

Universidad de las Ciencias Informáticas

Facultad 10



Título: Manual del Arquitecto de Software.

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

Autor(es): Annie Leyva Simón / Devorah Lisset Covas Ramos.

Tutor(es): Yusleydi Fernández del Monte / Sonia Guerrero Lambert.

Co-tutor: Dayron Hernández Ramos.

Junio 2009



“Pero la juventud tiene que crear. Una juventud que no crea es una anomalía realmente.”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado:

Manual del Arquitecto del Software.

Y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Devorah Lisset Covas Ramos.

Annié Leyva Simón.

Yusleydi Fernández del Monte.

Sonia Guerrero Lambert.

Tutor: Ing. Yusleydi Fernández del Monte.

- Graduada en el año 2008 de Ingeniero en Ciencias Informáticas.
- Tiene 4 años de experiencia en el trabajo vinculado con la Calidad de Software.
- Tiene 1 año de experiencia en el trabajo vinculado con la Gestión de Conocimientos.
- Tiene 1 año de experiencia en la docencia universitaria.
- Ha publicado y realizado presentaciones en eventos sobre temas de gestión de conocimientos y calidad de software.
- Se desempeña como profesora de la facultad 10

Tutor: Ing. Sonia Guerrero Lambert.

- Graduada en el año 2008 de Ingeniero en Ciencias Informáticas.
- Tiene 4 años de experiencia en el trabajo vinculado con la Calidad de Software.
- Tiene 1 año de experiencia en el trabajo vinculado con la Gestión de Conocimientos.
- Tiene 1 año de experiencia en la docencia universitaria.
- Ha publicado y realizado presentaciones en eventos sobre temas de gestión de conocimientos y calidad de software.
- Se desempeña como profesora de la facultad 10

Co-Tutor: Ing. Dayron Hernández Ramos.

- Graduado en el año 2007 de Ingeniero en Ciencias Informáticas.
- Ha participado como programador en diferentes proyectos de la Universidad y ha realizado presentaciones en eventos relacionados con el tema, los cuales relacionamos a continuación:
 - Seminario “El desarrollo de las infocomunicaciones en la República Rusa”, 2008.
 - Programador de bases de datos en la primera fase del proyecto Identidad 2005-2008.
 - Análisis, Diseño y Programación de la primera versión en Java del sistema para el Análisis y Monitoreo de Negocios. 2007-2008
 - Feria internacional de La Habana 2009.
 - Programación de la fase 1 del proyecto Identidad. 2005-2008
- Se desempeña como especialista en el grupo de Arquitectura y desarrollo de software de la Dirección Técnica – IP.

AGRADECIMIENTOS

Quisiera agradecer en primer lugar a nuestras tutoras Yusleydi Y Sonia por su apoyo, guía y esfuerzo personal, al igual que a nuestro cotutor Dayron quien fue la luz en momentos de oscuridad. A mi compañera de tesis por estar a la altura. A todos los que nos ayudaron con sus consejos quisiera agradecer. A mi tía Onelia por su ayuda. A mi hermana Ana por sus recomendaciones. A Raciél que me enseñó a estudiar. A la Vida, a la Revolución, a la UCI, a la Facultad X y mis profes por transmitirme los conocimientos necesarios para mi futura vida profesional, por darme esta oportunidad de estudiar y llegar a ser Ingeniera.

Devorah Lisset Covas.

Dedico todo mi esfuerzo y empeño plasmado aquí al Amor de mis Amores: Mi Madre Querida, que ha sabido quererme a pesar de mis defectos, que me ha sobrellevado y sacado adelante por encima de las miles de dificultades que hemos atravesado juntas en la vida, a ella que es parte de mí ser y me ha robado el corazón.

A mi padre, por darme su apoyo y siempre exigirme tan rigurosamente en el estudio.

A mi hermanita, que a pesar de ser seis años mayor que yo siempre me he creído su hermana mayor.

A Ale, por su paciencia y por ser la persona que yo quisiera ser en determinados momentos.

A Reina por su incondicionalidad, por ser más que mi suegra, mi amiga.

A mi suegro, que a través de sus majaderías me ha dado buenas lecciones.

A mi abuela Santiago por ayudarme tanto, a mis tías y tíos, a mis primos, en general a toda mi familia.

A Ana y Annié que son las otras hermanas que nunca tuve.

A Yiyi porque más que prima es amiga.

A Arletis por darme su tutela.

A mis amigos de la facultad cuatro.

A Lili y Abu.

A Vilmita por ser inigualable y por su ejemplo.

A mi antiguo y actual grupo, el primero por ser las personas lindas y especiales que son y el segundo por tener el nivel que tienen y enseñarme que uno nunca debe quedarse atrás y debe luchar por ser mejor cada día, por superarse y estar a la altura de los que te rodean.

A mis alumnos del grupo 10103 que dentro de cinco cortos años serán los ingenieros de la 7ma graduación UCI, por enseñarme a ver el lado de los profes. También a mis tres anteriores grupos.

A la memoria de mis seres queridos que ya no están entre nosotros, pero son inolvidables.

Devorah Lisset Covas Ramos.

RESUMEN

Con el propósito de erradicar los errores inesperados que surgen por la falta de conocimientos en la realización de la arquitectura de un producto determinado la presente investigación centra como objeto de estudio el proceso de desarrollo de software teniendo como objetivo elaborar un manual con las funciones del arquitecto de software para proyectos de la Universidad de las Ciencias Informáticas dirigido al que se desempeñe como arquitecto de software. Para ello se utilizaron diferentes métodos de investigación como el analítico - sintético, el histórico - lógico, el inductivo – deductivo, la entrevista, la encuesta, entre otros. Como resultado se obtuvo un manual que tiene como finalidad brindar conocimientos básicos de cómo desempeñar las tareas fundamentales a personas de poca o ninguna experiencia en este rol, sin importar una metodología en específico, sino que se enseñarán las actividades del rol de forma genérica. En el mismo se recoge el estado del arte del rol, las responsabilidades y competencias del mismo, se identifican los elementos que integran la Arquitectura de software(AS) donde se trata la importancia e influencia que ejercen los requerimientos no funcionales sobre la misma, se realiza el establecimiento de la línea base de la arquitectura, además se explica la importancia de documentar la arquitectura, se trata el tema de valorar la viabilidad de la arquitectura entre otros. Después de validar la propuesta planteada se demostró que es una solución eficiente que permite una mayor transmisión e innovación de conocimientos utilizando una muestra aleatoria de 30 estudiantes de la facultad 10.

PALABRAS CLAVE

Arquitectura de Software, manual.

TABLA DE CONTENIDO

TABLA DE CONTENIDOS	
AGRADECIMIENTOS	¡ERROR! MARCADOR NO DEFINIDO.
DEDICATORIA	¡ERROR! MARCADOR NO DEFINIDO.
RESUMEN	¡ERROR! MARCADOR NO DEFINIDO.
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	¡ERROR! MARCADOR NO DEFINIDO.
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	21
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	36
CONCLUSIONES	¡ERROR! MARCADOR NO DEFINIDO.
RECOMENDACIONES	57
BIBLIOGRAFÍA	59
ANEXOS	61
GLOSARIO	¡ERROR! MARCADOR NO DEFINIDO.

INTRODUCCIÓN

La Informática como ciencia se ha desarrollado en los últimos años, convirtiéndose en el centro de cualquier acción dentro de la sociedad moderna. No se concibe en la actualidad alguna nación apartada de la misma, se ha apoderado de ramas tan imprescindibles como las comunicaciones, los medios de difusión masiva, la educación y hasta de la salud donde observamos novedosas técnicas sustentadas en esta ciencia. Muchos de los logros anteriormente mencionados se deben en gran escala a los nuevos y numerosos aportes científicos hechos al desarrollo del Software, sobre el cual pesa gran responsabilidad ya que este debe poseer una arquitectura sólida que soporte el correcto funcionamiento de los sistemas usados en dichas ramas. Para lograr esto, se requiere de profesionales con experiencia y conocimiento suficiente para diseñar dicha arquitectura, estos profesionales son conocidos como los arquitectos del Software.

El rol del arquitecto, asegura que se alcance el máximo de robustez en la arquitectura del software, así como generar bosquejos, modelos, guías de usuario y prototipos que mantengan en acuerdo y al corriente a los ingenieros con los usuarios, es el individuo que le da forma desde diferentes perspectivas al producto que se desea elaborar.

A la hora de desarrollar la arquitectura de un sistema estos profesionales deben tener presente un conjunto de aspectos teniendo en cuenta que debe ser flexible y debe permitir su evolución, además de estar relacionada profundamente con los requerimientos que se determinen, ya que estos representan la funcionalidad de una aplicación determinada.

Según Philippe Kruchten¹: “La arquitectura de software, 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.”

¹ Philippe Kruchten:

Licenciado en Ingeniería Mecánica de la Ecole Centrale de Lyon en Francia.

Doctor en Ciencias de la Computación en el Instituto Francés de Telecomunicaciones.

En la Universidad de las Ciencias Informática (UCI), este rol es ejercido por muchos estudiantes y profesionales, que a partir del análisis y estudio de los resultados, sobre la base de las técnicas aplicadas en esta investigación, se aprecian los siguientes problemas:

- Los arquitectos no cuentan con suficiente bibliografía específica sobre el tema.
- La poca información con que cuentan los arquitectos varias veces no esta organizada correctamente.
- Los arquitectos no reciben capacitación para realizar su trabajo.
- En la UCI cuando surge un proyecto, el arquitecto y el equipo de trabajo determinan la arquitectura a usar, es decir comienzan a crear una arquitectura desde cero, sin usar la que ya está hecha por otros proyectos y que se pueden reutilizar. Esto provoca que los proyectos trabajen aislados y en su mayoría de formas diferentes sin aprovechar la experiencia ya existente.
- En ocasiones cuando comienzan los proyectos los clientes no saben que quieren, como consecuencia la arquitectura comienza a conformarse con elementos erróneos por lo que es muy probable que tenga que cambiar con el paso del tiempo y el arquitecto tenga que trabajar doble ¿por qué? por no haberse efectuado un correcto levantamiento de requisitos.
- Los arquitectos no documentan los componentes que usan en sus proyectos y no comparten los mismos para que puedan ser usados por otros arquitectos.
- El arquitecto que va a conformar la arquitectura no cuenta con todos los elementos esto es resultado de la incorrecta captura de los requisitos no funcionales.
- En muchos proyectos el rol no es definido adecuadamente, o sea se nombra a cualquier integrante para que asuma esta responsabilidad o simplemente no se nombra a nadie.

Todas las insuficiencias antes expuestas traen como consecuencias:

1. Que se construyan conocimientos sobre como desempeñar el rol de arquitecto sobre una base que se corre el riesgo que sea insuficiente.
2. Aparición de errores inesperados por falta de conocimientos en la elaboración de un producto determinado, provocando así divorcios entre la forma y la funcionalidad del mismo.
3. Construcción de un producto que puede no ser flexible, y por tanto no admitir cambios.
4. Perdida de tiempo en la formación del arquitecto, lo cuál puede retrasar el trabajo planteado.
5. Mala gestión de conocimientos al no proporcionarse medios y ambientes necesarios para que puedan fluir los conocimientos correctamente.

6. La no reutilización de componentes.

Por tanto surge el **Problema científico:**

¿Cómo proporcionar al que desempeña el rol de arquitecto en proyectos de la UCI, información dirigida a realizar su trabajo?

Partiendo del problema anterior se plantea como **objetivo de la investigación:**

Elaborar un manual con las funciones del arquitecto de software para proyectos de la UCI dirigido al que desempeñe este rol.

La presente investigación tiene como **objeto de estudio:**

EL Proceso de desarrollo de software.

Como campo de acción:

El rol del Arquitecto en el proceso de desarrollo de software.

En función del desarrollo de la investigación se establecen las siguientes **preguntas científicas:**

1. ¿Cuál es la tendencia del desarrollo histórico de la arquitectura de software en Cuba y en el mundo?
2. ¿Qué teorías avalan las funciones de la arquitectura de software en el desarrollo de los proyectos?
3. ¿Cuáles son las funciones que debe desarrollar un arquitecto de software?
4. ¿Será posible que con el desarrollo de un manual con las funciones que debe cumplir un arquitecto de software, se logre una adecuada asimilación por parte del que desempeñe ese rol?

Consecuentemente se proponen las siguientes Tareas de investigación:

- Estudiar la evolución y tendencias del rol del arquitecto para conocer su comportamiento actual.
- Estudiar la evolución de la arquitectura de software (en lo adelante AS) como disciplina.
- Aplicar encuestas para investigar cuales son los problemas existentes a la hora de desempeñar el rol de arquitecto.
- Estudiar las actividades y responsabilidades que desempeña el arquitecto para construir una base de conocimientos relacionada con la labor que este realiza.
- Investigar los tipos de manuales existentes para tener argumentos suficientes a la hora de elaborar el manual del arquitecto.

- Investigar el contenido que debe incluirse en el manual del arquitecto para seleccionar la información que va a conformar dicho manual.
- Confeccionar el manual del arquitecto para cumplir el objetivo de la investigación.
- Aplicar encuestas y entrevistas a los especialistas en el tema orientadas a determinar los beneficios que aporta el manual del arquitecto en el proceso de desarrollo de software.

Para realizar dichas tareas de investigación se utilizarán los siguientes **métodos científicos**:

- **Métodos teóricos:**

Análisis y Síntesis: Permite realizar un estudio del problema científico, procesar la información obtenida, determinar algunos de los resultados y, además, elaborar las conclusiones del trabajo.

Histórico – Lógico: Permite durante la investigación el estudio de la evolución del arquitecto como rol y para determinar las tendencias actuales del mismo. Permite estudiar la evolución de la arquitectura de software.

Inducción–Deducción: Durante la investigación, permite el estudio del problema, la comprensión de las particularidades, así como la elaboración de un Manual del arquitecto que contenga sus responsabilidades y el modo que debe darle solución a cada una de sus tareas.

- **Métodos empíricos:**

Observación: Permite observar los problemas que existen. Para ello se establece una guía de observación (ver Anexo 1).

Entrevistas y encuestas: Dirigidas a profesores y estudiantes que desempeñan el rol de arquitecto para conocer sus criterios respecto al tema, así como indagar sobre los problemas existentes a la hora de instruirse para llevar a cabo sus tareas (ver Anexo 2).

- **Métodos Estadístico:**

Delphy: Permite validar si el manual elaborado resuelve el problema planteado. Posibilita realizar pronósticos y predicciones, además de permitir a los expertos llegar a un consenso basado en la discusión del tema. Es un proceso repetitivo. Su funcionamiento se basa en la elaboración de un cuestionario que ha de ser contestado por los expertos. Una vez recibida la información, se vuelve a realizar otro cuestionario basado en el anterior para ser contestado de nuevo. Finalmente permite al responsable del estudio elaborar sus conclusiones a partir de la explotación estadística de los datos obtenidos.

Población y muestra: La población está constituida por estudiantes que desempeñan el rol de arquitecto en los proyectos productivos de la facultad 10 en la UCI, de ellos se tomara una muestra significativa.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LAS FUNCIONES DEL ARQUITECTO DE SOFTWARE.

1. Introducción.

En el presente capítulo se hace un análisis del surgimiento de la arquitectura de software y del rol del arquitecto de software y sus tendencias, así como la descripción de los principales conceptos asociados. Se analizará el tipo de manual a usar de acuerdo a las necesidades propias del arquitecto de software.

2. Antecedentes históricos de la Arquitectura de Software (AS).

La Arquitectura de Software como disciplina es bastante joven, las primeras señales de su existencia se observaron en las reflexiones de Edsger Dijkstra en 1968, donde este propuso que se estableciera una estructuración correcta de los sistemas de software antes de lanzarse a programar, escribiéndose códigos de cualquier manera. En 1969 Fred Brooks Jr y Ken Iverson llamaban arquitectura a la estructura conceptual de un sistema en la perspectiva del programador. En 1971, C. R. Spooner tituló uno de sus ensayos “Una arquitectura de software para los 70s” [1]. Luego, en 1975, Frederick Phillips Brooks se refiere a la arquitectura como interfaz de usuario. Hasta la década de los 90’s el término fue visto de distintas maneras, sobretodo muy ligado al diseño. En el año 1992 se define la arquitectura de software como disciplina de software².

El primer estudio en que aparece la expresión “arquitectura de software” en el sentido en que hoy lo conocemos es de Perry y Wolf [2] en 1992.

En esta década surgieron numerosos aportes a la AS como lo son:

- Surgimiento de la programación basada en componentes.
- Surgimiento de los patrones.
- Surgimiento de los lenguajes de descripción de arquitectura (ADLs).
- Se consolidó la concepción de las vistas arquitectónicas.

2.1 Conceptos fundamentales de la Arquitectura de Software.

Una definición reconocida de la AS es la de Clements [Cle96a] en el año 1996 la cual plantea:

² Definido en la publicación “Foundations for the study of software architecture” escrito por Dewayne Perry y Alexander Wolf.

Finalmente en el año 2000 el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) lanza la definición oficial de arquitectura de software, también conocido como el primer estándar oficial para la arquitectura del software o la arquitectura del sistema, en su documento IEEE 1471, el cual plantea : “La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”.

En An Introduction to Software Architecture, David Garlan y Mary Shaw plantean:

“Más allá de los algoritmos y estructuras de datos de la computación; el diseño y la especificación de la estructura general del sistema emergen como una clase nueva de problema. Los aspectos estructurales incluyen la estructura global de control y la organización general; protocolos de comunicación, sincronización y acceso de datos; asignación de funciones para diseñar elementos; distribución física, composición de elementos de diseño; ajuste y rendimiento; y selección entre otras alternativas de diseño”.

Por otro lado el Rational Unified Process (RUP) plantea: “La arquitectura de software representa la estructura o las estructuras del sistema, que consta de componentes de software, las propiedades visibles externamente y las relaciones entre ellas. “

La AS es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes se 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.

Las autoras definen la Arquitectura de Software como el diseño de más alto nivel de la estructura de un sistema, de sus componentes y de las relaciones entre estos. Es el maquetado del software desde diferentes perspectivas, es llevar las necesidades del usuario a un sistema usando las tecnologías y componentes precisos para su cabal funcionamiento.

3. Arquitecto de Software.

Ante el nacimiento de la arquitectura de software como disciplina ha surgido la necesidad de crear un rol conocido como arquitecto de software o arquitecto de sistemas.

El rol del arquitecto es crítico en los proyectos, se focaliza en la calidad de servicio y lidera el proceso de definición y la implementación de la arquitectura. El arquitecto reutiliza implementaciones de arquitecturas

exitosas, frameworks y patrones de diseño y no es un diseñador en un proyecto. Un arquitecto es un facilitador, no toma decisiones unilaterales, irracionales, evita riesgos en los proyectos y agrega valor.

A diferencia de un programador, el arquitecto de software debe dominar la mayor cantidad de tecnologías de software y prácticas de diseño, para así poder tomar decisiones adecuadas para garantizar el mejor desempeño, reúso, robustez, portabilidad, flexibilidad, escalabilidad y mantenibilidad de las aplicaciones. Estas decisiones sobre la estructura y dinámica de la aplicación son plasmadas en una notación formal estandarizada como lo es UML.

Como base, el rol de los arquitectos suele comprender las tareas de: definición de las vistas de la arquitectura de una aplicación, dar soporte técnico-tecnológico a desarrolladores, clientes y expertos en negocios, conceptualizar y experimentar con distintos enfoques arquitectónicos, crear documentos de modelos y componentes y especificaciones de interfaces, validar la arquitectura contra requerimientos, suposiciones y además tener una dosis de estrategia y política, o sea, ser, en parte, un consultor (ver Figura 1).

3.1 Definición del Arquitecto de Software.

Por su significado etimológico la palabra arquitecto nos llega de los griegos, quienes bautizaron tal papel con la palabra *architécton* que define al director de una construcción. Esta palabra proviene de la unión de dos raíces muy fuertes *archós*, que significa guía y *técton* que significa constructor. Pero al español llegó gracias a los romanos que llamaron *Architectus*, a los grandes guías de las impresionantes y avanzadas obras civiles del imperio mas grande del mundo antiguo.

El arquitecto “es una persona, equipo u organización responsable por la arquitectura del sistema” (IEEE 1471).

“Este rol dirige el desarrollo de la arquitectura de software del sistema, que incluye la promoción y la creación de soporte para las decisiones técnicas claves que restringen el diseño global y la implementación para el proyecto.”³

Abogar por el cliente es la piedra angular del rol del arquitecto. Para lograr el rol de un verdadero abogado, el arquitecto necesita un extenso repertorio de elementos de diseño en un aspecto de elección libre de cualquier atadura. Un arquitecto deja de ser un verdadero apoyo al cliente si se encuentra atado a un conjunto de tecnologías, herramientas o metodologías, restringiendo así las soluciones disponibles al

³ Definición dada por RUP

cliente.



Figura 1 El rol del Arquitecto de Software relacionado con las personas, el proyecto, la organización y las tecnologías.

3.2 Responsabilidades del arquitecto de software según: Meter Eeles de IBM, Software Engineering Institute, Rational Unified Process-RUP SUN Microsystems y Bredemeyer Consulting.

- Servir de interfaz con los usuarios y patrocinadores, así como cualquier otro que está involucrado en determinar sus necesidades.
- Generar los niveles más altos de requisitos del sistema, basados en las necesidades del usuario, así como, algunas otras limitantes tales como costos y tiempos.
- Asegurar de que el conjunto de requisitos sea consistente, completo, correcto, y operacional.
- Llevar a cabo análisis costo-beneficio para determinar que costos se cumplen mejor ya sea manualmente, por software o con hardware; maximizando así los componentes ya desarrollados o los comerciales.
- Asegurar que se alcance el máximo de robustez de arquitectura.
- Generar bosquejos, modelos, guías de usuario y prototipos que mantengan en acuerdo y al corriente a los ingenieros con los usuarios.
- Elaborar la arquitectura correcta para solucionar el problema que se encuentra desarrollando.
- Definir y documentar la solución, asegurándose que este acorde con el sistema deseado y que además es la correcta para su soporte y evolución.
- Asegurar que todos los involucrados estén utilizando la solución elaborada y la estén utilizando bien.
- Conocer cuales cualidades sistémicas, deben alcanzarse y en que medida.
- Responder las inquietudes relacionadas con la selección de herramientas y ambientes de desarrollo.
- Resolver conflictos y ayuda a generar acuerdos.
- Gerenciar las estrategias de identificación y mitigación de los riesgos asociados con la arquitectura.
- Conocer y ayudar a la implementación de la Metodología.
- Conocer a la perfección los Requerimientos y Restricciones.
- Definir los estándares y políticas de reusabilidad de componentes.

3.3 Ética del arquitecto en su profesión.

Un arquitecto, al igual que todo rol debe poseer cualidades que lo distingua como profesional, debe actuar de un modo correcto y poseer principios éticos propios, los cuales son señalados a continuación:

- Aprender a confiar en las habilidades y conocimientos de los otros integrantes de su equipo.
- Ayudar inteligentemente en caso de que haya retrasos.
- Antes de criticar el trabajo de otros miembros del equipo, empezar identificando puntos buenos.
- Generalizar la crítica poniendo énfasis en las guías disponibles, sin criticar directamente una instancia particular del trabajo.
- No hablar con la gente solo cuando haya algo malo. Debe hacerle saber a la gente cuando las cosas están bien y su trabajo es apreciado.

3.4 Principales errores cometidos por los arquitectos.

Los responsables de desempeñar este rol comúnmente comenten los siguientes errores:

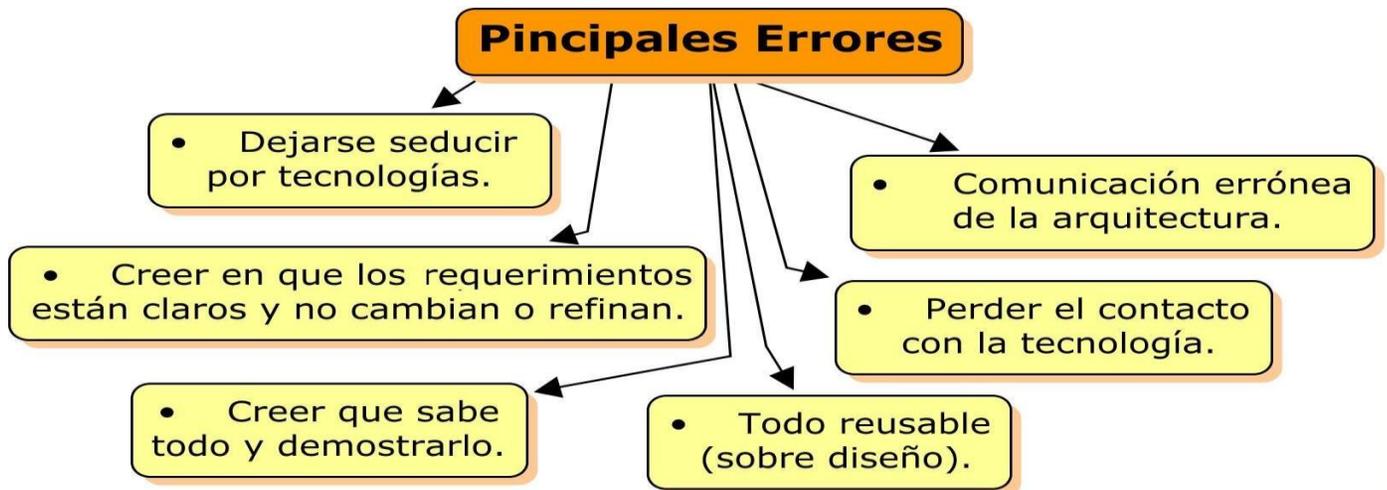


Figura 2 Errores en los cuales puede incurrir un arquitecto en el desempeño de sus tareas.

3.5 Metodologías.

La manera de plasmar correctamente el ciclo de vida del software, es a través de la utilización de metodologías que no son más que el camino a seguir para desarrollar software de forma sistemática. Las metodologías son una guía para desarrollar un proceso de software con un nivel de calidad elevado. Y pueden seguir uno o varios modelos del ciclo de vida, englobando un conjunto de métodos para abarcar el ciclo de vida completo. Según la metodología en la cual se apoye el desarrollo de un software, el arquitecto tendrá que realizar un grupo de determinadas tareas. Para dar respuesta al objetivo de la investigación (realización del manual) los autores no se basarán en ninguna metodología específica para la definición de las tareas del rol, sino que las establecerán de forma genérica para que todo aquel que vaya a desempeñar el rol del arquitecto, independientemente de la metodología en la que se base, sepa cómo realizar dichas tareas.

Dentro de las metodologías más usadas actualmente en la facultad 10 se encuentran RUP, SCRUM y XP, las cuales son reconocidas internacionalmente, además de estas se usan otras como por ejemplo Métricas V3. En la facultad mencionada, en los proyectos Unicornios del Polo de software libre se desarrolló una metodología conocida como SXP, la cual es usada por estos.

A continuación se muestran mapas conceptuales con la principal información de algunas de estas metodologías. La primera es una metodología pesada, conocida por sus siglas como:

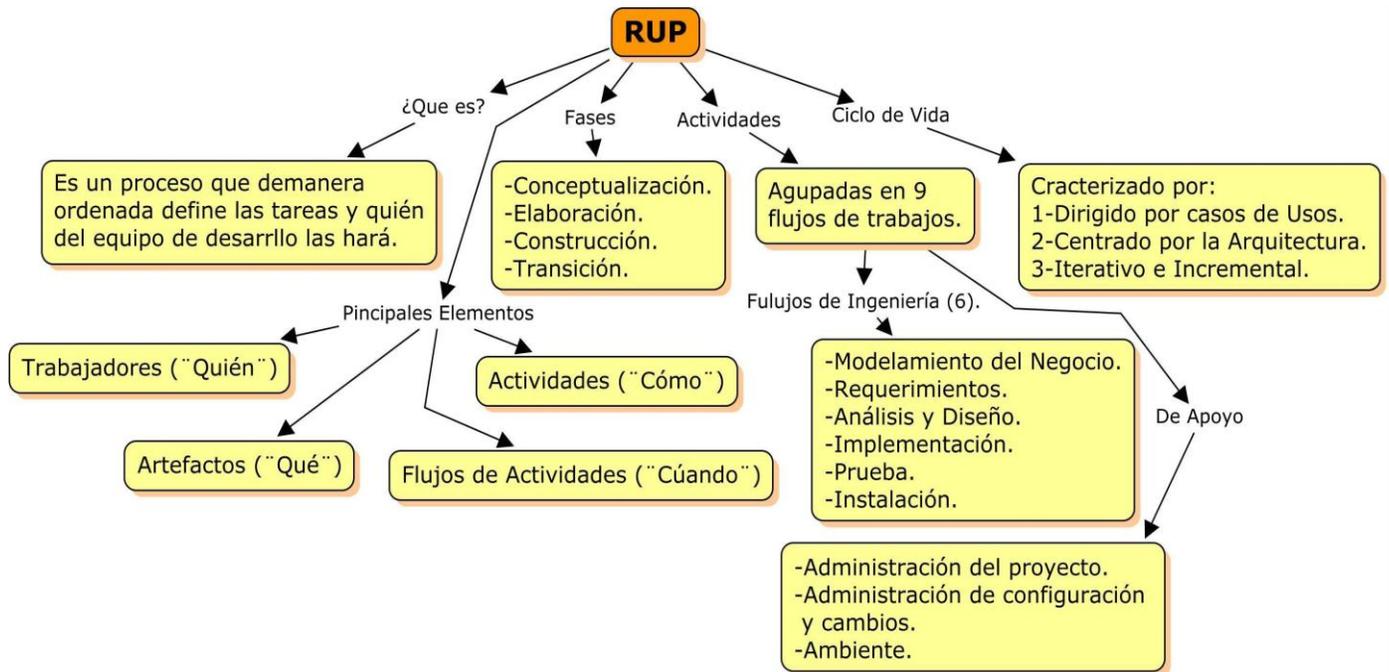


Figura 3 Metodología RUP (Proceso Unificado de Rational - Proceso Unificado de Desarrollo de Software).

Aunque la Figura 4 puede sugerir que los flujos de trabajo se desarrollan en cascada, la "lectura" de este gráfico tiene que ser vertical y horizontal. RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros en dependencia de la fase en que se encuentre.

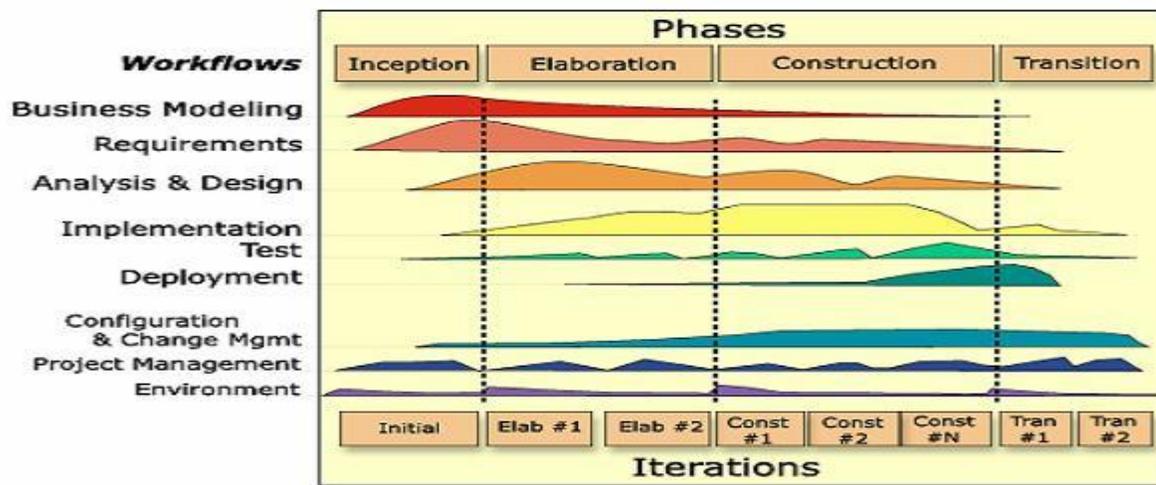


Figura 4 RUP en dos dimensiones.

RUP cuenta con un conjunto de roles, el arquitecto de software se encuentra específicamente en el conjunto de los desarrolladores. Otra metodología es la que mostraremos a continuación pero que en este caso no se trata de una pesada, sino de una ágil.

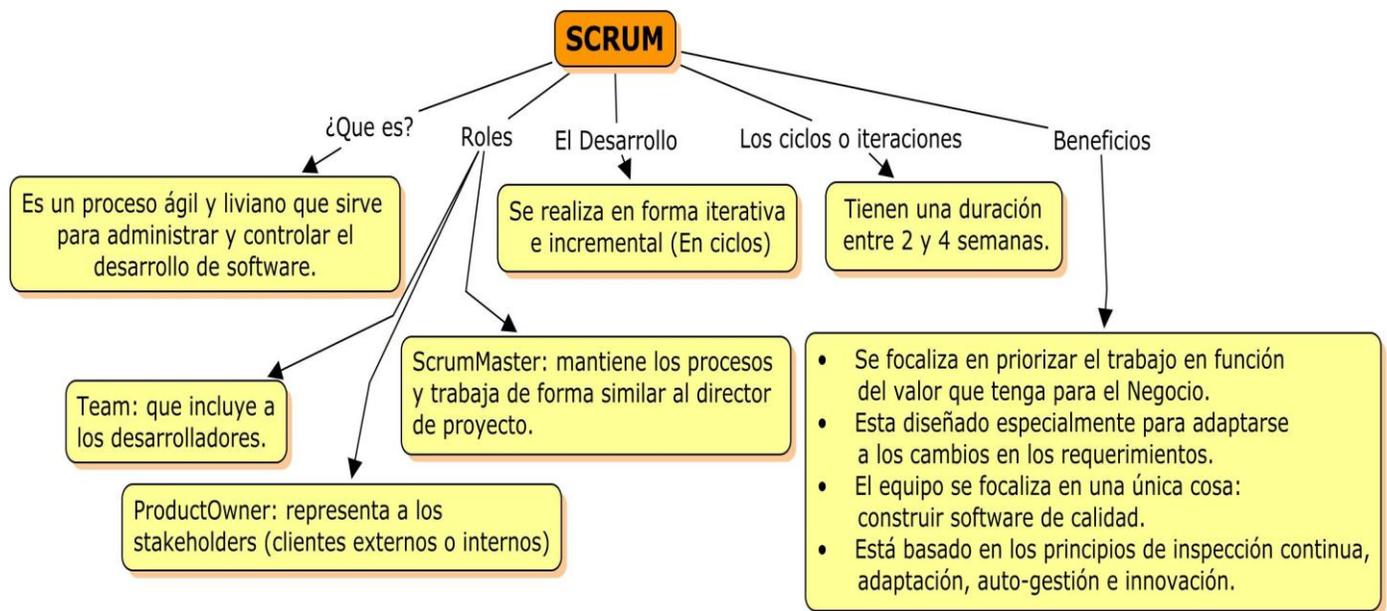


Figura 5 Metodología de desarrollo ágil SCRUM.

Por último, la metodología XP quizás la más usada y conocida actualmente de las ágiles.

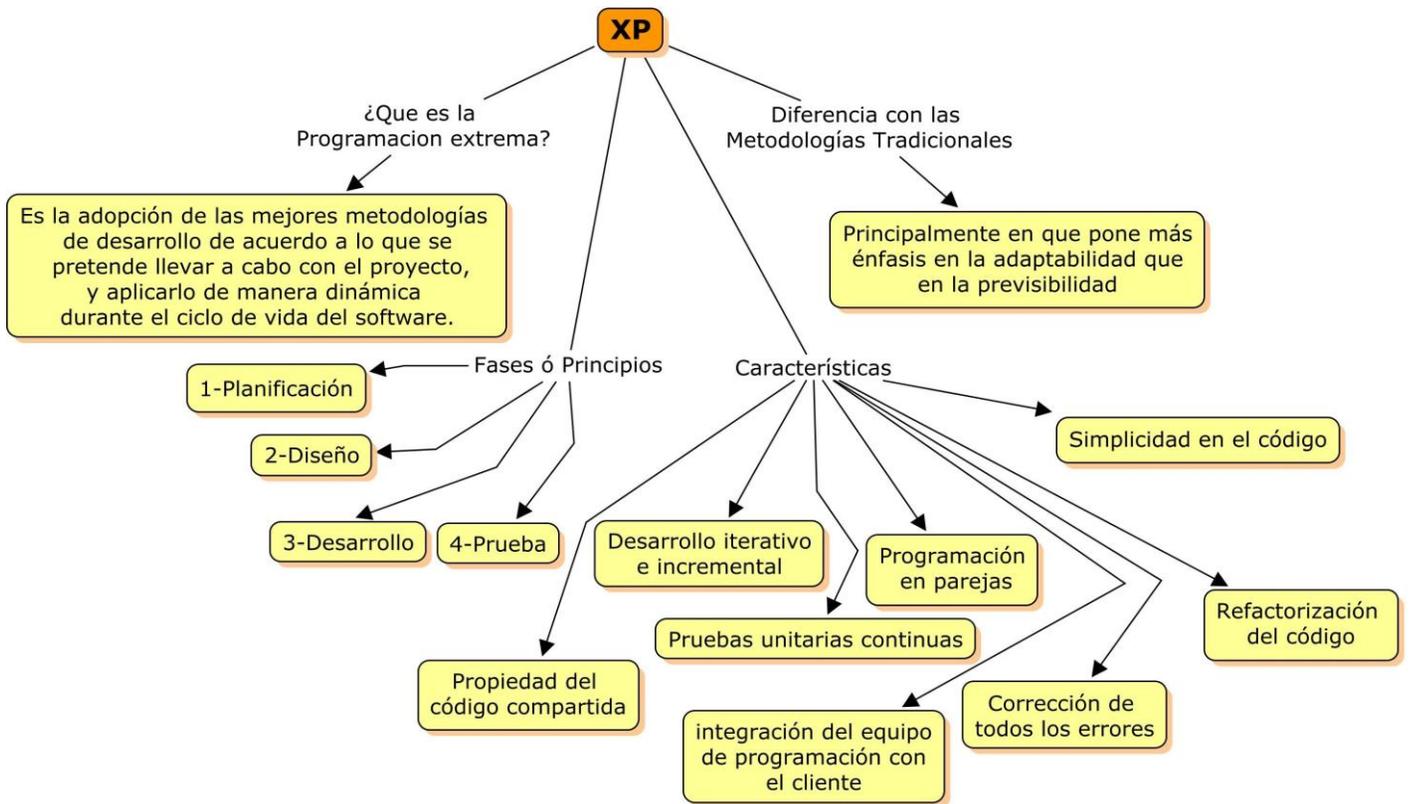


Figura 6 Metodología XP (Programación Extrema).

5. Entidades vinculadas al tratamiento de la AS en la UCI.

Enfocado al trabajo con la AS, la UCI cuenta con entidades cuya labor esta muy acorde a las tendencias actuales del mundo moderno en relación con esta rama, el actuar de las mismas puede servir de punto de partida para la realización del manual, por ejemplo en el mismo se aborda el tema de los estilos arquitectónicos y la **Dirección de Informatización de la UCI** guía el desarrollo de aplicaciones hacia una Arquitectura Empresarial Basada en Servicios (plasmado en sus lineamientos en <http://uddi.uci.cu/?q=arq>) lo cual es una tendencia de la AS o sea actualmente desarrollar arquitecturas con el estilo llamado Arquitectura Orientada a Servicios o conocido mayormente por sus siglas como SOA es muy recomendado.

Otra de estas entidades es la **Dirección Técnica** que es un grupo de trabajo que cuenta con un equipo de especialistas que se dedican a verificar la AS de los proyectos de la universidad, comprobando el uso

correcto de la metodologías empleadas en los mismos, que estén bien fundamentadas las herramientas usadas, además comprueban que en caso de usar algún framework se acople al diseño, en resumen estos especialistas se dedican a evaluar la arquitectura de los proyectos y esta actividad también forma parte de los temas que son tratados en el manual.

Otra entidad vinculada al tratamiento de la AS es el **Centro de Consultoría** el cual posee dentro de sus objetivos desarrollar modelos arquitectónicos para diferentes tipos de aplicaciones informáticas y establecer un proceso de desarrollo para la fase de implementación para cada modelo arquitectónico. También es su propósito dar soporte a los modelos arquitectónicos y componentes desarrollados. Además de formar arquitectos de software.

6. Fundamentos teóricos del manual.

Crear manuales supone recopilar los aspectos que permiten a un usuario utilizar con éxito y sin esfuerzo un dispositivo o entorno, y redactarlos de forma que indiquen cómo realizar tareas, incorporando los recursos visuales (imágenes, diagramas...) necesarios para que las explicaciones resulten más evidentes, adaptándose al nivel de conocimiento de sus destinatarios y con el lenguaje que le resulte más próximo. Escribir buenos manuales supone entender en profundidad por parte de quien los redacta el procedimiento que se está explicando, las circunstancias en las que se va a utilizar y ponerse en la situación de sus destinatarios, para comprender y cubrir sus necesidades de información. Existen diferentes tipos de manuales, que se agrupan de acuerdo a su contenido o función específica. El tipo de manual se determina dando respuesta al propósito que se ha de lograr con su uso.

6.1 Definición de manual

Conjunto de instrucciones y explicaciones escritas que cubren distintas materias como políticas, métodos, planes de cuentas, trabajos de un sistema de contabilidad o de un sistema de cómputo, documento guía de referencia como conocimiento básico en asuntos específicos.

6.2 Tipos de manuales.

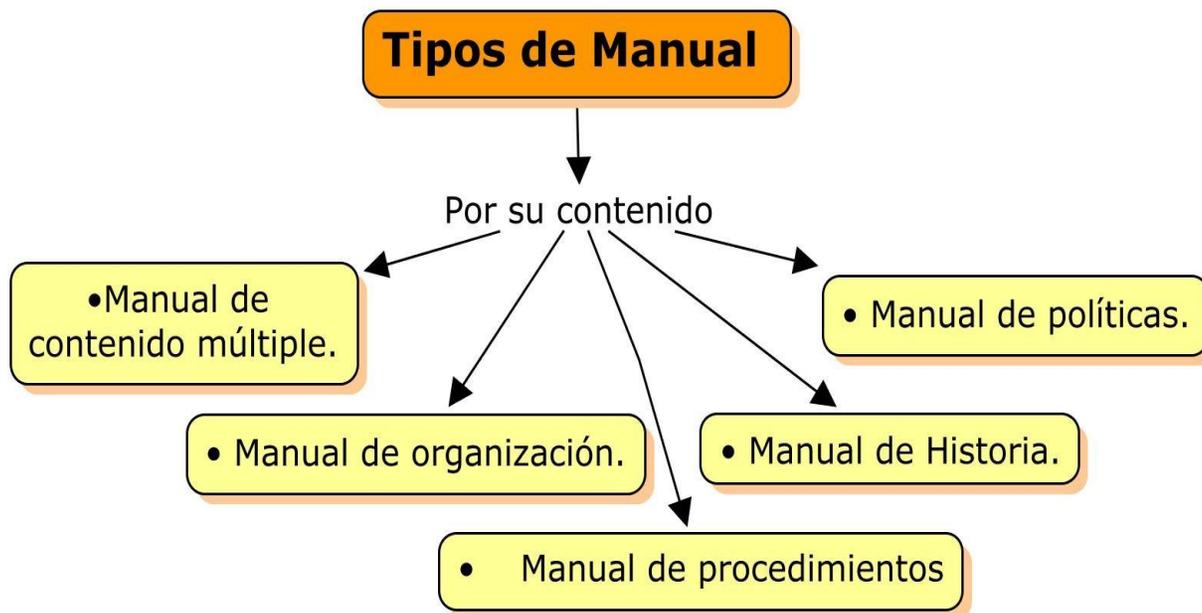


Figura 7 Tipos de manuales por su contenido.

7. Vinculación del manual al Sistema de Gestión de Conocimientos.

7.1 ¿Que es gestión de conocimientos?

Consiste en la capacidad de generar nuevos conocimientos a partir de datos y experiencias, además de organizar, distribuir y ordenar los conocimientos ya existentes en la misma entidad o sistema.⁴

7.2 Vinculación con el desarrollo del Manual del Arquitecto de Software.

El sistema de gestión de conocimiento tiene como objetivo principal brindar un modelo para gestionar conocimientos acerca de cualquier tema y brindar los ambientes y condiciones para que el conocimiento fluya eficazmente.

Para la implementación de este sistema se utiliza la modelo dML – UCI, la cual consta con 4 etapas de desarrollo, conformadas por un conjunto de actividades que están compuestas por tareas que permiten

⁴ Yusleydi Fernández, Sonia Guerrero. 2008. Capacitación orientada a la calidad de software, sistema de gestión de conocimientos para pruebas de software. Habana: s.n., 2008.

que se realicen los objetivos de cada etapa. Las etapas son Planificación y documentación, Adquisición del conocimiento, Desarrollo de los medios de transmisión del conocimiento y Evaluación y monitoreo del sistema.

Además se definen roles que son los encargados de ejecutar cada una de las etapas realizando las actividades que le corresponden, los cuales se mencionan continuación.

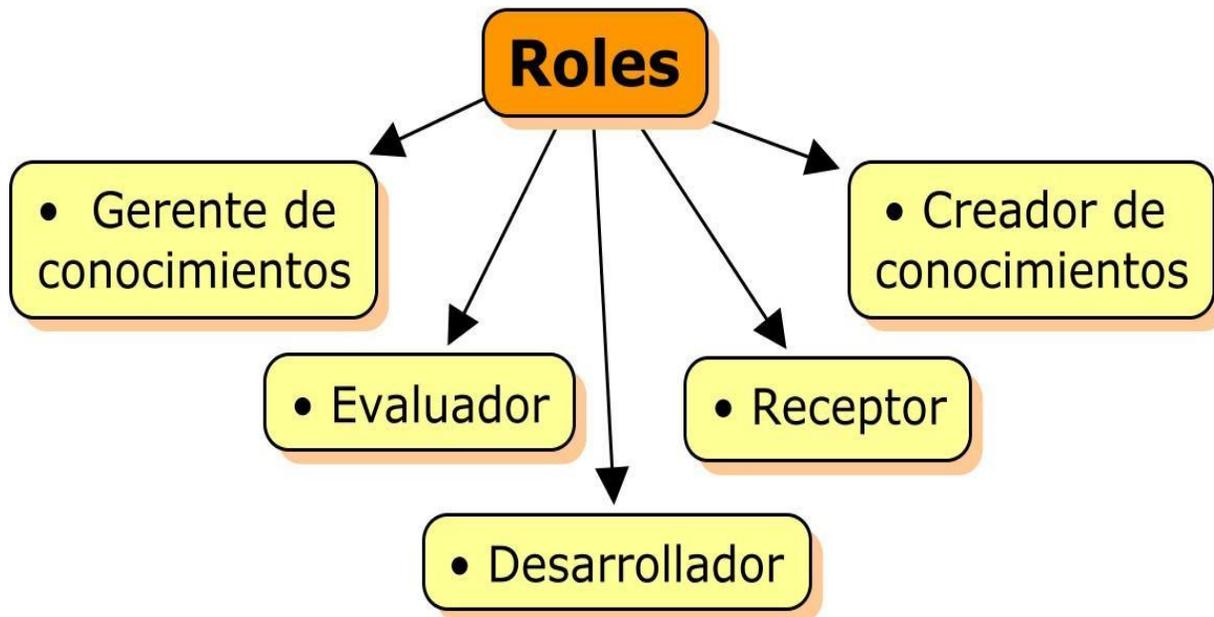


Figura 8 Roles de la metodología dML.

El proceso para la elaboración del manual del arquitecto del software estará implicado en el desarrollo de las 3 primeras etapas antes mencionadas con esto se logrará darle solución a las deficiencias existentes con respecto a las actividades que realiza este rol:

Primera etapa: Planificación y documentación.

Se realizará toda la estructuración del trabajo a desarrollar, así como la recopilación y retroalimentación de la información.

Segunda etapa: Adquisición del conocimiento

Se estudiará exhaustivamente la información obtenida para completar el conocimiento que será almacenado en el manual.

Tercera etapa: Desarrollo de los medios de transmisión de conocimiento.

Para representar la información dirigida a enseñar a los arquitectos a realizar sus actividades se determinó elaborar un manual, el cual permite conocer las principales tareas a desarrollar y los requerimientos del rol. Los manuales auxilian el adiestramiento y capacitación del personal ya que describen en forma detallada las actividades de cada puesto. Sirve para el análisis o revisión de los procedimientos de un sistema. Aumenta la eficiencia de los empleados, indicándoles lo que deben hacer y cómo deben hacerlo y construye una base para el análisis posterior del trabajo y el mejoramiento de los sistemas, procedimientos y métodos.

Como respuesta a la solución se escogió el Manual de Procedimiento: en el cual será generado, transmitido y creado todos los conocimientos.

Los roles que serán ejecutados para desarrollar el manual son los siguientes:

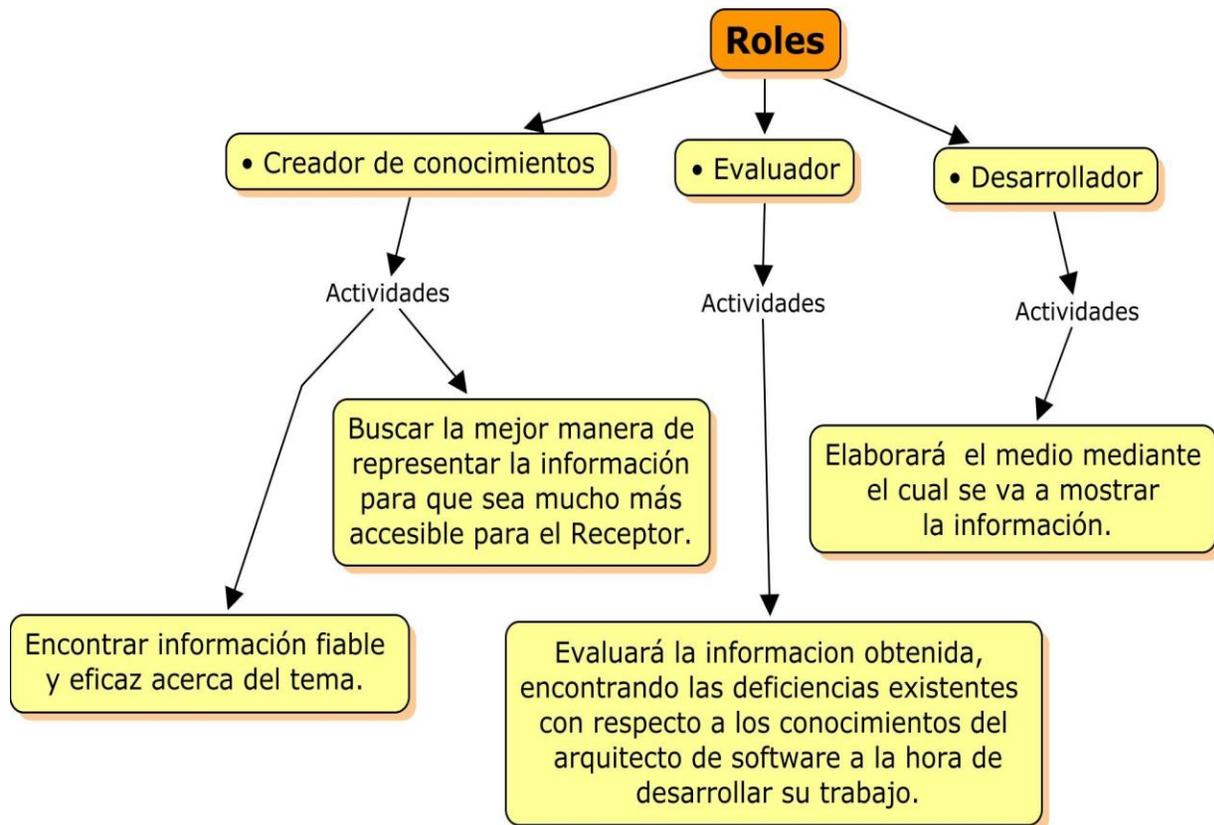


Figura 9 Roles a ejecutar para el desarrollo del manual y las actividades por cada uno de los mismos.

CAPÍTULO 2: MANUAL DEL ARQUITECTO DEL SOFTWARE.

Introducción:

Con el actual estado de innovación tecnológica y la competitividad que se pone de manifiesto y que tipifican al mundo moderno, se ha llegado a la conclusión que la buena elaboración y difusión de los objetivos, estructuras, actividades y normas de trabajo dentro del ámbito apropiado, son actos indispensables para obtener productos de software con la calidad requerida, dentro de esta calidad la AS es un pilar fundamental dependiendo totalmente del éxito que se obtenga al realizar las actividades definidas para el rol que desempeña esta tarea. Por estas razones se realizó el Manual del arquitecto del Software, en el cual se definen actividades que permiten brindarle los conocimientos precisos a personas de poca o ninguna experiencia como arquitectos de software.

2.1 Introducción al manual

El manual propuesto contiene los conocimientos necesarios de manera sencilla y amena, además presenta conceptos fundamentales del contenido a enseñar en forma de mapas conceptuales a lo largo del mismo para facilitar el aprendizaje de los usuarios. La finalidad de este manual es ofrecer una descripción actualizada, concisa y clara de las actividades del arquitecto del software contenidas en cada proceso.

Los objetivos específicos presentes en el manual de manera general son:

- Transmitir los conocimientos básicos introductorios del arquitecto de software para que se conozcan las características fundamentales que lo identifican y puedan desarrollar habilidades en este campo.
- Aprender a identificar los elementos fundamentales que integran la AS.
- Aprender a identificar los requerimientos no funcionales significativos para la AS.
- Aprender a definir la línea base de la arquitectura.
- Conocer por qué es necesario documentar la arquitectura.
- Aprender a documentar la arquitectura.
- Conocer por qué es necesario evaluar la arquitectura y los objetivos perseguidos con esto.
- Aprender a evaluar la arquitectura.

El manual consta de 5 capítulos, de ellos el primero es introductorio donde se sintetiza el estudio del estado del arte del rol, los tres siguientes se dedicarán a tres fases respectivamente donde estarán desglosadas todas las actividades generales en las cuales participa el arquitecto de software y en el capítulo 5 se resolverá de forma práctica un caso de estudio.

En la siguiente figura se muestra la estructura por capítulos del manual.

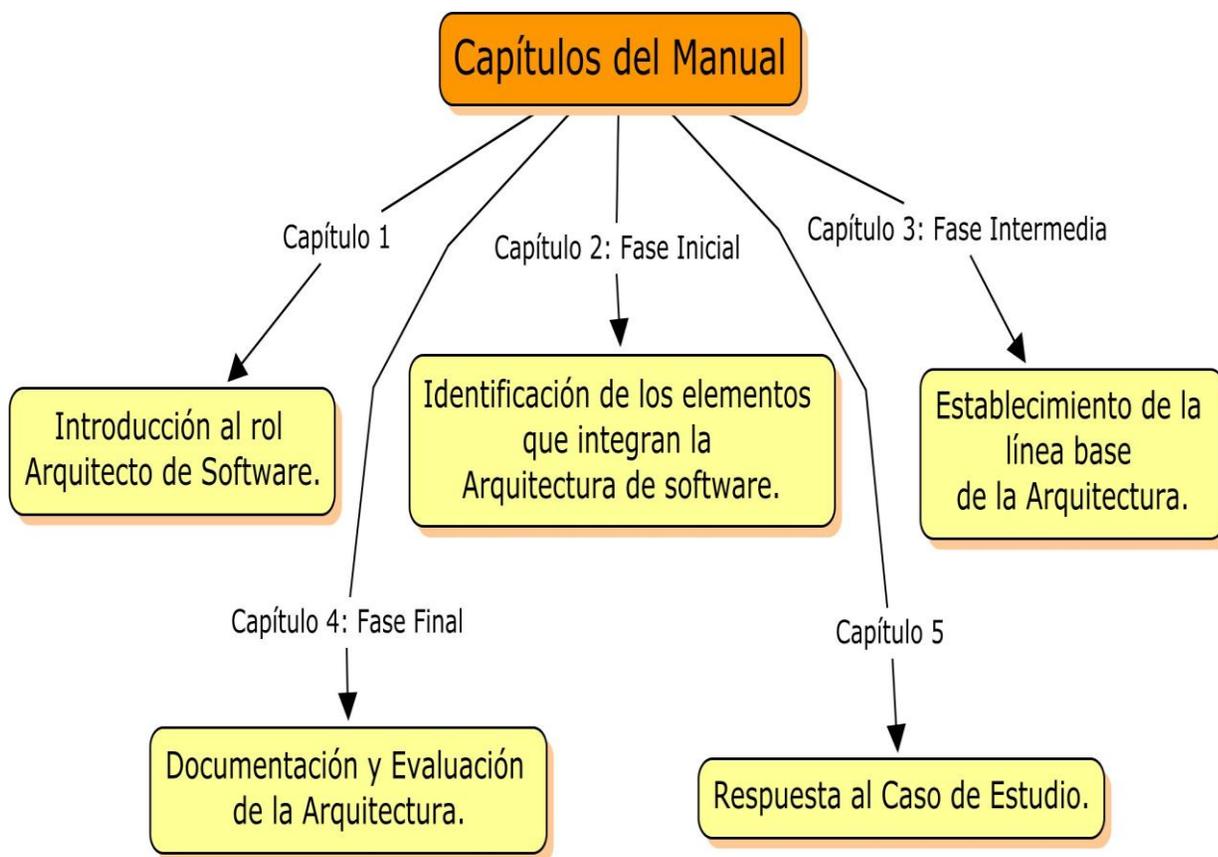


Figura 10 Estructura de los capítulos del Manual del Arquitecto del Software.

Los capítulos 2,3 y 4 poseen la siguiente estructura:

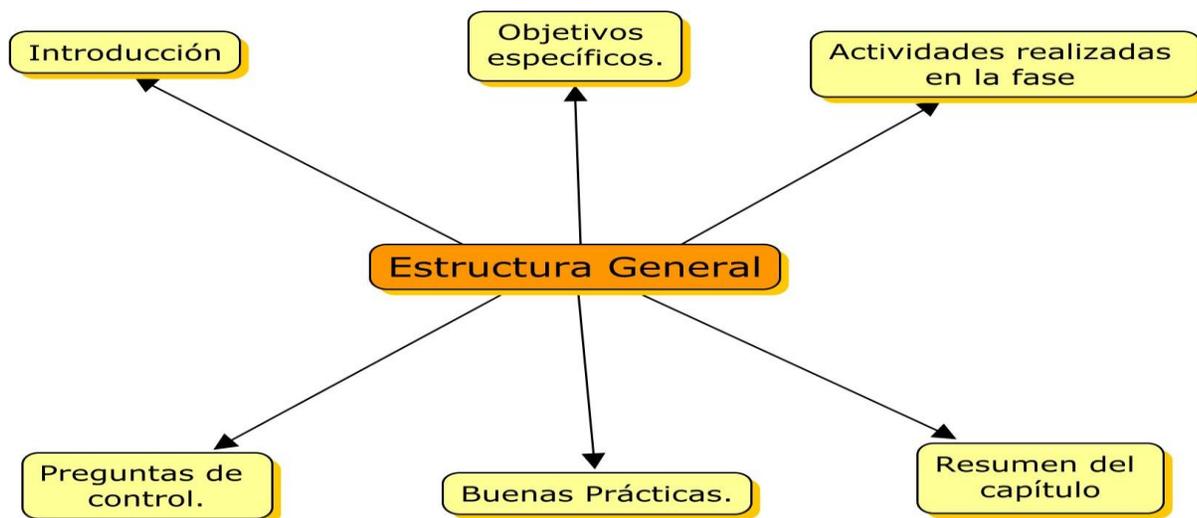


Figura 11 Estructura general de los capítulos 2, 3 y 4 del Manual.

En los capítulos a modo de resaltar los conceptos y que la lectura resulte más dinámica y menos monótona, se usa la siguiente imagen:  y serán escritos en letra cursiva y encerrados en un recuadro.

Cada uno de estos puntos y otros que aparecen en los capítulos 1 y 5 estarán representados por un ícono con el objetivo de amenizar la lectura y darle dinamismo al manual. En la siguiente tabla se muestran los mismos:

Punto del capítulo	Icono.
En las introducciones y otros puntos comunes.	
Para los objetivos.	
Para las actividades.	 ACTIVIDADES
En las preguntas.	

En buenos consejos.	
Los resúmenes.	
Caso de estudio.	
Soluciones Prácticas al caso de estudio.	
Para resaltar errores cometidos.	

Seguido de los buenos consejos en los capítulos que representan las fases definidas se incluyó una frase relacionada con la arquitectura o en similitud con el tema, el objetivo de las mismas es brindarle un momento de distracción al lector, con el fin de que la lectura no sea abrumadora.

Se definieron tres fases para el desarrollo de las actividades que serán ejecutadas por los arquitectos de software con el propósito de alcanzar organización en el contenido y de seguir una secuencia lógica en el proceso de realización de las actividades planteadas, las cuales se relacionan a continuación:

Fase Inicial: Identificación de los elementos que integran la Arquitectura de software.

- Seleccionar y evaluar los requisitos no funcionales que definen la arquitectura de software.
- Definir todos los elementos base de la arquitectura.

Fase Intermedia: Establecimiento de la línea base de la arquitectura.

- Determinar junto a los diseñadores las interfaces de integración tanto internas como externas.
- Construir la arquitectura base.
- Refinamiento de la Arquitectura.
 - Identificar los mecanismos de Diseño.
 - Identificar los elementos de Diseño.

- Incorporar elementos no construidos (flexibilidad).

Fase Final: Documentación y Evaluación de la Arquitectura.

- Evaluar la viabilidad de la arquitectura
- Elaborar el documento de la arquitectura.
- Velar por el cumplimiento de los requisitos de Hardware.
- Respuesta de la arquitectura ante cambios.

2.2 Estructura del Manual del Arquitecto de Software.

2.2.1 Capítulo 1: Introducción al rol del Arquitecto de Software.

Este capítulo 1 está constituido por un resumen realizado del estudio del arte de la arquitectura de software. Se le incorpora un complemento imprescindible de análisis que son los lineamientos de calidad mínimos que se relacionan con las actividades definidas en el manual, contribuyendo de manera positiva con la realización de las mismas, favoreciendo que cualquier producto de software que se desarrolle cuente con la calidad requerida y satisfaga las necesidades del cliente.

La siguiente Figura muestra el contenido del capítulo 1 del manual.

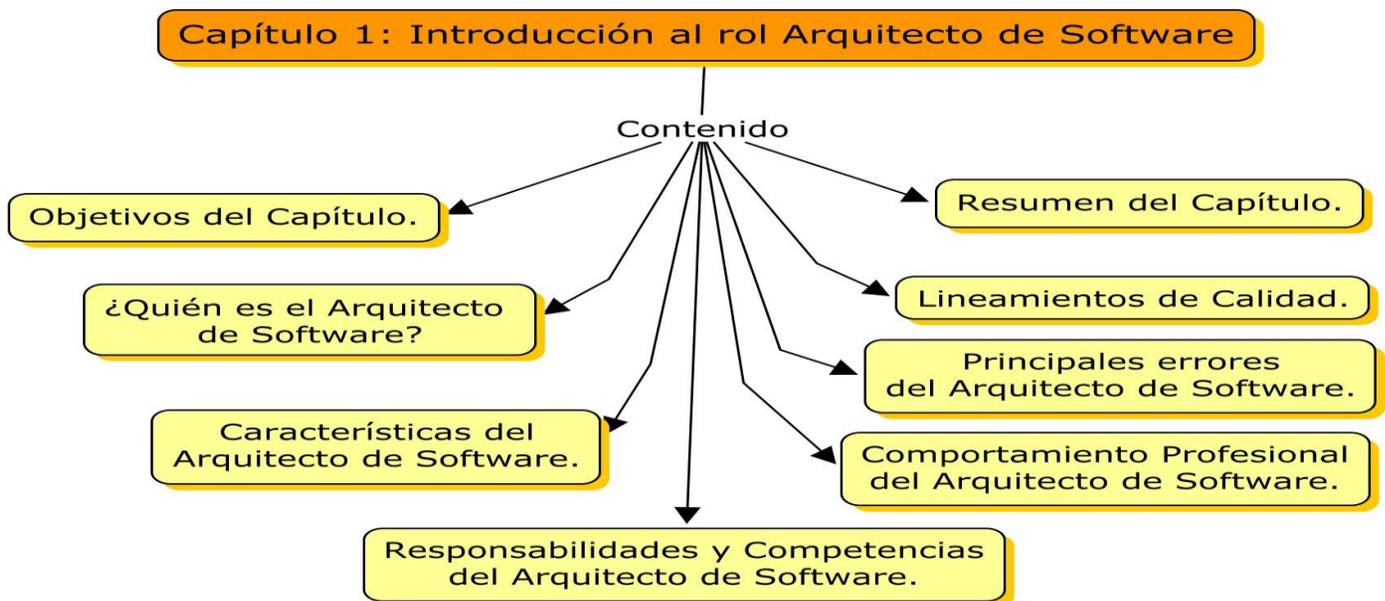


Figura 12 Estructura del Capítulo 1 del Manual del Arquitecto del Software.

2.2.2 Capítulo 2: Fase Inicial: Identificación de los elementos que integran la Arquitectura de Software.

La siguiente figura muestra el contenido del capítulo en cuestión.

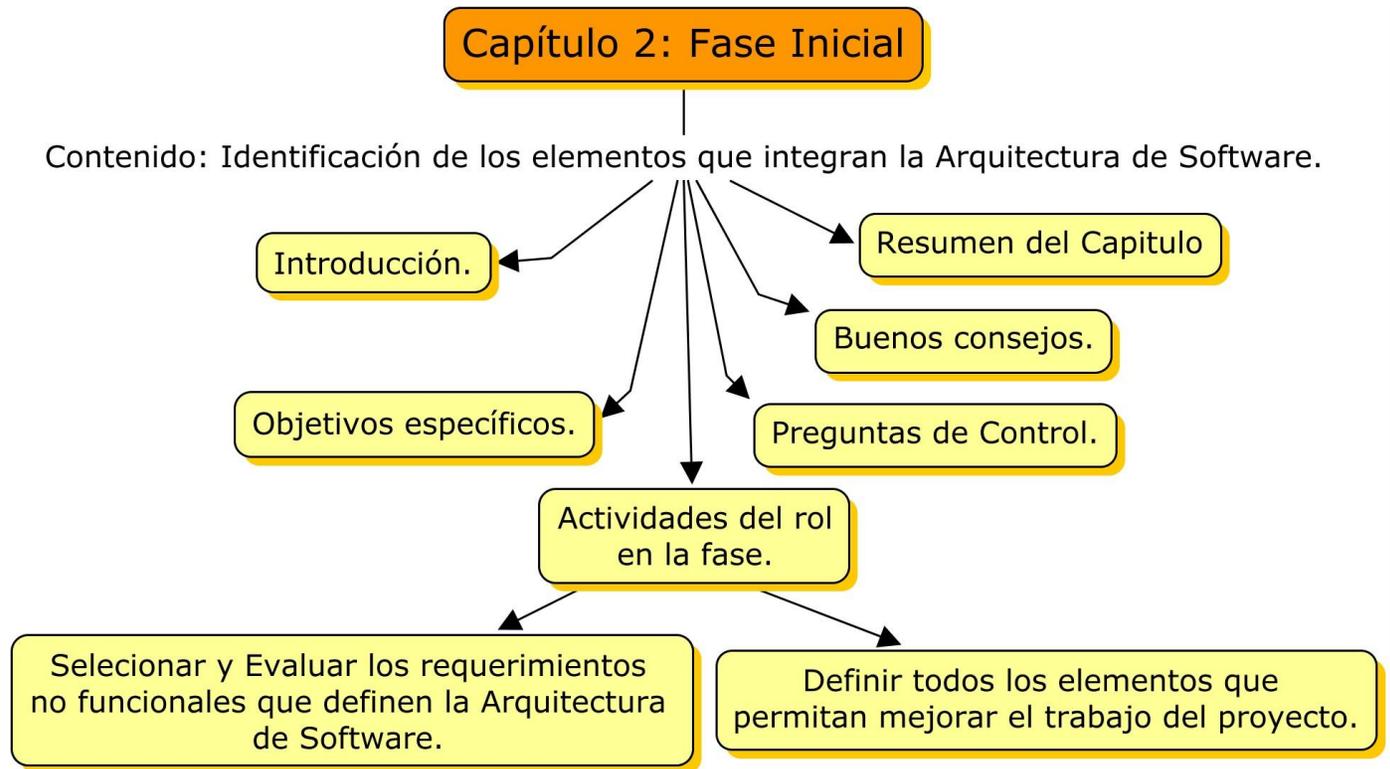


Figura 13 Estructura del Capítulo 2 del Manual del Arquitecto del Software.

Este capítulo y los dos siguientes presentan una estructura donde se incluye una introducción, seguido de los objetivos que se pretenden alcanzar y las actividades que serán ejecutadas por los arquitectos de software en cada fase.

Para consolidar y evaluar todos los conocimientos transmitidos, se formulará al final de cada capítulo preguntas de control, se brindarán buenos consejos para llegar a ser excelentes arquitectos y se concluye con un resumen a través de mapas conceptuales.

Las primeras actividades que se definieron se explicarán brevemente a continuación:

☞ Seleccionar y evaluar los requerimientos no funcionales que definen la Arquitectura de Software.

En la misma se plasma el concepto de los requerimientos no funcionales, la importancia que tienen para lograr que clientes y usuarios puedan valorar las características funcionales del producto, pues si se conoce que el mismo cumple con toda la funcionalidad requerida, como cuán usable, seguro, conveniente y agradable puede llegar a ser, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. Se explica además que estos requerimientos tienen que ser tomados totalmente en serio por los arquitectos desde el nacimiento de los proyectos de software.

Se desglosan los tipos de categorías que existen para clasificarlos seguidos de una explicación de la relación de los requerimientos con la arquitectura, específicamente con el diagrama de despliegue.

⇒ Definir todos los elementos (nodos, componentes, herramientas, framework, estilos, patrones, escenarios críticos y otros) que permiten mejorar el trabajo del proyecto.

En esta actividad se enseña cuáles son los elementos que permiten mejorar el trabajo del proyecto, a través de conceptos, características, imágenes, ejemplos prácticos, exponiendo sus ventajas, desventajas, utilidad, descripciones entre otros aspectos.

En este capítulo se profundiza en:

- ✓ Los tipos de nodos que se identifican en un sistema determinado, se aborda que en los mismos se ejecutan los componentes, que ellos representan el despliegue físico de estos componentes y lo más primordial es que se da a conocer la responsabilidad de los arquitectos en el momento de identificar los nodos que se incluyen en el sistema, ya sean Nodos Procesadores (Computadoras) o los conocidos como Dispositivos y hasta los Protocolos que comunican a estos.
- ✓ Los componentes, se tratan los principios que se tienen que tener en cuenta al diseñarlos, que no son más que su reusabilidad, extensibilidad, encapsulamiento, independencia, libres de un contexto específico.
- ✓ Las principales herramientas que pueden ser utilizadas en el ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

- ✓ Los estilos y patrones arquitectónicos cimienta esencial en la construcción de los sistemas, puesto que un buen arquitecto debe conocer la mayor cantidad de patrones y estilos posibles, y no solo conocerlos, sino dominarlos pues definen las reglas generales de organización en términos de un patrón y las restricciones en la forma y la estructura de un grupo numeroso y variado de sistemas de software. Un estilo determina el vocabulario de componentes y conectores que pueden ser utilizados en instancias de este estilo, con un conjunto de restricciones en las descripciones arquitectónicas.

- ✓ Los escenarios significativos de un sistema (casos de usos como se les llaman en RUP o historias de usuario en XP) de acuerdo al impacto que ejerzan sobre la arquitectura deben ser priorizados por el arquitecto pues de acuerdo a su prioridad es la significación e influencia que estos tendrán sobre la arquitectura, se clasifican en críticos, secundarios, auxiliares, opcionales.

2.2.3 Capítulo 3: Fase Intermedia: Establecimiento de la Línea Base de la Arquitectura.

La siguiente figura muestra el contenido del capítulo.

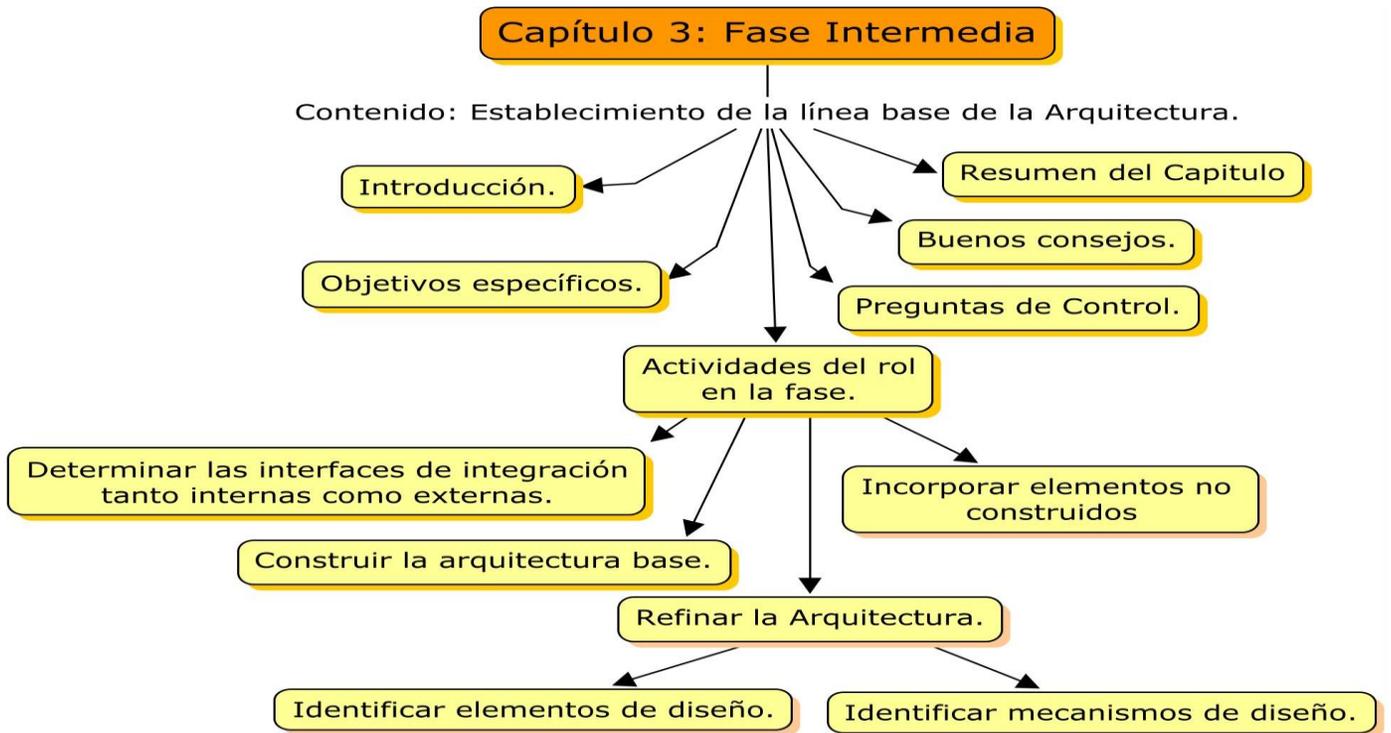


Figura 14 Estructura del Capítulo 3 del Manual del Arquitecto del Software.

Las actividades que se puntualizan en este capítulo permiten definir la Línea Base de la arquitectura, la cual le da forma al sistema y es el esqueleto del mismo. Esta Línea Base contiene las funcionalidades más significativas del sistema que nos ayudan a mitigar los riesgos más importantes y deben ser las más trascendentales para los usuarios del sistema. La línea base de la arquitectura es un conjunto de artefactos revisados y aprobados que representa un punto de acuerdo para la posterior evolución y desarrollo, y solamente puede ser modificada a través de un procedimiento formal, como la gestión y configuración de cambios.

A continuación se presentan cada una de estas actividades:

☞ Determinar las interfaces de integración tanto internas como externas.

En esta actividad se profundiza en:

- ✓ Las interfaces externas donde se explica que cada usuario a la hora de acceder a un sistema lo hace a través de la interfaz que este tiene definida para este usuario específico, que no son más que las interfaces de usuarios.
- ✓ Las interfaces internas donde se da a conocer que cada sistema posee internamente interfaces para que los componentes y subsistemas se comuniquen entre sí.



Construir la arquitectura base.

En esta actividad se tiene en cuenta que para lograr la arquitectura base del software se deben definir la vistas arquitectónicas y hay que integrar los elementos vistos en los capítulos anteriores del manual a través de los diagramas de implementación, estos diagramas son los de despliegue y de componentes.

En esta actividad se profundiza en:

- ✓ El modelo de las “4+1” vistas de Philippe Kruchten, donde se explican las vistas que se definen en el mismo a continuación se relacionarán :
 1. **Vista de casos de uso:** representa un extracto del modelo de casos de usos del sistema. La vista de casos de uso es igual al modelo de casos de uso, la única diferencia es que la vista de la arquitectura solo tiene aquellos casos de usos y actores que son arquitectónicamente significativos mientras que el modelo de casos de uso tienen todos los casos de usos del sistema. Esta vista puede ser modelada estáticamente mediante diagramas de casos de usos.
 2. **Vista lógica o de diseño:** representa los elementos de diseño más importantes para la arquitectura del sistema, describe las clases más importantes, su organización en paquetes y subsistemas, y estos a su vez en capas. También describe las realizaciones de casos de uso más importantes como por ejemplo los que describen aspectos dinámicos del sistema. Esta vista puede ser modelada estáticamente mediante diagramas de clases y objetos.
 3. **Vista de procesos:** representa una base para la comprensión de la organización de los procesos de un sistema, ilustrados en el mapeo de las clases y subsistemas en procesos e hilos. Solo suele usarse cuando el sistema presenta procesos concurrentes

o hilos. Puede ser modelada estáticamente mediante diagramas de interacción (secuencia y colaboración).

4. **Vista de componentes o de implementación:** representa la captura de las decisiones arquitectónicas tomadas para la implementación. Normalmente, esta vista contiene una enumeración de todos los subsistemas del modelo de implementación, diagramas de componentes que ilustran cómo se organizan los subsistemas en capas y jerarquías, ilustraciones de dependencias de importación entre subsistemas, puede ser modelado estáticamente a través del diagrama de componente.
- ✓ El diagrama de despliegue del cual el arquitecto es el responsable donde debe considerar los aspectos referentes a la futura distribución física o despliegue del sistema, el hardware es un punto clave para el correcto funcionamiento del sistema, ya que este no es capaz de ejecutarse en el “aire”, necesita de un hardware para correr. En ocasiones las condiciones del hardware tienen que ser específicas, la configuración de red existente tiene que cumplir ciertos parámetros y debe haber protocolos específicos disponibles, además se explica la necesidad de que esta configuración se realice correctamente porque de no ser sucederían problemas ilimitables. Se detallan los elementos que integran este diagrama, los protocolos que se usan y los pasos fundamentales para su correcta elaboración.

Refinamiento de la Arquitectura.

Cuando se realiza el refinamiento de la arquitectura se centra la atención en Identificar los Mecanismos de Diseño reutilizables que pueden ser necesarios a emplear en la elaboración del sistema, tales como mecanismos para garantizar la persistencia y la seguridad de los datos, así como la concurrencia de las operaciones y accesos a la información, conjuntamente se identifican elementos de diseño, significativos para la arquitectura.

En esta actividad se profundiza en:

- ✓ Identificar los elementos de diseño, se enseña cómo se identifican los subsistemas, las interfaces, los sucesos y las señales, lo que se obtiene al analizar los principales escenarios de un sistema a implementar, que no son más que aquellas que representan la interfaz de usuario entre un escenario y el autor (clases fronteras), o sea representan objetos gráficos como ventanas, diálogos o algún menú. Otro tipo de clase son las llamadas clases entidades las

cuales modelan objetos del dominio, entidades o acontecimientos del mundo real de los que el software debe utilizar información, estos objetos deben ser persistentes. Por último están las clases controladoras las cuales corresponden a los objetos internos del software y no persistentes. Estos tres tipos de clases representan elementos conceptuales que pueden ejecutar un comportamiento en específico, van evolucionando hasta convertirse en varios tipos de elementos de diseño como clases, subsistemas, clases activas, interfaces, sucesos y señales.

- ✓ Identificar los mecanismos de diseño donde en este proceso se refinan los mecanismos de análisis para adaptarlos a las restricciones del entorno de implementación. Se explica que los mecanismos de diseño están relacionados con los mecanismos de análisis, los cuales son perfeccionamientos adicionales, y pueden vincular uno o más patrones de diseño y de arquitectura. No existe necesariamente ninguna diferencia en escala entre el mecanismo de análisis y el mecanismo de diseño; por lo tanto es posible hablar de un mecanismo de permanencia a nivel de análisis y a nivel de diseño y querer decir lo mismo, pero a un nivel diferente de perfeccionamiento. Se explican y profundizan cada unos de los pasos a seguir para identificar correctamente estos mecanismos.

 Incorporar elementos no construidos (flexibilidad).

La flexibilidad es la capacidad que tiene un objeto o cosa de adaptarse a una nueva situación. No podemos olvidarnos que la necesidad más acuciante de gestión en la industria mediana es su necesidad de flexibilidad y de adaptación al mercado, es decir, atender de una forma ágil y fiable los requerimientos de los clientes, manteniendo productos y costes adecuados. Una buena selección de tecnología con base aporta en el tema de flexibilidad y agilidad un factor preponderante para el éxito o el fracaso de una empresa ante los cambios del mercado.

Un sistema debe ser capaz de aceptar nuevos cambios y nuevos componentes que pueden que ya estén desarrollados y solos tengamos que reutilizarlos, además tiene que aceptar actualizaciones de componentes que ya posee.

En esta actividad se profundiza en:

- ✓ En la explicación de cuatro pasos fundamentales para perfeccionar el modelo de diseño:

1. Identificar oportunidades de reutilización.
2. Revertir la ingeniería de los componentes.
3. Actualizar la organización del modelo de diseño.
4. Actualizar la vista lógica.

2.2.4 Capítulo 4: Fase Final: Documentación y Evaluación de la Arquitectura.

La siguiente figura muestra el contenido del capítulo:

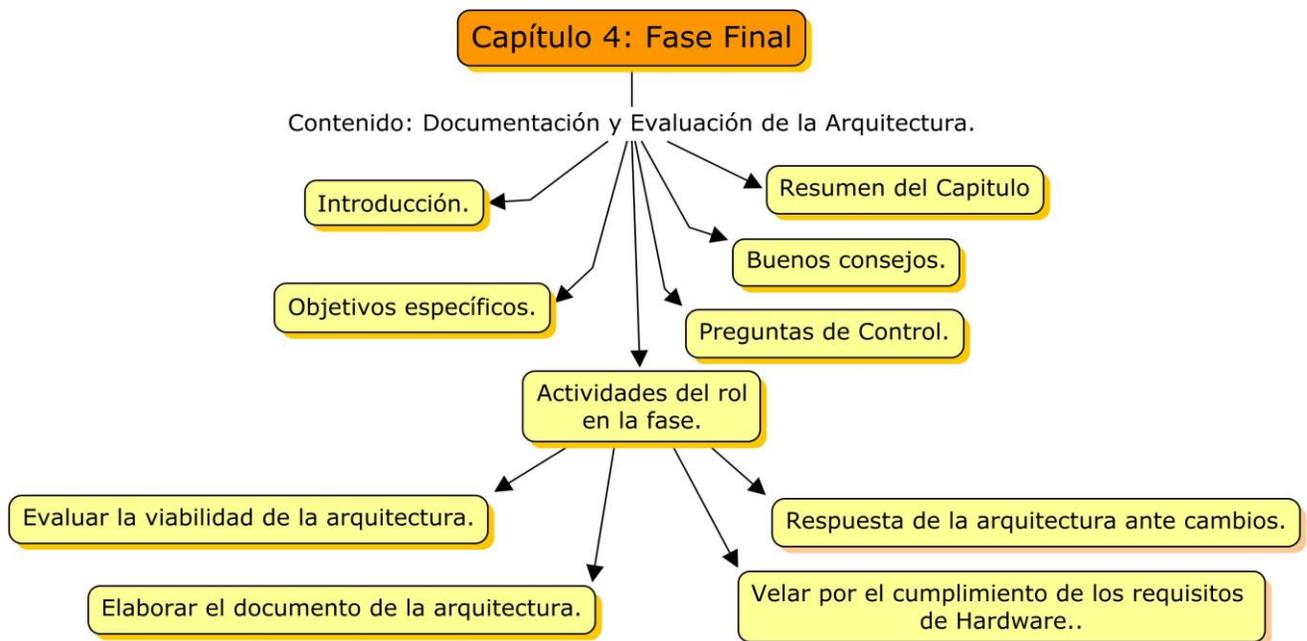


Figura 15 Estructura del Capítulo 4 del Manual del Arquitecto del Software.

La documentación de la arquitectura es de gran importancia, incluso una arquitectura perfecta es inútil si nadie la entiende. Si se desea crear una sólida arquitectura, debe describirse con suficientes detalles, sin ambigüedades, y organizado de tal manera que otros puedan encontrar rápidamente la información necesaria. En caso contrario, el esfuerzo realizado se ha perdido debido a que la arquitectura será inutilizable.

A continuación se presentan las actividades definidas en este capítulo.

 Evaluar la viabilidad de la arquitectura.

En esta actividad se profundiza en:

- ✓ Detectar posibles riesgos y verificar el cumplimiento de los atributos de calidad. Con la evaluación de la arquitectura se busca saber si se pueden habilitar los requerimientos, atributos de calidad y restricciones para asegurar que el sistema a ser construido cumple con las necesidades de los stakeholders, se especifican quienes participan en la evaluación de la arquitectura, que deben ser generalmente los miembros del equipo de desarrollo, los arquitectos y los diseñadores, pero también pueden participar personas especialistas en el tema y los clientes. En la UCI se realizan mediante auditorias organizadas por la dirección de la institución, ó empresa, o por la dirección de calidad central, se explica cuál es el papel del arquitecto en esta actividad ya que él es responsable de generar y documentar la arquitectura de software para el sistema estudiado, además de explicar a los stakeholders cuál es la arquitectura evaluada, pero también se encarga de generar y documentar una arquitectura de software para el sistema estudiado, debe explicar a los stakeholders cuál es la arquitectura evaluada. En este punto del manual se enseña a través de que cualidades es evaluada dicha arquitectura (puede hacerse por atributos de calidad observables como por ejemplo la disponibilidad, la funcionalidad, la confiabilidad, entre otros y a través de los atributos de calidad no observable como la integridad, la reusabilidad, la portabilidad, la escalabilidad, la modificabilidad, entre otros más existentes) y cuáles métodos pueden ser usados para estos fines, recomendando los métodos ATAM y MECABIC usados en la universidad por los especialistas de la dirección técnica, estos dos métodos contemplan una gran cantidad de atributos de calidad para valorar arquitecturas. También se recomiendan momentos en los cuales es conveniente realizar la evaluación, donde se explica que puede ser evaluada en cualquier momento pero es recomendable en dos etapas distintas: temprano y tarde. Otro aspecto fundamental analizado es una recomendación de pasos a seguir para la evaluación:
 1. Determinar los criterios de evaluación: estos criterios se obtienen a partir de los requisitos significativos arquitectónicamente que fueron incluidos en la arquitectura.

2. Evaluar la arquitectura: En este paso, la arquitectura se prueba comparándola con los criterios de evaluación. Por ejemplo, en el caso de un prototipo ejecutable, se puede realizar mediante una demostración; en el caso de un modelo conceptual, mediante la inspección y el razonamiento; o para una simulación, mediante la configuración y la ejecución del modelo de simulación con datos de entrada derivados de los criterios de evaluación, y la recopilación y el análisis posterior de los datos de salida del modelo.
3. Valorar los resultados: Los resultados de la evaluación se valoran para determinar no sólo si se pueden cumplir los requisitos significativos arquitectónicamente, sino también como comprobación de la validez de dichos requisitos. En este momento del desarrollo, los requisitos todavía pueden mutar y no siempre son bien comprendidos por los interesados.

Elaborar el documento de Arquitectura.

En esta actividad se profundiza en:

- ✓ La importancia que posee la elaboración del documento de arquitectura y que aspectos se recogen en el mismo. En el documento de arquitectura se registran las vistas, los diagramas tanto de despliegue como de componentes, el de clases, además también recogen las herramientas que se usan, los casos de usos que se priorizan, en fin, aquí queda plasmado todo el trabajo.

Respuesta de la arquitectura ante cambio.

En esta actividad se profundiza en:

- ✓ La actualidad con respecto a los cambios. Se aconseja crear una arquitectura de tal manera que se adelante a los posibles cambios y que de haber necesidad de realizar algún cambio esta responda positivamente y permita que se le realicen los cambios necesarios, es decir, la Arquitectura debe ser flexible a estos cambios.

Velar por el cumplimiento de los requisitos de Hardware.

En esta actividad se pide que el arquitecto vele por los requerimientos de hardware definidos antes.

3. Modo de uso para el Manual del Arquitecto de Software.

Las autoras proponen para una mejor comprensión de los conocimientos implícitos en dicho manual se realice una lectura dirigida al conocimiento que se quiera aprender, o sea, que si necesita saber de un tema en específico se dirija ala capítulo que se encarga de desarrollarlo. En cambio si el lector es principiante en el tema y no conoce mucho del mismo, es recomendable leer el manual desde el principio.

Capítulo 3 Validación del Manual del Arquitecto de Software.

3.1 Introducción.

Cada persona controla o examina la calidad de las acciones que realiza, ya sea dentro de su vida personal o en el contorno profesional, y es en este ámbito donde toma un connotado valor esta acción, pues en dependencia de los resultados obtenidos como fruto de su trabajo profesional obtiene el respeto, la admiración y alcanza prestigio ante los demás. Para conocer la validez de la propuesta se utilizaron los métodos siguientes: método de validación Delphy y método de validación por Casos de Estudio.

3.2 Delphy como método de validación de experto.

Es una técnica de investigación social que tiene como objetivo obtener de forma confiable un consenso de opiniones y valoraciones sobre un tema en específico. El método Delphy consiste en un proceso iterativo de rondas donde se le realizan encuestas a cada experto (en el caso de la presente investigación se realiza con especialistas debido a que en la universidad no se cuenta con expertos) de forma individual evitando la interacción. La encuesta se lleva a cabo de una manera anónima (actualmente es habitual realizarla haciendo uso del correo electrónico o mediante cuestionarios web establecidos al efecto).

La calidad de los resultados depende, sobre todo, del cuidado que se ponga en la elaboración del cuestionario y en la elección de los especialistas consultados. Siempre se comenzará este proceso enviando un modelo a los posibles especialistas con una explicación breve sobre los indicadores del trabajo y los resultados que se desean obtener. Para la aplicación práctica del método es necesario considerar dos cuestiones fundamentales:

- La selección de los especialistas.
- La elaboración del cuestionario.

A continuación se explica cómo fueron aplicadas estas cuestiones.

3.2.1 El Proceso de Selección de Especialistas.

Como bien se explica anteriormente, es muy importante después de haber decidido aplicar el método Delphy, seleccionar a los especialistas en el tema a evaluar. Estos, serán una persona o grupo de

personas con un gran conocimiento sobre el tema tratado, que podrán emitir criterios irrefutables de cualquier problema y valoraciones importantes con un alto nivel de conocimiento.

Para la selección de los especialistas que determinarán hasta qué punto es eficiente el manual, se tuvo en cuenta una serie de características, que posibilitaron obtener información curricular y actualizada de ellos. Basándose principalmente, en las participaciones de estos en eventos científicos, que contaran además con una ardua experiencia ocupacional en el tema. También se tuvo en cuenta los conocimientos que deben tener sobre el problema planteado. Estos conocimientos están basados principalmente en las siguientes áreas de trabajo:

- Proceso de Desarrollo del Software.
- Arquitectura de Software.

Basándose en las informaciones anteriores, la lista a consultar quedó conformada por 9 especialistas. Estos están integrados de manera directa a la producción en la UCI y de una forma u otra vinculados al trabajo con la AS. A este listado de especialistas, se le realizó una autoevaluación de los niveles de información y argumentación que tienen sobre el tema especificado. En la cual marcaron con una X, en una escala creciente del 1 al 10, el valor que se corresponde con el grado de conocimiento o información que tienen cada uno sobre el tema de prueba; donde los resultados obtenidos se muestran a continuación en la siguiente tabla:

	Grado de conocimiento o información sobre el tema									
Especialistas	1	2	3	4	5	6	7	8	9	10
1			x							
2						x				
3					x					
4						x				
5							x			
6								x		
7							x			

8					x					
9								x		

Tabla 1: Grado de conocimiento de los especialistas con respecto al tema.

Una vez analizado el grado de conocimiento con el que cuentan estos especialistas, se pasó a calcular el Coeficiente de Conocimiento o Información (Kc), a través de la siguiente fórmula.

$$Kc = n(0,1)$$

Donde:

Kc: Coeficiente de Conocimiento o Información.

n: Rango seleccionado por el especialista.

En la tabla siguiente se muestran los resultados obtenidos para cada especialista, en la que se puede apreciar que casi todos poseen un coeficiente de conocimiento medio:

Especialista	Valor de Kc
1	0,3
2	0,6
3	0,5
4	0,6
5	0,7
6	0,8
7	0,7
8	0,5
9	0,8

Tabla 2: Coeficiente de conocimiento

Seguidamente se valoran un grupo de aspectos que influyen sobre el nivel de argumentación o fundamentación del tema a estudiar, donde se determinan los aspectos de mayor influencia y que revelan conocimientos de AS. A partir de los valores reflejados por cada especialista en una tabla similar a esta, en la cual ellos podían marcar con una X el nivel que ellos tienen en estos aspectos. Posteriormente se contrastan con los valores de la siguiente tabla patrón:

FUENTES DE ARGUMENTACION	Alto	Medio	Bajo
Análisis teóricos realizados por usted.	0.3	0.2	0.1
Su experiencia obtenida.	0.5	0.4	0.2
Trabajos de autores nacionales.	0.05	0.05	0.05
Trabajos de autores extranjeros.	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

Tabla 3: Tabla patrón del nivel de argumentación.

Una vez vista la tabla patrón, se muestra un resumen con las fuentes de argumentación seleccionadas por cada especialista.

FUENTES DE ARGUMENTACION	Alto	Medio	Bajo
Análisis teóricos realizados por usted.	9	2,3,4,5,6,7,8	1
Su experiencia obtenida.	6	1,2,4,5,7,9	3,8
Trabajos de autores nacionales.	2,3		1,4,5,6,7,8,9
Trabajos de autores extranjeros.	6,7,8,9	1,3,5	2,4
Su propio conocimiento del estado del	6,8,9	3,4,5,7	1,2

problema en el extranjero.			
Su intuición.	2	3,4,5,7,8,9	1,6

Tabla 4: resumen del nivel de argumentación de los encuestados.

Estos aspectos que influyen sobre el nivel de argumentación o fundamentación del tema a estudiar, permitieron calcular el Coeficiente de Argumentación (Ka) de cada especialista, mediante la siguiente ecuación:

$$Ka = a \cdot n_i = (n_1 + n_2 + n_3 + n_4 + n_5 + n_6)$$

Donde:

Ka: Coeficiente de Argumentación.

n_i : Valor correspondiente a la fuente de argumentación i (1 hasta 6).

Los resultados obtenidos fueron que 2 de los especialistas poseen un nivel de coeficiente de argumentación igual a 0.9, y el resto poseen un nivel mayor o igual a 0.6.

Experto	Valor de Ka
1	0,7
2	0,8
3	0,6
4	0,8
5	0,8
6	0,9
7	0,8
8	0,6
9	0,9

Tabla 5: Coeficiente de argumentación.

Una vez obtenido los valores del Coeficiente de Conocimiento (Kc) y el Coeficiente de Argumentación (Ka) se procede a obtener el valor del Coeficiente de Competencia (K) que finalmente es el coeficiente que determina en realidad que experto se toma en consideración para trabajar en esta investigación Este coeficiente (K) se calcula de la siguiente forma:

$$K = 0,5 (Kc + Ka)$$

Donde:

K: Coeficiente de Competencia

Kc: Coeficiente de Conocimiento

Ka: Coeficiente de Argumentación

Posteriormente obtenido los resultados se valoran de la manera siguiente:

0,8 < K < 1,0 Coeficiente de Competencia Alto

0,5 < K < 0,8 Coeficiente de Competencia Medio

K < 0,5 Coeficiente de Competencia Bajo.

De manera general los cálculos realizados dadas las respuestas de los especialistas quedan como se muestran en la próxima tabla.

Especialista	Kc.	Ka.	K.	Grado del coeficiente de competencia.
1	0,3	0,7	0,5	Medio
2	0,6	0,8	0,7	Medio
3	0,5	0,6	0,55	Medio
4	0,6	0,8	0,7	Medio
5	0,7	0,8	0,75	Medio
6	0,8	0,9	0,85	Alto
7	0,7	0,8	0,75	Medio

8	0,5	0,6	0,55	Medio
9	0,8	0,9	0,85	Alto

Tabla 6: Resultados obtenidos.

Una vez confeccionado el listado de especialistas, se invitó a cada uno de ellos de forma personal para que participaran en el proceso de Validación y aceptación de la propuesta. Se detalló de forma clara y precisa los indicadores por los que se evaluaba la propuesta, así como la realización del cuestionario. Con este paso se termina el proceso de selección del personal que conformará el Panel de Especialistas. Una vez dado el consentimiento de cada uno de ellos se conforma el panel que no es más que el conjunto de especialistas.

3.2.2 Elaboración del Cuestionario.

Esta fase del método se centra en la elaboración del cuestionario que se envía a los especialistas, es el documento con el que se consigue que los especialistas interactúen, ya que en él se obtendrá la información para presentar los resultados. Para la presentación del cuestionario se tuvieron en cuenta preguntas con enfoques investigativos y que se centraran principalmente en los indicadores que debe cumplir la propuesta presentada, además de permitir que las respuestas fueran abiertas y en todos los casos con posibilidad de emitir su criterio personal (Anexo 3: Cuestionario a Miembros del Panel de Especialistas.). En este proceso los especialistas que conforman el panel recibieron de forma personal un resumen del Manual del Arquitecto de Software, unido al cuestionario que debían responder, en un mínimo de tiempo para analizar las respuestas y pedirles que realizaran preguntas en caso de que surgieran dudas durante el estudio de la misma.

3.2.3 Validación de la propuesta.

Para validar el Manual del Arquitecto de Software propuesto, se tuvieron en cuenta una serie de indicadores, que son reflejados en el cuestionario a través de las preguntas aplicadas en el mismo, la correspondencia de los indicadores con cada pregunta se muestra en la siguiente tabla:

Indicadores — Preguntas	1	2	3	4	5
1-Importancia de la Propuesta.	X				
2-Necesidad de Empleo.		X			
3-Posibilidad de uso.		X	X		
4-Adaptabilidad a los proyectos.					
5-Contribución a la Calidad de los productos de la universidad.			X		
6-Factores que pueden atentar Contra la utilización y modo de uso del manual.				X	X

Tabla 7: Indicadores de las preguntas.

El cuestionario consta de cinco preguntas, la primera responde al indicador número uno (Importancia de la Propuesta), la segunda tributa a la necesidad de los arquitectos de software, a la posibilidad de uso del Manual y a la adaptabilidad del mismo a los proyectos productivos. La tercera responde a la posibilidad de uso de manera que contribuya a la calidad de los productos. La cuarta y quinta pregunta son opiniones y criterios que pueden tener los especialistas sobre los factores que atentarían y los que facilitarían al uso del manual, así como recomendaciones al manual evaluado, en este indicador favorecería que al tener en cuenta los criterios de un mayor número de especialistas haría al producto más completo, pues estos conocedores del tema aportarían nuevas ideas. Cubriéndose así en cada una de las preguntas del cuestionario los indicadores trazados. Para un mejor análisis de la encuesta se analizan cada uno de los indicadores propuestos.

3.2.3.1 Importancia de la Propuesta.

Para valorar este aspecto se tuvo en cuenta las respuestas de los especialistas a la pregunta 1, referidos a la importancia de desarrollar un Manual para los Arquitectos de Software. Las respuestas podían ser:

- Positivas.
- Negativas.
- No se emite criterio.

El resultado se muestra a continuación:

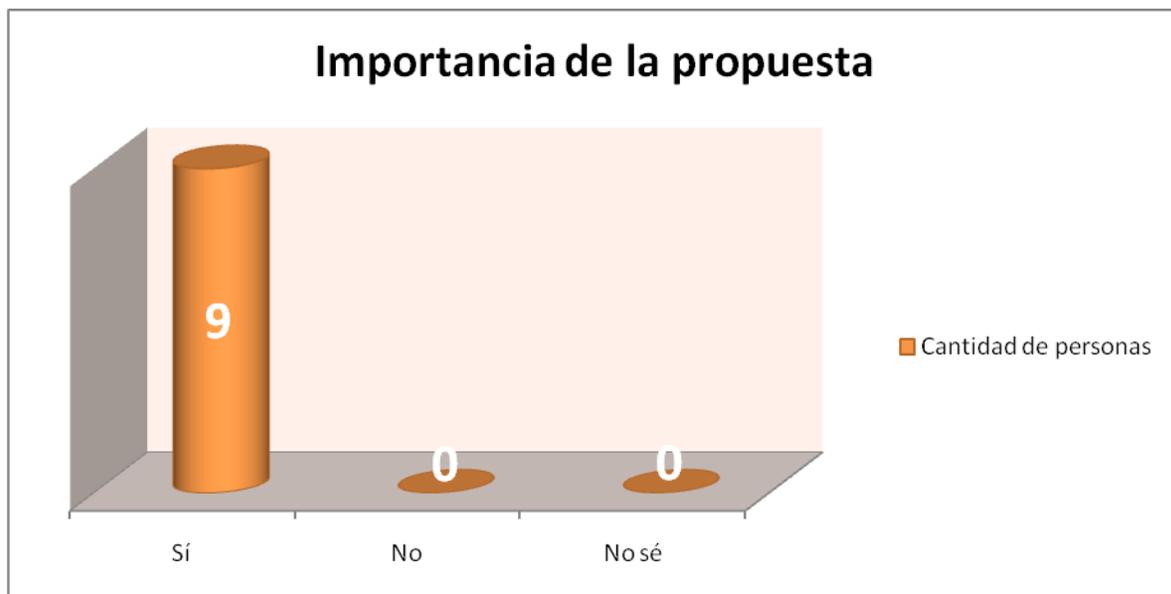


Figura 16 Importancia de la propuesta.

Como se observa en la gráfica todos los especialistas coincidieron en su totalidad que el Manual propuesto tiene gran importancia como material de consulta para el arquitecto, para facilitar el aprendizaje y trabajo del mismo, con un 100% de aceptación. Argumentando sus respuestas a través de los criterios que se emiten a continuación:

- Para conocer las actividades y responsabilidades del rol.
- Es importante porque mejoraría el proceso de aprendizaje de aquellos que tengan poca o ninguna experiencia en el tema.
- En la universidad hay pocos documentos que formalicen la arquitectura y por tanto poca guía para quienes comienzan el rol.
- Una documentación detallada y específica sobre esos temas es siempre importante y de gran ayuda, más cuando ese tipo de documentación es escasa.
- Es importante por lo difícil de la toma de decisiones asociadas a definiciones tecnológicas y de infraestructura productiva.
- Para que se sigan buenas prácticas y no se quede nada por evaluar.
- Ayudaría a crear una base porque en la universidad se cuenta con poca bibliografía del tema.
- Muestra ejemplos que ayudarán a los que se inician en el rol.
- Ayudaría al personal que desempeñan el rol en los proyectos productivos que carecen de experiencia.

3.2.3.2 Necesidad de Empleo.

En la pregunta 2 del cuestionario, los especialistas dieron una valoración en un rango del 1 al 5 siendo el 5 el mayor valor del grado de necesidad que existe en la Universidad de emplearse esta propuesta. El resultado se muestra a continuación:

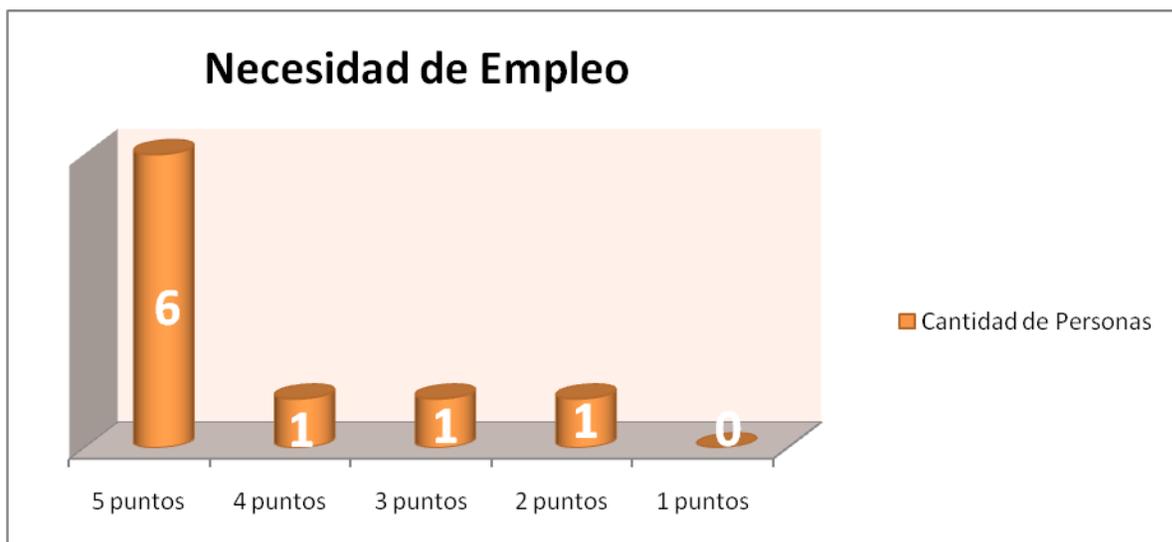


Figura 17 Necesidad de empleo del manual

La gran mayoría de los especialistas consideran que existe necesidad de empleo del manual, seis de ellos dan una puntuación de cinco puntos, lo que representa un gran porcentaje de necesidad.

3.2.3.3 Posibilidad de uso.

Este objetivo se evalúa de la misma manera que el anterior, los especialistas dieron una valoración en un rango del 1 al 5 siendo el 5 el mayor valor de la posibilidad de que el manual se use. El resultado se muestra a continuación:



Figura 18 Posibilidad de uso del manual.

Como muestra el gráfico prevalece la posibilidad de uso del manual, al ser la mayor puntuación la más proporcionada por los especialistas.

3.2.3.4 Adaptabilidad a los proyectos.

En este objetivo los especialistas dieron una valoración en un rango del 1 al 5 siendo el 5 el mayor valor. El resultado se muestra a continuación:



Figura 19 Adaptabilidad del manual.

La gráfica muestra contradicción en las respuestas, pero nuevamente prevalece la puntuación mayor.

3.2.3.5 Contribución a la Calidad de los productos de la universidad.

Para valorar este aspecto se tuvo en cuenta las respuestas de los especialistas a la pregunta 3, donde se analizó si el uso del manual en la universidad permitiría mejorar la calidad de los productos que se desarrollan en la misma. Las respuestas podían ser positivas, negativas o no se emite criterio. El resultado se muestra a continuación:



Figura 20 Contribución del manual a la calidad de los productos en la UCI

Como muestra el anterior gráfico, seis de ellos respondieron positivamente ante la posibilidad de que el manual mejore la calidad de los productos, dos dijeron no saber y solo uno se manifestó negativamente. Los especialistas argumentaron su respuesta a través de los criterios que se emiten a continuación:

- El manual puede servir de guía, pero no puede formar la experiencia necesaria para cumplir este rol.
- Porque mejoraría el proceso de aprendizaje de aquellos que tengan poca ó ninguna experiencia en el tema y al aprender, realizarían su trabajo con mayor calidad.
- Es un área de conocimiento que no se da en el proceso de formación durante la carrera.
- Porque los temas que aborda son bastante genéricos e importantes en los proyectos actuales en la universidad.
- Debe llevar un análisis y escenarios de aplicación más amplios.
- En la UCI existen facultades y centros productivos, es importante brindar una guía común.
- Es necesario porque es una propuesta innovadora y no siempre se tienen los conocimientos ni la experiencia para ejecutar el rol con calidad.
- El personal tiene ansias de conocimientos y superación en el tema, solo falta el medio y este puede serlo.

3.2.3.6 Factores que pueden atentar contra la utilización y modo de uso del manual.

Este objetivo es abordado por la pregunta cuatro donde se da la posibilidad a cada uno de los especialistas de especificar los factores que creen que pueden estar a favor de la aplicación del Manual del Arquitecto de Software y en contra del mismo. Mostrándose las siguientes opiniones:

En contra:

- Pueden atentar contra el manual la falta de experiencia y práctica que existe actualmente sobre el tema.
- La aplicación sin talleres y centros que se ocupen de rectorar el proceso.
- La resistencia a seguir una norma o reglamentación.
- La no adaptabilidad a los escenarios de aplicación.

- Diversidad de arquitecturas, no obligación de seguir una en común por la infraestructura productiva.
- Que no se dé a conocer la existencia del manual.
- Falta de conocimientos asociados a la disciplina.
- La experiencia en el tema que puedan tener los autores.
- La interdisciplinariedad de la arquitectura.
- Que no se publique en el ambiente de desarrollo y superación.

A favor:

- La necesidad de un trabajo como este y la calidad del mismo.
- La necesidad del conocimiento sobre el tema.
- La publicación en sitios Web.
- Que sea una norma de la Infraestructura Productiva.
- Que se de conocer y perfeccione constantemente.
- La presentación en conferencias y exposiciones.

Para evitar que tomen vida los criterios dados en contra del manual por parte de los especialistas, las autoras se proponen combatir los que dependen de ellas, por ejemplo, para impedir que no se dé a conocer la existencia del manual, este será publicado en el Sistema de Gestión y Conocimientos. Ante la dificultad existente en la universidad de que no se obliga a seguir una arquitectura en común por la infraestructura productiva, se sabe que en la universidad recientemente se creó el Centro de Consultoría el cual se encargará de eliminar este problema.

3.2.3.7 Resultados Finales.

Los resultados obtenidos en la validación realizada por el Método Delphy, dieron resultados positivos en cada uno de los indicadores planteados aunque hubo diversidad de criterios entre los especialistas que formaron el panel como resultado de los pocos conocimientos que poseen algunos, pero reinaron las respuestas positivas en cada objetivo analizado por lo que no fue necesaria la aplicación de otro cuestionario para validar la propuesta.

Resultados Finales de la Validación

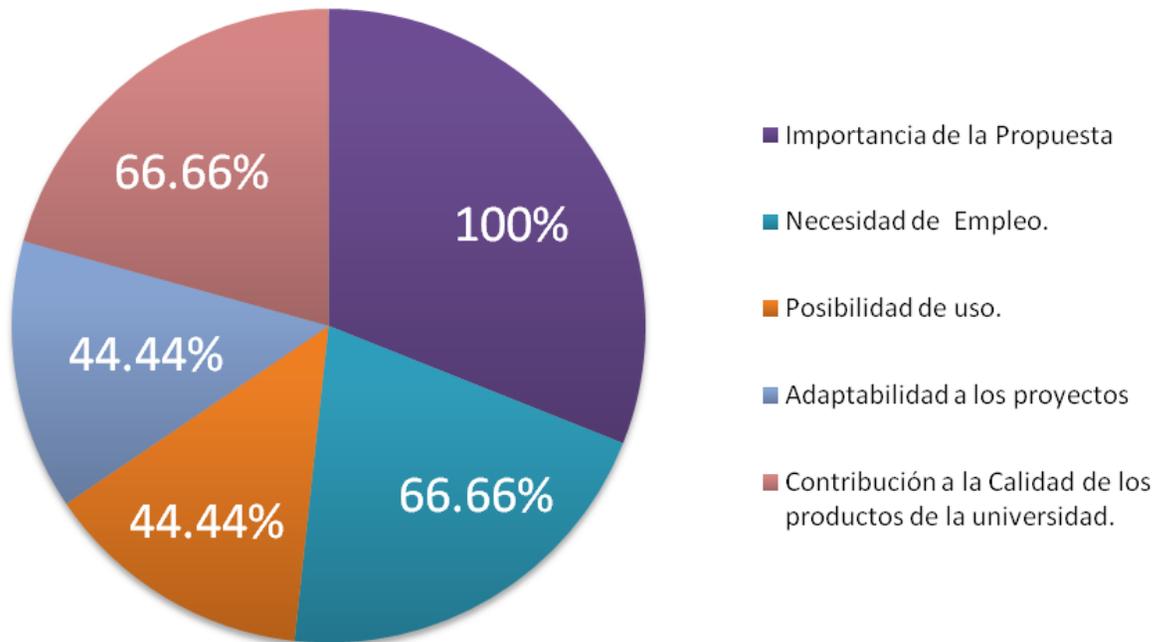


Figura 21 Porcentaje final de los indicadores.

Algunas de las opiniones aportadas por los especialistas en el punto número 5 del cuestionario:

- El manual debe usarse como una pequeña guía dentro de todos los conocimientos que debe poseer un arquitecto.
- Excelente idea y materialización en un trabajo como este, sería bueno que una vez aplicado dicho manual en algún proyecto se le incorpore al trabajo la experiencia y resultados del uso del mismo.
- El manual está orientado a un área muy general, sería bueno adaptarlo a cada área de la AS.
- Debe ser enriquecido con patrones más usados, ejemplos reales y tendencias arquitectónicas.
- Debe ser lucrado con opiniones de especialistas y mejorado constantemente.
- Como intención válida, necesaria y podría ser útil, demasiado alcance para una tesis de grado, aunque es importante que se hagan este tipo de trabajos.
- Se debe seguir estudiando y ampliando ante la aparición de nuevas metodologías, patrones, diseños arquitectónicos, etc.

El análisis de los resultados anteriores permite afirmar de modo general que el Manual del Arquitecto de Software, fue evaluado por los especialistas como efectivo, correcto y necesario para desarrollar la Arquitectura de los proyectos de la Universidad de las Ciencias Informáticas.

3.3 Validación por Caso de Estudio.

Para validar por casos de estudio se realiza un análisis del problema existente antes de existir el manual y después de ponerlo en disposición de todos los interesados. El manual tiene como fin eliminar todas las deficiencias existentes en cuanto al conocimiento abordado por él.

El caso de estudio en cuestión está dado por la situación existente con los Arquitectos de Software en la UCI, la cual evidencia que muchos de ellos no poseen los conocimientos necesarios, ni una bibliografía que se ajuste a las características de la universidad para obtener los mismos.

Las variables que se tienen en cuenta son:

- Tiempo: Da la medida del tiempo que se necesita para aprender.
- Conocimientos sobre AS: Muestra los conocimientos reales que existen sobre el tema.
- Existencia de un manual para los Arquitectos de Software: Para ayudar a los que desempeñan el rol a aprender a desarrollar las actividades que les competen.

3.3.1 Antes

Antes de existir el Manual la situación real era que los Arquitectos no cuentan con suficiente bibliografía específica sobre cómo realizar su trabajo, y la poca información con que cuentan los Arquitectos muchas veces no está organizada correctamente. Se tomó una muestra significativa de treinta personas de manera aleatoria, en los que se incluían el sexo masculino y el femenino, estudiantes que desempeñaban el rol en proyectos productivos y otros que no lo desempeñaban, también se tuvo en cuenta que fueran de diferentes cursos escolares, para aplicarle un cuestionario que permitiera evaluar los conocimientos reales sobre el tema. En este momento de la validación se observó que la variable tiempo estaba siendo afectada a razón de que las personas de la muestra necesitaban mucho más tiempo para estudiar sobre el tema debido a la falta de un manual donde encontrarán la información necesitada.

Una vez aplicadas las preguntas (Anexo 4: Cuestionario de validación por Caso de Estudio para antes de consultado el Manual) sin estos nunca antes consultar el Manual, se observó que existía desconocimiento del tema en la gran mayoría, los resultados se muestran a continuación:

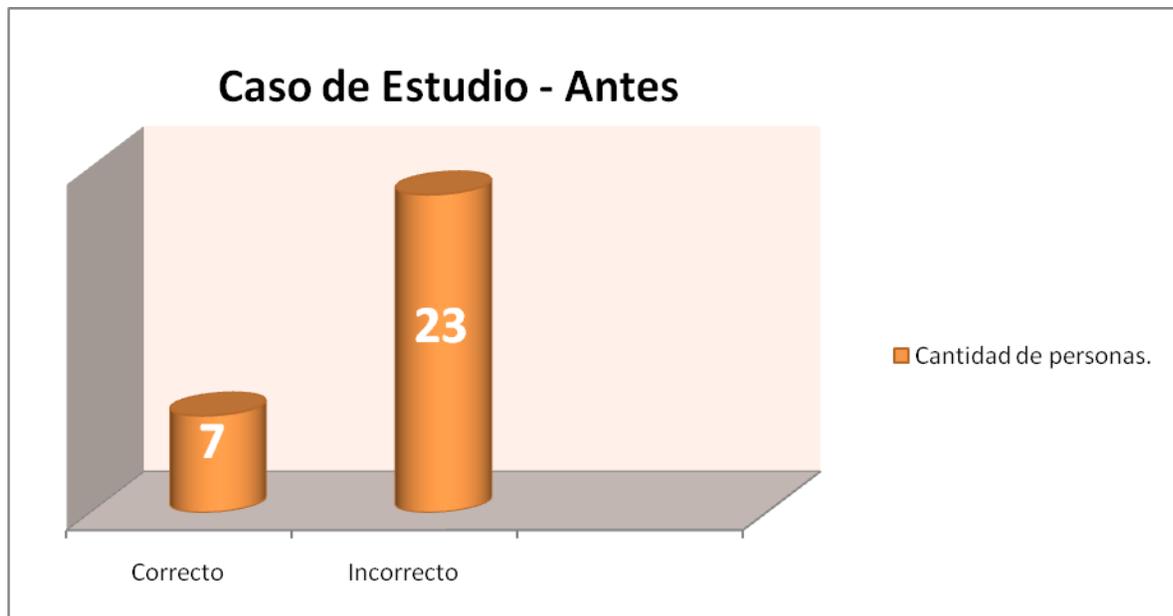


Figura 22 Primer cuestionario de la validación por Caso de Estudio.

Como se observa en la gráfica solo siete de los treinta evaluados respondieron positivamente la pregunta y veintitrés de manera negativa lo que confirma que existe desconocimiento del tema. Para dar respuesta al cuestionario se tardaron más del tiempo necesitado por una persona con conocimientos sobre el tema. En la pregunta aplicada en este cuestionario el encuestado no solo debía probar que poseía conocimiento sobre el tema, sino que debía aportar su criterio personal.

3.3.2 Después.

Luego de haber sido evaluado el conocimiento mediante el cuestionario antes de ser consultado el manual por los estudiantes que conformaban la muestra, se procede a poner a disposición de los mismos dicho Manual para que estos evacuen sus dudas respecto al tema o aprendan sobre el mismo. Después de dar un tiempo prudente para el estudio del mismo se aplica el segundo cuestionario (Anexo 5: Cuestionario de

validación por Caso de Estudio para después de consultado el Manual) para examinar si con el análisis del mismo los compañeros se apoderaron de nuevos conocimientos y validar así la eficacia del Manual.

En las dos preguntas con que consta este segundo cuestionario, la primera posee un carácter reproductivo de lo se aprendió en el Manual pero a la vez el encuestado debe dar su criterio personal, mientras que la segunda es un poco más profunda, porque posee un carácter aplicativo, porque el que la responde debe poner en práctica lo aprendido en el Manual. Los resultados de las mismas se muestran a continuación:

Caso de Estudio-Después-Pregunta 1.

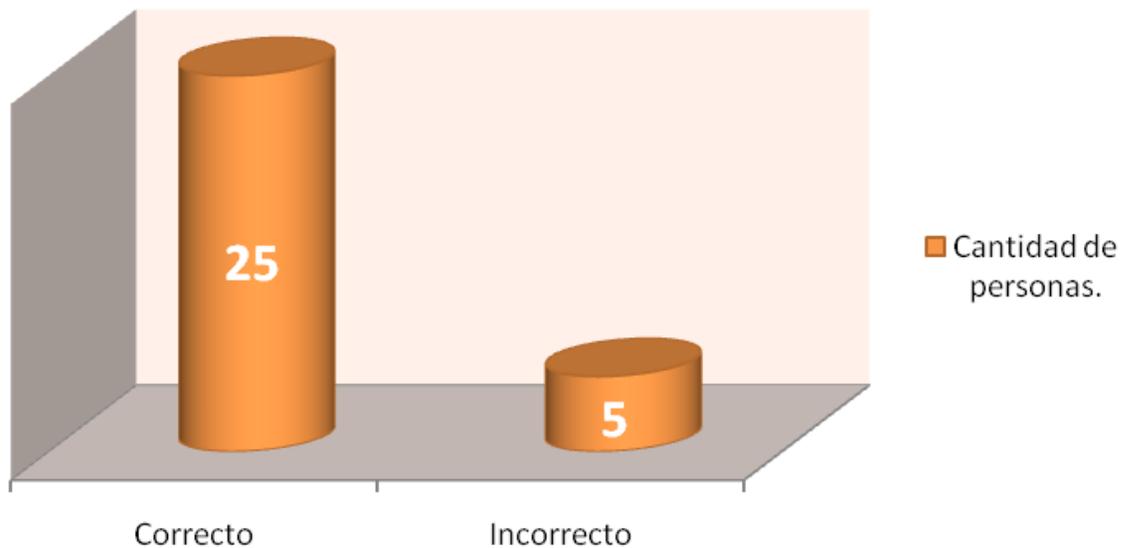


Figura 23 Primer pregunta del segundo cuestionario de la validación por Caso de Estudio.

Caso de Estudio-Después-Pregunta 2.

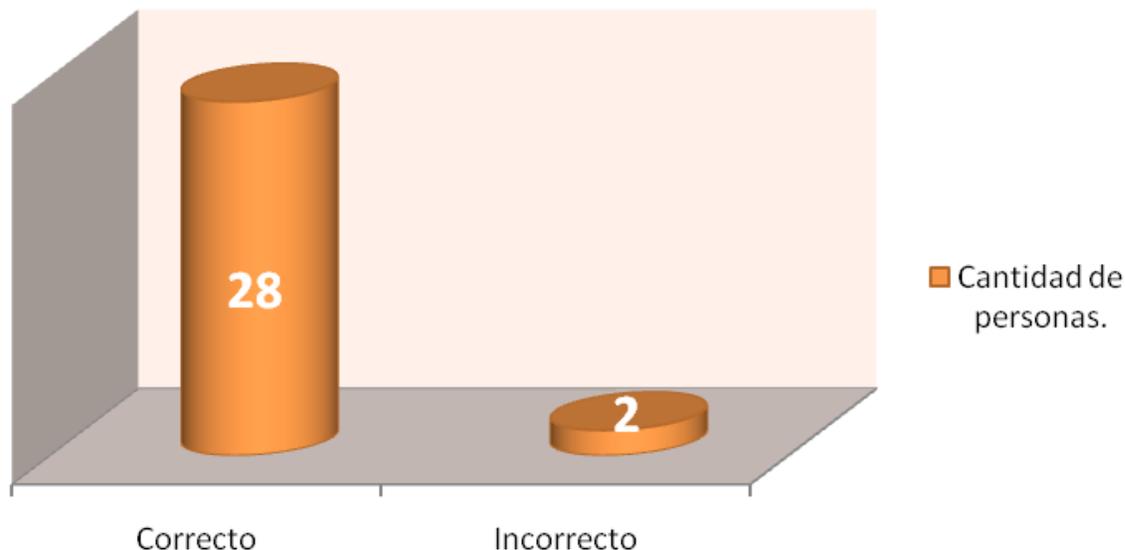


Figura 24 Segunda pregunta del segundo cuestionario de la validación por Caso de Estudio.

En los dos gráficos anteriores se muestra que el uso del manual aportó conocimientos que ayudaron a resolver la realidad que existía antes de su uso, pues los indicadores demuestran que la situación cambió debido a que antes solo existía un 2,33% de personas con conocimientos para responder correctamente el primer cuestionario y en la pregunta uno del cuestionario aplicado en este segundo momento de evaluación existe un 83,33% que respondió correctamente la misma y un 93,33% en la segunda pregunta que demuestra que los estudiantes pusieron en práctica lo aprendido, por tanto se puede afirmar que es viable el uso de manual para ayudar a los que desempeñan el rol a aprender a desarrollar las actividades que les competen, agregar que en este segundo momento de la evaluación las personas se tomaron menos tiempo para dar respuesta al cuestionario, lo que evidencia que la variable tiempo disminuyó positivamente ante la existencia del manual que brinda los conocimientos del tema.

En el siguiente gráfico a modo de resumen se muestra el porcentaje de respuestas positivas a cada una de las preguntas de ambos cuestionarios, o sea se mostrará el porcentaje de respuestas positivas antes del uso del manual y las positivas después de su uso, lo que evidencia una gran evolución en cuanto al aprendizaje del tema.

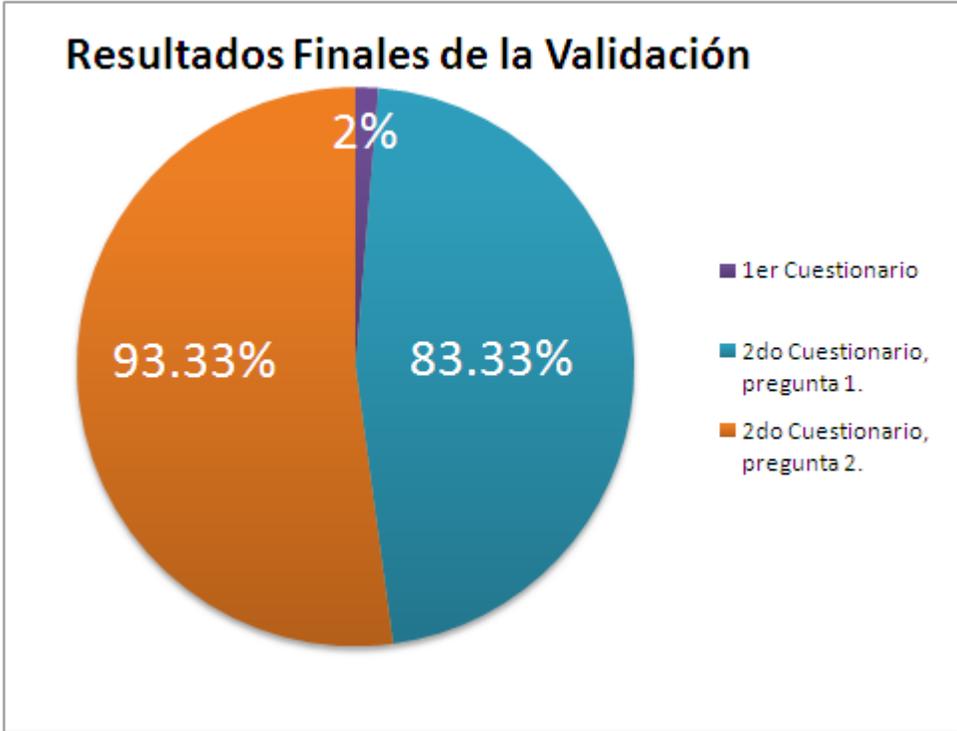


Figura 25 Porcentaje de respuestas correctas a los cuestionarios de la validación por Caso de Estudio.

CONCLUSIONES

Al culminar esta Investigación se ha llegado a las siguientes Conclusiones:

- Se propuso la utilización del Manual del Arquitecto de Software con el objetivo de eliminar las deficiencias existentes en las actividades realizadas por los arquitectos.

- Con la obtención del Manual del Arquitecto de Software, quedó plasmado un comportamiento organizado y eficaz, en las actividades a realizar por el arquitecto en la UCI, lo que contribuirá a la calidad del proceso de desarrollo de la arquitectura de software en la UCI.

- Se validó el manual teniendo como resultado la veracidad de la utilidad del mismo.

- Se comprobó que si las personas implicadas son capaces de usar correctamente el manual que está a su disposición, adquieren conocimientos referentes al tema, demostrando los resultados obtenidos en la validación

- Se demostró que la utilización de mapas conceptuales como medio de representación de la información posibilita una mayor asimilación de los conocimientos.

RECOMENDACIONES

En aras de contribuir al mejoramiento de la calidad del desarrollo de software se recomienda:

- Realizar nuevamente un estudio minucioso sobre los frameworks arquitectónicos y sobre los mecanismos y elementos de diseño.
- Valorar la necesidad de la incorporación de nuevas actividades a las propuestas en el manual.
- Automatizar el Manual del Arquitecto de Software.

BIBLIOGRAFÍA

Referencias bibliográficas.

[1] **C. R. Spooner.** "A Software Architecture for the 70's: Part I - The General Approach." *Software - Practice and Experience*, 1 (Enero-Marzo), pp. 5-37, 1971.

[2] **Dewayne E. Perry y Alexander L. Wolf.** "Foundations for the study of software architecture". *ACM SIGSOFT Software Engineering Notes*, 17(4), pp. 40–52, Octubre de 1992.

Bibliografía Consultada.

Frank DeRemer, Hans Kron. "Programming-in-the-large versus programming-in-the-small". *IEEE Transaction in Software Engineering*, 2, pp. 80-86, 1976.

Buschman, F., Meunier, R., Rohnert, H., Sommerlad, P. & Stal, M. (2000). *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*, Vol. 2. Estados Unidos: John Wiley & Sons.

Niklaus Wirth. "Program development by stepwise refinement", *Communications of the ACM*, 14(4), pp. 221-227, Abril de 1971.

Gómez, Omar Salvador. 2007. *Evaluando Arquitecturas de Software. Parte 1. Panorama General.* Evaluando Arquitecturas de Software. Parte 1. Panorama General. México: Brainworx, 2007.

Hidalgo, I (2004) *Patrones Arquitectónicos para Sistemas Colaborativos*, Tesis de Maestría en Ingeniería de Sistemas, Venezuela: Universidad Simón Bolívar.

Paul Clements. "A Survey of Architecture Description Languages". *Proceedings of the International Workshop on Software Specification and Design*, Alemania, 1996.

Erika Camacho, Fabio Cardeso, Gabriel Nuñez. 2004. *Arquitecturas de Software.* Arquitecturas de Software. 2004.

Gustavo Andrés Brey, Gastón Escobar, Nicolás Passerini y Juan Arías. 2005. *Arquitecturas de proyecto de IT. Evaluación de Arquitectura.* Arquitecturas de proyecto de IT. Evaluación de Arquitectura. Buenos Aires: s.n., 2005.

Indira Hidalgo, Anna Grimán, Luis Mendoza, y María Pérez. Evaluación de la Calidad de Patrones Arquitectónicos a través de Experimentos Cuantitativosπ. Evaluación de la Calidad de Patrones Arquitectónicos a través de Experimentos Cuantitativosπ. Venezuela: s.n.

Jorge Triñanes, María de las Nieves Freira. 2007. Gestión de Software. Gestión de Software. . [Online] octubre 20, 2007. [Cited: abril 18, 2009.] [Http: //www.fing.edu.uy/inco/cursos/gestsoft.'](http://www.fing.edu.uy/inco/cursos/gestsoft.)

Mollineda, Ramón. 2005. Arquitectura del Software: arte y oficio. Valencia: s.n., 2005.

2008. El Rol del Arquitecto del Software. [Book auth.] Pablo Andrés Carrillo, Sorey Bibiana García Zapata Ivis Rosa Vásquez Sierra. El Rol del Arquitecto del Software. Mdellín: s.n., 2008.

James Shore, Shane Warden. 2008. The Art of Agile Development. United States of America: O'Reilly Media, 2008.

José H. Canós, Patricio Letelier, M^a Carmen Penadés. Metodologías Ágiles en el Desarrollo de Software. Valencia: s.n.

Joskowicz, José. 2008. Reglas y Prácticas en eXtreme Programming. Canada: s.n., 2008.

Kniberg, Henrik. 2007. Scrum y XP desde las trincheras. Estados Unidos de América: C4Media Inc, 2007.

Len Bass, Paul Clements, Rick Kazman. 2003. Software Architecture in Practice, 2nd Edition. Software Architecture in Practice, 2nd Edition. S.I.: Addison-Wesley Professional, 2003.

2006. Marble Station. Marble Station. [Online] 2006. [Cited: marzo 5, 2009.] [http://www.marblestation.com/.](http://www.marblestation.com/)

Mejía, Nicolás Aristizábal. 2008. Un breve estudio sobre Arquitectura de Software. 2008.

Reynoso, Carlos Billy. 2004. Introducción a la Arquitectura de Software. Introducción a la Arquitectura de Software. Buenos Aires: s.n., 2004.

Slideshare. Slideshare. [Online] [Cited: enero 20, 2009.] <http://www.slideshare.net.>

Gricelda Medina Veloz, Fransisco Javier Luna Rosas, Jaime Muñoz Artiaga, Julio César Martínez Romo. 2008. Integrando mediante patrones de software una estrategia Fuzzy-Logic, en un servicio de balanceo dinámico de carga bajo corba. México: s.n., 2008. ISSN (Versión impresa):1405-5597.

ANEXOS

Anexo 1: Guía de observación.

- Actividades que realizan los arquitectos en su proyecto.
- Bibliografía consultada por los que desempeñan el rol.
- Capacitación a los Arquitectos.
- Deficiencias para los Arquitectos.
- Recursos con que cuentan los Arquitectos.
- Herramientas usadas por los Arquitectos.

Anexo 2: Encuesta a Arquitectos de proyectos en la UCI.

Nos encontramos realizando una investigación para la cual necesitamos de su ayuda, por esta razón le pedimos responder el cuestionario que a continuación le presentamos. Por favor, necesitamos su colaboración para transformar la realidad

Gracias.

1. Mencione las actividades que usted como Arquitecto realiza dentro de su proyecto.

2. ¿Posee usted bibliografía que le indique como hacer su trabajo?

Si_____ No_____

a. De ser positiva su respuesta menciónelas.

b. Tiene usted organizada esa bibliografía.

Si_____ No_____

c. De tener organizada la bibliografía diga de que modo tiene organizada dicha bibliografía.

3. ¿Recibe usted capacitación que le instruya como realizar su trabajo?

Si_____ No_____

Si recibe capacitación mencione el tipo de capacitación y quién o que institución o departamento se la brinda.

4. ¿Le alcanza a usted su tiempo para realizar su trabajo?

Si_____ No_____

De ser su respuesta negativa mencione los motivos:

5. Mencione algunas de las deficiencias que experimenta a la hora de aprender a ser arquitecto.
6. ¿Cuales recursos no dispone para realizar su trabajo?
7. Cuales herramientas usa para realizar su trabajo.
8. Conoce usted los lineamientos que establece la Dirección de Informatización de la UCI.
Si_____ No_____
9. Cuando surgió su proyecto, usted como arquitecto y el equipo que trabaja junto a usted decidieron:
Definir la arquitectura a usar, comenzando desde cero_____.
Reutilizar algunas cosas que ya estaban hechas por otros proyectos_____.

Anexo 3: Cuestionario a Miembros del Panel de Especialistas.

Nos encontramos realizando la validación de el Manual del Arquitecto de Software para la cual necesitamos de su ayuda, por esta razón le pedimos responder el cuestionario que a continuación le presentamos. El presente cuestionario tiene carácter anónimo. Por favor, necesitamos su colaboración para transformar la realidad.

1. ¿Considera usted que es importante el desarrollo de un Manual que presente las actividades a realizar por el Arquitecto de Software, para facilitar el aprendizaje y trabajo en la Universidad de las Ciencias Informáticas de aquellas personas que desempeñen dicho rol?

—Si —No —No Sé

¿Por Qué? _____
_____.

2. Valore el cumplimiento de los criterios mostrados a continuación en el manual, para ello de una puntuación de 1 a 5 según considere usted que el manual los recoge.

—Necesidad del empleo de la propuesta.

—Posibilidad de uso.

—Adaptabilidad a proyectos productivos.

3. ¿Considera usted que se pueda usar exitosamente en la universidad el Manual propuesto como bibliografía para consultar, de manera que permita mejorar la calidad de los productos que se desarrollan en la UCI dando el cumplimiento a su objetivo?

—Si — No —No Sé

¿Por Qué? _____
_____.

4. ¿Qué factores cree usted que podría atentar contra el correcto uso de este Manual en la UCI y cuales lo facilitarían?

_____.

5. Haga un comentario o aporte sobre el Manual que usted está evaluando. (El comentario es libre y debe reflejar algún elemento de interés que aporte elementos a la mejora del mismo):_____

_____.

Anexo 4: Cuestionario de validación por Caso de Estudio para antes de consultado el Manual.

Nos encontramos realizando la validación de el Manual del Arquitecto de Software para la cual necesitamos de su ayuda, por esta razón le pedimos responder el cuestionario que a continuación le presentamos. El presente cuestionario tiene carácter anónimo. Por favor, necesitamos su colaboración para transformar la realidad.

¿Sabe usted que son los estilos arquitectónicos? De saber, defínalos con sus palabras. Mencione algunos que usted conozca.

—Sí —No.

_____.

Anexo 5: Cuestionario de validación por Caso de Estudio para después de consultado el Manual.

Nos encontramos realizando la validación de el Manual del Arquitecto de Software para la cual necesitamos de su ayuda, por esta razón le pedimos responder el cuestionario que a continuación le presentamos. El presente cuestionario tiene carácter anónimo. Por favor, necesitamos su colaboración para transformar la realidad.

¿Considera usted que los requerimientos nos funcionales son fundamentales para el éxito de un producto de software? Justifique su respuesta.

—Sí. —No.

1. De las vistas arquitectónicas de un software, la Vista de Despliegue posee gran significación

a) ¿Mediante cual diagrama puede ser modelada?

b) Confeccione dicho diagrama a partir del siguiente fragmento extraído de un caso de estudio.

En la librería del centro de estudios Martianos cuentan con un terminal desde donde los lectores pueden acceder al catálogo de la misma, así como acceder a libros y revistas online. Es necesario el uso de un dispositivo de impresión para que los usuarios puedan imprimir determinados documentos. Por razones de seguridad el Servidor de Base de Datos se encuentra separado del Servidor de aplicaciones.

Glosario de Términos.(falta)

