Desarrollo rápido de aplicaciones y puesta a punto

Rapid Application Development and tuning

José Mario Martínez Castro[*]
René E. Cuevas Valencia[**]
Valentín Álvarez Hilario[***]

Fecha de recepción: 16 de febrero de 2011
Fecha de aceptación: 16 de junio del 2011

Resumen

El desarrollo de sistemas de información ha tenido grandes cambios con el advenimiento de la tecnología del software, como tal, en un principio, solo se contaba con lenguajes de programación con interfaces de texto en pantallas de color ámbar o verdes; posteriormente, el entorno gráfico dio la posibilidad de crear interfaces gráficas de usuario (GUI), lo que posibilitó darles mayor vida a las realizaciones de los programadores; por un buen tiempo, esto se mantuvo así, hasta la llegada de Internet y su WWW, que brindó posibilidades de captación de mercados sin explorar. Pero, al incrementar la tecnología de las interfaces, de forma paralela se incrementaba la complejidad para desarrollar esos mismos sistemas, que si bien cumplían funciones equivalentes a que sus predecesores, ahora eran más atractivos para sus usuarios: por la rapidez de desarrollo, mejor experiencia de uso, así como capacidad de adaptabilidad para satisfacer las necesidades de los clientes finales en el menor tiempo posible. Para lograr esto debieron evolucionar tanto metodologías eficientes como herramientas de software que incrementaran las capacidades de los desarrolladores; dentro de estas últimas

[*] Universidad Autónoma de Guerrero, Unidad Académica de Ingeniería, Chilpancingo Guerrero; México. Correo electrónico: jmmtzc@hotmail.com
[**] Universidad Autónoma de Guerrero, Unidad Académica de Ingeniería, Chilpancingo Guerrero; México. Correo electrónico: reneecuevas@hotmail.com
[***] Universidad Autónoma de Guerrero, Unidad Académica de Ingeniería, Chilpancingo Guerrero; México. Correo electrónico: valentin_ah@yahoo.com

encontramos: los entornos integrados de desarrollo (IDE), las Herramientas CASE, así como las herramientas RAD y RADD.

Palabras clave: RADD, CASE, sistemas de información, programación, software.

Introducción

Los sistemas de información se han incorporado a las instituciones y empresas; en la actualidad, no se concibe la idea de la operación de organizaciones de todos los tamaños sin el apoyo no solo de los equipos de cómputo, sino también de la implantación de estos sistemas, que se conciben como "Un conjunto de componentes interrelacionados que reúne (u obtiene), procesa, almacena y distribuye información para apoyar la toma de decisiones y el control en una organización" (Laudon y Laudon, 2008, p. 35). Para lograr el desarrollo de estos sistemas se requiere tanto de metodologías como de herramientas destinadas para tal fin (CMS, 2008, p.20). En el primer rubro se puede encontrar lo que sigue a continuación.

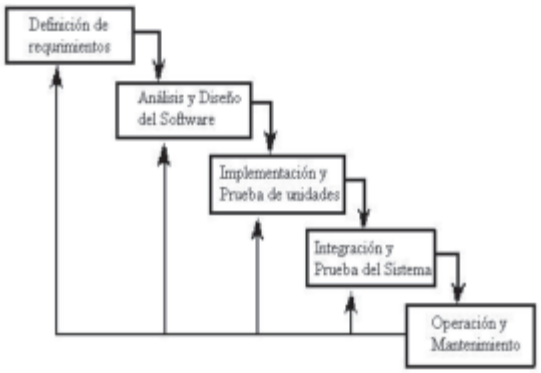
Metodología de cascada

- Se compone de: definición de requerimientos, análisis y diseño de software, implementación y pruebas de unidades, integración y pruebas del sistema, operación y mantenimiento (ver figura 1).
- Básicamente el contacto con el cliente de realiza en la primera etapa y no es posible detectar errores ni modificaciones a los requerimientos del cliente, si no hasta la etapa final (Nigthet ál., 2001, p. 54).

Metodología en espiral

- Se tienen como elementos: comunicación con el cliente, planificación, análisis de riesgos, ingeniería, construcción y entrega, evaluación del cliente (figura 2).
- Existe una alta comunicación con el cliente, para la evaluación del sistema con base en sus requerimientos y en la finalización de cada iteración programada.
- La planificación también cambia con cada iteración, lo que puede traer como consecuencia un sistema sin metas fijas (Green y DiCaterino, 1998, p. 40).

Figura 1. Metodología en cascada



Fuente: http://siteknox.blogspot.com/2010_09_01_archive.html

Figura 2. Metodología en espiral

Fuente: http://siteknox.blogspot.com/2010_09_01_archive.html

Metodología de prototipaje

- Se compone de los siguientes elementos: planeación, requerimientos, análisis y diseño, desarrollo, implementación, pruebas, evaluación, y la administración de los cambios (figura 3).
- De la misma forma que la metodología anterior se establece una alta comunicación con el cliente para la evaluación del sistema con base en sus requerimientos.
- Se requiere realizar una correcta administración de los cambios, así como el control de versiones del sistema por desarrollar (Fisher, et ál, 2010, 15).

Figura 3. Metodología prototipaje

Metodología de desarrollo rápido de aplicaciones

Los principios básicos son:

- Desarrollo y entrega rápida de sistemas de alta calidad y bajo costo.
- Intento de reducir los riesgos, seccionando en pequeños.
- Producir rápidamente sistemas de alta calidad, de forma primaria mediante el uso de prototipos iterativos (en cualquier fase del desarrollo), en el cual el usuario desempeña un rol activo y las herramientas computarizadas de desarrollo.
- Esta metodología engloba lo mejor de las metodologías anteriores que implican un mayor tiempo de desarrollo.
- Al tener un prototipo funcional de forma rápida permite un mayor acercamiento con el cliente y, a la vez, se genera mayor compromiso de este con el sistema, ya que lo siente parte de él, al ser testigo de su desarrollo (Benyon-Davies, 1998).

Herramientas CASE

Para lograr lo anterior se requieren de herramientas que permitan no solo la generación de código, como las herramientas CASE, que se definen como "Conjunto de herramientas y métodos asociados que proporcionan asistencia automatizada en el proceso de desarrollo del software a lo largo de su ciclo de vida" (Piattini, Calvo-Manzano, Cervera, Fernández, 1996); que aunque realizan la generación de partes de la aplicación, su producto final no puede ser mostrado al cliente, sin antes realizar los ajustes necesarios para convertirlo en algo ejecutable y funcional.

Herramientas RAD

Existen también las herramientas RAD (rapid application development) que evolucionaron de las anteriores, brindando la enorme posibilidad de generar aplicaciones funcionales, cuyos productos pueden interactuar con el cliente para realizar las evaluaciones y verificar si cumplen con los requerimientos (McBride, 2002).

Como ejemplos de estas herramientas podemos encontrar múltiples desarrollos que nos faciliten esta tarea, dentro de las que destacan:

- GeneXus.
- Leonardi.
- CakePHP (figura 4).
- Scriptcase (figura 5).
- WinDev (figura 6).
- 4D.

- Clarion.
- Codecharge Studio.
- FourJs Genero .
- Magic eDeveloper.
- PCSoft WinDev.
- PowerBuilder.
- WebSpeed Workshop.

Figura 4. CakePHP

Fuente: http://cakephp.org/

Figura 5. Scriptcase

Fuente:http://www.scriptcase.net/phpgenerator/home/home.php

Figura 6. WinDev

Fuente: http://www.windev.com.mx/

Herramientas RADD

Como una evolución a las herramientas anteriores, surge un subtipo denominado RADD, en el cual a la diferencia principal se denota en la última sigla *rapid application development and deployment,* que se podría realizar una traducción libre considerando su uso como: desarrollo rápido de aplicaciones y *puesta a punto*.

Este último concepto se refiere a los ajustes finales que se realizan a cada producto, si lo comparamos con la construcción de una casa habitación, sería tanto como una vez entregada la misma al cliente, es necesario realizar los *acabados* necesarios para que quede como se había idealizado, lo que lleva a la inversión de un mayor número de horas hombre, que se traducen en dinero.

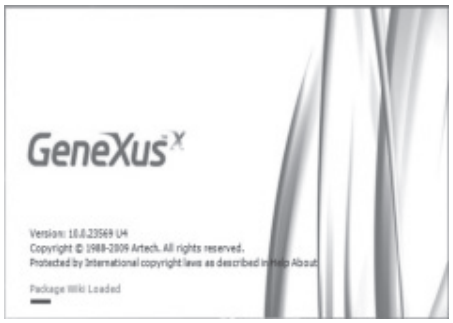
En las herramientas RADD estos detalles se pueden realizar sin problemas, ya que conservan la estructura básica y a partir de ella realizan las modificaciones correspondientes. Como tal, no existen muchas herramientas que ingresen en esta clasificación: por lo que a continuación se describe una de ellas.

GeneXus

Básicamente, es un programa que hace programas. Es una herramienta que parte de las visiones de usuarios, y a partir de ahí encapsula el conocimiento en lo que se llama *base de conocimiento*. Sistematiza ese conocimiento y desde allí automatiza el diseño, construcción y mantenimiento de la base de datos y los programas (Artech, 2008) (figura 7).

En resumen, es una herramienta basada en el conocimiento que diseña, genera y mantiene en forma automática los programas y la base de datos para lograr el desarrollo rápido de aplicaciones críticas en múltiples plataformas (figura 8).

Figura 7. Genexus X



Fuente: http://www.genexus.com/portal/hgxpp001.aspx

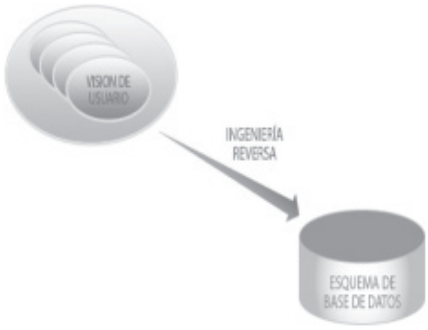
Figura 8. Genexus X Start Page

Fuente: http://www.genexus.com/portal/hgxpp001.aspx

Entender las necesidades del usuario final es una de las pocas tareas que no se pueden automatizar en el desarrollo de software. Al referirse a los desarrolladores de GeneXus como analistas de negocios en lugar de programadores, codificadores o desarrolladores (Márques y Fernández, 2008, p. 33).

La Metodología GeneXus se basa en la descripción de las entidades del usuario final (objetos reales tanto tangibles como intangibles) con las que se debe manejar su aplicación. Esto se hace describiendo las visiones de los usuarios finales sobre estas entidades, con un alto nivel de abstracción. Por este motivo, se realiza una programación declarativa y los analistas de negocios describen la realidad para que GeneXus cree el modelo de datos en una BD especificada y construya los programas de aplicación para cubrir las necesidades funcionales requeridas. Cuando esa realidad cambie, los analistas de negocios simplemente deberán describir la nueva realidad y será GeneXus el que se encargará de realizar los cambios necesarios en el modelo de datos y los programas para representar la nueva realidad descripta (figura 9).

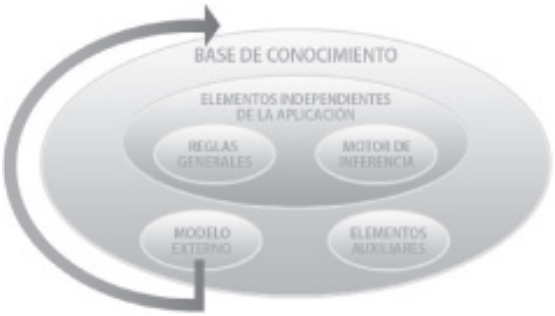
Figura 9. Visiones de los usuarios



Fuente: http://www.genexus.com/portal/hgxpp001.aspx

Para lograr todo esto, GeneXus tiene una base de conocimiento, que inicialmente tiene asociado un conjunto de mecanismos de inferencia y algunas reglas de aplicación general, como las que aseguran la consistencia (las de integridad referencial, por ejemplo). Luego, cuando el analista GeneXus comienza a describir la realidad creando objetos, estas descripciones (el modelo externo) son sistematizadas automáticamente y pasan a estar contenidas en la base de conocimiento. Además, sobre ese conocimiento, obtiene un conjunto de resultados que le ayudan a mejorar la eficiencia de las inferencias posteriores (figura 10).

Figura 10. Base de conocimiento



Fuente: http://www.genexus.com/portal/hgxpp001.aspx

Conclusiones (t1)

Dichas herramientas se han analizado en diversos ámbitos académicos tanto del Instituto Tecnológico de Chilpancingo, la Universidad Americana de Acapulco, como de la Unidad Académica de Ingeniería de la Universidad Autónoma de Guerrero. En las primeras instituciones, los estudiantes de las carreras de Ingeniería en Sistemas Computacionales e Ingeniería en Computación, respectivamente, han tenido contacto con dichas herramientas en sus versiones libres y de prueba en talleres de programación avanzada. Por su parte los estudiantes de la UAI de la UAG, cursan la carrera de Ingeniero en Computación y desarrollan actividades referentes al tema en la asignatura denominada Herramienta CASE.

Una vez evaluadas las opciones han seleccionado a la herramienta GeneXus para el desarrollo e implantación de proyectos finales de clase; ya que algunos consideran que es una herramienta que permite agilizar el aprendizaje.

Referencias

Artech (2008). *GeneXus X: Quick* Benyon-Davies, P..

Benyon-Davies, P. (1998). *Rapid Application Development: A review an Case Study*. Kane Thompson Centre. Recuperado de: http://www.com.glam.ac.uk/SOC_Server/research/gisc/RADbrfl.htm

CMS (2008). *Selecting a Development Approach*. USA: Department of Health & Human Services.

Fisher, P. et ál. (2010) System Development Life Cycle Models and Methodologies, Canadian Society for International Health Certificate Course in Health Information Systems. *Module 3: System Analysis & Database Development, Part 3: Life Cycle Models and Methodologies*. Recuperado de: http://famed.ufrgs.br/pdf/csih/mod3/Mod_3_3.htm

Green, D. y DiCaterino, A. (1998). *A Survey of System Development Process Models*. Center for Technology in Government. Recuperado de: http://www.ctg.albany.edu/publications/reports/survey_of_sysdev/

Laudon, J.P. y Laudon, K.C. (2008). *Sistemas de Información Gerencial*. Pearson.

Márques, D. y Fernández, C. (2008). *Genexus X Episodio Uno*. Artech

McBride, J.R. (2002). *Introduction to Systems Analysis, Topic 19, Rapid Application Development*. Prentice Hall. Recuperado de: http://www.scs.uvic.ca/~jmcbride/c375f19.pdf

Nigth, L. et ál. (2001). *System Development Methodologies for Web Enabled E-Business: A Customization Paradigm*. Recuperado de: http://www.kellen.net/SysDev.htm

Piattini, J.A., Calvo-Manzano, J, Cervera, Fernández, L. (1996). Análisis y diseño detallado de aplicaciones informáticas de gestión. Madrid: Ed. Ra-Ma.