Universidad de Oriente.

Nucleó Anzoátegui.

Escuela de Ingeniería y Ciencias Aplicadas.

Departamento de Ingeniería.

Desarrollo de Software.



Proceso Unificado de Rational Fase Construcción.

Tutor Profesor Ing. Víctor Mujica

Grupo N° 6. Integrantes:

Correa, Luis. C.I: 19.840.230

Dun, Manuel. C.I: 19.257.821

Rondón, Frank. C.I: 19.738.854

Copyright © 2015 por Luis Correa & Frank Rondón & Manuel Dun. Todos los derechos reservados.

Barcelona, 22 de Julio de 2015

**Índice**

**Pág.**

**1.** **Introducción** 3

**2.** **Implementación** 4

2.1. Implementación de la arquitectura 4

2.2. Implementar una clase e implementar un subsistema 4

2.3. Realizar pruebas de unidad 4

2.4. Integrar el sistema 5

**3.** **Prueba** 7

3.1. Planificar pruebas 8

3.2. Diseñar las pruebas 8

3.3. Realizar pruebas de integración 8

3.4. Realizar pruebas del sistema 8

3.5. Evaluar las pruebas 9

**4.** **Despliegue** 9

**5.** **Control del análisis de negocio** 9

**6.** **Gestión de cambio de configuración** 10

**7.** **Evaluación de las iteraciones y la fase de construcción** 10

**8.** **Planificación de la fase de transición** 11

**9.** **Conclusiones** 12

**10.** **Bibliografía** 13

**11.** **Glosario de términos** 14

## **Introducción**

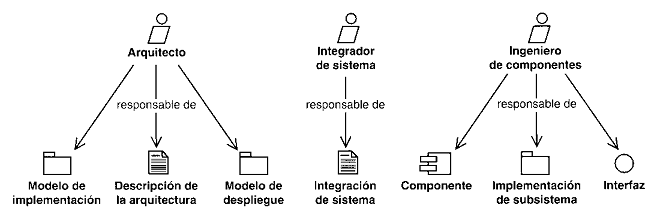
El Proceso Unificado de Rational (RUP) forma parte esencial en el desarrollo del software ya que te permite mediante sus faces lograr la creación de un software de calidad. En esta era de la información y con el surgimiento de nuevas tecnologías de comunicaciones, se cuenta con el RUP que es un sistema con un conjunto de metodologías adaptables al contexto y necesidades de cada organización. También se conoce por este nombre al software desarrollado por Rational (Empresa desarrolladora de software), que incluye información entrelazada de diversos [artefactos](https://es.wikipedia.org/wiki/Artefacto_%28dise%C3%B1o_de_software%29) y descripciones de las diversas actividades.

El RUP está incluido en el Rational Method Composer (RMC), que permite la personalización de acuerdo con las necesidades. En el presente caso, cada usuario podrá establecer los parámetros de uso del Software. En el presente trabajo se trata sobre la Fase de Construcción de RUP, en la cual están involucrados la implementación, prueba, control, gestión, evaluación y planificación del Software.

## **Implementación**

Este flujo de trabajo implementa y lleva a cabo las pruebas de unidad de todos los componentes, trabajando principalmente a partir del modelo de diseño. El resultado, después de varias iteraciones y de la integración y pruebas del sistema, es la versión operativa inicial, que representa el 100 por cien de los casos de uso. En esta subsección tratamos las actividades de implementación de la arquitectura, implementar una clase e implementar un subsistema, realizar pruebas de unidad e integrar el sistema.

Es en este flujo de trabajo en el que el proyecto lleva a cabo la mayor parte del trabajo de la fase de construcción, construyendo los componentes, como se describe en la Figura N°1. El proyecto rellena cada componente con más y más código, construcción tras construcción, iteración tras iteración, hasta que al final de la fase de construcción, todos los componentes están “llenos”.



**Figura N°1. Los trabajadores y artefactos involucrados en la implementación.**

### Implementación de la Arquitectura

En este momento la arquitectura está firmemente asentada. El papel del arquitecto, en lugar de realizar una vigilancia continua, será solo el de actualizarla si es necesario.

### Implementar una Clase e Implementar un Subsistema

Los ingenieros de componentes implementaran las clases y subsistemas del modelo de implementación. Implementaran clases stub cuando sea necesario para armar una construcción.

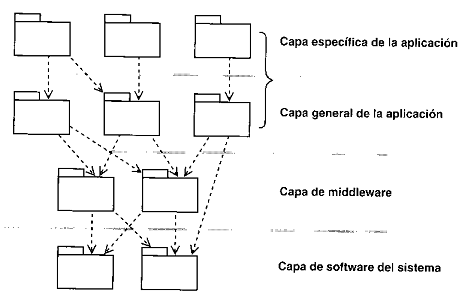
### Realizar Pruebas de Unidad

El ingeniero de componentes será el responsable de realizar las pruebas de unidad del componente que fabrique. Si es necesario, corregirá el diseño y la implementación del componente.

### Integrar el Sistema

El responsable de la integración del sistema creara un plan de integración de construcciones que perfile la secuencia de construcciones. Este plan mostrara los casos de uso o escenarios de un caso de uso que va a implementar la construcción. Estos casos de uso y escenarios nos conducirán a los subsistemas y componentes.

Los responsables de la integración de sistemas normalmente encuentran aconsejable, empezar por construir las capas inferiores de la jerarquía de la arquitectura en capas, como la capa del software del sistema o la capa intermedia (ilustradas en la Figura 2). Las construcciones siguientes se expandirán hacia arriba, hacia las capas general y especifica de la aplicación. Es difícil montar una construcción sin tener las funciones de apoyo que proporcionan las capas inferiores.



**Figura N°2. La arquitectura en capas organiza los sistemas en capas de subsistemas.**

Por ejemplo, la organización software de una determinada empresa química está pensada para realizar un incremento, en promedio, cada dos semanas. Se dice que los proyectos en Microsoft Corporation desarrollan una construcción cada día –una vez que el proyecto llega al punto de tener una construcción de código. Cada construcción es sometida a pruebas –pruebas de las nuevas adicciones y pruebas de regresión del código añadido con el ya existente—para asegurar que el código en desarrollo es capaz de funcionar. Este proceso diario de construcción comprueba cada día que las unidades de código comprobadas el día anterior son compatible. Esta práctica presiona a los desarrolladores para que no “rompan la construcción”. Al mismo tiempo sin embargo, reduce la presión a largo plazo sobre la organización software, ya que los problemas de integración son descubiertos durante las pruebas, normalmente cada noche, y resueltos poco después. El hacer una construcción cada día puede parecer que impone una considerable presión temporal sobre un proyecto, pero no necesariamente. Los desarrolladores comprobaran su código que no esté listo, posiblemente eso rompería la construcción. Sin embargo, los desarrolladores individuales estarán bajo la presión de comprobar su código a tiempo de cumplir con el plan de integración de construcciones.

La duración de cada periodo de construcción es ciertamente un asunto que cada organización debe considerar. El principio director es construir con una frecuencia suficiente como para obtener las ventajas que las construcciones frecuentes traen consigo.

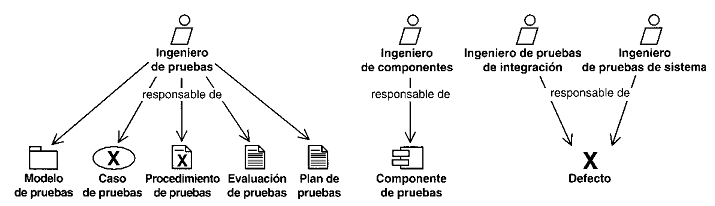
Para limitar el tamaño de cada construcción, a menudo el responsable de la integración del sistema encargara un STUB o un driver para representar un componente que no haya sido construido aun. Un STUB es un elemento muy sencillo que simplemente proporciona una respuesta a un estímulo –o a todos los estímulos que el componente puede revivir de otros componentes (aun no completados) de la construcción. De forma similar, un driver proporciona un estímulo a los otros componentes que son parte de la construcción que se está probando. Debido a que son sencillos, no es probable que STUBS y drivers introduzcan complicaciones adicionales.

Por tanto, el responsable de la integración del sistema, tendrá en cuenta el orden en el que ensamblar los componentes, con objeto de formar una configuración estable y que pueda ser probada. Documentara sus hallazgos en el plan de integración de construcciones, que mostrar en que momento es necesaria cada construcción para cumplir con el plan de integración y pruebas. El responsable de la integración del sistema reunirá clases completas y STUBS en una construcción ejecutable, de acuerdo con el plan de construcción, y hará construcciones cada vez mayores, mientras los encargados de las pruebas de integración y regresión las comprueban. En el paso final de cada iteración, y por último de la propia fase, el responsable de la integración del sistema conectara el sistema en su conjunto, y los responsables de probar el sistema y de realizar las pruebas de regresión lo comprobarán.

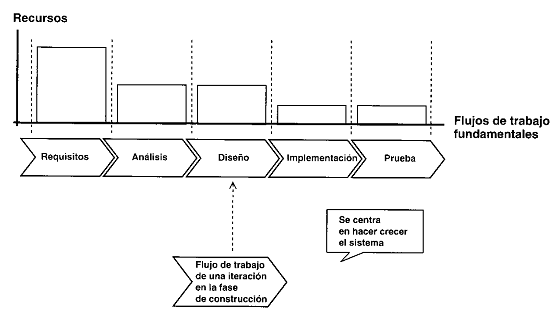
Esta planificación establecerá una secuencia que ordenara las construcciones de cada iteración y las iteraciones de la fase de construcción. Debido a que a menudo existen dependencias de compilación de las capas superiores de la arquitectura respecto a las capas inferiores, los responsables de la integración del sistema podrán tener que planificar la compilación de abajo a arriba.

## **Prueba**

Los esfuerzos de los ingenieros de pruebas para descubrir lo que puede ser comprobado de forma efectiva y para desarrollar casos de pruebas y procedimientos de pruebas para hacerlo, descritos en la Figura N°3, tendrán su fruto en la fase de construcción. Esta es una actividad fundamental de esta fase, como muestra la Figura N°4. Debido a que realizar pruebas de unidad es responsabilidad de los ingenieros de componentes, los ingenieros de pruebas deberán estar disponibles para proporcionar asistencia técnica. No obstante, los ingenieros de componentes y sus colaboradores, el encargado de las pruebas del sistema, son los responsables de la comprobación de las construcciones, es decir, los incrementos al final de las iteraciones, y en última instancia, de la construcción final, que construye la versión completa del sistema.



**Figura N°3. Los trabajadores y artefactos involucrados en las pruebas.**



**Figura N°4. El trabajo de una iteración durante la fase de construcción.**

### Planificar Pruebas

Los ingenieros de pruebas seleccionaran los objetivos que comprueben las sucesivas construcciones y, por último, el propio sistema.

### Diseñar las Pruebas

Los ingenieros de pruebas, determinaran como probar los requisitos en el conjunto de construcciones, con objeto de verificar los requisitos que puedan ser comprobados. Preparan casos y procedimientos de prueba con este propósito. De los casos y procedimientos de prueba de construcciones precedentes, seleccionaran aquellos que aun sean pertinentes y los modificaran para utilizarlos en las pruebas de regresión de las construcciones siguientes. Los ingenieros de pruebas verificaran los componentes que deberán ser comprobados de forma conjunta, tal y como es establecido originalmente en el plan de pruebas. El propósito de estas pruebas de integración es verificar las interfaces entre los componentes que estén siendo probados y comprobar que los componentes pueden trabajar conjuntamente.

### Realizar Pruebas de Integración

Los encargados de las pruebas de integración ejecutaran los casos de prueba, siguiendo los procedimientos de prueba. Si la construcción supera las pruebas, el responsable de la integración del sistema añadirá construcciones adicionales, según vayan estando disponibles, y el encargado de las pruebas de integración seguirá comprobándolas. Si las pruebas de integración detectan fallos, los encargados de las pruebas los registraran t notificaran al jefe de proyecto. El jefe de proyecto, o la persona que este designe, determinaran cual será el siguiente paso a dar. Esta persona puede ser el responsable de la integración del sistema, el cual posee el conocimiento técnico pertinente. El siguiente paso podría ser, por ejemplo, prolongar el trabajo en la misma construcción, diferir la corrección del defecto hasta la construcción siguiente o, en particular en el caso de un fallo especialmente grave, asignar una serie de personas específicamente cualificada para investigarlo.

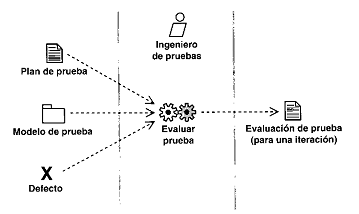
### Realizar Pruebas del Sistema

En el momento en que las sucesivas construcciones alcancen el final de una iteración, habrán alcanzado el status de versión parcial del sistema, y entraran en la jurisdicción del encargado de las pruebas del sistema los registrará y notificara al jefe de proyecto a la persona que este designe para su resolución.

Al final de la última iteración de la fase de construcción, el encargado de las pruebas del sistema comprobará la versión operativa inicial. De nuevo, notificara los fallos al jefe de proyecto para que encargue al ingeniero de componentes responsable de su corrección.

### Evaluar las Pruebas

A medida que transcurren las pruebas de integración y del sistema, los ingenieros de pruebas revisaran los resultados de las pruebas al final de cada construcción a la luz de los objetivos originalmente fijados en el plan de pruebas (posiblemente modificado durante las sucesivas iteraciones). El propósito de evaluar las pruebas es asegurarse de que estas alcancen sus objetivos. Si una prueba no alcanza sus objetivos, los casos y procedimientos de prueba deberán ser modificados para lograrlos ver Figura N°5



**Figura N°5. Entrada y resultado de la evaluación de prueba.**

## **Despliegue**

## **Control del análisis de negocio**

Uno de los propósitos de la apuesta de negocio, desarrollada al final de la fase de elaboración, es el de servir de guía al jefe de proyecto y los inversores para ejecutar la fase de construcción. Con este objetivo, el jefe de proyecto compara el progreso real al final de cada iteración con la agenda, esfuerzo y costes planificados. Revisará los datos de productividad del proyecto, cantidad de código desarrollado, tamaño de la base de datos y otras métricas.

Rara vez el número de líneas de código completadas es una buena medida del progreso en un desarrollo basado en componentes. Debido a que uno de sus objetivos es la reutilización de clases y componentes, un ingeniero de componentes u otros trabajadores pueden realizar un buen progreso con las construcciones e iteraciones y sin embargo escribir poco código nuevo. Una medida más pertinente del trabajo desarrolldado, en estas circunstancias, es la finalización de las construcciones e iteraciones de acuerdo con el plan.

Discrepancias mayores de un pequeño porcentaje especialmente en sentido negativo, llevaran al jefe de proyecto a tomar medidas. De forma similar, discrepancias de mayor tamaño llevaran a reuniones de revisión con los inversores.

A medida que el jefe de proyecto adquiere un mayor conocimiento sobre los costes y capacidades del producto durante la fase de construcción, puede encontrar necesario actualizar el análisis de negocio y comunicar el nuevo análisis a los inversores.

## **Gestión de cambio de configuración**

## **Evaluación de las iteraciones y la fase de construcción**

Sobre la base de una revisión de los resultados de las pruebas y otros criterios de evaluación, el jefe del proyecto y el grupo de evaluación:

* Material de usuario: En la fase de construcción se prepara una primera versión de los materiales escritos de ayuda a los usuarios finales, tales como guías de usuario, textos de ayuda, notas de versión, manuales de usuario, etc. Los criterios de evaluación. ¿Son suficientes para dar soporte a los usuarios en la fase de transición?
* Material de cursos: Una versión preliminar de los materiales para cursos que den soporte a los usuarios finales, tales como diapositivas, notas, ejemplos y tutoriales, se preparan también durante esta fase. ¿Son suficientes para dar soporte a los usuarios en la fase de transición?, los criterios de evaluación durante la fase de construcción, deberán tener presente si la capacidad operativa está suficientemente madura y estable para permitir la entrega de versiones beta o preliminares a los usuarios, sin que las organizaciones de software y las propias comunidades de usuarios sufran riesgos inaceptables o de difícil control.
* Revisaran lo logrado en una iteración contrastándolo con lo que había sido planificado.
* Planificaran en cuál de las iteraciones siguientes se habrá de llevar a cabo el trabajo no completado.
* Determinaran que la construcción está lista para entrar en la siguiente iteración.
* Actualizaran la lista de riesgos.
* Completaran el plan de la iteración siguiente.
* Al final de la última iteración de esta fase, determinaran que el producto ha superado las pruebas del sistema y que han alcanzado la capacidad operativa inicial.
* Autorizaran la entrada en la fase de transición.
* Actualizaran el plan de proyecto.

## **Planificación de la fase de transición**

El equipo de proyecto no puede esperar planificar de antemano la fase de transición con tanto detalle como hizo para las fases anteriores. Sus miembros saben, por supuesto, que van a producir versiones beta (o su equivalente) para que las evalúen usuarios seleccionados. Esta parte de la fase de transición: seleccionar a los encargados de probar las versiones beta, reproducir el código de operación, preparar instrucciones de pruebas, etc., es la que se puede planificar con cierto detalle.

La respuesta recibida –riesgos, problemas, fallos, sugerencias—difícilmente puede preverse de antemano. Si el equipo ha tenido experiencia en pruebas beta, podrá hacerse una idea de lo que cabe esperar. Sera capaz de estimar la cantidad aproximada de personal experimentado necesario para abordar los problemas que los usuarios de la beta descubran.

## **Conclusiones**

* Se concluye que RUP es una metodología de desarrollo de software de uso universal. Que está siendo muy utilizado por empresas importantes, ya que permite la personalización del software a la medida de cada usuario.
* El RUP es una metodología basada en Casos de Uso con el objetivo de identificar los actores involucrados y a partir de sus objetivos, encontrar los casos de uso.

## **Bibliografía**

* James Rumbaugh, Ivar Jacobson, Grady Booch 2000. El Proceso Unificado de Desarrollo de Software. Madrid. Addison Wesley.

## **Glosario de términos**

* STUB: Es un elemento muy sencillo que simplemente proporciona una respuesta a un estímulo –o a todos los estímulos que el componente puede revivir de otros componentes (aun no completados) de la construcción