



Universidad de las Ciencias Informáticas

Facultad # 4

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

Título: Procedimiento de Revisión Técnica Formal en el
Grupo de Calidad de FORTES de la facultad 4

Autor(a): Elena González Revilla

Tutor: Ing. Lizardo Ramírez Tabuada

Co-tutor(a): Ing. Lisandra Peña Espinosa

“Año 56 de la Revolución”

La Habana, Junio 2014

Declaración de auditoría

Declaro ser la autora de la presente tesis que tiene por título “Procedimiento de Revisión Técnica Formal en el Grupo de Calidad de FORTES de la facultad 4” y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Elena González Revilla

Firma del Autor(a)

Ing. Lizardo Ramírez Tabuada

Firma del Tutor

Ing. Lisandra Peña Espinosa

Firma del Co-tutor



*He cometido errores, pero ninguno
estratégico, simplemente táctico.*

Fidel Castro

Agradecimientos

*Hoy quiero darles las gracias a todas aquellas **personas** que confiaron en mí y que siempre me apoyaron en los momentos más difíciles de mi vida. A mi mamá **Eloisa** que me da fuerzas para vencer cualquier obstáculo que se presenta en mi camino, por ser mi guía y razón de ser, a mi papá **Oscar** por ser mi ejemplo a seguir y fuente de inspiración y a **Eloisita** por ser la mejor hermana del mundo.*

*A mi tutor **Lizardo** y a mi co-tutora **Lisandra** por su paciencia y dedicación, a los profesores consultantes **Arce**, **Ernesto**, **Oswaldo** y **Graciela** por su ayuda incondicional y a todos los **profesores** que de una forma u otra me ayudaron a mi formación como profesional.*

*A mis amistades **Ramoncito**, **Anali**, **Yanet**, **Yulix** y **Dany** que siempre se mantuvieron a mi lado desinteresadamente, a mis **compañeros** de aula por demostrarme que en la unidad está la fuerza y en especial a mi novio **Ramón** por su **AMOR** y comprensión.*

Elena

Dedicatoria

Dedico el presente trabajo de diploma a toda mi familia por el gran apoyo que me han brindado, en especial a mi mamá, papá, hermana y a mi novio por estar siempre conmigo cuando más lo necesitaba.

Elena

Resumen

Para una mejor calidad técnica en la documentación y entregables de un proyecto se elaboró un procedimiento de Revisión Técnica Formal en el Grupo de Calidad de FORTES de la facultad 4, el cual permite conocer el estado técnico del proyecto mediante las no conformidades que se encuentran en las etapas de arquitectura, diseño, requerimiento y modelo de datos. Para el desarrollo del procedimiento se realizó un análisis de los conceptos que tributan al avance del trabajo y se analizan algunas investigaciones de las revisiones de *software*, con el objetivo de obtener actividades y buenas prácticas para incorporarlas al procedimiento definido.

Se desarrollaron las actividades, se definieron los roles con sus responsabilidades y un conjunto de artefactos para documentar cada una de las tareas realizadas. Para probar el procedimiento, se aplicó a un proyecto real y en curso del centro FORTES de la facultad 4.

Palabras clave: procedimiento, revisión, revisión técnica formal.

Índice

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	7
1.1 PRINCIPALES CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	7
1.1.1 <i>Calidad del software</i>	7
1.1.2 <i>Revisiones de software</i>	7
1.1.3 <i>Tipos de técnicas de revisiones</i>	8
1.1.4 <i>Clasificación de las revisiones</i>	9
1.1.5 <i>Revisión Técnica</i>	11
1.1.6 <i>Clasificación de las revisiones según su grado de formalidad</i>	11
1.1.7 <i>Revisión Técnica Formal</i>	10
1.1.8 <i>Restricciones de la revisión técnica formal</i>	11
1.1.9 <i>Directrices de la revisión técnica formal</i>	11
1.2 ANÁLISIS DE PROCEDIMIENTOS DE REVISIONES DE SOFTWARE	12
1.2.1 <i>Propuesta de la Universidad del Valle-Cochabamba</i>	13
1.2.2 <i>Propuesta de la Universidad Nacional Experimental “Rómulo Gallegos”</i>	21
1.2.3 <i>Propuesta del Instituto Superior Politécnico José Antonio Echeverría</i>	22
1.2.4 <i>Propuesta del Centro de Estudios de Ingeniería de Sistema</i>	23
1.2.5 <i>Propuesta de la Universidad de la Ciencias Informáticas</i>	24
1.3 CONCLUSIONES	26
CAPÍTULO 2. ELABORACIÓN DEL PROCEDIMIENTO	21
2.1 OBJETIVO Y ALCANCE DEL PROCEDIMIENTO	21
2.2 DESCRIPCIÓN DEL PROCEDIMIENTO	21
2.3 ROLES Y SUS RESPONSABILIDADES	22
2.4 ACTIVIDADES GENERADAS EN LAS FASES DEL PROCEDIMIENTO	23
2.4.1 <i>Actividades en la fase de planificación de la revisión</i>	24
2.4.2 <i>Actividades en la fase de ejecución de la revisión</i>	24
2.4.3 <i>Actividades en la fase de análisis de los resultados</i>	25
2.5 ARTEFACTOS DEL PROCEDIMIENTO	26
2.5.1 <i>Acta de reunión de inicio</i>	27
2.5.2 <i>Listas de chequeo por etapas</i>	28

Índice

2.5.3 Registro de no conformidades	37
2.5.4 Acta de reunión de cierre	38
2.6 CONCLUSIONES.....	39
CAPÍTULO 3. PROBAR EL PROCEDIMIENTO	41
3.1 APLICACIÓN DEL PROCEDIMIENTO A UN PROYECTO PRODUCTIVO	41
3.1.1 Planificación de la revisión	41
3.1.2 Ejecución de la revisión	43
3.1.3 Análisis de los resultados.....	50
3.2 CONCLUSIONES.....	54
CONCLUSIONES GENERALES	55
BIBLIOGRAFÍA	56
ANEXOS	58

Índice de Figuras

FIGURA 1: TIPOS DE TÉCNICAS DE REVISIONES (ELABORACIÓN PROPIA).....	9
FIGURA 2: CLASIFICACIÓN DE LAS REVISIONES (ELABORACIÓN PROPIA).....	11
FIGURA 3: DESCRIPCIÓN DEL PROCEDIMIENTO (ELABORACIÓN PROPIA).....	22
FIGURA 4: ACTIVIDADES EN CADA FASE (ELABORACIÓN PROPIA).....	23
FIGURA 5: ACTA DE REUNIÓN INICIO (ELABORACIÓN PROPIA).....	27
FIGURA 6: LEYENDA DE LAS LISTAS DE CHEQUEO (ELABORACIÓN PROPIA).....	28
FIGURA 7: REGISTRO DE NO CONFORMIDADES (ELABORACIÓN PROPIA).....	38
FIGURA 8: ACTA DE REUNIÓN DE CIERRE (ELABORACIÓN PROPIA).....	39
FIGURA 9: RELACIÓN DE LAS NO CONFORMIDADES ENCONTRADAS EN LA RTF.....	53

Índice de Tablas

TABLA 1: LISTA DE CHEQUEO DE ARQUITECTURA (ELABORACIÓN PROPIA).	29
TABLA 2: LISTA DE CHEQUEO DE DISEÑO (ELABORACIÓN PROPIA).	34
TABLA 3: LISTA DE CHEQUEO DE REQUERIMIENTO (ELABORACIÓN PROPIA).....	35
TABLA 4: LISTA DE CHEQUEO DE MODELO DE DATOS (ELABORACIÓN PROPIA).....	36
TABLA 5: CRONOGRAMA.....	41
TABLA 6: ACUERDOS TOMADOS.	42
TABLA 7: LISTA DE CHEQUEO DE ARQUITECTURA.	43
TABLA 8: LISTA DE CHEQUEO DE DISEÑO.	46
TABLA 9: LISTA DE CHEQUEO DE REQUERIMIENTO.....	46
TABLA 10: LISTA DE CHEQUEO DE MODELO DE DATOS.	48
TABLA 11: NO CONFORMIDADES DE LA ETAPA DE ARQUITECTURA.	50
TABLA 12: NO CONFORMIDADES DE LA ETAPA DE DISEÑO.	51
TABLA 13: NO CONFORMIDADES DE LA ETAPA DE REQUERIMIENTO.....	51
TABLA 14: NO CONFORMIDADES DE LA ETAPA DE MODELO DE DATOS.	52

Introducción

En el mundo, la industria del *software* actualmente se caracteriza por ser muy competitiva. Muchas empresas se centran en la calidad de sus productos o servicios para mejorar su posición en el mercado y lograr una mayor satisfacción del cliente. Uno de los pilares fundamentales para garantizar la calidad son las revisiones de *software* en cada fase de la Ingeniería del *Software*.

Las empresas e instituciones cubanas tales como el Centro de Desarrollo de Tecnología para la Defensa (XETID), la Empresa Nacional de *Software* (DESOFT), el Complejo de Investigaciones de Tecnologías Integradas (CITI-CUJAE) y la Universidad de las Ciencias Informáticas (UCI) dan importantes pasos con el firme propósito de producir *software*, dentro de ellas se destaca la UCI que surge con el objetivo de formar profesionales comprometidos con su patria y altamente calificados en la rama de la Informática, producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación y servir de soporte a la industria del país.

La UCI ha centralizado todos sus proyectos en los centros productivos, entre ellos se encuentra el Centro de Tecnologías para la Formación (FORTES). Este centro, dentro de su estructura cuenta con un Grupo de Calidad que ofrece servicios para el aseguramiento, verificación y validación de los productos desarrollados.

El Grupo de Calidad de FORTES (GCF) está conformado por profesores y estudiantes y tiene como principal objetivo ofrecer servicios a todos los proyectos productivos del centro e interesados. Entre las prestaciones brindadas se encuentran las revisiones, auditorías, capacitaciones y pruebas de *software* en las diferentes etapas de desarrollo.

Actualmente en el GCF no existen servicios que permitan validar la completitud y corrección de los entregables en los proyectos que son revisados. No se cuenta con un mecanismo que permita prevenir los riesgos que puedan derivarse en etapas posteriores del proyecto, tales como inconsistencias, ambigüedades y no cumplimiento de estándares. En las revisiones que se realizan no se verifica la consistencia interna en el expediente de proyecto y su coherencia con los requerimientos, debido a que el alcance está enmarcado en que se generen los artefactos y no en una revisión técnica formal a la documentación de arquitectura, diseño, requerimiento y modelo de datos.

Introducción

Debido a la situación anteriormente expuesta, se plantea el siguiente **problema a resolver** ¿Cómo definir el procedimiento de revisión técnica formal en el Grupo de Calidad de FORTES de la facultad 4?

Para dar cumplimiento al problema planteado se establece como **objetivo general** de la presente investigación elaborar el procedimiento de revisión técnica formal en el Grupo de Calidad de FORTES de la facultad 4.

A partir del objetivo general se derivan como **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Definir las fases, actividades, artefactos y roles de la revisión técnica formal.
- Probar el procedimiento de revisión técnica formal.

Se define como **objeto de estudio** el procedimiento de revisión técnica formal, enmarcado en el **campo de acción** la revisión técnica formal en el Grupo de Calidad de FORTES de la facultad 4.

Con vista a resolver el problema planteado se propone la siguiente **idea a defender**: el procedimiento de revisión técnica formal en el Grupo de Calidad de FORTES permitirá detectar defectos técnicos en la documentación y entregables de los productos desarrollados en el Centro.

Para garantizar el cumplimiento de los objetivos específicos se proponen las siguientes **tareas de investigación**:

1. Estudio de los conceptos relacionados con la revisión técnica formal.
2. Análisis de los procedimientos similares.
3. Definición de las fases, actividades, artefactos, roles involucrados y sus responsabilidades en la propuesta de solución.
4. Probar la propuesta aplicándola a un proyecto del Centro.

Durante el desarrollo de esta investigación se hizo necesario profundizar en el estudio de los temas abordados, por lo que se utilizaron varios **métodos de investigación**:

Método teórico:

El método teórico consiste en crear condiciones para ir más allá de las características superficiales de la realidad y posibilita el conocimiento del estado del arte del fenómeno, su evolución en una etapa determinada y su relación con otros fenómenos. Los métodos teóricos que se aplican:

Introducción

- **Histórico-lógico:** se utilizó para constatar teóricamente la evolución de la revisión técnica formal. Este método se basa en el estudio crítico de los antecedentes asociados a la propuesta que se pretende construir y para utilizar estos como punto de referencia y comparación de los resultados alcanzados.
- **Analítico-sintético:** se utilizó para el análisis de diferentes tipos de bibliografía; permitiendo la extracción de los elementos más importantes que se relacionan con la revisión técnica formal. Este método permitió descomponer el problema de investigación en elementos por separado para así poder profundizar en el estudio de cada uno de ellos y luego sintetizarlos en la solución de la propuesta.

Métodos empíricos:

Los métodos empíricos se aproximan al conocimiento del objeto mediante su conocimiento directo y el uso de la experiencia. Para el procedimiento se aplica el **método de la observación** que consiste en la percepción directa del objeto de investigación. De los métodos de la observación se aplica el siguiente:

- **La observación investigativa:** es el instrumento universal del científico, además permite conocer la realidad mediante la percepción directa de los objetos y fenómenos. Se utilizó durante todo el desarrollo de la investigación, para el análisis de un antes y un después de acuerdo a la problemática presentada, así como al probar la propuesta.

El presente trabajo ha sido organizado en tres capítulos los cuales abordan la investigación del procedimiento a realizar. Se estructura de la siguiente forma:

Capítulo1: Fundamentación teórica

En este capítulo se presenta el marco teórico de la revisión técnica formal y se caracterizan procedimientos que se relacionan con las mismas.

Capítulo2: Elaboración del procedimiento

A partir de un estudio en la fundamentación teórica se describe la propuesta de solución y se definen los artefactos, roles involucrados y sus responsabilidades en el procedimiento desarrollado.

Capítulo3: Probar el procedimiento

En este capítulo se aplica la propuesta a un proyecto real y se analizan los resultados obtenidos del procedimiento diseñado.

Capítulo 1. Fundamentación teórica

En el presente capítulo se realiza un estudio de los conceptos que tributan al desarrollo del trabajo y se analizan algunas investigaciones de las revisiones de *software*, las cuales son de gran ayuda para la elaboración y aplicación de un procedimiento de revisión técnica formal en el Grupo de Calidad de FORTES (GCF).

1.1 Principales conceptos asociados al dominio del problema

1.1.1 Calidad del *software*

Es difícil establecer un concepto abarcador que satisfaga todas las variantes de lo que se entiende por calidad del *software*, para este término existen muchas definiciones, tales como:

- Según la IEEE estándar 610-1900 “La calidad del *software* es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario” (1).
- La norma ISO 8402:1994 plantea que es “El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas” (2).
- Roger Pressman define que es “La concordancia del *software* producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario” (3).

Para la presente investigación la autora considera más apropiada la definición de calidad de *software* de Roger Pressman, teniendo en cuenta que le presta mayor importancia a los requerimientos y estándares que el producto debe cumplir. Lo antes planteado es posible verificarlo a través de las revisiones de *software*.

1.1.2 Revisiones de *software*

Muchos autores han definido el concepto de revisiones de *software*. A continuación se citan los que para esta investigación han resultado de mayor relevancia:

- La IEEE estándar 1997 plantea que las revisiones de *software* sirven para validar la calidad y/o el estado de un producto. Este producto a revisar puede ser un documento, un módulo, un prototipo, etc. Las revisiones son un proceso o reunión durante el cual un producto de *software* se presenta

Capítulo 1. Fundamentación teórica

al personal del proyecto, administradores, usuarios, clientes u otras partes interesadas para comentario o aprobación (4).

- Roger Pressman en su edición en el 2005 define “las revisiones del *software* son un “filtro” para el proceso de ingeniería del *software*. Se aplican en varios momentos del desarrollo del *software* y sirven para detectar errores y defectos que puedan así ser eliminados” (5).

Al efectuar las revisiones de *software* se mejora la calidad del producto, teniendo en cuenta que estas (6):

- Posibilitan mediante su aplicación una mejora directa de la calidad del objeto que se examina.
- Al facilitar la comunicación entre los miembros del equipo de desarrollo provocan, de forma indirecta, una mejora en la calidad del proceso de desarrollo.
- Facilitan el control del costo y el tiempo.

Una revisión de *software* tiene como objetivos (7):

- Señalar la necesidad de mejoras en el producto.
- Continuar las partes de un producto en las que no es necesaria o no es deseable una mejora.
- Conseguir un trabajo técnico de una calidad más uniforme, o al menos más predecible, que la que puede ser conseguida sin revisiones, con el fin de hacer más manejable el trabajo técnico.

Después de analizar algunas definiciones se llega a la conclusión que las revisiones se realizan desde las primeras fases de desarrollo del proyecto. Estas se ejecutan a la documentación y entregables para detectar si alguno no está enteramente elaborado y hacer recomendaciones con el objetivo de validar la completitud del expediente del proyecto.

Las revisiones de *software* posibilitan una mejora directa de la calidad del objeto que se examina y permite un control del tiempo del desarrollo del mismo. Además, se consigue que la evaluación técnica no necesariamente sea realizada por personas involucradas en el equipo de desarrollo, sino en otras personas técnicamente competentes y objetivas.

1.1.3 Tipos de técnicas de revisiones

Para las revisiones existen dos tipos de técnicas que son importantes y hacen una combinación adecuada para detectar gran cantidad de defectos que mejoran la calidad del producto final, estas se clasifican en técnicas estáticas y dinámicas (6).

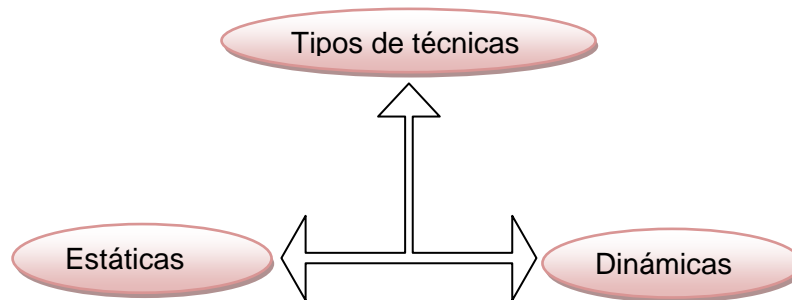


Figura 1: Tipos de técnicas de revisiones (Elaboración propia).

Estáticas: análisis y chequeo de documentos de requisitos, diagramas de diseño, código fuente. Son visuales y se realizan sin necesidad de que el *software* esté ejecutándose.

Dinámicas: pruebas sobre implementación real (sólo pueden usarse cuando ya se tiene código ejecutable). Son las que detectan los defectos ejecutando el *software*, fundamentalmente son las ejecutadas en la fase de prueba del proyecto.

1.1.4 Clasificación de las revisiones

La IEEE en su estándar 1028 y Edith Alhelí Martínez Mata (Consultor especializado en Aseguramiento de la Calidad) clasifican las revisiones en (4) (8):

1- Inspecciones de *software*

La inspección de *software* es un examen visual de un producto con el objetivo de detectar e identificar anomalías, incluyendo errores y desviaciones de los estándares y especificaciones. Son exámenes llevados a cabo por facilitadores imparciales que están entrenados en las técnicas de inspección. La toma de medidas correctivas o investigativas para una anomalía es un elemento obligatorio de una inspección de *software*, aunque la solución no debe ser determinada en la reunión de inspección. Se obtienen métricas (producto y proceso).

2- Administración de revisiones

Una evaluación sistemática de una adquisición de *software*, suministro, desarrollo, operación, o el proceso de mantenimiento realizadas por o en nombre de la administración que supervisa el progreso, determina el estado de los planes y cronogramas, confirma los requisitos y su sistema de asignación, o evalúa la efectividad de los enfoques administrativos usados para lograr la aptitud del propósito.

Capítulo 1. Fundamentación teórica

3- Auditorías

Un examen independiente de un producto de *software*, proceso de *software*, o un conjunto de procesos de *software* para evaluar el cumplimiento de las especificaciones, normas, acuerdos contractuales, u otros criterios.

4- *Walkthrough* (Visita guiada o recorridos)

Una técnica de análisis estático en la que un diseñador o programador guía a miembros del equipo de desarrollo y otras partes interesadas a través de un producto de *software*, y los participantes hacen preguntas y hacen comentarios sobre los posibles errores, violaciones de los estándares de desarrollo y otros problemas. No tienen métricas.

5- Revisiones cruzadas

Las revisiones cruzadas determinan el estatus del producto. Facilitan el intercambio de información típicamente utilizadas para diseño y codificación y se recolectan algunas métricas.

6- Revisiones de gestión

Las revisiones de gestión se efectúan con el objetivo de controlar la progresión del proyecto, aunque se pueden evaluar los riesgos asociados con relación al costo, escala de tiempo, recursos utilizados y calidad del producto. Estas revisiones se centran en el chequeo del proceso de desarrollo de *software*, lo cual incluye gestión de personal, desarrollo de cronograma y cumplimiento de los planes establecidos. Constituye una forma de chequear la correspondencia entre la planificación definida y la realizada.

7- Revisión técnica

La revisión técnica es una evaluación sistemática de un producto de *software* por un equipo de personal calificado los cuales examinan su idoneidad para el fin que se quiere, además de identificar las discrepancias con las especificaciones y estándares. También pueden ofrecer recomendaciones y exploración de distintas alternativas. Busca consenso útil como capacitación. Se expone el producto y aprobación para productos complejos o de nueva tecnología. Para su realización se requiere experiencia técnica y las métricas no son obligatorias.

Después de analizar varias clasificaciones de revisiones de *software* por diferentes autores se decide profundizar en las revisiones técnicas, porque estas no solo verifican que la documentación del proyecto esté, sino que se encuentre elaborada correctamente, analiza y comprueba sistemáticamente el producto y exige personal capacitado y altamente comprometidos con la tarea a realizar.

Capítulo 1. Fundamentación teórica

1.1.5 Revisión Técnica

La revisión técnica es una forma de verificar el trabajo de los desarrolladores, pues los errores se pueden introducir fácilmente y ningún producto está exento de ellos. Es factible para examinar el trabajo técnico desde otra perspectiva que no sea la de los propios desarrolladores, pues algunas clases de errores son más difíciles de detectar por el propio autor que por otra persona (7).

Las revisiones técnicas más comunes son:

- Revisión de la especificación de requisitos.
- Revisión del diseño.
- Revisión del código.
- Revisión de las pruebas.
- Revisión del manual de usuario.

Uno de los objetivos fundamentales de las revisiones técnicas es ofrecer a los gestores información fiable acerca de los aspectos técnicos del proceso de desarrollo de *software*, de la misma forma que les llega información fiable acerca de los costos y la programación del trabajo, información que puedan utilizar para tomar decisiones adecuadas y dirigir con éxito el proyecto (7).

1.1.6 Clasificación de las revisiones según su grado de formalidad

Las revisiones se clasifican según su grado de formalidad, estas pueden ser de tipo informal cuando no necesitan de una planificación y formal cuando se realizan de forma organizada. A continuación se definen estos tipos de revisiones.

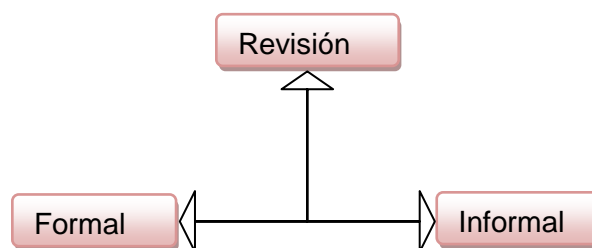


Figura 2: Clasificación de las revisiones (Elaboración propia).

Capítulo 1. Fundamentación teórica

Revisión informal

La revisión informal es aquella que no necesita de una planificación anticipada y se realiza de una forma informal como es el tratar o abordar un problema en una conversación en torno a una mesa bebiendo un café (5).

Este tipo de revisión, también llamada inadecuadamente sólo revisión (lo cual genera confusión con el nombre genérico de todas estas técnicas) se basan simplemente en un intercambio de opiniones entre los participantes; que pueden o no presentar una guía de pasos para su realización que es de forma constante, debido a las confusiones que esto puede generar, y la necesidad de registrar los datos obtenidos mediante este proceso este tipo de revisión evoluciona posteriormente, en función de estas características (9):

- No tienen un proceso definido.
- Usualmente no se planifican.
- No se definen roles para su realización.

Revisión formal

La revisión formal constituye una de la acción más necesaria para obtener un *software* con la calidad requerida; pues se enfoca en el producto y puede realizarse en cualquier momento del ciclo de desarrollo aunque lo más recomendable es ponerla en práctica desde el inicio (5).

En este tipo de revisiones los participantes son responsables de la fiabilidad de la evaluación y generan un informe que refleja el acto de la revisión (6).

- Es un evento público.
- Se informa por escrito los resultados.
- Todos los participantes son responsables de la calidad de la revisión.

Ventajas:

- Los informes que se generan sirven como hitos para el proyecto.
- Promueven una mejor preparación por parte de los participantes.
- Tiene objetivos definidos y procesos documentados.
- Tiene roles definidos.
- Su realización incluye métodos o técnicas para la detección de defectos.
- Se recolectan datos para el control del proceso.

Capítulo 1. Fundamentación teórica

Desventaja:

- La formalidad hace que este tipo de reuniones sean un tanto impersonales.

En este tipo de revisiones los informes generados sirven como hitos del proyecto y promueven una mejor preparación por parte de los participantes, sin embargo debido a su formalidad puede que en ocasiones sean un tanto impersonales (6).

Por otro lado según el objeto que se revise se suele diferenciar entre las revisiones con orientación técnica y las revisiones orientadas a la gestión, también conocidas como revisiones de proyecto (7).

Por tanto, en la presente investigación se propone como técnica de evaluación la revisión formal, pues esta genera informes que sirven de guía y ayuda a los proyectos evaluados. Garantiza que los participantes se encuentren preparados, tiene objetivos y roles definidos, incluye métodos o técnicas para la detección y descripción de los defectos y se obtienen datos que contribuirán al control del procedimiento.

1.1.7 Revisión Técnica Formal

La revisión técnica formal (RTF) purifica las actividades de Ingeniería de *Software*, se aplica en diversos momentos del desarrollo para detectar defectos. Valida la completitud y corrección de los entregables de un proyecto, previniendo en forma temprana sobre potenciales problemas y riesgos que puedan derivarse en etapas posteriores del proyecto como inconsistencias, ambigüedades, no cumplimiento de estándares. El servicio consiste en la revisión formal de la documentación de arquitectura, diseño, requerimiento y modelo de datos, con el objetivo de (10):

- Verificar la consistencia interna de la documentación y su coherencia con los requerimientos.
- Verificar el cumplimiento de estándares del cliente.
- Validar la completitud y facilidad de lectura de la documentación.
- Proponer mejoras, agregados y estándares nuevos.

Los objetivos de la RTF son (6):

- Descubrir errores en la función, la lógica o la implementación de cualquier representación de *software*.
- Verificar que el *software* bajo su revisión alcanza sus requisitos funcionales.
- Garantizar que el *software* ha sido desarrollado de acuerdo a los estándares predefinidos.

Capítulo 1. Fundamentación teórica

- Conseguir un *software* desarrollado uniformemente.

Los motivos básicos por los que realizar una RTF, se deben principalmente al trabajo revisado, porque existen errores que son percibidos más fácilmente por otras personas que por los creadores. Sirven como campo de entrenamiento permitiendo que los especialistas puedan observar los diferentes enfoques a la arquitectura, requerimiento, diseño y modelo de datos del *software*. También para promover la seguridad y la continuidad, porque varias personas se familiarizarán con partes del *software* que de otro modo no hubieran visto.

1.1.8 Restricciones de la revisión técnica formal

Para realizar la RTF es necesario regirse por algunas restricciones que ayuden al procedimiento a tener la mayor calidad (11).

- Se debe convocar a la RTF normalmente entre tres y cinco personas.
- Se debe preparar por adelantado, pero sin que requiera más de dos horas de trabajo previo por persona.
- La duración de la RTF debe ser menor a dos horas.
- Se debe centrar en una parte específica y pequeña del *software* total.

1.1.9 Directrices de la revisión técnica formal

Establecer de antemano directrices para conducir la RTF es una práctica recomendable, distribuyéndolas entre los revisores, para ser analizadas y finalmente seguidas (6).

- **Revisar el producto, no al productor:** una RTF involucra personas y egos. Se deben señalar los errores adecuadamente; el tono de la reunión debe ser distendido y constructivo.
- **Fijar una agenda y mantenerla (es decir no desviar el tema de la reunión):** un mal de las reuniones de todo tipo es la deriva. La RTF debe seguir un plan de trabajo concreto.
- **Limitar el debate y las impugnaciones:** cuando un revisor ponga de manifiesto un criterio, podrá no haber unanimidad sobre su impacto, pero deben evitarse los desacuerdos.
- **Enunciar áreas del problema, no intentar resolverlo:** a menudo la resolución de los problemas puede ser encargada al desarrollador por sí sólo o con la ayuda de otra persona.

Capítulo 1. Fundamentación teórica

- **Tomar notas escritas:** a veces es buena idea que el registrador tome las notas en una pizarra de forma que las declaraciones y la asignación de prioridades puedan ser comprobadas por el resto de los revisores, a medida que se va registrando la información.
- **Limitar el número de participantes, e insistir en la preparación anticipada:** se ha de mantener el número de participantes en el mínimo necesario.
- **Desarrollar una lista de comprobación para cada producto a ser revisado:** se han de desarrollar listas de chequeos para cada producto o artefacto a revisar, y de esta forma se organizará mejor el proceso de revisión.
- **Disponer de recursos y una agenda para la RTF (incluir como tarea del proyecto):** para que la RTF sean efectivas, se deben planificar como una tarea del proceso de la ingeniería del *software*. Además se debe trazar un plan de acción para las modificaciones inevitables que aparecen como resultado de una RTF, lo cual forma parte de la estrategia de gestión de la configuración.
- **Realizar un entrenamiento por parte de los revisores:** el entrenamiento se debe basar en los aspectos relacionados con el proceso, así como en las consideraciones de psicología humana que atañen a la revisión.
- **Repasar las revisiones anteriores:** el primer producto que se haya revisado puede establecer las propias directivas de revisión.

1.2 Análisis de procedimientos de revisiones de *software*

A continuación se describe el IEEE estándar 1028:1997 Revisiones del *software*. Este define los tipos de revisiones: revisión de gestión y técnica, inspección, recorrido y auditoría. En el estándar se explica detalladamente, de cada una de las revisiones, cuáles son los posibles productos de *software* a examinar, los responsables de llevarlas a cabo, o sea, los roles, los criterios y artefactos de entrada y salida. En general, entre los roles que definen se encuentra el jefe de decisión, quien decide la realización de la revisión, el líder del equipo de revisores, el escriba que se encarga de la documentación generada, el lector que presenta la revisión, el autor del producto objeto de revisión, el revisor y el técnico que provee la información necesaria para los revisores o participa activamente, de acuerdo al tipo de revisión. Cuando definen los artefactos de entrada de cada una de las revisiones la mayoría coincide en: una declaración de objetivos para la revisión, el producto de *software* a ser revisado, el nivel actual de defectos o la lista de resultados, documentos de procedimiento de la revisión, cualquier reglamento, norma, directriz, plan o

Capítulo 1. Fundamentación teórica

procedimiento bajo el cual el producto de *software* va a ser revisado y la categoría de los defectos. El estándar define que para que una revisión pueda ser realizada tienen que haber sido cumplidos los criterios de entrada, o sea, las precondiciones y la autorización de la revisión. Esta establece que las revisiones serán planificadas en los documentos de planificación del proyecto, mientras que la primera señala que se debe complementar con antelación, el establecimiento de una declaración de objetivos para la revisión y que las entradas requeridas para la misma estén disponibles (4).

1.2.1 Propuesta de la Universidad del Valle-Cochabamba

La Universidad del Valle –Cochabamba ubicada en Bolivia desarrolló una propuesta que permite mejorar el proceso de desarrollo del *software*. Esta consta de tres etapas que se describen a continuación (12):

Planificación

Se define la información requerida para llevar a cabo todo el proceso, antes de comenzar la etapa, el jefe de aseguramiento de calidad o coordinador, conjuntamente con el desarrollador, verifica si el artefacto está correctamente terminado y con la información necesaria para realizar la inspección:

1. Se selecciona a los inspectores, con su respectivo rol (cargo que desempeñará) y perfil (línea de inspección).
2. Se debe describir el proyecto el cual utilizará las inspecciones.
3. Debe describirse la inspección tomando en cuenta la siguiente información: nombre del artefacto, etapa que se encuentra el proyecto, estándares a utilizar, si se llevará a cabo sincrónicamente o asincrónicamente, fecha de inicialización, finalización y propósito de la inspección. Por último las reglas de partida, las cuales servirán para la preparación de los inspectores.
4. Se definen todos los documentos de apoyo, los cuales servirán para la inspección.
5. Se seleccionan las listas de comprobación, de acuerdo al propósito, el tipo de *software*, la etapa de desarrollo y el tipo de artefactos.
6. Deben definirse los pesos correspondientes a cada pregunta y objetivo respecto a los objetivos de la inspección.

Posteriormente se pasa a la preparación de los inspectores donde revisan los distintos documentos planificados para la inspección y realizan una reunión rápida con el propósito fundamental de conocer los objetivos de la inspección y cómo se va a desarrollar la misma.

Capítulo 1. Fundamentación teórica

Verificación

En esta etapa se emplean las listas de verificación planificadas en la etapa de planificación, sirviendo a los inspectores como una guía y recurso para los detalles del artefacto en inspección. Los inspectores al efectuar la lectura a cada una de las preguntas de las listas de comprobación y verificando la conformidad de cada una de estas, efectúa una valoración de acuerdo a su preparación, experiencia y visión, para luego realizar un resumen de los defectos que, a su parecer, se encuentran en el artefacto. De esta forma, al culminar con la verificación se tiene los posibles defectos que servirán como partida para la reunión de registro.

La información generada por el grupo de inspección es la siguiente:

1. Se obtienen listas de comprobación correctamente verificadas y una lista de defectos encontrados en el artefacto.
2. Se registra hora y día de inicio de la verificación.
3. Se obtiene el número de defectos y observaciones. Al finalizar la verificación, se realiza un conteo de los defectos encontrados.
4. Se registra el tiempo empleado en la verificación.
5. Se obtiene la evaluación de cada una de las listas de comprobación.

Resultado y conclusiones

En la reunión de registro el jefe de inspección y el asesor realizan un resumen de los defectos encontrados, para luego en conjunto con el coordinador discutir los defectos y clasificarlos por gravedad. Pasada la reunión de registro, se obtiene un resumen de defectos que es considerado por el jefe de inspección, el cual junto al asesor discuten si es necesaria la participación de personas con experiencias o formadas profesionalmente para reconstruir el artefacto y corregir los defectos. De acuerdo con las observaciones por cada defecto, se planifica un tiempo de reconstrucción y la asignación de las personas que participarán en la reconstrucción. Esta etapa forma parte del modelo de inspección propuesto; es la etapa final. La inspección de un artefacto finaliza cuando los defectos encontrados fueron corregidos por el desarrollador. El jefe de aseguramiento de calidad tiene la labor de analizar las métricas para establecer mejoras en el proceso de inspección.

Capítulo 1. Fundamentación teórica

1.2.2 Propuesta de la Universidad Nacional Experimental “Rómulo Gallegos”

La Facultad de Ciencias Exactas Físicas y Naturales de la Universidad Nacional Experimental de los Llanos Centrales “Rómulo Gallegos” ubicada en Venezuela presenta una propuesta de inspecciones de *software*, con el objetivo de detectar defectos sin examinar por qué surgieron. Esta consta de seis etapas que se describen a continuación (13):

Planificación

En esta etapa se realiza la asignación de roles a los inspectores. Las copias del producto y el material relacionado son distribuidas, en esta oportunidad se planifica la reunión de inspección.

Conferencia Introductoria

Se realiza una preparación general del proyecto en desarrollo a los inspectores que no están familiarizados con el mismo.

Preparación

Los inspectores se instruyen individualmente para la reunión de inspección, estudiando el material aportado en la planificación.

Reunión

Los inspectores se reúnen para examinar el producto de *software*. El lector presenta el trabajo mientras los demás investigan el producto en la búsqueda de defectos, los cuales luego son clasificados y registrados. Al final de la reunión los inspectores deciden si el trabajo es aceptado, re trabajo bajo la supervisión del moderador o re trabajado por una próxima inspección.

De la inspección surgen dos resultados: una lista de defectos encontrados y un reporte para la gerencia, indicando el producto inspeccionado, equipo de inspección, número de defectos encontrados y su gravedad.

Re trabajo

El autor del trabajo corrige los defectos detectados en la reunión de inspección.

Seguimiento

En esta fase las correcciones realizadas por el autor son chequeadas por el moderador, si este queda satisfecho la inspección queda oficialmente concluida.

Capítulo 1. Fundamentación teórica

1.2.3 Propuesta del Instituto Superior Politécnico José Antonio Echeverría

El Instituto Superior Politécnico José Antonio Echeverría de la Ciudad de la Habana desarrolló una propuesta de introducción de las revisiones en el proceso desarrollo de *software* para la Pequeñas y Medianas Empresas Productoras de *Software* en Cuba (PYME). Esta consta de tres fases (7):

Preparación de las Revisiones

En esta etapa la empresa deberá trazar pautas para el desarrollo de las revisiones, los roles que se ejecutarán, así como las políticas generales de aseguramiento de calidad. Las actividades propuestas son las siguientes:

1. Definir un administrador de aseguramiento de la calidad, encargado de establecer las políticas y velar por la calidad de los productos a nivel de empresa.
2. Definir roles adecuados para el grupo de desarrollo de cada proyecto de *software*.
3. Definir roles para las revisiones, que dependerán del tipo de revisión que se desarrolle.
4. Establecer los documentos para registrar el resultado de las revisiones y los involucrados dentro de la organización que recibirán estos informes.
5. Definir las herramientas para la ejecución de las revisiones.
6. Adiestrar al personal en el uso de las herramientas y procedimientos establecidos.

Gestión de productos terminados

El objetivo fundamental de este proceso es contar con la información del estado y calidad de los productos desarrollados por la empresa antes de introducir el modelo, de forma que la empresa pueda trazar acciones preventivas, para ello se ejecutan las actividades siguientes:

1. Registrar defectos reportados por los clientes, para el caso de productos en fase de explotación u operación.
2. Registrar defectos reportados por el personal que implanta el sistema, para el caso de los productos en fase de implantación.
3. Llevar métricas para ambos tipos de productos y trazar acciones preventivas.

Gestión de nuevos productos y de productos en desarrollo

La organización debe definir un conjunto de políticas con respecto a las revisiones, que deberán ser seguidas por desarrolladores e inspectores a lo largo del proceso de desarrollo de *software* y durante la ejecución de las revisiones. Se definirán los momentos, tipos de revisión y listas de chequeo o listas de

Capítulo 1. Fundamentación teórica

comprobación generales para cada tipo de proyecto, que son aquellas listas cuyos elementos consisten en parámetros o preguntas a considerar en la revisión y que pueden ser defectos que están presentes en el proyecto. Además es importante que no deje de declararse explícitamente con qué frecuencia se revisará la línea base del proyecto que pasará a formar parte del repositorio de elementos de configuración, como una versión estable y revisada de los elementos de configuración correspondientes.

1.2.4 Propuesta del Centro de Estudios de Ingeniería de Sistema

El Centro de Estudios de Ingeniería de Sistema (CEIS) de la Ciudad de la Habana desarrolló una propuesta que permite planificar, controlar y dar seguimiento a las revisiones de los proyectos de *software*. Esta consta de seis etapas, a continuación se describen cada una de ellas (14):

Planificación

El moderador designado para este proceso deberá establecer la conducta y progreso de la misma para lo cual llevará a cabo las siguientes tareas:

1. Identificación del equipo, asegurándose de que sus miembros estén disponibles para preparar adecuadamente la revisión.
2. Verificación de que los materiales necesarios estén disponibles y de acuerdo con los estándares.
3. Verificación de que están establecidos los criterios de entrada para la revisión en forma de lista de chequeo.
4. Programar tiempo y lugar de la reunión de inspección asegurando que el lugar esté reservado para tales fines.
5. Asignar la función de lector a un miembro seleccionado del equipo de revisión.
6. Citar a todos los miembros del equipo y otras partes interesadas para la reunión.

Revisión

El moderador programará un contacto con los miembros del equipo de revisión y el autor del producto a revisar, donde este último dará una breve descripción del *software*, qué tareas se han realizado, cómo se realizará cada tarea, una descripción de las interfaces y otros aspectos que pudieran resultar de interés y que serán acordados previamente entre el autor y el moderador. Si existiese algún problema significativo que viole los criterios de entrada establecidos, la revisión puede ser pospuesta hasta que el problema sea solucionado. Posteriormente el moderador citará a todos para la reunión de Inspección y entregará los materiales a los miembros del equipo de revisión.

Capítulo 1. Fundamentación teórica

Preparación

Los revisores revisan el producto y documentan cualquier discrepancia o defecto encontrado. Deben mediar como mínimo cinco días, por supuesto, este tiempo o fase de preparación dependerá de la complejidad del producto a revisar y será determinado por el moderador. Durante esta fase los revisores registran los defectos encontrados en un documento para tales efectos y será suministrado, conjuntamente con los materiales del producto a revisar, por el moderador. El lector debe registrar cualquier dificultad en el entendimiento de los materiales facilitados a los revisores.

Reunión de inspección

El moderador es responsable de conducir adecuadamente y asegurar que los miembros del equipo trabajen con profesionalidad, encontrando defectos en el producto que contribuyan a mejorar su calidad. El lector deberá mostrar el producto de forma lógica para que se vayan exponiendo las discrepancias encontradas. Los defectos son identificados, discutidos y registrados, atención particular merecen aquellos que no han sido detectados con anterioridad y que surgen como consecuencia de la discusión de los miembros del equipo. El objetivo de la reunión es encontrar defectos, no soluciones, esto último es responsabilidad del autor. Como resultado de la reunión se deberá elaborar un documento que contenga los defectos encontrados y que será entregado al autor y grupo de aseguramiento de la calidad de *software*.

Reproceso

El autor comienza a corregir los defectos y cuando estén listos lo informará al moderador.

Verificación

La corrección de los defectos es verificada. Si el moderador entiende que se requiere una re inspección, lo informará al autor y este comenzará a preparar el material necesario. Si es necesario el moderador puede designar a un miembro del equipo para que lo asista en la revisión de los defectos y una vez verificado, cada uno debe ser registrado en el documento correspondiente.

1.2.5 Propuesta de la Universidad de la Ciencias Informáticas

La Universidad de las Ciencias Informáticas (UCI) comenzó el programa de mejora, donde se implantan las áreas de proceso del nivel 2 de CMMI, y específicamente de PPQA (*Process and Product Quality Assurance*). El área de proceso encargada de verificar la adherencia de los procesos y productos de

Capítulo 1. Fundamentación teórica

trabajo y garantizar la información y resolución de las no conformidades ha redefinido el proceso de revisiones que se estaba siguiendo hasta el momento; estas revisiones ahora contarían con 4 etapas, a continuación se describen cada una de ellas (15):

Primera etapa: Planear las revisiones

Durante la planeación de las revisiones se elabora un plan de revisiones, se seleccionan los procesos y productos a evaluar, los revisores y se envían las notificaciones necesarias. Además se verifica con tiempo suficiente de antelación la viabilidad de la revisión.

Salidas: plan de revisiones y tres notificaciones (notificación de revisión, de revisores y de personal).

Segunda etapa: Ejecución de la revisión

La etapa de ejecución de la revisión corresponde a las actividades en el sitio del proyecto. Debe comenzar con la reunión de apertura, seguida de la revisión documental y terminar con la reunión de cierre, informe final de la revisión. Igualmente en esta etapa se realiza la evaluación de desempeño de los revisores.

Salidas: minutas de reunión de apertura y cierre, informe final de la revisión y evaluaciones de desempeño de los revisores.

Tercera etapa: Seguimiento de las no conformidades

La etapa de seguimiento de las no conformidades comprende desde la definición del plan de acción para la resolución de los defectos hasta la definitiva resolución de los mismos. El proceso está concebido de forma tal que cuando un defecto no se puede resolver en el ámbito del proyecto se eleve por distintos niveles de escalamiento hasta su resolución definitiva o hasta la emisión de un permiso de no conformidad.

Salidas: plan de acción para la resolución de las no conformidades, registros de escalamiento, solicitudes de permisos y registros de permisos.

Cuarta etapa: Análisis de resultados

El análisis de resultados es una tarea continua que se debe realizar con los resultados de las revisiones que se han llevado a cabo anteriormente. Este análisis comprende el estudio de las tendencias, el estado de las no conformidades y acciones correctivas a nivel organizacional, el registro de lecciones aprendidas y la toma de decisiones.

Salidas: solicitudes de mejora a procesos y productos, reporte de la calidad.

Capítulo 1. Fundamentación teórica

El estándar IEEE tiene en cuenta elementos importantes para la realización de una revisión y con su estudio se sentaron las bases para la elaboración del procedimiento que se propone en este trabajo. No existe una herramienta para la RTF que ayude a verificar que la documentación de cada proyecto en las distintas etapas esté técnicamente correcta, por lo que se analizan varias propuestas de revisiones de *software* de algunas instituciones para contribuir al desarrollo de la investigación. Este análisis permitió conocer que las revisiones de *software* sólo verifican que la documentación se encuentra y no si está correcta, ayudó a mejorar la organización de las ideas en cuanto a la realización y las fases que se utilizan en el desarrollo de las revisiones que se le ejecutan al *software*.

1.3 Conclusiones

El estudio realizado a los distintos conceptos relacionados con la RTF y algunas propuestas de revisiones de *software*, permitieron conocer que:

- ✓ Aunque existen diferentes procedimientos de revisiones de *software*, no se cuenta con una propuesta para realizar la RTF.
- ✓ La RTF profundiza en la revisión que se realiza al *software* porque permite verificar que este se encuentre técnicamente correcto.
- ✓ Se deben tener en cuenta elementos fundamentales como: fases, actividades, artefactos y roles para la elaboración de un procedimiento.
- ✓ La RTF debe ser realizada por especialistas en las distintas etapas de Ingeniería de *Software*, para garantizar que este se desarrolle con la calidad técnica requerida.

Capítulo 2. Elaboración del procedimiento

En este capítulo se describe la propuesta de solución, definiendo el objetivo y el alcance que tiene para los proyectos del centro FORTES de la facultad 4. Además, se detallan los artefactos, roles involucrados y sus responsabilidades permitiendo un mayor entendimiento del procedimiento que guiará a la RTF.

En la investigación se define el procedimiento como la descripción detallada de ciertos pasos a seguir para desarrollar una tarea. Este es guiado por actividades que son ejecutadas por los diferentes roles, generando en cada una de ellas los artefactos correspondientes.

2.1 Objetivo y alcance del procedimiento

El procedimiento definido tiene como objetivo detectar defectos y hacer recomendaciones en las etapas de arquitectura, diseño, requerimiento y modelo de datos, lo que posibilitará una mejor organización y control de la calidad técnica en los proyectos a revisar en el GCF de la facultad 4. Esto permitirá guiar mediante pasos lógicos, la RTF que realizarán los revisores técnicos, en las distintas etapas de desarrollo del *software*.

2.2 Descripción del procedimiento

A continuación se describen las fases que se llevarán a cabo para la confección del procedimiento: planificación de la revisión, ejecución de la revisión y análisis de los resultados (Ver figura 3). Las fases del procedimiento, son las etapas en las cuales se realizan un conjunto de actividades generando artefactos de entrada y salida.

1. Planificación de la revisión: su objetivo principal es darle cumplimiento al cronograma y acuerdos planteados en la reunión inicio. En esta fase la preparación al equipo revisor constituye una de las tareas más importantes para garantizar una mayor familiarización y un ambiente de trabajo más cómodo al realizar la RTF.
2. Ejecución de la revisión: es la fase donde se realiza la RTF. Esta se centra específicamente, en la etapa de arquitectura, diseño, requerimiento y modelo de datos. Su objetivo es detectar las no conformidades que puedan existir en cuanto a: descripción y captura de requisitos, relaciones entre clases, entregables, interfaz, diagramas entre otros.

Capítulo 2. Elaboración del procedimiento

3. Análisis de los resultados: en esta fase se integran los resultados de las revisiones por etapas, se elabora el registro de no conformidades con los comentarios realizados por los revisores técnicos para dar a conocer el estado del proyecto revisado.



Figura 3: Descripción del procedimiento (Elaboración propia).

2.3 Roles y sus responsabilidades

Los roles del procedimiento se definen según el conocimiento, experiencias y capacidades personales. Desempeñan un conjunto de tareas específicas, durante la ejecución de las actividades en la RTF. A continuación se describen los roles definidos en la presente investigación:

Revisor líder: está presente en todas las fases del procedimiento, es el encargado de convocar la reunión de inicio y cierre de la revisión, asigna el o los revisores que junto a los revisores técnicos realizarán la revisión en cada una de las etapas. Entrega el registro de no conformidades al líder del proyecto revisado.

Líder del proyecto: está presente en la planificación de la revisión y análisis de los resultados, participando directamente en la reunión de inicio y cierre de la revisión. Es el responsable de gestionar la capacitación para el equipo revisor.

Revisor técnico: pueden estar presentes en todas las fases del procedimiento. Son especialistas de arquitectura, diseño, requerimientos y modelo de datos que aplican la lista de chequeo al expediente de proyecto y entregables, por lo que se necesita que estos sean analistas, arquitectos, programadores y diseñadores de base de datos. Participan fundamentalmente en la ejecución de la revisión.

Capítulo 2. Elaboración del procedimiento

Revisor: está presente en todas las fases del procedimiento, es el rol que junto al revisor técnico realiza la revisión por etapas y registra todas las no conformidades. Se encarga de conformar el registro de no conformidades del proyecto revisado.

2.4 Actividades generadas en las fases del procedimiento

Las actividades son las tareas a realizar en cada una de las fases, que permiten realizar el procedimiento desde inicio hasta el fin. Por cada de una de estas actividades se definen artefactos y los roles que garantizarán el cumplimiento de estas tareas. En la figura 5 se relacionan las actividades en cada fase.

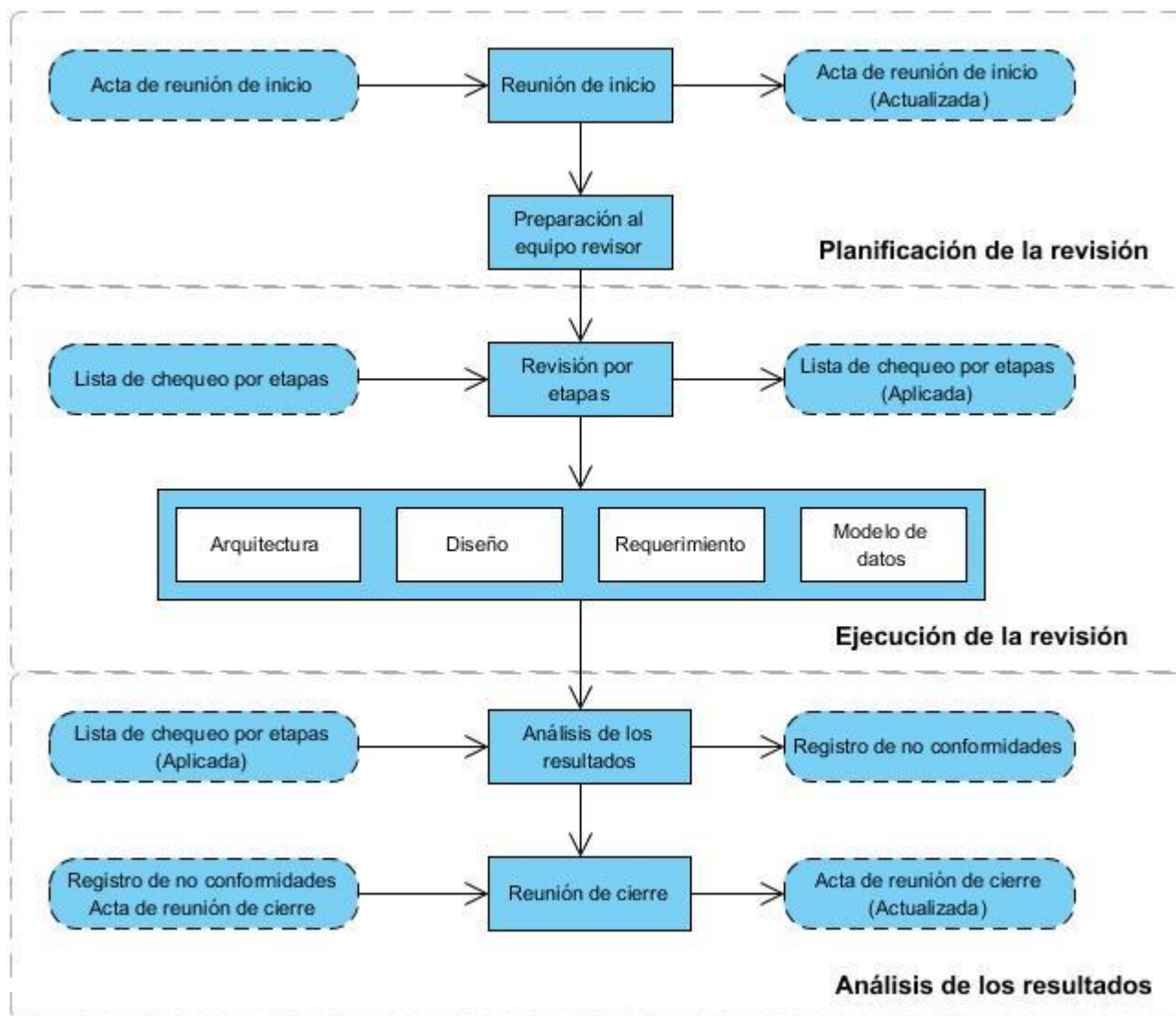


Figura 4: Actividades en cada fase (Elaboración propia).

Capítulo 2. Elaboración del procedimiento

2.4.1 Actividades en la fase de planificación de la revisión

Las actividades correspondientes a la fase de planificación son: reunión de inicio y preparación al equipo revisor.

Reunión de inicio: participan los involucrados en la RTF. En ella se concilia el cronograma conformado tanto por el proyecto a revisar como por el equipo revisor. Se presentan todas las actividades a realizar durante el período de revisión y se actualiza el acta de reunión de inicio.

Roles involucrados:

- ✓ Revisor líder: dirige la reunión y concilia los acuerdos con el proyecto a revisar.
- ✓ Revisor: presenta el cronograma de revisión y actividades por etapa.
- ✓ Líder del proyecto: aprueba el cronograma y la revisión.
- ✓ Revisor técnico (opcional): aprueba las revisiones por etapa, de acuerdo al cronograma.

Artefacto de entrada:

- ✓ Acta de reunión de inicio.

Artefacto de salida:

- ✓ Acta de reunión de inicio (Actualizada).

Preparación al equipo revisor: se realiza una capacitación por parte del equipo de desarrollo con el objetivo de familiarizar al equipo revisor con el negocio o funcionamiento del producto a revisar.

Roles involucrados:

- ✓ Líder del proyecto: gestiona la preparación al equipo revisor.
- ✓ Revisor técnico: recibe la preparación.
- ✓ Revisor: recibe la preparación.

2.4.2 Actividades en la fase de ejecución de la revisión

Las actividades correspondientes a la fase de ejecución son: revisiones a las diferentes etapas del *software*.

Revisiones por etapas: se aplican las listas de chequeo por etapas, se actualizan y se detectan las no conformidades.

Capítulo 2. Elaboración del procedimiento

Roles involucrados:

- ✓ Revisor líder: dirige el procedimiento.
- ✓ Revisor: junto al revisor técnico toma los comentarios que se transforman en no conformidades y recomendaciones.
- ✓ Revisor técnico: ejecuta la revisión.

Artefactos de entrada:

- ✓ Listas de chequeo por etapas.
 - Lista de chequeo de arquitectura.
 - Lista de chequeo de diseño.
 - Lista de chequeo de requerimiento.
 - Lista de chequeo de modelo de datos.

Artefactos de salida:

- ✓ Listas de chequeo por etapas (Aplicadas).
 - Lista de chequeo de arquitectura.
 - Lista de chequeo de diseño.
 - Lista de chequeo de requerimiento.
 - Lista de chequeo de modelo de datos.

2.4.3 Actividades en la fase de análisis de los resultados

Las actividades correspondientes a esta fase son: análisis de los resultados y la reunión de cierre:

Análisis de los resultados: se analiza la lista de chequeo y se elabora el registro de las no conformidades.

Roles involucrados:

- ✓ Revisor líder: dirige el procedimiento y elabora el registro de no conformidades.
- ✓ Revisor: elabora junto al revisor líder el registro de no conformidades.

Capítulo 2. Elaboración del procedimiento

Artefactos de entrada:

- ✓ Listas de chequeo por etapas (Aplicadas).
 - Lista de chequeo de arquitectura.
 - Lista de chequeo de diseño.
 - Lista de chequeo de requerimiento.
 - Lista de chequeo de modelo de datos.

Artefacto de salida:

- ✓ Registro de no conformidades.

Reunión de cierre: participan los involucrados en la revisión, se discute el registro de no conformidades, se informa el estado y la cantidad de no conformidades que tuvo el proyecto en las distintas etapas. Se actualiza el acta de reunión de cierre.

Roles involucrados:

- ✓ Revisor líder: dirige el procedimiento.
- ✓ Revisor: presenta los resultados por etapa.
- ✓ Revisor técnico (opcional): interviene en caso de dudas con las no conformidades.
- ✓ Líder del proyecto: recepciona el registro de no conformidades.

Artefactos de entrada:

- ✓ Registro de no conformidades.
- ✓ Acta de reunión de cierre.

Artefacto de salida:

- ✓ Acta de reunión de cierre (Actualizada).

2.5 Artefactos del procedimiento

Los artefactos contribuyen a que el procedimiento se efectúe de forma organizada y son la constancia de los resultados de la revisión llevada a cabo, se generan en cada fase y se toman como base para futuras revisiones. Los que se definen en el procedimiento de RTF son:

- ✓ Acta de reunión inicio.
- ✓ Listas de chequeo por etapas.

Capítulo 2. Elaboración del procedimiento

- Lista de chequeo de arquitectura.
 - Lista de chequeo de diseño.
 - Lista de chequeo de requerimiento.
 - Lista de chequeo de modelo de datos.
- ✓ Registro de no conformidades.
- ✓ Acta de reunión de cierre.

2.5.1 Acta de reunión de inicio

El acta de reunión de inicio es el documento que contiene los elementos a tratar en la reunión (Ver figura 6). En este artefacto se recogen los siguientes puntos:

- ✓ Lugar, fecha, hora: donde se realizará la reunión.
- ✓ Asistentes y ausentes: personal involucrado con la revisión.
- ✓ Orden del día: son los puntos que guiarán la reunión.
- ✓ Conciliación del cronograma: la realizan los involucrados en la revisión, concilian la fecha, hora, lugar y responsable de todas las actividades que se planifican en el cronograma.
- ✓ Acuerdos: son los planteamientos que se recogen con el objetivo de facilitar la revisión.
- ✓ Consideraciones generales: son asuntos de carácter general.
- ✓ Observaciones: son temas que en la reunión se requieren esclarecer.



Reunión de inicio

Autor		Fecha	
Lugar		Hora Inicio	
Proyecto/Grupo		Hora Terminación	
Asunto			
Asistentes			
Ausentes			

Orden del día

1. Conciliación del cronograma
2. Acuerdos
3. Consideraciones generales

Cronograma

Actividad	Fecha	Hora	Lugar	Responsable

Acuerdos tomados

No.	Acuerdo
1	
2	

Consideraciones generales:

Observaciones:

Figura 5: Acta de reunión inicio (Elaboración propia).

Capítulo 2. Elaboración del procedimiento

Este artefacto transita por dos estados, el primero que es cuando se presenta a los participantes en la reunión y sale de esta actividad como reunión de inicio (Actualizada), en este estado ya fue aprobado el cronograma y se aprobaron los acuerdos asignando responsables a cada uno.

2.5.2 Listas de chequeo por etapas

Las listas de chequeo por etapas son los artefactos principales del procedimiento, con la ejecución de las mismas se podrá revisar no solo que la documentación esté, sino que especialistas de las diferentes etapas de desarrollo constatarán que las mismas se hayan ejecutado con la calidad requerida.

Las listas de chequeo por etapas son cuatros, una de arquitectura, diseño, requerimiento y modelo de datos. Cada una de estas listas cuentan con los elementos siguientes.

- ✓ Introducción: se realiza una breve sinopsis de lo que recoge el informe de evaluación.
- ✓ Propósito y objetivos: se enuncian los objetivos y el propósito que se persigue con la realización del informe.
- ✓ Alcance: proyecto al que se involucra.
- ✓ Resumen: se realiza un pequeño resumen de lo que se enunciará en la plantilla

Estas listas de chequeo por etapas también incluyen una leyenda que no es más que una serie de elementos que permitirá identificar y clasificar los que serán revisados en los proyectos, ver figura 7.

Leyenda:

Código: está compuesto por cuatro dígitos, el primer par de números es el de la etapa y el segundo par del número del indicador.

01(Etapa de arquitectura)

02(Etapa de diseño)

03(Etapa de requerimiento)

04(Etapa de modelo de datos)

Ejemplo: 0101(los número que están de color rojo pertenece al código de la etapa de arquitectura y los restantes al número del indicador)

Indicadores a evaluar: son las actividades que deben cumplirse para que la etapa a revisar este con todos los elementos que se necesitan en el software.

Respuesta: si aplica se responde **Si** o **No** según los indicadores a evaluar y si no aplica **NP**.

Evaluación: se pone alta, media o baja según el cumplimiento del indicador.

Alta (A): se cumple el indicador a evaluar en la etapa a revisar.

Media (M): le faltan algunos detalles para cumplirse el indicador a evaluar.

Baja (B): no se cumple el indicador a evaluar.

Comentario: se explica cuando se de media o baja.

Figura 6: Leyenda de las listas de chequeo (Elaboración propia).

Capítulo 2. Elaboración del procedimiento

Las listas de chequeo por etapas están presente en varias actividades, comienzan en la ejecución de la revisión donde los revisores técnicos la aplican por cada etapa, y salen de estas actividades como listas de chequeo por etapas (Aplicadas), estas entran a la fase de análisis de los resultados donde con los comentarios realizados por los revisores se conforma el próximo artefacto.

A continuación se relacionan los indicadores a evaluar en cada una de las listas de chequeo por etapas.

Lista de chequeo de arquitectura:

Tabla 1: Lista de chequeo de arquitectura (Elaboración propia).

Código	Indicadores a evaluar
0101	1. ¿Existe el documento "Guía base"? <i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la arquitectura que se maneja en el proyecto]</i>
0102	1.1. ¿Se especifica en el documento cada una de las vistas? <i>[Debe explicarse cada una de las vistas y en caso de alguna no aplicar debe especificarse el por qué]</i>
0103	1.2. ¿Se especifica en el documento la Estrategia para Pruebas de Concepto? <i>[Debe explicarse la estrategia para las pruebas de concepto verificando los puntos siguientes]</i>
0104	1.2.1. ¿Se especifican y seleccionan las técnicas para pruebas de concepto adecuadas? <i>[Verificar de acuerdo al tipo de proyecto si usa la técnica adecuada, el especialista puede recomendar una técnica más óptima, en caso necesario]</i>
0105	1.2.2. ¿Se especifican y seleccionan las herramientas para las pruebas de estrés? <i>[Verificar que de acuerdo al tipo de proyecto es suficiente con las pruebas de estrés seleccionadas, puede recomendar otras]</i>
0106	1.2.3. ¿Se encuentra evaluado por criterios el modelo teórico de pruebas de concepto por cada vista? <i>[Verificar que se evalúe cada uno de los criterios en cada vista, de acuerdo al tipo de proyecto]</i>
0107	2. El documento " Vista de procesos". <i>[No aplica para FORTES]</i>

Capítulo 2. Elaboración del procedimiento

0108	<p>3. ¿Existe el documento " Vista de Sistema"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la vista de sistema, a continuación elementos que no deben faltar en esta vista]</i></p>
0109	<p>3.1. ¿Se especifican los requisitos priorizados del sistema?</p> <p><i>[Verificar que se listen los requisitos priorizados y que se siga el procedimiento correcto, explicado en el documento]</i></p>
0110	<p>3.2. ¿Se encuentra el agrupamiento y priorización de los requisitos siguiendo cada una de las acciones?</p> <p><i>[Verificar que se sigan todas las acciones explicadas en el documento]</i></p>
0111	<p>3.3. ¿Se identifican los paquetes y componentes principales siguiendo las reglas establecidas en el documento?</p> <p><i>[Verificar que se sigan todas las reglas explicadas en el documento]</i></p>
0112	<p>3.4. ¿Se representa la dependencia entre paquetes?</p> <p><i>[Verificar que se represente a través de diagrama todas las dependencias, y que estas estén correctas]</i></p>
0113	<p>3.5. ¿Se determinan los escenarios de la vista?</p> <p><i>[Verificar que se determinen todos los escenarios siguiendo las acciones descritas en el documento]</i></p>
0114	<p>4. ¿Existe el documento " Vista de presentación"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la vista de presentación, a continuación elementos que no deben faltar en esta vista]</i></p>
0115	<p>4.1. ¿Se definen en el documento las pantallas?</p> <p><i>[Verificar que se haga una correcta definición de las pantallas en correspondencia con el tipo de proyecto]</i></p>
0116	<p>4.2. ¿Se definen en el documento contenido, pantallas genéricas de subsistemas o módulos?</p> <p><i>[Verificar que se haga una correcta definición de los contenidos y pantallas en correspondencia con el tipo de proyecto]</i></p>
0117	<p>4.3. ¿Se definen las pantallas de áreas de trabajo de cada módulo?</p> <p><i>[Verificar que se haga una correcta definición de las pantalla por área de trabajo correspondencia con el tipo de proyecto]</i></p>

Capítulo 2. Elaboración del procedimiento

0118	<p>4.4. ¿Se describen los escenarios de la vista de presentación que se aplican a su solución?</p> <p><i>[Verificar que se describan los escenarios que se aplican a la solución de acuerdo al tipo de proyecto]</i></p>
0119	<p>5. ¿Existe el documento " Vista de Datos"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la Vista de Datos, a continuación elementos que no deben faltar en esta vista]</i></p>
0120	<p>5.1. ¿Se encuentra detallada la caracterización del producto según la vista?</p> <p><i>[Verificar que se identifiquen correctamente los diferentes tipos de datos que se quieren almacenar]</i></p>
0121	<p>5.2. ¿Se encuentran descritos los escenarios y patrones de solución?</p> <p><i>[Verificar que haya una correcta descripción de los escenarios y patrones de descripción en el documento]</i></p>
0122	<p>5.3. ¿Se encuentra en el diccionario de datos los elementos necesarios para identificar los conceptos para la modelación de la base de datos?</p> <p><i>[Verificar que los término se encuentren con la definición correcta y en caso de referenciar otro documento, verificar que en el mismo se encuentre correctamente]</i></p>
0123	<p>5.4. ¿Se encuentran las especificaciones del diseño físico de la solución en el Modelo de datos?</p> <p><i>[Verificar que se encuentren las especificaciones del diseño físico en el modelo de datos, en caso de referenciar algún documento, verificar que el mismo esté correcto]</i></p>
0124	<p>6. ¿Existe el documento " Vista de integración "?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la Vista de Integración, a continuación elementos que no deben faltar en esta vista]</i></p>
0125	<p>6.1. ¿Se seleccionan los estilos arquitectónicos a emplear?</p> <p><i>[Verificar que se seleccionen los estilos y patrones que utilizará, siguiendo las reglas brindadas en el documento]</i></p>
0126	<p>6.2. ¿Se selecciona la estrategia de diseño y desarrollo del proyecto?</p> <p><i>[Verificar que se seleccione la estrategia de diseño adecuada para el tipo de proyecto]</i></p>
0127	<p>6.3. ¿Se seleccionan los patrones de diseño, principales vista desde la integración entre las soluciones?</p>

Capítulo 2. Elaboración del procedimiento

	<i>[Verificar que se seleccionen los patrones de diseño adecuados para el tipo de proyecto]</i>
0128	6.4. ¿Se seleccionan estándares de codificación? <i>[Verificar que se seleccionen los estándares de codificación adecuados para el tipo de proyecto]</i>
0129	7. ¿Existe el documento " Vista de Entornos de desarrollo tecnológico"? <i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a los entornos de desarrollo tecnológico, a continuación elementos que no deben faltar en esta vista]</i>
0130	7.1. ¿Se selecciona la tecnología a utilizar según el Sistema Operativo? <i>[Verificar que se especifique por cada uno de los escenarios de despliegue el Sistema Operativo]</i>
0131	7.2. ¿Se selecciona la tecnología a utilizar por el Gestor de Base de Datos? <i>[Verificar que se especifique por cada uno de los escenarios el Gestor de Base de Datos, en caso de no seleccionar ninguna verificar que la tabla esté llena con la especificación de la que se va a utilizar]</i>
0132	7.3. ¿Se selecciona la tecnología ORM a utilizar? <i>[Verificar que se especifique la tecnología ORM adecuada para el tipo de proyecto, en caso de no seleccionar ninguna verificar que la tabla esté llena con la especificación de la que se va a utilizar]</i>
0133	7.4. ¿Se selecciona la tecnología del lenguaje de programación a utilizar? <i>[Verificar que se haga un correcto uso de los lenguaje de programación a Nivel de negocio y a Bajo nivel]</i>
0134	7.5. ¿Se selecciona la tecnología del marco de trabajo? <i>[Verificar que se especifique la selección del marco de trabajo adecuado para el desarrollo e interfaz usuario, en caso de seleccionar ninguna verificar que la tabla esté llena con la especificación de la que se va a utilizar]</i>
0135	8. ¿Existe el documento " Vista de Seguridad"? <i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la seguridad, a continuación elementos que no deben faltar en esta vista]</i>
0136	8.1. ¿Se identifican los requisitos de seguridad del producto? <i>[Verificar que se especifiquen en las preguntas mínimas que deben quedar respondidas y que deben permitir identificar requisitos de seguridad]</i>

Capítulo 2. Elaboración del procedimiento

0137	<p>8.2. ¿Se identifica la autenticación y la capa de presentación?</p> <p><i>[Verificar que se evalúe asignándole la puntuación máxima a los elementos que debe utilizar para el diseño y desarrollo de la aplicación]</i></p>
0138	<p>8.3. ¿Se especifica el nivel de la capa de negocio?</p> <p><i>[Verificar que se evalúe el nivel de la capa de negocio asignándole la puntuación máxima a los elementos que debe utilizar en el diseño y desarrollo de la aplicación]</i></p>
0139	<p>8.4. ¿Se especifica el nivel de la capa de datos?</p> <p><i>[Verificar que se evalúe asignándole la puntuación máxima a los elementos que debe utilizar para el diseño y desarrollo de la aplicación]</i></p>
0140	<p>8.5. ¿Se especifica la seguridad del servidor de la aplicación?</p> <p><i>[Verificar que se identifique el estado de seguridad de la aplicación vista desde el entorno]</i></p>
0141	<p>8.6. ¿Se especifica la tecnología para el sellado del código?</p> <p><i>[Verificar que se especifique la tecnología que utilizará para la protección del código]</i></p>
0142	<p>9. ¿Existe el documento " Vista de Infraestructura"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la Vista de Infraestructura, a continuación elementos que no deben faltar en esta vista]</i></p>
0143	<p>9.1. ¿Los elementos de los diagramas de los pasos 1, 2 y 3 son tratados para la descripción en el paso 4 de escenarios y patrones de solución?</p> <p><i>[Verificar que los elementos de los diagramas de los pasos 1, 2 y 3 sean tratados para la descripción en el paso 4 de escenarios y patrones de solución]</i></p>
0144	<p>9.2. ¿Se especifican los escenarios de infraestructura?</p> <p><i>[Verificar que a partir de los análisis en los pasos anteriores identifique los escenarios y los patrones de solución en los mismos]</i></p>
0145	<p>10. ¿Existe el documento "Vista de Despliegue"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la Vista de Despliegue, a continuación elementos que no deben faltar en esta vista]</i></p>
0146	<p>10.1. ¿Se especifica la solución vista desde las licencias de los componentes que la forman?</p> <p><i>[Verificar que se seleccione que tipos de licencias soportan los componentes de la solución]</i></p>
0147	<p>10.2. ¿Se especifican los requerimientos mínimos de instalación?</p> <p><i>[Verificar que se seleccionen los requerimientos mínimos para la instalación]</i></p>

Capítulo 2. Elaboración del procedimiento

0101	11. ¿Existe el documento "Guía base"? <i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la arquitectura que se maneja en el proyecto]</i>
------	--

Lista de chequeo de diseño:

Tabla 2: Lista de chequeo de diseño (Elaboración propia).

Código	Indicadores a evaluar
0201	1. ¿Existe el documento "Modelo de diseño"? <i>[Este documento se encuentra en el expediente de proyecto en Arquitectura y diseño. En este debe estar definido todo lo referente al Modelo de Diseño, a continuación elementos que no deben faltar en este modelo]</i>
0202	1.1. ¿Se encuentra el diagrama de paquetes? <i>[Verificar que el sistema esté organizado en pequeños paquetes y subsistemas]</i>
0203	1.2. ¿Se encuentra el diagrama de clases? <i>[Verificar que estén las interacciones entre ellas]</i>
0204	1.3. ¿Se realiza la descripción de cada una de las clases que intervienen en el diagrama de clases? <i>[Verificar que permitan determinar la capa a la que pertenecen, módulo, número de la clase, nombre de la clase, propósito y descripción]</i>
0205	2. ¿El tipo de aplicación es web? <i>[Verificar que en el documento Modelo de diseño, se encuentren los siguientes elementos]</i>
0206	2.1. ¿Se identifican los tres estereotipos para identificar las principales clases? <i>[Verificar que los estereotipos sean Server page, Client page y Form]</i>
0207	2.2. ¿Se encuentran los formularios relacionados con clase cliente? <i>[Verificar que todos los formularios estén relacionados con la clase cliente]</i>
0208	2.3. ¿Se encuentra una clase cliente que esté relacionada con una clase servidor? <i>[Verificar que al menos una clase cliente esté relacionada con una clase servidor y la relación sea de composición]</i>
0209	2.4. ¿Se encuentra la clase formulario relacionada con la clase controladora? <i>[Verificar que la relación de la clase formulario con la clase controladora es << submit>>]</i>
0210	2.5. ¿Se encuentra la clase servidor relacionada con la clase cliente?

Capítulo 2. Elaboración del procedimiento

	<i>[Verificar que la relación de la clase servidor con la clase cliente es <<build>>]</i>
0211	2.6. ¿Se encuentra la relación de clase cliente a cliente? <i>[Verificar que la relación de clase cliente a cliente es <<link>>]</i>
0212	2.7. ¿Se encuentran las sintaxis de las clases? <i>[Verificar que estén correctas todas las sintaxis de las clases]</i>
0213	2.8. ¿Se encuentren las sintaxis de las funcionalidades? <i>[Verificar que estén correctas todas las sintaxis de las funcionalidades]</i>
0214	2.9. ¿Se encuentran las relaciones entre clases? <i>[Verificar que todas las relaciones entre clases estén correctas]</i>
0215	2.10. ¿Se encuentra la distribución de las clases? <i>[Verificar que la distribución de las clases sea de acuerdo al framework o la librería que utilice]</i>
0216	2.11. ¿Existe la clase u ORM que permita la conexión a los datos? <i>[Verificar que la clase u ORM permita la conexión a los datos]</i>

Lista de chequeo de requerimiento:

Tabla 3: Lista de chequeo de requerimiento (Elaboración propia).

Código	Indicadores a evaluar
0301	1. ¿Existe el documento "Especificación de Requisitos"? <i>[Este documento se encuentra en el expediente de proyecto en Requisitos. En este debe estar definido todo lo referente a los requisitos, a continuación elementos que no deben faltar]</i>
0302	1.1. ¿Se encuentran los paquetes de análisis? <i>[Verificar que esté correctamente el siguiente elemento]</i>
0303	1.1.1. ¿Se describen los paquetes de análisis? <i>[Verificar que estén en orden alfabético y de cada paquete de análisis se describan sus objetivos y requisitos fundamentales]</i>
0304	1.2. ¿Se encuentra el o los diagramas de casos de uso que utiliza el proyecto? <i>[Verificar que en el diagrama de caso de uso se describan correctamente los siguientes elementos]</i>
0305	1.2.1. ¿Se encuentra la descripción de los actores? <i>[Verificar que se describa los actores primarios del sistema y las metas que estos pretenden alcanzar con el sistema]</i>

Capítulo 2. Elaboración del procedimiento

0306	1.2.2. ¿Se encuentran los requisitos funcionales? <i>[Verificar que se describa correctamente los requisitos funcionales]</i>
0307	1.3. ¿Se encuentran los requisitos no funcionales? <i>[Verificar que los requisitos no funcionales estén correctamente, mediante los siguientes elementos]</i>
0308	1.3.1. ¿Se identifican los requisitos que afectan la usabilidad? <i>[Verificar que se identifiquen todos los requisitos que afectan la usabilidad]</i>
0309	1.3.2. ¿Se identifican los requisitos relacionados con la confiabilidad? <i>[Verificar que se identifiquen correctamente los requisitos relacionados con la fiabilidad]</i>
0310	1.3.3. ¿Se encuentran las características de la eficiencia del sistema? <i>[Verificar que incluyan los tiempos de respuesta específico y que se hagan referencia a los casos de uso por el nombre]</i>
0311	1.3.4. ¿Se encuentra el requisito que refuerce el soporte del sistema? <i>[Verificar que se incluyan normas de codificación, convenciones para nombrado, bibliotecas de clase, el acceso y utilidades de mantenimiento]</i>
0312	1.3.5. ¿Se encuentra la restricción de diseño? <i>[Verificar que las restricciones de diseño sean las decisiones que se tomaron]</i>
0313	1.3.6. ¿Se encuentran especificados todos los requisitos legales, tales como patentes, derechos de autor, marca comercial, logotipo, requisito de licencia o restricción de uso? <i>[Verificar que se describan correctamente los requisitos legales]</i>

Lista de chequeo de modelo de datos:

Tabla 4: Lista de chequeo de modelo de datos (Elaboración propia).

Código	Indicadores a evaluar
0401	1. ¿Existe el documento "Vista de Datos"? <i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la Vista de Datos, a continuación elementos que no deben faltar en esta vista]</i>
0402	1.1. ¿Se encuentra detallada la caracterización del producto según la vista? <i>[Verificar que se identifiquen correctamente los diferentes tipos de datos que se quieren almacenar]</i>

Capítulo 2. Elaboración del procedimiento

0403	1.2. ¿Se encuentran descritos los escenarios y patrones de solución? <i>[Verificar que haya una correcta descripción de los escenarios y patrones de descripción en el documento]</i>
0404	1.3. ¿Se encuentran identificados los estándares? <i>[Verificar que se identifiquen correctamente los estándares que se utilicen]</i>
0405	1.4. ¿Se encuentra identificada en qué forma está la base de datos? <i>[Verificar que la forma normal esté identificada correctamente]</i>
0406	1.5. ¿Se encuentra en el diccionario de datos los elementos necesarios para identificar los conceptos para la modelación de la base de datos? <i>[Verificar que los términos se encuentren con la definición correcta y en caso de referenciar otro documento, verificar que en el mismo se encuentre correctamente]</i>
0407	1.6. ¿Se encuentran las especificaciones del diseño físico de la solución en el Modelo de datos? <i>[Verificar que se encuentren las especificaciones del diseño físico en el modelo de datos, en caso de referenciar algún documento, verificar que el mismo esté correcto]</i>
0408	1.7. ¿Se encuentran la descripción del modelo lógico del sistema en el Modelo de datos? <i>[Verificar que esté correcta la descripción del modelo lógico del sistema en el Modelo de Datos]</i>
0409	1.8. ¿Se utilizaron estándares para la identificación física de los ficheros de código fuente, scripts o de base de datos? <i>[Verificar que los estándares que se utilizaron para la identificación física estén correctos]</i>

2.5.3 Registro de no conformidades

El registro de no conformidades es el documento que se genera luego de analizadas las listas de chequeo por cada etapa, los comentarios realizados por los especialistas detectando problemas se traducen en no conformidades y estas son registradas para entregarla al proyecto revisado.

El registro de no conformidades en su primera parte consta del identificador del documento, especificando autor y quien revisa y aprueba el artefacto. También cuenta con la introducción, propósitos y objetivos, alcance y un resumen para guiar al revisor en las acciones que debe realizar. Para documentar las no conformidades, el artefacto establece una leyenda la cual se muestra en la figura 7 que muestra el código del indicador, la no conformidad que se deriva de ese indicador y una clasificación de la misma.

Capítulo 2. Elaboración del procedimiento



Registro de no conformidades

Leyenda:

Código: es el número que tiene en la lista de chequeo.

No conformidad detectada: son los errores encontrados durante la ejecución de la revisión.

Clasificación: es la importancia que tiene la no conformidad según el revisor técnico para el proyecto.

Etapa de arquitectura

Código	No conformidad detectada	Clasificación

Etapa de diseño

Código	No conformidad detectada	Clasificación

Etapa de requerimiento

Código	No conformidad detectada	Clasificación

Etapa de modelo de datos

Código	No conformidad detectada	Clasificación

Figura 7: Registro de no conformidades (Elaboración propia).

2.5.4 Acta de reunión de cierre

El acta de reunión de cierre es el documento que contiene los elementos a tratar en la reunión de cierre (Ver figura 9). En este artefacto se recogen los siguientes puntos:

- ✓ Lugar, fecha, hora: donde se realizará la reunión.
- ✓ Asistentes y ausentes: personal involucrado con la revisión.
- ✓ Orden del día: son los puntos que guiarán la reunión
- ✓ Cantidad de no conformidades por etapa: en esta tabla se relaciona la cantidad de no conformidad que se detectaron y documentaron en el artefacto registro de no conformidades.
- ✓ Estado del proyecto: es una valoración de la situación actual que tiene el proyecto en relación con la cantidad de no conformidades detectadas por etapa.
- ✓ Observación: son temas que en la reunión se requieren esclarecer.

Autor		Fecha	
Lugar		Hora Inicio	
Proyecto/Grupo		Hora Terminación	
Asunto			
Asistentes			
Ausentes			

Orden del día

1. Discusión del registro de las no conformidades
2. Informar el estado del proyecto

Cantidad de no conformidades por etapa

Etapa	Cantidad de no conformidades
Arquitectura	
Diseño	
Requerimiento	
Modelo de datos	

Estado del proyecto:

Observación:

Figura 8: Acta de reunión de cierre (Elaboración propia).

El artefacto acta reunión de cierre transita por dos estados, primero cuando se presenta en la reunión final y acta de reunión de cierre (Actualizada), como artefacto de salida de esta actividad. Con la salida de este artefacto se pone fin al procedimiento de revisión técnica formal.

2.6 Conclusiones

Los elementos fundamentales definidos para la elaboración del procedimiento permiten constatar que:

- ✓ Los roles y las responsabilidades de cada uno no se solapan y están en correspondencia con sus habilidades, lo que garantiza dar cumplimiento a las tareas específicas que se ejecutan durante la RTF.
- ✓ Las actividades definidas responden a la secuencia de pasos lógicos que guían la RTF en cada una de sus fases de ejecución.

Capítulo 2. Elaboración del procedimiento

- ✓ Los artefactos de entrada y salida que se generan en cada actividad permiten documentar el desarrollo de la misma y verifican el estado técnico del proyecto.

Capítulo 3. Probar el procedimiento

Para probar el procedimiento se aplica a un proyecto real y en curso, con el propósito de saber la fiabilidad que presenta el mismo. Además, conocer la veracidad de las actividades, el desempeño de los roles definidos y confirmar cuál es el estado actual del proyecto.

3.1 Aplicación del procedimiento a un proyecto productivo

Para probar el procedimiento se seleccionó un proyecto productivo del centro FORTES de la facultad 4. Con la elección realizada se demuestra la eficiencia del mismo y que puede ser aplicado a todos los proyectos del centro.

3.1.1 Planificación de la revisión

Reunión de inicio

La primera actividad que se realizó fue la reunión de inicio, en la que participó el revisor líder que fue el que dirigió la reunión, el líder del proyecto al cual se le realizará la revisión y los revisores que acompañarán al revisor técnico en la revisión por etapa. Se concilió el cronograma que se muestra en la tabla 5, se acordó realizar la preparación al equipo revisor, la cual estaría a cargo del líder de proyecto a revisar y se planificó para el día siguiente de la reunión de inicio.

Tabla 5: Cronograma.

Actividad	Fecha	Hora	Lugar	Responsable
Reunión inicio	19/05/2014	8.30 am	Laboratorio 103	Revisor líder
Preparación al equipo revisor	20/05/2014	8.30 am	Laboratorio 102	Líder de proyecto a revisar
Ejecución de la revisión	20/05/2014	10.00 am	Laboratorio 102	Revisor líder
Revisión a la etapa de arquitectura	20/05/2014	10.00 am	Laboratorio 102	Revisor 1
Revisión a la etapa de requerimiento	20/05/2014	10.00 am		Revisor 2

Capítulo 3. Probar el procedimiento

Revisión a la etapa de diseño	20/05/2014	2.00 pm	Laboratorio 102	Revisor 1
Revisión a la etapa de modelo de datos	20/05/2014	2.00 pm	Laboratorio 102	Revisor 2
Reunión de análisis de los resultados	21/05/2014	9.00 am	Laboratorio 103	Revisor líder
Reunión de cierre	21/05/2014	3.00 pm	Laboratorio 103	Revisor líder

Luego de conciliado el cronograma se tomaron los acuerdos que se relacionan en la tabla 6 y como consideraciones generales, que la reunión se efectuó a la hora y en el tiempo planificado.

Tabla 6: Acuerdos tomados.

No.	Acuerdos
1	La capacitación se impartirá en el laboratorio 102 por el líder de proyecto y deben estar los especialistas que serán revisores técnicos (Responsable líder del proyecto)
2	Se debe garantizar que el expediente de proyecto esté actualizado en el repositorio. (Responsable líder del proyecto)
3	Se debe garantizar que los revisores técnicos estén en el horario planificado para la revisión. (Responsable revisor líder)
4	Se debe garantizar que se entreguen las listas de chequeo en cuanto termine cada una de las revisiones por etapa (Responsables revisores)
5	Tener el registro de no conformidades listo para la reunión de cierre (Responsable revisor líder)

Preparación al equipo revisor

La actividad de preparación al equipo revisor se realizó como fue conciliada en la reunión de inicio. El responsable del proyecto a revisar le explicó al equipo revisor cómo es el funcionamiento del producto y le

Capítulo 3. Probar el procedimiento

mostró el expediente del proyecto para que se fueran familiarizando, para cuando realizaran la revisión se hiciera con la calidad requerida.

3.1.2 Ejecución de la revisión

Revisión por etapas

La actividad revisión por etapas comenzó al día siguiente de la preparación al equipo revisor. El revisor técnico junto al revisor aplicó la lista de chequeo según la etapa correspondida, encontrando no conformidades e indicadores que no se aplican. A continuación se muestran los errores encontrados en la lista de chequeo por etapas.

Lista de chequeo de arquitectura:

Tabla 7: Lista de chequeo de arquitectura.

Código	Indicadores a evaluar	Respuesta	Evaluación	Comentario
0102	1.1. ¿Se especifica en el documento cada una de las vistas? <i>[Debe explicarse cada una de las vistas y en caso de alguna no aplicar debe especificarse el por qué]</i>	No	M	Aunque el proyecto no aplica todas las vistas tiene que explicar el por qué de las vistas que no utiliza.
0105	1.2.2. ¿Se especifican y seleccionan las tecnologías requeridas para pruebas de concepto? <i>[Verificar que de acuerdo al tipo de proyecto si usa las tecnologías adecuada, el especialista puede recomendar otras en caso necesario]</i>	No	B	Se debe especificar las tecnologías requeridas para las pruebas de concepto de acuerdo al tipo de proyecto.
0107	1.2.4. ¿Se encuentra evaluado por criterios el modelo teórico de pruebas de concepto por cada vista? <i>[Verificar que se evalúe cada uno]</i>	Si	M	Sólo están algunas vistas evaluadas por el criterio el modelo teórico de pruebas

Capítulo 3. Probar el procedimiento

	<i>de los criterios en cada vista, de acuerdo al tipo de proyecto]</i>			de concepto, las que faltan no están evaluadas.
0109	3. ¿Existe el documento " Vista de Sistema"? <i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la Vista de Sistema, a continuación elementos que no deben faltar en esta vista]</i>	No	B	Se debe explicar por qué no se utiliza ésta vista.
0122	5.2. ¿Se encuentran descritos los escenarios y patrones de solución? <i>[Verificar que haya una correcta descripción de los escenarios y patrones de descripción en el documento]</i>	No	B	Se deben describir los escenarios y patrones de solución.
0123	5.3. ¿Se encuentra en el diccionario de datos los elementos necesarios para identificar los conceptos para la modelación de la base de datos? <i>[Verificar que los término se encuentren con la definición correcta y en caso de referenciar otro documento, verificar que en el mismo se encuentre correctamente]</i>	No	B	Se debe describir el diccionario de datos, para poder modelar correctamente los datos.
0124	5.4. ¿Se encuentran las especificaciones del diseño físico de la solución en el Modelo de datos? <i>[Verificar que se encuentren las especificaciones del diseño físico en el Modelo de datos, en caso de referenciar algún documento, verificar que el mismo esté correcto]</i>	NP		
0132	7.2. ¿Se selecciona la tecnología a utilizar por el Gestor de Base de Datos?	Si	R	Aunque selecciona la tecnología debe

Capítulo 3. Probar el procedimiento

	<i>[Verificar que se especifique por cada uno de los escenarios el Gestor de Base de Datos, en caso de no seleccionar ninguna verificar que la tabla esté llena con la especificación de la que se va a utilizar]</i>			llenar la tabla con los datos de la tecnología a utilizar.
0134	7.4. ¿Se selecciona la tecnología del lenguaje de programación a utilizar? <i>[Verificar que se haga un correcto uso de los lenguaje de programación a Nivel de negocio y a Bajo nivel]</i>	NP		
0141	8.5. ¿Se especifica la seguridad del servidor de la aplicación? <i>[Verificar que se identifique el estado de seguridad de la aplicación vista desde el entorno]</i>	NP		
0142	8.6. ¿Se especifica la tecnología para el sellado del código? <i>[Verificar que se especifique la tecnología que utilizará para la protección del código]</i>	NP		
0144	9.1. ¿Los elementos de los diagramas de los pasos 1, 2 y 3 son tratados para la descripción en el paso 4 de escenarios y patrones de solución? <i>[Verificar que los elementos de los diagramas de los pasos 1, 2 y 3 sean tratados para la descripción en el paso 4 de escenarios y patrones de solución]</i>	Si	B	Aunque se mencionan los escenarios de infraestructura se debe especificar si este es aplicable para la solución y de los mencionados ninguno de los escenarios de infraestructura es aplicable a la solución.

Capítulo 3. Probar el procedimiento

Lista de chequeo de diseño:

Tabla 8: Lista de chequeo de diseño.

Código	Indicadores a evaluar	Respuesta	Evaluación	Comentario
0204	1.4. ¿Se realiza la descripción de cada una de las clases que intervienen en el diagrama de clases? <i>[Verificar que permitan determinar la capa a la que pertenecen, módulo, número de la clase, nombre de la clase, propósito y descripción]</i>	No	B	Se debe realizar la descripción de todas las clases que intervienen en el diagrama de clases.

Lista de chequeo de requerimiento:

Tabla 9: Lista de chequeo de requerimiento.

Código	Indicadores a evaluar	Respuesta	Evaluación	Comentario
0302	1.1. ¿Se encuentran los paquetes de análisis? <i>[Verificar que esté correctamente el siguiente elemento]</i>	NP		
0303	1.1.1. ¿Se describen los paquetes de análisis? <i>[Verificar que estén en orden alfabético y de cada paquete de análisis se describan sus objetivos y requisitos fundamentales]</i>	NP		
0305	1.2.1. ¿Se encuentra la descripción de los actores? <i>[Verificar que se describa los actores primarios del sistema y las metas que estos pretenden alcanzar con el sistema]</i>	No	B	Se debe describir los actores primarios y las metas que estos pretendan alcanzar con el sistema.

Capítulo 3. Probar el procedimiento

0306	1.2.2. ¿Se encuentran los requisitos funcionales? <i>[Verificar que se describa correctamente los requisitos funcionales]</i>	Si	M	Para que un requisito funcional este descrito correctamente se debe describir además los campos que utiliza con su tipo de dato y su regla o restricciones.
0308	1.3.1. ¿Se identifican los requisitos que afectan la usabilidad? <i>[Verificar que se identifiquen todos los requisitos que afectan la usabilidad]</i>	No	B	Se debe identificar los requisitos que afectan la usabilidad del sistema.
0310	1.3.3. ¿Se encuentran las características de la eficiencia del sistema? <i>[Verificar que incluyan los tiempos de respuesta específico y que se hagan referencia a los casos de uso por el nombre]</i>	No	B	Se debe incluir los tiempos de respuesta específico y hacer referencia a los casos de uso por el nombre, para que exista una mejor eficiencia del sistema.
0312	1.3.5. ¿Se encuentra la restricción de diseño? <i>[Verificar que las restricciones de diseño sean las decisiones que se tomaron]</i>	No	B	Se debe poner las restricciones de diseño que emplearon en el sistema.

Capítulo 3. Probar el procedimiento

Lista de chequeo de modelo de datos:

Tabla 10: Lista de chequeo de modelo de datos.

Código	Indicadores a evaluar	Respuesta	Evaluación	Comentario
0403	1.2.1. ¿Se encuentran descritos los escenarios y patrones de solución? <i>[Verificar que haya una correcta descripción de los escenarios y patrones de descripción en el documento]</i>	NP		
0404	1.3. ¿Se encuentran identificados los estándares? <i>[Verificar que se identifiquen correctamente los estándares que se utilicen]</i>	No	B	Se debe identificar los estándares que se utilicen en el sistema.
0405	1.4. ¿Se encuentra identificada en qué forma está la base de datos? <i>[Verificar que la forma normal este identificada correctamente]</i>	No	B	Se debe especificar en qué forma está la base de datos, para saber si están correctas las cantidades de tablas que se generan.
0406	1.5. ¿Se encuentra en el diccionario de datos los elementos necesarios para identificar los conceptos para la modelación de la base de datos? <i>[Verificar que los términos se</i>	No	B	Se debe describir el diccionario de datos, para poder modelar correctamente los datos.

Capítulo 3. Probar el procedimiento

	<i>encuentren con la definición correcta y en caso de referenciar otro documento, verificar que en el mismo se encuentre correctamente]</i>			
0407	<p>1.6. ¿Se encuentran las especificaciones del diseño físico de la solución en el Modelo de datos?</p> <p><i>[Verificar que se encuentren las especificaciones del diseño físico en el Modelo de Datos, en caso de referenciar algún documento, verificar que el mismo esté correcto]</i></p>	NP		
0408	<p>1.7. ¿Se encuentra la descripción del modelo lógico del sistema en el Modelo de datos?</p> <p><i>[Verificar que este correcta la descripción del modelo lógico del sistema en el Modelo de Datos]</i></p>	NP		
0409	<p>1.8. ¿Se utilizaron estándares para la identificación física de los ficheros de código fuente, scripts o de base de datos?</p> <p><i>[Verificar que los estándares que se utilizaron para la identificación física estén correctos]</i></p>	No	B	Se debe identificar los estándares que utilizaron para la identificación física de los ficheros.

Capítulo 3. Probar el procedimiento

3.1.3 Análisis de los resultados

Análisis de los resultados

En la actividad análisis de los resultados, los revisores junto al revisor líder analizaron las listas de chequeo por etapas aplicadas y elaboraron el registro de no conformidades. A continuación se muestran las no conformidades detectadas por cada etapa y su clasificación.

Etapa de Arquitectura:

Tabla 11: No conformidades de la etapa de arquitectura.

Código	No conformidad detectada	Clasificación
0102, 0109,	En el documento guía base no se especifica por qué no utiliza todas las vistas. Ej. Sistema,	Significativa
0105	Se deben especificar las tecnologías requeridas para las pruebas de concepto de acuerdo al tipo de proyecto.	Significativa
0107	Sólo están algunas vistas evaluadas por el criterio el modelo teórico de pruebas de concepto. ¿Las que faltan no están evaluadas?	Significativa
0122	En el documento guía base los escenarios y patrones de solución que se describen no están en completa correspondencia con la implementación con que se desarrolla el producto.	Significativa
0123	En el diccionario de datos no están descritos todos los elementos que son útiles para la construcción del modelo de datos, lo que trae consigo que personal que no sea experto en la temática pueda entender la aplicación.	Significativa

Capítulo 3. Probar el procedimiento

0124	En el modelo de datos debería especificarse el diseño físico de la solución.	Significativa
0132	Aunque selecciona la tecnología debe llenar la tabla con los datos de la tecnología a utilizar.	No significativa
0133	Debe especificarse la tecnología para el sellado del código.	Recomendación
0141	No se especifica en el documento el estado de seguridad del proyecto.	Recomendación
0144	Aunque se mencionan los escenarios de infraestructura se debe especificar cuál es aplicable para la solución y de los mencionados ninguno de los escenarios de infraestructura es aplicable a la solución.	Significativa

Etapa de diseño:

Tabla 12: No conformidades de la etapa de diseño.

Código	No conformidad detectada	Clasificación
0204	Se debe realizar la descripción de todas las clases que intervienen en el diagrama de clases.	Significativa

Etapa de requerimiento:

Tabla 13: No conformidades de la etapa de requerimiento.

Código	No conformidad detectada	Clasificación
0302, 0303	No se encuentran los paquetes de análisis, por lo que no se puede definir cuáles son los requisitos fundamentales.	Significativa

Capítulo 3. Probar el procedimiento

0305	Se debe describir los autores primarios y las metas que estos pretendan alcanzar con el sistema.	Significativa
0306	Para que un requisito funcional este descrito correctamente se debe describir además los campos que utiliza con su tipo de dato y su regla o restricciones.	Significativa
0308	Se debe identificar los requisitos que afectan la usabilidad del sistema.	Significativa
0310	Se debe incluir los tiempos de respuesta específico y hacer referencia a los casos de uso por el nombre, para que exista una mejor eficiencia del sistema.	Significativa
0312	Se debe poner las restricciones de diseño que emplearon en el sistema.	No Significativa

Etapa de modelo de datos:

Tabla 14: No conformidades de la etapa de modelo de datos.

Código	No conformidad detectada	Clasificación
0403	Deberían describirse los patrones y escenarios de solución para una mejor comprensión del documento.	Recomendación
0404	Se debe identificar los estándares que se utilicen en el sistema.	Significativa
0405	Se debe especificar en qué forma está la base de datos, para saber si están correctas las cantidades de tablas que se generan.	Significativa
0406	Se debe describir el diccionario de datos, para una mejor modelación de los datos.	No Significativa

Capítulo 3. Probar el procedimiento

0407	Debería especificarse el diseño físico en la solución del modelo de datos.	Recomendación
0409	Se debe identificar los estándares que utilizaron para la identificación física de los ficheros.	Significativa

Reunión de cierre

Con la actividad reunión de cierre concluye el procedimiento, en ella se reunieron los revisores los cuales presentaron los resultados de las no conformidades por etapa, el revisor líder que fue el que dirigió la reunión de cierre y el líder del proyecto al cual le fue entregado el registro de no conformidades. A continuación se muestra la relación de las no conformidades encontradas en la revisión.

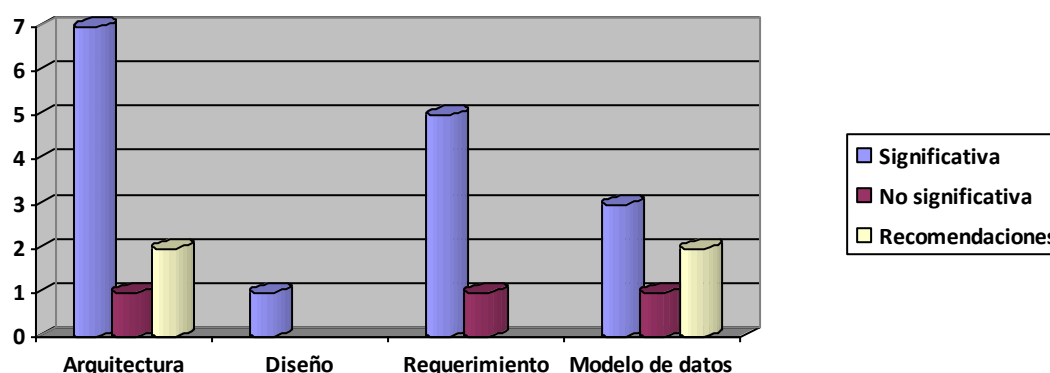


Figura 9: Relación de las no conformidades encontradas en la RTF.

En esta actividad también se analizó el estado del proyecto, donde los revisores informaron lo siguiente: en general el proyecto debe trabajar en las no conformidades significativas de modo que garantice una mejor calidad técnica, y no descuidar las no significativas y recomendaciones pues estas en un futuro podrían afectar al producto.

Con este resultado se puede constatar que la RTF permiten analizar con profundidad los documentos y entregables de un expediente de proyecto. Una RTF permite verificar no solo que los documentos estén, sino se puede corroborar con la ayuda de especialistas que estos están correctamente ejecutados. Al proyecto revisado primeramente se le aplicó una revisión y en esta no se detectaron no conformidades mientras que al aplicar la RTF si existieron incongruencias técnicas. Este resultado nos permite afirmar que es necesario revisar técnicamente los proyectos del centro.

Capítulo 3. Probar el procedimiento

3.2 Conclusiones

La aplicación del procedimiento a un proyecto real demostró que:

- ✓ Con la ejecución de cada una de las actividades se logró aplicar la RTF de forma organizada, permitiendo a todos los involucrados tener conocimiento de cómo ejecutarlo, cuándo realizarlo y qué deben hacer en cada una de ellas.
- ✓ Los artefactos generados respondieron a las necesidades de la RTF, permitiendo documentar el desarrollo de las actividades realizadas.
- ✓ Se hace necesaria la RTF para detectar errores técnicos que hasta el momento no eran evidentes en las revisiones de *software* a los diferentes proyectos del centro.

Conclusiones generales

Con la presente investigación se llega a las siguientes conclusiones:

- ✓ El estudio realizado a los distintos conceptos relacionados con la RTF permitió conocer que esta profundiza en la revisión que se realiza al *software*, verificando que se encuentre técnicamente correcto y debe ser ejecutada por especialistas, para garantizar que el producto cuente con la calidad técnica requerida.
- ✓ Los elementos fundamentales definidos para la elaboración del procedimiento, permiten constatar que los roles, actividades y artefactos que se generan, garantizan dar cumplimiento a las tareas específicas que se ejecutan, responden a la secuencia de pasos lógicos de la RTF, permiten documentar el desarrollo de la misma y verifican el estado técnico del proyecto.
- ✓ La aplicación del procedimiento a un proyecto real se ejecutó de forma organizada, los artefactos generados permitieron documentar el desarrollo de las actividades realizadas, permitiendo a los involucrados tener conocimiento en cada una de ellas y la detección de errores técnicos que no eran evidentes en las revisiones de *software*.

Bibliografía

1. **610-1900, IEEE estándar.** *Calidad de Software.* 1900. 610
2. **8402-1994, ISO.** *Calidad de Software.* 1994. 8402.
3. **Pressman, Roger.** *Reflexión de Calidad del Software de Roger Pressman.* 2010. Quinta edición.
4. **1028-1997, IEEE Standard.** *Software Reviews.* 1997. 1028.
5. **Pressman, Roger.** *Revisiones de Software.* 2005. Quinta edición.
6. **Garcia, Yinimary Ortega Montoya y Zayli Noda.** *Estrategia de Control de la Calidad mediante Revisiones y Auditorías.* La Habana, Cuba. : s.n., 2007. Universidad de las Ciencias Informáticas.
7. **Díaz, Indira Pérez.** *Procedimiento de Revisiones de Software.* La Habana, Cuba : s.n., 2012. Universidad de las Ciencias Informáticas.
8. **Mata, Edith Alhelí Martínez.** *Clasificación de las Revisiones de Software.* [En línea] 12 de diciembre de 2008. [Citado el: 12 de noviembre de 2013.] <http://sg.com.mx/content/view/708>. Consultor especializado en Aseguramiento de la Calidad.
9. **Wiegers, Karl E.** *Process Assets for Software Peer Reviews and Inspections.* Process Impact. [En línea] 2007. [Citado el: 25 de noviembre de 2013.] <http://www.processimpact.com>.
10. **9001:2008, ISO.** *Revisión Técnica Formal.* 2008. 9001.
11. **Barrueto, Miton Leonel Ibarra.** *Gestión de Calidad de Software,*. Manabi, Ecuador : s.n., 2013 . Universidad Laica "Eloy Alfaro" .
12. **Sotés, Roberto Félix Zamuriano.** *Proceso de Inspección de Software y Aseguramiento de la Calidad.* Bolivia : s.n., 2011.

Bibliografía

13. **Martin, A. A. O.** *Propuesta de Procedimiento*. Venezuela : s.n., 2011. Universidad Rómulo Gallegos.
14. **Delgados, M.D.** *Propuesta del Centro de Estudios de Ingeniería de Sistema*. Habana, Cuba : s.n., 2010.
15. **Segoviano, Z.M.** *Revisiones de Software* . Habana, Cuba : s.n., 2010. Universidad de las Ciencias Informáticas.
16. **830-1998, IEEE Std.** *IEEE Recommended Practice for Software Requirements Specifications*. 1998. 830.
17. **Pressman, Roger.** *Ingeniería del Software. Un enfoque práctico* . 2002. 5ta. Edición.
18. **Reynoso, Billy.** *Introducción a la Arquitectura de Software*. Buenos Aires : s.n. Universidad de Buenos Aires.
19. **Ruiz, Francisco.** *Diseño del Software* . Universidad Cantabria Facultad de Ciencias.
20. **Society, IEEE Computer.** *2004. Guide to the Software Engineering Body of Knowledge*.
21. **Sommerville, Ian.** *Ingeniería de Software*. 2002. 6ta. Edición.
22. **9000:2000, ISO.** *Principios de la Gestión de la Calidad*. 2000. 90000.

Anexos

Anexo # 1: Acta de la reunión de inicio.



Reunión de inicio

Autor		Fecha	
Lugar		Hora Inicio	
Proyecto/Grupo		Hora Terminación	
Asunto			
Asistentes			
Ausentes			

Orden del día

1. Conciliación del cronograma
2. Acuerdos
3. Consideraciones generales

Cronograma

Actividad	Fecha	Hora	Lugar	Responsable

Acuerdos tomados

No.	Acuerdo
1	
2	

Consideraciones generales:

Observaciones:

Anexo # 2: Lista de chequeo de arquitectura.



Lista de chequeo de RTF

**Lista de Chequeo de Revisión Técnica Formal de
Arquitectura**

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mm/yy>	<x.x>	<descripción de la actividad a realizar>	<nombre del especialista que elabora la lista de chequeo>
<dd/mm/yy>	<x.x>	<descripción de la actividad a realizar>	<nombre del especialista que aplica la lista de chequeo>

Introducción

[Se realiza una breve sinopsis de lo que recoge el informe de evaluación.]

Propósito y objetivos

[Se enuncian los objetivos y el propósito que se persigue con la realización del Informe]

Alcance

[Proyecto al que se involucra]

Resumen

[Se realiza un pequeño resumen de lo que se enunciará en la plantilla]

Leyenda:

Código: está compuesto por cuatro dígitos, el primer par de números es el de la etapa y el segundo par del número del indicador.

01 (Etapa de arquitectura)

02 (Etapa de diseño)

03 (Etapa de requerimiento)

04 (Etapa de modelo de datos)

Ejemplo: 0101 (los número que están de color rojo pertenece al código de la etapa de arquitectura y los restantes al número del indicador)

Indicadores a evaluar: son las actividades que deben cumplirse para que la etapa a revisar esté con todos los elementos que se necesitan en el software.

Respuesta: si aplica se responde **Si** o **No** según los indicadores a evaluar y si no aplica **NP**.

Evaluación: se pone alta, media o baja según el cumplimiento del indicador.

Alta (A): se cumple el indicador a evaluar en la etapa a revisar.

Media (M): le faltan algunos detalles para cumplirse el indicador a evaluar.

Baja (B): no se cumple el indicador a evaluar.

Comentario: se explica cuando se de media o baja.

Anexos

Código	Indicadores a evaluar	Respuesta	Evaluación	Comentario
0101	<p>12. ¿Existe el documento "Guía base"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la arquitectura que se maneja en el proyecto]</i></p>			
0102	<p>12.1. ¿Se especifica en el documento cada una de las vistas?</p> <p><i>[Debe explicarse cada una de las vistas y en caso de alguna no aplicar debe especificarse el por qué]</i></p>			
0103	<p>12.2. ¿Se especifica en el documento la Estrategia para Pruebas de Concepto?</p> <p><i>[Debe explicarse la estrategia para las pruebas de concepto verificando los puntos siguientes]</i></p>			
0104	<p>12.2.1. ¿Se especifican y seleccionan las técnicas para pruebas de concepto adecuadas?</p> <p><i>[Verificar de acuerdo al tipo de proyecto si usa la técnica adecuada, el especialista puede recomendar una técnica más óptima, en caso</i></p>			

Anexos

	<i>necesario]</i>			
0105	<p>12.2.2. ¿Se especifican y seleccionan las herramientas para las pruebas de estrés?</p> <p><i>[Verificar que de acuerdo al tipo de proyecto es suficiente con las pruebas de estrés seleccionadas, puede recomendar otras]</i></p>			
0106	<p>12.2.3. ¿Se encuentra evaluado por criterios el modelo teórico de pruebas de concepto por cada vista?</p> <p><i>[Verificar que se evalúe cada uno de los criterios en cada vista, de acuerdo al tipo de proyecto]</i></p>			
0107	<p>13. El documento " Vista de procesos".</p> <p><i>[No aplica para FORTES]</i></p>			
0108	<p>14. ¿Existe el documento " Vista de Sistema"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la vista de sistema, a continuación elementos que no deben faltar en esta vista]</i></p>			
0109	<p>14.1. ¿Se especifican los</p>			

Anexos

	<p>requisitos priorizados del sistema?</p> <p><i>[Verificar que se listen los requisitos priorizados y que se siga el procedimiento correcto, explicado en el documento]</i></p>			
0110	<p>14.2. ¿Se encuentra el agrupamiento y priorización de los requisitos siguiendo cada una de las acciones?</p> <p><i>[Verificar que se sigan todas las acciones explicadas en el documento]</i></p>			
0111	<p>14.3. ¿Se identifican los paquetes y componentes principales siguiendo las reglas establecidas en el documento?</p> <p><i>[Verificar que se sigan todas las reglas explicadas en el documento]</i></p>			
0112	<p>14.4. ¿Se representa la dependencia entre paquetes?</p> <p><i>[Verificar que se represente a través de diagrama todas las dependencias, y que estas estén correctas]</i></p>			
0113	<p>14.5. ¿Se determinan los escenarios de la vista?</p> <p><i>[Verificar que se determinen todos los escenarios siguiendo las acciones]</i></p>			

Anexos

	<i>descritas en el documento]</i>			
0114	<p>15. ¿Existe el documento " Vista de presentación"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la vista de presentación, a continuación elementos que no deben faltar en esta vista]</i></p>			
0115	<p>15.1. ¿Se definen en el documento las pantallas?</p> <p><i>[Verificar que se haga una correcta definición de las pantallas en correspondencia con el tipo de proyecto]</i></p>			
0116	<p>15.2. ¿Se definen en el documento contenido, pantallas genéricas de subsistemas o módulos?</p> <p><i>[Verificar que se haga una correcta definición de los contenidos y pantallas en correspondencia con el tipo de proyecto]</i></p>			
0117	<p>15.3. ¿Se definen las pantallas de áreas de trabajo de cada módulo?</p> <p><i>[Verificar que se haga una correcta definición de las pantalla por área de trabajo correspondencia con el tipo de proyecto]</i></p>			

Anexos

0118	<p>15.4. ¿Se describen los escenarios de la vista de presentación que se aplican a su solución?</p> <p><i>[Verificar que se describan los escenarios que se aplican a la solución de acuerdo al tipo de proyecto]</i></p>			
0119	<p>16. ¿Existe el documento " Vista de Datos"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la Vista de Datos, a continuación elementos que no deben faltar en esta vista]</i></p>			
0120	<p>16.1. ¿Se encuentra detallada la caracterización del producto según la vista?</p> <p><i>[Verificar que se identifiquen correctamente los diferentes tipos de datos que se quieren almacenar]</i></p>			
0121	<p>16.2. ¿Se encuentran descritos los escenarios y patrones de solución?</p> <p><i>[Verificar que haya una correcta descripción de los escenarios y patrones de descripción en el documento]</i></p>			
0122	<p>16.3. ¿Se encuentra en el</p>			

Anexos

	<p>diccionario de datos los elementos necesarios para identificar los conceptos para la modelación de la base de datos?</p> <p><i>[Verificar que los término se encuentren con la definición correcta y en caso de referenciar otro documento, verificar que en el mismo se encuentre correctamente]</i></p>			
0123	<p>16.4. ¿Se encuentran las especificaciones del diseño físico de la solución en el Modelo de datos?</p> <p><i>[Verificar que se encuentren las especificaciones del diseño físico en el modelo de datos, en caso de referenciar algún documento, verificar que el mismo esté correcto]</i></p>			
0124	<p>17. ¿Existe el documento " Vista de integración "?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la Vista de Integración, a continuación elementos que no deben faltar en esta vista]</i></p>			
0125	<p>17.1. ¿Se seleccionan los estilos arquitectónicos a emplear?</p> <p><i>[Verificar que se seleccionen los</i></p>			

Anexos

	<i>estilos y patrones que utilizará, siguiendo las reglas brindadas en el documento]</i>			
0126	<p>17.2. ¿Se selecciona la estrategia de diseño y desarrollo del proyecto?</p> <p><i>[Verificar que se seleccione la estrategia de diseño adecuada para el tipo de proyecto]</i></p>			
0127	<p>17.3. ¿Se seleccionan los patrones de diseño, principales vista desde la integración entre las soluciones?</p> <p><i>[Verificar que se seleccionen los patrones de diseño adecuados para el tipo de proyecto]</i></p>			
0128	<p>17.4. ¿Se seleccionan estándares de codificación?</p> <p><i>[Verificar que se seleccionen los estándares de codificación adecuados para el tipo de proyecto]</i></p>			
0129	<p>18. ¿Existe el documento " Vista de Entornos de desarrollo tecnológico"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a los entornos de desarrollo tecnológico, a</i></p>			

Anexos

	<i>continuación elementos que no deben faltar en esta vista]</i>			
0130	<p>18.1. ¿Se selecciona la tecnología a utilizar según el Sistema Operativo?</p> <p><i>[Verificar que se especifique por cada uno de los escenarios de despliegue el Sistema Operativo]</i></p>			
0131	<p>18.2. ¿Se selecciona la tecnología a utilizar por el Gestor de Base de Datos?</p> <p><i>[Verificar que se especifique por cada uno de los escenarios el Gestor de Base de Datos, en caso de no seleccionar ninguna verificar que la tabla esté llena con la especificación de la que se va a utilizar]</i></p>			
0132	<p>18.3. ¿Se selecciona la tecnología ORM a utilizar?</p> <p><i>[Verificar que se especifique la tecnología ORM adecuada para el tipo de proyecto, en caso de no seleccionar ninguna verificar que la tabla esté llena con la especificación de la que se va a utilizar]</i></p>			
0133	<p>18.4. ¿Se selecciona la tecnología del lenguaje de programación a utilizar?</p> <p><i>[Verificar que se haga un correcto uso de los lenguaje de programación a Nivel de negocio y a Bajo nivel]</i></p>			

Anexos

0134	<p>18.5. ¿Se selecciona la tecnología del marco de trabajo?</p> <p><i>[Verificar que se especifique la selección del marco de trabajo adecuado para el desarrollo e interfaz usuario, en caso de seleccionar ninguna verificar que la tabla esté llena con la especificación de la que se va a utilizar]</i></p>			
0135	<p>19. ¿Existe el documento " Vista de Seguridad"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la seguridad, a continuación elementos que no deben faltar en esta vista]</i></p>			
0136	<p>19.1. ¿Se identifican los requisitos de seguridad del producto?</p> <p><i>[Verificar que se especifiquen en las preguntas mínimas que deben quedar respondidas y que deben permitir identificar requisitos de seguridad]</i></p>			
0137	<p>19.2. ¿Se identifica la autenticación y la capa de presentación?</p> <p><i>[Verificar que se evalúe asignándole la puntuación máxima a los elementos que debe utilizar para el diseño y desarrollo de la aplicación]</i></p>			

Anexos

0138	<p>19.3. ¿Se especifica el nivel de la capa de negocio?</p> <p><i>[Verificar que se evalúe el nivel de la capa de negocio asignándole la puntuación máxima a los elementos que debe utilizar en el diseño y desarrollo de la aplicación]</i></p>			
0139	<p>19.4. ¿Se especifica el nivel de la capa de datos?</p> <p><i>[Verificar que se evalúe asignándole la puntuación máxima a los elementos que debe utilizar para el diseño y desarrollo de la aplicación]</i></p>			
0140	<p>19.5. ¿Se especifica la seguridad del servidor de la aplicación?</p> <p><i>[Verificar que se identifique el estado de seguridad de la aplicación vista desde el entorno]</i></p>			
0141	<p>19.6. ¿Se especifica la tecnología para el sellado del código?</p> <p><i>[Verificar que se especifique la tecnología que utilizará para la protección del código]</i></p>			
0142	<p>20. ¿Existe el documento " Vista de Infraestructura"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe</i></p>			

Anexos

	<i>estar definido todo lo referente a la Vista de Infraestructura, a continuación elementos que no deben faltar en esta vista]</i>			
0143	<p>20.1. ¿Los elementos de los diagramas de los pasos 1, 2 y 3 son tratados para la descripción en el paso 4 de escenarios y patrones de solución?</p> <p><i>[Verificar que los elementos de los diagramas de los pasos 1, 2 y 3 sean tratados para la descripción en el paso 4 de escenarios y patrones de solución]</i></p>			
0144	<p>20.2. ¿Se especifican los escenarios de infraestructura?</p> <p><i>[Verificar que a partir de los análisis en los pasos anteriores identifique los escenarios y los patrones de solución en los mismos]</i></p>			
0145	<p>21. ¿Existe el documento "Vista de Despliegue"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la Vista de Despliegue, a continuación elementos que no deben faltar en esta vista]</i></p>			

Anexos

0146	21.1. ¿Se especifica la solución vista desde las licencias de los componentes que la forman? <i>[Verificar que se seleccione que tipos de licencias soportan los componentes de la solución]</i>			
0147	21.2. ¿Se especifican los requerimientos mínimos de instalación? <i>[Verificar que se seleccionen los requerimientos mínimos para la instalación]</i>			

Nombre y Apellido del Revisor: _____

Anexo # 3: Lista de chequeo de diseño.

**Lista de Chequeo de Revisión Técnica Formal
de Diseño**

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Código	Indicadores a evaluar	Respuesta	Evaluación	Comentario
0201	<p>3. ¿Existe el documento "Modelo de diseño"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura y diseño. En este debe estar definido todo lo referente al Modelo de Diseño, a continuación elementos que no deben faltar en este modelo]</i></p>			
0202	<p>3.1. ¿Se encuentra el diagrama de paquetes?</p> <p><i>[Verificar que el sistema esté organizado en pequeños paquetes y subsistemas]</i></p>			
0203	<p>3.2. ¿Se encuentra el diagrama de clases?</p> <p><i>[Verificar que estén las interacciones entre ellas]</i></p>			
0204	<p>3.3. ¿Se realiza la descripción de cada una de las clases que</p>			

Anexos

	<p>intervienen en el diagrama de clases?</p> <p><i>[Verificar que permitan determinar la capa a la que pertenecen, módulo, número de la clase, nombre de la clase, propósito y descripción]</i></p>			
0205	<p>4. ¿El tipo de aplicación es web?</p> <p><i>[Verificar que en el documento Modelo de diseño, se encuentren los siguientes elementos]</i></p>			
0206	<p>4.1. ¿Se identifican los tres estereotipos para identificar las principales clases?</p> <p><i>[Verificar que los estereotipos sean Server page, Client page y Form]</i></p>			
0207	<p>4.2. ¿Se encuentran los formularios relacionados con clase cliente?</p> <p><i>[Verificar que todos los formularios estén relacionados con la clase cliente]</i></p>			
0208	<p>4.3. ¿Se encuentra una clase cliente que esté relacionada con una clase servidor?</p> <p><i>[Verificar que al menos una clase cliente esté relacionada con una clase servidor y la relación sea de composición]</i></p>			
0209	<p>4.4. ¿Se encuentra la clase formulario relacionada con la clase controladora?</p> <p><i>[Verificar que la relación de la clase formulario con la clase controladora es <<</i></p>			

Anexos

	<i>submit>>]</i>			
0210	<p>4.5. ¿Se encuentra la clase servidor relacionada con la clase cliente?</p> <p><i>[Verificar que la relación de la clase servidor con la clase cliente es <<build>>]</i></p>			
0211	<p>4.6. ¿Se encuentra la relación de clase cliente a cliente?</p> <p><i>[Verificar que la relación de clase cliente a cliente es <<link>>]</i></p>			
0212	<p>4.7. ¿Se encuentran las sintaxis de las clases?</p> <p><i>[Verificar que estén correctas todas las sintaxis de las clases]</i></p>			
0213	<p>4.8. ¿Se encuentren las sintaxis de las funcionalidades?</p> <p><i>[Verificar que estén correctas todas las sintaxis de las funcionalidades]</i></p>			
0214	<p>4.9. ¿Se encuentran las relaciones entre clases?</p> <p><i>[Verificar que todas las relaciones entre clases estén correctas]</i></p>			
0215	<p>4.10. ¿Se encuentra la distribución de las clases?</p> <p><i>[Verificar que la distribución de las clases sea de acuerdo al framework o la librería que utilice]</i></p>			
0216	<p>4.11. ¿Existe la clase u ORM que permita la conexión a los datos?</p>			

Anexos

	<i>[Verificar que la clase u ORM permita la conexión a los datos]</i>			
--	---	--	--	--

Nombre y Apellido del Revisor: _____

Anexo # 4: Lista de chequeo de requerimiento.

Lista de Chequeo de Revisión Técnica Formal de Requerimiento

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Código	Indicadores a evaluar	Respuesta	Evaluación	Comentario
0301	<p>2. ¿Existe el documento "Especificación de Requisitos"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Requisitos. En este debe estar definido todo lo referente a los requisitos, a continuación elementos que no deben faltar]</i></p>			
0302	<p>2.1. ¿Se encuentran los paquetes de análisis?</p> <p><i>[Verificar que esté correctamente el siguiente elemento]</i></p>			
0303	<p>2.1.1. ¿Se describen los paquetes de análisis?</p> <p><i>[Verificar que estén en orden alfabético y de cada paquete de análisis se describan sus objetivos y requisitos fundamentales]</i></p>			
0304	<p>2.2. ¿Se encuentra el o los diagramas de casos de uso que utiliza el proyecto?</p> <p><i>[Verificar que en el diagrama de caso de</i></p>			

Anexos

	<i>uso se describan correctamente los siguientes elementos]</i>			
0305	<p>2.2.1. ¿Se encuentra la descripción de los autores?</p> <p><i>[Verificar que se describa los actores primarios del sistema y las metas que estos pretenden alcanzar con el sistema]</i></p>			
0306	<p>2.2.2. ¿Se encuentran los requisitos funcionales?</p> <p><i>[Verificar que se describa correctamente los requisitos funcionales]</i></p>			
0307	<p>2.3. ¿Se encuentran los requisitos no funcionales?</p> <p><i>[Verificar que los requisitos no funcionales estén correctamente, mediante los siguientes elementos]</i></p>			
0308	<p>2.3.1. ¿Se identifican los requisitos que afectan la usabilidad?</p> <p><i>[Verificar que se identifiquen todos los requisitos que afectan la usabilidad]</i></p>			
0309	<p>2.3.2. ¿Se identifican los requisitos relacionados con la confiabilidad?</p> <p><i>[Verificar que se identifiquen correctamente los requisitos relacionados con la fiabilidad]</i></p>			
0310	<p>2.3.3. ¿Se encuentran las características de la</p>			

Anexos

	<p style="text-align: center;">eficiencia del sistema?</p> <p><i>[Verificar que incluyan los tiempos de respuesta específico y que se hagan referencia a los casos de uso por el nombre]</i></p>			
0311	<p style="text-align: center;">2.3.4. ¿Se encuentra el requisito que refuerce el soporte del sistema?</p> <p><i>[Verificar que se incluyan normas de codificación, convenciones para nombrado, bibliotecas de clase, el acceso y utilidades de mantenimiento]</i></p>			
0312	<p style="text-align: center;">2.3.5. ¿Se encuentra la restricción de diseño?</p> <p><i>[Verificar que las restricciones de diseño sean las decisiones que se tomaron]</i></p>			
0313	<p style="text-align: center;">2.3.6. ¿Se encuentran especificados todos los requisitos legales, tales como patentes, derechos de autor, marca comercial, logotipo, requisito de licencia o restricción de uso?</p> <p><i>[Verificar que se describan correctamente los requisitos legales]</i></p>			

Nombre y Apellido del Revisor: _____

Anexo # 5: Lista de chequeo de modelo de datos.

Lista de Chequeo de Revisión Técnica Formal de Modelos de datos

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Código	Indicadores a evaluar	Respuesta	Evaluación	Comentario
0401	<p>2. ¿Existe el documento "Vista de Datos"?</p> <p><i>[Este documento se encuentra en el expediente de proyecto en Arquitectura de software. En este debe estar definido todo lo referente a la Vista de Datos, a continuación elementos que no deben faltar en esta vista]</i></p>			
0402	<p>2.1. ¿Se encuentra detallada la caracterización del producto según la vista?</p> <p><i>[Verificar que se identifiquen correctamente los diferentes tipos de datos que se quieren almacenar]</i></p>			
0403	<p>2.2. ¿Se encuentran descritos los escenarios y patrones de solución?</p> <p><i>[Verificar que haya una correcta descripción de los escenarios y patrones de descripción en el documento]</i></p>			

Anexos

0404	<p>2.3. ¿Se encuentran identificados los estándares?</p> <p><i>[Verificar que se identifiquen correctamente los estándares que se utilicen]</i></p>			
0405	<p>2.4. ¿Se encuentra identificada en qué forma está la base de datos?</p> <p><i>[Verificar que la forma normal esté identificada correctamente]</i></p>			
0406	<p>2.5. ¿Se encuentra en el diccionario de datos los elementos necesarios para identificar los conceptos para la modelación de la base de datos?</p> <p><i>[Verificar que los términos se encuentren con la definición correcta y en caso de referenciar otro documento, verificar que en el mismo se encuentre correctamente]</i></p>			
0407	<p>2.6. ¿Se encuentran las especificaciones del diseño físico de la solución en el Modelo de datos?</p> <p><i>[Verificar que se encuentren las especificaciones del diseño físico en el modelo de datos, en caso de referenciar algún documento, verificar que el mismo esté correcto]</i></p>			
0408	<p>2.7. ¿Se encuentran la descripción del modelo lógico del sistema en el Modelo de datos?</p> <p><i>[Verificar que esté correcta la descripción del modelo lógico del sistema en el Modelo de</i></p>			

Anexos

	<i>Datos]</i>			
0409	2.8. ¿Se utilizaron estándares para la identificación física de los ficheros de código fuente, scripts o de base de datos? <i>[Verificar que los estándares que se utilizaron para la identificación física estén correctos]</i>			

Nombre y Apellido del Revisor: _____

Anexo # 6: Registro de no conformidades.



Registro de no conformidades

	Nombre y apellidos	Cargo
Redactado por	-	-
Revisado por	-	-
Aprobado por	-	-

Introducción

[Se realiza una breve sinopsis de lo que recoge el informe de evaluación.]

Propósito y objetivos

[Se enuncian los objetivos y el propósito que se persigue con la realización del Informe]

Alcance

[Proyecto al que se involucra]

Resumen

[Se realiza un pequeño resumen de lo que se enunciará en la plantilla]

Leyenda:

Código: es el número que tiene en la lista de chequeo.

No conformidad detectada: son los errores encontrados durante la ejecución de la revisión.

Clasificación: es la importancia que tiene la no conformidad según el revisor técnico para el proyecto.

Etapa de arquitectura

Código	No conformidad detectada	Clasificación

Etapa de diseño

Código	No conformidad detectada	Clasificación

Anexos

Etapa de requerimiento

Código	No conformidad detectada	Clasificación

Etapa de modelo de datos

Código	No conformidad detectada	Clasificación

Anexos

Anexo # 3: Acta de reunión de cierre.



Reunión de cierre

Autor		Fecha	
Lugar		Hora Inicio	
Proyecto/Grupo		Hora Terminación	
Asunto			
Asistentes			
Ausentes			

Orden del día

1. Discusión del registro de las no conformidades
2. Informar el estado del proyecto

Cantidad de no conformidades por etapa

Etapa	Cantidad de no conformidades
Arquitectura	
Diseño	
Requerimiento	
Modelo de datos	

Estado del proyecto:

Observación: