



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

## “El uso de las Metodologías de desarrollo de software para sistemas informáticos”

**Autor:** Ivet Espinosa Conde, [email:ivet.conde@gmail.com](mailto:ivet.conde@gmail.com)

Universidad “Máximo Gómez Báez”, Cuba, Profesor Asistente e Investigador del Departamento de Recursos y Servicios Informáticos, Máster en Informática Aplicada.

### Resumen

El alto impacto que han alcanzado las Tecnologías de la Información y las Comunicaciones (TIC), han posibilitado el desarrollo de herramientas informáticas cada vez más potentes. En la actualidad los sistemas informáticos se han convertido en poderosas herramientas que posibilitan a las empresas y entidades en sentido general, automatizar sus procesos y contar con medios necesarios para el desarrollo eficiente y exitoso de sus procesos. Estas herramientas deben seguir un proceso de desarrollo guiado por una metodología que permita obtener software con la calidad y agilidad requerida. Las metodologías de desarrollo de software constituyen la guía para este proceso, estableciendo los principios robustos de la ingeniería a fin de obtener de manera económica, software que sea fiable y que funcione eficientemente.

**Palabras claves:** Metodología de desarrollo de software, sistema informático, herramienta, proceso.

### Abstract

The high impact they have reached the Information Technology and Communications (ICT) have enabled the development of increasingly powerful computing tools. Today, the computer systems have become powerful tools that enable companies and organizations in general, automate their processes and



## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

have the means necessary for the efficient and successful development of their processes. These tools must follow a development process guided by a methodology to obtain software with quality and agility required. The software development methodologies provide guidance for this process, establishing robust engineering principles in order to obtain economically software that is reliable and works efficiently.

**Keywords:** Software development methodology, computer system, tool, process.

### Introducción

El software es actualmente un producto sumamente importante, que mide incluso las potencialidades de desarrollo de un país. Este producto se desarrolla, no se fabrica en el sentido clásico: el software no es un producto típico industrial, cualquier otro producto, tiene una etapa de diseño y de desarrollo de prototipos, hasta que se lleva a una cadena de producción que incluye los cálculos de costo y de calidad. El software por el contrario es un producto único, que tiene los mayores costos en su proceso de ingeniería, no en la fabricación.

Por otra parte cuando se hable de ingeniería del software se refiere a la aplicación de un enfoque sistémico, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software. Por tanto en el desarrollo de este producto es importante contar con las herramientas, los métodos y los procesos correctos.

La creación de un sistema automatizado es generalmente un proceso largo y complejo que requiere un gran volumen de trabajo por parte de especialistas de diferentes perfiles y además está determinado por la realización de un complejo de tareas entre las cuales existe una relación lógica.

La naturaleza misma de un sistema automatizado como un sistema dinámico y complejo determina que su creación sea un proceso que se realiza por etapas de



## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

aproximaciones sucesivas con un nivel dado de estabilidad, confiabilidad y perfeccionamiento de la dirección.

Uno de los enfoques o principios más generalizado para facilitar la realización, planificación y control de un proyecto informático lo constituye la organización del trabajo en las llamadas etapas y fases. Estas etapas y fases varían de una metodología a otra en cuanto a cantidad y contenido.

El **objetivo** del trabajo es contribuir al aprendizaje de los diferentes tipos de metodologías que existen para el desarrollo de sistemas informáticos.

### Desarrollo

**Materiales y Métodos:** Se utilizaron como métodos y técnicas: el análisis histórico- lógico, el inductivo-deductivo, análisis documental, enfoque de sistema, y la observación.

### Resultados:

El concepto de metodología, dentro de la Ingeniería del Software es, sin duda, uno de los más oscuros y que más confusión produce tanto en estudiantes como en profesionales involucrados en procesos de desarrollo de software. Tanto es así, que en muchos proyectos de desarrollo, la aplicación de una metodología brilla por su ausencia, siendo éste un concepto casi desconocido.

Además, la constante innovación tecnológica hace que cada vez sea necesaria la aplicación de nuevas metodologías adaptadas a los nuevos tiempos. No es que exista una metodología claramente superior a las demás. Todas las metodologías son, en esencia, bienintencionadas. Obviamente, las más modernas responden a problemas y necesidades más actuales.



## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

### 1.1 Tipos de metodología

La comparación y/o clasificación de metodologías no es una tarea sencilla debido a la diversidad de propuestas y diferencias en el grado de detalle, información disponible y alcance de cada una de ellas. A grandes rasgos, si se toma como criterio las notaciones utilizadas para especificar artefactos producidos en actividades de análisis y diseño, se pueden clasificar en dos grupos: Metodologías Estructuradas y Metodologías Orientadas a Objetos. Por otra parte, considerando su filosofía de desarrollo, aquellas metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales (o también denominadas Metodologías Pesadas, o Peso Pesado). Otras metodologías, denominadas Metodologías Ágiles, están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollo pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.

- **Metodologías Estructuradas**

Los métodos estructurados comenzaron a desarrollarse a fines de los 70's con la Programación Estructurada, luego a mediados de los 70's aparecieron técnicas para el Diseño (por ejemplo: el diagrama de Estructura) primero y posteriormente para el Análisis (por ejemplo: Diagramas de Flujo de Datos). Estas metodologías son particularmente apropiadas en proyectos que utilizan para la implementación lenguajes de 3ra y 4ta generación.

*Ejemplos de metodologías estructuradas de ámbito gubernamental: MERISE (Francia), MÉTRICA (España), SSADM (Reino Unido). Ejemplos de propuestas de métodos estructurados en el ámbito académico: Gane & Sarson, Ward & Mellor, Yourdon & DeMarco e Information Engineering.*



## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

- **Metodologías Orientadas a Objetos**

Su historia va unida a la evolución de los lenguajes de programación orientada a objeto, los más representativos: a fines de los 60's SIMULA, a fines de los 70's Smalltalk-80, la primera versión de C++ por Bjarne Stroustrup en 1981 y actualmente Java o C# de Microsoft. A fines de los 80's comenzaron a consolidarse algunos métodos Orientadas a Objeto.

En 1995 Booch y Rumbaugh proponen el Método Unificado con la ambiciosa idea de conseguir una unificación de sus métodos y notaciones, que posteriormente se reorienta a un objetivo más modesto, para dar lugar al Unified Modeling Language (UML), la notación Orientada a Objetos más popular en la actualidad.

*Algunas metodologías orientadas a objetos que utilizan la notación UML son: Rational Unified Process (RUP), MÉTRICA (que también soporta la notación estructurada).*

### **1.2 Metodologías Pesadas**

Existen una serie de metodologías que se suelen llamar tradicionales, propuestas casi todas ellas con anterioridad a los años 90 del siglo XX, y que pretendían ayudar a los profesionales indicando pautas para realizar y documentar cada una de las tareas del desarrollo del software.



## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

Son las más tradicionales, se centran en la planificación, y definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema. Este tipo de metodologías son, mas eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida.

Entre las principales metodologías tradicionales se tienen los ya tan conocidos RUP y MSF entre otros, que centran su atención en llevar una documentación exhaustiva de todo el proyecto y centran su atención en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto.

- **El Proceso Unificado de Desarrollo de Software o Rational Unified Process (RUP):** es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. En RUP se agrupan las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería (Modelamiento del negocio, Requerimientos, Análisis y diseño, Implementación, Prueba, Instalación) y los tres últimos como de apoyo (Administración del proyecto, Administración de configuración y cambios, Ambiente). Define las fases de Conceptualización (Concepción o Inicio), Elaboración, Construcción y Transición. Se caracteriza por ser: *Dirigido por casos de uso, Centrado en la arquitectura, Iterativo e Incremental.*

### 1.3 Metodologías Ágiles

En una reunión celebrada en febrero de 2001 en Utah-EEUU, nace el término "ágil" aplicado al desarrollo de software. En esta reunión participan un grupo de 17



## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Varias de las denominadas metodologías ágiles ya estaban siendo utilizadas con éxito en proyectos reales, pero les faltaba una mayor difusión y reconocimiento.

Tras esta reunión se creó *The Agile Alliance*, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía "ágil". El Manifiesto comienza enumerando los principales valores del desarrollo ágil. (principios y valores). Los valores inspiran los doce principios del manifiesto. Estos principios son las características que diferencian un proceso ágil de uno tradicional.

Muchas metodologías similares a las ágiles fueron creadas antes del 2000. Entre los más notables se encuentran: Scrum (1986), Crystal Clear (cristal transparente), programación extrema (en inglés eXtreme Programming o XP, 1996), desarrollo de software adaptativo, Método de desarrollo de sistemas dinámicos (en inglés Dynamic Systems Development Method o DSDM, 1995).

- **Programación Extrema o eXtreme Programing (XP):** Es una disciplina de desarrollo de software basada en los métodos ágiles, que evidencia principios tales como el desarrollo incremental, la participación activa del cliente, el interés en las personas y no en los procesos como elemento principal, y aceptar el cambio y la simplicidad (Beck et al., 2001). XP



## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

apuesta por un crecimiento lento del costo del cambio y con un comportamiento asintótico. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas.

### 1.4 Comparación de metodologías ágiles y tradicionales

<b>Metodologías Ágiles</b>	<b>Metodologías Tradicionales</b>
Están orientadas hacia las necesidades del cliente.	Están orientados hacia el proceso del software.
Basadas en heurísticas o estadísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios.
Proceso menos controlado, con pocas políticas para el desarrollo.	Procesos mucha más controlados, con numerosas políticas o normas.
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos	Más artefactos
Pocos roles	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

### 1.5 Metodologías de desarrollo Web





## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

El avance de Internet y las comunicaciones ha provocado en los últimos años el nacimiento de nuevas propuestas metodológicas para la web. La mayoría de ellas han centrado su trabajo principalmente en las etapas de diseño e implementación. Entre las metodologías o métodos para la Web podemos citar:

- **WSDM: Web Site Design Method. 1997:** Define el sistema en base a los grupos de usuario, Su proceso de definición de requisitos tiene por objetivo el detectar los perfiles de usuario mediante dos tareas: Clasificación de usuarios mediante el estudio del entorno y Descripción de los grupos de usuario.
- **RNA: Relationship Navigational Analysis. 1998:** Resalta la necesidad de trabajar con la especificación de requisitos. Plantea una secuencia de pasos en la que separa el tratamiento de diferentes requisitos.
- **OOHDM: Object Oriented Hypermedia Design Model. 1999:** Es una propuesta ampliamente aceptada para la web. Ha sido usada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia y, sobre todo, numerosos sitios web.
- **NDT: Navigational Development Techniques. 2004:** Es un proceso metodológico para especificar, analizar y diseñar sistemas web. NDT es un proceso Bottom-Up, el cual se centra en la ingeniería de requerimientos, la cual es guiada por los objetivos, contemplando la captura, definición y verificación de requerimientos. El proceso inicia con la definición de los objetivos, luego se describe la captura y definición de los requerimientos, los cuales son clasificados y tratados de acuerdo a diferentes criterios. Una vez definidos y validados los requerimientos, se generan los modelos.

### 1.6Aplicaciones Ricas de Internet (RIAs)



## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

En la última década las aplicaciones Web han experimentado importantes cambios en cuanto a su arquitectura, las funcionalidades que ofrecen y las interfaces de usuario que presentan. Esta nueva forma de entender las aplicaciones Web recibe el nombre de RIA (Rich Internet Application), y queda patente en la denominada Web 2.0.

Ante estas nuevas necesidades, las metodologías dirigidas por modelos para el desarrollo de aplicaciones Web se han visto obligadas a adaptar sus métodos para generar aplicaciones Web que ofrezcan estas nuevas prestaciones. (por ejemplo UWE, OOHDM).

Los principales aspectos que aportan las RIAs son por ejemplo interfaces más atractivas (ricas) y una mejor capacidad de respuesta e interactividad con el usuario, similar a la de aplicaciones de escritorio. Entre estas metodologías se encuentran:

- **OOH4RIA:** Propuesta para el modelado de RIAs que extiende el método OO-H [11]. Se centra en el aspecto de la Interfaz de Usuario (IU) proponiendo 2 nuevos modelos: el modelo de presentación y el de orquestación [6]. El modelo de presentación organiza y da estructura a la IU. el modelo de orquestación modela el comportamiento existente entre los widgets que componen la IU y el comportamiento entre éstos y el sistema.
- **WebML:** Es un método dirigido por modelos que también se ha adaptado al desarrollo de RIAs. Se centra en el aspecto que permite al cliente almacenar datos y ejecutar parte de la lógica de negocio [1] [4]. El enfoque de la metodología WebML combina componentes tradicionales bien conocidos por los desarrolladores, como el diseño conceptual de datos usando el modelo Entidad- Relación y la especificación de los casos de



## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

usos usando UML, con nuevos conceptos y métodos para el diseño de hipertextos, que son fundamentales para el desarrollo web.

- **UWE:** método dirigido por modelos para la construcción de aplicaciones Web. Su propuesta se enfoca en el comportamiento de la IU. UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. entre los diagramas: diagramas de estado, secuencia, colaboración y actividad.

### Conclusiones

- Las metodologías de desarrollo de software guían el proceso para obtener un software de calidad y pueden ser clasificadas atendiendo a diferentes parámetros.
- La elección de una metodología varía en correspondencia con las características del sistema a implementar.
- No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc. Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes. Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos que tienen estas características. Por otra parte, la aparición de las Aplicaciones Ricas de Internet (RIAs) ha llevado a que la mayoría de las metodologías enfoquen sus esfuerzos en adaptar el modelado de la Interfaz de Usuario al nuevo paradigma.



## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

### Bibliografía

1. Jacobson, I., Booch, G. y Rumbaugh, J. (2000). El proceso unificado de desarrollo de software. Addison-Wesley.
2. Matheron, J. Merise. (1990). Metodología de Desarrollo de Sistemas. Teoría aplicada. Paraninfo.
3. Schwaber, K., Beedle, M., y Martin, R. C. (2001). Agile Software Development with SCRUM. Prentice Hall
4. Stapleton, J. (1997). DSDM: Dynamic Systems Development Method: The Method in Practice. Addison-Wesley
5. Ingeniería del Software: Metodologías y ciclos de vida. Laboratorio Nacional de Calidad del Software. 2009. [https://www.incibe.es/file/N85W1ZWfHifRgUc\\_oY8\\_Xg](https://www.incibe.es/file/N85W1ZWfHifRgUc_oY8_Xg)
6. Proceso de desarrollo de software. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. [http://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjmysHXwKTKAhWGpB4KHYYijBlkQFggcMAA&url=http%3A%2F%2Fwww.dsic.upv.es%2Fasignaturas%2Ffacultad%2Flsi%2Fdoc%2FIntroduccionProcesoSW.doc&usg=AFQjCNGAyOzvXjChM6izslal\\_eLoQhtk2Q](http://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjmysHXwKTKAhWGpB4KHYYijBlkQFggcMAA&url=http%3A%2F%2Fwww.dsic.upv.es%2Fasignaturas%2Ffacultad%2Flsi%2Fdoc%2FIntroduccionProcesoSW.doc&usg=AFQjCNGAyOzvXjChM6izslal_eLoQhtk2Q)
7. Oiver Andrés Pérez A. Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP – SCRUM. junio de 2011. <http://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwj609i-xKTKAhUFmR4KHZJCCJoQFggcMAA&url=http%3A%2F%2Fbiblioteca.uniminuto.edu%2Ffojs%2Findex.php%2Finventum%2Farticle%2Fdownload%2F9%2F9&usg=AFQjCNHcdzgmE2iUDryY3FiHoFJG87KXMQ>
8. María José Escalona Cuaresma y José Mariano González Romano. 1. Metodología y Técnicas en Proyectos software para la Web. Universidad de Sevilla. 2007. <http://www.lsi.us.es/docencia/get.php?id=2086>
9. Patricio Letelier y M<sup>a</sup> Carmen Penadés. Departamento de Sistemas Informáticos y Computación (DSIC). Universidad Politécnica de Valencia



## “CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2016”

Multidisciplinario

21 y 22 de abril de 2016, Cortazar, Guanajuato, México

(UPV). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

10. José Alfonso Aguilar. Irene Garrigós. Sven Casteleyn. ose-Norberto Mazón. Una propuesta orientada a objetivos para el análisis de requisitos en RIAs. [http://rua.ua.es/dspace/bitstream/10045/25167/1/2011\\_Aguilar\\_etal\\_JISBD.pdf](http://rua.ua.es/dspace/bitstream/10045/25167/1/2011_Aguilar_etal_JISBD.pdf)