**Universidad de Oriente**

**Núcleo de Anzoátegui**

**Escuela de Ingeniería y Ciencias Aplicadas**

**Departamento de Computación y Sistemas**

**Área: Desarrollo de Software**

Integrantes:

Félix Rodríguez C.I: 20.170.112

Marcos Suniaga C.I: 18.278.702

Barcelona, Enero de 2013

***Grapple***

“Guía para la Ingeniería de Aplicaciones Rápidas” nace a raíz de estas metodologías antiguas e incoherentes y debido a Grady Booch, James Rumbaugh e Ivar Jacobson apodado como los “Tres Amigos” han creado el Proceso Rational Unificado (RUP), y ante ello cada persona o cada amigo tenía su propio proceso, las ideas en tales procesos son similares a GRAPPLE, de ahí que contiene varias de las mejoras prácticas que se aplican al buen desarrollo de aplicaciones rápidas, lo cual no es una firme metodología sino un conjunto de ideas adaptables y flexibles, permitiendo así da la oportunidad a un gerente de proyectos con creatividad, de agregar sus propias ideas respecto a lo que funcionará y extraer los pasos que no funcionen en una organización en particular, mediante el cual nos enseña el camino para mostrar al UML dentro de un buen contexto.

Grapple contiene 5 segmentos, cada segmento consta de diversas acciones, por consiguiente trae un producto de trabajo y cada acción es responsabilidad de un miembro del equipo. Los segmentos son:

\*Recopilación de necesidades.

\*Análisis.

\*Diseño.

\*Desarrollo.

\*Distribución.

Grapple se encausa a los sistemas orientados a objetos, por ello es que las acciones dentro de cada segmento se orientan a crear productos de trabajo de una naturaleza orientada a objetos.

***1-Recopilación de necesidades***

Si se intenta asignar una importancia relativa a cada segmento, este es un buen candidato para ser el número uno. Si no se comprende lo que desea el cliente nunca se podrá realizar el sistema adecuado. Todos los análisis de casos de uso en el proyecto no ayudarán si no se comprende las bases del dominio del cliente y el problema que se quiera resolver.

**\*Descubra los procesos de negocios**

Un analista entrevistará al cliente o una persona con el conocimiento necesario que sea designada por el, a quien se le preguntará los pasos importantes del proceso uno por uno. Como consecuencia el analista obtendrá un vocabulario de trabajo en un subconjunto de terminología, como resultado del producto de trabajo es un diagrama de actividades que captan los pasos y puntos decisivos en el proceso.

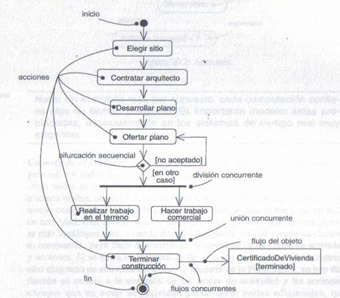
****

Diagrama de Actividades

**\*Realice un análisis del dominio**

El objetivo es comprender de la mejor manera los conceptos o terminologías recaudada en la fase anterior, adentrarse en el mundo del cliente en una entrevista y comprender las principales entidades en el dominio del cliente. Durante esta entrevista otro miembro del equipo tomará nota de forma óptima a través de un computador con las herramientas adecuada “modelador de objetos” el cual generará un diagrama de clases de alto nivel. Tomando como referencia los sustantivos y comenzar a convertir cada unos de ellos en una clase y algunos de ellos en atributos, los verbos se convertirán en las operaciones que realizaran cada clase.

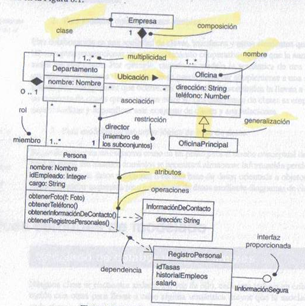
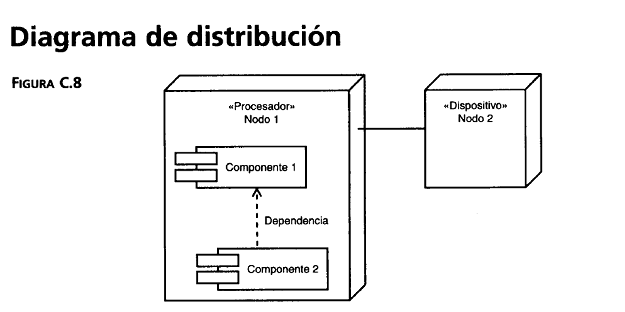
****

Diagrama de Clase

**\*Identifique los sistemas cooperativos**

Normalmente, los sistemas de negocios actuales no emergen de la nada, tienen que colaborar con otros. El equipo de desarrollo examinará y determinará de qué sistemas dependerá el nuevo sistema y cuales dependerán de él. Obteniendo como resultado un diagrama de distribución representando a los sistemas como nodos, con líneas de comunicación entre ellos, componentes residentes y dependencias entre componentes.

**\*Descubra las necesidades del sistema**

Descubrir las necesidades es muy importante, debido que con esta acción, el equipo realiza su primera sesión de JAD “Desarrollo Conjunto de Aplicaciones” y estas pueden conllevar a muchas otras durante este proceso. Una sesión JAD reúne a quienes toman las decisiones en la empresa del cliente, usuarios potenciales, miembros del equipo de desarrollo y un moderador cuya función es obtener una respuesta de quienes toman decisiones y de los usuarios acerca de lo que esperan que haga el sistema. Aparte deberán estar presente 2 miembros del equipo que tomen notas y el modelador de objetos deberá refinar el diagrama de clases que se obtuvo previamente. El producto del trabajo es un diagrama de paquetes. Cada paquete representa a un área de alto nivel de funcionalidad del sistema.

Cada paquete agrupa un conjunto de casos de uso. La complejidad del sistema será lo que determine la duración de la sesión, casi nunca será menor a media día laborar y podría durar hasta toda una semana laboral.

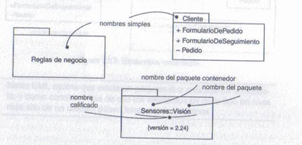
****

Diagrama de Paquetes

**\*Presentar los resultados al cliente**

Cuando el equipo finaliza todas las acciones de necesidades, el administrador de proyectos presentará los resultados al cliente. Algunas empresas podrían requerir la aprobación del cliente en este punto, para que pueda proceder el desarrollo. Otras podrían necesitar una estimación de los costos de acuerdo con los resultados. De esta manera el producto del trabajo podría variar de acuerdo con la empresa.

***2-Análisis***

En este segmento, el equipo profundiza en los resultados del segmento necesidades y aumentará su comprensión del problema. De hecho, partes de este segmento empezarán durante el segmento de necesidades, conforme el modelador de objetos empieza a depurar el diagrama de clases durante la sesión JAD (Desarrollo Conjunto de Aplicaciones) de Necesidades.

**\*Comprensión del uso del sistema**

Esta acción es un análisis de casos de uso de alto nivel. En una sesión JAD con usuarios potenciales, el equipo de desarrollo trabaja con los usuarios para descubrir a los actores que iniciarán cada caso de uso y quienes serán beneficiados. El equipo también intentara desarrollar nuevos casos de uso y casos de uso abstractos. El resultado será un conjunto de diagramas de casos de uso que muestren a los actores y las dependencias estereotipadas.

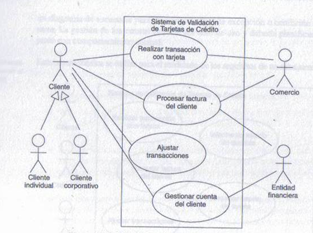
****

Diagrama de Casos de Uso

**\*Hacer realidad los casos de uso**

El equipo de desarrollo continúa su trabajo con los usuarios. El objetivo es analizar la secuencia de pasos en cada caso de uso. El producto del trabajo es una descripción textual de los pasos en cada caso de uso.

**\*Depurar los diagramas de clases**

En este punto, el modelador de objetos deberá rellenar los nombres de las asociaciones, clases abstractas, multiplicidades, generalizaciones y agregaciones. El producto del trabajo es un diagrama de clases depurado.

**\*Analizar cambios de estado en los objetos**

El modelador de objetos depurará el modelo mediante la presentación de cambios de estado conforme sea necesario. El producto del trabajo es un diagrama de estados.

****

Diagrama de Estados

**\*Definir la comunicación entre objetos**

Ahora que el equipo cuenta con un conjunto de diagramas de casos de uso y un diagrama depurado de clases, se definirá la forma en que los objetos se comunican. El modelador de objetos desarrollará un conjunto de diagramas de secuencias y de colaboraciones para delinear la comunicación. Deberán incluirse los cambios de estado. Estos diagramas conforman el producto del trabajo de esta acción.

**\*Analizar la integración con diagramas de colaboraciones**

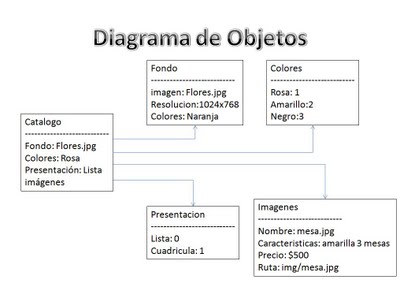
Al tiempo de realizar los pasos anteriores, el diseñador del sistema descubre los detalles específicos de la integración con los sistemas cooperativos. ¿Qué tipo de comunicación está envuelto?, ¿Cuál es la arquitectura de red?, si el sistema tendrá que utilizar bases de datos, un analista de bases de datos determinará la arquitectura (física o lógica) de ellas. Los productos del trabajo son diagramas de distribución detallados.

***3-Diseño***

En este segmento, el equipo trabajará con los resultados del segmento de Análisis para diseñar la solución. En el diseño y en el análisis se harán las revisiones pertinentes hasta que el diseño se haya completado. De hecho, algunas de las metodologías combinan al Análisis y al Diseño en una sola fase.

\* Desarrollo y depuración de los diagramas de objetos

Los programadores tomarán el diagrama de clases y generarán cualesquier diagramas de objetos que sea necesario. Darán vida a los diagramas de objetos mediante el análisis de cada operación y el desarrollo de un diagrama de actividades correspondiente. Los diagramas de actividades fungirán como la base de gran parte del código en el segmento de desarrollo. Los productos del trabajo serán los diagramas de objetos y los de actividades.



\* Desarrollo de diagramas de componentes

Los programadores serán quienes jueguen un importante papel en esta acción. La tarea era visualizar los componentes que resultarán del siguiente segmento y mostrar las dependencias entre ellos. Los diagramas de componentes serán el producto del trabajo.

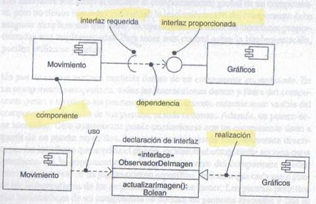


Diagrama de Componentes

\* Planeación para la distribución

Cuando se haya completado el diagrama de componentes, el diseñador del sistema empezará a planear la distribución e integración con sistemas cooperativos. Creará un diagrama de distribución que muestre el lugar donde se encontrarán los componentes. El producto del trabajo será un diagrama que sea parte de la distribución generado con anterioridad.

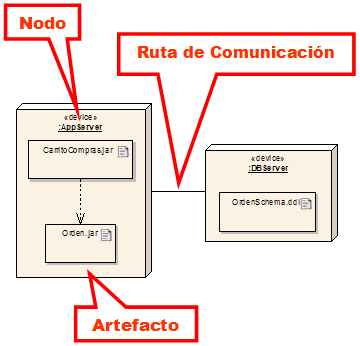
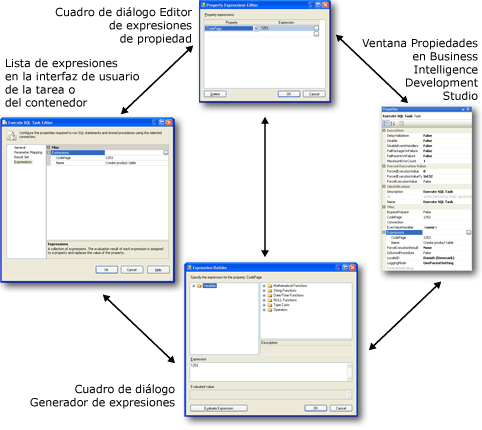


Diagrama de Distribución

\* Diseño y prototipos de la interfaz de usuario

Esto trae consigo otra sesión JAD con los usuarios. Aunque esto es parte del Diseño, esta sesión puede ser la continuación de anteriores sesiones JAD con los usuarios, un indicio de la interacción entre el Análisis y el Diseño. La interfaz del usuario debería permitir la consumación de todos los casos de uso. Para ello, un analista de GUI deberá trabajar con los usuarios para desarrollar prototipos, en papel, de las pantallas que corresponderán a grupos de casos de uso. Los usuarios pegarán papeletas removibles que representen los componentes de la pantalla (botones, casillas de verificación, listas de desplegables, menús, y cosas así). Cuando los usuarios queden satisfechos de la posición de los componentes, los desarrolladores generarán prototipos de las pantallas para que sean aprobados por los usuarios. Los productos del trabajo serán capturas de pantalla de los prototipos resultantes.



Prototipos de Interfaces

\*Pruebas de diseño

Los casos de uso permiten el diseño de pruebas del software. El objetivo es evaluar si el software hace lo que se supone que debería (esto es, que hace lo que se especifica en los casos de uso). Preferentemente, un desarrollador o especialista de pruebas externo al equipo de desarrollo deberá utilizar los diagramas de casos de uso para crear secuencias de comandos en herramientas automatizadas de pruebas. Tales secuencias de comandos conformarán el producto del trabajo.

\*Iniciar la documentación

Nunca es demasiado pronto para empezar a documentar el sistema para los usuarios finales y gerentes de sistemas. Los especialistas en la documentación trabajarán en conjunto con los diseñadores para empezar a generar un panfleto de la documentación y llegar a una estructura de alto nivel para cada documento. Tal estructura es el producto del trabajo.



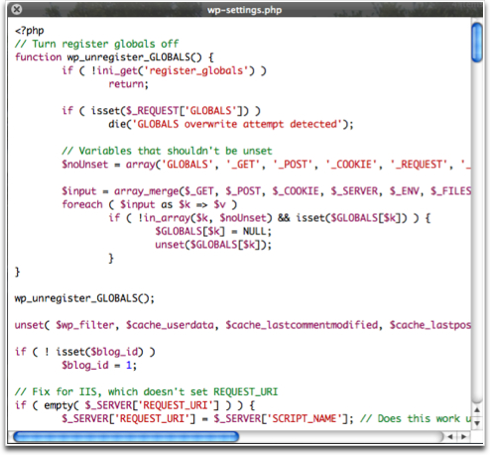
Documentación

***4-Desarrollo***

De este segmento se encargan los programadores. Con suficiente análisis y diseño, este segmento debería realizarse con rapidez y sin problemas.

\*Generación del código

Con los diagramas de clases, de objetos, de actividades y de componentes a la mano, los programadores generarán el código del sistema. Tal código es el producto del trabajo de esta acción.



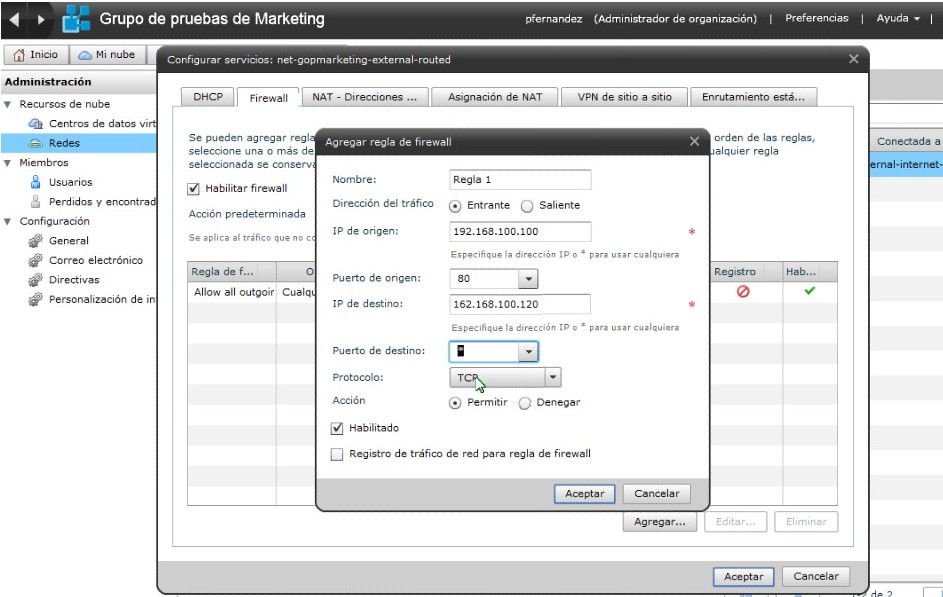
Código del Sistema

\*Verificación del código

Los especialistas en pruebas (no los desarrolladores) ejecutaran secuencias de comandos de prueba para evaluar si el código hace lo que se pretende. Los resultados de las pruebas son los productos del trabajo. Esta acción alimenta a la anterior y viceversa, hasta que el código pase todos los niveles de prueba.

\*Generación de interfaces del usuario, conexión con el código y prueba

Esta acción crea las interfaces de usuario ya aprobadas. El especialista en GUI las genera y conecta con el código. Las pruebas ulteriores aseguran que las interfaces funcionen adecuadamente. El sistema en funcionamiento junto con las interfaces de usuarios, son el producto del trabajo.



Interfaces Aprobadas

\*Consumación de la documentación

Durante el segmento de desarrollo, los especialistas en documentación trabajan en paralelos con los desarrolladores para asegurar la entrega oportuna de toda la documentación, la cual es el producto del trabajo de esta acción.



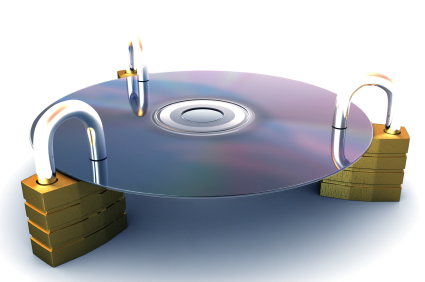
Consumación de la Documentación

***5-Distribución***

Cuando un sistema se ha finalizado, se distribuye en el hardware adecuado y se integra con los sistemas cooperativos. No obstante, la primera acción de este segmento puede iniciar antes de que el segmento de Desarrollo comience.

\*Planeación para copias de seguridad y recuperación

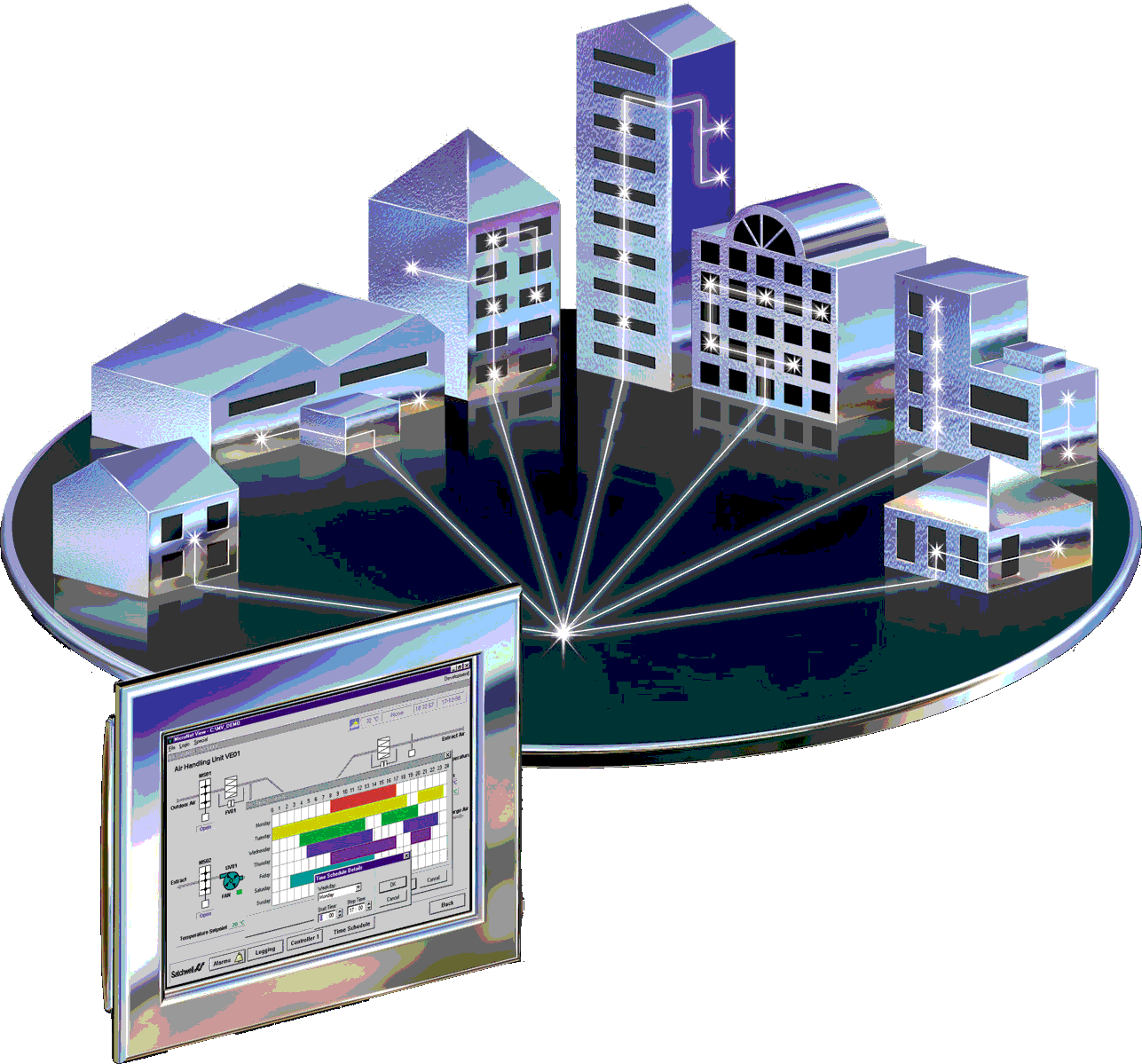
El diseñador del sistema creará un plan que incluya los pasos a seguir en caso de que el sistema falle. El plan, producto del trabajo de esta acción, establece lo que se deberá hacer para crear una copia de seguridad del sistema y para recuperarse del error.



Copias de Seguridad

\*Instalación del sistema terminado en el hardware adecuado

El diseñador del sistema, con toda la ayuda necesaria de los programadores, distribuye el sistema terminado en los equipos de cómputo adecuados. El producto del trabajo es el sistema completamente distribuido.



Sistema Distribuido

\*Verificación del sistema instalado

Finalmente, el equipo de desarrollo verifica el sistema instalado. ? Se ejecuta como se esperaba? El plan de copias de seguridad y recuperación funciona? Los resultados de estas pruebas determinarán si se necesita hacer una depuración ulterior. Tales resultados conforman el producto del trabajo de esta acción.

\*Celebración

Sin mayor explicación, el equipo de desarrollo podrá inventar los productos del trabajo de esta acción.

**Conclusión**

Una metodología de desarrollo estructura los segmentos y actividades en un proyecto de desarrollo de sistemas. Sin una metodología habría un caos y los desarrolladores no comprenderían el problema que se supone deberían resolver, así como los sistemas no cumplirían con las necesidades de los usuarios. Las metodologías de antaño forzaban a una secuencia “en cascada” de análisis, diseño, codificación y distribución. Este tipo de metodología secuencial podría fragmentar el desarrollo, de modo que un equipo de desarrollo podría no aprovechar la mejor asimilación que se obtiene durante la vida de un proyecto. Por lo general, también distribuye la mayor parte del tiempo en la codificación, y esto resta una enorme cantidad de tiempo al análisis y diseño. Es por esta razón que Grapple se basa en 5 segmentos, tratando de agilizar el proyecto, trabajando como se dice paso a paso, o segmento por segmento.

**Introducción**

La metodología de desarrollo de software GRAPPLE (Directivas para el Rápido Diseño de Aplicaciones), se refiere a un [framework](http://es.wikipedia.org/wiki/Framework) que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información.

A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad.

El framework para metodología de desarrollo de software consiste en:

* Una [filosofía de desarrollo de programas de computación](http://es.wikipedia.org/w/index.php?title=Filosof%C3%ADa_de_desarrollo_de_programas_de_computacion&action=edit&redlink=1) con el enfoque del proceso de desarrollo de software.
* Herramientas, modelos y métodos para asistir al proceso de desarrollo de software.

Estos frameworks son a menudo vinculados a algún tipo de organización, que además desarrolla, apoya el uso y promueve la metodología. La metodología es a menudo documentada en algún tipo de documentación formal.

La metodología GRAPPLE consta de cinco segmentos: Recopilación de Necesidades, Análisis, Diseño, Desarrollo y Distribución, los cuales le desarrollamos a continuación.