

DESARROLLO DE UN COMPONENTE Y REUTILIZACIÓN EN VARIAS APLICACIONES

Chambi Gareca Silvia

Universidad Autónoma Juan Misael Saracho

Tarija - Bolivia

Correo electrónico: chambi.gareca.silvia@gmail.com

RESUMEN

El desarrollo de software basado en componentes es el paso hacia una verdadera ingeniería y en la actualidad se ha convertido en uno de los mecanismos más efectivos para la construcción de aplicaciones software.

Siguiendo la línea de la programación basada en componentes vemos que la reutilización de componentes software permite optimizar el proceso de desarrollo de software.

El presente trabajo es una investigación que pretende mostrar la experiencia en la creación de un componente software bajo la plataforma framework.Net mostrando su consumo y reutilización en varias aplicaciones (Sistema de Almacenes Carolina y Sistema de la toma de pedido PIZZA) como elemento de mejoramiento del desarrollo de aplicaciones informáticas.

En la investigación se expone una descripción pormenorizada de las características que debe cumplir un componente como propiedades, atributos, métodos bajo el framework.Net. y se describen los procedimientos de compilación y utilización de componente, finalmente una síntesis de la experiencia realizada.

El objetivo que persigue la investigación es lograr la creación de un componente y consumirlo en varias aplicaciones informáticas, para lo cual se analizaron

y sintetizaron la teoría la programación basada en componentes, las características de un componente, plataformas de desarrollo y especificación sobre la creación de un componente.

La implantación del modelo de componentes tiene múltiples beneficios como el desarrollo de aplicaciones es más sencillo y en un tiempo y con un coste menor, la posibilidad de errores es mínima ya que los componentes son sometidos a un riguroso control de calidad antes de ser comercializados, finalmente las empresas de desarrollo especializadas pueden obtener ingresos adicionales vendiendo componentes a otras empresas.

En el ámbito local hay escasa información sobre la creación de componentes software así como en la programación basada en la reutilización de éstos, sin embargo la información de las redes fue determinante para alcanzar el objetivo.

PALABRAS CLAVE

Programación, software, componentes, TIC, ingeniería de software.

INTRODUCCIÓN

En el mercado mundial del software, el paradigma de los componentes se ha convertido en un estándar, y en la actualidad está siendo masivamente usado, por lo que el fin de este trabajo fue profundizar en la creación de componentes de software y experimentar su reutilización como elemento



de mejoramiento del desarrollo de aplicaciones informáticas.

La indiferencia hacia este nuevo paradigma trunca el desarrollo de la ingeniería del software, porque se deja de lado una programación que en la actualidad es una realidad, una tendencia, una nueva forma de hacer ingeniería del software.

Generar gente preparada con lo que usa y necesita la industria es lograr un profesional que pueda competir en el mercado, altamente capacitado para desenvolverse en cualquier área.

Por otro lado el desarrollo de la tecnología basada en componentes es el desarrollo de una ingeniería más madura, en la cual la disponibilidad de soluciones permite al ingeniero elegir entre varias opciones la más adecuada, en vez de perder el tiempo inventando una o dos soluciones; lo cual conduce a grandes ventajas enfocando de mejor manera los problemas de software resultado de un mejor proceso de ingeniería.

Una analogía es como la ingeniería civil alcanzó un grado de desarrollo cuando comenzó a utilizar piezas prefabricadas o fabricadas, por ejemplo cuando se construye una casa se compran las puertas, las aldabas, pero no fabrican desde cero, lo que se pretende con la investigación de programación basada en componentes es aprender a utilizar y manipular fragmentos de código prediseñados, e introducirlos en una aplicación que se construya, así mismo sobre la base de la manipulación de componentes crear un componente y consumirlo.

Para que la ingeniería del software alcance un grado de maduración se debe lograr la generalización de la reutilización de código existente en el desarrollo de software, todavía desarrollar una aplicación sigue requiriendo la escritura de una gran cantidad

de código, en cambio, en otras ingenierías como la electrónica el grado de reutilización es altísimo. En este ámbito, la construcción de un dispositivo electrónico se reduce a la integración de la manera adecuada de distintos componentes comerciales.

Los usuarios requieren grandes sistemas en tiempos cortos pero el desarrollo implica demasiado tiempo.

El desarrollo basado en componentes se define sobre la construcción de una aplicación a partir de componentes software comercial o gratuito ya existentes, limitando al mínimo necesario el desarrollo de código nuevo, para que este cambio se produzca es necesaria la creación de un mercado amplio de componentes software de calidad y con un ámbito de aplicación general, también es necesaria la adopción de una arquitectura estandarizada que garantice la interoperatividad de los componentes en distintas plataformas, lenguajes de programación y entornos de desarrollo.

Por lo que proponemos la creación de un componente y su consumo en aplicaciones empleando la teoría de componentes que permitirá evaluar en nuestro ámbito la fiabilidad, funcionalidad, y factibilidad del producto, que sustentará el planteamiento de una propuesta de reutilización de componentes en aplicaciones. Basadas en Componentes como una opción para mejorar el proceso de desarrollo del software.

2. MATERIALES Y MÉTODOS

Para la investigación se utilizó el enfoque de Programación Basada en Componentes considerando la reutilización como elemento fundamental.

Se propone un procedimiento para crear un componente mostrando las ventajas de reutilización del componente en varias aplicaciones



para contribuir al desarrollo de la ingeniería del software.

Las herramientas usadas para la investigación fueron: ENTERPRISE BEANS, Framework .NET

La metodología seguida se describe

Configuración del proyecto:

Iniciado el SharpDevelop se elige la opción de nueva solución, luego la plantilla biblioteca de controles de usuario de Windows, finalmente se escribe el nombre del proyecto como también la ruta donde se desea guardar.

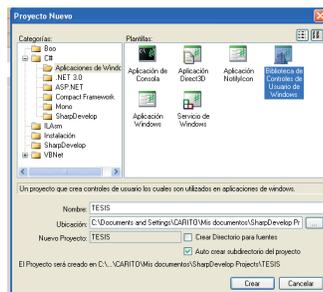


Fig. 1: Codificación del componente:

Se realizó la codificación del componente en el entorno de trabajo SharpDevelop 2.1 en el que se visualizan 3 secciones: en la parte izquierda se encuentra la sección de estructura del proyecto o solución, en la parte central la sección de codificación y en la parte derecha la sección de Propiedades como se observa en la figura de abajo.

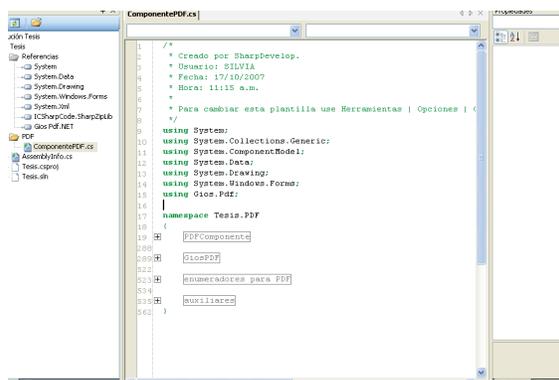


Fig 2: Compilación del componente

El componente creado fue establecido como biblioteca de clases por lo que el resultado de la compilación será una DLL (un ensamblado), donde para estar seguros de que el resultado de la compilación sea un DLL se debe verificar las propiedades del proyecto.

En esta figura se señalan dos aspectos que se deben verificar para que el componente sea compilado como una DLL y son el tipo de salida que es Biblioteca de clases y el nombre de salida que debe tener la extensión.dll

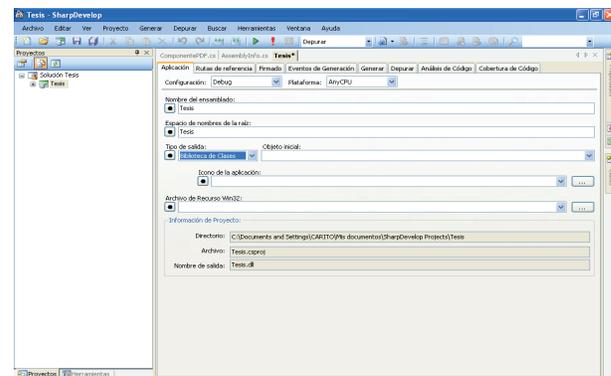


Fig 3: Reconocimiento del componente por parte del entorno de desarrollo.

Se debe personalizar la barra de herramientas escogiendo la opción personalizar barra Lateral

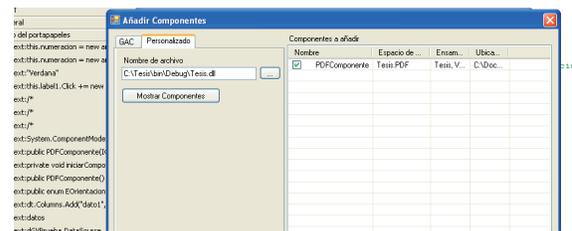


Fig. 4: Añadir Componentes.

Si el componente está correctamente definido la ventana de Añadir Componentes debe mostrar al ensamblado de nuestro componente más lista de si existiesen otros componentes.

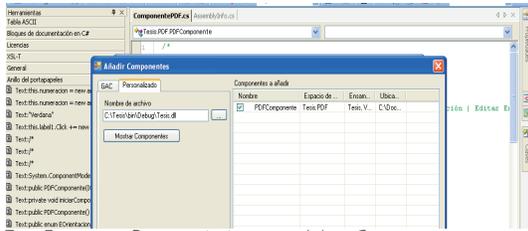


Fig. 5: Reconocimiento del Componente por Microsoft Visual C# 2005 Express Edition

El componente que exporta archivo a PDF creado por SharpDevelop es reconocido por otro entorno de desarrollo como se muestra en las pantallas siguientes.

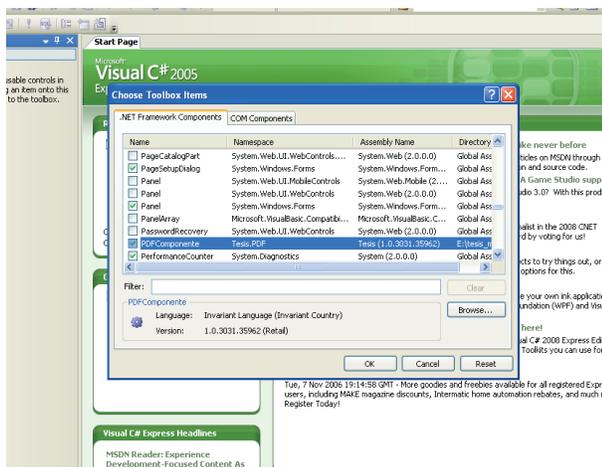


Fig. 6: Prueba de Funcionalidad del componente.

Para probar la funcionalidad y reutilización del componente creado se describe en de las tres aplicaciones que consumen al componente.

Se describe el consumo del componente por parte de la aplicación informática que es un Sistema de control de pedido que permite la conexión a base de datos y que consta de tres módulos: Toma de Pedido, Administración y Reportes.

El componente es consumido específicamente por el módulo de reportes el que contempla informes de ventas diarias, ventas por pizza, ventas por bebida, ventas por servicios, ventas por clientes, ventas a domicilio, pedidos emitidos, pedidos

cancelados, exportación de todos los informes a archivos Excel y pdf.

Esta investigación nace a raíz de que en esta aplicación en una primera versión al usarse las librerías de clases GIOSPDF para la generación de reportes mediante archivos PDF nace la necesidad de convertir a esta funcionalidad en un componente genérico reutilizable.

Es aquí donde se realizaron la mayor cantidad de experimentos para poder reutilizar el componente.

La figura se muestra el consumo del componente PDFComponente con el nombre pdfPizza.

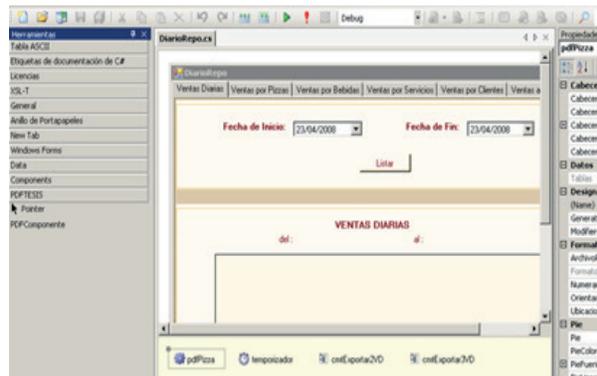


Fig. 7: Se presenta el código para la exportación del archivo a PDF.

```
void ExportarAPDFToolStripMenuItemClick(object sender, EventArgs
{
    if (dgvVentasDiarias.DataSource != null)
    {
        pdfPizza.Tablas = dgvVentasDiarias.DataSource as DataTable;
        pdfPizza.TituloTexto = "Ventas Diarias";
        pdfPizza.PieTexto = "Del: " + dtpfecha_inicio.Text + " al: " +
dtpfecha_fin.Text;
        pdfPizza.generarPDF();
        pdfPizza.salvarPDF(true);
    }
    else
    {
        Mensajes.ventana("No existen datos para exportar", tipoMensaje.Advertencia);
    }
}
}
```

Se muestra la ejecución de la aplicación y la generación del archivo en formato PDF.



Fig. 8: Vista



producto	cantidad	total
Pizza	13	300,00
Grande	8	200,00
Mediana	5	100,00
Bebida	4	26,00
Sandwich	5	25,00
total		351,00

Fig. 9: Reporte

3. DISCUSIÓN

El paradigma de los componentes es una tendencia sólida y es lo último que utiliza la industria del software, y la utilización de estas tecnologías logra aplicaciones fiables, robustas desarrolladas en tiempos cortos, permitiendo a los profesionales insertarse en proyectos de gran envergadura.

La utilización de plataformas de desarrollo de componentes (framework), permite la elaboración de componentes con un grado adecuado de reutilización por aplicaciones informáticas.

Se desarrolló un componente software generico.

CONCLUSIONES

- Con la creación de un componente se busca mejorar el desarrollo de aplicaciones informáticas esto se aplica en empresas de desarrollo, en equipos de investigación o grupos de desarrollo.
- Sin la base de la programación Orientada a Objetos programar con componentes es muy complicado y difícilmente se logrará construir sistemas en periodos cortos de tiempo.

- Después de la experiencia en la utilización de componentes propongo se implante este enfoque en los últimos cursos de programación en la carrera de Ing. Informática, donde ya nuestros estudiantes tienen el conocimiento previo de la programación orientada a objetos ya que un componente es una extensión de los objetos y posee características similares a ellos.
- Propongo la creación de un componente avanzado como tema de tesis ya que el componente es una aplicación que necesita análisis, diseño y prueba en las aplicaciones que lleguen a utilizarse.

BIBLIOGRAFÍA O REFERENCIAS

- [1] C. Aranda, M.L Callejo, "Construcción del concepto de dependencia lineal en un contexto de geometría dinámica". Revista Latinoamericana de Investigación en Matemática Educativa, 13(2), 129-158. 2010.
- [2] Grady Booch, Ivar Jacobson, James Rumbaugh, Jim Rumbaugh. "The Unified Modeling Language User Guide. Addison-Wesley", octubre de 1998. ISBN: 0-201-57168-4.
- [3] Desmond Francis D'Souza, Alan Cameron Wills. Objects, "Components and Frameworks with UML : The Catalysism Approach. Object Technology Series". Addison-Wesley, 1999. ISBN: 0-201-31012-0.
- [4] Jonás A. Montilva, Nelson Arapé2, Juan Andrés Colmenares. "Desarrollo de Software Basado en Componentes".
- [5] Fuentes Lidia, Troya Joséy Vallecillo Antonio Dept. "Lenguajes y Ciencias de la Computación".



Universidad de Málaga. ETSI Informática. Campus Teatinos, s/n. 29071 Málaga, Spain.

[6] Fuentes Lidia, Troya José Vallecillo Antonio Dept. “Lenguajes y Ciencias de la Computación”. Universidad de Málaga. ETSI Informática. Campus Teatinos, s/n. 29071 Málaga, Spain.

[7] Zorrilla Santiago, Torres Miguel. “Guía para elaborar la Tesis” Atlacomulco, agosto de 1998. ISBN: 970-10-0139-7

[8] Kraig Brockschmidt. Inside OLE. Second Edition. Redmond, WA: Microsoft

Press, 1993. (Development Library, Books and Periodicals).

[9] Agustín Cernuda del Río “Sistema de verificación de componentes software” Oviedo, Febrero de 2007.

Ejemplo 3: (Cita de una comunicación a un congreso)

Clayton, M. (1998) Computer-assisted, portafolio-nased composition: The next step for

freshman composition at MTSU. Paper presentet at Mid-South Instructional Technology

Conference, Muefreesboro, TN, Abril, 1998

[10] <http://nereida.deioc.ull.es/~cleon/doctorado/doc01/lang/poo.html>

[11] http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3985.asp Desarrollo de Software basado en Componentes. htm

[12] http://computacion.cs.cinvestav.mx/~sgarrido/cursos/ing_soft/Componentes/node55.html

[13] http://java.ciberaula.com/articulo/tecnologia_orientada_objetos/

<http://www.geocities.com/eztigma/tendencias.html>

[14] <http://msdn2.microsoft.com/es-es/library/>

[15] <http://commanet.blogspot.com/2006/02/java-no-est-orientado-componentes.html> (java no esta orientado a componentes)

[16] <http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/vbcon/html/vbwlkwalkthr>

