

# **UNIVERSIDAD LASALLISTA BENAVENTE**



## **ESCUELA DE INGENIERÍA EN COMPUTACIÓN**

Con estudios incorporados a la  
Universidad Nacional Autónoma de México  
CLAVE: 8793-16

**“SISTEMA DE ADMINISTRACIÓN DE METROLOGÍA”**

**TESIS**

PARA OBTENER EL TÍTULO DE:

**INGENIERO EN COMPUTACIÓN**

PRESENTA:

**JOSÉ EULOGIO GARCÍA CAMPOS**

**ASESOR: ING. ALEJANDRO GUZMÁN ZAZUETA**

**Celaya, Gto**

**Febrero de 2009**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## **Agradecimientos personales**

### **A Dios**

Por permitirme estar en este mundo. Y darme licencia de terminar mi carrera de ingeniería en computación, y actualmente le doy gracias porque tengo un trabajo que me gusta y que disfruto todos los días.

### **A mis padres y mis hermanos**

Por todo el apoyo que he recibido durante todos los años de mi vida. Por siempre estar conmigo en las buenas y en las malas, también les agradezco sus consejos ya que sin ellos tal vez yo no estaría ahora apunto de titularme. Gracias por quererme tanto son los mejores.

### **A mi esposa e hijo**

Porque siempre me apoyaron durante mis estudios, y fueron mi principal inspiración para terminar la carrera, los amo (Rosalba y Diego).

### **A mis maestros**

Por compartir con nosotros sus conocimientos desinteresadamente, Especialmente quiero agradecer al Ingeniero Alejandro Guzmán Zazueta primero por su apoyo en la realización de mi tesis y por compartir su vasto conocimiento con nosotros. También agradecer al ingeniero Anselmo Ramírez por su atención para con nosotros los estudiantes.

### **A mis Amigos**

Compañeros de carrera por el apoyo que nos brindamos durante la misma. A mis compañeros de trabajo porque gracias a ellos he aprendido mucho y gracias a ellos he realizado la documentación de este sistema que juntos llevamos a cabo. Gracias.

---

<b>Índice</b>	<b>Página</b>
Introducción.	
 <i>Capítulo I. Antecedentes</i>	
1.1 Antecedentes.	2
1.2 Empresa a la que se dirige.	4
1.2.1 Misión y visión.	4
1.2.2 Impacto tecnológico.	4
1.3 Operación y procesos del laboratorio de metrología.	5
1.4 Problemáticas en el laboratorio.	6
1.5 Solución propuesta para problemáticas.	7
 <i>Capítulo II. Desarrollo de sistema prototipo y ventajas de la plataforma .NET</i>	
2.1 Requisitos del sistema.	9
2.1.1 Software requerido para desarrollo de sistema.	9
2.1.2 Hardware requerido para desarrollo de sistema.	9
2.2 Esquematización del sistema prototipo.	10
2.3 Desarrollo de prototipo del sistema usando Visual Basic .NET.	11
2.4 Objetivos y ventajas de Visual Studio .NET.	15
2.4.1 Porqué usar Visual Basic .NET 2005.	15
2.4.2 Arquitectura del .NET Framework.	18
2.4.3 CLR – Common Language Runtime.	20
2.4.3.1 Modelo de ejecución del CLR.	23
2.4.4 .NET Framework Class Library.	24
2.4.5 Common Language Specification(CLS) y elección del lenguaje.	24
2.4.6 Ventajas de .NET.	25

**Capítulo III. Instalación de SQL Server 2005 e introducción al manejo de datos XML**

3.1	Introducción.	29
3.2	Ediciones de SQL Server 2005.	30
3.2.1	SQL Server 2005 Enterprise Edition (32 bits y 64 bits).	30
3.2.2	SQL Server 2005 Evaluation Edition (32 bits y 64 bits).	31
3.2.3	SQL Server 2005 Standard Edition (32 bits y 64 bits).	31
3.2.4	SQL Server 2005 Workgroup Edition (sólo 32 bits).	31
3.2.5	SQL Server 2005 Developer Edition (32 bits y 64 bits).	32
3.2.6	SQL Server 2005 Express Edition (sólo 32 bits).	32
3.2.7	SQL Server 2005 Compact Edition (sólo 32 bits).	33
3.2.8	SQL Server 2005 Runtime Edition (32 bits y 64 bits).	33
3.3	Instalación de SQL Server 2005.	34
3.3.1	Requisitos del sistema para la instalación.	34
3.4	Proceso de instalación de SQL Server 2005 Express Edition.	35
3.4.1	Instalar SQL Server 2005 Express Edition (Paso a Paso).	35
3.5	Proceso de instalación de SQL Server Management Studio Express.	42
3.5.1	Instalar SQL Server Management Studio Express.	42
3.5.2	¿Qué es SQL Server Management Studio?	46
3.6	Microsoft SQL Server y XML, Consultas con Xpath.	48
3.6.1	Introducción.	48
3.6.2	Nodos.	48
3.6.3	Relación entre nodos.	48
3.6.4	Selección de nodos.	49
3.6.5	Predicados.	51
3.6.6	Funciones.	53
3.6.7	Conclusión.	54
3.6.8	Apéndice: Variable XML.	55
3.7	Ejercicio de comparación de tiempo de respuesta usando consultas a datos XML.	56

**Capítulo IV. Funcionamiento del Sistema de administración de metrología**

4.1 Descripción de formas.	67
4.2 Autenticación.	67
4.3 Pantalla inicial del sistema.	68
4.4 Administrar Instrumentos.	70
4.4.1 Historial instrumento.	72
4.4.2 Nuevo.	74
4.4.3 Modificar.	75
4.4.4 Ver calibraciones.	76
4.4.5 Eliminar.	77
4.4.6 Ver informe.	77
4.4.7 Cerrar.	77
4.5 Calibraciones.	78
4.5.1 Ventana de calibraciones.	78
4.5.2 Imprimir reporte de cumplimiento de calibraciones.	79
4.5.3 Agregar.	80
4.5.4 Modificar.	81
4.5.5 Eliminar.	82
4.5.6 Cerrar.	82
4.6 Plan de calibraciones.	83
4.6.1 Imprimir plan de calibración.	84
4.6.2 Mover calibraciones y guardar cambios.	85
4.6.3 Ver calibraciones.	85
4.6.4 Regresar.	85
4.7 Informes de calibración.	86
4.7.1 Crear informe.	87
4.7.2 Ver mediciones.	94
4.7.3 Ver informe.	94

Conclusiones.

Bibliografía.

## Introducción

El presente trabajo de investigación es la culminación de la carrera de Ingeniería en Computación y la aplicación de los conocimientos de esta maravillosa carrera me ha instruido a que siempre y en todo momento tenemos la necesidad de actualizarnos en todo lo relacionado con las nuevas tecnologías ya que en la actualidad la tecnología crece a pasos descomunales.

De lo que ha surgido una aportación de los conocimientos y enseñanzas que me transmitieron los catedráticos de la carrera de Ingeniería en Computación, de esta manera expongo a la opinión del lector la siguiente investigación.

En el presente trabajo que lleva por título “Sistema de administración de metrología” ha sido documentado por un servidor José Eulogio García Campos a partir del sistema desarrollado para la empresa condomex cables de la ciudad de México y que lleva por nombre Sistema de administración de metrología mismo que aparece como título de esta investigación. Este sistema fue desarrollado por el equipo de desarrollo de software del CIDEC (centro de investigación y desarrollo carso) empresa para la cual laboro.

Hoy en día me doy cuenta que la sistematización y optimización de procesos me facilita muchas de las tareas que actualmente se hacen manualmente, tal y como, sistematizar inventarios, procesos de contabilidad, manejo de nóminas y en específico para este trabajo de investigación la administración de un laboratorio de metrología en el cual se requiere obtener información valiosa de forma fácil y rápida.

La presente tesis consta de cuatro capítulos en los cuales se trataron diversos puntos importantes desde un marco teórico del porqué se desarrolló el sistema de administración hasta el dar a conocer el funcionamiento del sistema y uso de herramientas auxiliares usadas.

En el primer capítulo nos involucramos en las actividades del laboratorio de metrología con el fin de conocer los procesos que se llevan a cabo dentro del mismo, para poder determinar los diferentes escenarios que se pueden presentar en los diferentes procesos existentes en el laboratorio.

En el segundo capítulo se creó un prototipo del sistema con el fin de que los usuarios que lo van a operar, prueben y aprueben el funcionamiento del futuro sistema. El prototipo será desarrollado en la plataforma .NET, con motivo del estudio de tesis se hará una breve pero importante introducción a la plataforma .NET.

En el tercer capítulo se explicará paso a paso la instalación y configuración de nuestro motor de base de datos, SQL Server 2005. Así como el uso del tipo de dato XML de SQL server 2005. Se tratará un ejercicio en el cual se podrá ver las ventajas de usar XQuerys y datos XML.

En el cuarto capítulo se dará a conocer el funcionamiento del sistema, del cómo operarlo, y de cómo sacar el mejor provecho del mismo.



# Capítulo I

## Antecedentes

## 1.1 Antecedentes<sup>1</sup>

Conductores mexicanos eléctricos y de telecomunicaciones S.A. de C.V., es una empresa líder en la fabricación de alambres, cables eléctricos y cables para telecomunicaciones, los cuales se comercializan exitosamente en los cinco continentes. Cometel se distingue por la insuperable calidad de sus productos que cuentan con la mejor tecnología de materiales, diseño y procesos de manufactura, únicos en México.

Cometel inició operaciones en 1954 y actualmente cuenta con cuatro plantas en el país.

- **Planta Potencia:** La cual inició operaciones en 1954 y se dedica al diseño y manufactura de cables eléctricos de aluminio y cobre utilizados en la generación, transmisión y distribución de energía eléctrica para alta, media y baja tensión, en corriente alterna y directa y que se usan en la industria, minas u hogares; así como el diseño y manufactura de cables control utilizados para la operación y protección de equipos y control de procesos.

Se ubica en Poniente 140 No. 720, Col. Industrial Vallejo, México D.F., C.P. 02300.

- **Planta Magneto.** Inicia operaciones en 1956, y se dedica a la fabricación de alambre magneto de cobre y aluminio; circulares, cuadrados y rectangulares; desnudos, esmaltados y forrados. También fabrica un amplio rango de calibres, construcciones y clases térmicas.

Se ubica en Poniente 140 No. 720, Col. Industrial Vallejo, México D.F., C.P. 02300.

- **Planta Conducon.** Inicia operaciones en 1971, y se dedica a la fabricación de cables de cobre con cubierta de PVC para baja tensión y compuestos de PVC flexibles, así como PVC para autoconsumo y consumo de otras plantas del Grupo (Nacel S.L.P. y Vallejo).

Se ubica en Calle 7 No. 872, Zona Industrial, Guadalajara, Jal., C.P. 44940.

---

<sup>1</sup> Información tomada de la página de Internet <http://www.condumex.com.mx/>

- **Planta Condutel.** Inicia operaciones en 1977, y se dedica a la fabricación de cables para telecomunicaciones, además de ser el único fabricante de cables de fibra en el país.

Se encuentra ubicada en el Km. 5, carretera San Juan del Río, Querétaro, C.P. 76800.

## **1.2 Empresa a la que se dirige**

La planta para la cual se desarrollará el sistema de administración de metrología será para la Planta Potencia de la ciudad de México. Esta será encargada de la operación del sistema ya que el laboratorio cuenta con los equipos y con las certificaciones necesarias para ser un laboratorio metrológico distinguido y reconocido oficialmente.

### **1.2.1 Misión y Visión**

Misión: Seguir siendo un laboratorio de metrología reconocido por la calidad en sus servicios que ofrece en la calibración de instrumentos de medición, tanto internos como externos.

Visión: Ser un laboratorio eficaz, que brinde seguridad y calidad en sus servicios.

### **1.2.2 Impacto Tecnológico**

Con la implementación del sistema de administración para el laboratorio de metrología de la planta cometel de la ciudad de México se pretende llevar una fácil y eficaz administración del laboratorio. Con el sistema implementado el jefe de aseguramiento de calidad podrá darse cuenta de la productividad del laboratorio, así como el laboratorista podrá llevar a cabo su trabajo de forma más organizada y de fácil acceso a la información que necesite.

### 1.3 Operación y Procesos del Laboratorio de Metrología

Los principales procesos que se llevan a cabo dentro del laboratorio son:

- **Calibración de instrumentos internos:** Esto es calibrar instrumentos de medición que pertenezcan a la planta ya sea de potencia o magneto y que para los cuales es necesario su calibración ya sea por reparación o porque se acerca su vigencia de calibración.
- **Calibración de Instrumentos externos:** Esto es calibrar instrumentos a clientes externos como su nombre lo dice, muchas veces clientes llegan al laboratorio de metrología y piden que se les calibre sus instrumentos y que se les expida un informe de calibración del instrumento calibrado, el cual funge como comprobante de calibración para fines legales.
- **Programar plan de calibraciones:** Esto es que el usuario planea las calibraciones, ya que muchos instrumentos requieren calibración por vigencia o también por reparación.
- **Generación de Informes de calibración:** Esto es cuando se calibra un instrumento se expide un certificado de calibración que lo ampara como instrumento vigente y que puede seguir siendo utilizado para sus fines para los cuales fue hecho.

### **1.4 Problemáticas en el Laboratorio de Metrología**

Una de las problemáticas dentro del laboratorio es que siendo tantos instrumentos dados de alta en el laboratorio se complicaba la organización de los mismos, es decir se da el caso en que a algunos instrumentos les tocaba calibración en un mes y no se llevaba a cabo sea por la razón que sea entonces, el equipo operaba sin haber sido calibrado tendiendo a ocasionar registrar posiblemente una mala medición o una medición no real aunque fuera por muy poco.

Otra de las problemáticas es que cuando se genera un informe calibración, este primeramente se crea en Excel para llevar a cabo las operaciones necesarias y la programación de fórmulas, entonces una vez terminado el informe en Excel éste es firmado por el jefe de aseguramiento de calidad, una vez firmado se imprime y se escanea para crear un archivo pdf, que se asocia a una calibración. En cierta forma se complica a veces rastrear algún informe de calibración de algún instrumento en específico.

La siguiente problemática es poder determinar de acuerdo a uso y mediciones registradas los instrumentos problema es decir instrumentos que ya no cumplen con los errores máximos permitidos. Y lo considero como problemática porque para poder determinar eso el laboratorista debería de hacer un conteo estadístico para ver el comportamiento de algún o algunos instrumentos y de acuerdo a los resultados estadísticos poder determinar si todavía cumplen con los errores máximos permitidos, que le apliquen de acuerdo al procedimiento o norma que los rijan.

## 1.5 Solución Propuesta Para Problemáticas

Soluciones propuestas para resolver estas problemáticas son primero que nada, crear un sistema el cual pueda ser capaz de programar con ayuda del laboratorista un plan de calibraciones es decir que el laboratorista pueda programar las calibraciones día con día, mes con mes y año con año, esto evitará que se pierda o se pase alguna calibración, esta funcionalidad sería primordial se pretende, que cada vez que se inicie el módulo de metrología este le diga al usuario si hay calibraciones programadas próximas ya sea calibraciones de la semana o del mes actual. Con esta funcionalidad el sistema incluso mandaría correos al jefe de aseguramiento de calidad para que esté enterado de las calibraciones que se deben de hacer día con día, y de igual forma enterado de la manera en que está trabajando el laboratorio.

Otra de las soluciones propuestas para solucionar la segunda problemática agilizar la generación de los informes de calibración el usuario capturará la información necesaria del informe, desde algunos datos generales como el error permitido o las condiciones ambientales a la cual se llevó a cabo la calibración así como los datos necesarios para llevar a cabo las operaciones requeridas para ese informe, las fórmulas estarán ya definidas en la programación y se generará el informe como archivo pdf. Con esto se evitará el escaneo de documento para convertir lo a pdf. Como reporteador se usará crystal report ya que por sus capacidades es el reporteador indicado.

La siguiente propuesta para solucionar la tercera problemática es generar reportes usando crystal reports en el cual el laboratorista podrá:

Consultar uno o varios instrumentos para ver su comportamiento a través de los resultados que ha arrojado durante sus calibraciones aplicadas. Con esto se podrá determinar si estos instrumentos aún funcionan de forma correcta. Ya que si no cumplen con la incertidumbre y errores máximos permitidos entonces deben de entrar a una revisión más a fondo.

# Capítulo II

## Desarrollo de Sistema Prototipo y Ventajas de la Plataforma .Net



## 2.1 Requisitos del Sistema.

Los requerimientos de software pueden variar, para el desarrollo del sistema se usará como IDE de programación Visual Studio .Net, este entorno de desarrollo es un paquete el cual consume muchos recursos por lo que para su uso adecuado al menos se requieren computadoras con capacidad de memoria RAM de 1Gb como mínimo, ya que como manejador de base de datos se usara SQLServer, este es un manejador muy robusto al cual también consume muchos recursos de la computadora.

### 2.1.1 Software Requerido Para Desarrollar Sistema

- Windows XP Profesional (Recomendado), Windows Server 2003, ó Windows Vista todas sus versiones.
- Internet Explorer 5.0, ó superior.
- Microsoft SQL Server 2005.
- Visual Studio.NET 2005, en cualquiera de sus versiones.

### 2.1.2 Hardware Requerido para desarrollar aplicaciones Visual Studio.NET.

- **Procesador:** superior a 1 GHZ, para este caso se utilizan computadoras procesador Centrino de 1.7 GHz
- **Memoria RAM:** como mínimo 1 GB
- Unidad de CD-ROM o DVD-ROM.
- Mouse.

## 2.2 Esquemmatización del Sistema Prototipo

El cliente quería ver un panorama más amplio de lo que el sistema sería capaz. Por lo que después de habernos involucrado en los procesos y operación del laboratorio se desarrolló un sistema prototipo el cual fue probado y aprobado tanto por el jefe de aseguramiento de calidad como por los laboratoristas. Este sistema prototipo fue el inicio de peticiones hacia nosotros, al ver que el sistema prototipo era capaz de agregar, consultar y borrar instrumentos así como agregar, actualizar y eliminar calibraciones, al cliente le pareció una buena base, así el cliente se dio cuenta de todas las necesidades, más tarde surgió la petición de que sería bueno agregar un apartado para los planes de calibración ya que es una de las problemáticas mas notadas.

## 2.3 Desarrollo del Sistema Prototipo usando .NET

Tanto para el desarrollo del sistema de administración de petrología como para su prototipo se utilizó Visual Studio .NET 2005 como plataforma de programación, Principalmente porque es una herramienta muy amigable y de fácil uso. A continuación se presentan las pantallas iniciales de lo que fue el sistema prototipo enfocado a dar a conocer lo que el sistema podría ser capaz de llevar cabo. Este se hizo a la par cuando se estaba conociendo las operaciones y los procesos en el laboratorio de metrología.



Fig. 2.1 Ventana de autenticación del sistema

Esta es la ventana de autenticación de usuario, en la cual el usuario se autentifica, introduciendo usuario y password. Al haber proporcionado la información requerida el sistema se encarga de Autenticar si el usuario existe y tiene los permisos para poder entrar al sistema.

Una vez autenticado el sistema prototipo muestra la siguiente ventana en la cual se pueden ver un menú de lado izquierdo con la opción de administrar instrumentos y calibraciones. (Ver Fig.2.2)



Fig. 2.2 Ventana inicial al ser autenticado

Aquí el usuario tiene la libertad de presionar cualquiera de las 2 opciones que se tenían en ese momento, administrar instrumentos y calibraciones. Veamos el comportamiento cuando el usuario presiona el botón de administración de instrumentos.

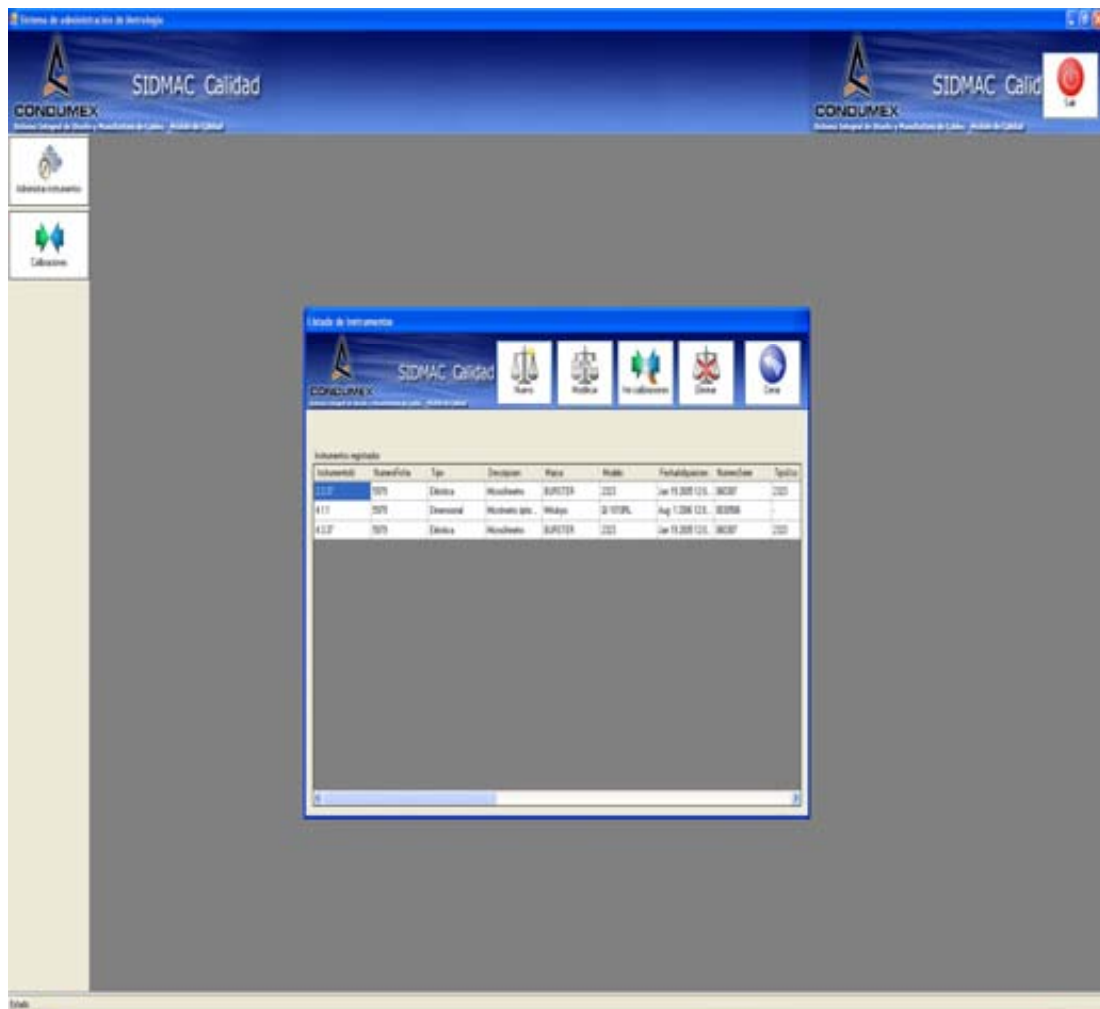


Fig. 2.3 Ventana al presionar la opción de administración de instrumentos

Aquí el usuario puede registrar nuevos instrumentos, modificarlos, ver calibraciones asociadas al instrumento y eliminar instrumentos. Así como ver las cartas de trazabilidad de algún instrumento en específico.

Ahora pasaremos a ver la ventana de calibraciones que se muestra cuando el usuario presiona calibraciones.

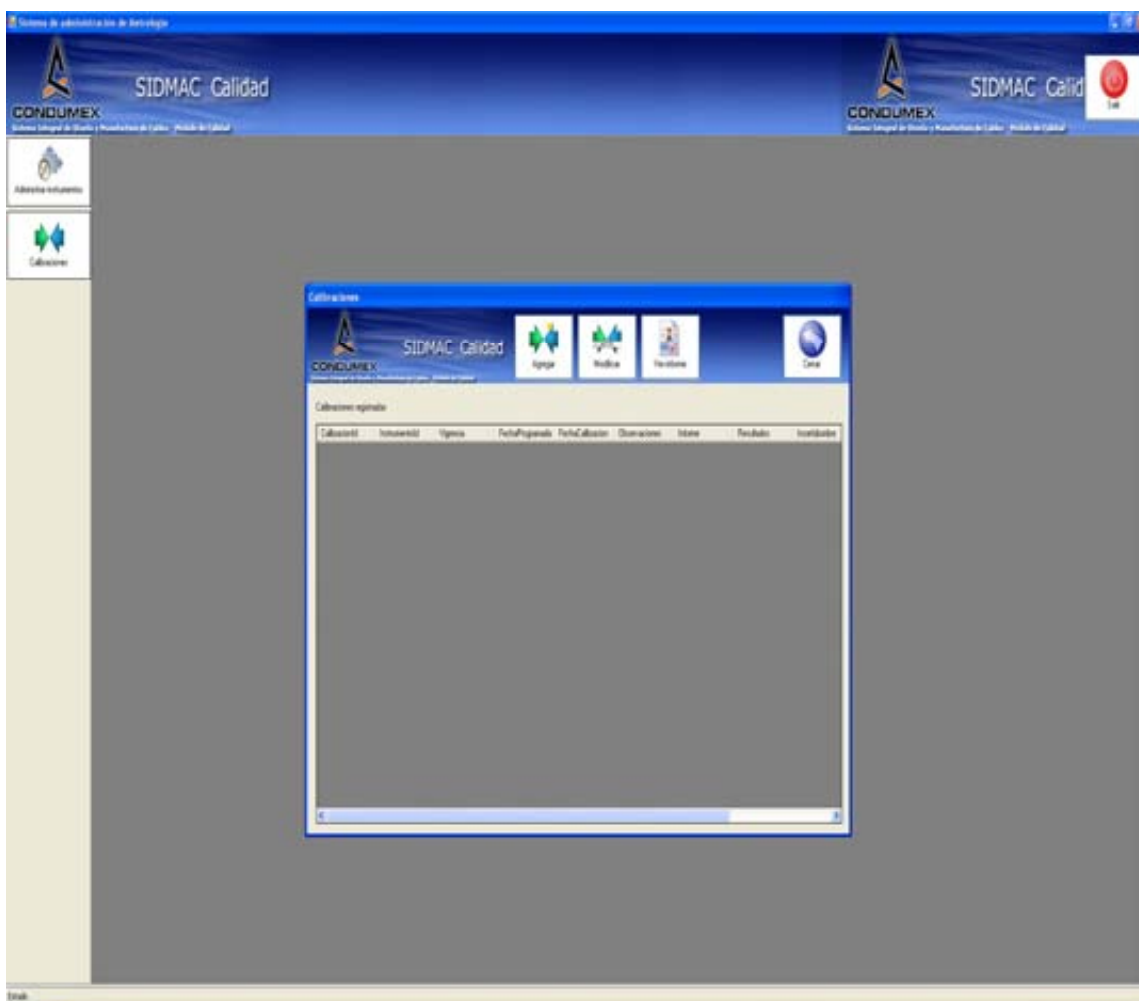


Fig. 2.4 Ventana cuando usuario presiona la opción de calibraciones.

Aquí el usuario podrá agregar y modificar calibraciones, así como la opción de ver informes de calibración de dichas calibraciones.

Inicialmente este fue el panorama que se presentó al jefe de aseguramiento de calidad y a los laboratoristas.

Como parte del objetivo de la creación del sistema prototipo fue primero que nada que a partir de nuestra poca experiencia en lo que eran los procesos que se llevan a cabo en el laboratorio de metrología pudiéramos crear una interfaz gráfica para el usuario y que a partir de esta tanto el usuario como nosotros desarrolladores nos diéramos cuenta de lo

capaz que podría ser el sistema. Una vez que el usuario conoció el sistema que, de alguna forma era un poco austera pero con la funcionalidad y dificultad de operación mínima se dio paso al desarrollo del sistema de administración de metrología, que en los capítulos más adelante se detallará. Por el momento cabe resumir que el crear un sistema prototipo ayuda a reafirmar y aclarar lo que el cliente necesita, así como también sirve para que el cliente se dé cuenta de lo que realmente necesita, entonces, una de las principales habilidades que tenemos que tener como desarrolladores y diseñadores de sistemas es presentarle al cliente ideas y/o soluciones a los problemas que se le complica resolver.

## **2.4 Objetivos y Ventajas de Visual Studio .NET 2005**

- Unifica los modelos de programación
- Simplifica aún más el desarrollo
- Provee un Entorno de Ejecución robusto y seguro
- Es independiente del lenguaje de programación
- Interoperabilidad con código existente
- Simplifica la instalación y administración de las aplicaciones
- Es Extensible

### **2.4.1 Porque usar Visual Studio .NET 2005**

Microsoft .NET es una plataforma de desarrollo y ejecución de aplicaciones. Esto quiere decir que no sólo nos brinda todas las herramientas y servicios que se necesitan para desarrollar modernas aplicaciones empresariales y de misión crítica, sino que también nos provee de mecanismos robustos, seguros y eficientes para asegurar que la ejecución de las mismas sea óptima. Los componentes principales de la plataforma .NET son:

- Un entorno de ejecución de aplicaciones, también llamado “Run time”, que es un componente de software cuya función es la de ejecutar las aplicaciones .NET e interactuar con el sistema operativo ofreciendo sus servicios y recursos.

- Un conjunto de bibliotecas de funcionalidades y controles reutilizables, con una enorme cantidad de componentes ya programados listos para ser consumidos por otras aplicaciones.
- Un conjunto de lenguajes de programación de alto nivel, junto con sus compiladores y linkers, que permitirán el desarrollo de aplicaciones sobre la plataforma .NET.
- Un conjunto de utilitarios y herramientas de desarrollo para simplificar las tareas más comunes del proceso de desarrollo de aplicaciones.
- Documentación y guías de arquitectura, que describen las mejores prácticas de diseño, organización, desarrollo, prueba e instalación de aplicaciones .NET.

Por otra parte, .NET representa la evolución COM (Component Object Model), la plataforma de desarrollo de Microsoft anterior a .NET y sobre la cual se basaba el desarrollo de aplicaciones Visual Basic 6 (entre otros tantos lenguajes y versiones).

A continuación se describirán algunas de las características principales de la plataforma .NET:

- Se dice que es una plataforma de ejecución intermedia, ya que las aplicaciones .NET no son ejecutadas directamente por el sistema operativo, como ocurre en el modelo tradicional de desarrollo. En su lugar, las aplicaciones .NET están diseñadas para ser ejecutadas contra un componente de software llamado Entorno de Ejecución (muchas veces también conocido como “Runtime”, o, “Máquina Virtual”). Este componente es el encargado de manejar el ciclo de vida de cualquier aplicación .NET, iniciándola, deteniéndola, interactuando con el Sistema Operativo y abasteciéndole servicios y recursos en tiempo de ejecución.
- La plataforma Microsoft .NET está completamente basada en el paradigma de Orientación a Objetos.



- .NET es multi-lenguaje: esto quiere decir que para poder codificar aplicaciones sobre esta plataforma no necesitamos aprender un único lenguaje específico de programación de alto nivel, sino que se puede elegir de una amplia lista de opciones.
- .NET es una plataforma que permite el desarrollo de aplicaciones empresariales de misión crítica, entendiéndose por esto que permite la creación y ejecución de aplicaciones de porte corporativo que sean críticas para la operación de tipos variados de organizaciones.
- .Net fue diseñado de manera tal de poder proveer un único modelo de programación, uniforme y consistente, para todo tipo de aplicaciones (ya sean de formularios Windows, de consola, aplicaciones Web, aplicaciones móviles, etc.) y para cualquier dispositivo de hardware (PC's, Pocket PC's, Teléfonos Celulares Inteligentes, también llamados "SmartPhones", Tablet PC's, etc.). Esto representa un gran cambio con respecto a las plataformas anteriores a .NET, las cuales tenían modelos de programación, bibliotecas, lenguajes y herramientas distintas según el tipo de aplicación y el dispositivo de hardware.
- Uno de los objetivos de diseño de .NET fue que tenga la posibilidad de interactuar e integrarse fácilmente con aplicaciones desarrolladas en plataformas anteriores, particularmente en COM, Ya que aún hoy existen una gran cantidad de aplicaciones desarrolladas sobre esa base.
- .NET no sólo se integra fácilmente con aplicaciones desarrolladas en otras plataformas Microsoft, sino también con aquellas desarrolladas en otras plataformas de software, sistemas operativos o lenguajes de programación. Para esto hace un uso extensivo de numerosos estándares globales que son de uso extensivo en la industria, y acerca de los cuales iremos aprendiendo a lo largo del curso. Algunos ejemplos de estos estándares son XML, HTTP, SOAP, WSDL y UDDI.

Por otra parte, .NET representa la evolución COM (Component Object Model), la plataforma de desarrollo de Microsoft anterior a .NET y sobre la cual se basaba el desarrollo de aplicaciones Visual Basic 6 (entre otros tantos lenguajes y versiones).

Por otro lado veremos una breve introducción al Framework .NET

El .NET Framework (traducido como “Marco de Trabajo”) es el componente fundamental de la plataforma Microsoft .NET, necesario tanto para poder desarrollar aplicaciones como para poder ejecutarlas luego en entornos de prueba o producción.

El .NET Framework tiene tres variantes principales, todas descargables gratuitamente desde Internet

- .NET Framework Redistributable Package: este es el mínimo componente de la plataforma .NET que se necesita para poder ejecutar aplicaciones. Normalmente ésta es la variante que se instala en los entornos productivos, una vez que el desarrollo y las pruebas de la aplicación han finalizado.

Está compuesto por:

- El entorno de ejecución de la plataforma .NET
- Las bibliotecas de funcionalidad reutilizable
- .NET Framework SDK: esta versión contiene herramientas de desarrollo de línea de comandos (compiladores, depuradores, etc.), documentación de referencia, ejemplos y manuales para desarrolladores de aplicaciones. Normalmente ésta variante se instala en los entornos de desarrollo de aplicaciones, y es más útil a los programadores que a los usuarios finales. Para poder instalar la versión SDK (Software Development Kit) es necesario instalar previamente el Redistributable Package.
- .NET Compact Framework: esta es una versión reducida del .NET Framework Redistributable, especialmente pensada para ser instalada en dispositivos móviles como Pocket PC's y SmartPhones.

## 2.4.2 Arquitectura del .NET Framework

En la figura se pueden apreciar las distintas partes que componen al .NET Framework, incluidas el entorno de ejecución de aplicaciones (CLR, en verde), el conjunto de bibliotecas de funcionalidad reutilizable (.NET Framework Class Library, en azul) y los

compiladores y herramientas de desarrollo para los lenguajes .NET (en rojo). Todos estos componentes se montan por encima de la familia de sistemas operativos Windows.

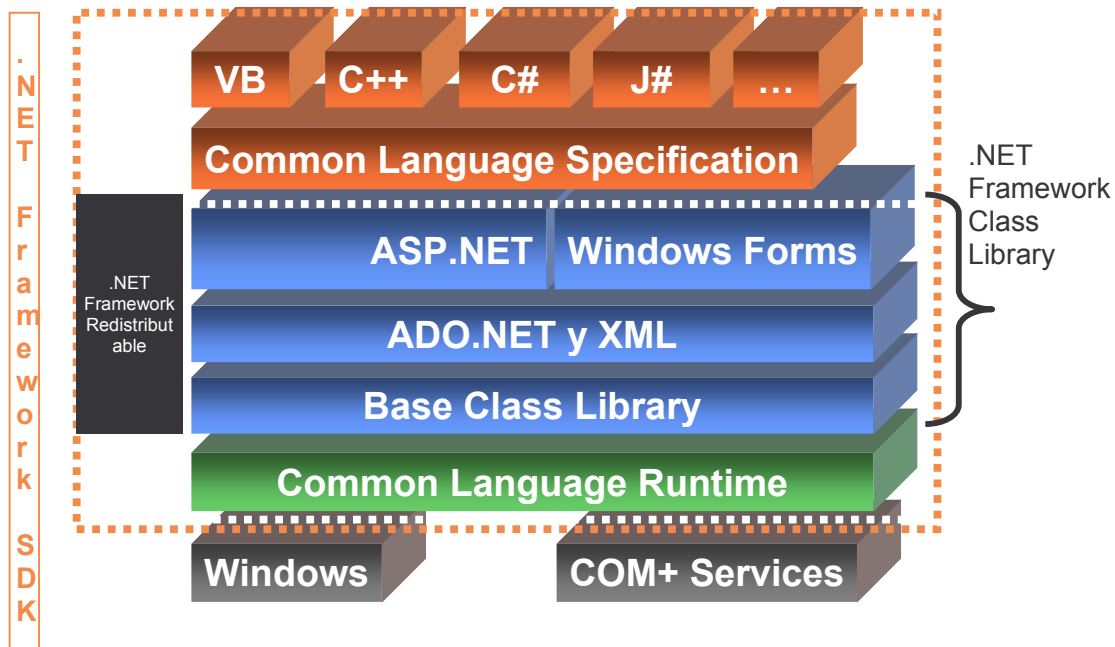


Fig. 2.5 Arquitectura del .NET Framework.

Dentro del conjunto de la .NET Framework Class Library se distinguen 4 sub-componentes principales:

- La Base Class Library (BCL - Biblioteca de Clases Base), que contiene la funcionalidad más comúnmente utilizada para el desarrollo de todo tipo de aplicaciones. Algunos ejemplos de la funcionalidad provista por la BCL son el manejo de colecciones, cadenas de texto, entrada/salida, threading, operaciones matemáticas y dibujos 2D.
- ADO.NET, que contiene un conjunto de clases que permiten interactuar con bases de datos relacionales y documentos XML como repositorios de información persistente.
- ASP.NET, que constituye la tecnología dentro del .NET Framework para construir aplicaciones con interfaz de usuario Web (es decir, aplicaciones cuya lógica se encuentra centralizada en uno o varios servidores y que los clientes pueden acceder

usando un browser o navegador mediante una serie de protocolos y estándares como HTTP y HTML).

- Windows Forms (o simplemente WinForms), que constituye la tecnología dentro del .NET Framework que permite crear aplicaciones con interfaz de usuario basada en formularios y ventanas Windows de funcionalidad rica y que se ejecutan directamente en los clientes.

### **2.4.3 CLR – Common Language Runtime**

Ahora conoceremos un poco del CLR (Common Language Runtime), que es el motor de ejecución runtime de .NET. Reconociendo algunos de sus principales servicios que les brinda a las aplicaciones que se ejecutan sobre él:

- **Compilación Just In Time (o Justo A Tiempo):** el CLR se encarga de compilar las aplicaciones .NET a código de máquina nativo para el sistema operativo y la plataforma de hardware en la que se está ejecutando. Esto lo hace sin intervención alguna del desarrollador o el usuario, y solamente a medida que se necesita.
- **Gestión Automática de Memoria:** el CLR abstrae a los desarrolladores de tener que pedir y liberar memoria explícitamente. Para esto, uno de sus componentes llamado Garbage Collector (Recolector de Basura) se encarga de liberar periódicamente la memoria que ya no está siendo usada por ninguna aplicación. Por otra parte, el CLR también abstrae a los desarrolladores del uso de punteros y del acceso a memoria de bajo nivel. Si bien estas características pueden ser consideradas poderosas, suelen hacer el desarrollo y mantenimiento de aplicaciones más propenso a errores y menos productivo.
- **Gestión de Errores Consistente:** como las aplicaciones .NET no se ejecutan directamente contra el Sistema Operativo, cualquier error no manejado que ocurra en tiempo de ejecución será atrapado por el CLR en última instancia, no afectando a ninguna otra aplicación que se esté ejecutando ni teniendo efecto alguno sobre su estabilidad.

- Ejecución Basada en Componentes: todas las aplicaciones .NET son empaquetadas en componentes reutilizables denominados genéricamente Assemblies, que el CLR se encarga de cargar en memoria y ejecutar.
- Gestión de Seguridad: el CLR provee una barrera más de contención a la hora de ejecutar aplicaciones manejadas, ya que permite establecer políticas de seguridad muy detalladas que las aplicaciones .NET que se ejecuten en una determinada computadora deberán cumplir.
- Multithreading: el CLR provee un entorno de ejecución multi-hilos por sobre las capacidades del Sistema Operativo, así como también mecanismos para asegurar su sincronización y acceso concurrente a recursos compartidos.

El desarrollo de una aplicación .NET comienza con la escritura de su código fuente en alguno de los lenguajes de alto nivel soportados por la plataforma. El mismo luego es compilado obteniéndose un ejecutable (que en Windows normalmente llevan la extensión .exe) o una biblioteca (que en Windows normalmente llevan la extensión .dll). A estos componentes .NET resultantes del proceso de compilación se los denomina genéricamente Assemblies, o Ensamblados.

Ahora bien, en lugar de contener código de máquina específico para el sistema operativo y el hardware en el cual fueron compilados (nativos), los assemblies contienen un código denominado MSIL (Microsoft Intermediate Language). EL MSIL es un set de instrucciones independientes de cualquier CPU existente y que puede ser convertido a código nativo muy eficientemente. MSIL incluye instrucciones para cargar, almacenar, inicializar e interactuar con objetos y sus atributos y métodos, así como también instrucciones aritméticas y lógicas, control de flujo, acceso directo a memoria, manejo de errores y otras operaciones. Antes de que el código MSIL pueda ser ejecutado debe convertirse a código nativo específico para un CPU y Sistema Operativo, tarea a cargo de los compiladores JIT incluidos en el CLR. Ver Figura 2.6:

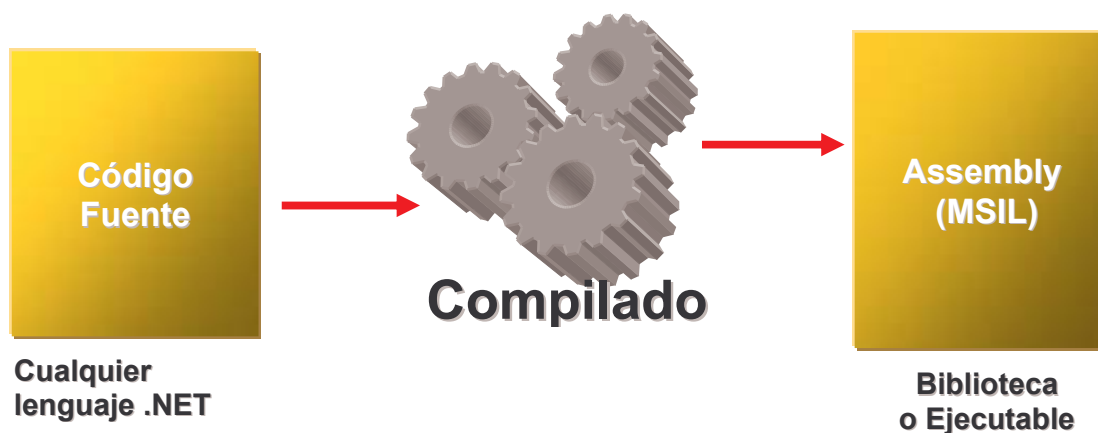


Fig. 2.6 Modelo de compilación y ejecución de aplicaciones .NET

Una aplicación .NET se compone, entonces, de uno o más assemblies. Otra de las características de los Assemblies es que no necesitan estar registrados en la Registry de Windows, como sus predecesores COM. De esta forma, instalar una aplicación .NET puede ser tan simple como copiar todos los assemblies necesarios a la computadora de destino, y basta con borrarlos a todos para tener una desinstalación limpia y completa.

Dado que .NET no depende de la Registry, y que cada assembly contiene información acerca de su versión y las versiones de los componentes de que depende, múltiples versiones de assemblies pueden coexistir sin ningún problema en la misma computadora.

Existen dos formas de que una aplicación pueda encontrar en tiempo de ejecución los assemblies de los que depende:

1) Ubicarlos en el mismo directorio. Esta es la opción preferida si esos assemblies sólo serán utilizados por esa única aplicación.

2) Ubicarlos en un repositorio centralizado de assemblies denominado Global Assembly Cache, en el cual se instalan todos los assemblies que serán utilizados por múltiples aplicaciones en la misma computadora. Para registrar un assembly en el GAC es necesario utilizar otra herramienta incluida en el SDK llamada gacutil.exe.

### 2.4.3.1 Modelo de Ejecución del CLR

La figura 2.6 representa el modelo de compilación y ejecución de aplicaciones .NET, al cual muchas veces se denomina “de compilación diferida”, o “de compilación en dos etapas”. Esto es así ya que el primer paso para poder ejecutar una aplicación dentro del CLR es compilar su código fuente para obtener un assembly con código MSIL. Este paso es realizado por cada uno de los compiladores de los distintos lenguajes de alto nivel soportados por .NET.

Luego, el CLR se encarga de compilar el código MSIL a código nativo que hace uso específico de los servicios del sistema operativo y la plataforma de hardware subyacente.

Todos los compiladores de los nuevos lenguajes .NET de Microsoft siguen este modelo de ejecución, con excepción de C++ .NET, que es el único lenguaje al que se le ha dejado la capacidad de emitir también código “no manejado”. Esto se debe a que ciertas aplicaciones, como los drivers de dispositivos, necesitan tener acceso a los recursos del sistema operativo a muy bajo nivel para lograr un rendimiento óptimo y mayor performance.

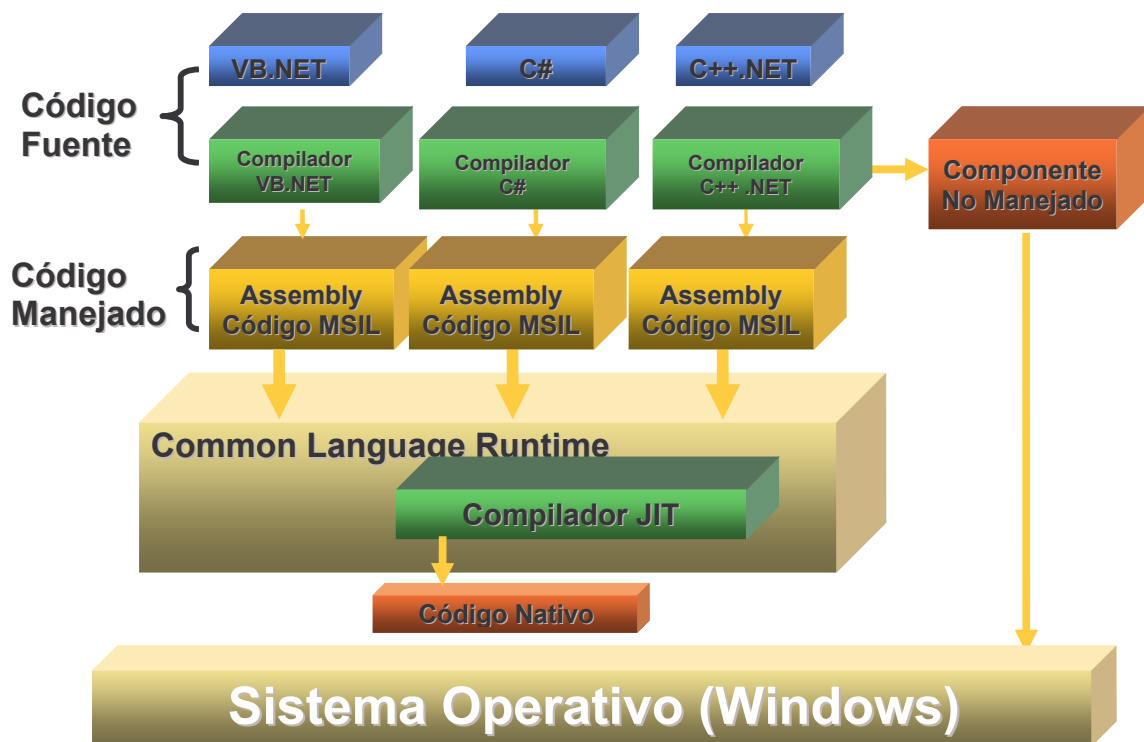


Fig. 2.6 Modelo de ejecución del CLR

#### **2.4.4 .NET Framework Class Library**

De muy poco serviría a los desarrolladores el contar con una plataforma de ejecución de aplicaciones tan sofisticada y robusta como el CLR sin tener además un conjunto de funcionalidades y componentes empaquetados listos para aprovechar y reutilizar en sus aplicaciones. Justamente ese es el propósito de la .NET Framework Class Library (Biblioteca de Clases del .NET Framework), que provee cientos de tipos básicos (clases e interfaces principalmente) orientados a objetos, extensibles mediante herencia, independientes del lenguaje de programación de alto nivel que se desee utilizar y organizados en namespaces jerárquicos, los Principales namespaces son:

- System: raíz de todos los otros namespaces, y dentro del cual podemos encontrar la mayoría de los namespaces correspondientes a la Base Class Library
- System.Data y System.Xml: en conjunto, estos dos namespaces constituyen la tecnología conocida como ADO.NET
- System.Web: dentro de este namespace se encuentran todos los tipos necesarios para programar aplicaciones y servicios Web ASP.NET
- System.Windows.Forms: dentro de este namespace se encuentran todos los tipos necesarios para programar aplicaciones de escritorio basadas en formularios y ventanas Windows.

#### **2.4.5 Common Language Specification (CLS) y elección del lenguaje**

Uno de los objetivos de diseño de la plataforma .NET fue el ser independiente del lenguaje de programación elegido para el desarrollo de aplicaciones. Para lograr esto es que se creó la Especificación de Lenguaje Común (o CLS, por sus siglas en inglés), que define y estandariza un subconjunto de todas las características soportadas por el CLR y que son necesarias en la mayoría de las aplicaciones. Todos los componentes desarrollados y compilados de acuerdo con la especificación CLS pueden interactuar entre si, independientemente del lenguaje de programación de alto nivel en el que fueron escritos.



Junto con el .NET Framework, Microsoft provee implementaciones de 4 lenguajes compatibles con CLS, junto con sus compiladores:

- Microsoft Visual Basic .NET
- Microsoft Visual C# .NET
- Microsoft Visual J#.NET
- Microsoft Visual C++.NET

Esto quiere decir que una aplicación escrita, por ejemplo, en Visual Basic.NET, puede incorporar sin problemas nuevas partes escritas en C# (Sharp) o C++ .NET.

Un punto importante a destacar es que la elección del lenguaje de alto nivel en el que debe escribirse una aplicación .NET prácticamente ha sido reducida a una cuestión de gustos personales y comodidad con la sintaxis. No hay prácticamente motivos tecnológicos sobresalientes que inclinen la balanza hacia algún lenguaje en particular, al menos entre los ofrecidos por Microsoft. Todos utilizan el mismo runtime, todos utilizan el mismo conjunto de bibliotecas de la misma forma, no existen diferencias notorias de performance entre ellos, todos tienen la misma potencia y todos tienen la misma capacidad de acceso a los recursos y servicios que expone el .NET Framework. De hecho, al cargar y ejecutar un assembly el CLR no sabe en qué lenguaje de programación de alto nivel éste fue escrito, ya que lo que el recibe como entrada es código MSIL.

#### **2.4.6 Ventajas de .NET**

Una vez dada una pequeña introducción a lo que es la plataforma .NET Resumiremos algunas de las ventajas que ofrece la plataforma Microsoft .NET:

- Unifica los modelos de programación
- Simplifica aún más el desarrollo
- Provee un Entorno de Ejecución robusto y seguro
- Es independiente del lenguaje de programación
- Interoperabilidad con código existente
- Simplifica la instalación y administración de las aplicaciones

- Es Extensible

Unificación de los modelos de programación, bibliotecas de funcionalidad y entornos de ejecución que existían anteriormente para distintos tipos de aplicaciones y distintos dispositivos.

Anteriormente a .NET existían lenguajes, bibliotecas, entornos de ejecución y herramientas de desarrollo distintas y específicas para cada tipo de aplicación y dispositivo (Visual Basic, Visual C++, ASP/VBScript, Embedded Visual C++, etc.).

.NET unifica todos esos modelos de programación ofreciendo una única API, un único entorno de ejecución, un único conjunto de bibliotecas y una única herramienta de desarrollo para cualquier tipo de aplicación.

Ofrece un modelo de desarrollo simplificado, basado en objetos que utilizan un sistema unificado de tipos de datos y se empaquetan en componentes reutilizables y auto-descriptivos (los assemblies).

Robusto entorno de ejecución (el CLR), que provee servicios a las aplicaciones en ejecución y maneja su ciclo de vida reforzando la

Seguridad y abstrayendo a los programadores de optimizaciones y manejos de memoria de bajo nivel.

Soporte a múltiples lenguajes de programación, lo cual acelera la curva de aprendizaje de los desarrolladores permitiendo que cada uno elija en base a sus gustos personales. Además, la posibilidad de utilizar las mismas herramientas de programación y tener las mismas capacidades de acceso a la plataforma independientemente del lenguaje le proporciona una flexibilidad sin precedentes.

.NET también simplifica, gracias al uso de assemblies auto-descriptivos, la instalación y administración de aplicaciones resolviendo gran parte de los problemas existentes en COM en lo que respecta la registración de componentes, manejo de múltiples versiones en paralelo y compatibilidad de aplicaciones.

Extensibilidad: Todas las clases incluidas en el .NET Framework son extensibles mediante los mecanismos de herencia propios de la orientación a objetos. Esto posibilita

que funcionalidades o controles gráficos que no cumplan exactamente con una determinada necesidad puedan ser extendidos para agregarle o modificarle comportamiento sin tener que escribir todo el código nuevamente.

También la plataforma .NET provee un altísimo grado de interoperabilidad con otras aplicaciones:

- Interoperabilidad entre aplicaciones .NET escritas en distintos lenguajes
- Interoperabilidad entre aplicaciones .NET y aplicaciones COM, mediante un módulo del CLR llamado COM-Interop. Esto permite reutilizar y aprovechar aplicaciones o componentes existentes desarrollados sobre la plataforma COM (por Ej. Visual Basic 6).
- Interoperabilidad entre aplicaciones .NET y múltiples tipos de aplicaciones desarrolladas sobre otras plataformas de software o hardware, incluso plataformas no Microsoft, mediante la tecnología de Servicios Web XML.

# Capítulo III

## Instalación de SQL Server 2005 e Introducción al Manejo de Datos Xml

### 3.1 SQL Server 2005 - Introducción

El cambio de SQL Server 2000 a **SQL Server 2005** tanto desde el punto de vista del programador, como desde la parte del administrador, da un giro de 180 grados. Acciones que se podían realizar desde la herramienta gráfica de SQL Server 2000, se han distribuido en dos grandes partes: El motor relacional de la base de datos, como herramienta para administrar una base de datos, y la herramienta Business Intelligent, como herramienta para desarrollar todo tipo de componentes, como puedan ser un proyecto de informes, uno de análisis de datos, incluso la importación y exportación de datos.

Muchas compañías se han visto desbordadas, debido a que la migración de algunos objetos era un trabajo muy tedioso, y muchos proyectos por falta de tiempo se seguían desarrollando contra SQL Server 2000. Una vez migrada la base de datos al nuevo nivel de compatibilidad, la vuelta atrás se antoja bastante compleja.

A favor de SQL Server 2005, muchísimas nuevas características que veremos a lo largo de esta investigación, para los que han dado el paso en su momento, están disfrutando de todas las características que nos ofrece esta nueva versión.

En definitiva este producto proporciona una solución integrada de gestión de la información que ayudará a empresas de cualquier dimensión a:

#### **-Disponibilidad:**

Las mejoras en la copia y restauración de datos, las nuevas opciones de replicación de datos, así como las nuevas tecnologías de alta disponibilidad que nos ofrece SQL Server 2005, permitirá a las empresas registrar acuerdos con sus clientes de una gran fiabilidad.

#### **-Escalabilidad:**

Las limitaciones de hardware y software con las cuales se encontraba SQL Server, han sido mejoradas con la inclusión de tres aspectos importantísimos:

\* El particionado de tablas ofrece un óptimo rendimiento, se puede ver con mayor profundidad en el módulo de optimización. Profundidad en el módulo de optimización.

\* Soporte de 64-bit, han abierto las puertas de empresas que demandaban un soporte de base de datos de mayor envergadura.

\* El particionado de índices, así como la incorporación de nuevos tipos, mejoran sensiblemente el resultado de las consultas.

#### **-Seguridad:**

Se han incluido ajustes de configuración que mejoran claramente los modelos de seguridad de versiones anteriores. En el módulo de administración y seguridad podremos ver más en profundidad a las características que nos referimos.

## **3.2 Ediciones de SQL Server 2005**

Las diferentes ediciones de SQL Server 2005 satisfacen los requisitos de rendimiento, tiempo de ejecución y precio únicos de organizaciones y personas.

A continuación daremos un vistazo a las características soportadas por las diferentes ediciones del SQL Server 2005

### **3.2.1 SQL Server 2005 Enterprise Edition (32 bits y 64 bits)**

Enterprise Edition es ampliable a los niveles de rendimiento necesarios para admitir:

- El procesamiento de transacciones en línea (**OLTP**) de las empresas más grandes.
- El análisis de datos de alta complejidad.
- Sistemas de almacenamiento de datos y sitios web.
- Las completas capacidades de análisis y Business Intelligence, y sus características de alta disponibilidad.

En definitiva **Enterprise Edition** es la edición más completa de SQL Server y es idónea para las organizaciones más grandes y para los requisitos más complejos.

### 3.2.2 SQL Server 2005 Evaluation Edition (32 bits y 64 bits)

SQL Server **Evaluation Edition** admite el mismo conjunto de características que SQL Server 2005 Enterprise Edition, pero únicamente puede usarse durante 180 días. Más adelante, si la empresa está interesada en utilizarlo como herramienta en el entorno de producción, Microsoft nos da la posibilidad de actualizar SQL Server Evaluation Edition para usarlo en producción, pasando a ser una licencia de la edición Enterprise Edition.

### 3.2.3 SQL Server 2005 Standard Edition (32 bits y 64 bits)

SQL Server 2005 **Standard Edition** es la plataforma de análisis y administración de datos para organizaciones pequeñas y medianas. Incluye la funcionalidad básica necesaria para soluciones de línea de negocio, almacenamiento de datos y comercio electrónico. Las características de alta disponibilidad y de Business Intelligence integradas de Standard Edition proporcionan a las organizaciones las capacidades básicas que necesitan para realizar sus operaciones. SQL Server 2005 Standard Edition es idóneo para aquella organización pequeña o mediana que necesite una plataforma completa de análisis y administración de datos.

### 3.2.4 SQL Server 2005 Workgroup Edition (sólo 32 bits)

SQL Server 2005 **Workgroup Edition** es la solución de administración de datos para pequeñas organizaciones que necesitan una base de datos sin límites de tamaño o número de usuarios. SQL Server 2005 Workgroup Edition se puede utilizar como servidor Web cliente o para operaciones de sucursales o departamentos. Incluye las características principales de base de datos de la línea de productos de SQL Server y es fácilmente actualizable a SQL Server 2005 Standard Edition o SQL Server 2005 Enterprise Edition. SQL Server 2005 Workgroup Edition es una base de datos idónea para comenzar que resulta confiable, sólida y fácil de administrar.

### 3.2.5 SQL Server 2005 Developer Edition (32 bits y 64 bits)

SQL Server 2005 **Developer Edition** permite a los programadores crear cualquier tipo de aplicación sobre SQL Server. Incluye toda la funcionalidad de SQL Server 2005 Enterprise Edition, pero su licencia permite utilizarlo como sistema de desarrollo y prueba, no como un servidor de producción.

SQL SERVER 2005 DEVELOPER EDITION

Es una opción ideal para:

- Fabricantes independientes de software
- Asesores.
- Integradores de sistemas.
- Proveedores de soluciones.
- Programadores corporativos que crean aplicaciones.
- Testeadores de aplicaciones.

Microsoft te da la posibilidad de actualizar SQL Server 2005 Developer Edition para utilizarlo en producción.

### 3.2.6 SQL Server 2005 Express Edition (sólo 32 bits)

**SQL Server Express** es gratuito. Esta edición puede redistribuirse (mediante un acuerdo); funciona como la base de datos de cliente, además de como una base de datos de servidor básica. SQL Server Express es la opción idónea para:

- Fabricantes independientes de software.
- Usuarios de servidor.
- Programadores no profesionales.
- Programadores de aplicaciones Web.
- Alojamiento de sitios Web.



Si se necesitan características de base de datos más avanzadas, SQL Server Express se puede actualizar sin problemas a versiones más sofisticadas de SQLServer, SQL Server Express también ofrece componentes adicionales que están disponibles como parte de SQL Server 2005 Express Edition with Advanced Services (SQL Server Express). Además de las características de SQL Server Express, SQL Server Express with Advanced Services ofrece lo siguiente:

- SQL Server Management Studio Express (SSMSE), un subconjunto de SQL Server Management Studio.
- Compatibilidad con catálogos de texto.
- Compatibilidad para ver informes mediante Reporting Services.

### **3.2.7 SQL Server 2005 Compact Edition (sólo 32 bits)**

Microsoft SQL Server 2005 **Compact Edition** es la base de datos compacta que amplía la funcionalidad de administración de datos corporativos a dispositivos.

Microsoft SQL Server 2005 Compact Edition puede replicar datos con SQL Server 2005 y SQL Server 2000, lo que permite a los usuarios mantener un almacén de datos móvil que se sincroniza con la base de datos primaria.

Microsoft SQL Server 2005 Compact Edition es la única edición de SQL Server que proporciona funcionalidad de administración de bases de datos relacionales para dispositivos inteligentes.

### **3.2.8 SQL Server 2005 Runtime Edition (32 bits y 64 bits)**

SQL Server 2005 **Runtime Edition** se ofrece a través del programa Microsoft ISV Royalty Program (programa de fidelidad para ISV). Según el Contrato de licencia para el usuario final de SQL Server 2005 Runtime Edition, un proveedor independiente de software (ISV) puede incrustar código de SQL Server en su solución siempre que el cliente no utilice el código de SQL Server para ejecutar otras aplicaciones ni en ningún otro contexto.

**Nota:** Todas las ediciones de SQL Server 2005 pueden usarse en entornos de producción, excepto SQL Server 2005 Evaluation Edition y SQL Server 2005 Developer Edition.

### 3.3 Instalación de SQL Server 2005

En este capítulo conoceremos los requisitos de software y hardware para poder instalar la versión **SQL Server 2005 Express Edition**. Realizaremos una instalación completa de **SQL Server 2005 Express Edition** y de su herramienta cliente **SQL Server Management Studio Express**.

Preparando la instalación de SQL Server 2005 Express Edition

#### 3.3.2 Requisitos del sistema para la instalación:

- **Sistemas operativos compatibles:** Windows 2000 Service Pack 4; Windows Server 2003 Service Pack 1; Windows XP Service Pack 2
- Equipo con procesador Intel Pentium III (o compatible) a 600 MHz o superior (se recomienda 1 GHz o más).
- 192 MB de RAM como mínimo (se recomiendan 512 MB o más).
- 525 MB de espacio disponible en disco duro
- Framework 2.0 (Descargar de la pagina de Microsoft)
- Descargar e instalar Windows Installer 3.1

He aquí los 4 archivos necesarios que para la correcta instalación de nuestro manejador de base de datos SQL Server 2005 Express Edition

- Windows Installer 3.1
- Framework 2.0
- SQL Server Express Edition
- SQL Server Management Studio

Si los hemos bajado a una misma carpeta, que es lo recomendable, nos quedaría como muestra la siguiente Fig. 3.1:

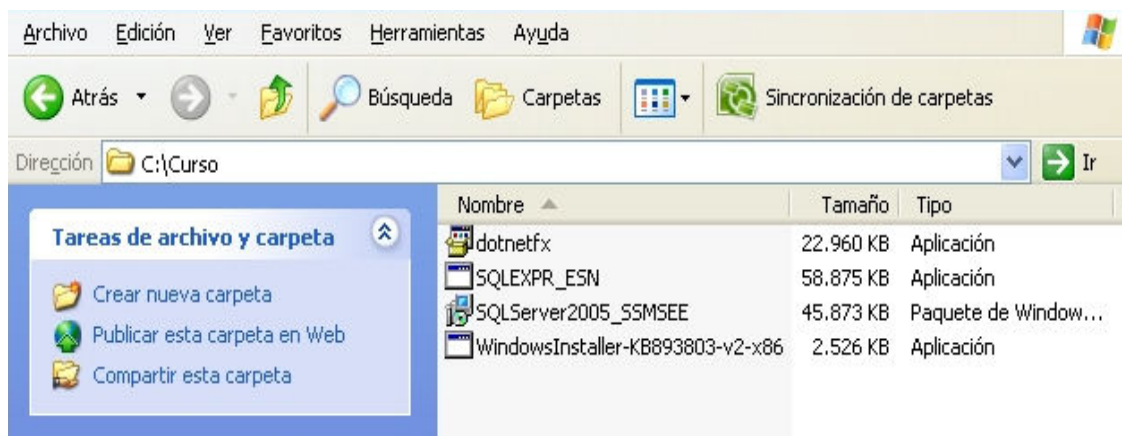


Fig. 3.1 Archivos necesarios para instalación de SQL server 2005

La instalación de Windows installer y framework 2.0 son necesarios para la instalación de esta versión, en versiones superiores vienen incluidos en la instalación, y por lo tanto no son necesarios descargar los ejecutables.

### 3.4 Proceso de instalación de SQL Server 2005 Express Edition

#### 3.4.1 Instalar SQL Server 2005 Express Edition

Para instalar SQL Server 2005 Express Edition hay que situarse en la carpeta donde tenemos nuestros ejecutables.

Ejecutamos el archivo **SQLEXPRESS.exe**

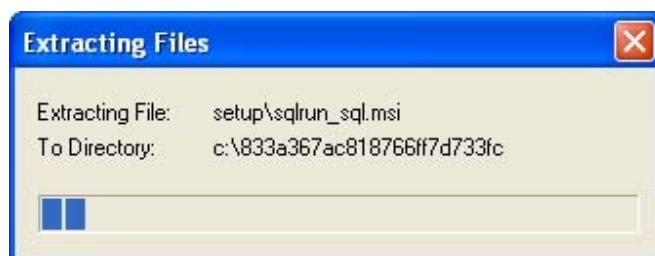


Fig. 3.2 Extracción de archivos.

De una forma o de otra comenzará la instalación, a continuación vamos a detallar gráficamente paso a paso la instalación de SQL Server 2005 Express.

Comenzamos aceptando la licencia una vez leída las condiciones.

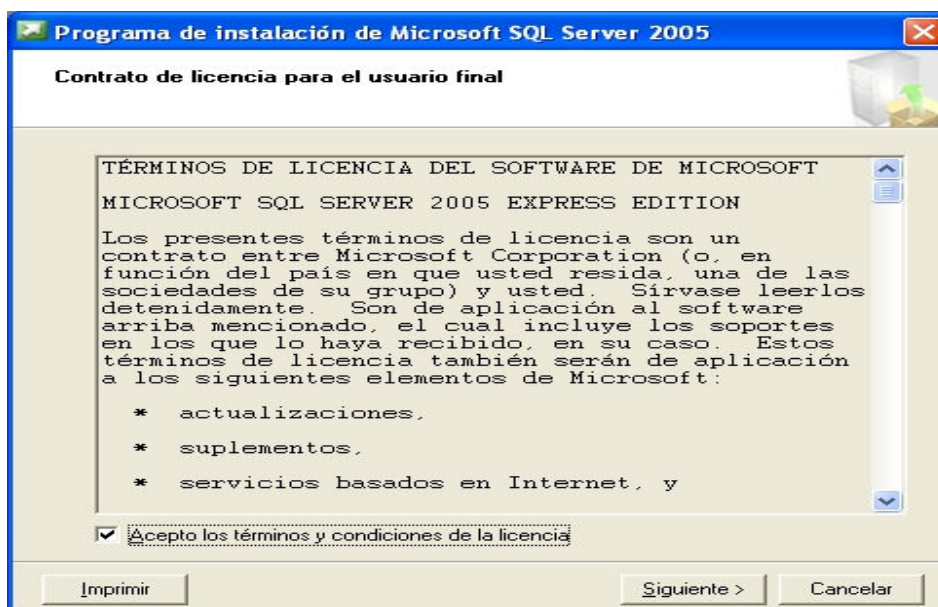


Fig. 3.3 Contrato de licencia

Ahora instalamos los requisitos previos para poder continuar con la instalación (Fig.3.4).



Fig. 3.4 Instalación de requisitos previos

La instalación en estos momentos comprueba los requisitos del sistema.

Si cumple el sistema con los requisitos solicitados, se lanza el asistente de instalación. El programa de instalación se asegura de que no se va a producir ningún error, estas comprobaciones puede llevar algunos minutos. Fig. 3.5

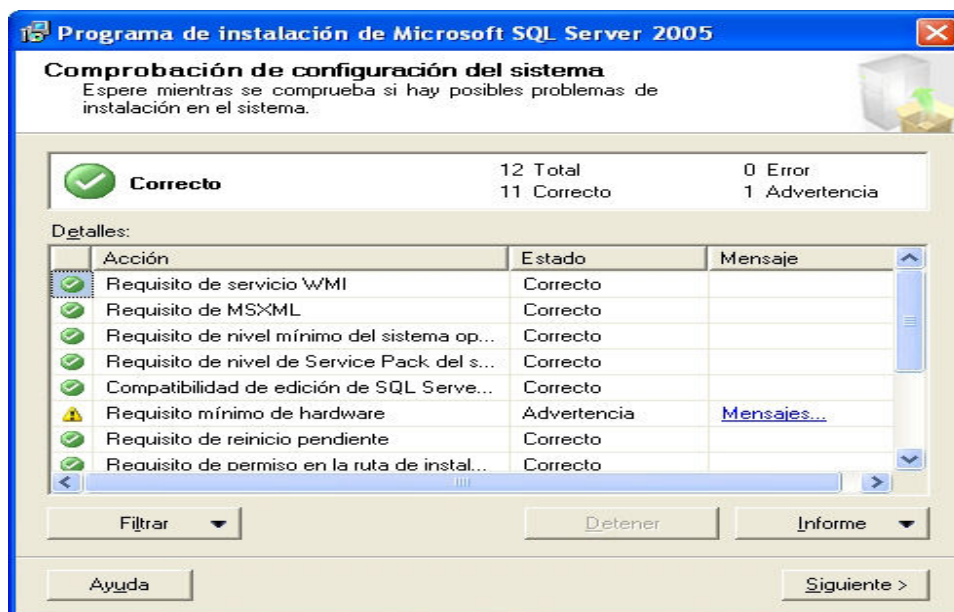


Fig. 3.5 Comprobación de configuración del sistema

Si aparece algún error habrá que ver la causa del mismo, dando clic sobre el vínculo que aparece a la derecha de la acción. También puede aparecer una advertencia, en este caso puede continuar la instalación, aunque es conveniente también comprobar el mensaje del mismo. Si todo ha ido bien, pulsamos el botón siguiente, y entonces comienza la instalación. Fig. 3.6

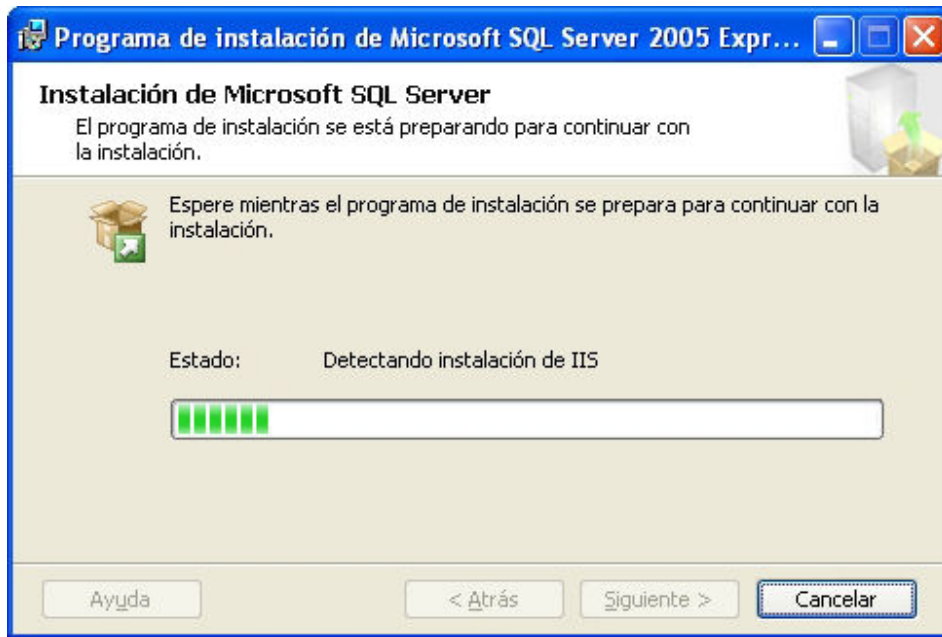


Fig. 3.6 Ventana de progreso de instalación de SQL Server

Registramos la aplicación con un nombre identificativo. Fig. 3.7

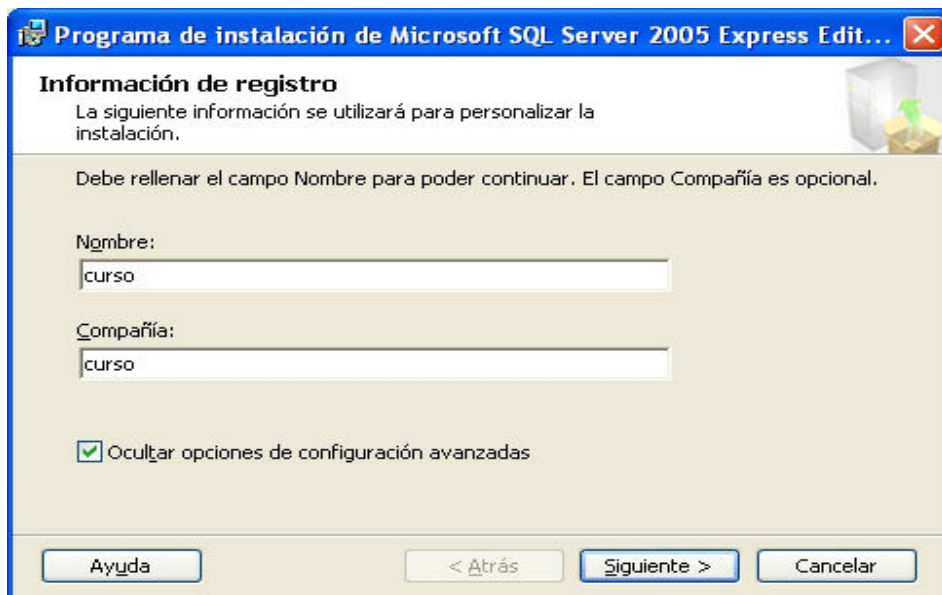


Fig. 3.7 Personalizar la instalación

También es muy importante seleccionar en que unidad y ruta vamos a instalar cada componente, cuando sólo tenemos una unidad no hay duda, pero si por ejemplo tenemos la

unidad C: bajo mínimos, y tenemos una unidad d: disponible con espacio suficiente, éste es el momento en el cual podemos modificar la ruta por defecto. Como veremos en la imagen, por defecto aparecen algunos componentes no incluidos, nosotros debemos seleccionar todas las características que SQL Server nos ofrece. Fig. 3.8

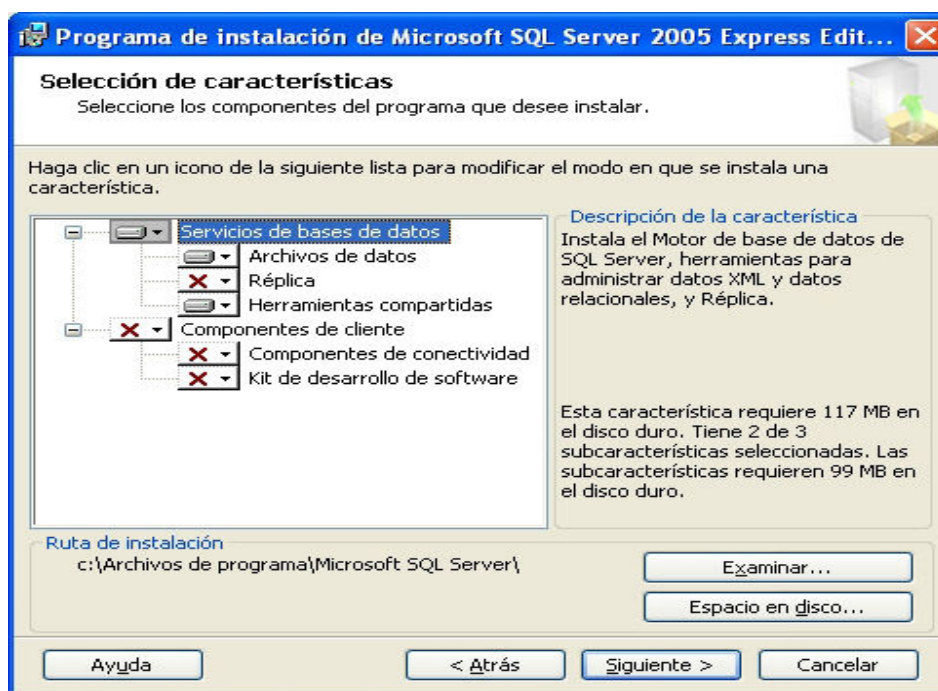


Fig. 3.8 Selección de características.

A continuación seleccionamos qué tipo de autenticación (Fig. 3.9) utilizaremos para conectar a la base de datos, generalmente se elige la autenticación mixta, ya que nos da la posibilidad de acceder a través de usuarios del sistema Windows, o bien a través de usuarios propios de SQL Server. Lógicamente, si seleccionásemos autenticación de Windows, la palabra de paso de acceso sería la misma del sistema. Como en este caso seleccionamos la autenticación mixta, tendremos que introducir una clave para el usuario administrador de la base de datos, llamado "sa". Para más información acceder al módulo de administración y seguridad.

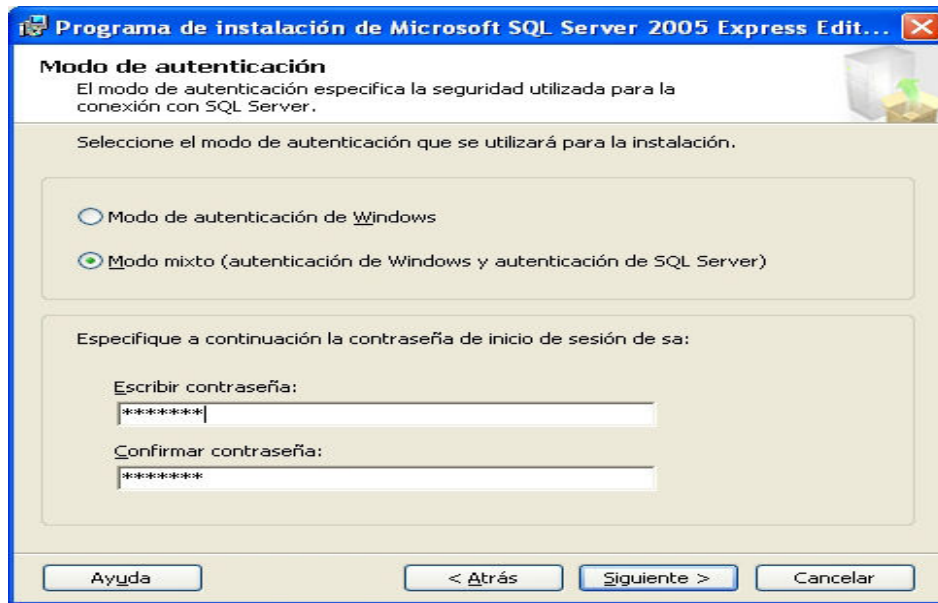


Fig. 3.9 Modo de Autenticación.

El siguiente paso nos anunciará que todos los requisitos previos han superado la prueba, y por tanto, que comenzará la instalación (Fig. 3.10).

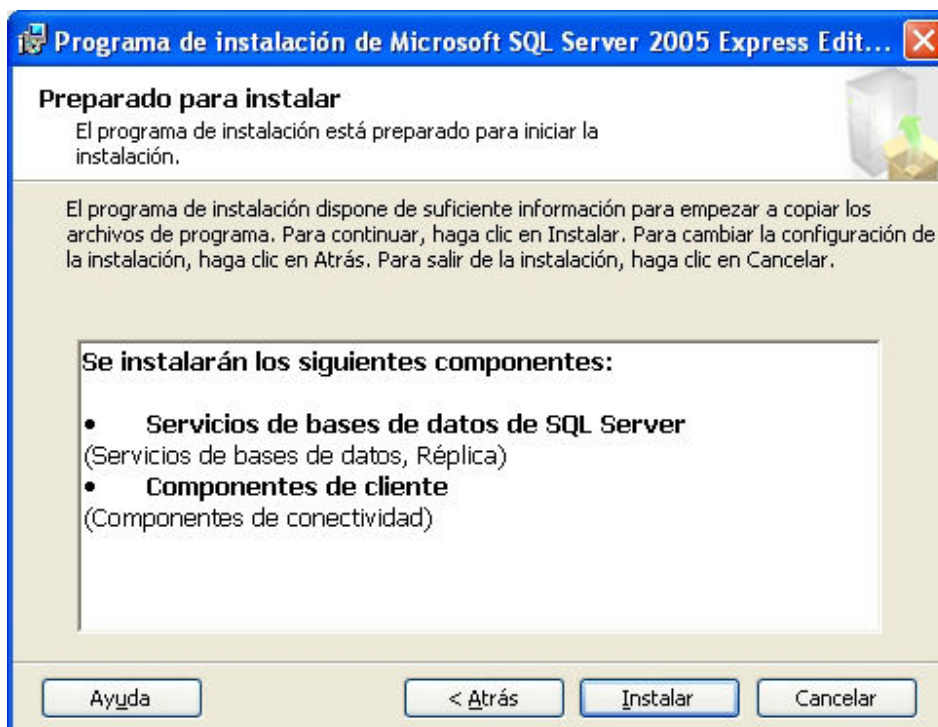


Fig. 3.10 Listo para instalar.



Comienza la instalación de todos los componentes. Si todo ha ido bien, aparecerá la siguiente imagen (Fig. 3.11), en caso contrario habrá que estudiar el informe que arroja la instalación.



Fig. 3.11 Instalación de componentes.

Una vez finalizada la instalación, el asistente nos deja un resumen de lo sucedido en todo este proceso. Recomendable repasar dicho informe por si viésemos algún aspecto de la instalación que no nos haya convencido. (Fig. 3.12)



Fig. 3.12 Resumen de instalación

Para comprobar que todo ha ido bien, pulsamos el botón de Windows, seleccionar todos los programas, y finalmente identificar un nuevo ítem llamado “SQL Server 2005”.

### 3.5 Proceso de instalación de SQL Server Management Studio Express

#### 3.5.1 Instalar SQL Server Management Studio Express

Para instalar SQL Server Management Studio Express hay que situarse en la carpeta donde hemos bajado nuestros ejecutables. (Fig.3.13)

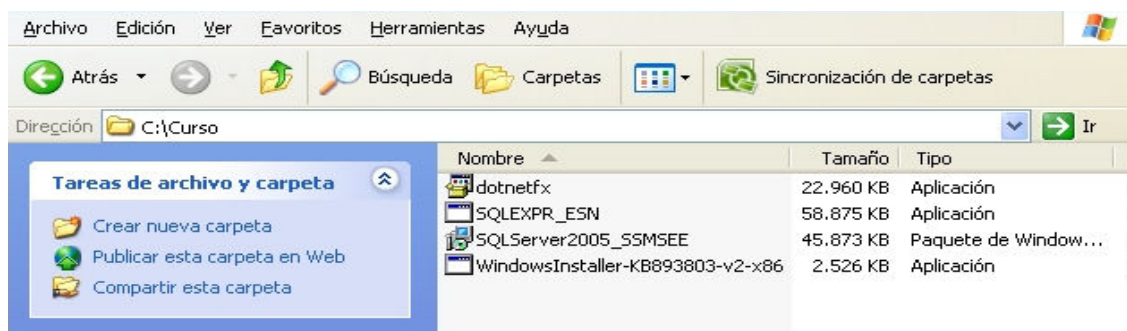


Fig. 3.13 Ventana de archivos para ejecutar.

Ejecutamos el fichero **SQLServer2005\_SSMSEE.exe**, se lanzará el asistente que nos guiará por la instalación. Fig.3.14



Fig. 3.14 Asistente de instalación, para SQL server management express.

Comenzamos aceptando la licencia una vez leída las condiciones (Fig. 3.15).

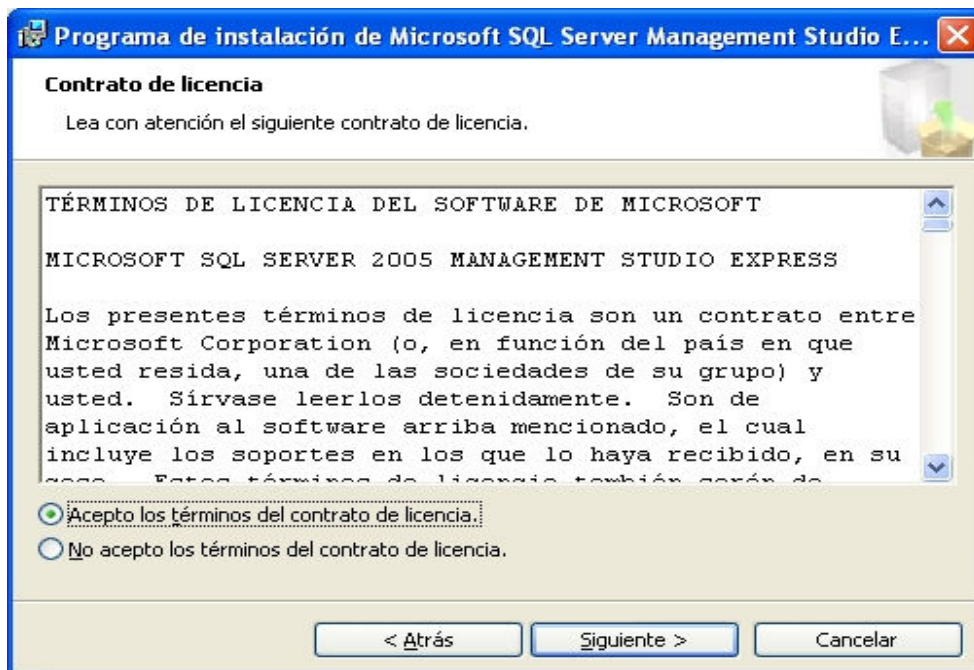


Fig. 3.15 Asistente de instalación

Personalizamos la instalación (Fig. 3.16).



Fig. 3.16 Personalización de Instalación.

Seleccionamos las características, en este caso la única que tenemos, la herramienta SSMS. Importante comprobar si tenemos espacio suficiente en disco (Fig. 3.17).

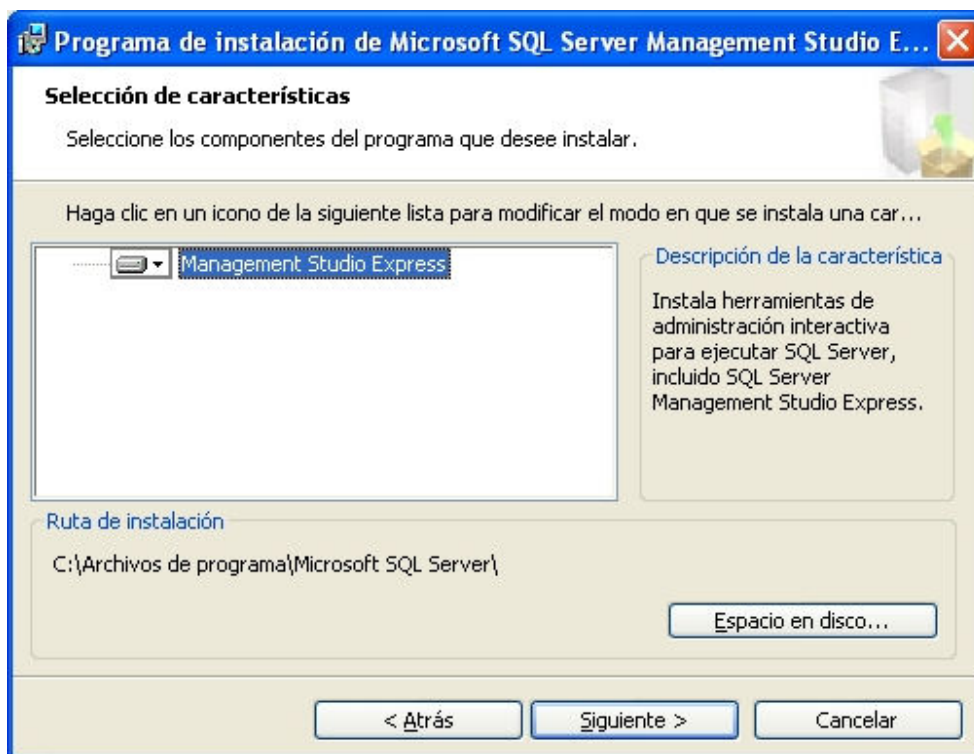


Fig. 3.17 Selección de características.

Ya está preparado el programa para comenzar la instalación. Basta con dar clic en Instalar para comenzar la instalación (Fig. 3.18).



Fig. 3.18 Preparado para la instalación.

Comienza la instalación. Completado el programa de instalación (Fig. 3.19).



Fig. 3.19 Instalación finalizada

Comprobamos que todo ha ido bien, dando clic en inicio de Windows, seleccionamos todos los programas, y por último elegimos la opción “Microsoft SQL Server 2005”, vemos en la nueva herramienta instalada SQL Server Management Studio Express.

Ejecutamos la herramienta, e intentamos conectar, por defecto podremos acceder como usuario de Windows, ya que en teoría nuestro usuario es administrador de nuestro propio PC, pero también tenemos la posibilidad de conectar al motor de base de datos, con la autenticación de SQL, con el usuario “sa” y la palabra de paso que indicamos durante la instalación. Ahora bien una vez conectados con nuestro servidor de base de datos, podemos dar de alta nuestras bases de datos. Mediante el SQL Server Management Studio

### 3.5.2 ¿Que es SQL Server Management Studio?

**SQL Server Management Studio** es una herramienta nueva para los administradores de bases de datos y programadores de SQL Server. Integrado en Microsoft Visual Studio, ofrece herramientas gráficas para la administración de bases de datos, junto con un completo entorno de desarrollo.

Management Studio incluye en una sola herramienta las funciones Administrador corporativo, Analysis Manager y Analizador de consultas SQL de SQL Server 2000, junto con la posibilidad de escribir instrucciones MDX, XMLA y XML.

- **MDX** (Expresiones multidimensionales): Es el lenguaje de consulta que se utiliza para trabajar con datos multidimensionales y para recuperarlos en Microsoft SQL Server Analysis Services.
- **XMLA**: Es el más reciente estándar creado para hacer entender las aplicaciones clientes con orígenes de datos multidimensionales.
- **XML**: Es un lenguaje que ya conocemos, que se compone de diferentes etiquetas bien estructuradas, de muy fácil uso, y donde su función principal es describir datos. Permite la lectura de datos desde diferentes aplicaciones.

Una nueva incorporación que ha realizado **SQL Server 2005** con respecto a otras versiones, es la posibilidad de guardar en un proyecto nuestros servidores registrados y nuestras consultas. Con la edición **SQL Server Management Studio Express** no nos aparecen todas las posibilidades con respecto a esta nueva funcionalidad, y si que se puede ver en el resto de ediciones de SQL Server 2005. Es similar a la forma de crear proyectos con **Visual Studio 2005**, la única diferencia es que en los proyectos de SQL Server

Management Studio guardamos conexiones y consultas, y en Visual Studio archivos, librerías, y demás funcionalidades que conlleva una aplicación.

La inclusión de los proyectos con **SQL Server 2005** da mucha facilidad de manejo a los administradores de bases de datos. Por ejemplo, podríamos tener 3 proyectos diferentes, uno de ellos tendrá las conexiones a los servidores del entorno de desarrollo, otro a los servidores de pruebas y finalmente un tercer proyecto que contenga las conexiones a los servidores del entorno de producción. Es un ejemplo de las ventajas que puede darnos esta nueva incorporación.

Ahora pasaremos a estudiar otra nueva innovación del SQL server 2005 y es la inclusión del dato de tipo XML. Como se comentó en líneas anteriores XML: Es un lenguaje que ya conocemos, que se compone de diferentes etiquetas bien estructuradas, de muy fácil uso, y donde su función principal es describir datos. Permite la lectura de datos desde diferentes aplicaciones. A través de ejemplos y ejercicios se intentara dar un panorama más claro de cómo poder usar los datos XML mediante consultas con Xpath.

### 3.6 Microsoft SQL Server y XML, consultas con XPath.

Transact-SQL soporta una parte importante del lenguaje XQuery mediante el cual se pueden extraer y preguntar al tipo de datos XML y sus componentes. Esta implementación se hizo cuando estaba en desarrollo el lenguaje XQuery por el World Wide Web Consortium (W3C). Vamos a ver en profundidad qué parte semántica está desarrollada en SQL Server 2005 en su unión con XPath.

#### 3.6.1 Introducción

XPath es el lenguaje para encontrar información dentro de una variable o campo XML. Se usa para navegar por los elementos y atributos de un objeto XML. Es un estándar del W3C en cuyas expresiones se basa XQuery para extraer información. Por esto, comprender XPath es fundamental para manejar documentos XML.

#### 3.6.2 Nodos

Dentro de Xpath hay siete clases de nodos: elemento, atributo, texto, espacio de nombres, instrucciones de procesamiento, comentarios y documento (el nodo raíz). Dentro del documento que vamos a usar para los ejemplos (ver apéndice), podemos apreciar los siguientes nodos:

Documento: `<librería>`

Elemento: `<autor>Antonio Machado</autor>`

Atributo: `lang="es"`

Texto: `29.99`

Espacio de nombres: `<?xml version="1.0" encoding="ISO-8859-1" ?>`

Comentarios: `(: Define los libros ... :)`

#### 3.6.3 Relaciones entre nodos

Una vez que conocemos las distintas partes de un documento XML, podemos apreciar las relaciones entre nodos para así aplicar correctamente las funciones XPath. Son bastante sencillas:

Padre: Cada elemento y atributo tiene un padre. En el ejemplo, el elemento libro es el padre de autor, título y precio.



**Hijos:** Cada elemento puede tener cero, uno o más hijos. En el ejemplo, el elemento libro tiene como hijos autor, título y precio.

**Hermanos:** Son los nodos que tienen el mismo padre. En el ejemplo, autor, título y precio son hermanos.

**Antepasados:** Son el padre de un nodo, el padre del padre, y así. En el ejemplo, el elemento título con texto *Stukas vs. Spitfires* tiene como antepasados libro, colección y librería.

**Descendientes:** El hijo de un nodo, el hijo del hijo, etc. En el ejemplo, los descendientes de librería son libro, colección, autor, título y precio.

### 3.6.4 Selección de nodos

Ahora sí que podemos entrar en los ejemplos de las distintas formas de extraer datos de un documento XML. Para ver el documento completo, hay varias formas. Poner el nombre del nodo raíz (sin olvidarnos nunca que se distingue entre mayúsculas y minúsculas):

```
SELECT @X.query('librería');
```

O usar la expresión «/» que significa el nodo raíz cuando no tiene nada a su izquierda:

```
SELECT @X.query('/');
```

Ambas formas son intercambiables. Por lo tanto, son equivalentes a:

```
SELECT @X.query('/librería');
```

Estas formas dan como resultado el documento completo:

```

<librería>
  <libro>
    <autor>Antonio Machado</autor>
    <título lang="es">Poesías completas</título>
    <precio>29.99</precio>
  </libro>
  <libro>
    <autor>Fernando Lázaro Carreter</autor>
    <título lang="es">El dardo en la palabra</título>
    <precio>23.99</precio>
  </libro>
  <libro>
    <autor>Antonio Buero Vallejo</autor>
    <título>La tejedora de sueños</título>
    <precio>19.95</precio>
  </libro>
  <colección>
    <título lang="eng">Battle over Britain</título>
    <libro>
      <título lang="eng">Stukas vs. Spitfires</título>
      <precio>49.95</precio>
    </libro>
    <libro>
      <título lang="eng">U. K. strikes back</título>
      <precio>49.95</precio>
    </libro>
  </colección>
</librería>

```

No aparece el comentario porque lo hemos quitado para que SQL Server prepare bien el texto XML.

Para traer un nodo específico, se encadenan los elementos mediante «/». Por ejemplo:

```
SELECT @X.query('/librería/libro');
```

Nos trae todos los nodos *libro* que son hijos del nodo raíz:

```

<libro>
  <autor>Antonio Machado</autor>
  <título lang="es">Poesías completas</título>
  <precio>29.99</precio>
</libro>
<libro>
  <autor>Fernando Lázaro Carreter</autor>
  <título lang="es">El dardo en la palabra</título>
  <precio>23.99</precio>
</libro>
<libro>
  <autor>Antonio Buero Vallejo</autor>
  <título>La tejedora de sueños</título>
  <precio>19.95</precio>
</libro>

```

Fijémonos que no trae los elementos *libro* que están debajo de *colección*. Para ello se usa «//» que significa traer los elementos estén donde estén. Por lo tanto, si queremos todos los elementos *libro* descendientes del nodo raíz, usaremos:

```
SELECT @X.query('/librería//libro');
```

Que nos trae:

```

└─ <libro>
  └─ <autor>Antonio Machado</autor>
  └─ <título lang="es">Poesías completas</título>
  └─ <precio>29.99</precio>
└─ </libro>
└─ <libro>
  └─ <autor>Fernando Lázaro Carreter</autor>
  └─ <título lang="es">El dardo en la palabra</título>
  └─ <precio>23.99</precio>
└─ </libro>
└─ <libro>
  └─ <autor>Antonio Buero Vallejo</autor>
  └─ <título>La tejedora de sueños</título>
  └─ <precio>19.95</precio>
└─ </libro>
└─ <libro>
  └─ <título lang="eng">Stukas vs. Spitfires</título>
  └─ <precio>49.95</precio>
└─ </libro>
└─ <libro>
  └─ <título lang="eng">U. K. strikes back</título>
  └─ <precio>49.95</precio>
└─ </libro>

```

Fijémonos que, aunque es el mismo resultado, esto es conceptualmente distinto de:

```
SELECT @X.query('//libro');
```

Que significa, trae todos los elementos libro estén donde estén. Téngase en cuenta que en la consulta anterior se especificó un nodo raíz y en esta no.

### 3.6.5 Predicados

Los predicados sirven para especificar aún más la búsqueda. Van siempre entre corchetes «[» y «]». Por ejemplo, queremos el precio del primer libro:

```
SELECT @X.query('librería/libro[1]/precio');
```

Podemos ver, además que en SQL Server el primer *libro* es el elemento [1]. Ahora bien, hay que tener cuidado cuando se usa la expresión descendiente:

```
SELECT @X.query('librería/libro[1]/precio');
```

Vemos cómo al navegar por el documento XML, SQL Server se encuentra con dos libros que son el primero de los hermanos: el que se encuentra debajo de *librería* y el que hay debajo de *colección*.

Podemos buscar el último de los libros:

```
SELECT @X.query('librería/libro[last()]');
```

```
<libro>
  <autor>Antonio Buero Vallejo</autor>
  <título>La tejedora de sueños</título>
  <precio>19.95</precio>
</libro>
```

También podemos buscar los dos primeros:

```
SELECT @X.query('librería/libro[position()<3]');
```

```
<libro>
  <autor>Antonio Machado</autor>
  <título lang="es">Poesías completas</título>
  <precio>29.99</precio>
</libro>
<libro>
  <autor>Fernando Lázaro Carreter</autor>
  <título lang="es">El dardo en la palabra</título>
  <precio>23.99</precio>
</libro>
```

«@»:

```
SELECT @X.query('//título[@lang="es"]');
```

Que nos da dos títulos si la ejecutamos, porque el tercer libro, aunque está en español, no tiene el atributo.

Para obtener todos los elementos que tienen un atributo, SQL Server no permite usar la expresión «//@lang», devolviendo:

**XQuery [query()]: El atributo no puede aparecer fuera de un elemento**

La expresión que sí acepta es usando comodines:

```
SELECT @X.query('//*[@lang');
```

Otro elemento de XPath que no acepta SQL Server 2005 es usar el operador «|» para obtener a la vez varios tipos de nodos. Devuelve:

**Xquery [query()]: No se admite la sintaxis XQuery 'union'**

### 3.6.6 Funciones

Si lo que queremos es lo que hay dentro de un nodo, tenemos que usar funciones. Por ejemplo, para obtener el texto sería:

```
SELECT @X.query('/librería/libro[1]/autor/text());
```

Si lo que queremos es lo que hay dentro del elemento, es otra función. Veamos dos ejemplos:

```
SELECT @X.query('/librería/libro[1]/node());
```

Aquí nos devuelve todo lo que hay dentro del primer libro

```
<autor>Antonio Machado</autor>  
<título lang="es">Poesías completas</título>  
<precio>29.99</precio>
```

Mismo que la función text():

```
SELECT @X.query('/librería/libro[1]/autor/node());
```

Para obtener el valor de un atributo no permite usar:

```
SELECT @X.query('/librería/libro[1]/título/@lang');
```

Pero sí se puede usar la función data() para obtenerlo:

```
SELECT @X.query('data(/librería/libro[1]/título/@lang)');
```

Por último, podemos usar funciones de agregado sobre el valor o los valores que nos devuelve una expresión.

Por tanto:

```
SELECT @X.query('sum(/librería/libro/precio)');
```

Devuelve la suma de todos los precios, de libros que están bajo el nodo raíz (73.93). Sin embargo:

```
SELECT @X.query('sum(//precio');
```

Devuelve la suma de todos los elementos *precio* estén donde estén (173.83). Se pueden usar las funciones count, avg, max, min y sum.

### 3.6.7 Conclusión

Bueno, pues después de todo este recorrido ya tenemos una idea de cómo extraer datos de una variable, XML. Para hacerlo de un campo XML se trabaja igual, sólo que usando el nombre del campo dentro de una consulta, donde ahora estamos usando la variable. De esta manera podemos ir usando campos XML para datos que no son estrictamente estructurados y que imponen una carga extra al tener que definir columnas redundantes, que casi siempre quedan vacías, en las tablas. A continuación se muestra el dato XML que se utilizó para los ejemplos anteriores.

### 3.6.8 Apéndice: Variable XML

```
DECLARE @X XML;  
SET @X = '<?xml version="1.0" encoding="ISO-8859-1"?>  
<librería>  
(: Define los libros y colecciones que hay en una librería :)  
<libro>  
<autor>Antonio Machado</autor>  
<título lang="es">Poesías completas</título>  
<precio>29.99</precio>  
</libro>  
<libro>  
<autor>Fernando Lázaro Carreter</autor>  
<título lang="es">El dardo en la palabra</título>  
<precio>23.99</precio>  
</libro>  
<libro>  
<autor>Antonio Buero Vallejo</autor>  
<título>La tejedora de sueños</título>  
<precio>19.95</precio>  
</libro>  
<colección>  
<título lang="eng">Battle over Britain</título>  
<libro>  
<título lang="eng">Stukas vs. Spitfires</título>  
<precio>49.95</precio>  
</libro>  
<libro>  
<título lang="eng">U. K. strikes back</título>  
<precio>49.95</precio>  
</libro>  
</colección>  
</librería>  
';
```

### 3.7 Ejercicio de comparación de tiempo de respuesta usando consultas a datos XML

Ejercicio real sobre el uso de consultas sobre datos XML. Primero que nada vamos a interactuar con una sencilla aplicación de Windows, en conjunto con nuestra base de datos. La finalidad de este ejemplo es comparar el tiempo de respuesta, de una consulta de posiciones leídas desde un dato XML almacenado en nuestra base de datos, por un lado se usarán Xqueries y por el otro lado se leerá el dato XML desde visual studio .net y se descompondrá para leer la información deseada. Como resultado, se desea obtener los siguientes campos (latitud, longitud, velocidad, fecha) que representara las posiciones de una unidad en un periodo de tiempo.

Para el siguiente ejercicio se necesita tener instalado las siguientes aplicaciones:

- SQL Server 2005 express edition.
- SQL server management studio.
- Visual Studio .NET express edition.

Como primer paso vamos a crear una pequeña base de datos, llamada Rastreo.

- Abrimos el SQL server 2005 management studio
- Abrir la opción de View que se encuentra en barra de herramientas, seleccionamos Object explorer.
- Ahora nos muestran los servidores y las bases de datos existentes, ahora dando clic derecho sobre la carpeta databases, seleccionamos (New Databases)
- Nos pide que nombremos la nueva base de datos, para este ejemplo la nombraré “Rastreo”.
- Una vez creada la base de datos, agregaremos una tabla la cual nombraremos “posiciones” la cual va simular un conjunto de posiciones asociados a una unidad vehicular. Esto es, se simulará, las notificaciones de posiciones que lanza un vehículo monitoreado vía satélite. Los campos que contendrá la tabla posiciones son los siguientes:



Campo	Tipo de dato	Permite nulos
posicionid	Int, identity(1,1)	Not null
unidadid	Int	Not null
velocidad	Decimal(18,0)	Not null
fecha	Datetime	Not null
definicion	Xml	Not null

Tabla "Posiciones"

- Una vez creada la base de datos y la tabla posiciones. Vamos a crear los siguientes Procedimientos almacenados. Abrimos la carpeta Programmability y damos clic derecho sobre Store procedure y seleccionamos "New store procedure" y copiamos el siguiente script y ejecutarlo con F5:

```
set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[posiciones_ConsultaHistorialSinXquery]
    @unidadId int,
    @fechaInicial datetime,
    @fechaFinal datetime
AS
SET NOCOUNT ON
=====
-- Descripcion: Regresa todos los registros
--                para la unidad entre fecha inicial y final

SELECT *
FROM [posiciones]
where [unidadId] = @unidadId and fecha between @fechaInicial and @fechafinal
ORDER BY fecha desc
```

- Repetimos el paso anterior para crear el siguiente Store procedure y copiamos el siguiente script y ejecutarlo con F5:

```

set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[posiciones_ConultaHistorialConXquery]
@unidadId int,
@fechaInicial datetime,
@fechaFinal datetime
AS
SET NOCOUNT ON
=====
-- Descripcion: Regresa el historial de posiciones
--                para la unidad entre fecha inicial y final
--
--=====
--***** Declaracion de variables
declare @definicion XML,
@latStr nvarchar(15),
@longStr nvarchar(15),
@lat float,
@long float,
@velocidad int,
@fecha datetime

create table #posiciones(
latitud float,
longitud float,
velocidad int,
fecha datetime)

DECLARE CurPosiciones CURSOR READ_ONLY FOR
SELECT [velocidad]
,[fecha]
,[definicion]
FROM [posiciones]
where [unidadId] = @unidadId and fecha between @fechaInicial and @fechafinal
ORDER BY fecha desc

OPEN CurPosiciones
FETCH NEXT FROM CurPosiciones
INTO @velocidad, @fecha, @definicion

WHILE @@FETCH_STATUS = 0
BEGIN
set @latStr = convert(nvarchar,(SELECT @definicion.query('/NewDataSet/posiciones/latitud/text()')))
set @longStr = convert(nvarchar,(SELECT
@definicion.query('/NewDataSet/posiciones/longitud/text()')))
set @long = @longStr
set @lat = @latStr
insert into #posiciones values(@lat, @long, @velocidad, @fecha)
FETCH NEXT FROM CurPosiciones
INTO @velocidad, @fecha, @definicion
END
CLOSE CurPosiciones
DEALLOCATE CurPosiciones
Select * from #posiciones
drop table #posiciones

```

- Para finalizar agregamos el ultimo Store procedure que nos auxiliara para efecto de nuestro ejemplo, copiar el siguiente script y ejecutarlo con F5:

```

//Inserta registros en la tabla posiciones de la base de datos "Rastreo
set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[Posiciones_insertarRegistros]
    @unidadid int,
    @velocidad decimal(18,0),
    @Fecha datetime,
    @definicion xml

AS
SET NOCOUNT ON
INSERT INTO [titulacion].[dbo].[posiciones]
    ([unidadid]
    ,[velocidad]
    ,[fecha]
    ,[definicion])
VALUES
    (@unidadid
    ,@velocidad
    ,@fecha
    ,@definicion)
    
```

Hemos concluido con lo que respecta a SQL Server 2005, ahora.

- Abrimos el Visual Studio .NET, en el menú nos vamos a archivo → Nuevo → Proyecto. Seleccionamos el tipo de proyecto, Visual Basic → Aplicación de Windows y le asignamos un nombre.
- Ahora a nuestra form1 agregamos los siguientes controles:

<i>Control</i>	<i>Name</i>	<i>Texto</i>
Button	ConsultaConXquery	Consulta Xquery
Button	ConsultasinXquery	Consulta sin XQuery
Datagridview	dgvSinXquery	
Datagridview	dgvXquery	
Label	LblDuracionConsulta	
Label	LblDuracionConsultaSinXquery	
Label	LblNoRegistrosconXquery	
Label	LblNoRegistrosSinXquery	

Ahora vamos al código de nuestra form1, dando doble clic sobre cualquier parte de la form1. Una vez en código copiamos los siguientes fragmentos de código.

Declaramos las siguientes variables dentro de la clase form1, no incluirlos dentro de alguna función o procedimiento.

```
Protected _db As Database
Dim number As New Random
Dim latitud As Double
Dim longitud As Double
Dim latitudanterior As Double
Dim longitudanterior As Double
Dim estado_activo As String
Dim alarmas As String
Dim fecha As DateTime
Dim velocidad As Double
```

Como podemos ver nos muestra un error con la siguiente variable

`Protected _db As Database` ‘El tipo 'Database' no está definido’ esto es porque no tenemos referencia hacia nuestro espacio de nombres correspondiente. Necesitamos agregar algunas referencias a nuestro proyecto, estas referencias fueron descargadas desde el sitio de Microsoft, son ensamblados gratuitos, y son bloques de programación que Microsoft ha desarrollado para facilitar la programación, el bloque que se usara para este ejemplo será el Data Access Block del cual su principal función es gestionar la base de datos.

Entonces las referencias que se necesitarán son:

Reference: Microsoft.Practices.EnterpriseLibrary.Common.dll

Reference: Microsoft.Practices.EnterpriseLibrary.Data.dll

Una vez agregadas las referencias continuaremos copiando los siguientes fragmentos de código. Las siguientes líneas copiarlas hasta la primera línea.

```
Imports Microsoft.Practices.EnterpriseLibrary.Common
Imports Microsoft.Practices.EnterpriseLibrary.Data
Imports System.IO
```

Ahora copiaremos el código del evento load de la forma:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    _db = DatabaseFactory.CreateDatabase("Rastreo")
    llenarEjemplotitulacion() ' Correr solo una vez ya que esta funcion nos servira para llenar
    nuestra base de datos.
    Me.Close()
End Sub
```

Este método nos servirá para poblar nuestra tabla “posiciones” de nuestra base de datos “Rastreo”

```

Private Sub llenarDbEjemplotitulacion()
    Dim dsposiciones As New DataSet
    dsposiciones.Tables.Add("posiciones")
    dsposiciones.Tables("posiciones").Columns.Add("latitud")
    dsposiciones.Tables("posiciones").Columns.Add("longitud")
    dsposiciones.Tables("posiciones").Columns.Add("latitudanterior")
    dsposiciones.Tables("posiciones").Columns.Add("longitudanterior")
    dsposiciones.Tables("posiciones").Columns.Add("estado_activo")
    dsposiciones.Tables("posiciones").Columns.Add("alarmas")
    dsposiciones.Tables("posiciones").Rows.Add()

    Dim dsGuardar As New DataSet
    dsGuardar.Tables.Add("informacion")
    dsGuardar.Tables("informacion").Columns.Add("unidadid")
    dsGuardar.Tables("informacion").Columns.Add("velocidad")
    dsGuardar.Tables("informacion").Columns.Add("fecha")
    dsGuardar.Tables("informacion").Columns.Add("definicion")
    dsGuardar.Tables("informacion").Rows.Add()
    'Llenar la base de datos con informacion valida
    'se llenara con 14000 posiciones para una unidad, existiendo 10 unidades

    Dim auxunidades, auxregistrosporunidad As Integer
    'Creamos un ciclo para hacer el llenado
    For auxunidades = 1 To 10
        fecha = System.DateTime.Now.AddMinutes(1)
        For auxregistrosporunidad = 1 To 14000
            latitud = number.Next(1, 999)
            longitud = number.Next(200, 600)
            latitudanterior = number.Next(30, 300)
            longitudanterior = number.Next(100, 800)
            estado_activo = "activo"
            alarmas = "ninguna"
            velocidad = number.Next(80, 150)
            dsposiciones.Tables("posiciones").Rows(0)("latitud") = latitud
            dsposiciones.Tables("posiciones").Rows(0)("longitud") = longitud
            dsposiciones.Tables("posiciones").Rows(0)("latitudanterior") = latitudanterior
            dsposiciones.Tables("posiciones").Rows(0)("longitudanterior") = longitudanterior
            dsposiciones.Tables("posiciones").Rows(0)("estado_activo") = estado_activo
            dsposiciones.Tables("posiciones").Rows(0)("alarmas") = alarmas
            Dim verif As Integer = _db.ExecuteNonQuery("Posiciones_insertarRegistros", New String()
            {auxunidades, velocidad, fecha, dsposiciones.GetXml})
        Next
    Next
End Sub

```

Ahora ejecutamos nuestro programita con F5. La aplicación mostrará un mensaje de que “La base de datos ha sido cargada”, aceptamos y continuamos con nuestra codificación de nuestra aplicación. Volvemos a nuestro evento load de la forma y comentamos las siguientes líneas:

```

'llenarDbEjemplotitulacion() ' Correr solo una vez ya que esta funcion nos servira para llenar nuestra base de datos.
'MsgBox("La base de datos ha sido cargada")
'Me.Close()

```

De no comentar las líneas, la aplicación estará cargando la base de datos con nuevos registros en la tabla de posiciones cada vez que se inicie la aplicación.

Ahora continuamos con los eventos de los controles de la forma:

Dentro de este manejador, se obtiene todos los registros, y uno por uno se recorre para leer el campo XML, en este caso es “definición” a partir de eso y conociendo la estructura del dato XML, vamos directo a los campos que necesitamos, “latitud”, “longitud”.

```
Private Sub ConsultasInXquery_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ConsultasInXquery.Click
    Dim dsResultados As New DataSet
    Dim dsfinal As New DataSet
    Dim renglon As Integer = 0
    Dim tiempoinicial As DateTime = System.DateTime.Now
    Dim datosxml As String = ""
    'Criterios de búsquedas para la unidadid = 1
    dsResultados = _db.ExecuteDataSet("posiciones_ConsultaHistorialSinXquery", New String() {1,
Convert.ToDateTime("24/01/2007"), Convert.ToDateTime("24/12/2009")})
    dsfinal.Tables.Add("datos")
    dsfinal.Tables("datos").Columns.Add("latitud")
    dsfinal.Tables("datos").Columns.Add("longitud")
    dsfinal.Tables("datos").Columns.Add("velocidad")
    dsfinal.Tables("datos").Columns.Add("fecha")

    While dsResultados.Tables(0).Rows.Count > renglon
        datosxml = ""
        datosxml = dsResultados.Tables(0).Rows(renglon)("definicion")
        Dim dsXml As New DataSet
        dsXml.ReadXml(New StringReader(datosxml))
        dsfinal.Tables("datos").Rows.Add(dsXml.Tables(0).Rows(0)("latitud"),
dsXml.Tables(0).Rows(0)("longitud"), dsResultados.Tables(0).Rows(renglon)("velocidad"),
dsResultados.Tables(0).Rows(renglon)("fecha"))
        renglon += 1
    End While

    Me.dgvSinXquery.DataSource = dsfinal.Tables(0)
    Dim tiempofinal As DateTime = System.DateTime.Now
    Dim tiempoTotal As TimeSpan = tiempofinal.Subtract(tiempoinicial)

    LblDuracionConsultaSinXquery.Text = tiempoTotal.TotalSeconds
    LblNoRegistrosSinXquery.Text = dsfinal.Tables(0).Rows.Count.ToString
End Sub
```

El siguiente manejador simplemente ejecuta un store procedure el cual realiza directamente lecturas sobre el tipo de dato XML, el cual utiliza Xquerys.

```
Private Sub ConsultarHistorialPosicionesUsandoXquerys()
    Dim dsResultados As New DataSet
    Dim tiempoinicial As DateTime = System.DateTime.Now
    dsResultados = _db.ExecuteDataSet("posiciones_ConsultaHistorialConXquery", New String() {1,
Convert.ToDateTime("24/01/2007"), Convert.ToDateTime("24/12/2009")})
    Me.dgvXquery.DataSource = dsResultados.Tables(0)
    Dim tiempofinal As DateTime = System.DateTime.Now
    Dim tiempoTotal As TimeSpan = tiempofinal.Subtract(tiempoinicial)

    LblDuracionConsulta.Text = tiempoTotal.TotalSeconds
    LblNoRegistrosconXquery.Text = dsResultados.Tables(0).Rows.Count.ToString
End Sub
```

Una vez terminada la codificación, vamos a hacer un acomodo de controles con el fin de que quede un poco mas claro de que se trata el ejemplo, vamos a acomodar los controles de preferencia como se muestra en la figura siguiente



Bueno hemos llegado al momento de ejecutar nuestra aplicación, la cual tenía como finalidad, comparar tiempos de respuesta en consultas sobre datos XML desde código VB.NET contra consultas directas sobre SQL server usando Xquerys. Ahora para hacer esta comparación vamos a ejecutar la aplicación con F5. Al mostrar la ventana vamos a presionar los botones, no importa el orden, los dos botones deben mostrar los mismos resultados ya que realizan la misma consulta, lo que realmente nos interesa es el tiempo de respuesta que tiene cada uno de los métodos, el que usa Xquerys para recorrer un dato XML

o el que lo recorre mediante código en VB .NET. ahora veremos los resultados en la siguiente imagen, los resultados pueden variar dependiendo de la capacidad de la computadora y del número de procesos que se estén corriendo al momento de la prueba.

**Form1**

**Comparativa de consultas usando Xquerys directamente en SQL server contra el proceso de lectura desde código VB .NET**

**Consulta sin XQuery**

No Registros : **14000**

Duracion consulta(s) : **10,26476**

latitud	longitud	velocidad	fecha
554	414	127	05/12/0
574	494	110	05/12/0
556	242	96	05/12/0
776	393	129	05/12/0
202	450	146	05/12/0
103	514	130	05/12/0
531	394	140	05/12/0
555	478	136	05/12/0
797	307	101	05/12/0
02	507	127	05/12/0
377	271	80	05/12/0
138	285	105	05/12/0
467	461	135	05/12/0
606	217	136	05/12/0
751	238	108	05/12/0
554	536	119	05/12/0
511	565	102	05/12/0
291	303	123	05/12/0
232	278	87	05/12/0
634	306	132	05/12/0
877	205	103	05/12/0
539	541	99	05/12/0
...	...	...	...

**Consulta Xquery**

No Registros : **14000**

Duracion consulta(s) : **3,8252128**

latitud	longitud	velocidad	fecha
554	414	127	05/12/0
574	494	110	05/12/0
556	242	96	05/12/0
776	393	129	05/12/0
202	450	146	05/12/0
103	514	130	05/12/0
531	394	140	05/12/0
555	478	136	05/12/0
797	307	101	05/12/0
02	507	127	05/12/0
377	271	80	05/12/0
138	285	105	05/12/0
467	461	135	05/12/0
606	217	136	05/12/0
751	238	108	05/12/0
554	536	119	05/12/0
511	565	102	05/12/0
291	303	123	05/12/0
232	278	87	05/12/0
634	306	132	05/12/0
877	205	103	05/12/0
539	541	99	05/12/0
...	...	...	...

Como se puede apreciar en la ventana de resultados los tiempos de respuesta si varían un poco, siendo más rápida la consulta que utiliza Xquerys y cursores directamente sobre el dato xml.



Conclusiones: Gracias a su Tipo de dato xml, ahora Sql Server permite almacenar gran cantidad de informacion en un solo campo con estructura definida o indefinida y con facil y rapido acceso a cuaquier tipo de informacion contenida en dicho campo, mediante el uso de Xquerys.

# Capítulo IV

## Funcionamiento del Sistema de Metrología

## 4.1 Descripción de Formas

La descripción de formas nos sirve para dar a conocer al usuario el funcionamiento del sistema así como cada una de las partes que lo componen.

## 4.2 Autenticación

La primera descripción de forma que haremos será la de autenticación para algún usuario. Esta pantalla es por la cual el usuario introduce su nombre de usuario y contraseña con el fin de que el sistema lo reconozca y le permita el acceso al sistema, de lo contrario si no lo reconoce simplemente se lo hará saber al usuario enviando un mensaje al usuario que verifique su password y/o contraseña.



The image shows a software window titled "Autenticación" (Authentication). The window has a blue header bar with the text "SIDMAC Calidad - Sistema de administración de Metrología". On the left side, there is a logo for "CONDUMEX" which consists of a stylized triangle with orange and black elements. On the right side, there are two text input fields: the first is labeled "Nombre de usuario" and the second is labeled "Contraseña". Below these fields are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Fig. 4.2 Pantalla inicial del sistema

### 4.3 Pantalla Inicial del Sistema:

Una vez que el usuario haya sido reconocido por el sistema como usuario del mismo entonces le mostrará una ventana tal y como se muestra en la figura 4.2

En la cual se puede apreciar cinco funcionalidades principales las cuales se muestran a lado izquierdo de la pantalla y son:

1. Administrar instrumentos.
2. Calibraciones.
3. Plan de calibraciones.
4. Calibraciones por atender.
5. Informes de calibración.

Estas opciones son parte de los requisitos iniciales del cliente, estas opciones están disponibles para el usuario para ser usadas inmediatamente si así lo desea.

La funcionalidad se vendrá dando poco a poco durante la descripción de las formas y su funcionamiento. Por el momento es necesario que el usuario sepa que existe una ventana la cual contiene el menú principal y una ventana que muestra las calibraciones que no han sido atendidas todavía.

Dentro del ambiente inicial se puede apreciar una ventana que aparece al iniciarse el sistema la cual su función principal es avisarle al laboratorista que se tienen algunas calibraciones pendientes y que se necesitan programar para su calibración. Esta forma se verá más adelante por el momento basta saber que el sistema le avisa al laboratorista si existen o no calibraciones que no han sido atendidas.

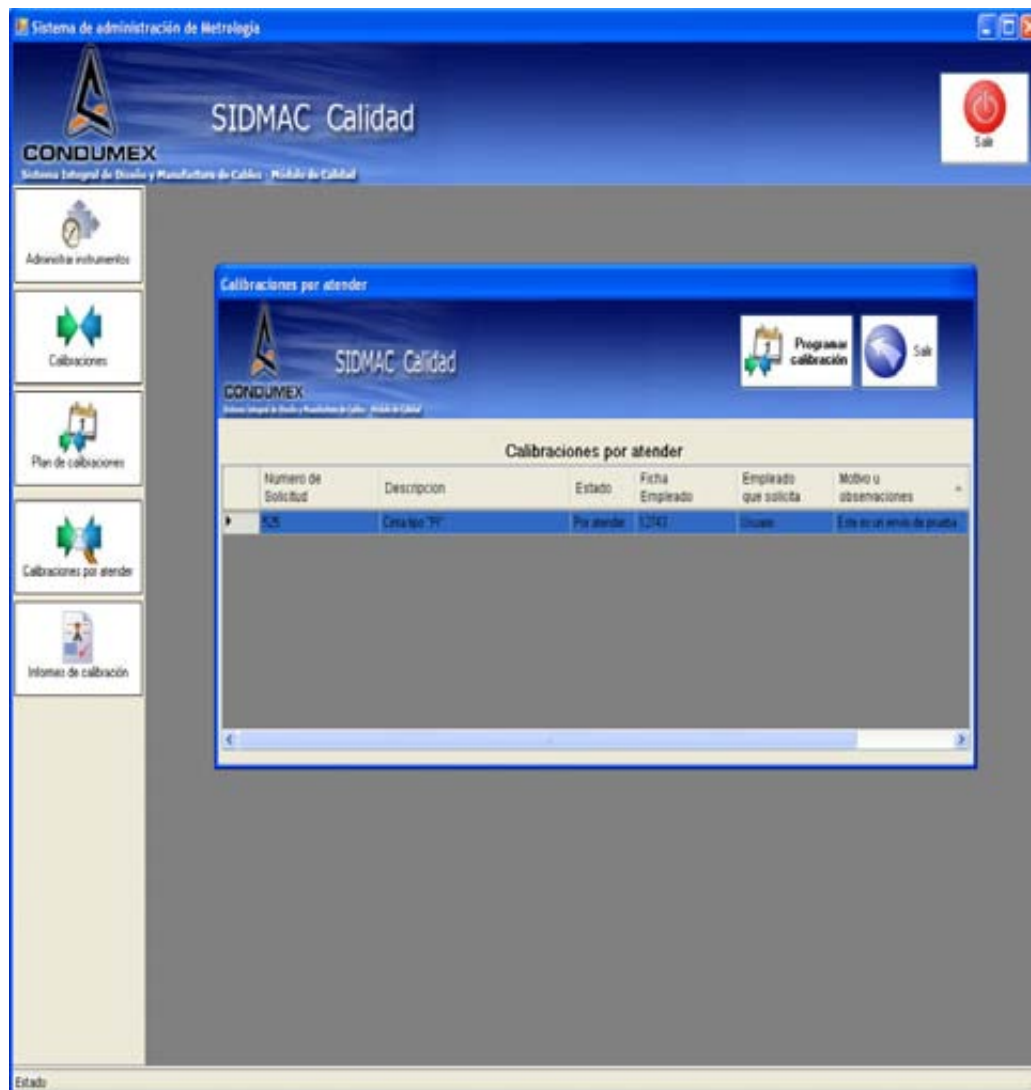


Fig. 4.2 Pantalla inicial del sistema

Después de haber conocido la ventana inicial la cual nos servirá como contenedora de más ventanas, esto es, esta ventana siempre estará detrás de todos los movimientos que hagamos. Ahora pasaremos a analizar cada una de las cinco funcionalidades principales del sistema de administración de metrología.

Pasemos entonces a la primera opción de arriba hacia abajo, vamos a describir la opción de administrar instrumentos.

#### 4.4 Administrar Instrumentos

En esta sección hablaremos del funcionamiento del sistema después de que el usuario presiona la opción de administrar instrumentos, en este momento el sistema muestra los instrumentos registrados en el sistema hasta el momento, ver (figura 4.3)

En figura 4.3 se muestran dichos instrumentos cabe mencionar que se muestra información del instrumento tal como:

- Clave del instrumento: Esta sirve para identificar al instrumento, no es única
- Área: Departamento o Área de la planta a la cual pertenece o bien a que planta externa pertenece.
- Ficha Responsable: Los instrumentos tienen a alguien como responsable del mismo.
- Tipo: Nos dice el tipo de magnitud de la cual es capaz de medir.
- Descripción: Es la descripción del instrumento.
- Estado: Como su nombre lo dice, no dice si el instrumento está en uso todavía o si ha causado baja temporal o definitiva.
- Observaciones: Dependiendo del comportamiento del instrumento se agregan observaciones al instrumento.
- Calibrado: Es una forma de saber si el instrumento aún sigue calibrado.

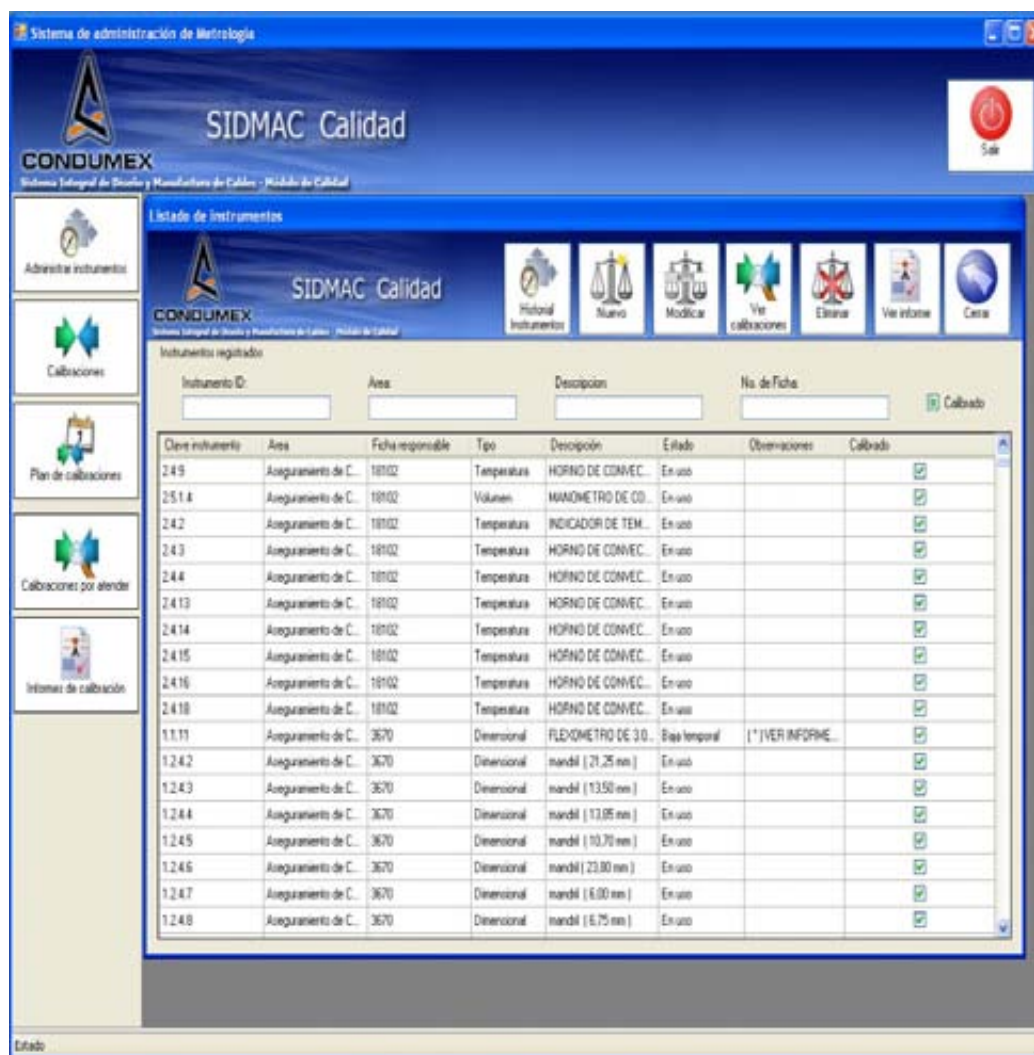


Fig. 4.3 Ventana de listado de instrumentos

Se puede ver que la ventana del listado de instrumentos (Fig. 4.3) también contiene funcionalidades en la parte superior se muestra el menú propio de esta ventana. El cual se lista y describe a continuación:

- Historial instrumento
- Nuevo
- Modificar
- Ver calibraciones
- Eliminar

- Ver informe
- Cerrar

#### 4.4.1 Historial Instrumento

Esta opción permite al usuario ver un reporte del comportamiento que ha tenido uno o más de un instrumento durante sus calibraciones, esto con el fin de comparar principalmente un historial de sus incertidumbres registradas en sus calibraciones anteriores. A continuación se presenta un ejemplo de un reporte de historial de instrumento.

Para generar un reporte,

- Para un reporte de un solo instrumento, simplemente posicionarse sobre el instrumento deseado y dar clic derecho del mouse y seleccionar Ver historial instrumento. Y listo el sistema se encargará de mostrar el reporte del instrumento seleccionado.
- Para un reporte de varios instrumentos, filtrar los instrumentos deseados, es decir, usar el control que aparece arriba del listado. Por ejemplo si se desea generar un reporte solo de instrumentos que tengan en común los caracteres M 1 en su clave de instrumento, entonces, escribo M 1 en el campo instrumento ID del control mencionado, esto nos filtrará y obtendremos los instrumentos deseados. Al tener los instrumentos que necesitamos, presionar el botón historial instrumentos, el sistema se encargará de generar el reporte. Este puede ser exportado a PDF o Excel. O bien ser impreso inmediatamente.

A continuación se muestra un ejemplo (Fig. 4.4) del reporte que formamos con los criterios descritos



**Hoja de características**

**SIDMAC CALIDAD**  
CONDUMEX  
Sistema Integral de Diseño y Manufactura de Cables - Módulo de Calidad

Informe principal

**SIDMAC - CALIDAD**  
**LAB. DE METROLOGIA**

**HISTORIAL INSTRUMENTOS**  
Viernes, 7 de Noviembre de 2008

	FechaCalibracion	Incertidumbre	Error Maximos Permitido
M 1.1.4	01-Feb-2008	5,93	0,0048
	12-Dic-1900		
M 1.1.5	06-Dic-2008	8,53	0,00037
	12-Dic-1900		
M 1.1.6	04-Dic-2008	9,67	0,00005
	12-Dic-1900		

Fig. 4.4 Reporte historial de instrumentos.

### 4.4.2 Nuevo

Al dar clic sobre esta opción el sistema le muestra al usuario una pantalla de captura de información del nuevo instrumento que se creará. A continuación se muestra la ventana de creación de nuevo instrumento.

**Información del instrumento**

**CONDUMEX** SIDMAC Calidad  
Laboratorio de metrología  
Sistema Integral de Diseño y Manufactura de Cables - Módulo de Calidad

Guardar    Guardar y continuar    Cancelar

Planta o Área:  Tipo de medición: Dimensional

Clave de instrumento:  Descripción:

Responsable:  Número de serie:

Marca:  Accesorios:

Modelo:

Fecha de adquisición: Viernes , 07 de Noviembre de 2008    Uso:

Alimentación:  Tipo:

Calibración interna     Uso colectivo

Carta de trazabilidad:  buscar...    Estado del instrumento:

Características:

Magnitud	Unidad	Alcance	Resolución	Exactitud
Dimensional				
Eléctrica				
Fuerza				
Masa				
Presión				
Temperatura				
Tiempo				
Velocidad				

Observaciones:

Fig. 4.5 Ventana de captura de nuevo instrumento

Aquí el usuario ingresa la información que tiene del nuevo instrumento, no es obligatorio llenar todos los campos que se muestran.

En esta ventana se observa que cuenta con 3 opciones que pueden ser usadas en cualquier momento que el usuario desee.

- Guardar: al dar clic en guardar el sistema cierra la ventana de captura y guarda el nuevo instrumento en la base de datos

- Guardar y continuar: al dar clic el sistema guarda el nuevo instrumento en la base de datos y limpia nuevamente la ventana de captura para ingresar otro nuevo instrumento si lo desea el usuario.
- Cancelar: al dar clic en cancelar, el sistema cancela la captura y cierra la ventana,

### 4.4.3 Modificar

Esta funcionalidad permite al usuario modificar la información de algún instrumento. Ver figura 4.6

**Información del instrumento**

CONDUMEX SIDMAC Calidad  
Laboratorio de metrología  
Sistema Integral de Diseño y Manufactura de Cables - Módulo de Calidad

Planta o Área:  Tipo de medición:

Clave de instrumento:  Descripción:

Responsable:  Número de serie:

Marca:  Accesorios:

Modelo:

Fecha de adquisición:

Alimentación:  Uso:

Calibración interna  Uso colectivo Tipo:

Carta de trazabilidad:   Estado del instrumento:

Características:
 

Magnitud	Unidad	Alcance	Resolución	Exactitud
<input type="text" value="Temperatura"/>	<input type="text" value="°C"/>	<input type="text" value="15 a 350"/>	<input type="text" value="+ 0,1"/>	<input type="text" value="+1%"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Observaciones:

Fig. 4.6 Ventana para modificar la información del instrumento.

Para modificar la información de algún instrumento el usuario deberá seleccionar algún elemento de la lista de instrumentos y después dar clic en modificar. Este comando

mostrara la información del instrumento seleccionado y el usuario podrá modificar la información del mismo. De igual forma esta ventana cuenta con sus respectivas funciones como son:

- Guardar: al dar clic en guardar el sistema cierra la ventana de captura y actualiza la información del instrumento en la base de datos
- Guardar y continuar: al dar clic, el sistema actualiza la información del instrumento en la base de datos y limpia la ventana de captura con opción de que el usuario si lo desea ingrese algún nuevo instrumento
- Cancelar: al dar clic en cancelar, el sistema cancela la actualización del instrumento y cierra la ventana de captura.

#### 4.4.4 Ver Calibraciones

Este comando le permite al usuario ver las calibraciones que tiene asociadas algún instrumento.

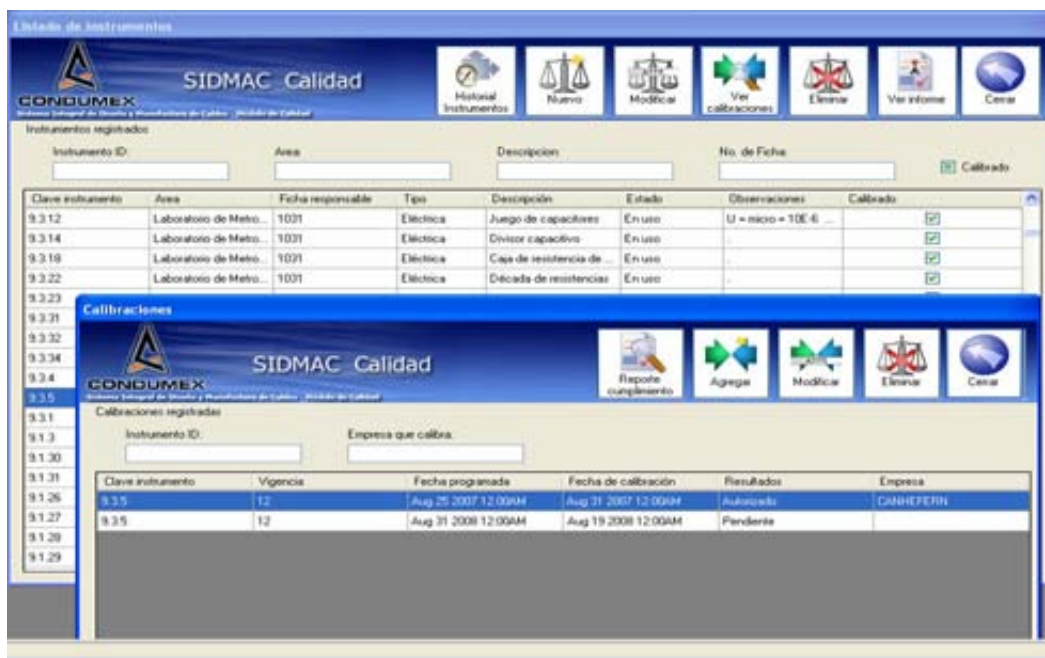


Fig. 4.7 Calibraciones asociadas al instrumento

Para conocer las calibraciones asociadas a algún instrumento primeramente se debe seleccionar algún instrumento de la lista inicial, una vez seleccionado procederemos a dar clic en la opción ver calibraciones esta nueva ventana mostrara las calibraciones efectuadas al instrumento si es que las tiene.

#### **4.4.5 Eliminar**

Como su nombre lo indica eliminará el instrumento seleccionado de la base de datos.

#### **4.4.6 Ver Informe**

Muestra las cartas de trazabilidad de algún instrumento seleccionado.

#### **4.4.7 Cerrar**

Este comando cierra la ventana de administración de instrumentos y nos regresa a la ventana contenedora inicial.

### 4.5 Calibraciones

Después de haber analizado el funcionamiento de la ventana de administración de instrumentos, ahora pasaremos a analizar el funcionamiento de la ventana de calibraciones. Esta se inicia cuando el usuario selecciona calibraciones del menú inicial que se encuentra en la parte izquierda de la ventana contenedora inicial. Una vez seleccionada la opción calibraciones, el sistema mostrará la ventana de calibraciones ver Fig. 4.8

#### 4.5.1 Ventana de Calibraciones

Muestra todo un listado de calibraciones capturadas en el sistema así como también el instrumento con el que están asociados



Fig. 4.8 Listado de calibraciones capturadas

De igual forma esta ventana de calibraciones tiene su menú propio que le da una funcionalidad independiente, dentro de este menú podemos encontrar:

- Reporte de cumplimiento
- Agregar
- Modificar
- Eliminar
- Cerrar

Así como también se puede apreciar el uso de un filtro para ejecutar consultas sobre el listado de calibraciones, para este caso se pueden filtrar las calibraciones ya sea por instrumento id y por empresa a la que se le calibra.

#### **4.5.2 Imprimir Reporte de Cumplimiento**

Esta funcionalidad es principalmente para generar reportes de calibraciones en un periodo de tiempo, este mostrará los instrumentos que se programaron en dicho periodo, aquí se mostrará un porcentaje de cumplimiento de calibraciones programadas.

Para generar un reporte de cumplimiento de calibraciones solo se necesita dar clic en el botón de reporte de cumplimiento, este mostrará una ventana en la que le pedirá al usuario seleccionar el periodo de fechas del cual desea que se genere el reporte. (Ver figura 4.9).

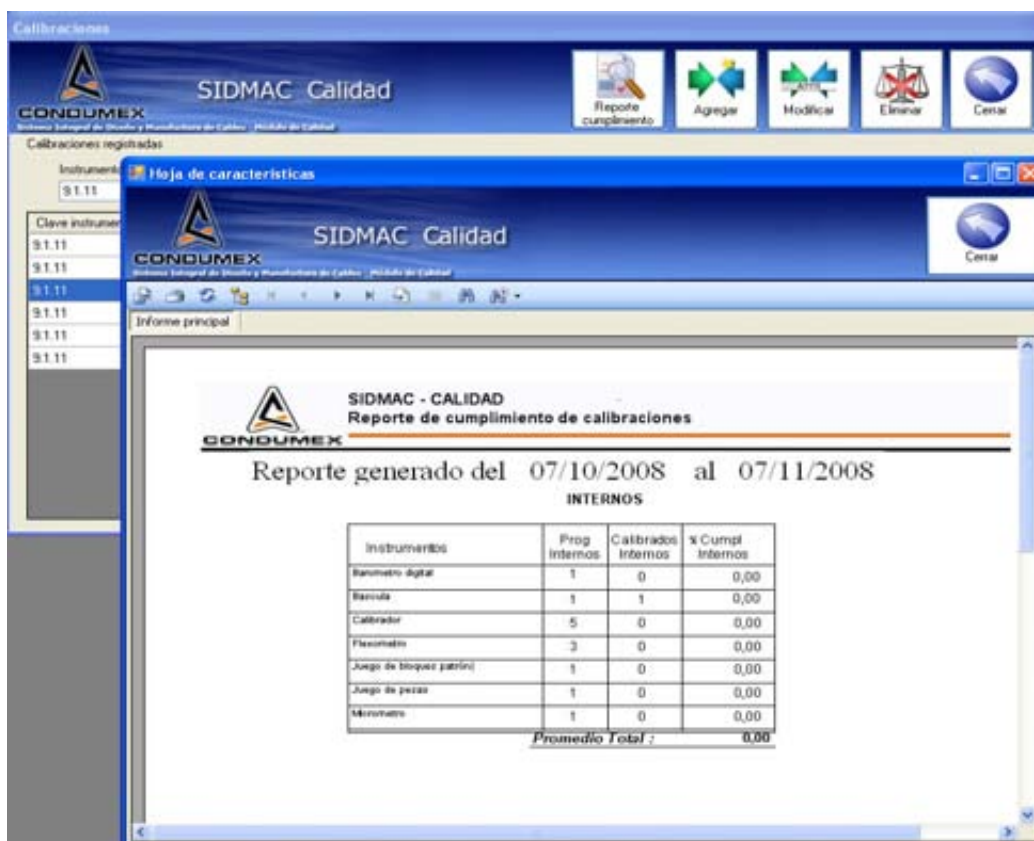


Fig. 4.9 Reporte de cumplimiento de calibraciones

### 4.5.3 Agregar

Nos permite agregar nuevas calibraciones al sistema, aquí nos pedirá llenar la información referente a la calibración.

Para agregar una nueva calibración primero, de la ventana inicial donde se muestra la lista de calibraciones dar clic sobre Agregar, ahora el sistema nos mostrará una ventana de captura de calibración, aquí nos pedirá que llenemos la información correspondiente, no es obligatorio llenar toda la información, los únicos campos obligatorios son:

- Instrumento
- Fecha programada
- Fecha calibración
- Informe
- Vigencia



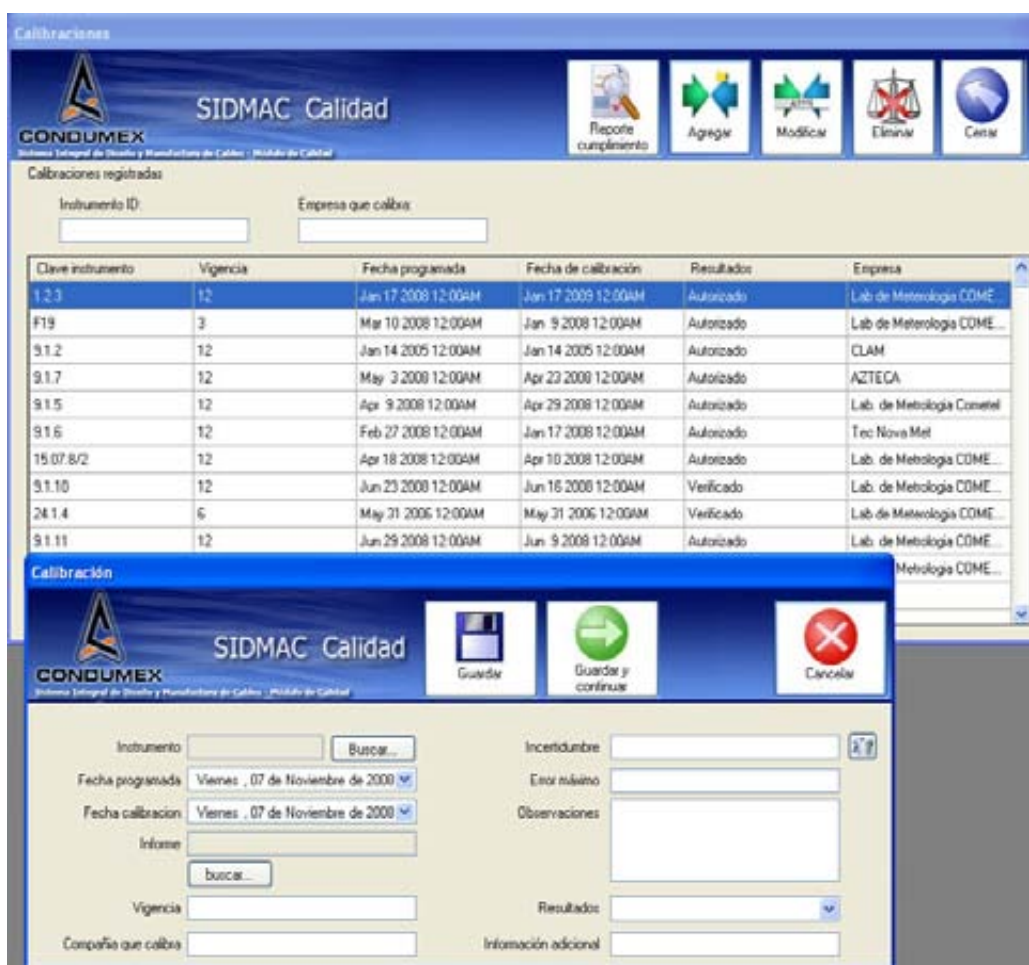


Fig. 4.10 Ventana de captura de nueva calibración

#### 4.5.4 Modificar

Nos permite modificar calibraciones existentes en el sistema

Para modificar alguna calibración, primero, de la lista de calibraciones inicial seleccionar dicha calibración a modificar, una vez seleccionada, entonces dar clic sobre el botón modificar, aquí el sistema muestra la información de la calibración, la cual puede ser editada por el usuario



Fig. 4.11 Listado de calibraciones

#### 4.5.5 Eliminar

Este comando nos permitirá eliminar calibraciones de nuestro sistema, no se borra físicamente de la base de datos simplemente se cambia su estado a eliminada.

Para hacer uso del botón eliminar, primero se tiene que seleccionar alguna calibración, una vez seleccionada se da clic sobre eliminar, el sistema le pregunta al usuario si realmente desea eliminar la calibración, una vez confirmada la eliminación, el sistema refresca la lista de calibraciones.

#### 4.5.6 Cerrar

Nos permite salir de la ventana de calibraciones.

### 4.6 Plan de Calibraciones

Al dar clic en plan de calibraciones muestra 2 botones adicionales los cuales son el de plan de calibraciones interno y externo. Al presionar el botón de calibraciones interno se muestra un calendario de calibraciones programadas 1 año antes y un año después del día que se haga la consulta. Esto es con el fin de que el laboratorista pueda ver la carga de trabajo que tiene para cada mes del año en curso y del próximo año así como terminar calibraciones pendientes que se debieron hacer anteriormente.



Fig. 4.12 Plan de calibración Interno.

Para ver el plan de calibración interno seleccionar el Plan de calibración, después el sistema mostrará dos opciones disponibles interno y externo

- Interno: Muestra el calendario de calibraciones programadas, pero solo para instrumentos que pertenecen a la planta
- Externo: Muestra el calendario de calibraciones programadas, solo se muestran los instrumentos de clientes externos

Este control tiene la capacidad de mover las calibraciones de un mes a otro dependiendo si el usuario así lo desea o si hay necesidad de reprogramar calibraciones.

Como podemos ver la ventana plan de calibración tiene consigo un grupo de opciones las cuales son:

- Imprimir plan de calibración
- Mover calibraciones
- Ver calibraciones
- Guardar cambios
- Regresar

#### 4.6.1 Imprimir Plan de Calibración

Genera un reporte de planes de calibración, con diferentes criterios de búsqueda.

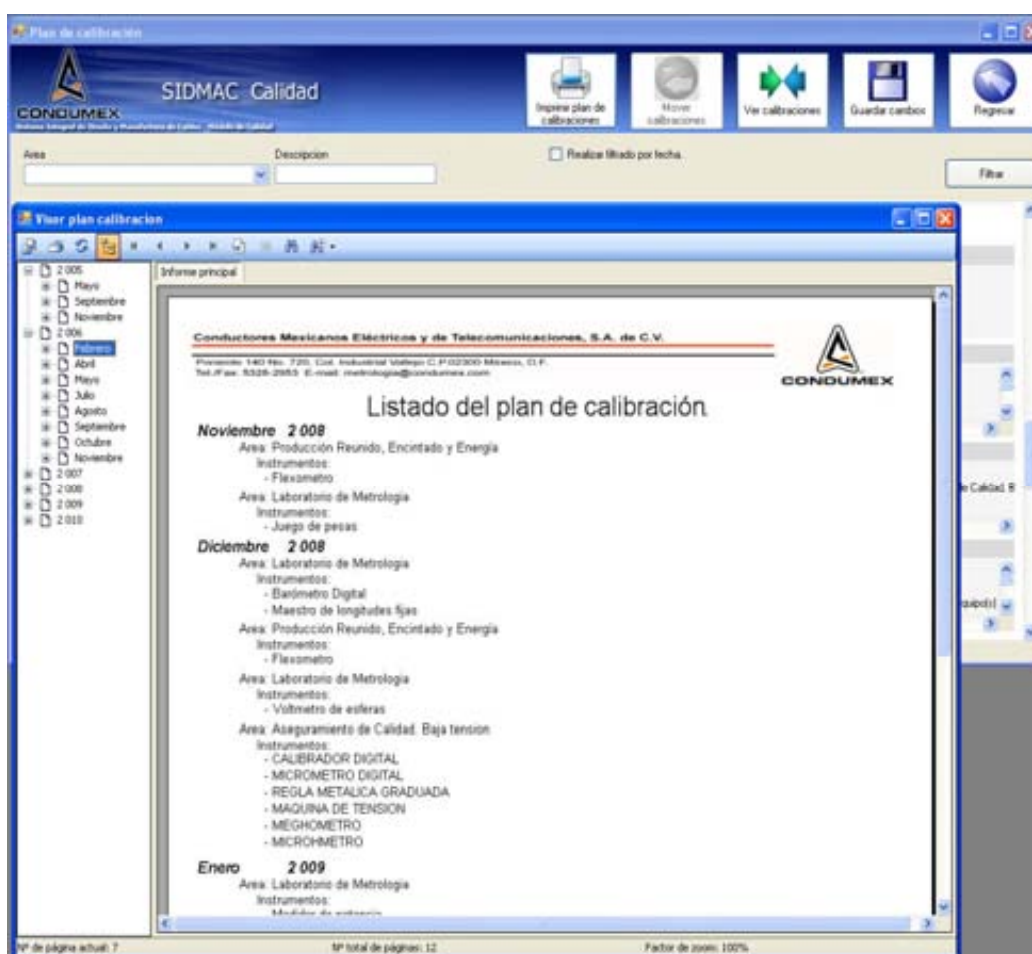


Fig. 4.13 Reporte de plan de calibración

Al dar clic en imprimir plan de calibraciones el sistema muestra un control, en el cual permite seleccionar la fecha inicial y la fecha final, que se desea para generar el reporte. El reporte se puede imprimir o bien generar el reporte archivo pdf para ser enviado vía correo electrónico si así se desea.

#### **4.6.2 Mover Calibraciones y Guardar Cambios**

Esta opción permite al usuario cambiar la programación de las calibraciones manualmente, funciona de tal forma que al dar clic sobre el botón mover calibraciones, todas las calibraciones pueden ser arrastradas de un mes a otro. Una vez terminando con los movimientos de calibraciones. Se presiona guardar cambios para actualizar las fechas de las calibraciones.

El objetivo de esta funcionalidad es el poder ver un calendario de calibraciones programadas por mes, ya que se puede ver la carga de trabajo para el laboratorio, y así de forma más amplia poder hacer una mejor programación de actividades

#### **4.6.3 Ver Calibraciones**

Con esta opción se pueden ver las calibraciones que se modificaron y la nueva fecha que se programo la ventana que lanza esta opción es igual que la figura 4.8 de este capítulo.

#### **4.6.4 Regresar**

La opción de regresar simplemente cierra la ventana de plan de calibración ya sea interno y externo. Y nos regresa a la pantalla inicial o en este caso a nuestra forma padre.

### 4.7 Informes de Calibración

Ahora pasaremos a describir el funcionamiento de una de las partes más importantes del sistema, que es la creación de los informes de calibración. Principalmente vamos a conocer la ventana mediante la cual podremos seleccionar diferentes tipos de informes. Ver Fig. (4.14)

Ahora pasaremos a describir el funcionamiento de una de las partes más importantes del sistema, que es la creación de los informes de calibración.

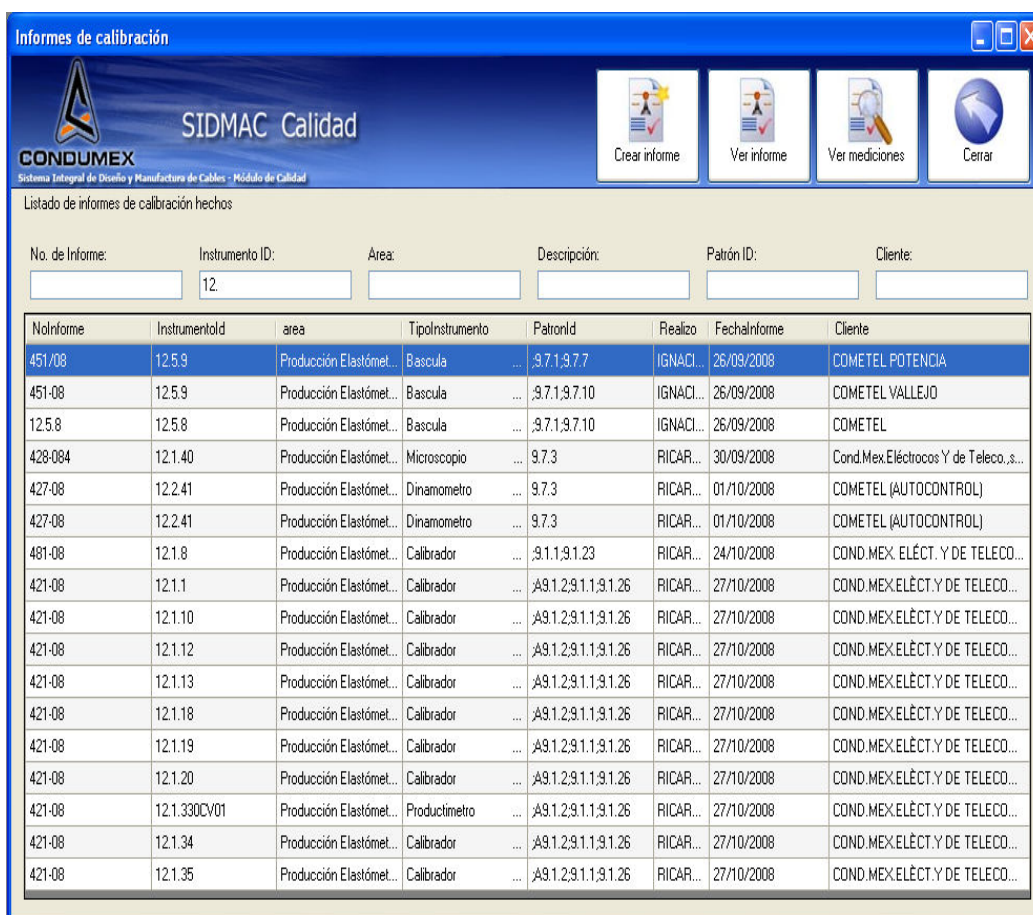


Fig. 4.14 Ventana inicial de informes de calibración

Como podemos ver en esta ventana se tienen 4 opciones las cuales el usuario puede disponer de ellas en cualquier momento. Estas opciones son:

- Crear informe
- Ver informe

- Ver mediciones
- Cerrar

### 4.7.1 Crear Informe

Esta opción permite al usuario crear un nuevo informe de calibración para uno o varios instrumentos.

Para crear un nuevo informe de calibración, primero dar clic sobre el botón de crear informe, en este momento el sistema mostrará al usuario una ventana la cual deberá ser llenada por el usuario ver figura 4.15

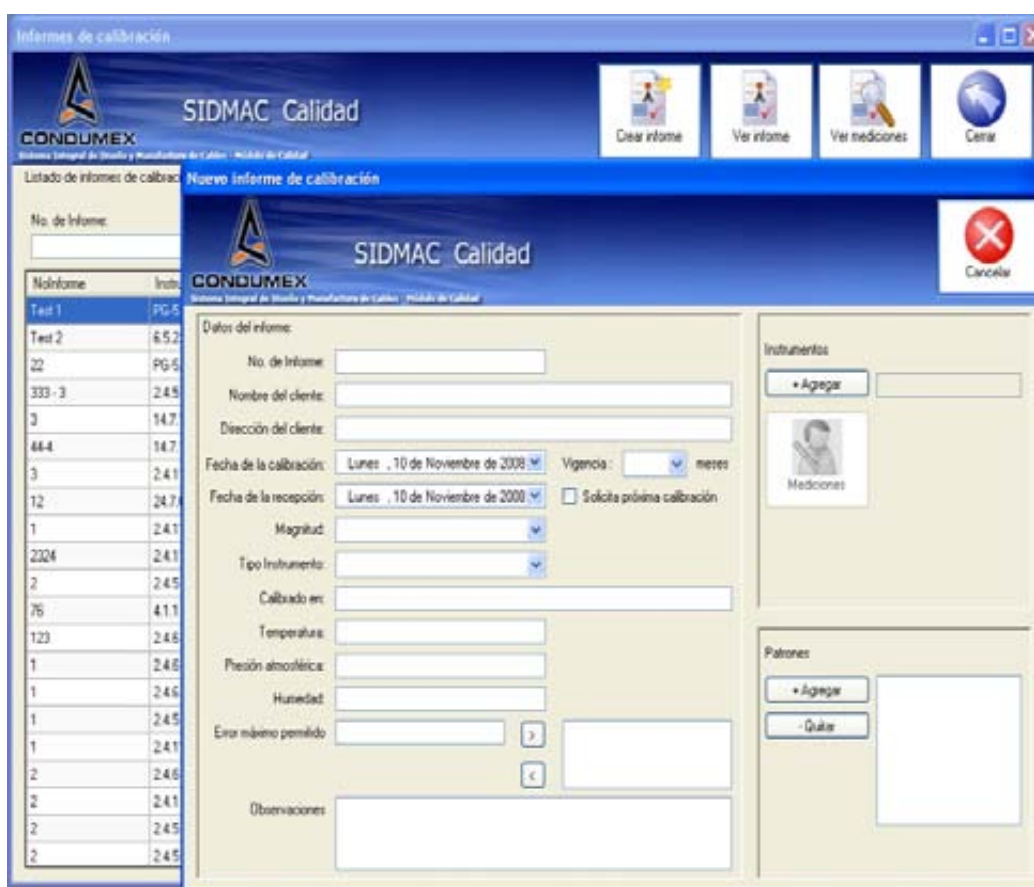


Fig. 4.15 Ventana de creación de nuevo informe de calibración.

El usuario será el encargado de llenar la información, estos campos se llenarán dependiendo de la información que cuente el usuario. Para llevar a cabo un mejor orden en la creación del informe llevar a cabo los siguientes pasos:

### Pasos para la creación de informes de calibración

1. Llenar el apartado de “Datos del informe” que se encuentra de lado izquierdo, en donde se pueden ver campos como no informe, magnitud, tipo de instrumento de entre los más importantes.
2. Para seleccionar la magnitud dar clic sobre el campo de magnitud, aquí se despliega una lista de magnitudes disponibles las cuales son:
  - Dimensional
  - Fuerza
  - Velocidad
  - Química
  - Masa
  - Temperatura
  - Eléctrica
3. Una vez seleccionada la magnitud, el sistema da al usuario la opción de seleccionar un instrumento que pertenezca a la magnitud seleccionada. Por ejemplo al seleccionar la magnitud *Fuerza*. El control de tipo de instrumento se llena con los instrumentos que pertenecen a Fuerza en este caso son:
  - Dinamómetro
  - Máquina de tensión con celda
  - Máquina de tensión con anillo
  - Máquina de tensión con pesas

Una vez seleccionado el instrumento ahora ya sabemos qué tipo de informe vamos a crear

4. Ahora asociaremos el informe a un instrumento. En la parte superior derecha tenemos un apartado llamado Instrumentos. Este es el encargado de asociar el informe a un instrumento que tengamos registrado en el sistema. Para llevar a cabo la asociación, dar clic sobre el botón agregar, en este instante el sistema mostrará



una ventana con la lista de instrumentos registrados en el sistema, aquí el usuario selecciona el instrumento deseado Ver (Fig. 4.16).

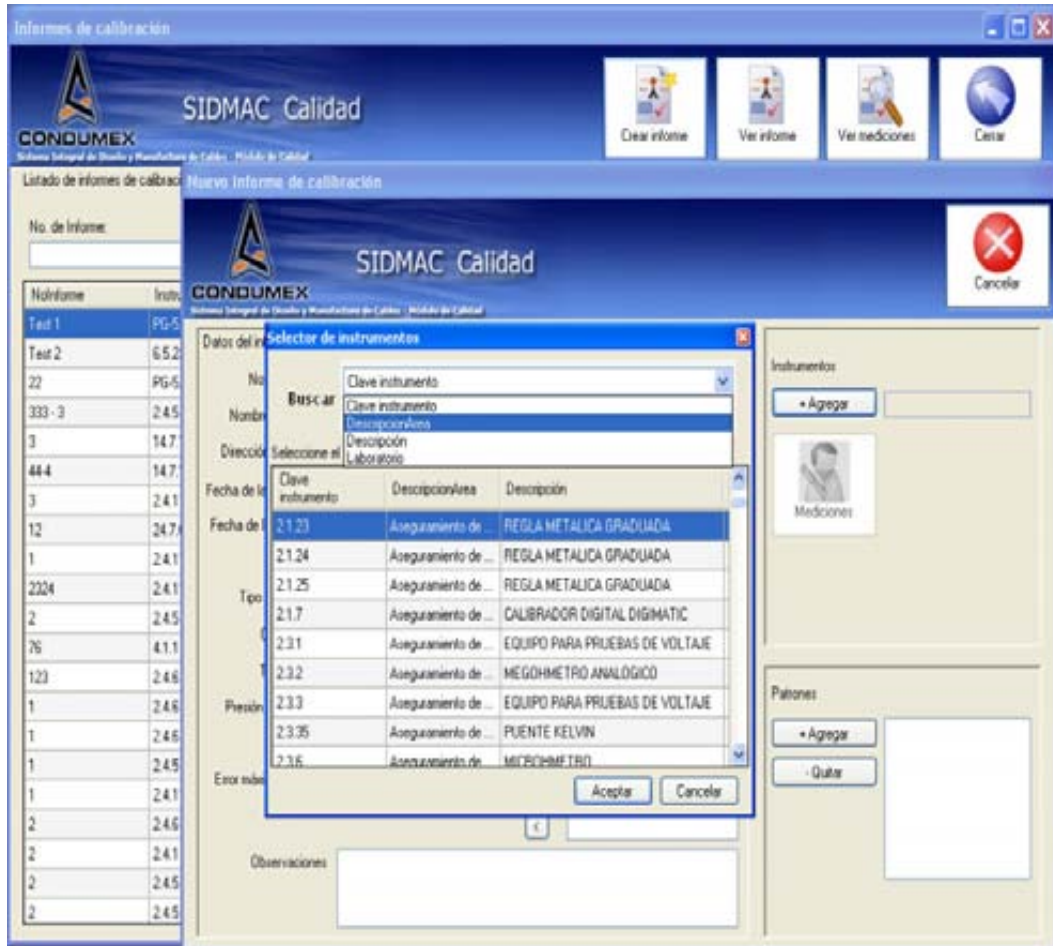


Fig. 4.16 Lista de instrumentos registrados en el sistema

NOTA: el instrumento todavía no se asocia al informe.

5. De igual forma asociaremos patrones al informe de calibración los cuales servirán como auxiliares para la calibración correcta del instrumento. El procedimiento para agregar patrones es muy similar al de agregar instrumentos al informe. En la parte inferior derecha de la ventana se encuentra un apartado llamado “Patrones” para agregar patrones al informe basta con dar clic sobre agregar, aquí el sistema volverá a mostrar la lista de patrones registrados en el sistema, el usuario puede seleccionar hasta cinco patrones. Los patrones conforme se vayan agregando se mostrarán en el cuadro de texto que se encuentra al lado de los botones agregar y quitar. Para quitar

algún patrón de la lista basta con seleccionarlo y dar clic sobre el botón de quitar, aquí el sistema quitara el patrón de la lista. Una vez completada la información el usuario puede agregar las mediciones correspondientes al instrumento que selecciono.

- Para agregar las mediciones correspondientes al informe, identificar nuevamente el apartado de “Instrumentos” allí aparece un botón con una etiqueta “Mediciones” al dar clic sobre este botón, el sistema mostrará una hoja de cálculo parecida a una hoja de Excel (Ver Figura 4.17), en la que el usuario podrá introducir los valores necesarios. Esta hoja de cálculo contiene fórmulas y campos bloqueados, es decir que el usuario no podrá editar las fórmulas... Este fue una petición del jefe de aseguramiento de calidad, con el fin de que el laboratorista no pueda modificar datos

The screenshot shows the 'SIDMAC Calidad' software interface. The top section is a form for 'Datos del informe' (Report Data) with fields for 'No. de Informe' (1254A), 'Nombre del cliente' (Conticon), 'Dirección del cliente' (Carretera panamericana), 'Fecha de la calibración' (07 de Octubre de 2000), 'Vigencia' (12 meses), and 'Fecha de la recepción' (02 de Septiembre de 2000). There is a 'Mediciones' button and a 'Cancelar' button.

The bottom section is a spreadsheet titled 'HOJA DE CÁLCULO PARA DIBIOMETRO'. It contains several tables and formulas:

**ACELERACION LOCAL DE LA GRAVEDAD**  
 $G1 = G * (1 + b1 * Sen^2O - b2 * Sen^22O) - 3,086 * 10^{-6} * H$

VALOR DEL	LECTURA DEL	ERROR %	ERROR %	ERROR %	ERROR %	U/DEL	U/DE	U/DE
PATRON	INSTRUMENTO	EXACTITUD	REPETIBILIDAD	CERO	REVERSIBILIDAD	PATRON	MECCION	MECCION
N	N	%	%	%	%	N	N	kg
120.45	44.3	-65.494	0.68	0	0.23	0.000814	0.2345	0.02391
257.01	86.3	-65.663	0.45	0.00	0.06	0.000814	0.3339	0.03404
385.61	131.9	-65.766	0.30	0.00	0.30	0.000814	0.3537	0.03606
514.20	176.2	-65.727	0.23	0.00	0.23	0.000824	0.3190	0.03252
642.73	220.7	-65.657	0.45	0.00	0.45	0.000813	0.7688	0.07940


**BITACORA DE MEDICIONES**

INCERT. A	LECTURA Patrón	LECTURA Inst. Asc.	LECTURA Inst. Des.	LECTURA Inst. Asc.	LECTURA Patrón
kg	N	N	N	N	kg
1.16E-01	44.4634	44.20	44.30	44.50	4.53401
1.66E-01	86.9281	86.10	86.15	86.50	9.06916
1.76E-01	133.4349	131.00	132.20	131.80	13.60657
1.59E-01	177.9517	176.00	176.40	176.30	16.14399
3.84E-01	222.4050	220.20	221.20	220.80	22.67900

Fig. 4.17 Hoja de cálculo para el tipo de instrumento seleccionado

7. Como se puede ver la ventana de mediciones tiene las siguientes opciones. Guardar – Guarda la hoja de cálculo y la asocia al instrumento. Para consulta o generación futura del informe. Y regresa a la ventana de creación del informe.
- Salir – nos regresa a la ventana de crear informe, no asocia las mediciones al instrumento.
- Para continuar con la creación del informe, tenemos que guardar nuestras mediciones. Para esto solo basta con dar clic sobre el botón de guardar. Una vez guardadas las mediciones estarán asociadas al instrumento, y el sistema nos regresara a la ventana de creación de informe.
8. Al estar nuevamente en la ventana de nuevo informe de calibración, y haber capturado la información para el informe, muestra las siguientes opciones: (Ver figura 4.18)
- Guardar y continuar: Guarda el informe, lo asocia a un instrumento y limpia la ventana para registrar un nuevo informe de calibración
  - Guardar: Solo guarda el informe, lo asocia a un instrumento y cierra la ventana de nuevo informe de calibración.
  - Ver informe: genera una vista del informe que se creó. El cual se puede imprimir o bien generarlo como archivo pdf. Ya sea para enviarlo vía correo electrónico u otros usos.
  - Cancelar: como su nombre lo indica, cancela la creación del nuevo informe de calibración. Simplemente cierra la ventana de nuevo informe.
9. Para guardar el informe y asociarlo al instrumento basta con dar clic sobre guardar o en guardar y continuar. En funcionamiento hacen casi lo mismo guardar el informe y asociarlo a un instrumento. Solo que guardar cierra inmediatamente la ventana de nuevo informe de calibración a diferencia de guardar y continuar que guarda el informe, lo asocia al instrumento y limpia los campos para registrar nuevo informe de calibración.
10. Para ver el informe de calibración que se generó de acuerdo al instrumento correspondiente, dar clic sobre el botón de ver informe. El sistema mostrará el informe de calibración (Ver Fig. 4.19).

**Nuevo informe de calibración**

 **SIDMAC Calidad**  
Sistema Integral de Diseño y Manufactura de Cables - Módulo de Calidad

**Datos del informe:**

No. de Informe:

Nombre del cliente:

Dirección del cliente:

Fecha de la calibración:  Vigencia:  meses

Fecha de la recepción:   Solicita próxima calibración

Magnitud:

Tipo Instrumento:

Calibrado en:

Temperatura:


Presión atmosférica:

Humedad:

Error máximo permitido:

Observaciones:

**Instrumentos**

 Mediciones

**Patrones**

Fig. 4.18 Ventana de nuevo informe de calibración para guardar y generar informe

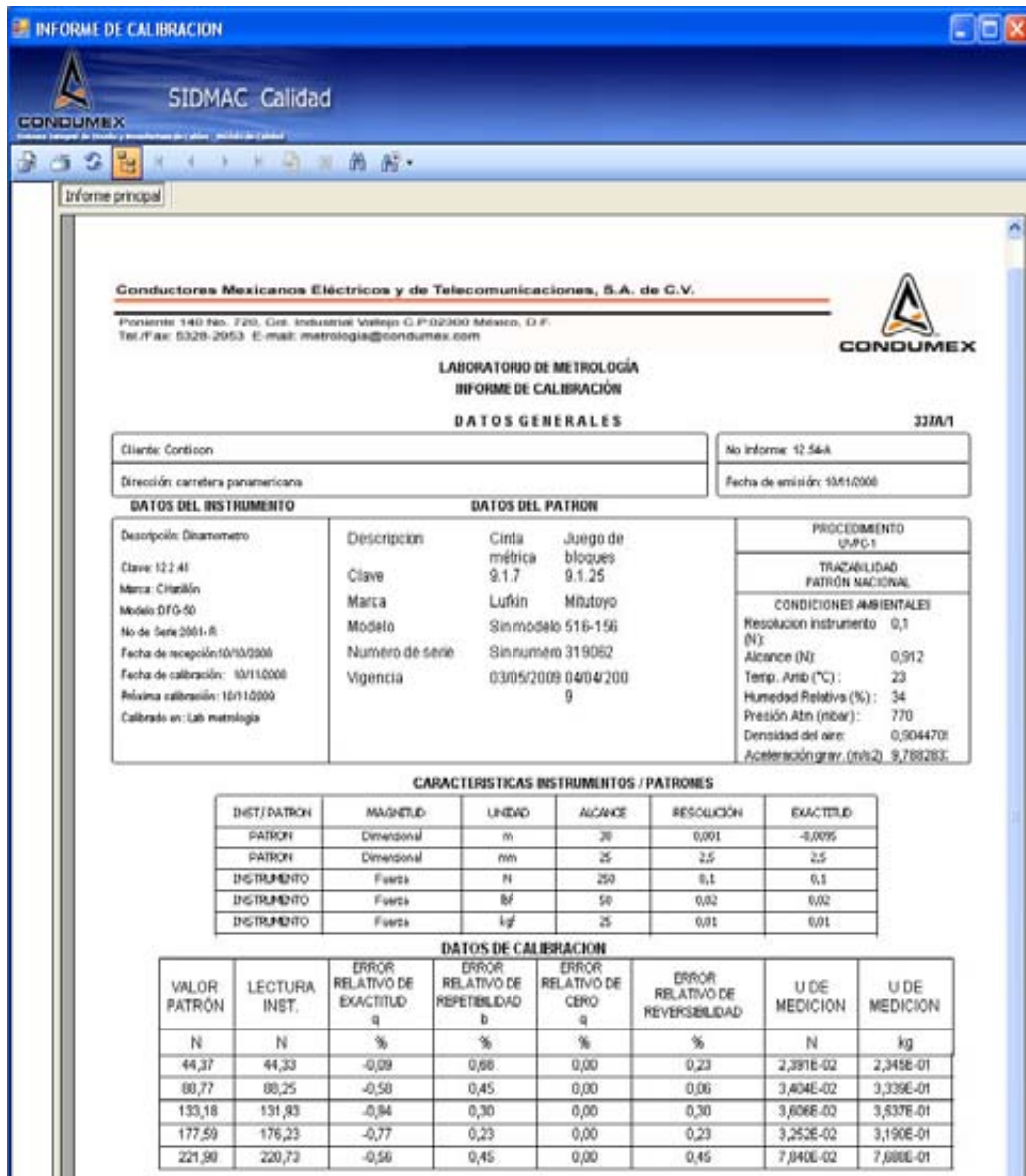


Fig. 4.19 Generación de Informe de calibración.

De esta manera se ha creado el informe de calibración el cual avala la calibración del instrumento en cuestión. Este informe se hizo bajo las normas y procedimientos de las organizaciones que rigen el cumplimiento de calibraciones entre ellas principalmente (CENAM Centro nacional de metrología).

Regresando a la ventana de informes de calibración, donde muestra todos los informes que se han registrado en el sistema, se tiene otras opciones aparte de crear informe que son:

#### **4.7.2 Ver Mediciones**

Esta opción nos permite ver los valores que el usuario registró y los cálculos que el sistema arrojó, por razones de validación esta hoja solo es de consulta. Una vez las mediciones registradas en el sistema solo son para consulta y generación de informes. Por razones de confiabilidad no se pueden editar.

#### **4.7.3 Ver Informe.**

Esta opción vuelve a generar el informe de calibración ya sea para consulta simplemente o para exportarlo a .pdf, ó para enviarlo por email.

Para esto solo se debe seleccionar un informe de la lista de informes y se presiona el botón de ver informe.

Es así como terminamos, este trabajo de tesis, basado principalmente en un sistema que fue desarrollado e implementado para condumex cables en la ciudad de México, Df. En el año 2008, para el laboratorio de metrología de condumex..

### **Conclusiones**

Habiendo realizado el estudio anterior utilizando la metodología de programación orientada a objetos, se desarrolló el sistema de administración de metrología para la empresa Condumex cables de la ciudad de México, y se llegó a las siguientes conclusiones:

PRIMERA: Cabe resaltar que con el uso del sistema la productividad del laboratorio ha sido notable, ésto fácilmente se puede observar al consultar los planes de calibración o bien las calibraciones que se han realizado.

SEGUNDA: Se cumplió con el requisito de encontrar los instrumentos problema y es que con los reportes de comportamiento de instrumentos es fácil determinar los que están causando problemas o bien que ya no cumplen con ciertas normas y especificaciones.

TERCERA: La generación de informes de calibración se hace de forma fácil y el usuario los puede tener a la mano de forma fácil, ya sea para consultarlos, imprimirlos o bien para enviarlos por email, ya que se generan en formato .pdf.

CUARTA: La programación de calibraciones es una de las más importantes funcionalidades ya que gracias a ésta, el sistema envía notificaciones a los laboratoristas avisando que existen calibraciones pendientes las cuales deben ser atendidas o bien reprogramadas.

QUINTA: El uso de Xquerys en SQL server facilita y agiliza las consultas sobre los datos XML, e Integrado con el Visual Studio .NET es una herramienta que vale la pena sacar el mayor provecho.

---

## Bibliografía

1. Ann L. Winblad, *Software orientado a objetos*, Ediciones, Díaz de Santos, 1993, Pág. 31.
2. Eric Freeman, Kathy Sierra, Bert Bates, *Head First Design Patterns*, Editorial O'Reilly, 2004, PAG 44.
3. Jesse Liberty, Dave Grundgeiger, *Programming Visual Basic .NET*, Edition 2 illustrated, Editorial O'Reilly, 2003, Pág. 4.
4. José A. Cerrada, José Antonio Cerrada Somolinos, Manuel Collado Machuca, Manuel Collado, Sebastián Gómez Palomo, José Félix Estivaliz López, *Introducción a la ingeniería del software*, Editorial Ramón Areces, 2000.
5. Thierry Groussard, Miguel Jiménez Antelo, *Visual Basic 2005(VB.NET): Programe con Visual Studio 2005*, Ediciones ENI, 2006, Pág. 153.
6. Thierry Groussard, Xavier Angelet, *Vb.net*, Ediciones ENI, 2003, Pág. 87.

## Internet

1. <http://msdn.microsoft.com/es-es/library/>  
MSDN Library - Herramientas y lenguajes de desarrollo - Documentación de visual studio - Introducción a Visual Basic.
2. <http://msdn.microsoft.com/es-es/library/>  
MSDN Library - Herramientas y lenguajes de desarrollo - Documentación de visual studio - Visual Basic.
3. <http://msdn.microsoft.com/es-es/library/>  
MSDN Library - SQLServer 2005 - Documentación de SQL Server 2005 - Tutoriales de SQL Server 2005.
4. <http://msdn.microsoft.com/es-es/library/>  
MSDN Library - Herramientas y lenguajes de desarrollo - Visual studio 2005 - Documentación de Visual studio - Programación de .net Framework - Documentos y datos XML
5. <http://www.condumex.com.mx>