

GROUPWARE PARA EL DISEÑO DE DIAGRAMAS UML EN TIEMPO REAL DENTRO DE UN AMBIENTE WEB

Groupware to UML Diagrams Design in Real Time within a Web Environment

Enrique de Jesús Alcocer Polo

Ingeniero de Sistemas
Universidad del Magdalena

Correo electrónico:
enrique.alcocer@gmail.com

Artículo de investigación

Fecha de Recepción: 2014-05-29
Fecha de Aceptación: 2014-08-23

Resumen

El objetivo principal del siguiente artículo es explicar algunos conceptos y teorías que servirán de base al desarrollo de un aplicativo groupware que permita a equipos virtuales y no virtuales pertenecientes a un proyecto de software diseñar diagramas UML de manera colaborativa enmarcados en un ambiente Web y en tiempo real.

Palabras clave: aplicativo, groupware, tiempo real, trabajo colaborativo, UML, Web 2.0.

Abstract

The main objective of this article is to explain some of the concepts and theories that serve as a basis in the development of a groupware application enabling virtual and non-virtual teams belonging to a software project to design UML diagrams in a collaborative way framed in a Web environment and in real time.

Keywords: application, groupware, real-time, collaborative work, UML, Web 2.0.

INTRODUCCIÓN

Existe una tendencia de llevar todo a la web hasta el punto que llegara un día en donde las computadoras no serán útiles si no tienen una conexión a Internet, los sistemas operativos no necesitaran memoria de disco para funcionar, solo una conexión a la red, varias de las aplicaciones que funcionan en los escritorios de las computadoras poco a poco van migrando a la web (editores de texto, editores de dibujo, etc.), cambiando la forma de vida, el modo de pensar y la percepción del mundo.

Surge el groupware como una nueva forma de trabajar de manera remota y grupal, nacen los equipos virtuales que utilizan la internet para llevar a cabo el desarrollo que implica cualquier proyecto a través del trabajo colaborativo, potencializando el trabajo desde casa; resultando de gran ayuda en problemas de distancia y ubicación geográfica. Mozilla [1] está promocionando el uso de una web abierta y libre, buscando un equilibrio entre el lucro comercial y el beneficio de todos, donde el manejo de la información sea segura y se respete la identidad del usuario.

Una imagen vale más que mil palabras cuenta el dicho popular, un modelo es un conjunto de imágenes y texto que representan algo, UML (Unified Modeling Language, Lenguaje Unificado de Modelado) [2] como lenguaje de modelado nos ayuda a realizar modelos de un sistema para su fácil entendimiento; los modelos son valiosos por muchas razones, en gran medida representado por imágenes y muchas veces las imágenes pueden transmitir más información que una gran cantidad de texto, los modelos son valiosos porque es más barato, rápido y fácil cambiar modelos que cambiar código, en este proyecto nos enfocamos en crear modelos que representen software.

En la práctica lo importante del desarrollo del software es la escritura del código, los diagramas son solo imágenes bonitas, lo que realmente le interesa al usuario es que el software funcione y no la belleza de los dibujos ¿Entonces por qué usar UML? Algunas de las razones son las siguientes:

- Ayuda a que la curva de aprendizaje del paradigma Orientado a Objetos sea más fácil.
- Permite una comunicación más fluida entre con los usuarios que conocen el negocio.
- Existe una comprensión más amplia del panorama general del problema a resolver.

Buscando un enfoque a la ingeniería de software, aprovechando las herramientas web 2.0 para el trabajo colaborativo en un proyecto a través del diseño de diagramas UML, no busca solucionar todos los problemas que surgen y son intrínsecos en cualquier proyecto de software, pero si agilizar la documentación derivada de los diagramas permitiendo el trabajo en grupo sin importar la ubicación geográfica, en un ambiente virtual donde el conocimiento es compartido y valorado por los miembros del mismo.

1. TRABAJO COLABORATIVO

El trabajo colaborativo es aquel que se realiza entre dos o más personas como una cooperación voluntaria que unen sus esfuerzos para lograr un objetivo común, dentro de un ambiente de trabajo colaborativo todas las partes se benefician aunque no en la misma proporción por existir diferencias asimétricas entre los participantes [3].

Según [4] el aprendizaje colaborativo dentro de una organización se encuentra en la necesidad de la interacción, la colabora-

ción y el cooperativismo entre pares, se toman el que mejor se desempeña y este enseña y transmite su conocimiento a los demás, con lo cual se gana tiempo en la formación autónoma e individual de lo que puede hacer cada persona.

Los equipos virtuales toman cada vez más fuerza dentro de las organizaciones, trabajar virtualmente tiene numerosas ventajas no solo para las organizaciones también para los usuarios a nivel individual, se reducen costes temporales y espaciales, existe una mayor flexibilidad, la diversidad del equipo favorece la creatividad personal y la innovación empresarial, hay una agilidad en la toma de decisiones y se producen mejores resultados. [5].

La incorporación de herramientas informáticas, específicamente aplicaciones de la web 2.0, ayudan al fomento de un ambiente colaborativo, desarrollando habilidades de comprensión, construcción y discusión sin necesidad de reuniones físicas, evitando de este modo, desperdiciar tiempo y dinero en traslados; el uso de estas herramientas ha significado pasar de una comunicación unidireccional estática, a una conversación bidireccional, donde la colaboración es la base de la interacción [6]; pero no se trata sólo de ahorrar tiempo y/o dinero, se trata de establecer una nueva relación con compañeros a través de las nuevas tecnologías [7].

El apoyo síncrono para las reuniones electrónicas con la ayuda de herramientas informáticas y la aparición de la investigación en esta área es lo que se llega a conocer como trabajo cooperativo apoyado por ordenador (CSCW) [8]. Al trabajar en un ambiente compartido apoyado en CSCW, se puede compartir conocimiento, a través de opiniones, información, tareas, etc. pero para que esto se pueda dar, es necesario tomar en cuenta los roles de los participantes

y las reglas así como están estructuradas las interacciones [9].

El trabajo colaborativo virtual [10] es un equilibrio que se logra entre las características sociales de todo ser humano para conectar y generar confianza y la capacidad arquitectónica bajo los conceptos y metodologías que le permite diseñar organizaciones y procesos en red, el trabajo en equipos virtuales, mediante el uso de TIC en las organizaciones actuales necesita de una reflexión y nuevos enfoques sobre el modo de dirigir al capital social en general, y más concretamente a los equipos de trabajo [11].

Dentro de la colaboración se puede definir de manera tentativa tres capas diferentes de capacidades donde cualquier organización puede identificar fácilmente su propia cobertura [12]:

- La primera capa se denomina “**Comunicación**” donde se intercambian información y datos entre las partes interesadas.
- La segunda capa se denomina “**Coordinación**”, las partes interesadas sincronizan tareas y objetos relacionados, como el flujo de trabajo o agenda compartida.
- La tercera capa se denomina “**Cooperación**”, en un espacio de trabajo compartido los actores realizan un trabajo colectivo de colaboración.



Fig 1. Capas de Colaboración. Future Workplaces, towards the “Collaborative” Web [12]

Las capas de comunicación y coordinación mostradas cuentan hoy en día con un gran despliegue, mientras la cooperación se reduce a menudo a un espacio de trabajo compartido donde las personas que colaboran son subidores de archivo y muy poco se ve el intercambio de los mismos entre los actores de manera sistemática, sin embargo, intercambiar o incluso compartir archivos no proporciona un nivel satisfactorio de apoyo a la colaboración entre personas, para llegar a una visión y entendimiento común deben ser capaces de colaborar de una manera eficiente, para los investigadores de hoy esta capa de cooperación es más compleja de lo que era antes.

2. SOFTWARE LIBRE

El software libre es aquel que puede ser distribuido, modificado, copiado y usado; por lo tanto, debe venir con el código fuente para que se pueda dar por hecho las libertades que lo caracterizan. Dentro del software libre hay matices que es necesario tener en cuenta. Por ejemplo, el software de dominio público es aquel que no está protegido por el copyright, por consiguiente, podrían crearse versiones no libres del mismo, en cambio el software libre protegido con copyleft impide a los redistribuidores incluir algún tipo de restricción a las libertades propias del software así concebido, es decir, garantiza que las modificaciones seguirán siendo software libre.

Es conveniente no confundir el software libre con el software gratuito, este no cuesta nada, hecho que no lo convierte en software libre, porque no es una cuestión de precio, sino de libertad. Para comprender este concepto, debemos pensar en la designación de libre como en “libertad de expresión”.

Existen cuatro clases de libertad para los usuarios de software:

- Libertad 0. La libertad para ejecutar el programa sea cual sea nuestro propósito.
- Libertad 1. La libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades —el acceso al código fuente es condición indispensable para esto.
- Libertad 2. La libertad para redistribuir copias y ayudar así a tu vecino.
- Libertad 3. La libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad —el acceso al código fuente es condición indispensable para esto. Software libre es cualquier programa cuyos usuarios gocen de estas libertades.

La libertad para utilizar un programa significa que cualquier individuo u organización podrán ejecutarlo desde cualquier sistema informático, con cualquier fin y sin la obligación de comunicárselo subsiguientemente ni al desarrollador ni a ninguna entidad en concreto [13].

La libertad para redistribuir copias supone incluir las formas binarias o ejecutables del programa y el código fuente tanto de las versiones modificadas, como de las originales, ya que debemos tener la libertad para redistribuir tales formas si se encuentra el modo de hacerlo, pues las libertades para hacer cambios y para publicar las versiones mejoradas requieren de la accesibilidad de código fuente, por supuesto de manera libre, condición necesaria del software libre.

Cuando hablamos de software libre, debemos evitar utilizar expresiones como “regalar” o “gratis”, ya que se puede caer en el error de interpretarlo como una mera cuestión de precio y no de libertad.

A partir de los años ‘60 y ‘70 una comunidad de expertos en informática y sistemas

de redes, autodenominados hackers, acordaron utilizar cooperativamente sus destrezas para descubrir nuevas posibilidades tecnológicas. Este acuerdo hizo posible la evolución de las computadoras y el trabajo colectivo en pro de lo que ellos denominaron el software libre [14].

El proyecto software libre nace en 1983 con el GNU [15] de Richard Stallman, investigador del Laboratorio de Inteligencia Artificial del Massachusetts Institute of Technology. Inició este proyecto con la intención de crear un sistema operativo compatible con Unix, pero de código abierto, de forma tal que cualquier persona pudiese utilizarlo, modificarlo y adaptarlo a sus necesidades. En 1985, Stallman funda la Free Software Foundation (FSF) con la idea fundamental de promover y divulgar el software de código abierto. Su meta principal consiste en “promocionar la libertad de los usuarios de ordenadores, y defender los derechos de los usuarios de software libre [16]. La FSF publica en 1989 la primera versión de GNU General Public License [15], la cual está orientada a “garantizar la libertad de compartir y modificar todas las versiones de un programa, con el fin de asegurarse de que sigue siendo software libre para todos sus usuarios” [15].

3. WEB 2.0

La Web 2.0 es la evolución de la web en donde los usuarios dejan de ser pasivos para pasar a ser activos, usuarios que participan y contribuyen al contenido de la red formando parte de una sociedad que se informa, comunica y genera conocimiento. Es una etapa que ha definido nuevos proyectos en Internet y está preocupándose por brindar mejores soluciones para el usuario final. Muchos aseguran que se ha reinventado lo que era el Internet, otros hablan de burbujas e inversiones, pero la realidad es que la evolución natural del

medio realmente ha propuesto cosas más interesantes.

La Web 2.0 busca fomentar la interacción social utilizando nuevas herramientas que permite conectar a una infinidad de fuentes de información y a los propios usuarios compartiendo contenidos, actividades, comentarios, fotografías, videos, etc. [9] Se debe pensar a la Web 2.0 como una configuración de tres vértices; tecnología, comunidad y negocio.

Los mayores consumidores de contenidos web son los jóvenes de los países ricos gracias a su alfabetización digital, aunque se está notando un incremento dentro de los países emergentes, desafortunadamente constituyen audiencias volátiles, que no tienen respeto a los derechos de copyright, ni lealtades más allá de su propia comunidad. Con ese perfil de audiencias y en un entorno publicitario conservador, el modelo de negocio está aún por aparecer, aunque la lógica de la larga cola (the Long Tail) permite confiar en desarrollos exitosos, más allá de Google [14].

Según [14] citando a O'Reilly, existen siete principios constitutivos de las aplicaciones Web 2.0:

1. La World Wide Web como plataforma.
2. Aprovechar la inteligencia colectiva.
3. La gestión de la base de datos como competencia básica.
4. El fin del ciclo de las actualizaciones de versiones del software.
5. Modelos de programación ligera. Búsqueda de la simplicidad.
6. El software no limitado a un solo dispositivo.
7. Experiencias enriquecedoras del usuario.

Internet surgió en 1989, cuando Tim Berners-Lee y Robert Cailiau desarrollaron un sistema basado en hipertextos (HTTP + HTML = WEB) para facilitar el acceso a la información del CERN (Organización Europea para la Investigación Nuclear). Se organizaba mediante una red de enlaces o hipervínculos entre diversos documentos que podían visualizarse con la ayuda de una aplicación específica (el navegador). Nace la Web 1.0 la cual se concibió como una colección de documentos estáticos enlazados para su consulta o descarga, donde los usuarios no podían interactuar con ellos, se ofrecía un recurso de manera unidireccional.

A medida que el sistema se hizo popular, aumentaron sus prestaciones. De manera progresiva, se desarrollaron métodos que permitiesen la creación de páginas con contenidos dinámicos, que dio lugar a la Web 1.5 en 1997 (más de uno recordara las páginas con efecto nieve y efectos en los cursores).

Aparecieron las aplicaciones escritas algún lenguaje de programación conocidos como los CGI que se ejecutaban en el servidor, y que podían recibir parámetros de los clientes. Pese a su gran utilidad, tendían a sobrecargar el servidor, de ahí que surgiesen sistemas de ejecución de módulos más integrados en el servidor y lenguajes de programación interpretados que permitían ejecutarse en el cliente.

No tardaron en aparecer diversas arquitecturas y lenguajes de programación, tanto en la parte del cliente con lenguajes de script como Javascript o VBscript u objetos incrustados en la página como los applets realizados en Java. En la parte del servidor, con lenguajes como PHP, ASP o JSP, que permitían el desarrollo de páginas web totalmente dinámicas e interactivas con acceso a Sistemas Gestores de Bases de Datos (SGBD).

Todas estas tecnologías dieron paso en 2003 a las aplicaciones Web 2.0, donde los usuarios empiezan a colaborar entre ellos, el consumo pasa a poder ser push, es decir, con tecnologías RSS: al usuario ahora le llegan los nuevos contenidos sin tener que ir a buscarlos y la actualización del contenido pasa a ser bidireccional. Los recursos son más fáciles de producir y consumir. Accediendo a través de la red mediante un navegador u otras aplicaciones específicas. Estas generan de manera dinámica una serie de páginas con tecnología AJAX que interpretará y representará el cliente [17]. La muestra la evolución de la web desde sus inicios:

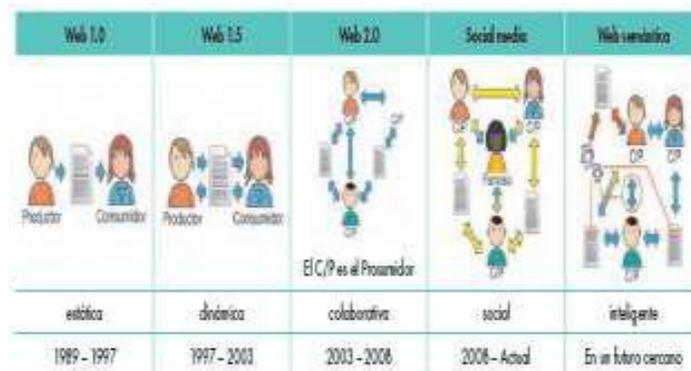


Fig. 2. Evolución de la Web. Aplicaciones Web [17]

En la etapa Web 2.0 con la aparición de tecnologías de acceso dedicado fijo (ADSL, Módem-cable) o móvil (UMTS/3G, LTE/4G) y la aparición de nuevas especificaciones estándares de W3C como XML, JSON, SOAP, UDDI, WSDL, RSS, ATOM o AMF en la parte nacen las aplicaciones interactivas de Internet enriquecidas (RIA por sus siglas en Internet de Rich Internet Application) las cuales son muchas más potentes y han desatado una nueva guerra de componentes con nuevas funciones multimedia (videos, gráficos vectoriales, animaciones).

Flash como plataforma de la compañía Adobe al sacar su marco de referencia cliente FLEX es una de las más activas, aunque han mantenido su hegemonía en los navegadores con Flash e incluso en los escritorios con AIR, se han visto aparecer tecnologías como JavaFX de Oracle o Silverlight de Microsoft, otras grandes compañías de Internet lideradas por Google, Amazon o Yahoo han venido trabajando con estándares como AJAX y han conseguido que en el 2012 la aprobación el nuevo estándar HTML5 del W3C con lo que se busca el dejar de usar complementos no estándares y así terminar con esta guerra por Internet [17].

4. UML

El lenguaje unificado de modelado o UML (Unified Modelillg Language) es el sucesor de la oleada de métodos de análisis y diseño orientados a objetos (OOA&D) que surgió a finales de la década de 1980 y principios de la siguiente. Entonces UML es un lenguaje de modelado, y no un método. La mayor parte de los métodos consisten, al menos en principio, en un lenguaje y en un proceso para modelar.

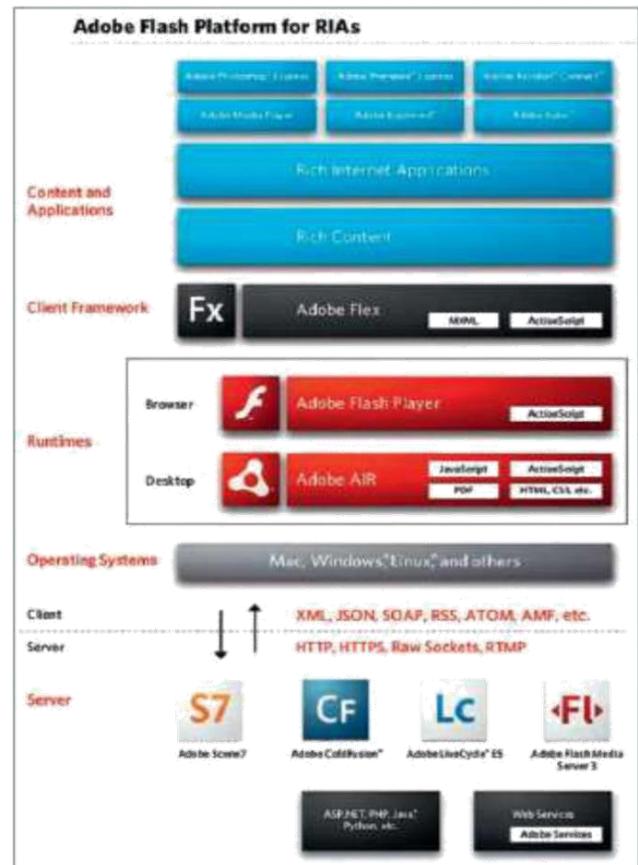


Fig. 3. Esquema de la plataforma Flash de Adobe para crear RIA. Planeta Web 2.0. Inteligencia colectiva o medios fast food [14]

El lenguaje de modelado es la notación (principalmente gráfica) de que se valen los métodos para expresar los diseños. El proceso es la orientación que nos dan sobre los pasos a seguir para hacer el diseño [18].

UML a través de sus diagramas permite una mejor comunicación entre las personas que conocen el negocio del sistema y las personas interesadas en la automatización del sistema, bien lo dijo [19]:

“UML (Lenguaje Unificado de Modelado) es una de las herramientas más emocionantes en el mundo actual del desarrollo de sistemas. Esto se debe a que permite a los

creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicárselas a otras personas”.

UML es un lenguaje precisamente como lo son el inglés o el afrikáans. UML a diferencia de otros lenguajes está conformado por símbolos y una gramática que define la manera en que se pueden usar estos símbolos. Si aprende los símbolos y la gramática, y sus imágenes serán comprensibles para todo aquel que reconozca estos símbolos y conozca la gramática [20].

Actualmente UML se encuentra en la versión 2.0 y está conformado por los siguientes diagramas:

Estructura

- Diagrama de clases
- Diagrama de objetos
- Diagrama de componentes
- Diagrama de estructura compuesta
- Diagrama de paquetes
- Diagrama de despliegue

Comportamiento

- Diagrama de casos de uso
- Diagrama de actividades
- Diagrama de estado interacción
- Diagrama de secuencia
- Diagrama de colaboración UML 1.X
- Diagrama de comunicación UML 2.0
- Diagrama de tiempo
- Diagrama de interacción

5. GROUPWARE

Son programas que ayudan a los individuos a trabajar juntos, usando una red local o Internet como medio de comunicación. Es el conjunto de tecnologías utilizadas en el CSCW (Trabajo en Grupo Asistido por Ordenador) y diseñadas al efecto. El trabajo realizado mediante herramientas groupware es una labor cualitativamente mejor que el planteado por los esquemas de organización tradicionales, ya que proporciona una intervención con los propios compañeros a tres niveles: Colaboración, coordinación y comunicación [21].

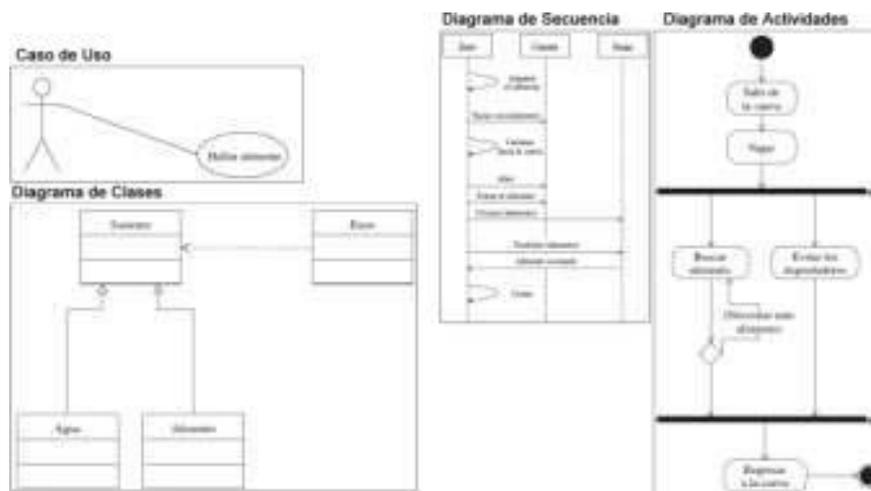


Fig. 4. Diagramas UML. Manual de UML [20]

También las podemos definir como:

- Cualquier producto o tecnología que permite el trabajo en equipo a grupos de personas.
- Orientadas a mejorar la productividad de grupos de trabajo o equipos colaborativos.
- Los programadores de software enfocan sus productos hacia algún aspecto relacionado con el groupware.
- Permite la replicación de información entre grupos de trabajo, ayudando a mantenerla sincronizada a lo largo de sitios dispersos geográficamente.

El término “groupware” fue introducido en 1978 por Peter y Trudy Johnson-Lenz. Definirlo no es sencillo, es un término que se refiere a la tecnología, a los sistemas que soportan la actividad de los grupos. Es un recurso en red que presenta una estructura que facilita las interacciones de un grupo ofreciendo herramientas que permiten manipular los materiales de interés común.

Básicamente en los sistemas groupware el desarrollo del trabajo está constituido por el intercambio de información, su gestión y control, la notificación de toda clase de actuaciones, utilización del conocimiento almacenado en espacios compartidos, entre otras acciones [22].

Existen diferentes tipos de grupos dependiendo de la forma de interactuar: Grupos electrónicos y grupos soportados electrónicamente. Los grupos electrónicos se podrían comunicar exclusivamente mediante ordenador; no conociendo personalmente un miembro a los otros miembros del grupo. Mientras que los grupos soportados electrónicamente lo hacen mediante el contacto personal además de las comunicaciones electrónicas. Groupware puede servir, tanto para encuentros de grupo cara a cara, como para grupos que no se encuentren en un mismo lugar geográfico:

Tabla 1. Matriz groupware espacio/tiempo

Matriz groupware espacio/tiempo		
Espacio/tiempo	Mismo Tiempo	Diferente Tiempo
Mismo Lugar	Interacción Cara a Cara	Interacción Asíncrona
Diferente Lugar	Interacción Síncrona Distribuida	Interacción Asíncrona Distribuida

Fuente: Un estudio sobre el groupware; propuesta de fundamentación teórica para el diseño de actividades colaborativas usando moodle [22]

Tabla 2. Matriz groupware 3x3

Matriz groupware 3x3			
Espacio/tiempo	Mismo Tiempo	Diferente Pero Predecible	Diferente e Impredecible
Mismo Lugar	Interacción Cara a Cara	Calendarios Electrónicos	Espacios de reuniones
Diferente Lugar predecible	Teleconferencias Videoconferencias Conferencias de escritorio	Interacción Correo electrónico	Editores de grupo
Diferente Lugar Impredecible	Seminarios Interactivos multiusuario	Tablón de anuncios	Manejo del flujo de trabajo

Fuente: Un estudio sobre el groupware; propuesta de fundamentación teórica para el diseño de actividades colaborativas usando moodle [22]

Los tres planos de intervención: comunicación, colaboración y coordinación. Cuando trabajamos con herramientas groupware, lo hacemos de manera distribuida, es decir, cada participante del grupo trabaja desde su propio ordenador, compartiendo información con el resto de miembros del grupo, pero manteniendo los tres planos de intervención referidos de las 3Cs del groupware: comunicación, colaboración y coordinación.

- **Comunicación:** Los miembros de un grupo de trabajo necesitan comunicarse los unos a los otros para intercambiar información.
- **Colaboración:** Mediante el intercambio de información, los miembros del grupo pueden colaborar realizando aportaciones, combinar opiniones y generar documentación de forma conjunta.
- **Coordinación:** Las tecnologías groupware permiten la coordinación de los distintos miembros del grupo para la realización de un determinado proyecto común.

Este tipo de herramientas pueden ser concebidas como un apoyo a un grupo que actúa cara a cara o a otro que trabaja físicamente distribuido. Pero también puede ser concebido como un sistema que ayuda a un grupo que interactúa en tiempo real o en forma asincrónica [21].

6. CICLO DE VIDA

El ciclo de vida del software muestra las fases por las cuales debe pasar todo proyecto de software, normalmente incluye las actividades de toma de requisitos, análisis, diseño, código, pruebas [23].

Existen diferentes modelos para ser aplicados a diferentes proyectos de software según su complejidad, tamaño, tiempo, etc.

A. Modelo en Cascada

El modelo en cascada es el más simple de todos ellos y sirve de base para el resto. En cada una de las fases se asignan unas actividades que al completarse proporcionarán los requisitos de la siguiente fase, entonces el proyecto no se diseña hasta que ha sido analizado, o se desarrolla hasta que ha sido diseñado, o se prueba hasta que ha sido desarrollado [24].

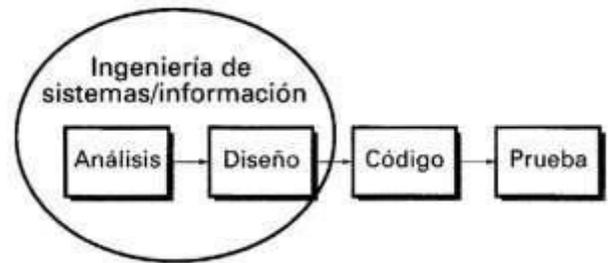


Fig. 5. Modelo en Cascada. Ingeniería del Software un Enfoque Práctico [24]

B. Modelo en Espiral

Conocido como modelo de proceso de software evolutivo combinando la construcción de prototipos con los aspectos controlados y sistemáticos del modelo línea secuencial o cascada, en este modelo se desarrolla en una serie de versiones incrementales del software e incluya la nueva variable que es el análisis del riesgo. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado [24].

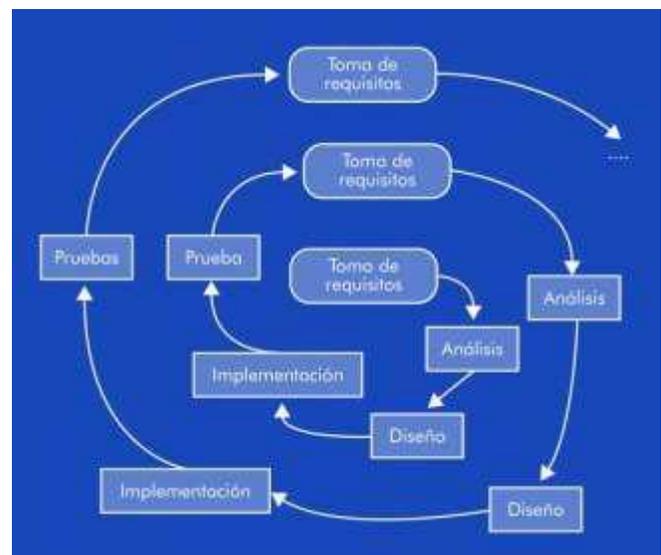


Fig. 6. Modelo en espiral. Ingeniería del Software en entornos de SL [24]

C. Modelo iterativo

El modelo iterativo o incremental combina elementos del modelo lineal secuencial con la filosofía interactiva de construcción de prototipos. El modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario [23].

El modelo iterativo o incremental es el más usado en el desarrollo de software libre (aunque sea involuntariamente) debido a la propia naturaleza cambiante de los requisitos o la aportación constante de nuevos colaboradores [24].

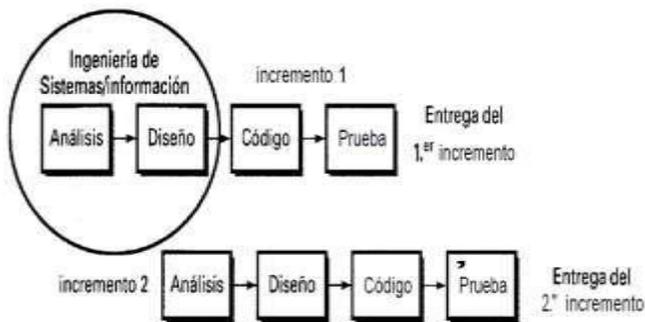


Fig. 7. Modelo iterativo. Ingeniería del Software: un enfoque práctico [23]

7. METODOLOGÍAS WEB

Con el nacimiento de aplicativos y completos sistemas de información desarrollados para funcionar en la nube, surge la necesidad de nuevos modelos y metodologías que puedan adaptarse a las aplicaciones web, entre ellas encontramos la OOHDM derivado de HDM.

A. HDM

El modelo HDM (Hypertext Design Mode), fue creado por Franca Garzotto, Paolo Paolini y Daniel Schwabe en 1991 [25]. El objetivo era crear un modelo que fuera de utilidad para realizar el diseño de una aplicación de hipertexto. HDM propone una serie de elementos que permiten

al diseñador especificar una aplicación. Estos elementos son las entidades, los componentes, las perspectivas, las unidades y los enlaces. Todos estos elementos pueden incorporarse en la semántica del clásico modelo Entidad-Relación. Pero, y a pesar de que términos como las entidades hayan sido heredados de los ERD han sido extendidos para poder representar una estructura compleja que contenga enlaces y una semántica de navegación interna.

Cada entidad está compuesta por una jerarquía de componentes que heredan las propiedades de dicha entidad. Los componentes no pueden existir sin la entidad de la que dependen. Los componentes son, por su parte, abstracciones para diseñar un conjunto de unidades o nodos que representan un mismo conjunto de información de la entidad. Una unidad, es pues un depósito de la información contenida en una aplicación.

Una unidad representa un fragmento del contenido de una entidad presentada bajo una perspectiva particular. De esta forma, la perspectiva permite representar la multiplicidad de presentaciones de un mismo contenido de información [26]. En la Fig. 8 se muestra una representación de un modelo HDM como un diagrama de clases utilizando el lenguaje de modelado UML.

1. La noción de que los objetos de navegación son vistas, en la base de datos sentido, de objetos conceptuales.
2. El uso de abstracciones apropiadas para organizar la navegación espacio, con la introducción de contextos de navegación.
3. La separación de los problemas de interfaz de problemas de navegación.
4. Una identificación explícita que hay decisiones de diseño que necesitan sólo hacerse en el momento de ejecución.

OOHDM propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cuatro etapas: diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación.

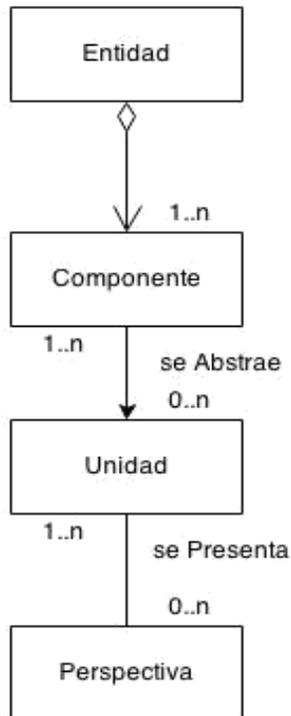


Fig. 8. Relaciones de los elementos de HDM. Metodologías para el desarrollo de sistemas de información global: análisis comparativo y propuesta [26].

A pesar de que en la actualidad HDM no se usa, ha servido como base a otras importantes metodologías como son RMM y OOHDM.

B. OOHDM

El modelo OOHDM (Object Oriented Hypermedia Design Methodology), creado para aplicaciones hipermedia, fue diseñado por Gustavo Rossi, Daniel Schwabe, Fernando Lyardet [27], [28]. Es una extensión de HDM pero está orientado a objetos, es una de las metodologías más utilizadas y ha sido usada para diseñar diferentes aplicaciones hipermedia como galerías interactivas, presentaciones multimedia y

numerosos sitios web. Las piedras angulares del enfoque OOHDM son:

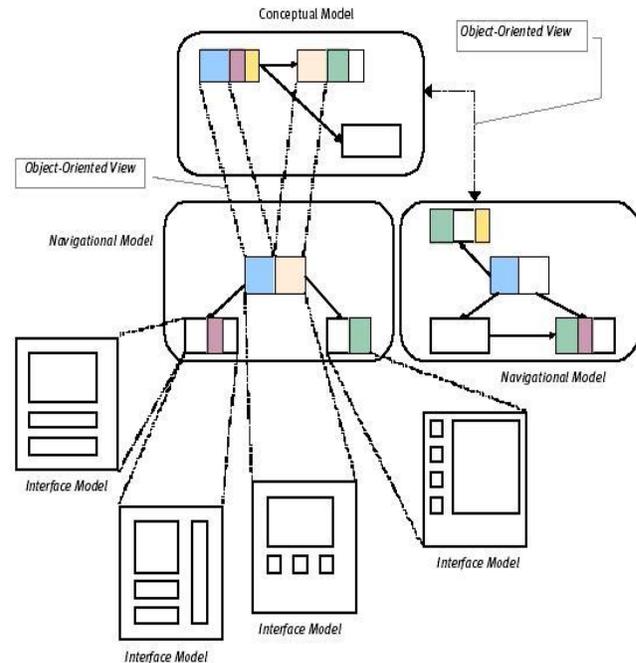


Fig. 9. Relación Conceptual entre la navegación y la interfaz de Objetos en OOHDM. (Las casillas sombreadas representan los atributos de clase). An Object Oriented Approach to Web-Based Application Design [27]

Modelado conceptual

Durante esta actividad, se construye un esquema conceptual que representa objetos, sus relaciones y colaboraciones existentes en el dominio de destino, está construido por clases, relaciones y subsistemas. Las clases son descritas como en los modelos orientados a objetos tradicionales. Sin embargo, los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real. En las aplicaciones hipermedia “convencionales”, es decir, aquellos en los que la componentes hipermedia no se modificarán durante su ejecución, que podría utilizar un modelo semántico estructural. Sin embargo, cuando sea la base de la información puede cambiar de forma dinámica o tenemos la intención de realizar

cálculos complejos o consultas sobre los objetos o el esquema, necesitamos un modelo orientado a objetos de comportamiento más rico.

Diseño Navegacional

La primera generación de aplicaciones web pretendía realizar la navegación a través de un espacio de información utilizando un solo modelo hipermedia de datos. La simplicidad de HTML como lenguaje de implementación y la falta de una cultura de diseño hipermedia resulta en los sitios web en los que los usuarios tienen una sobrecarga de experiencia cognitiva y desorientación mientras navegan por la Web.

Para empeorar las cosas, no hay buenas herramientas de orientación para los existentes navegadores y la mayoría de las herramientas existentes proporcionan un mapa de nivel “nodo y enlace”, que rápidamente se convierte en difícil de manejar. Además, estos mapas no son así definidos en el caso de páginas generados dinámicamente [27].

El diseño de navegación es expresado en dos esquemas: el esquema de clases navegacionales y el esquema de contextos navegacionales. En OOHDM existe un conjunto de tipos predefinidos de clases navegacionales: nodos, enlaces y estructuras de acceso.

La semántica de los nodos y los enlaces son las tradicionales de las aplicaciones hipermedia, y las estructuras de acceso, tales como índices o recorridos guiados, representan los posibles caminos de acceso a los nodos. La principal estructura primitiva del espacio navegacional es la noción de contexto navegacional.

Un contexto navegacional es un conjunto de nodos, enlaces, clases de contextos, y otros

contextos navegacionales (contextos anidados). Pueden ser definidos por comprensión o extensión, o por enumeración de sus miembros. Los contextos navegacionales juegan un rol similar a las colecciones y fueron inspirados sobre el concepto de contextos anidados [29].

Diseño de Interfaz Abstracta

Una vez que la estructura de navegación de la aplicación se ha definido, se debe especificar los aspectos de interfaz. Esto significa definir la forma en que aparecerán diferentes objetos de navegación, que se activarán los objetos de interfaz navegación y otras funcionalidades de la aplicación, y que transformaciones de la interfaz se llevarán a cabo y cuándo.

Una clara separación entre ambas preocupaciones, la navegación y el diseño abstracto de la interfaz, permite la construcción de interfaces diferentes para el mismo modelo de navegación, lo que lleva a un mayor grado de independencia frente a la tecnología de interfaz de usuario (User Interface). Además, esta separación permite una mejor comprensión de la estructura general de la aplicación, indicando claramente cuáles transformaciones en la interfaz son también las transformaciones de navegación. También indica que son simplemente transformaciones de interfaz locales que no afectan el estado de la navegación y por lo tanto no requieren el acceso a la Web servidor [27].

Implementación

En esta fase, el diseñador realmente implementa el diseño. Hasta ahora, todos los modelos se construyeron deliberadamente de una manera tal que sea independiente de la plataforma de implementación; en esta fase, el particular, entorno de ejecución se

tiene en cuenta. Diseños OOHDM se pueden implementar en la WWW, teniendo cuidado de no fijar un única alternativa, ya que hay muchos enfoques posibles a través del cual esto se puede lograr. Cuando se llega a la fase de implementación, el diseñador ya tiene definido los elementos de información que forman parte del dominio del problema. También ha identificado cómo estos elementos deben ser organizados de acuerdo con el perfil y tareas del usuario; él ha decidido cuál es la interfaz se verá así, y cómo se comportará [27].

CONCLUSIONES

Lo expuesto anteriormente servirá de base para la realización de un aplicativo groupware desarrollado para la web que permitirá el trabajo colaborativo de diagramas UML y sus características formales como artefacto tecnológico. La investigación para este proyecto no termina aquí a medida que se avance se estudiara y analizara nuevos conceptos, metodologías, modelos y tecnologías que se adapten a lo que se quiere hacer o bien se podrían descartar, es un camino que empieza en búsqueda constante de la innovación y mejora de los procesos propios de un proyecto de software y siempre enmarcado dentro de la Ingeniería de software.

REFERENCIAS

[1] Mozilla, “Mozilla”, 1998.
<http://www.mozilla.org/es-ES/>

[2] OMG, “UML”, 1997.
http://www.omg.org/gettingstarted/what_is_uml.htm

[3] L. Silva, Dorothy. “Tecnología y trabajo colaborativo en la sociedad del conocimiento”. Red Rev. Científicas América Lat. el Caribe, España y Port. Sist. Inf. Científica SILVA, vol. 23, pp. 107-122, 2013.

[4] R. Avello-Martínez, V. Gómez, “La construcción del conocimiento y las herramientas de trabajo colaborativo en red. Una aproximación teórica inicial,” Seminario Internacional Docencia Universitaria, 5, 2010, Cienfuegos (Cuba).

[5] V. Orengo, A. Zornoza, J. M. Peiró, “Gestión de equipos virtuales en las organizaciones laborales: algunas aportaciones desde la investigación”, Papeles del Psicólogo, vol. 32, pp. 82–93, 2011.

[6] G. Gmario, M. Marambio, “La organización del conocimiento en la web 2.0 en bibliotecas especializadas en Chile: su valor público”, Rev. F@ro, 2012, p. 8.

[7] F. C. y D. M. Juan Pablo Navarro, Hernán Sala, Vanina Limbiati, “Revista Electrónica de Didáctica en Educación Superior”, Buenos Aires, pp. 1-8, 2013.

[8] J. G. Davis, S. Konda, H. Granger, M. Collins, “Creating Shared Information Spaces to Support Collaborative Design Work,” vol. 1, no. 412, pp. 1-25, 2006.

[9] U. D. E. Les, I. Balears, M. G. Ezkauriatza, S. Ib., “Tesis”, 2011.

[10] C. F. Llambías, “El arte de facilitar el trabajo en Red”, TejeRedes. 2013, p. 58.

[11] V. Orengo, A. Zornoza, J. M. Peiró, “Gestión de equipos virtuales en las organizaciones laborales: algunas aportaciones desde la investigación” Papeles del Psicólogo, vol. 32, pp. 82-93, 2011.

[12] M. Pallot, W. Prinz, H. Schaffers, “Future Workplaces, towards the ‘Collaborative’ Web,” 2005.

[13] M. Culebro Juárez, W. G. Gómez Herrera, S. Torres Sánchez, “Software libre vs. Software propietario: ventajas y desventajas”, México: [s. f.], p. 170, 2006.

[14] C. Cobo, H. Pardo, Planeta Web 2.0. Inteligencia colectiva o medios fast food. Barcelona /México D.F., 2007, p. 162.

- [15] “GNU,” 2007.
<http://www.gnu.org/copyleft/gpl.html>
- [16] “Free Software Foundation (FSF)”, 2010. <http://www.fsf.org>
- [17] R. V. Lerma-Blasco, Aplicaciones web, McGraw Hill, España, 2013, p. 214.
- [18] M. Fowler, K. Seott, UML gota a gota. Pearson, Mexico D.F., 1999, p. 211.
- [19] J. Schemuller, Aprendiendo UML en 24 Horas, Prentice H. México D.F., 2001, p. 404.
- [20] P. Kimmel, Manual de UML, McGraw-Hill. México D.F., 2008, p. 257.
- [21] P. Román, “Posibilidades Formativas de las Herramientas Groupware. El aprendizaje colaborativo en la educación”, 2012.
http://tecnologiaedu.us.es/tecnoedu/images/stories/pedro_ro_man_groupware.pdf
- [22] N. R. A. G. y T. F. O. Pérez, “Un estudio sobre el groupware; propuesta de fundamentación teórica para el diseño de actividades colaborativas usando moodle,” Rev. Cub. Física vol. 24 No. 1, La Habana, pp. 55-58, 2007.
- [23] R. S. Pressman, Ingeniería del software. Un enfoque práctico, McGraw-Hill. España, 2002, p. 640.
- [24] D. A. Pérez, M. G. Ginestá, M. H. Matías, J. M. Hernández, Ingeniería del software en entornos de SL, UOC. Catalunya, 2007, p. 314.
- [25] F. Garzotto, L. I. Mainetti, P. Paolin, “Hypermedia Design, Analysis and Evaluation Issues,” 1995.
- [26] M. J. Escalona, “Metodologías para el desarrollo de sistemas de información global: análisis comparativo y propuesta,” Universidad de Sevilla, 2001.
- [27] D. Schwabe and G. Rossi, “An Object Oriented Approach to Web-Based Application Design,” 1998.
- [28] G. Rossi, D. Schwabe, and F. Lyardet, “Web Application Models are more than Conceptual Models,” LIFIA, Dep. Informática, UNLP. Argentina, 1999.
- [29] B. Mercerat and D. A. Silva, “Construyendo aplicaciones web con una metodología de diseño orientada a objetos,” vol. 115, 2002.