

# Universidad de las Ciencias Informáticas

## Facultad 3



**Título:** Modelo de desarrollo de software para el departamento  
Soluciones Aduanales del CEIGE.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Autor:**

Idalberto Carlos Alonso Driggs

**Tutor:**

Ing. Yisel Niño Benítez

**Cotutor(es):**

Ing. Daileny Caridad Arias Pupo

MsC. Mariano Flores López

Ciudad de la Habana, 2012

Año del 53 aniversario de la Revolución

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

Idalberto Carlos Alonso Driggs

Autor

\_\_\_\_\_

Ing. Daileny Arias Pupo

Cotutora

## **DATOS DE CONTACTO**

Tutora: Ing. Yisel Niño Benítez.

Graduada de Ingeniera en Ciencias Informáticas en el año 2009. Pertenece al centro de desarrollo CEIGE. Posee tres años de experiencia. Ocupa el cargo de Administradora de calidad del centro.

Categoría docente: Instructor recién graduado.

Correo electrónico: ynino@uci.cu

Cotutora: Ing. Daileny Arias Pupo.

Graduada de Ingeniera en Ciencias Informáticas en el año 2011. Pertenece al centro de desarrollo CEIGE. Posee un año de experiencia. Ocupa el cargo de Asesora de calidad del centro.

Categoría docente: RGA.

Correo electrónico: dcarias@uci.cu

## AGRADECIMIENTOS

*Gracias a mi Señor y Salvador Jesucristo, por regalarme la vida y darme las fuerzas para llevar acabo todo este proyecto, unas vez más puedo decir hasta aquí me ha ayudado Dios. Gracias mami, papi, togi y abuela por su amor y apoyo en cada momento de mi vida. Gracias a mi novia Roxy por tu amor y cariño, te quiero mucho y ya falta menos. Gracias a mis hermanos por todas las cosas buenas que me han enseñado en todo este tiempo, los quiero mucho y Dios me los bendiga. Gracias a mis amigos, compañeros de aula, apartamento, proyecto y todos aquellos que han estado a mi lado y me han regalado parte de su tiempo. Gracias a la tutora y cotutora Dailenis por ser los dos a la vez, por soportarme y toda la ayuda que me has dado en todo este período, muchas gracias. Gracias a la oponente Giselle y al tribunal en general, por su ayuda y los consejos en cada una de los cortes que me han ayudado a perfeccionar el trabajo y llegar hasta el final.*

*Gracias a todos.*

## DEDICATORIA

*Dedico este trabajo a mi familia, mi mamá, papá, mi hermana y abuela Doris. A la chacha más hermosa de toda la Tierra, Roxi. A mis hermanos y a todos aquellos que de una manera u otra fueron parte y ayudaron en este proyecto y al mejor de todos a Dios por permitir hacerlo realidad.*

## RESUMEN

Mejorar la organización para erradicar las dificultades de planificación, control y toma de decisiones en el proceso de desarrollo de software en el departamento de Soluciones Aduanales del Centro de Informatización de la Gestión de Entidades (CEIGE), ha llevado a organizar los procesos que deben desarrollarse en el mismo. Con este propósito y luego de un estudio detallado de los modelos de desarrollo de software existentes, se realiza una propuesta de modelo que incluye las actividades, roles y los artefactos que deben llevarse a cabo en el departamento y en cada proyecto perteneciente al mismo. En la conformación del modelo se tienen en cuenta y se incluyó además, las actividades por áreas de proceso que propone Capability Maturity Model Integration (CMMI) nivel dos, ya que la universidad está orientada por un programa de mejora, para certificar dicho nivel y es de vital importancia cumplir con lo establecido en este modelo. Se describe con claridad cada fase, actividad y artefacto que conforma el modelo de desarrollo propuesto, y se especifican los roles que intervienen en cada actividad. Finalmente el modelo se validó utilizando el método Delphi como una de las variantes de Métodos Expertos, haciendo uso del criterio de un panel de expertos con un alto nivel de competencia y conocimiento en la materia, se obtuvo como resultado la aprobación del modelo apoyado por la concordancia de los diferentes criterios de los expertos.

**Palabras clave:** artefactos, actividades, mejoras, modelo de desarrollo de software, roles, proceso de desarrollo de software.

## Índice de contenidos

Introducción .....	1
Capítulo 1. Fundamentación teórica .....	5
1.1. Introducción.....	5
1.2. Conceptos importantes .....	5
1.3. Tipos de modelos de desarrollo de software.....	8
1.4. Mejora de proceso en el desarrollo de software .....	18
Capítulo 2. Propuesta de solución .....	22
2.1. Introducción.....	22
2.1.1. Información del departamento de Soluciones Aduanales del CEIGE .....	22
2.2. Modelo de desarrollo de software.....	25
2.3. Propuesta de avances para alcanzar el nivel tres de CMMI.....	50
2.4. Conclusiones parciales .....	51
Capítulo 3. Validación de la propuesta de solución .....	52
3.1. Introducción.....	52
3.2. Método de evaluación de experto.....	52
3.3. Conclusiones parciales .....	63
Conclusiones generales .....	64
Recomendaciones.....	65
Referencias bibliográficas .....	66
Bibliografías .....	67
Glosario de términos .....	71

### Índice de figuras

Figura 1: Proceso de desarrollo de software. ....	5
Figura 2: Elementos del proceso de desarrollo de software.....	6
Figura 3: Modelo de desarrollo en cascada. ....	10
Figura 4: Modelo de desarrollo evolutivo. ....	11
Figura 5: Modelo de desarrollo en Espiral.....	13
Figura 6: Paradigma de programación automática.....	15
Figura 7: Desarrollo basado en reutilización de componentes.....	16
Figura 8: Modelo de desarrollo iterativo incremental.....	17
Figura 9: Estructura organizativa del departamento Soluciones Aduanales.....	23
Figura 10: Estructura organizativa de un proyecto del departamento Soluciones Aduanales.....	24
Figura 11: Estructura general del modelo de desarrollo de software del departamento aduana.....	25
Figura 12: Ciclo de vida de los proyectos del departamento Soluciones Aduanales.....	29
Figura 13: Actividades de la fase Estudio Preliminar.....	31
Figura 14: Actividades de la fase modelamiento del negocio.....	34
Figura 15: Actividades de la fase de Requisitos.....	37
Figura 16: Actividades de la fase de Análisis y diseño.....	40
Figura 17: Actividades de la fase de Implementación.....	43
Figura 18: Actividades de la fase Pruebas internas.....	44
Figura 19: Actividades de la fase Pruebas de liberación.....	45
Figura 20: Actividades de la fase de Despliegue.....	47
Figura 21: Relación de las actividades horizontales en el ciclo de desarrollo.....	48
Figura 22: Porcentaje del coeficiente de competencia de los expertos.....	56



### Índice de tablas

Tabla 1: Coeficiente de conocimiento.....	54
Tabla 2: Coeficiente de argumentación.....	54
Tabla 3: Coeficiente de expertos.....	55
Tabla 4: Datos obtenidos de los expertos.....	57
Tabla 5: Tabla de frecuencias absolutas.....	59
Tabla 6: Tabla de frecuencias acumuladas.....	60
Tabla 7: Tabla de frecuencias relativas acumuladas.....	61
Tabla 8: Puntos de corte.....	62
Tabla 9: Grados de adecuación.....	62

### Introducción

El desarrollo de software es uno de los pilares fundamentales de la informática, y se ha convertido en un aspecto indispensable en los negocios de las empresas, la ciencia, la ingeniería y en todo el avance científico técnico de la humanidad. Las empresas desarrolladoras de software deben ser capaces de gestionar y controlar el complejo proceso de desarrollo y mantenimiento del software, ser conscientes de la importancia de hacer buenos desarrollos y lo complicado que es su realización; ha llevado al nacimiento de modelos, metodologías, teorías y normas a seguir para hacer más fácil, rápida y organizada la construcción y el mantenimiento del software y así aportar las mejores soluciones al mercado.

Cuba aún no muestra resultados sobresalientes en América Latina. En el país la industria de software, es relativamente nueva y su desarrollo se ha visto obstaculizado por las consecuencias de un bloqueo económico. Pero a pesar de los contratiempos externos, muchas cosas se han hecho, el país ha estado trabajando en determinar una estrategia general, que no solo abarque la informatización del país, sino que contemple toda la industria como un factor clave para el desarrollo de la nación. Se han creado en el país organizaciones e instituciones para concretar estas ideas, un ejemplo de esto es la Universidad de las Ciencias Informáticas (UCI), nacida en una de las etapas más importantes de la nación cubana en busca de nuevas vías de progreso. La UCI con el objetivo de formar profesionales integrales y potenciar el desarrollo de software con calidad, hoy en día constituye un eslabón importante para el desarrollo de la industria de software cubana.

Es una gran oportunidad y un reto para la universidad desarrollar las estrategias que le permitan un posicionamiento y un reconocimiento internacional con productos competitivos de exportación, lo que requiere, la elección e implantación de un modelo de desarrollo de software, debido a que el mercado cada vez está más informado sobre los productos que necesita y requiere de empresas que le entreguen software efectivo, confiable, en tiempo y a un precio razonable.

Las propuestas de acción para el fortalecimiento de la industria del software han permitido que el país y la universidad identifiquen, como tarea imprescindible para tener éxito, la mejora de procesos en el ciclo de desarrollo del software. A partir de esto se ha iniciado en diferentes áreas productivas de la universidad un programa de mejoras de procesos, una de estas áreas es el Centro de Informatización de la Gestión de Entidades (CEIGE) certificado con nivel dos de CMMI permitiendo una mejora en las actividades del desarrollo de producto.

En el CEIGE existe un departamento de Soluciones Aduanales con la misión de desarrollar productos para la gestión de los principales procesos de la Aduana General de la República de Cuba, el cual tiene un número de dificultades relacionadas con el proceso de desarrollo de software:

- ✓ No posee una guía debidamente documentada que oriente y organice el proceso de desarrollo de software, de forma tal que se garantice un desarrollo en tiempo y con la calidad requerida, provocando innovaciones a partir de las competencias de los autores que intervienen en el proceso de desarrollo de software.
- ✓ El conocimiento adquirido durante el desarrollo de software en el departamento no se registra, el mismo lo poseen algunas personas, trayendo consigo que la ausencia en algún momento de estas provoque retraso en el cronograma de desarrollo, por concepto de preparación de un nuevo personal que asimile dicho conocimiento.
- ✓ La forma actual de desarrollar software en el departamento no se ajusta en su totalidad a las normas establecidas por el modelo de mejora de procesos de CMMI nivel dos.
- ✓ Muchos de los procesos se ejecutan parcialmente provocando atrasos en el cronograma de desarrollo.
- ✓ Las actividades que se deben realizar a lo largo del desarrollo de software no se encuentran formalmente documentadas.

Partiendo de lo mencionado anteriormente se plantea el siguiente **problema a resolver**: las dificultades en la planificación, el control de las actividades y la toma de decisiones durante el proceso de desarrollo de software en el departamento Soluciones Aduanales del CEIGE afectan la organización de su trabajo productivo.

Por tanto el **objeto de estudio** de esta investigación es: proceso de desarrollo de software.

Se plantea como **objetivo general** elaborar el modelo de desarrollo de software para el departamento Soluciones Aduanales del CEIGE con el fin de mejorar la organización de su trabajo productivo.

De ello se deriva que el **campo de acción** en que se realiza la investigación sea proceso de desarrollo de software del departamento Soluciones Aduanales del CEIGE.

Como guía de investigación se trazan los siguientes **objetivos específicos**:

- ✓ Realizar un estudio bibliográfico para establecer un marco teórico referente a los modelos de desarrollo existentes.

- ✓ Realizar un estudio referente al proceso de desarrollo de software en el departamento Soluciones Aduanales del CEIGE.
- ✓ Definir las actividades, artefactos y roles del modelo de desarrollo para el departamento Soluciones Aduanales del CEIGE.
- ✓ Aplicar criterio de experto para validar el modelo de desarrollo propuesto para el departamento Soluciones Aduanales del CEIGE.

Para dar cumplimiento a los objetivos específicos se trazan las siguientes **Tareas de la investigación:**

- ✓ Realizar un estudio bibliográfico sobre conceptos relacionados con el dominio del problema.
- ✓ Realizar un estudio de las áreas de proceso de CMMI nivel dos enmarcándose en el proceso de mejora que se lleva a cabo en la Universidad.
- ✓ Realizar de un estudio de las áreas de proceso de CMMI nivel tres.
- ✓ Levantar la información referente al proceso de desarrollo de software en el departamento Soluciones Aduanales del CEIGE.
- ✓ Definir las actividades del modelo de desarrollo para el departamento Soluciones Aduanales del CEIGE.
- ✓ Definir los artefactos generados por las actividades del modelo de desarrollo para el departamento Soluciones Aduanales del CEIGE.
- ✓ Definir de los roles del modelo de desarrollo para el departamento Soluciones Aduanales del CEIGE.
- ✓ Elaborar el objetivo de la validación.
- ✓ Seleccionar los expertos que conformaran el panel de validación.
- ✓ Elaborar el cuestionario para evaluar la propuesta de solución.
- ✓ Elegir la metodología.
- ✓ Ejecutar la metodología.
- ✓ Procesar la información y obtener los resultados.

De la presente investigación se esperan como **Posibles resultados:**

- ✓ El departamento Soluciones Aduanales del CEIGE contará con la definición de las actividades para guiar la organización de su trabajo productivo.
- ✓ El departamento Soluciones Aduanales del CEIGE contará con la definición de artefactos que deben generarse en cada una de las actividades que se realizan dentro del proceso de desarrollo.

- ✓ El departamento Soluciones Aduanales del CEIGE contará con la definición de los roles en el proceso de desarrollo de Sistemas de Gestión Aduanal.
- ✓ El departamento Soluciones Aduanales del CEIGE contará con un modelo de desarrollo de software para Soluciones de Gestión Aduanal.

### **Métodos y técnicas a utilizar**

Se utilizó como estrategia de investigación la exploratoria y la explicativa. Se exploraron los diferentes modelos de desarrollo de software más utilizados en el mundo, con el objetivo de identificar elementos reutilizables para el desarrollo del modelo que se propone. Se utilizaron métodos teóricos tales como el método Histórico lógico y el Sistemico ya que se planteó el problema como un todo donde los elementos del modelo que lo solucionan se comportan como un sistema sostenible e integral. Se realiza un estudio del estado del arte sobre el Proceso de desarrollo de software utilizando el método Histórico-Lógico, un análisis de las características para la mejora de proceso en el desarrollo del software y la situación actual del departamento al que se le va conformar el modelo mediante el método Analítico-Sintético.

### **Capítulo 1. Fundamentación teórica.**

En este capítulo se brinda un acercamiento a los principales conceptos asociados al dominio del problema y que son abordados a lo largo de la investigación. Se realiza un estudio y análisis del estado del arte de los modelos de desarrollo de software existentes en la actualidad, la mejora de procesos en las actividades del desarrollo de software como es CMMI como apoyo para la solución del problema planteado.

### **Capítulo 2. Propuesta de solución.**

En este capítulo se realiza una breve descripción de las principales características, estructura organizativa y el proceso de desarrollo del departamento Soluciones Aduanales del CEIGE. Además se muestra la propuesta de solución al problema planteado en la investigación, en el mismo se realiza una descripción general del modelo de desarrollo propuesto para el departamento Soluciones Aduanales del CEIGE.

### **Capítulo 3. Validación de la propuesta de solución.**

En este capítulo se realiza la validación de la solución propuesta mediante la aplicación del método Experto de forma tal que se pueda realizar una valoración de los resultados que se obtengan para verificar cuan eficiente será el aplicar dicho modelo en el departamento de Soluciones Aduanales del CEIGE.

# Capítulo 1. Fundamentación teórica

## 1.1. Introducción

En este capítulo se muestran los principales conceptos relacionados con el dominio del problema, se aborda un estudio detallado de los principales modelos de desarrollo de software que son utilizados a nivel mundial realizando una valoración y análisis crítico de cada uno de ellos. Adicionalmente se realiza un análisis de la mejora de los procesos en el ciclo de vida de un proyecto.

## 1.2. Conceptos importantes

En este epígrafe se muestra un análisis detallado de los principales conceptos relacionados con el problema, asociado al proceso y al modelo de desarrollo con el objetivo de lograr un mejor entendimiento de la solución final que se propone.

### **Proceso de desarrollo de software.**

El proceso de desarrollo de software como concepto puede verse desde diferentes puntos de vistas; a continuación se muestran algunos de los conceptos definidos por varios autores.

*Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. (Jacoboson, 2000)*

Dicho proceso es intensamente intelectual, afectado por la creatividad y juicio de las personas involucradas. De manera general este proceso se muestra en la Figura 1.

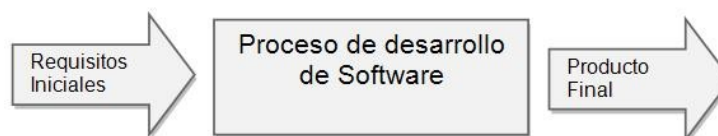


Figura 1: Proceso de desarrollo de software.

*Proceso de desarrollo de software se define como un marco de trabajo para las tareas que se requieren en la construcción de software de alta calidad. (Pressman, 2005)*

*Un proceso de desarrollo software es un conjunto de actividades y resultados asociados que producen un*

*producto de software. Estas actividades son llevadas a cabo por los ingenieros de software. (Sommerville, 2005)*

*Un proceso de desarrollo de software es un conjunto de actuaciones, decisiones, actividades y tareas que se encadenan de forma secuencial y ordenada para conseguir un resultado que satisfaga plenamente los requerimientos del cliente al que va dirigido. (Blaya, 2006)*

*Un proceso de desarrollo de software es el conjunto estructurado de actividades requeridas para desarrollar un sistema de software. (Gervás, 2007)*

Patricio Letelier plantea que “Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente” (Letelier, 2008)

Teniendo en cuenta los conceptos planteados anteriormente se concluye que un proceso de desarrollo de software es un conjunto de actividades que se llevan a cabo en el período de construcción de un producto informático en las que se involucran roles y se generan artefactos.

### Características de un proceso de desarrollo de software

Un proceso de desarrollo de software tiene algunos elementos fundamentales, a continuación se muestra como Pressman caracteriza el proceso de desarrollo de software ver Figura 2. (Pressman, 2005)

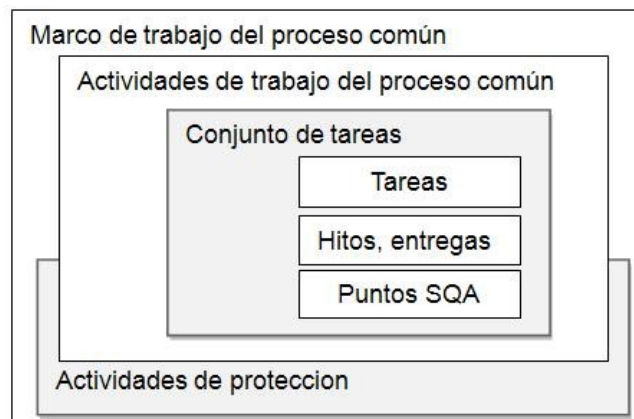


Figura 2: Elementos del proceso de desarrollo de software.

**Un marco común del proceso**, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia del tamaño o complejidad.

**Un conjunto de tareas**, cada uno es una colección de tareas de ingeniería del software, hitos de proyectos, entregas y productos de trabajo del software, y puntos de garantía de calidad, que permiten

que las actividades del marco de trabajo se adapten a las características del proyecto de software y los requisitos del equipo del proyecto.

**Las actividades de protección**, tales como garantía de calidad del software, gestión de configuración del software y medición, abarcan el modelo del proceso. Las actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.

En el proceso de desarrollo de software existe un conjunto de actividades fundamentales y comunes que se encuentran presentes en todos ellos:

**Especificación de software**: se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.

**Diseño e Implementación**: se diseña y construye el software de acuerdo a la especificación.

**Validación**: el software debe validarse, para asegurar que cumpla con lo que quiere el cliente.

**Evolución**: el software debe evolucionar, para adaptarse a las necesidades del cliente.

Además de estas actividades, Pressman también menciona un conjunto de actividades protectoras, que se aplican a lo largo de todo el proceso del software, que a continuación de describen (Pressman, 2005):

**Seguimiento y control del proyecto de software**: permite que el equipo de software evalúe el progreso comparándolo con el plan del proyecto y así elegir las acciones necesarias para mantener el programa.

**Gestión del riesgo**: evalúa los riesgos que pudieran afectar los resultados del proyecto o la calidad del producto.

**Aseguramiento de la calidad del software**: define y conduce las actividades requeridas para asegurar la calidad del software.

**Revisiones técnicas formales**: evalúa los productos del trabajo de la ingeniería del software en un esfuerzo encaminado a describir y eliminar los errores antes de que estos se propaguen hacia la siguiente acción o actividad.

**Medición**: define y recolecta mediciones del proceso, el proyecto y el producto para ayudar al equipo a entregar software que satisfaga las necesidades del cliente; se puede usar en conjunto con todas las otras actividades del marco de trabajo.

**Gestión de la configuración del software**: maneja los efectos del cambio a través del proceso de software.

**Gestión de la reutilización**: define los criterios para la reutilización de productos del trabajo y establece mecanismos para la creación de componentes reutilizables.



**Preparación y producción del producto de trabajo:** abarca las actividades requeridas para crear productos del trabajo como modelos, documentos, registros, formatos y listas.

## **Modelo de desarrollo de software**

Para llevar a cabo el desarrollo de un producto informático es necesario realizar un conjunto de tareas entre la idea inicial y el resultado final que se obtiene. Para que este conjunto de actividades que definen el proceso de desarrollo de software no se ejecuten de forma desorganizada, se hace necesario incorporar una estrategia cuyo objetivo fundamental sea ordenar dichas actividades en el desarrollo del software.

*Un modelo de desarrollo establece el orden en el que se harán las cosas en el proyecto, provee los requisitos de entrada y de salida para cada una de las actividades.* (Victorino, 2011)

Teniendo en cuenta lo mencionado anteriormente, se concluye que un modelo de desarrollo de software es una guía que establece un orden de las actividades en el período de desarrollo de un producto de software, en el cual se tienen en cuenta los artefactos, roles y responsabilidades en cada una de las acciones en el ciclo de vida del proyecto.

### **1.3. Tipos de modelos de desarrollo de software**

El crecimiento de la ingeniería de software ha traído consigo propuestas diferentes para mejorar los resultados en el proceso de construcción y obtención de un producto final. Los modelos de desarrollo de software, como guías, ocupan un lugar importante en la búsqueda del método adecuado para producir software de calidad en cualquier contexto de desarrollo. A continuación se muestra un análisis detallado de los distintos tipos de modelos de desarrollo que existen y los más utilizados.

#### **Codificar y corregir (Code-and-Fix)**

Este es el modelo básico utilizado en los inicios del desarrollo de software. Contiene dos pasos: (Universidad politécnica de Valencia, 2006)

- ✓ Escribir código.
- ✓ Corregir problemas en el código.

Se trata de implementar algo de código y luego pensar acerca de requisitos, diseño, validación y mantenimiento.

### **Ventajas:**

- ✓ El problema es claramente comprendido.
- ✓ Permite corregir errores durante el período de desarrollo de software. (Universidad politécnica de Valencia, 2006)

### **Desventajas:**

- ✓ Después de un número de correcciones, el código puede tener una muy mala estructura, provocando que los arreglos sean muy costosos.
- ✓ Frecuentemente, el software bien diseñado, no se ajusta a las necesidades del usuario, por lo que es rechazado o su reconstrucción es muy cara.
- ✓ El código es difícil de reparar por su pobre preparación para probar y modificar. (Universidad politécnica de Valencia, 2006)

### **Modelos de desarrollo de software prescriptivos.**

Cualquier organización de ingeniería del software debe describir un conjunto único de actividades dentro del marco de trabajo para el (los) proceso(s) de software que adopte. (Chi, 2009)

#### **Modelo en Cascada**

El modelo cascada fue el primer modelo de desarrollo de software que se publicó, se derivó de otros procesos de ingeniería. Este modelo toma las actividades fundamentales de especificación, desarrollo, validación y evolución del proceso de desarrollo de software y las representa como fases separadas del proceso de desarrollo. (Universidad politécnica de Valencia, 2006)

Pressman plantea que, *“El modelo en cascada, es un paradigma que sugiere un enfoque sistemático, secuencial, hacia el desarrollo del software, que se inicia con la especificación de requerimientos del cliente y que continua con la planeación, el modelado, la construcción y el despliegue para culminar en el soporte del software terminado.”* (Pressman, 2005)

El modelo en cascada consta de las siguientes fases:

**Definición de los requisitos:** los servicios, restricciones y objetivos son establecidos con los usuarios del sistema. Se busca hacer esta definición en detalle.

**Diseño de software:** se particiona el sistema en sistemas de software o hardware. Se establece la arquitectura total del sistema. Se identifican y describen las abstracciones y relaciones de los componentes del sistema.

**Implementación y pruebas unitarias:** construcción de los módulos y unidades de software. Se realizan pruebas de cada unidad.

**Integración y pruebas del sistema:** se integran todas las unidades. Se prueban en conjunto. Se entrega el conjunto probado al cliente.

**Operación y mantenimiento:** generalmente es la fase más larga. El sistema es puesto en marcha y se realiza la corrección de errores descubiertos. Se realizan mejoras de implementación. Se identifican nuevos requisitos.

La interacción entre fases puede observarse en la Figura 3. Cada fase tiene como resultado documentos que deben ser aprobados por el usuario.

Una fase no comienza hasta que termine la fase anterior y generalmente se incluye la corrección de los problemas encontrados en fases previas. (Universidad politécnica de Valencia, 2006)

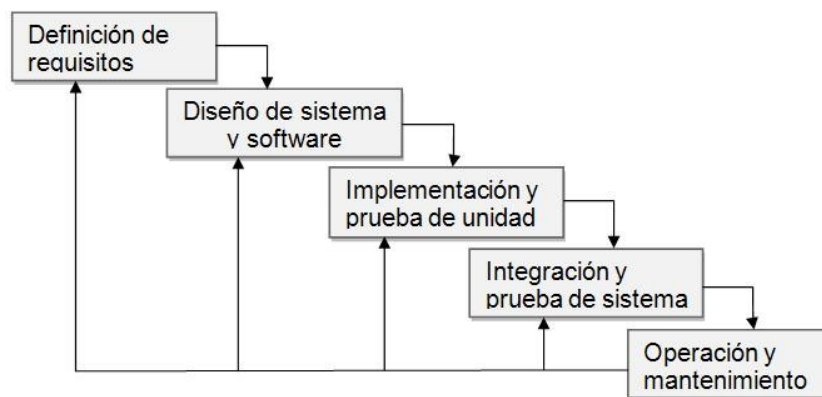


Figura 3: Modelo de desarrollo en cascada.

En la práctica, este modelo no es lineal, involucra varias iteraciones e interacciones entre las distintas fases de desarrollo.

### Ventajas:

- ✓ **La planificación es sencilla.**
- ✓ **Asegura la calidad del producto resultante.**
- ✓ Permite trabajar con personal poco cualificado.
- ✓ **Con este modelo se tiene un seguimiento de todas las fases del proyecto y de su cumplimiento.** (Universidad politécnica de Valencia, 2006)

### Desventajas:

- ✓ Las iteraciones son costosas e implican rehacer trabajo debido a la producción y aprobación de

documentos.

- ✓ Aunque son pocas iteraciones, es normal congelar parte del desarrollo y continuar con las siguientes fases.
- ✓ Los problemas se dejan para su posterior resolución, lo que lleva a que estos sean ignorados o corregidos de una forma poco elegante.
- ✓ Existe una alta probabilidad de que el software no cumpla con los requisitos del usuario por el largo tiempo de entrega del producto.
- ✓ Es inflexible a la hora de evolucionar para incorporar nuevos requisitos. Es difícil responder a cambios en los requisitos. (Universidad politécnica de Valencia, 2006)

Este modelo es recomendado usar, si se entienden a plenitud los requisitos. Aún se utiliza como parte de proyectos grandes.

### Modelos de desarrollo de software evolutivos

El software, como todos los sistemas complejos, evoluciona con el tiempo. Los requisitos de los negocios y productos a menudo cambian conforme se realiza el desarrollo. (Chi, 2009)

La idea detrás de este modelo es el desarrollo de una implantación del sistema inicial, exponerla a los comentarios del usuario, refinarla en N versiones hasta que se desarrolle el sistema adecuado. (Universidad politécnica de Valencia, 2006) Ver Figura 4.

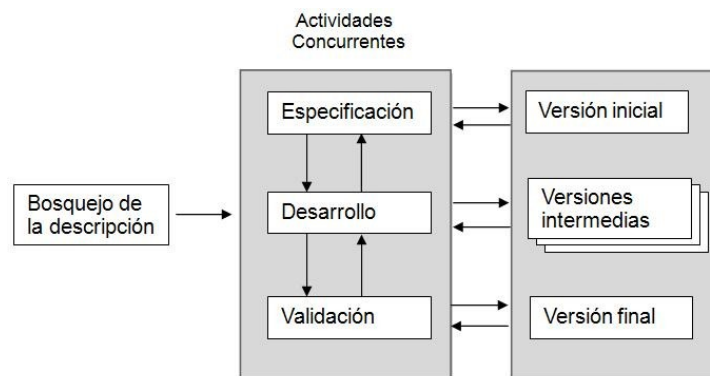


Figura 4: Modelo de desarrollo evolutivo.

En la figura mostrada anteriormente se observa como actividades concurrentes: especificación, desarrollo y validación, se realizan durante el desarrollo de las versiones hasta llegar al producto final.

Existen dos tipos de desarrollo evolutivo:

**Desarrollo Exploratorio:** El objetivo de este enfoque es explorar con el usuario los requisitos hasta llegar

a un sistema final. El desarrollo comienza con las partes que se tiene más claras. El sistema evoluciona conforme se añaden nuevas características propuestas por el usuario. (Universidad politécnica de Valencia, 2006)

**Enfoque utilizando prototipos:** El objetivo es entender los requisitos del usuario y trabajar para mejorar la calidad de los requisitos. A diferencia del desarrollo exploratorio, se comienza por definir los requisitos que no están claros para el usuario y se utiliza un prototipo para experimentar con ellos. El prototipo ayuda a terminar de definir estos requisitos. (Universidad politécnica de Valencia, 2006)

### **Desarrollo en espiral**

El desarrollo en espiral es un ejemplo de los modelos de proceso evolutivo (ver Figura 5) es actualmente uno de los más conocidos y fue propuesto por Boehm. El ciclo de desarrollo se representa como una espiral, en lugar de una serie de actividades sucesivas con retrospectiva de una actividad a otra. Pressman plantea que, *“el modelo en espiral es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de la construcción de prototipos con los aspectos controlados y sistemáticos del modelo cascada”* (Pressman, 2005)

Cada ciclo de desarrollo se divide en cuatro fases:

**Definición de objetivos:** Se definen los objetivos. Se definen las restricciones del proceso y del producto. Se realiza un diseño detallado del plan administrativo. Se identifican los riesgos y se elaboran estrategias alternativas dependiendo de estos.

**Evaluación y reducción de riesgos:** Se realiza un análisis detallado de cada riesgo identificado. Pueden desarrollarse prototipos para disminuir el riesgo de requisitos dudosos. Se llevan a cabo los pasos para reducir los riesgos.

**Desarrollo y validación:** Se escoge el modelo de desarrollo después de la evaluación del riesgo. El modelo que se utilizará depende del riesgo identificado para esa fase.

**Planificación:** Se determina si continuar con otro ciclo. Se planea la siguiente fase del proyecto.

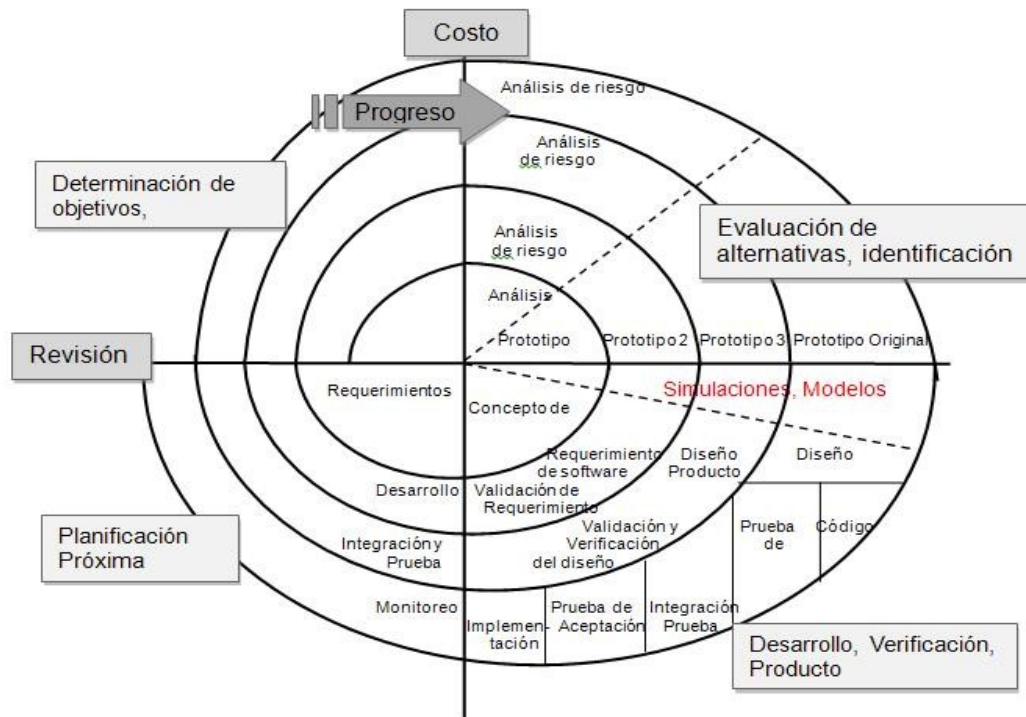


Figura 5: Modelo de desarrollo en Espiral.

Este modelo a diferencia de los otros, toma en consideración explícitamente el riesgo, esto es una actividad importante en la administración del proyecto. El ciclo de vida inicia con la definición de los objetivos. De acuerdo a las restricciones se determinan distintas alternativas. Se identifican los riesgos al sopesar los objetivos contra las alternativas. Se evalúan los riesgos con actividades como análisis detallado, simulación y prototipos. Se desarrolla un poco el sistema. Se planifica la siguiente fase.

### Ventajas:

- ✓ **Se obtiene una rápida realimentación del usuario, ya que las actividades de especificación, desarrollo y pruebas se ejecutan en cada iteración.**
- ✓ La especificación puede desarrollarse de forma creciente.
- ✓ **Los usuarios y desarrolladores logran un mejor entendimiento del sistema. Esto se refleja en una mejora de la calidad del software.**
- ✓ Es más efectivo que el modelo de cascada, ya que cumple con las necesidades inmediatas del cliente. (Universidad politécnica de Valencia, 2006)

### **Desventajas:**

- ✓ Los administradores necesitan entregas para medir el progreso. Si el sistema se necesita desarrollar rápido, no es efectivo producir documentos que reflejen cada versión del sistema.
- ✓ Los cambios continuos pueden ser perjudiciales para la estructura del software haciendo costoso el mantenimiento.
- ✓ Para el rápido desarrollo se necesitan herramientas que pueden ser incompatibles con otras o que poca gente sabe utilizar. (Universidad politécnica de Valencia, 2006)

Este modelo es efectivo en proyectos pequeños (menos de 100.000 líneas de código) o medianos (hasta 500.000 líneas de código) con poco tiempo para su desarrollo y sin generar documentación para cada versión. (Universidad politécnica de Valencia, 2006)

Para proyectos largos es mejor combinar lo mejor del modelo de cascada y evolutivo: se puede hacer un prototipo global del sistema y posteriormente reimplementarlo con un acercamiento más estructurado. Los subsistemas con requisitos bien definidos y estables se pueden programar utilizando cascada y la interfaz de usuario se puede especificar utilizando un enfoque exploratorio.

### **Modelo especializado de procesos**

Pressman plantea que, los modelos especializados de procesos adoptan muchas de las características de uno o más de los modelos convencionales, tienden a aplicarse cuando se ha elegido un enfoque de ingeniería del software definido de una manera muy estrecha. (Pressman, 2005)

### **Desarrollo formal de sistemas.**

Este modelo se basa en transformaciones formales de los requisitos hasta llegar a un programa ejecutable. La Figura 6 ilustra un paradigma ideal de programación automática. Se distinguen dos fases globales: especificación (incluyendo validación) y transformación. (Universidad politécnica de Valencia, 2006)

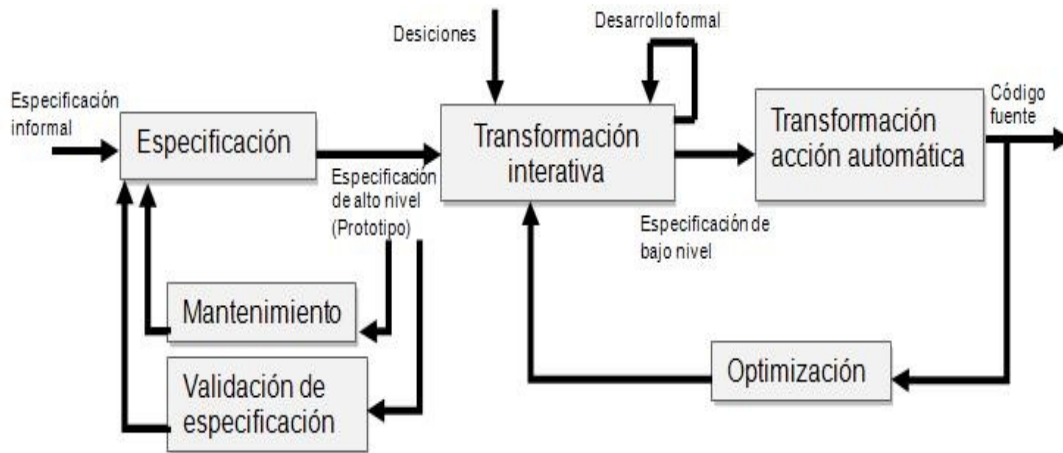


Figura 6: Paradigma de programación automática.

Las características principales de este paradigma son: la especificación (es formal, ejecutable y constituye el primer prototipo del sistema), la especificación es validada mediante prototipación. Posteriormente, a través de transformaciones formales la especificación se convierte en la implementación del sistema, en el último paso de transformación se obtiene una implementación en un lenguaje de programación determinado.

El mantenimiento se realiza sobre la especificación (no sobre el código fuente), la documentación es generada automáticamente y el mantenimiento es realizado por repetición del proceso (no mediante parches sobre la implementación). (Universidad politécnica de Valencia, 2006)

### Ventajas:

- ✓ Permite demostrar la corrección del sistema durante el proceso de transformación. Así, las pruebas que verifican la correspondencia con la especificación no son necesarias.
- ✓ Es atractivo sobre todo para sistemas donde hay requisitos de seguridad y confiabilidad importantes.
- ✓ Requiere desarrolladores especializados y experimentados en este proceso para llevarse a cabo. (Universidad politécnica de Valencia, 2006)

### Desventajas:

- ✓ Hace falta una formación especializada para aplicar la técnica
- ✓ Muchos aspectos de los de los sistemas reales son difíciles de especificar formalmente como:



interfaz de usuario y requisitos no funcionales. (Universidad politécnica de Valencia, 2006)

### Desarrollo basado en reutilización

Como su nombre lo indica, es un modelo fuertemente orientado a la reutilización. Este modelo consta de 4 fases ilustradas en la Figura 7. (Universidad politécnica de Valencia, 2006). A continuación se describe cada fase:

**Análisis de componentes:** Se determina qué componentes pueden ser utilizados para el sistema en cuestión. Casi siempre hay que hacer ajustes para adecuarlos.

**Modificación de requisitos:** Se adaptan los requisitos para concordar con los componentes de la etapa anterior. Si no se puede realizar modificaciones en los requisitos, hay que seguir buscando componentes más adecuados (fase 1).

**Diseño del sistema con reutilización:** Se diseña o reutiliza el marco de trabajo para el sistema. Se debe tener en cuenta los componentes localizados en la fase 2 para diseñar o determinar este marco.

**Desarrollo e integración:** El software que no puede comprarse, se desarrolla. Se integran los componentes y subsistemas. La integración es parte del desarrollo en lugar de una actividad separada.

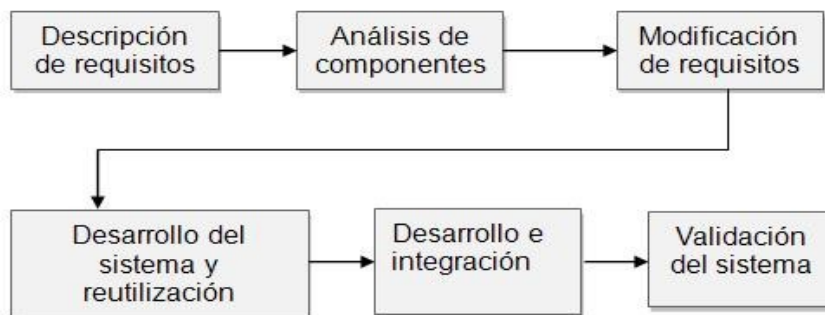


Figura 7: Desarrollo basado en reutilización de componentes.

### Ventajas:

- ✓ *Disminuye el costo y esfuerzo de desarrollo.*
- ✓ *Reduce el tiempo de entrega.*
- ✓ *Disminuye los riesgos durante el desarrollo.* (Universidad politécnica de Valencia, 2006)

### Desventajas:

- ✓ Los compromisos en los requisitos son inevitables, por este motivo puede que el software no cumpla las expectativas del cliente.
- ✓ Las actualizaciones de los componentes adquiridos no están en manos de los desarrolladores del sistema. (Universidad politécnica de Valencia, 2006)

### Modelos de procesos incrementales.

Los modelos incrementales del proceso de software producen software como una serie de entregas de incrementos. (Chi, 2009)

Según Sommerville, *“la entrega incremental es un enfoque intermedio que combina las ventajas de los modelos cascada y evolutivo. En un proceso de desarrollo incremental, los clientes identifican, a grandes rasgos los servicios que proporcionará el sistema.”* (Sommerville, 2005)

### Desarrollo incremental.

Mills, sugirió el enfoque incremental de desarrollo como una forma de reducir la repetición del trabajo en el proceso de desarrollo y dar oportunidad de retrasar la toma de decisiones en los requisitos hasta adquirir experiencia con el sistema (ver Figura 8). Es una combinación del Modelo de Cascada y el Modelo Evolutivo.

Reduce el rehacer trabajo durante el proceso de desarrollo y da oportunidad para retrasar las decisiones hasta tener experiencia en el sistema.

Durante el desarrollo de cada incremento se puede utilizar el modelo de cascada o evolutivo, dependiendo del conocimiento que se tenga sobre los requisitos a implementar. Si se tiene un buen conocimiento, se puede optar por cascada, si es dudoso, evolutivo.

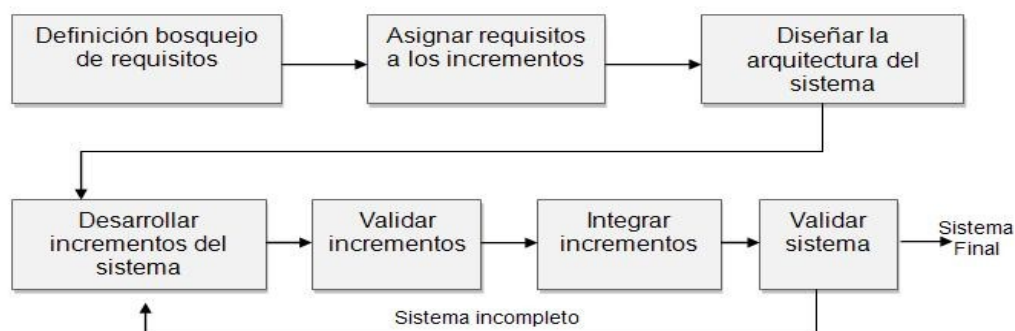


Figura 8: Modelo de desarrollo iterativo incremental.

### **Ventajas:**

- ✓ *Los clientes no esperan hasta el fin del desarrollo para utilizar el sistema. Pueden empezar a usarlo desde el primer incremento.*
- ✓ *Los clientes pueden aclarar los requisitos que no tengan claros conforme ven las entregas del sistema.*
- ✓ *Se disminuye el riesgo de fracaso de todo el proyecto, ya que se puede distribuir en cada incremento.*
- ✓ *Las partes más importantes del sistema son entregadas primero, por lo cual se realizan más pruebas en estos módulos y se disminuye el riesgo de fallos.* (Universidad politécnica de Valencia, 2006)

### **Desventajas:**

- ✓ Cada incremento debe ser pequeño para limitar el riesgo (menos de 20.000 líneas).
- ✓ Cada incremento debe aumentar la funcionalidad.
- ✓ Es difícil establecer las correspondencias de los requisitos contra los incrementos.
- ✓ Es difícil detectar las unidades o servicios genéricos para todo el sistema. (Universidad politécnica de Valencia, 2006)

Una vez estudiado los modelos expuestos anteriormente se concluye que no existe un modelo de desarrollo universal que se adapte completamente a las circunstancias del departamento por lo que se decide seleccionar los puntos fuertes a partir de las ventajas resaltadas de algunos de los modelos estudiados. Teniendo en cuenta aspectos tales como la calidad del producto resultante, la gestión de riesgos, la corrección durante el desarrollo y visión del progreso por parte del cliente y jefe de proyecto, se puede conformar un modelo de desarrollo adaptable al departamento y que proporcione efectividad en el desarrollo del producto.

### **1.4. Mejora de proceso en el desarrollo de software**

La gestión eficaz de los activos de las organizaciones es indispensable para el éxito de su actividad, por lo que necesitan de una guía que les ayude y facilite la gestión de las actividades en el ciclo de vida del software. En la actualidad existen modelos de madurez, estándares que proporcionan una mejora continua de las organizaciones para obtener productos de mayor calidad, un ejemplo de esto es CMMI,

estándar que permite la evaluación en los procesos de una organización.

Mary Beth plantea que: *El CMMI proporciona una oportunidad para evitar o eliminar estos canales y barreras, apoyándose en los modelos integrados que trascienden disciplinas. El CMMI para Desarrollo contempla las buenas prácticas relativas a las actividades de desarrollo y mantenimiento aplicadas a productos y servicios. Introducir y fundamentar el uso de los modelos de mejora para lograr la evaluación de la madurez de la organización que use el modelo de desarrollo escogido.* (Chrissis, 2009)

El modelo CMMI cuenta con dos representaciones distintas: continua y escalonada o por etapas. El modelo escalonado, o centrado en la madurez de la organización establece cinco Niveles de Madurez para clasificar a las organizaciones, en función de qué áreas de procesos consiguen sus objetivos y se gestionan con principios de ingeniería. A continuación una breve descripción de los niveles de madurez:

**Nivel 1 (Inicial):** El proceso es impredecible, es reactivo y pobremente controlado.

**Nivel 2 (Administrado):** El proceso es reactivo y se caracteriza por su aplicación a proyectos.

**Nivel 3 (Definido):** El proceso es proactivo y se ve a nivel de la organización.

**Nivel 4 (Administrado Cuantitativamente):** El proceso es medido y controlado.

**Nivel 5 (Optimizado):** El proceso se enfoca en la mejora continua.

### **CMMI nivel 2 (Administrado).**

El nivel dos de CMMI consta de siete áreas de procesos, que a continuación se describen brevemente:

**Gestión de requisitos (REQM):** gestión de los requisitos del proyecto y los productos esperados.

Identificación de inconsistencias entre los requisitos y el plan de proyecto.

**Planificación de proyectos (PP):** establecimiento y mantenimiento de planes que definen las actividades a ejecutar durante el proyecto.

**Seguimiento y control de proyectos (PMC):** control del progreso, identificación de desviaciones y toma de decisiones correctivas.

**Acuerdo con proveedores (SAM):** gestión de la adquisición de productos y servicios a través de la existencia de un acuerdo formal con el proveedor.

**Aseguramiento de la calidad (PPQA):** garantía de la calidad de los procesos aplicados y de los productos obtenidos.

**Gestión de la configuración (CM):** establecimiento y mantenimiento de la integridad de los productos generados en el proyecto.

**Medición y análisis (MA):** desarrollo, mantenimiento y uso de capacidades de medida que soporten las necesidades de información de la organización.

**CMMI nivel 3 (Definido).**

**Desarrollo de Requisitos (RD):** producir y analizar los requerimientos del cliente, del producto y de los componentes del producto.

**Solución Técnica (TS):** diseñar, desarrollar e implementar soluciones para los requerimientos.

**Integración del Producto (PI):** manejar el ensamblado de componentes y comprobación del funcionamiento del producto ensamblado antes de la entrega.

**Verificación (VER):** asegurar que los productos de trabajo seleccionados cumplen sus requerimientos especificados.

**Validación (VAL):** demostrar que un producto o componente de producto se ajusta a su uso previsto cuando se sitúa en su entorno previsto.

**Enfoque de Procesos Organizacionales (OPF):** planificar e implementar las mejoras a los procesos de la organización.

**Definición de Procesos Organizacionales (OPD):** establecer y mantener un conjunto de activos de proceso.

**Capacitación Organizacional (OT):** establece y mantener una capacidad de entrenamiento que permite desarrollar los conocimientos y habilidades necesarias para las actividades de la organización.

**Administración del Proyecto Integrado (IPM):** administrar el proyecto y el involucramiento de todos los interesados mediante un proceso que, basado en el proceso estándar de la organización, ha sido adaptado a las necesidades particulares del proyecto.

**Administración de Riesgos (RSKM):** identificar los problemas potenciales antes de que ocurran para que las actividades de gestión de riesgos puedan ser planificadas y utilizadas según sea necesario a lo largo de la vida del producto o del proyecto para mitigar los impactos adversos para alcanzar los objetivos.

**Equipo de Trabajos Integrados (IT):** establecer y mantener equipos integrados para el desarrollo de productos.

**Administración de Proveedores Integrados (ISM):** identificar potenciales proveedores de productos y establecer con ellos relaciones mutuamente beneficiosas.

**Análisis de Decisión y Resolución (DAR):** lograr que las decisiones sean tomadas de acuerdo a un

proceso formal.

**Ambiente Organizacional para la Integración (OEI):** proveer la infraestructura necesaria para que los distintos grupos implicados en el proyecto trabajen de manera integrada.

### 1.5. Conclusiones Parciales

Después de un estudio minucioso de cada uno de los modelos expuestos anteriormente y haciendo un análisis y valoración de los mismos, destacando los aspectos más importantes de cada uno de ellos para tenerlos en cuenta en la conformación del modelo que dará solución al problema planteado, se puede concluir que de manera general:

- ✓ No existe un modelo de desarrollo de software universal que sea efectivo para todos los contextos de proyectos de desarrollo.
- ✓ Fusionar los puntos fuertes de algunos de los modelos analizados, teniendo en cuenta las ventajas resaltadas para conformar un modelo de desarrollo híbrido, que proporcione efectividad en las actividades a lo largo del ciclo de vida de desarrollo de los productos del departamento Soluciones Aduanales del CEIGE.
- ✓ Hacer uso de las buenas prácticas de CMMI nivel dos para la mejora de los procesos en cada etapa de desarrollo de software y así lograr una mejor organización en las actividades.
- ✓ El empleo de CMMI no solo permitirá una mejora de los procesos, ofreciendo a la organización un nivel de capacidad y madurez elevado, sino que le permite además un posicionamiento reconocido en el mercado mundial.
- ✓ La incorporación de las buenas prácticas de CMMI en el modelo de desarrollo proporciona una mejor organización y gestión de las actividades en las organizaciones, mayor productividad, calidad de los productos y servicios a los clientes.
- ✓ Elaborar un modelo de desarrollo de software con las siguientes características: orientadas a la arquitectura, iterativa e incremental.

## Capítulo 2. Propuesta de solución

### 2.1. Introducción

En este capítulo se detallan las características, la estructura organizativa del departamento Soluciones Aduanales del CEIGE y los procesos que se deben llevar a cabo en la dimensión de desarrollo de los productos. En la dimensión de desarrollo se describen los procesos que tienen lugar dentro del departamento Soluciones Aduanales detallando para cada proceso el flujo de actividades, roles que las ejecutan y artefactos que se generan. Se utilizó la herramienta Case Visual Paradigm (VP) para el modelado de las actividades de las fases usando la representación modelado de procesos del negocio. Se hace referencia además a los procesos del nivel dos de CMMI que se llevan a cabo durante la producción del software, dada la importancia que tienen los mismos para lograr eficiencia, productividad y organización dentro del departamento.

#### 2.1.1. Información del departamento de Soluciones Aduanales del CEIGE

El departamento de Soluciones Aduanales tiene como principal responsabilidad desarrollar sistemas informáticos, para gestionar los procesos de negocio de las entidades aduaneras en su relación con el comercio exterior de nuestro país. La Aduana General de la República de Cuba (AGR) en su condición de órgano de control del comercio exterior establece regulaciones acerca de los procedimientos necesarios para informatizar el control de personas, artículos o mercancías y tripulantes en las fronteras cubanas. Para ello la AGR se basa en un conjunto de leyes y regulaciones que definen algunas reglas para la entrada y salida de los mismos y que forman parte del sistema de control estatal para garantizar la seguridad y protección de la economía nacional, el cumplimiento de la legislación nacional así como de la seguridad de todos los habitantes de la isla.

#### Antecedentes del departamento

La relación de trabajo entre la UCI y la AGR se remonta al curso 2004-2005. Inicialmente la planificación del desarrollo así como la identificación de los componentes a implementar era responsabilidad de los especialistas funcionales de la AGR. A medida que se fue ganando en madurez en la organización de los procesos de desarrollo de software en la universidad los equipos encargados de revisar hicieron posible que la proyección de los sistemas fuera más colegiada y coordinada entre ambas partes. La estructura productiva de la UCI fue evolucionando, se creó el Polo de Sistemas Aduaneros que agrupó y gestionó los

desarrollos para la AGR durante dos años. Al moverse hacia el trabajo en los centros productivos se pasan todos los desarrollos de dicho polo hacia el Centro de Informatización de la Gestión de Entidades (CEIGE) creando el departamento de Soluciones Aduanales.

### Estructura organizativa del departamento.

El Departamento de Soluciones Aduanales cuenta con una estructura central que se encarga de la gestión de los procesos primarios de toda estructura organizativa como son los recursos humanos y la gestión tecnológica. El organigrama del departamento (Ver Figura 9) es el que se muestra a continuación:

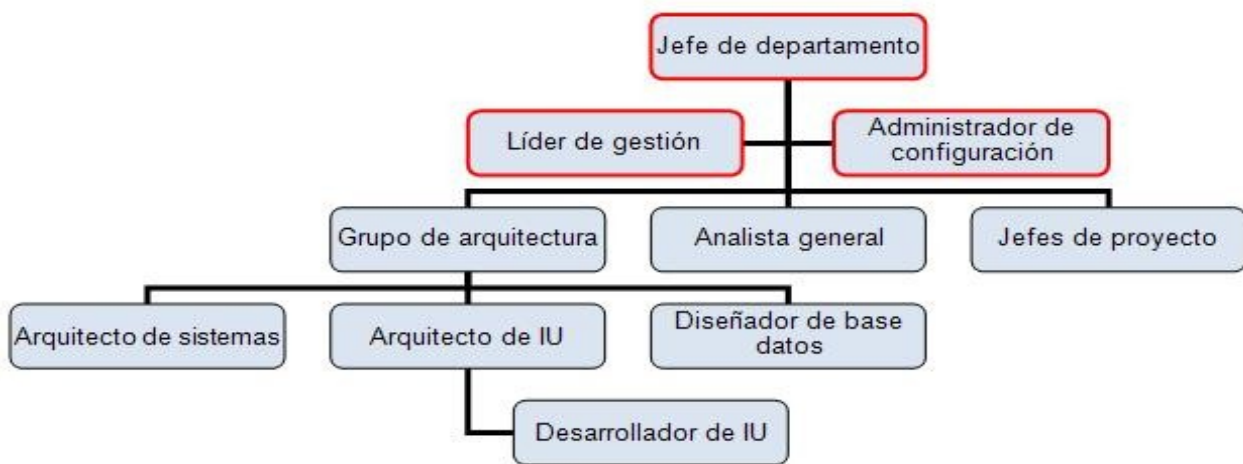


Figura 9: Estructura organizativa del departamento Soluciones Aduanales.

De este organigrama es conveniente detallar dos elementos particulares. En primer lugar la centralización de los elementos de desarrollo de Interfaz de Usuario (IU) en un grupo que no se subordina a ningún proyecto en particular sino que sirve a cada proyecto como clientes en función de una planificación también central que prioriza las tareas en función de los hitos pactados para entregar determinado componente. El otro elemento que sustenta la decisión de centralizar el desarrollo de IU radica en el esfuerzo que se está llevando a cabo de estandarizar de manera radical el diseño, pautas de codificación y estrategia de implementación de las pantallas. Para lograr de manera eficiente este proceso es imprescindible al menos al inicio tener centralizado el equipo de desarrollo de IU. La otra definición organizativa a destacar es la también centralización del trabajo de los arquitectos de datos. Producto de la alta integración que existe entre los subsistemas se hace necesario que los arquitectos de datos tengan control de todas las entidades de base de datos que vayan surgiendo en cada uno de los desarrollos. De



esta forma cada proyecto tendrá como asesor un arquitecto de datos pero las decisiones relativas a los modelos de datos serán a nivel general para todo el departamento. Ver Figura 10.

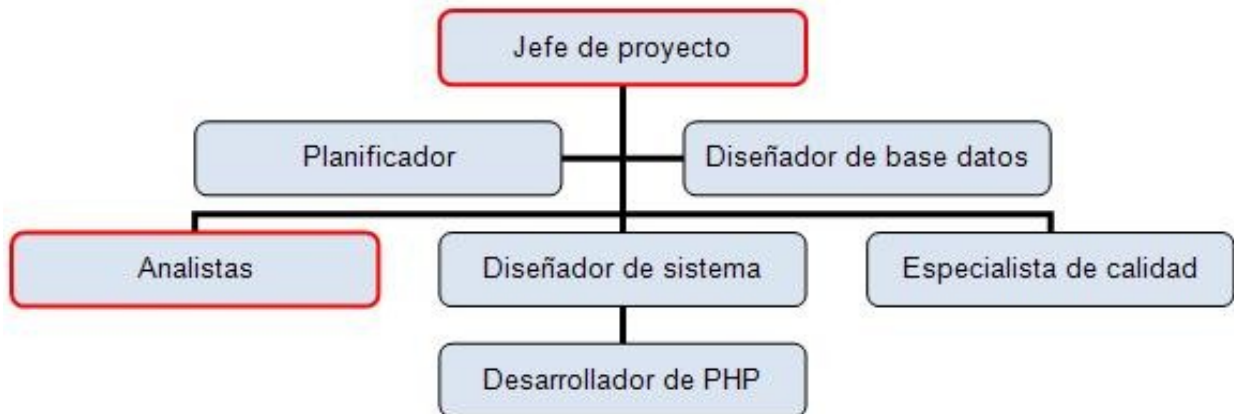


Figura 10: Estructura organizativa de un proyecto del departamento Soluciones Aduanales.

En un proyecto de los que se subordina al departamento existe al menos un diseñador de sistemas encargado de la identificación de los componentes, módulos y subsistemas necesarios para dar cumplimiento a los requisitos identificados por los analistas. Las tareas a los desarrolladores de PHP se definen en función de las necesidades del proyecto por lo tanto no están subordinados permanentemente a ningún subsistema o módulo, ni a ningún diseñador de sistema. Es necesario acotar que pudieran existir casos excepcionales de tareas que necesiten mucho nivel de especialización o dedicación a los que se podría asignar un grupo de programadores durante un espacio largo de tiempo. Los roles de planificador y arquitecto de datos se consideran en este ámbito de apoyo al jefe de proyecto en las tareas de organización y asignación de tareas así como las definiciones a nivel de modelo de datos. En cada proyecto existe un especialista de calidad encargado de dar seguimiento a la calidad de implementación del modelo de desarrollo y otras definiciones producto de la implantación del modelo CMMI nivel dos en el CEIGE. Además se encarga de gestionar la relación con todos los niveles de liberación de calidad. Actualmente el centro se encuentra en un proceso de reestructuración en el cual se definirá una nueva estructura para el departamento, los roles señalados en rojo se mantienen para esta nueva estructura.

## 2.2. Modelo de desarrollo de software

A continuación se expone el modelo de forma general que guiará el desarrollo de los productos en el departamento de Soluciones Aduanales, en el mismo se muestran los roles definidos y la descripción del flujo de actividades por fases a seguir en el proceso de desarrollo del software. Ver figura 11. La plantilla del modelo de desarrollo se muestra en el Anexo 5.

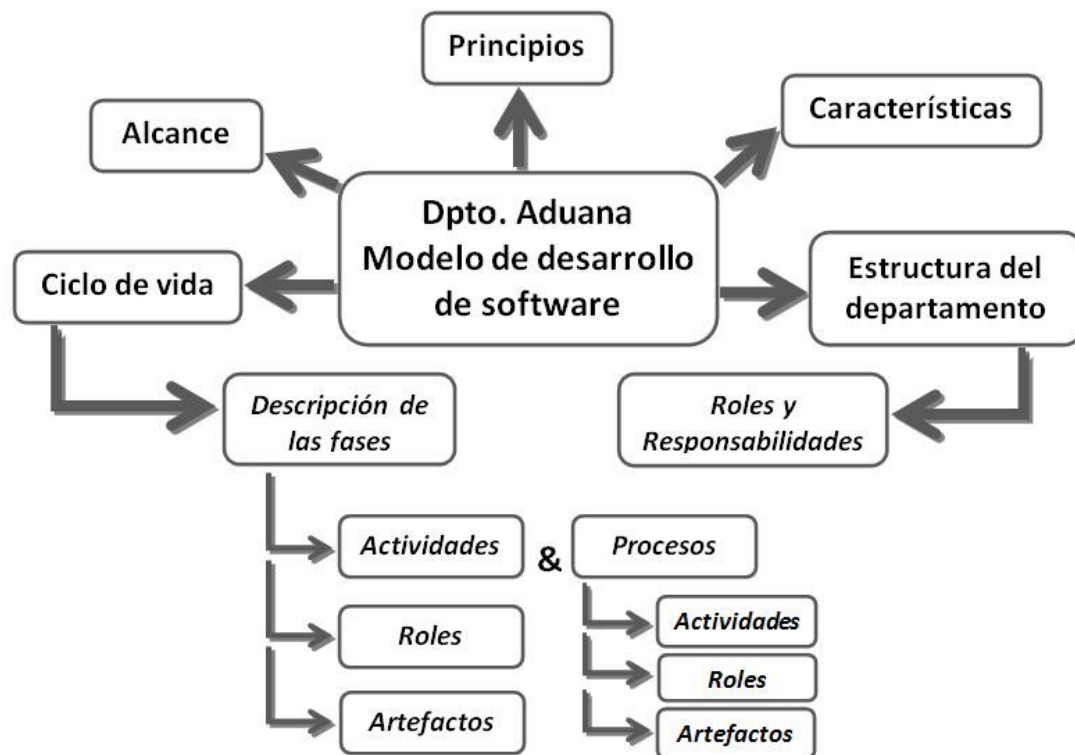


Figura 11: Estructura general del modelo de desarrollo de software del departamento aduana.

**2.2.1. Alcance:** todos los proyectos inscritos en el Departamento de Soluciones Aduanales.

### 2.2.2. Características del modelo

✓ **Orientado en la arquitectura**

Uno de los mayores riesgos que se corren en el desarrollo de sistemas grandes, es que, al hacer cambios a un fragmento del sistema, otras partes se ven afectadas, el no tener en cuenta tal riesgo puede convertir un proceso de cambio en algo muy crítico y difícil de manejar. Para esto se propone que el modelo tenga

una arquitectura resistente y flexible que organice los componentes de manera tal que sean manejables, evitando que los cambios conlleven a grandes esfuerzos. La arquitectura es una vista del diseño completo con las características más importantes del software; organiza los elementos (componentes, clases) por los que está formado un sistema, las relaciones entre ellos y el contexto en el que se implantarán, los principios que orientan su diseño y evolución. En el modelo de desarrollo de software propuesto en la fase de Requisitos se genera las vistas de arquitectura las cuales le permite al desarrollador ver el software, antes de comenzar su desarrollo, desde varios puntos de vistas: seguridad, infraestructura, integración, presentación, entre otros. En las fases Análisis y diseño e Implementación se refina la arquitectura para lograr un equilibrio entre los componentes a desarrollar y la arquitectura, los componentes se ajusten a la arquitectura definida y la arquitectura permita el desarrollo de los componentes y así el sistema evolucione no solo en esta etapa sino en sus versiones futuras.

### ✓ **Iterativo e incremental**

El desarrollo de software se le dedica muchas horas de esfuerzos que puede durar varios meses, es factible dividir el trabajo en partes más pequeñas, donde cada parte es una iteración que resulta en un incremento. En el modelo propuesto en la fase Estudio preliminar se obtiene el plan de desarrollo de software, en el mismo se desarrolla el plan de iteraciones donde se define la cantidad de iteraciones que se realizará a lo largo del proceso de desarrollo de software. Las iteraciones constituyen el paso por el flujo de actividades propuesto, y los incrementos el crecimiento del producto. El control de las iteraciones reduce el costo de los riesgo al costo de un solo incremento, si se tiene que repetir la iteración, solo se pierde el esfuerzo mal empleado de la iteración, no el valor del producto entero. Para esto al final de cada fase se realiza el proceso de creación y liberación de la línea base, una línea base es un conjunto de productos de trabajo formalmente revisados y aprobados que sirve de base para desarrollos posteriores o entregas, al liberarse una línea base los elementos que la componen son revisados y aprobados, un cambio a una línea base se realiza mediante una solicitud de cambio.

### **2.2.3. Principios**

El modelo persigue las tendencias de los modelos de desarrollo de software y políticas de las áreas de procesos de CMMI nivel dos estudiados para garantizar rapidez, buen funcionamiento y una mejora continua en el desarrollo de los procesos llevados a cabo en el departamento de Soluciones Aduanales. Los principios fundamentales en los que se basa son:

- ✓ Proporcionar una guía adecuada para ejecutar las actividades, desarrollar los productos de trabajo

y ofrecer los servicios del proceso de desarrollo de software. Este principio se evidencia en el proceso de definición del plan de proyecto, el cual se realiza en la fase Estudio preliminar donde se obtiene el plan de desarrollo de software que sirve de guía a lo largo del proceso de desarrollo de software.

- ✓ Garantizar un buen ambiente de trabajo y que los recursos necesarios para desarrollar las actividades definidas por el plan, estén disponibles cuando se necesitan. Este principio se evidencia desde la primera fase de desarrollo cuando se planea y se asignan recursos al proyecto, en el cual se tiene el plan de recursos humanos y el plan de recursos materiales y se solicitan los recursos planificados. En el despliegue de la administración de la configuración se crea el repositorio del proyecto, se genera el acceso al mismo, se identifican los niveles de seguridad y los mecanismos de respaldo y recuperación. Luego en cada una de las demás fases con el proceso de contabilización y auditoría de la configuración se gestiona la integridad de los productos, control y cambios sobre ellos y se mantiene la visibilidad de estos según los niveles de seguridad apropiados.
- ✓ Identificar los roles involucrados relevantes en cada proceso. Establecer y mantener la participación prevista de los mismos en actividades como planeación, decisión, compromiso, comunicación, coordinación, revisiones, definición de requisitos y resolución de problemas durante la ejecución del proceso. Este principio se evidencia en la ejecución de los procesos de revisiones y evaluaciones que se llevan a cabo a lo largo del ciclo de vida, por ejemplo PMC y PPQA donde el proyecto considera los roles involucrados en las evaluaciones, estos son las personas que los ejecutarán y los responsables de efectuar el seguimiento de los elementos durante todo el proceso.
- ✓ Tener una buena relación y entendimiento con el cliente a lo largo del proceso de desarrollo del software. Este principio se evidencia desde la participación del cliente en el inicio de desarrollo del producto, en la revisión y aprobación del proyecto técnico donde se chequea con el cliente que la solución propuesta se ajuste a sus necesidades, cumple sus expectativas y está dentro del presupuesto. En otras fases se realiza el proceso de compromiso al plan donde se realiza una reunión con el cliente, se revisan los compromisos, las fechas pactadas y entregables. Además, se comunica el estado del proyecto y se registran los datos de las revisiones permitiéndole al cliente una visión del desarrollo.

### **Ventajas:**

- ✓ Buena planificación.
- ✓ Buen control y seguimiento de las actividades definidas en el ciclo de vida permitiendo la visión del progreso del desarrollo.
- ✓ Mejora de la calidad del producto.
- ✓ Definición de entregables formales.

### **Desventajas:**

- ✓ El éxito está dado por una buena especificación de requisitos y buena gestión de los cambios que puedan surgir.
- ✓ Se necesita de un personal preparado y comprometido con el rol y sus responsabilidades.
- ✓ Se necesita un personal capacitado en las áreas de proceso de mejorar de CMMI.

#### **2.2.4. Elementos del modelo**

El modelo de desarrollo que se propone en este trabajo está conformado por 8 fases, un conjunto de actividades asociadas a cada una de las fases, los responsables de realizar cada una de las actividades y los artefactos generados una vez que se realizan las actividades.

#### **2.2.5. Roles y responsabilidades**

Se realizó una selección de los roles que intervienen en el modelo, así como sus responsabilidades y el tipo de tarea que deben de realizar dichos roles en el proceso de desarrollo de software, ver Anexo 1, para realizar dicha selección se tuvo en consideración lo que plantea CMMI nivel dos.

#### **2.2.6. Ciclo de vida**

A continuación se muestran las fases que conformarán el modelo de desarrollo, y para su definición se tuvo en cuenta lo planteado por el proceso de mejora que se lleva a cabo en la universidad, aunque no incluye la última fase que este propone (Soporte), pues en el departamento no se realiza ninguna actividad relacionada con esta fase. Se le agregó las actividades específicas del departamento por la forma en que la realizan y algunos artefactos que generan, las mismas se reflejan en los diagramas subrayadas en color rojo. Ver figura 12.

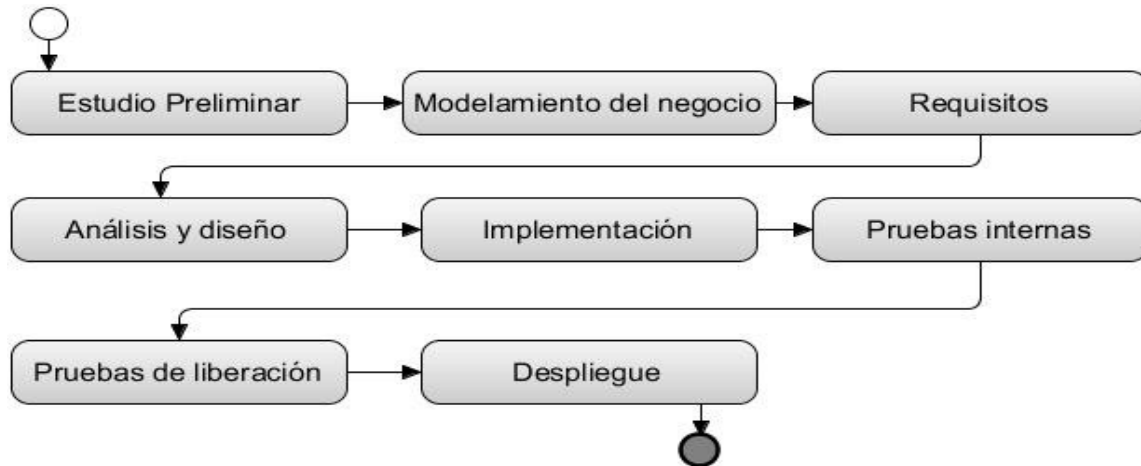


Figura 12: Ciclo de vida de los proyectos del departamento Soluciones Aduanales.

### 2.2.7. Descripción de las fases.

**Fase Estudio Preliminar:** en esta la fase se llevan a cabo las actividades relacionadas con la planeación del proyecto a un alto nivel. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto y realizar estimaciones de tiempo, esfuerzo y costo.

**Fase Modelado de negocio:** el Modelado del Negocio es la fase destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea automatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para la descripción y modelado de negocio pueden ser utilizadas diferentes técnicas como el Modelado Business Process Modeling Notation (BPMN).

**Fase de Requisitos:** el esfuerzo principal en la fase de Requisitos es desarrollar un modelo del sistema que se va a construir. Se incluye un conjunto de procesos, servicios que describen todas las interacciones que tendrán los usuarios con el software, estos responden a los requisitos funcionales del sistema. La especificación de los requisitos incluye también requisitos no funcionales.

**Fase Análisis y Diseño:** durante esta fase, si es necesario, a través de los modelos de análisis los requisitos descritos durante la fase de Requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de los mismos y una descripción que sea fácil de mantener y ayude a estructurar el sistema. Además durante esta fase es modelado el sistema y su forma para que cumpla todos los requisitos, incluyendo los requisitos no funcionales. Esto contribuye a una arquitectura sólida y

estable que se convierte en un plano para la implementación. Los modelos desarrollados en esta etapa son más formales y específicos de una implementación. Durante esta fase son desarrollados el documento de arquitectura, diagramas de clases, diagramas de entidad relación, diagrama de despliegue entre otros. Detallar que las actividades subrayadas en color rojo son actividades particulares del departamento que se realizan en esta fase.

**Fase Implementación:** en la implementación a partir de los resultados del análisis y diseño se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares. Detallar que las actividades subrayadas en color rojo son actividades particulares del departamento que se realizan en esta fase.

**Fase Pruebas internas:** durante esta fase el laboratorio de calidad del centro verifica el resultado de la implementación probando según sea necesaria cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas.

**Fase Pruebas de liberación:** pruebas diseñadas e implementada por el Laboratorio Industrial de Pruebas de Software a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.

**Fase de Despliegue:** durante esta fase se procede a la entrega de la solución, así como a la instalación, configuración, prueba y puesta en marcha del software en el entorno real del cliente. También deben realiza en este período la capacitación y acompañamiento a clientes para asegurar que adquieran los conocimientos necesarios en la manipulación del software.

2.2.7.1. Actividades de la fase Estudio preliminar

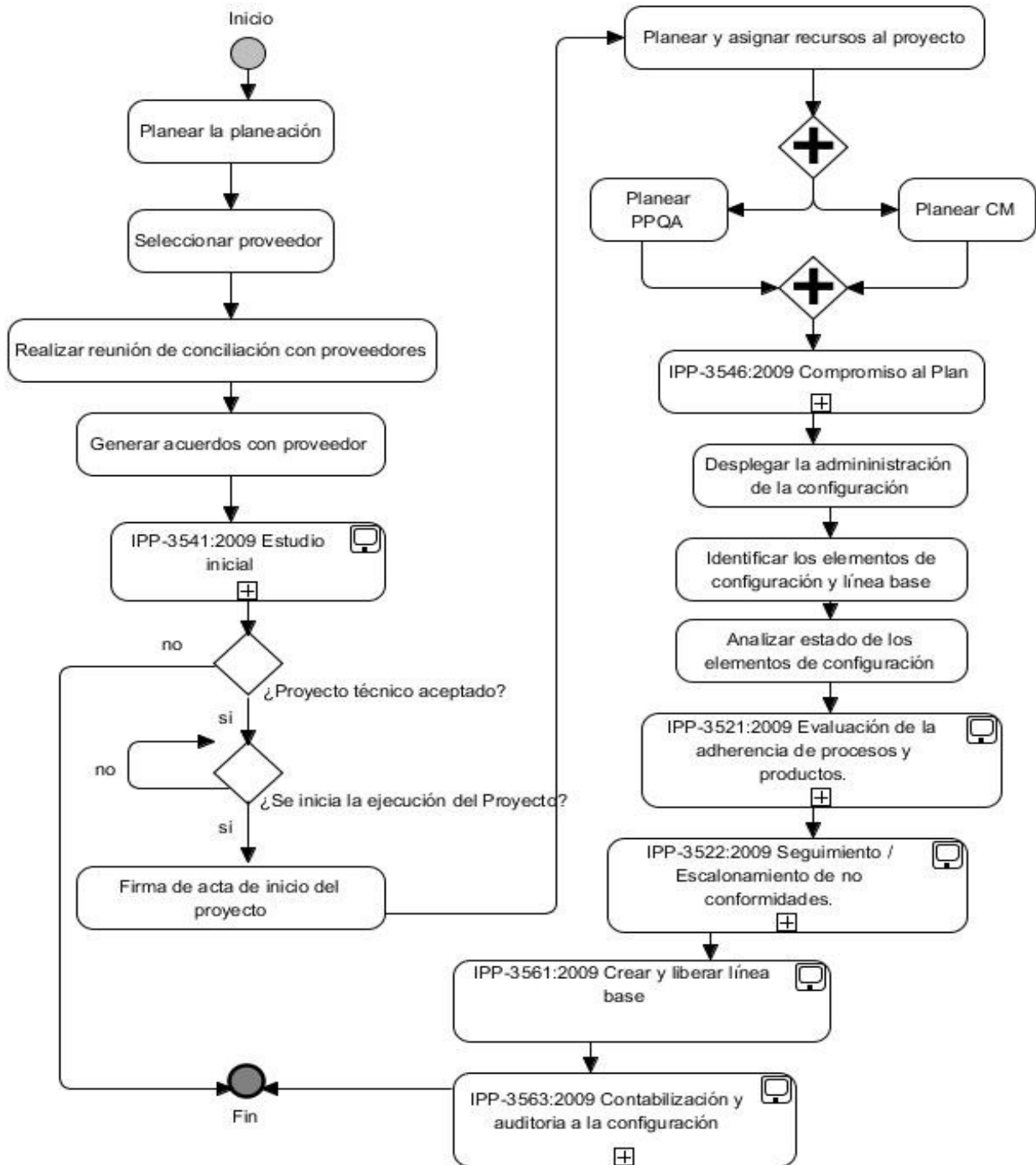


Figura 13: Actividades de la fase Estudio Preliminar.



### 2.2.7.1.1. Descripción de las actividades de la fase Estudio preliminar

#### **Actividad 1: Planear la Planeación**

Descripción: se define el calendario para las principales actividades del proceso de planeación del proyecto. Se definen los responsables para cada actividad y se le asigna un tiempo de realización.

Artefactos: WBS y Calendario (inicial, con las actividades de planeación).

Roles que ejecutan: Jefe de proyecto.

#### **Actividad 2: Seleccionar proveedor**

Descripción: se revisan las Propuestas de los proveedores. Se registran los posibles proveedores en la plantilla Evaluación y Selección de proveedores teniendo en cuenta los criterios que aparecen en esta, se evalúan cada uno de los proveedores. Basados en los criterios anteriores se selecciona al proveedor más adecuado.

Artefactos: Evaluación y selección de proveedores, Catálogo de proveedores (actualizado).

Roles que ejecutan: Jefe de proyecto, Alta gerencia.

#### **Actividad 3: Realizar reunión de conciliación con proveedores**

Descripción: se realiza la reunión de conciliación, en la reunión se negocian los intereses de ambas partes, se llega a un acuerdo y se elabora el cronograma de trabajo. Se llega a las bases del acuerdo que se pretende establecer.

Artefactos: Minuta de reunión.

Roles que ejecutan: Jefe de proyecto, Experto funcional, Proveedor, Analista.

#### **Actividad 4: Generar acuerdo con proveedores**

Descripción: una vez estudiados los requisitos y actualizado el WBS y calendario se llega a un acuerdo. Los acuerdos son aprobados y firmados por las partes. Actualizar en la plantilla Selección de proveedores, ponerle un vínculo al acuerdo que se estableció con el proveedor.

Artefactos: Acuerdo con el proveedor (firmado), WBS y calendario (aprobado), Selección de proveedores (actualizado con el vínculo al acuerdo), Plan de desarrollo (actualizado).

Roles que ejecutan: Jefe de proyecto, Proveedor.

#### **Actividad 5: Estudio inicial**

Descripción: ver proceso IPP-3541:2009 Estudio preliminar del Libro de proceso para la Planeación del proyecto. (Documento del programa de mejoras, 2009)

#### **Actividad 6: Firma de acta de inicio**

Descripción: se firma el Acta de inicio por ambas partes.

Artefactos: Acta de inicio (firmada).

Roles que ejecutan: Jefe de proyecto, Cliente, Alta gerencia.

### **Actividad 7: Planear y asignar recursos al proyecto.**

Descripción: se elabora el Plan de recursos humanos y el Plan de recursos materiales. Se solicitan a la alta gerencia los recursos planificados. Se asignan los recursos al proyecto.

Artefactos: WBS y Calendario (actualizado), Plan de desarrollo de software (Plan de recursos humanos actualizado), Planes y Registro de monitoreo (Plan de recursos materiales actualizado).

Roles que ejecutan: Jefe de proyecto, Alta gerencia.

### **Actividad 8: Planear PPQA.**

Descripción: se define la cantidad de revisiones que se van a hacer y en qué momento.

Artefactos: Plan de PPQA (Redmine), Registro de Evaluaciones del Proyecto, Registro de Evaluaciones de PPQA, WBS y calendario (actualizado).

Roles que ejecutan: Jefe de proyecto, Especialista de la calidad.

### **Actividad 9: Planear CM.**

Descripción: se elabora el Plan de Administración de la configuración donde establecen elementos importantes de la configuración como la identificación de los ECS, las líneas bases a establecer, las auditorías a realizar entre otros elementos.

Artefactos: Plan de Desarrollo de Software (Plan de Administración de la configuración).

Roles que ejecutan: Administrador de la configuración, Jefe de proyecto.

### **Actividad 10: Compromiso al plan.**

Descripción: ver proceso IPP-3546:2009 Compromiso al plan del Libro de proceso para la Planeación del proyecto. (Documento del programa de mejoras, 2009)

### **Actividad 11: Desplegar la administración de la configuración.**

Descripción: se crea el repositorio del proyecto. Se generar acceso al repositorio, se identifican los niveles de seguridad y se define los mecanismos de respaldo y recuperación.

Artefactos: Repositorio del Proyecto Accesos establecidos.

Roles que ejecutan: Administrador de la configuración, Jefe de proyecto.

### **Actividad 12: Identificar los elementos de configuración y línea base.**

Descripción: se nombran los elementos de configuración del proyecto que se van creando, siempre

acorde a lo planteado en el documento estándar de configuración.

Artefactos: Elementos de Configuración del proyecto.

Roles que ejecutan: Administrador de la configuración, Jefe de proyecto.

### **Actividad 13: Analizar estado de los elementos de configuración.**

Descripción: se hacer una actualización del repositorio, guiarse por lo descrito en el manual de la herramienta, se aplica cambios a un elemento de configuración, se revisa las actualizaciones e integridad de un elemento de configuración, publicar el elemento de configuración, guiarse por lo descrito en el manual de la herramienta.

Artefactos: Elemento de configuración (actualizado), Repositorio (actualizado).

Roles que ejecutan: Equipo de trabajo.

### **Actividad 14: Realizar procesos horizontales.**

Descripción: ver epígrafe procesos horizontales en el ciclo de vida del desarrollo del software.

#### **2.2.7.2. Actividades fase Modelamiento del negocio.**

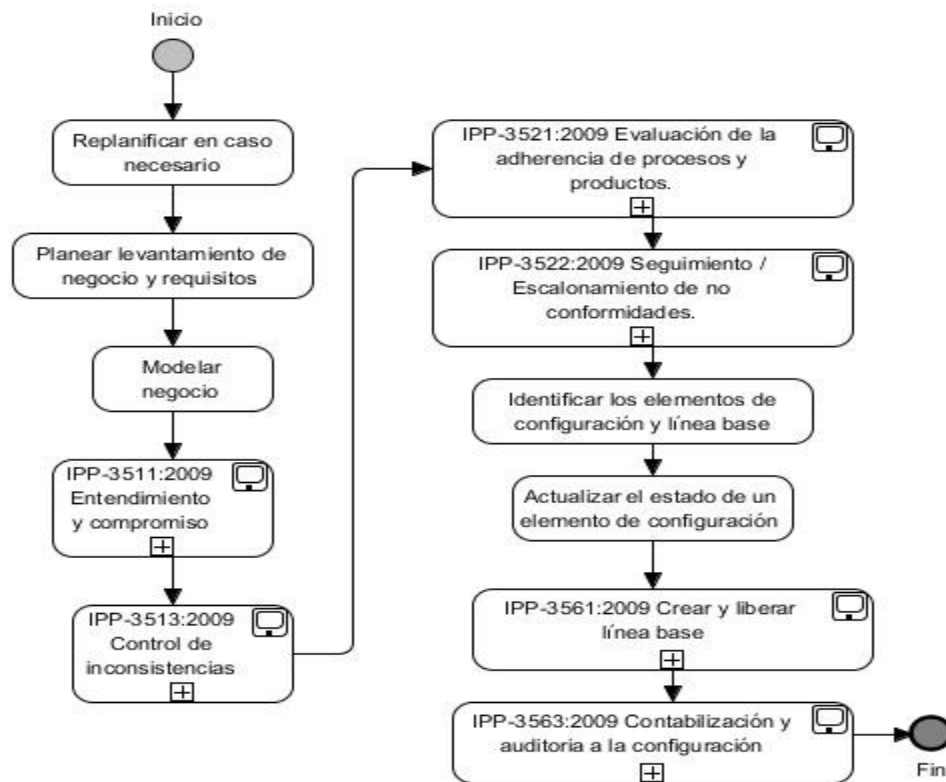


Figura 14: Actividades de la fase modelamiento del negocio.

### 2.2.7.2.1. Descripción de las actividades de la fase modelamiento del negocio.

#### **Actividad 1: Replanificar en caso que sea necesario.**

Descripción: se replanifica la fase en caso que sea necesario.

Artefactos: WBS y Calendario (actualizado). Planes de desarrollo de software (actualizados).

Roles que ejecutan: Jefe de proyecto.

#### **Actividad 2: Planear levantamiento de negocio y de requisito.**

Descripción: se actualiza el WBS y Calendario especificando las actividades relacionadas con el negocio y los requisitos.

Artefactos: WBS y Calendario (actualizado con las actividades de negocio y requisitos).

Roles que ejecutan: Jefe de proyecto.

#### **Actividad 3: Modelar negocio.**

Descripción: se recibe la documentación del negocio. Se identifican los procesos de negocio. Se registran los procesos identificados en el Mapa de procesos. Se revisan los mapas de procesos identificados previamente e identificar procesos de los procesos. Se realiza un estado del arte para ver lo que se ha hecho hasta el momento y lo que falta por hacer. Se desarrollan talleres, encuentros y entrevistas a los proveedores de requisitos (cliente) para obtener la información necesaria para completar la plantilla de procesos de negocio y aclarar las dudas existentes. Se completa la plantilla de procesos de negocio. Se hace una elaboración del modelo de procesos de negocios utilizando la herramienta VP, a partir de lo obtenido con el cliente.

Artefactos: Glosario de términos (actualizado), Modelo de Procesos de Negocio con BPMN, Mapa de procesos de Negocio. Descripción de procesos de negocio, Modelo Conceptual, Reglas del negocio

Roles que ejecutan: Analista, Proveedor de requisitos (cliente).

#### **Actividad 4: Entendimiento y Compromiso.**

Descripción: ver proceso IPP-3511:2009 Entendimiento y Compromiso del Libro de proceso para la Administración de Requisitos. (Documento del programa de mejoras, 2009)

#### **Actividad 5: Control de inconsistencias.**

Descripción: ver proceso IPP-3513:2009 Control de inconsistencias del Libro de proceso para la Administración de Requisitos. (Documento del programa de mejoras, 2009)

#### **Actividad 6: Evaluación de la Adherencia de procesos y productos.**

Descripción: ver proceso IPP-3521:2009 Evaluación de la Adherencia de procesos y productos del Libro

de proceso para PPQA. (Documento del programa de mejoras, 2009)

### **Actividad 7: Seguimiento/Escalamiento de las no conformidades.**

Descripción: ver proceso IPP-3522:2009 Seguimiento/Escalamiento de las no conformidades del Libro de proceso para PPQA. (Documento del programa de mejoras, 2009)

### **Actividad 8: Identificar los elementos de configuración y línea base.**

Descripción: se identifican los elementos de configuración del proyecto basado en criterios predeterminados. Se asigna el identificador único y se especifica las características de los elementos de configuración.

Artefactos: Elementos de configuración del proyecto.

Roles que ejecutan: Administrador de la configuración, Jefe de proyecto.

### **Actividad 9: Analizar estado de los elementos de configuración.**

Descripción: se hace una actualización del repositorio, guiarse por lo descrito en el manual de la herramienta, se aplica cambios a un elemento de configuración, se revisa las actualizaciones e integridad de un elemento de configuración, publicar el elemento de configuración, guiarse por lo descrito en el manual de la herramienta.

Artefactos: Elemento de configuración (actualizado), Repositorio (actualizado).

Roles que ejecutan: Equipo de trabajo.

### **Actividad 10: Crear y liberar línea base.**

Descripción: ver proceso IPP-3561:2009 Creación y liberación de líneas bases del Libro de proceso para la administración de la configuración. (Documento del programa de mejoras, 2009)

### **Actividad 11: Contabilización y auditoría a la configuración.**

Descripción: ver proceso IPP-3563:2009 Contabilización y auditoría a la configuración del Libro de proceso para la administración de la configuración. (Documento del programa de mejoras, 2009)

2.2.7.3. Actividades de la fase de Requisitos

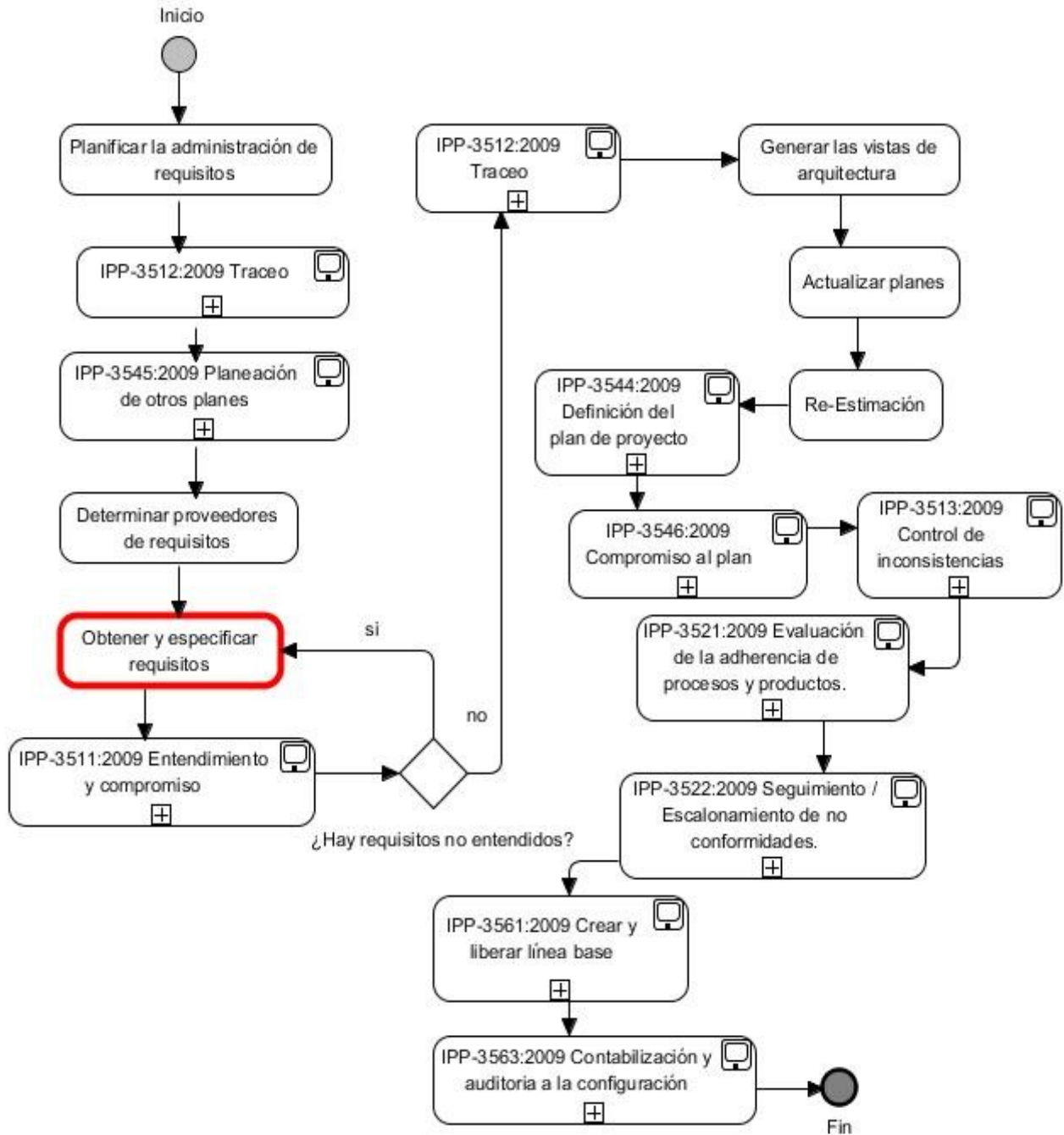


Figura 15: Actividades de la fase de Requisitos.

### 2.2.7.3.1. Descripción de las actividades de la fase de Requisitos.

#### **Actividad 1: Planear administración de requisitos.**

Descripción: se actualiza el glosario de términos. Se actualiza las actividades concernientes a la administración de requisitos en el WBS y Calendario y se preparan las herramientas de trabajo.

Artefactos: WBS y Calendario (actualizado), Glosario de términos, Herramientas (instaladas y configuradas).

Roles que ejecutan: Jefe de proyecto, Planificador, Analista.

#### **Actividad 2: Traceo.**

Descripción: ver proceso IPP-3512:2009 Traceo del Libro de proceso para la Administración de Requisitos. (Documento del programa de mejoras, 2009)

#### **Actividad 3: Planeación de otros planes.**

Descripción: ver proceso IPP-3545:2009 Planeación de otros planes del Libro de proceso para la Planeación del proyecto. (Documento del programa de mejoras, 2009)

#### **Actividad 4: Determinar proveedores de requisitos.**

Descripción: se validan los proveedores contra los criterios de validación. Se seleccionan los proveedores válidos y se crea una lista con estos proveedores. Se selecciona el proveedor responsable y se identifica el proveedor responsable en la lista de proveedores válidos. Se asegura que el cliente está de acuerdo con la selección.

Artefactos: Lista de proveedores válidos.

Roles que ejecutan: Proveedor de requisitos, Analista, Jefe de proyecto.

#### **Actividad 5: Obtener y especificar requisitos.**

Descripción: se refinan los requisitos identificados. Se describen los requisitos funcionales y no funcionales. Se evalúa los requisitos según su complejidad y prioridad y actualizar la herramienta de gestión de requisitos si el proyecto lo requiere. Se describe las salidas del sistema si el proyecto lo requiere. Se elabora el prototipo de interfaz de usuario. Se analizan los requisitos del cliente. Se identificar ambigüedades, inconsistencias, requisitos comunes y escenarios arquitectónicos requeridos para dar soporte a los requisitos funcionales. Se describen las salidas del sistema si el proyecto lo requiere. Se elaborar prototipo de interfaz de usuario utilizando la herramienta Axure. Se define la arquitectura de información si al proyecto le da valor agregado.

Artefactos: Especificación de requisitos de software, Descripción de requisitos, Documento de salidas del

sistema (actualizado), Arquitectura de información.

Roles que ejecutan: Analista.

### **Actividad 6: Entendimiento y Compromiso.**

Descripción: ver proceso IPP-3511:2009 Entendimiento y Compromiso del Libro de proceso para la Administración de Requisitos. (Documento del programa de mejoras, 2009)

### **Actividad 7: Traceo.**

Descripción: ver proceso IPP-3512:2009 Traceo del Libro de proceso para la Administración de Requisitos. (Documento del programa de mejoras, 2009)

### **Actividad 8: Generar las vistas de arquitectura.**

Descripción: una vez establecidos los requisitos funcionales de y las características técnicas de la aplicación, se generan las vistas de arquitectura.

Artefactos: Especificación de requisitos funcionales, Descripción de requisitos funcionales, Modelo conceptual (refinado), Salidas del sistema, Diseño de casos de prueba.

Roles que ejecutan: Analista.

### **Actividad 9: Actualizar planes.**

Descripción: ver proceso IPP-3545:2009 Planear otros planes del Libro de proceso para la Planeación del proyecto. (Documento del programa de mejoras, 2009)

### **Actividad 10: Re estimación.**

Descripción: se realiza una segunda estimación del proyecto con la información de los requisitos a realizar. Utilizando la herramienta de estimación se realiza una segunda estimación del tiempo de desarrollo, costo y recursos humanos del proyecto.

Artefactos: Reporte de re-estimación del proyecto.

Roles que ejecutan: Jefe de proyecto.

### **Actividad 11: Definición del plan de proyecto.**

Descripción: ver proceso IPP-3544:2009 Definición del plan de proyecto del Libro de proceso para la Planeación del proyecto. (Documento del programa de mejoras, 2009)

### **Actividad 12: Compromiso al plan.**

Descripción: ver proceso IPP-3546:2009 Compromiso al plan de proyecto del Libro de proceso para la Planeación del proyecto. (Documento del programa de mejoras, 2009)

### **Actividad 13: Realizar procesos horizontales.**



Descripción: ver epígrafe procesos horizontales en el ciclo de vida del desarrollo del software.

**2.2.7.4. Actividades de la fase Análisis y Diseño**

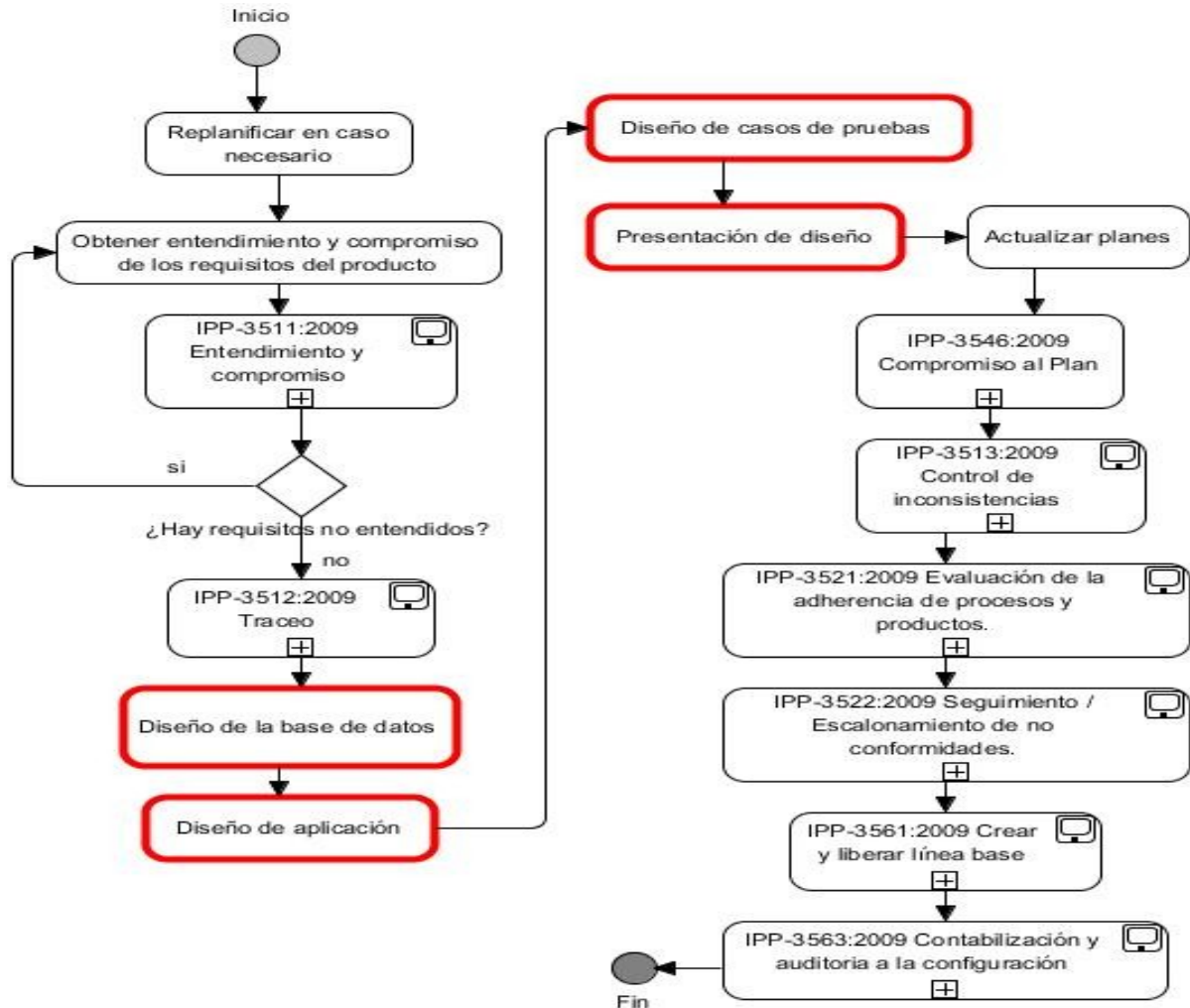


Figura 16: Actividades de la fase de Análisis y diseño.

**2.2.7.4.1. Descripción de las Actividades de la fase de Análisis y diseño.**

**Actividad 1: Replanificar en caso que sea necesario.**

Descripción: se replanifica la fase en caso que sea necesario.

Artefactos: WBS y Calendario (actualizado).Planes de desarrollo de software (actualizados).

Roles que ejecutan: Jefe de proyecto.

**Actividad 2: Obtener entendimiento y compromiso de los requisitos del producto.**

Descripción: se obtiene un entendimiento de los requisitos. Se define los artefactos a obtener en cada fase del proyecto, asociado a cada requisito.

Artefactos: Descripción de requisitos del producto: Diagrama de clases, Diagrama entidad relación, Diagrama de interacción, Diagrama de despliegue, Ejecutables, Documentos de arquitectura, Código fuente, Casos de Prueba.

Roles que ejecutan: Jefe de proyecto, Analista, Arquitecto de software.

### **Actividad 3: Entendimiento y Compromiso.**

Descripción: ver proceso IPP-3511:2009 Entendimiento y Compromiso del Libro de proceso para la Administración de Requisitos. (Documento del programa de mejoras, 2009)

### **Actividad 4: Traceo.**

Descripción: ver proceso IPP-3512:2009 Traceo del Libro de proceso para la Administración de Requisitos. (Documento del programa de mejoras, 2009)

### **Actividad 5: Diseño de la base de datos.**

Descripción: se crea el Script de la BD a partir del modelo de datos y el diccionario de datos. Se diseña la BD utilizando la herramienta Erwin XML. Se muestra a los funcionales el resultado y los mismos deciden si está correcto el modelo o si necesita algunos cambios. Se realiza un reporte en HTML de la base de datos.

Artefactos: Script de BD, Diseño de la BD, Reporte de HTML.

Roles que ejecutan: Jefe de proyecto, Arquitecto de datos, Analista, Funcionales.

### **Actividad 6: Diseño de la aplicación.**

Descripción se realiza el diagrama de clases y el modelado. Se realiza el diagrama de flujo para cada uno de los métodos. Se refina la arquitectura.

Artefactos: Modelo de diseño (donde se incluyen el diagrama de secuencia orientado a actividades), Modelo de descripción del modelo de datos (incluye el Modelo de datos), Modelo de integración, Arquitectura de software.

Roles que ejecutan: Analista, Diseñador del sistema, Arquitecto de software.

### **Actividad 7: Diseño de casos de prueba.**

Descripción: a partir de la Especificación de procesos del negocio, prototipo de IU, reporte HTML y descripción de la arquitectura se realiza el diseño de caso de pruebas.

Artefactos: Diseño de Casos de Prueba, Fichero de pruebas para plugin Symfony.

Roles que ejecutan: Analista, Funcionales, Jefe de proyecto.

**Actividad 8: Presentación de diseño.**

Descripción: se muestra a los funcionales el diseño de la aplicación y los mismos deciden si está correcto el diseño o necesita algún cambio.

Artefactos: Aprobación interna del diseño.

Roles que ejecutan: Analista, Funcionales, Jefe de proyecto.

**Actividad 9: Actualizar planes.**

Descripción: ver proceso IPP-3545:2009 Planear otros planes del Libro de proceso para la Planeación del proyecto. (Documento del programa de mejoras, 2009)

**Actividad 10: Compromiso al plan.**

Descripción: ver proceso IPP-3546:2009 Compromiso al plan del Libro de proceso para la Planeación del proyecto. (Documento del programa de mejoras, 2009)

**Actividad 11: Realizar procesos horizontales.**

Descripción: ver epígrafe procesos horizontales en el ciclo de vida del desarrollo del software.

**2.2.7.5. Actividades de la fase Implementación.**

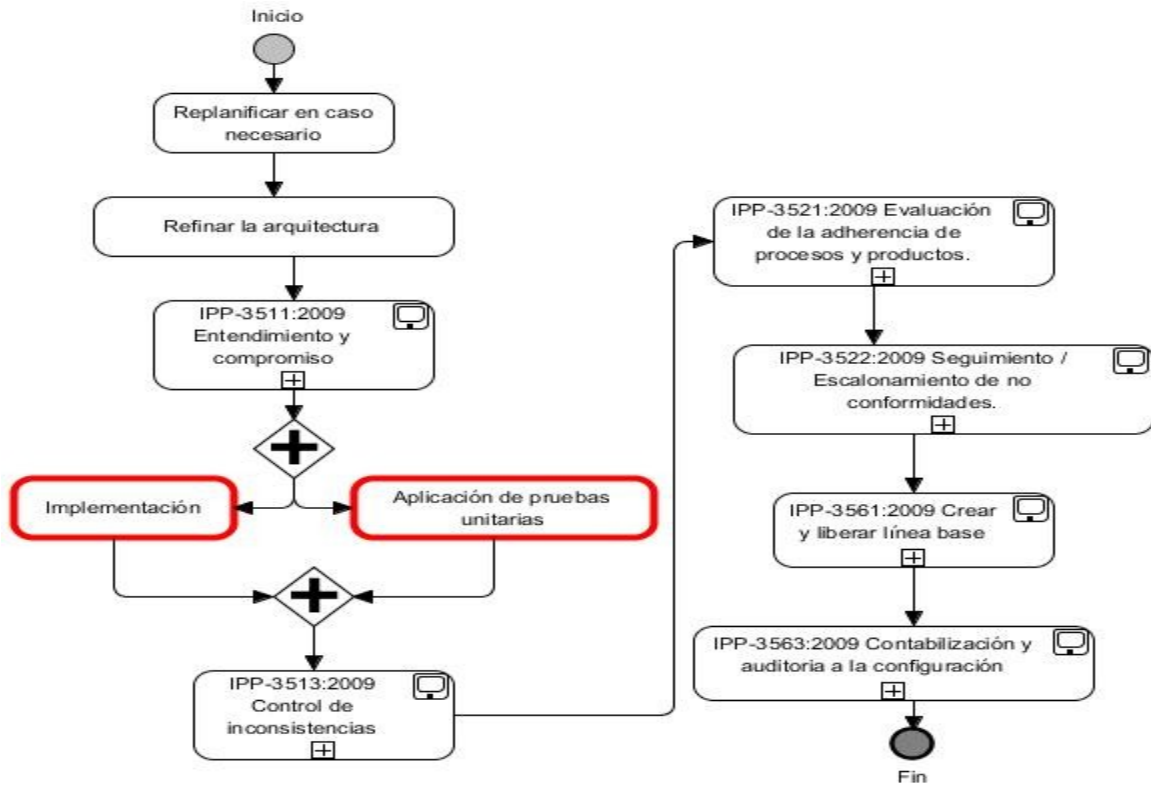


Figura 17: Actividades de la fase de Implementación.

**2.2.7.5.1. Descripción de las actividades de la fase Implementación.**

**Actividad 1: Replanificar en caso que sea necesario.**

Descripción: se replanifica la fase en caso que sea necesario.

Artefactos: WBS y Calendario (actualizado).Planes de desarrollo de software (actualizados).

Roles que ejecutan: Jefe de proyecto.

**Actividad 2: Refinar la arquitectura.**

Descripción: se refina la arquitectura.

Artefactos: Manual de usuario.

Roles que ejecutan: Analista, Arquitecto de software.

**Actividad 3: Entendimiento y Compromiso.**

Descripción: Ver proceso IPP-3511:2009 Entendimiento y Compromiso del Libro de proceso para la Administración de Requisitos. (Documento del programa de mejoras, 2009)

### Actividad 4: Implementación y Aplicación de pruebas unitarias.

Descripción: Se implementa toda la lógica de negocio diseñada. Se le aplican pruebas a la implementación para encontrar posibles errores en el código.

Artefactos: Código fuente, Resultados de las prueba unitarias.

Roles que ejecutan: Programador.

**Actividad 5: Realizar procesos horizontales.** Descripción: ver epígrafe procesos horizontales en el ciclo de vida del desarrollo del software.

#### 2.2.7.6. Actividades de la fase Pruebas internas.

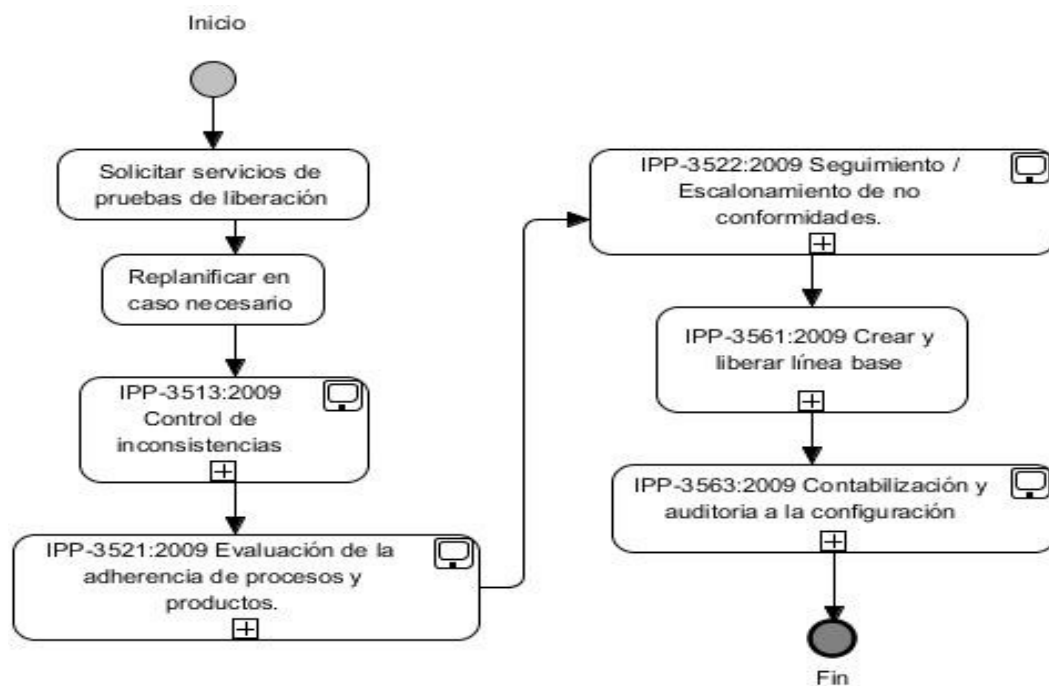


Figura 18: Actividades de la fase Pruebas internas.

#### 2.2.7.6.1. Descripción de las actividades de la fase Pruebas internas.

##### Actividad 1: Solicitar servicios de Pruebas de liberación interna.

Descripción: se solicita los servicios de pruebas internas al asesor de calidad del centro.

Artefactos: Solicitud de servicios de pruebas internas.

Roles que ejecutan: Jefe de proyecto.

##### Actividad 2: Replanificar en caso que sea necesario.

Descripción: se replanifica la fase en caso que sea necesario.

Artefactos: WBS y Calendario (actualizado). Planes de desarrollo de software (actualizados).

Roles que ejecutan: Jefe de proyecto.

**Actividad 3: Realizar procesos horizontales.** Descripción: ver epígrafe procesos horizontales en el ciclo de vida del desarrollo del software.

### 2.2.7.7. Actividades de la fase Pruebas de liberación.

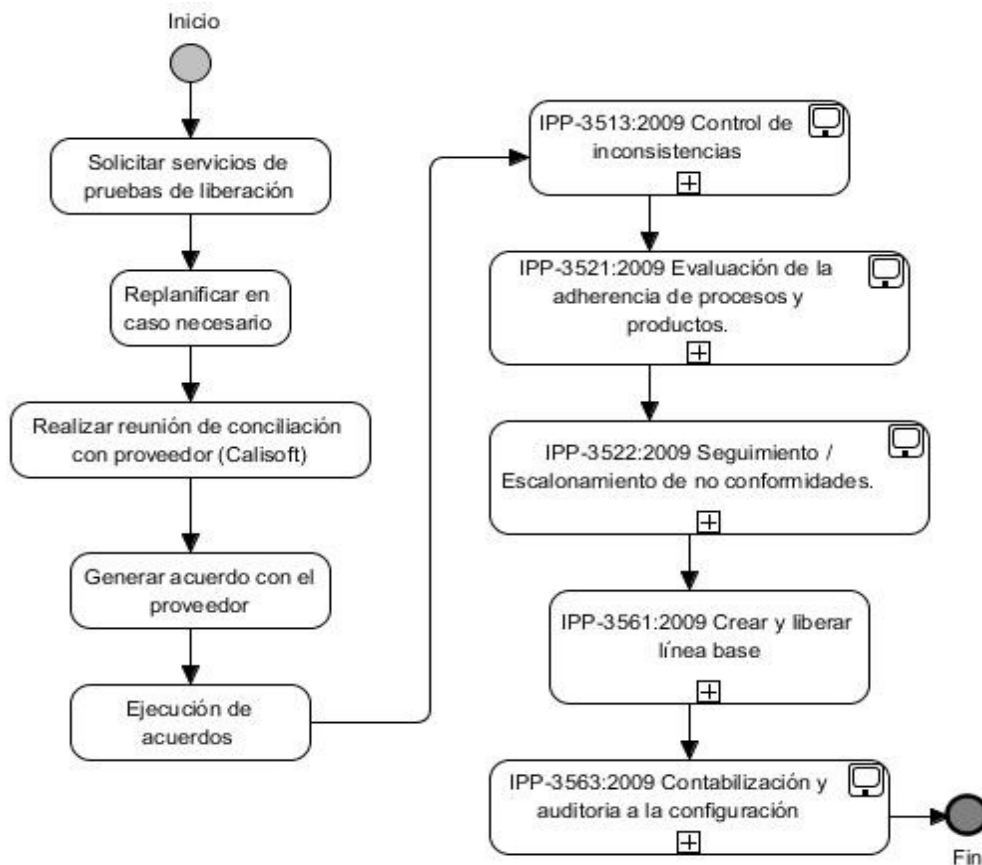


Figura 19: Actividades de la fase Pruebas de liberación.

#### 2.2.7.7.1. Descripción de las actividades de la fase Pruebas de liberación.

##### Actividad 1: Solicitar servicios de Pruebas de liberación.

Descripción: se solicita los servicios de pruebas de liberación a Calisoft.

Artefactos: Plantillas de solicitud de pruebas de liberación, plantilla de artefactos para pruebas de

liberación.

Roles que ejecutan: Jefe de proyecto.

### **Actividad 2: Replanificar en caso que sea necesario.**

Descripción: se replanifica la fase en caso que sea necesario.

Artefactos: WBS y Calendario (actualizado).Planes de desarrollo de software (actualizados).

Roles que ejecutan: Jefe de proyecto.

### **Actividad 3: Realizar reunión de conciliación con proveedores.**

Descripción: se realiza la reunión de conciliación en la que participan el proveedor (Calisoft), el Jefe de proyecto, el Analista y el Experto Funcional. En la reunión se negocian los intereses de ambas partes, se llega a un acuerdo y se elabora el cronograma de trabajo. Se llega a las bases del acuerdo que se pretende establecer.

Artefactos: Minuta de reunión.

Roles que ejecutan: Jefe de proyecto, Experto funcional, Proveedor, Analista.

### **Actividad 4: Generar acuerdo con proveedores.**

Descripción: el Jefe de proyecto y el Proveedor, una vez estudiados los requisitos y actualizado el WBS y Calendario, se llega a un acuerdo. Si el proveedor cuenta con una plantilla para este fin, el acuerdo es recogido en la Plantilla del proveedor, en caso contrario se documenta en la plantilla: Acuerdo con el Proveedor. Los acuerdos son aprobados y firmados por las partes. Actualizar en la plantilla Selección de Proveedores, ponerle un vínculo al acuerdo que se estableció con el proveedor.

Artefactos: Acuerdo con el Proveedor (Firmado), WBS y Calendario (Aprobado), Selección de proveedores (actualizado con el vínculo al acuerdo), Plan de Desarrollo (actualizado).

Roles que ejecutan: Jefe de proyecto, Proveedor.

### **Actividad 5: Realizar procesos horizontales.**

Descripción: ver epígrafe procesos horizontales en el ciclo de vida del desarrollo del software.

### 2.2.7.8. Actividades de la fase de Despliegue.

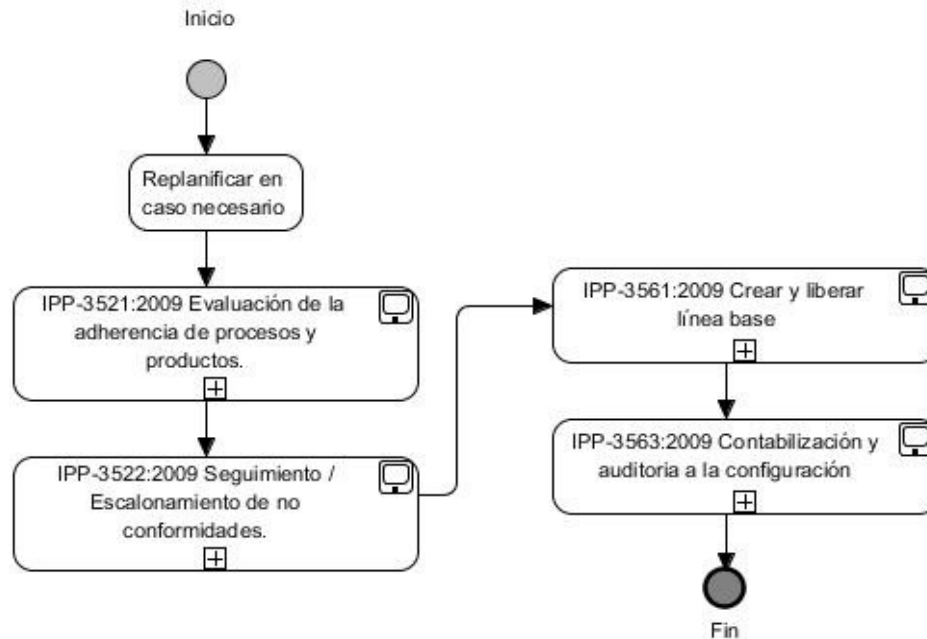


Figura 20: Actividades de la fase de Despliegue.

#### 2.2.7.8.1. Descripción de las actividades de la fase Despliegue.

##### **Actividad 1: Replanificar en caso que sea necesario.**

Descripción: se replanifica la fase en caso que sea necesario.

Artefactos: WBS y Calendario (actualizado). Planes de desarrollo de software (actualizados).

Roles que ejecutan: Jefe de proyecto.

##### **Actividad 2: Realizar procesos horizontales.**

Descripción: ver epígrafe procesos horizontales en el ciclo de vida del desarrollo del software.

### 2.2.7.9. Actividades periódicas

#### **Actividad: Monitoreo y control de proyecto**

Descripción: se revisan los valores reales de los indicadores del proyecto (tareas, recursos materiales y humanos, riesgos, compromisos, datos, participación de los involucrados relevantes y solicitudes de cambio) y se comparan con valores planeados. A partir de estas revisiones se identifican las desviaciones del proyecto. Para más información ver IPP-3530\_2009 Libro de Proceso Monitoreo y Control de Proyecto. (Documento del programa de mejoras, 2009)



**2.2.7.10. Actividades eventuales**

**Actividad: Medición y Análisis**

Descripción: Ver el proceso IPP 3553:2009 Recolectar y Almacenar los datos de la medición y el proceso IPP 3554:2009 Analizar datos de la medición y comunicar los resultados del Libro de proceso para la medición y análisis. (Documentos del programa de mejoras)

**Actividad: Solicitud de Cambios**

Descripción: ver el proceso IPP 3562:2009 control de cambios del Libro de proceso para la administración de la configuración. Estas actividades eventuales de solicitudes de cambio se realizan independientemente de la fase en la que se encuentre el proyecto, solo depende de que aparezca un cambio. (Documento del programa de mejoras, 2009)

**2.2.7.11. Procesos horizontales en el ciclo de vida de desarrollo.**

Los procesos Evaluación de la adherencia de procesos y productos, Seguimiento/Escalonamiento de las no conformidades, Crear y liberar línea base y Contabilización y auditoría a la configuración se realizan a lo largo de todo el ciclo de vida de desarrollo. El proceso Control de inconsistencia se realiza desde la fase modelamiento de negocio hasta la fase de pruebas de liberación. El orden en que se realizan cada una de ellas se detalla en el diagrama de la fase correspondiente. Ver figura 21.

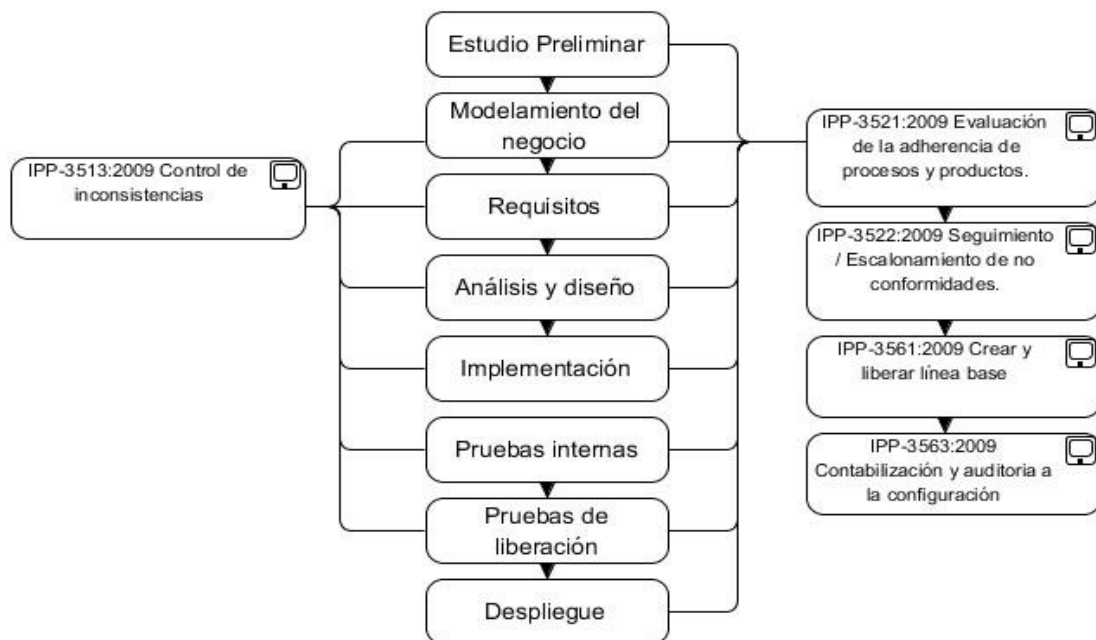


Figura 21: Relación de las actividades horizontales en el ciclo de desarrollo.

**Proceso: IPP-3513:2009 Control de inconsistencias.**

Descripción: se realizan revisiones con el fin de identificar las inconsistencias entre los planes de proyecto, los productos de trabajo, los requisitos y sus cambios. Debe identificarse la fuente de la inconsistencia, los cambios a realizar en los productos de trabajo y deben iniciarse las acciones correctivas pertinentes. Para más información ver IPP-3513:2009 Control de inconsistencias del Libro de proceso para la Administración de Requisitos. (Documento del programa de mejoras, 2009)

**Proceso: IPP-3521:2009 Evaluación de la Adherencia de procesos y productos.**

Descripción: se realizan evaluaciones de adherencia para determinar el grado de conformidad de los procesos, productos de trabajo y servicios con las especificaciones, estándares, normas u otras reglas aplicables. Para más información ver proceso IPP-3521:2009 Evaluación de la Adherencia de procesos y productos del Libro de proceso para PPQA. (Documento del programa de mejoras, 2009)

**Proceso: IPP-3522:2009 Seguimiento/Escalamiento de las no conformidades.**

Descripción: se identifican no conformidades, estas han de resolverse en un tiempo establecido, se comunican y son escaladas a otros niveles en caso de ser necesario. Para más información ver proceso IPP-3522:2009 Seguimiento/Escalamiento de las no conformidades del Libro de proceso para PPQA. (Documento del programa de mejoras, 2009)

**Proceso: IPP-3561:2009 Crear y liberar línea base.**

Descripción: se crea y libera la línea base para uso interno y para entregar al cliente. Una línea base es un conjunto de productos de trabajo formalmente revisados y aprobados que sirve de base para desarrollos posteriores o entregas. Para más información ver proceso IPP-3561:2009 Creación y liberación de líneas bases del Libro de proceso para la administración de la configuración. (Documento del programa de mejoras, 2009)

**Proceso: IPP-3563:2009 Contabilización y auditoría a la configuración.**

Descripción: se realizan auditorías a la configuración para mantener la integridad de las líneas bases de la configuración. Para más información ver proceso IPP-3563:2009 Contabilización y auditoría a la configuración del Libro de proceso para la administración de la configuración. (Documento del programa de mejoras, 2009)

### 2.3. Propuesta de avances para alcanzar el nivel tres de CMMI.

La visión continua de los diferentes centros productivos de la universidad como el CEIGE de realizar procesos eficaces, establecer objetivos y prioridades en la mejora de procesos, y así obtener una mayor calidad y evaluación de sus procesos actuales, los lleva a optar por un nivel de madurez superior al obtenido. Partiendo de lo mencionado anteriormente se propone como sugerencias dar un paso más, estudiando el impacto y las mejoras que puede traer incorporar al modelo de desarrollo algunas de las buenas prácticas de CMMI nivel tres para abrir camino a la certificación de dicho nivel.

Para alinear el actual proceso de desarrollo con CMMI nivel tres, se debe incorporar once nuevas áreas de procesos, asumir todas estas responsabilidades de manera rápida no es nada fácil porque será necesario realizar cambios en la mayoría de los procesos actuales, mejorar las actuales herramientas de trabajo, e incorporar algunas nuevas, a continuación se mencionan algunas de las prácticas que se sugieren y se pueden incorporar para preparar el camino al nivel tres:

**Verificación y Validación** que son procesos que permitirán la evaluación del producto para determinar si se satisfacen las necesidades del negocio y si se están construyendo acorde a las especificaciones. En el caso de la Verificación permitirá responder si “¿Se está construyendo el producto de la manera correcta?” y la Validación si “¿Se está construyendo el producto correcto?” quiere decir que la verificación implica comprobar que el software está de acuerdo con su especificación, comprobarse que satisface sus requisitos funcionales y no funcionales y la validación, sin embargo, asegurara que el sistema satisface las expectativas del cliente. Haciendo más allá de la comprobación de que el sistema satisface su especificación para demostrar que el software hace lo que el cliente espera que haga, ya que las especificaciones del sistema no siempre reflejan los deseos o necesidades reales de los usuarios y los propietarios del sistema.

**Gestión de riesgos** para la prevención y anticipación a los riesgos. Además de que esta área es una evolución de las prácticas básicas de manejo de riesgo incluidas en PP y PMC pertenecientes al nivel dos, y en esta área se plantea una perspectiva sistemática para planear, anticipar y mitigar riesgos minimizando su impacto en el proyecto.

Además de reforzar los objetivos conseguidos con el nivel dos, el aplicar e implementar estas prácticas del nivel tres permitirá:

- ✓ Disminuir los errores y defectos en los entregables.
- ✓ Prevenir el efecto de los riesgos más críticos identificados en el proyecto.

- ✓ Disponer de una visión integrada de las tareas y participantes en el proyecto.
- ✓ Una mejora en la planificación debido a que hará necesario realizar una planificación cuidadosa para obtener el máximo provecho de las inspecciones y pruebas, y control de los procesos.

Por lo tanto el hacer uso de estas buenas prácticas de Verificación, Validación, Gestión de riesgos del nivel tres de CMMI no solo abrirá las puertas a la certificación sino que permitirá comprobar que el producto está hecho para un propósito y que el producto es lo suficientemente bueno para su uso previsto.

### **2.4. Conclusiones parciales**

En este capítulo se da cumplimiento al objetivo relacionado con la elaboración del modelo de desarrollo para el departamento de Soluciones Aduanales del centro CEIGE. Se define la estructura, las características y los principios con los que debe cumplir el modelo propuesto para su buen funcionamiento. El modelo propuesto permite dirigir y gestionar el proceso de desarrollo de software desde la dimensión de desarrollo realizando una descripción de los procesos asociados a las mismas. En esta dimensión de desarrollo se describe el proceso de construcción del producto por fases y se hace uso de las buenas prácticas de CMMI nivel dos obteniendo una mejora continua en cada uno de los procesos. En la especificación de cada proceso se describe brevemente el flujo de actividades a desarrollar en cada una de las fases del desarrollo, los roles involucrados y los artefactos generados en cada actividad. Quedando constituida de esta forma una guía práctica para la implantación de las tareas en el ciclo de vida del proyecto.

### Capítulo 3. Validación de la propuesta de solución

#### 3.1. Introducción

En el presente capítulo para la validación y aceptación del modelo propuesto en el Capítulo 2, se tomó como herramienta el uso del Criterio de un panel de expertos y el empleo de técnicas propuestas en el Método Delphi, y se realiza una descripción de cómo fue ejecutado el método y los resultados que fueron obtenidos.

#### 3.2. Método de evaluación de experto

##### Método Delphi

El Método Delphi, considerado como una de las técnicas subjetivas de pronosticación más confiables, conocido también como Delfos, nombre cuyo origen proviene del oráculo de la antigua Grecia, el método Delphi fue creado alrededor de los años 1963-1964 por la Rand Corporation, y específicamente por Olaf Helmer, Dalkey y Gordon, con el objetivo de elaborar pronósticos a largo plazo, referentes a posibles acontecimientos en varias ramas de la ciencia, la técnica y la política. Así pues el método Delphi es la utilización sistemática del juicio intuitivo de un grupo de expertos para obtener un consenso de opiniones informadas. (Gordon, Helmer, Delkey, 1963-1964)

Este método Delphi presenta tres características fundamentales que a continuación se describen:

**Anonimato:** No debe existir contacto entre los participantes, pero el encargado de la encuesta sí puede identificar a cada participante y sus respuestas.

**Iteración y realimentación controlada:** La iteración se consigue al presentar varias veces el mismo cuestionario. Como se van presentando los resultados obtenidos con los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.

**Respuesta del grupo en forma estadística:** La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.

La esencia de este método consiste en la organización de un diálogo anónimo entre los expertos consultados individualmente, mediante cuestionarios, con vistas a obtener un consenso general o, al

menos, los motivos de la discrepancia, la confrontación de las opiniones se lleva a cabo mediante una serie de interrogantes sucesivas, entre cada una de las cuales la información obtenida sufre un procesamiento estadístico - matemático.

Entre las etapas principales de realización de la evaluación de una investigación a través del método de evaluación de expertos se encuentran las siguientes:

### **3.2.1. Elaboración del objetivo.**

Formulación del objetivo de la evaluación por los expertos: Valorar el modelo propuesto para la aplicación en el departamento Soluciones Aduanales del centro CEIGE y la organización del trabajo productivo que se espera alcanzar con su aplicación.

### **3.2.2. Selección de los expertos.**

Uno de los principales aspectos a tener en cuenta para la aplicación del método es la selección del grupo de expertos a encuestar. Se considera como experto al individuo en sí, grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia.

Partiendo de la definición planteada anteriormente se toma como criterios básicos para la selección de un experto los que a continuación se plantean:

- ✓ La efectividad de la actividad profesional que realiza.
- ✓ La experiencia que posee en el tema a analizar, dado por los años de trabajo.
- ✓ Los conocimientos adquiridos a través de las diferentes formas de superación.
- ✓ El grado académico o científico que posee del tema y su participación en diversos eventos científicos.

Elegir los expertos atendiendo a las características mencionadas propicia obtener resultados con calidad, junto a otras cualidades propias de estos como pueden ser: la seriedad, la honestidad, la sinceridad, la responsabilidad y otras en este sentido, que hacen que las opiniones brindadas sean confiables y válidas para el objetivo propuesto.

Para llevar a cabo el método se realizó la selección un grupo expertos del CEIGE y de otros centros de la universidad teniendo en cuenta su disposición a participar en la encuesta y apreciación en el tema investigado, modelo de desarrollo de software. Luego de hacer esta previa selección se aplica una metodología usada en el método Delphi la cual resulta completa y sencilla para determinar la competencia

## Capítulo 3. Validación de la propuesta de solución

de los expertos. Para realizar las evaluaciones se realizó una encuesta (Ver Anexo 2) la cual fue resuelta por los posibles expertos.

La metodología establece que lo primero que el posible experto debe hacer es marcar con una **x** en una escala creciente de 0 a 10, ver Tabla 1 el valor que se corresponde con el grado de conocimiento o información que tienen sobre el tema de estudio. Con los datos se calcula el coeficiente de conocimiento **Kc** para cada uno de los expertos donde:

**Kc** = El valor que marcó el experto en la tabla por (0.1).

0	1	2	3	4	5	6	7	8	9	10

Tabla 1: Coeficiente de conocimiento.

Lo segundo es pedirles a los expertos su autoevaluación de sus niveles de argumentación o fundamentación sobre el tema de estudio, para esto debe llenar con una **x** en la Tabla 2 de coeficiente de argumentación que se muestra a continuación:

Fuente de argumentación	Grado de influencia		
	Alto	Medio	Bajo
Análisis teóricos realizados por Ud.			
Su experiencia obtenida.			
Estudio de trabajos de autores nacionales.			
Estudio de trabajos de autores extranjeros.			
Su propio conocimiento del estado del problema.			
Su intuición.			

Tabla 2: Coeficiente de argumentación.

A partir de la autoevaluación realizada y el patrón de valores para el coeficiente de argumentación, que se muestra en el Anexo 3, se calcula el coeficiente de argumentación **Ka**. Dónde:

**Ka** =  $\Sigma$  de los valores que se obtienen de sustituir las cruces de la Tabla por los valores correspondientes en su posición de la tabla de valores para el coeficiente de argumentación.

Luego de calculado los valores de **Kc** y **Ka**, se calcular el coeficiente de competencia **K** mediante la

fórmula:

$$K = \frac{Kc + Ka}{2}$$

Una vez calculado el valor de **K**

- ✓ Si  $0,8 < \mathbf{K} < 1$  el coeficiente de competencia alto.
- ✓ Si  $0,5 < \mathbf{K} < 0,8$  el coeficiente de competencia medio.
- ✓ Si  $\mathbf{K} < 0,5$  el coeficiente de competencia bajo.

Para obtener mejores resultados en la aplicación del Delphi es conveniente utilizar los expertos cuyo coeficiente de competencia sea alto o medio.

Luego de realizada la encuesta de autoevaluación a los posibles expertos, solo siete fueron seleccionados para formar parte del grupo de validación de la propuesta, pues fueron aquellos cuyos resultados arrojaron un coeficiente de competencia alto y medio (ver Tabla 3). Los resultados del análisis del coeficiente de competencia se muestran a continuación:

Experto	Kc	Ka	K	Competencia
1	0.8	0.8	0.8	Medio
2	0.8	0.7	0.75	Medio
3	0.8	0.9	0.85	Alto
4	0.8	0.9	0.85	Alto
5	0.6	1.0	0.8	Medio
6	0.7	0.9	0.8	Medio
7	1.0	1.0	1.0	Alto

Tabla 3: Coeficiente de expertos.

En la figura 22 que se muestra a continuación se representa el resultado de acuerdo al coeficiente de competencia del grupo resultante para la validación de la propuesta, donde los expertos con coeficiente alto representan un 50% de total y un 50% los que poseen coeficiente medio.



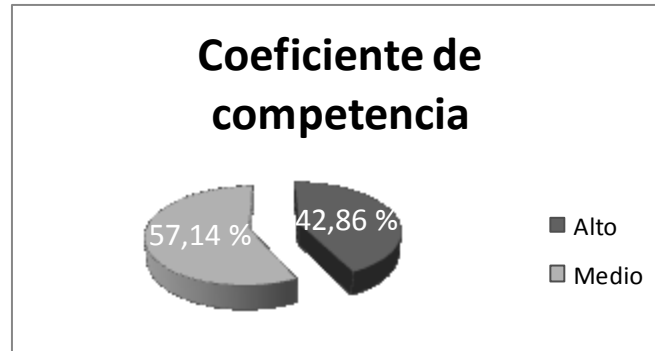


Figura 22: Porcentaje del coeficiente de competencia de los expertos.

### 3.2.3. Elaboración del cuestionario.

El cuestionario elaborado para validar la solución propuesta consta de 12 preguntas orientadas a evaluar aspectos críticos de la solución, el cual fue realizado por los siete expertos seleccionados los cuales evaluaron en las categorías ordinales de Muy Adecuado (MA), Bastante Adecuado (BA), Adecuado (A), Poco Adecuado (PA), y No Adecuado (NA). Con las respuestas dadas por los expertos se podrá obtener la concordancia entre ellos y la aceptación y validez de la propuesta. El cuestionario de validación utilizado en la investigación se muestra en el Anexo 4.

#### 3.2.3.1. Establecimiento de la concordancia entre los expertos.

Cuando se tienen datos de tipo ordinal el coeficiente de Kendall, el cual toma en consideración el orden, es usualmente un instrumento estadístico muy apropiado para indicar el grado de asociación de las evaluaciones ordinales hechas por evaluadores múltiples cuando se evalúa la misma muestra. Los valores del coeficiente deben oscilar entre 0 y 1. Entre mayor sea el valor Kendall, más fuerte será la asociación. (Alvarado, 2008)

Un coeficiente Kendall significativo implica que los evaluadores están aplicando esencialmente el mismo estándar cuando evalúan la muestra. Después de haber realizado el cuestionario, y con las respuestas de los expertos, se verificará si existe concordancia entre ellos mediante el coeficiente de Kendall, el cual se calcula mediante la siguiente fórmula:

$$W = \frac{12 \cdot S}{K^2 (N^3 + N)} \quad (1)$$

Dónde: **W**: es el coeficiente de concordancia. **S**: es la suma de los cuadrados de las desviaciones observadas de la media de  $S_j$  (rangos); y se calcula mediante la expresión:

## Capítulo 3. Validación de la propuesta de solución

$$S = \sum_{j=1}^n (S_j - \bar{S})^2 \quad (2)$$

Dónde:  $\bar{S}$  es la suma de los rangos dividido entre la cantidad de preguntas. **N**: Cantidad de preguntas, en este caso N=12. **K**: Cantidad de expertos, por tanto K=7.

$$\bar{S} = \frac{\sum_{j=1}^n S_j}{N} \quad (3)$$

A continuación se muestran los cálculos realizados para determinar la concordancia de los expertos:

	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7	Si
P1	4	3	4	4	4	4	3	26
P2	5	4	5	5	4	5	4	32
P3	5	4	4	4	3	5	4	29
P4	4	5	5	3	3	4	5	29
P5	4	4	5	4	4	4	5	30
P6	4	3	5	3	2	3	5	25
P7	5	3	3	4	4	4	4	27
P8	5	3	3	4	4	3	4	26
P9	4	4	4	3	4	5	5	29
P10	3	3	4	3	5	4	5	27
P11	4	4	4	3	4	3	5	27
P12	5	5	4	4	5	3	5	31

Tabla 4: Datos obtenidos de los expertos.

Sustituyendo los valores en la ecuación (3), se calcula el valor el valor de  $\bar{S}$  y se obtiene el valor de:

$$\bar{S} = \frac{338}{12} = 28.17$$

Luego para determinar la desviación media (**S**) se sustituye los valores en la ecuación (2) y se obtiene el resultado de **S** = 51.67.

Sustituyendo los valores en la ecuación (1) se obtiene el valor de Kendall:

$$W = \frac{12 * 51.67}{7^2((12^3) - 12)}$$

$$W = 0.00737358$$

Una vez calculado el coeficiente de Kendall (w) es necesario conocer su probabilidad de significancia, para lo cual se debe transformar éste valor en Chi cuadrado de Pearson mediante la fórmula siguiente:

$$X^2_{cal} = K(N - 1)w$$

Sustituyendo los valores en la ecuación se obtiene:

$$X^2 = 7 * (12 - 1) * 0.00737358$$

$$X^2 = 0.57$$

Para buscar el Chi cuadrado tabulado en la tabla de distribución, mostrada en el Anexo 6 se calcula el grado de libertad (gl) donde:  $gl = N - 1$

Sustituyendo los valores en la ecuación se obtiene:  $gl = 12 - 1 = 11$

$X^2_{tabla} = X^2_{\alpha;gl}$   $\alpha$  es el nivel de significación utilizado para calcular el nivel de confianza. El nivel de confianza es igual a  $100(1 - \alpha)$  %, es decir, un  $\alpha$  de 0,05 indica un nivel de confianza de 95 %. Una vez obtenido los valores se comparan  $X^2_{real}$  y  $X^2_{tabla}$ , si  $X^2_{real} < X^2_{tabla}$  entonces existe concordancia de criterios entre los expertos.

$$X^2_{real} < X^2_{tabla}$$

$$0.57 < 19.6752$$

Por lo tanto se puede concluir que hay concordancia entre los expertos.

### 3.2.4. Elección de la metodología.

La evaluación de los expertos se desarrolló a través del Método Delphi, ya que es considerado como uno de los métodos subjetivos de pronosticación más confiables, constituye un procedimiento para

confeccionar un cuadro de la evolución de situaciones complejas, a través de la elaboración estadística de las opiniones de los expertos en el tema tratado.

### 3.2.5. Ejecución de la metodología.

La ejecución de la metodología se inicia con la entrega a cada experto de la encuesta con los aspectos a evaluar, donde los mismos deben expresar sus ideas y criterios sobre las mejoras, deficiencias que presenta el proceso y que pudiera presentar al ser aplicado en la práctica.

### 3.2.6. Procesamiento de la información.

Los expertos que formaron parte del panel recibieron la propuesta como documentación para contestar los criterios encuestados, de igual manera el cuestionario a responder con un total de doce preguntas, los cuales algunos fueron enviados vía e-mail garantizando el anonimato de los mismos, debido a que fueron enviados a cada uno por separado, evitando que se supiera el nombre del resto de los miembros del panel. Se realizó una sola ronda de preguntas y luego se prosiguió a analizar los resultados.

Se confeccionaron tablas para ir recogiendo los resultados aportados por los expertos. Para ello se utilizó el programa Excel 2010 y dichos resultados se recogieron en una tabla como la que sigue:

Tabla de frecuencias absolutas						
No.	MA	BA	A	PA	NA	Total
1	0	5	2	0	0	7
2	4	3	0	0	0	7
3	2	4	1	0	0	7
4	3	2	2	0	0	7
5	2	5	0	0	0	7
6	2	1	3	1	0	7
7	1	4	2	0	0	7
8	1	3	3	0	0	7
9	2	4	1	0	0	7
10	2	2	3	0	0	7
11	1	4	2	0	0	7
12	4	2	1	0	0	7

Tabla 5: Tabla de frecuencias absolutas.

Después de haber tabulado todos estos datos se procede mediante los siguientes pasos para lograr los

## Capítulo 3. Validación de la propuesta de solución

resultados esperados:

**Paso 1:** Construir una tabla de frecuencias acumuladas, esto se hace sumándole a cada número en la fila el número anterior, excepto al primer número.

Tabla de frecuencias acumuladas					
No.	MA	BA	A	PA	NA
1	0	5	7	7	7
2	4	7	7	7	7
3	2	6	7	7	7
4	3	5	7	7	7
5	2	7	7	7	7
6	2	3	6	7	7
7	1	5	7	7	7
8	1	4	7	7	7
9	2	6	7	7	7
10	2	4	7	7	7
11	1	5	7	7	7
12	4	6	7	7	7

Tabla 6: Tabla de frecuencias acumuladas.

**Paso 2:** Copiar la tabla anterior y borrar los resultados numéricos con el objetivo de construir la tabla de Frecuencias relativas acumuladas. Los datos numéricos que aparecen en esta tabla se obtienen mediante la división de cada uno de los números de la tabla de **Frecuencias absolutas acumuladas** para cada pregunta por el número total de expertos presentes en la validación, en este caso serían siete.

Tabla de frecuencias relativas acumuladas					
No.	MA	BA	A	PA	NA
1	0	0.7142	1	1	1
2	0.5714	1	1	1	1
3	0.2857	0.8571	1	1	1
4	0.4285	0.7142	1	1	1
5	0.2857	1	1	1	1

## Capítulo 3. Validación de la propuesta de solución

6	0.2857	0.4285	0.8571	1	1
7	0.1428	0.7142	1	1	1
8	0.1428	0.5714	1	1	1
9	0.2857	0.8571	1	1	1
10	0.2857	0.5714	1	1	1
11	0.1428	0.7142	1	1	1
12	0.5714	0.8571	1	1	1

Tabla 7: Tabla de frecuencias relativas acumuladas.

**Paso 3:** Finalmente, se buscan las imágenes de los elementos de la tabla anterior por medio de la función. Estas imágenes se representan en la misma tabla anterior, sólo que se le adicionan 3 columnas y una fila para representar los siguientes resultados:

**Suma de las columnas:** esta nueva fila recoge en cada una de sus celdas la suma de todos los valores de la columna correspondiente.

**Suma de las filas:** esta nueva fila recoge en cada una de sus celdas la suma de todos los valores de la fila correspondiente.

**Para hallar el promedio General (N),** se divide la suma de las sumas de las filas (la cual tiene que ser igual a la suma de las sumas de las columnas) entre el resultado de multiplicar el número de aspectos que se están evaluando por el número de preguntas, en este caso se divide por 48 porque quedan sólo cuatro categorías, ya que la última ha sido eliminada.

El valor **N-P** da el valor promedio que otorgan los expertos a cada elemento propuesto. En este caso **N=0.76** y **P=promedio**. Con los resultados obtenidos hasta ahora ya se pueden calcular los puntos de corte, los cuales se van a colocar en la tabla como una fila nueva que va a recoger en cada una de las celdas el promedio de los valores de la aplicación de la función de distribución normal de la columna correspondiente. Los puntos de corte se calculan dividiendo la suma de las columnas entre la cantidad de preguntas realizadas.

Puntos de corte						N=0.76	
No.	MA	BA	A	PA	Suma	P	N-P
1	0	0.71	1	1	2.71	0.7	0.06
2	0.57	1	1	1	3.57	0.9	-0.14

## Capítulo 3. Validación de la propuesta de solución

3	0.29	0.86	1	1	3.14	0.8	-0.04
4	0.43	0.71	1	1	3.14	0.8	-0.04
5	0.29	1	1	1	3.29	0.8	-0.04
6	0.2857	0.43	0.86	1	2.57	0.6	0.16
7	0.14	0.71	1	1	2.86	0.7	0.06
8	0.14	0.57	1	1	2.71	0.7	0.06
9	0.29	0.86	1	1	3.14	0.8	-0.04
10	0.29	0.57	1	1	2.86	0.7	0.06
11	0.14	0.71	1	1	2.86	0.7	0.06
12	0.57	0.86	1	1	3.43	0.9	-0.14
Suma	3.43	9	11.86	12	36.28		
Puntos de corte	0.29	0.75	0.99	1			

Tabla 8: Puntos de corte.

La suma obtenida de las 4 primeras columnas da los puntos de corte. Estos puntos de corte se utilizan para determinar el grado de adecuación o categoría de cada aspecto encuestado según los expertos. El grado de adecuación se muestra en la tabla siguiente:

Muy adecuado	Bien adecuado	Adecuado	Poco adecuado	No adecuado
0.29	0.75	0.99	1	

Tabla 9: Grados de adecuación.

Si el valor promedio de adecuación del elemento a evaluar es:

- ✓ Menor o igual que 0.29 el nivel de adecuación es Muy adecuado.
- ✓ Mayor que 0.29 y menor o igual que 0.75 el nivel de adecuación es Bastante adecuado.
- ✓ Mayor que 0.75 y menor o igual que 0.99 el nivel de adecuación es Adecuado.
- ✓ Entre 0.99 y 1 el nivel de adecuación es Poco adecuado.
- ✓ Mayor que 1 el nivel de adecuación es No adecuado.

Una vez aplicado el método Delphi, las encuestas realizadas a los 7 expertos arrojaron resultados satisfactorios:

- ✓ Todas las preguntas fueron valoradas por los expertos de Muy adecuada.
- ✓ El 100% de los expertos coinciden en la utilidad que tiene la puesta en práctica de la propuesta en

el departamento Soluciones Aduanales del CEIGE.

### **3.3. Conclusiones parciales**

Con la validación de la propuesta de solución queda demostrado que los objetivos trazados se cumplieron y los expertos de la investigación corroboraron esta afirmación con su valoración. Y así lo demuestra el análisis estadístico de los resultados obtenidos en la encuesta, en el cual los expertos definieron muy adecuado todos los elementos mostrados. De esta forma no se hizo necesario realizar otra iteración del método, porque todos los expertos estuvieron de acuerdo con el modelo de desarrollo de software planteado, por lo que se puede concluir que:

- ✓ El modelo de desarrollo de software propuesto contribuye a mejorar la planificación, el control de las actividades y la toma de decisiones durante el proceso de desarrollo de software en el departamento Soluciones Aduanales del CEIGE, por tanto mejorará la organización de su trabajo productivo.
- ✓ Los roles y las responsabilidades definidas para los mismos son suficientes para llevar a cabo su trabajo de manera efectiva en el departamento Soluciones Aduanales del CEIGE.
- ✓ Los artefactos propuestos son suficientes para recoger las evidencias necesarias para el departamento Soluciones Aduanales del CEIGE.
- ✓ Las actividades de CMMI propuestas permiten ejecutar las prácticas necesarias por áreas de proceso del nivel dos para dar cumplimiento a los objetivos del programa de mejora llevado a cabo por la universidad.



### **Conclusiones generales**

Para alcanzar los objetivos trazados en el presente trabajo de diploma se llevaron a cabo una serie de tareas que permitieron obtener las metas propuestas, por lo que se puede concluir de forma general lo siguiente:

Se realizó el estudio bibliográfico de los principales modelos de desarrollo de software existentes, arribando a la selección de los puntos fuertes de algunos de ellos para fusionarlos y conformar el modelo de desarrollo para el departamento Soluciones Aduanales del CEIGE.

Al realizar un estudio referente al proceso de desarrollo de software en el departamento de Soluciones Aduanales del CEIGE se definió las actividades, artefactos y roles para el modelo de desarrollo, se confeccionó la propuesta de solución, incluyendo las buenas prácticas del nivel dos de CMMI para la mejora de procesos en el ciclo de desarrollo del producto, el cual ayudará a mejorar la organización del trabajo productivo del departamento.

Como parte final de la investigación, se validó la propuesta a través del Método Delphi como variante de los Métodos de Expertos, arrojando como resultado la aceptación de los elementos propuestos respaldado por el grado de coincidencia en los criterios del grupo de expertos.

### Recomendaciones

Partiendo de que con el desarrollo del presente trabajo se dio cumplimiento a su objetivo general se recomienda lo siguiente:

- ✓ Adoptar el modelo de desarrollo en los proyectos productivos del departamento Soluciones Aduanales del CEIGE.
- ✓ Al elaborar el modelo de desarrollo del CEIGE tener en cuenta la propuesta del modelo de desarrollo de software realizado para el departamento Soluciones Aduanales del CEIGE.
- ✓ Analizar las áreas de procesos del nivel tres de CMMI, para tenerlas en cuenta en futuras mejoras de la propuesta.
- ✓ Tener en cuenta para futuras versiones del modelo la nueva estructura que se está aplicando actualmente en el centro.

### Referencias bibliográficas

- Blaya, Inmaculada. 2006.** Gestion por procesos. 2006.
- Chi, Genaro Alberto Gómez. 2009.** Modelos de desarrollo de software. 2009.
- Chrissis, Mary Beth. 2009.** Guía para la integración de procesos. 2009.
- Documento del programa de mejoras. 2009.** IPP-3510\_2009 Libro de Proceso para la Administración de Requisitos. 2009.
- Documento del programa de mejoras. 2009.** IPP-3520\_2009 Libro de Proceso para PPQA v2.0. 2009.
- Documento del programa de mejoras. 2009.** IPP-3530\_2009 Libro de Proceso Monitoreo y Control de Proyecto 2.1. 2009.
- Documento del programa de mejoras. 2009.** IPP-3540\_2009 Libro de Proceso para la Planeación del Proyecto. 2009.
- Documento del programa de mejoras. 2009.** IPP-3560\_2009 Libro de Proceso para la Administración de la Configuración. 2009.
- Documentos del programa de mejoras. IPP-3550-2009\_** Libro de Proceso para Medición y Analisis v2.0.
- Documento del programa de mejoras. 2009.** IPP-3570\_2009 Libro de Proceso para la Administración de Acuerdos con Proveedores. 2009.
- Gervás, Pablo. 2007.** Proceso de desarrollo de software. 2007.
- Gordon, Helmer, Delkey. 1963-1964.** Método Delphi. [En línea] 1963-1964.]  
[http://www.12manage.com/methods\\_helmer\\_delphi\\_method\\_es.html](http://www.12manage.com/methods_helmer_delphi_method_es.html).
- Jacobson.Booch, Rumbaugh. 2005.** El proceso unificado de desarrollo de software. 2005.
- Letelier. 2008.** Introducción Proceso de Software. 2008.
- Picado Alvarado, Federico. 2008.** Análisis de concordancia de atributos. [En línea] diciembre de 2008.  
[http://www.tec.ac.cr/sitios/Vicerrectoria/vie/editorial\\_tecnologica/Revista\\_Tecnologia\\_Marcha/pdf/tecnologia\\_marcha\\_21-4/cap%203.pdf](http://www.tec.ac.cr/sitios/Vicerrectoria/vie/editorial_tecnologica/Revista_Tecnologia_Marcha/pdf/tecnologia_marcha_21-4/cap%203.pdf).
- Pressman. 2005.** Ingeniería del software enfoque práctico. 2005.
- Sommerville. 2005.** Ingeniería del software. Madrid : s.n., 2005. 84-7829-074-5.
- Universidad politécnica de Valencia. 2006.** Introducción al proceso de desarrollo de software. 2006.
- Victorino, Alejandro Pérez Cortés y José Bernardo Parra. 2011.** Implementación de tableros de control en el área de mejora continua en una empresa de manufactura. 2011.

### Bibliografías

**ALLSOFT. 2008.** El modelo CMMI. Monterrey : s.n., 2008.

**Blaya, Inmaculada. 2006.** Gestion por procesos. 2006.

**Boehm, B. W. 1988.** A Spiral Model of Software Development and Enhancement. 1988. IEEE Computer.

**C.V., ALLSOFT S.A. de. 2003.** Modelos de Desarrollo. Monterrey, N.L. : s.n., 2003.

**Chi, Genaro Alberto Gómez. 2009.** Modelos de desarrollo de software. 2009.

**Chrissis, Mary Beth. 2009.** Guía para la integración de procesos. 2009.

**Dalkey, Norman C, Brown, Bernice y Cochran, S. 1969.** The Delphi Method, III: Use of self rating to improve group estimates. Technological Forecasting and Social Change. [En línea] noviembre de 1969. [http://www.rand.org/pubs/research\\_memoranda/2006/RM6115.pdf](http://www.rand.org/pubs/research_memoranda/2006/RM6115.pdf).

**Documento del programa de mejoras. 2009.** Roles y responsabilidades. 2009.

**Documento del programa de mejoras. 2009. 2009.** Libro de Proceso para la Administración de Requisitos. 2009.

**Documento del programa de mejoras. 2009. 2009.** Libro de Proceso para PPQA v2.0. 2009.

**Documento del programa de mejoras. 2009. 2009.** Libro de Proceso Monitoreo y Control de Proyecto 2.1. 2009.

**Documento del programa de mejoras. 2009. 2009.** Libro de Proceso para la Planeación del Proyecto. 2009.

**Documento del programa de mejoras. 2009. 2009.** Libro de Proceso para la Administración de la Configuración. 2009.

**Documentos del programa de mejoras.** Libro de Proceso para Medición y Analisis v2.0.

**Documento del programa de mejoras. 2009. 2009.** Libro de Proceso para la Administración de Acuerdos con Proveedores. 2009.

**Gervás, Pablo. 2007.** Proceso de desarrollo de software. 2007.

**Gordon, Helmer, Delkey. 1963-1964.** Método Delphi. [En línea] 1963-1964. [Citado el: 5 de 4 de 2012.] [http://www.12manage.com/methods\\_helmer\\_delphi\\_method\\_es.html](http://www.12manage.com/methods_helmer_delphi_method_es.html).

**Instituto Nacional de Tecnologías de la Comunicación,. 2009.** Guía de validación y verificación. 2009.

**Isaías Carrillo Pérez, Rodrigo Pérez González, Aureliano David Rodríguez Martín. 2008.** Metodología de desarrollo del software. 2008.

**Jacobson.Booch, Rumbaugh. 2005.** El proceso unificado de desarrollo de software. 2005.

- Jiménez, Hugo F. Arboleda. 2005.** ACIS. Modelos de ciclo de vida en desarrollo de software. [En línea] 2005. <http://www.acis.org.co/index.php?id=551>.
- José H. Canós, Patricio Letelier yM<sup>a</sup> Carmen Penadés. 2007.** Metodologías Ágiles en el Desarrollo de Software. [En línea] 2007. <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>.
- Letelier. 2008.** Introducción Proceso de Software. 2008.
- MEDINA, YENNY FIGUEROA 2006.** Metodologías de desarrollo de software. 2006.
- Méndez, Gonzalo. 2008.** Proceso Software y Ciclo de Vida. 2008.
- Mills, H., O'Neill, D. 1980.** The Management of Software Engineering, IBM Systems. 1980.
- Picado Alvarado, Federico. 2008.** Análisis de concordancia de atributos. [En línea] diciembre de 2008. [http://www.tec.ac.cr/sitios/Vicerrectoria/vie/editorial\\_tecnologica/Revista\\_Tecnologia\\_Marcha/pdf/tecnologia\\_marcha\\_21-4/cap%203.pdf](http://www.tec.ac.cr/sitios/Vicerrectoria/vie/editorial_tecnologica/Revista_Tecnologia_Marcha/pdf/tecnologia_marcha_21-4/cap%203.pdf).
- Pressman. 2005.** Ingeniería del software enfoque práctico. 2005.
- Real Academia Española. 2008.** Diccionario de la Real Academia Española. [En línea] 2008. <http://www.rae.es/rae.html>.
- Royce, W. 1970.** Managing the development of large software systems: concepts and technique. 1970. IEEE Westcon.
- Sommerville. 2005.** Ingeniería del software. Madrid : s.n., 2005. 84-7829-074-5.
- Universidad politécnica de Valencia. 2006.** Introducción al proceso de desarrollo de software. 2006.
- Victorino, Alejandro Pérez Cortés y José Bernardo Parra. 2011.** Implementación de tableros de control en el área de mejora continua en una empresa de manufactura. 2011.
- Zavala-Ruiz, J. 2008.** Organizational Analysis of Small Software Organizations: Framework and Case Study" en H. Oktaba y M. Piattini (eds) Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies. 2008.

### Glosario de términos

**Artefactos:** Un artefacto es todo tipo de información producida, cambiada o usada por el proceso, es un producto del proceso que puede tener asociado una plantilla, como guía de su contenido. Los artefactos podrán asociarse a las actividades como entrada o salida de las mismas.

**Actividad:** Conjunto de operaciones o tareas propias de una persona o entidad que permite que el trabajo se va a realizar sea descrito y entendido de manera precisa por aquellos que tienen que ejecutarlo.

**BPMN:** Modelado de procesos del negocio.

**Calidad:** Conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas.

**CMMI:** Modelo de Madurez de las Capacidades Integrado.

**Despliegue:** Producir una versión del producto y entregar el software a sus usuarios finales.

**Eficiencia:** Capacidad para lograr un fin empleando los mejores medios posibles

**Eficacia:** Capacidad para obrar o para conseguir un resultado determinado.

**Estrategia:** Conjunto de acciones que se llevan a cabo para lograr un determinado fin.

**Fases:** Las fases están conformadas por un conjunto de objetivos definidos y solamente se puede dar por concluida una vez que se hayan alcanzado estos objetivos.

**Gestión de proyectos:** Es la aplicación de varios conocimientos, habilidades, herramientas y técnicas para planificar y satisfacer las actividades de un proyecto.

**Herramientas:** Utensilios o provisiones necesarias para poder emprender un proyecto de software. Soportan los procesos de desarrollo de software modernos.

**Hitos:** Tareas con un valor, representa una fecha importante en un proyecto, como la finalización de una fase del proyecto, o la fecha de un informe en particular es vencimiento.

**Línea base:** es un conjunto de productos de trabajo formalmente revisados y aprobados que sirve de base para desarrollos posteriores o entregas.

**Organización:** Grupo social formado por personas, tareas y administración que interactúan en el marco de una estructura sistemática para cumplir con sus objetivos.

**Producto de software:** Son los artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables, y documentación.

**Proceso:** Es un conjunto de actividades o eventos que se realizan o suceden con un determinado fin.

**Proceso de software:** Es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto para transformar los requisitos de usuario en un producto.

**Planificación:** La planificación es el establecimiento de objetivos, y la decisión sobre las estrategias y las tareas necesarias para alcanzarlas.

**Prototipo:** Es una representación limitada del diseño de un producto que permite a las partes responsables de su creación experimentar, probarlo en situaciones reales y explorar su uso.

**Repositorio:** Es un sitio centralizado donde se almacena y mantiene información digital, como bases de datos o archivos informáticos.

**Requerimientos:** Conjuntos de requisitos que debe cumplir el sistema a desarrollar.

**Recursos:** Son todos aquellos elementos necesarios, tanto tangibles como intangibles, para que una organización cumpla con sus objetivos.

**Software:** Todos los componentes intangibles de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).

**Salidas:** Resultado del procedimiento.