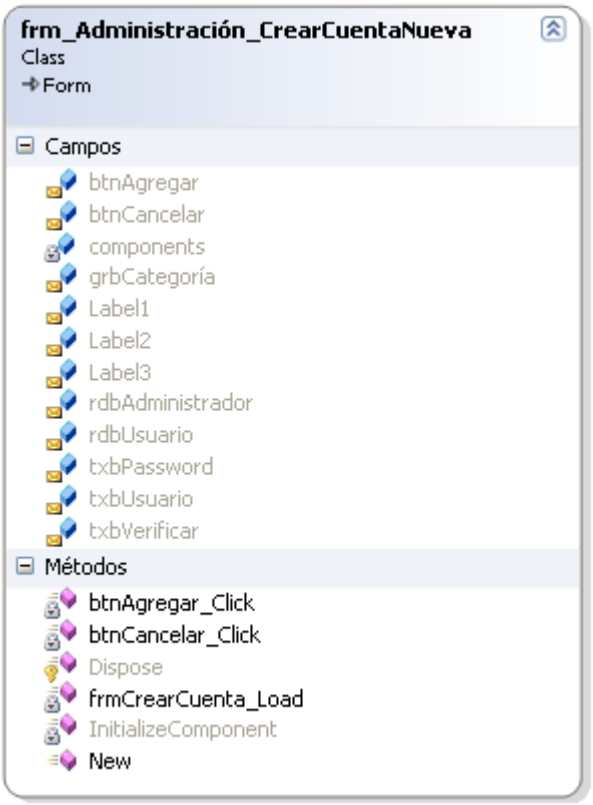
Campos: *btnAceptar, btnCancear, components, contraseña, cuenta, label1, label3, lblUsuario y txbContraseña.*

Métodos: *btnAceptar_Click, btnCancelar_Click, Dispose, frmContraseñaNueva_Load e InitializeComponent.*

Vínculos: *modFunciones y frm_Administración_Cuentas.*

IV.3.14 CLASE frm_Administración_CrearCuentaNueva

Líneas	Archivo
100	frm_Administración_CrearCuentaNueva.vb



El propósito de esta clase es el de brindar al *Operador del Sistema* con privilegios de súper-Administrador o Administrador, la posibilidad de crear cuentas de operadores nuevas. El súper-Administrador tiene acceso a todas las posibilidades por lo que podrá crear cuentas de Administrador y de operador, mientras que los Administradores sólo podrán crear cuentas de operadores.

Campos: *btnAgregar, btnCancelar, components, grbCategoría, components, Label1, Label2, Label3 rdbAdministrador, rdbUsuario, txbPassword, txbUsuario y txbVerificar.*

Métodos: *btnAgregar_Click, btnCancelar_Click, Dispose, frmCrearCuenta_Load, InitializeComponent y New.*

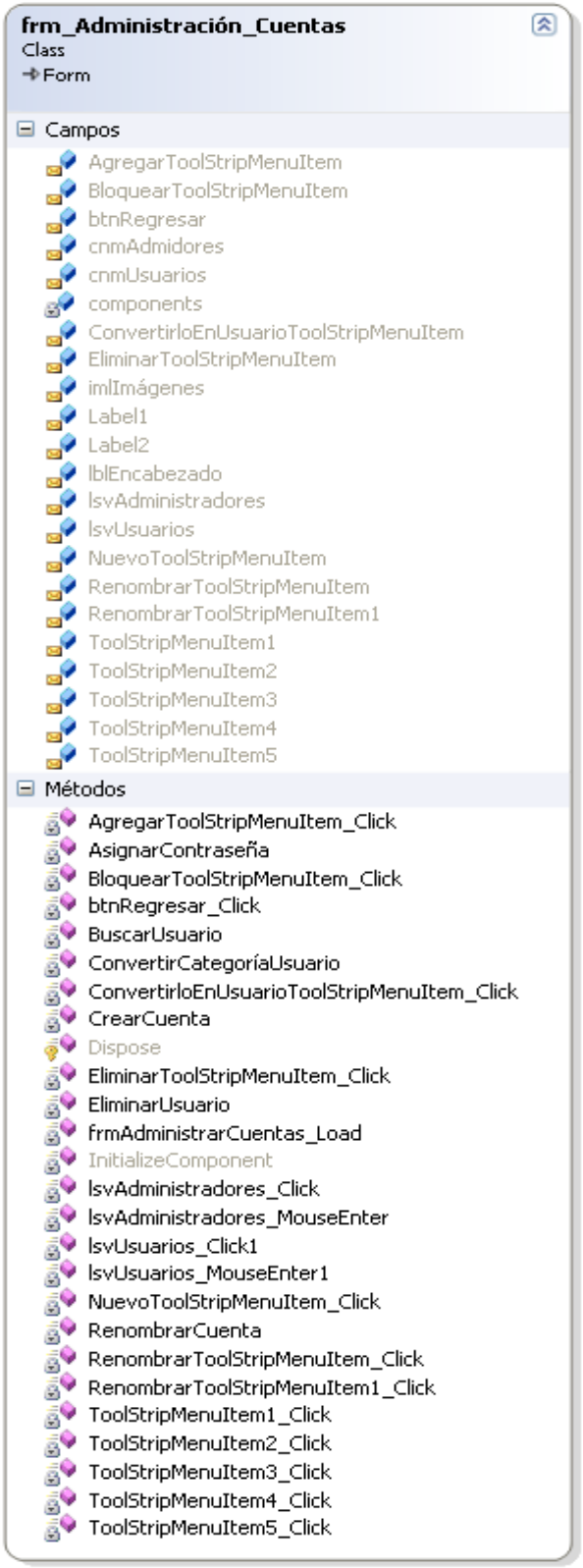
Vínculos: *clsGeneral, clsParametros, clsSeguridad, clsTransacción, modFunciones y frmPrincipal.*

IV.3.15 CLASE frm_Administración_Cuentas

Líneas	Archivo
389	frmAdministración-Cuentas.vb

El propósito de esta clase es la de proporcionar a los operadores del Sistema con categoría única de súper-Administrador, acceder a las funciones (formularios – clase) relacionadas con las tareas de administración de todas las cuentas de los operadores del Sistema y por lo tanto, constituye una herramienta indispensable para el fin que se ha señalado.

Campos: *AgregarToolStripMenuIten, BloquearToolStripMenuItem, btnRegresar, cnmAdmidores, cnmUsuarios, components, ConvertirloEnUsuarioToolStripMenuIten, EliminarToolStripMenuItem, imlImágenes, Label1, Label2, lblEncabezado, lsvAdministradores, lsvUsuarios, NuevoToolStripMenuItem, RenombrarToolStripMenuItem, RenombrarToolStripMenuItem1, ToolStripMenuItem1, ToolStripMenuItem2, ToolStripMenuItem3, ToolStripMenuItem4 y ToolStripMenuItem5.*

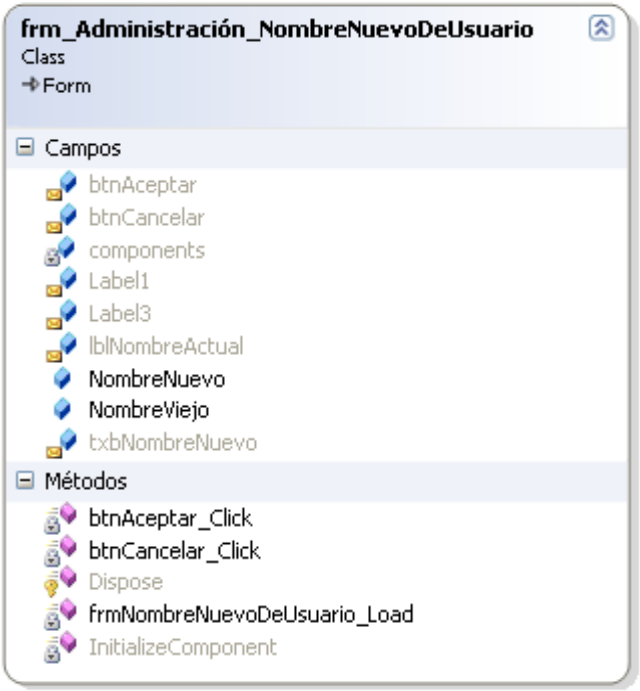


Métodos: *AgregarToolStripMenuItem_Click, AsignarContraseña, BloquearToolStripMenuItem_Clic, btnRegresar_Click, BuscarUsuario, ConvertirCategoríaUsuario, ConvertirloEnUsuarioToolStripMenuItem_Click,CrearCuenta, Dispose, EliminarToolStripMenuItem_Click, EliminarUsuario, frmAdministrarCuentras_Load, InitializeComponent, lsvAvministradores_Click, lsvAdministradores_MouseEnter, lsvUsuarios_Click1, lsvUsuarios_MouseEnter1, NuevoToolStripMenuItem_Click, RenombrarToolSTripMenuItem1_Click, ToolStripMenuItem1_Click, ToolStripMenuItem2_Click, ToolStripMenuItem3_Click, ToolStripMenuItem4_Click y ToolStripMenuItem5_Click.*

Vínculos: *clsGeneral, clsSeguridad, clsTransacción, modFunciones, modSeguridad, frm_Administración_ContraseñaNueva y frmPrincipal.*

IV.3.16 CLASE frm_Administración_NombreNuevoDeUsuario

Líneas	Archivo
22	frmAdministración-NombreNuevoDeUsuario.vb



La finalidad de esta clase (formulario) es la de permitir al *Operador del Sistema* realizar las tareas de administración sobre las cuentas de operadores. En este caso se trata del cambio de nombre o alias de alguno de los operadores que se encuentran registrados y sólo se puede realizar desde una cuenta de súper-Administrador.

Campos: *btnAceptar, btnCancelar, components, Label1, Label3, lblNombreActual, NombreNuevo, NombreViejo, txbNombreNuevo.*

Métodos: *btnAceptar_Click, btnCancelar_Click, Dispose, frmNombreNuevoDeUsuario_Load e InitializeComponent.*

Vínculos: *modFunciones y frm_Administración_Cuentas.*

IV.3.17 CLASE frmAdministraciónPanel

Líneas	Archivo
346	frmAdministración-Panel.vb

Esta clase (formulario), es el punto de partida para todos los formularios básicos de administración de cuentas de operadores de manera particular, debido a que desde aquí se acceden a todas las funciones de administración de manera rápida y se encuentra orientada para que sólo el súper-Administrador pueda acceder a esta.

Este formulario concentra en un solo grupo de opciones y características a todas las funciones básicas de operación del panel de operación del Sistema en su conjunto.

Campos: *btnBuscarUsuario, btnLiberarPanel, btnRegresar, components, imlPanel, lsvDetalle, lsvMódulos, mnuApagadoForzado, mnuEncendidoForzado, mnuLiberarMóodulo, mnuMódulos, mnuUsuarios, mnux15minutos, mnux30mminutos, mnux45minutos, mnux60minutos, munxAgregar, mnuxEntegarMateriales, mnuxPrestarMateriales, mnuxQuitarUsuario, tlsComandos, tlsSep0, tmrEstatus, tmrIndicador, tmrMódulosXVencer, tmrRelojes, ToolTiip y tsmSep0.*

frmAdministraciónPanel
Class
Form

Campos
- btnBuscarUsuario
- btnLiberarPanel
- btnRegresar
- components
- imlPanel
- lsvDetalle
- lsvMódulos
- mnuApagadoForzado
- mnuEncendidoForzado
- mnuLiberarMódulo
- mnuMódulos
- mnuUsuarios
- mnux15minutos
- mnux30minutos
- mnux45minutos
- mnux60minutos
- mnuxAgregar
- mnuxEntregarMateriales
- mnuxPrestarMateriales
- mnuxQuitarUsuario
- tlsComandos
- tlsSep0
- tmrEstatus
- tmrIndicador
- tmrMódulosXVencer
- tmrRelojes
- ToolTip
- tsmSep0

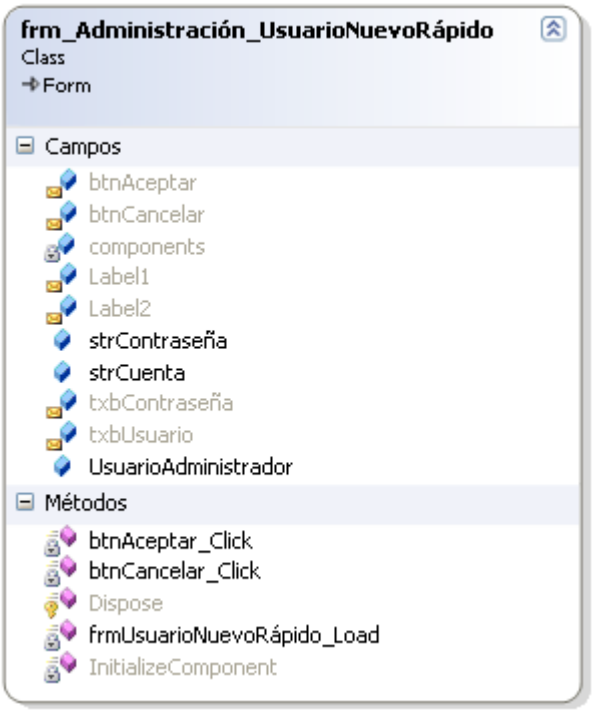
Métodos
- AgregarMásTiempo
- btnBuscarUsuario_Click
- btnLiberarPanel_Click
- btnRegresar_Click
- Dispose
- frmAdministraciónPanel_Load
- IconoDeMódulo
- InitializeComponent
- lsvDetalle_ItemSelectionChanged
- lsvDetalle_MouseEnter
- lsvMódulos_SelectedIndexChanged
- mnuApagadoForzado_Click
- mnuEncendidoForzado_Click
- mnuLiberarMódulo_Click
- mnux15minutos_Click
- mnux30minutos_Click
- mnux45minutos_Click
- mnux60minutos_Click
- mnuxAgregar_Click
- mnuxEntregarMateriales_Click
- mnuxPrestarMateriales_Click
- mnuxQuitarUsuario_Click
- RefrescarDetalle
- tmrEstatus_Tick
- tmrIndicador_Tick
- tmrMódulosXVencer_Tick
- tmrRelojes_Tick

Métodos: *AgregarMásTiempo, btnBuscarUsuario_Click, btnLiberarPanel_Click btnRegresar_Click, Dispose, frmAdministraciónPanel_Load, InicioDeMódulo, IntializeComponent, lsvDetalle_ItemSelectionChanged, lsvDetalle_MouseEnter, lsvMódulos_SelectedIndexChanged, mnuApagdoForzado_Click, mnuEncendidoForzado_Click, mnuLiberarModulo_Click, mnux15minutos_Click, mnux30minutos_Click, mnux45minutos_Click, mnux60minutos_Click, mnuxAgregar_Click, mnuxEntregarMateriales_Click, mnuxPrestarMateriales_Click, mnuQuitarUsuario_click, RefrescarDetalle, tmrStatus__Tick, tmrMódulosXVencer_Tick y tmrRelojes_Tick.*

Vínculos: *clsModulo, clsParametros, clsTransacción, modFunciones, modInventario, modModulos, modPanel, modUsuarios y frmPrincipal.*

IV.3.18 CLASE frm_Administración_UsuarioNuevoRápido

Líneas	Archivo
47	frmAdministración-UsuarioNuevo.vb



La finalidad de de esta clase (formulario) es la de proporcionar a los operadores del Sistema con categoría de súper-Administrador y Administradores, el mecanismo mediante el cual se puede crear a un usuario nuevo de manera rápida. Esta característica se accede desde las opciones de administración de cuentas.

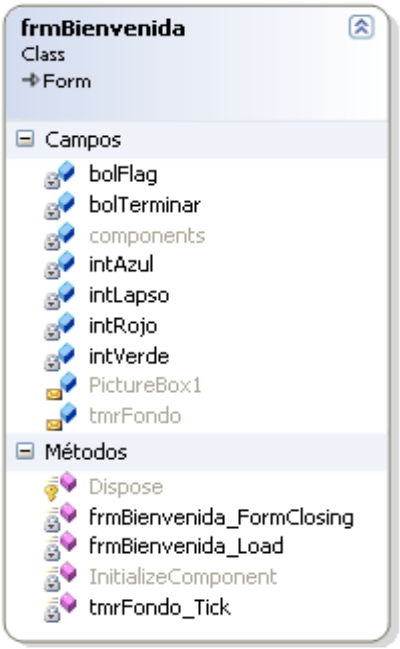
Campos: *btnAceptar, btnCancelar, components, Label1, Label2 strContraseña, strCuenta, txbContraseña, txbUsuario y UsuarioAdministrador.*

Métodos: *btnAceptar_Click, btnCancelar_Click, Dispose, frmUsuarioNuevoRápido_Load e InitializeComponent.*

Vínculos: *modFunciones y frm_Administración_Cuentas.*

IV.3.19 CLASE frmBienvenida

Líneas	Archivo
276	frmBienvenida.vb



La finalidad de esta clase (formulario) es la de presentar en una ventana transparente al escudo de la UNAM y junto con ella unos textos de leyenda en donde se indican los créditos y nombre del Sistema. Éste aparecerá después de haber superado la fase de identificación de credenciales del *Operador del Sistema*.

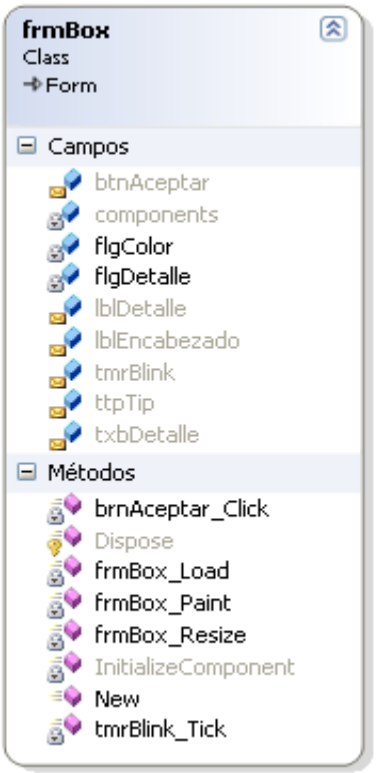
Campos: *bolFlg, bolTerminar, components, intAzul, intLapso, intRojo, intVerde, PictureBox1 y tmrFondo.*

Métodos: *Dispose, frmBienvenida__FormClosing, frmBienvenida_Load, InitializeComponent y tmrFondo_Tick.*

Vínculos: *frmPrincipal.*

IV.3.20 CLASE frmBox

Líneas	Archivo
113	frmBox.vb



La finalidad de esta clase es de servir la base de la construcción y prestación de los mensajes de aviso del Sistema, así como del cuadro de solicitud de aceptación o rechazo de la sugerencia que se ofrece a los operadores en situaciones concretas dentro de la operación normal del Sistema.

Campos: *btnAceptar, components, flgColor, flgDetalle, lblDetalle, lblEncabezado, tmrBlink, ttpTip y txbDetalle.*

Métodos: *btnAceptar_Click, Dispose, frmBox_Load, frmBox_Paint, frmBox_Resize, InitializeComponent, new y tmrBlink_Tick.*

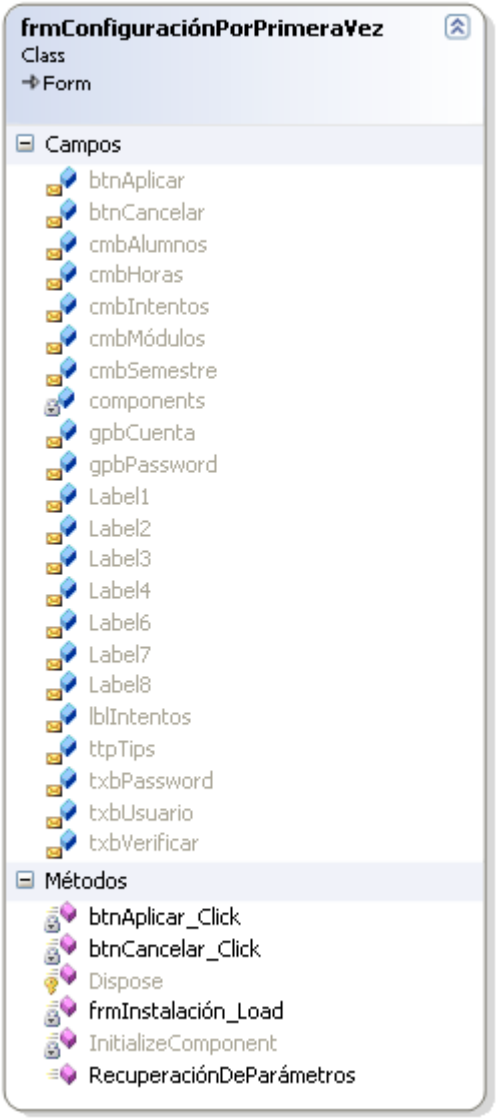
Vínculos: *modFunciones y frmPregunta.*

IV.3.21 CLASE frmConfiguraciónPorPrimeraVez

Líneas	Archivo
346	frmConfiguraciónPorPrimeraVez.vb

La finalidad de esta clase (formulario) es de mostrar al *Operador del Sistema* la llamada ventana de configuración de primera vez; esta ventana será mostrada durante el proceso de inicio de sesión, una vez detectado que el Sistema ha sido instalado de manera correcta y en consecuencia, se ofrece la posibilidad de establecer los parámetros de inicio necesarios para poder comenzar a trabajar con el Sistema.

En esta ventana se configura el semestre que se está cursando, el número de usuarios máximo por modulo, el número de horas máximo de servicio, el número de intentos al momento de iniciar sesión cuando el operador intenta acceder con su cuenta y contraseña; y, aquí se debe de establecer el nombre del primer Administrador (no es el súper-Administrador) y su contraseña. Cuando se elige guardar cambios, esta ventana no aparecerá nunca más a no ser que se vuelva a instalar el Sistema.



Campos: *btnAceptar, btnCancelar, cmbAlumnos, cmbHoras, cmbIntentos, cmbMódulos, cmbSemestre, components, gpbCuenta, gpbPassword, Label1, Label2, Label3, Label4, Label5, label6, label7, label8, lblIntentos, ttpTips, txbPassword, txbUuario y txbVerificar.*

Métodos: *btnAplicar_Click, btnCancelar_Click, Dispose, frmInstalación_Load, InitializeComponent y RecuperaciónDeParámetros.*

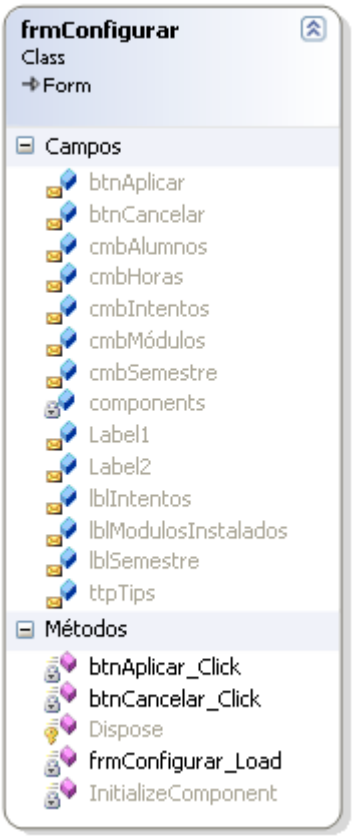
Vínculos: *clsGeneral, clsModulo, clsParametros, clsSeguridad, modFunciones, modInicio, modModulos y modSeguridad.*

IV.3.22 CLASE frmConfigurar

Líneas	Archivo
45	frmConfigurar.vb

La finalidad de esta clase, es la de contener al formulario encargado de realizar los ajustes a los parámetros de configuración del Sistema. Desde aquí se puede cambiar el número de módulos instalados para ser operados desde el Sistema, el semestre que se está cursando, el número máximo de usuarios que se permitirán por módulo, así como el número de horas máxima por sesión. Por sus características, esta es la clase complementaria a frmConfiguraciónPorPrimeraVez.

Campos: *btnAplicar, btnCancelar, cmbAlumnos, cmbHoras, cmbIntentos, cmbMódulos, cmbSemestre, components, Label1, Label2, lblIntengos, lblModulosInstalados, lblSemestre y ttpTips.*

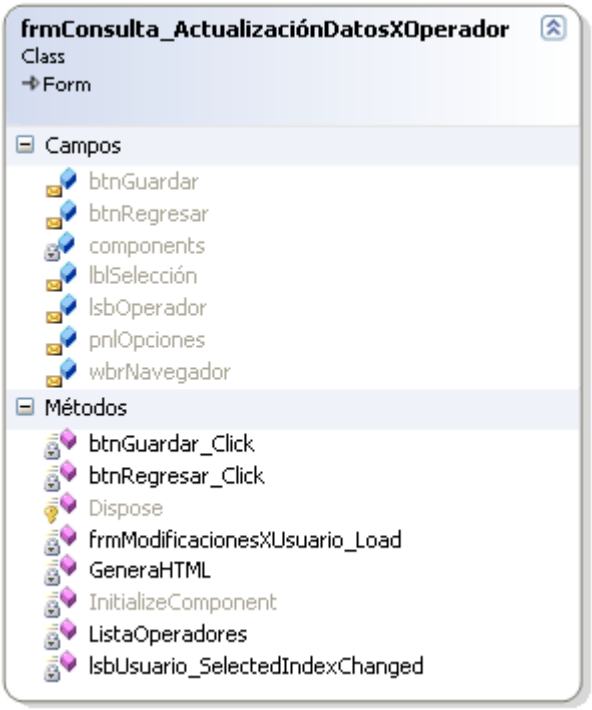


Métodos: *btnAplicar_Click, btnCancelar_Click, Dispose, frmConfigurar_Load y InitializeComponent.*

Vínculos: *clsParámetros, modFunciones y frmPrincipal.*

IV.3.23 CLASE frmConsulta_ActualizaciónDatosXOperador

Líneas	Archivo
224	frmConsulta-Actualización de datos de los usuarios por operador.vb



La finalidad de esta clase es la de mostrar el formulario el resultado de la consulta de todos los datos de usuarios que se han ofrecido por *Operador del Sistema*. Los resultados solicitados se mostrarán en formato HTML que se encontrará dentro de la distribución de la clase – formulario y sólo será podrán hacerlo los operadores con privilegios de administración.

Los resultados a los que se hace referencia, se pueden guardar en alguna ubicación que prefiera el operador para su posterior revisión, impresión o análisis según convenga y sea el caso.

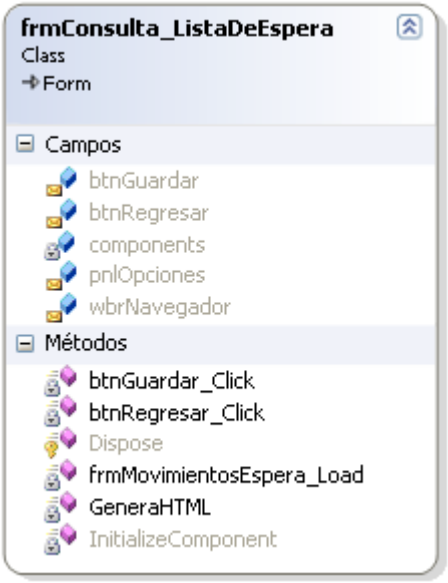
Campos: *btnGuardar, btnRegresar, components, lblSelección, lsbOperador, pnlOpciones y wbrNavegador.*

Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmModificacionesXUsuario_Load, GeneraHTML, InitializeComponent, ListaOperadores y lbUsuario_SelectedIndexChanged.*

Vínculos: *clsGeneral, modFunciones y frmPrincipal.*

IV.3.24 CLASE frmConsulta_ListaDeEspera

Líneas	Archivo
215	frmConsulta-Lista de espera.vb



La finalidad de esta clase-formulario es la de proporcionar al *Operador del Sistema* una herramienta en la que se pueda ofrecer un informe sobre el comportamiento de la lista de espera. Al igual que en el caso anterior, los resultados se reflejan en formato HTML que se encontrará incrustado dentro del formulario, permitiendo al operador guardar los resultados generados en la ubicación que desee para su posterior análisis o impresión según sea la decisión y necesidad del *Operador del Sistema*.

Campos: *btnGuardar, btnRegresar, components, pnlOpciones y wbrNavegador.*

Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmMovimientosEspera_Load, GeneraHTML e InitializeComponent.*

Vínculos: *clsGeneral, modFunciones y frmPrincipal.*

IV.3.25 CLASE frmConsulta_MaterialesXUsuario

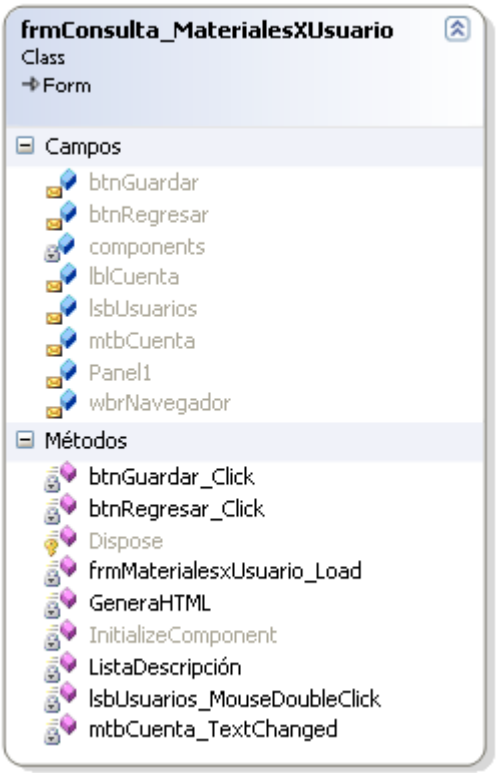
Líneas	Archivo
272	frmConsulta-Materiales prestados por usuario.vb

Esta clase-formulario, ofrece al *Operador del Sistema* la posibilidad de consultar los movimientos registrados en el Sistema de todos los materiales que han sido prestados a un usuario en particular. Al igual que en la clase anterior, esta muestra los resultados generados en formato HTML incrustado dentro de la distribución y diseño del formulario, contando con la posibilidad de guardar su contenido en una ubicación en particular para su análisis o impresión posterior.

Campos: *btnGuardar, btnRegresar, components, lblCuenta, lsbUsuarios, mtbCuenta, Panel1 y wbrNavegador.*

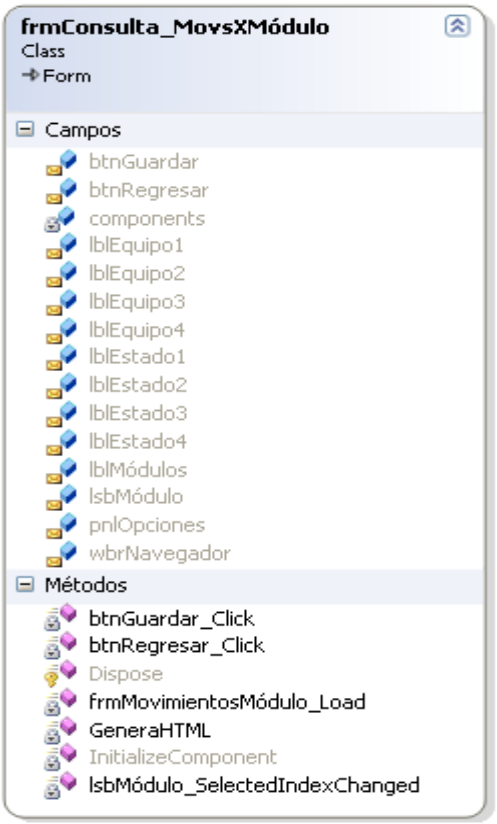
Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmMaterialesxUsuario_Load, GeneraHTML, IntializeComponent, ListaDescripcióh, lsbUsuarios_MouseDoubleClick y mtbCuenta_TextChanged.*

Vínculos: *clsGeneral, modFunciones, modUsuarios y frmPrincipal.*



IV.3.26 CLASE frmConsulta_MovsXMódulo

Líneas	Archivo
317	frmConsulta-Movimientos por Módulo.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema, consultar los movimientos registrados sobre los distintos módulos instalados en el Laboratorio. El formulario se encuentra estructurado de manera tal en la que las condiciones deseadas reflejen los resultados en formato HTML con la posibilidad de proporcionar al operador, guardar los datos en la posición que desee para su posterior consulta o impresión.

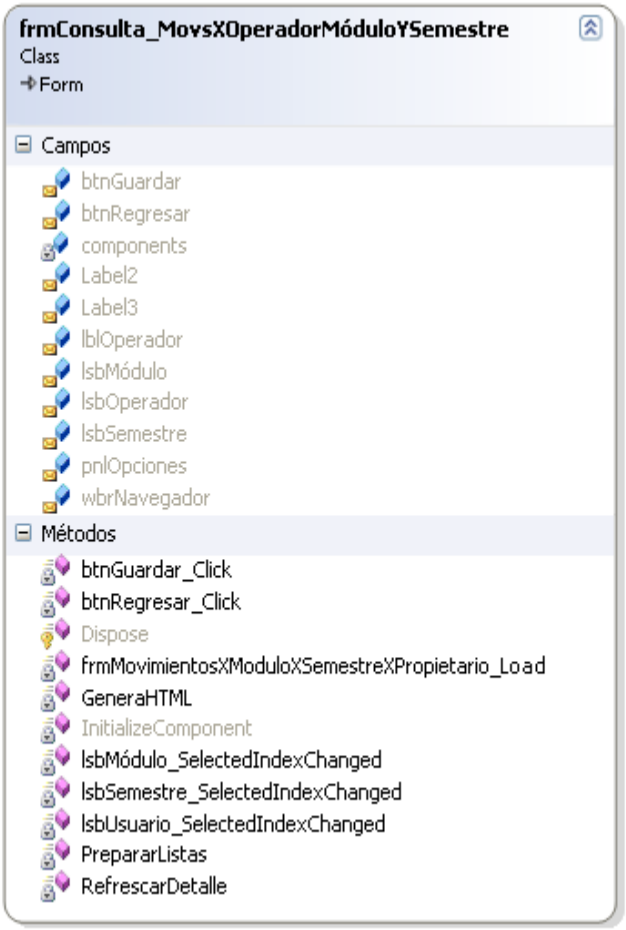
Campos: *btnGuardar, btnRegresar, components, lblEquipo1, lblEquipo2, lblEquipo3, lblEquipo4, lblEstado1, lblEstado2, lblEstado3, lblEstado4, lblMódulos, lsbMódulo, pnlOpciones y wbrNavegador.*

Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmMovimientosMódulo_Load, Dispose, frmMovimientosMódulo_Load, GeneraHTML, InitializeComponent y lsbMódulo_SelectedIndexChanged.*

Vínculos: *clsGeneral, clsMódulo, modFunciones, modUsuarios y frmPrincipal.*

IV.3.27 CLASE frmConsulta_MovsXOperadorMóduloYSemestre

Líneas	Archivo
398	frmConsulta-Movimientos por operador, módulo y semestre.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema una herramienta de consulta sobre los movimientos que se han registrado en el Sistema en general, agrupados por operador, módulo y semestre; estas condiciones las podrá seleccionar el operador para su análisis inmediato o posterior con la posibilidad de ser impresos o llevar un control histórico para los fines de administración y seguimiento en las tareas de administración del Laboratorio.

Campos: *btnGuardar, btnRegresar, components, Label2, Label3, lblOperador, lsbMódulo, lsbOperador, lsbSemetre, pnlOpciones y wbrNavegador.*

Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmMovimientosXModuloXSemestreXPropietario_Load, GeneraHTML, InitializeComponent, lsbMódulo_SelectedIndexChanged, lsbSemestre_SelectedIndexChanged, lsbUsuario_SelectedIndexChanged, PrepararListas y RefrescarDetalle.*

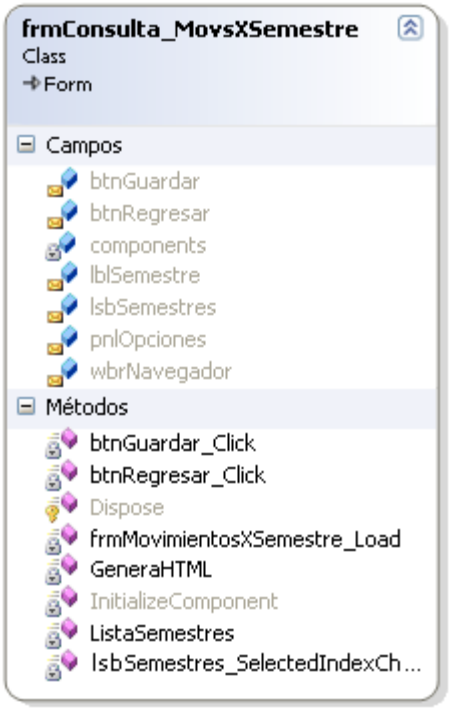
Vínculos: *clsGeneral, modFunciones y frmPrincipal.*

IV.3.28 CLASE frmConsulta_MovsXSemestre

Líneas	Archivo
258	frmConsulta-Movimientos Por Semestre.vb

La finalidad de esta clase-formulario, es la de brindar a los operadores del Sistema la posibilidad de consultar los movimientos registrados en las instalaciones del Laboratorio agrupados por semestre, según lo elija el operador en turno. Los resultados que se generan se muestran dentro del diseño del formulario mediante el formato HTML, ofreciendo la posibilidad de guardar los resultados obtenidos para su análisis inmediato o posterior, así como para las posibilidades de impresión de los mismos.

Campos: *btnGuardar, btnRegresar, components, lblSemestre, lsbSemetres, pnlOpciones y wbrNavegador.*

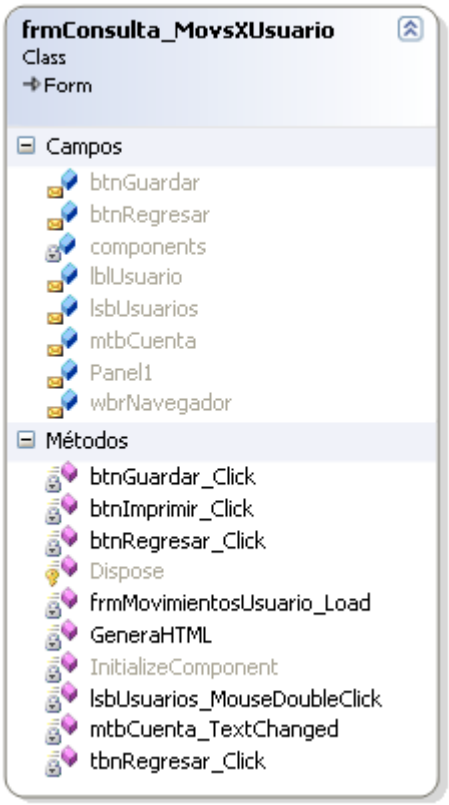


Métodos: *btnGuardar__Click, btnRegresar_Click, Dispose, frmMovimientosXSemestre_Load, GeneraHTML, InitializeComponent, ListaSemestres y lsbSemestres_SelectedIndexChanged.*

Vínculos: *clsGeneral, modFunciones, y frmPrincipal.*

IV.3.29 CLASE frmConsulta_MovsXUsuario

Líneas	Archivo
247	frmConsulta-Movimientos por Usuario.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema, una herramienta mediante la cual puede consultar los movimientos que los usuarios han registrado sobre el uso que han dado a las instalaciones del Laboratorio durante la solicitud de los servicios que éste brinda. La clase mediante el diseño del formulario ofrece la posibilidad de elegir al usuario que se desea consultar y los resultados se mostraran en el formulario en formato HTML para su análisis inmediato o posterior así como para poder imprimir esta información.

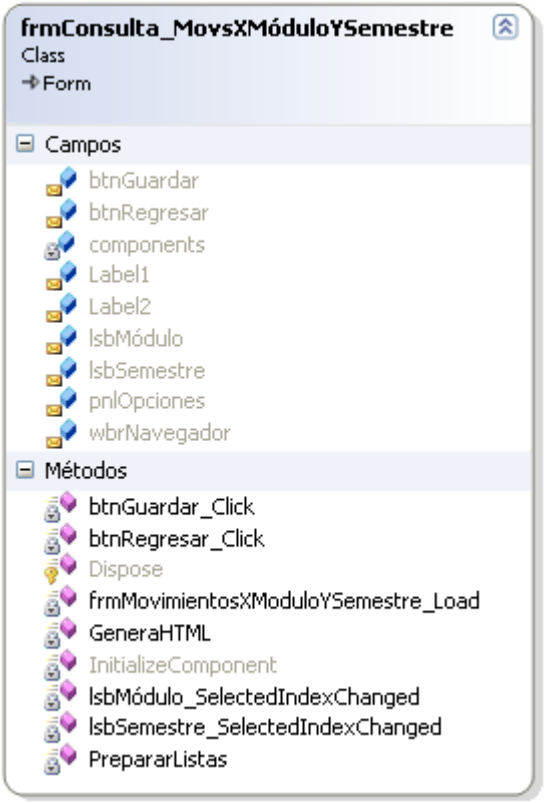
Campos: *btnGuardar, btnRegresar, components, lblUsuario, lsbUsuarios, mtbCuenta, Panel1 y wbrNavegador.*

Métodos: *btnGuardar__Click, btnImprimir_Click, btnRegresar_Click, Dispose, frmMovimientosUsuario_Load, GeneraHTML, InitializeComponent, lsbUsuarios_MouseDoubleClick, mtbCuenta_TextChanged y tbnRegresar_Click.*

Vínculos: *clsGeneral, modFunciones, modUsuarios y frmPrincipal.*

IV.3.30 CLASE frmConsulta_MovsXMóduloYSemestre

Líneas	Archivo
323	frmConsulta-Movimientos x Módulo y Semestre.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema una herramienta de consulta sobre los movimientos que se han registrado durante la operación del Sistema. Esta información se puede configurar mediante la selección de las opciones que se ofrecen para elegir al módulo deseado y el Semestre para acotar la consulta. Los resultados que se generan se colocan dentro de un control que mantiene la información en formato HTML para su análisis inmediato o posterior al ser guardada la información o para su impresión. Esta opción tiene la finalidad de proporcionar a los operadores del Sistema una herramienta para dar seguimiento y control de las actividades que realizan día a día dentro de las instalaciones del Laboratorio.

Campos: *btnGuardar, btnRegresar, components, Label1, Label2, lsbMódulo, lsbSemestre, pnlOpciones y wbrNavegador.*

Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmMovimientosXModuloYSemestre_Load, GeneraHTML, IntializeComponent, lsbMódulo_SelectedIndexChanged, lsbSemestre_SelectedIndexChanged y prepararListas.*

Vínculos: *clsGeneral, modFunciones y frmPrincipal.*

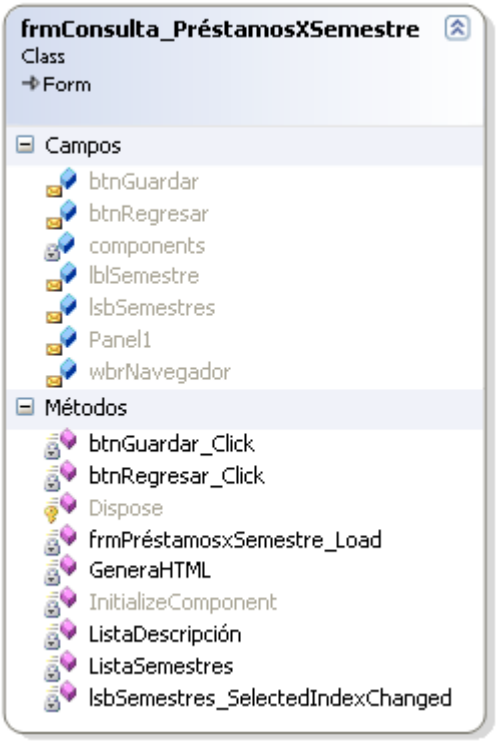
IV.3.31 CLASE frmConsulta_PréstamosXSemestre

Líneas	Archivo
281	frmConsulta-Préstamos por Semestre.vb

La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema una herramienta mediante la cual puedan consultar los distintos préstamos de materiales o equipos que forman parte de los bienes del Laboratorio y agrupados por semestre, el cual se deberá de elegir para generar la información de la consulta solicitada. Los resultados generados se muestran dentro del formulario en formato HTML permitiendo a los operadores del Sistema su análisis inmediato o posterior al ser guardados en alguna localidad del disco duro del Sistema, así como para su impresión u otras tareas deseadas. Esta herramienta se ha incluido para el apoyo en las tareas cotidianas de la administración del Laboratorio.

Campos: *btnGuardar, btnRegresar, components, lblSemestre, lsbSemestres, Panel1 y wbrNavegador.*

Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmPréstamosXSemestre_Load,*

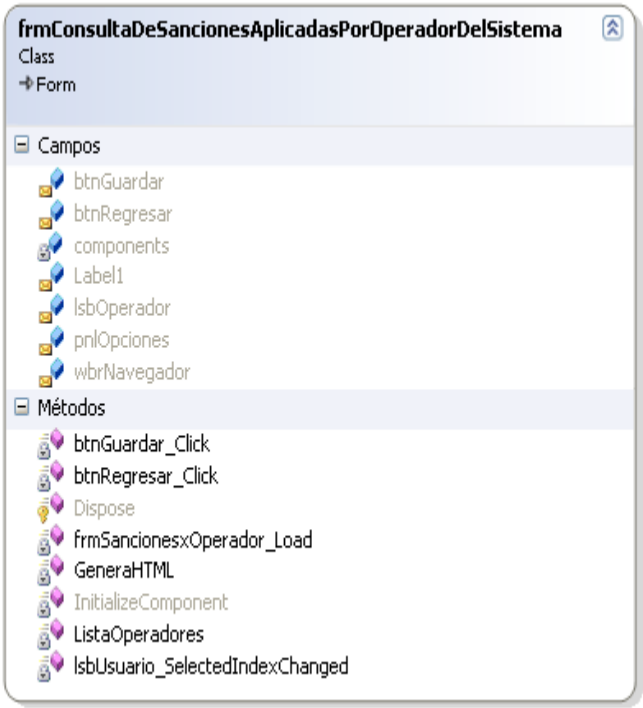


GeneraHTML, InitializeComponent, ListaDscripción, ListaSemestres, lsbSemestres_selectedIndexChanged.

Vínculos: *clsGeneral, modFunciones y frmPrincipal.*

IV.3.32 CLASE frmConsultaDeSancionesAplicadasPorOperadorDelSistema

Líneas	Archivo
241	frmConsulta-Sanciones por operador.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema, una herramienta de consulta de los movimientos que se han registrado sobre las sanciones que ha aplicado cada uno de los operadores del Sistema a los usuarios de los servicios del Laboratorio. La manera que se muestran los resultados de las consultas, es mediante el empleo de un contenedor de información en formato HTML, que permite a los operadores el análisis inmediato o posterior (si se guardan los resultados) para fines de seguimiento y control de las tareas administrativas sobre el Laboratorio.

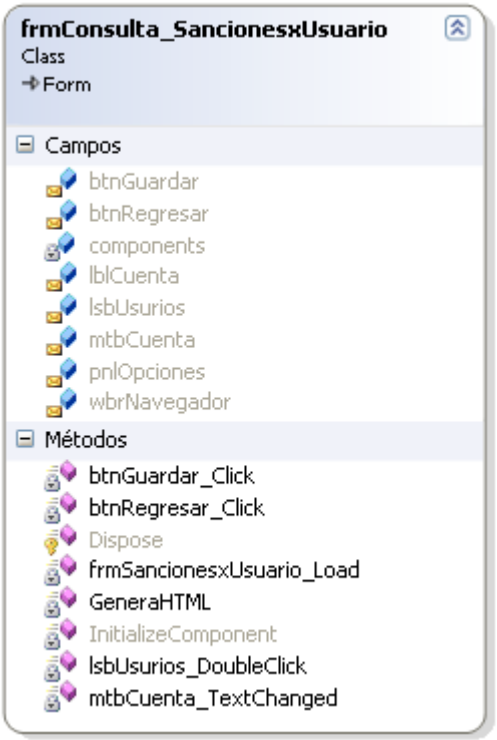
Campos: *btnGuardar, btnRegresar, components, Label1, lsbOperador, pnlOpciones y wbrNavegador.*

Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmSancionesXOperador_Load, GeneraHTML, InitializeComponent, ListaOperadores y lsbUsuario_SelectedIndexChanged.*

Vínculos: *clsGeneral, modFunciones y frmPrincipal.*

IV.3.33 CLASE frmConsulta_SancionesxUsuario

Líneas	Archivo
211	frmConsulta-Sanciones por Usuario.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema una herramienta que les permita consultar las sanciones que de manera histórica se han registrado en la operación del Sistema sobre alguno de los usuarios en particular de los servicios del Laboratorio Abierto.

El formulario proporciona los medios necesarios para que se seleccione al usuario que se desea consultar y una vez que se ha seleccionado, se mostrará en el formulario dentro de un contenedor de información en formato HTML, la información que se ha localizado.

Esto permite al operador hacer análisis inmediatos o si se desea guardar los resultados, para un análisis posterior o impresión posterior.

Campos: *btnGuardar, btnRegresar, components, lblCuenta, lblUsuarios, mtbCuenta, pnlOpciones y wbrNavegador.*

Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmSancionesxUsuario_Load, GeneraHTML, InitializeComponent, lsbUsuarios_DoubleClick y mtbCuenta_TextChanged.*

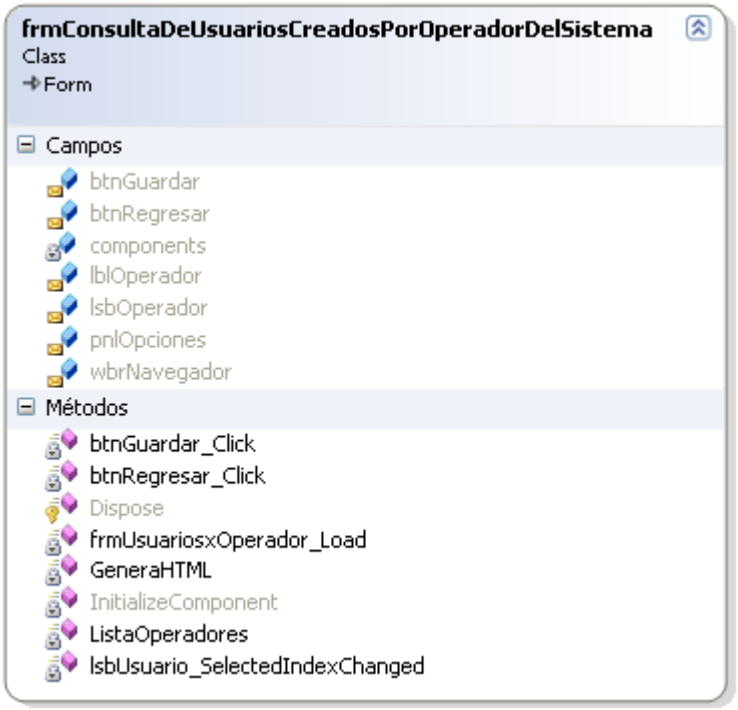
Vínculos: *clsGeneral, modFunciones, modUsuarios y frmPrincipal.*

IV.3.34 CLASE frmConsultaDeUsuariosCreadosPorOperadorDelSistema

Líneas	Archivo
225	frmConsulta-Usuarios creados por operador.vb

La finalidad de esta clase-formulario es la de proporcionar a los operadores del Sistema una herramienta de consulta para el análisis de la información registrada en el Sistema durante su operación ordinaria sobre los usuarios que han sido creados (ingresados al Sistema), clasificados por *Operador del Sistema*.

Los resultados que se generen a partir de la selección de las opciones que en el Sistema se solicitan, se mostraran en formato HTML permitiendo su análisis inmediato o guardarlos para su uso posterior.



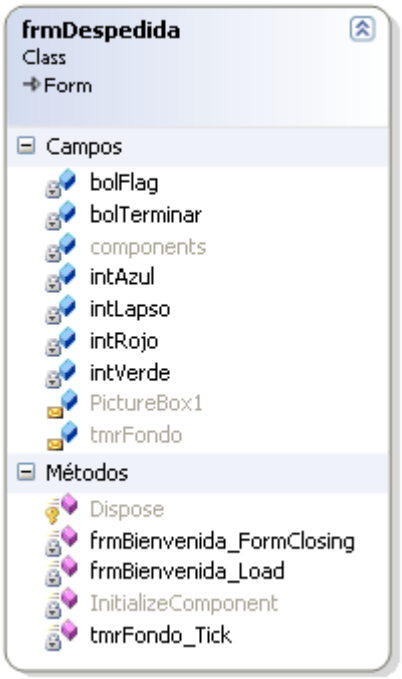
Campos: *btnGuardar, btnRegresar, components, lblOperador, lsbOperador, pnlOpciones y wbrNavegador.*

Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmUsuariosxOperador_Load, GeneraHTML, InitializeComponent, ListaOperadores y lsbUsuario_selecteIndexChanged.*

Vínculos: *clsGeneral, modFunciones y frmPrincipal.*

IV.3.35 CLASE frmDespedida

Líneas	Archivo
249	frmDespedida.vb



La finalidad de esta clase, es la de mostrar a los operadores del Sistema un mensaje de despedida al momento de cerrar las operaciones del Sistema. Esta despedida está compuesta por un escudo de la Facultad de Ingeniería, incluyendo los mensajes relacionados con la versión y derechos del Sistema.

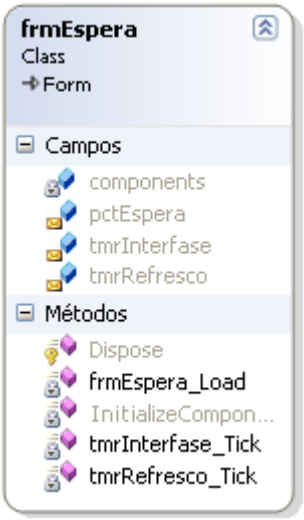
Campos: *bolFlg, bolTerminar, components, intAzul, intLapso, intRojo, intVerde, PictureBox1 y tmrFondo.*

Métodos: *Dispose, frmBienvenida_FrmClosing, frmBienvenida_Load, InitializeComponent y tmrFondo_Tick.*

Vínculos: *frmPrincipal.*

IV.3.36 CLASE frmEspera

Líneas	Archivo
52	frmEspera.vb



La finalidad de esta clase es la de ofrecer apoyo al Sistema para mostrar un indicador simbólico en forma de estrella giratoria en los momentos en que se pierda contacto con la *interfaz electrónica*. Si tras el paso de 10 segundos no hay respuesta, se ofrece al *Operador del Sistema* que seleccione entre intentar la re-conexión o cancelar todas las operaciones pendientes en el Sistema.

Campos: *components, pctEspera, tmrInterfaz y tmrRefresco.*

Métodos: *Dispose, frmEspera_Load, IntializeComponent, tmrInterfaz_Tick y tmrRefresco_Tick.*

Vínculos: *clsInterfase, modFunciones y frmPrincipal.*

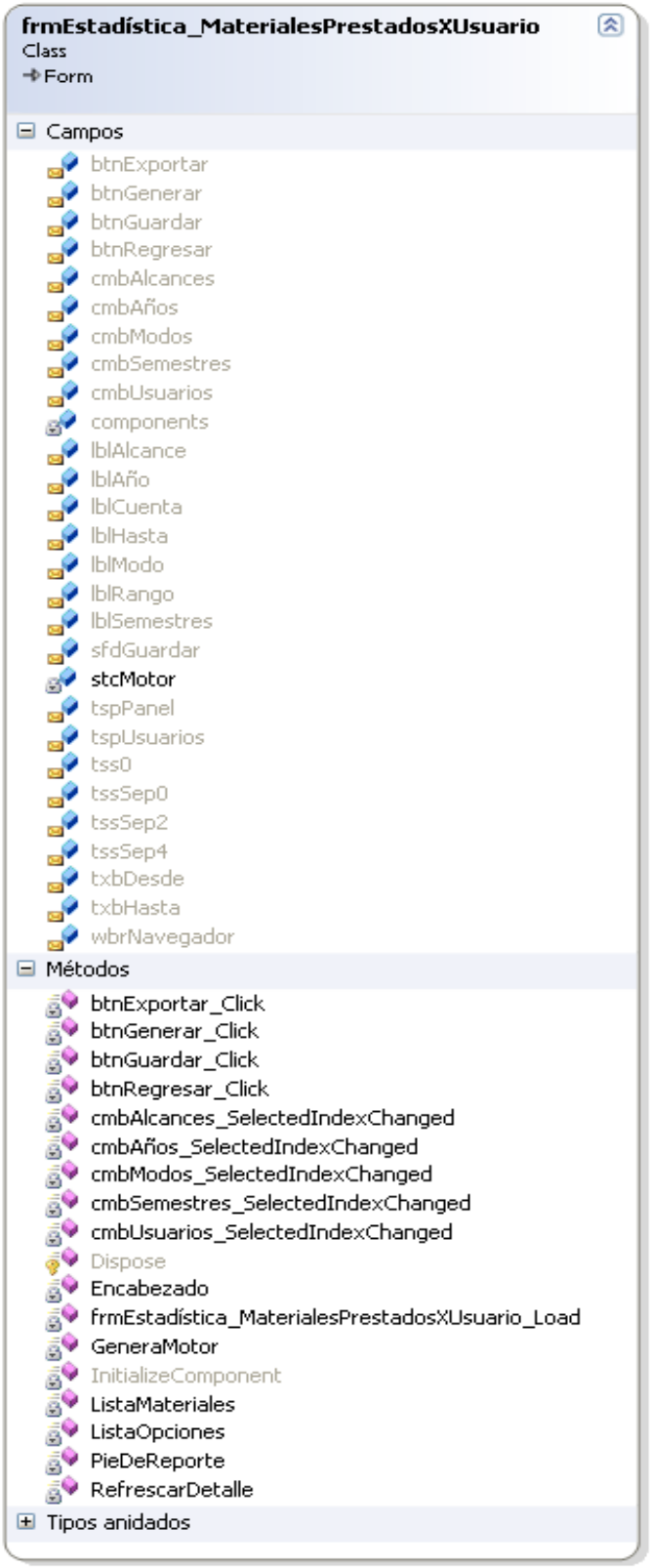
IV.3.37 CLASE frmEstadística_MaterialesPrestadosXUsuario

Líneas	Archivo
662	frmEstadísticas-Materiales prestados por usuario.vb

La finalidad de esta clase-formulario es la de permitir a los operadores del Sistema realizar consultas delimitadas de los materiales que se han prestado a los usuarios, ofreciéndole un mecanismo para realizar la selección de los usuarios que se hayan seleccionado.

Estas opciones son similares a las equivalencias de las consultas, con la diferencia de que permiten guardar los resultados en formato HTML o en un formato delimitado por comillas y comas (formato csv) con la finalidad de poder importar dicha información a otros programas como es el caso de Excel.

Campos: *btnExportarn, btnGenerar, btnGuardar, btnRegresar, cmbAlcances, cmbAños, cmbModos, cmbSemestres, cmbUsuarios, components, lblAlcance, lblAño, lblCuenta, lblHasta, lblModo, lblRango, lblSemestres, sfdGuardar, stcMotor, tspPanel, tspUsuarios, tss0, tssSep0, tssSep2, tssSep4, txbDesde, txbHasta y wbrNavegador.*

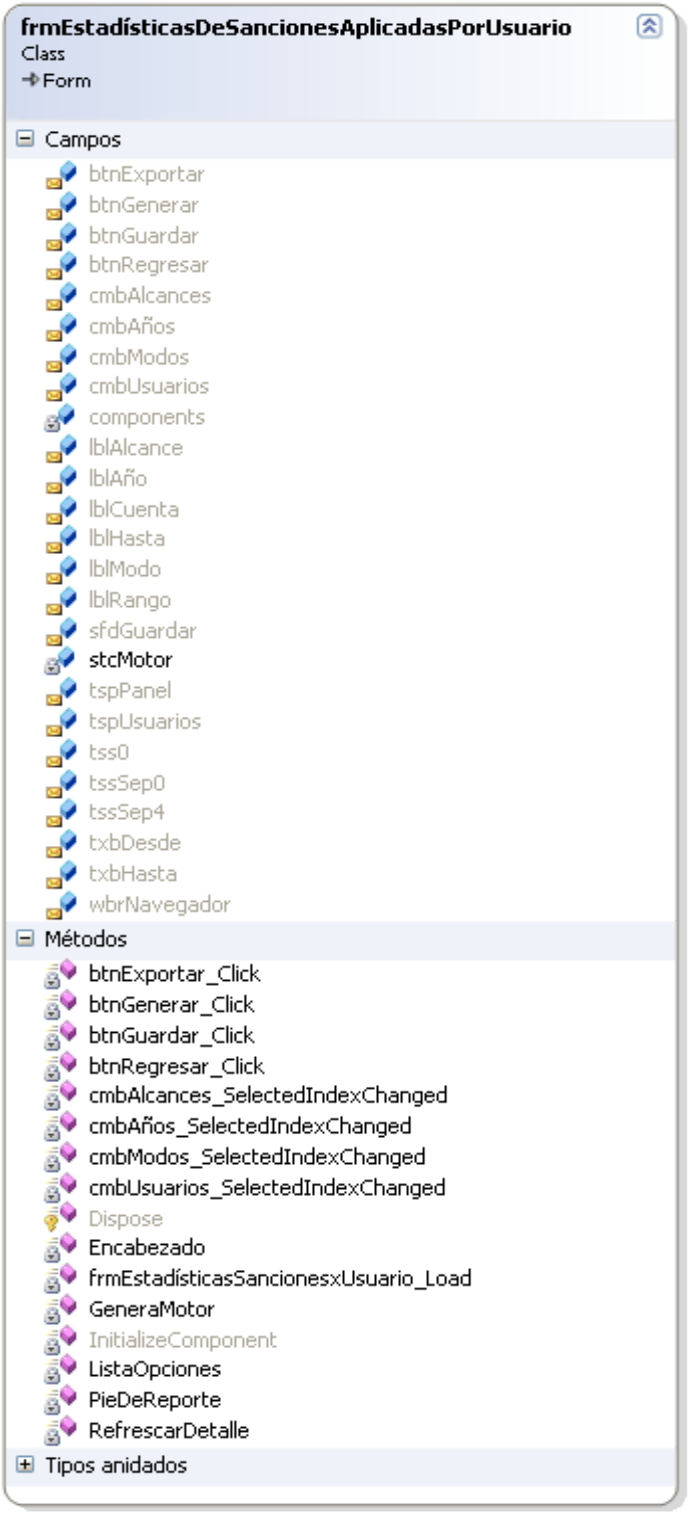


Métodos: *bthExportar_Click, btnGenerar_Click, btnGuardar_Click, btnGuardar_Click, btnRegresar_Click, cmbAlcances_SelectedIndexChanged, cmbAños_SelectedIndexChanged, cmbModos_SelectedIndexChanged, cmbSemetres_SelectedIndexChanged, cmbUsuarios_SelectedIndexChanged, Dispose, Encabezado, frmEstadística_MaterialesPrestadosXUsuario_Load, GenerarMotor, InitializeComponent, ListaMateriales, ListaOpciones, PieDeReporte y RefrescarDetalle.*

Vínculos: *clsGeneral, modFunciones, modUsuarios y frmPrincipal.*

IV.3.38 CLASE frmEstadísticas_MaterialesPrestados

Líneas	Archivo
963	frmEstadísticas-Materiales prestados.vb



La finalidad de esta clase es la de ofrecer a los operadores del Sistema una herramienta de consulta de estadísticas relacionadas con los movimientos registrados sobre los materiales prestados en sus distintas modalidades y combinaciones. Para esto, se ofrece al operador los mecanismos para acotar la consulta.

De manera complementaria a las clases de consulta, esta clase ofrece guardar los resultados en formatos delimitados por comillas y comas (csv) con la finalidad de poder utilizarla en programas externos como Excel.

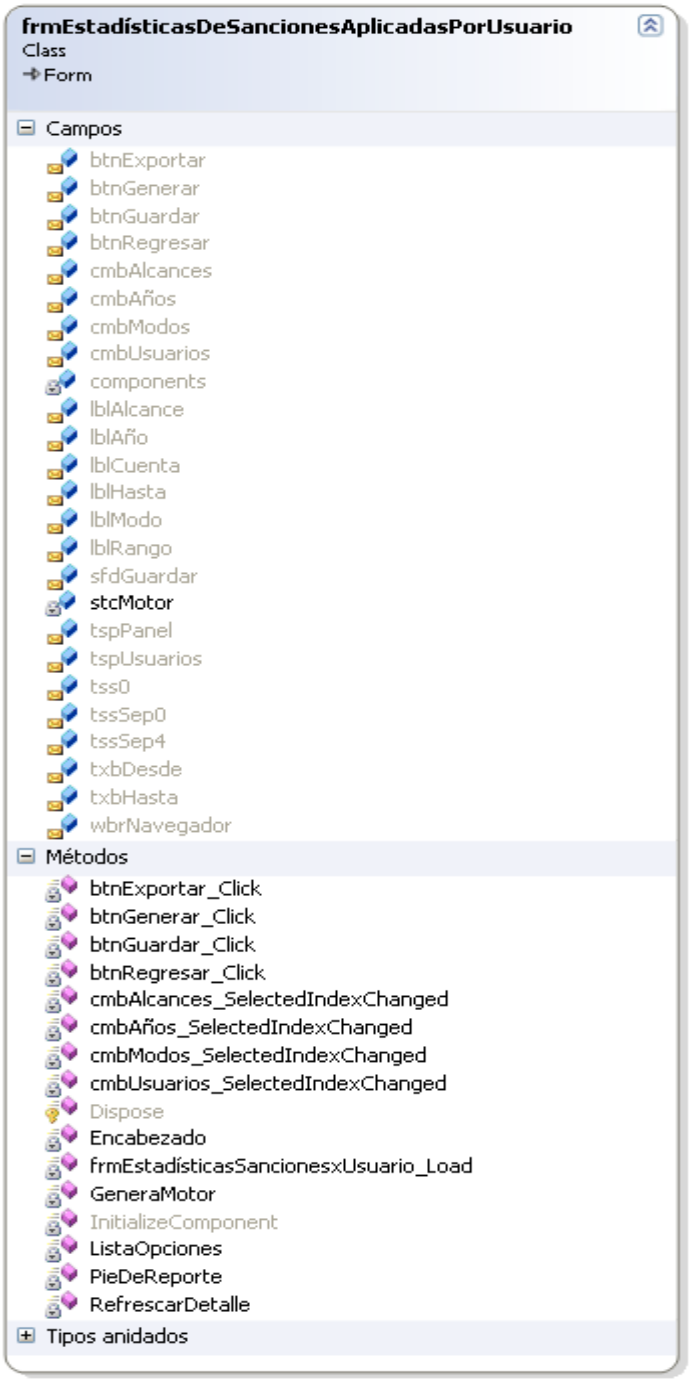
Campos: *btnExportarm btnGenerar, btnGuardar, btnRegresar, cmbAlcances, cmbAños, cmbCarreras, cmbModos, cmbSemestres, components, lblAlcance, lblCarreras, lblHasta, lblModo, lblRango, lblSemestres, sfdGuardar, stcMotor, ToolStripLabel1, tspPanel, tssSep0, tssSep1, tssSep2, tssSep4, txbDesde, txbHasta y wbrNavegador.*

Métodos: *AgregarMateriales, btnExportar_Click, btnGenerar_Click, btnGuardar_Click, btnRegresar_Click_1, cmbAlcances_SelectedIndexChanged, cmbAños_SelectedIndexChanged, cmbCarreras_SelectedChanged, cmbModos_SelectedIndexChanged, Dispose, Encabezado, frmEstadisticasPréstamoMateriaeles_Load, GeneraMotor, InformeGeneral, InformeUsuarios, InitializeComponent, ListaOperaciones y PierDeReporte, RefrescarDetalle.*

Vínculos: *clsGeneral, modFunciones y frmPrincipal.*

IV.3.39 CLASE frmEstadísticasDeSancionesAplicadasPorUsuario

Líneas	Archivo
587	frmEstadísticas-Sanciones aplicadas a usuarios.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema un mecanismo mediante el cual se puede consultar todos los movimientos relacionados por las sanciones que se han aplicado a los usuarios de manera personalizada e histórica para los fines que convengan en las tareas de administración del Laboratorio. Además, se permite exportar los datos generados a un archivo en formato delimitado por comillas y comas con la finalidad de poder ser utilizados desde otros programas como Excel.

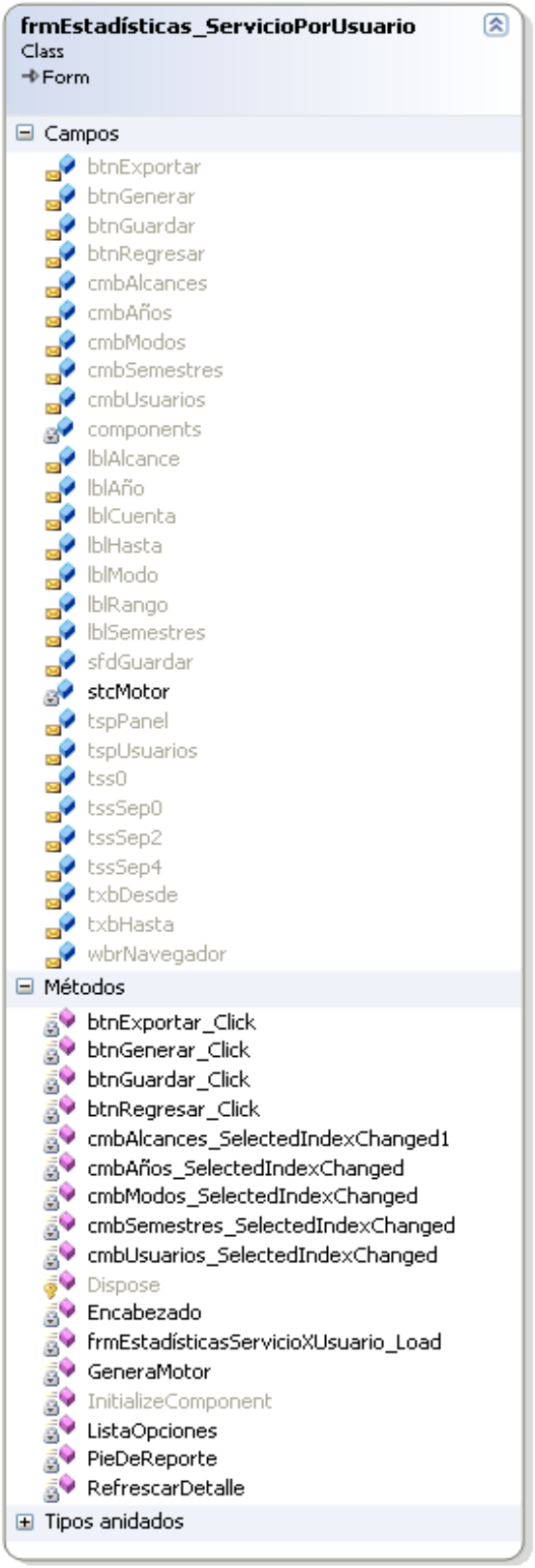
Campos: *btnExportar, btnGenerar, btnGuardar, btnRegresar, cmbAlcances, cmbAños, cmbModos, cmbUsuarios, components, lblAlcance, lblAño, lblCuenta, lblHasta, lblModo, lblRango, sfdGuardar, stcMotor, tspPanel, tspUsuarios, tss0, tssSep4, txbDesde, txbHasta y wbrNavegador.*

Métodos: *btnExportar_Click, btnGenerar_Click, btnGuardar_Click, btnRegresar_Click, cmbAlcances_SelectedIndexChanged, cmbAños_SelectedIndexChanged,, cmbModos_SelectedIndexChanged, cmbUsuarios_SelectedIndexChanged, Dispose, Encabezado, frmEstadísticasSancionesxUsuario_Load, GeneraMotor, InitializeComponent, ListaOperaciones, PieDeReporte y RefrescarDetalle.*

Vínculos: *clsGeneral, modFunciones, modUsuarios y frmPrincipal.*

IV.3.40 CLASE frmEstadísticas_ServicioPorUsuario

Líneas	Archivo
686	frmEstadísticas-Servicio proporcionado por usuario.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema una herramienta mediante la cual se pueda consultar de manera selectiva y a manera de estadísticas, los servicios que se le han proporcionado a un usuario del Sistema según se halla registrado en las operaciones normales del Sistema. Al igual que las demás opciones estadísticas, proporciona al usuario la posibilidad de exportar en formato delimitado por comas y comillas (formato .csv) para que pueda se utilizada esta información en otras aplicaciones, por ejemplo, Excel.

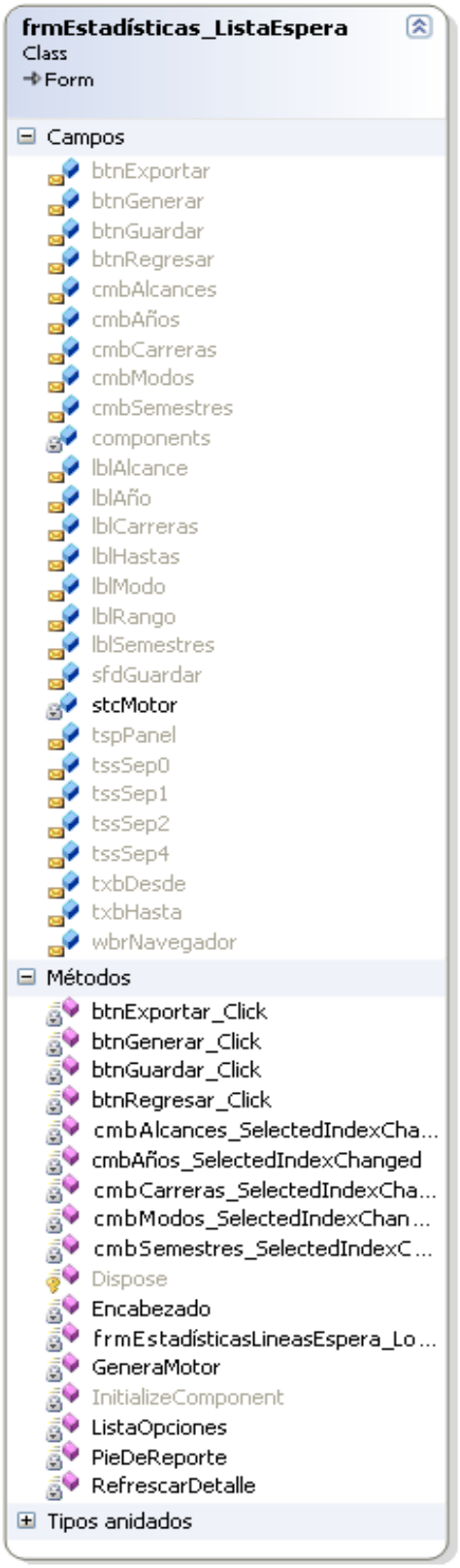
Campos: *btnExportar, btnGenerar, btnGuardar, btnRegresar, cmbAlcances, cmbAños, cmbModos, cmbUsuarios, components, lblAlcance, lblAño, lblCuenta, lblHasta, lblModo, lblRango, sfdGuardar, stcMotor, tspPanel, tspUsuarios, tss0, tssSep0, tssSep4, txbDesde, txbHasta y wbrNavegador.*

Métodos: *btnExportar_Click, btnGenerar_Click, btnGuardar_Click, btnRegresar_Click, cmbAlcances_SelectedIndexChanged, cmbAños_SeletedIndexChanged, cmbModos_SelectedIndexChanged, cmbUsuarios_SelectedIndexChanged, Dispose, Encabezado, frmEstadísticasSancionesxUsuario_Load, GeneraMotor, InitializeComponent, ListaOpciones, PieDeReporte, RefrescarDetalle.*

Vínculos: *clsGeneral, modFunciones, modUsuarios y frmPrincipal.*

IV.3.41 CLASE frmEstadísticas_ListaEspera

Líneas	Archivo
740	frmEstadísticas-Tiempos en la lista de espera.vb



La finalidad de esta clase-formulario es la de brindar una herramienta de administración a los operadores del Sistema, en la que se detallan los movimientos que se han registrado en la lista de espera. La consulta se puede acotar de acuerdo a las necesidades del operador. Además de arrojar los datos estadísticos según la modalidad elegida, permite guardar a dichos datos en algún archivo en formato delimitado por comas y comillas (csv) para que puedan ser reutilizados en otras aplicaciones como Excel, además de poder guardarlos en forma HTML como ya se indicó en el caso de las consultas.

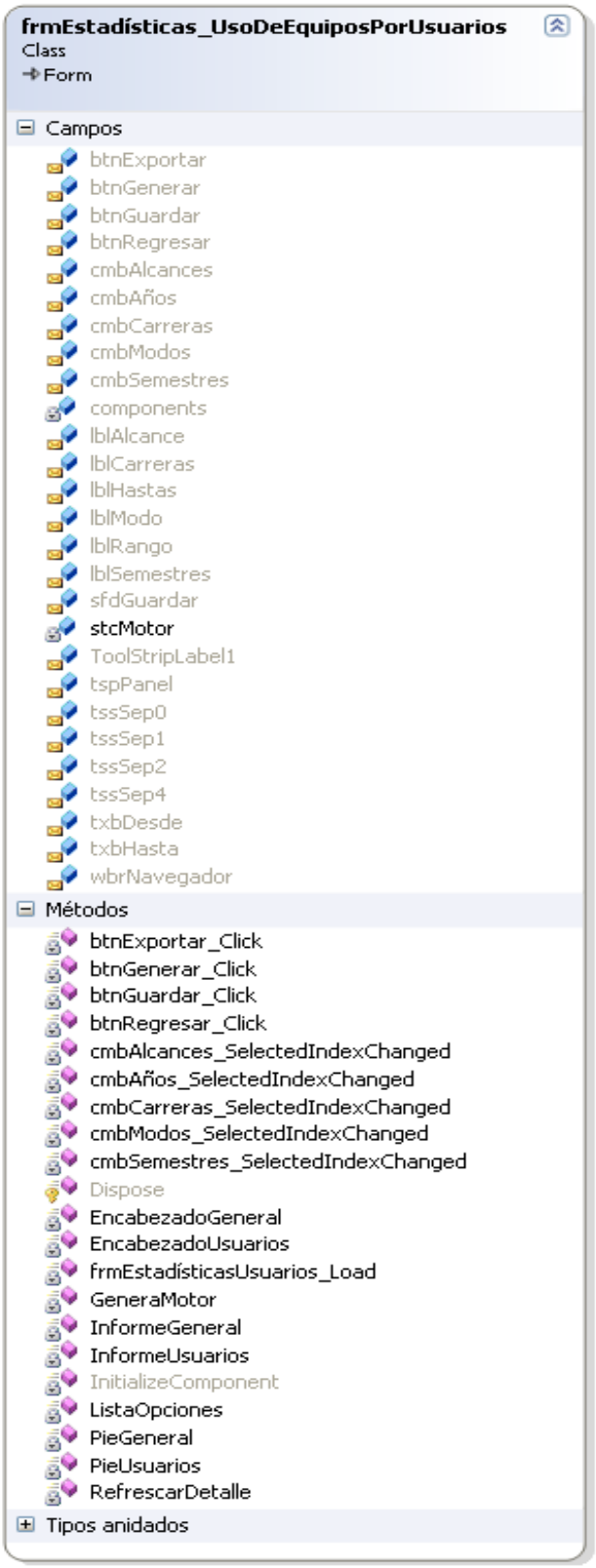
Campos: *btnExportar, btnGenerar, btnGuardar, btnRegresar, cmbAlcances, cmbAños, cmbModos, cmbSemestres, cmbUsuarios, components, lblAlcance, lblAño, lblCuenta, lblHasta, lblModo, lblRango, lblSemestres, sfdGuardar, stcMotor, tspPanel, tspUsuarios, tss0, tssSep0, tssSep2, tssSep4, txbDesde, txbHasta y wbrNavegador.*

Métodos: *btnExportar_Click, btnGenerar, btnGuardar_Click, btnRegresar_Click, cmbAlcancesSelectedIndexChanged, cmbAñosSelectedIndexChanged, cmbCarrerasSelectedIndexChanged, cmbModosSelectedIndexChanged, cmbSemestresSelectedIndexChanged, Dispose, Encabezado, frmEstadísticasLineasEspera_load, GeneraMotor, InitializeComponent, ListaOpciones, PieDeReporte y RefrescarDetalle.*

Vínculos: *clsGeneral, modFunciones y frmPrincipal.*

IV.3.42 CLASE frmEstadísticas_UsoDeEquiposPorUsuarios

Líneas	Archivo
976	frmEstadísticas-Uso de equipos por usuario.vb



Esta clase-formulario ofrece a los operadores del Sistema, una opción mediante la cual se puedan consultar las estadísticas sobre el uso que se les ha dado a los equipos del Laboratorio por parte de los usuarios que de manera histórica se van registrando para los fines administrativos de este Sistema. Esta opción permite además de guardar los resultados generados; en formato HTML, también en un formato de delimitación de textos por comas y comillas (csv) para su uso en otras aplicaciones como es el caso de Excel.

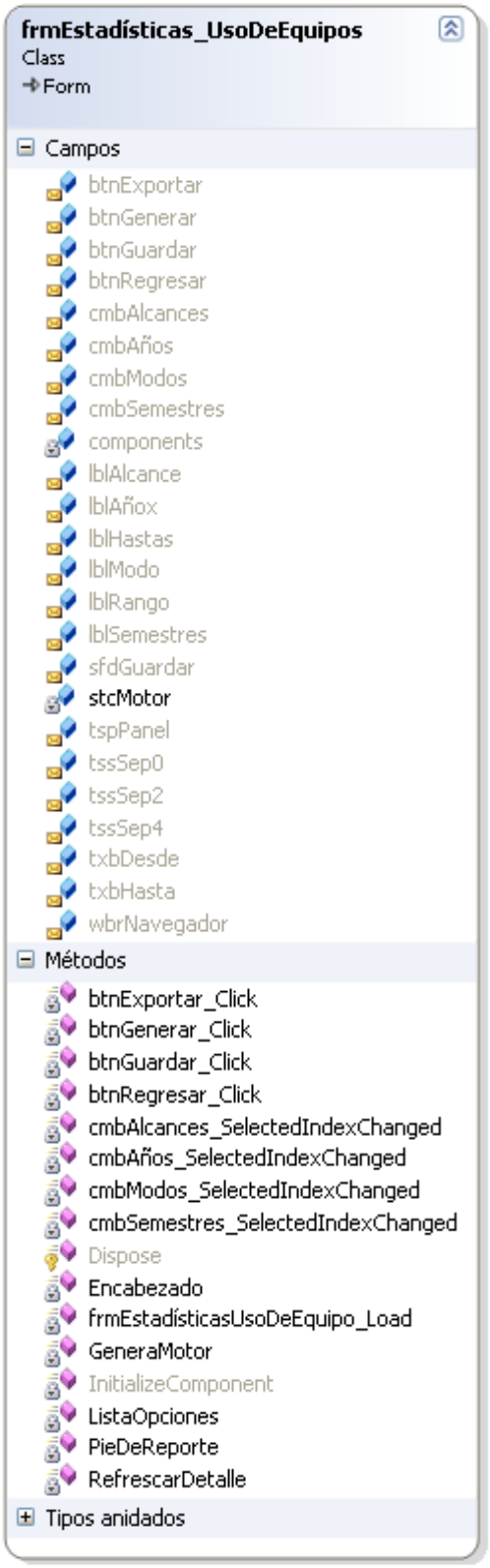
Campos: *btnExportar, btnGenerar, btnGuardar, btnRegresar, cmbAños, cmbCarreras, cmbModos, cmbSemestres, contents, lblAlcance, lblCarreras, lblHastas, lblModo, lblRango, lblSemestres, sfdGuardar, stcMotor, ToolStripLabel1, tspPanel, tssSep0, tssSep1, tssSep2, tssSep4, txbDesde, txbHasta y wbrNavegador.*

Métodos: *btnExportar_Click, btnGenerar_Click, btnGuardar_Click, btnRegresar_Click, cmbAlcances_SelectedIndexChanged, cmbAños_SelectedIndexChanged, cmbCarreras_SelectedIndexChanged, cmbModos_SelectedIndexChanged, cmbCarreras_SelectedIndexChanged, cmbSemestres_SelectedIndexChanged, Dispose, EncabezadoGeneral, EncabezadoUsuarios, frmEstadísticasUsuarios_Load, GeneraMotor, InformeGeneral, InformeUsuarios, InitializeComponent, ListaOpciones, PieGeneral, PieUsuarios y RefrescarDetalle.*

Vínculos: *clsGeneral, modFunciones y frmPrincipal.*

IV.3.43 CLASE frmEstadísticas_UsoDeEquipos

Líneas	Archivo
659	frmEstadísticas-Uso de los equipos.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema una herramienta de consulta de estadísticas sobre el uso de los equipos, misma que se acotará de acuerdo a los parámetros que ofrece el Sistema. Los resultados que se generen se visualizan primeramente en el formulario de esta clase según el formato HTML, permitiendo guardar su contenido para posteriores referencias o consultas, así como la posibilidad de exportar en formato delimitado de comas y comillas (csv) para poder ser utilizado en otras aplicaciones como Excel.

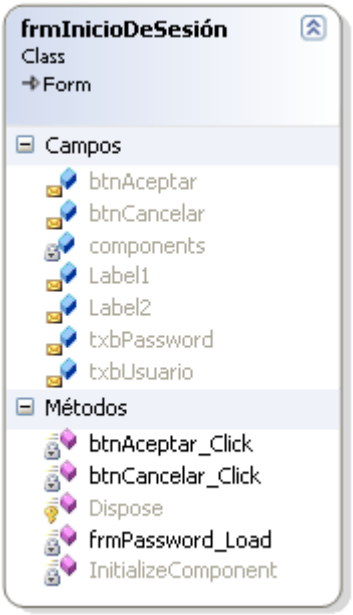
Campos: *btnExportar, btnGenerar, btnGuardar, btnRegresar, cmbAlcances, cmbAños, cmbMods, cmbSemestres, components, lblAlcance, lblAñox, lblHastas, lblModo, lblRango, lblSemestre, sfdGuardar, stcMotor, tspPanel, tssSep0, tssSep2, tssSep4, txbDesde, txbHasta y wbrNavegador.*

Métodos: *btnExportar_Click, btnGenerar_Click, btnGuardar_Click, btnRegresar_Click, cmbAlcances_SelectedIndexChanged, cmbAños_SelectedIndexChanged, cmbModos_SelecedIndexChanged, cmbSemestres_SelectedIndexChanged, Dispose, Encabezado, frmEstadisticasUsoDeEquipo_Load, GeneraMotor, InitializeComponent, ListaOpciones, PieDeReporte y RefrescarDetalle.*

Vínculos: *clsGeneral, modFunciones y frmPrincipal.*

IV.3.44 CLASE frmInicioDeSesión

Líneas	Archivo
127	frmInicioDeSesión.vb



La finalidad de esta clase-formulario es el de servir como la ventana de identificación de credenciales; es decir, es donde se solicita al usuario que introduzca el nombre de su cuenta de operador y su contraseña para que pueda operar al Sistema.

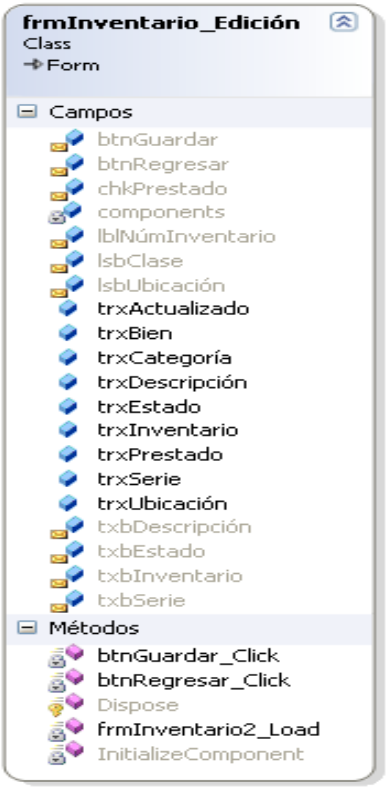
Campos: *btnAceptar, btnCancelar, components, Label1, Label2, txbPassword y txbUsuario.*

Métodos: *btnAceptar_Click, btnCancelar_Click, Dispose, frmPassword_Load e InitializeComponent.*

Vínculos: *clsGeneral, clsParametros, clsSeguridad, clsTransacciones, modFunciones y modInicio.*

IV.3.45 CLASE frmInventario_Edición

Líneas	Archivo
38	frmInventario-Edición.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema un mecanismo desde el cual se pueden actualizar los datos de alguno de los bienes del Laboratorio, para actualizar datos como la ubicación en donde se encuentra el bien, la categoría, las condiciones en que se encuentra, etc. El único valor que no puede cambiar es el número de bien registrado dentro de los registros del **SAEALABFI.**

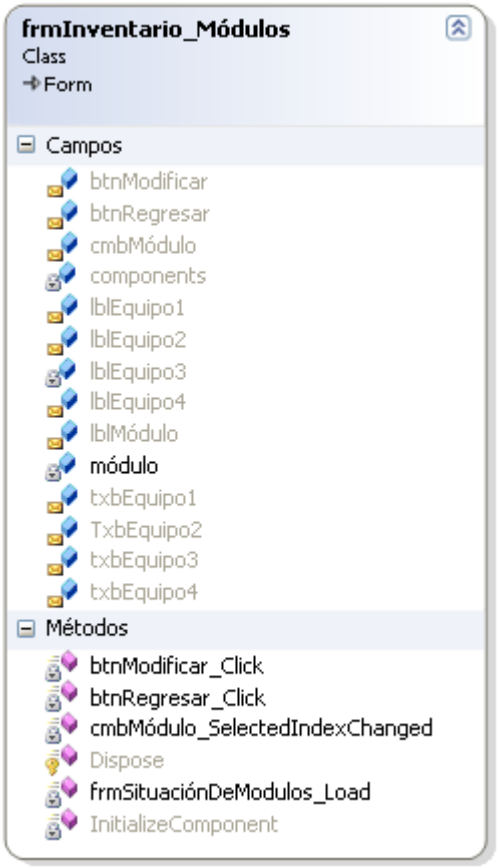
Campos: *btnGuardar, btnRegresar, chkPrestado, components, lblNumInventario, lsbClase, lsbUbicación, trxActualizado, trxBien, trxCategoría, trxDescripción, trxEstado, trxInventario, trxPrestado, trxSerie, trxUbicación, txbDescripción, txbEstado, txbInventario y txbSerie.*

Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmInventario2_Load e InitializeComponent.*

Vínculos: *modFuncione, frmInventario_ConsultaDeLosMovimientos*

IV.3.46 CLASE frmInventario_Módulos

Líneas	Archivo
56	frmInventario-Módulos.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores una manera en que se actualice de manera particular, la situación que guardan cada uno de los componentes de los módulos instalados en el Laboratorio. Mediante una lista desplegable se selecciona al módulo deseado y dentro de la casilla de selección-escritura se escribirá las condiciones en que se mantiene el equipo en particular.

Campos: *btnModificar, btnRegresar, cmbMódulo, components, lblEquipo1, lblEquipo2, lblEquipo3, lblEquipo4, lblMódulo, módulo, txbEquipo1, txbEquipo2, txbEquipo3 y txbEquipo4.*

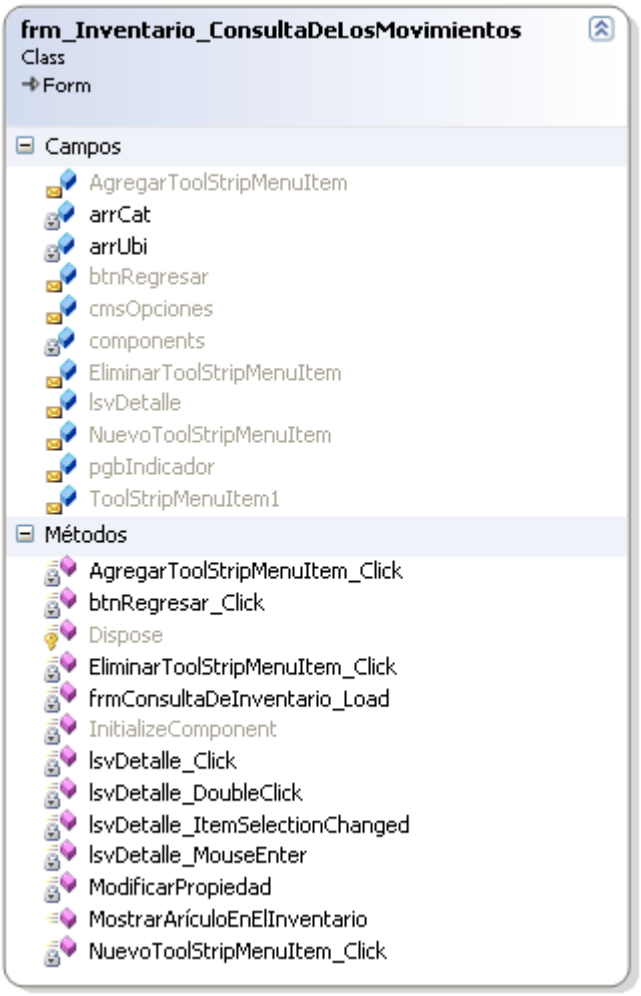
Métodos: *btnModificar_Click, btnRegresar_Click, cmbMódulo_SelectedIndexChanged, Dispose, frmSituaciónDeModulos_Load e InitializeComponent.*

Vínculos: *clsModulo, clsParametros, modFunciones y frmPrincipal.*

IV.3.47 CLASE frm_Inventario_ConsultaDeLosMovimientos

Líneas	Archivo
398	frmInventario-Movimientos.vb

Esta clase-formulario ofrece a los operadores del Sistema una manera rápida de acceder a todas las operaciones que se han incorporado para el manejo de la información en que se ha estructurado el registro de los bienes del inventario del Laboratorio y mediante el uso de menús contextuales se puede realizar tareas de agregar bienes nuevos, de editar los existentes o de eliminarlos según corresponda el caso.



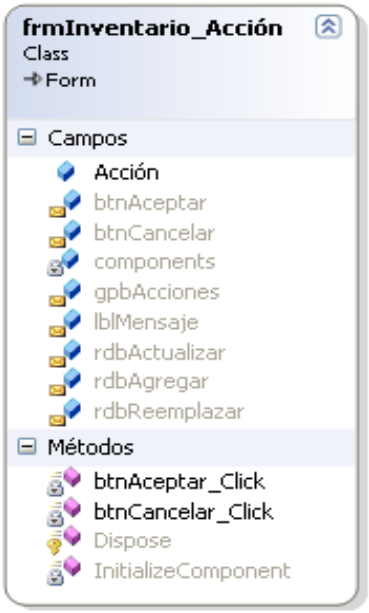
Campos: *AgregarToolStripMenuItem, arrCat, arrUbi, btnRegresar, cmsOpciones, components, EliminarToolStripMenuItem, lsvDetalle, NuevoToolStripMenuItem, pgbIndicador, ToolStripMenuItem1.*

Métodos: *AgregarToolStripMenuItem_Click, btnRegresar_Click, Dispose, EliminarToolStripMenuItem_Click, frmConsultaDeInventario_Load, InitializeComponent, lsvDetalle_Click, lsvDetalle_DoubleClik, lsvDetalle_ItemSelectionChanged, lsvDetalle_MouseEnter, ModificarPropiedad, MostrarArticuloEnElInventario y NuevoToolStripMenuItem_Click.*

Vínculos: *clsGeneral, clsParametros, clsTransacción, modFunciones, modInventario, frmInventario_Edición, frm_Inventario_ConsultaDeLosMovimientos y frmPrincipal*

IV.3.48 CLASE frmInventario_Acción

Líneas	Archivo
24	frmInventario_Acción.vb



Esta clase-formulario es auxiliar en el proceso de importación de información del detalle de descripción de los bienes del Laboratorio. Dentro de las opciones que aquí se ofrecen se encuentran: Actualizar, Agregar o Reemplazar.

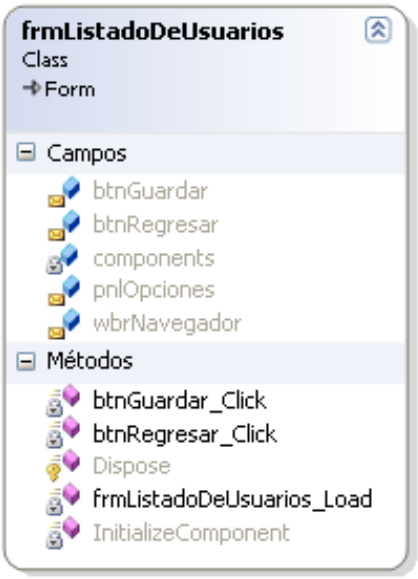
Campos: *Acción, btnAceptar, btnCancelar, components, gbpAcciones, lbMensaje, rdbActualizar, rdbAgregar y rdbReemplazar.*

Métodos: *btnAceptar_Click, btnCancelar_Click, Dispose e InitializeComponent.*

Vínculos: *modFunciones y modInventario.*

IV.3.49 CLASE frmListadoDeUsuarios

Líneas	Archivo
95	frmListadoDeUsuarios.vb



La finalidad de esta clase-formulario es la de ofrecer a los operadores del Sistema una opción desde la cual se pueda saber los usuarios que se encuentran dentro de las instalaciones del Laboratorio. La consulta agrupa por módulo a los usuarios indicando cuantos de ellos se encuentran en los módulos y en seguida un los números de cuenta y nombres de los usuarios.

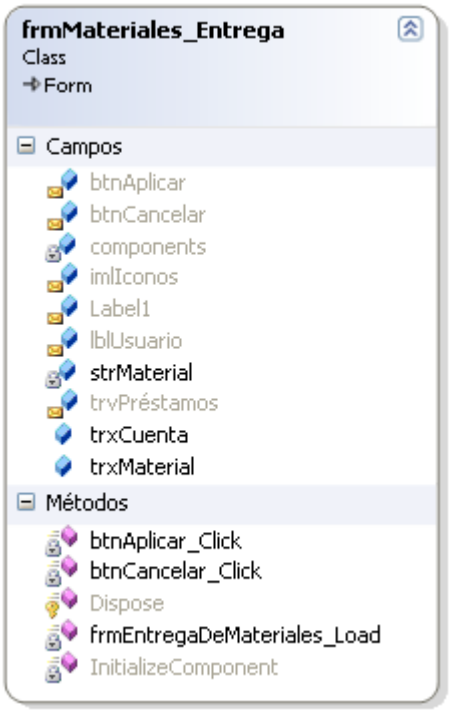
Campos: *btnGuardar, btnRegresar, components, pnlOpciones y wbrNavegar.*

Métodos: *btnGuardar_Click, btnRegresar_Click, Dispose, frmListadoDeUsuarios_Load e InitializeComponent.*

Vínculos: *clsGeneral, clsModulo, clsParametros, modFunciones, modUsuarios y frmPrincipal.*

IV.3.50 CLASE frmMateriales_Entrega

Líneas	Archivo
92	frmMateriales-Entrega.vb



La finalidad de esta clase-formulario se utiliza como un mecanismo auxiliar mediante el cual los usuarios devuelven los materiales que les han sido prestados. Mediante un control de árbol de vista, se colocaran cada uno de los materiales y de ellos se podrá seleccionar cual o cuales de ellos se devuelven al aplicar los cambios para que surjan efecto.

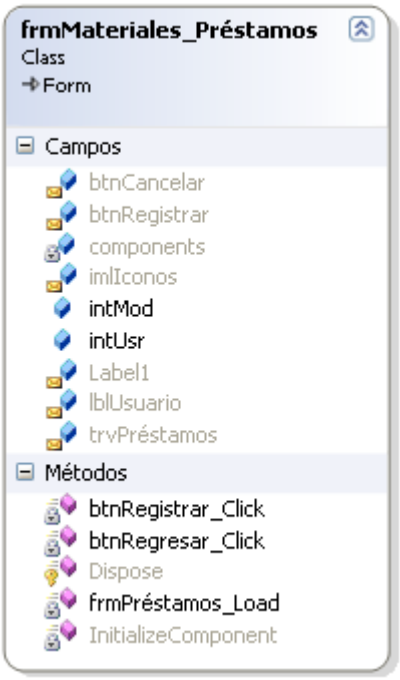
Campos: *btnAplicar, btnCancelar, components, imlIconos, Label1, lblUsuario, strMaterial, trvPréstamos, trxCuenta y trxMaterial.*

Métodos: *btnAplicar_Click, btnCancelar_Click, Dispose, frmEntregaDeMateriales_Load e InitializeComponent.*

Vínculos: *clsGeneral, modFunciones, modInventario y modUsuarios.*

IV.3.51 CLASE frmMateriales_Préstamos

Líneas	Archivo
175	frmMateriales-Préstamos.vb



La finalidad de esta clase-formulario es la de auxiliar al Sistema en las tareas de préstamo de materiales a los usuarios del Laboratorio. Se basa en un control de árbol de vista donde se colocan los íconos de grupo de materiales. En esta versión del Sistema se han considerado las categorías: Herramientas, Materiales y Manuales y mediante la expansión de las ramas del árbol se mostraran a cada uno de los bienes registrados y clasificados bajo ese grupo y serán asociadas al usuario al que se les adjudicará en calidad de préstamo.

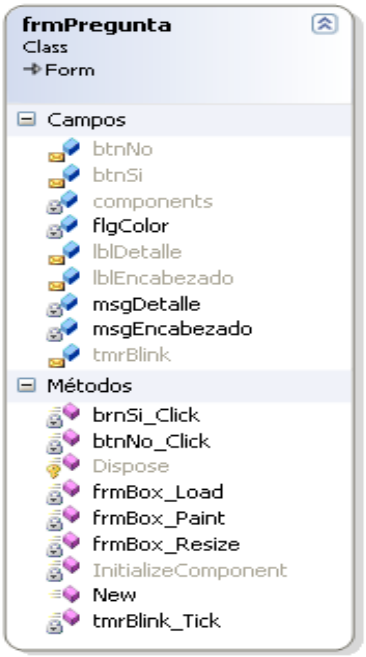
Campos: *btnCancelar, btnRegresar, components, imlIconos, intMod, intUsr, Label1, lblUsuario y trvPréstamos.*

Métodos: *btnRegistrar_Click, btnRegresar_Click, Dispose, frmPréstamos_Load e InitializeComponent.*

Vínculos: *clsGeneral, clsMódulo, clsParámetros, clsTransacción, modFunciones, modInventario y modUsuarios.*

IV.3.52 CLASE frmPregunta

Líneas	Archivo
103	frmPregunta.vb



El propósito de esta clase-formulario, es el de apoyar al Sistema para interactuar con el *Operador del Sistema* en las situaciones en que se requiere pedir que confirme o rechace las acciones que le son sugeridas ante situaciones como errores o consideraciones que implican la toma de decisiones.

Campos: *btnNo, btnSi, components, flgColor, lblDetalle, lblEncabezado, msgDetalle, msgEncabezado y tmrBlink.*

Métodos: *btnSi_Click, btnNo_Click, Dispose, frmBox_Load, frmBox_Paint, frmBox_Resize, InitializeComponent, New y tmrBlink_Tick.*

Vínculos: *modFunciones.*

IV.3.53 CLASE frmPrincipal

Líneas	Archivo
1,837	frmPrincipal.vb

Esta es la clase-formulario que conforma el núcleo de operaciones de todo el Sistema, en su composición se encuentra un objeto de vista de lista en la que se encuentran simbolizados todos los módulos instalados en el Laboratorio. Cuenta con otras tres vistas de lista que se utilizan para administrar los tiempos próximos a vencer por los usuarios de los módulos, una para los materiales pendientes por entregar y una más para la lista de usuarios que se encuentran en espera cuando las instalaciones están ocupadas a su máxima capacidad.

Desde aquí se han construido todos los menús de opciones directas agrupadas en las categorías: *Archivos, Consulta, Estadísticas, Usuarios, Administración, Inventario, Ayuda y Acerca De…*

Además, se encuentran todos los elementos en los que se identifica a la aplicación, al *Operador del Sistema*, el Semestre que se está cursando y el estatus que guardan los módulos utilizados.

Complementando las opciones del Sistema, se han construído menús contextuales que se asocian a los módulos dentro de la vista de lista en la que son simulados y de manera complementaria a todas las características del Sistema.

También se encuentran los códigos necesarios para poder incluir en la barra de tareas del Sistema Operativo un mecanismo de minimización de la aplicación, dotándolo de las opciones más frecuentes y con éste, se encuentran los mecanismos de avisos en la barra de tareas del Sistema Operativo sobre las tareas más comunes como lo son el encendido o apagado de los módulos.

Por las dimensiones de esta clase-formulario, su esquema de composición será mostrado el esquema en dos grandes grupos: uno para los ***Campos*** y uno para los Métodos ya que en su tamaño normal no se puede mostrar de manera completa a sus componentes.

En las siguientes páginas, se muestran a manera de tabla en dos columnas y dos renglones que abarcan dos hojas en forma seccionada a cada una de las secciones que componen la lista de ***Campos*** y en seguida se comienza a mostrar la lista de éstos. Posteriormente, de manera similar, se realiza el seccionamiento en tres bloques (en una hoja a las dos secciones iniciales, en un tramo final la otra) que se utilizan para mostrar la distribución de los ***Métodos*** que serán seguidos por la distribución de cada uno de éstos.

frmPrincipal
Class
Form

Campos

- bolTerminar
- btnCerrar
- btnInterfase
- btnMaximizar
- btnMinimizar
- btnNormal
- cmnCronómetro
- cmnEspera
- cmnMenu
- cmuMateriales
- cnmTray
- components
- dmuEntrega1
- dmuEntrega2
- dmuEntrega3
- dmuEntrega4
- dmuLiberar1
- dmuLiberar2
- dmuLiberar3
- dmuLiberar4
- dmuPréstamo1
- dmuPréstamo2
- dmuPréstamo3
- dmuPréstamo4
- enmStat
- imlCronómetro
- imlMódulos
- intAlto
- intAncho
- lblBusqueda
- lblCategoría
- lblDummy
- lblEspera
- lblEtiquetaBuscar
- lblEtiquetaCategorías
- lblEtiquetaMóduloActivo
- lblEtiquetaSemestre
- lblEtiquetaUsuario
- lblEtiquetUsuarios
- lblMódulo
- lblMódulos
- lblPrestados
- lblReloj
- lblSemestre
- lblStatus
- lblUsuario
- lblUsuarios
- lblVencimiento
- lsvCronómetro
- lsvEspera
- lsvPanel
- lsvPréstamos
- mnu15Minutos
- mnu30minutos
- mnu45minutos
- mnu60minutos
- mnuAcercaDe
- mnuActivarCuenta
- mnuActualizarDatosUsuario
- mnuAdministración
- mnuAdministraciónCuentas
- mnuAdministrarActualizarInterfase
- mnuAdministrarPanel
- mnuAgregarUsuario
- mnuAjusteSeguridad
- mnuAlumnosQueUsaronServicio
- mnuApagarMódulos
- mnuArchivo
- mnuArchivoTransacciones
- mnuAyuda
- mnuBloquearCuenta
- mnuBorrarCuenta
- mnuCambiarContraseña
- mnuConfigurar
- mnuConsultas
- mnuCrearCuenta
- mnuDatosActualizadosXOperador
- mnuEfectos
- mnuEspaciosDisponibles
- mnuEstadísticas
- mnuEstadoMódulos
- mnuEstSancionesPorUsuario
- mnuEstSancionesXUsuario
- mnuExportarInventario
- mnuImportarInventario
- mnuInventario
- mnuLiberarPanel
- mnuLíneaTiempoEspera
- mnuMaterialesXUsuario
- mnuMovsInventario
- mnuMovsLíneaEspera
- mnuMovsMódulo
- mnuMovsUsuarios
- mnuMovsXMóduloXSemestre
- mnuMovsXMóduloXSemXUsuario
- mnuMovsXSemestre
- mnuPréstamoMateriales
- mnuPréstamoMatXSemestre
- mnuPréstamosMatXUsuario
- mnuPrincipal
- mnuQuitarSanciones
- mnuRaya0
- mnuRaya1
- mnuRaya2
- mnuRestauraciónSistema
- mnuSalir
- mnuSanciones
- mnuSancionesXOperador

mnuSancionesXUsuario
mnuSep0
mnuSep01
mnuSep1
mnuSep2
mnuSep3
mnuSep4
mnuSep5
mnuSep6
mnuServicioXUsuario
mnuTransDesdeCero
mnuTransPosiciónActual
mnuUsoDeEquipos
mnuUsrsCreadosXOperador
mnuUsuarios
mnuUsuariosEnSesión
mnuUsuariosExportar
mnuUsuariosImportar
mnuxAdministrarPanel
mnuxAgregar
mnuxApagarMódulo
mnuxAsignarMódulo
mnuxBuscarEspacios
mnuxEncenderMódulo
mnuxEntregadeMateriales
mnuxEntregar
mnuxLiberarMódulo
mnuxLiberarPanel
mnuxLiberarUsuario
mnuxPréstamodeMateriales
mnuxQuitar
mnuxSep0
mnuxSep1
mnuxSep2
mnuxSep3
mnuxUsuarioNuevo
noiSAEALABFI
pgrBar
stbStatus
tlbUsuario
tlsBusqueda
tlsEtiqueta
tlsMensajes
tmrCronómetro
tmrEspera
tmrInterfase
tmrReloj
tmrVencimiento
tmsiSalir
ToolStripMenuItem1
ToolStripMenuItem16
ToolStripMenuItem2
ToolStripSeparator1
trayAcercaDe
trayActivarCuenta
trayActualizar
trayAdministración

trayAgregar
trayBloquearCuenta
trayBorrarCuenta
trayConfigurar
trayCrearCuenta
trayEfectos
trayEspaciosDisponibles
trayExportarInventario
trayExportarUsuarios
trayImportarInventario
trayImportarUsuarios
trayInventario
trayLiberarPanel
trayListaDeUsuarios
trayMódulos
trayMovimientos
trayQuitarSanciones
traySalir
traySanciones
traySep0
traySep1
trayUsuarios
tryArchivo
tryContraseñas
tryTransacciones
tspOcupado
tsSep0
tsSep1
tsSep2
ttpControl
txtBuscar
Métodos

Campos: *bolTerminar, btnCancelar, btnInterfaz, btnMaximizar, btnMinimizar, btnNormal, cmnCronómetro, cmnEspera, cmnMenu, cmnuMateriales, cmnTray, components, dmuEntrega1, dmuEntrega2, dmnuEntrega3, dmuEntrega4, dmuLiberar1, dmuLiberar2, dmnuLiberar3, dmuLiberar4, dmuPréstamo1, dmuPréstamo2, dmnuPréstamo3, dmuPréstamo4, enmSat, imlCronómetro, inlMódulos, intAlto, intAncho, lblBusqueda, lblCategoría, lblDummy, lblEspera, lblEtiquetaBuscar, lblEtiquetaCategorías, lblEtiquetaMóduloActivo, lblEtiquetaSemestre, lblEtiquetaUsuario, lblEtiquetUsuarios, lblMódulos, lblPrestados, lblReloj, lblSemestre, lsblStatus, lblUsuario, lblUsuarios, lblVencimiento, lsvCronómetro, lsvEspera, lsvPanel, lsvPréstamos, mnu15Minutos, mnu30Minutos, mnu45Minutos mnu60Minutos, mnuAcercaDe, mnuActivarCuenta, mnuActualizarDatosUsuario, mnuAdministración, mnuAdministraciónCuentas, mnuAdministrarActualizarInferfase, mnuAdministrarPanel, mnuAgregarUsuario, mnuAjusteSeguridad, mnuAlumnosQueUsaronServicio, mnuApagarMódulos, mnuArchivo, mnuArchivoTransacciones, mnuAyuca, mnuBloquearCuenta, mnuBloquearCuenta, mnuBorrarCuenta, mnuCambairContraseña, mnuConfigurar, mnuConsultsa, mnuCrearCuenta, mnuDatosActualizadosXOperador, mnuEfectos, mnuEspaciosDisponibles, mnuEstadísticas, mnuEstadoMódulos, mnuEstSancionesPorUsuario, mnuEstSancionesXUsuario, mnuExportarInventario, mnuImportarInventario, mnuInventario, mnuLiberarPanel, mnuLíneaTiempoEspera, mnuMaterialesXUsuario, mnuMovsInventario, mnuMovsLíneaEspera, mnuMovsMódulo, mnuMovsUsuarios, mnuMovsXMóduloXSemestre, mnuMovsXMóduloXSemXUsuario, mnuMovsXSemestre, mnuPréstamoMateriales, mnuPréstamoMatXSemestre, mnuPréstamosMatXUsuario, mnuPrincipal, mnuQuitarSanciones, mnuRaya0, mnuRaya1, mnuRaya2, mnuRestauración*Sistema*, mnuSalir, mnuSanciones, mnuSancionesXOperador, mnuSancionesXUsuario, mnuSep0, mnuSep01, mnuSep2, mnuSep3, mnuSep4, mnuSep5, mnuSep6, mnuServicioXUsuario, mnuTrasDesdeCero, mnuTransPosiciónActual, mnuUsoDeEquipos, mnuUsrsCreadosXOperador, mnuUsuarios, mnuUsuariosEnSesión, mnuUsuariosExportar, mnuUsuariosImportar, mnuxAdministrarPanel, mnuxAgregar, mnuxApagarMódulo, mnuxAsignarMódulo, mnuxBuscarEspacios, mnuxEncenderMódulo, mnuxEntregadeMateriales, mnuxEntregar, mnuxLiberarMódulo, mnuxLiberarPanel, mnuxLiberarUsuario, mnuxPréstamodeMateriales, mnuxQuitar, mnuxSep0, mnuxSep1, mnuxSep2, mnuxSep3, mnuxUsuarioNuevo, noi**SAEALABFI**, pgBar, strbStatus, tlbUsuario, tlsBusqueda, tlsEtiqueta, tlsMensajes, tmrCronómetro, tmrEspera, tmrInterfaz, tmrReloj, tmrVencimiento, tmslSair, ToolStripMenuItem1, ToolStripMenuItem16, ToolStripMenuItem2, ToolStripSeparator1, trayAcercaDe, trayActivarCuenta, trayActualizar, trayAdministrar, trayAgregar, trayBloquearCuenta, trayBorrarCuenta, trayConfigurar, trayCrearCuenta, trayEfectos, trayEspaciosDisponibles, trayExportarInventario, trayExportarUsuarios, trayInventario, trayLiberarPanel, trayListaDeUsuarios, trahMódulos, trayMovimientos, trayQuitarSanciones, traySalir, traySanciones, traySep0, traySep1, trayUsuarios, tryArchivo, tryContraseñas, tryTransacciones, tspOcupado, tsSep0, tsSep1, tsSep2, ttpControl y txbBuscar.*

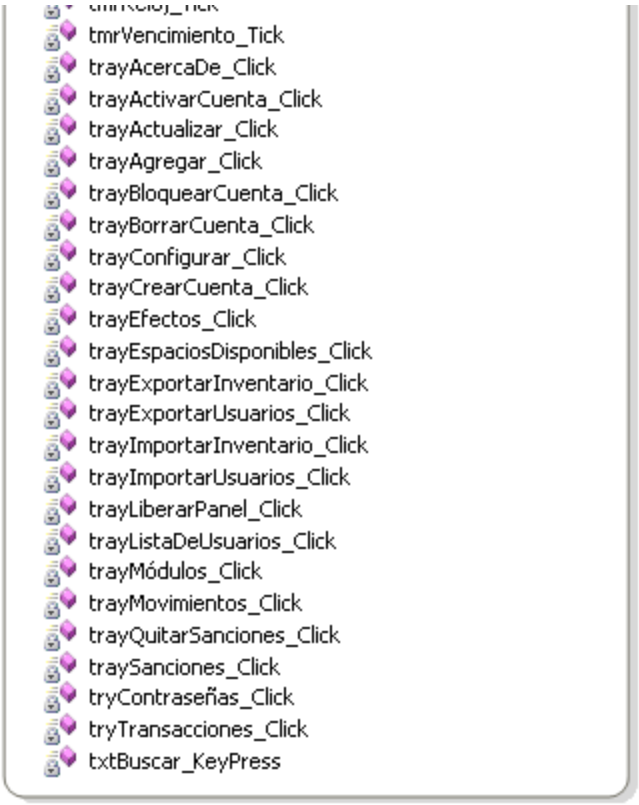
Métodos: *ActivarDialogo, ACtualizarInterfaz, AgregarTiempoAlModulo, Ayuda_Click, btnCerrar_click, tnCerrar_Click, btnMaximizar_Click, btnMinimizar_Click, BuscarESpacios, BuscarEspaciosDisponiblesToolStripMenuItem_Click, cmnMenu_MouseEnter, Dispose, dmuEntrega1_Click, dmuEntrega2_Click, dmuEntrega3_Click, dmuEntrega4_Click, dmuLiberar1_Click, dmuLiberar2_Click, dmuLiberar3_Click, dmuLiberar4_Click, dmuPréstamo1_Click, dmuPréstamo2_Click, dmuPréstamo3_Click, dmuPréstamo4_Click, Finalize, frmPrincipal_FormClosing, frmPrincipal_Load, frmPrincipal_ResizeEnd, InitializeComponent, lsvCronómetro_SelectedIntexChanged, lsvEspera_MouseEnter, lsvPanel_DoubleClick, lsvPanel_SelectedIndexChanged, lsvEspera_MouseEnter, lsvPanel_DoubleClick, lsvPanel_electedIndexChanged, lsvPréstamos_SelectedIndexChanged, mnu115Minutos_Click, mnu30Minutos_Click, mnu45Minutos_Click, mnu60Minutos_Click, mnuAcercaDe_Click,*

frmPrincipal
Class
Form

Campos

Métodos

- ActivarDialogo
- ActualizarInterfase
- AgregarTiempoAlModulo
- Ayuda_Click
- btnCerrar_Click
- btnMaximizar_Click
- btnMinimizar_Click
- btnNormal_Click
- BuscarEspacios
- BuscarEspaciosDisponiblesToolStripMenuItem_Click
- cmnMenu_MouseEnter
- Dispose
- dmuEntrega1_Click
- dmuEntrega2_Click
- dmuEntrega3_Click
- dmuEntrega4_Click
- dmuLiberar1_Click
- dmuLiberar2_Click
- dmuLiberar3_Click
- dmuLiberar4_Click
- dmuPréstamo1_Click
- dmuPréstamo2_Click
- dmuPréstamo3_Click
- dmuPréstamo4_Click
- Finalize
- frmPrincipal_FormClosing
- frmPrincipal_Load
- frmPrincipal_ResizeEnd
- InitializeComponent
- lsvCronómetro_SelectedIndexChanged
- lsvEspera_MouseEnter
- lsvPanel_DoubleClick
- lsvPanel_SelectedIndexChanged
- lsvPréstamos_SelectedIndexChanged
- mnu15minutos_Click
- mnu30minutos_Click
- mnu45minutos_Click
- mnu60minutos_Click
- mnuAcercaDe_Click
- mnuActivarCuenta_Click
- mnuActualizarDatosUsuario_Click
- mnuAdministraciónCuentas_Click
- mnuAdministrarActualizarInterfase_Click
- mnuAdministrarPanel_Click
- mnuAgregarUsuario_Click
- mnuAjusteSeguridad_Click
- mnuAlumnosQueUsaronServicio_Click
- mnuApagarMódulos_Click
- mnuArchivoTransacciones_Click
- mnuBloquearCuenta_Click
- mnuBorrarCuenta_Click
- mnuCambiarContraseña_Click
- mnuConfigurar_Click
- mnuCrearCuenta_Click
- mnuDatosActualizadosXOperador_Click
- mnuEfectos_Click
- mnuEspaciosDisponibles_Click
- mnuEstadoMódulos_Click
- mnuEstSancionesPorUsuario_Click
- mnuExportarInventario_Click
- mnuImportarInventario_Click
- mnuLiberarPanel_Click
- mnuLíneaTiempoEspera_Click
- mnuMaterialesXUsuario_Click
- mnuMovsInventario_Click
- mnuMovsLíneaEspera_Click
- mnuMovsMódulo_Click
- mnuMovsUsuarios_Click
- mnuMovsXMóduloXSemestre_Click
- mnuMovsXMóduloXSemXUsuario_Click
- mnuMovsXSemestre_Click
- mnuPréstamoMateriales_Click
- mnuPréstamoMatXSemestre_Click
- mnuPréstamosMatXUsuario_Click
- mnuQuitarSanciones_Click
- mnuRestauraciónSistema_Click
- mnuSalir_Click
- mnuSanciones_Click
- mnuSancionesXOperador_Click
- mnuSancionesXUsuario_Click
- mnuServicioXUsuario_Click
- mnuTransDesdeCero_Click
- mnuTransPosiciónActual_Click
- mnuTraySalir_Click
- mnuUsoDeEquipos_Click
- mnuUsrsCreadosXOperador_Click
- mnuUsuariosEnSesión_Click
- mnuUsuariosExportar_Click
- mnuUsuariosImportar_Click
- mnuxAdministrarPanel_Click
- mnuxAgregar_Click
- mnuxApagarMódulo_Click
- mnuxAsignarMódulo_Click
- mnuxEncenderMódulo_Click
- mnuxEntregar_Click
- mnuxLiberarMódulo_Click
- mnuxLiberarPanel_Click
- mnuxQuitar_Click
- mnuxUsuarioNuevo_Click
- New
- noiSAEALABFI_MouseDoubleClick
- RefrescarStatus
- Salir
- tmrCronómetro_Tick
- tmrEspera_Tick
- tmrInterfase_Tick
- tmrReloj_Tick



mnuActualizarDatosUsuario_Click, mnuAdministraciónCuentas_Click, mnuAdministrarActualizarInterfaz_Click, mnuAdministrarPanel_Click, mnuAgregarUsuario_Click, mnuAjusteSeguridad_Click, mnuAlumnosQueUsaronServicio_Click, mnuApagarModulos_Click, mnuArchivoTransacciones_Click, mnuBloquearCuenta_Click, mnuBorrarCuenta_Click, mnuCambiarContraseña_Click, mnuConfigurar_Click, mnuCrearCuenta_Click, mnuDatosActualizadosXOperador_Click, mnuEfectos_Click, mnuEspaciosDisponibles_Click, mnuEstadoMódulos_Click, mnuEstSancionesPorUsuario_Click, mnuExportarInventario_Click, mnuImportarInventario_Click, mnuLiberarPanel_Click, mnuLíneaTiempoEspera_Click, mnuMaterialesXUsuario_Click, mnuMovsInventario_Click, mnuMovsLíneaEspera_Click, mnuMovsMódulo_Click, mnuMovsUsuarios_click

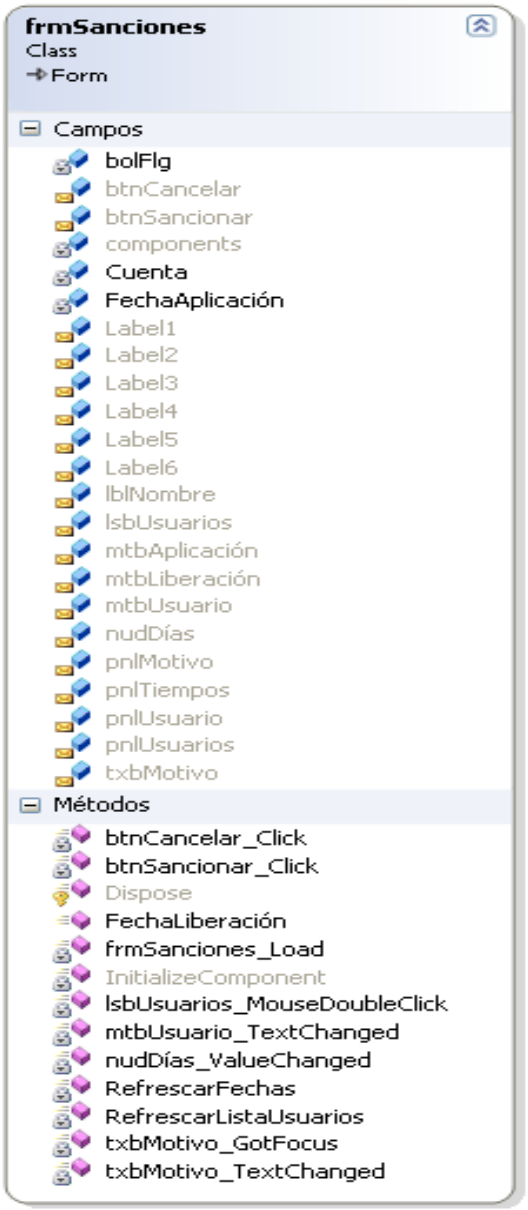
*mnuMovsXMóduloXSemestre_Click, mnuMovsXMóduloXSemXUsuarios_Click, mnuMovsXMóduloXSemestre_Click, mnuMovsXSemXUsuario_Click, mnuMovsXSemestre_Click, mnuPréstamoMateriales_Click mnuPréstamoMatXSemestre_Click, mnuPrésamoMatXUsuario_Click, mnuQuitarSanciones_Click, mnuRestauración*Sistema*_Click, mnuSalir_Click, mnuSanciones_Click, mnuSancionesXOperador_Click, mnuSancionesXUsuario_Click, mnuTransDesdeCero_Click, mnjTransPosiciónActual_Click, mnuTraySalir_Click, mnuUsoDeEquipos_Click, mnuUsrsCreadosXOperador_Click, mnuUsuriosEnSesión_Click, mnuUsuariosExportar_Click, mnuUsuariosImportar_Click, mnuxAdministrarPanel_Click, mnuxAgregar_Click, mnuxApagarMódulo_Click, mnuxAsignarMódulo_Click, mnuxEncenderMódulo_Click, mnuxEntregar_Click, mnuxUsuarioNuevo_Click, New, noi**SAEALABFI**_MouseDoubleClick, RefrescarStatus, Salir, tmrCronómetro_Tick, tmrEspera_Tick, tmrInterfaz_Tick, tmrReloj_Tick, tmrReloj_Tick, tmrVencimiento_Tick, trayAcercaDe_Tick, trayActivarCuenta_Click, trayActualizar_Click, trayAgregar_Click, trayBloquearCuenta_Click, trayBorrarCuenta_Click, trayConfigurar_Click, trayCrearCuenta_Clic, trayEfectos_Click, trayEspaciosDisponibles_Click, trayExportarInventario_Click, trayExportarUsuarios_Click, trayImportarInventario_Click, trayExportarUsuarios_Click, trayLiberarPanel_Clic, trayListaDeUsuarios_Click, trayMódulos_Click, trayMovimientos_Click, trayQuitarSanciones_Click, traySanciones_Click, tryContraseñas_Click, tryTransacciones_Click y txtBuscar_KeyPress.*

Vínculos: *clsGeneral, clsInterfase, clsMódulo, clsParámetros, clsSeguridad, clsTransacción, modFunciones, modInicio, modInventario, modModulos, modPanel, modSanciones, modSeguridad, modUsuarios, frmAcercaDe, frm_Administración_ActivarCuenta, frm_Administración_ArchivoDeTransacciones, frm_Administración_BloquearCuenta, frm_Administración_BorrarCuenta, frm_Administración_CambiarContraseña, frm_Administración_CrearCuentaNueva, frm_Administración_Cuentas, frm_AdministraciónPanel, frmBienvenida, frmConfigurar, frmConsulta_ActualizaciónDatosXOperador, frmConsulta_ListaDeEspera, frmConsulta_MaterialesXUsuario, frmConsulta_MovsXMódulo, frmConsulta_MovsXOperadorModuloYSemestre, frmConsulta_MovsXSemestre,*

*frmConsulta_MovsXUsuario, frmConsulta_MovsXMóduloYSemestre, frmConsulta_PréstamosXSemestre, frmConsultaDeSancionesAplicadasPorOperadorDel*Sistema*, frmConsulta_SancionesxUsuario, frmConsultaDeUsuariosCreadosPorOperadorDel*Sistema*, frmDespedida, frmEspera, frmEstadística_MaterialesPrestadosXUsuario, frmEstadísticas_MaterialesPrestados, frmEstadísticasDeSancionesAplicadasPorUsuario, frmEstadísticas_ServicioPorUsuario, frmEstadísticas_ListaEspera, frmEstadísticas_UsoDeEquiposPorUsuarios, frmEstadísticas_UsoDeEquipos, frmInventario_Módulos, frm_Inventario_ConsultaDeLosMovimientos, frmListadoDeUsuarios, frmSanciones, frmUsuarios_Actualizar, frmUsuarios_Agregar y frm_Usurarios_SelecciónDeUsuario*

IV.3.54 CLASE frmSanciones

Líneas	Archivo
312	frmSanciones.vb



La finalidad de esta clase-formulario es la de permitir a los operadores del Sistema, llevar a cabo las tareas de aplicación de sanciones o de reprogramación de la vigencia de las ya existentes sobre algún usuario en particular, en este formulario también se indicaran los motivos por el cual se ha aplicado la sanción y se deberá seleccionar el tiempo en número de días hábiles de sanción.

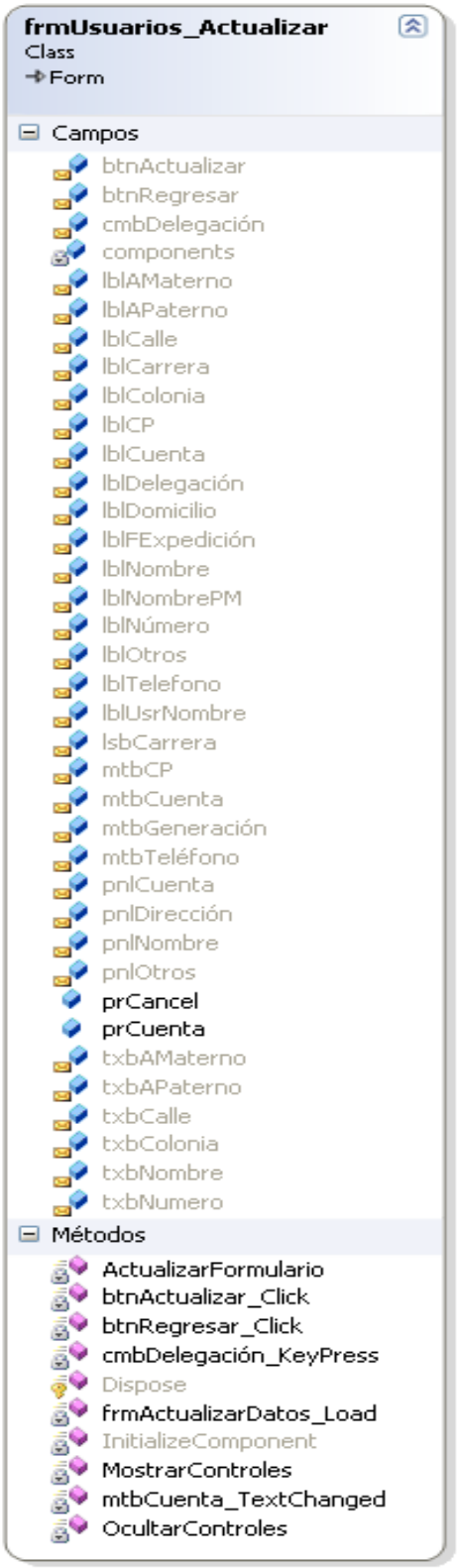
Campos: *bolFlg, btnCancelar, btnSancionar, components, Cuenta, FechaAplicación, Label1, Label2, Label3, Label4, Label5, Label6, lblNombre, lblUsuarios, mtbAplicación, mtbLiberación, mtbUsuario, nudDías, pnlMotivo, pnlTiempos, pnlUsuario, pnlUsuarios y txbMotivo.*

Métodos: *btnCancelar_Click, btnSancionar_Click, Dispose, FechaLiberación, frmSanciones_Load, InitializeComponent, lsbUsuarios_MouseDoubleClick, mtbUsuario_TextChanged, nuDías_ValueChanged, RefrescarFechas, RefrescarListaUsuarios, txbMotivo_GotFocus y txbMotivo_TextChanged.*

Vínculos: *clsGeneral, clsModulo, clsParametros, clsTransacción, modFunciones, modUsuarios y frmPrincipal.*

IV.3.55 CLASE frmUsuarios_Actualizar

Líneas	Archivo
261	frmUsuarios-ActualizarDatos.vb



La finalidad de esta clase-formulario es la de permitir a los operadores del Sistema, actualizar los datos de los usuarios registrados en el Sistema, tomando como dato de referencia al número de cuenta y previa verificación de la existencia de esta en la Base de Datos, entonces se procede a la edición de de los datos asociados al usuario seleccionado.

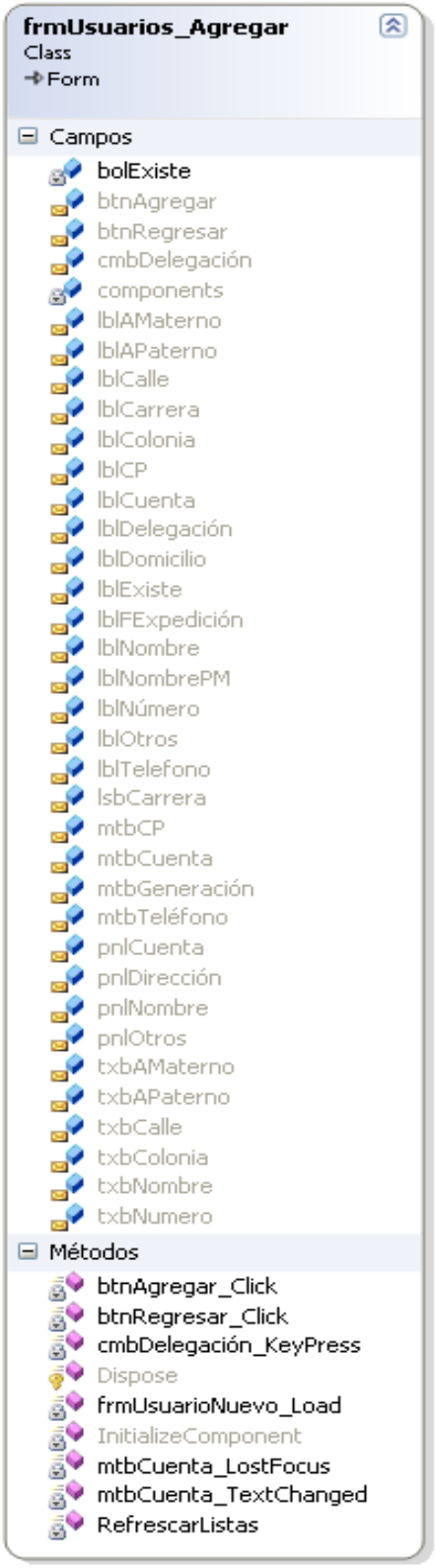
Campos: *btnActualizar, btnRegresar, cmbDelegación, components, lblAMaterno, lblAPaterno, lblCalle, lblCarrera, lblColonia, lblCP, lblCuenta, lblDelegación, lblDomicilio, lblFExpedición, lblNombre, lblNombrePM, lblNúmero, lblOtros, lblTelefono, lblUsrNombre, lsbCarrera, mtbCP, mtbCuenta, mtbGeneración, mtbTeléfono, pnlCuenta, pnlDirección, pnlNombre, pnlOtros, prCancel, prCuenta, txbAMaterno, txbAPaterno, txbCalle, txbColonia, txbNombre y txbNumero.*

Métodos: *ActualizarFormulario, btnActualizar_Click, btnRegresar_Click, cmbDelegación_KeyPress, Dispose, frmActualizarDatos_Load, InitializeComponent, MostrarControles, mtbCuenta_TextChanged y OcultarControles.*

Vínculos: *clsGeneral, clsSeguridad, clsTransacción, modFunciones, modUsuarios, frmPrincipal y frm_Usuarios_SelecciónDeUsuario.*

IV.3.56 CLASE frmUsuarios_Agregar

Líneas	Archivo
239	frmUsuarios-AgregarNuevo.vb



La finalidad de esta clase-formulario es la de proporcionar a los operadores del Sistema el mecanismo mediante el cual se podrán agregar usuarios nuevos a la Base de Datos.

Se verificará primeramente que el usuario no esté registrado, si existe entonces se indicará mediante un mensaje de error sobre la existencia del usuario y si no existe, entonces se procede a desplegar el formulario mediante el cual se podrá anexar un usuario a la Base de Datos.

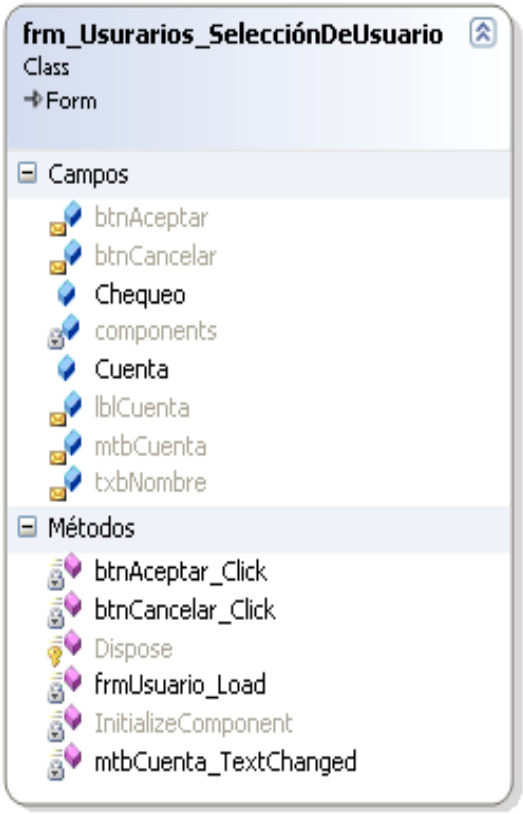
Campos: *bolExiste, btnAgregar, btnRegresar, cmbDelegación, components, lblAMaterno, lblAPaterno, lblCalle, lblCarrera, lblColonia, lblCP, lblCuenta, lblDelegación, lblDomicilio, lblExiste, lblFExpedición, lblNombre, lblNombrePM, lblNúmero, lblOtros, lblTelefono, lsbCarrera, mtbCuenta, mtbGeneración, mtbTeléfono, pnlCuenta, pnlDirección, pnlNombre, pnlOtros, txbAMaterno, txbAPaterno, txbCalle, txbColonia, txbNombre y txbNumero.*

Métodos: *btnAgregar_Click, btnRegresar_Click, cmbDelegación_KeyPress, Dispose, frmUsuarioNuevo_Load, InitializeComponent, mtbCuenta_LostFocus, mtbCuenta_TextChanged y RefrescarListas.*

Vínculos: *clsGeneral, clsParámetros, clsSeguridad, clsTransacción, modFunciones, modUsuarios y frmPrincipal.*

IV.3.57 CLASE frm_Usurarios_SelecciónDeUsuario

Líneas	Archivo
158	frmUsuarios-SelecciónDeUsuario.vb



La finalidad de esta clase-formulario es la de apoyar al Sistema en las tareas comunes en las que se requiere solicitar el número de cuenta de un usuario; se apoya en el uso de una caja de edición de texto configurada para que solo acepte dígitos numéricos y al ser tecleados 9 se hace una búsqueda en la Base de Datos con la finalidad de que se verifique que exista, si existe en la Base de Datos, dentro del mismo dialogo mostrará el nombre asociado al número de cuenta de no ser el caso se indica que no está registrado ese número de cuenta, permitiéndole al operador corregir el número de cuenta para intentar de nueva cuenta checar si existe o no el nuevo número de cuenta.

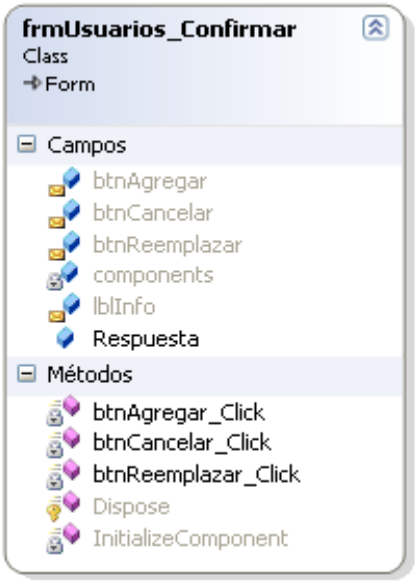
Campos: *btnAceptar, btnCancelar, Chequeo, components, Cuenta, lblCuenta, mtbCuenta y txbNombre.*

Métodos: *btnAceptar_Click, btnCancelar_Click, Dispose, frmUsuario_Load, InitializeComponent y mtbCuenta_TextChanged.*

Vínculos: *clsGeneral, modFunciones, modModulos y modUsuarios.*

IV.3.58 CLASE frmUsuarios_Confirmar

Líneas	Archivo
158	frmUsuarios_Confirmar.vb



La finalidad de esta clase-formulario es la de apoyar al Sistema en el proceso de importación de datos de los usuarios. Se mostrará entonces al *Operador del Sistema* este dialogo en el que se solicita que seleccione entre las opciones: *Agregar, Reemplazar* y *Cancelar proceso.*

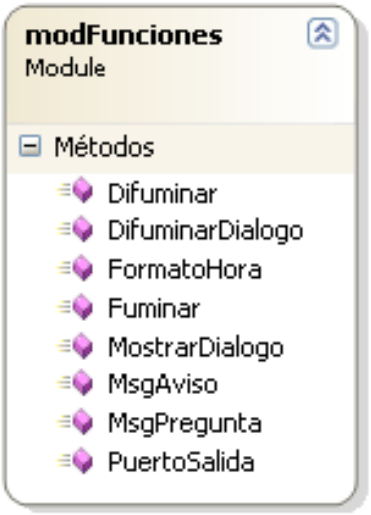
Campos: *btnAgregar, btnCancelar, btnReemplazar, components, lblInfo y Respuesta.*

Métodos: *btnAgregar_Click, btnCancelar_Click btnReemplazar_Click, Dispose e InitializeComponent.*

Vínculos: *modFunciones y modUsuarios.*

IV.3.59 MODULO modFunciones

Líneas	Archivo
214	modFunciones.vb



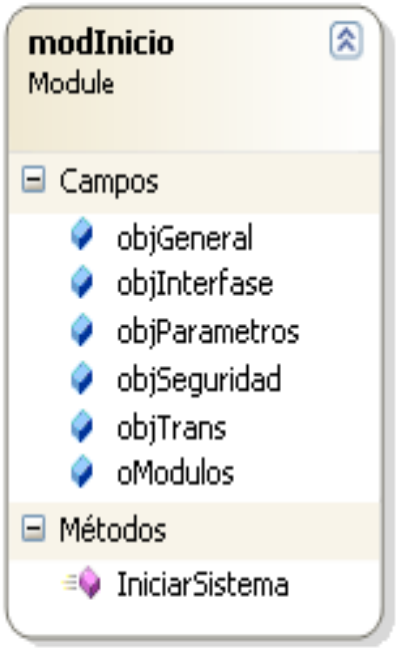
La finalidad de este módulo es contener los métodos de uso común para todo el Sistema en conjunto. Los métodos que contiene los que se encuentran asociados a los efectos de video, el despliegue de diálogos especiales para mostrar de manera uniforme a los mensajes que se generan con la operación del Sistema, el cuadro de dialogo en el que se pide a los operadores confirme alguna advertencia de operación, formatos de hora para despliegue en los tiempos de uso de los módulos y en las operaciones de comunicación con la *interfaz electrónica* mediante el puerto serie como ya ha sido indicado en la justificación de esta decisión.

Métodos: *Difuminar, DifuminarDialogo, FormatoHora, Fuminar, MostrarDialogo, MsgAviso, msgPregunta y PuertoSalida.*

Vínculos: *clsInterfase y clsParametros.*

IV.3.60 MODULO modInicio

Líneas	Archivo
222	modInicio.vb



La finalidad de este modulo es el de contener todos los elementos clave del funcionamiento del Sistema. El Método IniciarSistema se invoca antes de que se creen todos los controles del formulario frmPrincipal y en éste se realizaran todas las funciones básicas de verificación para comenzar a operar el Sistema. Durante estas fases de operación se encuentran las de preparación de los objetos de uso General, de la Interfaz, de los parámetros del Sistema, Seguridad, Transacciones y la matriz contenedora de los módulos instalados en el Laboratorio.

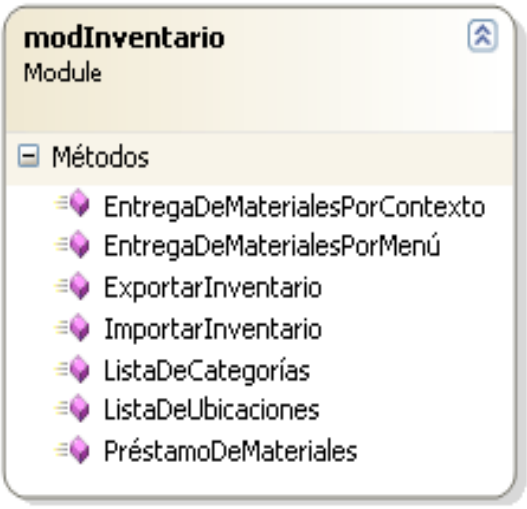
Campos: *objGeneral, objInterfaz, objParametros, objSeguridad, objTrans y oModulos.*

Métodos: *Iniciar*Sistema.

Vínculos: *clsGeneral, clsInterfase, clsModulo, clsParametros, clsSeguridad, clsTransacción, modFunciones, modMódulos y modSeguridad.*

IV.3.61 MODULO modInventario

Líneas	Archivo
652	modInventario.vb



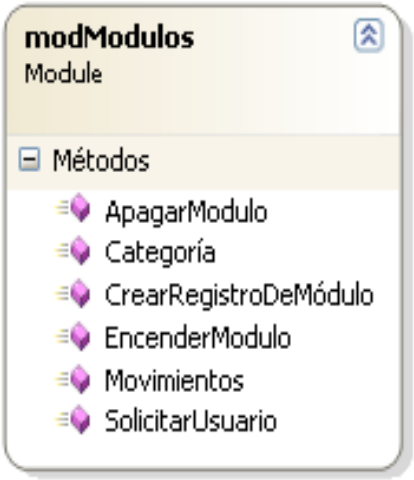
La finalidad de este módulo es el de contener a todos los métodos que se encargan de administrar las tareas relacionadas con el Inventario general del Laboratorio, así como del estado que guardan los equipos de los módulos dentro del Laboratorio. Entre estos métodos se encuentran los de entrega de materiales mediante el acceso del menú contextual de los módulos o mediante el menú contextual en el panel de materiales pendientes por entregar; la exportación e importación de inventario; la lista de categorías de los bienes del Laboratorio, la lista de ubicaciones registradas para los bienes del Laboratorio y el préstamo de materiales.

Métodos: *EntregaDeMaterialesPorContexto, EntregaDeMaterialesPorMenú, ExportarInventario, ImportarInventario, ListaDeCategorías, ListaDeUbicaciones Y PréstamoDeMateriales.*

Vínculos: *clsGeneral, clsInterfase, clsModulo, clsParametros, clsSeguridad, clsTransacción, modFunciones y modInventario.*

IV.3.62 MODULO modModulos

Líneas	Archivo
265	modInventario.vb



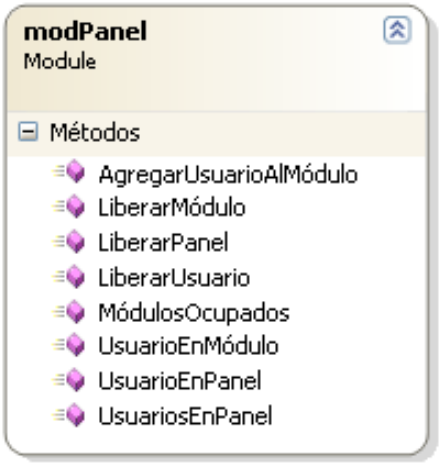
La finalidad de este modulo es el de contener todos los métodos de uso común dentro de la operación del Sistema en su conjunto en relación a la manera en que operan los módulos. Estas funciones consideran las de encender y apagar los módulos mediante la *interfaz electrónica*, las categorías de los operadores, los registros de módulos, el registro de movimientos generales sobre los módulos y la solicitud de un usurario para realizar las tareas de administración del Laboratorio.

Métodos: *ApagarModulo, Categoría, CrearRegistroDeMódulo, EncenderMódulo, Movimientos y SolicitarUsuario.*

Vínculos: *clsGeneral, clsMódulo, clsParametros, clsTransacción y modFunciones.*

IV.3.63 MODULO modPanel

Líneas	Archivo
315	modPanel.vb



La finalidad de este módulo es el de contener a todos los métodos encargados de realizar las tareas comunes en relación al uso del panel principal donde se simula y lleva el control de cada uno de los módulos de manera independiente.

Métodos: *AgregarUsuarioAlMódulo, LiberarMódulo, LiberarPanel, LiberarUsuario, MódulosOcupados, UsuarioEnMódulo, UsuarioEnPanel y UsuariosEnPanel.*

Vínculos: *clsGeneral, clsInterfase, clsMódulo, clsParametros, clsTransacción, modFunciones, modMódulos y modUsuarios.*

IV.3.64 MODULO modSanciones

Líneas	Archivo
51	modSanciones.vb

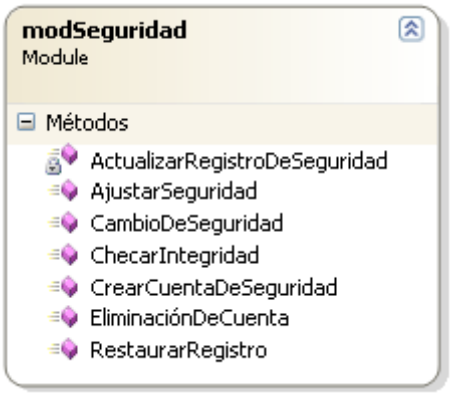
La finalidad de este módulo es el de agrupar a los métodos encargados de realizar un acceso a las operaciones relacionada con las Sanciones de los usuarios desde la operación general del Sistema.

Métodos: *QuitarSanciones.*

Vínculos: *clsGeneral, modFunciones, modModulos y modUsuarios.*

IV.3.65 MODULO modSeguridad

Líneas	Archivo
513	modSeguridad.vb



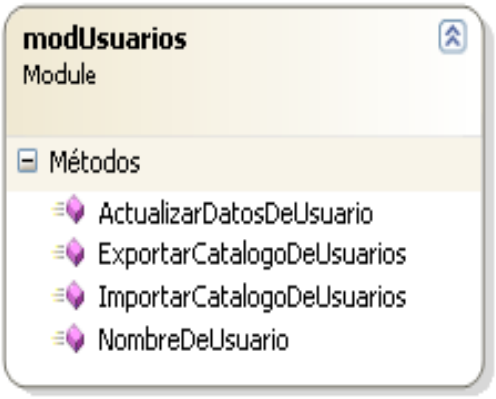
La finalidad de este módulo es el de contener todos los métodos que se emplean para realizar las operaciones generales del Sistema en cuanto a la integridad y a las operaciones que están vinculadas a las tareas de seguridad y de las operaciones de transacciones básicas.

Métodos: *ActualizarRegistroDeSeguridad, AjustarSeguridad, CambioDeSeguridad, ChecarIntegridad, ChecarCuentaDeSeguridad, EliminaciónDeCuenta y RestaurarRegistro.*

Vínculos: clsGeneral, clsSeguridad, clsTransacción y modFunciones.

IV.3.66 MODULO modUsuarios

Líneas	Archivo
421	modUsuarios.vb



La finalidad de este módulo es el de contener todos los métodos asociados a las operaciones relacionadas con los usuarios, como lo son la actualización de datos, creación de usuarios, exportación de catalogo de usuarios, importación de catalogo de usuarios y la obtención de los nombres de los usuarios a partir de su número de cuenta.

Métodos: *ActualizarDatosDeUsuario, ExportarCatalogoDeUsuarios, ImportarCatalogoDeUsuarios y NombreDeUsuario.*

Vínculos: *clsGeneral, clsParametros, clsTransacción y modFunciones.*

CAPÍTULO V: INTERFAZ ELECTRÓNICA

V.1 Elección de componentes

La *interfaz electrónica*, tendrá la finalidad exclusiva de activar o desactivar el suministro del flujo eléctrico a cualquiera de los módulos que se encuentran instalados dentro del Laboratorio. En la actualidad se cuentan con 30 módulos.

Para lograr esto, se requiere la utilización de componentes de tipo digital y de componentes analógicos para poder implementar la finalidad definida para esta parte del proyecto.

El elemento principal para esta *interfaz* estará formado por un componente de memoria independiente para cada módulo en el que se almacene el estado que guarda el módulo que será asociado a dicho componente de memoria y que a su vez servirá para activar un conmutador electrónico de corriente alterna.

Al comenzar a revisar el catalogo de componentes tipo TTL que existen en el mercado, se encontró que existen diversos encapsulados que permiten la implementación de memoria como es el caso de los flip-flops y como ejemplo se tiene el encapsulado SN74F273 que implementa 8 flip-flops tipo D y que utilizan simultáneamente una señal de habilitación, similarmente se encuentra el caso del encapsulado SN74F574. En ambos casos se controlan señales de 3 estados. El encapsulado SN74F74 implementa 2 flip-flops independientes tipo D con señal de habilitación independiente así como una señal denominada preset. Otro encapsulado es el SN74F174A que implementa 8 flip-flops tipo D con una señal de habilitación y de retención conjunta (los 8 flip-flops). De manera similar existen otros encapsulados que implementan a 4 flip-flops (SN74F175), otros que implementan 8 flip-flops (SN74F377A); encapsulados con flip-flop tipo J-K (SN74F109 o el SN74F112) y otros más, como es el caso de los encapsulados SN74LS373, SN74LS374 y SN74LS73A y algunos más.

En cuanto a los registros se encontró con un encapsulado que implementa 8 bits de corrimiento y almacenamiento simultáneo con salidas de tres estados, éste corresponde al encapsulado SN74F299, un encapsulado de registro de corrimiento de 8 salidas en paralelo y una entrada en serie (SN74164 o SN74LS164), un encapsulado de 8 entradas en paralelo y una salida en serie (SN74165A, SN74LS164A, SN74166A O SN74LS166A), un encapsulado de 4 registros de cambio bidireccional (SN74194, SN74LS194A o SN74S194A), un encapsulado de registros de corrimiento de 4 bits de acceso paralelo (SN74195, SN74LS195A o SN74S195A) y la lista sigue con registros de corrimiento asociados a cambios serial-parelelo, paralelo-paralelo y combinaciones de ambos.

En cuanto a los latches se encontró al encapsulado SN74F373 que corresponde a un latch octal tipo D con salidas de 3 estados y similarmente al encapsulado SN74F573; un encapsulado de 4 bits de registro tipo D con salidas de 3 estados (SN74LS173A), un encapsulado con latches transparentes de tipo D y flip-flops disparados por rizo (correspondiente a las familias SN74LS373, SN74LS374, SN74S373 y SN74S374). Existe un encapsulado de archivo de registros de 4x4 con salidas de 3 estados (SN74LS670), un encapsulado de un latch biestable de 4 bits (SN7475 y SN74LS75), existe un encapsulado de latches de 8 bits direccionable (SN74259 o SN74LS259B), un encapsulado de 4 latches S-R (SN74279 y SN74LS279A) y un encapsulado de 4 latches biestabes (SN74LS375) entre otros.

Como es de notarse; se requieren al menos de 30 dispositivos de retención y los encapsulados en el mejor de los casos nos ofrece 8 de estos componentes, pero para poder direccionar y

almacenar el estado deseado en dicho componente se requeriría colocar la palabra de 8 bits (4 bits ó 2 bits según sea el caso del encapsulado) para que con la señal de habilitación sean retenidos simultáneamente todos los bits que requiere el componente. Esta situación podría traer el problema de que al hacer el cambio, pudiese haber un pequeño lapso de transición en el que hubiese un flasheo en las salidas y por consiguiente, se vería reflejado en las líneas de corriente.

Analizando por separado cada uno de los dispositivos previamente indicados, se llegó a la conclusión de que el dispositivo ideal para este proyecto es el **SN74LS259B** que ofrece la independencia de 8 latches que se pueden direccionar mediante 3 bits de entrada que sirven como dirección a cualquiera de los 8 latches, una línea que sirve como dato a escribir en la localidad direccionada y finalmente una línea de habilitación, una de reset y por supuesto, las 8 líneas de salida.

Se hubiera podido escoger encapsulados de 2 ó 4 flip-flops, pero para el primer caso se requerirían 4 veces la cantidad de dispositivos que el elegido y en el segundo de los casos el doble y con el problema de que la señal de reloj es común a los flip-flops por lo que no existe independencia física y se estaría en el caso de tener que colocar los datos independientes de manera simultánea previos a la señal de habilitación.

Con el dispositivo elegido, se requerían en total 4 encapsulados para poder direccionar 32 módulos y como ya se ha mencionado, se encuentran instalados 30 módulos con lo que se cubrirá de manera total a cada módulo.

La señal de habilitación para este dispositivo es mediante una señal de transición alta-baja-alta y, para que esto se pueda realizar, se empleará un decodificador que permitirá mediante la señal de habilitación que en éste último se generará, la selección de alguno de los 4 bloques.

Revisando el catálogo, existen varios dispositivos enfocados a la decodificación de señales, entre ellos se encuentran los enfocados a conversión BCD-Decimal, Control de displays de 7 segmentos, un decodificador/demultiplexor de 3 lineas a 8 salidas con latch direccionable (SN74LS137), un encapsulado de decodificador/demultiplexor de 3 líneas a 8 líneas para alta velocidad en memorias y transmisión de datos (SN74LS138 y SN74LS138A), decodificadores duales de dos líneas a 4 (SN74LS139 o SN74LS139A), seleccionadores de datos/multiplexores (SN74150, SN74151A, SN74LS151 o SN74S151), selectores/multiplexores duales de 4 líneas a una línea (SN74153, SN74LS153 o SN74S153), decodificadores/demultiplexadores duales de 2 lineas a 4 lineas (SN74155, SN74156, SN74LS155A o SN74LS156), selectores/multiplexores cuádruples de dos líneas a una (SN74157, SN74LS157, SN74LS158, SN74S157 o SN74S158), multiplexores/seleccionadores de datos con salidas de 3 estados (SN74251, SN74LS251 o SN74S251), selectores/multiplexores de datos duales de 4 líneas a una línea con salidas de 3 estados (SN74LS253 o SN74S253), selectores /multiplexores cuádruple de 2 líneas a una línea (SN74LS257B, SN74LS258B, SN74S257 o SN74S258) y multiplexores cuádruple de 2 entradas con almacenamiento (SN74298, SN74LS298 o SN74LS399).

De los componentes indicados en el párrafo anterior, se ha decidido utilizar al encapsulado **SN74LS155N** por ser un dispositivo cuyas salidas son en señal baja; tiene la peculiaridad de mantener de manera independiente dos decodificadores con dos líneas de entrada y 4 de salida. Para el caso del proyecto se necesitaría uno de los dos decodificadores para tener la posibilidad de activar alguno de los 4 bloques, pero otra característica que ofrece el encapsulado es que en ambos decodificadores se cuenta con una línea más de direccionamiento y si esta se puentea entre los dos decodificadores, se comporta como un decodificador de 3

entradas a 8 salidas. Aprovechando esta característica, se ha decidido implementarlo de esta manera para que el Sistema en conjunto permita habilitar hasta 8 tarjetas de bloque y si cada bloque soporta hasta 8 módulos, entonces se estará en posibilidad de controlar hasta 64 módulos.

Los bits de datos que se requieren para direccionar de manera individual a cada uno de los módulos se describen más adelante. Por ahora sólo se justifica el uso de los componentes de la *interfaz electrónica*.

Para la comunicación entre el equipo de cómputo y la interfaz electrónica se ha considerado utilizar un puerto serie como ya se ha mencionado; por las características de este puerto en cuanto a los niveles de voltaje de las señales que controla, primeramente será necesario convertirlas a un rango de señales de tipo TTL (0 a 5 volts). Esta tarea se puede realizar mediante comparadores de voltaje, arreglo de transistores y otras técnicas más que, en relación a costos y espacio resultarían más complejos. Sin embargo, existe en el mercado distintos dispositivos que permiten realizar el intercambio bidireccional de los niveles de las señales entre el protocolo NRZ del puerto serial y los niveles TTL de los dispositivos que conforman a la interfaz en conjunto.

Revisando las hojas de especificaciones de estos productos, se ha llegado a la conclusión que el que más se ajusta al proyecto es el **MAX232** que implementa dos pares de conversión de señales de transmisión y de recepción. Para garantizar que los niveles enviados y recibidos al puerto serial de la computadora, la configuración del dispositivo se apoya en el uso de 5 capacitores electrolíticos de 1 µF como se puede consultar en el Capítulo III.

En cuanto al control de la *interfaz*, se podría realizar mediante la conversión de señales serie a paralelo y posteriormente la generación de las señales de control hacia la etapa siguiente de la *interfaz* haciendo uso de registros de corrimiento de conversión serie-paralelo y otros dispositivos auxiliados en temporizadores para lograr este fin. Considerando que en la actualidad existen dispositivos micro-controladores que pueden ser programados y que además entre sus características pueden realizar todas las tareas necesarias para implementar un mecanismo de control. Revisando los dispositivos que en este ramo existen en el mercado, se ha encontrado a la familia de micro-controladores de la empresa **Microchip** que ofrece productos de diversa índole y capacidades en cuanto a niveles de escala e implementación de tareas, de entre estos se ha seleccionado al **PIC16F84A** por ser un dispositivo de escala media y contar con dos puertos bidireccionales nombrados puerto A y puerto B. en este proyecto, se utilizará el puerto A para la comunicación bidireccional con el equipo de cómputo apoyándose en la conversión de niveles proporcionada por el dispositivo **MAX232** y para la generación de las señales de habilitación del decodificador que tendrá como entrada los 3 bits de direccionamiento de selección de bloque que saldrán del puerto B y en conjunto con otros 3 bits que direccionaran al módulo dentro del bloque seleccionado en el decodificador, generará la señal de habilitación del bloque para actualizar de manera específica el estatus del módulo en particular, este mecanismo se detallan más adelante en este Capítulo. Para este fin, el programa del micro-controlador hará la tarea de conversión serie-paralelo y viceversa así como el establecimiento del mecanismo de reconocimiento entre el equipo de cómputo y la *interfaz electrónica*.

Al final de la etapa de memoria por cada módulo, el valor que se encuentre en cada dirección será conectado a un dispositivo opto-acoplador (cuyo principio se ha explicado en el Capítulo III) que será el intermediario entre la etapa digital y analógica, misma que estará conformada por un TRIAC. El opto-acoplador que se utilizará es el **MOC3011** por tratarse de un dispositivo diseñado para ser utilizado en conjunto con TRIACs. La tolerancia de corriente

para los TRIACs se desprende de los valores de consumo de corriente de los módulos según se muestra en la siguiente tabla.

Módulo	Componente	Consumo máximo
0	Equipo de cómputo	1.50 A
	Monitor	2.00 A
	Taladro	3.00 A
	Lámpara UV	0.25 A
	Grabador de memorias	0.25 A
TOTAL		7.0 A
1-29	Osciloscopio	2.0 A
	Fuente dual	2.5 A
	Generador de funciones	0.5 A
	Multímetro digital	0.5 A
TOTAL		5.5 A

Tabla V-1. Consumo máximo de corriente por módulo

Como se puede apreciar, el módulo 0 en condiciones máximas de consumo de corriente, se encuentra en el orden de los 7 Amperes, mientras que los demás módulos en un rango máximo de 5.5 Amperes. Se ha considerado TRIACs de 8 Amperes primero para que se garantice que al menos el nivel de consumo máximo por cada módulo sea satisfecho y, para tener un margen en cuanto al consumo que los usuarios realicen de manera extra ya que en las líneas de alimentación de los módulos también conectan cautines, equipos de cómputo personal y otros más. Para proteger a la *interfaz electrónica*, se está considerando un fusible que se encuentre por lo menos medio ampere del abajo máximo tolerado por el TRIAC.

Es importante considerar que en ocasiones acuden usuarios a realizar proyectos o tareas de materias relacionadas con potencia y en estos casos es incuantificable la cantidad de energía demandada pues se ha ocurrido que en este tipo de proyectos se ha provocado cortos, se botan las pastillas y en alguna ocasión el switch general del edificio.

Para prever esta situación, se sugiere utilizar la tarjeta de bloque 3 que estará destinada para alimentar a los módulos 23, 24, 25, 26, 27, 28 y 29 para que utilice un TRIAC que iguale a la corriente soportada por las pastillas del conmutador (15 Amperes) y que de igual manera se utilice un fusible de medio ampere por debajo de este nivel de corriente o la otra alternativa, es utilizar relevadores en cuyo caso, tendría que hacerse el ajuste en el diseño del PCB para esta tarjeta de bloque.

V.2 Antecedentes

En este Capítulo se muestran los detalles de la *interfaz electrónica*. Una vez que se han definido los conceptos previos al diseño e implementación de la misma, se ha considerado retomar los conceptos clave del puerto Serie ya que la manera en que opera la *interfaz* se encuentra ligada estrechamente a la forma en que se han implementado las normas RS232.

La norma RS232 establece que la comunicación serie se basa en un mecanismo de comunicación

NRZ con niveles de voltaje de -12 a +12 volts; teniendo que para el **1** lógico, el rango de voltaje es de -3 a -12 volts y para el **0** lógico, el rango es de +3 a +12 volts.

En este estándar se establece que la comunicación se realiza mediante una línea denominada TxD y recibe por otra línea denominada RxD. La comunicación se realiza mediante un proceso asíncrono; es decir, no hay un mecanismo implícito mediante el cual se deban de sincronizar el equipo transmisor (en este caso la computadora) y el equipo receptor (en nuestro caso la *interfaz electrónica*) y por este motivo, la norma indica que la línea habilitada para la transmisión se mantendrá en un **1** lógico en espera de que se tenga algún dato en el búfer para ser transmitido, cuando llega este momento se coloca un bit de arranque que es un **0** lógico y en seguida se comienzan a enviar los bits del byte a transmitir, al fin de la transmisión del byte se envían los bits de parada y después de nueva cuenta se coloca la salida de esta línea en un **1** lógico en espera de que se encuentre disponible otro byte para ser transmitido.

El proceso contrario; el de recepción, realiza la interpretación de este mecanismo para que se pueda saber en qué momento comienza a recibir bytes de parte de la *interfaz*. Para que ambos equipos se puedan entender, se debe configurar los parámetros fundamentales que norman este protocolo, es decir: se requiere configurar la velocidad en baudios en que será transmitida o recibida la información en el momento en que se reciba o envíe el bit de arranque, la cantidad de bits que componen al byte dé datos, la cantidad de bits de parada que se requieren para terminar la transmisión o recepción del byte en turno y, un parámetro adicional que corresponde al bit de paridad; éste último es utilizado para determinar si el byte recibido ha llegado de manera correcta, si no fuese el caso, se debería de enviar un mensaje al transmisor para indicarle que no ha llegado de manera correcta y por consiguiente, se le está solicitando que vuelva a transmitir ese byte.

Para el caso de esta aplicación, los valores que se utilizarán para estos parámetros es el que se muestra en la siguiente tabla:

Parámetro	Valor
Puerto	**COM1**
Velocidad	**9600 Baudios**
Bits de datos	**8**
Bits de parada	**2**
Bits de paridad	**0**

Tabla V-2. Parámetros de comunicación entre la aplicación y la interfaz electrónica

Durante la elaboración del modelo de la *interfaz electrónica* ocurrieron distintas circunstancias que condujeron a una situación relacionada la elección del micro controlador que se utiliza en la *interfaz*, originado principalmente por que en el Laboratorio Abierto y en otros que se encuentran dentro de las mismas instalaciones del Edificio y que se utiliza para el fin de grabar los dispositivos electrónicos programables como es el caso de los micro controladores y dispositivos GAL entre otros. Por alguna razón, parece ser que dicho grabador no es compatible con los micro controladores de la familia **PIC**, pues tras la programación y aparente grabación de la aplicación en el micro controlador no hubo problemas y se grabó de forma adecuada, pero en el momento de tratar de echar a andar el prototipo, simplemente no funcionaba; se intentó con otros métodos de programación del mismo con el uso del microBasic y microAssambler y los resultados no eran ni siquiera los más mínimos, se pensó que podría ser el oscilador y como consecuencia, fue adquirido otro nuevo y seguía sin funcionar; se volvió a checar el prototipo y los componentes fueron re ensamblados y el problema persistió.

Tras de realizar todas estas pruebas, conjuntamente con el tutor de tesis, se tomó la decisión de cambiar de micro controlador distinto de escala media y que incluye en su estructura de hardware a una USART (el **PIC16C73B-04/SP**), se preparó una aplicación elemental en la que mediante el uso de interruptores push-button en un puerto de entrada, se reflejara en puerto de salida mediante el uso de leds los valores representados por los push-buttons y también se realizó un pequeño programa para hacer prueba con la USART.

Tras esta preparación, una vez más se acudió a grabar el dispositivo en los equipos indicados y al igual que en el caso del micro controlador anterior, no hubo los resultados esperados. Se recurrió a la asesoría de un profesor de la Facultad que cuenta con experiencia con estos dispositivos y lenguajes para su uso y tras de sí, no hubo manera de poder resolver la situación presentada.

Finalmente, el tutor acudió a uno de sus Jefes laborales quien accedió a prestar un programador original de MicroChip (**PICSTART PLUS**), se hizo la prueba con el micro controlador que inicialmente fue considerado (**PIC16F84A**) y con el último (**PIC16C73B-04/SP**), con el primero no hubo mayor contratiempo, mientras que con el segundo hubo errores registrados que condujo a determinar que el dispositivo se había dañado.

Al grabar el **PIC16F84A**, se procedió a incluirlo en el prototipo ya montado en protoboards y "sorpresa, si funcionó", a partir de este momento, la tarea se enfocó a terminar de implementar la comunicación entre la *interfaz electrónica* y la aplicación. En los siguientes apartados el esfuerzo está enfocado a la descripción de la estructura, su funcionamiento y la prueba con dispositivos que requieren de alimentación eléctrica de tipo alterno.

V.3 Protocolo de comunicación con la *interfaz electrónica*

Ya se ha justificado el medio de comunicación que se utilizará para la conexión entre el Software de aplicación y la *interfaz electrónica*, de esto se ha de aclarar que la *interfaz electrónica* será capaz de administrar el suministro de energía eléctrica para los 30 módulos instalados en el Laboratorio Abierto de Electrónica de la Facultad de Ingeniería.

Debido a la naturaleza de un puerto serial en cuanto a los niveles eléctricos que se utiliza en ellos, no representa mayor problema la distancia que existe entre el equipo de cómputo para la aplicación y la *interfaz electrónica* que es de aproximadamente 7.5 metros de longitud.

En el extremo correspondiente a la *interfaz electrónica*, está tendrá primeramente como tarea, la conversión de los niveles lógicos RS-232 a TTL y en seguida, su conversión a formato paralelo. Esto se explica con más detalle en el siguiente Capítulo.

Por ahora hay que considerar que la forma en que se entenderá el software de aplicación y la *interfaz electrónica* que estará gobernada por un micro controlador **PIC16F84A.**

El protocolo es simple; se basa en el envío de información codificada en un byte y a cambio recibe un byte de respuesta de aceptación o rechazo en un tiempo límite de 500 µSeg y si la respuesta no se recibe en este lapso, entonces se activará un proceso en el que se checa la respuesta de la *interfaz electrónica* en la que se le dará al operador la oportunidad de decidir entre continuar el intento de identificación y comunicación con la *interfaz* o de cancelar todas las operaciones pendientes, pues se está en el caso de que se ha dejado de registrar comunicación por cualquier razón ajena a la operación del Sistema en conjunto (vg. Interrupción de energía eléctrica en la *interfaz electrónica*).

El byte que la computadora enviará a la *interfaz electrónica* mediante el puerto serial tiene el formato que se muestra en la siguiente ilustración.

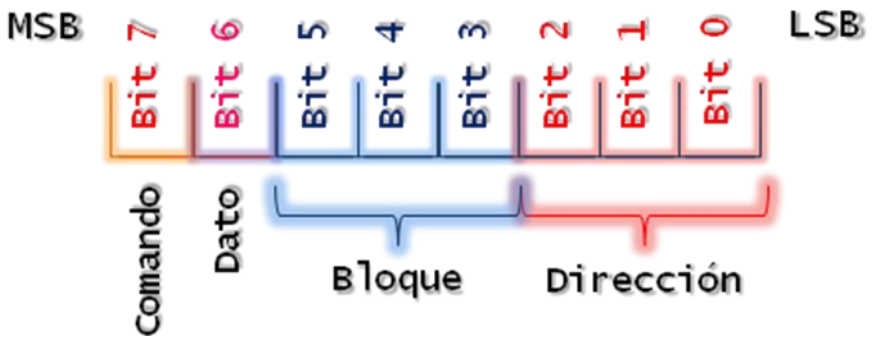


Ilustración V-1. Formato de los bytes de comunicación con la interfaz electrónica

Como se puede observar, el byte formado por 8 bits se agrupa en 4 secciones que en seguida se comentan:

- **Dirección**. Estos tres bits (0 a 2) en conjunto forman 8 combinaciones posibles (0 al 7) que corresponderá al conmutador electrónico dentro de la tarjeta de bloque para que se pueda acceder de manera individual a cada uno de los módulos instalados en el Laboratorio Abierto.

- **Bloque.** Este es un conjunto de bits corresponde al selector de bloque (bits 3 al 5) que conjuntamente tienen la posibilidad de realizar la habilitación de hasta 8 tarjetas de bloque (por las combinaciones posibles con 3 bits). Para el caso de este trabajo de tesis basta con tan solo 2 bits pues con éstos, podemos seleccionar hasta 4 bloques de 8 conmutadores arrojando un total de 32 y si consideramos que en el Laboratorio se encuentran instalados 30 módulos queda entonces cubierta la capacidad con los bits 3 y 4, pero para efectos de la posibilidad de expansión, el Sistema se ha ajustado para que pueda soportar hasta 64 bloques al utilizar todos los bits de dirección y también los bits de bloque.

- **Dato**. Este corresponde al bit 6 y en este bit se colocará el estado que reflejará el módulo respecto de la alimentación eléctrica. En otras palabras: si el valor es **0** lógico entonces se apagará el módulo seleccionado mediante los grupos de bloque y dirección; en caso contrario, si el bit tiene un valor de 1 lógico, se activará la alimentación del módulo seleccionado mediante la combinación de los grupos de bloque y dirección.

- **Comando.** Este bit se interpretará como un comando si se encuentra establecido a un valor de **1** lógico (más adelante se describen los comandos programados) y un valor de **0** lógico conducirá a que el byte se interprete como un encendido o apagado del módulo que se ha seleccionado mediante los bits 0-6.

Como consecuencia de esta agrupación de bits del byte que envía el Sistema, la *interfaz electrónica* realizará los ajustes pertinentes en los componentes electrónicos con la finalidad de que se vean reflejados los cambios solicitados y esperará un 500 µSeg para que el Sistema se estabilice y esté en espera de recibir un nuevo valor para que sea interpretado de acuerdo a lo que se ha descrito en los párrafos anteriores.

Por ejemplo, supóngase que se quiere encender el módulo 4 del bloque 2; el Sistema enviará a

la *interfaz electrónica* un valor con el formato: 01010100 que de manera física corresponderá al módulo 20.

En la siguiente tabla se muestran los valores que corresponderán a cada módulo físico con su combinación de sección de bloque y dirección.

Módulo físico	*Sección de bloque*	*Sección de dirección*
0	000	000
1	000	001
2	000	010
3	000	011
4	000	100
5	000	101
6	000	110
7	000	111
8	001	000
9	001	001
10	001	010
11	001	011
12	001	100
13	001	101
14	001	110
15	001	111
16	010	000
17	010	001
18	010	010
19	010	011
20	010	100
21	010	101
22	010	110
23	010	111
24	011	000
25	011	001
26	011	010
27	011	011
28	011	100
29	011	101
30	011	110
31	011	111
32	100	000
33	100	001
34	100	010
35	100	011
36	100	100
37	100	101
38	100	110
39	100	111
40	101	000
41	101	001
42	101	010
43	101	011
44	101	100

Módulo físico	Sección de bloque	Sección de dirección
45	101	101
46	101	110
47	101	111
48	110	000
49	110	001
50	110	010
51	110	011
52	110	100
53	110	101
54	110	110
55	110	111
56	111	000
57	111	001
58	111	010
59	111	011
60	111	100
61	111	101
62	111	110
63	111	111

V.3.1 Comandos implementados en la *interfaz electrónica*

Al finalizar este proyecto sólo se han implementado dos comandos con los cuales se pueden realizar dos tareas de vital importancia para el Sistema en conjunto: *La identificación de la interfaz electrónica* y *el reinicio de la interfaz*.

El primero de los comandos, es la identificación de la *interfaz electrónica* y se utiliza para determinar si la *interfaz* está operando sin problema alguno. El comando como ya se mostró en la Ilustración IV-16, comienza con un bit "1" y en seguida los restantes 7 bits que identifican la tarea que se desea que se ejecute en la *interfaz electrónica* que para el caso de la identificación de ésta última tiene un valor **1111 0000 (0xF0)**.

El segundo comando que se ha incluido en este formato para propósito de la realización de tareas por parte de la *interfaz*; es el comando de reinicio o "reset" de toda la etapa electrónica y se ha identificado mediante el valor **1111 0001 (0xF1)**.

Estos son los únicos comandos incluidos, si se enviase un valor de comando (que empiece con bit 1 en el MSB) se interpretarán como comando no conocido y responderá con un valor de respuesta alusivo al error de reconocimiento de comando

V.3.2 Respuesta de la *interfaz electrónica*

Una vez establecidas las bases de la comunicación entre la aplicación y la *interfaz electrónica*, toca el turno a las respuestas que ésta última retorna a la aplicación con la finalidad de indicarle a la aplicación que el byte con la configuración del módulo seleccionado se ha realizada de manera correcta.

Si se ha recibido y ejecutado de manera correcta la opción que le envió el programa de

aplicación, la *Interfaz electrónica* responderá con un valor **1111 1111 (0XFF)**; si por alguna razón por causas imprevistas no se pudo identificar y ejecutar la acción deseada de manera correcta, la *interfaz electrónica* responderá con un valor **1111 000 (0xF0)**. Cuando se trata del envío de un comando, tras la evaluación de este, la *interfaz electrónica* responderá con un valor **1111 0001 (0xF1)** si el comando recibido no se encuentra implementado y si se pudo ejecutar de manera correcta, entonces responderá con un valor de aceptación (**1111 1111 ó 0xFF**).

V.4 Modelo conceptual de la *interfaz electrónica*

Una vez que he establecido las bases de operación del Sistema y la *interfaz electrónica* en conjunto, así como el protocolo en que se entenderán para que las tareas propias del **SAEALABFI**, es necesario decir que luego de una búsqueda de los distintos dispositivos que existen en el mercado así como sus características, su uso, simulación y, finalmente la construcción de un prototipo, el modelo que aquí se muestra, funciona al 100%.

En la Ilustración V-1, se presenta en forma esquemática el modelo conceptual de la *interfaz electrónica* de donde se puede apreciar que se está considerando un total de 64 módulos y para el caso del proyecto sólo necesitan 30 y por las características binarias de los dispositivos lógicos se hace un ajuste a un total de 32, aunque por las características de expansión podrá soportar hasta 64. Los bloques del 4 al 7 se encuentran opacados para indicar esta situación.

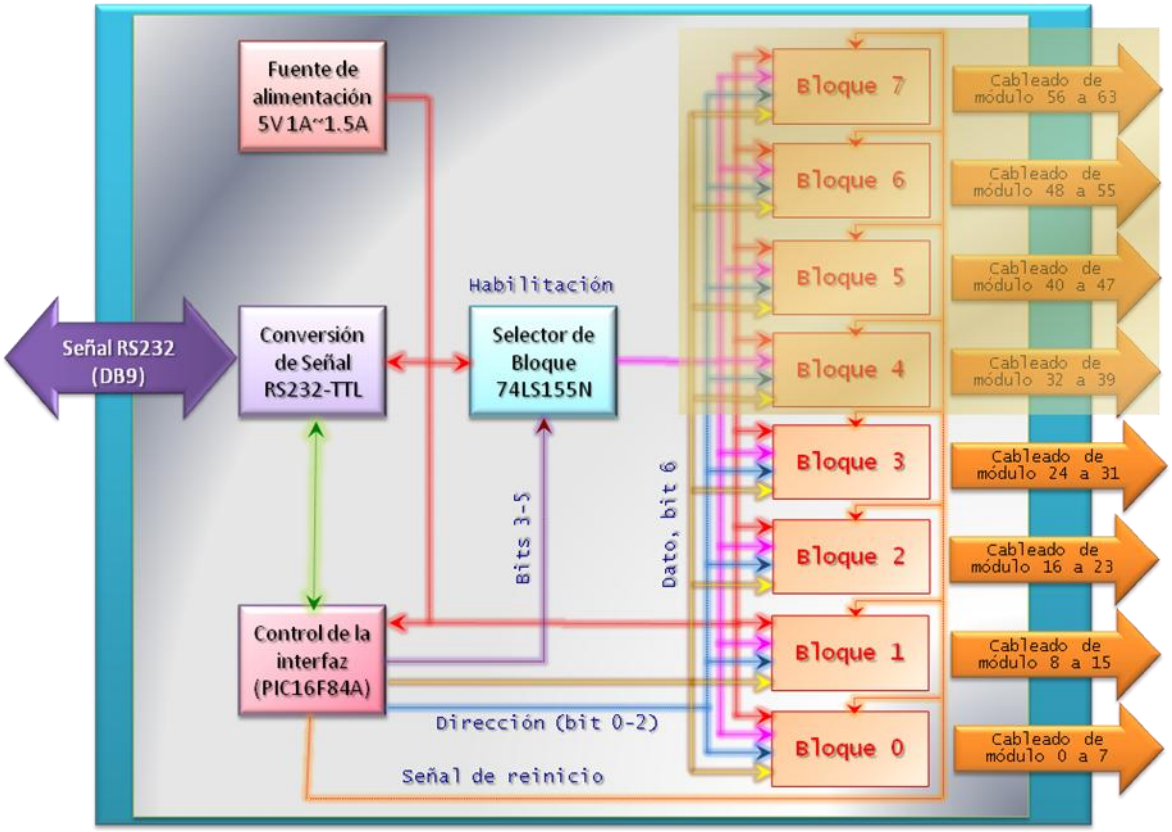


Ilustración V-2. Modelo conceptual de la interfaz electrónica

En el esquema de la ilustración V-3, se observan las distintas etapas de la *interfaz electrónica*, teniendo como entrada a la *interfaz* un suministro de datos digitales en tamaño de un byte cuya estructura ya ha sido indicada en el Capítulo anterior al definir el

protocolo de comunicación.

La comunicación es por consiguiente bidireccional y las fases que conducen al procesamiento e interpretación de los bits recibidos se realizan en las siguientes etapas:

- **Conversión de señal RS232-TTL.** Este es un dispositivo nombrado MAX232 de Maxim (propiedad de Dallas Semiconductor). Este dispositivo ha sido desarrollado para el fin específico de convertir señales RS232 (ya mencionado) a niveles TTL y viceversa; dentro del encapsulado se encuentran dos pares de conversores de este tipo. El funcionamiento de este dispositivo se basa en el apoyo de 5 capacitores electróliticos que garantizan que los niveles RS232 se encuentren en el orden de los -10 volts y +10 volts (los niveles RS232 están dentro del rango de -12 volts a +12 volts). Los datos recibidos y convertidos a los niveles TTL se alimentan a la siguiente fase, desde la cual también recibirán datos que serán sometidos al proceso inverso de conversión, es decir, de niveles TTL a RS232 y serán enviados al equipo de cómputo mediante el puerto Serial.

- **Control de la *interfaz*.** Esta etapa está conformada por un micro controlador **PIC16F84A** que se encarga de llevar el control de la *interfaz electrónica*. Su propósito es que mediante un programa desarrollado en su propio "dialecto", se encargue de recabar todos los bits que le son suministrados por la etapa anterior y realizar las acciones pertinentes para que se cumpla con el objetivo de la *interfaz* en conjunto. Esta etapa realizará un reinicio de toda la interfaz al momento de que se ha alimentado eléctricamente debido a que con los dispositivos utilizados para el prototipo, no operan de manera igual en cuanto al estado que guardan los componentes de los bloques. En este caso, se utilizó 3 de un fabricante y uno de otro. Al ser alimentados, los tres mantienen sus salidas en un valor de **1**, mientras que el otro que no pertenece al mismo fabricante, los mantiene en un valor de **0** lógico. Cuando se determina que el valor recibido corresponde a un proceso de conmutación de energía eléctrica, el micro controlador colocará en las salidas del puerto utilizado para este fin a los bits que corresponden a la dirección del puerto dentro del bloque y, los otros bits que corresponden al selector del bloque en la siguiente fase para que una vez que se ha preparado a los componentes, se genere una señal de habilitación (en flanco de transición de alto a bajo) para el decodificador (3 a 8) que finalmente activará una de las 8 salidas que corresponderá a una de los 8 bloques mostrados en el diagrama conceptual y con esto, el efecto será la actividad concreta que se ha seleccionado por medio del byte recibido por el puerto Serial. Cuando se trata de la interpretación de un comando de identificación, el programa de este dispositivo simplemente regresará el valor correspondiente a la respuesta de identificación y si todo marcha bien, lo hará en un lapso no mayor a 500 µSeg. En caso que el comando recibido sea el de reinicio de la Interfaz, generará una señal de habilitación (flanco de transición de alto a bajo) de la señal de reinicio de la *interfaz* como se puede apreciar en el modelo mostrado en la Ilustración V-1.

- **Selector de bloque.** Este en realidad es un decodificador de tres entradas y como consecuencia 8 salidas; se habilita en un flanco de transición de alto-bajo y que las salidas de este a su vez sirven como señales de habilitación de los bloques ubicados en la última etapa del modelo conceptual. De esta manera se garantiza que el estado que se quiere mantener en el interruptor electrónico deseado mantendrá el estatus deseado mediante el envío de los bytes recibidos por los mecanismos ya indicados. El dispositivo empleado y que cumple con las condiciones aquí expresadas es el **74LS155N**.

- **Bloques.** Cada uno de los bloques configurados en la *interfaz electrónica*, según se muestra en el modelo conceptual tiene como propósito el agrupar a 8 interruptores electrónicos, su estructura se describe en el siguiente apartado.

- **Fuente de alimentación.** Para completar el modelo conceptual es obvio que todos los componentes tienen que ser alimentados eléctricamente para que puedan operar. Por tratarse de dispositivos lógicos, se requiere de una fuente de alimentación del orden de 5 volts con una corriente del orden de 1.5 A para garantizar que todos los componentes de la *interfaz electrónica* operen de forma adecuada.

V.5 Descripción general de bloques de módulos

Cada uno de los bloques que se han indicado en el apartado anterior, se encuentran estructurados de acuerdo a la Ilustración V-4.

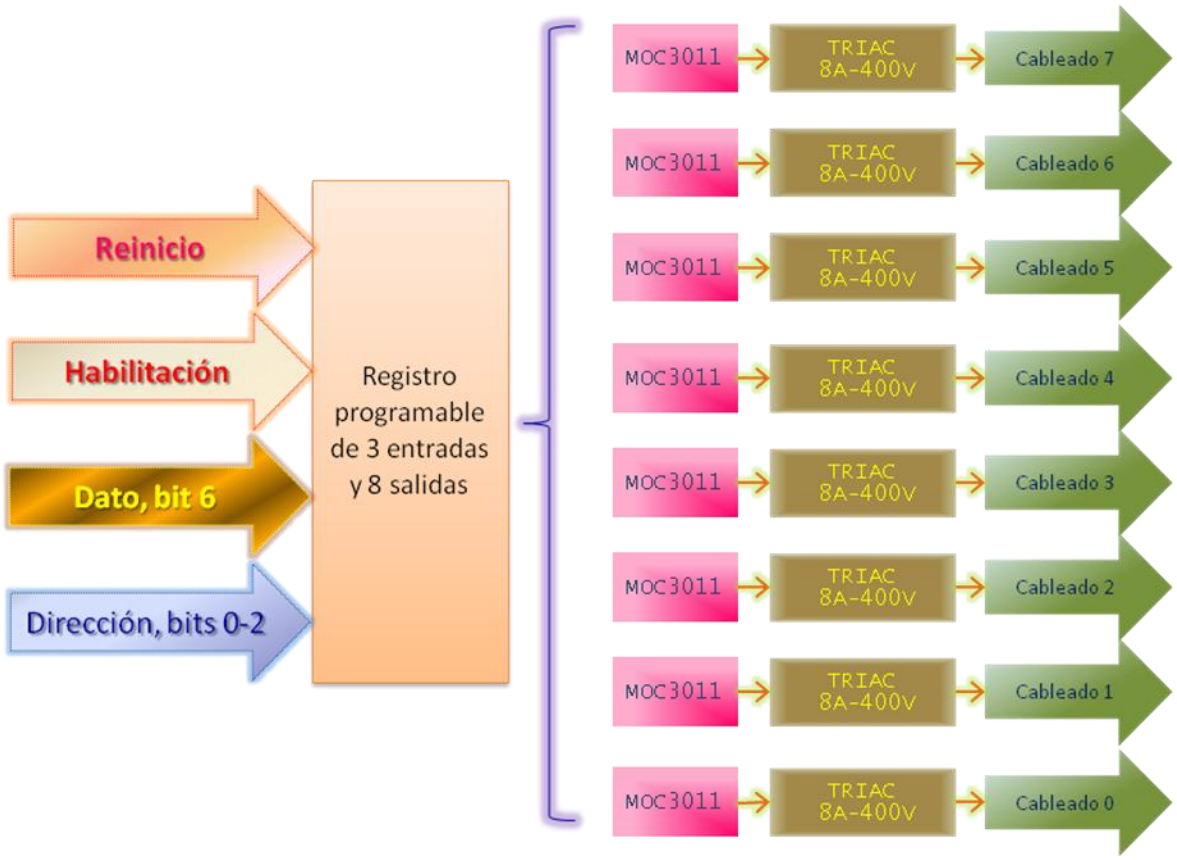


Ilustración V-3. Diagrama general de bloque

En esta ilustración se puede observar que cada bloque se encuentra suministrado por 4 elementos principales:

- **Dirección.** Se encuentra formado por un conjunto de 3 bits que corresponden a los tres bits de menor orden del byte de datos recibidos desde el puerto serial y se utilizan para direccionar al módulo que se quiere actualizar en relación a la conmutación electrónica de suministro eléctrico.

- **Dato**. Es el valor que representa el estado que mantendrá el módulo seleccionado mediante los bits de dirección. Un **"1"** lógico alimentará eléctricamente al módulo y un **"0"** lógico, le cortará el suministro eléctrico.

- **Habilitación**. Es la señal que permite al registro programable asumir los valores que se actualizarán de acuerdo a la entrada en cuanto a la dirección del registro o latch interno que retendrá la situación del módulo. Esta señal es de flanco de transición de activación de alta a baja.

- **Reinicio**. Es una señal que provoca que todos los componentes del bloque sean reiniciados, es decir, las salidas del registro serán **"0"** y por lo tanto, se cortará el suministro eléctrico de los módulos.

La siguiente etapa del módulo es un registro programable de 3 entradas que sirven como direccionamiento del latch de salida que por su composición y configuración electrónica es de 8 salidas, para el caso de este proyecto, se ha seleccionado al dispositivo **74LS259** que cumple con esta función. Tiene como valores de entrada a la dirección del latch que se desea modificar y el dato que se ubicará en la dirección seleccionada en el momento en que se genere la señal de habilitación.

Cada una de las salidas están ligadas a un latch o flip-flop interno que servirá como un elemento de memoria que en su salida mantendrá el valor que se le haya asignado hasta que ocurra un cambio de dato programado o cuando se reciba una señal de reinicialización.

Cada salida se conecta a un dispositivo opto acoplador (también llamado opto aislador) como ya se indicó en el Capítulo III. Su función será pues, la de encender o apagar en combinación con el TRIAC como dispositivo de conducción bidireccional de corriente alterna. Para este proyecto se ha empleado el dispositivo **MOC3011** que ha sido diseñado para activar por disparo a un TRIAC.

El TRIAC, como ya se indicó, se encarga de realizar una conmutación (switch) electrónico, disparado por el valor que guarda el opto acoplador mediante la compuerta de disparo según las hojas de especificaciones de estos dispositivos.

Finalmente, al final de cada salida, se realiza el cableado en el que se conectan físicamente al TRIAC las terminales que conducen la energía eléctrica necesaria para cerrar el circuito y permitir así el flujo eléctrico hacia los módulos. Se ha considerado utilizar a un TRIAC de 8 amperes con una tolerancia de 400 volts por estimar que el consumo máximo por módulo no rebasa los 4 amperes, pero ha ocurrido ocasiones en las que para algunos tipos de proyectos y concretamente de tipo "de potencia", el consumo se dispara y eso podría ocasionar problemas; por ello se está considerando 8 amperes para tener margen ante estas situaciones, aunque sería conveniente que se destinara un módulo o bloque de módulos para que se empleen para este tipo de proyectos y aumentar la capacidad del TRIAC para ese (os) módulo (s) en especial.

Como resultado de lo que hasta ahora se ha expresado, en la siguiente tabla se muestra la relación que hay entre bloques y módulos.

Bloque	Módulos
0	0,1,2,3,4,5,6,7
1	8,9,10,11,12,13,14,15
2	16,17,18,19,20,21,22,23
3	24,25,26,27,28,29,30,31
4	32,33,34,35,36,37,38,39
5	40,41,42,43,44,45,46,47
6	48,49,50,51,52,53,54,55
7	56,57,58,59,60,61,62,63

Tabla V-3. Relación de bloques y módulos

V.6 Control de la *interfaz*

Ya se ha indicado que el control de la *interfaz* se realiza mediante un micro-controlador **PIC16F84A**. En las próximas líneas se incluyen las características más importantes de este dispositivo, justificando su uso y describiendo un poco más el detalle de la manera en que opera el software que se ha desarrollado para los fines de este proyecto.

Este dispositivo se puede encontrar en distintos tipos de encapsulado. En este proyecto es utilizado el tipo **PDIP**, se encuentra constituido por 18 terminales. Es un dispositivo desarrollado con una tecnología **RISC** de alto desempeño. Sus principales características son:

- Solo hay que aprender 35 instrucciones de lenguaje ensamblador.
- Todas las instrucciones se ejecutan en un solo ciclo a excepción de las instrucciones de salto que utilizan dos ciclos.
- La velocidad de operación es de 4 Mhz de reloj en corriente directa por lo que los ciclos de instrucción son de 4ns.
- Cuenta con una memoria para programa de 1024 palabras.
- Cuenta con 68 bytes de RAM.
- Cuenta con 64 bytes de datos en EEPROM.
- Palabras de instrucción de 14 bits de longitud.
- Bytes de datos de un ancho de 8 bits.
- Cuenta con 15 registros de funciones especiales de hardware.
- Cuenta con una pila implementada en hardware de una profundidad de 8 niveles.
- Se puede utilizar el direccionamiento directo, indirecto y relativo.
- Cuenta con cuatro fuentes de interrupción: pin de interrupción externa RB0/INT. Sobreflujo de cronómetro TMRO. Interrupciones en los cambios PUERTO B <7:4>. Escritura completa en EEPROM.
- Cuenta con 13 pines de control de dirección de entrada/salida.
- Corriente suficiente para manejo de leds. 25 mA por pin.
- Un contador preescalar programable de 8 bits en TMR0.

En la siguiente Ilustración se muestra el esquema de distribución de pines de este dispositivo. De éste se especificará las señales que se usan en la *interfaz electrónica* y su propósito dentro del proyecto en general.

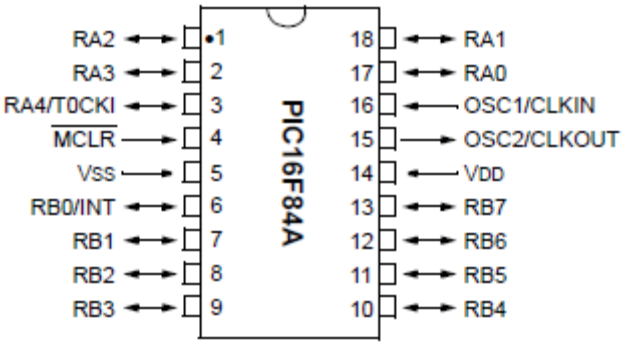


Ilustración V-4. Distribución de salidas del PIC16F84A

En la siguiente tabla se muestra el detalle del significado de cada una de las terminales del micro controlador.

Número de pin	Nombre del pin	Descripción
16	OSC1/CLKI	Entrada de Oscilador de cristal / entrada a una fuente de reloj externa
15	OSC2/CLKOUT	Salida del oscilador de cristal. Conecta al cristal o al resonador en modo de oscilador de cristal. En modo RC, el pin OSC2 da salida a CLKOUT, que tiene ¼ de la frecuencia de OSC1 y denota la tasa de ciclos de instrucción.
4	MCLR	Limpieza maestra (Reset). Entrada de voltaje programable. Este pin es un Reset en valor "bajo".
17	RA0	PORTA: Es un puerto bidireccional También se puede seleccionar como entrada de reloj para el cronómetro/contador TMR0. La salida es de tipo drenaje abierto.
18	RA1	
1	RA2	
2	RA3	
3	RA4/TOCKI	
6	RB0/INT	PORTB: Es un puerto bidireccional. Puede ser programado con software para poner en alta impedancia a todas las entradas. RB0/INT también se puede seleccionar como un pin de interrupción externa. RB4, RB5, RB6 y RB7: Pines de interrupción en el cambio de valor. RB6: Programación serial del reloj RB7: Programación serial de datos.
7	RB1	
8	RB2	
9	RB3	
10	RB4	
11	RB5	
12	RB6	
13	RB7	
5	VSS	Tierra de referencia para la lógica y los pines de E/S
14	VDD	Suministro de energía eléctrica para la lógica y los pines.

Tabla V-4. Significado de las terminales del dispositivo

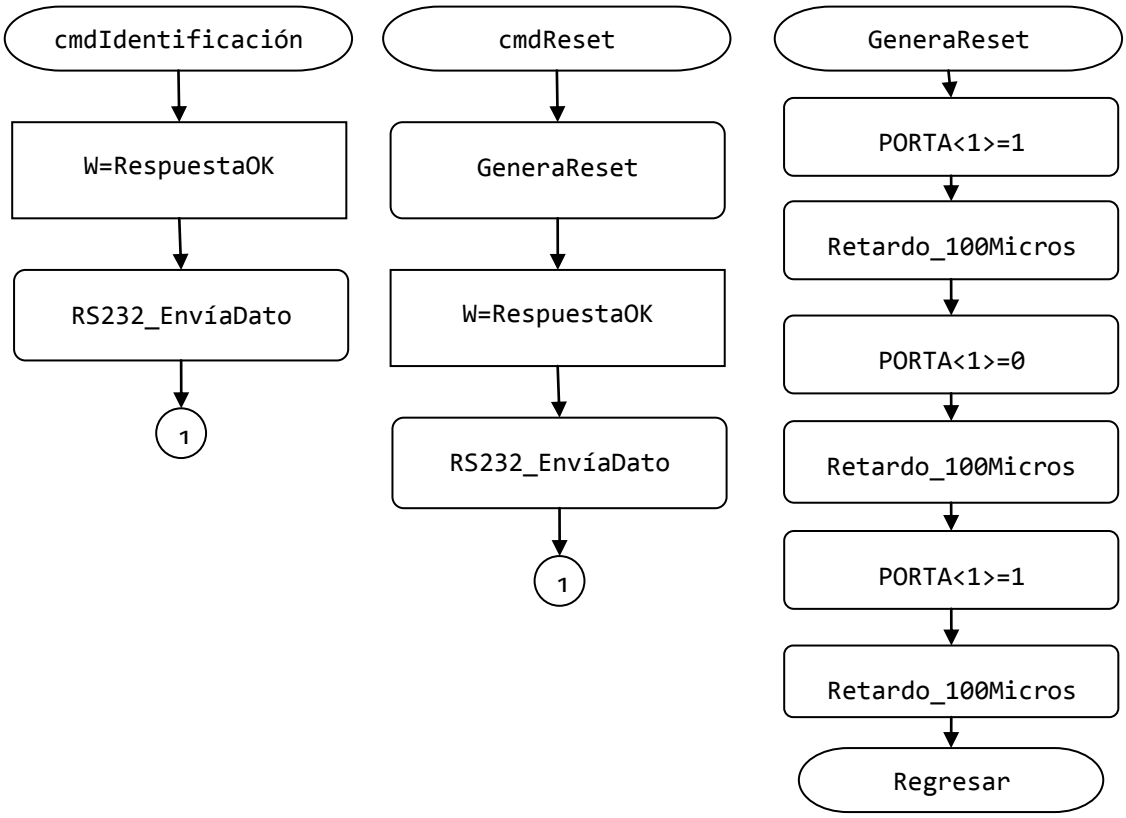
De la tabla anterior y para propósitos de asociación con los valores utilizados dentro del

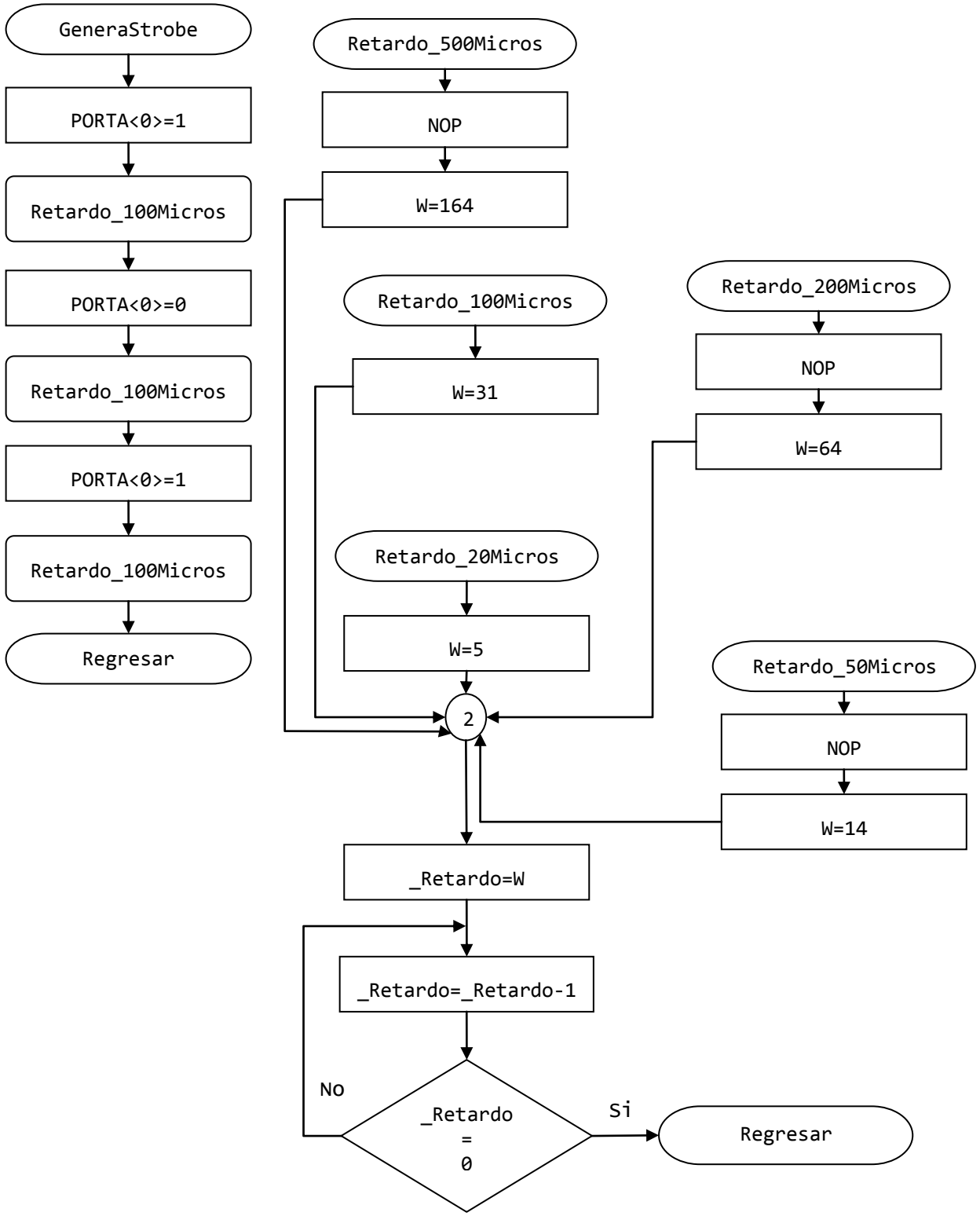
proyecto se puede asociar con los pines del micro controlador y del proyecto, según se ha descrito hasta ahora, se muestra la siguiente tabla de uso de las terminales y de los bits dentro de la *interfaz electrónica*.

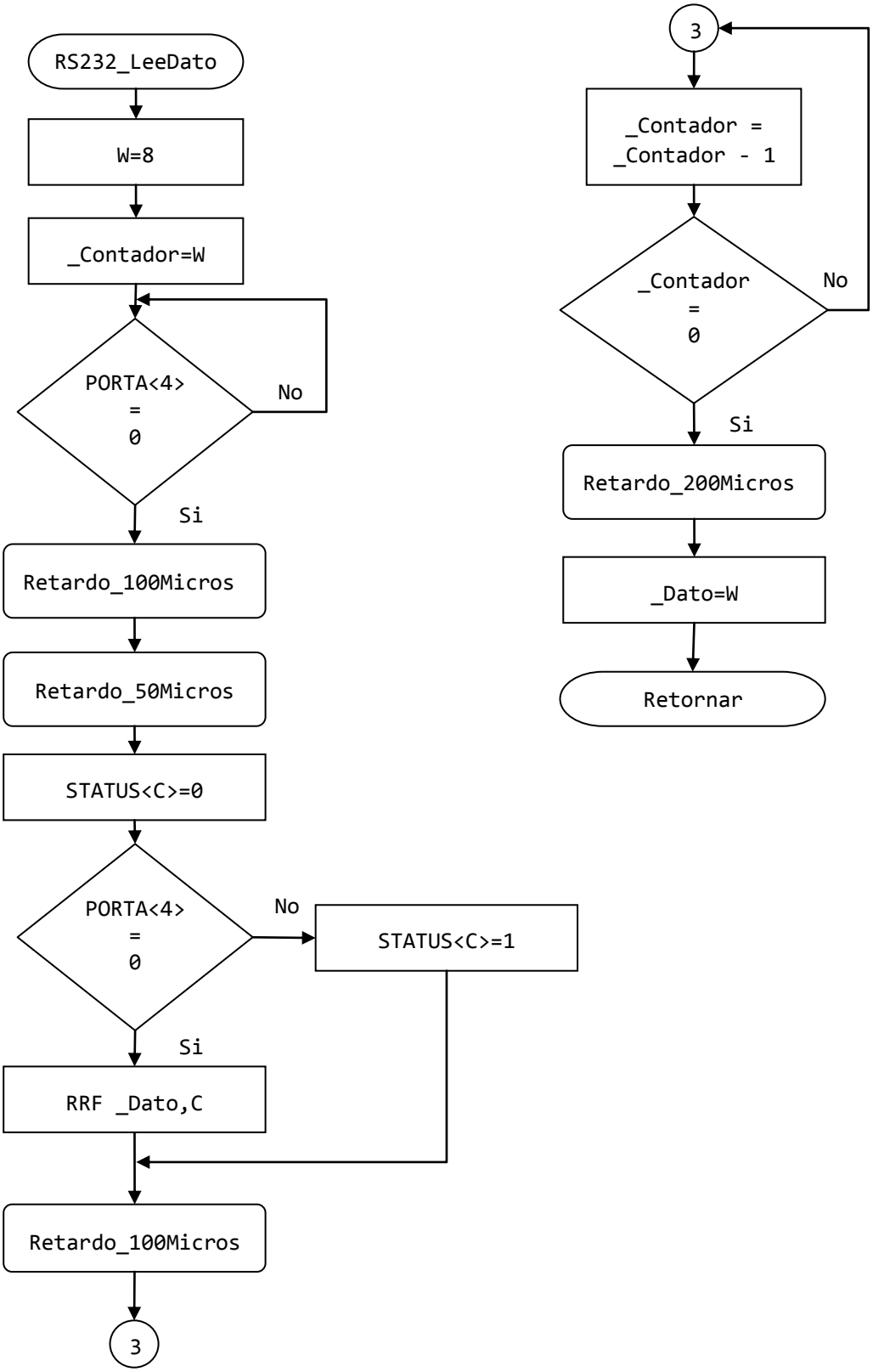
Bit	Pin	Significado
0	RB0 – 6	Dirección 0
1	RB1 – 7	Dirección 1
2	RB2 – 8	Dirección 2
3	RB3 – 9	Bloque 0
4	RB4 – 10	Bloque 1
5	RB5 – 11	Bloque 2
6	RB6 - 12	Dato
	RA0 - 17	Habilitación
	RA1 – 18	Reinicio
	RA3 – 1	TxD
	RA4 – 2	RxD

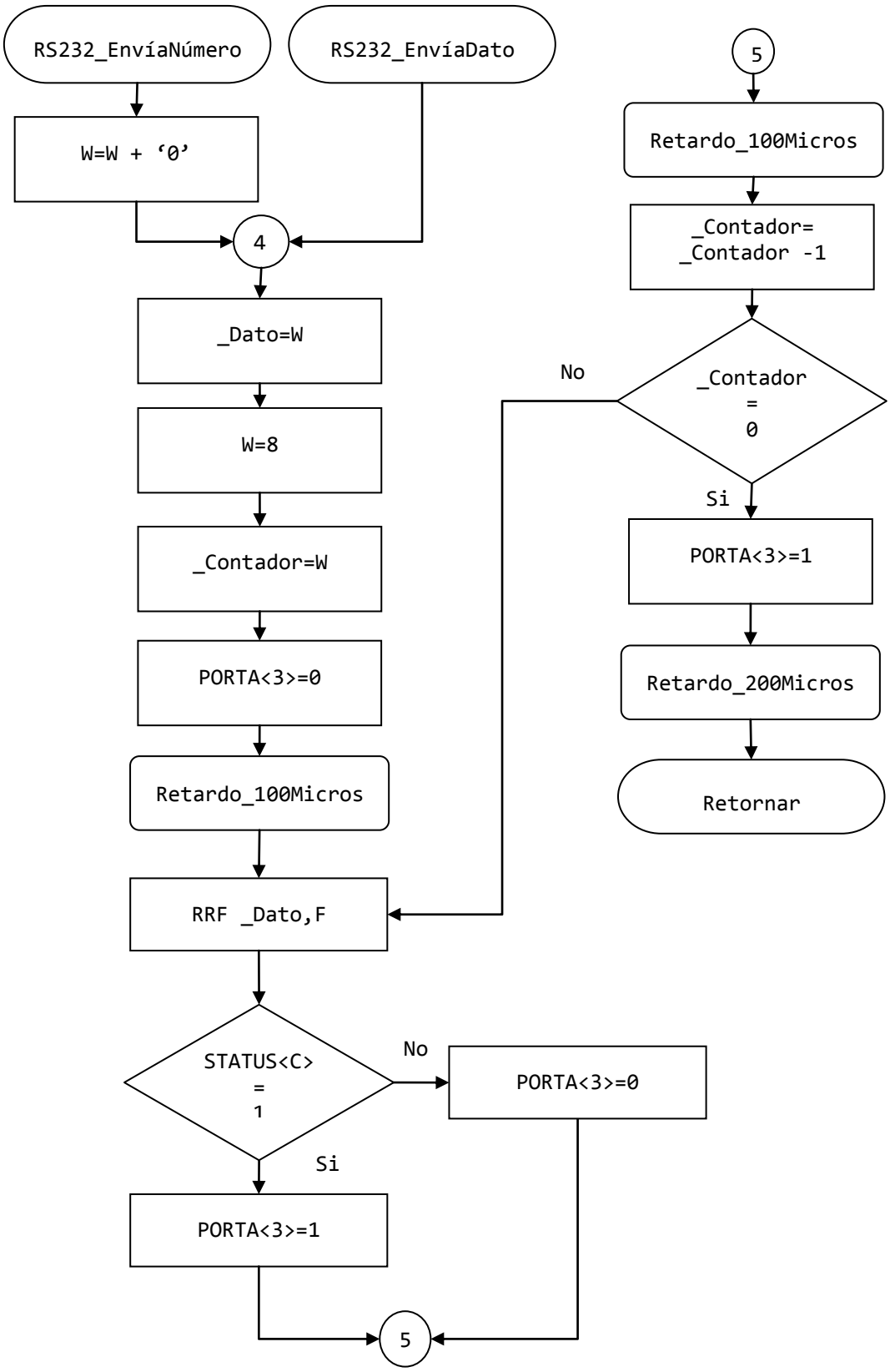
Tabla V-5. Pines utilizados asociados al protocolo de comunicación

V.7 Diagramas de flujo del software el Micro controlador




GeneraStrobe
PORTA<0>=1
Retardo_100Micros
PORTA<0>=0
Retardo_100Micros
PORTA<0>=1
Retardo_100Micros
Regresar
Retardo_500Micros
NOP
W=164
Retardo_100Micros
W=31
Retardo_200Micros
NOP
W=64
Retardo_20Micros
W=5
2
Retardo_50Micros
NOP
W=14
_Retardo=W
_Retardo=_Retardo-1
No
_Retardo
=
0
Si
Regresar


RS232_LeeDato
W=8
_Contador=W
PORTA<4>
=
0
No
Si
Retardo_100Micros
Retardo_50Micros
STATUS<C>=0
PORTA<4>
=
0
No
STATUS<C>=1
Si
RRF _Dato,C
Retardo_100Micros
3
3
_Contador =
_Contador - 1
_Contador
=
0
No
Si
Retardo_200Micros
_Dato=W
Retornar


RS232_EnvíaNúmero
RS232_EnvíaDato
W=W + ‘0’
4
_Dato=W
W=8
_Contador=W
PORTA<3>=0
Retardo_100Micros
RRF _Dato,F
STATUS<C>
=
1
No
PORTA<3>=0
Si
PORTA<3>=1
5
5
Retardo_100Micros
_Contador=
_Contador -1
_Contador
=
0
No
Si
PORTA<3>=1
Retardo_200Micros
Retornar

V.8 Listado del código fuente del programa del PIC16F84A

```
;
;Proyecto de comunicación y sincronización del SAEALABFI con la interfaz
;electrónica versión 4
;
;Para la implementación de este proyecto, se debe configurar al equipo de cómputo
;para que se utilice
; el puerto serial COM1 con los siguientes parámetros de comunicación:
;     - Velocidad de transmisipon de 9600 baudios. La duración de cada bit será
;       104 µSeg.
;     - Un bit de inicio o Start a nivel bajo
;     - Dato de 8 bits
;     - Sin paridad
;     - Dos bits de final o Stop a nivel alto

      __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC

      LIST  P=16F84A
      INCLUDE     <P16F84A.INC>
      INCLUDE     <RETARDOS.INC>
      INCLUDE     <COMUNICACIONES.INC>

      CBLOCK 0X0C
            _Retardo
            _Contador
            _Dato
      ENDC

      #DEFINE     RespuestaOK             0xFF
      #DEFINE     RespuestaError          0XF0
      #DEFINE     RespuestaComandoError   0XF1
      #DEFINE     ComandoIdentificación   0xF0
      #DEFINE     ComandoReset                  0xF1

      ORG         0x0000
      GOTO  Inicio

      ORG   0x0005
Inicio
      ; Se activa el banco de memoria 1 para configurar los puertos
      ; A y B.
      BSF         STATUS,RP0

      ; Strobe --------| - TRISA,0  Salida
      ; Reset --------|| - TRISA,1  Salida
      ; xxxxx -------||| - TRISA,2  Salida
      ; Tx ---------|||| - TRISA,3  Salida
      ; Rx --------||||| - TRISA,4  Entrada
      ;            vvvvv

      CLRF  TRISA       ; Primero se configuran todas las señales como
                        ; salida.
      BSF   TRISA,4     ; Ahora se configura la señal 4 como de entrada.

      CLRF  TRISB       ; PUERTOB <0:7>, se establece a cero por lo que cada línea
                        ; se ha configurado como salida

      ; Una vez configurados los puertos, seleccionamos el banco de memoria 0 para
```

```
      ; poder continuar con la lógica del programa.

      ; Seleccionamos el banco de memoria 0
      BCF         STATUS,RP0


      ; Primer paso.- Iniciar el puerto A para que al momento de ser cambiada
      ; la dirección de los puertos no se genere una sañal de habilitación

      ; Strobe---------|
      ; Reset---------||
      ; xxxx---------|||
      ; Tx----------||||
      ; Rx---------|||||
      ;            VVVVV
      BSF         PORTA,0
      BSF         PORTA,1
      BSF         PORTA,3

      ; De manera similar al caso anterior, se establecen valores de inicio
      ; para las líneas que corresponden al puerto B, para este caso, cada
      ; línea se pondrá en cero.

      CLRF  PORTB

      ; Ahora, activamos el banco de memoria 1 desde el que se configurarán
      ; las líneas de los puertos A y B.
      ; un bit 0 en la posición de la línea, la habilitará para salida, mientras
      ; que un 1 corresponderá a una entrada.

      ; Paso 2: Generación de señal de Reset
      CALL  GeneraReset

Principal
      CALL  RS232_LeeDato     ; Lee un dato del RS232 y lo deja en W

      ; Si el bit 7 del dato recibido está encendido, entonces se irá
      ; a procesar la interpretación de comandos, sino es así, entonces
      ; se coloca el dato recibido en el puerto B.
      BTFSC _Dato,7
      GOTO  InterpretarComando

      ; si no comienza con "1", entonces se trata de dato
      MOVWF PORTB                   ; El dato leído se coloca en el puerto B
      CALL  GeneraStrobe                  ; Se genera la señal de habilitación
      MOVLW RespuestaOK             ; Respuesta a la computadora de que no hubo
                                    ; error
      CALL  RS232_EnvíaDato
      GOTO  Principal

InterpretarComando
      ; El comando es la identificación de la interfaz electrónica (IE)
      MOVF  _Dato,W
      SUBLW ComandoIdentificación
      BTFSC STATUS,Z
      GOTO  CmdIdentificación

      ; El comando es el reset de la Interfaz electrónica
      MOVF  _Dato,W
      SUBLW ComandoReset
      BTFSC STATUS,Z
```

```
      GOTO  CmdReset

      ; Si no hay un comando válido, se retorna el código de error
      MOVLW RespuestaComandoError
      CALL  RS232_EnvíaDato

      GOTO  Principal

CmdIdentificación
      ; Acciones que se realizan cuando se selecciona el comando de
      ; identificación de interfaz electrónica.
      MOVLW RespuestaOK
      CALL  RS232_EnvíaDato
      GOTO  Principal

CmdReset
      ; Acciones que se realizan cuando se selecciona el comando de
      ; Reinicialización de la Interfaz Electrónica.
      CALL  GeneraReset
      MOVLW RespuestaOK
      CALL  RS232_EnvíaDato
      GOTO  Principal

; Subrutina para generar un reseteo en la interfaz electrónica:
; primero pone en alto la línea por 100 µSeg, luego en bajo otros
; 100 µSeg y, finalmente, en alto otros 100 µSeg.
GeneraReset
      BSF   PORTA,1
      CALL  Retardo_100Micros
      BCF   PORTA,1
      CALL  Retardo_100Micros
      BSF   PORTA,1
      CALL  Retardo_100Micros
      RETURN

; Subrutina para generar un strobo en la interfaz electrónica, para
; que los datos escritos en el puerto B sean reflejados:
; primero pone en alto la línea por 100 µSeg, luego en bajo otros
; 100 µSeg y, finalmente, en alto otros 100 µSeg.
GeneraStrobe
      BSF   PORTA,0
      CALL  Retardo_100Micros
      BCF   PORTA,0
      CALL  Retardo_100Micros
      BSF   PORTA,0
      CALL  Retardo_100Micros

      RETURN
```

V.9 Tarjetas de la *interfaz electrónica*

La *interfaz electrónica* quedará distribuida en dos tipos de tarjeta: la primera corresponderá a la del módulo principal desde la que se realizará todo el proceso de conversión de señales y control de los componentes de la Interfaz y, las tarjetas de bloque, mismas que tendrán el mismo aspecto para cada bloque y lo único que les hará diferente de las demás, será la conexión en la tarjeta principal pues será allí donde se generará la señal de habilitación y como ya se ha mencionado, esta será única al momento de ejecutar una acción pues será el

punto de partida que garantizará que no se encienda o apague más de un módulo a la vez.

V.9.1 Tarjeta principal

Esta tarjeta recibe desde un cable DB9 las 3 señales necesarias para que se pueda comunicar el equipo de cómputo mediante el puerto serial; estas señales corresponden a la tierra física, la señal de transmisión y la de recepción que en conjunto implementan el protocolo y mecanismo de comunicación entre el equipo de cómputo y la *interfaz electrónica*.

Esta tarjeta basa su operación según lo mostrado en el apartado V.1, en donde se ha incluido el modelo conceptual de la *interfaz electrónica*. Aquí es donde se realiza la conversión de señales RS232 a TTL mediante el uso de un dispositivo MAX232 y, en seguida se procesan las señales recibidas y convertidas con el uso de un micro controlador de baja integración correspondiente al **PIC16F84A** y mediante el programa de control que ya se ha mostrado mediante diagramas de flujo, generará las señales necesarias para que se active el módulo seleccionado desde el programa de solución que ha sido diseñado para cubrir las necesidades del Laboratorio Abierto de electrónica.

La señal de habilitación la realizará mediante el uso de un decodificador de 3 a 8 salidas con salidas invertidas, pues los latches de las tarjetas de bloque se activan en flanco de bajada; el dispositivo que se utiliza para este fin es el 74LS155N.

Dentro de la tarjeta principal se ha incluido 8 conectores tipo SIL o headers de 8 pares de pines machos rectos que sirven para conectarse allí mediante conectores de este tipo y cables tipo listón de 16 hilos a las tarjetas de bloque. La correspondencia de bloques es como se muestra en la ilustración V-4. En la parte inferior derecha se encuentra la conexión para el bloque 0 (con el nombre de *Bloque 1*), en seguida se encuentran en forma ascendentes los conectores para los bloques restantes (nombrados como *Bloque 2 y hasta el Bloque* 8).

Las señales que en cada una de ellas se coloca es en el siguiente orden: +5 Volts, Tierra, Bit de dirección 0, bit de dirección 1, bit de dirección 2, Dato, Reset y Señal de habilitación; todas las terminales se encuentran ubicadas de derecha a izquierda se ubican referenciadas por las muescas de sesgadas de la representación gráfica del SIL o header.

El diseño del PCB de esta tarjeta se muestra en el Apéndice A.

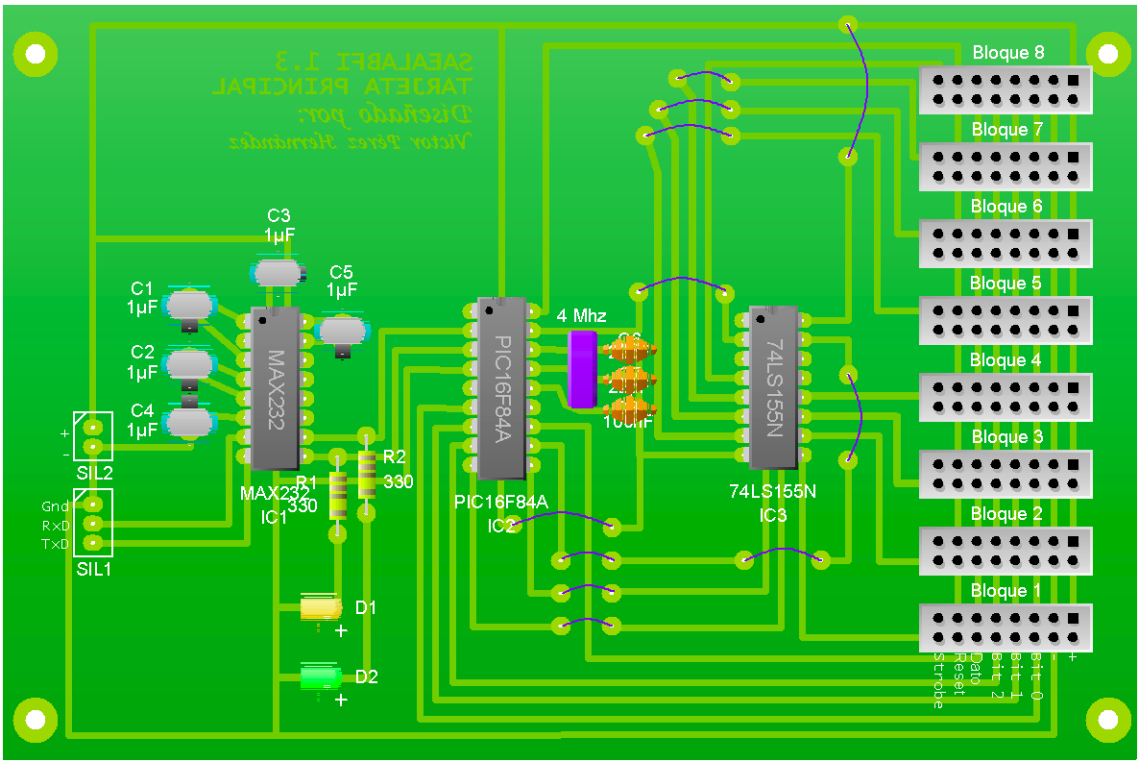


Ilustración V-5. Aspecto de la tarjeta principal

V.9.1.1 Lista de materiales

Para la implementación de esta tarjeta electrónica que corresponde a la tarjeta principal de la *interfaz electrónica* se requiere de los materiales que enseguida se muestran. Se requiere hacer hincapié en que para la alimentación de la *interfaz electrónica* se requiere de una fuente de alimentación externa de 5 volts a 1.5 Amperes, y en la tarjeta electrónica se ha incluido un conector SIL de dos terminales para este fin como se aprecia en la Ilustración V-6.

Cantidad	Parte
5	Capacitores electrolíticos de 1µF a 16 volts
2	Capacitor cerámico de 22 pF
1	Capacitor cerámico de 100 nF
8	Conectores SIL o headers machos planos de 8 pines de doble hilera
1	Conector SIL o header macho plano de 3 pines
1	Conector SIL o header macho plano de 2 pines
1	Cristal de cuarzo de 4 Mhz
1	Tarjeta fenólica de 15x10 cm
1	Dispositivo **MAX232**
1	Micro controlador **PIC16F84A**
1	Decodificador **SN74LS155N**
2	Resistencias de 330 Ω a ¼ de Watt
2	Leds de 5 v
2	Bases de 16 terminales para circuitos integrados
1	Base de 18 terminales para circuito integrado

Tabla V-6. Lista de materiales de la tarjeta principal

V.9.2 Tarjeta de bloque

La tarjeta de bloque corresponde a cada uno de los bloques descritos con anterioridad, dentro de su estructura se encuentra un dispositivo **SN74LS259** que corresponde a un latch programable mediante una dirección de entrada de 3 bits con los cuales se puede mantener en "memoria" a 8 valores binarios, mismos que son utilizados para activar o desactivar, según sea el caso al led interno de cada uno de los optoacopladores **MOC3011** que se utilizan para activar o desactivar el flujo de corriente eléctrica controlada por el correspondiente TRIAC que en las terminales asociadas a M1 y M2 da entrada al flujo eléctrico.

Cada tarjeta de bloque se conecta a la tarjeta principal mediante un cable de listón y un conector de tipo SIL de 8 terminales en doble hilera que unirá a esta placa con la principal en el conector que se ha incluido como salida para los bloques del 0 al 7 (formalmente Módulo 1 a Módulo 8).

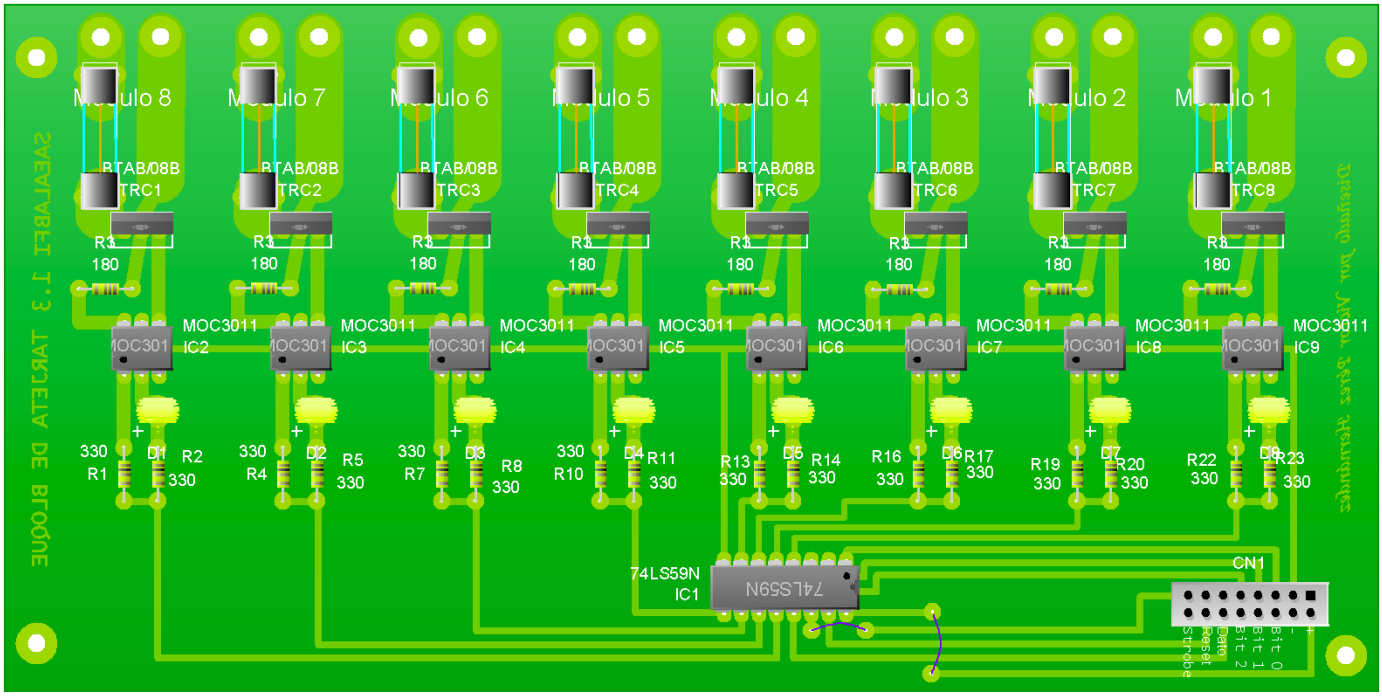


Ilustración V-6. Aspecto de la tarjeta de bloque

Las señales que se envían en este conector se indican en la siguiente tabla:

Terminal	Señal
1	+5 Volts
2	Tierra
3	Bit de dirección 0
4	Bit de dirección 1
5	Bit de dirección 2
6	Dato
7	Reset
8	Habilitación

Tabla V-7. Distribución de señales entre Las tarjetas de la interfaz electrónica

El diseño del PCB de esta tarjeta se muestra en el Apéndice A.

V.9.2.1 Lista de materiales

Para poder implementar cada una de las tarjetas de bloque, se requiere de la siguiente lista de materiales. Previo a esto es necesario recalcar que para el caso de este proyecto se requiere de 4 tarjetas de bloque (0 a 3) que en conjunto soportan a un total de 32 módulos ya que físicamente se encuentran instalados 30.

La *interfaz electrónica* se encuentra preparada para conectar hasta 8 tarjetas de bloque (0 a 7) para dar soporte a un total de 64 módulos de trabajo.

Cantidad	Parte
1	Tarjeta fenólica de 20x10 cm
8	Resistencias de 180 Ω a ¼ de watt
16	Resistencias de 330 Ω a ¼ de watt
1	Latch programable **SN74LS259**
1	SIL o header macho recto de 8 pines de doble hilera
8	Optoacopladores **MOC3011**
8	TRIACs BTA08-400B (8 Amperes a 400 volts)
8	Leds de 5 v
8	Disipadores de calor de tipo clip TO-220
8	Bases para circuito integrado de 6 terminales
1	Base para circuito integrado de 16 terminales
8	Pares de bases para fusible tipo europeo
8	Fusibles tipo europeo de 7.5 A

Tabla V-8. Lista de materiales de una tarjeta de bloque

V.9.3 Materiales complementarios

Adicionalmente a los materiales indicados en las tarjetas principal y de bloque, se requieren de materiales para la conexión entre las tarjetas; estos materiales básicamente son conectores, cables, soldadura, cloruro férrico y herramientas para dicho propósito. En la siguiente tabla, se enlistan dichos materiales.

Cantidad	Parte
1	Conector DB-9 Hembra
1	Concha para conector DB9
2	Metros de cable de listón de 16 hilos
7.5	Metros de cable para conector serial de 5 hilos
1	Conector hembra tipo SIL de 3 terminales
1	Conector hembra tipo SIL de 2 terminales
1	Bloque de zapatas para conectores hembra SIL
1	Litro de Cloruro Férrico
1	Rollo de soldadura
1	Recipiente para creación de PCBs
5	Brocas de distintos calibres

Tabla V-9. Lista de materiales complementarios

V.10 Configuración del cable de conexión entre la computadora y la *Interfaz Electrónica*

Como ya se indicó en el Capítulo III, la manera en que opera un puerto serial está basada en una comunicación asíncrona mediante un protocolo handshaking y una codificación de señales de tipo NRZ. El propósito original de este tipo de puertos fue el de conectar equipos de cómputo personal con otros a distancia mediante un módem, motivo por el cual, se encuentran enfocados a este fin.

Para la conexión de la computadora y la *Interfaz Electrónica,* se empleará un conector DB9 con la configuración de módem nulo (por parte de la computadora) y un conector SIL 3 (por parte de la *Interfaz Electrónica*).

Por la razón expuesta en la Ilustración V-6 se muestra la configuración de las terminales entre el conector DB9 y el conector SIL y es necesario observar que para la conexión sólo se requerirá de 3 señales del conector DB9 a saber: Ground (terminal 5), TxD (terminal 3) y RxD (terminal 2).

En la tabla V-9 se enlistan las señales del conector DB9 con su correspondiente número de terminal que corresponden a las marcadas en la Ilustración V-8.

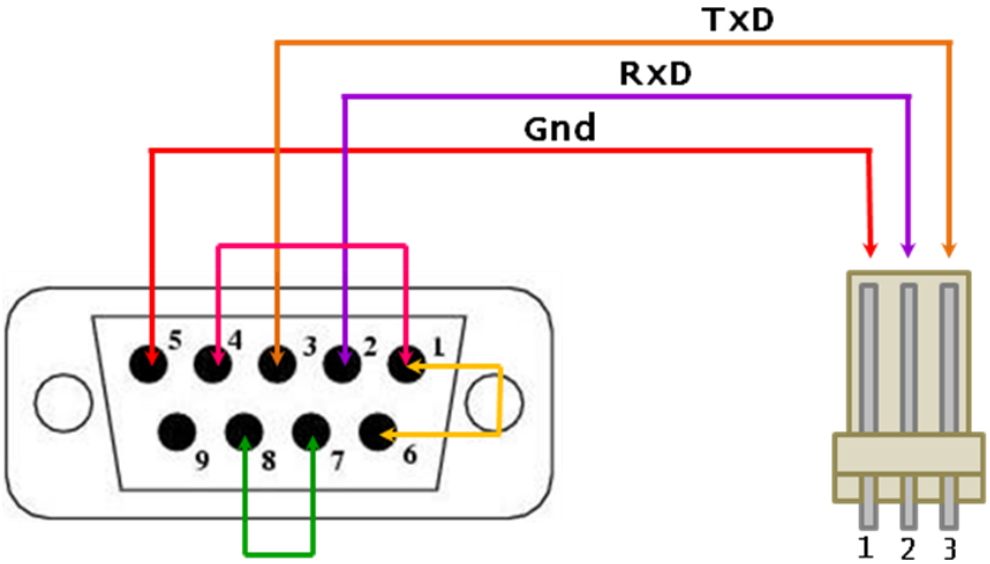


Ilustración V-7. Configuración del cable de conexión serial PC-Interfaz Electrónica

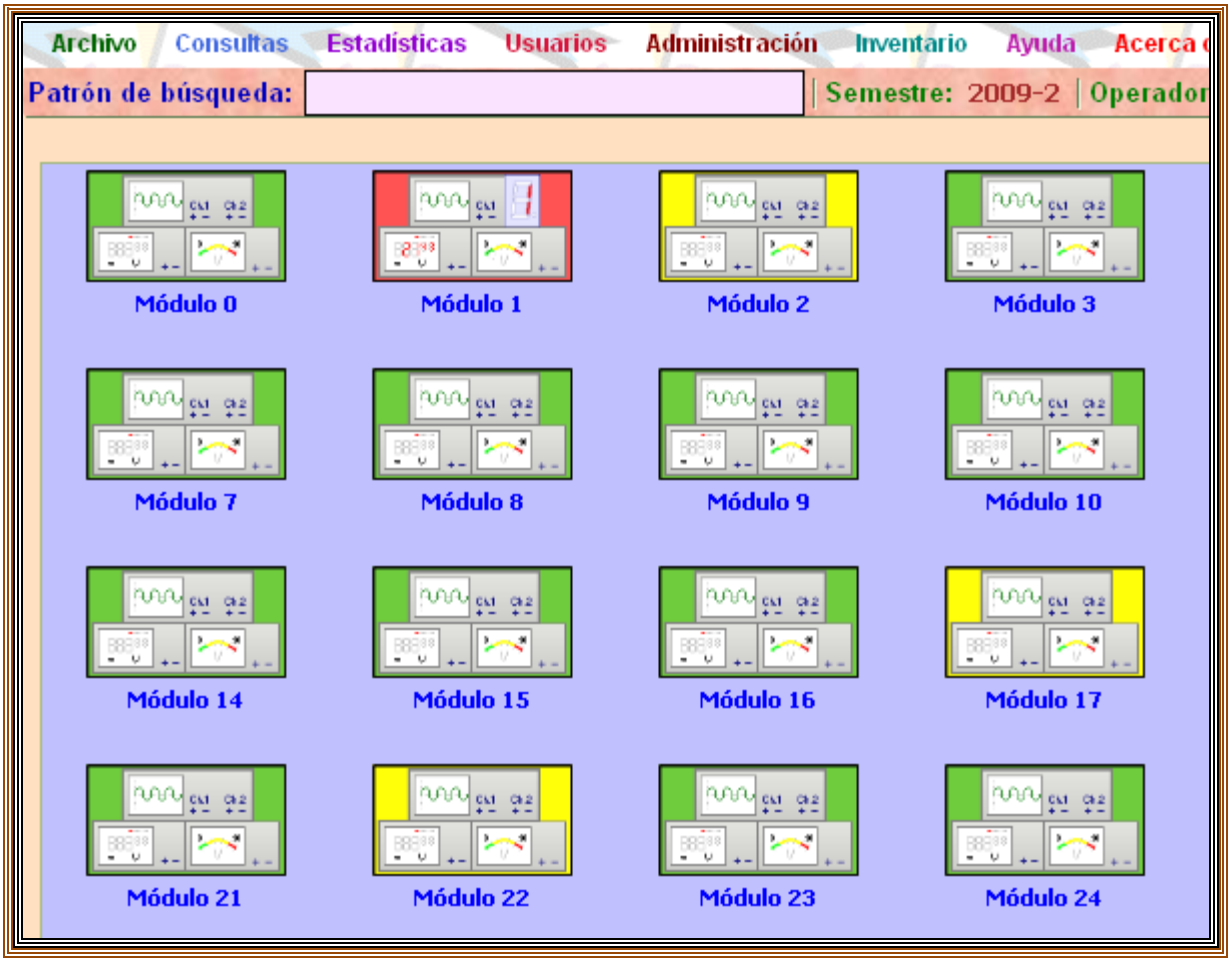
PIN	SEÑAL
1	Data Carrier Detect (DCD)
2	Received Data (RxD)
3	Transmitted Data (TxD)
4	Data Terminal Ready (DTR)
5	Signal Ground (SG)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Ring Indicator (RI)

Tabla V-10. Señales en el conector DB9

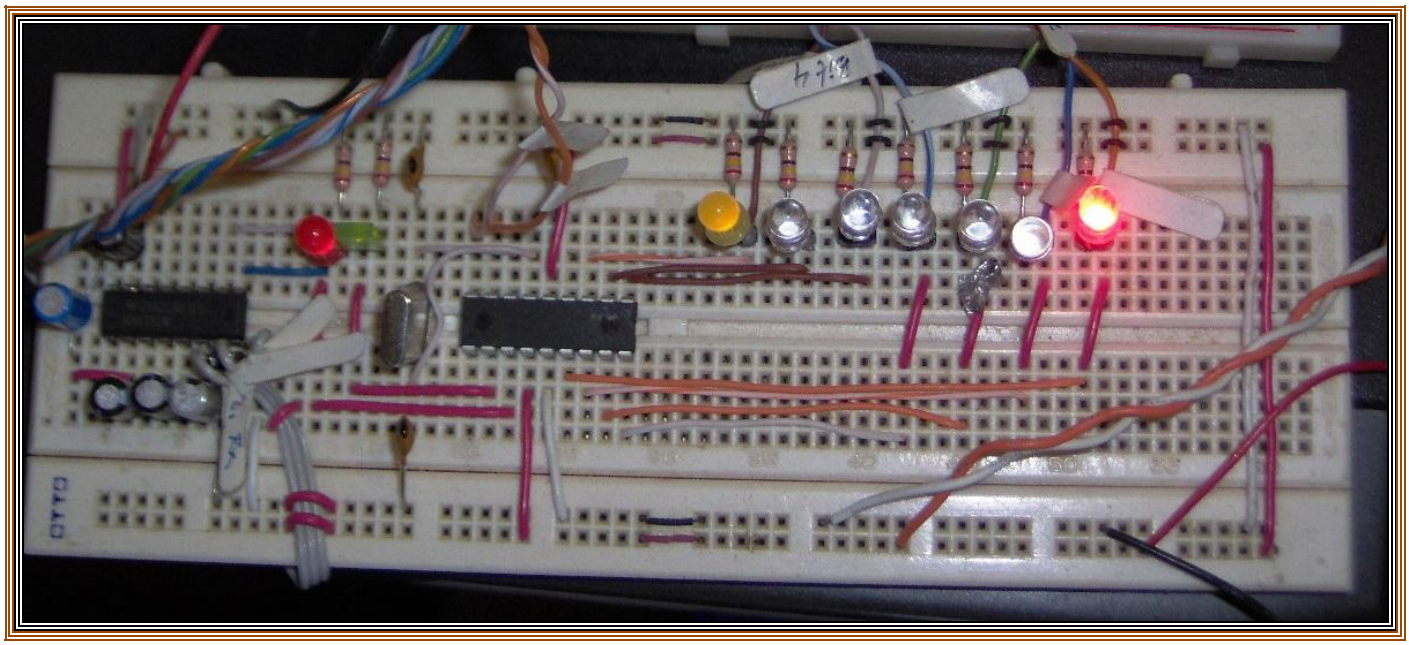
V.11 Prototipo de prueba

Para realizar las pruebas de operación de la *interfaz electrónica*, se ha construido un prototipo en tarjetas protoboard en las que se han distribuido los componentes de las distintas fases en tres tarjetas. La primera de ella contiene la fase de conversión de señales RS232 a TTL y el micro controlador y con el apoyo de leds para mostrar el flujo de señales en la etapa de conversión y la de colocación de datos en la fase de interpretación de señales. En la segunda tarjeta, se ha incluido el decodificador y los latches programables y, para visualizar los datos retenidos por cada uno de los 32 módulos, se ha utilizado un led por cada "módulo" para estar seguro de que en las líneas de habilitación de la etapa de conmutación se encuentren las señales que se han seleccionado con la operación del Sistema. Finalmente, en la tercer tarjeta, se han colocado dos optoacopladores y TRIACs que alimentan a dos focos de manera respectiva para simular al módulo 0 y al módulo 1.

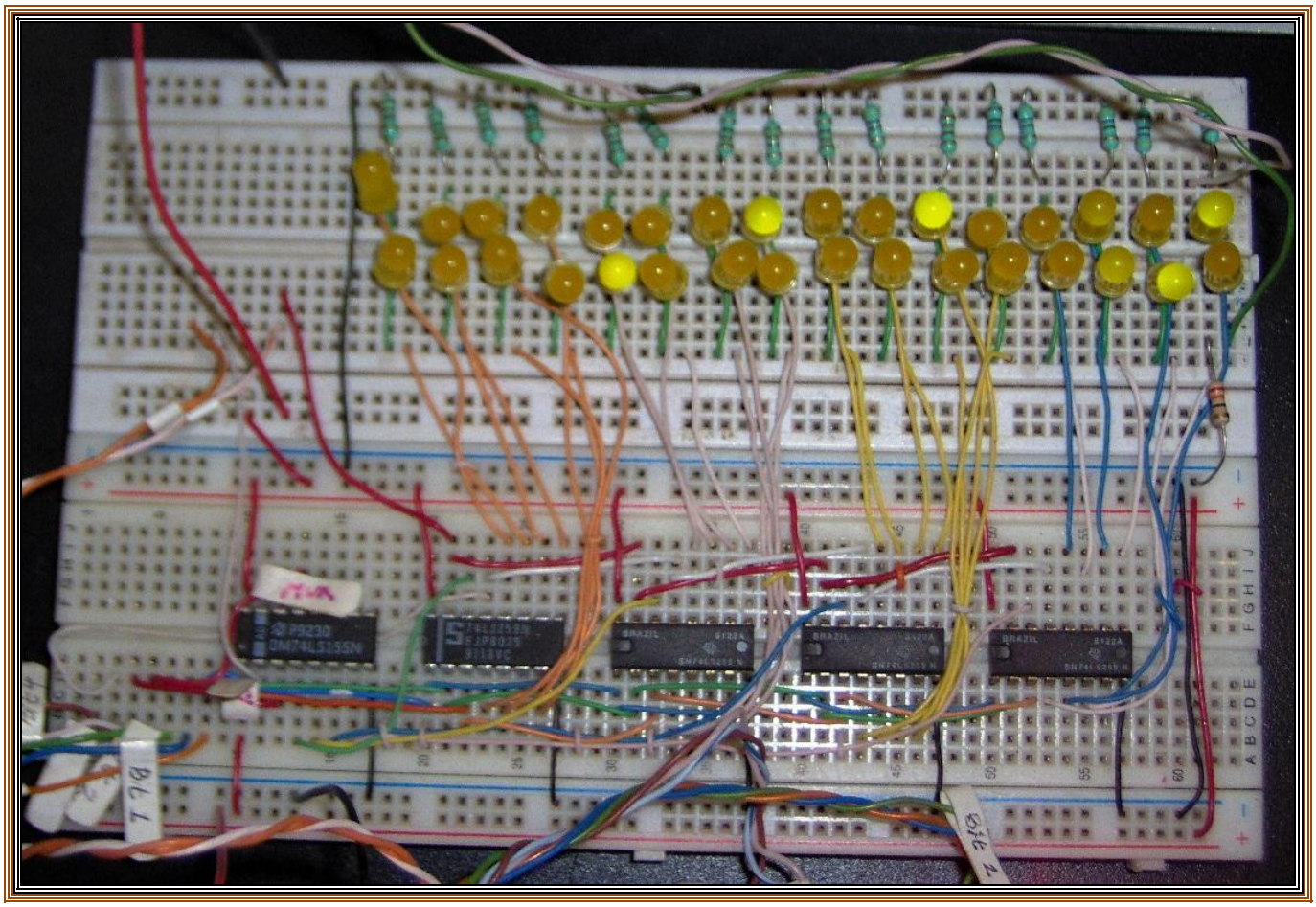
En la siguiente serie de fotografías se muestran los distintos aspectos que guardan cada una de las etapas del prototipo. Para el caso particular, se ejemplifica la activación del Módulo 1 que ha sido asignado a un usuario previamente.



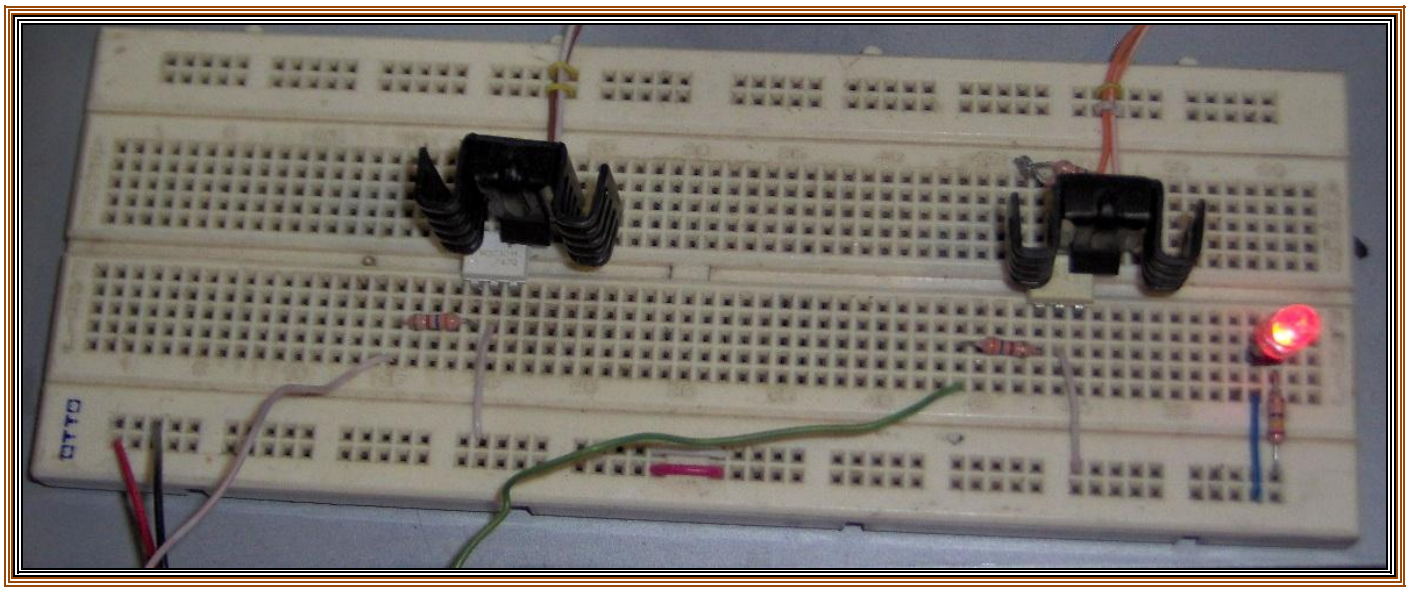
Fotografía V-1. Ajuste en algunos módulos desde el panel



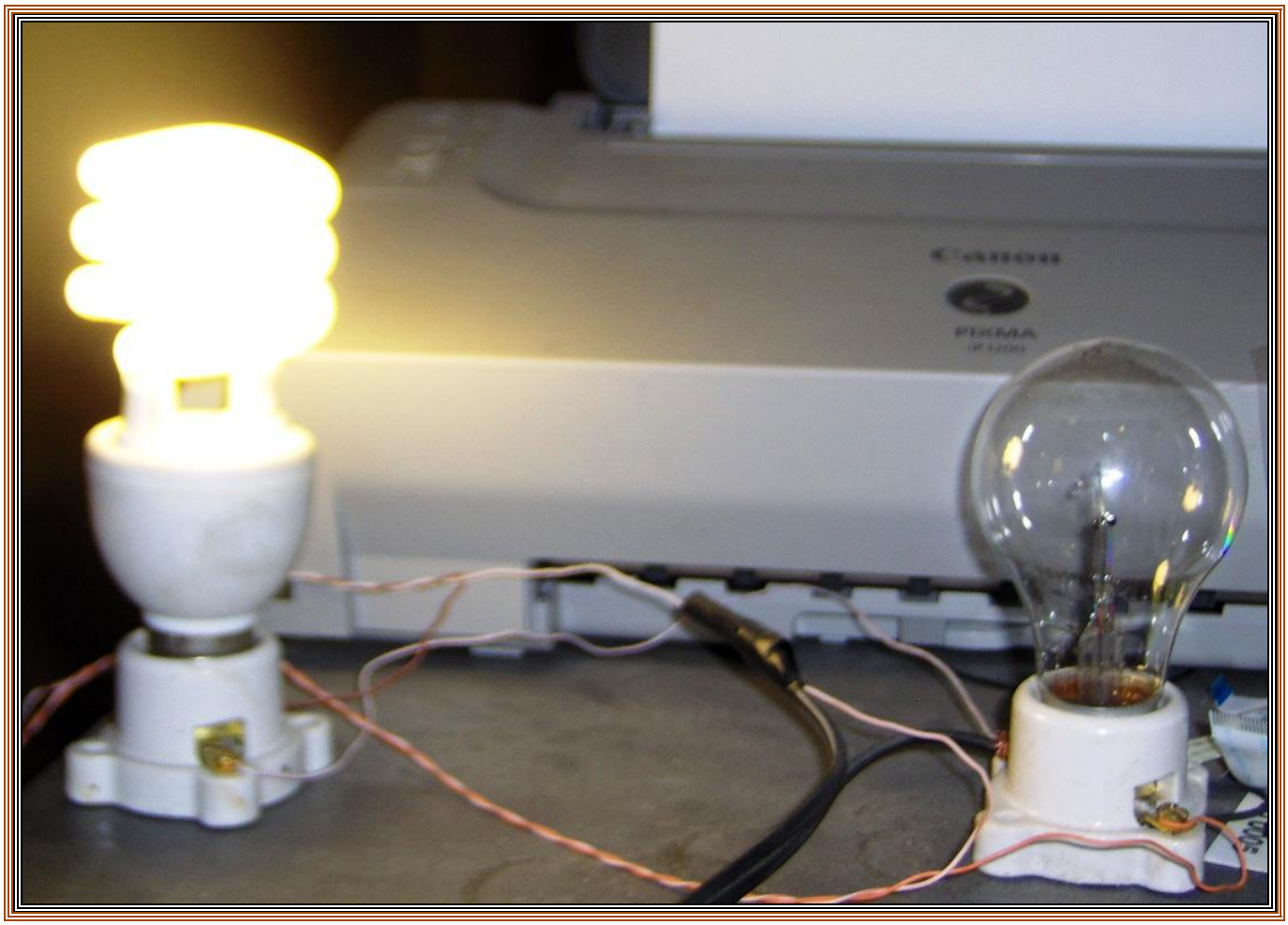
Fotografía V-2. Tarjeta de conversión de señal y control del prototipo



Fotografía V-3. Etapa de retención de información



Fotografía V-4. Tarjeta de alambrado de los optoacopladores y TRIACs



Fotografía V-5. Simulación de carga en el módulo 1.

CONCLUSIONES

Durante la elaboración de este trabajo se han presentado situaciones que en ocasiones se encuentran fuera de alcance resolver; por una parte, las de tipo técnico y por otra, las de tipo económico.

Durante las distintas fases de elaboración de este Sistema, se han tenido que retomar distintos conocimientos adquiridos hace más de 10 años durante la etapa de estudio y que por razones que de no haberse presentado, en estos momentos no se tendría por que haber preocupaciones por retomar este proceso, para obtener mi título y cédula profesional.

Hoy que este trabajo ha sido concluido, puede decirse que la frase tan utilizada "nunca es tarde para comenzar" es cierta y encaja al 100% en la situación particular y de igual manera seria un ejemplo o invitación para todos aquéllos quienes no han concluido este proceso para lo hagan. En este caso, por razones de salud y laborales no existieron las condiciones ni el apoyo para tomar la decisión y, ahora que aprovechando el momento de desempleo vivido, lejos de verlo como una situación difícil, se ha visto como un momento de oportunidad para dar un respiro y concluir varias cosas que en su momento no se pudieron concretar.

Este trabajo sin duda, ofrece una solución a la problemática que ha sido descrita en los primeros Capítulos y sin duda, debería de ser considerada para que sea implementada al 100%. Esto, fundamentalmente porque aún existen personas que, en su cabeza tienen ideas "cuadradas" y lejos de considerar a este trabajo como una herramienta que facilite las tareas de administración y optimización de los servicios en el Laboratorio Abierto, lo ven como un estorbo. Por lo menos, la intención no es esa y crean que ha constituido un gran esfuerzo por llegar a esta etapa del trabajo de tesis y por consiguiente, no sería recomendable que quedara como algo que deba archivarse.

Durante las distintas visitas hechas a las instalaciones del Laboratorio, se tuvo la oportunidad de observar de cerca las distintas situaciones que allí se presentan y de ello, se obtuvo los materiales necesarios para idear la estructura del **SAEALABFI**.

Al iniciar la elaboración de este trabajo se estimaba que terminaría entre 7 y 9 meses el trabajo en conjunto. Pero, hubo necesidad de resolver problemas técnicos que estuvieron lejos del alcance (vg. Grabación del **PIC16F84A**). Estos problemas generaron algunas lagunas que se interpusieron en el camino.

Otro aspecto de grande importancia fue el de la elaboración de la documentación del Sistema, que en este momento se encuentra conformado por un total de 276 páginas y que solo abarca la operación y características del Sistema y como resultado, no se consideran los detalles técnicos de la implementación en conjunto con la *interfaz electrónica*. De cualquier manera, no es nada despreciable este esfuerzo, pues requirió de una gran decisión y tiempo al frente del equipo de cómputo para concretar esta etapa.

Cuando se tuvo lista la documentación del Sistema, se procedió a la elaboración del programa de instalación y para esto fue utilizada la aplicación *Setup Factory 8*. Esta aplicación de uso comercial permite proveer a los programas personalizados de instalación, las herramientas necesarias para incluir todos aquéllos archivos de toda índole que las aplicaciones requieren para su correcto funcionamiento. Dentro de estas características se encuentra la determinación del Sistema Operativo en que será instalado el programa; las utilerías complementarias que para el caso del **SAEALABFI** fue el Microsoft Framework 2.0, Microsoft SQL 2005 Express y algunas fuentes tipográficas.

Con esto, el programa instalador está preparado para determinar que en el equipo en que se instale se cuente con los recursos mínimos de memoria y espacio en disco duro para que pueda operar el Sistema en él.

Durante este proceso de elaboración de este trabajo, se liberaron 3 versiones del mismo en las que se fueron incluyendo las mejoras y herramientas que en reuniones con los encargados del Laboratorio se depuraron. En la última de ellas (versión 1.2.0000.0) se incluye de manera integral a todas estas observaciones y se encuentra dentro del programa de instalación junto con la documentación de éste.

Otro aspecto al que se tuvo que enfrentar fue que en el planteamiento original del tema de tesis, se consideró al puerto paralelo como el mecanismo de comunicación entre el equipo de cómputo en el que se instaló el Sistema y la propia *interfaz* electrónica. Esta decisión tuvo que cambiarse en la medida en que la marcha del trabajo avanzaba pues, si bien es cierto que este puerto ofrece amplias ventajas en la implementación de la parte electrónica, también presenta limitaciones en cuanto a la distancia máxima que debe existir entre el equipo de cómputo y el equipo periférico (en nuestro caso la *interfaz electrónica*).

Se consideró utilizar un puerto USB, pero los costos de los kits para implementarlos dentro de una solución, no justifican su utilización y, aunado a esto, una vez más la distancia es la limitante, pues por implicar niveles de voltaje del orden de -5v y +5v, la norma que define a este tipo de dispositivos indica que la distancia (cable) no debe exceder de 1.8m.

Por lo anteriormente dicho, quedó como la solución viable, el puerto serie que utiliza niveles de voltaje del rango de -12V a 12V y por la norma RS232 que rige a este tipo de conexiones; permite tener distancias de hasta 15 metros entre el equipo de cómputo y el periférico. Esto justifica la distancia física que habrá entre la computadora y la *interfaz electrónica* que es del orden de los 7.5 metros.

Al seleccionar a este puerto, la tarea de la implementación de la *interfaz electrónica* se complicó un poco más, pues se requiere primeramente de la conversión de los niveles eléctricos del protocolo RS232 a TTL y, aunque esta tarea no es difícil pues ya existe en el mercado dispositivos electrónicos dedicados a este fin particular (vg. MAX232), la tarea complementaria es la de la conversión de la serie de bits enviados secuencialmente y, una vez recibidos, convertirlos a formato paralelo que es la manera en que la *interfaz electrónica* procesa; los datos de habilitación y des habilitación de módulos y la ejecución de señales de control para la operación de la *interfaz* en conjunto.

Para solucionar esta situación, fue necesario recurrir al uso de un microcrontrolador. En el caso particular del proyecto es el **PIC16F84A** cuya tarea es la de realizar la comunicación asíncrona dúplex entre la computadora y la parte electrónica. En esta comunicación, solo es necesario la implementación y uso de tres líneas del puerto serie: tierra, transmisión y recepción.

En la búsqueda de compatibilidad entre distintos equipos de distintas generaciones, y para garantizar que la comunicación pueda ser implementada en distintos equipos, se realizó un análisis de distintos equipos de la generación del Pentium 4 (y anteriores) y los microprocesadores de tipo de doble núcleo y core duo. Se pudo observar que la mayor parte de los equipos de las generaciones nuevas ya no incluyen en su extensibilidad a los puertos serie y paralelo. Esto no representa mayor problema, pues también existen tarjetas de expansión vía puertos PCI o PCI Express que implementan a uno o dos puertos serie con lo que se puede dar soporte a esta aplicación.

También se encontró que existen convertidores de señal de tipo USB a serie, pero con ellos se cae en la situación de la distancia, pues los niveles de voltaje máximos son de -5V a +5V y nuevamente la limitante es la distancia que es del orden de 1.8 metros máximo.

Para implementar el proyecto en conjunto, durante el mes de diciembre de 2008 se buscó entre las autoridades de la Facultad el apoyo para este fin. Se nos dijo que se contaba con una partida presupuestal para la implementación de proyectos por el área de computación y solicitaron un proyecto de presupuesto para considerarse en la partida presupuestal indicada; se nos notificó que se había autorizado el presupuesto.

El presupuesto incluye la adquisición de un equipo de cómputo nuevo con un lector de código de barras, tarjeta de expansión PCI a puertos serie, los componentes para la elaboración de una tarjeta principal y una de reserva, 4 tarjetas de bloque y 2 más de reserva, así como los insumos necesarios para su elaboración (tarjetas fenólicas, cloruro férrico, hojas de transferencia térmica y un grabador de micro controladores lite). Los precios se consultaron durante el mes de enero de 2009 y el total ascendía a un total de $16,626.75 y a la fecha no hay noticia alguna.

Por la premura de terminar el proyecto, se realizaron pruebas con 2 módulos completos y el resto simulándolo mediante leds a las salidas de los registros o latches contenedores de los estados de cada módulo y que por recomendaciones de uno de los sinodales se hicieron las pruebas de implementación para los dos modulos con el material con que se contaba.

Finalmente, es de considerar que deberían de realizar la adquisición de al menos un grabador de los micro controladores de Microchip, pues con los que se cuenta, aunque aparentemente los graba, no dieron los resultados esperados hasta que por medio del préstamo de un grabador que nos hizo un profesor, se pudo echar a andar el micro controlador y se obtuvieron los resultados esperados.

También es importante diseñar un nuevo reglamento del Laboratorio, para que las características del **SAEALABFI** sean empleadas a fondo y para que los usuarios tomen conciencia de la importancia que implica el correcto uso de los recursos del Laboratorio para poder brindar un mejor servicio en las instalaciones con los principios de equidad, apoyo incondicional y sujeto a las normas establecidas y como consecuencia, un mejor aprovechamiento de las condiciones de los equipos e instrumentos para darle un mayor tiempo de vida útil y aportar un granito de arena en el tema del uso de los recursos no renovables en cuanto al uso de la energía eléctrica se refiere.

AMPLIACIONES, CONTINUIDAD Y MEJORAS DEL SISTEMA

Primeramente, es necesario recordar que el proyecto en un principio se consideró para dar soporte a 30 módulos que se encuentran instalados en el Laboratorio Abierto y, por razones técnicas de la lógica combinacional binaria de la electrónica digital, la infraestructura básica aquí expresada puede sin mayor problema expandirse a 32 módulos.

Aunque las mismas características técnicas al utilizar el decodificador **SN74LS155N**, el Sistema se ha ajustado para que de soporte real hasta 64 módulos agrupados en 8 tarjetas de bloque con 8 módulos cada uno como ya se ha expresado a través de este trabajo de tesis.

Para agregar nuevas tarjetas de bloque con sus respectivos módulos, tan solo es necesario crear las tarjetas y montar los componentes siguiendo el órden que se mostró en el Capítulo V en los subtemas enfocados a la *interfaz electrónica* y realizar el ajuste en los parámetros del Sistema en la opción *Archivo->Configurar->Módulos instalados* para que se considere a todos los módulos que allí se expresen.

Otra de las características que se deben de considerar es el caso que se ha manifestado en relación a los proyectos de potencia. Motivo por el cual se ha sugerido que por lo menos en la última tarjeta de bloque que contendrá 6 módulos, sea ajustada de manera que esos seis módulos sean considerados para ser asignados a proyectos generales y específicos para los relacionados con "Potencia" pues la demanda de corriente es mayor y en ocasiones incuantificable de manera fácil. Para ello se ha sugerido que en lugar de utilizar TRIACs de 8 Amperes, se considere igualar su capacidad al de las pastillas del panel de switcheo (15 Amperes) o bien utilizar relevadores en cuyo caso se tendría que realizar la modificación del PCB para que en él se puedan montar dichos componentes. En cualquiera de los casos e incluyendo los módulos "normales" se recomienda incluir un fusible cuya capacidad de corriente sea a lo más medio Ampere debajo de la capacidad del dispositivo electrónico que se utilice para el propósito de conmutación de la energía eléctrica para evitar que se suscite algún daño a la *interfaz electrónica* y para garantizar que la operación de que el Sistema en su conjunto opere de manera normal.

Al principio de este proyecto, se consideró incluir un lector de código de barras para la lectura del número de cuenta de los usuarios y no tener que digitarlo, a estas fechas no fue posible lograr esta finalidad por motivos ecónomicos aunque de ser necesario, tan solo será necesario hacer ajustes en la clase ***frmUsuarios_SelecciónDeUsuario*** que corresponde a la caja de diálogo desde donde se accede al ingreso de números de cuenta de los usuarios y desde las clases ***frmSanciones***, ***frmUsuarios_Actualizar***, ***frmUsuarios_Agregar*** y ***frm_Usuarios_SelecciónDeUsuario***.

Una mejora que pudiese implementarse en la continuidad de este trabajo y como un tema de tesis complementario sería el de vincular al Sistema todo un Sistema de monitoreo inhalambrico de cada uno de los módulos en los que se determine de manera automática el estado que guarda cada uno de los instrumentos de cada módulo y se reporte al Sistema dicha situación para que los Administradores puedan verificar y determinar con certeza lo que está ocurriendo con la operación física y directa de los equipos del Laboratorio.

Complementariamente, se podría utilizar un Sistema electrónico en que se coloque un display en cada módulo y en él se anuncie a los integrantes de los módulos, quién o quiénes deben estar en dicho módulo; los integrantes que se encuentran asignados a ese módulo, la cantidad de tiempo que llevan utilizando el módulo y lo más importante, la cantidad de tiempo que les

resta antes de que les sea interrumpido de manera automática el flujo eléctrico y así ellos tengan que acudir con el administrador del Laboratorio a solicitar una prorroga en el tiempo de servicio según se ha descrito en el instructivo de operación del Sistema y en el Manual de referencia del mismo.

Por lo anterior, es aconsejable rediseñar el reglamento interno de uso de las instalaciones del Laboratorio, de sus equipos y servicios para que en conjunto se cubra al 100% los requerimientos de nuestros tiempos de equipos que cada vez son más escasos, se prolongue el tiempo de vida útil de los mismos y se cree una cultura de uso eficiente de los bienes públicos de la UNAM.

Finalmente, y de manera complementaria al Sistema, se podría crear un Sistema alterno en el que se ponga a disposición de los usuarios del Laboratorio, un catalogo en línea de consulta de todos los manuales de referencia de los dispositivos electrónicos más utilizados y de toda índole inherentes a la operación y trabajo dentro del Sistema con la finalidad de que se vincule a la *Base de Datos* del **SAEALABFI** y registrar su uso como si se tratase de un bien físico como los manuales de referencia. Este vínculo se podría realizar en las clases ***frmMateriales_Entrega*** y ***frmMateriales_Préstamo***. Con esto, ya no sería necesario hacer uso de demasiado papel y, con ello apoyar en los esfuerzos por la preservación ecológica que es tema de los tiempos que nos ha tocado vivir y que en adelante será de vital importancia para la especie humana conservar.

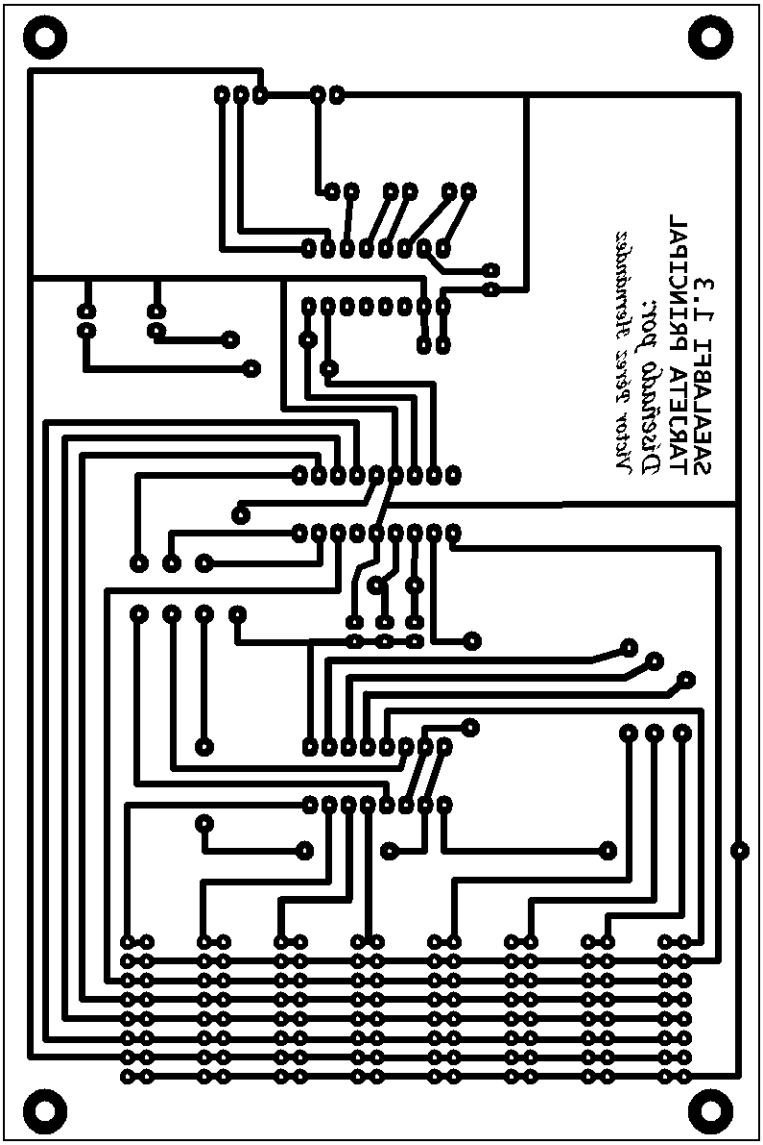
MANTENIMIENTO AL SISTEMA

Este es sin duda una de las inquietudes que pudiése preocupar más, pero por la naturaleza del Sistema, se considera que es mínimo el mantenimiento que se debe realizar al mismo. Por una parte, se encuentra el mantenimiento que debe realizarse al Software en cuyo caso se ha provisto de un conjunto de opciones agrupado en el menú **Administración** para que se realicen los ajustes necesarios a la *Base de Datos* y los parámetros generales y de propósito específico del Sistema; aunque conviene recordar que la mayoría de estas opciones sólo las puede realizar el súper-administrador o los Administradores del mismo.

En cuanto a la interfaz electrónica, el mantenimiento es mínimo pues los encapsulados y componentes principales y más delicados de la misma se encuentran montados en bases que retienen a dichos componentes y en consecuencia solo se requerirá realizar un monitoreo en caso de falla para reemplazar el dispositivo dañado sin tener que desoldar, desmontar o hacer algo más que lo ya indicado. En el caso de que no haya suministro en los módulos aunque en la fase digital asi se indique, conviene realizar una medición del flujo eléctrico para determinar la posible falla, aunque de implementarse un proyecto complementario en forma de un escáner, se podría ahorrar mucho trabajo.

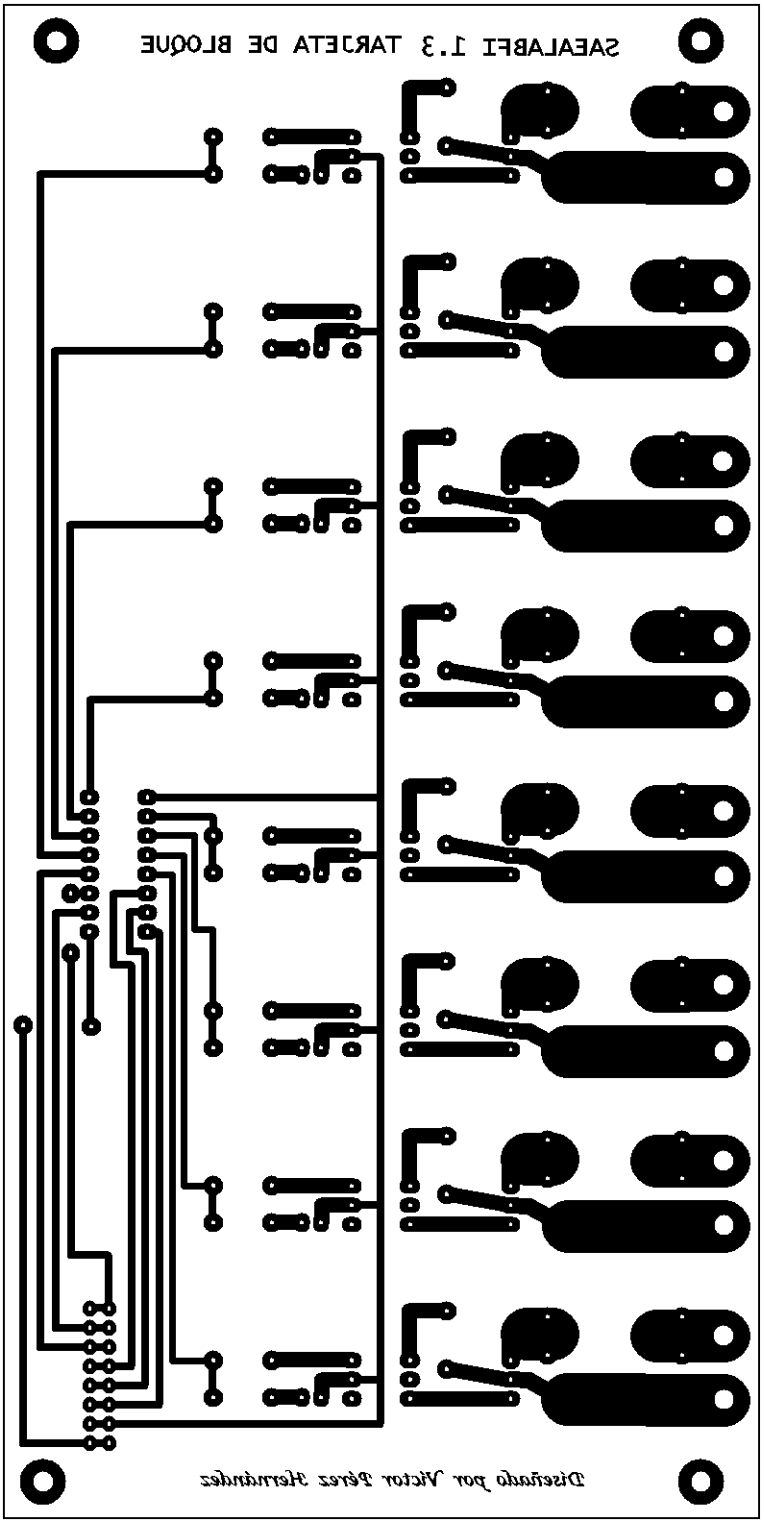
APÉNDICE A: TARJETAS PCB

TARJETA PRINCIPAL



PCB 1. Tarjeta Principal en tamaño real de 10x15 cm

PCB DE LA TARJETA DE BLOQUE



PCB 2. Tarjeta de Bloque en tamaño real de 20x10 cm

APÉNDICE B: PRUEBAS, MEDICIONES E IMPLEMENTACIÓN

Para la implementación del proyecto, se realizaron gestiones con las autoridades académicas correspondientes con la finalidad de solicitar un equipo de cómputo en el que se pudiese instalar la aplicación que es tema de esta tesis; en respuesta a esta petición, resolvieron otorgarnos un equipo básico para este propósito.

El equipo corresponde a un equipo DELL Optiplex serie Gx220, cuya configuración es la siguiente:

- Microprocesador Intel Pentium 4 @ 2 Ghz
- Disco duro con capacidad de 40 GB
- Memoria RAM de 128 Mb tipo DDR @ 266 Mhz
- Monitor de resolución máxima de 1024 x 478 pixeles
- Teclado
- Mouse
- Unidad de Disquete
- Unidad de CD-ROM

Para la implementación de las utilerías necesarias en la aplicación se requiere de un mínimo de 256 Mb de memoria RAM, por lo que de manera conjunta entre el asesor y el tesista, le fue incluido un módulo más de 256 Mb de capacidad para hacer un total de 384 Mb y de este total queda libre 380 Mb pues los otros 4 Mb los ocupa el Sistema Operativo para las funciones de video.

Fotografía 0-1. Preparación del equipo para instalar el SAEALABFI

En la fotografía anterior, se muestra el equipo al que se ha hecho referencia. Se encuentra

en proceso de preparación para comenzar a operar al Sistema en modo libre de conexión; es decir, aún sin la implementación del reconocimiento de señales y de la *interfaz electrónica*.

Una vez que se tuvo listo al equipo, éste fue trasladado al cuarto de control del Laboratorio Abierto de Electrónica, en donde se colocó al equipo para poder comenzar a realizar los registros de actividad de uso por parte de los usuarios que solicitan los servicios del Laboratorio.

Fotografía 0-2. Operación del Sistema con el equipo asignado al proyecto

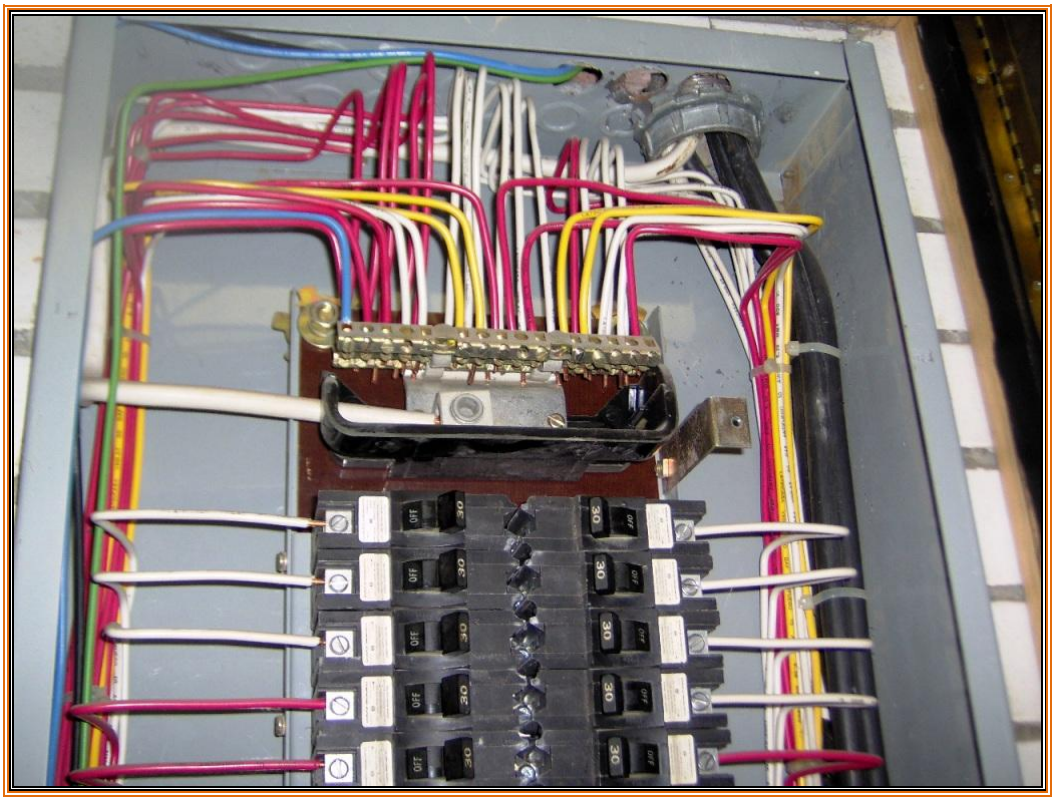
En la fotografía anterior se puede apreciar al equipo ya colocado en las instalaciones del Laboratorio Abierto operando de manera normal en el que se ha incluido en el casillero de identificaciones a aquéllas que corresponden a los usuarios que han solicitado el servicio del Laboratorio.

Cabe resaltar que para el inicio del uso del Sistema, se preparó una Base de Datos con la información de 750 alumnos que, sumado a los 53 con que ya se contaba en pruebas anteriores se contó de inicio con un total de 803 alumnos (usuarios) con derecho al uso de los servicios del Laboratorio.

Al transcurrir los días, se han ido agregando usuarios que no se encuentran registrados en el Sistema de manera que la información con que se cuenta ha ido aumentando y dando así, una mejor certeza de que la finalidad del **SAEALABFI** proporciona confiabilidad y en la medida de que se registren los movimientos de los usuarios, los productos del Sistema serán explotados a su máxima capacidad.

En complemento a la instalación del Sistema en el equipo y su puesta en marcha en el

Laboratorio, se realizaron pruebas de medición de corriente de manera directa en el tablero o panel de switcheo. En dicho panel se observó que eléctricamente se encuentra alambrado en forma trifásica de manera que en cada fase se encuentran distribuidos 10 módulos. En la siguiente fotografía se muestra parte del alambrado del panel de switcheo.

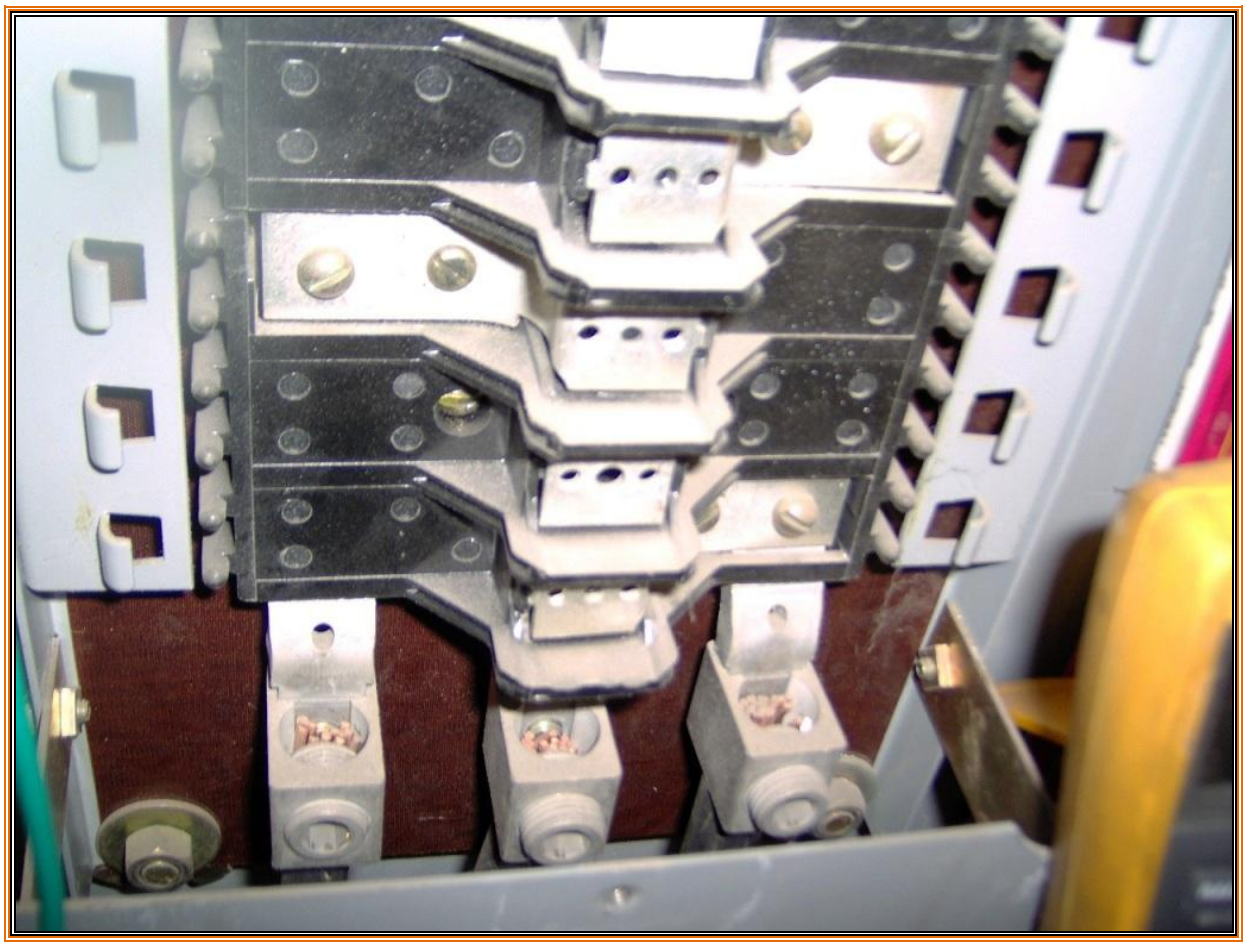


Fotografía 0-3. Aspecto del alambrado del panel de switcheo

En la fotografía 4 mostrada en la siguiente página, se muestra la entrada de las tres fases de corriente alterna al panel de switcheo, de donde se puede observar que de abajo hacia arriba las pastillas de los switches correspondientes a los módulos del Laboratorio, se van alternando de 3 en tres de abajo hacia arriba, unos módulos cierran su circuito con conexión a la izquierda (fase 1), otros a la derecha (fase 2) y otros al centro (fase 3).

En el alambrado de estas conexiones, se ha observado que utilizan alambre de calibre 10 de un solo hilo, de aquí se ha considerado que para conectar a la *interfaz electrónica* con el panel se requerirá de cable flexible con varios hilos para que la alteración de las conexiones de alimentación de energía eléctrica sean las mínimas y para que se pueda colocar de manera externa el alambrado hacia la *interfaz*. Ésta última, se estima contenerla en una caja metálica para cumplir con la finalidad aquí expresada.

En la fotografía 5 se muestra una de las pruebas de medición de corriente con una carga de prueba considerada dentro de los rangos de operación normal. El consumo de corriente para la mayoría de los proyectos que en el Laboratorio se realizan, se encuentra en un consumo dentro del rango de 0 a 2 Amperes. Sin embargo, el consumo de corriente máxima se encuentra relacionada a los rangos de consumo máximo de los instrumentos de medición con que cuenta



Fotografía 0-4. Conexión de las tres fases en el panel de switcheo

cada módulo. Estos rangos se muestran en la tabla V-1, donde se resume el consumo máximo por módulo. En suma, el consumo en condiciones máximas de corriente para una operación normal del instrumental dentro de un módulo es del orden de 5 amperes, no obstante, no hay que descartar que en algunas ocasiones los usuarios además de la corriente que se consume en estos instrumentos, pueden hacer uso de otros equipos o instrumentos propios y que pueden conectarlos a las líneas de alimentación de corriente lo que no nos permite saber con certeza la cantidad de corriente límite que se pudiese demandar en un momento dado.

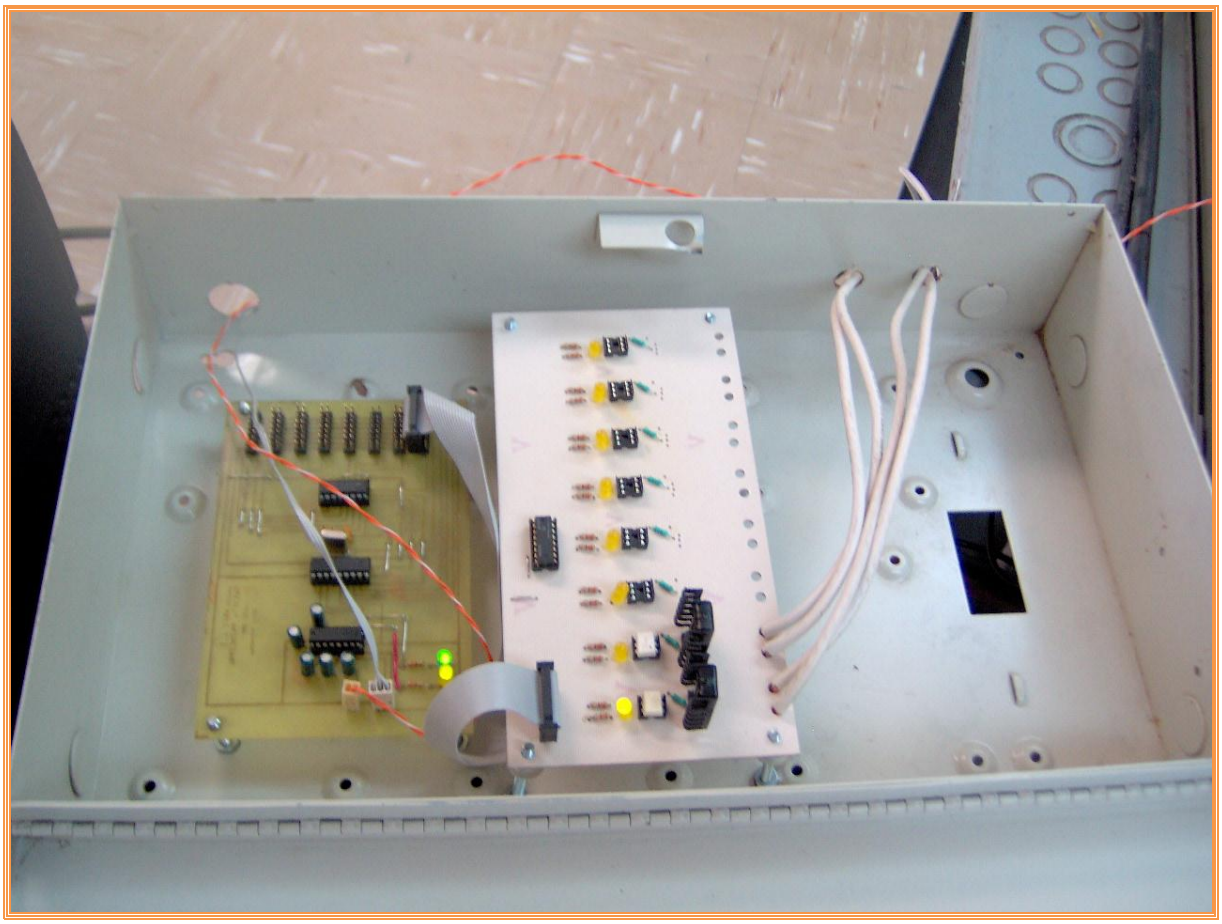
Por esta razón, se ha elegido a TRIACs de un consumo máximo de 8 amperes tal y como ya fue justificado en la sección V.1.

Fotografía 0-5. Medición de corriente con carga de prueba

Pruebas de implementación

Para propósito de verificar que el trabajo que se ha desarrollado en este tema de tesis, por la premura de tiempo y por la escasez de recursos económicos para implementarlo al 100%; se cuenta con los componentes que permiten la implementación de dos módulos. Las tarjetas que se utilizaron para este fin son: tarjeta principal y una tarjeta de bloque con los componentes necesarios para hacer las pruebas con el módulo 0 y 1.

Por razones del uso del Laboratorio Abierto de Electrónica de la Facultad de Ingeniería, se obtuvo el consentimiento de las autoridades competentes a fin de realizar dichas pruebas en uno de los laboratorios alternos en donde a estas fechas ya no están siendo utilizadas por haber concluido el semestre y por lo tanto a continuación se muestran las fotografías que dan testimonio de este proceso, mismo que será comentado después de cada una de las fotografías obtenidas para este propósito.



Fotografía 0-6. Montaje de las tarjetas principal y de bloque

En la fotografía previa, se muestra la colocación de las tarjetas indicadas montadas en una caja metálica. En ésta se puede observar que de la tarjeta de bloque se han ramificado dos pares de cables que servirán para ser conectados en el panel de switcheo y cumplir con la finalidad de ésta *interfaz*. Se puede observar que al lado izquierdo del eje central de la placa de bloque se encuentran colocados unos leds, mismos que servirán para indicar el estado que deberá estar reflejado en la parte de la electrónica analógica. Para este caso se observa que de abajo hacia arriba se encuentran colocados los optoacopladores y los TRIACs para el módulo 0 en posición más extrema hacia la parte de debajo de la fotografía y en seguida, el módulo 1.

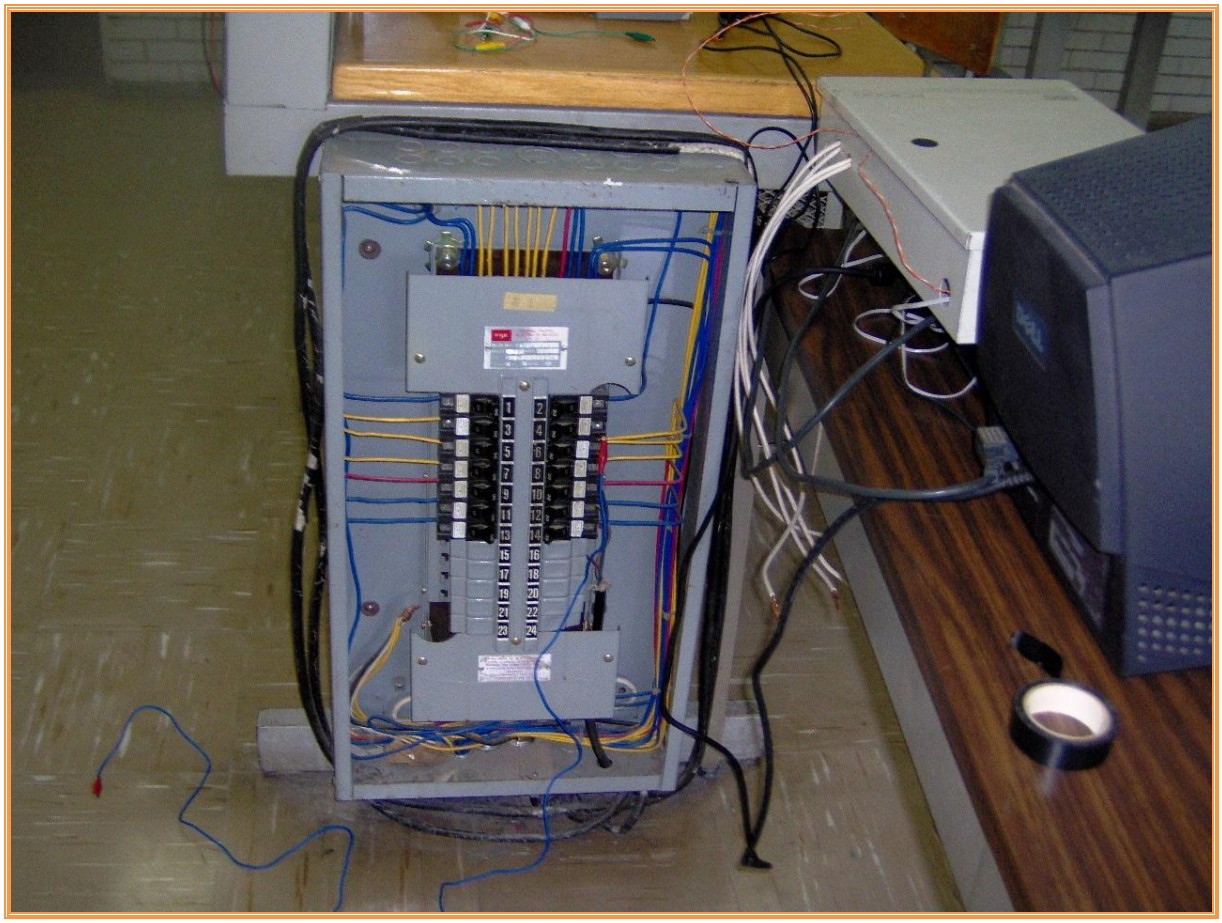
Es pertinente mencionar que para la expansión del conjunto de la *interfaz electrónica* se deberá continuar en secuencia hacia arriba en la colocación de los componentes para cubrir la primer tarjeta de bloque; montando por supuesto el optoacoplador, el TRIAC y el par de cables que corresponderá al módulo 3 y así sucesivamente hasta llenar la tarjeta de bloque (hasta el módulo 7). Una vez que se haga lo propio pero en una tarjeta de bloque más que corresponderá a los módulos 8 a 15; ésta tarjeta deberá ser conectada a la tarjeta principal mediante un cable de listón pero siguiendo el orden mostrado en la tarjeta principal de derecha a izquierda conforme se aprecia la fotografía.

Esto mismo se podrá realizar hasta alcanzar el máximo de 8 tarjetas de bloque y para que el Sistema pueda enviar las señales de control a cada uno de los módulos, entonces deberá realizarse el ajuste de la cantidad de módulos instalados en la *interfaz electrónica* desde la opción *Archivo->Configurar->Módulos instalados*. Para tener un panorama más claro, favor de revisar el anexo incluido en este material o bien, para más detalle revise el Capítulo III y Capítulo VI del manual de referencia del Sistema ubicado en el CD 2 o bien, una vez instalado

el Sistema desde el menú de programas de Windows dentro del grupo de programas del SAEALABFI.

En la fotografía 7, se observa el cableado entre la *interfaz electrónica* en el que para este fin se han utilizado al módulo 3 del tablero para simular al módulo 0 del Laboratorio Abierto y el módulo 4 al módulo 1 del Laboratorio Abierto (el módulo 3 corresponde al alambrado del lado izquiero y el módulo 4 al del lado derecho).

En la fotografía 8 se muestra una representación en la que se están utilizando a los módulo 0 y 1 del panel de control del Sistema para controlar el flujo eléctrico a los respectivos módulos instalados en la *interfaz electróncia*.

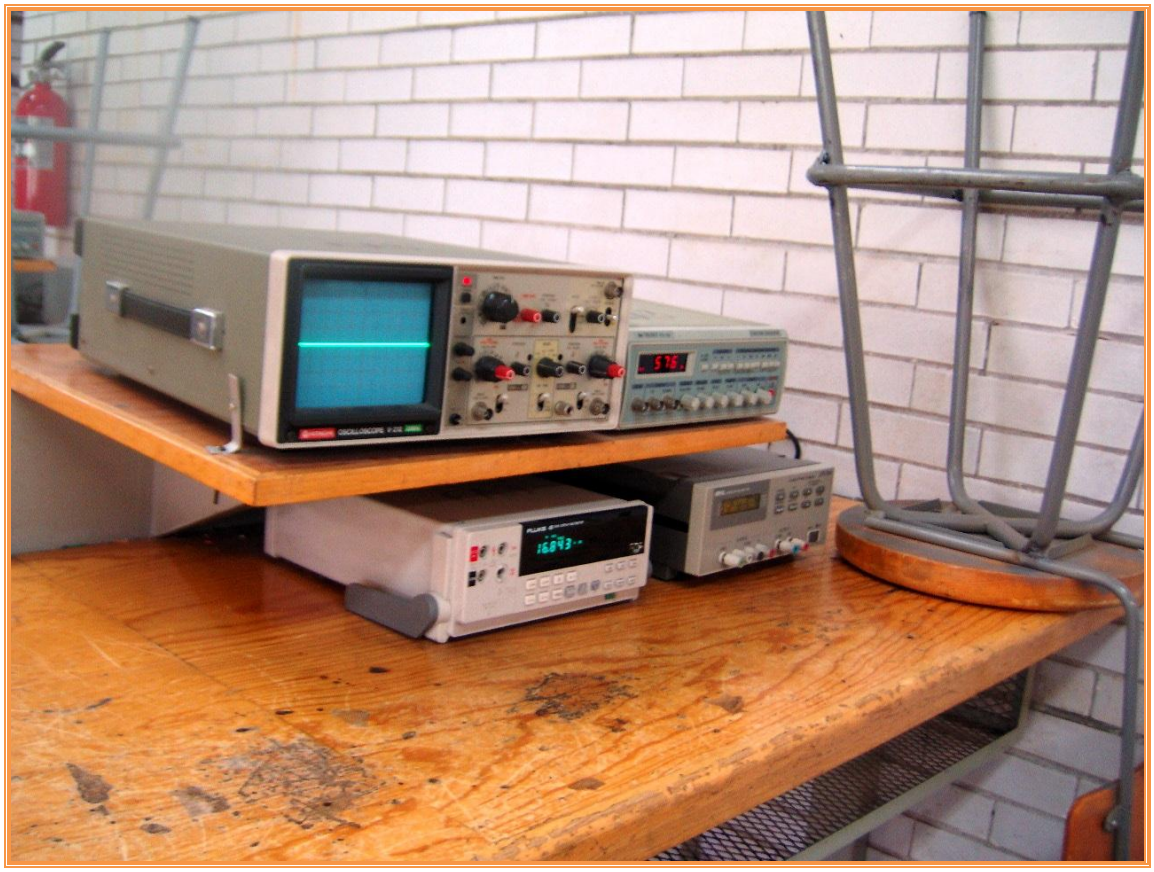


Fotografía 7. Conexión de la interfaz electrónica con el panel de switcheo

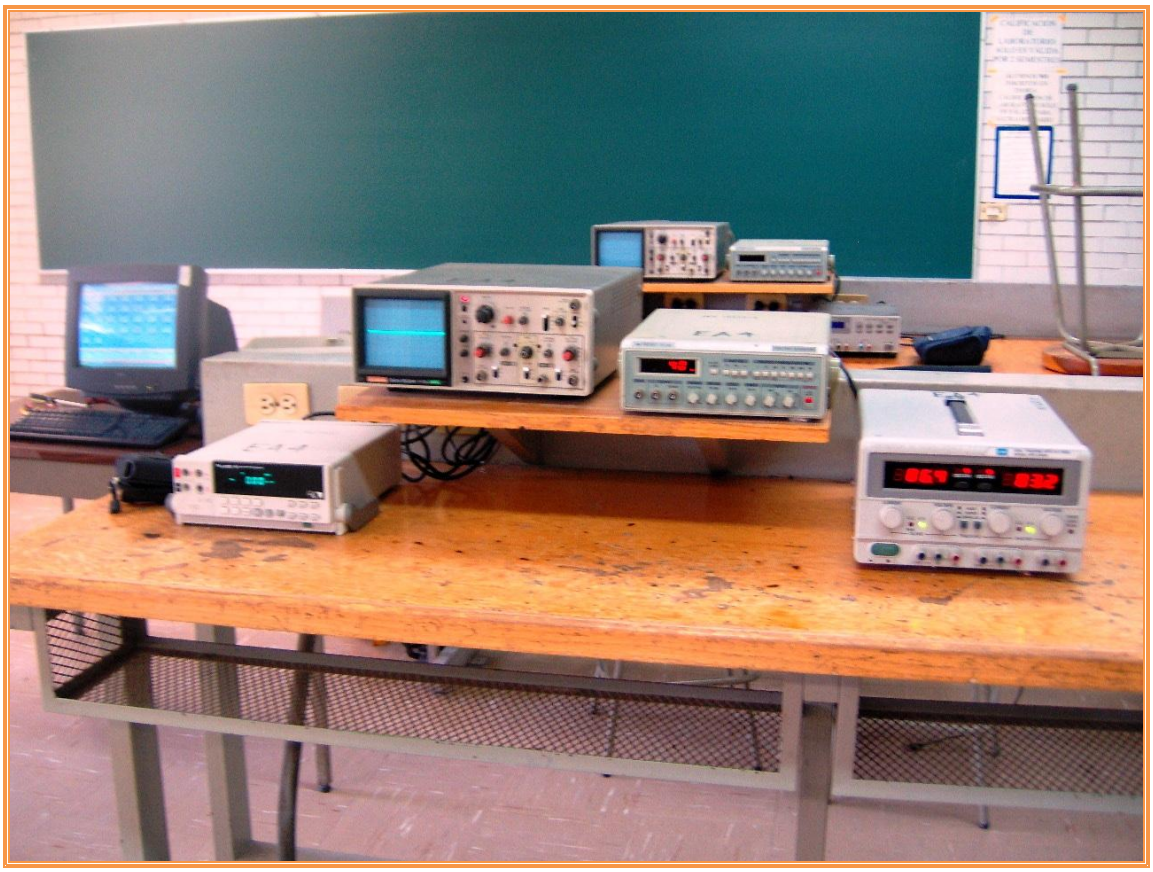
En el mismo se puede observar que en la parte superior izquierda se muestra un ícono de color amarillo que corresponde al módulo 0 y que indica que se ha establecido el atributo para que el módulo sea encendido de manera automática (forzada) cada vez que se inicie el Sistema y, en ícono que le sigue en color rojo indica que ha sido asignado a él un usuario que se encontrará trabajando en el módulo 1, motivo por el cual le será permitido el flujo eléctrico para que realice sus tareas, trabajos, proyectos o investigaciones que le han sido encomendadas.

Fotografía 8. Operación en los módulos 0 y 1 del Sistema

El reflejo de la situación mostrada en la fotografía 8 aprecia en las fotografías 9 y 10.



Fotografía 9. Reflejo del control del módulo 0



Fotografía 10. Reflejo del control del módulo 1

ANEXO: INSTRUCTIVO DE USO DEL SISTEMA

En este anexo se realiza una descripción rápida de las características intrínsecas del **SAEALABFI**, mostrando las opciones, secciones y operaciones que se pueden realizar con el Sistema, se abordará solamente la explicación de lo que se ofrece de manera directa desde el panel de control del Laboratorio por medio del **SAEALABFI**; si se desea conocer más acerca de las distintas opciones provistas por favor revise el manual del Sistema, mismo que lo podrá realizar desde el CD 2 o bien, una vez instalado el Sistema, desde el grupo de programas SAEALABFI desde el menú del Sistema Operativo se encuentra un vínculo al manual indicado.

Antes de hacer la descripción general del **SAEALABFI** es pertinente hacer mención que el Sistema se encuentra basado en el registro minucioso y detallado de cada operación que en éste se realiza, además de que los operadores del mismo (administradores del Laboratorio) se han clasificado en tres categorías: súper-administrador, administradores y operadores. Para cada uno de este tipo de operadores se han establecido derechos que indican las opciones u operaciones a que pueden acceder. Al final de este anexo se muestra la lista general de derechos en el que se indican cada una de las opciones del Sistema y quién o quienes tienen derecho a usarlas.

En la Ilustración 1, se muestra la composición y distribución de los distintos componentes con que ha sido construido el Sistema para ofrecer a los operadores del mismo a las herramientas que en conjunto apoyen en las tareas de administrativas de las instalaciones del Laboratorio, así como de sus componentes. Se puede apreciar que las zonas de trabajo se dividen en:

- Menú de Opciones
- Barra de identificación y búsqueda
- Barra de título
- Reloj
- Indicador de conexión de la *interfaz electrónica*
- Botones de estado
- Panel de control
- Paneles complementarios
- Barra indicadora del uso del panel
- Icono en la barra de tareas de Windows

A continuación se describe cada una de estas secciones. Para mayor información sobre estos componentes, por favor consulte el manual de operación del **SAEALABFI** mismo que puede ubicarse en el CD 2 o bien, al instalar el Sistema en el grupo de programas del SAEALABFI en el menú del Sistema Operativo aparecerá el acceso directo a dicho manual.

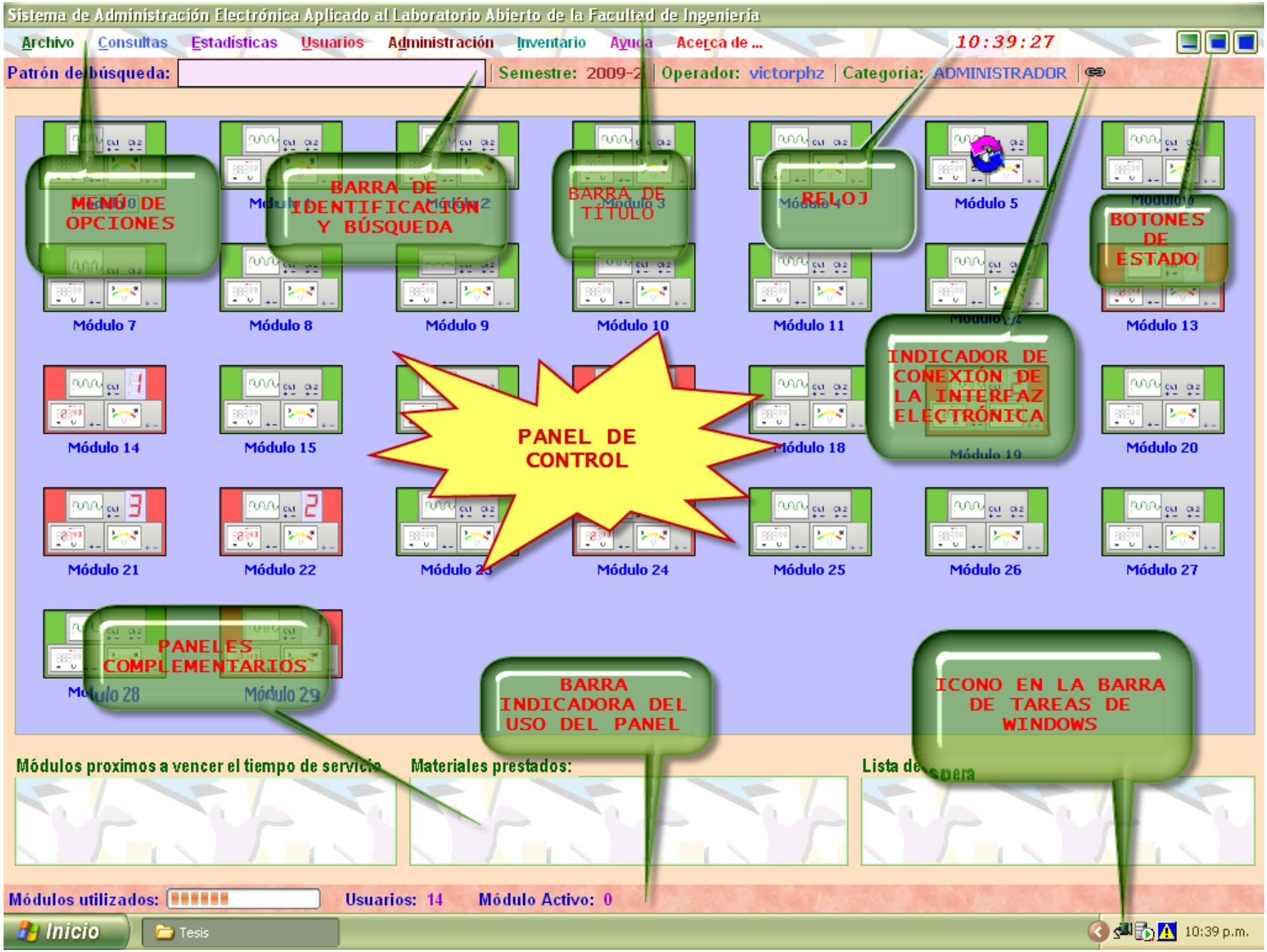


Ilustración 1. Aspecto de la ventana principal del SAEALABFI

Menús de Opciones

El menú de opciones es la estructura jerárquica ubicada en la parte superior de la ventana principal del Sistema desde la cual se puede acceder a los distintos comandos y opciones que el **SAEALABFI** proporciona a los operadores como apoyo en las tareas cotidianas de administración del Laboratorio. El conjunto de menús se encuentra agrupado en: *Archivo*, *Consultas*, *Estadísticas*, *Usuarios*, *Administración*, *Inventario*, *Ayuda* y *Acerca de…*

En las siguientes ilustraciones se muestra la estructura de cada uno de los menús que se han indicado en el párrafo anterior, observándose que cada una de las ramas de los menús se encuentra iluminado de un color distinto que se ha utilizado para identificar las opciones de cada uno de los menús.

Menú Archivo

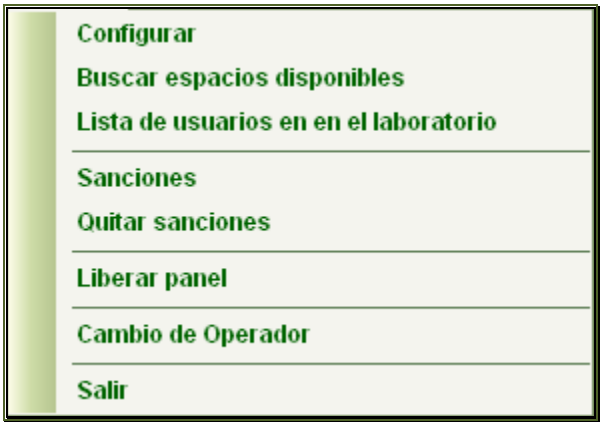


Ilustración 2. Menú Archivo

Como se aprecia, las opciones de este menú son: *Configurar, Buscar espacios disponibles, Lista de usuarios en el Laboratorio, Sanciones, Quitar Sanciones, Liberar panel, Cambio de operador y Salir*. En seguida se indica el propósito general de cada una de estas opciones:

- ***Configurar***, le permite al operador realizar los ajustes en los parámetros generales del Sistema, siempre y cuando el operador cuente con los derechos para acceder a esta opción. Al final de este anexo se incluye una lista de las opciones y los derechos que los distintos tipos de operador cuentan para este fin.

- ***Buscar espacios disponibles***. Es una opción que muestra al operador un listado con los lugares que quedan disponibles en cada uno de los módulos instalados, con la finalidad de checar la posibilidad de incluir a un usuario nuevo en alguno de los módulos.

- ***Lista de usuarios en el laboratorio***. Esta opción muestra al operador del Sistema, un listado en formato HTML en que se muestran cada uno de los módulos instalados en el Laboratorio, con sus respectivos usuarios que se encuentran ocupándolo.

- ***Sanciones***. Esta opción, permite a los operadores del Sistema aplicar sanciones a los usuarios que a criterio de éste y por la normatividad del uso de las instalaciones, lo

hayan infringido.

- ***Quitar sanciones***. Esta opción es la parte complementaria a la anterior y cuya finalidad es intuitiva; es decir, permite al operador, cancelar sanciones que se encuentren vigentes para los usuarios.

- ***Liberar panel***. Esta opción permite al operador del Sistema, dejar libre a todos los usuarios que se encuentren asignados a los módulos de trabajo y en este proceso si alguno de los usuarios tienen en resguardo alguno de los materiales propiedad de las instalaciones, serán enviados al panel de los materiales pendientes por entregar como se explicará más adelante con mayor detalle.

- ***Cambio de Operador***. Esta opción permite realizar un ajuste en relación al operador que se encuentre trabajando en el momento indicado con el Sistema. Debe notarse que todas las operaciones internas del Sistema se firman con el operador del Sistema con la finalidad de dar seguimiento al uso que se hace de las instalaciones por parte de los usuarios y de los operadores y es importante en el caso en que hay cambio de turno o encargado de la administración del Laboratorio.

- ***Salir***. Esta opción permite terminar con la operación del Sistema y ocurrirá al cierre de las instalaciones. Es importante notar que el Sistema sólo permitirá cerrarse cuando no haya ningún usuario registrado utilizando las instalaciones y, cuando no haya material alguno que esté pendiente por entregar.

Menú Consultas

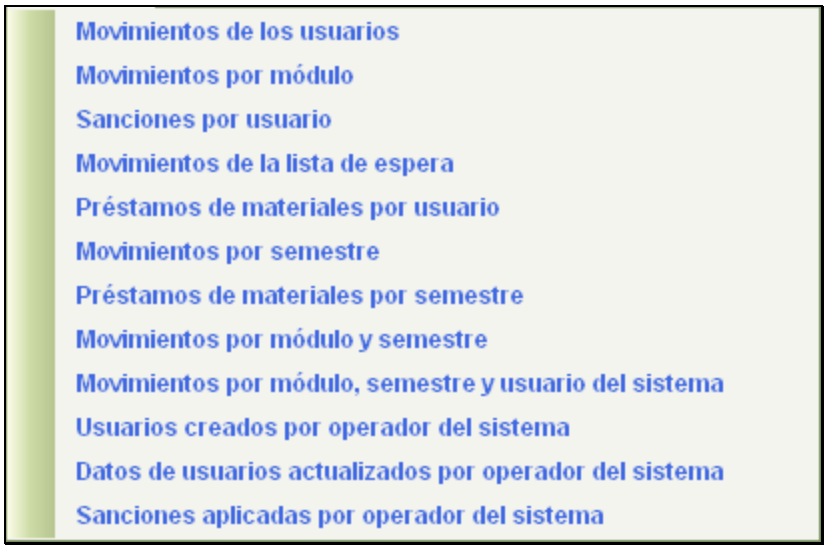


Ilustración 3. Opciones del menú Consultas

Las opciones agrupadas en este menú tienen la finalidad de ofrecer al operador del Sistema un conjunto de herramientas relacionadas con la consulta de los movimientos que se hayan realizado durante las operaciones normales del Sistema. Las opciones son *Movimientos de los usuarios*, *Movimientos por módulo*, *Sanciones por usuario*, *Movimientos de la lista de espera*, *préstamos de materiales por usuario*, *Movimientos por semestre*, *Préstamos de materiales por semestre*, *Movimientos por semestre*, *Préstamos de materiales por usuario*, Movimientos por

semestre, *Préstamos de materiales por semestre, Movimientos por módulo y semestre, Movimientos por módulo, semestre y usuarios del sistema, Usuarios creados por operador del sistema, Datos de usuarios actualizados por operador del sistema* y *Sanciones aplicadas por operador del sistema*.

En conjunto, todas estas opciones ofrecen una manera de conocer los movimientos que se registran día a día dentro de las instalaciones del Laboratorio; todas las opciones muestran los resultados en formato HTML y proporcionan distintas combinaciones de criterios de selección de la información deseada. Los parámetros más comunes son: semestre, año, número de cuenta, carrera, número de módulo y operador del Sistema. Para un mayor de detalle de cada una de estas opciones revísese el Capítulo VII del manual de referencia del **SAEALABFI** (ubicado en el CD 2 o desde el grupo de programas del **SAEALABFI** dentro del menú de inicio de Windows una vez instalado el Sistema desde el CD 1).

Menú Estadísticas

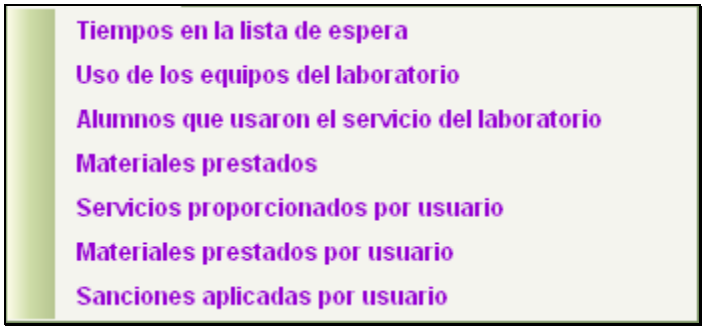


Ilustración 4. Opciones del menú Estadísticas

En este menú se agrupan opciones que están relacionadas con las estadísticas que involucran las distintas operaciones que se pueden realizan con el Sistema, se han incluido con la finalidad de dar seguimiento a los puntos críticos en las tareas cotidianas en relación con la administración del Laboratorio. Las opciones comprendidas en este grupo son: *Tiempo en la lista de espera, Uso de los equipos del laboratorio, Alumnos que usaron el servicio del laboratorio, Materiales prestados, Servicios proporcionados por usuarios, Materiales prestados por usuario* y *Sanciones aplicadas por usuario*.

Todas estas opciones muestran los resultados en formato HTML y se caracterizan por ofrecer al operador una manera en que la información consultada se pueda exportar a un formato delimitado por comillas y comas comúnmente conocido como formato ***CSV*** que puede ser importado en aplicaciones externas como es el caso de Excel.

Para delimitar los criterios de consulta, se han provisto de parámetros de consulta que en todos los casos son: alcance (todo, semestre o rango, en algunas opciones también se incluyen carrera y usuario), modo (día o mes) y año. Para mayor referencia, revísese el Capítulo VIII del manual de referencia del **SAEALABFI**.

Menú Usuarios

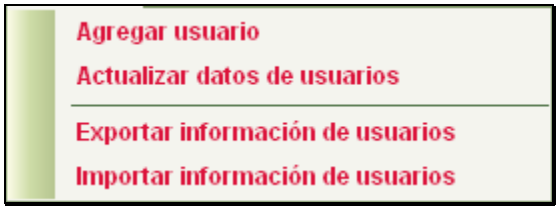


Ilustración 5. Opciones del menú Usuarios

Este menú está enfocado a las operaciones básicas que se pueden realizar sobre la información relacionada con los usuarios que tienen derecho a solicitar los servicios que se brindan en el Laboratorio. Las opciones que aquí se incluyen son: *Agregar usuario*, *Actualizar datos de usuarios*, *Exportar información de usuarios* e *Importar información de usuarios*.

En seguida se indica la finalidad de cada una de las opciones aquí mostradas:

- ***Agregar usuario***. La finalidad de esta opción es la de registrar a un usuario en la Base de Datos del **SAEALABFI** que antes no se encontraba registrado. La información que se incluye es: Número de Cuenta, Nombre(s), Apellido Paterno, Apellido Materno, Calle, Número, Colonia, Delegación o Municipio, Código Postal, Teléfono, Carrera y Fecha de expedición de la credencial.

- ***Actualizar datos de usuarios***. Esta opción es complementaria a la anterior y su finalidad es la de actualizar la información de alguno de los usuarios que ya se encuentran registrados en el Sistema. Es importante indicar que la vigencia de la información de los usuarios es de 6 meses. Después de este período y cuando se intente solicitar servicio se le solicitará al operador que actualice la información de los usuarios con la finalidad de mantener actualizada la Base de Datos.

- ***Exportar información de usuarios***. Esta opción permite al operador del Sistema generar un archivo en formato plano (texto) la lista de los usuarios que se encuentran registrados en el Sistema. Para consultar el formato del archivo y mayor detalle, revísese el Capítulo IX del manual de referencia del **SAEALABFI**.

- ***Importar información de usuarios***. Esta opción permite al operador realizar la operación contraria a la anterior, siguiendo el mismo formato que el de exportación; es útil cuando se desea incorporar varios usuarios de manera simultánea y que de forma externa se puede concentrar sin tener que rellenar el formulario de alta de usuarios para cada uno de los usuarios nuevos.

Menú Administración

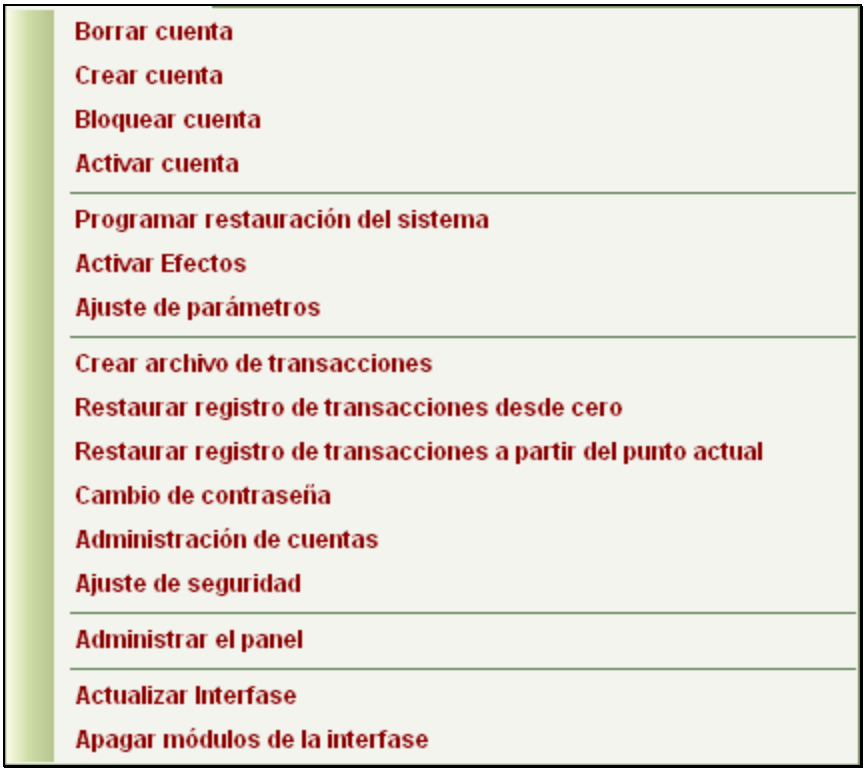


Ilustración 6. Opciones del menú Administración

El grupo de opciones que se encuentran incluidos en este menú se encuentran enfocadas hacia las tareas de administración de los recursos y operaciones propias del Sistema en conjunto. Las opciones que están contenidas en este menú son: *Borrar cuenta, Crear cuenta, Bloquear cuenta, Activar cuenta, Programar restauración del sistema, activar efectos, Ajuste de parámetros, Crear archivo de transacciones, Restaurar registro de transacciones desde cero, Restaurar registro de transacciones a partir del punto actual, Cambio de contraseña, Administración de cuentas, Ajuste de seguridad, Administrar el panel, Actualizar Interfaz y Apagar módulos de la Interfaz*.

Las opciones *Borrar cuenta* y *Crear cuenta* se utilizan para borrar o crear cuentas de operadores del Sistema, sujetándose a la jerarquía de los operadores quienes acceden a esta opción y según la lista de opciones y sus correspondientes derechos que se muestran al final de éste anexo.

Las opciones *Bloquear cuenta* y *Activar cuenta* son empleadas para dar mantenimiento a las cuentas de los operadores que han sido creadas en el Sistema. De igual manera al caso anterior, la operación de estas opciones se rige en los niveles jerárquicos y de derechos de los operadores del Sistema.

La opción *Programación restauración del sistema* se utiliza para dar mantenimiento mayor al Sistema en un momento en que se quiera restablecer al nivel de movimientos que se tenía al instalar el Sistema con la característica de que se conserva la lista de usuarios que tienen derecho a solicitar los servicios del Laboratorio.

La opción *Activar/Desactivar efectos* se utiliza para activar o desactivar los efectos visuales del **SAEALABFI**.

La opción *Ajuste de parámetros* se utiliza para actualizar los parámetros del Sistema con forme al Script de instalación del Sistema. Tiene la característica de que desde aquí se pueden incluir nuevas categorías de materiales para ser prestados, Carreras, Ubicaciones y otros parámetros de significado especial para el Sistema.

La opción *Crear archivo de transacciones* tiene la finalidad de crear un archivo en formato plano (texto) donde se detallan las transacciones o detalles de los movimientos en el Sistema y se pueden delimitar en tres tipos: todo, rango de fechas o rango de número de transacciones.

La opción *Restaurar registro de transacciones desde cero* se ha incluido para que el registro de transacciones comience de nueva cuenta desde cero y es de utilidad cuando se tiene la necesidad de comenzar a operar a partir de un estado similar al de instalación sin alterar la estructura de la información del Sistema.

De manera similar al caso anterior, la opción *Restaurar registro de transacciones a partir del punto actual* salvo que el contador de transacciones no se altera.

La opción *Cambio de contraseña* se utiliza para que el operador en turno pueda cambiar su contraseña de acceso por lo que esta tarea es de carácter personal.

La opción *Administración de cuentas* se utiliza para realizar de manera centralizada todas las operaciones relacionadas con las cuentas de los operadores y sólo está disponible para el súper-administrador ya que desde aquí podrá reasignar contraseñas a los operadores en momentos en que se les halla sido olvidadas. Revísese el Capítulo X del manual de referencia del **SAEALABFI** para más detalle.

La opción *Ajuste de seguridad* tiene la finalidad de realizar ajuste en los registros de seguridad de acceso de los operadores al Sistema; tiene utilidad en los casos en que por alguna razón “alguien” de manera mal intencionada o por error, altere esta información, claro está, siempre y cuando no se haya alterado por completo su contenido.

La opción *Administrar el panel* tiene la finalidad de ofrecer al súper-administrador una manera rápida y concentrada de las operaciones asociadas con la operación del panel de control del **SAEALABFI** (vid Capítulo X del manual de referencia del **SAEALABFI** para más detalle).

Las últimas dos opciones *Actualizar Interfaz* y *Apagar módulos de la Interfaz* están asociadas con la operación de la Interfaz electrónica. La primera opción reenviará el estatus que guarda cada uno de los módulos en el panel, tiene utilidad en el caso en que se haya presentado algún problema en la Interfaz electrónica y entonces actualizar el estado previo a la falla. La segunda opción se utiliza para apagar todos los módulos de la Interfaz.

Menú Inventario

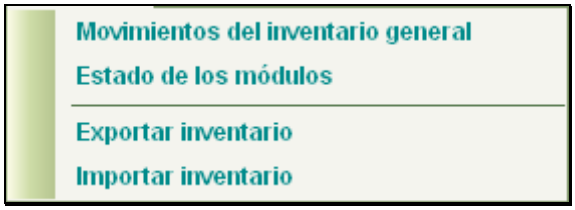


Ilustración 7. Opciones del menú Inventario

La finalidad de este menú es la de ofrecer a los operadores del Sistema una manera de mantener actualizado el inventario de los bienes pertenecientes al Laboratorio y de los instrumentos de cada uno de los módulos. Las opciones que se encuentran agrupadas en este menú son: *Movimientos del inventario general*, *Estado de los módulos*, *Exportar inventario* e *Importar inventario*.

A continuación se da una explicación breve de cada opción:

- ***Movimientos del inventario general***. En esta opción se ofrecen distintas opciones agrupadas en menús contextuales que permiten agregar, actualizar o eliminar la información de los bienes del Laboratorio.,

- ***Estado de los módulos***. En esta opción se puede hacer el ajuste de la situación que guardan los instrumentos de los módulos. Cada uno de ellos mantiene un estatus que por el momento se encuentra definido como "Funcionando" o "En reparación" aunque se está considerando incluir otros en el futuro. Cuando se modifica el estado de los instrumentos de los módulos y en caso de no ser "Funcionando" se mostrará el ícono del módulo correspondiente con un símbolo en la parte central con la finalidad de indicarle al operador que alguno de los instrumentos se encuentra en reparación.

- ***Exportar inventario***. Es una opción que permite exportar en formato plano (texto) con de manera delimitada de acuerdo al formato que se describe en el Capítulo XI del manual de referencia del **SAEALABFI**.

- ***Importar inventario***. Esta opción es la contraparte de la opción anterior, su finalidad es la de incorporar el inventario de los bienes del Laboratorio y ha sido incluida con la finalidad de realizar la concentración de la información desde otra aplicación como Excel y después incorporarla al Sistema y así evitar capturar de manera individual cada uno de los bienes.

Opción Ayuda

Esta opción muestra al operador el navegador de archivos en formato PDF (Adobe Acrobat) en el que se mostrará al operador el archivo de la ayuda del Sistema asociado a la forma en que se encuentra estructurado el **SAEALABFI** y las distintas opciones con que está dotado. En la siguiente Ilustración se muestra un ejemplo de la apariencia que guarda este archivo.

Ilustración 8. Ventana de la ayuda del SAEALABFI

Opción Acerca de…

Esta opción muestra al operador del sistema, los derechos y parámetros de identificación de la versión que está utilizando del Sistema, como puede observarse en la siguiente Ilustración.

Ilustración 9. Ventana de créditos del Sistema

Barra de identificación y búsqueda

En esta barra se proporciona una caja de texto en la cual se puede escribir una secuencia o combinación de palabras, letras o números delimitados por espacios que se utilizan como patrones de búsqueda dentro del Sistema para mostrar una lista en la cual se refleja los módulos que contienen alguno de los delimitadores dentro del número de cuenta del usuario o de su nombre dentro del Panel de control, después realiza la búsqueda dentro de la lista de materiales pendientes por entregar y, finalmente realiza la búsqueda dentro de la lista de espera si es que hay algún usuario en espera de ser atendido.

Suponiendo que llega un usuario a buscar a un compañero con el cual se encuentra desarrollando un proyecto, pero por disposición oficial la aplicación del reglamento indica que sólo podrán estar dentro de las instalaciones aquéllos quienes hayan solicitado el servicio del Laboratorio y que haya disponibilidad para tal efecto, pero no recuerda el nombre del compañero o el número de cuenta completo, sólo recuerda que el número de cuenta comienza con los números 3010. Una manera rápida de averiguar si esta persona se encuentra dentro del Laboratorio, es escribir en el cuadro de texto de búsqueda de usuarios la secuencia 3010 y después de esto oprimir la tecla *ENTER*, un posible resultado de esta acción es la que se muestra en las siguientes imágenes:

Ilustración 10. Ejemplo de patrón de búsqueda

Ilustración 11. Resultados de búsqueda

Como puede observarse en las Ilustraciones 10 y 11, se observa que el cuadro “Resultados de búsqueda” visualiza una lista con los módulos donde hubo coincidencia así como el número de cuenta y el nombre del usuario. Si hubiese usuarios con materiales pendientes por entregar y este a su vez cumpla con el patrón de búsqueda, un resultado posible puede ser el que se muestra en la Ilustración 12.

Ilustración 12. Resultados de búsqueda en caso de que hubiese coincidencias en la lista de materiales pendientes por entregar

En esta última Ilustración se observa que además de los módulos indicados, se refleja la leyenda "Materiales por entregar" y en seguida se muestra el número de cuenta y el nombre del usuario que se encuentra en esta situación.

Si se llegara a dar el caso que hubiese además un usuario en la línea de espera y se realiza la búsqueda con el mismo patrón indicado, un resultado posible podría ser el que se muestra en la Ilustración 13.

Ilustración 13. Resultados de búsqueda con resultados de coincidencia en el Panel de Control, lista de materiales por entregar y la lista de espera

En caso de no existir coincidencias con el patrón de búsqueda, el dialogo de resultados es similar al que se muestra en la Ilustración 14.

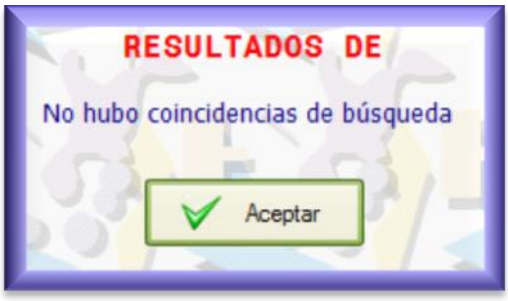


Ilustración 14. Resultados sin coincidencia de búsqueda

Después de la sección de búsqueda, se muestran 3 secciones separadas por una barra vertical en la que se muestra: el semestre sobre el cual se están registrando las operaciones del Sistema (se puede actualizar, cambiando los parámetros en la opción de configuración), el nombre del operador del Sistema y, finalmente la categoría a la que pertenece el operador en turno (como ya ha sido mencionado, las categorías de operadores son: ***SUPERADMINISTRADOR***, ***ADMINISTRADOR*** *y* ***USUARIO***). Esto se puede observar en la Ilustración 15.

Ilustración 15. Sección de Identificación

Barra de título

Esta barra es la que Windows muestra en cualquier ventana de aplicaciones, en el caso del **SAEALABFI** siempre mostrará la leyenda ***Sistema de Administración Electrónica Aplicado al Laboratorio Abierto de la Facultad de Ingeniería***.

Reloj

Se muestra la hora del Sistema cuyo propósito es el de apoyar al operador del Sistema a llevar a cabo aquéllas tareas que regularmente realiza de manera manual y que requiere la consulta de la hora en que se registra algún suceso (por ejemplo, el otorgamiento de permiso a algún usuario para acudir al sanitario e indicarle a éste que cuenta con determinado tiempo para tal fin).

Indicador de conexión de la *interfaz electrónica*

Este es un pequeño ícono en forma de cadena que se utiliza para visualizar el estado que guarda la conexión de la *Interfaz Electrónica* con el **SAEALABFI**. Si se encuentra dicha Interfaz, el ícono tiene la apariencia de una cadena cerrada y si es el caso de que no se ha encontrado a la Interfaz electrónica o si por alguna razón se ha perdido la conexión con ella, el ícono tiene el aspecto de una cadena rota como se muestran en las Ilustraciones 16 y 17 respectivamente.

Ilustración 16. Indicador de conexión de la Interfaz Electrónica

Ilustración 17. Indicador de conexión rota o no establecida con la Interfaz Electrónica

Botones de estado

En esta sección se muestran botones de reemplazo a los estándares de Windows para realizar las funciones de Cerrar ventana, Reducción a la barra de tareas de Windows, Restauración al estado previo y Maximizar la ventana. Estas opciones se ejemplifican en la Ilustración 18.

Ilustración 18. Botones de estado

El significado de los botones de izquierda a derecha es el siguiente: Cerrar, Minimizar, Normal y Maximizar, siendo el primero configurable al momento de la instalación del Sistema, para revisar cómo se puede cambiar su comportamiento por favor diríjase al Capítulo II del manual del **SAEALABFI**.

Panel de Control

Sin lugar a duda, esta es la sección más importante del **SAEALABFI** debido a que es aquí donde se registran y visualizan la mayor parte de las operaciones que han sido desarrolladas en el Sistema.

Para empezar se comenzará describiendo el aspecto que muestra el Panel de Control. El panel de control contiene un ícono por cada uno de los módulos instalados en el Laboratorio, incluyendo al módulo 0 que está reservado para contener en éste a las herramientas e instrumentos de uso general, es decir, el módulo de grabación de memorias, al medidor de inductancia y al taladro de banco.

Todos los demás módulos contienen cuatro instrumentos electrónicos a saber: una fuente de poder, un generador de funciones, un multímetro digital y un osciloscopio.

El número de módulos se establece en el proceso de configuración de primera vez y posteriormente se puede modificar en la opción del menú *Archivo -> Configurar*, como ya se ha descrito anteriormente en la sección dedicada al menú de opciones Archivo.

Cabe hacer mención que el número de módulos máximo que se pueden instalar y Configurar es de 63 (más el módulo 0), aunque por la naturaleza de la configuración de equipos instalada en el Laboratorio, se encuentra limitado a 31 módulos, pues también se considera al módulo 0 haciendo un total de 32 módulos (para el caso de este proyecto, esta cantidad de módulos es la que la *Interfaz electrónica* soporta y que ha sido adaptada para soportar máximo a los 64 módulos ya señalados.

Iconos de los módulos del Panel de Control

Los íconos utilizados para representar a los módulos dentro del Panel de Control, se han agrupado en tres colores distintos cuyo color y significado se resume en la siguiente tabla.

Apariencia	Significado	Descripción
	Modulo libre	Como su nombre indica, son módulos que están disponibles para ser asignados a usuarios que solicitan el servicio.
	Modulo ocupado 1	Representa a un módulo que se encuentra asignado a un usuario. El número que le acompaña en la parte superior derecha, indica la cantidad de usuarios que se encuentran utilizando al módulo.
	Módulo ocupado 2	De manera similar al caso anterior con la diferencia de que se encuentran dos usuarios utilizando el módulo.
	Modulo ocupado 3	De manera similar al caso anterior con la diferencia de que se encuentran tres usuarios utilizando el módulo.
	Modulo ocupado 4	De manera similar al caso anterior con la diferencia de que se encuentran cuatro usuarios utilizando el módulo.
	Modulo forzado	Los módulos forzados, son aquéllos que se pueden programar para que sean encendidos de manera automática al momento de arrancar el Sistema. Debe considerarse que no necesariamente tiene asignado algún usuario dentro de él. Un caso que ejemplifica este tipo de módulos es el módulo 0, que por su naturaleza debe estar encendido siempre, mientras que el servicio que se ofrece para el uso de éste, sigue el procedimiento normal de asignación de módulos.
	Modulo forzado 1	Este ícono representa a un módulo forzado; muestra en la esquina superior derecha el número uno que representa la cantidad de usuarios que se encuentran utilizando el módulo en ese momento.
	Modulo forzado 2	Este ícono representa a dos módulos forzado; muestra en la esquina superior derecha el número uno que representa la cantidad de usuarios que se encuentran utilizando el módulo en ese momento.

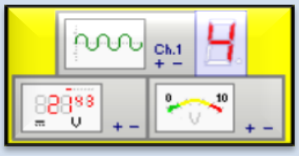
Apariencia	Significado	Descripción
	Modulo forzado 3	Este ícono representa a tres módulos forzado; muestra en la esquina superior derecha el número uno que representa la cantidad de usuarios que se encuentran utilizando el módulo en ese momento.
	Modulo forzado 4	Este ícono representa a tres módulos forzado; muestra en la esquina superior derecha el número uno que representa la cantidad de usuarios que se encuentran utilizando el módulo en ese momento.
	Módulo disponible con algún instrumento en reparación	Cuando alguno de los componentes o instrumentos asociados a un módulo se encuentra en reparación mediante la opción *Inventario -> Estado de los módulos*. Se colocará el símbolo en forma de reciclaje que se encuentra al centro del ícono.
	Módulo ocupado con algún instrumento en reparación	De manera similar al caso anterior, los módulos ocupados que tienen reporte de algún instrumento en reparación mediante la opción indicada, aparecerá el símbolo de reciclaje en la parte central del ícono; este símbolo se mostrará de manera respectiva para el caso de 1,2,3 ó 4 usuarios que ocupan al módulo.
	Módulo forzado con algún instrumento en reparación	De manera similar al caso anterior, los módulos ocupados o vacios que se han programado para ser encendidos de manera forzada y que tienen reporte de algún instrumento en reparación mediante la opción indicada, aparecerá el símbolo de reciclaje en la parte central del ícono; este símbolo se mostrará de manera respectiva para el caso de 1,2,3 ó 4 usuarios que ocupan al módulo o simplemente el símbolo de reciclaje cuando no se encuentre asignado el módulo.

Tabla 1. Significado de los íconos de los módulos

Se considera pertinente mencionar que la cantidad de usuarios máxima que se pueden asociar a un módulo es de 4 y se ajusta desde la opción *Archivo -> Configurar*. Esta cantidad variará de acuerdo al criterio del operador del Sistema y es aconsejable reconsiderar dicha cantidad en momentos en que la demanda de uso de las instalaciones aumenta según los momentos críticos en el transcurso de los semestres.

Cada módulo posee propiedades intrínsecas asociadas a los usuarios que a él están asignados, dichas propiedades son el número de cuenta del usuario, el nombre del usuario y otros de operación interna que permiten al Sistema controlar los recursos y tiempos que se asignan a los módulos.

Los módulos tienen asociado un cronómetro cuya duración está indicada en los parámetros de

instalación o de configuración del Sistema pudiendo ser 1, 2, 3 ó 4 horas según se indique; esta flexibilidad obedece a que la demanda del uso de los equipos no está distribuida uniformemente a medida que transcurre el semestre en curso, por lo que en algunos momentos podría ser factible establecer en 3 o hasta en 4 horas el tiempo límite de uso del Laboratorio aunque regularmente y por reglamento el tiempo máximo de uso de los equipos es de 2 horas.

El tiempo asignado a los módulos comienza a contar desde el momento en que el primer usuario del equipo solicita el servicio y le es asignado un módulo y por consiguiente, si llegan a utilizar el mismo módulo otros integrantes del equipo, el tiempo que ya fue utilizado por él o los usuarios anteriormente no será repuesto para los subsecuentes usuarios. También aplica en el caso en que se encuentren usuarios en la lista de espera y se libera algún módulo, el operador tiene la posibilidad de seleccionar quien ocupará el módulo y en ese momento arranca su cronómetro.

En cuanto al cronómetro, es necesario mencionar que el Sistema está preparado para indicarle al operador del Sistema desde la cantidad de minutos establecida en los parámetros de instalación por primera vez o en la opción de configuración. En cuanto a este parámetro se puede decir que se permite al operador seleccionar entre 5 y 40 minutos. Esta cantidad de minutos servirá para indicarle al operador con la anticipación de los minutos señalados de que alguno o algunos de los módulos se encuentran próximos a vencer el tiempo de uso de los módulos a través de un panel de ***"Módulos próximos a vencer el tiempo de servicio"***, desde el cual a criterio del operador, se le puede hacer una prorroga de 15, 30, 45 ó 60 minutos al módulo que se seleccione de la lista indicada.

Si dentro del número de minutos seleccionado aún no se ha tomado ninguna decisión respecto de la prorroga de servicio, el **SAEALABFI** de manera automática desconectará al módulo de la alimentación de energía eléctrica y se liberaran los integrantes de dicho módulo procediendo a dejar disponible el módulo, y si alguno o algunos de los integrantes del equipo en el módulo tuviese materiales prestados, dichos usuarios se colocarán en el panel de ***"Materiales prestado"*** y serán quitados de la lista hasta que el usuario llegue con el operador a entregar dichos materiales.

Durante el proceso de liberación automática o manual de algún módulo, el Sistema verifica si hay usuarios en la ***"Lista de espera"***, de ser así, en el momento en que se libera al módulo de los usuarios se le proporcionará al operador la libertad de elegir quién ocupará el módulo liberado si es que hay dos o más usuarios en la lista de espera, pero si sólo hay uno, de manera automática le será asignado el módulo que se ha desocupado y se notificará al operador del Sistema que el usuario ha sido asignado al módulo y en ese preciso momento comienza a correr el tiempo del servicio según se haya configurado. Como ejemplo de esto, obsérvese la Ilustración 19.

Ilustración 19. Mensaje de asignación automática

En la Ilustración anterior, se muestra un mensaje de aviso de que el usuario cuyo número de cuenta es 301005674 y de nombre “NADYELLI BECERRIL GALVAN”, ha sido asignado al módulo 29 que previamente ha sido liberado a través de un proceso automático o manual.

Cada módulo contiene una etiqueta que es su descripción simbólica de la forma “Módulo *x*”, donde x corresponde al módulo del Laboratorio seleccionado, además de esto internamente cada módulo posee un texto descriptivo que aprovecha la capacidad que el Sistema Operativo Windows proporciona para mostrar tooltips o cuadros de tips asociados al objeto (control) seleccionado y que para esta versión del Sistema tiene forma de un pequeño globo de mensajes.

El tooltip se mostrará por encima del módulo seleccionado al momento de posicionar el apuntador (cursor) del mouse sobre el ícono que representa al módulo. En esa etiqueta se despliega primeramente el número de módulo y el tiempo que resta de servicio, enseguida muestra una lista de los equipos instalados en el módulo así como el estado en que se encuentra cada uno de estos y finalmente, muestra la lista de usuarios que se encuentran asignados a dicho módulo.

Un ejemplo de esto se muestra en la Ilustración 20. En esta Ilustración se aprecia como lo he descrito la información correspondiente al módulo 10, que le resta una hora con 36 minutos de servicio, los equipos funcionando y se encuentran 2 usuarios asignados a este módulo.

En la Ilustración 21, se muestra el tooltip correspondiente al módulo 0, destacándose que a diferencia del cualquier otro módulo, la lista de equipos es lo único que se diferencia de los demás módulos como ya ha sido descrito con anterioridad.

Si el módulo se encuentra libre, se mostrará un tooltip que sólo contendrá el número de módulo y la lista de equipos asociados al módulo seleccionado como se puede apreciar en la Ilustración 22.

Ilustración 20. Tooltip de un módulo en uso

Ilustración 21. Tooltip del módulo 0 en uso

Ilustración 22. Tooltip de un módulo libre

Menús contextuales del Panel de Control

Los menús contextuales son una de las características con que Windows dota a los desarrolladores y usuarios del Sistema Operativo para poder hacer más robustas las aplicaciones que se desarrollan sobre él.

La finalidad de los menús contextuales o emergentes es el de acceder de manera rápida, concisa e inmediata a las opciones o comandos asociados a un control u objeto dentro de los Sistemas de Software. Para acceder a ellos es necesario oprimir el botón derecho sobre el control/objeto deseado y en seguida, si es que hay alguna opción programada para dicho fin aparecerá en forma convencional posicionada justo a la derecha de la ubicación en donde se registró la pulsación del botón derecho.

Para el caso del Panel de Control, existen 4 variantes del menú contextual que están asociadas a los módulos ocupados, módulos vacíos, módulos forzados y área en donde no se encuentran visualizados los módulos. Para éste último caso, además de lo ya explicado, se considera la categoría del operador del Sistema con la finalidad de permitir o limitar el acceso hacia opciones propias de administración como es la opción de Administración del panel.

Antes de empezar a describir los menús, se precisa indicar que dentro de la operación del panel hay otras funciones asociadas a la agrupación de los módulos y la asignación de ellos a los usuarios. Cuando un usuario que ocupa un módulo sólo o en equipo, ha decidido terminar de utilizar el servicio del Laboratorio, tan solo será necesario que acuda con el operador del Sistema para que haga el canje de su identificación contra su liberación dentro del Laboratorio y registrándose en ese momento el término de uso del módulo y si éste hubiese sido el único usuario entonces el módulo queda a disposición de quien así lo solicite, y en caso de haber usuarios en lista de espera, se procede a la asignación automática o selectiva. A esta característica le he nombrado *"Liberación de* usuario".

Un módulo puede también ser liberado antes de tiempo de manera manual y a criterio del operador del Sistema. Esta función se ha nombrado *"Liberación de módulo",* y es considerada como una opción útil cuando alguno de los miembros del equipo de trabajo asignado al módulo seleccionado ha infringido las reglas de uso del Laboratorio o cuando el equipo completo ha terminado de trabajar sin tener que liberar uno a uno de los usuarios del módulo que se les había asignado.

Además de las dos modalidades de liberación indicadas, he agregado una más llamada *"Liberación del Panel"*, y como su nombre lo indica se utiliza para liberar a todos los módulos del Laboratorio, se considera útil cuando se vaya a cerrar el Laboratorio o cuando el Laboratorio esté asignado a un grupo de clase que se encuentre con su profesor para revisión de proyectos o clases especiales y hayan concluido y por consiguiente, todos los usuarios han dejado de utilizar el servicio del Laboratorio.

Menú contextual de un módulo libre

Este menú está formado por dos opciones, estas son la de *"Asignar módulo"* y *"Encendido Forzado del módulo"*. Como ya he mencionado, los módulos pueden reflejar 3 estados posibles: módulo ocupado, módulo desocupado y módulo encendido de manera forzada. El aspecto visual del menú para el caso de este apartado es el que se muestra en la Ilustración 23.

Ilustración 23. Menú contextual de un módulo libre

Al seleccionar la primera opción, el Sistema mostrará una ventana en la que el operador tiene que introducir el número de cuenta del usuario que se está asignando al módulo seleccionado como se muestra en la Ilustración 24.

Ilustración 24. Introducción de cuenta para el usuario al que se le brindará el servicio

Una vez que se ha terminado de teclear el número de cuenta del usuario, el Sistema verificará que esté registrado dicho usuario en la Base de Datos del **SAEALABFI** y mientras se va tecleando dicho número, la ventana mostrará u ocultará el botón de *Aceptar* primeramente si es que el número de cuenta es de longitud menor a 9 números, y después, al llegar a los 9 caracteres, determina si es que existe dicho número de cuenta.

Si se encuentra registrado el usuario, la ventana de introducción de número de cuenta tendrá la apariencia mostrada en la Ilustración 24, mientras que la apariencia que se muestra cuando no se ha introducido un número de cuenta válido, es el que se muestra en la Ilustración 25.

Ilustración 25. Indicación de que el número de cuenta no está registrado

Cuando el número de cuenta es válido, aparece el nombre de usuario asociado a ese número de cuenta y, en este momento se puede elegir continuar con el proceso de asignación al oprimir el botón *Aceptar* o rechazar la asignación del módulo al usuario al oprimir el botón *Cancelar*.

Una vez que se ha decidido proceder a la asignación del módulo al usuario, el Sistema verificará que la información del usuario no tenga más de 6 meses sin ser actualizada, si es el caso, mostrará un mensaje indicando que es tiempo de actualizarse y después se enviará al cuadro de diálogo de actualización de información.

Ilustración 26. Indicación de actualización de información

En la Ilustración 26, muestra el mensaje de la necesidad de actualizar la información del usuario del cual se ha introducido el número de cuenta, mientras que en la Ilustración 27 se aprecia la estructura que presenta la ventana *Actualización de datos de usuarios* mediante la cual se proporciona al operador el medio de actualización de la información correspondiente al usuario seleccionado. La estructura de esta información se detalla en el Capítulo IX del manual de referencia del SAEALABFI.

Si se realiza algún cambio a la información o si se determina que los datos no han cambiado, tan solo es necesario oprimir en el botón *Actualizar*, si por el contrario sólo se oprime el botón *Regresar*, el Sistema interpretará a esta acción como la anulación del proceso y de regreso a la ventana de captura de número de cuenta, desaparecerá el botón de *Aceptar* y por consiguiente, no habrá más opción que cancelar el servicio al usuario que está solicitando el uso de las instalaciones del Laboratorio.

Una vez que se ha superado el proceso anterior, el **SAEALABFI** procede a verificar si dicho usuario tiene sanciones vigentes, de ser así, por reglamento del laboratorio no será posible brindarle el servicio al usuario y por consiguiente, se procederá a indicar que no es posible brindarle el servicio a dicho usuario en ese momento, además indicará a partir de cuándo será posible seguirle ofreciendo los servicios del Laboratorio y/o de los materiales y herramientas con que se cuenta.

Actualización de datos de usuarios

Número de cuenta: 089256121 VICTOR PEREZ HERNANDEZ

Nombre del alumno:

Nombre: VICTOR
Apellido Paterno: PEREZ
Apellido Materno: HERNANDEZ

Domicilio:

Calle: 1A. CALLE DE CONTRERAS
Número: 36
Colonia: BARRIO LAS CALLES
Delegación/Municipio: MAGDALENA CONTRERAS
CP: 10840*
Teléfono: 56-31-70-33 **

Otros datos:

Carrera: COMPUTACIÓN, ELECTRÓNICA, INDUSTRIAL, MECATRÓNICA
Fecha de expedición: 10/10/2009

Actualizar | Regresar

Ilustración 27. Aspecto de la ventana de actualización de información de los usuarios

Ilustración 28. Respuesta a la cancelación de actualización de información de los usuarios

Ilustración 29. Notificación de sanciones vigentes

Cuanto se ha superado las dos condiciones anteriores, se procede a brindarle al usuario el servicio solicitado siempre y cuando el número de usuarios máximo permitido por módulo no se haya rebasado,

Una manera alternativa de asignar el módulo a otro u otros usuarios, es oprimiendo doble clic sobre la representación gráfica del módulo que se ha seleccionado para tal fin.

Cuando se selecciona la segunda opción del menú contextual de los módulos libres, es decir, "*Encendido forzado del módulo*", el Sistema realizará las operaciones necesarias para que el módulo sea abastecido de energía eléctrica y en consecuencia, la representación gráfica de dicho módulo cambiará a color amarillo según se describe en la Tabla 1.

Cabe destacar también que una vez que se han superado las verificaciones antes citadas, el **SAEALABFI** buscará en todos los módulos si hay algún registro con el número de cuenta que se ha seleccionado, de ser así el Sistema responderá con un mensaje de advertencia en el cual se muestra que el usuario ya se encuentra en otro módulo, indicando el número en el que se encuentra el usuario como se muestra en la Ilustración 30.

Ilustración 30. Respuesta a un usuario que ya se encuentra utilizando un módulo

Menú contextual de un módulo forzado no asignado

Al igual que en el caso anterior, el menú contextual está formado por 2 opciones: *Asignar módulo y Apagado forzoso del módulo*. La apariencia del menú se muestra en la Ilustración 31.

Ilustración 31. Menú contextual de un módulo forzado sin ser asignado

Para el caso de la primera opción, la forma en que opera y se procesa la asignación del módulo es idéntica a la que se ha descrito en el apartado de *Menú contextual de un módulo libre* (apartado anterior), mientras que para la segunda, realizará el proceso contrario al de encender módulo y en consecuencia la representación gráfica del estado que guarda el módulo cambiará a color verde cuyo significado se describe también en la Tabla 1.

Menú contextual de un módulo ocupado

En el caso de los módulos ocupados, se muestra una lista de opciones en la que se muestra: *Asignación de módulo, Préstamo de materiales, Liberar usuario, Liberar Módulo y Liberar Panel*, la representación gráfica de esta lista se muestra en la Ilustración 31.

Ilustración 32. Menú contextual de un módulo ocupado con capacidad para más usuarios

Cuando el módulo está ocupado por completo, la opción *Asignación de módulo*, desaparecerá de la lista de opciones. Esto lo puede observar en la Ilustración 33.

Ilustración 33. Menú contextual de un módulo ocupado sin capacidad para más usuarios

Como puede apreciarse en las Ilustraciones 32 y 33, ambos casos mantienen las operaciones comunes que se pueden realizar con los módulos sobre los cuales se muestra la lista de opciones desplegadas. En el caso de la opción *Asignar módulo*, el proceso de ejecución es el mismo que el que ya he descrito en el caso del *Menú contextual de un módulo libre* y del *Menú contextual de un módulo forzado no asignado*.

Es de destacarse que el número de usuarios que se permiten relacionar con un módulo depende de los parámetros de configuración por primera vez o de la opción de Configuración del Sistema como ya se ha descrito en temas previos. Cuando se ha excedido ese número y se intenta asignar otro usuario al módulo, el Sistema responderá con un mensaje de advertencia en el que se anuncia la imposibilidad de brindar el servicio a un usuario más. Esto lo puede observar en la Ilustración 34.

Ilustración 34. Respuesta a un modulo que está cubierto totalmente

Para la opción *Préstamo de materiales*, al abrir la opción el menú desplegará un submenú en el que se muestran los nombres de cada uno de los usuarios que se encuentran utilizando el módulo y en éste se podrá seleccionar al usuario al que se le va a prestar el material, herramientas o instrumentos con que cuenta el Laboratorio y que se pueden registrar mediante el Menú Inventario como ya se ha descrito en apartados anteriores dedicados a la explicación de las opciones de los menús y concretamente el menú Inventario. El despliegue citado presenta una apariencia similar a la mostrada en la Ilustración 35.

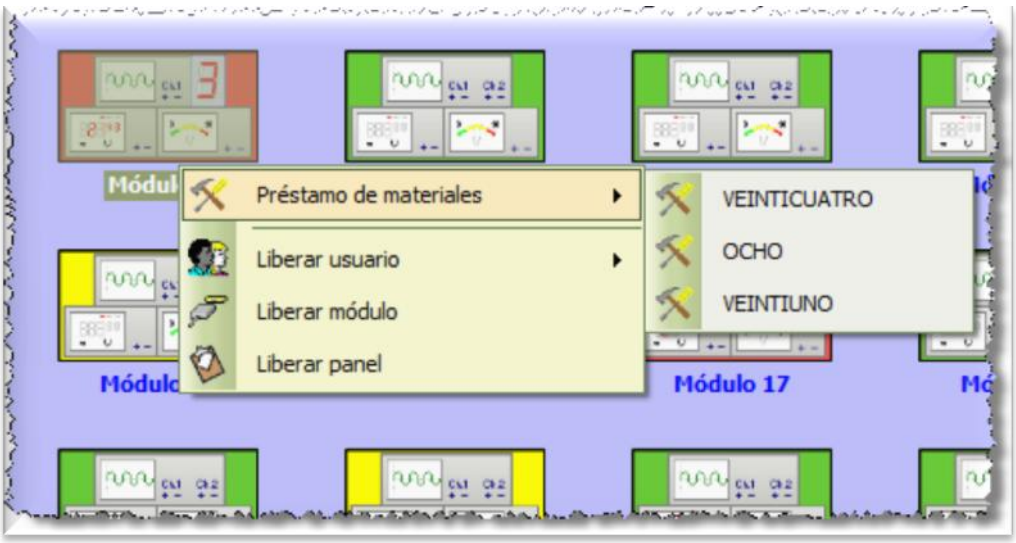


Ilustración 35. Menú préstamo de materiales

Cuando se selecciona a cualquiera de los usuarios (para este caso VEINTICUATRO, OCHO y VEINTIUNO), el **SAEALABFI** procederá a mostrar la ventana de préstamo de materiales en la cual se muestra de manera agrupada la lista de bienes que aún no han sido prestados y que quedan a disposición de quien así lo requiera. Para el ejemplo que aquí se muestra, se ha seleccionado al usuario OCHO. En la versión de liberación del Sistema, se han incluido tres categorías para los materiales y/o equipos las cuales son: *Manuales*, *Herramientas* e *Instrumentos*; estas categorías se pueden establecer en las opciones de Exportación o Importación del Inventario que ya se describió y si se desea mayor detalle revísese el Capítulo XI del manual del **SAEALABFI.**

La ventana de préstamo de materiales se presenta en la Ilustración 36.

Al expandir cada una de las categorías, se desplegará cada uno de los bienes que se encuentran disponibles proporcionándose una casilla de verificación mediante la cual se puede seleccionar al material desplegado de tal manera que se puede asignar uno o más bienes de la misma o de diferente categoría para el usuario que ha sido seleccionado.

Una vez que se ha establecido la lista, existe la posibilidad de que por cualquier razón ajena o fortuita, se decida no continuar con dicha asignación y para ello se ha proporcionado al botón *Cancelar*, que cerrará la ventana y no hará ninguna asignación y a su vez dejará libres todos los bienes que pudieran haberse seleccionado para el usuario. Al oprimir el botón *Registrar*, se procede a la asignación de los materiales seleccionados al usuario y quedarán registrados en el Sistema bajo resguardo del usuario seleccionado.

Ilustración 36. Ventana préstamos de materiales

En el momento en que queda registrado el resguardo de materiales a cualquiera de los ocupantes del módulo seleccionado, se agrega una opción más en el menú contextual que es la de *Entrega de Materiales* que al ser seleccionado, expandirá un submenú más en donde se muestra la lista de los usuarios que tienen resguardo de materiales en dicho módulo como en el ejemplo mostrado en la Ilustración 37.

Ilustración 37. Menú entrega de materiales

Al seleccionar alguno de los usuarios marcados con resguardo (desplegados en el submenú *Entrega de materiales*), el Sistema mostrará una ventana en la cual se despliega el listado de los bienes que tiene resguardados el usuario. Se debe considerar que al momento de hacer la entrega de materiales no necesariamente tiene que devolverse todos a la vez pero quizá sólo acuda a entregar uno de ellos, dos o varios, y en caso de que aún quede con al menos uno bajo resguardo, dicho usuario seguirá apareciendo en la lista indicada del menú de materiales

prestados a los usuarios. Si el tiempo de servicio vence o si el usuario se retira del Laboratorio, entonces la lista de los bienes resguardados se colocarán en el panel de *Materiales prestados* y es de suma importancia ser liberados antes de que se termine de utilizar al Sistema pues éste cuando se le solicite cerrar la aplicación indicará que aún hay materiales pendientes por entregar motivo por el cual no se puede proceder a cerrar el Sistema aún cuando no haya ningún módulo ocupado. En la ilustración 38 se percibe esta situación.

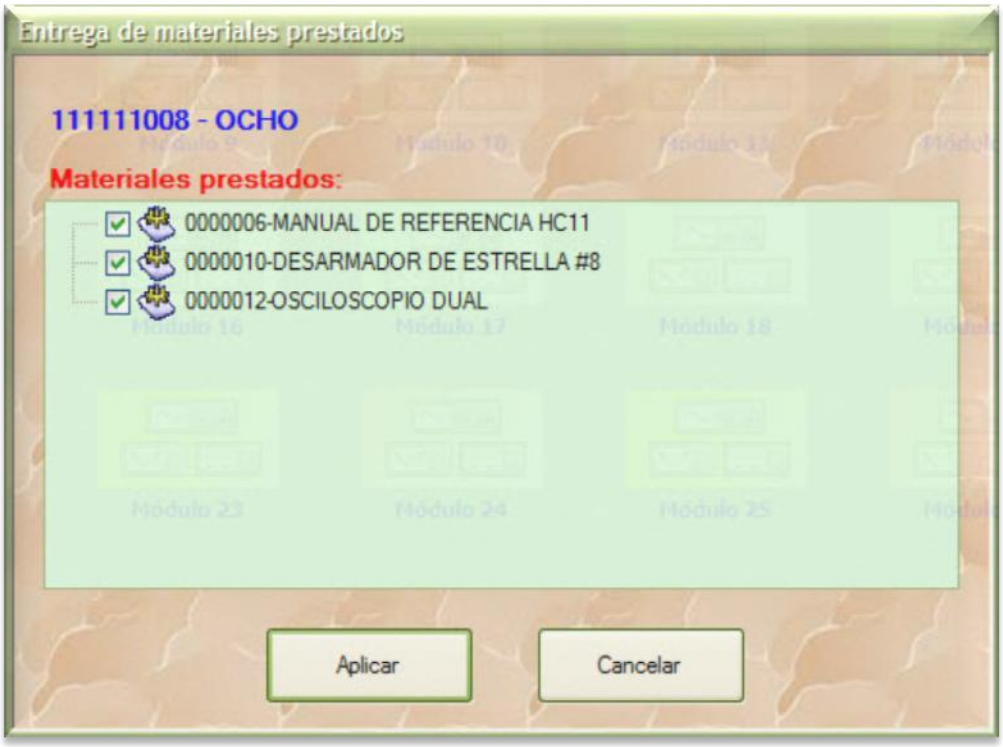


Ilustración 38. Ventana de entrega de materiales

Si por cualquier razón el usuario se acerca al operador del Sistema a solicitar que se le preste cualquier otro bien y éste ya tiene bajo su reguardo otros, esto no presenta mayor problema pues al ingresar a la opción de préstamo de materiales seguirá apareciendo el usuario y al seleccionarlo desde dicho submenú, el Sistema desplegará la ventana de préstamo de materiales indicando con una marca en la casilla de verificación aquéllos bienes que ya tiene bajo resguardo y mostrando sin verificar sólo aquéllos bienes que aún no han sido asignados.

Dentro del menú contextual, se encuentra la opción de *Liberar usuario*, ésta mostrará un submenú con los nombres de cada uno de los usuarios que se encuentran utilizando al módulo seleccionado y permite elegirlo como una opción de menú con la finalidad de Liberarlo del módulo y registrar en el Sistema que ya ha dejado de utilizar el servicio, si tiene bienes resguardados, será el momento de pasarlos a la lista de *Materiales por entregar* si es que no se realizó el proceso previamente a su liberación.

Si fuese el caso de que éste usuario sea el único utilizando el módulo, entonces además de lo ya expuesto, el módulo quedará libre para ser reutilizado por cualquier otro usuario que solicite el servicio a partir del momento de su liberación y considerando que si se encuentra algún usuario en la *Lista de espera*, de manera automática será asignado al módulo. La apariencia del menú de liberación de usuarios se muestra en la Ilustración 39.

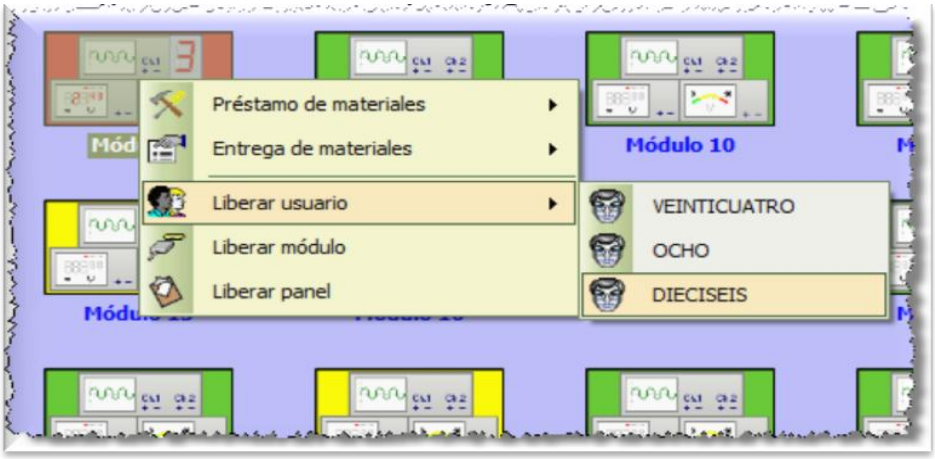


Ilustración 39. Menú de liberación de usuarios

En cuanto a la opción de liberación de módulo, ésta brinda al operador la posibilidad de liberar a cada uno de los usuarios que se encuentran haciendo uso del módulo seleccionado, evitándole al operador la tarea de liberar a cada uno de los usuarios por separado, la utilidad de ésta opción ya ha sido descrita pero una vez más cabe hacer mención de ella: si por razón alguna se sanciona a los integrantes del equipo que utiliza el módulo, en ese momento se puede suspender el servicio y complementariamente se puede registrar una sanción para cada uno de los usuarios del módulo.

Si uno o más usuarios tienen bajo resguardo algún bien del Laboratorio y no se entregaron antes de que les fuese suspendido el servicio, una indicación de dicho material se colocara en la lista de materiales pendientes por entregar y quedará activo hasta que no sea devuelto.

No hay reversibilidad en esta decisión, por lo que si así es necesario, cualquiera de los integrantes del equipo o todos a la vez son susceptibles de ser reasignados al módulo o a otro sometiéndose a las condiciones de uso del Laboratorio y si fuera necesario, anotarse en la lista de espera para que se les vuelva a proporcionar el servicio teniendo en consideración que éste usuario o el equipo completo no tenga sanciones registradas sin vencer en el Sistema.

La última opción que aquí se muestra, es la de *Liberación del Panel* cuyo propósito también ya se ha indicado y recordándola. Se puede decir que es un proceso en el cual todos y cada uno de los usuarios de cada uno de los módulos son liberados procediéndose a cortar el suministro de energía eléctrica a los módulos, salvo a aquéllos equipos cuyo estatus sea el de *Encendido forzado*. Como consecuencia de ello, los usuarios que se encuentran en lista de espera se asignarán a los módulos liberados y los resguardos pasarán a la lista de *Materiales pendientes por entregar*, como ya ha sido indicado en los casos anteriores.

Ilustración 40. Advertencia de liberación del panel

En el caso de esta opción, se brinda la posibilidad de retractarse del proceso antes de que comience mediante el cuadro de advertencia que se muestra en la Ilustración 40.

Menú contextual de un módulo forzado y ocupado

Las opciones que están asociadas a los módulos forzados y ocupados trabajan de manera exacta a los módulos ocupados como ya se ha descrito en el apartado anterior por lo que la apariencia y comportamiento del menú contextual es el mismo.

Menú contextual del área del Panel de Control donde no se encuentran los módulos

El menú contextual que se despliega cuando se oprime el botón derecho del mouse sobre el Panel de Control en cualquier área fuera de la zona de módulos, está conformada básicamente por tres opciones y en caso de que el operador del Sistema que se encuentre en sesión corresponda a la categoría de Administrador o Súper administrador, aparecerá una cuarta opción correspondiente a la *Administración del panel*, ésta última se describe con detalle en la descripción del *Menú de Administración* que se indicó en el apartado dedicado a la explicación del Menú Administración o para mayor información consúltese el Capítulo X del manual de referencia del **SAEALABFI.** Esta opción es un modo de acceso rápido la opción de Administración de panel que en el menú indicado y se incluye con fines de agilizar las tareas administrativas, su apariencia se puede observar en las Ilustraciones 41 y 42.

Ilustración 41. Menú contextual del panel de control fuera del área de módulos como operadores

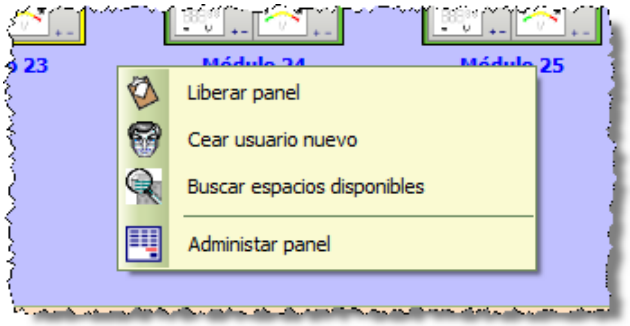


Ilustración 42. Menú contextual del panel de control fuera del área de módulos como administradores o súper-administrador

La primer opción es la de *Liberar panel*, opción que ya se ha descrito en los menús contextuales anteriores y cuya explicación queda a la revisión de los mismos en la opción *Liberación del panel* en el menú *Archivo* o desde las opciones de los menús contextuales ya descritos en párrafos anteriores.

La opción *Crear usuario nuevo*, es una opción mediante la cual se puede acceder a la ventana de creación de usuarios nuevos en el catálogo de usuarios que tienen derecho a hacer uso de los servicios que se ofrecen en el Laboratorio según se ha especificado en el reglamento interno del Laboratorio y que se hará saber a los mismos en el momento oportuno. Esta opción es un enlace directo a la opción *Agregar Usuario* del menú *Usuarios* por lo que para tener más detalles favor de referirse al Capítulo IX del manual de referencia del **SAEALABFI** o en la descripción del menú *Usuarios* en de este anexo. La apariencia de esta ventana se observa en la Ilustración 43.

Ilustración 43. Apariencia de la ventana de creación de usuarios nuevos

La tercera opción es *Buscar espacios disponibles*, también es un modo de acceso rápido a la opción *Buscar espacios disponibles* del *Menú Archivo* como ya fue descrito en este anexo y para mayor detalle, consulte el Capítulo VI del manual de referencia del **SAEALABFI**. Esta opción busca en todos los módulos cuantos usuarios aún faltan para que los módulos sean cubiertos por completo.

El resultado de dicha búsqueda se muestra con un cuadro que despliega la información relacionada a los módulos y sus respectivos espacios, en caso de que el Laboratorio esté

cubierto al cien por ciento, se desplegará un mensaje en el cual se indica que no hay espacios disponibles y por consiguiente no podrá acomodarse ni un usuario más en los espacios que pudiesen quedar. La forma de este cuadro de diálogo se muestra en la Ilustración 44.

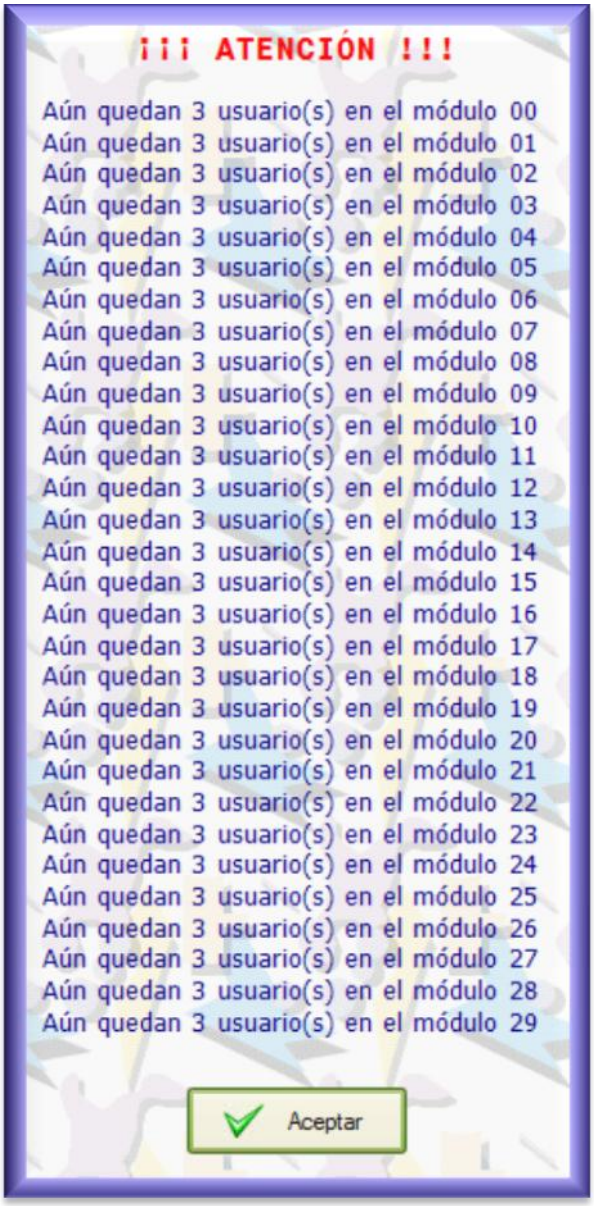


Ilustración 44. Resultado de la búsqueda de espacios

Paneles complementarios

Son tres los paneles complementarios en los que se apoya el Sistema para realizar las tareas de administración del Laboratorio, cuya significado y apariencia se describen en seguida.

Panel de Módulos próximos a vencer el tiempo de servicio. Este panel es empleado para indicar al operador cuales módulos se encuentran próximos a terminar el tiempo máximo configurado para brindar servicio a los usuarios del **SAEALABFI**, este aviso, por acuerdo en las reuniones de análisis de requerimientos se puede configurar en un rango de 5 a 40 minutos, tiempo en el cual el operador del Sistema tiene la oportunidad de hacer una prorroga de tiempo de servicio a los usuarios del módulo seleccionado por 15, 30, 45 ó 60 minutos.

Si no es atendido el aviso, al transcurrir el tiempo reglamentario máximo del cual puede brindarse el servicio a los usuarios, se procederá a liberar el módulo, dejándolo libre para ser reasignado a cualquier otro usuario e incluso a los ocupantes del módulo indicado, sometiéndose en este último caso a las condiciones generales de servicio a los usuarios.

Cuando se selecciona a un módulo dentro del panel y le es asignado una prorroga de tiempo, el módulo seleccionado se quitará de la lista y dentro de los registros internos del Sistema se actualizará el cronómetro con los minutos que se les ha agregado de más.

El panel se encuentra localizado en la parte inferior izquierda del área de trabajo, su apariencia es la que se muestra en la Ilustración 45, mientras no se encuentre algún módulo en espera de atender el tiempo de vencimiento, el menú contextual de prórroga de tiempo quedará deshabilitado y de igual manera sucederá cuando se oprima el botón derecho del mouse sobre el área fuera de la lista de módulos.

Ilustración 45. Apariencia general del Panel de módulos próximos a vencer el tiempo de servicio

De manera complementaria, este panel se administra de manera conjunta cuando se ha habilitado la opción de *Administración de panel* que se explica a detalle en el Capítulo X del manual de referencia del **SAEALABFI**.

A medida que va transcurriendo el tiempo, la etiqueta que indica el tiempo que le resta al módulo será actualizada, esto se realiza con una frecuencia de intervalos de 15 segundos y está sujeta al cronómetro general del Sistema que actualiza la información en periodos de cada minuto. Un ejemplo de la lista que contiene módulos que se encuentran en proceso de vencer el tiempo de servicio, se puede observar en la Ilustración 46.

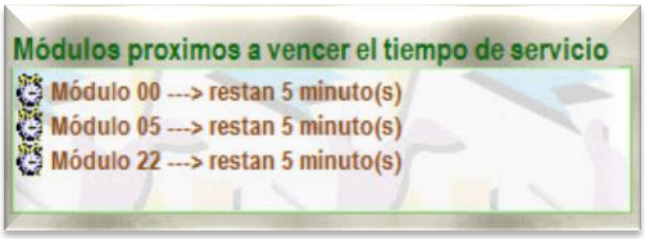


Ilustración 46. Apariencia general del Panel de módulos próximos a vencer

Panel de préstamo de materiales. Es un pequeño panel colocado en la parte inferior central del entorno de trabajo del Sistema cuya finalidad es la de depositar en él a todos aquéllos usuarios que tienen bajo resguardo materiales y/o bienes que forman parte del Inventario del Laboratorio y que ya han dejado de tener servicio por cualquiera de las razones que se han expresado a lo largo de éste anexo.

La lista se va actualizando en la medida de que se va haciendo uso del servicio del Laboratorio y sólo serán colocados aquéllos usuarios a los que se les ha agotado el tiempo de servicio, a quienes se les ha desconectado el suministro de energía y como consecuencia a

aquéllos usuarios de los módulos que se encontraban colocados en el *Panel de Módulos próximos a vencer el tiempo de servicio*. La apariencia del panel, es la que se muestra en la Ilustración 47.

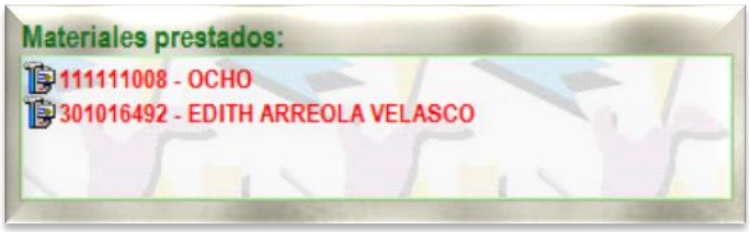


Ilustración 47. Apariencia del panel de materiales prestados

Cuando existe al menos un usuario en la lista, se activará una opción de menú contextual con una opción cuya etiqueta es *Entregar,* como se muestra en la Ilustración 48.

Ilustración 48. Menú contextual de entrega de materiales

Al seleccionarse dicha opción sobre un usuario de la lista, aparecerá la ventana de entrega de materiales que se representa en la Ilustración 38. En dicha ventana se muestran los datos básicos de identificación del usuario. Es decir, su número de cuenta y nombre y en seguida la lista de materiales que están resguardados por el usuario, aparecen con una casilla de verificación en la parte izquierda de cada línea de la lista indicando qué bienes tiene y por consiguiente entregará.

Si por alguna razón queda pendiente de entregar al menos un bien para el usuario y al oprimir el botón *Aplicar*, el usuario continuará vigente en la lista de materiales por entregar, por lo que sólo será liberado del panel cuando se haya hecho la entrega física de todos los bienes y se haya registrado en el Sistema.

Se debe destacar que para poder cerrar el Sistema, una de las condiciones necesarias para cumplir dicha tarea es la de que el panel de Materiales prestados se encuentre vacía.

Panel de Lista de espera. Este panel se encuentra ubicado en la parte inferior derecha del área de trabajo del **SAEALABFI**, su propósito es el de contener la lista cronológica de aquéllos usuarios que acuden al Laboratorio a solicitar el servicio, pero en ese momento se encuentran ocupados todos los módulos o no se encuentra algún compañero de equipo dentro del Laboratorio.

La espera de los usuarios será cronometrada por lo que un pequeño reloj aparecerá en la parte izquierda del número de cuenta y nombre del usuario en la lista, teniendo dos sucesos posibles; primero, sí es liberado alguno de los módulos por cualquiera de las vías indicadas, en ese momento se asignará al módulo liberado de manera selectiva (cuando hay 2 o más usuarios en espera) o de manera automática (cuando sólo exista un usuario en lista de espera). La otra opción es que el usuario desista de seguir esperando turno.

El aspecto del panel se puede observar en la Ilustración 49.

Ilustración 49. Aspecto del panel de lista de espera

El panel está dotado de un pequeño menú contextual que se actualizará a medida de que se haga uso del mismo. Dichas opciones son las de *Agregar* y *Quitar*, la primer opción permite anexar un usuario nuevo en la lista de espera y será la única opción mostrada cuando no haya ningún usuario en la lista o cuando habiendo algún usuario en la lista y se oprime el botón derecho en un área fuera de la lista de usuarios pero dentro del panel.

Si se selecciona la opción Agregar cuando aún hay módulos por asignar, el Sistema responderá con el mensaje que se muestra en la Ilustración 50.

Ilustración 50. Mensaje que indica que aún hay módulos para ser ocupados por los usuarios

Cuando hay usuarios en la lista, aparecerá la segunda opción y se mostrará cuando se oprima el botón derecho o secundario del mouse sobre la identificación de un usuario, al seleccionar la opción *Quitar*, el Sistema se da por enterado de que ese usuario ha decidido no continuar esperando a ser atendido al encontrarse un módulo desocupado por cualquiera de las vías ya mencionadas.

Barra indicadora del uso del panel

La barra indicadora del uso del panel está formada por cuatro secciones, tres de ellas siempre visibles y una que será visualizada cuando se realizan determinadas tareas secundarias.

En primer lugar, está la sección de *Módulos utilizados* donde mediante una barra indicadora de progreso se refleja gráficamente qué tan ocupados se encuentran los módulos del Laboratorio. El aspecto de esta sección se muestra en la Ilustración 51.

Ilustración 51. Sección de módulos utilizados

La segunda sección es la sección de *Usuarios en el Laboratorio*, en donde se indica mediante un número cuántos usuarios se encuentran repartidos entre los módulos que están instalados en el Laboratorio. Su aspecto se muestra en la Ilustración 52.

Ilustración 52. Sección de usuarios en el Laboratorio

La tercer sección es la sección de *Módulo activo*, que indicará qué módulo es el que está seleccionado al momento de realizar las operaciones propias del **SAEALABFI**, si no se encuentra seleccionado alguno se mostrará la etiqueta *"Ninguno"*, en caso contrario, el número del módulo activado. Su aspecto se muestra en la Ilustración 53.

Ilustración 53. Sección de Módulo Activo

Finalmente, la cuarta sección aparentemente se encuentra oculta y sólo se activará cuando se acceda a alguna de las opciones que se encuentran agrupadas dentro del árbol de menús. Aquí se muestra una barra de progreso que indica que el Sistema está en espera de que dicha opción termine de operar según se muestra en la Ilustración 54.

Ilustración 54. Sección de espera de terminación de opciones del árbol de menús

Consideraciones de la operación del Sistema con la *Interfaz Electrónica*

En esta liberación del SAEALABFI, se ha integrado y considerado todas las posibilidades que pudiesen suscitarse al operar al Sistema en manera conjunta con la Interfaz Electrónica de manera física. Como se indica en el Capítulo IV del manual de operación del **SAEALABFI**, en la fase de inicio de sesión dentro del apartado *fase de identificación de la Interfaz electrónica;* el Sistema desde el inicio de Sesión verificará mediante el protocolo de comunicación con la *Interfaz Electrónica* si es que ésta se encuentra conectada, alimentada y lista para ser utilizada, si se cumplen estas tres condiciones, el **SAEALABFI** comenzará a operar de manera ordinaria y de no ser así se le notificará al operador de que hay problemas en la identificación de la Interfaz. En este caso el Sistema detendrá su funcionamiento hasta que se intente de nuevo ingresar a él y el problema con la *Interfaz Electrónica* haya sido resuelto.

Pérdida de comunicación con la *Interfaz Electrónica*

Se puede presentar el caso en que por alguna situación no prevista, se pierda la comunicación con la *Interfaz Electrónica*; esto se puede deber a variaciones en el suministro de energía eléctrica o en algún problema que se presente en el uso de los módulos del Laboratorio por parte de los usuarios registrados y con derecho a hacer uso de él o por cualquier otra que no sea previsible. Cuando sucede esta situación, el Sistema primeramente cambiará el ícono de conexión dentro de la barra de identificación y búsqueda por la forma de una cadena rota y en seguida atenuará la ventana principal del **SAEALABFI** al 85% de su intensidad normal y mostrará un ícono al centro del panel en forma de estrella rotando en espera de poderse restablecer la comunicación con la *Interfaz Electrónica* como se puede observar en la Ilustración 55.

Ilustración 55. Indicador de espera de respuesta de la Interfaz electrónica

Cuando aparece el ícono de espera de la *Interfaz Electrónica*, el **SAEALABFI** se encontrará en un proceso de verificación constante en espera de que haya respuesta, si tras el paso de aproximadamente 10 segundos no se ha reconocido nuevamente la conexión, el Sistema le solicitará al operador que seleccione la opción de continuar buscando o de cancelar la búsqueda, dicha solicitud la hará saber mediante el cuadro de diálogo que aparece en la Ilustración V-56.

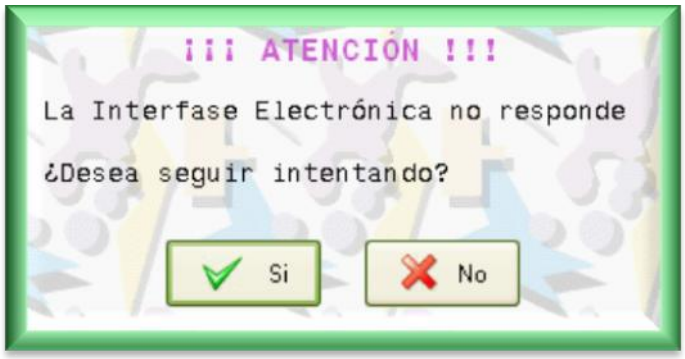


Ilustración 56. Pregunta sobre el intento de búsqueda de la Interfaz Electrónica

Como se puede observar, se le ofrecen dos opciones al operador del Sistema en relación a la pérdida de conexión con la *Interfaz Electrónica*: Si se oprime en el botón *Si*, el **SAEALABFI** continuará buscando y si durante el proceso de búsqueda se logra volver a establecer la conexión con la *Interfaz Electrónica* entonces se ocultará el cuadro de diálogo mostrado en la Ilustración anterior, se ocultará la estrella que indica la búsqueda de la *Interfaz Electrónica*, la opacidad del panel central volverá a ser del 100% y el ícono indicador de conexión cambia a una cadena cerrada. Una vez efectuadas las acciones anteriores, se realizará un ajuste para actualizar el estado de los módulos para que corresponda con el que se encuentra indicado en el panel principal del **SAEALABFI**. Ésta última acción también se puede realizar desde la opción *Actualizar Interfaz* del menú *Administración* como ya se expuso en el tema dedicado al menú Administración.

Si el operador decide no continuar con la búsqueda de la *Interfaz Electrónica*, el **SAEALABFI** procederá a retornar al panel de control (panel principal), restableciendo la opacidad del panel al 100%, desaparecerá la estrella de búsqueda de Interfaz y colocará en el ícono de identificación de Interfaz a una cadena rota. Una vez que se ha concluido con estas acciones, el Sistema realizará un proceso de Liberación del Panel sin solicitar la confirmación al

operador con la finalidad de liberar a todos los usuarios que pudiesen estar en ese momento haciendo uso del Laboratorio y para poner en el panel de entrega de materiales a quienes tengan materiales en resguardo. La operación de liberar al panel se indicó en la descripción del menú *Archivo* dentro de este anexo.

En este momento, el Sistema está listo para que desaloje a todos los usuarios del Sistema y terminar la operación de éste, pero si quiere volver a intentar la re conexión de la *Interfaz Electrónica*, se encuentra preparado para que en el primer intento de asignar algún módulo a algún usuario se reactive la secuencia ya descrita en este apartado como se puede observar en la siguiente Ilustración.

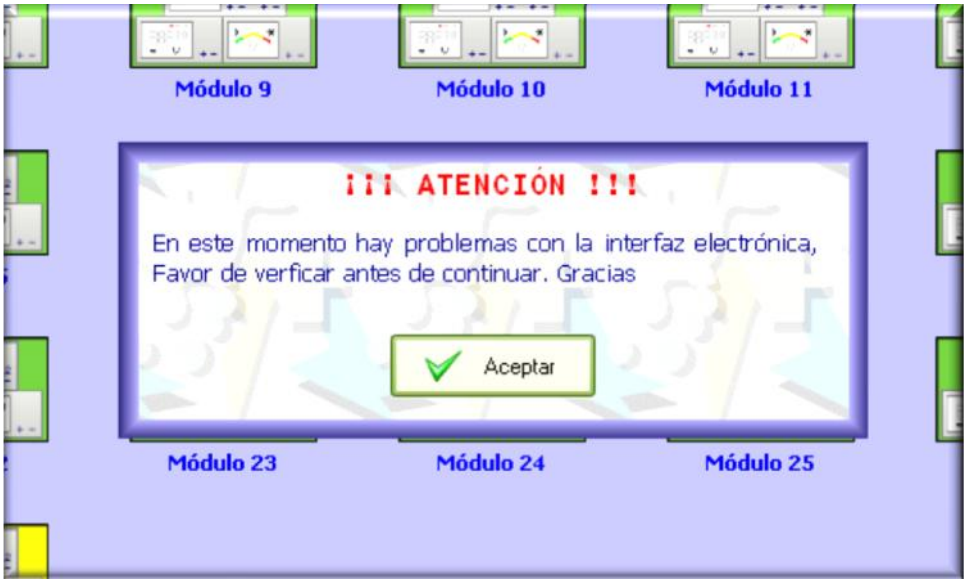


Ilustración 57. Aviso de la no disponibilidad de Servicio por falla en la Interfaz Electrónica

Icono en la barra de tareas de Windows

El **SAEALABFI** tiene incorporado un ícono que se coloca en el área de notificación de la barra de tareas de Windows, su finalidad es la de ofrecerle al operador una manera rápida de acceder a las funciones más comunes que le ayudan a las tareas de administración.

Los íconos de este tipo aparecerán siempre visibles a no ser que explícitamente se le asocie con un estatus de *siempre oculto* en el área de notificación de la barra de tareas de Windows que corresponde al área que se encuentra en la parte inferior derecha de la ventana principal del Sistema Operativo. Como ejemplo de esto obsérvese la Ilustración 58.

Ilustración 58. Sección del área de notificación

Cuando se termina de inicializar al **SAEALABFI**, éste colocará un ícono en el área mostrada en la zona de la barra de tareas, cuya apariencia es la de un conector de un cable. Este ícono mostrará información sobre las operaciones de encendido y apagado se realicen hacia la *Interfaz Electrónica* como se observa en la Ilustración 60.

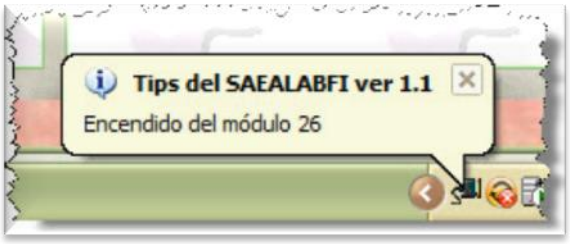


Ilustración 59. Notificación de acciones realizadas en los módulos

El botón tiene asociado un menú contextual el cual hace referencia a las funciones más comunes del Sistema para que dese allí se puede acceder a ellas. Si se oprime dos veces seguidas el botón principal sobre el ícono del **SAEALABFI**, se determinará primero, si la ventana principal está vacía, entonces se ocultará y quedará realizando las operaciones normales en segundo plano. Además de que el globo (figura de notificación) cambiará y le indicará al operador que oprima con el botón secundario sobre el ícono para mostrar el menú contextual, o en su defecto, que oprima dos veces el botón principal para ocultar o mostrar la ventana principal según sea su estado actual. Este globo se puede observar en la Ilustración 60.

Ilustración 60. Aviso sobre el uso del ícono del SAEALABFI

Si la ventana principal del Sistema se encontraba previamente oculta, al oprimir dos veces seguidas al botón, la ventana se mostrará en su estado previo (*Normal o Maximizado*), y cuando se encuentre con la ventana visualizada se oprime el botón minimizar, el Sistema lo tratará como si se hubiese oprimido dos veces sobre el botón en el área de notificación y trayendo como consecuencia la ocultación de la ventana principal y en seguida mostrando el globo de *Opciones de acceso rápido del* ***SAEALABFI*** como se observa en la Ilustración 60.

Ahora, sin importar cual sea el estado de la ventana principal del **SAEALABFI**, si el operador del Sistema oprime el botón secundario sobre el ícono en el área de notificación del Sistema Operativo, se le mostrará al operador el menú contextual cuya apariencia se muestra en la Ilustración 61.

Ilustración 61. Menú contextual del ícono del SAEALABFI en el área de notificación

Como se puede observar, se han incluido las opciones de Archivo, Usuarios, Administración, Inventario, Acerca de… y Salir.

Para el caso de la opción Archivo, el Sistema a su vez desplegará otro menú con la apariencia mostrada en la Ilustración 62.

Ilustración 62. Menú Archivo

En este menú, se observan las opciones *Configurar*, *Buscar espacios disponibles*, *Sanciones*, *Quitar sanciones*, *Liberar panel* y *Lista de usuarios en el laboratorio*. Todas estas opciones corresponden a las que se describieron en el menú *Archivo* de la ventana principal del **SAEALABFI** y su operación está sujeta a los niveles de seguridad establecidos para cada opción. Para más detalle, revise el apartado del menú *Archivo*.

Para el caso de la opción *Usuarios*, el Sistema mostrará al menú usuarios cuya apariencia se puede observar en la Ilustración 63.

Ilustración 63. Menú Usuarios

Obsérvese que las opciones para este menú son las de *Agregar*, *Actualizar*, *Exportar información de usuarios* e *Importar información de usuarios*. Estas opciones corresponden a las opciones que se agrupan dentro del menú *Usuarios* de la ventana principal; su acceso y uso están sujetos a los criterios de seguridad que para cada opción se han establecido, por lo que se sugiere revisar el apartado en el que se detallan cada una de las opciones.

Para el caso de la opción *Administración*, el Sistema mostrará un submenú en el que se muestran las opciones más comunes en relación a la administración del Sistema como se muestra en la Ilustración 64.

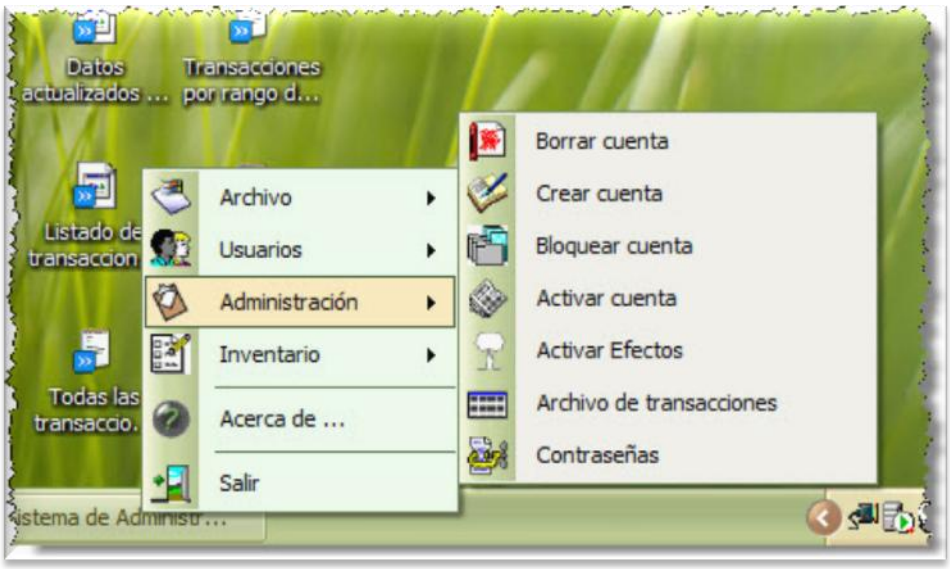


Ilustración 64. Menú Administración

Para este caso, se observa que las opciones que se disponen a través de esta vía son: *Borrar cuenta*, *Crear cuenta*, *Bloquear cuenta*, *Activar cuenta*, *Activar/Desactivar efectos*, *Archivo de transacciones* y *contraseñas*. Estas opciones corresponden a las opciones más frecuentes en cuanto a la administración del Sistema y por consiguiente, la función de cada una de ellas corresponde con las que se han descrito en el apartado del *Menú Administración*. Al igual que en los casos anteriores, el uso de cada una de estas opciones esta sujeto a los criterios de Seguridad que se han establecido para cada opción.

Para el caso de la opción *Inventario*, el Sistema le mostrará al operador un submenú en el cual se muestran las opciones más comunes relacionadas con el Inventario general del Sistema. Como se muestra en la Ilustración 65.

En la Ilustración se aprecian las opciones que están asociadas el menú *Inventario*: *Movimientos*, *Módulos*, *Exportar* e *Importar*. Todas estas tienen su equivalencia dentro del menú principal de la ventana del Sistema por medio de la opción *Inventario*. La manera en que funcionan, su alcance así como sus restricciones se encuentran explicados en el apartado dedicado a las opciones del Menú *Inventario*.

La opción *Acerca de …*, visualiza la ventana de descripción de la versión del **SAEALABFI** y los derechos asociados como se muestra en la Ilustración 66.

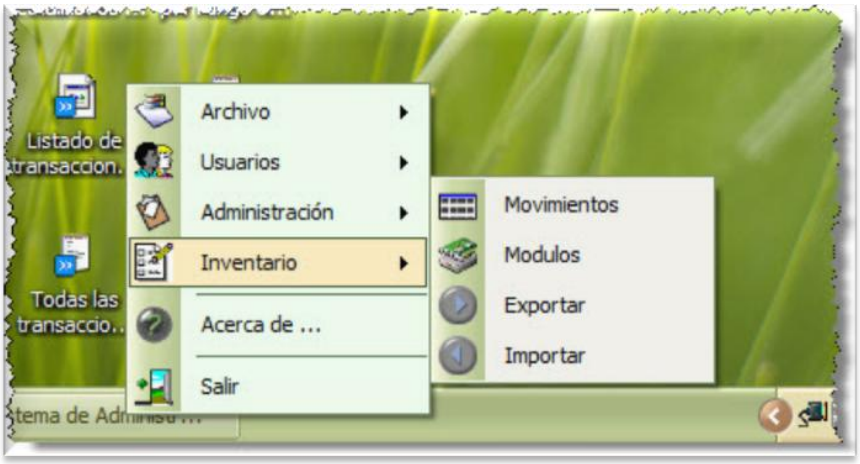


Ilustración 65. Menú Inventario

Ilustración 66. Ventana Acerca de SAEALABFI

Por último, la opción *Salir* permite al operador terminar la ejecución del programa siempre y cuando se cumplan las condiciones que ya se indican en el Capítulo VI del manual de operación del SAEALABFI.

Derechos por categoría de Operador

Finalmente, se incluye una tabla en la que se concentran todas las opciones agrupadas de cada una de las opciones con que se ha integrado al **SAEALABFI**, según el menú al que pertenecen, además de los operadores cuya categoría puede acceder.

S	A	U	Opción	Menú
✓	✓	✓	Configurar	**Archivo**
✓	✓	✓	Buscar espacios disponibles	**Archivo**
✓	✓	✓	Lista de usuarios en el Laboratorio	**Archivo**
✓	✓	✓	Sanciones	**Archivo**
✓	✓	✓	Quitar sanciones	**Archivo**
✓	✓	✓	Liberar panel	**Archivo**
✓	✓	✓	Cambio de Operador	**Archivo**
✓	✓	✓	Salir	**Archivo**
✓	✓	✓	Movimientos de los usuarios	**Consultas**
✓	✓	✓	Movimientos por módulo	**Consultas**
✓	✓	✓	Sanciones por usuario	**Consultas**
✓	✓	✓	Movimientos de la lista de espera	**Consultas**
✓	✓	✓	Préstamo de materiales por usuario	**Consultas**
✓	✓	✓	Movimientos por semestre	**Consultas**
✓	✓	✓	Préstamos de los materiales por semestre	**Consultas**
✓	✓	✓	Movimientos por módulo y semestre	**Consultas**
✓	✓	✗	Movimientos por módulo, semestre y usuario del Sistema	**Consultas**
✓	✓	✗	Usuarios creados por operador del Sistema	**Consultas**
✓	✓	✗	Datos de usuarios actualizados por operador del Sistema	**Consultas**
✓	✓	✗	Sanciones aplicadas por operador del Sistema	**Consultas**
✓	✓	✗	Tiempo en la lista de espera	**Estadísticas**
✓	✓	✗	Uso de los equipos del laboratorio	**Estadísticas**
✓	✓	✗	Alumnos que usaron el servicio del laboratorio	**Estadísticas**
✓	✓	✗	Materiales prestados	**Estadísticas**
✓	✓	✗	Servicios proporcionados por usuario	**Estadísticas**
✓	✓	✗	Materiales prestaos por usuario	**Estadísticas**
✓	✓	✗	Sanciones aplicadas por usuario	**Estadísticas**
✓	✓	✗	Agregar usuario	**Usuarios**
✓	✓	✗	Actualizar datos de usuarios	**Usuarios**
✓	✓	✗	Exportar información de usuarios	**Usuarios**
✓	✓	✗	Importar información de usuarios	**Usuarios**

S	A	U	Opción	Menú
✓	✓	✗	Borrar cuenta	**Administración**
✓	✓	✗	Crear cuenta	**Administración**
✓	✓	✗	Bloquear cuenta	**Administración**
✓	✓	✗	Activar cuenta	**Administración**
✓	✗	✗	Restaurar Sistema / Anular programación de restauración	**Administración**
✓	✓	✗	Activar / Desactivar efectos	**Administración**
✓	✓	✗	Ajuste de parámetros	**Administración**
✓	✓	✗	Crear archivo de transacciones	**Administración**
✓	✗	✗	Restaurar registro de transacciones desde cero	**Administración**
✓	✗	✗	Restaurar registro de transacciones a partir del punto actual	**Administración**
✓	✓	✓	Cambio de contraseña	**Administración**
✓	✗	✗	Administración de cuentas	**Administración**
✓	✗	✗	Ajuste de seguridad	**Administración**
✓	✗	✗	Administrar el panel	**Administración**
✓	✓	✓	Actualizar Interfaz	**Administración**
✓	✓	✓	Apagar Módulos de la Interfaz	**Administración**
✓	✓	✓	Movimientos del inventario general	**Inventario**
✓	✓	✓	Estado de los módulos	**Inventario**
✓	✓	✗	Exportar inventario	**Inventario**
✓	✓	✗	Importar inventario	**Inventario**

S.- *Súper – Administrador;* **A**.- *Administrador;* **U**.- *Usuario*

BIBLIOGRAFÍA

Enrique Palacios, Fernando Remiro, Lucas J. López
Microcrontrolador PIC16F84, Desarrollo de proyectos, 2ª edición
Editorial Alfaomega Ra-Ma

José Ma. Angulo Usategui, Ignacio Angulo Martínez
Microcrontroladores PIC16F84
Diseño Práctico de aplicaciones
3a. Edición
Editorial Mc Graw Hill

Ronald J. Tocci
Sistema**s Digitales, Principios y Aplicaciones, quinta edición**
Editorial Prentice Hall

Barry B. Brey
Los microprocesadores Intel, Arquitectura, programación e interfaces
Tercera edición
Editorial Prentice Hall

Fernando G. Guerrero and Carlos Eduardo Rojas
SQL Server 2000 Programming by example
Editorial QUE

Libros en Pantalla de Microsoft SQL Server 2005
Septiembre de 2007
Microsoft

Luis Miguel Blanco
Programación en Visual Basic **.NET**
Editorial grupo Eidos

Gary Coronell and Johnathan Morrison
Programming VB.NET: A guide for Experienced Programmers
Editorial Apress

Tim Patrick, Steven Roman, Ron Petrusha, Paul Lomax
Visual Basic **2005 in a Nutshell, Third Edition**
Editorial O'Reilly Media Inc.

Gretory Brill
CodeNotes for .NET
Editorial Random House

Craig Utley
A Programmer's Introduction to Visual Basic**.NET**
Editorial SAMS

Introducción a la Programación Orientada a Objetos **en** Visual Basic **.NET**
Departamento de Informática y automática
Universidad de Salamanca

Arquitectura de de aplicaciones .NET: Diseño de aplicaciones y servicios
Patterns & Practices
Microsoft

Microsoft .NET Framework Primer for the Visual Basic **Developer**
Microsoft

Visual Basic **.NET y .NET Framework**
Libros en pantalla de Microsoft Visual Studio .NET
Microsoft

Wilder Ramírez Delgado
Implementación de una *interfaz* de comunicación serial virtual
Con un micro controlador PIC16F84
Centro de investigación y desarrollo
Facultad de Ingeniería
Universidad de Mazinales, España

El Mundo de la Electrónica, TV, AUDIO y VIDEO
Los componentes en corriente alterna
Enciclopedia Visual de la Electrónica
Capítulo 5
http://www.legga.com

A tutorial on parallel port Interfacing
http://logix4u.net

RS232 Data Interface
A Tutorial on Data Interface and cables
http://www.arcelect.com

+5V-Powered, Mutichannel RS-232 Drivers/Receivers
MAX220-MAX249
19-4323; Rev 15; 1/06
http://www.maxim-ic.com

BTA/BTB 08 B TRIACS
SGS-THOMSON Microelectronics

MOC3011 Semiconductor Technical Data
Motorola
MOC3010/D
http://www.motorola.com

SN74155 DUAL 2-LINE TO 4-LINE DECODERS/DEMULTIPLEXERS
Specification datashet
SDLS057
Texas Instruments

SN74LS259B 8-BIT ADDESSABLE LATCHES
Specification datashet
SDLS086
Texas Instruments

MICROCHIP PIC16FX DataSheet
DS30430C
http://www.microchip.com

MICROCHIP PIC16F84A
Data Sheet
DS35007B
http://www.microchip.com

MICROHCIP PICmicro Mid-Range MCU Family
Reference Manual
DS33023A
http://www.microchip.com

MICROCHIP MPASM Assembler,
MPLINK Object Linker,
MPLIB Object Librarian
User's Guide
DS33014J
http://www.microchip.com

ÍNDICE ALFABÉTICO

A

Acerca de… 194
Administración 191
AMPLIACIONES, CONTINUIDAD Y MEJORAS DEL SISTEMA 171
ANEXO: INSTRUCTUCTIVO DE USO DEL SISTEMA 185
Antecedentes 141
Antecedentes de la solución 21
APENDICE A: TARJETAS PCB 174
APÉNDICE B: PRUEBAS, MEDICIONES E IMPLEMENTACIÓN 176
Archivo 187
Área común de trabajo 13
Ayuda 193

B

Barra de identificación y búsqueda 195
Barra de título 197
Barra indicadora del uso del panel 220
Bases de datos 29
BIBLIOGRAFÍA 230
Bit de inicio o arranque 39
Bit de parada 40
Bit de paridad 40
Bits de datos 39
Bloque 144
Bloques 149
Botones de estado 198

C

CAPÍTULO I EL LABORATORIO ABIERTO 10
CLASE clsGeneral 86
CLASE clsInterfase 86
CLASE clsModulo 87
CLASE clsSeguridad 89
CLASE frm_Administración_ArchivoDeTransacciones 92
CLASE frm_Administración_BloquearCuenta 93
CLASE frm_Administración_BorrarCuenta 93
CLASE frm_Administración_CambiarContraseña 94
CLASE frm_Administración_CrearCuentaNueva 95
CLASE frm_Administración_Cuentas 95
CLASE frm_Administración_NombreNuevoDeUsuario 97
CLASE frm_Administración_UsuarioNuevoRápido 99
CLASE frm_Inventario_ConsultaDeLosMovimientos 120
CLASE frm_Usurarios_SelecciónDeUsuario 133
CLASE frmAdministraciónPanel 97
CLASE frmBienvenida 99
CLASE frmBox 100
CLASE frmConfiguraciónPorPrimeraVez 100

CLASE frmConfigurar 101
CLASE frmConsulta_ActualizaciónDatosXOperador 102
CLASE frmConsulta_ListaDeEspera 103
CLASE frmConsulta_MovsXMódulo 104
CLASE frmConsulta_MovsXMóduloYSemestre 107
CLASE frmConsulta_MovsXOperadorMóduloYSemestre 105
CLASE frmConsulta_MovsXSemestre 105
CLASE frmConsulta_MovsXUsuario 106
CLASE frmConsulta_PréstamosXSemestre 107
CLASE frmConsulta_SancionesxUsuario 109
CLASE frmConsultaDeSancionesAplicadasPorOperadorDelSistema 108
CLASE frmConsultaDeUsuariosCreadosPorOperadorDelSistema 109
CLASE frmDespedida 110
CLASE frmEspera 111
CLASE frmEstadística_MaterialesPrestadosXUsuario 111
CLASE frmEstadísticas_MaterialesPrestados 113
CLASE frmEstadísticas_ServicioPorUsuario 115
CLASE frmEstadísticas_UsoDeEquipos 118
CLASE frmEstadísticas_UsoDeEquiposPorUsuarios 117
CLASE frmEstadísticasDeSancionesAplicadasPorUsuario 114
CLASE frmInicioDeSesión 119
CLASE frmInventario_Acción 121
CLASE frmInventario_Edición 119
CLASE frmInventario_Módulos 120
CLASE frmListadoDeUsuarios 122
CLASE frmMateriales_Entrega 122
CLASE frmMateriales_Préstamos 123
CLASE frmPregunta 123
CLASE frmPrincipal 124
CLASE frmSanciones 130
CLASE frmUsuarios_Actualizar 131
CLASE frmUsuarios_Agregar 132
CLASE rmUsuarios_Confirmar 133
Código 1. Creación de la Base de Datos SAEALAB 61
Código 10. Creación de la Entidad tbSanciones 73
Código 11. Creación de la Entidad tbSeguridad 74
Código 12. Creación de la Entidad tbSysParms 75
Código 13. Creación de la Entidad tbSysTemp 76
Código 14. Creación de la Entidad tbTransacción 77
Código 15. Creación de la Entidad tbUsuarios 79
Código 2. Creación de la Entidad tbDump 63
Código 3. Creación de la Entidad tbEspera 64
Código 4. Creación de la Entidad tbInventario 65
Código 5. Creación de la Entidad tbInvTemp 66
Código 6. Creación de la Entidad tbModulos 69
Código 7. Creación de la Entidad tbMovimientos 70
Código 8. Creación de la Entidad tbMovsModulos 71
Código 9. Creación de la Entidad tbPréstamos 72
Columnas de categorías 34
Columnas de datos sin procesar 34

Columnas de identificadores 34
Columnas relacionales o diferenciales 34
Comando 144
Comandos implementados en la interfaz electrónica 146
Conceptos generales 27
CONCLUSIONES 168
Configuración del cable de conexión entre la computadora y la *Interfaz Electrónica* 164
Consideraciones de la operación del Sistema con la Interfase Electrónica 221
Consultas 188
Control de la interfaz 148, 151
Conversión de señal RS232-TTL 148
Convertidor de señal de niveles RS232 a TTL 57
Crear modelos de los objetos 34
Cuarto de Control 11
Cuarto de Control. 11

D

Dato 144, 150
Decodificadores 52
Demanda de los servicios del Laboratorio 16
Derechos por categoría de Operador 228
Desarrollo de la base de datos bajo SQL Server Express 61
DESARROLLO DEL SISTEMA 59
Descripción del Sofware 79
Descripción funcional sobre los servicios proporcionados por el Laboratorio Abierto de la Facultad de Ingeniería 14
Descripción general de bloques de módulos 149
Diagramas de flujo del software el Microcontrolador 153
Dirección 144, 149

E

Elección de componentes 138
Elección del puerto de comunicaciones 59
Encapsulados 56
Entidad tbDump 63
Entidad tbEspera 63
Entidad tbInventario 64
Entidad tbInvTemp 65
Entidad tbModulos 66
Entidad tbMovimientos 69
Entidad tbMovsModulos 70
Entidad tbPréstamos 71
Entidad tbSanciones 72
Entidad tbSeguridad 74
Entidad tbSysParms 75
Entidad tbSysTemp 75
Entidad tbTransacción 76
Entidad tbUsuarios 77
Equipo de cómputo y software de implementación 23
Estadísticas 189
Estructura de la solución 23

Estructura y composición del Laboratorio Abierto 10

F

Facultades de los administradores del Laboratorio 15
Formato de un byte 39
Fototransistor o lineal 55
Fototriac de paso por cero 55
Fuente de alimentación 149
Funcionamiento 55

H

Habilitación 150
Hardware 36

I

Icono en la barra de tareas de Windows 223
Iconos de los módulos del Panel de Control 198
Identificar las relaciones entre los objetos 35
Identificar los objetos 34
Identificar los tipos de información para cada objeto 34
Indicador de conexión de la *interfaz electrónica* 197
INDICE 238
ÍNDICE ALFABÉTICO 233
Integridad de dominio 31
Integridad de entidad 31
Integridad definida por el usuario 32
Integridad referencial 31
INTERFASE FÍSICA (ELÉCTRICA) 45
Interfaz electrónica 23, 48
INTERFAZ ELECTRÓNICA 138
INTRODUCCIÓN 8
Inventario 193

L

Laboratorio abierto de la Facultad de Ingeniería 11
Lenguaje de control de datos (DCL) 33
Lenguaje de definición de datos (DDL). 33
Lenguaje de manipulación de datos (DML) 33
Lenguajes de programación 28
Lista de materiales 161, 163
Listado del código fuente del programa del PIC16F84A 157

M

MANTENIMIENTO AL SISTEMA 173
Materiales complementarios 163
Menú Administración 191
Menú Consultas 188
Menú contextual de un módulo forzado no asignado 208
Menú contextual de un módulo forzado y ocupado 215
Menú contextual de un módulo libre 204

Menú contextual de un módulo ocupado 209
Menú contextual del área del Panel de Control donde no se encuentran los módulos 215
Menú de Opciones 187
Menú Estadísticas 189
Menú Inventario 193
Menú Usuarios....190
Menús contextuales del Panel de Control 203
Microcontroladores.... 56
Modelaje de las bases de datos 33
Modelo conceptual de la interfaz electrónica....147
Módulo activo 221
MODULO modFunciones 134
MODULO modInicio 134
MODULO modInventario 135
MODULO modModulos 135
MODULO modPanel 136
MODULO modSanciones 136
MODULO modSeguridad 136
MODULO modUsuarios 137
Módulos de trabajo 13
Módulos en el Laboratorio.... 24
Módulos utilizados 220

N

Niveles lógicos RS232 38

O

Objetivo general.... 5
OBJETIVOS 5
Objetivos específicos.... 5
Opción Acerca de....194
Opción Ayuda....193
Optoacopladores.... 54
Optotiristor 55
Optotriac 55

P

Panel de Control....198
Panel de la lista de espera 22
Panel de Lista de espera 219
Panel de materiales por entregar.... 22
Panel de Módulos próximos....217
Panel de módulos próximos a vencer el tiempo de servicio 22
Panel de préstamo de materiales....218
Panel de switcheo....12, 23
Paneles complementarios 217
PCB DE LA TARJETA DE BLOQUE 175
Pérdida de la comunicación con la Interfase Electrónica....221
Procedimiento para otorgar servicios a los usuarios 15
Protocolo de comunicación con la interfaz electrónica 143

Protocolo de conexión 23
Prototipo de prueba 165
Pruebas de implementación 180
Puerto paralelo 40
Puerto serial 37
Puerto USB 43

R

Recopilar la información 34
Registros (latches) 48
Reinicio 150
Reloj 197
Respuesta de la interfaz electrónica 146

S

Selector de bloque 148
Servicios bridados por el Laboratorio Abierto 15
Software 27
Software de desarrollo 24
Solución a la problemática presentada 21

T

Tarjeta de bloque 162
Tarjeta principal 160
TARJETA PRINCIPAL 174
Tarjetas de la interfaz electrónica 159
Tipos de Optoacopladores 54
TRANSMISIÓN ASINCRÓNICA 46
TRANSMISIÓN BULK 48
TRANSMISIÓN EN EL USB 46
TRANSMISIÓN ISOCRONICAS 47
TRANSMISIÓN SINCRÓNICA 47
TRANSMISIONES DE CONTROL 48
TRANSMISIONES DE INTERRUPCIÓN 48
TRIAC con acoplado óptico (opto coupler TRIAC) 54
Triacs 53

U

Usuarios 190
Usuarios en el Laboratorio 221

RELACIÓN DE CLASES DEL SAEALABFI 1.2.0000.0

| Clase/Módulo | clsGeneral | clsInterfase | clsModulo | clsParametros | clsSeguridad | clsTransacción | modFunciones | modInicio | modInventario | modModulos | modPanel | modSanciones | modSeguridad | modUsuarios | frmAcercaDe | frm_Administración_ActivarCuenta | frm_Administración_ArchivoDeTransacciones | frm_Administración_BloquearCuenta | frm_Administración_BorrarCuenta | frm_Administración_CambiarContraseña | frm_Administración_ContraseñaNueva | frm_Administración_CrearCuentaNueva | frm_Administración_Cuentas | frm_Administración_NombreNuevoDeUsuario | frmAdministraciónPanel | frm_Administración_UsuarioNuevoRápido | frmBienvenida | frmBox | frmConfiguraciónPorPrimeraVez | frmConfigurar | frmConsulta_ActualizaciónDatosXOperador | frmConsulta_ListaDeEspera | frmConsulta_MaterialesXUsuario | frmConsulta_MovsXMódulo | frmConsulta_MovsXOperadorMóduloYSemestre | frmConsulta_MovsXSemestre | frmConsulta_MovsXUsuario | frmConsulta_MovsXMóduloYSemestre | frmConsulta_PréstamosXSemestre | frmConsultaDeSancionesAplicadasPorOperadorDelSistema | frmConsulta_SancionesxUsuario | frmConsultaDeUsuariosCreadosPorOperadorDelSistema | frmDespedida | frmEspera | frmEstadística_MaterialesPrestadosXUsuario | frmEstadísticas_MaterialesPrestados | frmEstadísticasDeSancionesAplicadasPorUsuario | frmEstadísticas_ServicioPorUsuario | frmEstadísticas_ListaEspera | frmEstadísticas_UsoDeEquiposPorUsuarios | frmEstadísticas_UsoDeEquipos | frmInicioDeSesión | frmInventario_Edición | frmInventario_Módulos | frm_Inventario_ConsultaDeLosMovimientos | frmInventario_Acción | frmListadoDeUsuarios | frmMateriales_Entrega | frmMateriales_Préstamos | frmPregunta | frmPrincipal | frmSanciones | frmUsuarios_Actualizar | frmUsuarios_Agregar | frm_Usuarios_SelecciónDeUsuario | frmUsuarios_Confirmar |
|---|---|---