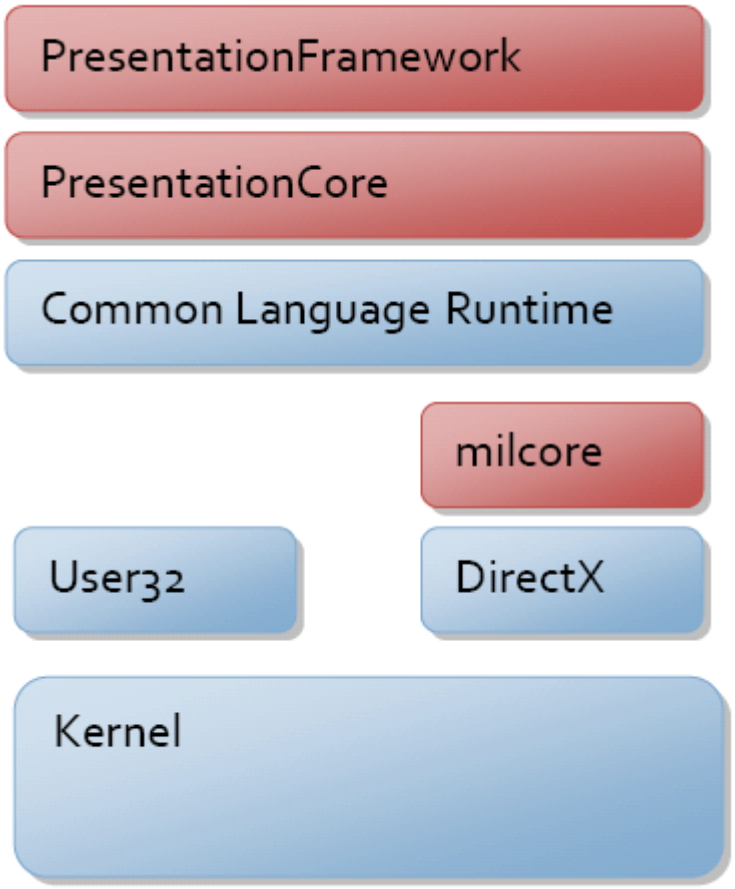
CAPÍTULO III



RESUMEN

Después de hablar de las tecnologías que un lenguaje de programación tiene, mencionaremos la estructura con la que la tecnología tiene y esta es la arquitectura, dicha arquitectura mensiona todos los módulos que manejan y los procesos que manipulan, desde el manejo de los dispositivos de la máquina hasta el modo de compilar el código que los programadores han implementado en las aplicaciones.

La tecnología WPF ofrece una amplia infraestructura y potencialidad gráfica con la que se podrán desarrollar aplicaciones de atractiva apariencia manejando con gran potencialidad la imagen y el video, con esto podemos

decir que las aplicaciones desarrolladas necesitaran más requerimiento de tarjeta de video y asi sacar todo el potencial que esta tecnología puede dar. Cuando hablamos del requerimiento de la parte gráfica también damos a conocer el runtime de ejecución que se maneja, el mismo que es el DirectX.

También daremos a conocer las diferentes arquitecturas de las bases de datos que se manejan en la actualidad, así como también conoceremos las más óptimas para el acoplamiento con la tecnológia WPF. El manejo de las bases de datos también tienen su estudio para poder definir lo mejor y las ventajas que cada una puede propinar, ADO DB es una arquitectura que se acopla con el lenguaje de programación .NET y esta a la vez se acopla con la tecnología WPF.

Tanto el ODBC como el OLE DB tienen la ventaja de que pueden ser utilizados por el cliente. Ambas tecnologías tienen una capacidad completamente diferente. ODBC estaba primeramente diseñada para el uso relacionado con data. De cualquier manera, data no guarda relación con los demás formatos. En relación a los nuevos formatos de data, ésta tiene nuevos lugares, como Internet.

Conexión de Acceso Cliente Servidor

Finalmente, el marco de Microsoft Component Object Model (COM) nos ayudara a la conexión y acceso entre cliente servidor llevando la información de un lugar a otro de una forma segura, la idea de usar una buena arquitectura de comunicación de información es importante ya que existen aplicaciones que requieren de toda la seguirar a la hora de transmitir los datos.

Los Objetos COM pueden ser instanciados y referenciados en un proceso, a través de las fronteras de un proceso dentro de equipo y, a través de una red, usando la tecnología DCOM. Salir del proceso y de los objetos remotos puede utilizar serialización para enviar las llamadas a los métodos y valores de retorno hacia atrás y hacia delante. La serialización es invisible para el objeto y el código usando el objeto.

Acceso a Datos (OLE DB)

En términos simples, OLE DB es una sucesión de la tecnología estándar de Open Database Connectivity (ODBC). OLE DB es una interfaz de un bajo nivel de juego para trabajar con data.
Tanto el ODBC como el OLE DB tienen la ventaja de que pueden ser utilizados por el cliente. Ambas tecnologías tienen una capacidad completamente diferente.

ODBC estaba primeramente diseñada para el uso relacionado con data. De cualquier manera, data no guarda relación con los demás formatos. En relación a los nuevos formatos de data, ésta tiene nuevos lugares, como Internet. Finalmente, el marco de Microsoft Component Object Model (COM) requiere una mejor tecnología de acceso. Aclarando un poco esta cuestión, ODBC no tiene direcciones nuevas; es una nueva tecnología lo que es necesario.

3. Arquitecturas usadas por WPF

3.1. Descripción de las arquitecturas compatibles para la tecnología WPF

Para la revisión de la este tema dividiremos la arquitectura en dos puntos, el uno será la arquitectura de desarrollo y la otra será la arquitectura de base de datos o acceso a datos.

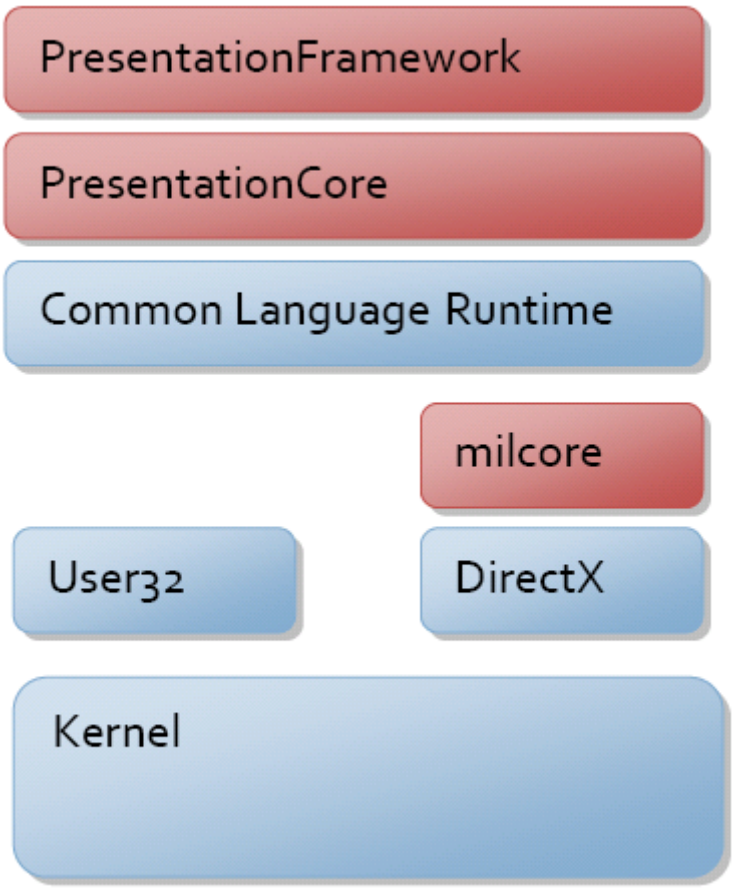


Figura3.1: Arquitecturas compatibles par la tecnología WPF Ref: http://msdn.microsoft.com/es-es/library/ms750441.aspx

PresentationFramework.- Se aplica al usuario final elementos de presentación, incluyendo los diseños, dependiente del tiempo, animaciones basadas en efectos y escenas, y enlace de datos. Ofrece una amplia infraestructura y potencialidad gráfica con la que se podrán desarrollar aplicaciones de excitante y atractiva apariencia, con mayores y más

funcionales facilidades de interacción que incluyen animación, vídeo, audio, documentos, navegación, gráfica 3D. Separa, con el lenguaje declarativo XAML y los lenguajes de programación de .NET, la interfaz de interacción de la lógica del negocio, propiciando una arquitectura Modelo Vista Controlador para el desarrollo de las aplicaciones.

PresentationCore.- Proporciona un contenedor administrado para MIL e implementa los servicios básicos de WPF, incluyendo un sistema de propiedad que es consciente de las dependencias entre los organismos de elaboración y de los consumidores de los bienes, el envío de un mensaje del sistema por medio de un objeto Dispatcher implementar un sistema de eventos y servicios especializados que pueden poner en práctica un sistema de diseño, como la medición de elementos de la interfaz.

Common Language Runtime.- o CLR (Lenguaje común en tiempo de ejecución) es el componente de máquina virtual de la plataforma .Net de Microsoft. Es la implementación del estándar Common Language Infrastructure (CLI) que define un ambiente de ejecución para los códigos de los programas. El CLR ejecuta una forma de código intermedio (bytecode) llamada Common Intermediate Language (CIL, anteriormente conocido como MSIL Microsoft Intermediate Language), la implementación de Microsoft del CLI.

Los desarrolladores que usan CLR escriben el código en un lenguaje como C# o VB.Net. En tiempo de compilación, un compilador.NET convierte el código a MSIL (Microsoft Intermediate Language). En tiempo de ejecución, el compilador en tiempo de ejecución (Just-in-time compiler) del CLR convierte el código MSIL en código nativo para el sistema operativo. Alternativamente, el código MSIL es compilado a código nativo en un proceso separado anterior a la ejecución. Esto acelera las posteriores ejecuciones del software debido a que la compilación de MSIL a nativo ya no es necesaria.

Milcore.- Es el componente que interactua directamente con Direct-X y ofrece soporte básico para las superficies en 2D y 3D, temporizador de control de manipulación de los contenidos de una superficie con el fin de exponer la

animación construye en un nivel superior, y la composición de los elementos individuales de una aplicación WPF 3D en una final " escena "que representa la interfaz de usuario de la aplicación y la vuelve a la pantalla. Los códecs multimedia, también se aplican en el código no administrado.

User32.- Es la plataforma donde se va a ejecutar el sistema desarrollado, no es que las aplicaciones se limitan a esta plataforma si no que está abierta para poder ejecutarse en una de 64 sin ningún inconveniente. Dejando al Kernel la limitación de la resolución y el potencial que le dará a la máquina para su correcta respuesta.

3.1.1. Arquitectura de Desarrollo

Entre las arquitecturas de desarrollo tenemos la arquitectura de Windows forms.

Las clases del espacio de nombres **System.Windows.Forms** agrupadas en categorías:

Control, control de usuario y formulario: La mayoría de las clases del espacio de nombres *System.Windows.Forms* derivan de la clase *Control*. La clase *Control* proporciona la funcionalidad base de todos los controles que se muestran en un objeto *Form*. La clase *Form* representa una ventana dentro de una aplicación. Incluye cuadros de diálogo, ventanas no modales y ventanas cliente y principal de la Interfaz de documentos múltiples (MDI). También puede crear sus propios controles derivando de la clase UserControl.

Menús y barras de herramientas: Windows.Forms contiene un amplio conjunto de clases para que pueda crear sus propias barras de herramientas y menús personalizadas, con un aspecto y comportamiento modernos (apariencia y percepción). Las clases ToolStrip, MenuStrip, ContextMenuStripy StatusStrip se pueden utilizar para crear barras de herramientas, barras de menús, menús contextuales y barras de estado, respectivamente.

Controles: El espacio de nombres System.Windows.Forms dispone de diferentes clases de controles que se pueden utilizar para crear interfaces de

usuario completas. Algunos controles están diseñados para la entrada de datos en la aplicación, por ejemplo, los controles TextBox y ComboBox. Otros controles muestran datos de la aplicación, por ejemplo, Label y ListView. El espacio de nombres también dispone de controles para invocar comandos en la aplicación, por ejemplo, Button. El control WebBrowser y las clases HTML administradas, como HtmlDocument, permiten mostrar y manipulan páginas HTML desde su propia aplicación administrada de formularios Windows Forms. El control MaskedTextBox es un control de entrada de datos avanzado que permite definir máscaras que aceptan o rechazan automáticamente los datos proporcionados por el usuario. Además, puede utilizar el control PropertyGrid para crear su propio Diseñador de Windows Forms que muestra las propiedades visibles de diseñador de los controles.

Diseño: Hay varias clases importantes en Windows.Forms que ayudan a controlar el diseño de los controles en una superficie de presentación, como un formulario o control.FlowLayoutPanel dispone en serie todos los controles que contiene y TableLayoutPanel permite definir celdas y filas para disponer los controles en una cuadrícula fija. SplitContainer divide la superficie de presentación en dos o más partes ajustables.

Datos y enlace de datos: Windows.Forms define una arquitectura enriquecida para enlazar a los orígenes de datos como las bases de datos y los archivos XML. El control DataGridView proporciona una tabla personalizable para mostrar los datos y le permite personalizar celdas, filas, columnas y bordes. El control BindingNavigator representa una forma estandarizada de explorar datos y trabajar con ellos en un formulario; BindingNavigator se empareja frecuentemente con el control BindingSource para recorrer los registros de datos de un formulario e interactuar con ellos.

Componentes: Además de los controles, el espacio de nombres System.Windows.Forms proporciona otras clases que, aunque no se derivan de la clase Control, también aportan características visuales a las aplicaciones basadas en Windows. Algunas clases, como ToolTip y ErrorProvider, amplían las capacidades o proporcionan información al usuario. Con las clases Help y

HelpProvider, puede mostrar información de Ayuda al usuario de sus aplicaciones.

Cuadros de diálogo comunes: Windows proporciona varios cuadros de diálogo comunes que se pueden utilizar para ofrecer a la aplicación una interfaz de usuario coherente a la hora de realizar tareas como abrir y guardar archivos, manipular la fuente o el color del texto o imprimir. Las clases OpenFileDialog y SaveFileDialog proporcionan la funcionalidad para mostrar un cuadro de diálogo que permita al usuario buscar o escribir el nombre del archivo que desea abrir o guardar. La clase FontDialog muestra un cuadro de diálogo para cambiar los elementos del objeto Font que utiliza la aplicación. Las clases PageSetupDialog, PrintPreviewDialog y PrintDialog muestran cuadros de diálogo que permiten al usuario controlar la impresión de documentos. Para obtener más información sobre cómo imprimir desde una aplicación basada en Windows, vea el espacio de nombres System.Drawing.Printing. Además de los cuadros de diálogo comunes, el espacio de nombres System.Windows.Forms dispone de la clase MessageBox para mostrar un cuadro de mensaje que pueda presentar datos del usuario y también recuperarlos de éste.

3.1.2. Arquitectura de Base de Datos

Las arquitecturas de conexión de datos más destacados en la actualidad son las que mencionamos y explicamos a continuación.

3.1.2.1. ¿Por qué usamos la tecnología ADO (Access Data Object)?

ActiveX Data Objects (ADO) es actualmente la más novedosa y atractiva tecnología de Microsoft. ADO está concebido con data, que es una tecnología especialmente interesante para desarrolladores.

Si hablamos de ADO, estamos haciéndolo sobre dos elementos diferentes: el Actives data objects en sí mismo y sobre la tecnología de Microsoft Universal Data, a parte también se incorpora el OLE DB.

3.1.2.2. OLE DB y Universal Data Access

En términos simples, OLE DB es una sucesión de la tecnología estándar de Open Database Connectivity (ODBC). OLE DB es una interfaz de un bajo nivel de juego para trabajar con data.

Tanto el ODBC como el OLE DB tienen la ventaja de que pueden ser utilizados por el cliente. Ambas tecnologías tienen una capacidad completamente diferente. ODBC estaba primeramente diseñada para el uso relacionado con data. De cualquier manera, data no guarda relación con los demás formatos. En relación a los nuevos formatos de data, ésta tiene nuevos lugares, como Internet. Finalmente, el marco de Microsoft Component Object Model (COM) requiere una mejor tecnología de acceso. Aclarando un poco esta cuestión, ODBC no tiene direcciones nuevas; es una nueva tecnología lo que es necesario. La tecnología es OLE DB, y esto comienza aquí.

Los clientes pueden trabajar directamente con OLE DB o pueden hacerlo con él a través del interfaz de ADO (esto último es lo más frecuente). OLE DB tiene acceso a cualquiera de estos dos sistemas, ya sea a través de SQL data o a través de ODBC. OLE DB proveedor viene provisto de un acceso directo con OLE DB. Por lo cual OLE DB puede ser usado con una gran variedad de SQL data, además de que data existe en ordenadores centrales. La facilidad de acceso de data a través de interfaces comunes, sin recordar la localización o estructura de data, son realmente el poder entre ADO y OLE DB.

Mientras ODBC usa drivers, OLE DB usa proveedores. Un proveedor es un software provisto de un tipo específico de data emparejado con el OLE DB especificado. Varios OLE DB proveedores que existen hoy en día, están incluidos en Microsoft SQL Server y Oracle. El uso de ODBC está muy extendido, por eso el OLE DB proveedor para ODBC tiene creado un orden de facilidad para la migración de ODBC hasta OLE DB.

Actualmente varios proveedores están siendo desarrollados. Quizás el más desarrollado de ellos actualmente es el OLE DB proveedor para Microsoft Outlook. El MS Remote es un proveedor especial que permite a data el acceso

directo a Internet. Este breve aquí expuesto hace referencia a una tercera parte de los proveedores comunes comprometidos con OLE DB, además, hay una gran cantidad de nuevos proveedores en desarrollo.

ADO.- OLE DB es un conjunto de interfaces de bajo nivel provistas de un acceso a data con una gran variedad de formatos y localizaciones. Aunque poderosas, las interfaces de OLE DB pueden ser muy voluminosas para trabajar con ellas directamente. Afortunadamente, ADO está provisto de un conjunto de alto nivel, un desarrollador con interfaces que hacen que trabajar con OLE DB y universal data access sea relativamente simple. La interfaz usada es ADO, que se turna en uso con OLE DB.
ADO a sí mismo es un conjunto de objetos. Por sí mismo, ADO no es capaz de nada. Para poder funcionar, ADO necesita de los servicios de OLE DB proveedor. El proveedor se turna para usar el nivel bajo de la interfaz de OLE DB con access para trabajar con data. Algunos ADO usan una conexión con SQL Server OLE DB proveedor y otros ADO utilizan una conexión con Oracle OLE DB proveedor. Aunque la interfaz es constante, la compatibilidad puede ser muy diferente porque los OLE DB proveedores son muy diferentes, en el cual el punto culminante es el polimorfismo natural del OLE DB.
Como desarrolladores, anhelamos la consistencia. ADO nos provee de una interfaz consistente para los códigos de nuestro programa.
La versión actual de ADO (2.1) es la cuarta versión de ADO que ha salido en los últimos dos años. ADO 1.0 fue primeramente limitado para trabajar con páginas Active Server. Sólo existía un OLE DB proveedor, era el OLE DB proveedor para ODBC Drivers.

ADO (2.5) – Compatibiliza con la nueva versión de Microsoft Windows 2000.

ADO (2.1) Compatibiliza con la nueva versión de Microsoft Web browser, Internet Explorer 5.0. Si se discute sobre dato o algo relacionado con Internet, lo más posible es que se mencione XML. XML, el Extensible Markup Language es un lenguaje mark-up que nos permite crear etiquetas tal y como estamos acostumbrados con data. XML está rápidamente favorecido por el formato universal corriente del que está provisto. El primer formato de almacenaje en

Office 2000 para documentos fue XML. ADO (2.1) client-side tiene una opción para salvar documentos en XML.

ADO (2.0) Representa una enorme ganancia en funcionalidad. Una de las más notables características es la habilidad para crear client-side recordsets. El client-side recordsets tiene una fila local en el hard-drive que puede ser abierta sin tener que estar conectada a la red.

ADL (1.5) Introdujo nuevas capacidades y proveedores a ADO. Alguno de los nuevos proveedores fueron OLE DB proveedor para Jet (el proveedor JOLT). El MS Remote proveedor, cuya fuerza es Remote Data Services (RDS), también fue introducida. Con esta versión se introdujo la habilidad de crear recordsets desconectados.

3.1.3. Arquitectura de Conexión

Para el proceso de comunicación de la aplicación con los datos ya sean por FTP, HTTP o una conexión a Base de datos se ha visto varios tipos los mismos que explicaremos a continuación.

Component Object Model (COM) es una plataforma de Microsoft para componentes de software introducida por dicha empresa en 1993. Esta plataforma es utilizada para permitir la comunicación entre procesos y la creación dinámica de objetos, en cualquier lenguaje de programación que soporte dicha tecnología. El término COM es a menudo usado en el mundo del desarrollo de software como un término que abarca las tecnologías OLE, OLE Automation, ActiveX, COM+ y DCOM. Si bien COM fue introducido en 1993, Microsoft no hizo énfasis en el nombre COM hasta 1997.

Esencialmente COM es una manera de implementar objetos neutrales con respecto al lenguaje, de manera que pueden ser usados en entornos distintos de aquel en que fueron creados, a través de fronteras entre máquinas. Para componentes bien creados, COM permite la reutilización de objetos sin conocimiento de su implementación interna, porque fuerza a los implementadores de componentes a proveer interfaces bien definidos que están separados de la implementación. Las diferentes semánticas de reserva

de memoria están acomodadas haciendo a los objetos responsables de su propia creación y destrucción por medio del contador de referencias. Se puede hacer casting entre distintos interfaces de un objeto por medio de la función QueryInterface(). El método preferido de herencia en COM es la creación de sub objetos a los que se delegan las llamadas a métodos (llamado agregación).

Los Objetos COM pueden ser instanciados y referenciados en un proceso, a través de las fronteras de un proceso dentro de equipo y, a través de una red, usando la tecnología DCOM. Salir del proceso y de los objetos remotos puede utilizar serialización para enviar las llamadas a los métodos y valores de retorno hacia atrás y hacia delante. La serialización es invisible para el objeto y el código usando el objeto.

Hay que tener en cuenta que esta forma de conexión tendrá todos los beneficios solo si la red donde está funcionando contiene un DNS, el cual no es más que un directorio de servidor de nombres este es administrado por Active Directory. Como no es de nuestro estudio el análisis y la configuración de una red que contenga DNS vale mencionar que esta configuración se la obtiene instalando Windows Server 2003 0 2008.

Web Serices Esta forma de conexión usa el protocolo HTTP por lo que esta no limita a la conexión de los datos desde el cliente hacia el servidor, la única parametrización que se debe tener es que el servidor debe tener configurado la salida a internet o publicar el Servicio Web en Internet, este servicio web debe estar corriendo en un servidor de aplicaciones web, en nuestro caso será en IIS (Internet Information Services) para que el cliente pueda acceder a él siempre y cuando se le otorgue los permisos para hacerlo.

Esta forma de conexión es peligrosa ya que la información puede ser receptada por terceros y ocasionar un peligro. Es por esta razón que la información que sale del servidor debe estar encriptado y solo los clientes autorizados para ver la información podrán desencriptarlo.

3.2. Requerimientos de la arquitectura

Para el uso de las arquitecturas es necesario considerar la evolución de la tecnología, en tiempos pasados se tenía que conformar con lo pantalla negra o azul del DOS, una de las arquitecturas usadas eran el manejo de archivos planos como es el caso de C++, Fox, entre otros.
Todas las tecnologías que se usaban en tiempos pasados no tenían tantos requerimientos para que sus aplicaciones puedan correr ya que todo tiene que desarrollarse.

En la actualidad cada arquitectura que aparece es en base las necesidades de los usuarios, y cabe recalcar que la tecnología está dando grandes pasos, tanto que a un año más, nadie usará mouse, ya que todo es con touch screen (pantalla táctil), solo se manejara con los dedos.

Todos estos avances se han dado a que tanto la tecnología como la forma de usar las diferentes framework's han ido creciendo de igual forma, es por eso que la forma de manejar estos framework's se reduce en usar arquitecturas que soporten a los mismos.

A continuación numeraremos los requerimientos para que una arquitectura pueda correr en WPF.

- Framework 2, 3, 3.5, 2 SP1, 3 SP1, 3.5 SP1.
- Tarjeta grafica como mínimo de 128 para tener una buena resolución, si esta no viene incluida en el mainboard.
- Memoria Ram de por lo menos 2GB para no interrumpir con los demás procesos.

Estos requerimientos para el funcionamiento de las aplicaciones desarrolladas en WPF. Un framework, en el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado.

Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.
Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

3.3. Manejo de los controles de la arquitectura

Al hablar de WPF decimos que es una nueva tendencia, y como toda nueva tendencia tiene sus pruebas antes de presentar una versión definitiva, es por esta razón que los usuarios no usa mucho la tecnología nueva no antes de que esta haya tenido su tiempo en el mercado o hasta escuchar que la tecnología realmente es eficiente y eficaz para realizar los diferentes procesos a manejar.

A continuación daremos a conocer los controles más usados y potentes que presenta WPF a la hora de realizar una aplicación de escritorio o web.

Animación.- La funcionalidad de este componente dentro de WPF es usada en los diferentes controles que se manejen en las aplicaciones, dando un aspecto más interactivo y atractivo a la vista de los usuarios.

El uso de las animaciones en WPF hace que las aplicaciones usen al máximo los componentes gráficos de una máquina, tal es eso que las maquinas de la actualidad vienen implementadas con aceleradoras graficas, esto sucedió desde el lanzamiento del sistema operativo Windows Vista, el sistema operativo Windows 7 maneja tecnología WPF, es por esa razón de ser tan atractivo e interactivo al momento de trabajar.

Las animaciones son usadas en botones, cuadros de texto, formularios así a todos los controles de WPF, además se pueden crear nuevos controles con más funcionalidades que los normales, claro que todos se derivan de los controles existentes.

A continuación mostraremos un pequeño esquema de animación de un botón.

Botón Normal

Figura 3.2: Botón Normal

Botón MouseOver

Figura 3.3: Botón MouseOver

Botón MousePress

Figura 3.4: Botón MousePress

Este y muchos tipos de animaciones podemos realizar con el uso de XAML (Ver Figura 3.2, 3.3, 3.4), es en estos archivos donde se maneja la programación especificando las diferentes acciones que va a ejecutar el control al momento de presentare un evento.

3.4. Análisis de las ventajas y desventajas de la arquitectura

Entre las ventajas encontradas en esta arquitectura podemos las siguientes:

Aprovecha todas las bondades de la maquina donde está ejecutando la aplicación ya sea de escritorio o web, ya que una aplicación web desarrollada

con esta arquitectura corre en los clientes y no en el servidor como las aplicaciones web anteriores como era ASP de .NET.

La programación y el diseño grafico es manejado en un mismo GUI de desarrollo dando, esto se lo realiza mediante el uso de templates, que una vez desarrollados se los puede reutilizar.

Hace que los sistemas sean más atractivos a las vista de los usuarios claro está sin perder la eficacia y eficiencia que un sistema debe tener.

Entre las desventajas encontradas en esta arquitectura tenernos las siguientes:

Para que una aplicación desarrollada con esta arquitectura corra en una maquina debe cumplir con requerimientos actuales, caso que los ordenadores antiguos no lo tienen para lo cual deberán aumentar memoria, aceleradoras graficas.

Los plugins necesarios para el correcto funcionamiento de las aplicaciones de WPF hacen que el ordenador pida más características y los ordenadores antiguos algunos tendrán que desecharse.

3.5. División de las capas usadas por la arquitectura

La arquitectura de WPF se encuentra dividida en varias capas para el funcionamiento de las aplicaciones que en ellas se implementan, las capas se encuentran asignadas en la imagen se indica a continuación.

Los diferentes niveles de la figura 3.5 se lo describe a continuación para una mejor comprención.

Servicios de documentos.- es la capa que se encargara de procesar y visualizar los reportes y esta se lo realiza mediante archivos XPS. Los controles a usarcé para la visualización de documentos XPS son los ViewDocument el mismo que se le transformar un archivo XAML a XPS el cual se lo podrá visualizar mediante el control antes mencionado.

Interfaz de Usuario.- en esta capa se implementa los controles que se usan en las aplicaciones así como los la información y los templates usados por los controles.

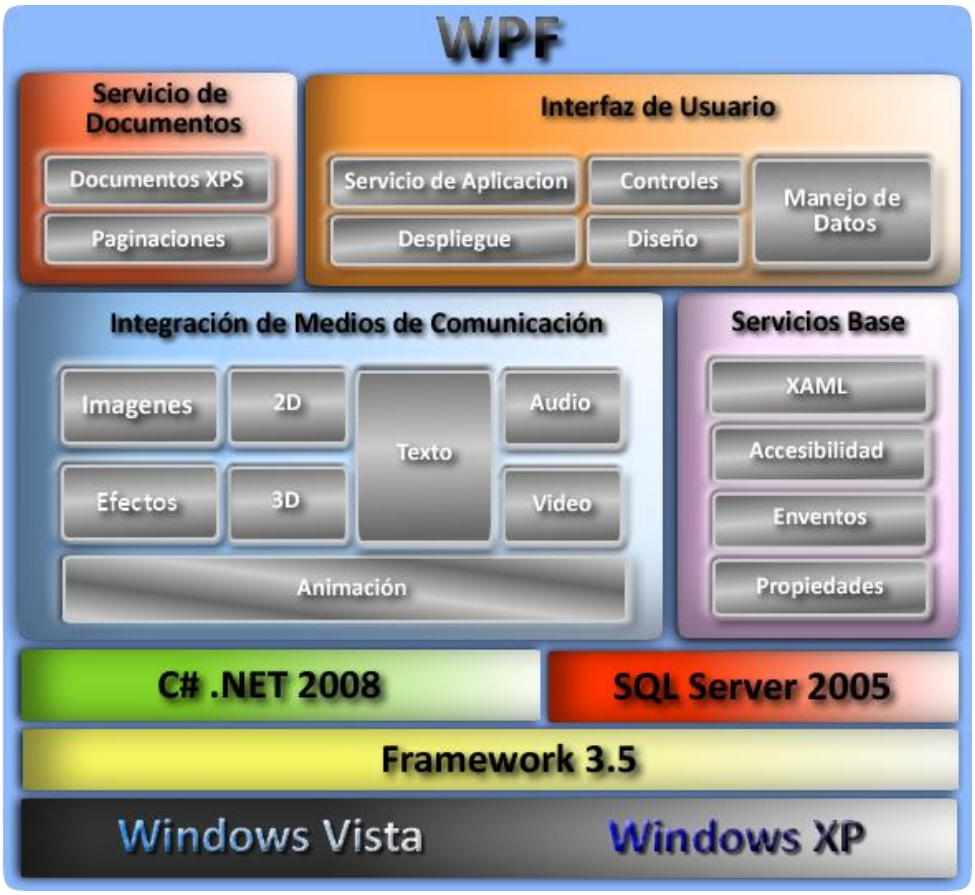


Figura 3.5: División de Capas

Integración de Medio.- esta capa es representada por el manejo de imágenes 2D, 3D, animaciones, texto, audio, video, los cuales aran que la aplicación se torne más amigable al usuario.

Servicios Base.- en esta capa se incluyen lo que son los accesos a base de datos, manipulación de información cliente servidor. Mediante esta capa se puede interactuar con la información ya sea usando componentes COM+ o servicios web para la conexión y manipulación de la información.

Y la capa primordial donde se implementan y corren las aplicaciones es la compuesta por el sistema operativo, motor de base de datos, GUI de desarrollo para la implementación de las aplicaciones, el motor que codificara la aplicación en funcionamiento.

3.6. Evaluación de la arquitectura COM+

El modo de conexión a las diferentes base de datos hacen que las aplicaciones sean más codiciadas por los clientes a la hora de implementar un sistema comercial, contable en fin entre más flexibilidad exista para conectarse a una o varias fuentes de datos la aplicación tendrá más importancia. El otro lado del funcionamiento de una aplicación es la forma de conectarse a las fuentes de datos, en la actualidad existen varias formas de conexión a una base de datos ya sea usando COM+, Web services, ADO, OLEDB, todas son validas pero pocas son las que manejan con fiabilidad la conexión a una Base de datos.

Para el estudio de la arquitectura de WPF usaremos como medio de conexión dos formas importantes y eficientes en la actualidad como es el caso de COM+ y Web Services.

COM+ (Component Object Model).- es una plataforma de Microsoft para componentes de software introducida por dicha empresa en 1993. Esta plataforma es utilizada para permitir la comunicación entre procesos y la creación dinámica de objetos, en cualquier lenguaje de programación que soporte dicha tecnología.

El término *COM* es a menudo usado en el mundo del desarrollo de software como un término que abarca las tecnologías OLE, OLE Automation, ActiveX, COM+ y DCOM. Si bien COM fue introducido en 1993, Microsoft no hizo énfasis en el nombre *COM* hasta 1997.

Esencialmente COM es una manera de implementar objetos neutrales con respecto al lenguaje, de manera que pueden ser usados en entornos distintos de aquel en que fueron creados, a través de fronteras entre máquinas. Para

componentes bien creados, COM permite la reutilización de objetos sin conocimiento de su implementación interna.

Las diferentes semánticas de reserva de memoria están acomodadas haciendo a los objetos responsables de su propia creación y destrucción por medio del contador de referencias. Se puede hacer *casting* entre distintos interfaces de un objeto por medio de la función QueryInterface(). El método preferido de herencia en COM es la creación de sub-objetos a los que se delegan las llamadas a métodos.

3.7. Manejo del Servidor de componentes

El servidor de componentes es un complemento de Windows, este modulo se lo encuentra desde las versiones de Xp, Vista, 7 y sistemas servidores Windows desde 2000 en adelante es decir Windows Server 2000, 2003, 2008. Este módulo administra los componentes que correrán en el servidor, allí se le da las distintas acciones que cada componente va a realizar cuando el cliente realice una petición.

Como revisamos anterior mente los componentes no son más que clases o dll's compiladas con ciertas funcionalidades adicionales para que sean aceptados por el servidor de componentes, estos componentes están configurados para que se transaccional entre otros componentes o no lo sean.

Un módulo importante para que funcione correctamente un servidor de componentes es tener instalado el servicio de DTC (Coordinador de Transacciones Distribuidas), la parametrización necesaria que debe tener este servicio es el que se hace conocer a continuación.

El servidor de componentes lo podemos encontrar en el administrador de tareas que se encuentra en el panel de control del sistema operativo instalado (Windows 2003 Server), para más información a continuación daremos a conocer el servidor de componentes, su configuración y sus componentes en funcionamiento, para la explicación se tendrá ciertos componentes ya integrados.

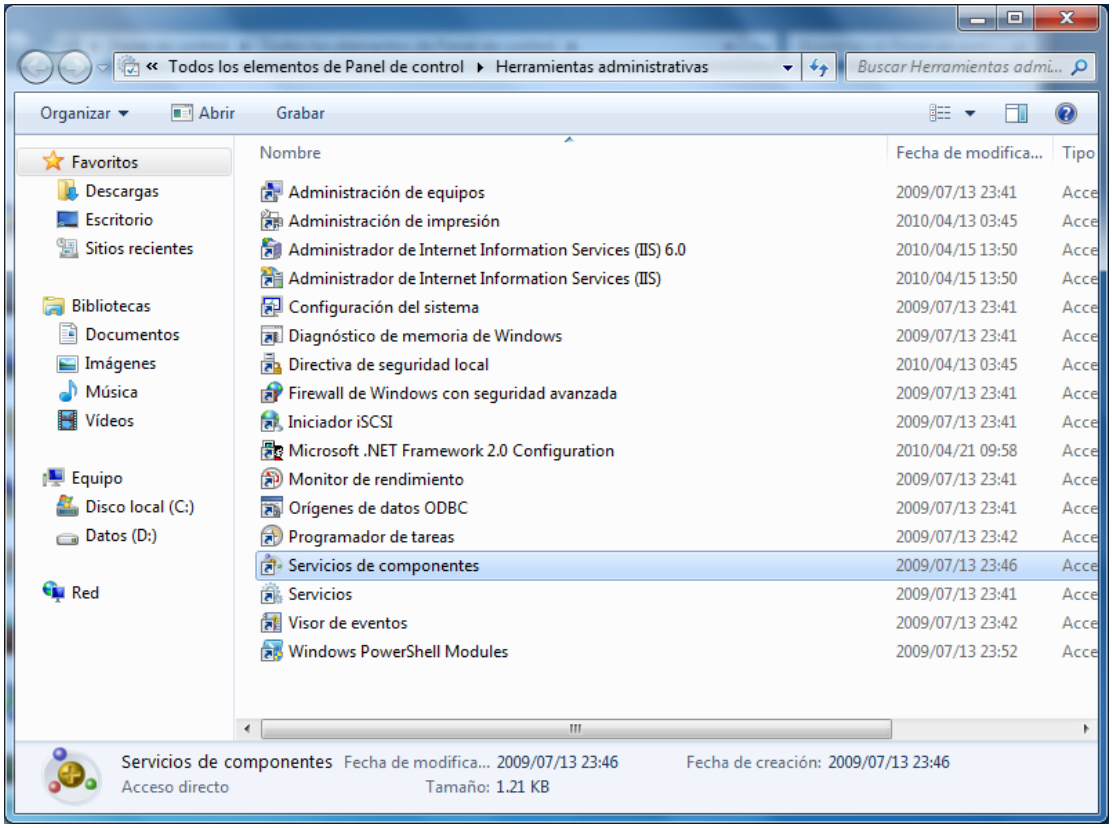


Figura 3.6: **Manejo del Servidor de Componentes**

El gráfico señala la opción del DTC (Coordinador de Transacciones Distribuidas)

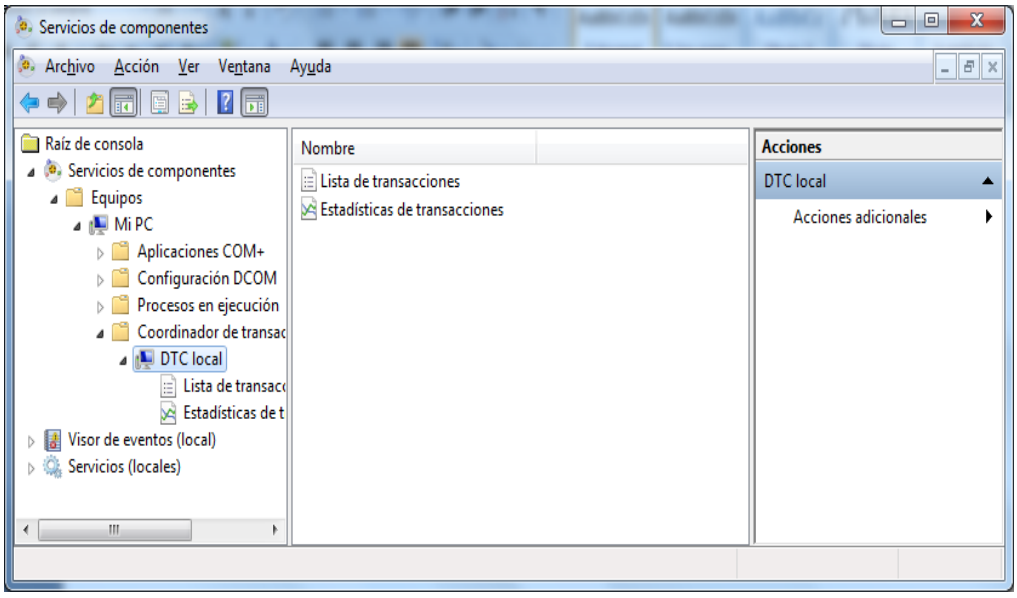


Figura 3.7: **DTC (Coordinador de Transacciones Distribuidas)**

Propiedades de DTC (Coordinador de Transacciones Distribuidas)

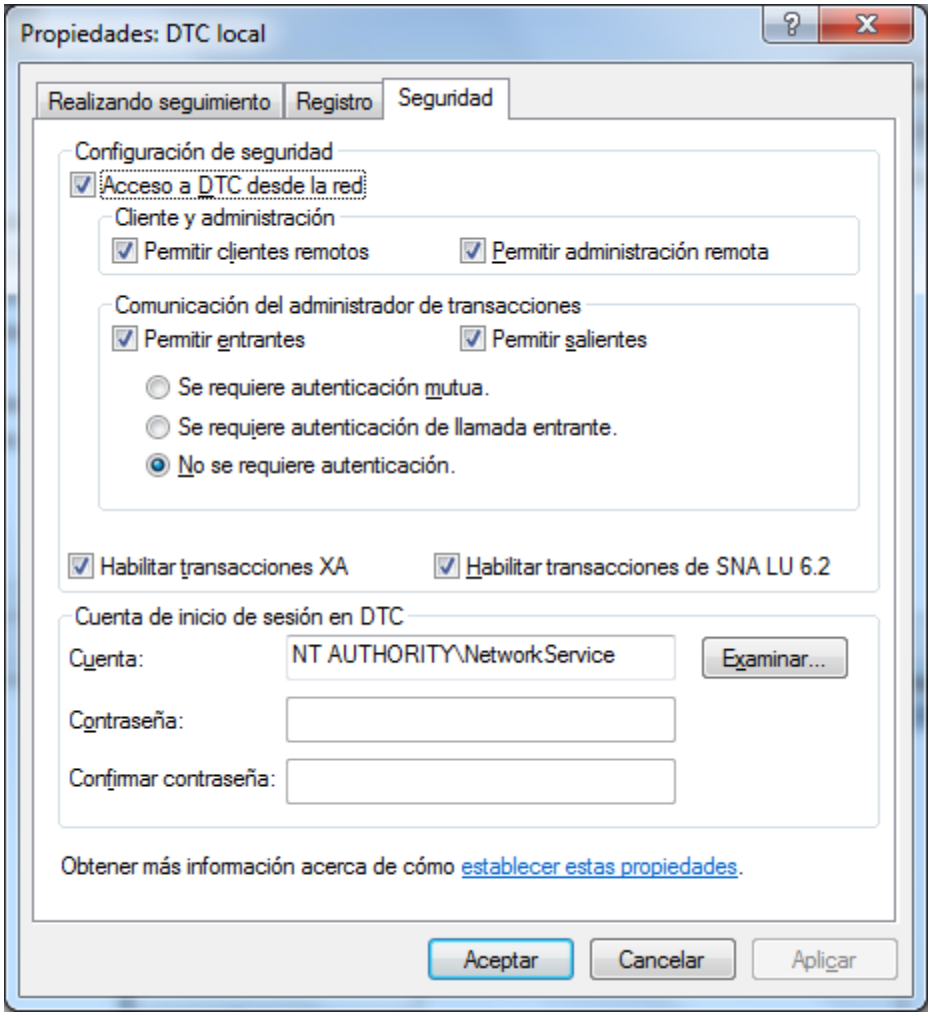


Figura 3.8: Propiedades de DTC (Coordinador de Transacciones Distribuidas)

Esta configuración es la que se debe de tener tanto en el servidor como en los clientes que se conecten al servidor en proceso, de esta forma certificamos que la conexión remota se la realice correctamente.

Es importante recordar que la conexión con los componentes se los realiza mediante la red de forma remota, usando todo el canal del ftp y http para su mejor funcionamiento.

A continuación mostraremos la configuración de los diferentes componentes que corren en el servidor de componentes, el componente a instalar es el que permitirá realizar las conexión hacia el servidor de base de datos para obtener las peticiones de los diferentes clientes.

Este componente puede leer información desde el servidor usando un nombre y una clave que se almacenan en el archivo de configuraciones del servidor de aplicaciones, este archivo contiene las conexiones a las diferentes bases de datos como también puede almacenar información que el cliente desee recuperar.

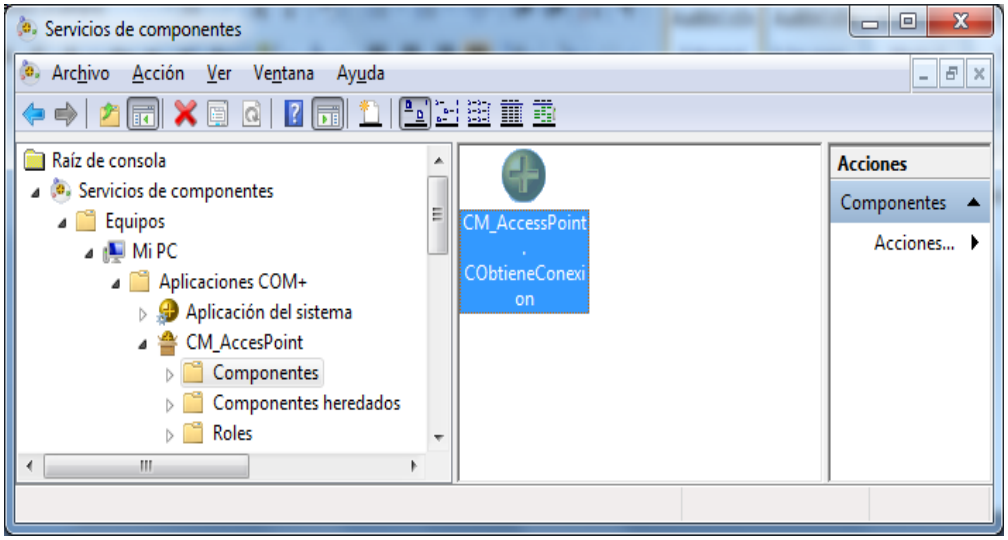


Figura 3.9: Configuración Acces Point

Propiedades del componente

Seguridad.- Esta opción sirve para indicarle al componente que niveles de autenticación se van a usar. Para el nivel de autenticación para llamadas debe ser *'LLAMAR'* y el nivel de suplantación que tendrá el componente será *'IDENTIFICAR'*.

El check de Exigir comprobación debe de estar seleccionado debido a que todas las aplicaciones que usan los componentes no corren en un mismo sistema operativo y por ende los certificados son diferentes.

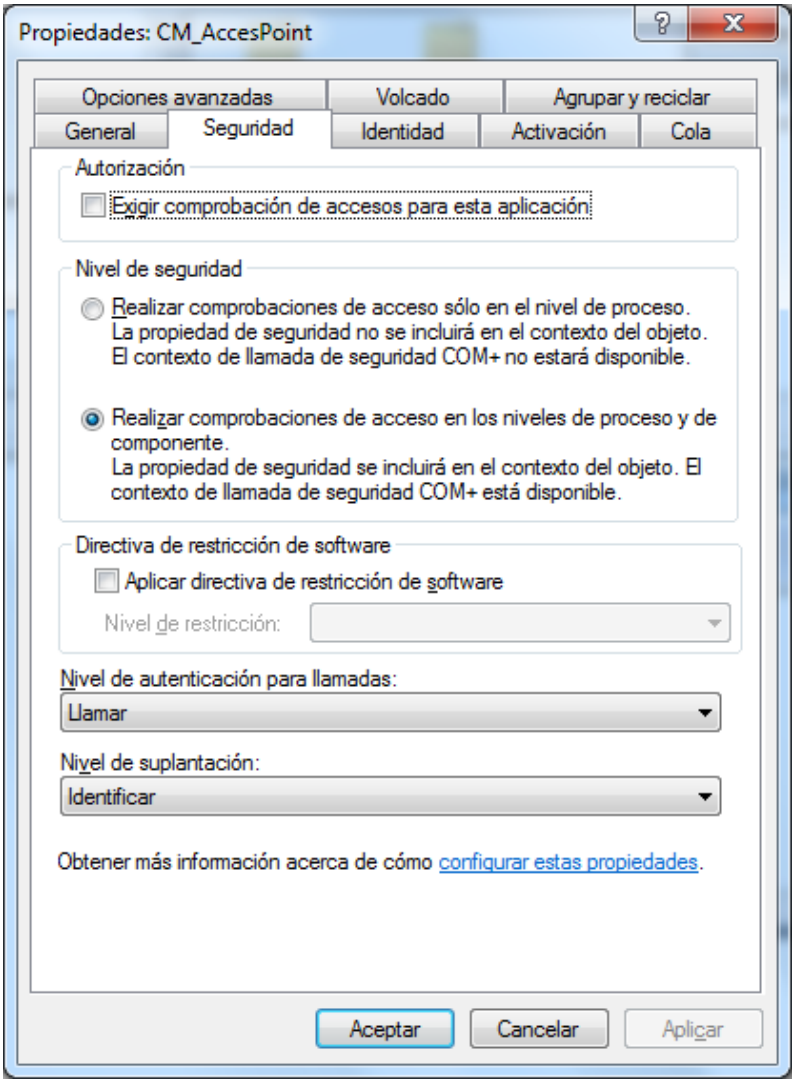


Figura 3.10: Propiedades del Componente (Seguridad)

Identidad.- Se indicará la identidad del servidor componentes, seleccionado el administrador del equipo administrador con su respectivo password.

Esta opción sirve para que el componente se identifique como administrador y así poder acceder a las funciones y métodos de los componentes, esta opción es usada mas en servidores de dominio, ya que cada cliente debe estar dentro de un grupo de dominio, este dominio será nuestro administrador y por ende será el pase para poder acceder sin ningún problema a los diferentes componentes que estén funcionando en el servidor de componentes.

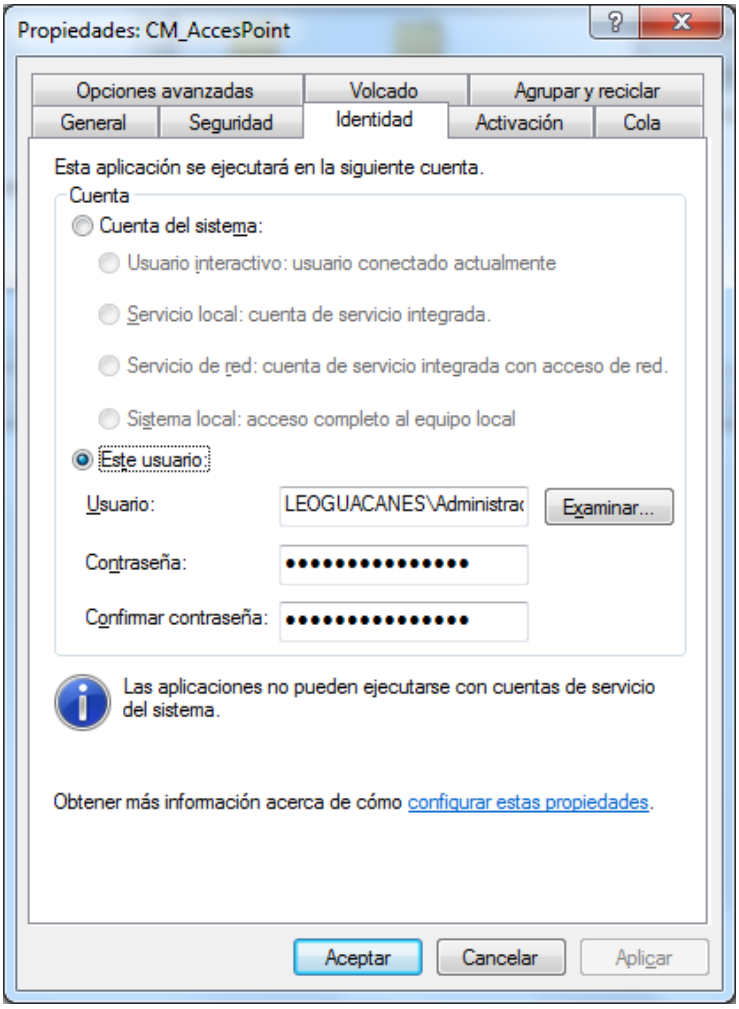


Figura 3.11: Propiedades del Componente (identidad)

Estas son las opciones básicas para el correcto funcionamiento de los componentes y así darnos un correcto funcionamiento al momento de realizar las conexiones hacia el servidor.