

# Metodología Híbrida para Desarrollo de Software en México.

## CICIC 2012

**Eréndira M Jiménez-Hernández**

Tecnología de Software y Bases de Datos, Centro de Investigación en Computación (CIC), IPN.  
Ciudad de México, C.P. 07738, México

y

**Sandra D Orantes-Jiménez**

Tecnología de Software y Bases de Datos, Centro de Investigación en Computación (CIC), IPN.  
Ciudad de México, C.P. 07738, México

### RESUMEN

*En este artículo se presenta una metodología para desarrollo de software basada en la nueva tendencia en el área de Ingeniería de Software: las metodologías híbridas. La metodología que se propone, fue diseñada a partir de un estudio realizado con las empresas mexicanas dedicadas a desarrollar software.*

**Palabras clave:** Ingeniería de Software, Metodologías Ágiles, Metodologías Híbridas, Metodología Propuesta, Metodologías Tradicionales.

### 1. INTRODUCCIÓN

Actualmente existen muchas metodologías para el desarrollo de software, las cuales se engloban en dos grandes categorías: las metodologías tradicionales y las metodologías ágiles.

Las metodologías tradicionales propician las buenas prácticas que existen dentro de la Ingeniería de Software; sin embargo, requieren de mucha disciplina para seguir con el riguroso proceso que éstas conllevan [1].

Por otro lado, las metodologías ágiles presentan respuestas rápidas al cambio y son flexibles, aunque generan poca documentación y no hacen uso de métodos formales.

En este artículo se propone una metodología para desarrollo de proyectos de software basada en la nueva tendencia en el área de Ingeniería de Software: las metodologías híbridas.

Las metodologías híbridas incorporan algunas prácticas existentes dentro de las metodologías tradicionales y ágiles, proporcionando así una gran ventaja.

Para crear la metodología que se propone, el primer paso consistió en investigar la factibilidad de utilizar una metodología híbrida en el contexto actual de las empresas desarrolladoras de software en México.

Una vez que se comprobó por medio de un estudio la viabilidad, se diseñó la metodología, la cual considera las necesidades de dichas empresas y combina algunas prácticas existentes dentro de las metodologías RUP (*Rational Unified Process*, Proceso Unificado de Rational) [2], XP (*eXtreme Programming*, Programación Extrema) [3] y Scrum [4].

La metodología que se propone, se creó para ayudar a las empresas mexicanas; por lo que se consideró a México como caso de estudio.

Sin embargo, esta metodología puede ser utilizada en empresas de otros países.

### 2. MARCO CONCEPTUAL

De acuerdo con el Diccionario de la Real Academia Española, la palabra *metodología* es un conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal [5].

Retomando la definición anterior, en el área de Ingeniería de Software, el término *metodología* [1] se ha adecuado para referirse a un marco de trabajo empleado para estructurar, planificar y controlar el proceso de desarrollo de sistemas computacionales.

Por lo que al utilizar una metodología para desarrollo de software se espera que ésta pueda proveer un conjunto de prácticas y herramientas que faciliten el proceso de desarrollo; ofreciendo un producto que satisfaga las expectativas del cliente, con alta calidad y seguro.

#### 2.1. Metodologías Tradicionales

Algunos ejemplos de este tipo de metodologías son *Yourdon* [6], OMT (*Object Modeling Technique*, Técnica de Modelado de Objetos) [7], RUP, Métrica 3 [8], entre otras.

Las metodologías tradicionales contribuyeron positivamente al ser incrementales e iterativas; promovieron la asignación de roles dentro del equipo de desarrollo, facilitaron la división del sistema en varios subsistemas y fomentaron el reúso de componentes.

De manera general, las metodologías tradicionales consideraron la importancia de documentar el sistema, permitiendo así, entender, extender y mantener el software.

Sin embargo, tienen algunas desventajas, entre las cuales se puede señalar que se requiere de un alto grado de disciplina, no se tiene respuesta rápida a cambios, se genera documentación innecesaria y se invierte mucho tiempo en el modelado del sistema. Además estas metodologías tienen un plan de proyecto muy riguroso. Por lo que de manera general, no consideran que el análisis, el diseño y la construcción son impredecibles la mayoría de las veces [9].

#### 2.2. Metodologías Ágiles

El esquema propuesto por las metodologías tradicionales no resulta ser el más adecuado para muchos de los proyectos actuales en donde el entorno del sistema es muy cambiante y se

exige reducir drásticamente los tiempos de desarrollo. En este escenario nacieron las metodologías ágiles, que en esencia combinan una filosofía y un conjunto de directrices de desarrollo [10].

La filosofía busca la satisfacción del cliente y la entrega temprana del software incremental; equipos de proyectos pequeños y con alta motivación; métodos informales y un mínimo de productos de trabajo.

Las directrices de desarrollo resaltan la entrega sobre el análisis y el diseño, la comunicación activa y continúa entre los desarrolladores y los clientes.

Las metodologías ágiles, al igual que las metodologías tradicionales poseen ciertas ventajas y desventajas.

Algunas de las ventajas de las metodologías ágiles es que presentan respuestas rápidas y efectivas al cambio. Tienen un plan de proyecto flexible y una gran simplicidad de manera general en el desarrollo.

Sin embargo, las metodologías ágiles generan poca documentación y no hacen uso de métodos formales.

Algunas de las metodologías ágiles más conocidas son: *XP*, *Crystal* [11] y *Scrum*.

### 2.3. Metodologías Híbridas

Hay una gran diversidad de metodologías, aunque todas ellas caen dentro de alguna de las dos clasificaciones mencionadas: ágiles o tradicionales.

Sin embargo, existe una nueva categoría: las metodologías híbridas.

Las metodologías híbridas pretenden retomar las ventajas de las metodologías existentes, de tal forma que son una combinación de las mejores prácticas descritas en cada una de ellas.

Dentro las metodologías híbridas se puede mencionar a *EssUP* (*Essential Unified Process*, Proceso Esencial Unificado) [12] como la pionera.

**EssUP:** es una metodología creada por Ivar Jacobson en el 2010, basada en el Proceso Unificado (PU) [13], los métodos ágiles y la madurez de procesos.

EssUP es ágil porque no pretende imponer un proceso específico, además toma en cuenta que es necesario tener flexibilidad y respuestas rápidas ante los cambios.

Sin embargo, EssUP menciona que es necesario documentar y modelar en UML [14], con lo cual retoma una importante característica de las metodologías tradicionales.

Por lo que, EssUP es una metodología híbrida, aunque conceptualmente, ya que en la práctica el equipo de desarrollo de software que pretenda utilizar esta metodología debe seleccionar el modelo de ciclo de vida de desarrollo de software que mejor se adapte a sus necesidades, asignar los roles que crean convenientes y seleccionar las mejores prácticas; con lo cual se presenta un gran problema si no se tiene la experiencia y el conocimiento necesario para saber elegir las mejores prácticas dentro de la Ingeniería de Software y aplicarlas de manera adecuada en cada proyecto.

EssUP presenta una nueva tendencia en metodologías para desarrollo de software, ya que intenta retomar algunas ventajas de las metodologías tradicionales y de las ágiles, convirtiéndola en la primera metodología híbrida.

## 3. CASO DE ESTUDIO

En México, más del 50% de las empresas dedicadas al desarrollo de software son candidatas a utilizar una metodología híbrida. Por lo cual, las metodologías existentes al ser

tradicionales o ágiles, no se acoplan a las necesidades actuales de las empresas.

Así que, diseñar una metodología híbrida para que la puedan utilizar las empresas de desarrollo de software en México, es una buena opción para incrementar la productividad en dichas empresas.

Lo anterior se comprobó a través de un estudio realizado con las empresas de software en México.

Para realizar el estudio, se diseñó una encuesta con 19 preguntas, con respuestas de opción múltiple. Además se recurrió a los datos estadísticos de INEGI (Instituto Nacional de Estadística y Geografía). Según INEGI [15], en el año 2010 se contabilizaron 9540 empresas en México dedicadas al desarrollo de software.

Por lo que, para calcular el tamaño de la muestra, se aplicó la encuesta a un grupo piloto de 20 empresas y se obtuvieron los valores estadísticos de  $p$  y  $q$  respectivamente, mismos que se utilizaron en la Ec. (1).

$$n = \frac{Z^2 pqN}{NE^2 + Z^2 pq} \quad (1)$$

Donde:

- $n$  tamaño de la muestra
- $Z$  nivel de confianza
- $p$  variabilidad positiva
- $q$  variabilidad negativa
- $N$  tamaño de la población
- $E$  precisión o el error

Considerando que el nivel de confianza es del 95% y con una población de 9540 empresas, se obtuvo el tamaño de la muestra sustituyendo como en la Ec. (2):

$$n = \frac{(0.95)^2(0.40)(0.60)(9540)}{(9540)(0.05)^2 + (0.95)(0.40)(0.60)}$$

$$n = 85.86$$

$$n \approx 86 \quad (2)$$

De la lista de empresas dedicadas a desarrollar software en México que proporcionó INEGI, se seleccionaron aleatoriamente a 86 para aplicarles la encuesta.

Es importante mencionar que la selección de las empresas a las cuales se les aplicó la encuesta fue en forma aleatoria, que no todas las empresas señaladas por INEGI se pudieron encontrar y que tampoco no todas mostraron interés en participar, de tal manera que en ese caso fue sustituida en forma aleatoria por otra empresa hasta completar las 86 de la muestra.

Al analizar los datos, se encontró que el número de empresas que prefieren los tres tipos de metodologías se distribuyen de la siguiente manera:

Tipo de metodología	Número de empresas
Metodologías Ágiles	22
Metodologías Híbridas	50
Metodologías Tradicionales	14
TOTAL	86

De donde se puede observar que el número de empresas que prefieren metodologías híbridas es 50 y no híbrida es 36. Para determinar si una empresa tiene inclinación por usar una determinada metodología no se hizo con una sola pregunta, sino considerando los valores de las preguntas que distinguen a dicha metodología. En las preguntas donde era permitido responder más de una opción, se sumó la puntuación y se dividió entre el número de respuestas contestadas de una determinada pregunta para encontrar el promedio. La prueba de hipótesis se muestra a continuación:

$$H_0: p_1 \geq 0.50$$

$$H_1: p_1 < 0.50$$

$$\frac{z_\alpha}{2} = \frac{z_{0.05}}{2} = z_{0.025} = -1.96$$

Donde:

$p_1$ : Proporción de empresas desarrolladoras de software con una inclinación hacia metodologías híbridas

La interpretación de las hipótesis es la siguiente:

$H_0$ : El 50% o más de las empresas desarrolladoras de software tienen una inclinación hacia el uso de metodologías híbridas.

$H_1$ : Menos del 50% de las empresas desarrolladoras de software tienen una inclinación hacia el uso de metodologías híbridas.

El estadístico de prueba es:

$$z = \frac{\hat{p} - p}{\sqrt{\frac{p(1-p)}{n}}} \quad (3)$$

Donde:

$z$  estadístico de prueba

$\hat{p}$  proporción de la muestra que tiene inclinación hacia el uso de metodologías híbridas

$p$  proporción de la población = 0.50

$n$  tamaño de la muestra

Sustituyendo los valores correspondientes se puede ver la Ec. (4):

$$z = \frac{\left(\frac{50}{86}\right) - 0.50}{\sqrt{\frac{0.50(1-0.50)}{86}}} = 1.5096 \quad (4)$$

La información obtenida en el estudio tiene una distribución normal, de tal manera que con un nivel de significancia (o error) del 5% el área de rechazo o no rechazo de la hipótesis nula  $H_0$  es como se muestra en la siguiente figura:

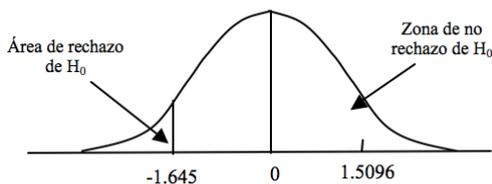


Figura 1. Área bajo la curva de la distribución normal

NOTA: El valor del área bajo la curva normal (-1.645) se obtuvo de tablas estadísticas [16]

Se puede observar en la Figura 1 que el estadístico de prueba cae en la zona de no rechazo, por lo tanto no se rechaza  $H_0$  y se dice que:

*“El 50% o más de las empresas desarrolladoras de software tiene una inclinación hacia el uso de metodologías híbridas”.*

Para corroborar la prueba de hipótesis se realizó un intervalo de confianza con un nivel del 95%. Los límites de dicho intervalo se muestran a continuación en la Figura 2:

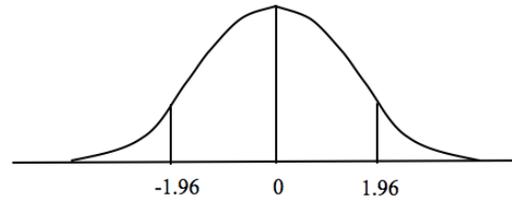


Figura 2. Intervalo de confianza

En donde:

$$\hat{p} - z_{\frac{\alpha}{2}} \sqrt{\frac{pq}{n}} < p < \hat{p} + z_{\frac{\alpha}{2}} \sqrt{\frac{pq}{n}}$$

$$\frac{50}{86} - 1.96 \sqrt{\frac{\left(\frac{50}{86}\right)\left(1 - \frac{50}{86}\right)}{86}} < p < \frac{50}{86} + 1.96 \sqrt{\frac{\left(\frac{50}{86}\right)\left(1 - \frac{50}{86}\right)}{86}}$$

$$0.4770 < p < 0.6855 \quad (5)$$

Lo cual implica que con un error del 5% la proporción de desarrolladores de software que prefieren metodologías híbridas se encuentran entre el 47.77% y el 68.5%, tal como se observa en la Ec. (5).

Mediante el estudio realizado con las empresas dedicadas al desarrollo de software se comprobó que dichas empresas tienen una inclinación y son candidatas a utilizar la nueva tendencia en el área de Ingeniería de Software: las metodologías híbridas.

Además, el estudio permitió encontrar información que sirvió para diseñar la metodología que se propone. A continuación se menciona de manera resumida dichos resultados adicionales:

- El 65% de las empresas tienen entre 1 y 5 años de antigüedad.
- El 79% de las empresas desarrollan aplicaciones Web.
- En el 67% de las empresas el tiempo de desarrollo de los proyectos oscila entre 2 y 6 meses.
- El 82% de las empresas tienen equipos conformados por a lo más 10 personas.
- El 73% de las empresas utilizan el Paradigma de Programación Orientado a Objetos.
- El 59% de las empresas invierten más del 50% del tiempo de desarrollo en la codificación.
- El 88% de las empresas realizan sólo pruebas de caja negra.
- Las metodologías más utilizadas son: XP (36%), RUP (26%) y Scrum (16%).
- El 67% de las empresas no utiliza estándares para aseguramiento y gestión de la calidad.
- El 52% de las empresas utilizan para modelar UML.

En este estudio, se puede ver que una metodología híbrida será de gran ayuda para las empresas desarrolladoras de Software de la República Mexicana, pero dadas las similitudes de México con los países latinoamericanos seguramente estos resultados se pueden extrapolar también para esos países, con las respectivas diferencias propias de cada una de esas naciones, empresas de desarrollo de software y grupos de trabajo.

## 4. METODOLOGÍA PROPUESTA

La metodología que se propone es una metodología híbrida para desarrollo de proyectos de software. Fue creada tomando en cuenta las necesidades actuales de las empresas mexicanas dedicadas al desarrollo de software. Esta metodología combina algunas prácticas existentes dentro de las metodologías RUP, XP y Scrum; por lo cual es un híbrido entre lo tradicional y lo ágil.

Para utilizar esta metodología basta saber:

- Los roles que deben de existir.
- Los principios metodológicos.
- El proceso de desarrollo.

### 4.1. Características de la Metodología

La metodología propuesta fue diseñada para desarrollar proyectos de software con las siguientes características:

- Proyectos de desarrollo de aplicaciones Web como *e-business* (*electronic-business*, negocio electrónico) [17], también *e-commerce* (*electronic-commerce*, comercio electrónico) [17].
- Proyectos que se desarrollen en un lapso de 2 a 6 meses.
- Equipos de desarrollo conformados a lo más por 10 integrantes (sin contar a los usuarios y al cliente).

### 4.2. Roles de la Metodología

En esta metodología se plantea que deben existir los siguientes roles o papeles dentro del equipo de desarrollo, véase Tabla 1:

Tabla 1. Roles

ROL	DESCRIPCION
<b>Cliente</b>	Especifica los requerimientos y financia el proyecto de software.
<b>Líder del proyecto</b>	Se encarga de negociar los proyectos, por lo que es el integrante del equipo de desarrollo que tiene más relación con el cliente/usuarios. Es además el intermediario entre el cliente y el administrador del proyecto. Debe de tener la habilidad de unir ideas, personas y recursos. Así como tener facilidad para la toma de decisiones.
<b>Administrador del proyecto</b>	Se encarga de coordinar a los programadores, al probador y al documentador. Además organiza las reuniones necesarias para analizar los requerimientos, realizar el diseño, el plan de proyecto y las pruebas.
<b>Programador</b>	Son los responsables de codificar el diseño.
<b>Probador</b>	Su función consiste en verificar que se realizan las actividades de manera adecuada en cada fase del proceso de desarrollo de software. Además deberá cumplir con la tarea de asegurar la calidad, haciendo uso del Ciclo y los 14 Principios de Deming [18].
<b>Documentador</b>	Su función principal es generar los documentos que respalden y documenten lo que se va generando a lo largo del proceso de software. Ayuda en la elaboración de los rotafolios.
<b>Usuarios finales</b>	Son las personas que interactúan con el software una vez que se libera para su uso productivo.

En esta metodología se retoma una importante característica de las metodologías ágiles al incluir a los usuarios finales y al cliente como parte del equipo de desarrollo.

### 4.3. Principios Metodológicos

Para poder utilizar adecuadamente la metodología se deben conocer los siguientes principios metodológicos:

- Hacer uso de papel rotafolio.
- Realizar lluvias de ideas durante las juntas.
- Hacer uso del Ciclo de Deming para la Gestión de Riesgos durante todo el proceso de desarrollo.
- Considerar los 14 Principios de Calidad de Deming, para garantizar calidad en el proceso y en el producto de software.

Por lo que con dichos principios se guían las actividades dentro del proceso de desarrollo de esta metodología.

### 4.4. Proceso de Desarrollo

Como se puede apreciar en la Figura 3, la metodología tiene 4 etapas, las cuales se describirán más adelante:

1. Planteamiento
2. Preparación
3. Construcción
4. Implantación



Figura 3. Proceso de desarrollo

**Planteamiento.** En esta etapa se deben cumplir los siguientes objetivos:

- Definir los requerimientos del proyecto que se va a desarrollar
- Realizar la estimación de costos de tiempo, de recursos (tanto materiales como humanos) y monetario. Se sugiere hacer uso de métodos de estimación como Puntos de Función y/o COCOMOII.
- Elaborar, pactar y firmar un contrato que estipule los costos y los requerimientos, así como penalizaciones y otros aspectos legales.
- Elaborar un plan general del proyecto. El primer paso para crear el plan general, consiste en calcular el tiempo que se tendrá para llevar a cabo cada una de las 4 fases. Después se debe realizar una lista de actividades, así como asignarle la precedencia a cada una.

En esta etapa, se puede hacer uso de una serie de ciclos, considerando que el software tiene una naturaleza evolutiva; estos ciclos servirán para pactar acuerdos entre los involucrados.

**Preparación.** En esta etapa se deben establecer las tareas y responsabilidades entre los miembros del equipo. Por lo que esta etapa se resume en la realización del análisis y el diseño a través de algunas iteraciones si se requieren, es decir a

través de varias juntas en donde el líder de proyecto, el administrador del proyecto, los programadores, el probador y el documentador colaboren con su experiencia y sus ideas para poder llevar a cabo los objetivos de esta fase.

En esta etapa se deben realizar los diagramas de Entidad-Relación (en caso de ser necesarios), los diagramas de interfaz, los diagramas de navegación y los diagramas de UML que se requieran. Los diagramas deberán estar pegados en un lugar visible para todos, para agilizar la comunicación.

También deben seleccionarse los lenguajes de programación y las tecnologías Web (como JSP, PHP, ASP).

**Construcción.** En esta etapa se lleva a cabo la codificación del diseño que se planteó en la fase de preparación. Además se hará uso de algunas prácticas propuestas en otras metodologías como:

- Juntas semanales de 1 hora. Estas juntas permitirán generar un ciclo de comunicación en esta etapa; de tal manera, que se puedan identificar avances y problemas.
- Programación en parejas. La programación en parejas se llevará a cabo, tal como lo propone XP. Con la finalidad de agilizar la codificación, pero esto se aplicará cuando se tenga poco tiempo para desarrollar la aplicación y se pueda contar con los recursos necesarios.

**Implantación.** En esta etapa se instala el software de manera operativa y se llevan a cabo las pruebas finales con los usuarios a través de una serie de retroalimentaciones.

## 5. CONCLUSIONES

Las metodologías de desarrollo de software han ido evolucionando a medida que los paradigmas de programación también lo han hecho, de tal manera que se acoplaron a ellos tratando de ajustarse a las nuevas necesidades.

Dichas metodologías han proporcionando herramientas y prácticas que ayudan en el proceso de desarrollo de software.

Pero a pesar de la evolución que han tenido no existe una metodología que se aplique para todos los tipos de proyecto de software, por lo cual es necesario investigar cuáles son las necesidades reales del ámbito donde se desea implementar una determinada metodología. Así como tomar en cuenta las prácticas de las metodologías existentes para poder seleccionar las que mejor convengan.

Se puede observar en los resultados de la investigación, que las empresas que se dedican a desarrollar software en México no solamente son candidatas para usar metodologías híbridas, sino que seguramente obtendrán mejores resultados al desarrollar software con una metodología híbrida.

Por lo que esta metodología ayudará a estas empresas a mejorar la calidad de sus productos de software a menor costo y con un tiempo de desarrollo menor.

La metodología será de gran ayuda para las empresas desarrolladoras de Software de la República Mexicana, pero dadas las similitudes de México con los países latinoamericanos seguramente estos resultados se pueden extrapolar también para esos países, con las respectivas diferencias propias de cada una de esas naciones, empresas de desarrollo de software y grupos de trabajo.

La metodología se pretende implantar en algunas empresas mexicanas dedicadas a desarrollar software, esto permitirá probar la metodología, para hacer las adecuaciones necesarias. Además, se tiene contemplado terminar la codificación de la herramienta que servirá de apoyo a la gestión del proyecto al proporcionar la automatización de los elementos que se

mencionan en el proceso de desarrollo de la metodología, como los diagramas de UML, el contrato, el plan de proyecto, etc.

Esta investigación muestra que es factible utilizar una metodología híbrida como la presente metodología para desarrollar software en México. Además, proporciona información que puede ser empleada para otras investigaciones relacionadas con el área de Ingeniería de Software en México.

## 6. REFERENCIAS

- [1] Pressman, Roger. "Software Engineering: a practitioner's approach" (6th ed.). McGraw Hill. 2005
- [2] IBM, "IBM Rational Unified Process (RUP)". Extraído el 18 de Septiembre del 2011, de <http://www-01.ibm.com/software/awdtools/rup/>
- [3] Beck, Kent. "Extreme programming (XP): a gentle introduction". Extraído el 3 de Octubre del 2011, de <http://www.extremeprogramming.org/>
- [4] Scrum group. "Scrum". Extraído el 2 de Octubre del 2011, de <http://www.scrum.org/>
- [5] Real Academia Española. "Metodología". Extraído el 8 de Octubre del 2011, de <http://www.rae.es/rae.html>
- [6] Yourdon, Edward. "Techniques of Program Structure and Design". Prentice Hall. 1976
- [7] Rumbaugh et al. "Object-Oriented Modeling and Design". Prentice-Hall. 1990
- [8] Ministerio de Presidencia de España. "Métrica v.3". Extraído el 8 de Octubre del 2011, de [http://administracionelectronica.gob.es/?\\_nfpb=true&\\_pageLabel=P60085901274201580632&langPae=es](http://administracionelectronica.gob.es/?_nfpb=true&_pageLabel=P60085901274201580632&langPae=es)
- [9] Sommerville, Ian. "Software Engineering" (7th ed). Prentice Hall. 2009
- [10] Braude, Eric. "Ingeniería de Software. Una perspectiva orientada a objetos". Alfaomega. 2000
- [11] Cockburn, Alistar. "Crystal Clear: a human-powered methodology for small teams". Addison-Wesley Professional. 2004
- [12] Jacobson, Ivar. "The Essential Unified Process (EssUP)". Extraído el 5 de Septiembre del 2011, de [http://www.ivarjacobson.com/process\\_improvement\\_technology/essential\\_unified\\_process\\_software/](http://www.ivarjacobson.com/process_improvement_technology/essential_unified_process_software/)
- [13] Jacobson, Ivar. "The Unified Process Lifecycle Practice". Extraído el 2 de Septiembre del 2011, de [http://www.ivarjacobson.com/Unified\\_Process\\_Lifecycle.aspx](http://www.ivarjacobson.com/Unified_Process_Lifecycle.aspx)
- [14] Object Management Group. "UML". Extraído el 14 de Septiembre del 2011, de <http://www.uml.org>
- [15] INEGI. "Estadística". Extraído el 14 de Febrero del 2011, de <http://www.inegi.org.mx>
- [16] Walpole, Ronald. "Probabilidad y Estadística para Ingenieros" (6th ed). Prentice Hall. 2000
- [17] Ramirez, Willy. "E-commerce & E-Business". Extraído el 24 de Septiembre del 2011, de <http://www.aiu.edu/publications/student/spanish/E-COMMERCE%20&%20EBUSINESS.htm>
- [18] Deming, Edwards. "The Deming System". Extraído el 14 de Septiembre del 2011, de <http://deming.org/>