

Universidad de las Ciencias Informáticas
FACULTAD 15



“Modelo de gestión de la arquitectura de software de un sistema de planificación empresarial.”

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

Autor: Mailyn Hernández Castellanos.

Tutor(es): Ing. Mileidy Magalys Sarduy Pérez.
Ing. César Lage Codorníu.

Ciudad de La Habana, Junio del 2010.

“Año 52 de la Revolución.”

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 15 de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los __ días del mes de _____ del año _____.

AUTOR

Mailyn Hernández Castellanos.

TUTOR

Ing. Mileidy Magalys Sarduy Pérez.

TUTOR

Ing. César Lage Codorniú.

En el Departamento de Tecnología del Centro de Informatización para la Gestión de Entidades (CEIGE), se hace imprescindible diseñar un modelo que garantice la eficacia en el desarrollo de las aplicaciones pues se carece de un modelo de gestión de la Arquitectura de Software (ASW). Con esta finalidad y luego de un estudio detallado de los principales conceptos y las diferentes áreas de trabajo o vistas en las que se organiza la ASW, y el estudio de las metodologías de desarrollo de software existente principalmente en aquellas que se caracterizan por ser ligeras o ágiles, se realiza una propuesta que consta de un flujo de actividades que están relacionadas de forma directa con la ASW, se identifican los roles fundamentales, los artefactos generados en cada actividad, se incluye además como gestionar ese flujo de actividades a partir de herramientas y técnicas, con el objetivo de tener un control del avance y desarrollo durante el ciclo de vida del producto. Además se explica cómo gestionar las desviaciones y riesgos, que influyen en el avance de la construcción de los componentes. Luego esta propuesta es validada a través del Método de Expertos, realizando una correcta selección de los participantes a través del coeficiente de competencia, después de haber aplicado el cuestionario de validación se desarrolló el estableciendo del nivel de concordancia entre los miembros que conforman el panel de expertos, se prosiguió con la tabulación de los resultados que arrojaron una aceptación de muy adecuado por parte de los participantes.

Palabras claves:

Arquitectura, modelo, metodologías.

Contenido

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....	3
INTRODUCCIÓN.....	3
ARQUITECTURA DE SOFTWARE.....	3
1.1 ESCUELAS O CORRIENTES DE LA ARQUITECTURA	4
MODELOS DE PROCESOS O DESARROLLO DE SOFTWARE.....	9
1.1 TIPOS DE MODELOS DE DESARROLLO DE SOFTWARE	9
METODOLOGÍA EN EL DESARROLLO DE SOFTWARE	11
1.1 METODOLOGÍAS TRADICIONALES O PESADAS	11
1.2 METODOLOGÍAS LIGERAS O ÁGILES	12
CAPÍTULO 2 MODELO DE DESARROLLO.....	17
INTRODUCCIÓN.....	17
¿CÓMO INTERVIENEN LAS VISTAS ARQUITECTÓNICAS EN EL CICLO DE VIDA DEL PRODUCTO?	17
2.1 CONCEPCIÓN.....	18
2.2 ELABORACIÓN	19
2.3 CONSTRUCCIÓN.....	20
2.4 CIERRE	21
2.5 RESULTADOS.....	21
ESTRUCTURA DEL MODELO.....	23
2.6 ALCANCE DEL MODELO	24
2.6.1 PREMISAS Y PRINCIPIOS DEL MODELO	24
2.6.2 CARACTERÍSTICAS DEL MODELO	24
2.7 DESCRIPCIÓN DEL MODELO.....	25
2.7.1 FLUJO DE ACTIVIDADES.....	25
2.7.2 ARTEFACTOS	28
2.7.3 ROLES	30
2.7.4 HERRAMIENTAS Y TÉCNICAS.....	32
2.7.5 GESTIÓN DE RIESGOS Y DESVIACIONES	33
CONCLUSIONES PARCIALES.....	35
CAPÍTULO 3 VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	36
INTRODUCCIÓN.....	36
EL MÉTODO DE EXPERTOS.....	36
EL MÉTODO DELPHI.....	37
APLICACIÓN DEL MÉTODO DELPHI	38

3.1	SELECCIÓN DE LOS EXPERTOS.....	39
3.2	CÁLCULO DEL COEFICIENTE DE COMPETENCIA	39
3.3	ELABORACIÓN DEL CUESTIONARIO DE VALIDACIÓN DE LA PROPUESTA	41
3.4	ESTABLECER LA CONCORDANCIA DE LOS EXPERTOS MEDIANTE EL COEFICIENTE DE KENDALL	42
3.5	DESARROLLO PRÁCTICO Y EXPLORATORIO DE LOS RESULTADOS	43
3.6	RESULTADOS DE LA VALIDACIÓN DEL MODELO.....	47
3.6.1	RESULTADOS OBTENIDOS DE LA ENCUESTA DE VALIDACIÓN.....	48
	CONCLUSIONES PARCIALES.....	51
	CONCLUSIONES GENERALES	52
	RECOMENDACIONES	53
	BIBLIOGRAFÍA.....	54
	GLOSARIO DE TÉRMINOS.....	56
	ANEXOS.....	58

INTRODUCCIÓN

En la actualidad la gestión empresarial juega un rol muy importante en las empresas, ya que de la dirección que tenga la empresa, la eficiencia, las habilidades y estrategias que se utilicen, dependerá en gran medida la obtención de buenos resultados.

Actualmente, existe abundante información que permite seleccionar diferentes herramientas como guías y frameworks^{*1} contribuyendo que en la ejecución de los proyectos ayuden a resolver los aspectos técnicos de los desarrollos. Sin embargo, a medida que crece la complejidad de las aplicaciones, y se extiende el uso de sistemas basados en componentes estos deben ser extensibles y adaptables a las soluciones generales, donde aún con los avances actuales, las metodologías, herramientas y procesos con los que se cuenta, la gestión y el desarrollo de la arquitectura se torna compleja.

Este proceso de desarrollo y gestión de la arquitectura se evidencia en el Departamento de Tecnología del CEIGE, donde se torna difícil el desarrollo de la ASW; debido a que el grupo de desarrollo está aumentando como consecuencia al gran alcance que tiene el proyecto; provocando de esta manera, que el tamaño del equipo condicione un mayor esfuerzo en cuanto a la definición de normas y estándares que se deben cumplir para el desarrollo del software, así como la solución de diseños en conjunto y la adaptación de soluciones particulares a casos generales. Además de que no se encuentran claras las actividades ni los artefactos a generar por cada uno de los roles de la arquitectura de software, y no está bien definido el modelo de gestión que guíe y oriente al equipo de trabajo en el departamento.

Ante la situación descrita anteriormente se traza como *problema a resolver* de esta investigación: la carencia de un modelo de gestión de la ASW afecta la eficacia en el desarrollo de las aplicaciones.

Siendo el *objeto de estudio* de esta investigación el modelo de desarrollo para la gestión de la arquitectura de software. De ello se deriva que el *campo de acción* sea las metodologías de desarrollo de software en sistemas de gestión empresarial.

Se plantea como *objetivo general* elaborar un modelo de gestión de la ASW para un sistema de planificación empresarial.

Como guía de la investigación se establecen los siguientes *objetivos específicos*:

- Definir marco teórico de la investigación.

¹ El * significa que la palabra está en el Glosario de Términos

- Diseñar un modelo de gestión de la arquitectura para sistemas de planificación empresarial.
- Validar el modelo a partir de métricas e indicadores.

Como *idea a defender* de esta investigación se plantea: la elaboración de un modelo para gestionar la arquitectura de software, garantizará la eficacia en el desarrollo de aplicaciones.

Se espera como *resultado* de la investigación un Modelo de gestión de la arquitectura de un sistema de planificación empresarial.

Estructura del documento

Capítulo 1: Se aborda el estado del arte de las metodologías para el desarrollo de software, enfocando el estudio en las metodologías de desarrollo de software ágiles, se definen los principales conceptos respecto a la arquitectura de software y las diferentes áreas de trabajo o vistas en las que se organiza la misma.

Capítulo 2: Se propone el Modelo de Gestión para la construcción e implementación de la arquitectura de software.

Capítulo 3: Se realiza la evaluación a través de métricas e indicadores, correspondiente al Modelo de Gestión propuesto en el capítulo anterior.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

INTRODUCCIÓN

En el presente capítulo se realiza un estudio detallado sobre el estado del arte de la arquitectura* del software, así como los principales conceptos y las diferentes áreas de trabajo o vistas en las que se organiza la misma. Se centra la investigación en el estudio de las metodologías de desarrollo de software*, principalmente en aquellas que se caracterizan por ser ligeras o ágiles.

ARQUITECTURA DE SOFTWARE

Los primeros estudios realizados referentes a la ASW fueron guiados por *Davis Garlan* y *Mary Shaw* en la década de los '90 del siglo pasado. Al realizar un análisis de los precedentes históricos de la ASW que tanta aceptación tiene hoy en el mundo del desarrollo de software, mediante este estudio se hace referencia al surgimiento del concepto de ASW por *Edsger Dijkstra*² en el año 1960. Luego *Dewayne Perry* de AT&T Bell Laboratories de New Jersey y *Alexander Wolf* de la Universidad de Colorado retoman el elemento teórico, es decir lo revitalizan por lo que se les puede calificar como padres de la arquitectura del software. Desde ese período se le dio comienzo a uno de los dilemas más importantes de la actualidad y de la industria de software en sí: definir el concepto de la arquitectura del software, pues se encuentran muchas definiciones respecto al tema de ASW y ninguna de ellas ha sido reconocida.

La definición de la ASW según *IEEE** Std 1471-2000 plantea:

“La ASW es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución” (Josep).

Philippe Kruchten, *Grady Booch*, basados en un trabajo preliminar de *David Garlan* y *Mary Shaw* plantean el siguiente concepto de arquitectura del software:

“La arquitectura comprende un conjunto de decisiones significativas con respecto a la organización de un sistema de software incluyendo la selección de los elementos estructurales y sus interfaces que componen al sistema, el comportamiento y las especificaciones de la colaboración entre esos componentes, la interdependencia entre los elementos estructurales y de comportamiento en sistemas de gran envergadura y el estilo arquitectónico que guía esta organización”. (Meier)

² Dijkstra: Entre sus contribuciones a la informática está el problema del camino más corto, también conocido como el algoritmo de Dijkstra, la Notación polaca inversa y el relacionado algoritmo Shunting Yard.

Otra definición aceptada es la de *Clements*: “La ASW es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones”. (Reynoso)

Kazman plantea: “La ASW de un programa o sistema de computación es la(s) estructura(s) del sistema que comprende los componentes del software, las propiedades visibles de esos componentes y las relaciones entre ellos.” (Bass)

Billy Reynoso: Vista estructural de alto nivel, define estilo o combinación de estilos para una solución, se concentra en requerimientos* no funcionales, los requerimientos funcionales se satisfacen mediante modelado y diseño de aplicación, esencial para éxito o fracaso de un proyecto. (Reynoso)

Luego de un estudio de estos y otros conceptos de la arquitectura del software se puede llegar a la conclusión que: la ASW es la representación o diseño de alto nivel de un sistema. Posee dimensiones de gestión, vistas de organización o representación, y elementos de solución. Está estructurada por las partes que conforman el software, principios de diseño y evolución además del ambiente tecnológico de implantación.

1.1 Escuelas o corrientes de la arquitectura

Actualmente la gran mayoría de los elementos y definiciones que contiene la disciplina de la ASW se dictaminan y se rigen por seis corrientes fundamentales entre las que se encuentran:

- ✓ Arquitectura como etapa de ingeniería y diseño orientada a objetos: basada en el modelo de Ivar Jacobson³. Este modelo está estrechamente ligado al mundo de UML* y Rational. En esta postura, la arquitectura se restringe a las fases iniciales y preliminares del proceso y concierne a los niveles más elevados de abstracción. La definición de la arquitectura que se promueve en esta corriente tiene que ver con los aspectos formales a la hora del desarrollo. Otras definiciones revelan que la arquitectura de software, en esta perspectiva, concierne a decisiones sobre organización, selección de elementos estructurales, comportamiento, composición y estilo arquitectónico susceptibles de ser descritas a través de las cinco vistas clásicas del modelo “4+1” de Kruchten.

³ Es el padre de los componentes y de la arquitectura de componentes, de la ingeniería moderna de negocios, del desarrollo de software orientado a aspectos, del UML (Unified Modelling Language), y del proceso Unificado Racional (Rational Unified Process). Es el líder del pensamiento en el mundo del software donde ha hecho varias contribuciones decisivas.

- ✓ Arquitectura estructural, basada en un modelo estático de estilos, ADLs⁴ y vistas: constituye la corriente fundacional y clásica de la disciplina. Los representantes de esta corriente son todos académicos, mayormente de la Universidad Carnegie Mellon en Pittsburgh: Mary Shaw, Paul Clements, David Garlan. En toda la corriente, el diseño arquitectónico no sólo es el de más alto nivel de abstracción, sino que además no tiene por qué coincidir con la configuración explícita de las aplicaciones; rara vez se encontrarán referencias a lenguajes de programación o piezas de código.
- ✓ Estructuralismo arquitectónico radical: se trata de un desprendimiento de la corriente anterior, mayormente europeo, que asumen una actitud más confortativa con el mundo UML. En el seno de este movimiento hay al menos dos tendencias, una que excluye de plano la relevancia del modelado orientado a objetos para la AS y otra que procura definir nuevos estereotipos de UML como correctivos de la situación.
- ✓ Arquitectura basada en patrones: se basa esencialmente en la redefinición de los estilos como patrones POSA⁵. El diseño consiste en identificar y articular patrones preexistentes, que se definen en forma parecida a los estilos de arquitectura.
- ✓ Arquitectura procesual: Desde comienzos del siglo XXI, con centro en el SEI y con participación de algunos arquitectos de Carnegie Mellon de la primera generación y muchos nombres nuevos de la segunda como: Rick Kazman, Len Bass, Paul Clements, Felix Bachmann, Fabio Peruzzi, Jeromy Carrière, Mario Barbacci, Charles Weinstock. Intenta establecer modelos de ciclo de vida y técnicas de diseño de requerimientos, diseño, análisis, selección de alternativas, validación, comparación, estimación de calidad y justificación económica específicas para la ASW. Otras variantes dentro de la corriente procesual se caracterizan principalmente por cambios en la etapa del proceso de extracción de arquitectura, generalización y reutilización.
- ✓ Arquitectura basada en escenarios: es la corriente más nueva, se trata de un movimiento predominantemente europeo, con centro en Holanda. Recupera el nexo de la ASW con los requerimientos y la funcionalidad del sistema, ocasionalmente borroso en la arquitectura estructural clásica. En esta corriente suele utilizarse diagramas de casos de usos UML como herramienta informal u ocasional, dado que los casos de uso son uno de los escenarios posibles y estos no están orientados a objetos. Los autores vinculados con esta modalidad han sido, aparte de los codificadores de ATAM, CBAM, QASAR y demás métodos del SEI, los arquitectos holandeses de la Universidad Técnica de Eindhoven, de la Universidad Brije, de la Universidad de Groningen y de

⁴ Architecture Description Language (Lenguajes de Descripción Arquitectónica)

⁵ Pattern Oriented Software Architecture.

Philips Research: Mugurel Ionita, Dieter Hammer, Henk Obbink, Hans de Bruin, Hans van Vliet, Eelke Folmer, Jilles van Gorp, Jan Bosch. La representación de holandeses es significativa; la Universidad de Eindhoven es, incidentalmente, el lugar en el que surgió lo que P. I. Sharp proponía llamar la escuela arquitectónica de Dijkstra. (Reynoso, 2004)

Software Engineering Institute (SEI): dentro del campo de la arquitectura el SEI juega un gran papel ya que provee técnicas y métodos para crear, documentar, evaluar y reconstruir una arquitectura robusta. Aportan la herramienta Architecture Expert (ArchE) que es una herramienta de diseño auxiliar que ayuda a los arquitectos explorar diseños arquitectónicos impulsados por los atributos de calidad. Arche muestra al arquitecto propuestas para mejorar la arquitectura actual y permite al arquitecto decidir cuál es la mejor alternativa. (Larisa, y otros)

Debido a las características del CEIGE el ciclo de vida de los proyectos de desarrollo de software está *centrado en la arquitectura*, pues la arquitectura establece la línea base así como los elementos estructurales tomando como partida inicial los elementos de la arquitectura de negocio, interviene en la gestión de cambios diseñando la evolución e integración del producto, orienta las prioridades del desarrollo, resuelve las necesidades tecnológicas y da soporte para el desarrollo del software. Otra característica es que el desarrollo está *orientado a componentes* dado que las iteraciones son orientadas por el significado arquitectónico de los componentes, y estos últimos son abstracciones arquitectónicas de los procesos de negocio en conjunto con los requisitos asociados que modelan, lo que trae como resultado que el componente sea la unidad de medición y ordenamiento de las iteraciones en el desarrollo del software. El desarrollo es *iterativo e incremental* planificado y coordinado con el equipo de arquitectura en conjunto con el cliente y la alta gerencia, teniendo en cuenta que cada iteración representa el desarrollo de componentes, estos a su vez son integrados, permitiendo la evolución incremental en el desarrollo del software. Todo este desarrollo tiene que ser *ágil y adaptable al cambio* pues con el desarrollo de las partes solo se formaliza las características principales de la solución, de ahí que es fundamental priorizar los talleres y la comunicación entre las personas. En el proyecto están involucrados el cliente y los funcionales estos comparten las responsabilidades del éxito del mismo. Los cambios que se realizan son conciliados, discutidos y aprobados. (proyecto)

Teniendo presente las características anteriormente mencionadas y gran cantidad de componentes de software que componen el proyecto, las constantes y complejas integraciones entre componentes, hacen necesario el uso de vistas arquitectónicas, como una forma de organización y análisis de la arquitectura a establecer para el software, por otra parte se hace indispensable el uso de un lenguaje de descripción arquitectónico (ADL, del inglés), el cual permitirá obtener un modelo de la interacción entre distintos

componentes de software, utilizado luego para la toma de decisiones sobre el diseño y futuro desarrollo del software.

Basado en el estudio de cada una de las corrientes existentes a nivel mundial y en las características del sistema a desarrollar mencionadas con anterioridad, se asume la corriente arquitectónica procesual del centro del SEI, pero no todas las problemáticas originadas por las características propias del sistema y del equipo que compone el proyecto ERP*-Cuba quedan cubiertas con los elementos que nos provee esta corriente, por lo que se ajustan parte de sus características, como la relación con los requerimientos y los métodos de evaluación, incorporando elementos arquitectónicos de otras corrientes como los escenarios arquitectónicos, las vistas, y los lenguajes de descripción arquitectónicos, de tal forma que se constituye una guía compuesta por los elementos arquitectónicos mencionados que tiene su basamento fundamentalmente en la escuela arquitectónica procesual que promueve el SEI.

En el proyecto ERP-Cuba, luego de un profundo estudio basado en la documentación técnica disponible y la consulta y estudio de varios proyectos realizados en nuestra universidad, se definieron las siguientes vistas:

- Vista de la Arquitectura de Sistema

La vista de sistema de la Arquitectura de Software, representa una proyección simétrica de alto nivel de los procesos de negocio o arquitectura de negocio que se trabaja, expresada en elementos, conectores, configuraciones y restricciones. (González Alvares, y otros, 2009)

- Vista de la Arquitectura de Datos

Define como se almacenarán, manejarán y usarán los datos en un sistema. También establece pautas para las operaciones con los datos para poder hacer posible predecir, modelar, guiar y controlar el flujo de datos en un sistema. Es donde se modela todo lo persistente y relacionado a la BD. Identifica y precisa las mejores clases de datos que apoyan las funciones del negocio definidas en el modelo de negocios. Tiene como objetivo puntualizar los principales tipos y fuentes de datos necesarios para dar soporte a las actividades de la empresa, de manera que sean: completas, consistentes y estables.

- Vista de la Arquitectura de Integración

El objetivo que persigue esta vista, es plasmar detalladamente qué es lo que se hace en ella y cómo se hace. Aquí se analizan los procesos de integración del negocio; se identifican los principales flujos de colaboración o unificación en la arquitectura; se establecen los conceptos más críticos en la integración de sus procesos, de acuerdo al nivel de incidencia en los mismos,

partiendo del análisis de su arquitectura, lo que permitirá conocer las áreas más críticas de integración y por último se analiza entradas y salidas de cada componente identificado por la arquitectura de sistema.

- Vista de la Arquitectura de Seguridad

Chequea e implementa todos los aspectos relacionados con el acceso a la aplicación, la modificación, lectura o eliminación de la información, etc. Es la encargada de establecer toda la política de seguridad a seguir en las diferentes fases o entornos de un sistema. Para esto deben identificar claramente todas las debilidades que puedan ser aprovechadas para un ataque. Estas debilidades deben ser agrupadas para crear soluciones que impidan violaciones de seguridad. Lleva a cabo la autorización, autenticación y auditorias al sistema. Proporciona un modelo arquitectónico coherente en el cual se pueden integrar distintas soluciones y se pueden realizar evoluciones y actualizaciones de forma controlada y con mínimos riesgos.

- Vista de la Arquitectura de Presentación

Como luce el software, cuáles son los colores que lleva la aplicación, como son los botones, los *link*. Tiene la responsabilidad como disciplina de definir y desarrollar las tecnologías para la representación de la información y usabilidad de esta. Entre las responsabilidades que cubre se encuentran los aspectos tecnológicos asociados a garantizar atributos de calidad en el producto como son los aspectos de rendimiento, usabilidad, reutilización y portabilidad de la capa de presentación en la arquitectura. (Verdecia, y otros, 2009)

- Vista de la Arquitectura de Tecnología

Es la base del software, propicia los elementos necesarios para crear el producto. Tiene como objetivo especificar y describir las tareas y competencias de los roles así como las actividades y *artefactos* del área de tecnología, y por tanto, definir las habilidades y los conocimientos que deben desarrollar las personas que trabajan en esta área de la ASW para lograr un buen desempeño. Es la responsable de garantizar un soporte tecnológico para el desarrollo de las configuraciones que propone el resultado de una arquitectura de sistema, así como las bases tecnológicas para los frameworks especializados de la arquitectura de sistema o de otras áreas como la de integración. Esta es el área responsable de identificar las tecnologías y herramientas a usar en la realización de la aplicación, además de definir la factibilidad técnica del producto.

Para tres de estas vistas (AS, AD, AT) el centro tiene definido métodos de construcción de las mismas, los cuales se encuentran formalizados en los libros de procesos que se describen en el CEIGE. Estas tres

vistas tienen un gran impacto en el ciclo de vida del proyecto específicamente en la disciplina de diseño, pues asentando las bases en una correcta definición de la línea base, un buen diseño de los componentes y del rendimiento esperado del servidor tendrá éxito la implementación del proyecto.

Existen métodos para construir y evaluar la arquitectura, sin embargo la gestión de la arquitectura está relacionada con las características propias que presente el software y de cómo se implemente la arquitectura en el mismo, para ello es necesario el estudio de los diferentes modelos de desarrollo de software que existen en la actualidad.

MODELOS DE PROCESOS O DESARROLLO DE SOFTWARE

Cuando se realiza el desarrollo de un producto de software, se llevan a cabo una serie de actividades o tareas, entre la idea inicial y el resultado final: "el producto".

*Sommerville*⁶ define modelo de proceso de software como "Una representación simplificada de un proceso de software, representada desde una perspectiva específica. Por su naturaleza los modelos son simplificados, por lo tanto un modelo de procesos del software es una abstracción de un proceso real."

Según *Letelier*⁷ un modelo de proceso de software es una representación simplificada de un proceso de software que conlleva una estrategia global para abordar el desarrollo de software.

Un modelo de desarrollo establece el orden en el que se crearán las cosas en el proyecto, provee los requisitos de entrada y los de salida para cada una de las actividades.

Es importante enfatizar que el ciclo de vida del proyecto ayuda a controlar las actividades del proyecto, desde su inicio hasta el fin del mismo, donde el modelo de desarrollo ayuda a la forma en que será construido el producto. Con la integración de ambos se genera el producto desde el punto de vista técnico y administrativo.

A continuación se darán a conocer las principales modelos de ciclo de vida que existen.

1.1 Tipos de modelos de desarrollo de software

Entre los principales modelos de desarrollo se encuentra *Codificar y Corregir (Code-and-Fix)*, el cual en los inicios del desarrollo de software fue el modelo básico utilizado. Consta solamente de dos pasos: escribir código y corregir los problemas en el mismo. Es aplicable a desarrollos muy pequeño con claridad de objetivos, requerimientos pequeños o de mantenimientos con bajo impacto.

⁶ Profesor de Ingeniería de Software de la Universidad de StAndrews en Escocia. Autor de un libro de texto ampliamente utilizado en la Ingeniería de Software, actualmente en su 8^{va} edición.

El *Modelo de Cascada* es otro de los primeros modelos que surgió en el desarrollo de software, este describe un orden secuencial en la ejecución de los procesos que tenga asociados, no acepta cambios en los requisitos. Consta de cinco fases: definición de los requisitos, diseño de software, implementación y pruebas unitarias, integración y pruebas del sistema, operación y mantenimiento. Es aplicable cuando existen especificaciones amplias de los requerimientos del cliente. (ALLSOFT S.A.) (Arboleda Jiménez)

El modelo basado en el *Desarrollo evolutivo* tiene como idea 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. Tiene como ventaja una rápida retroalimentación del usuario, ya que las actividades de especificación, desarrollo y pruebas se ejecutan en cada iteración. Es más efectivo que el *Modelo en Cascada*, pues cumple con las necesidades inmediatas del cliente. Los desarrolladores y usuarios logran un mejor entendimiento del sistema, lo que refleja una mejor calidad del producto. Es recomendable para proyectos pequeños o medianos, con poco tiempo para su desarrollo donde no se genere documentación para cada versión. Es aplicable cuando se requiere reemplazar el antiguo desarrollo con uno nuevo que satisfaga las nuevas necesidades según las redefiniciones del problema. Existen dos tipos de desarrollo evolutivo: *Desarrollo Exploratorio* y el *Enfoque Utilizando Prototipos*.

El *Modelo basado en Componentes* incorpora muchas de las características del *Modelo en Espiral*. Es un *Modelo Evolutivo*, exige un enfoque iterativo para la creación de software. Trae consigo la reutilización de software y esta a su vez proporciona beneficios a los ingenieros, brindando una reducción en el ensamblaje de componentes. Permite que las pruebas al software sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados. Permite que los componentes puedan ser construidos y luego mejorados continuamente lo que facilita a una mejor calidad del producto final.

El *Desarrollo Incremental* por Harlan D. Mill⁸. Este enfoque incremental de desarrollo surgió 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. Es una combinación del *Modelo de Cascada* y *Modelo Evolutivo*. Permite construir el proyecto en etapas incrementales en donde cada etapa agrega funcionalidad. Cada etapa consiste de requerimientos, diseño, codificación, pruebas y entrega, por lo que permite entregar al cliente un producto más rápido en comparación con el *Modelo en Cascada*. Permite que las pruebas e integraciones sean constantes.

⁷ Patricio Letelier Torres profesor titular de la Universidad Politécnica de Valencia.

⁸ Harlan D. Mills: Fundador de la Ingeniería de Software. Fue profesor de ciencias de la computación en el Instituto de Tecnología de la Florida.

Otro modelo es en *Espiral*, este surgió con el objetivo de darle una alternativa al *Modelo en Cascada*. Tanto el *Modelo en Cascada* como el *Modelo en Espiral*, no tienen en cuenta que los requerimientos del cliente pueden tener cambios en el transcurso del desarrollo del software. Se representa como una espiral, en lugar de una serie de actividades sucesivas, con retroceso de una actividad a otra. Cada ciclo de su desarrollo se divide en cuatro fases: definición de objetivos, evaluación y reducción de riesgos, desarrollo y validación. (Arboleda Jiménez) (ALLSOFT S.A.)

Ver Anexo 2. Comparación entre los modelos de proceso de software donde se expone un cuadro comparativo teniendo en cuenta algunos criterios básicos para la selección de un modelo de proceso.

METODOLOGÍA EN EL DESARROLLO DE SOFTWARE

Un proceso de software completo y detallado suele denominarse *Metodología*, estas se basan en la combinación de los modelos de proceso genéricos (*cascada*, *evolutivo* e *incremental* etc.) se hace necesario para el desarrollo de esta investigación, un estudio de las mismas.

Una metodología debe definir con claridad los artefactos, roles así como las actividades, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, así como una guía para el uso adecuado de las herramientas de apoyo, etc.

Patricio Letelier define que la metodología en un proyecto de desarrollo de software define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

Considerando la filosofía de desarrollo, las metodologías con mayor énfasis en la planificación y control del proyecto en cuanto a descripción de requisitos y modelado, reciben el nombre de Metodologías Tradicionales o Pesadas. Otras metodologías orientadas a la generación de código con ciclos cortos de desarrollo utilizadas en equipos de desarrollo pequeños son denominadas Metodologías Ligeras o Ágiles. Estas hacen hincapié en aspectos humanos asociados al trabajo en equipo e involucran de forma activa al cliente en el proceso. (Canós, y otros)

A continuación se estudia brevemente cada una de las categorías de las metodologías.

1.1 Metodologías tradicionales o pesadas

Las metodologías tradicionales tiene como principal característica la planeación con resultados predecibles o medibles de cada proceso que se realice. Sin embargo se torna difícil, pues no es posible predecir desde el comienzo de un proyecto los resultados que se van a obtener. Esto está dado a las

características que presenta cada software, pues estas pueden ser semejantes pero nunca iguales. Estas metodologías generan grandes cantidades de artefactos, están más bien orientadas al desarrollo de software donde el ciclo de vida de proyectos sea largo.

Dentro de estas metodologías se destaca el:

Proceso Unificado de Desarrollo (RUP): Es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo. Abarca más allá del análisis y el diseño, es orientada a objetos facilitando el uso de técnicas que soportan el ciclo completo de desarrollo de software. Tiene como principales características:

- ✓ Guiado por los casos de uso: Los casos de uso son los instrumentos principales para validar la ASW y realizar los casos de prueba.
- ✓ Centrado en la arquitectura: Los modelos son las proyecciones del análisis y el diseño pues componen la arquitectura del fruto a desarrollar.
- ✓ Iterativo e incremental: A través de todo el proceso de desarrollo se realizan versiones incrementales, es decir se acercan cada vez más al producto terminado, del producto en desarrollo.

1.2 Metodologías ligeras o ágiles

El término ágil surge en Febrero de 2001, el surgimiento fue dado al realizarse una reunión en Utah-EU donde participan expertos de la industria del software, entre ellos, creadores e impulsores de metodologías de software. El objetivo, era idear principios que facilitaran y respondieran a los cambios en la vida de desarrollo del proyecto, y que así permitieran el trabajo al equipo de desarrollo. Con esta reunión se creó *The Agile Alliance*, organización que se dedica a promover los conocimientos sobre el desarrollo ágil de software, así como brindar ayuda a aquellas organizaciones para que adopten dichos conocimientos.

Las metodologías ágiles están dirigidas al desarrollo de software, donde el ciclo de vida sea corto teniendo como principal habilidad la respuesta ante los cambios en los requisitos o a lo largo del desarrollo del proyecto. Son bastante flexibles ya que no existe un contrato tradicional, el cliente forma parte del equipo de desarrollo. Están conformadas por grupos de menos de 10 integrantes y trabajando en el mismo sitio. Poseen menos artefactos al igual que pocos roles, y tienden a tener menos acento en la arquitectura de software.

Entre las metodologías ágiles se encuentran:

- ✓ *Extreme Programming (XP)*: Es una de las metodologías más difundidas, pues define pocas prácticas y pocas reglas. Es adaptable a los procesos debido a los principios y prácticas que presenta. Quienes trabajan con esta metodología deben seguir procesos disciplinados y a la vez combinarlos con la adaptabilidad del proceso. Las prácticas en que se basa esta metodología hacen que el desarrollo del producto sea lo más ligero y exacto posible, entre las cuales se encuentra: el juego de la planificación, versiones pequeñas, metáforas del sistema, diseño sencillo, pruebas continuas, programación en parejas, propiedad colectiva del código, 40- horas de trabajo a la semana, cliente in-situ, estándares de codificación. (Jacobson, y otros)

- *Ventajas y desventajas*

Esta metodología puede ser implementada en forma parcial, es decir utilizar algunas de sus características, lo que facilita que pueda ser adaptable a cualquier equipo de desarrollo. Exige una comunicación más fluida con el cliente, que este a su vez tenga mayor participación en el proceso de desarrollo del producto. Se realizan pruebas constantemente al sistema. Sin embargo no es una metodología para equipos de trabajo grandes(a lo sumo 15). No está ampliamente probado ya que es una metodología nueva. Su equipo de programadores tiene que ser altamente especializados o con una gran experiencia.

- ✓ *SCRUM*: Promovida por *Ken Schwaber, Jeff Sutherland y Mike Beedle*. Esta metodología puntualiza un marco para la gestión de proyectos. Está dirigida a aquellos proyectos en el cual sea necesario el cambio rápido de requisitos. Tiene como características principales, el desarrollo de software a través de iteraciones de 30 días de duración, llamadas *sprints*, donde el resultado de cada sprint es un incremento ejecutable que se le muestra al cliente. Otra característica que presenta e importante son las reuniones a lo largo del desarrollo del proyecto, principalmente las reuniones diarias de 15 minutos con todo el equipo, para realizar ajustes de coordinación e integración. (Serrano)

- *Ventajas y Desventajas*

Esta metodología facilita la entrega de un producto funcional al finalizar cada Sprint. Permite la visualización del proyecto día a día evitando estancamiento en el desarrollo del producto, así como un seguimiento y comunicación del equipo. Alcance acotado y viable. Aumenta la comunicación con el equipo. Evita estancamientos en el proyecto. Sin embargo no genera toda la documentación que se generan en otras metodologías. No es apto para todos los proyectos. Tal vez sea necesario

complementarlo con otros procesos, o con otras características de otras metodologías como XP. (Alvarez M.)

- ✓ *Crystal Methodologies*: Desarrollada por *Alistair Cockburn*. Esta metodología plantea la existencia de tipos diferentes de proyectos, los cuales requieren metodologías diferentes. Es un conjunto de metodologías de desarrollo de software, las cuales se centran en los integrantes del equipo y la reducción del número de artefactos. Depende de dos factores fundamentales: el número de personas y las consecuencias de los errores, por lo que es imprescindible el papel que juega el equipo de desarrollo. Para lograr el resultado trazado se realizan grandes esfuerzos, orientados a limar habilidades y políticas de trabajo a seguir, en conjunto, como equipo. Estas políticas tienen gran relación, en cuanto al tamaño del equipo, por lo que definen colores para su clasificación. Ejemplo: *Crystal Clear* tiene de 3 a 8 integrantes, *Crystal Orange* tiene de 25 a 50 integrantes.
- ✓ *Dymanic Systems Development Method (DSDM)*: Surge en 1994 con la idea de crear una metodología RAD⁹ unificada. Puntualiza el marco para desarrollar un proceso de producción de software. Se caracteriza por un proceso iterativo e incremental, donde el usuario como el equipo de desarrollo trabajan en conjunto. Plantea cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción e implementación. Siendo estas tres últimas iterativas, además de existir retroalimentación a todas las demás fases. (Canós, y otros)
- ✓ *Adaptive Software Development (ASD)*: Fue promovida por *Jim Highsmith* y *Sam Bayer* a comienzos de 1990. Tiene como principales características: orientado a los componentes, más que a las tareas y a los cambios que puedan ocurrir en el desarrollo del proyecto, además de ser iterativo. Para su ciclo de vida propone tres fases fundamentales: especulación, colaboración y aprendizaje. En la primera fase, se da inicio al proyecto, donde se planifican las características que este va a tener, en la segunda se desarrollan estas características, y en la tercera, finalmente se revisa lo antes desarrollado, es decir, la calidad y se le da entrega al cliente. Con la revisión de los componentes se asimila los errores y se vuelve a iniciar el ciclo de desarrollo.

- *Ventajas y Desventajas*

Anticipa cambios y trata automáticamente con ellos dentro de un programa en ejecución, sin la necesidad de un programador. Difunde la colaboración entre los integrantes, es decir la interacción de personas. Sin embargo no está dirigido a aquellas empresas grandes, donde sea necesario un control estricto de los procesos y de las personas.

- ✓ *Feature Driven Development (FDD)*: Fue inspirada por *Jeff de Luca* y *Peter Coad* a finales de 1990. Planea un proceso iterativo el cual tiene cinco pasos. Se centraliza en las fases de diseño e implementación, partiendo de las características que debe tener el software. Las iteraciones que ejecuta son cortas de dos semanas (como máximo).
- ✓ *Lean Developmet (LD)*: Fue definida por *Bob Charette's*, en los años 80. Después de tener experiencia en proyectos de industria japonesa del automóvil, utilizada en proyectos de telecomunicaciones en Europa, Bob da surgimiento a esta metodología, la cual tiene como característica implantar un mecanismo para implementar cambios. Estos son considerados como riesgos para el desarrollo del proyecto, sin embargo si se realiza un manejo adecuado de estos, pueden ser oportunidades para perfeccionar la productividad del cliente. (Canós, y otros)
- ✓ *Agile Unified Process (AUP)*: *Scott Ambler* poniendo su empeño en la mejora de Rational Unified Process en 1999 desarrolla Enterprise Unified Process (EUP). A partir de 2001 se intenta imprimir aspectos ágiles y en 2005 se libera la primera versión de AUP. Agile Unifed Process a través del uso de técnicas ágiles y aplicando las mejores características de RUP, AUP de forma simple y fácil describe el desarrollo de software funcional. Entre las técnicas comunes que aplica se encuentra el diseño conducido por pruebas, desarrollo dirigido por modelos ágiles, gestión ágil de los cambios, técnicas de refactorización de la base de datos para mejorar la productividad. Define cuatro fases al igual que RUP (Inicio, Elaboración, Construcción, Transición) y siete disciplinas: Modelo, Puesta en práctica, Prueba, Despliegue, Gerencia de la Configuración, Gerencia de Proyecto y Ambiente. (CGI)

Conclusiones Parciales

Con el estudio del estado del arte, se logró realizar un análisis sobre la Arquitectura de Software, así como los fundamentales modelos y metodologías del software existentes teniendo en cuenta sus principales características, ventajas y desventajas.

A través del estudio realizado, se ha podido apreciar que la información disponible presenta características importantes que se deben tener en cuenta para orientar y perfeccionar la producción; garantizando la obtención de excelentes resultados. Las metodologías estudiadas están orientadas al desarrollo del software y no a la gestión de los procesos, destacando algunas que si están orientadas a la gestión de los procesos pero la implantación de estas en su totalidad, no responde a las necesidades imperantes en el departamento de tecnología de nuestro centro. Se propone un modelo de gestión de la

⁹ Rapid Application Development (Desarrollo Rápido de Aplicaciones): proceso de desarrollo de software que comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE.

arquitectura, basado en los modelos Iterativo-Incremental y modelo basado en Componentes a partir de características adaptables de la metodología SCRUM y sobre la definición de las vistas de la arquitectura definidas en el Departamento de Tecnología, centrándose en las áreas de trabajo (vistas) de: Arquitectura de Sistema, Arquitectura de Datos y Arquitectura de Tecnología.

CAPÍTULO 2 MODELO DE DESARROLLO

INTRODUCCIÓN

El presente capítulo está dedicado a describir y explicar el modelo para la gestión de la arquitectura de un sistema de planificación empresarial. En este se definen los elementos que conforman el modelo, como el flujo de actividades, los artefactos que se generan por cada una de ellas así como los roles que intervienen y las responsabilidades asociadas a los mismos. La solución se basa en un modelo que puede ser utilizado en proyectos de desarrollo de software de gestión, donde la propuesta de la arquitectura, en el desarrollo del ciclo de vida del producto, tenga presente la vista de Arquitectura de Sistema (AS), la vista de Arquitectura de Datos (AD) y la vista de Arquitectura de Tecnología (AT).

¿CÓMO INTERVIENEN LAS VISTAS ARQUITECTÓNICAS EN EL CICLO DE VIDA DEL PRODUCTO?

El Departamento de Tecnología del CEIGE se centra en el desarrollo tecnológico, y se caracteriza fundamentalmente por la implementación de soluciones particulares (componentes), que forman parte de soluciones generales (software). Es importante tener en cuenta que para la gestión del desarrollo tecnológico, este está orientado por el ciclo de vida del proyecto, específicamente donde se describe el ambiente de desarrollo, se define la arquitectura del sistema y se realiza el diseño de cada componente del software, que responde a los requisitos impuestos por el cliente (la disciplina de Diseño). En el presente epígrafe se realiza un análisis profundo teniendo en cuenta las características particulares del proyecto, concentrándose principalmente en el ambiente de desarrollo del producto.

En la siguiente figura se muestra cómo influyen las vistas de AS, la vista de AD y la vista de AT en las dos primeras fases del ciclo de vida del proyecto.

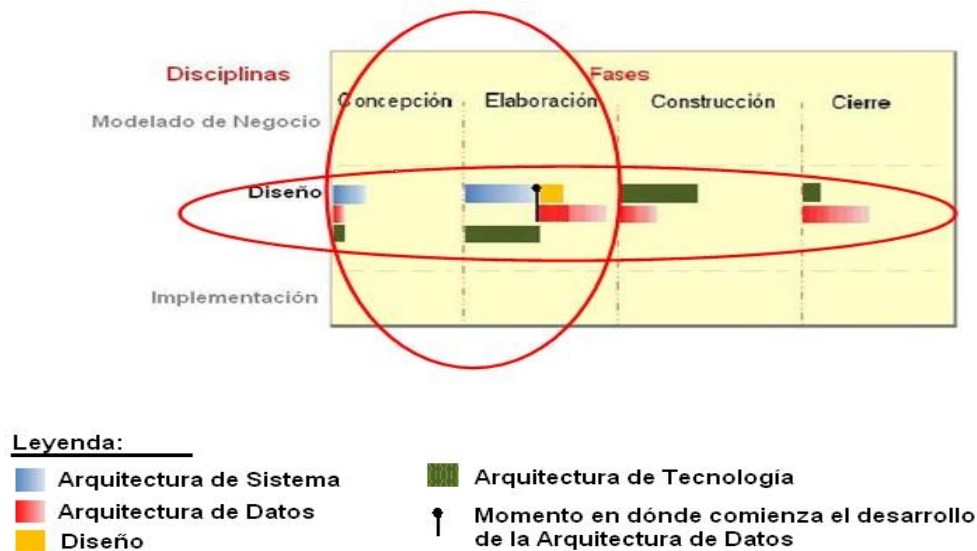


Figura 1. Comportamiento de las vistas de AS, AD y AT en las fases de Concepción y Elaboración.

2.1 Concepción

En las dos primeras fases del ciclo de vida del proyecto (Concepción y Elaboración), la vista de AS tiene un mayor peso; teniendo en cuenta que en la fase de Elaboración, la AS comparte su influencia con la AD y con la AT, estas últimas también tienen un gran peso en esta fase.

Retomando el análisis en la fase de Concepción, donde se realiza un flujo de actividades propias de la AS; se efectúa el análisis de los artefactos generados en la disciplina de Negocio; se identifican y delimitan responsabilidades de negocio a nivel macro, con el propósito de agrupar funcionalidades en subsistemas, módulos y componentes; se realiza el análisis de los atributos de calidad (AC), se identifican los escenarios arquitectónicos (EA) a partir de los requisitos de software (RS). Para garantizar la gestión de estas actividades mencionadas anteriormente se realizan talleres. Se realiza el agrupamiento de los requisitos funcionales identificándose las responsabilidades, los niveles de empaquetamiento así como las estructuras capaces de garantizar la mejor reutilización y colaboración posible.

La vista de AT y la vista de AD también tienen participación en esta fase, pero con un menor peso. Por su parte la AT realiza de manera global, definiciones de políticas de validación, estándares para la codificación y la integración de los componentes a la solución general, define la taxonomía de diseño para el desarrollo de interfaces, define métricas asociadas a los atributos de calidad y posibles pruebas que se pueden aplicar en la etapa de Elaboración y Construcción del software o componente.

También en la AD se definen de manera global un grupo de políticas de trabajo, elementos de la arquitectura y del diseño de la base de datos (BD); entre los que se encuentra la configuración técnica para llevar a cabo la configuración, la concurrencia y arquitectura en la herramienta a utilizar para el montaje de la BD; una serie de normas para realizar las consultas DQL¹⁰, las funciones y estándar SQL¹¹; se define la estructura del expediente; como se llevará a cabo la gestión de cambio, el versionado, el indexado, el mantenimiento de la BD así como la normalización, la nomenclatura y los comentarios; se define cómo llevar la seguridad de la BD y el acceso de los usuarios a la misma; se determina cómo llevar la configuración y la implementación de backup¹² en caso de pérdida de información o problemas de fallas; además se define cómo se llevará la integración con otros subsistemas.

¹⁰ DQL: Data Query Language, Lenguaje Estructurado de Datos.

¹¹ SQL: Structure Query Language, Lenguaje Estructurado de Consultas.

¹² Backup: se utilizan como sistema de salvadas en caso de pérdidas de información. Lo genera la herramienta que se utiliza para montar la BD.

2.2 Elaboración

Por su parte en la fase de Elaboración, la AS de manera general, identifica y diseña la configuración de los elementos de mayor abstracción que posee (los subsistemas, módulos y componentes); se realiza una propuesta candidata de los patrones de diseño a utilizar en los componentes a desarrollar; se identifican los estilos arquitectónicos y patrones arquitectónicos propicios, se examina el cumplimiento de los patrones GRAPS* en la propuesta del diseño arquitectónico; se identifican y formalizan en elementos arquitectónicos las principales funcionalidades horizontales para desarrollar todo el diseño arquitectónico; se realiza un análisis de reutilización de los elementos arquitectónicos del sistema así como de las bases tecnológicas que se encuentran almacenadas en los repositorios de componentes de reutilización; se diseñan los componentes más importantes de la arquitectura; se realiza el cálculo de la significación arquitectónica de los componentes del sistema, tomando en cuenta su impacto en la arquitectura y la prioridad de implementación de los mismos, se discute el plan de estimación y planificación* del proyecto, donde se realiza un análisis de la capacidad técnica del equipo, la disponibilidad tecnológica y productiva, además se analiza el cronograma de desarrollo e implementación de la línea base; se realiza una trazabilidad general, desde la disciplina de Negocio hasta los componentes y estructuras internas de los mismos, se revisa que todos los requisitos identificados estén respaldados por algún componente, es decir se realiza una revisión integral de la arquitectura de sistema; se desarrolla un taller para posibles ajustes previos de la AS propuesta, es decir la Línea Base de la Arquitectura (LBA) propuesta.

Se prosigue a “*realizar el diseño del modelo de datos*”; es en este punto (actividad*) donde se relaciona la vista de AS con la vista de AD. Desde este momento se inicia la etapa de desarrollo de la BD.

Se realizan actividades propias del diseño como identificar qué soluciones de reutilización se pueden emplear, partiendo del componente expresado en sus escenarios y requisitos; se elaboran los diagramas de clases para la capa de negocio; partiendo de los componentes arquitectónicamente significativos para la arquitectura se diseña un componente de prueba; se examina completamente el diseño a bajo nivel, realizando un refinamiento de los artefactos generados durante el diseño, persiguiendo como objetivo llevar a la fase de Construcción el diseño lo más claro y detallado posible.

Como se menciona anteriormente es en esta fase, donde la AS le da inicio al desarrollo de la BD. Para ellos se realiza el análisis y diseño del modelo y diccionario de datos, artefactos que definen el diseño de la BD y de las clases persistentes del negocio; se realiza la descripción del diseño estructural del modelo de datos generado en el diseño; se realizan pruebas de concepto al diseño para validar el diseño de los modelos; se realiza la clasificación de las tablas que conforman la BD.

En esta fase también tiene participación la vista de la AT, definiendo a partir de los AC y de los RF y no funcionales (RNF), los EA; se realiza un estudio de las herramientas, plataforma de desarrollo y

metodologías que se utilizan en la construcción de un software; se efectúa un estudio de reutilización con el objetivo de analizar los posibles componentes, o funcionalidades de estos que se puedan reutilizar; se realiza un prototipo de prueba con el objetivo de comprobar la factibilidad de los componentes, o funcionalidades de estos que se reutilizarán; se realiza el diseño de solución, por el cual se efectuará la integración de los componentes, o funcionalidades de estos que se van reutilizar; se realiza un taller de solución para analizar los resultados del estudio de reutilización y de las políticas de integración a seguir, el objetivo es especificar la solución a implementar; se realiza el diseño de la solución que posteriormente será implementada.

2.3 Construcción

En la siguiente figura se muestra cómo influyen las vistas de AD y la vista de AT en las dos últimas fases del ciclo de vida del proyecto* (Construcción y Cierre); donde estas vistas se reinciden.

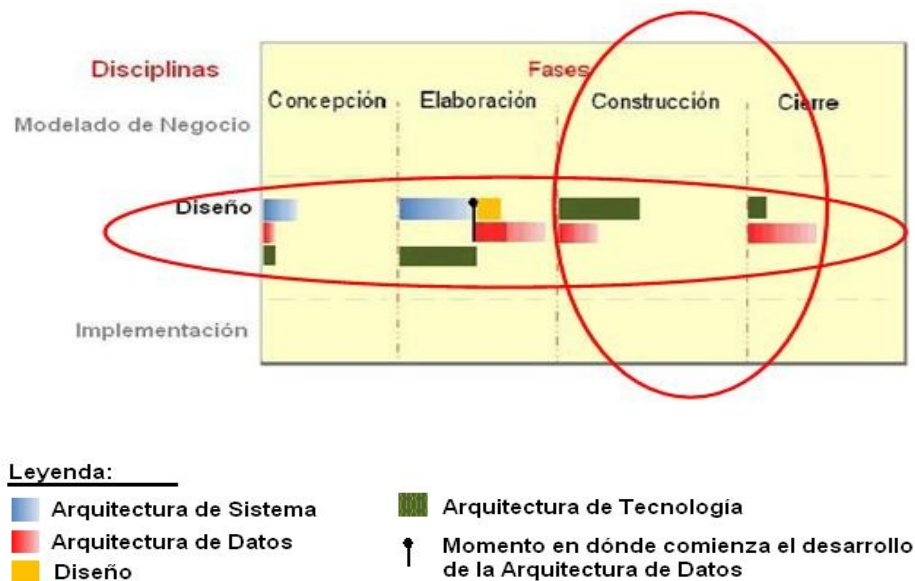


Figura 2. Comportamiento de las vistas de AS, AD y AT en las fases de Construcción y Cierre.

Analizando la fase de Concepción, la AT tiene un peso notable, donde se lleva a cabo la implementación de la solución propuesta en la fase de Elaboración; se realizan pruebas unitarias a esa solución propuesta; se realizan pruebas de *stress*¹³, pruebas de rendimiento y pruebas de seguridad en caso de que la solución propuesta lo requiera; se efectúa la integración de la solución particular a la solución general.

La AD como se menciona anteriormente tiene participación en esta fase, pero con menos peso que la AT. A grandes rasgos se procede a realizar la integración de los subsistemas; se efectúan pruebas de

¹³ Pruebas de stress o pruebas de resistencia, verifica hasta qué punto puede rendir el software antes de que falle.

conceptos para evaluar el rendimiento del servidor de BD; se realiza el cálculo del crecimiento de la BD, estos valores se registran y se puede prever un problema por capacidad en disco; se realiza el montaje de los script de instalación para el despliegue del producto, y de los script de actualización, los que constituyen actualizaciones que contribuyen a la mejora del producto en despliegue.

2.4 Cierre

Por su parte en la fase de Cierre, se destacan la vista de la AT y la vista de la AD.

La AT tiene influencia en esta fase del ciclo de vida del proyecto, donde de forma general realiza el cierre de la documentación que servirá de ayuda al usuario final; además de un caso de estudio.

La AD tiene una intervención de forma relevante en esta fase, pues influye con mayor peso que la AT; de manera general realiza el montaje de los scripts de actualización que contribuyen a la mejora del producto, además brinda soporte a incidencias que puedan ocurrir durante el despliegue.

2.5 Resultados

Después de haber realizado el análisis por cada una de las fases del ciclo de vida del proyecto y observando el comportamiento de las vistas de la AS, la vista de AD y la vista de la AT; se llega a la conclusión que el desarrollo de la arquitectura está regido por la Arquitectura de Sistema AS, donde en las dos primeras fases (Concepción y Elaboración) tiene mayor participación; destacando que en la fase de Elaboración su participación es compartida con las vistas de AD y la vista de AT, las cuales tienen gran influencia en esa fase. En la fase de Construcción tiene una mayor influencia la vista de AT y en la fase de Cierre la mayor influencia es por parte de la vista de AD. En la siguiente figura se puede apreciar gráficamente este comportamiento.

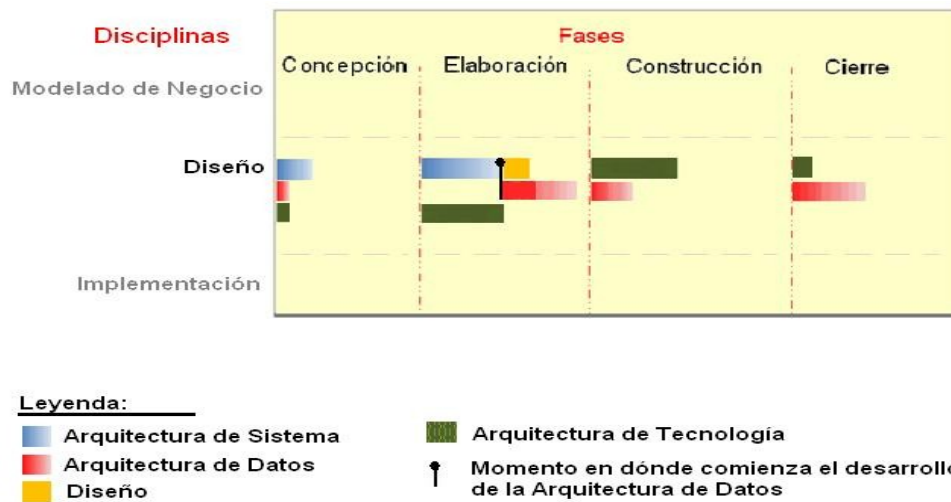


Figura 3. Comportamiento de las vistas de AS, AD y AT en el ciclo de vida del proyecto.

Después del análisis realizado se identificaron un grupo de actividades que forman parte del desarrollo y gestión de la arquitectura; estas actividades se caracterizan principalmente por el seguimiento de gestión que se les debe dar, permitiendo a través de este verificar su desarrollo tecnológico. En la siguiente figura se muestra el flujo de actividades.

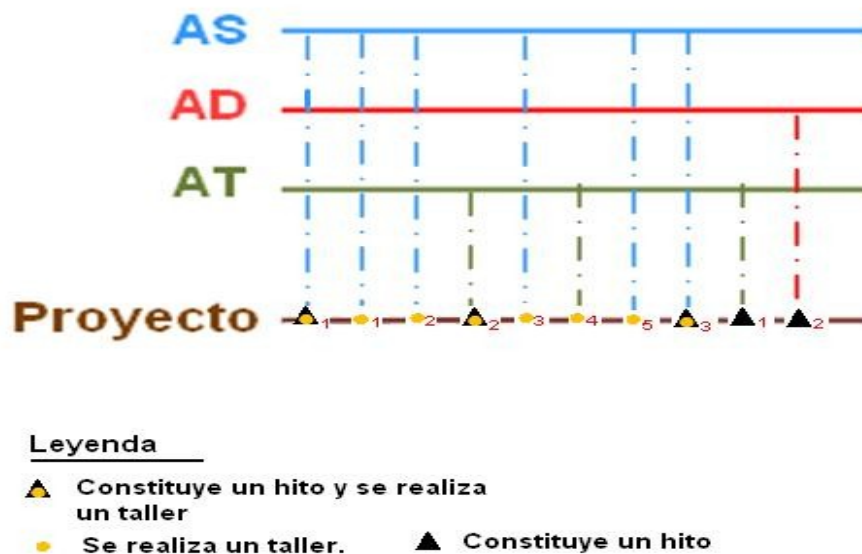


Figura 4. Flujo de actividades

▲₁: Analizar artefactos generados en el Negocio.

■₁: Identificar estructura de empaquetamiento.

- ₂: Identificar los atributos de calidad.
- ▲₂: Definir escenarios arquitectónicos.
- ₃: Identificar los componentes horizontales de la arquitectura.
- ₄: Realizar taller de solución.
- ₅: Discutir plan de estimación y planificación del proyecto.
- ▲₃: Realizar ajustes de la AS propuesta.
- ▲₁: Realizar pruebas unitarias a la solución.
- ▲₂: Evaluación del rendimiento del servidor.

ESTRUCTURA DEL MODELO

En el desarrollo de este epígrafe se expone la estructura que conforma el modelo propuesto, además de los elementos a tener presente para la implantación del mismo.

El modelo está basado en Sprint (iteraciones) preferentemente de tres meses donde se destacan tres fases fundamentales, Fase de inicio donde se define el alcance del proyecto, Fase de ejecución se define y establece la línea base de la arquitectura y por último la Fase de cierre donde se realiza una revisión del cumplimiento de los hitos*. Para el desarrollo y gestión* de estas fases se realiza el Taller de Conceptualización y las reuniones de Chequeo Tecnológico semanalmente donde se identifican los problemas que impidan seguir en el avance del cumplimiento de las tareas. Ver en Anexo 1. Diagrama del Modelo.

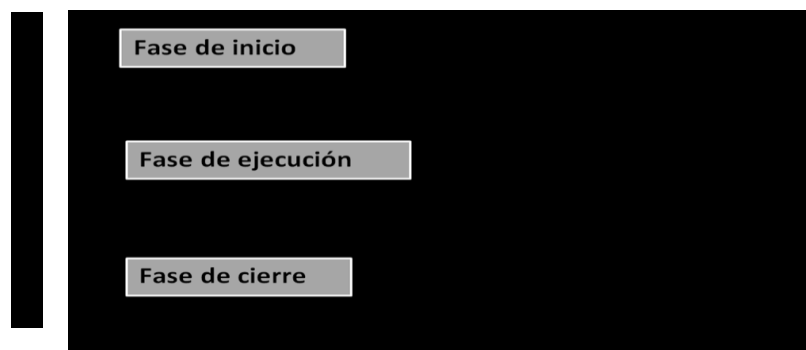


Figura 5. Estructura del modelo.

En cada fase está conformada por:

- ✓ Flujo de actividades.
- ✓ Artefactos.

- ✓ Roles.
- ✓ Herramientas* y técnicas.
- ✓ Desviaciones y riesgos.

2.6 Alcance del modelo

El modelo que se propone, puede ser aplicado a proyectos de desarrollo de software de gestión, donde la propuesta de la arquitectura, en el desarrollo del ciclo de vida del producto, tenga presente la vista de AS, la vista de AD y la vista de AT; además tienen que estar conformados por equipos de desarrollo medianos o grandes, y estos, a su vez, estructurados por pequeños equipos de trabajo.

2.6.1 Premisas y principios del modelo

Para que se lleve una correcta aplicación del modelo es necesario tener presente las siguientes premisas:

- ✓ Personal capacitado: se hace necesario de un personal capacitado y con experiencia en el desarrollo y gestión de la arquitectura.
- ✓ Trabajo en equipo: para lograr el éxito de este modelo es necesario que se fomente el trabajo en equipo, donde se puedan recibir ayuda de las personas que sean capaces de brindarla en el menor tiempo posible.
- ✓ Buena comunicación entre todo el equipo de desarrollo: esta premisa fomenta a obtener un mejor funcionamiento del modelo y anima el trabajo en equipo.
- ✓ Equipo de desarrollo caracterizado por ser de mediano a grandes, y organizado en pequeños equipos de trabajo donde no excede de más de 10 integrantes.

El modelo está basado en los principios de las Metodologías Ágiles, específicamente en los principios de SCRUM los cuales se plantean a continuación:

- ✓ Como principal prioridad está: satisfacer al cliente.
- ✓ Permitir cambios en los requerimientos, incluso tarde en el desarrollo.
- ✓ Tener presente que un software que funcione es la principal medida de progreso en el desarrollo del producto, por lo que se realizan entregas frecuentes de software funcional en escalas cortas de tiempo (semanas hasta un par de meses). (¿Estoy o no estoy usando Scrum?)

2.6.2 Características del modelo

El modelo que se propone está basado en algunas prácticas y características de SCRUM como:

- ✓ Sprint Backlog: este artefacto de la metodología SCRUM, contiene el listado de tareas que se debe realizar para la(s) iteración(es) (*Sprint*) en un período de tiempo definido; estas actividades

contribuyen al cumplimiento de hitos o de las mismas actividades planificadas, marcando en su avance funcionalidades al desarrollo del producto final.

- ✓ Reunión de SCRUM: estas reuniones se adaptaron a las características del Departamento de Tecnología. (¿Estoy o no estoy usando Scrum?)

2.7 Descripción del modelo

Para un correcto entendimiento de la estructura del modelo propuesto se describe detalladamente en este epígrafe cada uno de los elementos que conforman cada una de las fases del modelo.

2.7.1 Flujo de actividades

Se especifica el flujo de actividades a desarrollar, teniendo en cuenta los hitos a cumplir en cada una de las fases del ciclo de vida del proyecto, teniendo presente que el desarrollo de las vistas de AS, vista de AD y vista de AT, un atraso en el cumplimiento de estas tareas implica atraso en la planificación del proyecto provocando además un atraso en la etapa de implementación del producto.

Tabla 1. Flujo de actividades de la Fase de Inicio.

Fase de Inicio					
Objetivo	Actividad	Descripción	Artefactos de entrada	Artefactos de salida	Rol
Establecer el alcance del proyecto.	Realizar estimación y planificación del proyecto.	En esta actividad se reúne el equipo de trabajo y se realiza una planificación del proyecto.	[No posee.]	Sprint Backlog. Planificación del Proyecto. Minuta de acuerdos.	Arquitecto Principal. Líder de Gestión. Arquitecto de Sistema.
	Analizar artefactos generados en el Negocio.	Se realiza un estudio y análisis preciso de los artefactos generados en el Negocio.	Mapa de procesos. Descripción de los procesos de Negocio. Modelo conceptual. Listado de requisitos. Descripción de los requisitos. Glosario de términos. Visión del proyecto.	Minuta de acuerdos.	Arquitecto de Datos. Arquitecto de Tecnología. Analista.

Tabla 2. Flujo de actividades Fase de Ejecución.

Fase de Ejecución					
Objetivo	Actividad	Descripción	Artefactos de entrada	Artefactos de salida	Rol
Definir y establecer la Línea Base de la Arquitectura.	Identificar estructura de empaquetamiento.	En esta actividad se identifican y delimitan las responsabilidades de negocio a nivel macro (agrupar futuras funcionalidades en subsistemas, módulos y componentes).	Mapa de Procesos. Descripción de procesos de Negocio.	En esta actividad se modifica el artefacto: Línea Base de Proyecto (Estructura de empaquetamiento y breve descripción). Minuta de acuerdos	Arquitecto Corporativo o Principal. Arquitecto de Sistema. Arquitecto de Datos. Arquitecto Temático o de Componentes.
	Identificar los atributos de calidad.	En esta actividad se realiza el análisis de los atributos de calidad (AC) a favorecer con el modelo de la solución. Se identifican los escenarios arquitectónicos (EA), a partir de los requisitos de software (RS).	Listado de requisitos.	En esta actividad se modifica el artefacto: Línea Base de Proyecto (Matriz de RS, EA, AC). Minuta de acuerdos.	Arquitecto Corporativo o Principal. Arquitecto de Sistema. Arquitecto de Tecnología. Arquitecto Temático o de Componentes.
	Definir escenarios arquitectónicos.	En esta actividad se definen los escenarios arquitectónicos que debe cubrir la solución.	Atributos de calidad. Requisitos funcionales. Requisitos no funcionales.	Minuta de acuerdos.	Arquitecto de Tecnología. Programador. Analista.
	Identificar los componentes horizontales de la arquitectura.	En esta actividad se identifican y formaliza en elementos arquitectónicos de las principales funcionalidades horizontales para todo el diseño arquitectónico.	Mapa de Proceso.	En esta actividad se modifica el artefacto: Línea Base de Proyecto (Relación de componentes horizontales, actualización del mapa de componentes). Minuta de acuerdos	Arquitecto Corporativo o Principal. Arquitecto de Sistema. Arquitecto de Datos.

Realizar taller de solución.	En esta actividad se precisa y se detalla la solución que posteriormente se va a diseñar.	Políticas de Integración. Informe de Reutilización. Informe Técnico.	Minuta de acuerdos	Arquitecto Corporativo o Principal. Arquitecto de Tecnología. Programador
Discutir plan de estimación y planificación del proyecto.	En esta actividad se propone un primer esquema del cronograma de implementación y desarrollo de la línea base.	Documento Línea Base de Subsistema.	Documento Aprobación de solución. Minuta de acuerdos.	Arquitecto Corporativo o Principal. Arquitecto de Sistema. Arquitecto de Datos. Arquitecto Temático o de Componentes. Jefe de Proyecto.
Realizar ajustes de la arquitectura de sistema propuesta.	En esta actividad se detallan aspectos y se realizan ajustes de la línea base propuesta.	Documento Línea Base de Subsistema. Documento Línea Base de Proyecto. Documento Aprobación de solución	Minuta de acuerdos.	Arquitecto Corporativo o Principal. Arquitecto de Sistema. Arquitecto de Datos. Arquitecto de Tecnología. Arquitecto Temático o de Componentes. Jefe de Proyecto.
Realizar pruebas unitarias a la solución.	En esta actividad se realizan las pruebas unitarias a la solución.	Código fuente. Políticas de pruebas. Informe técnico.	Informe de pruebas unitarias. En esta actividad se modifica el artefacto: Informe técnico con los resultados obtenidos en la aplicación de las pruebas unitarias.	Arquitecto de Tecnología. Administrador de la calidad. Programador.
Evaluación del rendimiento del	En esta actividad se realizan las pruebas	[No posee]	Pruebas de Concepto del	Arquitecto de Datos.

	servidor.	de conceptos para evaluar el rendimiento del servidor.		Servidor.	
--	-----------	--	--	-----------	--

Tabla 3. Flujo de actividades de la Fase de Cierre.

Fase de Cierre						
Objetivo	Actividad	Descripción	Artefactos de entrada	Artefactos de salida	Rol	
Revisión u cumplimiento de los hitos trazados en la Fase de Inicio.	Realizar chequeo tecnológico.	En esta actividad se realiza un chequeo tecnológico a gran escala de los hitos trazados en la fase de Inicio que tributan al cumplimiento del Sprint. Además se realiza un análisis de las desviaciones y riesgos que constituyeron atrasos en el cumplimiento de la planificación del proyecto, tomándolo como experiencias para próximas iteraciones.	Sprint Backlog. Plan de Proyecto.	Minuta de acuerdos.	Arquitecto Corporativo o Principal. Líder de Gestión. Arquitecto de Sistema. Arquitecto de Datos. Arquitecto de Tecnología.	

2.7.2 Artefactos

A continuación se nombran los artefactos que se generan durante la ejecución del modelo; estos pueden ser artefactos de otras disciplinas y que son necesarios para llevar a cabo el proceso de gestión de la arquitectura, o pueden ser artefactos que se generan durante el proceso de gestión y desarrollo de la arquitectura.

Artefactos de entrada de otras disciplinas

Mapa de Proceso: este artefacto se genera en la disciplina de Negocio. Constituye un artefacto de entrada para el desarrollo de la AS y de la propia disciplina de Diseño; contiene el listado de los procesos de negocio con la información que intercambian entre ellos y las relaciones existentes entre los mismos.

Descripción de procesos de negocio: este artefacto se genera en la disciplina de Negocio. Constituye también un artefacto de entrada para el desarrollo de la AS y de la propia disciplina de Diseño; describe el flujo de actividades de los procesos de negocio.

Modelo Conceptual: este artefacto constituye también un artefacto de entrada para el desarrollo de la AS y de la propia disciplina de Diseño; contiene los diagramas de clases, el diagrama del modelo conceptual, además del diccionario de datos. Este artefacto es generado por las disciplinas de Negocio y Requisitos con el diccionario de datos que abarca cada una de estas disciplinas de manera independiente.

Descripción de requisitos: este artefacto se genera en la disciplina de Requisitos. Constituye un artefacto de entrada para el desarrollo de la AS y de la propia disciplina de Diseño; contiene una descripción de los requisitos funcionales, es decir lo que debe hacer el software.

Glosario de términos: este artefacto se genera en la disciplina de Requisitos. Constituye un artefacto de entrada para el desarrollo de la AS y de la propia disciplina de Diseño; se encarga de establecer un vocabulario común para todos los participantes en el proyecto.

Listado de requisitos: este artefacto se genera en la disciplina de Requisitos. Constituye un artefacto de entrada para el desarrollo de la AS y de la propia disciplina de Diseño; contiene el listado de requisitos de los proveedores.

Artefactos generados en el desarrollo de la arquitectura

Línea Base de Subsistema: este artefacto define las responsabilidades arquitectónicas de un subsistema. A través de este artefacto se establece la comunicación y dependencias entre los componentes y módulos que integran el subsistema. Además de plasmar la criticidad y complejidad de los componentes con futuras vistas a su reutilización y prioridad en el desarrollo del producto. El Arquitecto Temático o de Componentes es el encargado de realizar este artefacto.

Línea Base de Proyecto: este artefacto contiene una vista y descripción de forma global de las partes que integran o conforman el sistema. Contiene diagramas y matrices que establecen la integración y dependencias entre los distintos niveles de abstracción. Este artefacto establece la horizontalidad, prioridad así como las responsabilidades de cada nivel. El Arquitecto de Sistema es el encargado de realizar este artefacto.

Aprobación de Solución: este artefacto establece la aprobación y consentimiento de las líneas de desarrollo, en correspondencia con el diseño arquitectónico propuesto. El Arquitecto de Sistema es el encargado de elaborar este artefacto y el Jefe de Proyecto lo aprueba.

Pruebas de Concepto al Servidor: este artefacto contiene la configuración del gestor de BD. Refleja el crecimiento de la BD, mostrando una futura idea para prevenir la capacidad que debe tener el servidor de datos. El Arquitecto de Datos es el encargado de elaborar este artefacto.

Informe Técnico: este artefacto contiene la factibilidad técnica y económica de la solución propuesta, además de la información del estado del arte para el desarrollo de la solución a desarrollar. El Arquitecto de Tecnología es el encargado de elaborara este artefacto.

Informe de Pruebas Unitarias: este artefacto contiene el resultado de las pruebas unitarias realizadas a la solución propuesta. El encargado de elaborara este artefacto es el Administrador de la Calidad en conjunto con el Programador.

Artefactos para la gestión de la arquitectura

Minuta de acuerdos: este artefacto contiene los puntos tratados, acuerdos tomados, participantes, lugar y tema abordado en el desarrollo del taller de conceptualización. El Líder de Gestión es el encargado de elaborar este artefacto. Ver Anexo 8. Plantilla del artefacto minuta de acuerdos.

Sprint Backlog: este artefacto contiene el listado de hitos que se debe realizar para la(s) iteración(es) (Sprint) en un período de tiempo definido. El Líder de Gestión es el encargado de elaborar y darle seguimiento a este artefacto. Ver Anexo 9. Plantilla del artefacto Sprint Backlog

Planificación del proyecto: este artefacto contiene un cronograma de tareas a cumplir para el desarrollo del proyecto, incluye sistema de trabajo. Contiene la información de los recursos con los que se cuenta para efectuar la planificación.

2.7.3 Roles

En esta epígrafe se resumen de los roles que han sido definidos para llevar a cabo el modelo propuesto. Estos roles con el apoyo de un grupo de involucrados internos, son los que intervienen en el desarrollo de la arquitectura por cada una de las vistas de AS, la vista de AD y la vista de AT. A continuación se define la estructura de los roles que llevan el proceso de gestión de la arquitectura.

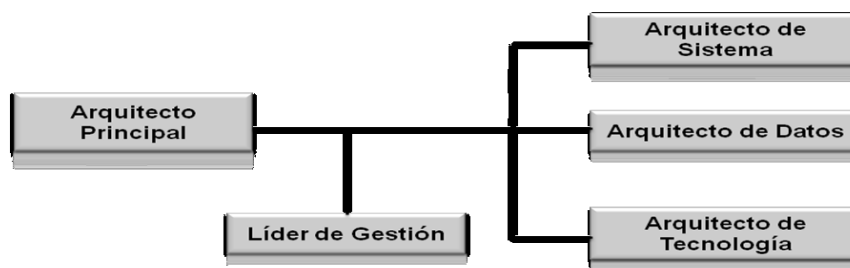


Figura 6. Roles propuestos por el Modelo.

Roles propuestos por el modelo

Arquitecto Corporativo o Principal: es el encargado de identificar las tareas de implementación, las necesidades tecnológicas, los hitos tecnológicos y la línea base. Es el máximo responsable de las decisiones arquitectónicas. Es el encargado de decidir cómo tratar los temas de reutilización así como los servicios tecnológicos al desarrollo y construcción tecnológica del producto. Orienta de forma estratégica la organización de la producción de software.

Arquitecto del Sistema: es el encargado de establecer y velar que se cumplan las políticas y estándares definidos en la AS que contribuyen al desarrollo de la línea base, además lleva las decisiones de integración en el proyecto.

Arquitecto de Datos: es el encargado de construir la Vista de Datos, establece la integración de la Arquitectura de Sistema en la dimensión de la AD. Vela por el cumplimiento de los elementos metodológicos de desarrollo, políticas, estándares, diseño y configuración definidos para la BD.

Arquitecto de Tecnología: es el encargado de estudiar y determinar la factibilidad técnica desde el punto de vista de tecnología del producto. Define y especifica la arquitectura de tecnología. Realiza el estudio tecnológico de las plataformas, framework y realiza el dictamen técnico de las mismas. Define el estilo arquitectónico vertical de la arquitectura, es decir la vista de tecnología.

Líder de Gestión: es el encargado del monitoreo y control de las tareas, la gestión del capital humano, la gestión de riesgos e incidencias. Lleva la planificación y gestión de los recursos del proyecto. Además de gestionar las necesidades técnicas del proyecto.

Involucrados internos en el desarrollo de la arquitectura

Analista: es el encargado de identificar y describir los requisitos de los interesados, establece los requisitos de los componentes del producto, estos pueden ser de los componentes de negocio o de los componentes de soporte. Aparte de de identificar y describir las interfaces entre los componentes participa en la ubicación de requisitos por componentes a desarrollar, valida los requisitos y determina el tamaño y la complejidad de los mismos.

Jefe de Proyecto: es el encargado de gestionar los recursos humanos y materiales asignados, planifica las iteraciones y versiones del producto; es el responsable de garantizar el cumplimiento del cronograma de trabajo y de los compromisos de la línea; además de guiar el proceso de identificación y mitigación de los riesgos.

Arquitecto de Componente o Temático: es el encargado de identificar, definir y formalizar los componentes de la línea temática que especializa; define la línea base de la línea temática que especializa; prioriza los

componentes y define los estándares, normas, protocolos para el desarrollo arquitectónico de la línea temática en que se especializa.

Programador: es el encargado de diseñar e implementar la lógica de negocio, construye las especificaciones del sistema en código fuente ejecutable, lleva a cabo la integración de los componentes que constituirán funcionalidades del producto final. Analiza, registra y ejecuta los casos de prueba.

Administrador de la calidad: es el encargado de gestionar las pruebas ya sean internas, de liberación o de aceptación; lleva el control de la documentación, revisa las políticas de control de cambios, control de errores y control de configuración, gestiona las revisiones y auditorías.

Ver Anexo 3. Competencias para cada rol que propone el modelo.

2.7.4 Herramientas y técnicas

En este epígrafe se enuncian las técnicas y herramientas que se aplican al modelo propuesto.

Talleres

Los talleres de conceptualización consisten en la reunión de un grupo de personas con conocimiento respecto al tema a tratar. Estos talleres están dirigidos a estudiar y analizar problemas, donde de manera conjunta se discuten o proponen soluciones, además de definir normas estándares y políticas a utilizar posteriormente en el desarrollo. Para el desarrollo de estos, se combinarán actividades como el trabajo en grupo, se realizarán sesiones generales donde se incluirá la elaboración y presentación de documentos, actas e informes; además de una organización y ejecución de trabajos a través de comisiones. Se fomentará el trabajo en grupos pequeños, donde se garantice un espacio que permita la movilidad de los participantes para que puedan trabajar con facilidad y se desempeñen de forma ordenada. Estos talleres están dirigidos a aquellas actividades donde sea necesaria la participación tanto de los roles propuestos por el modelo como de los roles involucrados en el proceso de desarrollo de la arquitectura.

Reuniones

Como parte de las características del modelo propuesto se encuentra la reunión de SCRUM; estas reuniones fueron adaptadas a las peculiaridades del Departamento de Tecnología; desarrollándose “reuniones de chequeo tecnológico”. Para llevar el desarrollo de estas reuniones se realizan encuentros semanales y reuniones trimestrales; con el objetivo de verificar el cumplimiento de las tareas y objetivos estratégicos planificados en el período, donde se identifican posibles desviaciones y riesgos que afectarán el cumplimiento de las tareas y traerá como consecuencia el atraso en el desarrollo del Sprint.

Los encuentros semanales se efectúan a principio de semana (lunes), donde están presentes los Jefes de cada equipo, el Jefe del Departamento y el Líder de Gestión; entre los objetivos a cumplir en estos encuentros está:

- ✓ Chequeo de las tareas a partir de la herramienta de monitoreo y control (Redmine).
- ✓ Revisión del estado de los componentes y de su documentación.
- ✓ Análisis de las desviaciones que imposibilitaron el avance en el desarrollo de los componentes.
- ✓ Análisis de las tareas atrasadas.

Diariamente el Jefe de cada Equipo tiene la tarea de chequear y asesorar el trabajo de los integrantes de su equipo, donde al final del día inspecciona el cumplimiento de las tareas planificadas; en caso de que detecte alguna desviación lo sube para el Redmine y lo informa de inmediato al Jefe de Departamento, para tomar medidas que no interrumpan el desarrollo del proyecto.

Las reuniones al es donde se desarrolla el cierre de los hitos acordados en la Fase de Inicio verificando su total cumplimiento. Se identifican los hitos que no fueron cumplidos, se definen los hitos a realizar en el próximo Sprint donde se pueden adicionar nuevas tareas a cumplir, se pueden refinar las que estaban planificadas anteriormente. Participan todos los profesionales del departamento.

Herramientas

En este epígrafe se mencionan las herramientas que se utilizan en la ejecución e implantación del modelo propuesto.

- ✓ *Procesadores de texto*: estos garantizan la elaboración de los documentos.
- ✓ *Redmine*: es una herramienta de monitoreo y control que permite llevar la gestión del proyecto. Permite definir los hitos y las tareas a realizar para cumplir cada uno de ellos, así como asignar esas tareas a cada uno de los responsables de darle cumplimiento. Posibilita llevar el tiempo de estimación, cronogramas de trabajo, el tiempo dedicado y el porcentaje de avance de las tareas asignadas; además del control de incidencias y riesgos.
- ✓ *Alfresco*: es una herramienta que facilita la búsqueda y obtención de documentos, gestiona el ciclo de vida de estos documentos en un repositorio centralizado. Representa un sistema de gestión documental en las empresas garantizando la seguridad y permitiendo la colaboración entre equipos.

2.7.5 Gestión de riesgos y desviaciones

En este epígrafe se exponen las posibles desviaciones que pueden constituir riesgos en el avance de entregas o del propio desarrollo del software. Además de cómo se puede llevar la gestión de las mismas. Para un mejor entendimiento respecto a este tema podemos preguntarnos:

¿Cómo se identifican las incidencias y los riesgos?

El modelo propuesto, posee una serie de reuniones a desarrollar en la que uno de los objetivos que se persigue es identificar cuáles son aquellos problemas que impiden o impedirán el avance en el desarrollo del producto. Estos problemas, constituyen desviaciones o riesgos. Hay desviaciones que no se pueden gestionar, ejemplo: caída de un servidor central; sin embargo los riesgos se pueden gestionar y se elaboran estrategias para mitigar su impacto. Las desviaciones son identificadas por los Jefes de Equipo en el transcurso de la semana.

¿Cómo se gestionan las desviaciones y los riesgos?

Después de haber identificado las desviaciones y los posibles riesgos; se prosigue a efectuar la gestión de los mismos a través del Redmine. Cada jefe de equipo sube las desviaciones y el líder de gestión va supervisando las mismas diariamente. En el caso de los riesgos es fundamental idear un plan de mitigación. El plan de mitigación de riesgos no es más que una serie de estrategias o medidas a establecer; estas garantizarán que el impacto de los incidentes ocurridos o de los futuros riesgos identificados sea leve; contribuyendo en ocasiones a que estas no constituyan un obstáculo para el desarrollo del producto, o simplemente permita recuperar el tiempo de atraso provocado por los mismos.

Problemas que pueden generar ocurrencias de desviaciones y riesgos

- ✓ Externos: este tipo de problema representa aquellos eventos o condiciones que se producen en un ambiente externo y no controlado por el proyecto. Estos problemas son difíciles de mitigar, en la gran mayoría solo se puede tener un plan de contingencia para que las afectaciones tengan un impacto mínimo para el avance del proyecto.
- ✓ Proyecto: son todos aquellos eventos o condiciones que amenazan al plan del proyecto; es decir, si los riesgos del proyecto se hacen realidad, es probable que la planificación temporal del proyecto se atrase. Los riesgos del proyecto identifican los problemas potenciales de presupuesto, planificación temporal, personal (asignación y organización), recursos, cliente y requisitos y su impacto en un proyecto de software.
- ✓ Personal: son todos aquellos eventos o condiciones que preocupan al personal vinculado al desarrollo y ejecución de la solución; va más encaminado al estado de ánimo o los problemas sociales de convivencia, de motivación con el proyecto en el cual están trabajando, las preocupaciones del personal por la carga docente y productiva; así como la necesidad que tienen las personas de que sea reconocido su trabajo. Esto puede provocar que la persona se sienta frustrada y que se convierta en un freno para el equipo de desarrollo.
- ✓ Tecnológicos: son todos aquellos eventos o condiciones que amenazan la calidad y la planificación temporal del software que hay que producir. Si un riesgo técnico se convierte en realidad, la

implementación puede tornarse compleja. Los riesgos técnicos identifican problemas potenciales de diseño, implementación, de interfaz. Los riesgos técnicos ocurren porque el problema es más difícil de resolver de lo que se pensaba.

- ✓ Otros: Son todos aquellos eventos o condiciones que no hayan quedado evidenciadas en ninguna de las clasificaciones antes descritas.

CONCLUSIONES PARCIALES

A través de la solución propuesta en este capítulo se da cumplimiento al objetivo relacionado con la confección del modelo. Con el análisis del comportamiento y desarrollo de la arquitectura en la disciplina de diseño en el ciclo de vida del proyecto se define la propuesta de solución por la gestión de la arquitectura, donde se define el alcance, los principios, las características y las premisas para su aplicación; se describe la estructura y elementos que conforman el modelo, las técnicas y herramientas utilizadas en cada actividad que conforma el flujo de actividades para llevar el proceso de gestión de la arquitectura.

CAPÍTULO 3 VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

INTRODUCCIÓN

Con el propósito de validar la solución propuesta en el capítulo anterior, se decide conformar un *Panel de Expertos* los cuales emitirán su criterio acerca de la adecuación del modelo. El proceso de validación se desarrolló mediante el *Método Delphi*, el cual es una técnica de investigación que tiene como propósito obtener una opinión grupal irrefutable partiendo de un grupo de expertos en el tema. Es un método de organización, favoreciendo la comunicación entre un grupo de personas las cuales pueden aportar ideas, conocimiento y experiencia para la resolución de un problema complejo. (Landeta, 2003)

EL MÉTODO DE EXPERTOS

Al realizar una investigación, uno de los principales inconvenientes que tiene asociado es la posibilidad de verificar y demostrar la confiabilidad de la propuesta resultante. Con el propósito de eliminar este problema surgen los Métodos de Expertos, estos utilizan como fuente de información y validación un grupo de personas a las que se le atribuye un supuesto conocimiento elevado en la materia que se va a tratar.

Los Métodos de Expertos presentan las siguientes ventajas y desventajas. (GTIC)

La opinión de varias personas que se consulten es más valiosa que la opinión individual de la persona con más experiencia en el tema. De ahí se deriva la idea de que varias cabezas son mejor que una. Además tomando en cuenta un grupo de personas estos siempre tendrán presente mayor número de factores para realizar la evaluación que los que tomará en cuenta una sola persona, proporcionando que cada experto pueda aportar al debate general la idea que tiene sobre el tema que se trata desde su propia área de conocimiento.

Sin embargo la presión social que ejerce el grupo respecto a sus participantes puede provocar acuerdos con la mayoría de los participantes, aunque la opinión de ésta sea errónea. Por lo que puede provocar que un experto pueda renunciar a la defensa de su opinión o idea, ante la presencia del grupo en no aceptarla. En estos grupos se da en ocasiones que el argumento que triunfa es el más citado, aunque este no sea el más válido. Otro voto en contra es la vulnerabilidad ante la posición y personalidad de algunos de los individuos, pues un participante con buena facilidad de expresión y comunicación, puede convencer al resto del panel de expertos aunque su opinión no sea la más acertada. Esta situación se puede dar también cuando uno de los expertos tenga un alto cargo en la organización, ya que sus subordinados no le revocarían sus argumentos con fuerza.

EL MÉTODO DELPHI

Se hace necesario un método que extraiga los beneficios de la interacción y comunicación directa de los métodos de expertos el cual descarte los inconvenientes que puedan surgir. El Método Delphi¹⁴ intenta seguir esa filosofía. Para la validación y aceptación del modelo que se presenta en el Capítulo 2 se hace uso del método Delphi.

El Delphi es uno de los métodos 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 un grupo de expertos en el tema tratado. (Fernández Aedo)

El Método Delphi proporciona un medio para agregar sistemáticamente los juicios informados de un grupo de expertos dentro de su campo de conocimiento. Permite rebasar el marco de las condiciones actuales más señaladas de un fenómeno y alcanzar una imagen integral y más amplia de su posible evolución, reflejando las valoraciones individuales de los expertos que pueden estar basadas en un análisis lógico, como en su experiencia intuitiva. (Aragón)

Este método permite la formación de un criterio o idea con mayor grado de integridad. Permite valorar alternativas de decisión. Por tener como característica el anonimato favorece a no crear conflictos entre los participantes creando un clima favorable al desarrollo de la creatividad garantizando el éxito del método. Por ser anónimo y confidencial favorece a la libertad de opinión pues ningún experto debe conocer que a su igual se le está solicitando opiniones.

Se basa en la organización de un diálogo anónimo, entre los expertos consultados de modo individual, con la aplicación de un cuestionario, con el propósito de obtener un consenso general o los motivos discrepantes entre los participantes. Los expertos seleccionados se someten a una serie de interrogantes sucesivas, cuyas respuestas son sometidas a procesos estadísticos para conocer la coincidencia o diferencia que estos tienen respecto al tema que se consulta. Este proceso iterativo, en el que cada cuestionario se informa los resultados del precedente, le brinda al experto la posibilidad de modificar sus respuestas anteriores, en función de los elementos aportados por otros expertos.

Este método posee características importantes como:

- El anonimato, los expertos responden las preguntas sin consultarse mutuamente. Es recomendable que dos expertos no conozcan entre sí que están opinando sobre un mismo tema.

¹⁴ El nombre de este método proviene del oráculo de Delphos, que se encontraba en la antigua Grecia, al que se acudía para hacer preguntas al dios a través de una sacerdotisa. El oráculo de Delphos poseía gran reputación por la certeza de sus predicciones.

- La retroalimentación controlada después de una ronda de preguntas, se tabulan las respuestas y se procesan antes de la siguiente ronda, para que los participantes puedan valorar los resultados de la ronda anterior, así como las razones dadas para cada respuesta y su dispersión del promedio. Permite que aumente el acuerdo al transcurrir varias rondas del proceso.
- Respuesta estadística del grupo donde el procesamiento de cada ronda se realiza con métodos estadísticos, característica que le hace diferencia de otros métodos existentes.

Por lo expresado anteriormente, es que se decide usar este método, en este caso la variante propuesta por Silvia Colunga y Georgina Amayuela (Colunga, y otros), a su vez empleada por el Licenciado Carlos Álvarez Martínez de Santelices en su tesis de maestría: “Experimentos virtuales para la enseñanza del Electromagnetismo” (Santelices, y otros), donde aparecen las conclusiones del estudio de numerosas tesis de maestría y doctorado relacionadas con ese tipo de investigación y la tesis de maestría del Ingeniero Rolando Quintana Aput: “Propuesta de indicadores para medir competencias del personal según el rol en proyectos multimedia”. Dichos autores no utilizan el Método Delphi para la valoración de expertos, sino una variante para propiciar mayor objetividad a los criterios de los especialistas a partir de la introducción de escalas valorativas.

Para la realización del método se realiza la selección de un grupo de expertos los cuales llevan a cabo el proceso de validación. En el transcurso de este período, ningún experto conocerá la identidad de los otros que forman parte del conjunto de debate, logrando así, que un integrante pueda cambiar o defender sus opiniones sin que eso cree una pérdida de imagen, en el caso de que sean erróneos, impide que un miembro del grupo sea influenciado por la reputación de otro de los miembros o por el peso que supone oponerse a la mayoría. Al realizar una educada selección de estos expertos proporciona la seguridad de un buen resultado, y al mismo tiempo un alto grado de credibilidad.

APLICACIÓN DEL MÉTODO DELPHI

Para llevar a cabo la aplicación del método se siguieron las siguientes etapas:

- ✓ Selección de expertos.
- ✓ Elaborar el cuestionario para la validación de la propuesta.
- ✓ Determinación de la concordancia de los expertos.
- ✓ Desarrollo exploratorio y práctico de los resultados.

3.1 Selección de los expertos.

Para un mayor entendimiento un experto es aquella persona o grupo de personas, que tienen amplios conocimientos o aptitudes en un área particular del conocimiento; estos son capaces de valorar, formular conclusiones y formular recomendaciones objetivas acerca del problema en cuestión.

Los expertos se seleccionaron teniendo en cuenta que cumplieran con los siguientes criterios:

- ✓ Graduado de Nivel Superior.
- ✓ Un año de experiencia como mínimo en temas de ASW o gestión de software.
- ✓ Vinculado al desarrollo de productos informáticos, preferentemente a proyectos de software de gestión.
- ✓ Vinculado a proyectos de software de gama alta.
- ✓ Prestigio en el colectivo de trabajo.

La selección de expertos atendiendo a estos criterios, proporciona la obtención de resultados con calidad, junto a otras cualidades propias de ellos como: la responsabilidad, la honestidad y la sinceridad, proporcionando que las opiniones brindadas sean confiables para el objetivo propuesto.

Para el desarrollo y puesta en práctica del método, se seleccionaron doce posibles expertos dentro de la UCI ¹⁵ para hacerle la propuesta, de los cuales diez respondieron afirmativamente para colaborar con la investigación y formar parte de la validación.

La autovaloración de los expertos se obtuvo luego de realizar una encuesta con el objetivo de determinar los coeficientes de competencia de los expertos seleccionados, recopilar información más detallada y actualizada sobre la labor que desempeñan, la calificación profesional, los años de experiencia en el tema y la categoría docente y científica. Para acceder a la encuesta aplicada consultar Anexo 4. Encuesta de autovaloración.

3.2 Cálculo del coeficiente de competencia

La selección de los expertos se hace de acuerdo a la valoración de sus competencias, para estos es necesario calcular el coeficiente de competencia (K), este se basa en los resultados de la encuesta de autovaloración, específicamente en el coeficiente de conocimiento (K_c) que es el resultado de la primera pregunta de la encuesta de autovaloración y el coeficiente de argumentación del conocimiento (K_a) el cual se obtiene luego de analizar los resultados de la tabla de la Pregunta 2 de la encuesta.

El análisis se hace de la siguiente forma:

Los expertos deben marcar, según su criterio y su grado de competencia sobre los aspectos sometidos a consideración, a estas marcas se le asignan valores de acuerdo a la siguiente escala:

Tabla 4. Grados de influencia en la determinación del coeficiente de argumentación.

FUENTES DE ARGUMENTACIÓN	Grado de influencia de cada fuente		
	Alto	Medio	Bajo
Análisis realizado por Ud. Respecto al tema	0.3	0.2	0.1
Experiencia obtenida en relación con el tema de investigación	0.5	0.4	0.2
Trabajos investigados de autores extranjeros	0.05	0.05	0.05
Trabajos investigados de autores nacionales	0.05	0.05	0.05
Su propio conocimiento del estado del problema en el extranjero	0.05	0.05	0.05
Su intuición	0.05	0.05	0.05
Totales	1.0	0.8	0.5

El coeficiente de argumentación del conocimiento (K_a) será igual a la suma de los valores donde el posible experto haya marcado.

Con estos datos se procede a calcular el coeficiente de competencia (K) a través de la siguiente fórmula:

$$K = \frac{K_c + K_a}{2}$$

Intervalos para determinar el nivel del coeficiente de competencia (K):

Si $0,8 < k < 1,0$ el coeficiente de competencia es alto.

Si $0,5 < k < 0,8$ el coeficiente de competencia es medio.

Si $k < 0,5$ el coeficiente de competencia es bajo.

Los expertos seleccionados para formar parte del grupo de validación de la propuesta fueron aquellos cuyos resultados arrojaron un coeficiente de competencia alto y medio.

De los diez expertos a los que se les hicieron la encuesta de autovaloración, solo nueve fueron seleccionados para continuar con la ejecución del método, los resultados se muestran a continuación:

Tabla 5. Coeficiente de competencia de los expertos.

Expertos	Coeficiente de Conocimiento (Kc)	Coeficiente de Argumentación del Conocimiento (Ka)	Coeficiente de Competencia (K)	Nivel
Experto 1	0,8	0,9	0,85	Alto
Experto 2	0,5	0,8	0,65	Medio
Experto 3	0,6	0,8	0,7	Medio
Experto 4	0,4	0,8	0,6	Medio
Experto 5	0,8	0,9	0,85	Alto
Experto 6	0,8	1	0,9	Alto
Experto 7	0,6	0,9	0,75	Medio
Experto 8	0,6	0,7	0,65	Medio
Experto 9	0,5	0,9	0,7	Medio
Experto 10	0,3	0,5	0,4	Bajo

El experto 10 deja de formar parte del grupo, dado que su coeficiente de competencia es bajo.

3.3 Elaboración del cuestionario de validación de la propuesta

Una vez seleccionados los expertos, se prosigue con la elaboración de la encuesta de validación, para lo cual se hace necesario elaborar un cuestionario de forma tal que se adapte a las condiciones de los expertos.

Para la elaboración de las preguntas se tuvieron en cuenta tres objetivos generales:

1. Evaluar las actividades que se proponen como parte de cada fase del modelo, son las necesarias y suficientes para llevar a cabo el desarrollo y gestión de la ASW para un sistema de gestión empresarial.
2. Evaluar si se realiza una correcta definición de roles, artefactos de entrada y salida, técnicas y herramientas para desarrollar cada una de las actividades.
3. Evaluar el sistema de trabajo que propone la solución para llevar a cabo la gestión de la arquitectura garantiza la eficacia del modelo propuesto.

El cuestionario fue elaborado de forma tal que las respuestas fueran categorizadas en: Muy adecuado (C1), Bastante adecuado (C2), Adecuado (C3), Poco adecuado (C4) y No adecuado (C5).

Para acceder al cuestionario para la validación del modelo, consultar Anexo 5. Encuesta de validación.

3.4 Establecer la concordancia de los expertos mediante el coeficiente de Kendall

Un perfecto acuerdo entre los expertos dará mayor validez a la propuesta, por lo que se necesita calcular el coeficiente de Kendall que ayuda a comprobar el grado de coincidencia de las valoraciones emitidas por los expertos.

El Coeficiente de Concordancia de Kendall, constituye un estadígrafo muy útil en estudios de confiabilidad entre expertos de una materia, al determinar la asociación entre distintas variables. Es una medida de coincidencia entre ordenaciones que pueden ser objetos o individuos. En este caso el coeficiente concordancia (W) será un índice de la divergencia del acuerdo efectivo entre los expertos mostrado en los datos del máximo acuerdo posible.

Para la aplicación del Coeficiente de Concordancia de Kendall (W), se construye una tabla de Aspectos a evaluar contra Expertos donde se sitúan los rangos de valoración (términos numéricos de 1 a 5) de cada aspecto evaluado contra cada uno de los expertos; estos datos son tomados a partir de los resultados arrojados luego de realizar la encuesta de validación. Para acceder a la tabla, consultar Anexo 6. Cálculo del coeficiente de concordancia de Kendall..

Después de la elaboración de la tabla se realizan los siguientes pasos:

- ✓ Determinar la suma de los valores numéricos asignados a cada aspecto a evaluar, según el criterio dado por cada experto (R_j).
- ✓ Determinar el valor medio de las R_j, dado por la sumatoria de los R_j entre N, siendo N el total de aspectos a evaluar (los aspectos serán las preguntas del cuestionario, en este caso N = 22).
- ✓ Determinación de la desviación media, dada por la diferencia entre cada R_j y el valor de la media.
- ✓ Determinación de la suma de los cuadrados de las desviaciones medias, S.
- ✓ Determinación del cuadrado del número total de expertos, K. En este caso K = 9.
- ✓ Determinación del cubo del número total de aspectos a evaluar, N.
- ✓ Determinación de la diferencia entre el cubo de N y N, y su multiplicación por el cuadrado de K.

Una vez que se obtienen todos estos datos es posible calcular el Coeficiente de Kendall (W) a través de la siguiente fórmula:

$$W = \frac{12 * S}{K^2 (N^3 - N)}$$

El coeficiente W ofrece el valor que posibilita decidir el nivel de concordancia entre los expertos. El valor de W siempre es positivo y oscila entre 0 y 1. Con el coeficiente de Kendall se puede calcular el Chi

cuadrado real, para demostrar si existe o no concordancia entre los expertos, el mismo se obtiene a través de la siguiente fórmula:

$$x^2 = K(N - 1)W$$

El Chi cuadrado calculado se compara con el de las tablas estadísticas. Si $X^2_{\text{real}} < x^2(\infty, N - 1)$, entonces existe concordancia en el trabajo de los expertos. Luego la realización de los cálculos pertinentes, estos arrojaron que $X^2_{\text{real}} = 0,554318$ y el $X^2_{(0,01, 21)} = 38,9322$, lo cual confirma el cumplimiento de la comparación y por tanto, existe concordancia entre los expertos. Para acceder a los pasos realizados para obtener estos resultados, consultar Anexo 6. Cálculo del coeficiente de concordancia de Kendall..

3.5 Desarrollo práctico y exploratorio de los resultados

Los expertos que conformaron el panel recibieron un resumen de la propuesta de solución como documentación primaria para responder los temas encuestados, además del cuestionario con un total de 7 preguntas. Los cuestionarios fueron enviados vía e-mail y en algunos casos fueron entregados impresos, con una breve explicación de las condiciones prácticas del desarrollo de la encuesta (plazo de respuesta y garantía de anonimato). Se realizó solo una ronda de preguntas y luego se prosiguió a analizar los resultados de las opiniones emitidas por los expertos.

Los resultados de los cuestionarios se procesaron según Colunga y Amayuela. Con el objetivo de recoger y visualizar los resultados aportados se fueron confeccionando tablas. Para ello se utilizó el Microsoft Excel 2007.

En la siguiente tabla de doble entrada se muestran los resultados que se recogieron:

Tabla 6. Frecuencias absolutas para cada pregunta de la encuesta.

Frecuencias absolutas:							
No	Elementos	C1	C2	C3	C4	C5	Total
1	Pregunta 1.1	7	0	2	0	0	9
2	Pregunta 1.2	5	4	0	0	0	9
3	Pregunta 1.3	6	2	0	1	0	9
4	Pregunta 1.4	6	2	0	0	1	9
5	Pregunta 1.5	7	1	0	1	0	9
6	Pregunta 1.6	4	4	1	0	0	9
7	Pregunta 1.7	4	3	1	0	1	9
8	Pregunta 1.8	5	3	1	0	0	9
9	Pregunta 1.9	4	4	0	0	1	9
10	Pregunta 1.10	1	6	1	1	0	9
11	Pregunta 2,1	5	3	0	1	0	9
12	Pregunta 2,2	6	1	1	1	0	9
13	Pregunta 2,3	5	1	2	1	0	9

14	Pregunta 2,4	7	1	0	1	0	9
15	Pregunta 2,5	4	3	1	1	0	9
16	Pregunta 3	4	1	3	0	1	9
17	Pregunta 4,1	5	3	1	0	0	9
18	Pregunta 4,2	6	2	0	1	0	9
19	Pregunta 5,1	1	4	2	1	1	9
20	Pregunta 5,2	1	4	2	2	0	9
21	Pregunta 6	4	3	2	0	0	9
22	Pregunta 7	2	6	0	1	0	9

Luego de tener los datos tabulados se procede a realizar los siguientes pasos para poder obtener los resultados esperados:

Primer paso: Se construye una tabla de frecuencias acumuladas. Esto es, cada número en la fila, excepto el primero se obtiene sumándole el anterior.

Tabla 7.Frecuencias absolutas acumuladas.

Tabla de frecuencias absolutas acumuladas:						
No	Aspectos	C1	C2	C3	C4	C5
1	Pregunta 1.1	7	7	9	9	9
2	Pregunta 1.2	5	9	9	9	9
3	Pregunta 1.3	6	8	8	9	9
4	Pregunta 1.4	6	8	8	8	9
5	Pregunta 1.5	7	8	8	9	9
6	Pregunta 1.6	4	8	9	9	9
7	Pregunta 1.7	4	7	8	8	9
8	Pregunta 1.8	5	8	9	9	9
9	Pregunta 1.9	4	8	8	8	9
10	Pregunta 1.10	1	7	8	9	9
11	Pregunta 2,1	5	8	8	9	9
12	Pregunta 2,2	6	7	8	9	9
13	Pregunta 2,3	5	6	8	9	9
14	Pregunta 2,4	7	8	8	9	9
15	Pregunta 2,5	4	7	8	9	9
16	Pregunta 3	4	5	8	8	9
17	Pregunta 4,1	5	8	9	9	9
18	Pregunta 4,2	6	8	8	9	9
19	Pregunta 5,1	1	5	7	8	9
20	Pregunta 5,2	1	5	7	9	9

21	Pregunta 6	4	7	9	9	9
22	Pregunta 7	2	8	8	9	9

Segundo paso: Se copia la tabla anterior y se borran los resultados numéricos. Ahora, en esta nueva tabla, se construye la tabla de frecuencias relativas acumuladas.

Tabla 8. Frecuencias relativas acumuladas.

Tabla de frecuencias relativas acumuladas						
No	Aspectos	C1	C2	C3	C4	C5
1	Pregunta 1.1	0,77778	0,77778	0,9999	0,9999	0,9999
2	Pregunta 1.2	0,55556	0,9999	0,9999	0,9999	0,9999
3	Pregunta 1.3	0,66667	0,88889	0,88889	0,9999	0,9999
4	Pregunta 1.4	0,66667	0,88889	0,88889	0,88889	0,9999
5	Pregunta 1.5	0,77778	0,88889	0,88889	0,9999	0,9999
6	Pregunta 1.6	0,44444	0,88889	0,9999	0,9999	0,9999
7	Pregunta 1.7	0,44444	0,77778	0,88889	0,88889	0,9999
8	Pregunta 1.8	0,55556	0,88889	0,9999	0,9999	0,9999
9	Pregunta 1.9	0,44444	0,88889	0,88889	0,88889	0,9999
10	Pregunta 1.10	0,11111	0,77778	0,88889	0,9999	0,9999
11	Pregunta 2,1	0,55556	0,88889	0,88889	0,9999	0,9999
12	Pregunta 2,2	0,66667	0,77778	0,88889	0,9999	0,9999
13	Pregunta 2,3	0,55556	0,66667	0,88889	0,9999	0,9999
14	Pregunta 2,4	0,77778	0,88889	0,88889	0,9999	0,9999
15	Pregunta 2,5	0,44444	0,77778	0,88889	0,9999	0,9999
16	Pregunta 3	0,44444	0,55556	0,88889	0,88889	0,9999
17	Pregunta 4,1	0,55556	0,88889	0,9999	0,9999	0,9999
18	Pregunta 4,2	0,66667	0,88889	0,88889	0,9999	0,9999
19	Pregunta 5,1	0,11111	0,55556	0,77778	0,88889	0,9999
20	Pregunta 5,2	0,11111	0,55556	0,77778	0,9999	0,9999
21	Pregunta 6	0,44444	0,77778	0,9999	0,9999	0,9999
22	Pregunta 7	0,22222	0,88889	0,88889	0,9999	0,9999

Los datos de esta tabla se obtienen dividiendo cada uno de los números de la Tabla 3 por el número total de expertos. En este caso el número total de expertos es nueve.

Tercer paso: Se buscan las imágenes de los elementos de la tabla anterior por medio de la función (Dist. Normal. Standard Inv.).

A la misma tabla se le adicionan tres columnas y una fila para colocar los resultados que se mencionan a continuación:

- ✓ Suma de las columnas.
- ✓ Suma de filas.
- ✓ Promedio de las columnas.
- ✓ Los promedios de las filas se obtienen de forma similar, en este caso también se divide por cuatro porque quedan cuatro categorías ya que la última se eliminó.
- ✓ Para hallar N, se divide la suma de las sumas entre el resultado de multiplicar el número de indicadores por el número de preguntas.
- ✓ El valor N - P da el valor promedio que otorgan los expertos para cada indicador propuesto.

En la siguiente tabla resume lo dicho en los puntos anteriores:

Tabla 9. Puntos de corte

Puntos de corte:								
No	Aspectos	C1	C2	C3	C4	Suma	N-P	
1	Pregunta 1.1	0,76	0,76	3,72	3,72	8,97	-1,04	Muy adecuado
2	Pregunta 1.2	0,14	3,72	3,72	3,72	11,30	-1,62	Muy adecuado
3	Pregunta 1.3	0,43	1,22	1,22	3,72	6,59	-0,45	Muy adecuado
4	Pregunta 1.4	0,43	1,22	1,22	1,22	4,09	0,18	Bastante adecuado
5	Pregunta 1.5	0,76	1,22	1,22	3,72	6,93	-0,53	Muy adecuado
6	Pregunta 1.6	-0,14	1,22	3,72	3,72	8,52	-0,93	Muy adecuado
7	Pregunta 1.7	-0,14	0,76	1,22	1,22	3,07	0,44	Bastante adecuado
8	Pregunta 1.8	0,14	1,22	3,72	3,72	8,80	-1,00	Muy adecuado
9	Pregunta 1.9	-0,14	1,22	1,22	1,22	3,52	0,32	Bastante adecuado
10	Pregunta 1.10	-1,22	0,76	1,22	3,72	4,48	0,08	Bastante adecuado
11	Pregunta 2,1	0,14	1,22	1,22	3,72	6,30	-0,37	Muy adecuado
12	Pregunta 2,2	0,43	0,76	1,22	3,72	6,14	-0,33	Muy adecuado
13	Pregunta 2,3	0,14	0,43	1,22	3,72	5,51	-0,17	Muy adecuado
14	Pregunta 2,4	0,76	1,22	1,22	3,72	6,93	-0,53	Muy adecuado
15	Pregunta 2,5	-0,14	0,76	1,22	3,72	5,56	-0,19	Muy adecuado

16	Pregunta 3	-0,14	0,14	1,22	1,22	2,44	0,59	Bastante adecuado
17	Pregunta 4,1	0,14	1,22	3,72	3,72	8,80	-1,00	Muy adecuado
18	Pregunta 4,2	0,43	1,22	1,22	3,72	6,59	-0,45	Muy adecuado
19	Pregunta 5,1	-1,22	0,14	0,76	1,22	0,90	0,98	Bastante adecuado
20	Pregunta 5,2	-1,22	0,14	0,76	3,72	3,40	0,35	Bastante adecuado
21	Pregunta 6	-0,14	0,76	3,72	3,72	8,06	-0,81	Muy adecuado
22	Pregunta 7	-0,76	1,22	1,22	3,72	5,40	-0,15	Muy adecuado
Suma		-0,55	22,58	40,93	69,33	132,29		
Puntos de corte		-0,02	1,03	1,86	3,15			

Las sumas obtenidas en las cuatro primeras columnas dan los puntos de corte:

Los puntos de corte se utilizan para determinar la categoría o grado de adecuación de cada criterio según la opinión de los expertos consultados. Con ellos se opera del modo siguiente:

Tabla 10 Grados de adecuación

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	No Adecuado
-0,02	1,03	1,86	3,15	

3.6 Resultados de la validación del modelo

Participaron en la selección de criterios para medir competencias nueve expertos representados como se muestra en la Figura 10.

El 78% del total de los expertos son Ingenieros en Ciencias Informáticas y el 22% son Licenciado en Educación. Todos los expertos están relacionados al desarrollo de software y temas relacionados a la investigación.

Los miembros del panel fueron seleccionados de la UCI y se pudo observar que la experiencia de los mismos oscilaba en un rango de uno a cinco años.

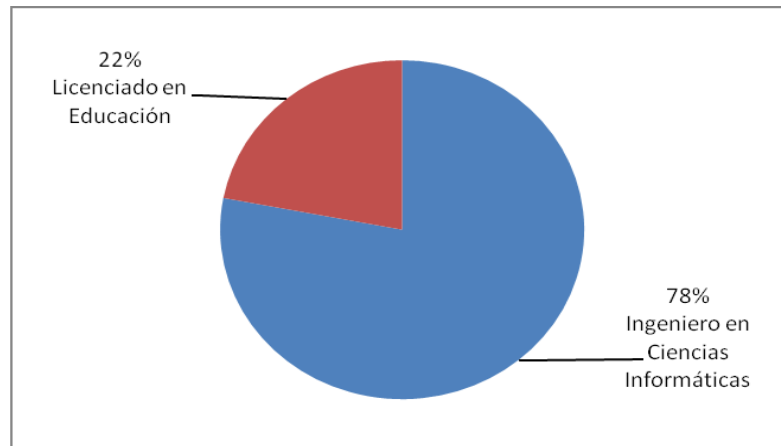


Figura 7. Representatividad de los expertos por especialidad

De todos los expertos se seleccionaron aquellos que tenían un coeficiente de competencia medio y alto.

En la Figura 11 se muestra el resumen de los resultados obtenidos de la encuesta de autovaloración del nivel de competencia de cada uno de los especialistas.

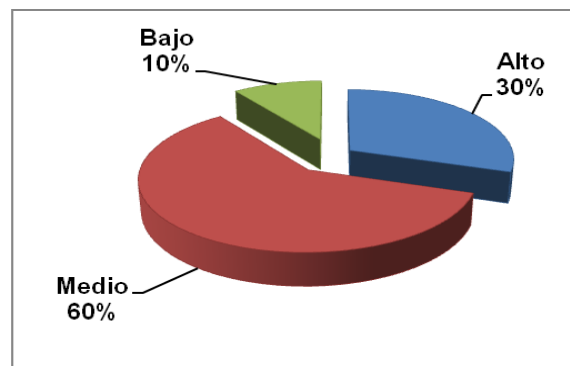


Figura 8. Coeficiente de Competencia de los expertos

En la misma se evidencia que el 30% de los encuestados tiene un nivel de competencia Alto, el 60% Medio y el resto Bajo. Estos datos demuestran que el 90% de los expertos tienen el conocimiento necesario para validar la propuesta, dando un alto valor a sus criterios con respecto a cada una de las preguntas realizadas en la encuesta.

3.6.1 Resultados obtenidos de la encuesta de validación

En este epígrafe se incluirán los resultados arrojados de la tabulación por preguntas y por objetivos.

Tabulación de los resultados por preguntas

El resultado arrojado del análisis estadístico de las respuestas de los expertos se puede observar en la Figura 12. Aquí solo se mostrará el nivel de adecuación de las respuestas a las preguntas 3, 4.1, 4.2, 5.1, 5.2, 6 y 7; para desglosar de manera más detallada el de las preguntas restantes con sus incisos.

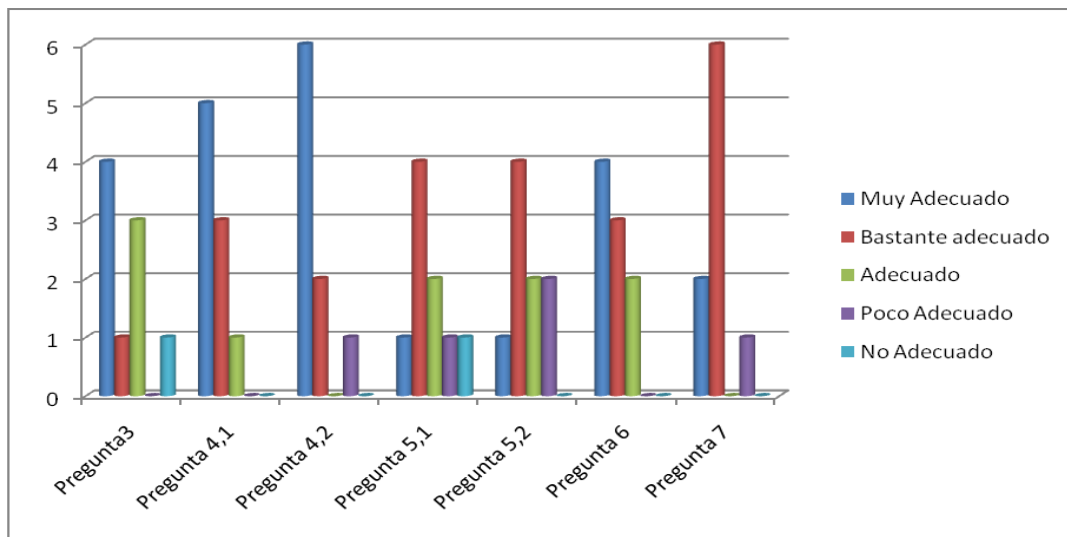


Figura 9. Nivel de adecuación de las Preguntas 3, 4.1, 4.2, 5.1, 5.2, 6 y 7

Por su parte las respuestas a la pregunta 1, correspondientes a la adecuación del flujo de actividades que se propone en el modelo, muestra que todos los expertos coinciden en que estos son adecuados, muy adecuados y bastante adecuados, tal como se muestra en la siguiente figura:

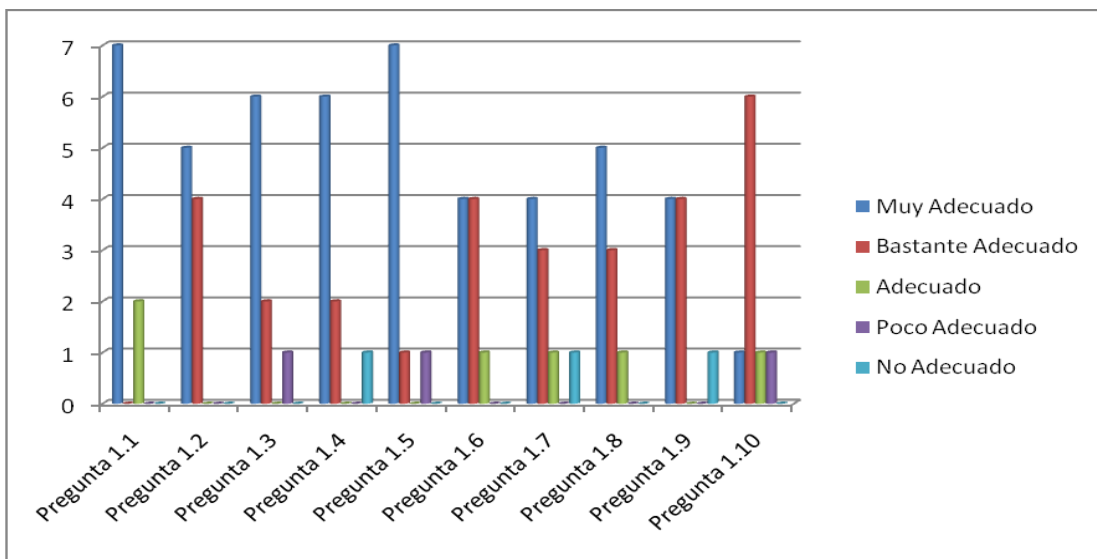


Figura 10. Nivel de adecuación de las Preguntas 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10

Por su parte la pregunta dos, corresponde a la adecuación de los roles que se proponen en el modelo, muestra que todos los expertos coinciden en que estos son adecuados, muy adecuados y bastante adecuados, tal como se muestra en el siguiente gráfico:

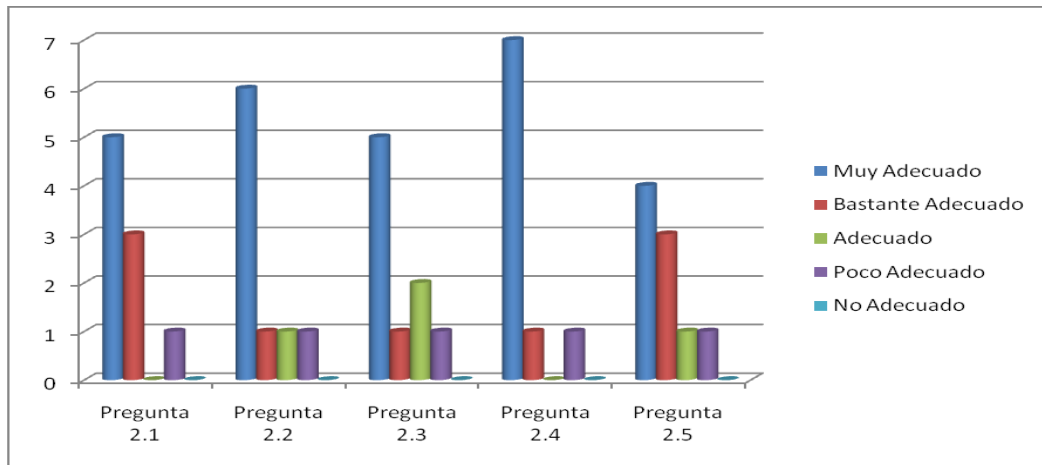


Figura 11. Nivel de adecuación de la pregunta 2

Con todos estos resultados, se obtuvo el 32% de las preguntas fueron catalogadas de bastante adecuadas y el 68% de muy adecuadas, tal como se muestra en el gráfico de la siguiente figura:

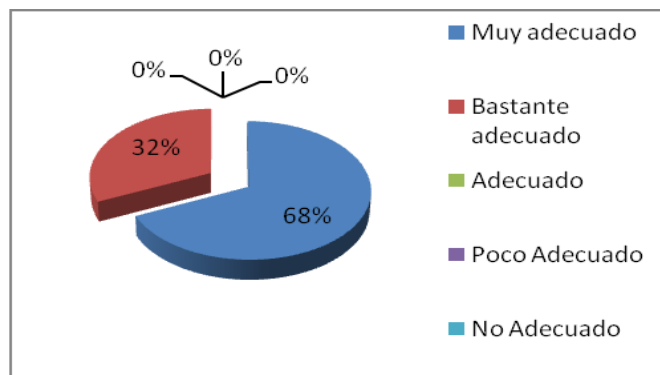


Figura 12. Nivel de adecuación de las preguntas de la encuesta

Tabulación de los resultados por objetivos

Luego de la tabulación de los cálculos anteriores, se realizó un resumen de los niveles de adecuación por objetivos, los resultados obtenidos se muestran a continuación:

Objetivo 1: Este objetivo general se desglosa en el objetivo específico que se plantean a continuación y que están relacionados con la Pregunta 1. Los resultados son los siguientes:

Tabla 11. Evaluación del objetivo 1

Pregunta	Objetivos específicos	Resultado obtenido
1	Demostrar que las actividades propuestas son las necesarias y suficientes para cumplir con el desarrollo y gestión de la ASW para un sistema de gestión empresarial.	Muy adecuado

Objetivo 2: Se desglosa en cuatro objetivos específicos relacionados con las Preguntas 2, 3, 5, 7 y 4, de los cuales se describen y se brindan los resultados alcanzados a continuación:

Tabla 12. Evaluación del objetivo 2

Pregunta	Objetivos específicos	Resultado obtenido
2	Demostrar que todos los roles propuestos son suficientes y necesarios.	Muy adecuado
3	Demostrar que las técnicas propuestas son adecuadas y necesarias para desarrollar las actividades.	Muy adecuado
5	Demostrar que las herramientas propuestas son adecuadas.	Bastante adecuado
4	Demostrar que los artefactos propuestos son suficientes y necesarios.	Muy adecuado

Objetivo 3: Se desglosa en los siguientes objetivos específicos relacionados con las Preguntas 6 y 7, de los cuales se describen y se brindan los resultados a continuación:

Tabla 13. Evaluación del objetivo 3

Pregunta	Objetivos específicos	Resultado obtenido
6	Demostrar la importancia y necesidad de la identificación de desviaciones y riesgos para la gestión de la arquitectura.	Muy adecuado
7	Demostrar la importancia y necesidad de establecer un modelo para la gestión de la arquitectura.	Bastante adecuado

CONCLUSIONES PARCIALES

La aplicación del cuestionario a los 9 expertos seleccionados arrojó resultados satisfactorios:

- ✓ Todas las preguntas fueron categorizadas de muy adecuadas, bastante adecuadas y adecuadas por el 100% de los expertos.
- ✓ Los tres objetivos generales a partir de los cuales se elaboraron las preguntas fueron cumplidos satisfactoriamente.
- ✓ Más del 95% de los expertos consideraron que la implantación del modelo puede llegar a ser muy adecuado o bastante adecuado.

Se puede concluir que la propuesta fue validada por la totalidad de los miembros del panel de expertos, lo cual demuestran las opiniones emitidas.

CONCLUSIONES GENERALES

Con este trabajo, se logró dar cumplimiento al objetivo principal de la investigación; realizar una propuesta de modelo para la gestión de la ASW para un sistema de planificación empresarial. Se identificaron artefactos, roles así como las herramientas y técnicas que intervienen en el desarrollo de la gestión de la arquitectura, a partir del modelo propuesto. Se caracterizaron todos los aspectos asociados al dominio del problema, los cuales contribuyeron a obtener los conocimientos necesarios para realizar una propuesta sólida. Se logró realizar un análisis crítico y valorativo de las principales metodologías y modelos de desarrollo que existen actualmente. Además se logró validar el modelo propuesto a través de los criterios aportados por un grupo de expertos, los cuales catalogaron más del 95% de los elementos del modelo como muy adecuado y bastante adecuados.

RECOMENDACIONES

Al concluir el presente trabajo de diploma, y luego de considerar cumplidos los objetivos trazados en el mismo, se recomienda:

- ✓ Incluir en los libros de proceso del centro la propuesta de solución.
- ✓ Profundizar en los tipos de desviaciones que propone el programa de mejora.
- ✓ Realizar los ajustes necesarios al modelo de acuerdo con las características del proyecto en el que se vaya a implantar, teniendo en cuenta la flexibilidad que ofrece el mismo.

BIBLIOGRAFÍA

Software Engineering Institute(SEI). [En línea] <http://www.sei.cmu.edu>.

Seminario sobre RUP en un entorno empresarial de desarrollo. [En línea] [http://www-5.ibm.com/services/learning/es/tairis.nsf/\(ExtCourseNr\)/RUPS1ES](http://www-5.ibm.com/services/learning/es/tairis.nsf/(ExtCourseNr)/RUPS1ES).

<http://www.microsoft.com>. [En línea]

¿Estoy o no estoy usando Scrum? **Silver., Melanie G.**

Alianza Ágil. Alianza Ágil. [En línea] <http://www.agilealliance.org>.

Alistair. *Desarrollo de Software Ágil*. [En línea]

ALLSOFT S.A. *Modelos de desarrollo*. [En línea]

Alvarez M., Itzcoalt. *Desarrollo Ágil con SCRUM*. [En línea] <http://www.sg.com.mx/sg07/presentaciones/Mejora%20de%20procesos/SG07.P02.Scrum.pdf>.

Aragón, Salvador. *Método Delphi*. [En línea]

Arboleda Jiménez, MSc. Hugo F. *Modelos de Ciclo de Vida de Desarrollo de Software en el Contexto de la Industria Colombiana de Software*. [En línea]

Barrientos Enríquez, Aleida Mirian. *El desarrollo de sistemas de información empleando el lenguaje de modelado unificado UML*. [En línea]

Bass, Len , Clements , Paul and Kazman , Rick. *Software Architecture in Practice*. [En línea] 0-321-15495-9.

C.V., ALLSOFT S.A. de. *Modelos de desarrollo*. [En línea] <http://www.google.com.cu/url?sa=t&source=web&ct=res&cd=3&ved=0CA4QFjAC&url=http%3A%2F%2Fwww.allsoft.com.mx%2Frecursos%2FModelosD.ppt&rct=j&q=%22Modelos+de+Desarrollo%22%2BALLSOFT+&ei=GkneS4fBMIS-9QTTJiYGrBw&usg=AFQjCNFfCsUOXQGDmUxatKieW6U7cT5Btg>.

Canós, José H., Letelier, Patricio y Penadé, M^a Carmen. *Métodologías Ágiles en el Desarrollo de Software*. [En línea]

Casanovas, Josep. *Usabilidad y arquitectura del software*. [En línea] <http://www.desarrolloweb.com/articulos/1622.php>.

CGI. *Introducción al Agile UP*. [En línea] CGI. <http://cgi.una.ac.cr/AUP/html/overview.html>.

Colunga, Silvia y G A. *La Psicología Educativa, su objeto, métodos y problemas principales*. [En línea]

Echeverría Cossío, Yanelis. *Modelo Ágil de Desarrollo de Proyectos de Software:Paradigma 3P*. [En línea]

Fernández Aedo, Raúl Rubén. *Modelo Informático para la autogestión del aprendizaje para la universalización de la enseñanza*. [En línea]

Fowler, Martín. *La Nueva Metodología. La Nueva Metodología*. [En línea] <http://www.programacion.net>.

GTIC. *Grupo de Tecnologías de la Información y las Comunicaciones. SSR - ETSI Telecomunicación - UPM*. [En línea] <http://www.gtic.ssr.upm.es/encuestas/delphi.htm>.

<http://www.dc.uba.ar>. [En línea]

<http://www.dybox.cl/metodologia/rup.html>. [En línea] <http://www.dybox.cl/metodologia/rup.html>.

- Ing. Ariel y Temis.** MODELO DE PRODUCCIÓN PARA EL FLUJO DE ARQUITECTURA DE DATOS DEL PROYECTO ERP-CUBA. *MODELO DE PRODUCCIÓN PARA EL FLUJO DE ARQUITECTURA DE DATOS DEL PROYECTO ERP-CUBA*. [En línea]
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. [En línea]
- Josep.** DesarrolloWeb. *DesarrolloWeb*. [En línea] [Citado el: 20 de Enero de 2010.] <http://www.desarrolloweb.com>.
- Landeta, Jon. 2003.** Aplicación del Método Delphi en la elaboración de la tabla simétrica de las tablas input-output 2001 de Catalunya. [En línea] 2003.
- Larisa y Osmar.** MODELO DE PRODUCCIÓN PARA EL FLUJO DE ARQUITECTURA DE SISTEMA DEL PROYECTO ERP-CUBA.
- Lic. Carlos Leone (MBA), Ing. Nicolás Passerini y Ing. Gustavo A. Brey.** *Metodologías Iterativas de Desarrollo*. [En línea] http://apit.wdfiles.com/local--files/start/01_apit_metodologias.pdf.
- Manifiesto para el Desarrollo de Software Ágil. *Manifiesto para el Desarrollo de Software Ágil*. [En línea] <http://www.agilemanifesto.org>.
- Meier, J D.** *Application Architecture Guide 2.0*. [En línea]
- mitecnologico. *mitecnologico*. [En línea] <http://www.mitecnologico.com/Main/ModeloBasadoEnComponentesDise%F1oDeSistemas>.
- Palacio, Juan.** *Flexibilidad con Scrum*. [En línea]
- proyecto, colectivo de autores del.** Ciclo de vida del proyecto. *Ciclo de vida del proyecto*. [En línea]
- Reynoso, Billy.** Architect Academy: Seminario de Arquitectura de Software. [En línea]
- Reynoso, Carlos Billy.** <http://www.pdfgratis.org>. <http://www.pdfgratis.org>. [En línea] <http://www.pdfgratis.org/Introduccion-a-la-arquitectura/>.
- . Introducción a la Arquitectura de Software. *Introducción a la Arquitectura de Software*. [En línea] [Citado el: 23 de Enero de 2010.] <http://www.willydev.net>.
- Santelices y L.C.Á.M.D.** Experimentos virtuales para la enseñanza del Electromagnetismo. [En línea]
- Serrano, Jorge.** *Explicando Scrum a mi abuela*. [En línea]
- Vázquez, Ivis Rosa, García, Bibiana y Carrillo Sorey, Pablo Andrés.** *El Rol de un Arquitecto de Software*. [En línea]
- Verdecia, Osvaldo Díaz y Quevedo Campins, Virgen Damaris.** *Una guía práctica de Arquitectura de Software*.
- Vistas, Introduccion a las.** Introduccion a las Vistas. [En línea] <http://www.dc.uba.ar/materias/arq-soft/2007/cuat1/descargas>.
- Welicki, León E.** *Implementando Extreme Programming en la plataforma.NET*. [En línea] <http://www26.brinkster.com/lwelicki/articles/Implementando%20Extreme%20Programming%20en%20la%20plataforma%20NET.pdf>.

GLOSARIO DE TÉRMINOS

“ -- A -- ”

Arquitectura: Tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.

“ -- E -- ”

ERPs (Enterprise Resource Planning): Los sistemas de planificación de recursos de la empresa son sistemas de gestión de información que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa. Los sistemas ERP son sistemas integrales de gestión para la empresa. Se caracterizan por estar compuestos por diferentes partes integradas en una única aplicación.

“ -- F -- ”

Frameworks: Es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

“ -- G -- ”

Gestión: Gestión es la acción y efecto de gestionar o la acción o efecto de administrar. Comprende todas las actividades de una organización que implican el establecimiento de metas u objetivos, así como la evaluación de su desempeño y cumplimiento; además del desarrollo de una estrategia operativa que garantice la supervivencia de la misma, según al sistema social correspondiente.

“ -- H -- ”

Herramientas: Es un dispositivo artificial cuya función es facilitar la aplicación de energía a una pieza o material durante la realización de una tarea. Es frecuente usar el término herramienta, por extensión, para denominar dispositivos o procedimientos que aumentan la capacidad de hacer ciertas tareas. Tal es el caso de herramientas de programación, herramientas matemáticas o herramientas de gestión. Esto frecuentemente viola la característica básica de las herramientas de ser medios para la aplicación controlada de energía.

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.

“ -- I -- ”

IEEE: Siglas del Instituto de Ingenieros Eléctricos y Electrónicos, (Institute of Electrical and Electronics Engineers). Fundada en 1984, es una asociación técnico-profesional mundial dedicada a la estandarización. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación, ingenieros en informática e ingenieros en telecomunicación.

Ingeniería de software: La aplicación de un enfoque sistemático disciplinado y cuantificable al desarrollo, operación y mantenimiento del software.

“ -- P -- ”

Proyecto: Proceso único consistente en un conjunto de actividades coordinadas y controladas con fechas de inicio y de finalización, llevadas a cabo para lograr un objetivo conforme con los requisitos específicos, incluyendo las limitaciones de tiempo, costo y recursos.

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

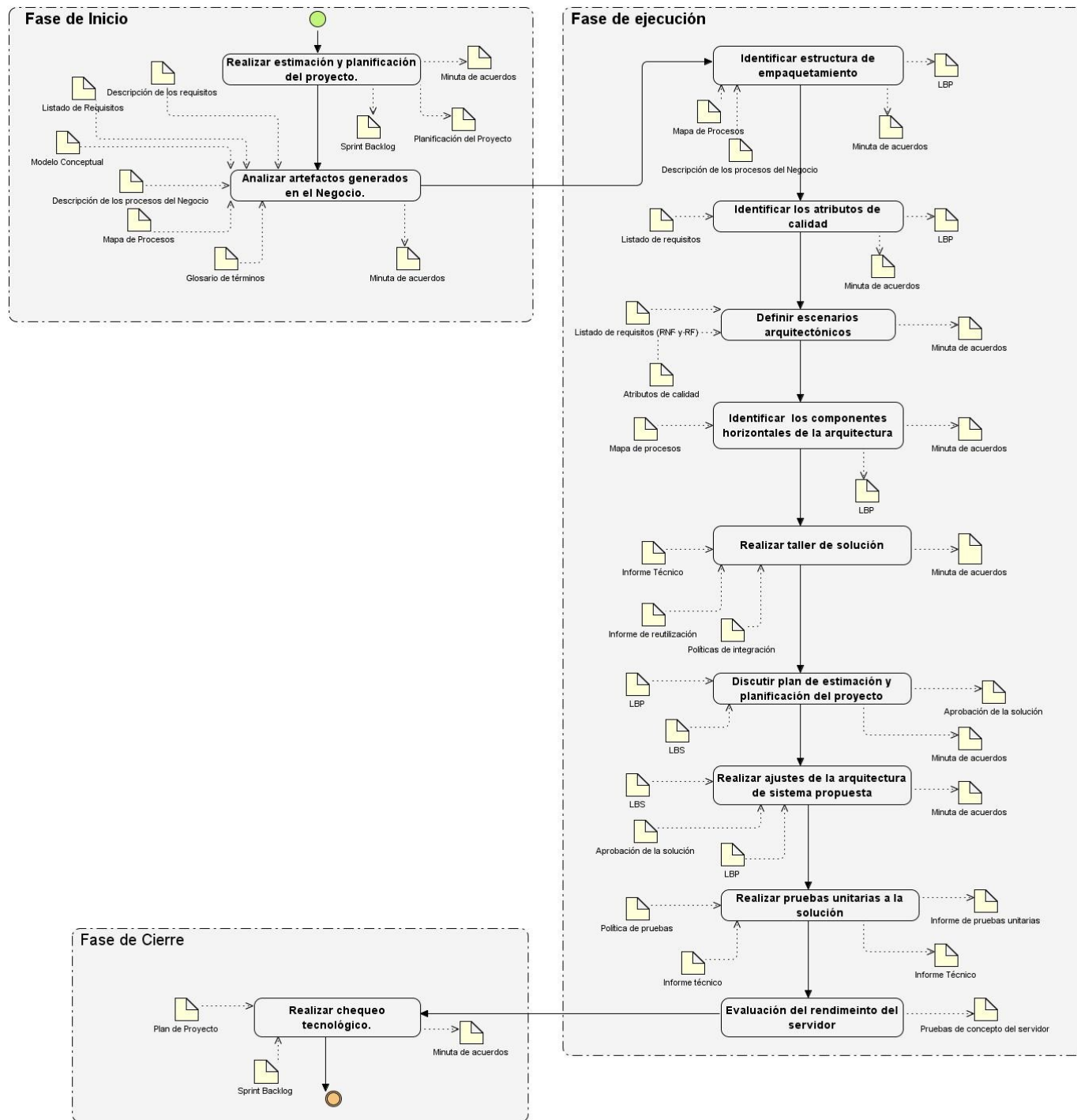
Patrones GRAPS: Patrones generales de software para asignación de responsabilidades, es el acrónimo de "General Responsibility Assignment Software Patterns". Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software.

“ -- S -- ”

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).

ANEXOS

Anexo 1. Diagrama del Modelo.



Anexo 2. Comparación entre los modelos de proceso de software

Modelos de proceso de software	Funciona con requisitos y arquitectura no predefinidos	Produce software altamente fiable	Gestión de riesgos	Permite correcciones sobre la marcha	Visión del progreso por el Cliente y el Jefe del proyecto
Codificar y corregir.	Bajo	Bajo	Bajo	Alto	Medio
Cascada	Bajo	Alto	Bajo	Bajo	Bajo
Evolutivo exploratorio	Medio o Alto	Medio o Alto	Medio	Medio o Alto	Medio o Alto
Evolutivo prototipado	Alto	Medio	Medio	Alto	Alto
Desarrollo formal de sistemas	Bajo	Alto	Bajo a Medio	Bajo	Bajo
Desarrollo orientado a reutilización	Medio	Bajo a Alto	Bajo a Medio	Alto	Alto
Incremental	Bajo	Alto	Medio	Bajo	Bajo
Espiral	Alto	Alto	Alto	Medio	Medio

Anexo 3. Competencias para cada rol que propone el modelo.

Rol	Competencias
Arquitecto Corporativo o Principal.	<ul style="list-style-type: none"> • Amplio conocimiento del entorno de desarrollo. • Dominio del trabajo con las herramientas tecnológicas, de modelado y de gestión. • Ser constante en el control y la organización del trabajo y evolución del desarrollo. • Habilidades de comunicación. • Habilidades para trabajar en equipo. • Capacidad de redacción y concreción. • Conocimiento de Ingeniería de Software. • Conocimientos generales de planificación y gestión de proyectos. • Conocimientos profundos de la tecnología utilizada en el desarrollo o relacionadas con el proyecto. • Conocimiento de modelos de desarrollo de software y arquitectónicos. • Conocimiento de los cánones teóricos de la arquitectura del software. • Conocimiento de estilos, patrones arquitectónicos, patrones de diseño y patrones de datos. • Conocimiento de la disciplina de arquitectura empresarial.

	<ul style="list-style-type: none"> • Conocimiento de notaciones de modelado BPMN, IDEFF, UML, ADL. • Alta capacidad de abstracción. • Alta capacidad de integrar conocimientos. • Capacidad predictiva de los sucesos, problemas y riesgos del proyecto. • Conoce la organización que abstrae en las soluciones arquitectónicas con amplitud y precisión de sus procesos y estructura organizacional, así como las leyes en que se enmarca
Líder de Gestión.	<ul style="list-style-type: none"> • Dominio del trabajo con las herramientas de gestión definidas por el centro. • Ser constante en el control y la organización. • Conocimientos generales de riesgos. • Habilidades de comunicación • Habilidades para trabajar en equipo • Capacidad de redacción y concreción. • Conocimiento de Ingeniería de Software. • Conocimientos generales de planificación y gestión. • Dominio del marco de trabajo definido por el centro. • Liderazgo.
Arquitecto de Sistema.	<ul style="list-style-type: none"> • Dominio de la tecnología, estrategia de negocios, política organizacional y consultoría. • Dominio de las técnicas de negociación. • Alta capacidad de abstracción. • Conoce arquitectura de bases de datos, técnicas de modelados de base de datos. • Conoce técnicas de modelado y gestión de integración, conoce los estilos arquitectónicos y patrones de integración. • Liderazgo. • Conocimientos sobre estilos arquitectónicos, patrones de diseño, metodologías de desarrollo. • Habilidades gerenciales. • Debe estar capacitado para asistir a la organización en la selección de la arquitectura más adecuada para responder a las necesidades reales del negocio. • Conoce el proceso de negocio que abstrae en la solución arquitectónica. • Conoce modelos de desarrollo y modelado como BPM, TOGAF, EA. • Conoce las notaciones de modelados existentes ADL, UML, IDEFF, BPMN.
Arquitecto de Datos.	<ul style="list-style-type: none"> • Posee amplios conocimientos en la teoría relacional de Base Datos (BD) (cálculo, álgebra y normalización). • Posee amplios conocimientos de diferentes herramientas de modelado para BD, tanto las de creación de una BD como Postgre, SQL, MS SQL, Oracle, como las de modelado ER/Studio, Visual Paradigm. • Conoce teorías de diseño base de datos, MER, normalización y desmoralización. • Amplios conocimientos de la arquitectura tecnológica de los gestores de bases de datos con que trabaja, así como los elementos metodológicos de configuración, optimización, y gestión de los mismos. • Posee conocimientos sobre optimización de bases de datos y de SQL estándar. • Posee conocimiento de los patrones de diseño de datos. • Posee conocimiento de los patrones de diseño. • Dirige y lidera es otra de las características principales de este arquitecto, mas cuando se hable de un proyecto de envergadura donde exista un equipo completo de datos. • Debe ser una persona responsable con sus tareas, estudioso y que se mantenga actualizado

	siempre en el ámbito de las base de datos.
Arquitecto de Tecnología.	<ul style="list-style-type: none"> • Amplios conocimientos de lenguajes y paradigmas de programación. • Amplios conocimientos de las características tecnológicas y arquitectónicas de las plataformas y tecnologías asociadas al proceso o marco institucional del tipo de producto de software en que se trabaja. • Alta capacidad de asimilación de lenguajes y tecnologías. • Amplio conocimiento de herramientas de modelado y diseño arquitectónico. • Deben ser programadores técnicos avanzados que además de tener las habilidades de un especialista técnico de latísimo nivel debe tener características de liderazgo y gestión, con poder y autoridad; deben tener capacidad de elocuencia y convencimiento sobre el equipo de desarrollo que trabaje. • Altos conocimientos de los dominios de solución y paradigmas más avanzados de la tecnología del software y ramos asociados, como la telemática, las telecomunicaciones y los dispositivos tecnológicos a fines con el tipo de software o producto que se produce. • Dominio de las técnicas de negociación. • Alta capacidad de abstracción. • Liderazgo. • Conocimientos sobre estilos arquitectónicos, patrones arquitectónicos y patrones de diseño. • Altos conocimientos de modelos y metodologías de desarrollo. • Habilidades gerenciales. • Debe estar capacitado para asistir a la organización en la selección de la arquitectura tecnológica más adecuada para responder a las necesidades del desarrollo técnico del producto de software. • Conoce las notaciones de modelados existentes ADL, UML, IDEFF, BPMN.

Anexo 4. Encuesta de autovaloración.

ENCUESTA DE AUTOVALORACIÓN PARA DETERMINAR EL COEFICIENTE DE COMPETENCIA DE LOS EXPERTOS

Compañero(a):

En la ejecución de la presente tesis, se desea someter a la valoración de un grupo de expertos, la propuesta de Modelo para la gestión de la ASW de un sistema de planificación empresarial para evaluar el sistema de gestión que se establece en la propuesta, los responsables seleccionados para llevar a cabo la gestión de la arquitectura así como las técnicas y herramientas que se necesitan para el desarrollo.

Para ello se necesita conocer el grado de dominio que usted posee acerca del tema de investigación y con ese propósito se desea que responda lo que se le pide a continuación.

Nombre (s) y Apellidos: _____

Centro de trabajo: _____

Labor que realiza: _____

Años de experiencia: _____

Calificación profesional: ___ Ingeniero ___ Licenciado en Educación ___ Máster ___ Doctor

___ Categoría docente: ___ Prof. Instructor ___ Prof. Asistente ___ Prof. Auxiliar ___ Prof. Titular ___ Prof. ___ Adjunto

1. Seleccione en una escala del 1 al 10 el valor que corresponda con el grado de conocimientos que usted posee acerca del tema de investigación que se desarrolla: la arquitectura de software, teniendo en cuenta que 1 representa no tener ningún conocimiento y 10 el pleno conocimiento de la problemática tratada.

1	2	3	4	5	6	7	8	9	10

2. Seleccione en una escala del 1 al 10 el valor que corresponda con el grado de conocimientos que usted posee acerca del tema de investigación que se desarrolla: gestión de software, teniendo en cuenta que 1 representa no tener ningún conocimiento y 10 el pleno conocimiento de la problemática tratada.

1	2	3	4	5	6	7	8	9	10

3. Valore el grado de influencia que cada una de las fuentes de argumentación que se le presenta a continuación ha tenido en su conocimiento y criterios sobre el tema que se investiga.

FUENTES DE ARGUMENTACIÓN	Grado de influencia de cada fuente		
	Alto	Medio	Bajo
Análisis realizado por Ud. Respecto al tema			
Experiencia obtenida en relación con el tema de investigación			
Trabajos investigados de autores extranjeros			
Trabajos investigados de autores nacionales			
Su propio conocimiento del estado del problema en el extranjero			
Su intuición			

Anexo 5. Encuesta de validación.

ENCUESTA A EXPERTOS PARA LA VALIDACIÓN DEL MODELO

Compañero(a):

La presente encuesta forma parte de la aplicación del Método de Valoración de Expertos. Se solicita su valiosa colaboración para evaluar si las actividades, roles, artefactos así como las técnicas y herramientas que se propusieron son correctos. Para darle cumplimiento a este objetivo se han elaborado un conjunto de preguntas que permiten medir la eficacia del modelo. Se le garantiza a usted que nadie podrá saber quién es el encuestado además se garantiza que sus opiniones se tendrán en cuenta para la posterior aplicación del modelo propuesto.

Valore el grado de factibilidad de cada pregunta de acuerdo a la siguiente escala:

Muy Adecuado (C1)

Bastante Adecuado (C2)

Adecuado (C3)

Poco Adecuado (C4)

No Adecuado (C5)

Preguntas	Criterio de expertos				
	C1	C2	C3	C4	C5
1. Categorice cada una de las actividades que propone el modelo:					
1.1 Analizar artefactos generados en el Negocio.					
1.2 Identificar estructura de empaquetamiento.					
1.3 Identificar los atributos de calidad.					
1.4 Definir escenarios arquitectónicos.					
1.5 Identificar los componentes horizontales de la arquitectura.					
1.6 Realizar taller de solución.					
1.7 Discutir plan de estimación y planificación del proyecto.					
1.8 Realizar ajustes de la Arquitectura de Sistema propuesta.					
1.9 Realizar pruebas unitarias a la solución.					
1.10 Evaluación del rendimiento del servidor.					
2. Categorice los roles propuestos para llevar a cabo la implantación del modelo.					
2.1 Arquitecto Corporativo o Principal: es el encargado de identificar las tareas de implementación, las necesidades tecnológicas, los hitos tecnológicos y la línea base. Es el máximo responsable de las decisiones arquitectónicas. Es el encargado de decidir cómo tratar los temas de reutilización así como los servicios tecnológicos al desarrollo y construcción tecnológica del producto. Orienta de forma estratégica la organización de la producción de software.					

2.2 Líder de Gestión: es el encargado del monitoreo y control de las tareas, la gestión del capital humano, la gestión de riesgos e incidencias. Lleva la planificación y gestión de los recursos del proyecto. Además de gestionar las necesidades técnicas del proyecto.					
2.3 Arquitecto de Sistema: es el encargado de establecer y velar que se cumplan las políticas y estándares definidos en la Arquitectura, lleva las decisiones de integración en el proyecto y la Arquitectura del Sistema, así como moderar el Taller de Diseño.					
2.4 Arquitecto de Datos: es el encargado de construir la Vista de Datos, establece la integración de la Arquitectura de Sistema en la dimensión de la Arquitectura de Datos. Vela por el cumplimiento de los elementos metodológicos de desarrollo, políticas, estándares, diseño y configuración definidos para la base de datos.					
2.5 Arquitecto de Tecnología: es el encargado de estudiar y determinar la factibilidad técnica desde el punto de vista de tecnología del producto. Define y especifica la arquitectura de tecnología. Realiza el estudio tecnológico de las plataformas y realiza el dictamen técnico de las mismas. Define el estilo arquitectónico vertical de la arquitectura, es decir la vista de tecnología.					
3. La propuesta de utilizar los talleres y las reuniones como técnicas para llevar la gestión del modelo es:					
4. Categorice los artefactos propuestos para llevar a cabo la gestión de la arquitectura.					
4.1 Minuta de acuerdos: este artefacto contiene los puntos tratados, acuerdos tomados, participantes, lugar y tema abordado en el desarrollo del taller de conceptualización.					
4.2 Sprint Backlog: este artefacto contiene el listado de hitos que se debe realizar para la(s) iteración(es) (<i>Sprint</i>) en un período de tiempo definido.					
5. Categorice las herramientas propuestas para llevar a cabo la gestión de la arquitectura.					
5.1 Redmine: es una herramienta de monitoreo y control que permite llevar la gestión del proyecto. Permite definir los hitos y las tareas a realizar para cumplir cada uno de ellos, así como asignar esas tareas a cada uno de los responsables de darle cumplimiento. Posibilita llevar el tiempo de estimación, cronogramas de trabajo, el tiempo dedicado y el porcentaje de avance de las tareas asignadas; además del control de incidencias y riesgos.					
5.2 Alfresco: es una herramienta que facilita la búsqueda y obtención de documentos, gestiona el ciclo de vida de estos documentos en un repositorio centralizado. Representa un sistema de gestión documental en las empresas garantizando la seguridad y permitiendo la colaboración entre equipos.					

6. Categorice el desarrollo de la identificación de desviaciones y riesgos, así como la gestión de estos.					
7. Se le pide su criterio acerca de la utilidad que puede tener la aplicación del modelo propuesto, para llevar la gestión de la arquitectura.					
8. Expresé otros criterios o recomendaciones que pudieran servir para perfeccionar el modelo propuesto:					

Gracias por su cooperación.

Anexo 6. Cálculo del coeficiente de concordancia de Kendall.

	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Experto 6	Experto 7	Experto 8	Experto 9	RJ
Preg 1.1	5	5	5	5	5	5	3	5	3	41
Preg 1.2	5	4	5	5	5	4	4	5	4	41
Preg 1.3	5	4	5	5	5	5	2	4	5	40
Preg 1.4	5	5	5	5	4	5	1	5	4	39
Preg 1.5	5	5	5	5	5	5	2	5	4	41
Preg 1.6	5	5	4	4	5	4	3	5	4	39
Preg 1.7	4	5	5	5	4	5	1	4	3	36
Preg 1.8	5	4	5	5	5	5	3	4	4	40
Preg 1.9	5	4	5	4	5	5	1	4	4	37
Preg 1.10	4	3	4	4	4	4	2	4	5	34
Preg 2,1	5	4	5	5	4	4	2	5	5	39
Preg 2,2	5	5	5	5	5	4	2	3	5	39
Preg 2,3	5	5	3	5	5	4	2	5	3	37
Preg 2,4	5	5	5	5	5	4	2	5	5	41
Preg 2,5	4	5	2	5	5	5	3	4	4	37
Preg 3	5	3	3	5	5	3	1	5	4	34
Preg 4,1	5	5	4	5	4	5	4	5	3	40
Preg 4,2	5	4	5	5	5	5	2	5	4	40
Preg 5,1	5	4	4	2	4	3	1	4	3	30
Preg 5,2	4	4	4	4	2	5	2	3	3	31
Preg 6	5	3	4	5	4	5	3	5	4	38
Preg 7	5	4	4	5	4	4	2	4	4	36

K es el número de expertos que intervienen en el proceso de validación, por lo que toma el valor 9.

N cantidad de aspectos a validar. En este caso es N = 22.

Rj es la suma de los rangos asignados a cada pregunta por parte de los expertos.

\bar{R}_j es la media de los rangos y se determina mediante la siguiente fórmula:

$$\bar{R}_j = \frac{\sum_{j=1}^n R_j}{N}$$

Obteniendo el valor:

$$\bar{R}_j = \frac{4(41) + 4(40) + 4(39) + 2(36) + 3(37) + 2(34) + 30 + 31 + 38}{22} = 37,73$$

$$\bar{R}_j = \frac{4(41)+4(40)+4(39)+2(36)+3(37)+2(34)+30+31+38}{22} = 37,73$$

S es la suma de los cuadrados de las desviaciones y se calcula de la siguiente forma:

$$S = \sum_{j=1}^n (R_j - \bar{R}_j)^2$$

Donde S = 210,36.

W es el coeficiente de Kendall y se calcula mediante la fórmula siguiente:

$$W = \frac{12 * S}{K^2(N^3 - N)}$$

Sustituyendo los valores obtenidos en la ecuación

Luego se procede con el cálculo del Chi-Cuadrado para poder ver si existe concordancia entre expertos:

$$X^2 = K(N-1)W = 9(22-1)0,002933 = 0,554318$$

Este Chi-Cuadrado se compara con el de la tabla inversa de la función de distribución de la variable Chi-Cuadrado con una probabilidad de error de 0,01. Consultar Anexo 7. Distribución de Chi-Cuadrado..

Si el Chi-Cuadrado real (X^2 real) es menor que el Chi-Cuadrado de la tabla ($X^2(\alpha, N - 1)$) entonces hay concordancia:

$$X^2 \text{ real} < X^2(\alpha, N - 1)$$

$$X^2 \text{ real} < X^2(0, 01, 21)$$

$$0, 554318 < 38, 9322$$

Anexo 7. Distribución de Chi-Cuadrado.

v/p	0,001	0,0025	0,005	0,01	0,025	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
1	10,8274	9,1404	7,8794	6,6349	5,0239	3,8415	2,7055	2,0722	1,6424	1,3233	1,0742	0,8735	0,7083	0,5707	0,4549
2	13,8150	11,9827	10,5965	9,2104	7,3778	5,9915	4,6052	3,7942	3,2189	2,7726	2,4079	2,0996	1,8326	1,5970	1,3863
3	16,2660	14,3202	12,8381	11,3449	9,3484	7,8147	6,2514	5,3170	4,6416	4,1083	3,6649	3,2831	2,9462	2,6430	2,3660
4	18,4662	16,4238	14,8602	13,2767	11,1433	9,4877	7,7794	6,7449	5,9886	5,3853	4,8784	4,4377	4,0446	3,6871	3,3567
5	20,5147	18,3854	16,7496	15,0863	12,8325	11,0705	9,2363	8,1152	7,2893	6,6257	6,0644	5,5731	5,1319	4,7278	4,3515
6	22,4575	20,2491	18,5475	16,8119	14,4494	12,5916	10,6446	9,4461	8,5581	7,8408	7,2311	6,6948	6,2108	5,7652	5,3481
7	24,3213	22,0402	20,2777	18,4753	16,0128	14,0671	12,0170	10,7479	9,8032	9,0371	8,3834	7,8061	7,2832	6,8000	6,3458
8	26,1239	23,7742	21,9549	20,0902	17,5345	15,5073	13,3616	12,0271	11,0301	10,2189	9,5245	8,9094	8,3505	7,8325	7,3441
9	27,8767	25,4625	23,5893	21,6660	19,0228	16,9190	14,6837	13,2880	12,2421	11,3887	10,6564	10,0060	9,4136	8,8632	8,3428
10	29,5879	27,1119	25,1881	23,2093	20,4832	18,3070	15,9872	14,5339	13,4420	12,5489	11,7807	11,0971	10,4732	9,8922	9,3418
11	31,2635	28,7291	26,7569	24,7250	21,9200	19,6752	17,2750	15,7671	14,6314	13,7007	12,8987	12,1836	11,5298	10,9199	10,3410
12	32,9092	30,3182	28,2997	26,2170	23,3367	21,0261	18,5493	16,9893	15,8120	14,8454	14,0111	13,2661	12,5838	11,9463	11,3403
13	34,5274	31,8830	29,8193	27,6882	24,7356	22,3620	19,8119	18,2020	16,9848	15,9839	15,1187	14,3451	13,6356	12,9717	12,3398
14	36,1239	33,4262	31,3194	29,1412	26,1189	23,6848	21,0641	19,4062	18,1508	17,1169	16,2221	15,4209	14,6853	13,9961	13,3393
15	37,6978	34,9494	32,8015	30,5780	27,4884	24,9958	22,3071	20,6030	19,3107	18,2451	17,3217	16,4940	15,7332	15,0197	14,3389
16	39,2518	36,4555	34,2671	31,9999	28,8453	26,2962	23,5418	21,7931	20,4651	19,3689	18,4179	17,5646	16,7795	16,0425	15,3385
17	40,7911	37,9462	35,7184	33,4087	30,1910	27,5871	24,7690	22,9770	21,6146	20,4887	19,5110	18,6330	17,8244	17,0646	16,3382
18	42,3119	39,4220	37,1564	34,8052	31,5264	28,8693	25,9894	24,1555	22,7595	21,6049	20,6014	19,6993	18,8679	18,0860	17,3379
19	43,8194	40,8847	38,5821	36,1908	32,8523	30,1435	27,2036	25,3289	23,9004	22,7178	21,6891	20,7638	19,9102	19,1069	18,3376
20	45,3142	42,3358	39,9969	37,5663	34,1696	31,4104	28,4120	26,4976	25,0375	23,8277	22,7745	21,8265	20,9514	20,1272	19,3374
21	46,7963	43,7749	41,4009	38,9322	35,4789	32,6706	29,6151	27,6620	26,1711	24,9348	23,8878	22,8876	21,9915	21,1470	20,3372
22	48,2676	45,2041	42,7957	40,2894	36,7807	33,9245	30,8133	28,8224	27,3015	26,0393	24,9390	23,9473	23,0307	22,1663	21,3370
23	49,7276	46,6231	44,1814	41,6383	38,0756	35,1725	32,0069	29,9792	28,4288	27,1413	26,0184	25,0055	24,0689	23,1852	22,3369
24	51,1790	48,0336	45,5834	42,9798	39,3641	36,4150	33,1962	31,1325	29,5533	28,2412	27,0960	26,0625	25,1064	24,2037	23,3367
25	52,6187	49,4351	46,9280	44,3140	40,6465	37,6525	34,3816	32,2825	30,6752	29,3388	28,1719	27,1183	26,1430	25,2218	24,3366
26	54,0511	50,8291	48,2898	45,6416	41,9231	38,8851	35,5632	33,4295	31,7946	30,4346	29,2463	28,1730	27,1789	26,2395	25,3365
27	55,4751	52,2152	49,6450	46,9628	43,1945	40,1133	36,7412	34,5736	32,9117	31,5284	30,3193	29,2266	28,2141	27,2569	26,3363
28	56,8918	53,5939	50,9936	48,2782	44,4608	41,3372	37,9159	35,7150	34,0266	32,6205	31,3909	30,2791	29,2486	28,2740	27,3362
29	58,3006	54,9662	52,3355	49,5878	45,7223	42,5569	39,0875	36,8538	35,1394	33,7109	32,4612	31,3308	30,2825	29,2908	28,3361

Anexo 8. Plantilla del artefacto minuta de acuerdos.

Autor

[Nombre y apellido del autor]

Lugar

[Lugar o local en donde se efectúa la reunión]

Participantes

[Nombre y apellidos de los participantes, puede incluir el cargo que ocupa.]

Puntos tratados

[Temas o puntos a debatir]

Acuerdos

[Acuerdos tomados durante el desarrollo de la actividad, incluye fecha de cumplimiento y responsable]

Actividades a realizar

[Son aquellas tareas que no se toman como acuerdos pero sería de interés dejarlas plasmada para darle cumplimiento, incluye responsable y fecha de cumplimiento.]

Anexo 9. Plantilla del artefacto Sprint Backlog

N° Sprint

[Es el número de iteración a realizar]

Listado de hitos o tareas

[Es el listado de hitos o tareas a cumplir y desarrollar durante el Sprint]

PD: El resto de las plantillas de los artefactos se encuentran en la versión digital en los anexos.