

Aplicación del modelo de prototipos: Caso de estudio Software RedbotGamesShop**Application of the prototype model: Case study RedbotGamesShop Software**

ARMENTA-BENITEZ, Brayan†, RODRIGUEZ-ESPINOZA, Indelfonso*, MEDINA-MUÑOZ, Luis Arturo y GONZALEZ-LÓPEZ, Samuel

ID 1^{er} Autor: *Brayan, Armenta-Benitez*

ID 1^{er} Coautor: *Indelfonso, Rodríguez-Espinoza* / ORC ID: 0000-0001-6405-9886, Researcher ID Thomson: I-1513-2018, CVU CONACYT ID: 346230

ID 2^{do} Coautor: *Luis, Medina-Muñoz* / ORC ID: 0000-0001-9598-1451, Researcher ID Thomson: H-4735-2018, CVU CONACYT ID: 454558

ID 3^{er} Coautor: *Samuel González-López* / ORC ID: 0000-0002-1511-1227, Researcher ID Thomson: N-6460-2018, CVU CONACYT ID: 345102

Recibido: 20 de Junio, 2018; Aceptado 29 de Agosto, 2018

Resumen

El objetivo es implementar el modelo de prototipos, utilizado en la Ingeniería de software, pertenece a los modelos de desarrollo evolutivo, con el fin de motivar a los alumnos en la asignatura de programación. En todo desarrollo de sistemas software, es de suma importancia el seguir algunas especificaciones, por lo cual es necesario aplicar metodologías o alguna una disciplina que durante todas las etapas del desarrollo ayuden a su elaboración, aún más cuando se pretende que alumnos de nivel TSU se les enseña a programar. La mayoría de los proyectos que se desarrollan, finalizan tarde y cuesta mucho más de lo estimado. La Ingeniería de Software es una disciplina que ofrece métodos y técnicas, razón básica por la que se requiere disponer de un proceso de desarrollo es mejorar la seguridad de trabajo eliminando riesgos innecesarios y conseguir y mantener calidad en el producto a desarrollar, utilizando la esencia básica del modelo que dice que el prototipo debe ser construido en poco tiempo y no se debe utilizar muchos recursos.

Aplicación, Ingeniería de software, modelo de prototipos

Abstract

The objective is to implement the model of prototypes, used in software engineering, belongs to the models of evolutionary development, in order to motivate students in the programming subject. In any development of software systems, it is very important to follow some specifications, which is why it is necessary to apply methodologies or some discipline that during all stages of development help in their elaboration, even more when it is intended that students of TSU level teaches them to program. Most of the projects that are developed, end late and cost much more than estimated. Software Engineering is a discipline that offers methods and techniques, basic reason why it is required to have a development process is to improve job security by eliminating unnecessary risks and achieve and maintain quality in the product to be developed, using the basic essence of the model that says that the prototype should be built in a short time and not many resources should be used.

Application, Software engineering, prototype model

Citación: ARMENTA-BENITEZ, Brayan, RODRIGUEZ-ESPINOZA, Indelfonso, MEDINA-MUÑOZ, Luis Arturo y GONZALEZ-LÓPEZ, Samuel. Aplicación del modelo de prototipos: Caso de estudio Software RedbotGamesShop. Revista de Simulación Computacional. 2018. 2-5: 8-13.

* Correspondencia al autor (correo electrónico: irodriguez@utnogales.edu.mx)

† Investigador contribuyendo como primer autor

Introducción

Disciplina que establece el uso de principios de ingeniería robustos, orientados a obtener software económico, que sea confiable y funcione de manera eficiente [1]. Las metodologías para el desarrollo de software se consideran una serie de pasos que se realizan con exactitud para que, de tal forma, se pueda estructurar, planear y controlar el proceso de desarrollo de software.

Aquí es donde personas adquieren distintos roles temporales y acorde a ellos se realizan distintas actividades, hay que tomar en cuenta el tamaño y las características del software a desarrollar para así, usar una correcta metodología acorde al proyecto.

La elección de este tema fue dada gracias a una observación propia, basada en la institución donde actualmente curso mis estudios, y en la forma en que mis compañeros han comenzado con el desarrollo de software sin apearse a un modelo en específico.

Es muy importante utilizar un modelo de desarrollo de software, ya que, después de realizar una debida investigación se llegó a la conclusión de que la utilización de dichos modelos facilita la planeación, el desarrollo y la culminación de un proyecto de software y que también facilita una mayor satisfacción para el cliente y sus requerimientos.

Desarrollar un software significa construirlo simplemente mediante su descripción. Tomando en cuenta el concepto anterior, desarrollar un software es entender los requerimientos del cliente y desarrollarlos orientándonos a un modelo de desarrollo de software.

De acuerdo con Sommerville, un modelo del proceso del software es una representación abstracta de un proceso del software. Cada modelo de proceso representa un proceso desde una perspectiva particular, y así proporciona sólo información parcial sobre ese proceso. Se especifica que los modelos generales no son descripciones definitivas de los procesos del software, mas bien, son abstracciones de los procesos que se pueden utilizar para explicar diferentes enfoques para el desarrollo de software [9].

Metodología

Modelo de prototipos

Pertenece a la rama de los modelos evolutivos tales como el modelo incremental, en este modelo se hace el uso de prototipos los cuales represe tan partes esenciales del software, sin embargo, a diferencia del modelo incremental los prototipos no son funcionales y en su mayoría únicamente corresponden a representaciones del diseño visible para el usuario y no del diseño interno tales como algoritmos y manejo de la Información, a al desarrollador una mejor comprensión de lo que hay que hacer [2].

Características del modelo

Su característica principal es la retroalimentación, este modelo permite crear una muestra de la solución, para comprobar si será funcional, esta se consulta con el cliente y en base a la opinión del cliente, se hacen las modificaciones necesarias hasta llegar a un modelo final que satisfaga las necesidades del cliente en cuestión. Sus ventajas principales son el menor fallo de errores ya que se encuentra con una constante comunicación con el cliente. Además de que es una excelente herramienta para la recolección de requisitos del cliente.

Que hacer, a lo largo de todo el ciclo de vida del software, para construir un producto bueno, de calidad, dentro del presupuesto y a tiempo. La utilización de técnicas para un desarrollo de software de calidad y que cumpla con todos los requerimientos que el cliente solicitó.

En el proyecto del desarrollo de software la metodología, como se aprecia en la figura 1, puntualiza “quién debe hacer qué, cuándo y cómo debe hacerlo”, no existe una metodología de software universal.

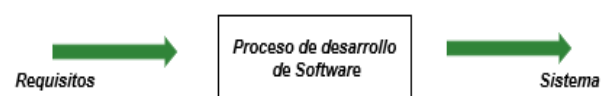


Figura 1 Proceso de desarrollo del software

El modelo es rentable cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.

También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción usuario-máquina.

Tomando en cuenta las características ya mostradas así como también sus ventajas y desventajas, se hizo un análisis de dicha metodología y se comprendió que fue la acorde al proyecto ya realizado, argumentando que para caso práctico-estudio, funciona bastante bien, este modelo fue aplicado por un estudiante el cual pretendía como objetivo el desarrollo de un software, y consultando con el cliente, se llegó a la conclusión, que los requisitos eran muy claros y precisos, por lo cual se eligió la metodología como tal, aplicándose a una de sus ventajas principales.

Se ejecutó la metodología partiendo del primer paso, saber los requisitos del cliente, y tomar en cuenta la claridad de los mismos, para así partir con el siguiente paso, que es la ejecución del prototipo para saber si será funcional para el cliente, una vez decidido y dialogado los cambios a realizar mediante su debida retroalimentación como lo indica el modelo, partimos a la mejora del prototipo en cuestión, y así, sucesivamente utilizando la característica principal del modelo, que es la retroalimentación, hasta llegar a un sistema finalizado que cumpla con todos los requerimientos solicitados.

Ciclo de desarrollo del modelo prototipos

Los sistemas prototipo desarrollan un comportamiento que satisface los requisitos y necesidades que se han entendido claramente. Realiza, por tanto, un proceso real de datos, para contrastarlo con el usuario. Se va modificando y desarrollando sobre la marcha, según las apreciaciones del cliente (el maestro).

Esto hace más lento el proceso de desarrollo y disminuye la fiabilidad, puesto que el software está constantemente variando, pero, a la larga, genera un producto más seguro, en cuanto a la satisfacción de las necesidades del cliente.

Una aplicación terminada es de calidad si cumple rigurosamente con todos y cada uno de sus requisitos.

Como se muestra en la figura 2, es muy fácil comprender la metodología de desarrollo de prototipos funcional evolutiva, primeramente se realiza un debido análisis en donde se está en constante comunicación con el cliente para determinar todos los requerimientos solicitados y saber si los objetivos son totalmente claros, luego nos pasamos al diseño, en donde ya se incluye como parte de la elaboración del prototipo, continuando con la codificación incluyendo en gran parte del desarrollo, después de realizan las pruebas pertinentes, y en base a ello, surge la retroalimentación para la evolución del software, el número de iteraciones, variara dependiendo los requerimientos y/o modificaciones solicitadas por el cliente durante el proceso.

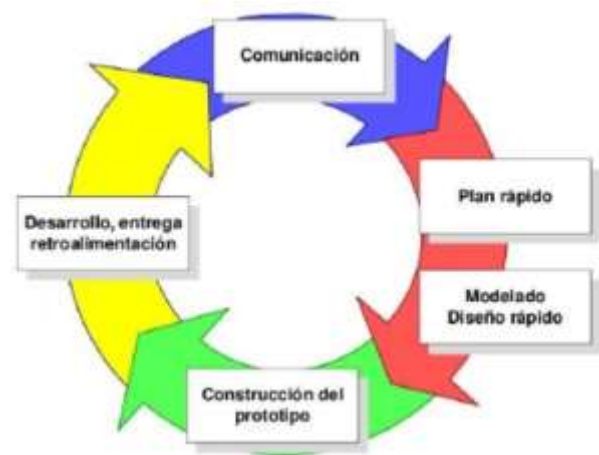


Figura 2 Gráfica del ciclo de desarrollo del modelo de prototipos [4]

De acuerdo a lo citado en el párrafo anterior un prototipo puede tener alguna de las tres formas siguientes:

- Un prototipo, ejecutable en la computadora o en papel, que describa la interacción hombre-máquina y los listados de la aplicación.
- Un prototipo que realice algún subconjunto de la función requerida, y que sirva para evaluar la utilidad de un algoritmo o las necesidades de capacidad de almacenamiento y velocidad de cálculo del programa final.
- Un sistema que realice en todo o en parte la función deseada pero que tenga características como rendimiento y consideraciones.

Ventajas del modelo prototipos

Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.

Además, también ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.

Desventajas del modelo prototipos

Su principal desventaja es que una vez que el cliente ha dado su aprobación final al prototipo y cree que está a punto de recibir el proyecto final, se encuentra con que es necesario reescribir buena parte del prototipo para hacerlo funcional, porque lo más seguro es que el desarrollador haya hecho compromisos de implementación para hacer que el prototipo funcione rápidamente. Es posible que el prototipo sea muy lento, muy grande, no muy amigable en su uso, o incluso, que esté escrito en un lenguaje de programación inadecuado.

El cliente ve funcionando lo que para él es la primera versión del prototipo que ha sido construido con "plastilina y alambres", y puede desilusionarse al decirle que el sistema aún no ha sido construido [8].

El desarrollador puede ampliar el prototipo para construir el sistema final sin tener en cuenta los compromisos de calidad y de mantenimiento que tiene con el cliente [3].

Herramientas CASE

CASE es un acrónimo para Computer-Aided Software Engineering, aunque existen algunas variaciones para lo que actualmente se entiende por CASE, esencialmente, un CASE es una herramienta que ayuda al ingeniero de software a desarrollar y mantener software. A continuación, se presentan algunas definiciones dadas para el término CASE [7].

Las herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero principalmente.

Actualmente existe un gran desarrollo y una gran cantidad de este tipo de herramientas solo por mencionar algunas: Microsoft Project, Rational Rose, JDeveloper, Magic Draw, Visual Paradigm, Microsoft Visio, BoUML.

Resultados

Las pruebas realizadas a la aplicación RedbotGamesShop, consistieron en evaluaciones periódicas, mientras el maestro responsable de la materia las valoraba, conforme se avanzaba de acuerdo con la metodología aplicada.

Pruebas

En uno de los prototipos entregados el cliente solicitó un login funcional acorde a lo requerido, solicitó un entorno en donde todo el personal ingresara al sistema, validando el rango con el cual contaba el empleado a ingresar en este caso, se solicitó distinguir empleados comunes, de administradores del sistema, en donde un cajero común, solo tuviera acceso al entorno de cobro de productos, mientras que un usuario administrador tuviera acceso a un mayor número de información valiosa para el negocio. Esto surgió en base al proceso de la metodología de prototipo de la etapa de retroalimentación del prototipo, el resultado final se muestra en la figura 3, tomando en cuenta que los diferentes tipos de usuarios, tuvieran un mismo login, pero no el mismo tipo de acceso a la información del sistema, validando el nombre de usuario y clave de acceso, como lo vemos en la figura 3.

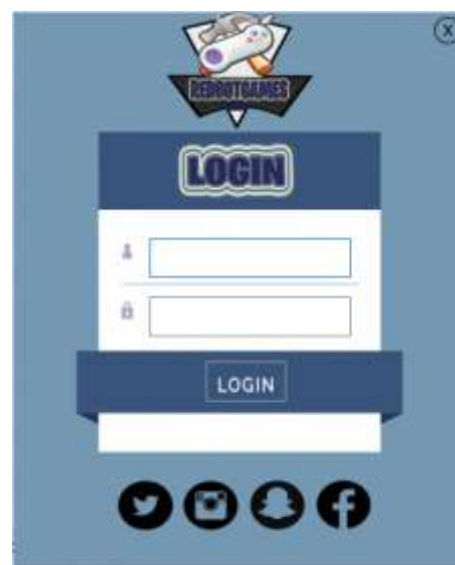


Figura 3 Acceso al del sistema

Como resultado se obtuvo un prototipo “Punto de Venta”, como se puede mostrar en la figura 4, que vienen a automatizar el proceso de salida de mercancía y cobro con el cual los empleados podrán generar de manera automática el “Ticket” de venta y la impresión de reportes para la toma de decisiones, así mismo el cierre de caja mostrara la información actualizada de las ventas y transacciones realizadas durante el día, además capturar la información de cada producto.

El utilizar una metodología permite organizar la forma en la que el proyecto será realizado y, de esta forma obligamos a revisar que los resultados son los correctos, antes de continuar con el siguiente paso, y a su vez marcar metas intermedias a realizar para controlar el avance del proyecto, además, la una inversión será mínima, para que esto sea efectivo, la metodología debe ser aplicada con rigor, como se fue aplicando en el prototipo mostrado, durante cada de revisión.



Figura 4 Pantalla prototipo de venta

Como se muestra en la figura 4 el cliente después de revisar el primer prototipo, nos solicitó una interfaz más amigable, pero a su vez sencilla, después de analizar el requerimiento, se aplicó su debida retroalimentación, como requisito esencial del modelo de desarrollo prototipo, logrando el resultado mostrado previamente.

```

RedbotGamesShop S.A. de C.V.
EXPEDIDO EN LOCAL PRINCIPAL
DIREC: DESCONOCIDA
TELEF: 6318000000
R.F.C.: XXXXXXXX-XX
EMAIL: aguillarbenitezbrayan@gmail.com

CAJA#1                                TICKET # 21
*****

CLIENTE: CLIENTE GENERICO
LE ATIENDE: JUAN

FECHA: 21/08/2018                      HORA: 20:41
*****
ARTICULO | CANT | PRECIO | IMPORTE
*****
ASSASINS CREED      1  1000  1000
FIFA                 1  1500  1500
RESIDENT EVIL        1   350   350
MINECRAFT            1   200   200
-----
SUBTOTAL.....$      3.050,00
IVA.....$              ,00
TOTAL.....$          3.050,00

EFFECTIVO.....$      3.100,00
CAMBIO.....$         -50,00

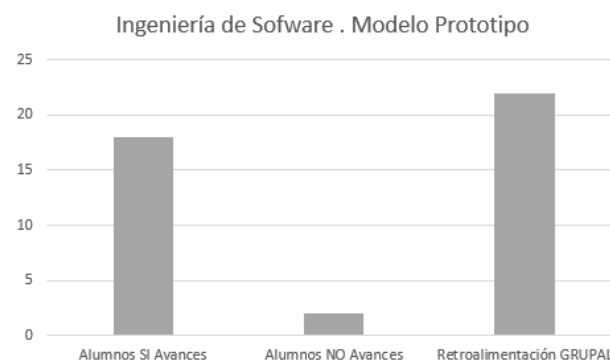
GRACIAS POR SU COMPRA! VUELVA PRONTO

```

Figura 5 Ticket de venta

Ya casi al finalizar la aplicación, el cliente solicitó la impresión automática del “Ticket” después de cada compra, se concluyó con un resultado exitoso, como se muestra previamente en la figura 5.

El trabajo docente desempeña sin duda un valor muy importante en la educación, la evaluación del rendimiento y adquisición de conocimientos en el proceso enseñanza-aprendizaje en los alumnos es tan sólo una de ellas, por lo que a continuación se muestran los resultados obtenidos de la aplicación del modelo prototipo en la gráfica 1.



Gráfica 1 Resultados de la aplicación del modelo prototipo

Como podemos apreciar en la gráfica anterior, el grupo avaluado corresponde al cuatrimestre mayo-agosto del año en curso con un total de 22 alumnos de la asignatura de programación.

Los alumnos que si fueron registrando su avance conforme lo pide el Modelo Prototipo y lo confirma el maestro en clases, con el fin de ir retroalimentado sobre el proyecto y corregir o disipar dudas en el alumno en caso de haberlas.

En la gráfica también se ve que el grupo también tuvo asesorías y retroalimentación grupal, esto con el fin de reforzar y avanzar todos a un mismo ritmo y que los alumnos compartan sus experiencias tanto de éxito como de fracaso. Para el caso de los alumnos que, por frustración, desmotivación e indiferencia por la asignatura no asistieron a lo mucho a dos consultas en el cuatrimestre es importante mencionar que su proyecto no cumplió con las perspectivas, aun cuando se le motivo y busco la manera de ayudarlo, sin caer o llegar a consentirlos.

Conclusiones

Las universidades tienen la responsabilidad de transmitir y preservar la educación, por ello es importante consolidar proyectos y promover este tipo de metodologías, argumentando que sirve de experiencia hacia el alumno, además se cumple con el programa educativo, así mismo se termina de manera favorable consiguiendo todos los requisitos solicitados por el cliente en este caso el maestro. La materia ha sido aprobada satisfactoriamente, para finalizar, se concluye con una aplicación totalmente funcional y a su vez, con una gran experiencia.

Esta investigación se centra en estudiar y proporcionar la metodología para la realización de software. El trabajo hasta ahora logrado ha despertado muchas curiosidades y ha dejado nuevos retos para el grupo.

Es importante mencionar que todos los modelos de software tienen ventajas y desventajas en su uso para el desarrollo de software, buscando el aprendizaje de los alumnos se deberá utilizar y aplicar la metodología que en su momento convenga.

References

S. Lawrence Pfleeger, Prentice Hall, Ingeniería de Software. Argentina, ISBN: 9789879460719, 2002.

Sommerville, I. "Ingeniería del Software". 10ª Edition, Addison-Wesley. 2016.

Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., Fernández, L. "Análisis y diseño detallado de aplicaciones informáticas de gestión". Rama. 2007 https://ecorfan.org/bolivia/researchjournals/Tecnologia_e_innovacion/vol2num5/Tecnologia_e_Innovacion_Vol2_Num5_6.pdf

[Bernd Bruegge, Dutoit Allen. Object-Oriented Software Engineering: Using UML, Patterns, and Java, 2004, Prentice Hall, segunda edición. Capítulo 15.

Pressman, Roger. (2002) "Ingeniería de Software: Un enfoque Práctico". McGraw Hill.

Bernadó, Ester y Garrell, Josep Maria y Román, Manuel y Salamó, Maria y Camps, Joan y Abella, Jaume. "Introducción a la programación en el ámbito de diversas ingenierías", Jenui, 1998.

Gómez Castro, R., Galvis Panqueva, A., Mariño Drews, O. (1998) Ingeniería de software educativo con modelaje orientado por objetos: un medio para desarrollar micromundos interactivos. Revista de Informática Educativa. 11 (1), 9 – 30

SOMMERVILLE Ian: Software Engineering, Addison-Wesley International Computer Science series, UK, 1993