

Servidor de Impresión Genérico para Impresoras Fiscales

Altamiranda, Pamela Carolina – Ferreyra, Walter Rafael
pamelacaltamiranda@gmail.com – walterferreyra@gmail.com

Universidad Tecnológica Nacional
Facultad Regional Tucumán
www.ftt.utn.edu.ar

Resumen. Servidor de Impresión Genérico para Impresoras Fiscales es una aplicación que administra los documentos enviados desde una PC hacia una impresora fiscal, concentrándolos en una cola de impresión y brindando seguimientos de sus estados. Con este servidor podremos conectar una impresora fiscal a cualquier PC de la red e imprimir de las mismas, compartiendo así este recurso. Esta aplicación admite más de un modelo de impresora fiscal lo cual brinda la comodidad de ser adaptado a diversas organizaciones que ya cuentan con impresoras fiscales o a empresas que deseen cambiar dicho hardware.

Para la realización del presente trabajo, se procedió a la investigación del funcionamiento, modos y objetivos del uso de las impresoras fiscales, organismos que la proveen, empresas usuarios de este dispositivo, empresas proveedoras de software para cumplir con el propósito de la impresora fiscal.

Palabras Claves: Servidor de Impresión, Impresoras Fiscales,

Propósito del trabajo: Este presente trabajo se realizó en el marco, de trabajo final, de la materia "Programación de aplicaciones distribuidas", impartida por el Ing. José De La Cruz, correspondiente a Cuarto Año de la carrera Ingeniería en Sistemas de Información, dictada en la Universidad Tecnológica Nacional - Facultad Regional Tucumán.

Como objetivo del proyecto deseamos transmitir la experiencia de la aplicación de: lo aprendido en la cátedra de Programación de Aplicaciones Distribuidas, combinado con experiencias adquiridas en otras cátedras e interactuar con herramientas de uso real en el mercado como lo son, C# con Visual Studio 2010, PowerBuilder, MySQL, Impresoras Fiscales y Sistemas de Gestión Comercial. Además de la interacción con empresas del mercado.

Queremos dejar un ambiente inicial de desarrollo de una aplicación en C# con Visual Studio 2010, haciendo uso de interoperabilidad con PowerBuilder y base de datos MySQL aplicadas a un caso práctico como lo es El Servidor de Impresión Genérico para Impresoras Fiscales.

1 Introducción

El servidor de impresión genérico para impresoras fiscales presta solución a cualquier organización comercial que desee imprimir comprobantes por impresoras de este tipo, de cualquier modelo. Soporta escalabilidad, puesto que será factible agregar en un futuro modelos de impresoras no implementados. Brinda diversos informes que se asocian a los reportes generados por la impresora para dar soporte al análisis de la información y posterior toma de decisiones.

Este Servidor es una aplicación que permite usar la impresora fiscal desde cualquier computadora de una red de Windows o TCP/IP. Su configuración es sencilla, lo cual permite que podamos ubicar la impresora en la maquina que deseemos.

La aplicación trabaja bajo el modelo de cliente-servidor, lo que nos permitió partir de un modelo y aplicaciones existentes y así brindar mejoras con conocimientos exactos del alcance del proceso de negocio.

Este trabajo está integrado por los siguientes componentes:

- **Servidor de Impresión:** La aplicación se instala en la computadora servidor de la red (o en cliente) que administra la cola de impresión entre otras funcionalidades y envía los comprobantes a imprimir a través del puerto serial.
- **Base de datos:** Esta permite almacenar los comprobantes a imprimir (cola de impresión). Se instala en la computadora servidor (o en cliente).

- **Active Fiscal:** El ActiveX Fiscal es un objeto COM que puede utilizarse en cualquier entorno de 32 bits que soporte controles ActiveX. Debe registrarse en la PC donde está conectada la impresora fiscal.
- **Archivo .dll:** Driver específico de cada modelo de impresora que contiene funciones para interactuar con la misma. Librería que acompaña siempre al objeto OCX. No se instala ni se registra.

2 Relevamiento

Nuestro trabajo comenzó con el relevamiento de información suficiente sobre impresoras fiscales. En este caso accedimos al sitio <http://www.welcome.to/ifdrivers>.

Este Sitio Web contiene toda la información que se necesita acerca de la programación de impresoras fiscales y los drivers fiscales. Son desarrolladores de drivers fiscales para la mayoría de las impresoras fiscales que se venden en la Argentina. Los mismos están disponibles para bajar del área de drivers. Estos drivers son 100% funcionales e incluyen ejemplos en los lenguajes de programación más comunes, junto con su manual de programación y guía de referencia. Para su uso en producción se requiere de su registración.

Nuestro segundo paso fue consultar a diversas empresas comerciales sobre el uso y objetivo de las impresoras fiscales. También consultamos el sitio <http://www.afip.gov.ar/> para obtener información sobre las impresoras.

Nuestro tercer relevamiento consistió en entrevistar a empresas proveedoras de software para impresoras fiscales y a desarrolladores independientes. En este caso la información obtenida fue realmente importante debido a que ellos conocían realmente como interactuar con una impresora fiscal a través de un lenguaje de programación.

Y por último transmitimos esta información a la Cátedra de Programación de Aplicaciones Distribuidas para luego diseñar el plan de desarrollo y el uso de las herramientas.

3 Requerimientos no funcionales de la versión inicial

Como versión inicial, el Servidor de Impresión Genérico para impresoras fiscales soporta:

1. La impresión de facturas para el modelo de impresora fiscal EPSON TM2000AF.
2. Una impresora fiscal conectada a la misma computadora (Servidor) donde se encuentra la aplicación. En la versión final se podrá conectar la impresora fiscal al puerto serial de una PC de Windows e imprimir desde cualquier terminal de una red de Windows o TCP/IP. Por ejemplo, usted podría conectar la impresora fiscal a un puerto serie en su computadora, luego poner en funcionamiento el Servidor instalado en otra PC y después puede imprimir desde cualquier otra PC de su red. Esto permitiría utilizar cualquier programa de facturación para imprimir por la impresora fiscal en forma local o remota desde cualquier PC situada en la misma red.
3. Múltiples conexiones entrantes como servidor (Cola de impresión).
4. Configuración simple.
5. C# como lenguaje de programación principal
6. PowerBuilder como lenguaje utilizado para la interoperabilidad
7. Ingreso de datos a la base de datos de la cola de impresión a través de comandos de SQL. En la versión final se utilizará .dll que deberán ser entregadas a programador de la aplicación de gestión comercial.

4 Descripción del funcionamiento del Servidor de Impresión Genérico

El Servidor de impresión recibe las tareas de impresión a través de la red y los redirecciona al puerto serie donde está conectada la impresora fiscal.

El sistema funciona con un esquema cliente - servidor, donde la computadora donde está instalada la aplicación, funciona como servidor de impresión y las computadoras que acceden a sus servicios, son terminales clientes.

La impresora fiscal cuando recibe los comandos del Servidor los ejecuta imprimiendo el comprobante, luego de la ejecución de cada comando la impresora envía la

respuesta por el puerto serie al Servidor donde se actualiza en la base de datos el estado y el número del último comprobante impreso asignado por la fiscal. Entonces, si las demás terminales desean conocer esta información, deben acceder a través de la red a la base de dato que administra la cola de impresiones.

Con fines prácticos en el desarrollo de aplicaciones de este tipo, se provee de un emulador fiscal para simular la conexión física de la PC con la impresora.

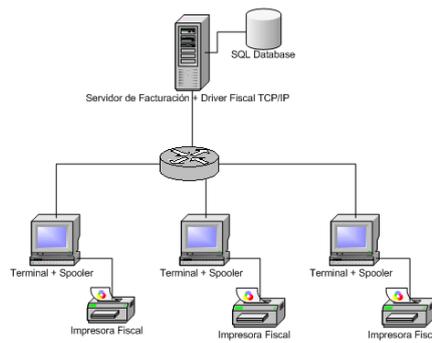
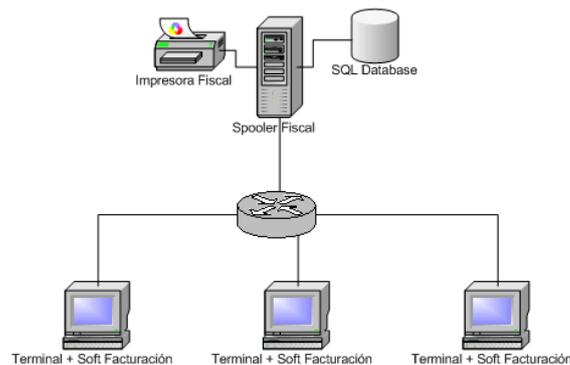


Figura 1: Modelo Cliente – Servidor.

La Fig. 1 muestra la conexión de más de una impresora fiscal a la red (versión final del proyecto). Esta versión inicial del servidor de impresión está desarrollado para una única impresora fiscal conectada a una PC (Servidor), en donde diferentes máquinas enviarán peticiones de impresión al Servidor en el cual se encuentra la base de datos (la misma, podría ser almacenada en cualquier máquina) que administra la cola de impresiones y la aplicación (que debe estar almacenada en la máquina que contiene la impresora). Las peticiones serán atendidas por orden de llegada,

Quedando el esquema descrito en la Fig. 2 que se muestra a continuación:



5 Metodología de desarrollo

Con respecto a la metodología utilizada es la de Craig Larman, basada en UML. Una de las razones principales es que contamos con la experiencia del cursado de materias referidas al tema coordinadas por la Ing. Mabel Torres y con su tutoría para este proyecto.

6 Desarrollo de la Aplicación

Para construir la versión inicial de esta aplicación, realizamos las siguientes acciones, las cuales fueron ordenadas en base a su uso inicial.

1. Una vez definido el objetivo de la versión inicial del proyecto: Imprimir facturas en la impresora Fiscal EpsonTM2000AF, comenzamos con la búsqueda de lo necesario.
2. El lenguaje a utilizar como base es C# con Visual Studio 2010 debido a que la cátedra así lo requiere. Pero también deseamos interactuar con otro lenguaje para conocer un poco sobre esta filosofía de la interoperabilidad. El lenguaje elegido fue PowerBuilder debido a nuestra experiencia.
3. Seleccionamos el Sistema de Base de Datos que vamos a implementar. Para esto recurrimos a nuestra experiencia adquirida en MySQL en la cátedra de Base de Datos I a cargo del Ing. José María Rodríguez.
4. Investigamos lo necesario para interactuar con la impresora fiscal EpsonTM2000AF. Determinamos que era necesario un Control ActiveX (OCX) más una librería (tm2032.dll) que fueron descargadas del sitio <http://www.welcome.to/ifdrivers> perteneciente a la empresa IfDrivers, como versiones de prueba.
5. Debido a que no contamos con la impresora fiscal física, adquirimos también un emulador fiscal descargado del sitio <http://www.welcome.to/ifdrivers> como versión de prueba para poder testear las interacciones desarrolladas desde C# con Visual Studio 2010.
6. Una vez que contamos con los componentes necesarios, iniciamos nuestro desarrollo:
 - a. Desarrollamos la base de datos.

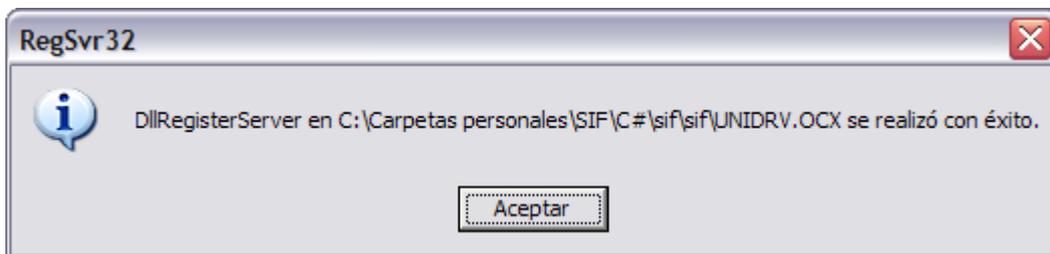
- b. Creamos un nuevo proyecto en Visual Studio 2010 C# y posteriormente agregamos controles necesarios en los que se encuentra el objeto OCX. Previamente es necesario registrar el OCX en regsvr32
- c. Agregamos la referencia a una dll desarrollada en PowerBuilder usada para recuperar información del comprobante a imprimir.
- d. Configuramos el Emulador fiscal para poder testear el desarrollo de los comando enviados a la impresora

7 Control ActiveX Fiscal

El ActiveX Fiscal es un objeto COM que puede utilizarse en cualquier entorno de 32 bits que soporte controles ActiveX. Esto incluye Visual Basic, VisualFox Pro, Delphi, PowerBuilder, ASP, etc.

Instalación

- a) Descargar el control OCX del sitio <http://www.welcome.to/ifdrivers> de acuerdo al modelo de impresora a utilizar
- b) Registrar en la línea de comandos regsvr32: inicio - ejecutar - regsvr32 - ctrl + arrastrar el archivo unidrv.ocx (es el que se descargó en el paso 1) hacia la línea de comando al lado de regsvr32, luego presionamos aceptar y nos mostrará el siguiente mensaje:



Nota: Para realizar los pasos 1 y 2 nos basamos en el contenido del siguiente link:

http://www.lawebdelprogramador.com/foros/Visual_Basic.NET/648989-Agregar_OCX_a_Visual_Basic_2005_Express.html

- c) En nuestro caso el driver que vamos a utilizar es tm2032.dll correspondiente a la impresora Epson TM2000AF, el cual puede ser adquirido a través del sitio <http://www.welcome.to/ifdrivers>.

Nota: Copiar el Ocx Fiscal **unidrv.ocx** junto con la librería tm2032.dll en un directorio o subdirectorio de su aplicación.

Importante: las DLL NO deben registrarse.

Una vez registrado se lo puede encontrar entre los demás controles ActiveX registrados e insertarlo en su programa.

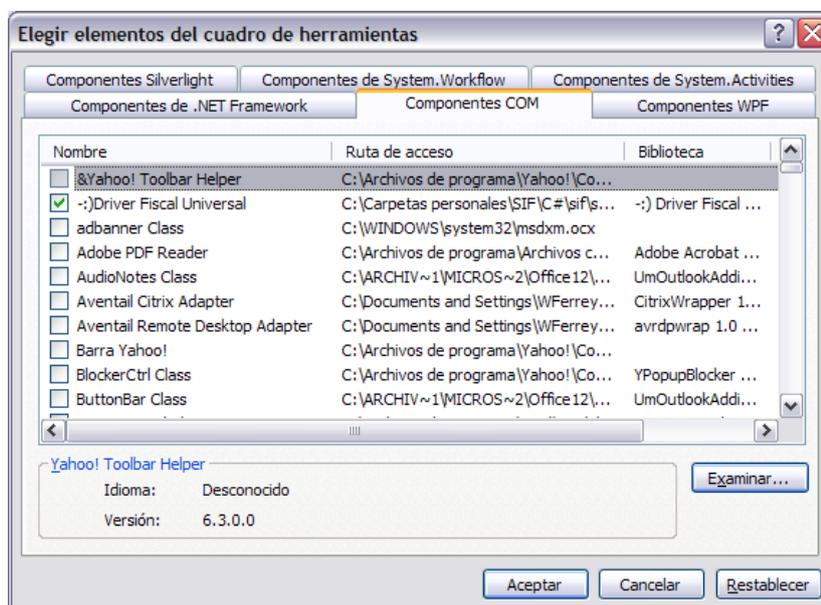
El control funciona como un wrapper (ver glosario) alrededor de las DLL fiscales de 32 bits (tm2032.dll), por lo cual es necesaria la presencia de estas DLL tanto al registrarlo como al distribuirlo. Una vez registrado, en C#, por ejemplo, se lo podrá encontrar en la ventana de componentes:

El ocx fiscal debe estar siempre junto a las bibliotecas dinámicas (DLL) fiscales para poder funcionar correctamente.

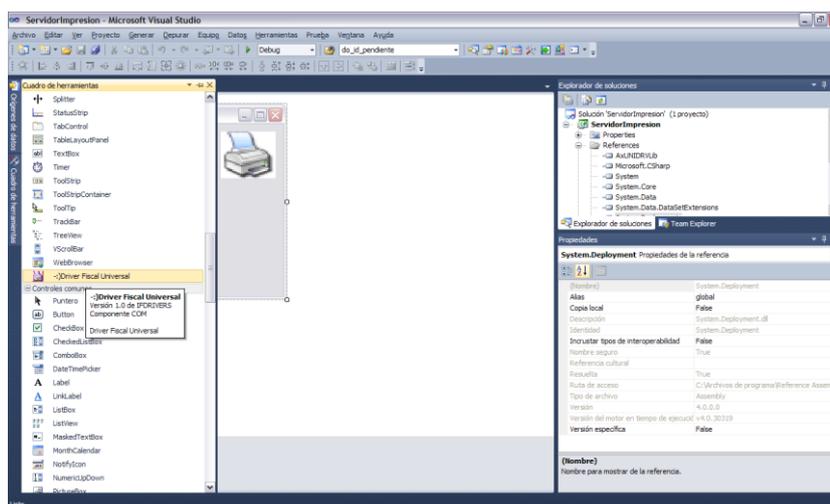
Una vez instalado y registrado, el ocx fiscal no debe moverse del directorio de instalación. Si por alguna razón quiere moverlo, deberá repetir el procedimiento de registración descrito antes.

Cómo usar un control ActiveX en C# con Visual Studio

Para ello hacemos click derecho sobre el cuadro de herramientas: “Todos los Formularios Windows Forms”, seleccionamos “elegir elementos”, en la ventana emergente activa y nos posicionamos en el tab “Componentes COM”, luego hacemos click en “examinar”, seleccionamos el objeto “unidrv.ocx” descargado y posteriormente presionamos aceptar.



Se agregará en el cuadro de herramientas “Todos los formularios Windows Forms” un nuevo componente llamado “driver fiscal universal”, seleccionamos dicho componente y lo arrastramos hacia la ventana.



8 Herramientas utilizadas

1. Base de datos:
 - a. MySQL 5.0 through 5.0.25
 - b. MySQL Administrator Versión 1.2.6 rc
 - c. MySQL Query Browser Versión 1.2.6 beta
 - d. MySql.Data.dll
2. Desarrollo
 - a. Microsoft Visual Studio 2010 Versión 10.0.30319.1 RTMRel
 - b. SyBase PowerBuilder Enterprise Versión 11.5 Build 2506
3. Modelo Relacional
 - a. MyManager.exe y libmySQL.dll

10 pamelacaltamiranda@gmail.com – walterrferreyra@gmail.com

b. ER Studio Version 7.0

4. Interacción con la Impresora Fiscal

a. IfDRivers UNIDRV.OCX, tm2032.dll

b. IfDRivers Emulador Fiscal Versión 4.2.1 (build 1110)

Ejemplo del funcionamiento del Servidor de Impresión Genérico para impresoras fiscales:

<http://www.youtube.com/watch?v=toynI8rWJUl>

9 Conclusiones

En el presente trabajo se desarrolló una herramienta de apoyo a los procesos de facturación de organizaciones comerciales. La aplicación es fácilmente adaptable a ambientes de redes y modelos de impresoras de diferentes marcas, simplificando la tarea al sistema de gestión comercial, encargándose exclusivamente, ésta aplicación, de la impresión de los documentos legales. El desarrollo incremental permitió construir un ambiente inicial de programación de una aplicación servidor de impresión genérico para impresoras fiscales para evolucionar hacia una versión final. Se reunieron valiosos conocimientos a cerca de procesos comerciales y la administración de impresoras fiscales, con el fin de registrar la documentación necesaria y administrar los libros exigidos por la AFIP.

La implementación inicial fue exitosa alcanzando los objetivos planteados al comenzar el trabajo, como por ejemplo: Interoperabilidad entre C#.Net con Visual Studio 2010 y PowerBuilder 11.5, Administrar la cola de impresión de comprobantes con una base de datos MySQL y comunicación entre cualquier sistema de gestión comercial con el servidor de impresión genérico para impresoras fiscales.

Con esta herramienta la organización podrá disminuir los costos de hardware al no ser necesario adquirir una gran cantidad de impresoras fiscales.

Desde una visión práctica esta aplicación admite más de un modelo de impresora fiscal lo cual brinda la comodidad de ser adaptado a diversas organizaciones que ya cuentan con impresoras fiscales o a empresas que deseen cambiar dicho hardware.

10 Agradecimientos

Agradecemos a las cátedras Programación de Aplicaciones Distribuidas impartida por el Ing. José De La Cruz e Introducción a la Ingeniería del Software a cargo de la Mg. Ing. Mabel Torres por habernos brindado la oportunidad de encarar un proyecto bajo su coordinación y apoyo.

Agradecemos también a las siguientes empresas:

IFDrivers, por brindarnos toda la información que se necesita acerca de la programación de impresoras fiscales y los drivers fiscales.

Global ID, por asesorarnos en cómo desarrollar un software de este tipo y el acceso a sus empresas clientes para entrevistas sobre el uso de impresoras fiscales.

11 Referencias

[1] Wikipedia: http://es.wikipedia.org/wiki/Component_Object_Model

[2] IfDrivers: <http://www.welcome.to/ifdrivers>

[3] PowerBuilder: <http://www.sybase.com/>

[4] Global ID, Sistemas Informáticos: <http://www.globalid.com.ar/>

[5] Visual Studio:

<http://www.frt.utn.edu.ar/>

http://msdn34.eacademy.com/elms/Storefront/Home.aspx?campus=utn_tucumon

[6] MySQL: <http://www.mysql.com/>

[7] SOMMERVILLE 2005: Sommerville Ian, Ingeniería del Software, Séptima Edición, Pearson Educación, Madrid, 2005.

[8] LARMAN 2003: Larman Craig, UML y patrones Una introducción al análisis y diseño orientado a objetos y al proceso unificado, Segunda Edición, Prentice Hall, Pearson, Pearson Educación, Madrid, 2003.

[9] COLOURIS et al. 2003: Colouris George, Jean Dollimore, Tim Kindberg, Sistemas Distribuidos, Conceptos y Diseño, Tercera Edición, Addison Wesley, 2003.

12 pamelacaltamiranda@gmail.com – walterferreyra@gmail.com