



Área Académica de Administración de Tecnologías de Información

“Diseño de sistema para el aseguramiento y el control de la calidad en los proyectos de software del TEC Digital”

Informe de trabajo final de graduación para optar por el título de  
Administración de Tecnología de Información  
con el grado académico de licenciatura

Michael Lizandro Sánchez Soto

I semestre

Cartago  
Junio, 2017

**ÁREA ACADÉMICA DE ADMINISTRACIÓN DE TECNOLOGÍAS DE INFORMACIÓN  
GRADO ACADÉMICO: LICENCIATURA**

Los miembros del Tribunal Examinador del Área de Administración de Tecnologías de Información, recomendamos que el presente informe del Trabajo Final de Graduación del estudiante Michael Lizandro Sánchez Soto sea aceptado como requisito parcial para obtener el grado académico de Licenciatura en Administración de Tecnología de Información.



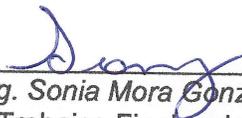
Ing. Julia Beatriz Espinoza Guzmán  
Profesora Asesora



Ing. Luis Javier Chavarría Sánchez, M.EdT.  
Miembro Tribunal Examinador



Lic. Néstor Morales Rodríguez  
Miembro Tribunal Examinador



Ing. Sonia Mora González, M.B.A.  
Coordinadora de Trabajos Finales de Graduación  
Licenciatura en Administración de Tecnología de Información

Junio, 2017

## **Dedicatoria**

A mis padres, Maureem y Guido, por todo el apoyo y formación, por ellos he llegado hasta donde estoy. Las metas que he alcanzado y las que conseguiré, son tan mías como tuyas.

A Dessiré, desde que llegó a mi vida me ha inspirado a ser un mejor hombre en todos los aspectos, por medio de su amor, humildad y genialidad. Con ella, el pasado siempre me provoca una sonrisa, el presente me llena de constante alegría y el futuro, me genera una gran esperanza.

A Nicole, como herramienta de motivación para que alcance todos sus sueños y anhelos.

## **Agradecimientos**

A Julia Espinoza, por abrirme las puertas del TEC Digital y por supuesto, por su impulso, guía y constante instrucción durante este proceso final.

A Mario Chacón, por darme la oportunidad de realizar este trabajo y ser fuente de inspiración en mi crecimiento profesional y académico.

A los colaboradores y las colaboradoras del TEC Digital que me ayudaron en este proceso.

A los profesores y las profesoras que me inspiraron y me formaron durante este ciclo de educación superior.

## Resumen

En el presente documento de trabajo final de graduación, se expone la investigación y el diseño de un sistema para el aseguramiento y el control de la calidad en los proyectos de software de la Unidad TEC Digital, instancia del Instituto Tecnológico de Costa Rica.

La calidad es un tema que organizaciones desarrolladoras de software como el TEC Digital, deben abordar para cumplir el objetivo de ofrecer productos y servicios, los cuales satisfagan los requerimientos y las necesidades de sus usuarios finales. Sin embargo, los procesos y las actividades para propiciar dicha calidad, en ocasiones representan elementos con alto costo en cuanto a los recursos necesarios para su implementación y ejecución.

Por este motivo, se plantea el diseño del sistema mencionado, con el fin crear una ruta sistematizada y planificada de las acciones requeridas, las cuales se dividen según las etapas de la metodología de desarrollo de software de la Unidad, para asegurar y controlar la calidad del software que desarrolla esta organización, a la vez que se aprovecha de forma efectiva, los recursos con los cuales cuenta.

El diseño del sistema se propone en función de tres elementos principales: roles, actividades y artefactos. Estos tres componentes desarrollan las interacciones que se plantean para el sistema y, de forma conjunta, despliegan la ruta sistematizada y planificada que se indicó.

Además, se realiza la propuesta de integración de dos elementos para la ejecución del propio sistema: 1) herramientas de software que soporten las actividades planteadas y 2) un modelo de calidad de software para la verificación, validación y evaluación de la misma a través de todo el ciclo de vida de desarrollo de los proyectos de software del TEC Digital.

**Palabras clave:** software, sistema, control de calidad de software, aseguramiento de la calidad, modelo de calidad de software.

## **Abstract**

In the present final graduation document, it is elaborated the design of a system for the assurance and quality control in the software projects of the TEC Digital Unit, instance of the Technological Institute of Costa Rica.

Quality is an issue that software development organizations such as TEC Digital must address to accomplish the goal of offering products and services that meet the requirements and needs of its end users. However, the processes and activities to promote such quality, sometimes represent elements with a lot of cost as far as the necessary resources for its implementation and execution.

For this reason, the design of the mentioned system is proposed to create a systematized and planned route of the required actions, which are divided according to the stages of the software development methodology of the Unit, to ensure and control the quality of the software that this Unit develops, effectively taking advantage of the resources with which it counts.

The design of the system is based on three main elements: roles, activities and artifacts. These three components develop the interactions that arise for the system and jointly develop the mentioned systematized and planned route.

In addition, it is suggested the integration of two elements for the execution of the system: 1) software tools that support the proposed activities and 2) a software quality model for the verification, validation and evaluation of such quality through the entire development cycle of the TEC Digital software projects.

**Keywords:** software, system, software quality control, software quality assurance, software quality model.

## Índice general

Índice de tablas .....	X
Índice de figuras .....	xi
<b>CAPÍTULO I: INTRODUCCIÓN.....</b>	<b>1</b>
1.1. Descripción general .....	1
1.2. Antecedentes .....	5
1.3. Descripción de la organización .....	7
1.4. Trabajos similares.....	12
1.5. Planteamiento del problema .....	18
1.6. Situación problemática .....	20
1.7. Beneficios esperados o aportes del proyecto .....	22
1.8. Objetivos del proyecto .....	25
1.9. Alcance .....	26
1.10. Entregables.....	29
1.11. Supuestos .....	31
1.12. Restricciones .....	32
<b>CAPÍTULO II: MARCO TEÓRICO.....</b>	<b>33</b>
2.1. Proceso de negocio .....	33
2.2. Modelado de procesos de negocio .....	34
2.3. Proyecto.....	36
2.4. Sistema .....	38
2.5. Software.....	38
2.6. Ciclo de vida de desarrollo de software .....	39
2.7. Ingeniería de requerimientos de software .....	41
2.8. Producto de software .....	43
2.9. Calidad.....	43
2.10. Calidad de software .....	46
2.11. Aseguramiento de la calidad de software .....	47
2.12. Modelo de calidad de software .....	53
<b>CAPÍTULO III: DESARROLLO METODOLÓGICO .....</b>	<b>62</b>
3.1. Generalidades de la metodología .....	62
3.2. Alcance de investigación .....	63
3.3. Enfoque de investigación.....	64
3.4. Diseño de investigación .....	67
3.5. Fuentes de recopilación de datos e información .....	70
3.6. Selección de la muestra .....	71
3.7. Métodos y técnicas de investigación en general .....	72
3.8. Procedimientos de recolección – procesamiento de datos e información .....	75
3.9. Instrumentos adicionales de apoyo .....	78
3.10. Etapas de la metodología .....	79
<b>CAPÍTULO IV: ANÁLISIS DE RESULTADOS .....</b>	<b>82</b>
4.1. Formulación y confirmación de base teórica sobre aseguramiento y control de la calidad de software .....	82

4.2. Revisión, análisis y descripción de problemática actual de calidad en los proyectos de software del TEC Digital .....	93
4.3. Elaboración de diseño del sistema para el aseguramiento y el control de la calidad en los proyectos de software del TEC Digital.....	126
4.4. Trabajos futuros .....	137
4.5. Cambio en el trabajo final .....	138
<b>CAPÍTULO V: PROPUESTA DE SOLUCIÓN .....</b>	<b>140</b>
5.1. Aspectos generales del sistema .....	140
5.2. Interacciones de los elementos principales del sistema .....	146
5.3. Integración de software en el sistema .....	155
5.4. Integración de modelo de calidad de software .....	157
5.5. Diseño del sistema .....	158
5.6. Lineamientos y características del sistema .....	163
5.7. Plan de capacitación sobre calidad de software.....	165
<b>CAPÍTULO VI: CONCLUSIONES .....</b>	<b>167</b>
<b>CAPÍTULO VII: RECOMENDACIONES .....</b>	<b>171</b>
<b>GLOSARIO.....</b>	<b>174</b>
Acrónimos.....	174
Abreviaturas.....	174
<b>ANEXOS .....</b>	<b>175</b>
Anexo A. Plantilla de Control de Calidad.....	175
Anexo B. Guía de Programación .....	177
Anexo C. Actualización sobre Plantilla de Control de Calidad .....	193
Anexo D. Plantilla de autoevaluación .....	194
Anexo E. Documento de Arquitectura .....	195
<b>APÉNDICES.....</b>	<b>209</b>
Apéndice A. Plantilla de minuta de reunión del trabajo final de graduación.....	209
Apéndice B. Minuta número uno del trabajo final de graduación .....	210
Apéndice C. Minuta número dos del trabajo final de graduación .....	213
Apéndice D. Minuta número tres del trabajo final de graduación .....	216
Apéndice E. Minuta número cuatro del trabajo final de graduación .....	218
Apéndice F. Minuta número cinco del trabajo final de graduación .....	220
Apéndice G. Minuta número seis del trabajo final de graduación .....	222
Apéndice H. Minuta número siete del trabajo final de graduación .....	224
Apéndice I. Minuta número ocho del trabajo final de graduación.....	226
Apéndice J. Minuta número nueve del trabajo final de graduación.....	228
Apéndice K. Minuta de solicitud de cambio a la Coordinación de Trabajos Finales de Graduación .....	230
Apéndice L. Minuta de solicitud de cambio a los involucrados en el TEC Digital del trabajo final de graduación .....	232
Apéndice M. Cronograma del trabajo final de graduación.....	234
Apéndice N. Mediciones .....	235
Apéndice Ñ. Entrevista elaborada al profesional experto número uno.....	236
Apéndice O. Entrevista elaborada al profesional experto número dos.....	238

Apéndice P. Cuestionario aplicado a colaboradores del TEC Digital .....	240
REFERENCIAS BIBLIOGRÁFICAS .....	245

## Índice de tablas

Tabla 1. Notación de BPMN 2.0 desarrollada por el <i>Object Management Group</i> .....	35
Tabla 2. Resumen de documentos examinados del TEC Digital .....	101
Tabla 3. Descripción y análisis de las causas de la categoría “Roles” .....	117
Tabla 4. Descripción y análisis de las causas de la categoría “Proceso / Actividades” .....	119
Tabla 5. Descripción y análisis de las causas de la categoría “Artefactos” .....	123
Tabla 6. Descripción y análisis de las causas de la categoría “Ambiente” .....	125
Tabla 7. Requerimientos de funcionalidades para herramienta de software del sistema .....	129
Tabla 8. Análisis comparativo de herramientas de software para el sistema .....	131
Tabla 9. Etapa uno del plan de implementación: inventario de artefactos.....	134
Tabla 10. Etapa dos del plan de implementación: preparación de artefactos.....	135
Tabla 11. Etapa tres del plan de implementación: pruebas y sistematización.....	135
Tabla 12. Etapa cuatro del plan de implementación: lineamientos .....	136
Tabla 13. Etapa cinco del plan de implementación: capacitación del TEC Digital.....	136
Tabla 14. Etapas del sistema y referencias teóricas para la preparación de sus artefactos ...	143
Tabla 15. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de iniciación y planeación .....	146
Tabla 16. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de análisis .....	147
Tabla 17. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de diseño .....	148
Tabla 18. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de desarrollo.....	149
Tabla 19. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de pruebas .....	150
Tabla 20. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de capacitación .....	152
Tabla 21. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de cierre de proyecto.....	152
Tabla 22. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de post-liberación del proyecto .....	154
Tabla 23. Propuesta de integración del modelo de calidad de la ISO/IEC 9126 en el sistema	157

## Índice de figuras

Figura 1. Cantidad de proyectos de software del TEC Digital desde su creación .....	5
Figura 2. Organigrama del TEC Digital .....	11
Figura 3. Organización de los elementos de la notación BPMN desarrollada por la OMG.....	34
Figura 4. Los cinco componentes de la administración de proyectos según Turner .....	37
Figura 5. Clasificaciones de los procesos específicos de software según la ISO/IEC 12207 IEEE Std 12207-2008 .....	41
Figura 6. Fases o etapas del ciclo de desarrollo de software según Avella y Gómez .....	51
Figura 7. Las seis categorías y subcategorías de calidad de software de acuerdo con la ISO/IEC 9126 .....	56
Figura 8. Subcaracterísticas de la calidad en uso según la ISO/IEC 9126-1.....	57
Figura 9. Flujos de interacción entre las perspectivas de calidad .....	58
Figura 10. Métricas presentes en la sub característica idoneidad de calidad .....	59
Figura 11. Fases metodológicas del trabajo final de graduación .....	63
Figura 12. Descripción de etapas de un diseño de integración .....	69
Figura 13. Diagrama de índices del marco teórico.....	82
Figura 14. Estado de calidad del software desarrollado en el TEC Digital .....	96
Figura 15. Criterio sobre existencia de situaciones negativas o problemas en la calidad del software del TEC Digital .....	97
Figura 16. Nivel de conocimiento sobre los activos de documentos del TEC Digital.....	99
Figura 17. Actividades del “Analista de Gestión de Calidad” del TEC Digital .....	104
Figura 18. Metodología resumida para el desarrollo de los proyectos de software del TEC Digital .....	108
Figura 19. Modelación del proceso de control de la calidad de software del TEC Digital.....	110
Figura 20. Sección de seguridad y rendimiento de la Plantilla de Control de Calidad .....	113
Figura 21. Sección de hallazgos de la Plantila de Control de Calidad .....	114
Figura 22. Diagrama de Ishikawa de la problemática de calidad en los proyectos de software en el TEC Digital .....	116
Figura 23. Elementos del sistema para el aseguramiento y el control de la calidad en los proyectos de software del TEC Digital .....	141
Figura 24. Sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital .....	159

## CAPÍTULO I: INTRODUCCIÓN

En este capítulo, se describe la información introductoria al trabajo final de graduación; inicialmente, se desarrolla los antecedentes del proyecto y se brinda una contextualización de la entidad donde se efectúa; además, se describe la problemática que origina el proyecto, así como los beneficios que se esperan brindar al TEC Digital (en adelante TD o Unidad) a raíz de la ejecución del proyecto.

De igual forma, se detallan tanto el objetivo general como los específicos del proyecto y su alcance. Finalmente, se indican un supuesto y una restricción que se propiciaron durante el trabajo final de graduación.

### 1.1. Descripción general

El presente documento tiene como objetivo exponer el desarrollo, la metodología, el análisis de los resultados encontrados y los elementos de la propuesta de solución, los cuales surgen a partir de la ejecución de trabajo final, en donde se diseñó un sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD.

El TD es una Unidad del Instituto Tecnológico de Costa Rica (en adelante TEC), adscrita a la Vicerrectoría de Docencia y de acuerdo con Miranda (2014), su objetivo general es *“incorporar las tecnologías de información y comunicación (en adelante TIC) en el desarrollo de las actividades docentes”*; según ese fin, la Unidad ha definido como una de sus principales líneas de acción el desarrollo de software enfocado hacia la implementación de las TIC como herramientas digitales de soporte y ayuda en el ambiente académico superior del TEC.

Actualmente, sobre la línea de acción que se indicó antes, el TD ejecuta el aseguramiento y control de la calidad (en adelante procesos de calidad) de los proyectos de software, según K. M. Gómez (comunicación personal, 02 de febrero, 2017) de una forma empírica; es decir, basados en la experiencia de los colaboradores que han tenido la responsabilidad de ejecutarlos.

El trabajo final de graduación surgió a raíz de la problemática generada en los proyectos de software, de forma específica, en el área de Gestión de la Calidad y sus procesos de calidad.

Así, de acuerdo con K. M. Gómez (comunicación personal, 02 de febrero, 2017), la ejecución de los procesos de calidad, sin contar con una base teórica y objetiva, ha ocasionado consecuencias negativas dentro del ciclo de desarrollo de software del TD. Dentro de estas se podría mencionar: larga duración en ejecución de los procesos de calidad (al menos de cinco a ocho días hábiles) dentro de los proyectos de software, curva de aprendizaje elevada (de seis a nueve meses) sobre forma de ejecución de los procesos de calidad, falta de estandarización en las revisiones de calidad de los proyectos, entre otras.

Con este trabajo se pretende abordar dicha problemática, mediante el diseño de un sistema que establezca una ruta de ejecución e interacción de los procesos de calidad junto a los roles contenidos en el desarrollo de software del TD presente en los proyectos de este tipo.

Este sistema incluye varios componentes de solución, uno de estos es el diseño como tal de la interacción entre roles, actividades y artefactos<sup>1</sup> en el ciclo de vida de desarrollo del software. Otro elemento es la propuesta de integración de futuras herramientas de software para gestionar y ejecutar el sistema. Así también, se plantea el usar un modelo de calidad para medir y evaluar la calidad de los proyectos de software de la Unidad de una forma objetiva.

La propuesta de integración de las herramientas de software se adapta a los requerimientos y las necesidades de la Unidad; pues la mayoría de estas fueron desarrolladas por el mismo TD (aunque para otros fines), solamente, una de las herramientas propuestas para el sistema no se elaboró por la organización. Sin embargo, esta cumple con la característica que busca la Unidad, establecida así por la

---

<sup>1</sup> Término que se desarrolla en el capítulo cinco del presente documento.

Vicerrectoría de Docencia (2010), con el fin de utilizar software libre como propuesta principal de implementación de las TIC en sus actividades.

Este documento presenta once secciones, mediante las cuales se describe el trabajo requerido para el diseño del sistema de los procesos de calidad en los proyectos de software del TD. De dichas secciones, cabe destacar siete: capítulo uno de introducción, capítulo dos de marco teórico, capítulo tres de desarrollo metodológico, capítulo cuatro de análisis de resultados, capítulo cinco de la propuesta de solución y capítulos seis y siete, que muestran las conclusiones y las recomendaciones, respectivamente.

En el primer capítulo se presenta la contextualización general de la entidad donde se desarrolló el proyecto. Este capítulo contiene, además, un estudio de los antecedentes y la situación actual que generó la necesidad de elaborar este trabajo final. Así también, se detallan objetivos, alcance, entregables, un supuesto y una restricción sobre el trabajo final de graduación.

En el capítulo dos, se muestra el fundamento conceptual de todos los temas vinculantes para el trabajo final, los cuales fueron necesarios para el análisis de resultados del mismo y principalmente, para el abordaje de solución planteado.

Seguidamente, en el capítulo tres se expone el marco metodológico que se siguió para obtener los resultados del proyecto. En el mismo se desglosan las generalidades y las etapas de desarrollo del trabajo. Así también, todos los elementos que componen el modelo de ejecución desde el punto de vista metodológico: alcance, enfoque, diseño, métodos, técnicas, fuentes de recopilación de datos e instrumentos adicionales que fueron requeridos.

En el capítulo cuatro, se describe el análisis de los resultados de la aplicación de esta metodología. Esta descripción se construye en virtud de las etapas metodológicas desarrolladas en el capítulo que le precede. De esa forma, se ubica en el capítulo cuatro, lo siguiente: formulación y confirmación de base teórica, problemática presente en la Unidad (revisión, descripción y análisis de la misma) y el plan de implementación

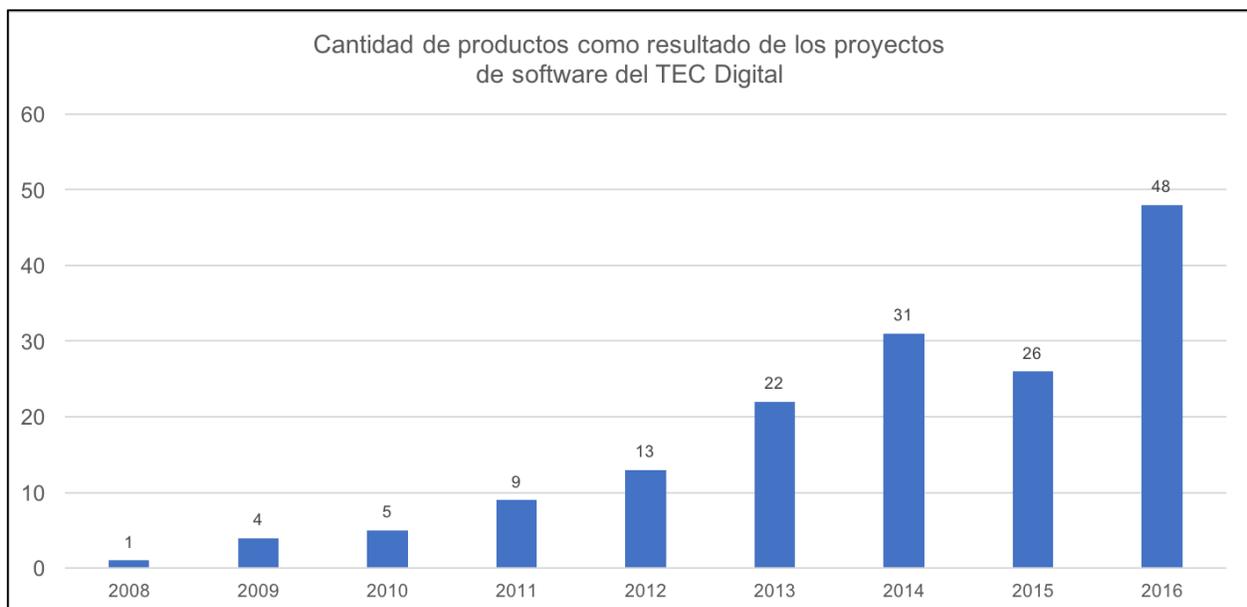
para apoyar la solución del problema. Adicionalmente, se encuentran los trabajos futuros y un cambio realizado sobre el trabajo final, que resultaron del análisis documentado previo a estos.

Posteriormente, en el capítulo cinco se encontrará la descripción específica del abordaje de solución propuesto. Y se finaliza en los capítulos seis y siete, con las conclusiones y las recomendaciones para el TD, respectivamente.

## 1.2. Antecedentes

La cantidad de proyectos de software que se realizan en la Unidad ha venido incrementándose de forma progresiva desde el momento de su creación (ver Figura 1); en el gráfico de columnas mostrado seguidamente, se denota un crecimiento en la cantidad de proyectos desarrollados<sup>2</sup> en el TD desde su año inicial al año 2016; este incremento muestra una tasa promedio de aumento anual de 78 por ciento para los productos de los proyectos de software de esta organización.

Figura 1. Cantidad de proyectos de software del TEC Digital desde su creación



Fuente: Elaboración TEC Digital.

Debido a este crecimiento, existe la posibilidad que para estos proyectos de la Unidad, el control de la calidad que se ejecuta tanto en el ciclo de vida de su desarrollo como para sus productos finales (software, manuales, entre otros), resulte en un conjunto de actividades que cada vez significará una carga mayor de trabajo y esfuerzo en recursos.

---

<sup>2</sup> Desarrollados: se refiere a nuevos proyectos o actualizaciones que se realizaron y tuvieron al menos un producto final de software como resultado o algún cambio sobre un producto final de proyectos anteriores.

Por lo mencionado anteriormente, se podría indicar entonces, que la rapidez de ejecución del proceso de control de la calidad es una característica del mismo, que de forma probable, se convertiría vital para la liberación<sup>3</sup> continua de los productos finales de dichos proyectos.

Además, la calidad de los proyectos de software desarrollados por la Unidad y sus productos finales, podría implicar de forma directa, en el nivel de satisfacción percibida por los usuarios finales del software. Estos usuarios son todos los estudiantes, profesores y funcionarios del TEC (en adelante comunidad TEC).

Debido a esto, la calidad mencionada y los procesos relacionados que se ejecutan sobre los proyectos de software de la Unidad, de acuerdo con M. Chacón (comunicación personal, 01 de febrero, 2017), son aspectos que el TD está en búsqueda de reforzar y aplicar una mayor eficiencia, en términos de recursos de tiempo y personal.

El presente trabajo final surge de esta forma, como un esfuerzo puntual enfocado hacia el cumplimiento de ese objetivo en el TD, con la meta final de apoyar el mejoramiento de los procesos de calidad en el ciclo de vida de desarrollo de los proyectos de software y los resultados generales de este.

---

<sup>3</sup> Liberación: etapa del ciclo de vida de desarrollo de software en la que se realiza la entrega del software a los usuarios finales que le darán uso al mismo.

### 1.3. Descripción de la organización

El TD es la Unidad que soporta la plataforma de aprendizaje electrónico del TEC, el tecDigital<sup>4</sup>. Esta plataforma hace posible la interacción entre los usuarios de la comunidad TEC. La misma presenta software y aplicaciones que ofrecen apoyo a la docencia, las labores administrativas y los estudiantes en general.

Seguidamente, se describen los hitos principales, de acuerdo con Chacón y Garita (2010), que conforman las etapas de creación de la Unidad y su plataforma de aprendizaje electrónico como tal:

1. Conceptualización del proyecto: de agosto a diciembre de 2007.
2. Plan piloto de la plataforma: de febrero de 2008 a febrero de 2009.
  - Conformación de equipo: semestre I de 2008.
  - Integración de aplicaciones y capacitación: semestre II de 2008.
3. Implantación a nivel institucional de la plataforma: de febrero a diciembre de 2009.
4. Consolidación de la plataforma: de enero de 2010 a marzo de 2013.
5. Conformación de la Unidad: de abril de 2013.

Además, Miranda (2014) describe siete líneas de acción que reúnen las áreas de desarrollo del TD:

1. Coordinación y gestión: se encarga de efectuar las tareas relacionadas con el control y la ejecución del presupuesto, coordinación entre las diferentes secciones del proyecto, adquisición de equipo e implementación del mismo para apoyo de las escuelas del TEC.
2. Investigación: se definen áreas de interés en investigación y cómo estas a su vez orientan las acciones del equipo de trabajo.

---

<sup>4</sup> tecDigital: se refiere a la plataforma de aprendizaje digital que brinda la Unidad TEC Digital.

3. Desarrollo de proyectos de programación: se desarrolla software a la medida, contemplando sus diferentes etapas: levantamiento de requerimientos, diseño, programación, pruebas e implantación. Estos proyectos atienden necesidades institucionales, innovación en aprendizaje electrónico y se integran con aplicaciones externas (*CmapServer*, *Pentaho* y otras).
4. Soporte a usuarios: se brinda apoyo a la comunidad TEC sobre la plataforma de aprendizaje electrónico, el uso de la misma, sus componentes e incidentes relacionados a esta. Esta plataforma está basada en *.Learn, Research, Network™* (en adelante *.LRN™*), entorno de software libre utilizado a nivel mundial, especialmente, por comunidades digitales y organizaciones que buscan implementar soporte digital para el aprendizaje electrónico (*.Learn, Research, Network™*, 2017).
5. Creación de contenidos de aprendizaje electrónico: servicio orientado a la creación de cursos académicos aplicando un proceso, originado por el diseño instruccional y generando objetos de aprendizaje con estándares internacionales (*IMS-LD, SCORM*, entre otros), los cuales son compartidos con investigadores y docentes a nivel mundial.
6. Componente pedagógico y capacitación: se elabora un modelo pedagógico de aprendizaje electrónico y por medio del desarrollo de cursos se capacita a los profesores en el uso de la plataforma y la generación de competencias para este tipo de aprendizaje y enseñanza.
7. Soporte a bases de datos y servidores: se efectúan labores de investigación, afinamiento y monitoreo para mantener la plataforma de aprendizaje electrónico actualizada y en buen funcionamiento. Se realizan respaldos diarios sobre las bases de datos, así también, tareas de monitoreo constante sobre los discos duros de los servidores.

Adicional a la actividad de soportar la plataforma mencionada y realizar las tareas y las funciones que se detallaron sobre las líneas de acción, el TD ejecuta

proyectos de investigación y desarrollo, los cuales resulten en abordajes puntuales sobre situaciones de problemáticas sociales, por ejemplo: el Traductor LESCO, primer traductor digital a nivel costarricense de lenguaje escrito a LESCO<sup>5</sup>. Estos proyectos se destacan por su naturaleza innovadora, en búsqueda del beneficio de la comunidad académica superior y también de la sociedad costarricense.

### **1.3.1. Misión**

“Promover el uso y aprovechamiento de las TIC en el ámbito docente-educativo, en el TEC, con el objetivo de mejorar la calidad en el ámbito universitario; proveyendo servicios de soporte, capacitación, desarrollo e innovación en TIC para la educación”. (Miranda, 2014)

### **1.3.2. Visión**

“Ser una entidad consolidada en el área de tecnologías educativas con altos niveles de excelencia e innovación, y un gran reconocimiento a nivel institucional, nacional e internacional”. (Miranda, 2014)

### **1.3.3. Valores**

De acuerdo con Miranda (2014), la organización posee los siguientes valores:

- Integridad profesional.
- Vocación de servicio y colaboración.
- Excelencia.
- Espíritu innovador.
- Compromiso en el quehacer.

---

<sup>5</sup> LESCO: acrónimo para LEngua de Señas de COstarricense.

#### **1.3.4. Organigrama**

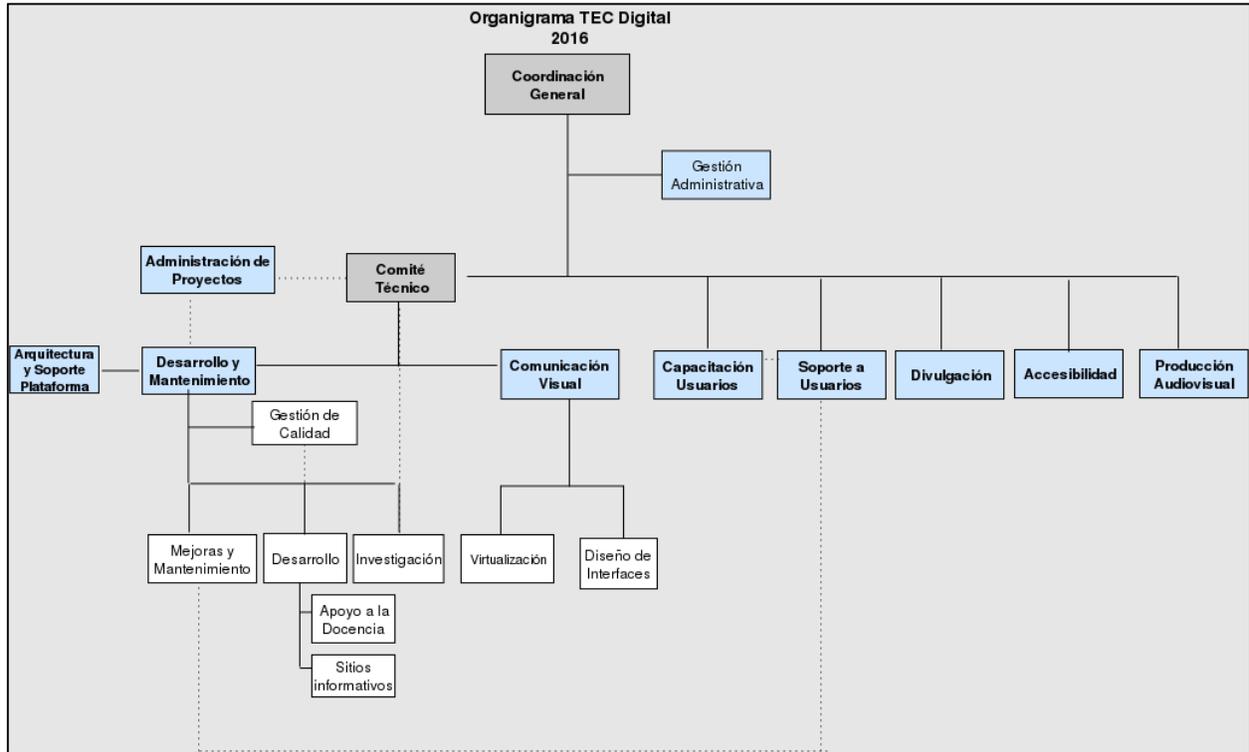
El TD cuenta con varias áreas de trabajo que componen los equipos de cada proyecto de desarrollo de software de forma dinámica, de acuerdo con las características y las necesidades de cada uno de estos proyectos.

El enfoque del trabajo final de graduación descrito en el documento, se centra en el área de Gestión de la Calidad (específicamente, para los proyectos de desarrollo de software) de la Unidad. En la actualidad, esta área se compone de un colaborador, el cual realiza las actividades del proceso de control de la calidad de software en los proyectos de software de la TD.

Además, para el proyecto se tomó en cuenta la participación de las áreas de Coordinación General, Comunicación Visual, Capacitación de usuarios (en adelante Capacitación) y el equipo de Desarrollo (en adelante Equipo Desarrollador); dado que forman parte del ciclo de vida de progreso de los proyectos de software de la Unidad.

De seguido, en la Figura 2 se presenta la distribución organizacional del TD para el 2016:

Figura 2. Organigrama del TEC Digital



Fuente: Elaboración TEC Digital.

## **1.4. Trabajos similares**

Anterior a este trabajo final de graduación, tanto a nivel interno del TD como externo, se han realizado trabajos, investigaciones o esfuerzos que muestran algún tipo de vínculo con el proyecto; estos fungen de entradas y base para el desarrollo del mismo. Estos se muestran enseguida.

### **1.4.1. Trabajos similares internos**

El primer esfuerzo realizado por parte de la Coordinación General de la Unidad, de acuerdo con K. M. Gómez (comunicación personal, 02 de febrero, 2017), fue asignar a una persona a cargo de la gestión de la calidad en esta entidad en agosto de 2013, con sus funciones y responsabilidades definidas. Donde su primera directriz fue definir mecanismos y técnicas para evaluar la calidad de los proyectos de software.

Dentro de esos mecanismos y técnicas, a nivel interno del TD se realizaron tres esfuerzos puntuales, información que se recabó con la ayuda de K. M. Gómez (comunicación personal, 02 de febrero, 2017). Estos esfuerzos se resumen en la creación de una plantilla de control de la calidad para los proyectos de software de la Unidad y, de forma posterior, la adición de más aspectos de revisión y escalas de medición sobre los procesos de control de la calidad. Así también, se elaboró un plan de aseguramiento de la calidad para la organización. Estos esfuerzos se profundizan inmediatamente.

#### **1.4.1.1. Creación de Plantilla de Control de la Calidad**

En agosto de 2013, la colaboradora encargada en ese periodo del área de Gestión Calidad en el TD, recopiló en Internet plantillas de control de calidad y elaboró una plantilla adaptada para los proyectos de desarrollo de software de dicha organización, esta plantilla lleva como nombre Documento de Control de Calidad y se documenta para cada uno de estos proyectos; en la organización se le conoce como Plantilla de Control de la Calidad.

Esta plantilla se implementa como un documento de hoja de cálculo (ver anexo A) en la herramienta *Google Drive*, donde se crea, se almacenan y se trabaja de forma colaborativa documentos de distintos formatos (Google, 2017a); el diseño de este

archivo está enfocado en aspectos de revisión de calidad, sobre los proyectos de desarrollo de software implementados en la plataforma de aprendizaje electrónico de la Unidad.

La plantilla mencionada contiene las siguientes secciones (dentro del documento se refleja en pestañas de la hoja de cálculo) para cada instancia de su documentación en los proyectos de software:

- Generalidades: se presenta información general del proyecto, así como un resumen redactado del estado de las pruebas sobre el mismo.
- Plan de Pruebas: el equipo desarrollador efectúa este plan donde se incluyen las entradas, las actividades con esas entradas y el resultado final que debería tener cada una de las actividades; es decir, los flujos del software. También se detalla problemas, errores o comentarios asociados con cada actividad.
- Hallazgos: se describe los hallazgos relacionados con un componente o actividad, así también el tipo de hallazgo (error, mejora o duda) y la prioridad de cada uno de estos (alta, media, baja).
- Portal TD: se describe el contenido mínimo que deben tener los portales de los proyectos, los cuales se encuentran ubicados dentro de la misma plataforma del TD, así también presenta el resultado de la comprobación de dicho contenido.
- Compatibilidad: se detalla las pruebas que se deben realizar sobre la compatibilidad del proyecto con diferentes ambientes web (navegadores).
- Calidad de los paquetes de software: se muestra la estructura y las características que deben contener los paquetes de software de los proyectos de desarrollo, definidas previamente, en la Guía de Programación del TD (ver anexo B).
- Interfaz/Navegación: describe las pruebas y las características que deben contener los componentes de la interfaz gráfica de usuario de los proyectos de software.

- Sección de seguridad: describe las pruebas sobre aspectos de seguridad que se deben realizar a los proyectos de desarrollo de software del TD.

Las pruebas contenidas en esta plantilla las ejecuta el colaborador encargado de Gestión de Calidad de la Unidad. Los resultados se comunican al equipo desarrollador y se inicia un proceso iterativo entre nuevas revisiones y correcciones, hasta que se resuelven los hallazgos y, de ese modo, se aprueba la liberación de los proyectos de software a los distintos ambientes de desarrollo.

#### **1.4.1.2. Adición de aspectos y escalas en Plantilla de Control de la Calidad**

En el 2014, otra colaboradora asumió el área de Gestión de Calidad en el TD y en el periodo comprendido entre abril y octubre del mismo año, efectúa una actualización de la Plantilla de Control de la Calidad (ver anexo C) y adiciona las escalas dentro de las secciones, las cuales se siguen utilizando a la fecha actual:

- Escalas: introduce escalas en las diferentes secciones de la Plantilla de Control de la Calidad de los proyectos de desarrollo de software del TD. Estas escalas son de rangos numéricos y los mismos varían de sección a sección.

Además de la adición de estas escalas, se realizaron varias plantillas de auto-calificación (ver anexo D) para los equipos desarrolladores, donde debían completar el cumplimiento que seguían respecto a una serie de aspectos de revisión y características definidas de calidad . Estas plantillas también se desarrollaron en una hoja de cálculo de la herramienta *Google Drive*.

#### **1.4.1.3. Plan de aseguramiento de la calidad de software**

Entre los años 2015 y 2016, el licenciado Luis Alexander Calvo Valverde, como trabajo final de graduación para la licenciatura en Ingeniería Informática y Calidad de Software, realizó un “Plan de Aseguramiento de la Calidad para reducir el número de errores detectados en el entregable final en el desarrollo de paquetes de software en el TEC-Digital”, el cual se centró en la gestión y la trazabilidad de los requerimientos de los proyectos de software de la Unidad.

Para la elaboración del plan, se utilizó el estándar IEEE 730-2014, el cual define una serie de procesos, que son aplicables en una organización dedicada al desarrollo de software, buscando dotar de un enfoque de aseguramiento de calidad a la misma. Como se menciona en el párrafo anterior, este plan se desarrolló principalmente, en torno a la la gestión de requerimientos de los proyectos de software.

El resultado central de ese proyecto, sobre aseguramiento de la calidad de software, fue el desarrollo de una primera iteración del plan y la elaboración de una herramienta de gestión y trazabilidad de requerimientos de proyectos del TD; la misma se desarrolló mediante una hoja de cálculo de *Google Drive*.

Actualmente, de acuerdo con K. M. Gómez (comunicación personal, 02 de febrero, 2017), en la Unidad no se ha implementado este plan de aseguramiento, por la complejidad técnica que se presenta en la herramienta propuesta; asimismo, la organización no cuenta aún con la madurez organizacional, en términos de procesos de calidad y actividades derivadas de los mismos para implantarlo.

#### **1.4.2. Trabajo similar externo**

Por otro lado, se indagó sobre trabajos o investigaciones externas al TD que tuvieran relación directa con el tipo de trabajo final de graduación que se desarrolló, así como con el tipo de planteamiento de solución buscado y el estado actual de la Unidad en la gestión de la calidad del software.

A pesar de que se encontraron diversos estudios (Ejaz, Nazmeen y Zafar, 2010; Elberzhager, Munch, Rombach y Freimut, 2011; Hampp, 2012; Lavallée y Robillard, 2011; Lochmann, Ramadani y Wagner, 2013; Mehdi y Aharpour, 2011; Nazir, 2005; Saif, Khan y Arif, 2010) sobre calidad de software con temas y propuestas relacionadas, el siguiente trabajo es el único que muestra un vínculo directo; pues se presenta una similitud en cuanto al abordaje de solución, donde se buscó un conjunto de actividades para asegurar y controlar la calidad, por medio de las etapas del ciclo de vida de desarrollo del software, para incorporar estándares o mejores prácticas para la validación, la verificación y la evaluación de la calidad del mismo, así como incluir un modelo de calidad en esas actividades. Seguidamente, se describe dicho trabajo.

#### **1.4.2.1 Metodología integrada para la aplicación de inspecciones y pruebas de software**

En el 2011, Clara Patricia Avella Ibáñez y Juan Federico Gómez Estupiñán, investigadores del programa de Ingeniería de Sistemas de la Universidad de Boyacá (Colombia), publican un artículo donde se expone un resumen de una investigación que desarrollaron en la línea de calidad de software.

Dicha investigación contiene el desarrollo de una metodología que integra conceptos y actividades de aseguramiento y control de la calidad, junto con elementos de revisión e inspección en un solo método, el cual busca la calidad en los proyectos de software.

La investigación surge, de acuerdo con Avella y Gómez (2011), *“al observar que en los procesos de desarrollo de software se aplicaban pruebas e inspecciones como tareas puntuales e independientes, pero no se evidenciaba una metodología que integrara estas actividades”*. (p. 269).

La metodología propuesta por Avella y Gómez, contiene como base, la utilización de estándares desarrollados por el *Institute of Electrical and Electronics Engineers* (en adelante IEEE), organización técnica de profesionales más grande a nivel mundial dedicada al avance tecnológico para el beneficio de la humanidad (IEEE, 2017) y también por la *International Organization for Standardization* (en adelante ISO), la cual es una organización internacional, independiente, no gubernamental y con una membresía de 162 cuerpos de estándares internacionales para múltiples áreas de conocimientos (ISO, 2017).

Los estándares o normas vinculantes para el trabajo final de graduación, las cuales presentan Avella y Gómez (2011), se detallan a continuación:

- IEEE Std 1028-2008.
- IEEE Std 829-2008.
- ISO/IEC 9126.

En el trabajo de investigación de dichos investigadores, se realiza una especificación de documentos (en el capítulo cinco se les dará el nombre de artefactos) y actividades por etapa del ciclo de vida de desarrollo de los proyectos de software. Estos documentos y actividades son, de acuerdo con los autores de la investigación, un acercamiento a la integración de varios cuerpos de conocimiento y metodologías sobre aseguramiento y control de la calidad.

### **1.5. Planteamiento del problema**

Actualmente, el TD cuenta con una arquitectura para el desarrollo de los proyectos de software, la cual consiste en tres ambientes principales: Desarrollo, Pruebas y Producción. Estos ambientes se reflejan en servidores separados donde los equipos desarrolladores instalan sus paquetes de software, según el estado en el ciclo de vida de desarrollo en el que se encuentren. Este estado lo determinan las aprobaciones brindadas por parte del proceso de control de la calidad, proceso que ejecuta el colaborador de Gestión de Calidad de la Unidad.

En el proceso de control de la calidad de los proyectos de software específicamente, se realiza una serie de pruebas en varios escenarios sobre los tres ambientes mencionados de forma anterior; todo esto basado en el plan de pruebas que detallan los equipos desarrolladores en la Plantilla de Control de la Calidad, la cantidad de pruebas y los escenarios requeridos son determinados y ejecutados por el colaborador de Gestión de Calidad, quien administra el proceso.

De manera resumida, el proceso de control de la calidad inicia con la solicitud para pruebas en el ambiente de Pruebas, posteriormente, cuando el software haya superado todas las revisiones y las pruebas a las cuales fue sometido, se migra al ambiente de Producción, con el requisito aprobado por el colaborador administrador del proceso. A continuación, se profundiza dicho proceso.

Todo proyecto de software de la Unidad debe pasar por este proceso de control de calidad, para buscar liberar el software en el ambiente final: Producción. Para esto se efectúa una primera solicitud al encargado de Gestión de Calidad y, de esa forma, se inicia una instancia del proceso; esta solicitud se realiza cuando los equipos desarrolladores lo consideren pertinente.

Esta solicitud de revisión se efectúa mediante un correo electrónico que se le envía al encargado de Gestión de Calidad. Las revisiones se ejecutan ya sea para un nuevo proyecto de software o un proyecto con algún cambio y se modificó su paquete de software.

De esa solicitud en adelante, la persona encargada del proceso de Control de la Calidad y el equipo desarrollador, inician una constante comunicación e interacción para que el proceso de control de la calidad se realice.

Esta interacción entre el equipo desarrollador y el encargado de Gestión de la Calidad de la Unidad consiste en lo siguiente: una vez que el equipo desarrollador registra el plan de pruebas y efectúa la solicitud de revisión, el encargado del proceso inicia la revisión de calidad del software con la herramienta principal: la Plantilla de Control de la Calidad; cuando se finaliza la revisión de las actividades del plan de pruebas, dicho encargado envía un correo electrónico de respuesta a la solicitud inicial y el equipo desarrollador se responsabiliza de ofrecer respuesta a lo documentado por el encargado de Gestión de la Calidad y resolver los hallazgos pertinentes.

Este proceso es iterativo desde que se encuentran los hallazgos en cada proyecto hasta que se resuelven por parte del equipo desarrollador; una vez el equipo desarrollador responde a los hallazgos, se revisa de nuevo para corroborar la resolución de los mismos; así de forma continua hasta que el encargado de Gestión de la Calidad considere que los hallazgos correspondientes han sido resueltos.

El proceso de calidad termina cuando el encargado de Gestión de la Calidad del TD aprueba el proyecto y el software (producto final) es liberado en el ambiente de Producción. La aprobación se muestra de forma explícita a los encargados de la arquitectura y servidores de la plataforma de aprendizaje electrónico de la Unidad, mediante el envío de un correo electrónico con copia al equipo desarrollador involucrado. Después de esto, no se tiene control o seguimiento sobre el uso y la percepción de calidad que se tenga del software, por parte de los usuarios finales.

## **1.6. Situación problemática**

Anteriormente, se realizó una contextualización mediante el planteamiento del problema en la forma de ejecución actual del proceso de control de la calidad en los proyectos de software del TD. Como resultado de la ejecución de este proceso, surgen los problemas específicos que se detallan a continuación:

### **1.6.1. Desalineación del control de la calidad con base teórica**

El proceso actual de control de la calidad no tiene una base teórica de acuerdo con mejores prácticas, metodologías, normas o marcos de referencias relacionados a la calidad en el desarrollo de software; por lo tanto, no se cuenta con un sistema establecido formalmente, que brinde un fundamento objetivo y estándar sobre el cual determinar la calidad de un proyecto de desarrollo de software de forma certera.

### **1.6.2. Control de la calidad sin aseguramiento**

A pesar que el TD cuenta con el proceso de control de la calidad en el desarrollo de los proyectos de software, se ha encontrado la situación actual que la única actividad, la cual permite ese control buscado con respecto a ciertos atributos de calidad en los proyectos, se realiza una vez estos proyectos se desarrollan.

Esto significa que el proceso no abarca una ruta sistematizada para todo el ciclo de desarrollo del software en la organización; no se toma en cuenta los pasos o etapas iniciales como son: iniciación de los proyectos, levantamiento de requerimientos, análisis y diseño de las soluciones; así como otros aspectos, fases o actividades de los proyectos, las cuales deberían tener un vínculo con un enfoque de calidad, para no solamente, controlar, sino asegurar la calidad en los proyectos de software de la Unidad.

### **1.6.3. Carencia de automatización en el control de la calidad**

El proceso de control de la calidad se efectúa de forma manual; es decir, no hay utilización de herramientas de software que realicen automáticamente, actividades del mismo; esto genera que el proceso requiera de un tiempo excesivo para su ejecución; por lo tanto, se ha convertido en una actividad que ralentiza los proyectos de software.

Además, con esta característica manual del proceso, se crea el riesgo de omisiones de pasos o actividades durante la ejecución del mismo, las cuales podrían obedecer a la mayoritaria intervención del factor humano dentro de este proceso.

#### **1.6.4. Desconocimiento de estado de calidad de los proyectos de software a nivel interno**

Actualmente, el TD carece de datos y estadísticas, los cuales permitan identificar el valor real del proceso y la correcta ejecución del mismo. Algunos ejemplos de posibles datos y estadísticas, son: cuántas veces un proyecto ha sido parte del proceso de control de la calidad, cuántas revisiones se han efectuado por ambiente y proyecto, cantidad de hallazgos y tipos de estos por revisión, cantidad de errores encontrados en Producción después de haber sido liberado el producto final de software en ese ambiente, entre otras.

Por esto, el proceso actual de control de calidad no cuenta con un medio o herramienta que permita conocer el estado general o métricas de calidad de los proyectos de desarrollo de software, para ninguna etapa de ese ciclo. Además, no se tiene una trazabilidad de qué proyecto y sus versiones se encuentran en cada ambiente.

#### **1.6.5. Desconocimiento sobre el nivel de calidad percibido por usuarios finales de los proyectos de software**

El proceso actual de control de calidad no permite conocer el nivel de calidad percibido por los usuarios finales, quienes utilizan los productos de los proyectos de software de la Unidad, una vez liberados estos. Esto se debe a que no presenta algún medio, herramienta, proceso o actividad, el cual brinde soporte a ese fin. Por lo tanto, actualmente, el TD carece de una medición puntual del nivel de calidad percibido de sus proyectos de software por parte de la comunidad TEC.

## **1.7. Beneficios esperados o aportes del proyecto**

Con los elementos planteados en la propuesta de solución del presente trabajo final, se pretende dotar a la Unidad de los beneficios que se detallan seguidamente.

### **1.7.1. Propuesta con fundamentos teóricos**

Generar una propuesta de solución a la problemática, con base en la teoría de mejores prácticas, metodologías, normas o marcos de referencias relacionados a la calidad en el desarrollo de software, la cual se espera se convierta en el fundamento para la alineación del control de la calidad y la implementación del aseguramiento de la calidad. Un ejemplo de esto, se observa en la propuesta de solución, mediante la integración de un modelo de calidad de software basado en la ISO/IEC 9126.

### **1.7.2. Diseño de sistema y plan de implementación para el aseguramiento y control de la calidad de software**

Se pretende mejorar el estado de los procesos de calidad, específicamente, en el área de desarrollo de proyectos de software, por medio del diseño del sistema, el cual incluya: procesos de calidad, actividades, roles y artefactos. Así también, se propone un plan para implementar dicho proceso. Esta implementación se excluye del alcance del trabajo final de graduación.

Este diseño del sistema pretende incluir una ruta de aseguramiento de la calidad del software aunado al control de esta que ya se ejecuta, para velar así por la gestión de la calidad de los proyectos de software del TD desde el momento de su iniciación y aún después de su liberación. En resumen, se busca proponer un enfoque integral de calidad en la Unidad durante el ciclo de vida de desarrollo de los proyectos.

### **1.7.3. Automatización de procesos de calidad**

Para ejecutar los procesos de calidad y sus actividades respectivas que son propuestas, en los proyectos de desarrollo de software del TD, se propone integrar herramientas de software, con lo cual se pretende agilizar las actividades de estos procesos para aumentar la eficiencia de los mismos. El implementar las herramientas se excluye del trabajo final de graduación.

Se espera que dicha eficiencia permita al personal de Gestión de la Calidad del TD un mayor aprovechamiento del recurso de tiempo para la ejecución de los procesos de calidad y, a su vez, se acorte el tiempo de ejecución de los proyectos de software.

Además, otro beneficio esperado con esta automatización, es la reducción del riesgo de omisión de pasos o evaluaciones dentro de los procesos de calidad del TD, los cuales resultan del accionar humano dentro de los mismos. Esto traería a la ejecución del sistema características como: repetibilidad, constancia, estandarización, entre otras.

#### **1.7.4. Conocimiento de estado de calidad de los proyectos de software**

A través de la propuesta de integración de herramientas de software y con los datos obtenidos de estas (una vez que se implementen de forma posterior al trabajo final), la Unidad tendrá un conocimiento del estado de calidad de los proyectos de desarrollo de software. Dentro de los datos, la información y las estadísticas que se podrían obtener cabe mencionar, entre otras:

- Registro de las versiones de proyectos.
- Cantidad de revisiones por cada una de estas versiones y en total por aplicación o software.
- Calificación automática de calidad según rúbricas de evaluación.
- Cantidad de hallazgos de cada una de estas versiones (encontrados, resueltos, reincidentes, sin resolver, entre otros estados).
- Cantidad de proyectos que se encuentran en los procesos de calidad por cada ambiente.
- Promedio de hallazgos por equipo desarrollador, proyecto y, en general, aplicación o software.
- Promedio de tiempo de resolución de hallazgos.
- Bitácora automática de los procesos de calidad.

Estos datos e información permitirán obtener un conocimiento claro del estado de calidad de los proyectos de software de la Unidad, con ello se brindaría así, la capacidad de toma de decisiones basada en información certera y real.

#### **1.7.5. Conocimiento sobre nivel de calidad percibido por usuarios finales de los proyectos de software**

Se busca dotar al TD de mecanismos específicos de retroalimentación, mediante la inclusión de actividades y artefactos dentro del sistema propuesto, los cuales permitan medir y categorizar el nivel de calidad percibido por los usuarios finales, quienes hacen uso de los proyectos de software desarrollados por la Unidad.

## **1.8. Objetivos del proyecto**

### **1.8.1. Objetivo general**

- Diseñar un sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD.

### **1.8.2. Objetivos específicos**

1. Examinar las mejores prácticas, normas, marcos de referencia y metodologías de aseguramiento y control de la calidad de software para la adquisición de una base teórica de gestión de la calidad en proyectos de desarrollo de software.
2. Analizar la brecha existente entre la base teórica adquirida de gestión de la calidad, respecto a la ejecución de los procesos de aseguramiento y control de la calidad en los proyectos de desarrollo de software del TD.
3. Determinar el alcance de solución aplicable a la situación que se presenta actualmente, en la gestión de la calidad de los proyectos de software del TD.

## **1.9. Alcance**

El trabajo final de graduación descrito en el presente documento consiste en realizar el diseño de un sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD; se toma como base un estudio de mejores prácticas, normas, estándares, marcos de referencia y metodologías de calidad de software.

El estudio que se menciona, busca proveer a la Unidad del conocimiento teórico necesario para incluir el aseguramiento y el control de la calidad en el ciclo de vida de desarrollo de sus proyectos de software. Dicho conocimiento permitirá la aplicación, dentro del sistema y los procesos de calidad, de aspectos y métricas de calidad de software, tales como: seguridad, accesibilidad, integración, usabilidad, compatibilidad, rendimiento y otras características más. Todo esto con el fin de asegurar el pleno cumplimiento de las necesidades y los requerimientos que se tienen de los usuarios finales para los proyectos de software de la Unidad.

Como parte de los elementos del desarrollo metodológico que se utilizaron para alcanzar esas metas, los cuales se mencionan anteriormente, se encuentran, entre otros: dos entrevistas a profesionales expertos en el área de la calidad de software, el análisis de activos de la Unidad (datos, estadísticas y demás información) relacionados a la gestión de la calidad de software de la entidad, así como un cuestionario que se aplicó a diez colaboradores vinculados a dicha gestión en el TD y una descripción del proceso de control de calidad de software, mediante un modelado de proceso de negocio.

Los elementos mencionados anteriormente, son los insumos utilizados para realizar una descripción de la problemática actual en la gestión de la calidad del software en la Unidad; esta se elaboró por medio de dos instrumentos de análisis: un diagrama de Ishikawa, donde se buscó evidenciar las principales causas de la problemática mencionada y, por otra parte, el estudio analítico de las brechas que se generan a través de esas causas, entre la gestión de calidad de software que presenta el TD y el estudio de la base teórica obtenida sobre aseguramiento y control de la calidad.

Estos instrumentos de análisis significaron herramientas esenciales para el desarrollo de la propuesta de solución del presente trabajo final, así como para definir sus componentes, alcance y plan de implementación dentro de la Unidad. Esta propuesta de solución fue revisada y aprobada por representantes de las áreas con las cuales cuenta el TD y tienen relación directa o indirecta con el ciclo de vida de desarrollo de los proyectos de software de la entidad.

Como se mencionó antes, se elabora también un plan de implementación de la propuesta de solución; pues esta supone una serie de elementos que incidirán en el TD y, de acuerdo con L. A. Calvo (comunicación personal, 4 de abril, 2017), para que la transición entre estados de una organización resulte efectiva y se produzcan resultados con verdadero valor práctico, se debe realizar un cambio progresivo, con objetivos claros y realistas.

Por otra parte, el sistema que se diseña como propuesta de solución para el aseguramiento y el control de la calidad en los proyectos de software del TD, consiste básicamente, en la descripción de la interacción planteada a través de tres elementos principales: roles, actividades y artefactos. De tal forma que se documentan, mediante las fases del ciclo de vida de desarrollo de los proyectos mencionados, el rol o los roles que ejecutan ciertas actividades, con las cuales se utilizan o se crean determinados artefactos.

Además, para la ejecución efectiva de dicho sistema, se propone integrar un conjunto de herramientas de software a los procesos de calidad, de tal manera, se realice el registro y el soporte de las actividades de evaluación y documentación de los procesos mencionados, una vez se implemente la propuesta.

Con dicha integración de herramientas de software, se busca que en un futuro la Unidad disponga de datos útiles para el desarrollo exitoso de sus proyectos, de tal forma, se apunta a maximizar el aprovechamiento de los recursos para la gestión de la calidad de software de la Unidad.

También, mediante la propuesta de integración mencionada anteriormente, se plantea además, incorporar funcionalidades en la ejecución del sistema como, entre otras: gestión de las bitácoras de los proyectos en cuanto al estado de calidad de los mismos de forma centralizada y automática, soporte de registro (rúbricas de evaluaciones, planes de pruebas, entre otros) de las actividades de los procesos de calidad de los proyectos de software del TD, desde el inicio de los proyectos hasta después de la liberación del producto de software y es utilizado por los usuarios finales.

El software que se propone implementar fue analizado y seleccionado según las políticas y las líneas de trabajo de la Unidad, las cuales buscan utilizar software libre y código abierto como primera opción y como medio para introducir de forma exitosa las TIC al ambiente de docencia y aprendizaje, según M. Chacón (comunicación personal, 01 de febrero, 2017).

Otro elemento importante de mencionar en el presente alcance del trabajo, es la selección de un modelo internacional de calidad de software para incorporarlo al sistema propuesto, buscando introducir una fundamentación sistematizada y objetiva para la validación, la verificación y la calificación del estado de calidad de los proyectos de software del TD.

De esta manera, dentro del alcance del presente proyecto, se incluye lo descrito anteriormente; sin embargo, se excluye implementar el sistema como tal, así también la implementación y la integración de las herramientas de software mostradas en la propuesta de procedimiento y la introducción del modelo de calidad de software. El resultado final de todos los entregables se propone enseguida.

## **1.10. Entregables**

El presente trabajo final de graduación cuenta con tipos de entregables, tanto de gestión como de productos. Seguidamente, se desglosan cada uno de estos.

### **1.10.1. Gestión del proyecto**

Para un adecuado control del presente trabajo final, se entregan los documentos de gestión que se detallan de seguido:

#### **1.10.1.1. Minutas**

Se realizaron minutas (basadas en la plantilla del apéndice A) de las reuniones con los encargados e involucrados del presente trabajo final de graduación, con la idea de tener respaldos de los contenidos discutidos; dentro de estas minutas se presenta el avance del proyecto en cuanto a objetivos propuestos, cambios y metas alcanzadas. Dichas minutas se muestran del apéndice B al apéndice J.

#### **1.10.1.2. Cronograma de proyecto**

El cronograma de proyecto, el cual se visualiza en el apéndice M, fue una herramienta de apoyo clave para las fechas de entrega de los productos; además, fungió como un documento de control de avance.

### **1.10.2. Entregables de producto**

En esta sección se describe los entregables, los cuales resultan como productos del presente trabajo final de graduación. Seguidamente, se detallan:

#### **1.10.2.1. Base teórica sobre aseguramiento y control de la calidad de software**

Se produce para el TD un marco teórico de buenas prácticas, estándares internacionales, marcos de referencia, entre otros, que funge como base de consulta para la conceptualización de elementos y temas relacionados a la calidad de software, gestión, aseguramiento y control de la misma.

#### **1.10.2.2. Análisis de problemática actual de calidad de software en el TD**

Cabe destacar el resultado del análisis que se efectúa sobre la problemática en la gestión de la calidad de software de la Unidad, la cual se desarrolla mediante

elementos metodológicos de consulta, especificación de hallazgos, contraste con lo mostrado en la base teórica y síntesis de los mismos.

#### **1.10.2.3. Diseño del sistema para los procesos de calidad del TD**

La base del trabajo final de graduación es el sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD, el cual comprende la descripción de los elementos que lo componen y la interacción que se genera entre estos (roles, actividades y artefactos), así como la propuesta de integración de herramientas de software y de un modelo de calidad de software al mismo.

#### **1.10.2.4. Propuesta de integración de software al sistema**

Una vez el sistema se implemente, el desarrollo del mismo se propone realizar, junto con la propuesta de integración con herramientas de software que brindan el soporte digital requerido para ejecutar los procesos de calidad. Dentro de esta propuesta se incluye la proposición de herramientas con las cuales la misma Unidad cuenta en su plataforma de aprendizaje electrónico y una adicional, para documentar y ejecutar planes de prueba de los proyectos de software del TD, la cual es software libre y con código abierto.

#### **1.10.2.5. Propuesta de modelo de calidad de software**

Con el objetivo de realizar la futura elaboración de los artefactos que se proponen en el sistema, se confecciona la propuesta de introducción de un modelo de calidad de software, basado en la ISO/IEC 9126 para verificar, validar y calificar la calidad de software de los proyectos de software del TD.

#### **1.10.2.6. Plan de implementación del sistema**

Además, se desarrolla una propuesta de plan de implementación para el sistema de aseguramiento y control de calidad en la Unidad, así como para todos los elementos requeridos del mismo. Este plan se elabora mediante etapas que describen lo siguiente: actividades por realizar, roles que se encargan de estas y un esfuerzo estimado en términos de tiempo.

### **1.11. Supuestos**

Para el trabajo final de graduación se plantea un supuesto para medir y evaluar la calidad de software:

- Se debió fundamentar el utilizar un modelo de calidad para la propuesta de introducción de atributos y características de calidad del software en los artefactos planteados para el TD, este modelo de calidad se encuentra localizado en la norma ISO/IEC 9126 propuesto por la ISO (2001). Actualmente, esta norma se halla en proceso de sustitución por parte de las normas ISO 25000, las cuales contarán con un modelo que integra más características y atributos de calidad de software; sin embargo, tres partes de esta nueva norma no se encuentran publicadas aún por su organización autora, por este motivo, se decide utilizar la ISO/IEC 9126. Entonces, surge la suposición de que luego, cuando este compendio de normas sea publicado en su totalidad y una vez el modelo de calidad de software de la ISO/IEC 9126 se haya implantado, se realizará la migración al mostrado por las ISO 25000.

### **1.12. Restricciones**

Como parte del desarrollo del presente trabajo final, se estableció una condición para el resultado del producto, seguidamente, se detalla:

- Se debió proponer la implementación de herramientas de software libre y con código fuente abierto para la propuesta de integración con el diseño del sistema. Esto obedece a que la Unidad, de acuerdo con M. Chacón (comunicación personal, 01 de febrero, 2017), tiene como objetivo específico el implementar herramientas de software libre para incorporarlas a las TIC en el desarrollo de sus actividades. Además, según Chacón (2015), el TD aboga por implementar y adaptar proyectos y herramientas de software de otras organizaciones con esa característica, motivo por la cual se presentó el requisito de proponer un software con código abierto.

## CAPÍTULO II: MARCO TEÓRICO

En este capítulo se fundamentan los conceptos necesarios para el desarrollo del presente trabajo final. Se aborda de forma inicial, los conceptos más generales, como lo son: proceso, proyecto y software; los cuales dan el sustento para la especificación del ciclo de vida de desarrollo de software, así como el producto de software.

Luego, se define el concepto de calidad y se enfatiza en la calidad de software para, seguidamente, establecer el campo del aseguramiento de la calidad de software y, dentro de este, el modelo de calidad de software. Al finalizar el capítulo, se indica también el concepto de sistema. A continuación, se profundiza cada uno de los conceptos descritos.

### 2.1. Proceso de negocio

Freund, Rücker y Hitpass (2014) definen un proceso como una *“concatenación lógica de actividades, a través del tiempo y lugar, impulsadas por eventos y que a través de su proceso de transformación, cumplen un determinado fin”* (p. 2), dicha definición se enfoca en la ejecución de un proceso dentro de una entidad y los autores lo llaman: un proceso de negocio.

Dentro de esa definición, los autores detallan una serie de elementos, los cuales describen cuatro características con las que cuenta un proceso de negocio:

- Evento: lo tratan como un tipo de *“ocurrencia externa”* que actúa como detonadora de un proceso de negocio.
- Objetivo: explican que todo proceso de negocio debe tener un objetivo o finalidad específico.
- Actividades: definen una actividad como una *“acción sobre un objeto”*; es decir, son las actividades las que realizan la transformación buscada al ejecutar el proceso de negocio.

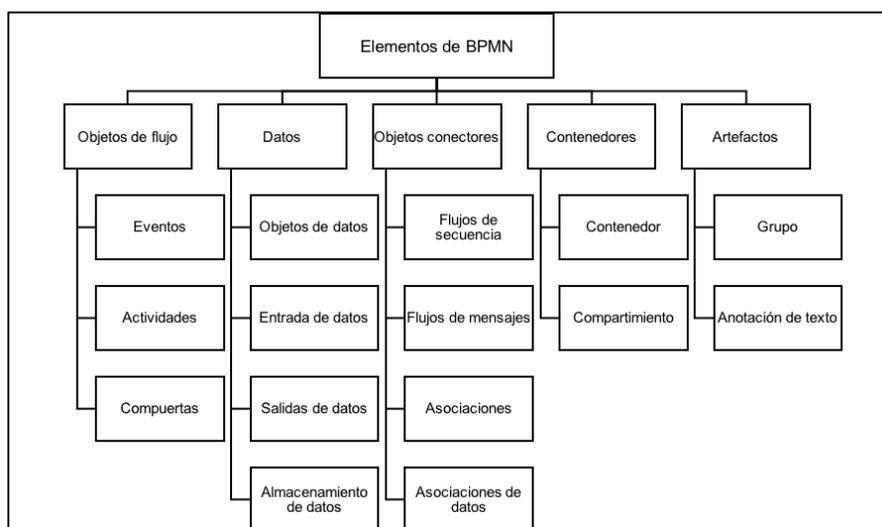
- Secuencia: las actividades de cualquier proceso de negocio deberán seguir siempre, según los autores, una secuencia lógica que se determina de acuerdo con las “*condiciones del negocio*”.

## 2.2. Modelado de procesos de negocio

Moreno, Rodríguez, Snoeck, Moreno, Casas y González (2014) detallan el modelado de los procesos de negocio como “*la descripción y visualización de los procesos por medio de un modelo, que los presente, de manera formal, informal o en la forma de un gráfico o diagrama*” (p. 2). Además, los autores detallan que esta modelación de procesos “*es uno de los primeros pasos hacia el logro de las metas organizacionales*” (p.1).

Para dicha modelación, el *Object Management Group* (en adelante OMG), consorcio internacional sin fines de lucro que se dedica a definir estándares de modelado para elementos de tecnología (OMG, 2017), crea el estándar *Business Process Model and Notation v2.0* (en adelante BPMN 2.0), como un set de herramientas gráficas para esta modelación; este estándar actualiza la notación desarrollada por la *Business Process Management Initiative*. En la Figura 3 se visualiza la estructura para los componentes gráficos de la notación del BPMN 2.0.

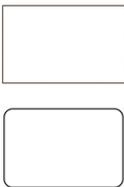
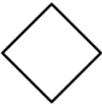
Figura 3. Organización de los elementos de la notación BPMN desarrollada por la OMG

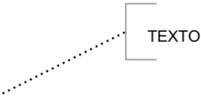


Fuente: Elaborada según el Object Management Group (2011), pp. 27-28.

Seguidamente, en la Tabla 1 se describen nueve de los elementos principales que se detallan en la BPMN 2.0; esta descripción incluye una muestra gráfica de su representación y el detalle de la función tarea para cada uno de estos:

Tabla 1. Notación de BPMN 2.0 desarrollada por el *Object Management Group*

Elemento	Representación gráfica	Descripción
Evento		<p>Un evento es algo que “ocurre” antes o durante un proceso, tienen un disparador y un impacto o resultado, por esta razón afecta el flujo del modelo del proceso. Se representa mediante un círculo y el contenido interior varía de acuerdo con el tipo de evento.</p>
Actividad		<p>Una actividad lleva el nombre de un término genérico de algún trabajo o tarea que la organización realiza en algún proceso, puede ser atómica o no atómica (tiene uno o más subprocesos interiores), esto genera los tipos de actividades como tarea o subproceso, respectivamente. Se representan mediante un rectángulo con o sin las esquinas redondeadas.</p>
Compuertas		<p>Una compuerta se utiliza para cambiar el flujo y la secuencia del modelo del proceso y respectivas actividades o elementos, puede divergir o convergir estos flujos o secuencias. Se representa mediante un rombo y su contenido interior depende del tipo de compuerta.</p>
Objeto de datos		<p>Un objeto de datos brinda información sobre requerimientos de las actividades del flujo o la secuencia del proceso en general.</p>
Flujo de secuencia		<p>El flujo de secuencia muestra los posibles flujos del modelo del proceso que se pueden realizar. Se presenta mediante líneas con terminales de flechas, ambas de fondo sólido.</p>

Elemento	Representación gráfica	Descripción
Flujo de mensajes		<p>El flujo de mensajes se usa para detallar el intercambio de comunicación existente entre dos o más participantes (personas, sistemas, software, entre otras opciones) de un proceso. Se representa mediante una línea segmentada con terminales de flecha sin fondo sólido.</p>
Contenedor		<p>Es la representación gráfica de un proceso, sus actividades y elementos asociados al mismo. Se representa mediante un rectángulo con una separación para el nombre de su título, la orientación puede ser tanto horizontal como vertical.</p>
Compartimiento		<p>Un compartimiento es una partición dentro de un contenedor que muestra el conjunto de elementos ejecutados y asociados por o hacia un participante en particular, se proporciona un título al compartimiento haciendo alusión al nombre, rol o identificación del participante.</p>
Anotación de texto		<p>Una anotación de texto es un tipo de componente que se utiliza para proveer información adicional sobre el proceso o alguno de sus elementos.</p>

*Object Management Group (2011), pp. 29-40.*

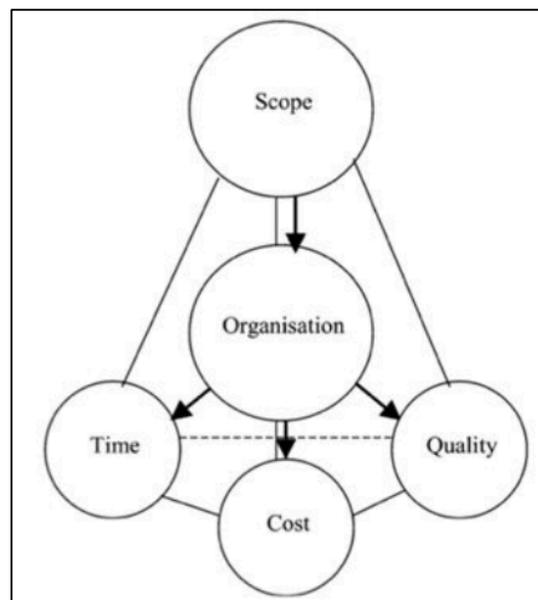
### 2.3. Proyecto

Según el *Project Management Institute (PMI, 2013)*, se define un proyecto como un “esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único” (p. 3). De acuerdo con la línea de concepto definido por el PMI, el proyecto, por lo tanto, tendrá los siguientes parámetros o características (esto de forma independiente del área de desarrollo o ejecución del mismo):

- Alcance: al buscarse “*crear un producto, servicio o resultado único*”, el esfuerzo que conlleva dicha creación deberá ser medido y controlado desde el momento de su planificación hasta su ejecución.
- Costo: el esfuerzo y los recursos necesarios para la ejecución del proyecto tendrán un costo que generalmente, se traduce en términos económicos.
- Tiempo: es un esfuerzo “temporal”; es decir, tiene un inicio y un final; este periodo debe ser planificado y controlado.
- Calidad: el PMI define el éxito del proyecto con base en la calidad del producto o servicio final que se obtiene como resultado del proyecto.

De acuerdo con Stoshikj, Kryvinska y Strauss (2013), quienes citan a Turner (2008), además de las características anteriores, de igual forma la organización (ver Figura 4) donde se desarrolla el proyecto juega un papel determinante en cada uno de estos.

Figura 4. Los cinco componentes de la administración de proyectos según Turner



Fuente: Tomada de Stoshikj, M., Kryvinska, N. y Strauss, C. (2013).

## 2.4. Sistema

De acuerdo con Chiavenato 2007, un sistema se define como un:

[...] conjunto de elementos dinámicamente relacionados en interacción (formando una red de comunicación cuyos elementos son interdependientes) que realizan una actividad, operación o proceso para alcanzar un objetivo o propósito que operan sobre datos, energía o materia tomados del ambiente que circunda al sistema (y con el cual el sistema interactúa dinámicamente) para producir información, energía o materia (que constituyen las salidas o resultados de la actividad del sistema) (p. 12).

Según lo anterior, se puede establecer que, dos principales características de los sistemas son que, se constituyen como dinámicos e interdependientes. Lo anterior se refuerza con lo establecido por Gustafson y Branch (1997), quienes mencionan que además de las descritas, otras características fundamentales son la sinergia y la cibernética. Para estos autores, cada una de estas características se refieren a lo siguiente:

1. Interdependiente: *“ningún elemento se puede separar del sistema, ya que mediante cada uno de estos se logran los objetivos del sistema”*.
2. Sinérgico: *“juntos todos los elementos, tienen más valor que cada uno por aparte, por lo que se puede deducir que un conjunto es mayor a la suma de sus partes”*.
3. Dinámico: *“los sistemas tienen que tener la capacidad de adaptarse a cambios, según las modificaciones del entorno en donde se encuentran o interactúan”*.
4. Cibernético: *“cada elemento que constituye el sistema se puede comunicar entre sí”* (p. 17).

## 2.5. Software

De acuerdo con Pressman (2010), el concepto de software se explica, desde tres perspectivas, como:

uno) instrucciones (programas de cómputo) que cuando se ejecutan proporcionan las características, la función y el desempeño; dos) estructuras de

datos, las cuales permiten que los programas manipulen en forma adecuada la información; tres) información descriptiva tanto en papel como en formas virtuales que describen la operación y uso de los programas (pp. 3-4).

La segunda perspectiva, presentada por Pressman (2010), tiene similitud con lo que propone el ISO y el IEEE (2011) en la norma ISO/IEC/IEEE 24765, en la cual se explica lo que es software mediante dos acercamientos:

1. todo o parte de programas, procedimientos, reglas, y documentación asociada de un sistema de procesamiento de información (p. 329).
2. programas de computadoras, procedimientos, posible documentación asociada y datos pertinentes a la operación de un sistema de computadora (p. 329).

De forma resumida, se detalla software como el conjunto de programas de cómputo, procedimientos, reglas y documentación que buscan, mediante su funcionamiento, ejecutar una acción o proceso a través de un sistema de computadora.

## **2.6. Ciclo de vida de desarrollo de software**

Según la ISO/IEC/IEEE 24765 propuesta por la ISO y el IEEE (2011), el ciclo de vida de desarrollo de software se define como *“el período de tiempo que inicia cuando un producto de software es concebido y termina cuando el software ya no se encuentra disponible para uso”* (p. 32).

De acuerdo con Pressman (2010), el ciclo de vida de desarrollo de software se detalla como un proceso acotado en cinco etapas o como él las llama *“actividades estructurales genéricas”*, en lo que describe como el *“flujo del proceso del software”*; según lo descrito por el autor, ese flujo se compone de:

1. Comunicación: es donde se inicia la concepción del software, donde se transmite la necesidad por parte de un usuario o grupo de usuarios; así también, empieza la participación de los involucrados y se conocen los requerimientos específicos del software.

2. Planeación: es la actividad donde se fija el plan a seguir para la construcción del software. Se define en función del producto o servicio que se construirá y se especifican las actividades generales de gestión y ejecución.
3. Modelado: en esta actividad se realiza un bosquejo<sup>6</sup> de la creación final, con el fin de tener un panorama general del software y su construcción.
4. Construcción: es la aplicación de todo lo planeado y modelado anteriormente, se incluye también las actividades de pruebas y validación del software.
5. Despliegue: se realiza la liberación y entrega del software al usuario final o usuarios finales.

Por otro lado, el PMI y la IEEE Computer Society (2013) realizan un resumen de lo que se menciona en el ISO/IEC 12207 IEEE Std 12207-2008 (elaborada por la ISO y el IEEE [2011]) y establecen el ciclo de vida de desarrollo de software como el conjunto de procesos y subprocesos necesario para la creación de software, el cual se divide en las siguientes seis categorías:

1. Análisis.
2. Arquitectura.
3. Diseño.
4. Construcción.
5. Integración.
6. Prueba.

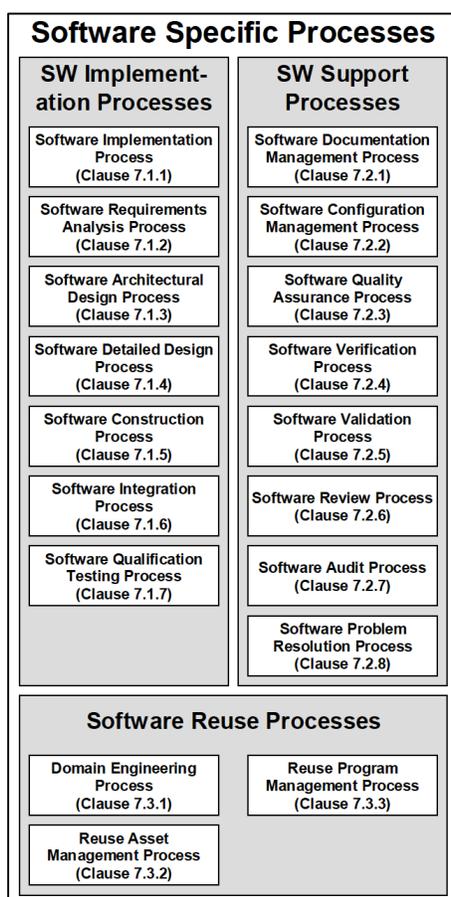
Además de estas categorías, el PMI menciona algunas adicionales enfocadas eso sí, en el ciclo de vida de desarrollo de software en general, las cuales son: liberación, soporte, mantenimiento, evolución, reemplazo y retiro del software; sin embargo, dicha organización no aborda de forma profunda las mismos.

---

<sup>6</sup> Bosquejo: se refiere a construcciones abstractas que fungen de modelos o diseños para la siguiente etapa del flujo del proceso del software.

Abordaje que se presenta efectivamente, en el estándar ISO/IEC 12207 IEEE Std 12207-2008 desarrollado por el ISO y el IEEE (2011); pues se describe de forma detallada todos las categorías y los procesos anteriores (ver Figura 5) y aunque llevan otros nombres, se dividen en las siguientes categorías principales: procesos de implementación de software, de soporte y de reutilización de software. Estas categorías se componen de los siguientes procesos:

Figura 5. Clasificaciones de los procesos específicos de software según la ISO/IEC 12207 IEEE Std 12207-2008



Fuente: Tomada de ISO y el IEEE (2011), p. 14.

## 2.7. Ingeniería de requerimientos de software

Sommerville (2011, p. 83) define la ingeniería de requerimientos de software como la etapa del ciclo de vida de desarrollo de software donde se descubre, se analiza, se documenta y se verifica los requerimientos de un software. Además, explica

que los requerimientos de software son las *“descripciones de lo que el sistema<sup>7</sup> debe hacer: el servicio que ofrece y las restricciones en su operación”*. Arias (2007) agrega que para realizar esta etapa se pueden seguir los siguientes pasos o actividades:

1. Extracción.
2. Análisis.
3. Especificación.
4. Validación.

Arias (2007) además, expone que existen dos categorías para los requerimientos de software: funcionales y no funcionales. Por otro lado, Sommerville (2011, pp. 84-85) describe las siguientes definiciones para estas categorías de requerimientos: los funcionales los *“definen las funciones que el sistema será de capaz de realizar”*; es decir, las *“transformaciones que el sistema debe realizar sobre las entradas para producir salidas”*. Mientras, los requerimientos no funcionales tienen relación con atributos o características que *“de una u otra forma pueden limitar el sistema”* o su uso.

Adicionalmente, Arias (2007) expone que un requerimiento debe poseer ciertas características para aplicar la ingeniería de requerimientos de software de forma correcta, dichas características se describen a continuación:

- Especificado: el requerimiento debe ser documentado donde se detalle por escrito la elaboración del mismo.
- Verificable: el requerimiento debe brindar la posibilidad de ser validado y probado a lo largo del ciclo de vida de desarrollo del software.

---

<sup>7</sup> Sistema: en este contexto de ingeniería de requerimientos de software, el término software también será referido como “sistema”.

- Conciso: el requerimiento debe ser fácil de comprender por las partes involucradas en el desarrollo del software, esto significa que debe ser redactado de forma simple y clara.
- Completo: el requerimiento no debe necesitar de alguna explicación o redacción de texto adicional de la que se detalla; pues se debe brindar la información necesaria para la comprensión del mismo.
- Consistente: un requerimiento no debe contradecir a otro o ser contradicho por otro.
- No ambiguo: el requerimiento debe tener una sola interpretación para todas las partes involucradas.

## 2.8. Producto de software

Si se parte de lo que presenta el ISO y el IEEE (2011) en la ISO/IEC/IEEE 24765 donde se define un producto de software como un *“conjunto de programas de cómputo, procedimientos, y todos los datos y documentación posiblemente asociada”*, se establece entonces un producto de software como un resultado posible del ciclo de vida de desarrollo de software definido de manera previa; es decir, tanto el programa de cómputo como tal así como todo lo resultante de este proceso: la documentación, diseños, los manuales de usuario, datos, políticas de uso, entre otros.

## 2.9. Calidad

Crosby (1987) definió calidad como la *“conformidad con los requerimientos”* que se tienen para un producto o servicio, los cuales se formulan a raíz de una necesidad inicial; el autor menciona como un error usual asociar el término calidad con significados de carácter positivo, por ejemplo, con algo “bueno” o “lujoso”; pues la calidad puede satisfacer o no, los requerimientos solicitados para un producto o servicio. Por esto, el autor explica que se tienen distintos niveles de calidad: mala, regular, alta, entre otros niveles adicionales.

Deming (2000) por otra parte, explica que el concepto de la calidad de un producto o servicio se podrá definir sólo *“en términos del agente”*, el agente es el

usuario, grupo de usuarios o destino final por los cuales, en primera instancia, se crea la necesidad de un nuevo producto o servicio. El autor describe de forma implícita que la calidad se debe definir solamente en términos de satisfacción de la necesidad inicial por parte del “agente”.

Por otro lado, Shewhart (1931) atribuyó dos peculiaridades principales al término calidad, que reúne las dos perspectivas anteriormente descritas, las cuales son la característica subjetiva, asociada a la satisfacción directa que tiene el cliente o grupo de clientes como resultado del uso del producto o el servicio; y, la objetiva, relacionada a las propiedades y las características propias del producto o servicio, pero que son independientes de la satisfacción del cliente; es decir, las orientadas en el proceso de creación de ese producto o servicio y reguladas por algunos lineamientos o estándares de calidad definidos *a priori*.

Finalmente, en la norma ISO 9000, creada por la ISO (2015), se menciona que la calidad se determina mediante el nivel de cumplimiento de los requerimientos de un cliente para un producto o servicio. De forma adicional, en dicha norma se desarrollan siete principios, los cuales se deben cumplir para conseguir la calidad en una organización, para sus procesos y sus resultados en general. Estos principios son independientes del tipo de entidad, área o proceso y son descritos de seguido:

1. Enfoque en el cliente: el objetivo primario para alcanzar la calidad siempre deberá ser el cumplir los requerimientos de los clientes y aportarles valor a dichos requisitos. Cuatro de los beneficios que se exponen en la norma como resultado de este principio son:
  - a. Se incrementa el valor que percibe el cliente.
  - b. Se aumenta la satisfacción del cliente.
  - c. Se mejora la reputación de la entidad.
  - d. Se logra la fidelidad del cliente.

2. Liderazgo: se debe crear una cultura para que los líderes de todos los niveles impulsen condiciones en donde se busque la calidad de forma transversal. En la norma se mencionan los siguientes beneficios a raíz de la implementación de este principio:
  - a. Se incrementa la eficiencia y la eficacia de la organización para cumplir sus objetivos.
  - b. Se mejora la coordinación de los procesos de la entidad.
  - c. Se desarrolla y se mejora la capacidad de la organización para entregar los resultados deseados.
3. Compromiso de los colaboradores: se debe adquirir compromiso por parte de todos los niveles de la organización para crear un enfoque de calidad. De acuerdo con este principio, mediante la aplicación de la norma se obtienen los siguientes resultados positivos:
  - a. Se difunde el entendimiento de la organización sobre los objetivos de calidad.
  - b. Se involucra a los colaboradores en el mejoramiento de las actividades.
  - c. Se mejora la confianza y la ayuda de los colaboradores que constituyen la entidad.
4. Acercamiento al proceso: se debe generar un entendimiento a los colaboradores del funcionamiento de la organización mediante un enfoque de procesos y sus actividades. De seguido, se describen los posibles beneficios que se brindan al cumplir este principio:
  - a. Se mejora la habilidad para enfocar esfuerzos sobre los procesos claves de la organización y las oportunidades de mejora.
  - b. Resultados consistentes y predecibles mediante el sistema de alineación de los procesos.

5. **Mejoramiento:** se debe poseer un enfoque continuo para el mejoramiento en general de la organización. Las consecuencias positivas al cumplir este principio pueden ser:
  - a. Se mejora la ejecución de los procesos, las capacidades de la entidad, sus resultados y, por ende, la satisfacción de los clientes.
  - b. Mejoramiento de la dirección hacia la innovación.
6. **Toma de decisiones basada en evidencias:** se debe tomar decisiones sobre el análisis de datos e información de la entidad, sus procesos y sus resultados. Se detallan tres de los posibles beneficios, según la norma, de la implementación de este principio:
  - a. Se mejora la toma de decisiones y sus resultados.
  - b. Se mejora la eficiencia y eficacia operacional de la organización.
  - c. Se mejora la habilidad para revisar, retar y cambiar opiniones y decisiones de la organización.
7. **Gestión de las relaciones:** se debe gestionar las relaciones con los involucrados de la organización para mantener la calidad sobre sus procesos y resultados a través del tiempo. Dos de los beneficios de cumplir este principio, según menciona la norma, se detallan a continuación:
  - a. Constante retroalimentación entre las partes involucradas, lo cual genera un mejoramiento continuo de la entrega de resultados.
  - b. Se mejora la capacidad para entregar resultados de valor a los involucrados.

## **2.10. Calidad de software**

En la ISO/IEC/IEEE 24765 que elaboraron la ISO y el IEEE (2011), se define la calidad de software como la *“capacidad de un producto de software de satisfacer las necesidades implicadas y establecidas cuando es usado bajo condiciones específicas”*;

cabe destacar el énfasis que se realiza sobre el producto del software, esto significa que el término calidad lo especifican de acuerdo con el cumplimiento del producto de software, no según el proceso de elaboración o el ciclo de vida del software.

La definición anterior se asemeja con la presente en la UNE-ISO/IEC 90003 que se desarrolló por la Asociación Española de Normalización y Certificación, ISO y el IEEE (2005), donde se especifica un concepto de calidad de software relacionado en forma directa, a la satisfacción de los requerimientos del software; esto significa que no se toma en cuenta para dicho término todo el proceso constructivo requerido para llegar a ese cumplimiento de requisitos; pues se define se calidad de software en virtud de únicamente, los requerimientos, no según las condiciones de uso en las cuales se prueba o valida la calidad.

Por otro lado, de acuerdo con Nazir (2005), el concepto de calidad de software está ganando cada vez más importancia en el ambiente del software; el autor utiliza como base la misma definición de la ISO/IEC/IEEE 24765 y menciona que la calidad del software es una característica crítica para el éxito de un proyecto de este tipo.

Wagner, Chulani y Wong (2011) adicionan que la calidad del software no es una propiedad establecida o universal del software, más bien se depende de cada proyecto o construcción en específico, para efectos de definición de la misma.

Wagner et al. (2011) mencionan que la calidad del software depende completamente, del contexto y las metas que tienen los involucrados con el software elaborado. Además, según los autores, la única forma de brindar calidad al producto de software es controlando la presencia de ésta a través de todo el ciclo de vida de desarrollo del software.

### **2.11. Aseguramiento de la calidad de software**

La ISO/IEC/IEEE 24765 elaborada por la ISO y el IEEE (2011) proporciona varias definiciones para el aseguramiento de la calidad como concepto general (no asociado a software), dentro de estas se encuentran:

- *“Un patrón planeado y sistemático de todas las acciones necesarias para proveer la confianza adecuada que un ítem o producto satisface los requerimientos técnicos establecidos”* (p. 287).
- *“Un conjunto de actividades diseñadas para evaluar el proceso mediante el cual son desarrollados y manufacturados productos”* (p. 287).

Estas dos definiciones no se orientan de forma específica en software; sin embargo, si se toma como referencia la definición de producto de software generada anteriormente, entonces se puede definir el aseguramiento de la calidad de software como el conjunto de procesos y actividades planeadas y ejecutadas, para proveer la certeza que se está cumpliendo con los requerimientos establecidos para un producto de software, tanto en sus etapas de creación así también como en su resultado final.

Por otra parte, Pressman (2010) describe el aseguramiento de la calidad de software como un tipo de *“actividad sombrilla”* dentro del ciclo de vida de desarrollo de software, esta característica de *“sombriilla”* se refiere a que se realiza durante todo este ciclo, con el fin de asegurar la calidad y el cumplimiento de requerimientos del software; además la define como la actividad que establece y ejecuta las sub actividades requeridas, para garantizar la calidad de software; de igual forma, dicha actividad se debe adaptar de acuerdo con el tipo de proyecto que se desarrolla.

Además, Pressman (2010) detalla que la meta del aseguramiento de la calidad es, dentro del campo de desarrollo de software, *“proveer al equipo administrativo y técnico los datos necesarios para mantenerlo informado sobre la calidad del producto, con lo que se obtiene perspectiva y confianza en que las acciones necesarias para lograr la calidad del producto funcionan”*. (p. 351).

Para cumplir esa meta, Pressman (2010) establece una serie de requisitos o pasos a seguir, los cuales se deben llevar a cabo para lograr una correcta ejecución del aseguramiento de la calidad de software:

1. Un proceso establecido de aseguramiento de la calidad de software.

2. Tareas específicas de aseguramiento y control de calidad que pueden incluir las pruebas o revisiones de calidad y las estrategias definidas para la ejecución de las mismas.
3. Métodos y herramientas que permitan la ejecución de prácticas eficaces de aseguramiento de la calidad de software.
4. Control de todos los productos de software y los cambios que reciben a través del ciclo de desarrollo de software.
5. Un procedimiento para asegurar el cumplimiento de los estándares de desarrollo de software (en caso que aplique).
6. Mecanismos para la medición y el reporte sobre aseguramiento de la calidad de software.

Por otro lado, Avella y Gómez (2011) formulan una *“metodología integrada al proceso de construcción de software para aplicar inspecciones y pruebas”*. Los mismos autores señalan que en el ambiente específico de aseguramiento de la calidad del software, se encontraron con un panorama muy amplio sobre actividades, procesos, técnicas y herramientas, los cuales se pueden utilizar para propiciar la calidad del software; sin embargo, no se encontró algún tipo de metodología o marco de referencia específico que integrara todo eso en una propuesta sobre la cual sustentarse, con el fin de proveer el aseguramiento de la calidad a cualquier proyecto de software o producto de este tipo.

Para esta metodología integrada, los autores utilizan como base de evaluación los estándares IEEE Std 1028-2008, IEEE Std 829-2008 y también el modelo de calidad propuesto en la ISO/IEC 9126. Esta metodología se encuentra enfocada hacia el aseguramiento y el control de la calidad de software, al usar como elementos principales los procesos de calidad de inspección y pruebas de la calidad del software.

De acuerdo con Avella y Gómez (2011), surge una diferencia entre lo que es una inspección de software y una prueba de software, ambos conceptos contenidos dentro del aseguramiento de la calidad del software. La primera se refiere a lo que

usualmente, se conoce como revisiones técnicas, las cuales son planeadas y ejecutadas desde el inicio del ciclo de vida del software y se enfocan más hacia el proceso en sí y ciertos requerimientos en el ciclo de vida de construcción del software. Por otra parte, las pruebas de software las caracterizan como procedimientos manuales o automáticos realizados sobre dicho ciclo.

Según Avella y Gómez (2011), uno de los fines principales de las inspecciones de software es la detección temprana de errores dentro del proceso de ciclo de vida de software y, de esta manera, evitar la transformación de estos errores en defectos propiamente, sobre los productos finales, defectos que implicarán esfuerzos y recursos extras para la corrección de los mismos. Mientras los objetivos fundamentales de las pruebas de la calidad del software son demostrar el cumplimiento satisfactorio de los requisitos y descubrir los defectos propios sobre el producto final del software, los cuales evitan el correcto funcionamiento del mismo.

Pressman (2010) por otra parte, expone el beneficio de reducción de costos obtenidos a partir del aseguramiento y control de la calidad, desde las etapas iniciales de los proyectos de software por medio de las inspecciones propuestas por Avella y Gómez (2011); pues establece que el *“costo de la calidad”* está asociado con la prevención, evaluación y fallas de los productos de software. Menciona que este costo de la calidad, principalmente, el de evaluación y las fallas ocasionadas por la falta de planificación de calidad se elevan de manera cuantiosa, cuando no se invierte los recursos pertinentes en el aseguramiento de la calidad desde etapas tempranas.

De acuerdo con Avella y Gómez (2011), las inspecciones y las pruebas de software son actividades complementarias mutuamente, estas no deben verse como procesos o acciones aisladas; todo lo contrario, deben utilizarse de forma conjunta en la validación y la verificación de la calidad a través de todo el ciclo de vida del software.

La construcción de esta metodología sobre aseguramiento y control de la calidad del software, guió a Avella y Gómez (2011) a dividir el ciclo de vida de desarrollo del software en etapas específicas sobre las cuales pueden ser planeadas y ejecutadas las

actividades de la metodología. Estas etapas son las que se muestran en la Figura 6 a continuación:

Figura 6. Fases o etapas del ciclo de desarrollo de software según Avella y Gómez



Fuente: Tomada de Avella y Gómez (2011), p. 278.

Esta metodología propuesta por los autores define una serie de actividades, técnicas y herramientas utilizables para aplicarlas a las inspecciones y las pruebas de software que mencionan; de seguido se presenta a modo de síntesis, las actividades requeridas y algunas de las herramientas necesarias para cada etapa, de acuerdo con el modelo de ciclo de vida y construcción de software que Avella y Gómez (2011) utilizan:

- Etapa de análisis:
  - Revisión del documento de ingeniería de requerimiento o especificación de requisitos, los autores definieron una lista de chequeo con los elementos mínimos y características con las cuales debe contar dicho documento.
  - Creación y revisión de un plan de las pruebas enfocadas en el cumplimiento de los requerimientos que se realizarán de forma posterior, estas son descritas de forma general. Para esta actividad recomiendan el uso del IEEE std 829-2008.
- Etapa de diseño arquitectónico:
  - Revisión del documento de arquitectura del software; para esta actividad los autores adaptaron una lista de chequeo.

- Creación y revisión de un plan de pruebas de integración, con el fin de identificar errores entre los componentes del software que se construirá.
- Etapa de diseño:
  - Revisión del documento de diseño software (incluye el modelado de las clases del software), utilizando como herramienta evaluadora una lista de chequeo.
  - Creación y revisión de un plan de pruebas unitarias, orientadas en las estructuras de datos, la lógica y la formulación de los algoritmos. Este plan se realiza por medio de casos de prueba.
- Etapa de codificación:
  - Inspección del código fuente del software construido a partir de alguna base técnica, como manual de programación o buenas prácticas.
  - Ejecución de los casos de prueba formulados anteriormente, los autores recomiendan que el software a probar cuente con interfaz gráfica para facilitar la ejecución de estas pruebas.
  - Reporte de los errores encontrados mediante la ejecución de los casos de prueba.
  - Verificación de las correcciones surgidas a partir del reporte de los errores del software.
  - Ejecución de pruebas de regresión total o parcial, las cuales garanticen que las correcciones realizadas no afectan las funcionalidades revisadas anteriormente.

Estas actividades, herramientas y métodos, de acuerdo con Avella y Gómez (2011), se plantean con el fin de apoyar el aseguramiento buscado de la calidad, para los proyectos y los productos de software; además, se deben ejecutar y utilizar de forma complementaria, junto a la integración de la ISO/IEC 9126.

## 2.12. Modelo de calidad de software

Un modelo de calidad se define de acuerdo con lo descrito por el ISO y el IEEE (2011) en la ISO/IEC/IEEE 24765 como un “*conjunto definido de características, y las relaciones entre estas, las cuales proveen la base para especificar los requerimientos de calidad y evaluar la calidad*” (p. 289); al tomar en cuenta esta definición y al realizar una asociación con un modelo de calidad de software, este término se adapta específicamente, a los requerimientos y la evaluación de los productos de software que se crean a partir de un proyecto o producto de este tipo.

Cuando se utiliza un modelo de calidad para la construcción de software, de acuerdo con Dutil, Rose, Suryn y Thimot (2010), se propicia una correcta definición para las mediciones de la misma, la cual contribuye a que se alcance el nivel esperado de calidad del producto de software.

Además, de acuerdo con Al-Elaimat y Al-Ghuwairi (2015), con la utilización de un modelo de calidad de software, además de brindar la posibilidad de dar uso a las medidas de calidad para validar y verificar la misma de forma posterior a la construcción del software, con el enfoque de cumplimiento de estas medidas se logra propiciar un conjunto de acciones y herramientas que son empleables a través de todo el ciclo de vida de desarrollo del software.

Bianchi, Soares y Romero (2015), por su parte, resaltan la utilización de un modelo de este tipo con el objetivo de describir, estimar, propiciar y predecir la calidad del software por medio de la descomposición de características, atributos y métricas de calidad de software.

Un ejemplo de un modelo de calidad de software (el cual se especificará posteriormente) y la descomposición del mismo lo presentan Lochmann y Goeb (2011), citados por Nandakumar, Lal y Parmar (2014); esta descomposición contiene las siguientes características o atributos de calidad de software: practicidad funcional, eficiencia de ejecución, compatibilidad, usabilidad, confiabilidad, seguridad, mantenibilidad y portabilidad.

Dicha descomposición de características, atributos y métricas de calidad que presentan Nandakumar et al. (2014), se muestra en varios modelos de calidad. De acuerdo con Lochmann, et al. (2013) existe gran cantidad de modelos de calidad de software utilizables en diversos ámbitos, tanto para desarrollo de software con un enfoque formal como para un desarrollo más ágil, ya sea orientados en ciertas necesidades o de carácter más general.

Los siguientes autores, mencionan una serie de modelos de calidad de software, los cuales se resumen a continuación:

- Lochmann y Goeb (2011) describen los siguientes modelos de calidad de software:
  - El propuesto en la norma ISO/IEC 9126 que desarrolla el ISO, donde se especifica un modelo de calidad de software en su parte uno (tiene un total de cuatro partes). Este modelo se desarrolla desde distintas perspectivas de calidad, las otras partes describen tres reportes técnicos.
  - El modelo de calidad presente en la norma 25010, también propuesta por la ISO.
- Wagner et al. (2011) exponen los siguientes modelos:
  - Un modelo de calidad desarrollado por el *Software QUALity Enhancement (SQUALE) project* que utiliza un marco de trabajo de análisis de código de software de forma complementaria.
  - El modelo de calidad de software elaborado por el *Quamoco Consortium* como una síntesis de varios modelos de calidad basados en actividades, para el control de la calidad de un producto de software.
- Nandakumar et al. (2014) por su parte, muestran el siguiente modelo:
  - Modelo de calidad *Cost-Benefit (CoBe)* propuesto para las actividades de aseguramiento de la calidad del software.

- Chawla y Chhabra (2015) por otro lado, detallan el siguiente modelo:
  - El *Software Quality Model for Maintainability Analysis* (SQMMA), el cual desarrollan con el fin de cuantificar características de calidad relacionadas a la mantenibilidad como atributo.

Dentro de estos modelos de calidad de software, cabe resaltar el modelo presente en la norma ISO/IEC 9126, específicamente, en su parte uno; dicho modelo, de acuerdo con St-Louis y Suryan (2012) es uno de los mejores modelos de este tipo existentes en la actualidad (haciendo referencia a su medición y aplicabilidad).

La norma que contiene ese modelo, según Abran, Al-Qutaish, Desharnais y Habra (s.f.), fue publicada inicialmente, por la ISO en 1991 donde se describía el conjunto de características enfocadas en la evaluación de productos de software; ya para el periodo comprendido entre los años de 2001 a 2004, dicha entidad publicó una versión extendida de la misma.

Con esta versión extendida, la ISO (2001) detalla para esta norma en su parte uno (ISO/IEC 9126-1), el modelo de calidad enfocado en las tres perspectivas de calidad, las cuales ISO propone como ópticas para abordar el tema de calidad de software; las perspectivas son: la calidad interna, la calidad externa y la calidad en uso.

En las otras tres partes (ISO/IEC/TR 9126-2 desarrollada por la ISO y el IEEE [2003a], ISO/IEC/TR 9126-3 elaborada por la ISO y el IEEE [2003b] y ISO/IEC/TR 9126-4 construida por la ISO y el IEEE [2004]) se describe, de acuerdo con Al-Qutaish (2009), reportes técnicos que contienen métricas específicas para la medición de la calidad del software, definidas, según las tres perspectivas de calidad mencionadas anteriormente.

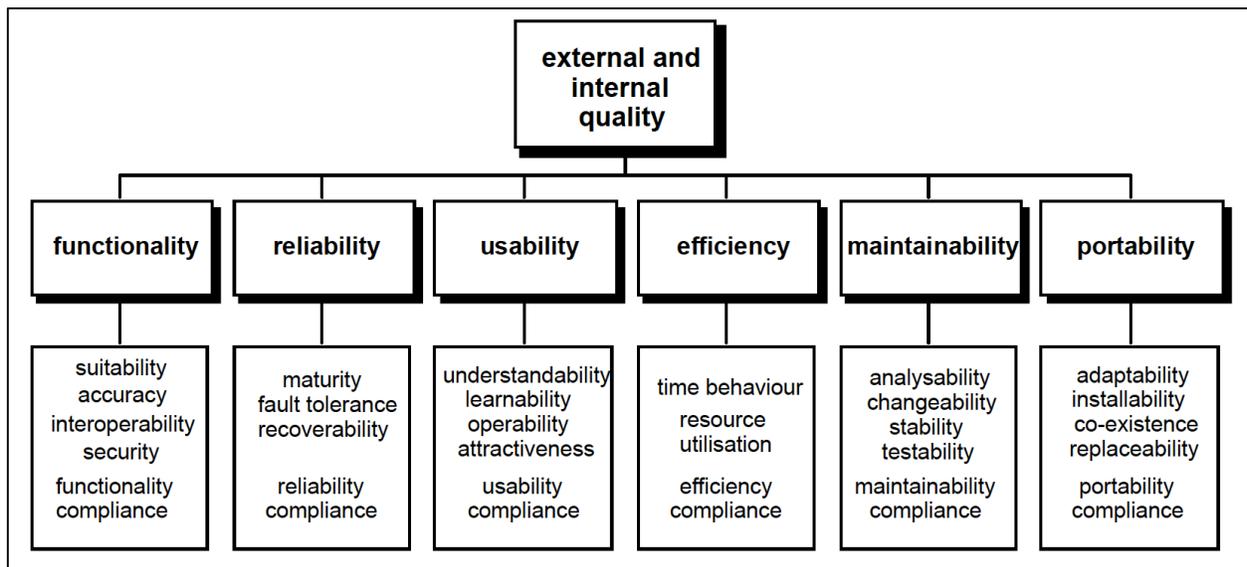
La calidad tanto interna como externa descritas en la ISO/IEC 9126-1 y en sus correspondientes reportes técnicos, forman la primera parte del modelo de calidad de software propuesta por dicha norma y de acuerdo con la ISO (2001), estas calidades se pueden encontrar manifiestas de forma externa en las características del software propiamente, al ser construido y probado y también como resultado de una serie de

atributos internos que se propician durante todo el ciclo de vida de desarrollo del software.

Por otro lado, según la ISO (2001), la calidad en uso, la cual es la segunda parte del modelo de calidad de software definido en la ISO/IEC 9126-1, se puede describir como un efecto combinado de las características de calidad interna, externa y propiamente, del uso por parte de los usuarios finales del software como producto.

Además, Lochmann y Goeb (2011), al tomar como fundamento la primera parte del modelo de calidad de software descrito en la ISO/IEC 9126-1, describen las seis características de calidad presentes en este modelo: eficiencia y las cinco características conocidas como las “ilidades” (funcionalidad, fiabilidad, usabilidad, mantenibilidad y portabilidad). Cada una de estas características se descomponen en una serie de subcaracterísticas, tal y como se muestra en la Figura 7. A su vez, para cada una de estas subcaracterísticas se formulan, en los reportes técnicos, una serie de métricas que hacen posible la medición y la evaluación de la calidad del software.

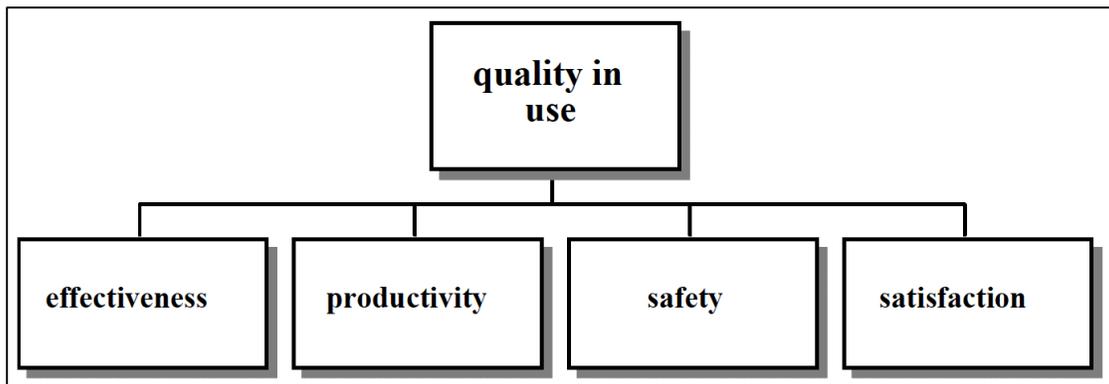
Figura 7. Las seis categorías y subcategorías de calidad de software de acuerdo con la ISO/IEC 9126



Fuente: Tomada de ISO y el IEEE (2001), p. 7.

Además, de acuerdo con la ISO (2001), en la segunda parte de la ISO/IEC 9126-1, se detalla el modelo de calidad de software que se utiliza para la calidad en uso, el cual contrasta en número de características y métricas (detalladas en su correspondiente reporte técnico) de medición y validación de calidad de software; estas características de calidad se muestran en la Figura 8, la cual se presenta a continuación:

Figura 8. Subcaracterísticas de la calidad en uso según la ISO/IEC 9126-1

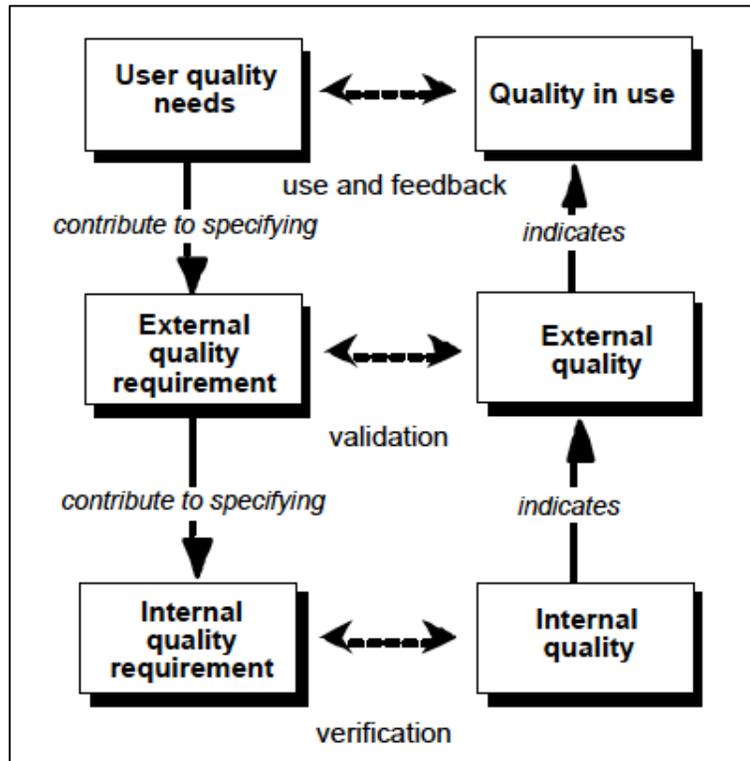


Fuente: Tomada de ISO y el IEEE (2001), p. 5.

Las características que propone la ISO/IEC 9126-1 para la calidad en uso, al compararse con las características pertinentes a la calidad externa e interna, varían tanto en cantidad así como en su estructura y objetivo; pues el fin principal del modelo de la calidad de software, enfocado en el uso del mismo, busca medir y evaluar el nivel de calidad por medio de un enfoque por completo orientado al usuario final del software. Por esta razón, las características de esta parte del modelo son cuatro: eficacia, productividad, seguridad y satisfacción.

Además, las características y las subcaracterísticas de la calidad presentes en la ISO/IEC 9126-1, para sus tres perspectivas descritas anteriormente, están enlazadas de forma directa y subsecuente (ver Figura 9).

Figura 9. Flujos de interacción entre las perspectivas de calidad



Fuente: Tomada de ISO y el IEEE (2001), p. 4.

En la figura anterior, se reconoce que todo atributo o estado de la calidad interna de un producto de software tendrá un efecto directo en la calidad externa del mismo, este, a su vez, se vincula o tiene un impacto en la definición de la calidad en uso por parte de los usuarios finales del software.

Por esta razón, Avella y Gómez (2011) proponen, como buena práctica de desarrollo de software, que los requerimientos o las características de calidad sean propiciados y evaluados desde las etapas iniciales del ciclo de vida de desarrollo del software.

Según lo mencionado por Al-Qutaish (2009), la conformación del modelo de calidad de software propuesto en la ISO/IEC 9126-1, se basa en tres perspectivas de calidad, cada una de estas a su vez tienen ciertas características o atributos y en estos se especifican subcaracterísticas asociadas a métricas, las cuales las definen en un

rango que hace posible la verificación, la validación y la evaluación objetiva del software.

Un ejemplo de estas métricas y sus componentes, se denota en la Figura 10 que se muestra a continuación:

Figura 10. Métricas presentes en la sub característica idoneidad de calidad

Table 8.1.1 Suitability metrics									
External suitability metrics									
Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value	Metric scale type	Measure type	Input to measurement	ISO/IEC 12207 SLCP Reference	Target audience
Functional adequacy	How adequate are the evaluated functions?	Number of functions that are suitable for performing the specified tasks comparing to the number of function evaluated.	$X=1-A/B$ A= Number of functions in which problems are detected in evaluation B= Number of functions evaluated	$0 \leq X \leq 1$ The closer to 1.0, the more adequate.	Absolute	X= Count/ Count A= Count B= Count	Requirement specification (Req. Spec.) Evaluation report	6.5 Validation, 6.3 Quality Assurance, 5.3 Qualification testing	Developer, SQA
Functional implementation completeness	How complete is the implementation according to requirement specifications?	Do functional tests (black box test) of the system according to the requirement specifications. Count the number of missing functions detected in evaluation and compare with the number of function described in the requirement specifications.	$X = 1 - A / B$ A = Number of missing functions detected in evaluation B = Number of functions described in requirement specifications	$0 \leq X \leq 1$ The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/ Count	Req. spec. Evaluation report	6.5 Validation, 6.3 Quality Assurance, 5.3 Qualification testing	Developer, SQA
<b>FOOTNOTES</b>									
1 Input to the measurement process is the updated requirement specification. Any changes identified during life cycle must be applied to the requirement specifications before using in measurement process.									
2 This metric is suggested as experimental use.									

Fuente: Tomada de ISO y el IEEE (2003a), p. 5.

En Figura 10 se muestran dos ejemplos de métricas pertenecientes a la perspectiva de calidad externa, presentes en el reporte técnico ISO/IEC/TR 9126-2 elaborado por la ISO (2003a); dichas métricas son generadas a partir de la característica de funcionalidad, la cual a su vez presenta la sub característica de practicidad. En este ejemplo estas dos métricas llevan el nombre de suficiencia funcional y completitud de implementación funcional.

La estructura que presentan las métricas en los reportes técnicos de la ISO/IEC 9126 contiene los siguientes componentes:

- Nombre de la métrica: este atributo permite identificar, de forma única, cada métrica; en el caso de métricas de calidad externa e interna, se tienen métricas con nombres similares pero con medidas y rangos distintos.

- Propósito de la métrica: mediante una pregunta, describe el fin por el cual se debe aplicar la métrica.
- Método de aplicación de la métrica: provee una guía de cómo aplicar la métrica en casos específicos.
- Medida, fórmula y cálculos necesarios: brinda cálculos y fórmulas para aplicar el método de aplicación.
- Interpretación del valor medido: se muestra cómo interpretar el valor calculado para propiciar mejores índices de calidad.
- Tipo de escala de métrica: se describe el tipo de escala para la métrica.
- Tipo de medida: describe los factores de las medidas y fórmulas utilizadas para el cálculo.
- Medida de la entrada: describe posibles procesos, etapas o artefactos del ciclo de vida de desarrollo del software donde es posible utilizar la métrica.
- Referencia al proceso del ciclo de vida de desarrollo del software en el cual puede utilizarse: hace una asociación directa con el proceso del ciclo de vida de desarrollo del software que se encuentra en el estándar ISO/IEC 12207 IEEE Std 12207-2008.
- Audiencia objetivo: describe los posibles roles o usuarios dentro del ciclo de vida de desarrollo del software, a quienes les podría ser útil el resultado obtenido de la métrica.

Todos los conceptos que se desarrollaron en el marco teórico anterior, fueron de vital relevancia para ejecutar la metodología planeada, así también para elaborar la propuesta de solución, conclusiones y recomendaciones del presente trabajo final de graduación. Estos conceptos se desarrollaron y se ordenaron de forma tal, que brindaran la base teórica precisa, la cual permitió un estudio sobre las áreas y los

conocimientos específicos y necesarios para el abordaje de la problemática presente en el TD sobre la calidad en sus proyectos de software.

## **CAPÍTULO III: DESARROLLO METODOLÓGICO**

En este capítulo se especifica la metodología utilizada para el desarrollo del presente trabajo final. La misma contiene los siguientes elementos: generalidades, diseño de investigación, métodos utilizados, sus respectivas técnicas e instrumentos de apoyo para ejecutar las mismas y finalmente, una descripción de las etapas aplicadas de la metodología.

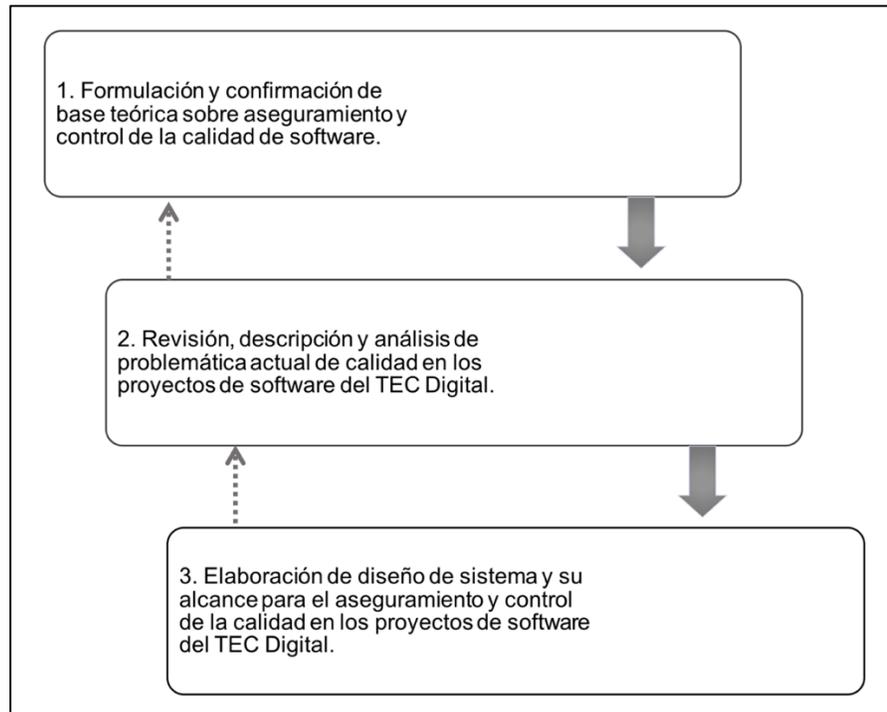
### **3.1. Generalidades de la metodología**

Para Rodríguez y Valldeoriola (2009) la metodología resulta esencial en cualquier proceso de investigación; dado que esta determina la forma de ejecución de este desarrollo. En primera instancia, se describe el enfoque de investigación utilizado, el desarrollo del diseño general y específico de investigación que se emplearon, así también el desglose de los métodos de investigación, técnicas y herramientas de apoyo necesarios para la ejecución de los mismos.

Todos los componentes mencionados anteriormente, se ordenan de forma mediante la definición de tres etapas metodológicas, las cuales se explican al final de este capítulo. En la Figura 11 se muestra, de forma gráfica y resumida, dichas etapas.

Una característica importante de destacar de estas etapas, sus actividades y demás elementos es que, entre los mismos existió procesos iterativos. Esta característica es propia del enfoque y diseño de investigación seleccionados, los cuales se detallan de manera posterior, en los apartados 3.3. y 3.4., respectivamente.

Figura 11. Fases metodológicas del trabajo final de graduación



Fuente: Elaboración propia.

### 3.2. Alcance de investigación

La investigación que se desarrolló incluye, de acuerdo con lo descrito por Hernández, Fernández y Baptista (2014), tres tipos de alcances: exploratorio, descriptivo y explicativo.

El primero se empleó debido a que, fue necesario examinar sobre el tema de aseguramiento y control de la calidad, para el cual se presentaban dudas y no se había abordado de una manera formal en el TD anteriormente; por lo tanto, resultó necesaria dicha exploración para sentar el estudio teórico que brindara el cimiento de los componentes y los resultados del trabajo.

El segundo tipo de alcance desarrollado, el descriptivo, surge a raíz de la necesidad explícita que existió en la estructura del trabajo final de graduación, de realizar una descripción del problema y el estado actual en la gestión de la calidad del

software. Además de la búsqueda de describir los resultados obtenidos mediante la ejecución de las técnicas e instrumentos investigativos.

Finalmente, se utiliza además un alcance explicativo, para elaborar una profundización sobre las causas, los efectos y demás factores influyentes que se encontraron determinantes sobre la problemática; de esta forma se apunta hacia la propuesta de solución basada en el sistema de aseguramiento y control de la calidad en los proyectos de software del TD.

### **3.3. Enfoque de investigación**

Para el trabajo final de graduación descrito en el presente documento, se utilizó un enfoque de investigación mixto, el cual incluye tanto el enfoque cuantitativo y cualitativo; descrito por Hernández et al. (2014), citando a Hernández y Mendoza (2008), como:

[...] el conjunto de procesos sistemáticos, empíricos y críticos de investigación e implican la recolección y el análisis de datos cuantitativos y cualitativos, así como su integración y discusión conjunta, para realizar inferencias producto de toda la información recabada (metainferencias) y lograr un mayor entendimiento del fenómeno bajo estudio (p. 534).

Este marco de referencia, para el enfoque cuantitativo, establece que tiene una característica de secuencialidad y necesita probado; estas características no fueron utilizadas para el presente trabajo final. Sin embargo, los autores señalan que mediante este enfoque se pueden realizar mediciones de algún fenómeno o situación estudiados, lo cual en este caso, si fue necesario para el análisis cuantitativo de datos presentes en las ejecuciones del control de la calidad de los proyectos de software del TD.

Por otro lado, de acuerdo con Hernández et al. (2014), el enfoque cualitativo muestra características, que efectivamente, se utilizan en el presente trabajo final de graduación, entre otras:

- Se revisa literatura y se construye el marco para formular la base teórica sobre calidad del software.
- La característica de iteraciones entre etapas predomina en el proyecto.
- Se realiza una inmersión en el área de Gestión de la Calidad de la Unidad y el proceso y las actividades que desempeña, para estudiar de una forma participativa el fenómeno seleccionado.
- Se plantea un problema al inicio del proyecto relacionado a la calidad del software del TD, mas no se sigue un proceso específico desde el principio para resolverlo, sino, conforme se desarrolla el trabajo investigativo, se afinan los elementos de análisis y criterios para abordar la propuesta de solución.
- No se crean ni se prueban hipótesis respecto a la problemática planteada del trabajo final de graduación.
- Se utilizan métodos de recolección y análisis de datos e información no estandarizados ni predeterminados; pues las condiciones y los recursos de estudio de la problemática en la Unidad no se prestan para dicha estandarización.
- Se fundamenta una perspectiva interpretativa del significado de los resultados obtenidos mediante la ejecución de todas las técnicas e instrumentos investigativos que se utilizan en el estudio de la situación del TD.

De esta manera, se realizó una combinación de componentes, datos e información de los enfoques de investigación cuantitativo y cualitativo; por lo tanto, no se puede catalogar como solo de uno de estos. Cabe destacar que se identificó, según Hernández et al. (2014), lo conocido como una preponderancia cualitativa en la mayoría del desarrollo metodológico del trabajo final; pues se desplegaron anidaciones de elementos cuantitativos, tal y como se explica en el diseño de investigación posteriormente. Sin embargo, la mayoría del proyecto siguió un enfoque cualitativo.

Con en el uso de este enfoque mixto, se buscó proveer las características para el presente trabajo final que son descritas de seguido, sustentadas, de igual forma, en lo mostrado por Hernández et al. (2014):

- Propiciar una visión más general y profunda a la vez, del objeto de estudio. Esto debido a que se realiza el estudio de la calidad presente en los proyectos de software del TD, puntualmente, se toma como base las etapas del ciclo de vida de desarrollo de estos (visión general); sin embargo, se profundiza en los componentes relacionados en cada una de estas que fueron vinculantes en el tema estudiado (profundización).
- Producir y analizar datos e información más “ricos” y variados, provenientes de la ejecución de los procesos de calidad de la Unidad y su entorno dentro de la organización por igual; por mencionar algunos: documentación, manuales y demás recursos vinculantes.
- Formular, con mayor claridad, el planteamiento de la situación problemática que presenta la Unidad en la arista de calidad de software, al utilizar varios tipos de datos e información (cuantitativos y cualitativos).
- Potenciar la creatividad teórica, por medio de múltiples procedimientos críticos y, de esa forma, la base teórica sobre calidad de software se convierte en el fundamento de la propuesta de solución.
- Apoyar el desarrollo de nuevas competencias y destrezas en el ámbito de la investigación.

También, una de las principales pretensiones básicas que se buscó con dicho enfoque mixto es la complementación, de acuerdo con lo propuesto por Greene (2007), Teddlie y Tashakkori (2009), Hernández y Mendoza (2008) y Bryman (2008), autores citados por Hernández et al. (2014).

La complementariedad de estos enfoques se empleó con la meta de realizar una mayor comprensión, entendimiento, clarificación e ilustración de los productos o

resultados obtenidos sobre la calidad de software en los proyectos del TD, al ejecutar los métodos cuantitativo y cualitativo propuestos.

Además, aunque los métodos mencionados presentan un enfoque distinto, la complementación descrita se realizó con la meta de sumar las virtudes y las bondades de ambos, a través de la recopilación y el procesamiento de los resultados recabados de la metodología.

### **3.4. Diseño de investigación**

El tipo de diseño de investigación seleccionado para el presente trabajo final fue el mixto de integración; el mismo es descrito por los autores Hernández et al. (2014), quienes a su vez citan a Hernández y Mendoza (2008), como un tipo de diseño donde la máxima es la combinación de los enfoques cualitativos y cuantitativos, de esta forma, se sigue el enfoque mencionado anteriormente.

Además, los autores agregan que, un objetivo de este tipo de diseño es que la combinación propuesta se realice a través de todo el proceso investigativo, o por lo menos, en la mayoría de las etapas de una metodología de investigación; meta requerida para el desarrollo del presente trabajo final, principalmente, al elaborar los capítulos cuatro y cinco, donde se desarrolló un análisis de resultados donde se encuentran datos e información de ambos tipos (cuantitativa y cualitativa).

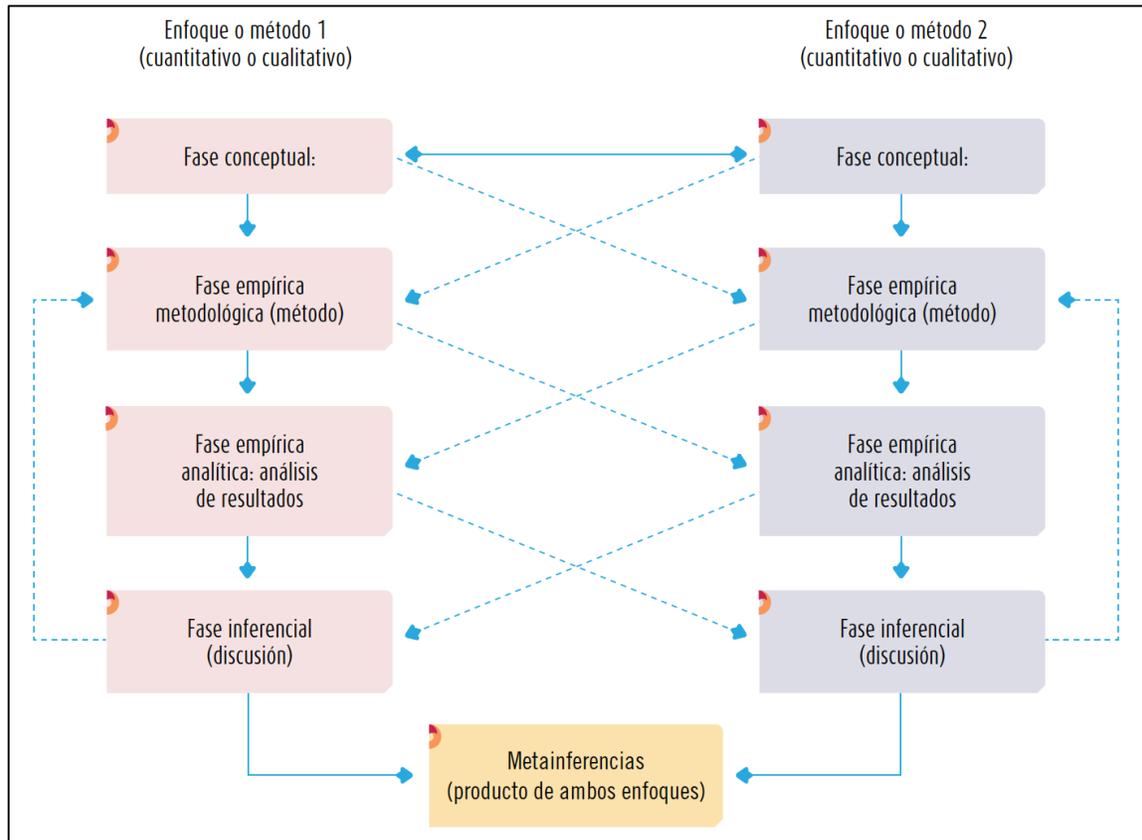
Cinco razones por las cuales se eligió el diseño investigativo mencionado, se detallan seguidamente, junto con su aplicación específica dentro del proyecto (las mismas son descritas de igual forma por Hernández et al. [2014]):

- Recolectar datos e información de tipos cuantitativos y cualitativos en una forma concurrente y en secuencia. Específicamente, se recaban datos cuantitativos con las mediciones de datos obtenidos como resultados de las instancias de ejecución del proceso de control de la calidad del TD (más adelante especificados), también de tres preguntas incluidas dentro un cuestionario aplicado a colaboradores de la Unidad (de igual forma se detalla luego). Por otro

lado, se recopila una mayoría de datos e información de carácter cualitativo que se expondrán en apartados posteriores.

- Como consecuencia de la característica anterior, el análisis que se requiere para los datos e información extraídos para el trabajo final de graduación es de dos tipos por igual: cuantitativo y cualitativo. Dichos análisis se realizan con los métodos específicos para cada enfoque, tal y como se explica más adelante en los apartados 3.7. y 3.8.
- El proceso de investigación permite las iteraciones en sus distintas etapas metodológicas de desarrollo, tal y como se muestra en la Figura 12. Esto se requiere, debido a la constante exigencia de contar con un respaldo teórico al momento de realizar el análisis de resultados y la construcción de la propuesta de solución.
- El diseño de integración permitió lidiar con un tipo de problema que se considera complejo a nivel organizacional, por su existencia dentro del ciclo de vida de desarrollo de los proyectos de software del TD, el cual tiene vínculo directo e indirecto con todas las áreas de la Unidad y representa una de sus principales líneas de acción.
- A pesar de la naturaleza iterativa y entremezclada de los procesos investigativos de este diseño, el reporte de los resultados definitivos del presente trabajo final de graduación se efectúa al finalizar este proyecto, tanto al TD como a la Coordinación del mismo; tal y como se muestra en la etapa de metainferencias presente en la Figura 12.

Figura 12. Descripción de etapas de un diseño de integración



Fuente: Tomado de Hernández et al. (2014), p. 550.

La Figura 12 presenta el desarrollo por etapas de una investigación, de acuerdo con un modelo general de diseño de integración, propuesto por Hernández et al. (2014). Dicho diseño mostró una similitud explícita respecto a los capítulos desarrollados para el presente trabajo final, seguidamente, se explica la semejanza con referencia a las etapas e interacciones mostradas en la Figura 12:

- La fase conceptual de la Figura 12 se representa mediante el desarrollo de los capítulos uno y dos del presente trabajo final de graduación.
- La fase empírica metodológica de la Figura 12 se visualiza como el presente capítulo, el cual muestra la forma de desarrollo del trabajo final.

- La fase empírica analítica mostrada en la Figura 12 tiene su similitud (incluso en el nombre) con el capítulo cuatro del presente documento; es decir, el análisis de resultados. En el mismo, en cierta medida también se desarrolla la siguiente fase inferencial de la Figura 12, debido a que se realiza la descripción de la problemática encontrada, la cual simboliza un tipo de discusión de esta.
- Finalmente, en la Figura 12 se muestra las metainferencias, el resultado del análisis de todos los enfoques, los métodos pertinentes, así como sus datos e información recibidos; el producto final de esta etapa dentro del trabajo final de graduación se podría indicar que es la propuesta de solución, así como las conclusiones y las recomendaciones.

### **3.5. Fuentes de recopilación de datos e información**

Para el desarrollo del presente trabajo final fueron consultadas distintas fuentes de datos e información, seguidamente, se detallan las mismas, según la clasificación que proponen los autores Silvestrini y Vargas (2017):

#### **3.5.1. Fuentes primarias**

De acuerdo con Silvestrini y Vargas (2017), una fuente primaria posee información original; es decir, sin ser modificada después de publicarse, así tampoco ha recibido algún tipo de filtro, interpretación o evaluación por ninguna otro estudio de contenido. Surgen a partir de cualquier tipo de actividad de carácter creativo o nuevo.

Por lo tanto, para el presente trabajo final de graduación se utilizaron las siguientes fuentes primarias:

- Colaboradores del TD.
- Profesionales expertos en calidad de software.
- Normas, estándares y marcos de referencia.
- Plantillas y estándares de la Unidad.
- Artículos científicos.

### **3.5.2. Fuentes secundarias**

De acuerdo con los conceptos establecidos por Silvestrini y Vargas (2017), se define una fuente secundaria como las que provienen de una fuente primaria pero que han sido ya sea sintetizadas, reorganizadas o evaluadas mediante alguna técnica. La preparación y el trato de este tipo de fuentes se enfoca en facilitar y dar el máximo acceso a los contenidos y a las mismas fuentes primarias.

Para el desarrollo del presente trabajo final de graduación se usaron las siguientes fuentes de orden secundario:

- Artículos científicos que contienen revisiones de literatura o estados del arte.
- Plantillas y estándares desarrollados en el TD y que han recibido alguna revisión o cambio.

### **3.5.3. Fuentes terciarias**

Por último, en los tipos de fuentes de información, Silvestrini y Vargas (2017) definen una terciaria como un tipo de guía tanto física como virtual, esta posee datos o información sobre las fuentes secundarias; dentro de este tipo de fuentes, como ejemplos se pueden mencionar: guías de obras que funcionan como referencia, bibliografías, entre otros.

En el presente trabajo se utilizó la fuente terciaria que se detalla a continuación:

- Catálogos de la Biblioteca “José Figueres Ferrer” del TEC.
- Catálogos de la Biblioteca del Sistema de Bibliotecas, Documentación e Información de la Universidad de Costa Rica.

### **3.6. Selección de la muestra**

Según Hernández et al. (2014) se utilizó un muestreo por propósito (elementos cuantitativos y cualitativos), el cual a su vez, Teddlie y Yu (2007) mencionan que se puede utilizar en cualquier diseño que siga el enfoque mixto, como es el caso del diseño mixto de integración presente en este trabajo. Este muestreo se diferencia del

estadístico al permitir utilizar muestras que se consideren pertinentes para el estudio del fenómeno, según requerimientos o propósitos del propio estudio.

De acuerdo con Hernández et al. (2014), es recomendable para investigadores que efectúan un estudio mixto (como el presente) por primera vez la selección de pequeñas muestras con un propósito. El propósito en este caso fue la revisión, descripción y análisis de la problemática de calidad en los proyectos de software del TD. Por este motivo, se tuvo una dependencia directa con el área de Gestión de la Calidad, respecto a los datos e información que se podía utilizar como muestra. La selección de cada una de las muestras se presenta en cada uno de los procedimientos mostrados de forma posterior en el apartado 3.8.

### **3.7. Métodos y técnicas de investigación en general**

Los métodos utilizados para el trabajo final de graduación son descritos por Hernández (2007) y Hernández et al. (2014); los cuales se aplican para tipos de investigación donde se necesita realizar el vínculo entre teoría y práctica. Estos se definen a continuación, junto al detalle de sus características y técnicas respectivas.

#### **3.7.1. Método por índices**

Para la construcción del marco teórico del presente proyecto investigativo, se empleó el método que Hernández et al. (2014) definen como el de índices, el mismo consiste en desarrollar inicialmente, un índice para dicha base teórica, lo más general posible; luego, conforme avanza la extensión de la pesquisa se debe ir afinando hasta llegar a temas y aristas específicas de conocimiento.

#### **3.7.2. Método empírico**

Según Hernández (2007), un método empírico hace posible observar y elaborar datos experienciales para brindar las condiciones necesarias y conocer los hechos que fundamentan y generan las características de los fenómenos en estudio; capacidades requeridas para el trabajo final de graduación. Además, detalla junto a Hernández et al. (2014), cuatro de las principales técnicas de este método, como las siguientes:

- Observación: la establecen como la técnica usada para comprender las características de algún fenómeno en estudio desde una perspectiva externa o interna (depende del fin que se tenga para la misma).
- Medición: técnica para efectuar una descripción y clasificación cuantitativa sobre estadísticas, propiedades, relaciones u otros datos vinculantes de un fenómeno en estudio.
- Entrevista: indican que es un tipo de instrumento investigativo para recabar datos e información (de enfoque cualitativo mayoritariamente) por medio de la interacción entre un entrevistador (quien realiza la entrevista) y un entrevistado (a quien se le aplica la entrevista); además, presentan características como la flexibilidad y la apertura a cambios en el momento de su ejecución. Los autores, a su vez, citando a Grinnel y Unrau (2007), mencionan el tipo de entrevista semiestructurada, la cual presenta una guía de organización para sus preguntas o asuntos; sin embargo, el entrevistador puede cambiarla e introducir o modificar preguntas a conveniencia del estudio. Kerlinger (1983) había sumado por su parte que, para este tipo de técnica, se debe gestionar una serie de preguntas que actúen a modo de sondeo o embudo, o sea, se debe ir desde lo más general a los más específico.
- Cuestionario: conjunto de preguntas en referencia a una o más variables a medir. Describen también que un cuestionario puede contar tanto con preguntas abiertas como cerradas. Las preguntas abiertas muestran una estructura, con el fin de que la persona entrevistada no se limite o tenga un sesgo para su respuesta; por otro lado, las preguntas cerradas contienen opciones o categorías para su respuesta, las cuales fueron delimitadas. Escobar y Bonilla (2009) la definen como una técnica de recolección de datos, la cual es posible que se ejecute mediante una entrevista grupal semiestructurada.

Las técnicas de este método y su representación específica dentro del trabajo final de graduación presente serán detalladas luego, en los procedimientos de

recolección de datos e información, donde se extenderá a profundidad la estructura y otras características de dichas técnicas.

### **3.7.3. Método de procesamiento de los datos y la información**

Para el caso de la selección de datos o información de tipo cuantitativo se hizo uso de una técnica de procesamiento que Hernández (2007) detalla como estadística (descriptiva), la cual fue aplicada de forma manual y con calculadora, para el conteo y la elaboración de porcentajes encontrados en dichos datos.

Por otro lado, para los análisis y los resultados provenientes de datos e información de carácter cualitativo, se utilizaron las técnicas que Hernández et al. (2014) detallan como la codificación y la evaluación temática, propias del enfoque mixto de investigación.

Finalmente, la característica principal que resume el procesamiento y el análisis de los datos y la información es la triangulación, propuesta por Hernández et al. (2014), la cual es una de las pretensiones básicas del enfoque mixto de investigación y con esta técnica, se buscó contar con diferentes fuentes para que el proceso de elaboración de sus resultados muestre un desarrollo robusto y sólido; sobre este y tomando como base el cumplimiento de dicha triangulación, se cimentó la propuesta de solución del presente trabajo final.

### **3.7.4. Método teórico**

Para Hernández (2007), un método teórico busca interpretar de forma conceptual y concreta datos o información de algún fenómeno, hecho o situación, por medio del desarrollo y el análisis a través de técnicas, las cuales, en este caso, siguen un enfoque tanto cuantitativo como cualitativo.

Además, Hernández (2007) expone que cinco ejemplos de las técnicas principales que se utilizan dentro de este campo teórico, las cuales fueron aplicadas para el presente proyecto investigativo, son las siguientes:

- **Análisis:** lo define como una acción intelectual que realiza una separación de los componentes que forman un todo, así también de sus relaciones y demás elementos. Esta tesis la apoya también Ruiz (2007).
- **Síntesis:** establece la síntesis como la posible unión que se puede realizar sobre partes de algún estudio que se analizaron de forma previa.
- **Inducción:** describe el proceso a través del cual se logra una generalización y toma como base una serie de hechos singulares, esto significa que se trasciende de lo específico a lo general.
- **Deducción:** es el procedimiento que se fundamenta en las generalizaciones para desarrollar (de forma inversa a la inducción) inferencias particulares, lo cual significa que se deduce de lo general a lo singular.
- **Modelación:** se entiende como una técnica donde se conceptualiza de forma esquemática y gráfica, algún término, proceso o paradigma de algún fenómeno de investigación.

Todas estas técnicas del método teórico se emplearon para el desarrollo de múltiples etapas y actividades del trabajo final de graduación; sin embargo, cabe destacar su uso concurrente, continuo y enfático para efectuar el análisis de resultados, la propuesta de solución, así como para producir las conclusiones y las recomendaciones.

### **3.8. Procedimientos de recolección – procesamiento de datos e información**

La aplicación de las técnicas de investigación detalladas en los métodos de investigación previamente, determinan los procedimientos utilizados para recabar, procesar y analizar los datos e información del trabajo final. El resultado y evidencias de estos procedimientos se muestran en el capítulo cuatro del presente documento. De seguido, se muestra el detalle de los instrumentos y dichas técnicas utilizadas dentro del presente trabajo final.

### 3.8.1. Observación

Esta técnica se utilizó para observar el comportamiento y demás elementos de la ejecución de los procesos de calidad del TD. Según Hernández et al. (2014), la observación fue de tipo participativa; pues el sujeto es quien ejecutó cuatro instancias del proceso de control de la calidad de software sobre el proyecto *Evaluaciones*<sup>8</sup> de esta entidad; por este motivo, surgió una interacción entre el sujeto y el objeto de estudio en particular. El resultado de la observación se visualiza en la descripción de la metodología de los proyectos de software de la Unidad y modelado del proceso de control de calidad presentes en la sección 4.2.4.1.

### 3.8.2. Mediciones

Para el caso del presente trabajo, se realizó una medición y un análisis cuantitativo descriptivo sobre los resultados de cuatro instancias de ejecución del proceso de control de la calidad de software del TD sobre el proyecto *Evaluaciones* (ver apéndice N) de dicha Unidad.

Además, se llevó a cabo un tipo de medición de opiniones que tienen los colaboradores sobre algunos criterios respecto a la calidad de los proyectos de software que se desarrollan en el TD.

### 3.8.3. Entrevistas individuales

Se aplicaron dos entrevistas diferentes (ver apéndice Ñ y apéndice O) a dos profesionales expertos en la calidad de software (una entrevista a cada profesional). Los entrevistados fueron: uno) el licenciado en Ingeniería Informática y Calidad de Software, Luis Alexander Calvo Valverde (en adelante el entrevistado uno) y el máster en Ciencias de la Computación, Ignacio Trejos Zelaya (en adelante el entrevistado dos), experto reconocido de calidad de software a nivel costarricense y latinoamericano y representante por Costa Rica ante la *Hispanic America Software Testing Qualifications Board* (en adelante HASTQB), “*asociación jurídica trans-sectorial y trans-nacional sin fines de lucro fundada por empresas, instituciones, organizaciones y*

---

<sup>8</sup> Evaluaciones es un proyecto de software del TD.

*personas especializadas en el campo de testeo y la industria del software” (HASTQB, 2017).*

Estas entrevistas se efectuaron para confirmar la base teórica adquirida, la cual produjo como resultado la elaboración del marco teórico del proyecto investigativo y con el objetivo de consultar, de igual forma, el criterio de dichos expertos sobre los componentes de la propuesta de solución desarrollada.

Tales entrevistas son semiestructuradas y se aplicaron personalmente, grabadas mediante un dispositivo con grabadora de voz y transcritas para su debido análisis. Esta última actividad, la cual se elaboró según lo que Hernández et al. (2014) definen como categorías de análisis. Los autores señalan que estas categorías pueden ser *“conceptos, experiencias, ideas, hechos relevantes y con significado”*, los cuales surgen como elementos principales de análisis de un tipo de instrumento como la entrevista.

La entrevista para el profesional uno es diferente a la aplicada al profesional dos; pues el primero realizó un proyecto sobre aseguramiento de la calidad del software en el TD; tal y como se indicó en el apartado 1.4.1.3. sobre los trabajos similares internos de la Unidad. Por este motivo, la entrevista para el profesional uno contiene preguntas enfocadas en ese proyecto para indagar sobre su consistencia y finalidad.

#### **3.8.4. Revisión documental**

Se realizó una recopilación de ocho documentos que posee la Unidad vinculantes a la calidad en el desarrollo de software, la cual se encuentra en el apartado 4.2.3. Dentro de los tipos de documentos e información recopilados se encuentran: plantillas, guías, estándares, modelos de procesos, entre otros documentos. Todos estos se hallan disponibles en formatos digitales y fueron descargados de una comunidad perteneciente a la plataforma de aprendizaje electrónico desarrollada y soportada por el TD.

### **3.8.5. Cuestionario**

Se aplicó un cuestionario (ver apéndice P) a diez colaboradores de la Unidad, quienes fungieron como representantes de cada área del TD que tiene relación directa o indirecta con el desarrollo del software y su calidad: coordinación general, administración de proyectos, arquitectura y soporte, desarrollo, comunicación visual, capacitación a usuarios, soporte a usuarios y producción audiovisual (áreas presentes en el capítulo uno, Figura 2).

Dicho cuestionario contenía preguntas tanto abiertas como cerradas y fue auto suministrado por medio de un correo electrónico; se utilizó la herramienta llamada *Google Forms*, la cual brinda la funcionalidad de cuestionarios en línea y reporte gráfico automático de resultados, de acuerdo con Google (2017b). Además, su análisis respectivo también se concretó mediante el uso de categorías, al igual que las entrevistas a los profesionales expertos en calidad de software.

### **3.8.6. Modelación de procesos de negocio**

Se modeló la ejecución de las cuatro instancias observadas del proceso de control de la calidad del software en un proyecto del TD y se efectuó una descripción general de la misma, respecto a la forma como se interactúa con la metodología de desarrollo de dichos proyectos de la Unidad; esto, para encontrar las causas de la problemática que presenta esta entidad en los procesos de calidad mencionados. Esta modelación se encuentra en el apartado 4.2.4.1.

## **3.9. Instrumentos adicionales de apoyo**

Además de los métodos y las técnicas de investigación detalladas anteriormente, también se requirió utilizar dos instrumentos de apoyo, con el fin de realizar el análisis final de la etapa de resultados y completar la triangulación de datos que se estaba buscando. De seguido, se describen estos instrumentos.

### **3.9.1. Diagrama de Ishikawa**

De acuerdo con Madison (2005), un diagrama de Ishikawa (nombrado así por el creador del diagrama y la técnica del mismo: Kaoru Ishikawa), conocido también como diagrama de espina de pescado o causa y efecto, expone mediante una explicación

esquemática las causas y el efecto de estas; este efecto principal es la problemática estudiada. En este caso, el diagrama de Ishikawa se utilizó para abordar la problemática de calidad del software del TD e identificar sus causas principales.

### **3.9.2. Análisis comparativo**

De acuerdo con Tonon (2011), quien a su vez utiliza lo expuesto por Sartori (1984), el método de análisis comparativo *“tiene como objetivo la búsqueda de similitudes y disimilitudes [...]”* entre un grupo de elementos a comparar.

Para Tonon (2011), el análisis comparativo ejecuta una confrontación de dos o varias propiedades enunciadas para dos o más objetos de estudio, en determinado momento y bajo ciertos criterios.

En el caso específico del presente trabajo final de graduación, se utilizó este tipo de análisis para comparar cuatro herramientas de software (consultar el apartado 4.3.2.), las cuales se recabaron para efectuar la propuesta de integración del abordaje de solución con un elemento de este tipo. Dicha tabla expone las categorías de análisis seleccionadas en conjunto con la encargada responsable del trabajo final por parte de del TD.

### **3.10. Etapas de la metodología**

La metodología de la investigación desarrollada en el presente trabajo, se dividió en tres etapas, con el fin de responder a los objetivos establecidos para el proyecto, estas etapas son:

#### **3.10.1. Formulación y confirmación de base teórica sobre aseguramiento y control de la calidad de software**

Por medio del método de índices que detallan Hernández et al. (2014) y se describió anteriormente, se construyó el marco teórico, el cual proporciona la base para el trabajo final de graduación.

Posteriormente, se realizaron las entrevistas a profesionales expertos en el área de la calidad del software, donde se incluyeron elementos para confirmar la selección y la pertinencia de esta base para el proyecto.

### **3.10.2. Revisión, descripción y análisis de problemática actual de calidad en los proyectos de software del TEC Digital**

Por medio de la técnicas e instrumentos de recolección de datos e información que se detallaron anteriormente, se revisó, describió y analizó el panorama actual de los proyectos de software del TD, en cuanto a su calidad, lo cual genera la problemática abordada del presente trabajo.

Las actividades o los pasos principales que se realizaron dentro de esta etapa, se resumen en la siguiente lista:

- Análisis cuantitativo: se incluyó los resultados y el análisis de las mediciones realizadas sobre cuatro instancias del proceso de control de la calidad de software en la Unidad, específicamente, sobre el proyecto Evaluaciones.
- Análisis de cuestionarios: se documentó y analizó las respuestas de diez colaboradores del TD, a quienes se les consultó sobre algunos criterios de la calidad de software que presentan los proyectos de esta organización.
- Análisis de documentos: se realizó la revisión documental de ocho archivos y se identificaron elementos de análisis para los documentos y demás recursos de información con los cuales cuenta la Unidad.
- Descripción y análisis de problemática: se describió de forma general, elementos asociados a la calidad de software en los proyectos de software del TD, basados en la observación de cuatro instancias del proceso de control de calidad sobre el proyecto Evaluaciones; esto dio como resultado lo siguiente:
  - Descripción del ciclo de vida de desarrollo de los proyectos de software.
  - Descripción del proceso de control de la calidad de software.
- Diagrama de Ishikawa: a modo de puntualización, se desarrolló un diagrama de Ishikawa, en donde se pretendió identificar las principales causas que provocan la problemática de calidad en los proyectos de software de la Unidad.

- Estudio analítico de las causas de la problemática: a partir de las causas identificadas en la actividad previa a esta, se realizó un estudio analítico de las mismas por medio de una triangulación de criterios que buscó identificar la brecha entre lo que dicta la base teórica y la situación actual del TD para la calidad de sus proyectos de software.
- Síntesis de problemática: finalmente, se realiza una síntesis que resume la problemática por medio de prosa, la cual se concluyó por medio de todas las actividades, técnicas e instrumentos ejecutados de forma previa.

### **3.10.3. Elaboración de diseño de sistema y su alcance para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital**

Se realizó un análisis de los requerimientos del sistema para el aseguramiento y control de la calidad en los proyectos de software de la Unidad, el cual es el elemento principal que conforma la propuesta de solución.

Como resultado, se elaboró el diseño del sistema mencionado, junto con una propuesta de integración de un conjunto de herramientas de software para el mismo y de un modelo de calidad para aplicar en este. Además, se incluye un plan de capacitación sobre calidad de software, así como del mismo sistema (para realizarse de forma posterior al trabajo final de graduación).

Finalmente, también se propuso un plan de implementación (en el apartado 4.3.4. del capítulo cuatro) del sistema para la calidad en los proyectos de software del TD. Este plan se desglosa por fases y contiene las actividades, los roles y una estimación del recurso (tiempo) necesario para la implementación de dicho sistema.

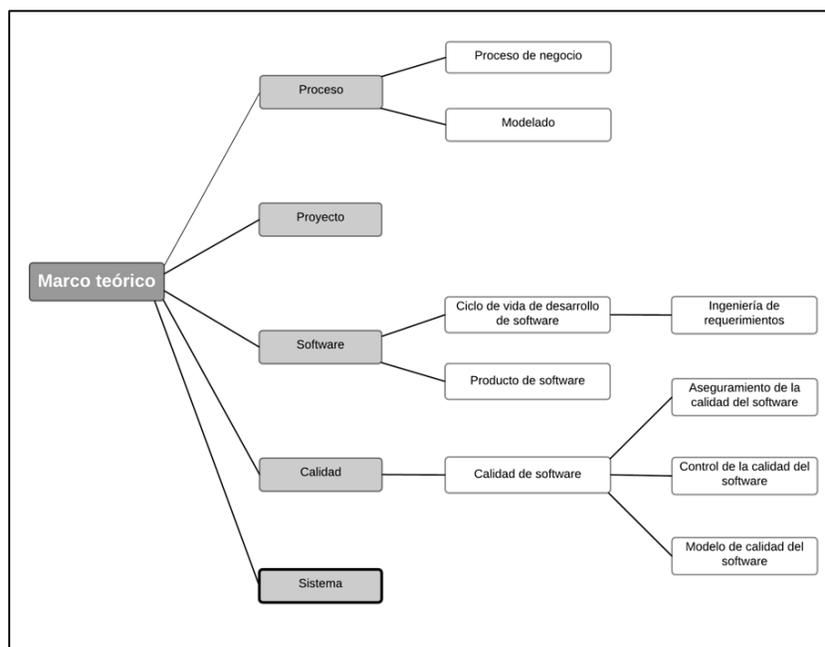
## CAPÍTULO IV: ANÁLISIS DE RESULTADOS

En el presente capítulo se muestra el resultado de la aplicación de los métodos, técnicas e instrumentos investigativos utilizados para el desarrollo del trabajo final de graduación. Dicho resultado se desglosa mediante las etapas definidas previamente, en la metodología, las cuales se profundizan de seguido. De forma agregada y, aunque no se encuentran dentro de la etapas metodológicas planteadas en el capítulo anterior, se detallan al final del presente capítulo, los trabajos futuros y un cambio realizado sobre el desarrollo del proyecto.

### 4.1. Formulación y confirmación de base teórica sobre aseguramiento y control de la calidad de software

Para la primera etapa se recopiló la base teórica sobre aseguramiento y control de la calidad de software, fundamentada en la investigación realizada sobre las mejores prácticas, normas, marcos y metodologías vinculantes con el tema y área de calidad de software. La estructura de índices, descrita en el apartado 3.5.1., apoyó en el desarrollo de esta base, la misma se puede consultar en la Figura 13.

Figura 13. Diagrama de índices del marco teórico



Fuente: Elaboración propia.

La presente investigación se elaboró mediante la consulta de las bases de datos electrónicas asociadas al TEC; principalmente, se extrajo información del catálogo electrónico de la *Association for Computing Machinery (ACM)*, de la *IEEE Digital Library*, de *EBSCOhost Web* y del propio repositorio del TEC.

La base teórica sobre aseguramiento y control de la calidad fue forzada a una validación mediante las entrevistas a dos profesionales expertos en el área de la calidad del software, como se indicó en el apartado 3.8.3. Los resultados de dichas entrevistas y su análisis correspondiente, se detallan a continuación:

#### **4.1.1. Primera entrevista**

L. A. Calvo (comunicación personal, 4 de abril, 2017) detalló que la calidad de software puede ser vista como el cumplimiento de los requerimientos del cliente, donde se provee al mismo tiempo satisfacción en su experiencia de uso y, además, dicho software debe haber sido producido con procesos efectivos.

Además, el entrevistado indicó que, si se busca tratar el tema de calidad de software, primero se debe introducir lo referente a la gestión de la misma; ésta, a su vez, se dividirá en dos temas principales: el control y el aseguramiento.

Ahora bien, si se profundiza en una de esas divisiones de la gestión de calidad, el control de calidad de software, L. A. Calvo (comunicación personal, 4 de abril, 2017) definió este término como el cumplimiento que se obtiene cuando se compara el resultado de algún proceso de desarrollo de software respecto a lo que se planteó se iba a realizar (planteamiento conocido como los requerimientos de un software). Es decir, si se dice que de algún proceso o actividad se tendrá cierto resultado, se valide que sea el producto esperado y no otro.

El aseguramiento de la calidad de software, L. A. Calvo (comunicación personal, 4 de abril, 2017) lo explica como el conjunto de procesos y actividades requeridos a través de todo el ciclo de desarrollo del mismo, para propiciar la calidad de sus productos finales. Por este motivo, se puede implicar entonces que, para la ejecución efectiva de estos procesos y sus actividades correspondientes, la elaboración de un

camino o ruta sistematizada de ejecución, donde se detalle las herramientas o artefactos necesarios para cada etapa o fase, es un posible abordaje sobre el cual una entidad puede implementar o mejorar dicho aseguramiento.

Por otro lado, a raíz del proyecto que elaboró en el TD entre el año 2015 y el 2016 para el desarrollo de un plan de aseguramiento de la calidad de software de dicha instancia, L. A. Calvo (comunicación personal, 4 de abril, 2017) detalla que encontró en la Unidad una gran problemática en su ciclo de desarrollo de software: la gestión de los requerimientos y su trazabilidad a lo largo de los proyectos. El entrevistado explicó que el TD no mantenía ningún registro de cambios de los requerimientos y, en muchos casos, indagó que ni siquiera se documentaban los requerimientos de los proyectos de software.

Por este motivo, L. A. Calvo (comunicación personal, 4 de abril, 2017) indicó que en ese momento decidió cambiar el enfoque de su proyecto e iniciar por dotar a la Unidad de una herramienta para ofrecer soporte a la gestión y la trazabilidad de los requerimientos de los proyectos de software. Esta herramienta se desarrolló mediante una hoja de cálculo de *Google Drive*, en la cual se realiza el registro de los requerimientos y los cambios sobre los mismos tienen un control y seguimiento en unidades de tiempo.

Además de esa problemática, el entrevistado uno indicó también otra situación que generaba problemática, no sólo en el control de la calidad de los proyectos de software, sino en todo el ciclo de vida de los mismos: una ausencia de comunicación asertiva entre las diferentes áreas por medio de sus procesos e interacciones. Este problema de igual forma se encuentra en los hallazgos del entrevistado sobre el área de Gestión de la Calidad; pues determinó que no hay proceso definido para la comunicación de aprobaciones o etapas en las cuales se encuentra la revisión de la calidad.

De igual forma, en el desarrollo del trabajo investigativo, L. A. Calvo (comunicación personal, 4 de abril, 2017) menciona que, otra problemática encontrada sobre la actual gestión de la calidad de software en la Unidad es que, no se mantiene

una sola base de evaluación de la calidad, no existen parámetros determinados y objetivos que permitan verificar y validar la calidad y esta evaluación se realiza únicamente, después del desarrollo, lo cual eleva el costo de posibles correcciones o mejoras.

Por otra parte, L. A. Calvo (comunicación personal, 4 de abril, 2017) agrega que, algunos pasos claves que guiarán al TD hacia el mejoramiento, en términos de calidad de sus productos, son la repetición y la constancia en la ejecución de sus procesos y actividades, junto a utilizar estándares de artefactos o herramientas. La repetición, según el entrevistado, propiciaría la capacidad de medir tanto los productos de los procesos de calidad como los procesos en sí. Además, detalla utilizar herramientas estándares para la ejecución de los procesos de calidad, brindaría condiciones favorables en búsqueda de la continuidad mencionada antes.

Por último, respecto a la propuesta de solución que se planteó en la entrevista, según el criterio de L. A. Calvo (comunicación personal, 4 de abril, 2017), una herramienta de software que brinde soporte a los procesos y las actividades del aseguramiento y el control de la calidad de la Unidad, genera de esta forma un sistema (desde un punto de vista integral), lo cual constituye un abordaje válido para la problemática del TD. Además, realiza la sugerencia técnica de consultar herramientas que ya proveen esas funcionalidades y pueden ser de ayuda para esta organización, tales como: *Tarantula*, *TestLink* o *Selenium*.

Concluye al mencionar que, la generación de estadísticas e información a partir de esa herramienta de software que se propone para la propuesta de solución, sería un buen apoyo a los colaboradores de la Unidad que les compete la toma de decisiones.

#### **4.1.2. Segunda entrevista**

I. Trejos (comunicación personal, 25 de abril, 2017) explica que la calidad de software, según la literatura que se puede consultar sobre el tema, es definida como la satisfacción de los requerimientos o las necesidades por los cuales el software fue desarrollado. Sin embargo, explica que este concepto va más allá de esa “corta definición”.

De acuerdo con I. Trejos (comunicación personal, 25 de abril, 2017), un paso clave para buscar esa calidad del software, es trascender de lo implícito a lo explícito, refiriéndose principalmente, a todo el ciclo de vida de su desarrollo. Esto significa que se deben realizar esfuerzos puntuales, procesos o actividades mediante una ruta sistematizada y planificada, la cual brinde las condiciones ideales para que el software elaborado en una entidad vaya contando con atributos de calidad tanto funcionales como no funcionales desde etapas muy tempranas de su concepción. Cuatro actividades o iniciativas que el entrevistado uno menciona se pueden realizar, son:

- La utilización de la ISO/IEC 9126 como modelo de calidad de software. Explica que utilizar este modelo permite priorizar los atributos de calidad no funcionales buscados para el software que se desarrolla. Además, detalló que este modelo funciona de forma práctica para implementarlo, se refiere a que tiene un menor costo en recursos para su adaptación en una entidad, al compararlo, por ejemplo, con el modelo presente en las ISO 25000.
- Toma eficiente y trazabilidad de requerimientos, los cuales son el insumo principal para construir un diseño de software que satisfaga las necesidades de los clientes y dicha relevancia se mantiene, de igual forma, para todas las etapas siguientes en el desarrollo del software. Recomienda el uso del IEEE 830-1998 para este fin.
- Elaboración de prototipos (funcionales o no funcionales) en etapas tempranas de desarrollo del software que se elaborará, esto con el fin de verificar y validar atributos de calidad, los cuales permitan disminuir costos sobre cambios que se podrían requerir en etapas más avanzadas en el ciclo de vida de desarrollo de los sistemas digitales.
- Incorporación de otros estándares de organismos internacionales, tales como: el IEEE std 1028-2008 para revisiones y auditorías de software, el IEEE 829-2008 para la documentación de los planes de prueba o el IEEE 1016-2009 para la elaboración del diseño de software.

Otro punto que resalta I. Trejos (comunicación personal, 25 de abril, 2017) es el aspecto socio-cultural que se encuentra intrínseco en una organización, el cual es una arista que se debe analizar para proponer alguna mejora sobre procesos de calidad y del cual no se ha extendido mucho en la literatura sobre calidad de software. La ISO 9000 (propuesta por la ISO [2015]) por su parte, respecto a este tema, detalla un principio del enfoque de calidad relacionado a lo que comenta el entrevistado dos, el cual lleva como nombre “*compromiso de los colaboradores*”.

Por otro lado, I. Trejos (comunicación personal, 25 de abril, 2017) indica que si se quiere dar una formalidad al tema de gestión de la calidad del software (incluye el aseguramiento y el control) en una organización, se debe tomar en cuenta tres elementos que de forma integrada ejecutan el ciclo de vida de desarrollo del software: roles, actividades y artefactos. El rol está compuesto del comportamiento y las responsabilidades atribuidas a los colaboradores dentro del proceso de construcción del software. Las actividades son las instancias prácticas y ejecuciones de dichos roles y los artefactos son las herramientas o productos que se utilizan para ejecutar o se tienen como resultados de las actividades mencionadas.

El entrevistado dos menciona que, para la propuesta de solución del presente trabajo final, se puede abordar esta “tríada” de elementos para definir los componentes que formarán el sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD; de esa forma, se genera un concepto al cual llama “tríada de calidad”. Debido a que, sobre estos elementos es posible establecer la medición de los procesos de calidad y mejorarlos de ahí en adelante.

En cuanto a los roles, I. Trejos (comunicación personal, 25 de abril, 2017) señala que es necesario entrelazar responsabilidades claras y explica que esto dependerá mucho de la cantidad y la disponibilidad de recursos humanos con los cuales cuente la Unidad para las mismas. Respecto a las actividades y los artefactos, el entrevistado detalla que se puede adaptar lo dispuesto por organizaciones o cuerpos de conocimiento de la IEEE y otras organizaciones, tales como:

- *The Guide to the Software Engineering Body of Knowledge* (en adelante SWEBOK).
- El IEEE 730-2014 sobre los procesos para el aseguramiento de la calidad de software (la versión anterior fue utilizada por el entrevistado uno para desarrollar un plan de aseguramiento de la calidad para el TD).
- El IEEE std 830-1998 para la gestión de los requerimientos del software de la Unidad.

Por otra parte, al conocer de forma superficial la situación problemática con la cual cuenta la Unidad en el área de gestión de la calidad del software, I. Trejos (comunicación personal, 25 de abril, 2017) comenta que los esfuerzos de verificación que se están realizando son positivos; sin embargo, sería positivo agregar, adicionalmente, la validación del software construido.

Lo anterior significa que no sólo se verifique que se está construyendo software con calidad, la cual cumple con ciertos estándares y parámetros a nivel interno del TD, sino también, se está asegurando que dicho software cumple y satisface a cabalidad los requerimientos y las necesidades de los clientes, en este caso, son la comunidad TEC. Se debe empezar por acciones como la elaboración del plan de pruebas en conjunto con los colaboradores encargados de la gestión de la calidad; pues en la actualidad, dicha tarea la realizan solo los equipos desarrolladores.

Por otro lado, I. Trejos (comunicación personal, 25 de abril, 2017) menciona que en Costa Rica, a nivel curricular de programas de las instituciones de educación superior que forman sobre computación, desarrollo de software y temas relacionados, se encuentra una carencia de formación que explicaría el porqué muchos profesionales a nivel costarricense no cuentan con esas habilidades técnicas para desarrollar un aseguramiento y control de la calidad en esos ámbitos.

Finalmente, I. Trejos (comunicación personal, 25 de abril, 2017) señala, con referencia a la propuesta de solución de implementar una herramienta de software para la automatización de actividades de soporte del aseguramiento y el control de la calidad

del software de la Unidad, que dicha herramienta sería un gran apoyo para el TD. Además, explica que se podría realizar una integración desde el inicio de los proyectos de software de esta entidad, al definir casos de pruebas e inspecciones basadas en los casos de uso de los requerimientos, los cuales se definen en las primeras etapas.

#### **4.1.3. Análisis de entrevistas**

A partir de las entrevistas que se aplicaron a los expertos en calidad de software, surgen tres categorías que sintetizan los criterios emitidos por los mismos. Estos criterios funcionan como aristas de análisis de los resultados de los instrumentos investigativos y se exponen a continuación:

##### **4.1.3.1. Definiciones sobre calidad de software**

Los dos profesionales expertos entrevistados coinciden en que la calidad de software se define como la satisfacción de los requerimientos o las necesidades de los usuarios finales; concepto que se refuerza con lo presente en el marco teórico (apartado 2.9.) donde se define ese término como *“la capacidad de un producto de software de satisfacer las necesidades implicadas”*.

Además, el entrevistado dos adiciona que el concepto de calidad de software va más allá de esa definición, por ejemplo, el entrevistado uno, mencionó que además de satisfacerse las necesidades y los requerimientos de los usuarios finales, se debe velar porque el ciclo de vida de desarrollo también cuente con características de calidad, como la efectividad en la utilización de los recursos necesarios.

Por otra parte, según el criterio del entrevistado uno, el aseguramiento de la calidad del software se establece como el conjunto de procesos y actividades que se requiere a lo largo de todo el ciclo de vida de desarrollo del mismo, para propiciar el nivel de calidad buscado por una entidad. El entrevistado dos comparte esta definición y la especifica con otras palabras, como la forma sistemática y planificada en la cual se integra la tríada de calidad (roles, actividades y artefactos) dentro de dicho ciclo de desarrollo, para lograr la calidad esperada para sus productos.

Por esta razón, como se muestra en el apartado 2.11., Pressman (2010) menciona que el aseguramiento de la calidad debe ser una actividad sombrilla; es decir, que se presente en todas las etapas del ciclo de desarrollo.

Por otro lado, el control de la calidad del software, el entrevistado uno lo caracteriza como el grado de cumplimiento que se encuentra en las etapas avanzadas del ciclo de vida de desarrollo o bien, en los productos de software, una vez elaborados.

Finalmente, unificando los conceptos detallados por los entrevistados, se puede decir entonces que, para satisfacer los requerimientos y las necesidades de los usuarios finales, se debe planificar y sistematizar los elementos (roles, actividades y artefactos) que velarán porque el nivel de calidad buscado se alcance y cómo se medirá ese cumplimiento; presentándose eso sí, ciertas condiciones y objetivos de niveles definidos por dichos usuarios.

El tipo de interacción de un sistema como el que se menciona, está detallado, de forma similar, en lo propuesto por Avella y Gómez (2011) en su metodología integrada sobre el proceso de construcción de software para aplicar inspecciones y pruebas de calidad.

#### **4.1.3.2. Elementos clave para la calidad de software**

A través del resumen y análisis de ambas entrevistas se identificaron aspectos o elementos claves que se consideran insumos fundamentales para la elaboración de la propuesta de solución, los mismos se explican seguidamente:

- El entrevistado uno confirma que dos características fundamentales que se deben buscar con el diseño del sistema para el TD, son la repetición del mismo, tipo de procesos y actividades de aseguramiento y control de la calidad de su software, así como la constancia con la que se deben realizar.
- El entrevistado dos, comenta se debe utilizar idear y planificar la consistencia e implementación de la propuesta de solución, con la premisa eso sí, del costo de la calidad; término que se abordó también en el apartado 2.11. por Pressman

(2010), donde se detalla los beneficios en términos de recursos que se obtienen al realizar un aseguramiento y control de la calidad del desarrollo de software desde etapas iniciales de dicho ciclo.

- Tanto el entrevistado uno como el dos, así también Avella y Gómez (2011), quienes se citan en el marco teórico, coinciden en la utilización de estándares (con cierto nivel de adaptación) para ejecutar un sistema o metodología de gestión e integración de la calidad en el desarrollo de software, tales como:
  - ISO/IEC 9126 como modelo de evaluación de calidad de software dentro de las etapas, actividades y artefactos posibles de la Unidad.
  - IEEE 730-2014 para la elaboración de un plan de aseguramiento de calidad de software.
  - IEEE std 830-1998 como estándar para gestionar los requerimientos del software del TD.
- Adicionalmente, el entrevistado dos sugiere darle importancia al aspecto socio-cultural de una organización al momento de implementar una propuesta de mejora como la que se planteó en el presente trabajo final de graduación. Explica que, con este tipo de proyectos, siempre surge una gran resistencia al cambio; por lo tanto, se debe tratar no únicamente, la parte de software, sistemas, metodologías y demás temas técnicos, sino también la parte humana; para lograr de esta forma lo que la ISO 9000 define como el compromiso de los colaboradores, principio fundamental de un sistema de gestión de calidad.
- Un aspecto importante que resalta el entrevistado dos, es la transición de lo implícito a lo explícito, por medio de esfuerzos que puntualicen la solución a las necesidades que presenta la Unidad para sus procesos de calidad de software, específicamente, menciona que se pueden realizar tres acciones:

- Elaboración y trazabilidad de requerimientos con base en un estándar. Acción que apoya el entrevistado uno; pues menciona que esta acción resulta clave para definir el nivel de calidad alcanzado y, en la actualidad, según su observación y criterio, en el TD no se está realizando de una forma estándar y formal.
- Prototipado en etapas tempranas del ciclo de vida de desarrollo de proyectos de software; ya sea de tipo no funcional o incluso funcional, se refiere a elementos visuales como *wireframes*, modelos de *interfaces* gráficas o funcionales en aplicaciones donde se establezca algún tipo de navegación entre las vistas del software.
- Definición de casos de prueba con base en los casos de uso (cuando esta técnica fuera utilizada en la Unidad para el detalle de la ingeniería de requerimientos). Detalla esta acción, pues de acuerdo con la línea definida para calidad de software, los productos de su desarrollo deben satisfacer los requerimientos y las necesidades de los usuarios finales, quienes se deberían tener descritos en los casos de uso.

#### **4.1.3.3. Criterio sobre propuesta de solución**

En la ejecución de las entrevistas a los dos profesionales expertos se les expuso, de forma general y breve, indicios de las características y elementos de la propuesta de solución elaborada en el presente trabajo final de graduación. Ambos entrevistados concluyen en que los elementos planteados son pertinentes a una problemática como la que se comentó presenta el TD en el área del aseguramiento y control de la calidad de software.

Además, el entrevistado uno enfatizó sobre la importancia de elaborar antes de su implementación, un plan de implementación con etapas y actividades, claramente, definidas y coherentes, con el objetivo primordial de que el sistema sea aceptado por la Unidad TD y, aún más relevante, utilizado.

## **4.2. Revisión, análisis y descripción de problemática actual de calidad en los proyectos de software del TEC Digital**

Para describir la problemática presente en la actual gestión de la calidad en los proyectos de software del TD, se requirió el análisis de distintos elementos, aspectos, documentos y actividades que posee y realiza dicha Unidad. Con base en esta situación, se identificaron en los siguientes sub apartados (4.2.1. al 4.2.4) y en el apartado 4.3. del presente capítulo, las principales causas, consistencia y efectos de esa problemática.

De esta manera, a través de las técnicas y los instrumentos de recolección de datos e información que se detallaron de forma previa, se revisó, se analizó y se describe seguidamente, la situación actual de los procesos de calidad del TD y todo lo vinculante a estos.

### **4.2.1. Análisis cuantitativo de datos e información**

Tal y como se explicó en el apartado 3.7.2., se llevó a cabo el análisis cuantitativo descriptivo mediante la observación participativa y la medición del proceso de control de la calidad (mostradas en el apartado 4.2.4.1.), específicamente, sobre cuatro instancias del proceso, realizadas al proyecto de Evaluaciones del TD en el 2016 (ver apéndice N).

Los datos y la información que se muestran a continuación, fueron medidos e interpretados según los resultados presentes en las revisiones de dicho proyecto, las cuales se efectuaron al utilizar el plan de pruebas para el proyecto seleccionado (plan desarrollado según la Plantilla de Control de Calidad que se muestra en el anexo A).

Con el presente análisis, se encontraron seis hallazgos significativos para la propuesta de solución, los cuales se desglosan seguidamente:

- La cantidad de hallazgos (tipo error) en proporción al número de tareas revisadas para cada instancia de proceso (errores divididos entre la cantidad de tareas revisadas) fue aumentando de forma progresiva: en marzo una proporción de 0.18, mayo de 0.42, junio de 0.76 y setiembre de 0.69. Este

suceso coincide con el cambio que se realizó, con criterio empírico en ese momento, donde conforme avanzaban las instancias, se solicitó al equipo desarrollador que se enviara a revisión de calidad en etapas más tempranas (en desarrollo específicamente), lo cual ocasionó que se encontraran más errores en dicha fase y no de forma posterior, al liberar Evaluaciones a la comunidad TEC.

- El porcentaje de errores respecto al total de hallazgos fue aumentando en cada instancia (se inició en un 46% y se finalizó con el 100%); mientras que el porcentaje de mejoras del total de hallazgos disminuyó hasta reducirse a cero en la última revisión. Esto se podría interpretar de la siguiente forma: conforme se construían los lineamientos de mejora sobre Evaluaciones, las siguientes iteraciones de desarrollo seguían esas instrucciones, así que, si se presentaban hallazgos posiblemente, iban a ser errores.
- El promedio de respuesta (determinado en días por el período comprendido entre la emisión de los hallazgos y cuando se recibe una respuesta por parte de los desarrolladores) no sigue alguna tendencia; sin embargo, a pesar de que dicho promedio depende de diversos factores (disponibilidad de personal para el proyecto, carga de trabajo en otros proyectos, disposiciones directas de la coordinación del TD, entre otros), se determinó que sería importante emplearlo dentro de la propuesta de solución a modo de semáforo de gestión sobre el proceso de control de la calidad del software.
- El promedio de días que se necesitaron para resolver los hallazgos encontrados, también resulta un factor que se puede consultar como un indicador para valorar el tiempo de respuesta de los equipos desarrolladores ante esas situaciones, específicamente, para determinar el grado de disponibilidad para el proyecto.
- Se denota una ausencia de información de registro general de las revisiones, por ejemplo: fechas exactas de revisión, cantidad de visitas al documento, registro de actividad por usuario y fecha (bitácora) y descripción de hallazgos por número de instancia del proceso y versión del software.

- De igual forma, no se determinaron datos para hacer posible la evaluación del desarrollo de software según las métricas de calidad de la ISO/IEC 9126; dentro de dichas proporciones faltantes se podrían mencionar:
  - Reincidencia de hallazgos por cada tipo.
  - Esfuerzo requerido medido en unidades (minutos, horas o días) de tiempo para revisión y corrección de los hallazgos.
  - Cantidad y tipos de errores que se asocian a atributos específicos de calidad ante su aparición.
  - Porcentaje de cumplimiento de requerimientos de acuerdo con los casos de uso propuestos.

Estos hallazgos recabados de los datos de ejecuciones de control de calidad sobre Evaluaciones, resultaron claves para determinar el estado actual de registro respecto a dicho proceso, definir cuáles de estos se podrían implementar en la propuesta de solución y cuáles elementos se debían priorizar para su incorporación.

#### **4.2.2. Análisis de cuestionario**

Como se mencionó en el apartado 3.8.5, se aplicó un cuestionario (ver apéndice P) a diez colaboradores de la Unidad, los cuales representaban las áreas que componen el TD y tienen relación directa o indirecta con el desarrollo del software y su calidad. De seguido, se describe el análisis correspondiente de las respuestas recibidas por medio de la identificación y la explicación de las siguientes categorías:

##### **4.2.2.1. Conceptualización de calidad**

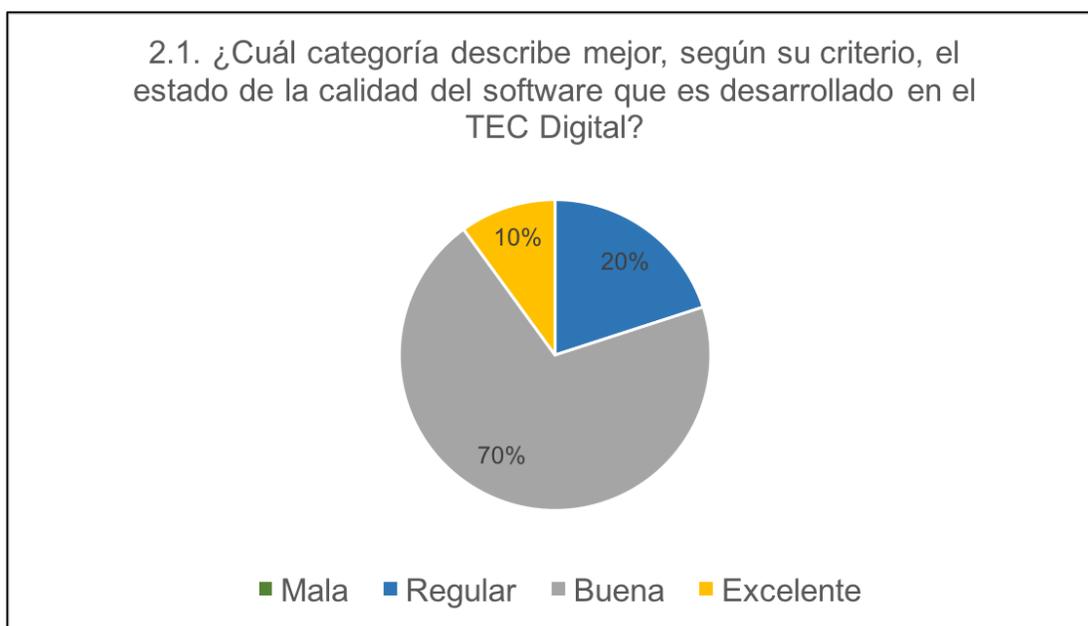
En cuanto a las definiciones de calidad (tomando su significado más general) que se muestran en cada una de las respuestas de los colaboradores (aunque con distintas palabras), se resume el término de calidad que todos manejan como lo siguiente: nivel de cumplimiento de un producto o servicio según las necesidades o los requerimientos de los usuarios finales. En ese sentido, se puede decir que los colaboradores de la Unidad tienen una conceptualización de calidad que se alinea al criterio mostrado, tanto en la ISO 9000 (se detalla en el apartado 2.10.) como por los

autores detallados en dicho apartado y, también, por parte de los profesionales expertos entrevistados.

#### 4.2.2.2. Estado actual de calidad de software del TEC Digital

Como se denota en la Figura 14, las respuestas a la pregunta 2.1. realizada en el cuestionario indican que el 70% de los colaboradores en el TD entrevistados, consideran que la calidad del desarrollo y el producto de software de la Unidad es “buena”.

Figura 14. Estado de calidad del software desarrollado en el TEC Digital



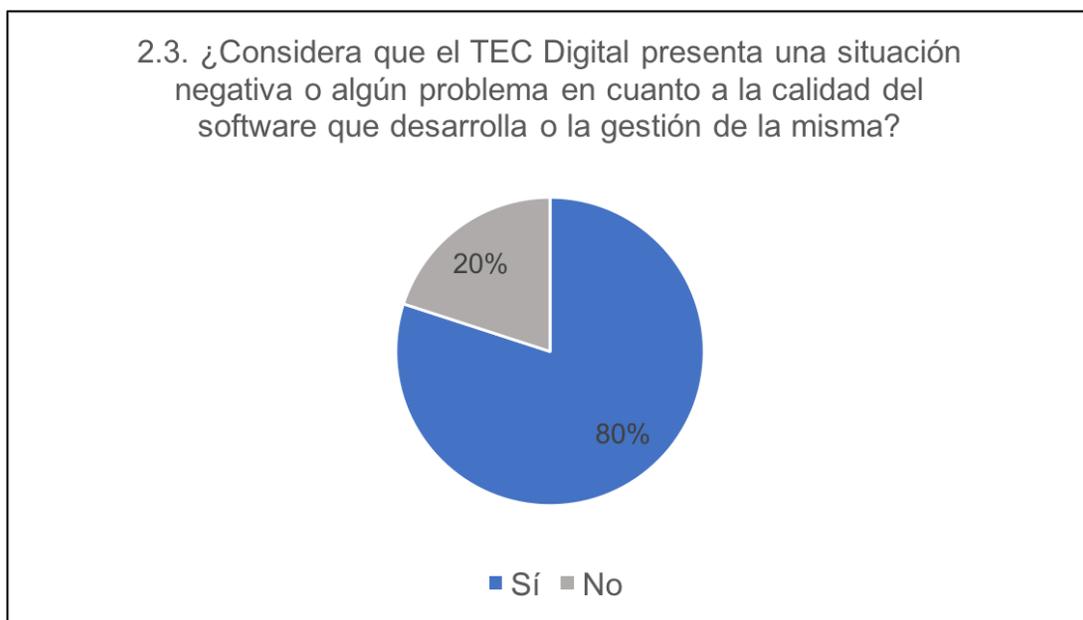
Fuente: Elaboración propia.

En la siguiente pregunta de este cuestionario, los entrevistados emitieron sus apreciaciones para fundamentar la selección de esa categoría, cuatro de las cuales se resaltan a continuación:

- Es “normal” en una organización no tener excelencia en todo; sin embargo, a nivel institucional el TD se distingue de otras instancias por su búsqueda del mejoramiento continuo.

- Se hacen bien las cosas en la Unidad; no obstante, se necesita más comunicación con los usuarios o los clientes finales para conocer su opinión sobre el software desarrollado.
- Se debe buscar un mejoramiento en la detección de errores en el software producido en el TD; pues se tienen muchos reportados a la hora de lanzarse los productos en el ambiente de la comunidad TEC.
- El esfuerzo y los recursos requeridos para ejecutar el control de la calidad del desarrollo y productos del software en el TD son muy elevados; por lo tanto, se debe encontrar la forma de cómo evaluar la calidad para reducir, eso sí, el costo de la calidad que se tiene.

Figura 15. Criterio sobre existencia de situaciones negativas o problemas en la calidad del software del TEC Digital



Fuente: Elaboración propia.

A pesar de que ocho colaboradores considera “buena” la calidad del software de la Unidad, un 10% más, como se puede visualizar en la Figura 15, contestaron en la pregunta 2.3. que esta entidad presenta situaciones negativas o problemas en cuanto a

calidad de software se refiere. Al sintetizar las justificaciones que los colaboradores brindaron del por qué esa respuesta, se detallan:

- La gestión de la calidad es una actividad muy importante dentro de un tipo de organización como el TD; sin embargo, no se le ha otorgado los recursos necesarios para aprovechar ese potencial.
- No se tiene conocimiento generalizado en toda la Unidad sobre los estándares existentes para el desarrollo de software y todas las etapas de su ciclo.
- Se posee una guía de programación para el software que se desarrolla en el TD; sin embargo, se difunde poco en la organización y no presenta un contenido de gran valor para los desarrolladores.
- La ausencia de métricas definidas para la evaluación de la calidad del software ha ocasionado que el resultado sea un asunto de perspectiva; es decir, termina siendo subjetivo totalmente. Esto debido a que al existir transición entre los colaboradores que se encargan de dicha tarea, en ocasiones se tienen conceptos o iniciativas muy diferentes.
- No se tiene control sobre la gestión de cambios en el TD, principalmente, enfocados en el área de vinculante al desarrollo de proyectos de software.
- Hay tareas dentro del proceso de control de la calidad que se podrían automatizar; sin embargo, no se ha realizado algún esfuerzo puntual para llegar a eso.
- No se está ejecutando un control de la calidad en su sentido más estricto; pues realmente, no se están registrando, en muchos casos, los requerimientos del software que se desarrolla; por lo tanto, no se puede definir qué nivel de cumplimiento de necesidades se tiene de los usuarios o clientes finales.
- A pesar de que existen manuales de procedimientos en el TD, no se tienen claras las funciones o las responsabilidades de los colaboradores por área, de

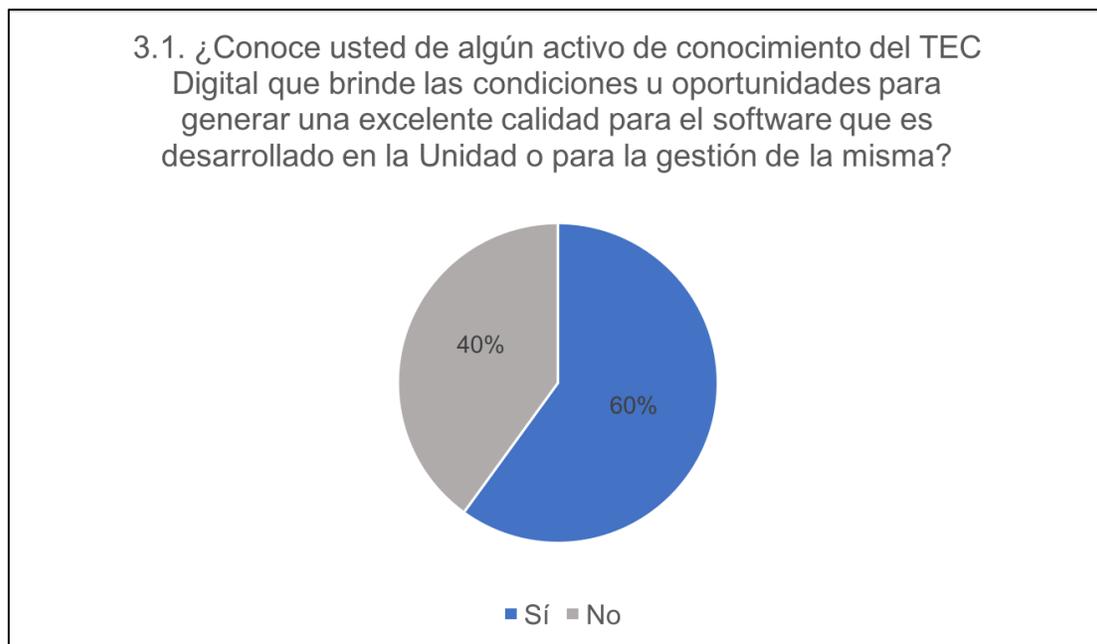
forma que no se identifica el papel que juegan dentro del ciclo de vida de desarrollo del software.

De esta forma, es posible concluir que, en la Unidad se tiene una noción de que el resultado del ciclo de vida de desarrollo de los proyectos de software elabora productos con buena calidad; sin embargo, se tienen situaciones negativas, las cuales aún no se han abordado dentro de la gestión de la calidad.

#### 4.2.2.3. Herramientas para la calidad de software del TEC Digital

En el TD, de acuerdo con las respuestas a la pregunta 3.1. del cuestionario (ver Figura 16), un 40% de los colaboradores entrevistados no tiene conciencia de los activos de conocimiento (estándares, plantillas, guías, entre otros) que asisten en la gestión y el desarrollo de las actividades inmersas en las etapas del ciclo de vida de construcción del software. Este hallazgo fue importante para definir uno de los elementos de la propuesta para solucionar el presente trabajo, principalmente, para el plan de implementación de la misma.

Figura 16. Nivel de conocimiento sobre los activos de documentos del TEC Digital



Fuente: Elaboración propia.

#### **4.2.2.4. Propuestas de mejora para la calidad del software del TEC Digital**

En el cuestionario suministrado se aplicaron dos preguntas para consultar sobre posibles ideas, iniciativas o cambios que se deberían realizar en el TD para promover el mejoramiento de la calidad del software de la Unidad, a continuación, se muestra un recuento de las opciones que se detallaron:

- Elaboración de instrumentos automatizados para ejecutar actividades en cada etapa del ciclo de vida de desarrollo del software.
- Implementación de metodologías ágiles para el desarrollo de software.
- Basar la gestión de la calidad del software en estándares de la ISO.
- Más participación del área de Comunicación Visual de la Unidad en el proceso de elaboración de interfaces gráficas del software.
- Promover la participación del aseguramiento y el control de la calidad del software a lo largo de todo el ciclo de vida de desarrollo de los proyectos de este tipo en el TD.
- Actualización y fortalecimiento de los procesos y los activos de conocimiento del ciclo de vida de construcción del software.
- Culturizar con un enfoque de calidad a todos los colaboradores de la Unidad.

#### **4.2.3. Análisis de documentos**

Se realizó una revisión y un análisis cualitativo de ocho activos documentales con los cuales cuenta el TD y tienen relación directa o indirecta con el ciclo de vida de desarrollo de los proyectos de software de la Unidad. Seguidamente, se describe por medio de la Tabla 2 un resumen de la información general de cada uno de estos documentos.

Tabla 2. Resumen de documentos examinados del TEC Digital

Nombre	Descripción	Fecha de creación	Fecha de actualización
Documento de arquitectura	Plantilla de documento donde son desarrollados la arquitectura y el diseño del software que se construye en el TD.	Desconocida.	Desconocida.
Git – Hoja de trucos	Documento que resume comandos y acciones sobre el repositorio de los paquetes de software de la Unidad.	Desconocida.	Desconocida.
Git – Manual	Manual que describe de forma amplia las acciones de configuración y uso del repositorio de los paquetes de software de dicha instancia.	Desconocida.	Desconocida.
Guía de Programación	Es la guía de programación computacional que debe utilizar el equipo desarrollador del TD.	Desconocida.	Mayo, 2016.
Manual de procedimiento del aseguramiento de la calidad del software	Documento que describe todo el proceso de revisión de calidad de software y los elementos requeridos por el mismo.	Octubre, 2014.	Octubre, 2014.
Metodología de trabajo	Documento que describe la metodología de trabajo para el desarrollo de proyectos de software por área de la Unidad.	Agosto, 2014.	Agosto, 2014.
Plan de Aseguramiento de la Calidad para reducir el número de errores detectados en el entregable final en el desarrollo de paquetes de software libre en el Tec-Digital	Documento que contiene un plan de aseguramiento de la calidad enfocado en la gestión y trazabilidad de los requerimientos de los proyectos de software del TD.	Agosto, 2016.	Agosto, 2016.
Plantilla de Control de Calidad	Plantilla con la que se desarrollan y evalúan los planes de prueba para los proyectos de software de la Unidad.	Agosto, 2013.	Desconocida.

Fuente: Elaboración propia.

Cada uno de estos documentos que se detallan en la Tabla 2, serán especificados y analizados en los siguientes sub apartados.

#### **4.2.3.1. Documento de arquitectura**

Este documento presenta lo que la organización requiere para el desarrollo de software; por lo tanto, contiene elementos de ingeniería de requerimientos, arquitectura y diseño como: especificación de requerimientos, arquitectura de la aplicación, modelo conceptual, diagrama general de casos de uso, especificación de casos de uso, diagrama de clases, diseño de la base de datos, seguridad, modificaciones a la plataforma y lo requerido para la instalación y configuración.

Para este estándar, no se tiene conciencia de la fecha cuando fue creado y tampoco se tiene registro de alguna actualización que haya recibido. Sin embargo, por su contenido, cabe indicar que se constituye en un documento fundamental para el desarrollo de los proyectos de software del TD.

#### **4.2.3.2. Git – Manual y Hoja de Trucos**

Estos documentos presentan un manual y un resumen de acciones realizables sobre el repositorio de paquetes de software que gestiona la Unidad. El repositorio es un versionador de código fuente, el cual se basa en la tecnología de *Gitlab* y utiliza dicha instancia para el registro del progreso en los proyectos de software y también, funciona como herramienta de respaldo.

Tanto el Manual como la Hoja de Trucos contienen instrucciones puntuales y precisas, las cuales ayudan al equipo desarrollador del TD a reconocer las acciones que deben efectuar durante todo el proceso de guardado y versionamiento de los paquetes de software que van realizando durante el ciclo de vida de desarrollo.

No se conoce de forma explícita en los documentos, su fecha de creación o actualización, así tampoco el o los autores de dichos activos de conocimiento de la entidad.

#### **4.2.3.3. Guía de Programación**

La Guía de Programación es un documento que contiene las instrucciones básicas de cómo ejecutar las actividades requeridas, para desarrollar los paquetes de

software basados en la plataforma LRN™, los cuales utilizan como lenguaje de programación el *TCL*.

En este documento no se presenta la fecha de creación; sin embargo, ha recibido una actualización en el 2015 y, dos cambios en el 2016, por tanto, cabe indicar que dicho activo ha presentado al menos un seguimiento anual en los pasados dos años.

#### **4.2.3.4. Manual de procedimiento de aseguramiento de la calidad de software**

Este manual contiene información relevante sobre el proceso que el TD define como *“aseguramiento de la calidad”*. Sin embargo, de acuerdo con las definiciones de lo que se muestra en el apartado 2.11. y según los expertos profesionales entrevistados, más bien, en este documento se detalla la información del proceso de control de la calidad, específicamente, la forma de ejecución de las revisiones y pruebas de calidad. Este manual de procedimiento fue elaborado en el 2014 y desde entonces, no ha recibido ninguna revisión o actualización.

En el manual, se encuentra toda la documentación correspondiente para conocer las especificaciones de los siguientes elementos del proceso: objetivo, alcance, normas, políticas, responsables del proceso, formularios vinculados (se relaciona la Plantilla de Control de Calidad), frecuencia y descripción de las actividades que se realizan durante el proceso.

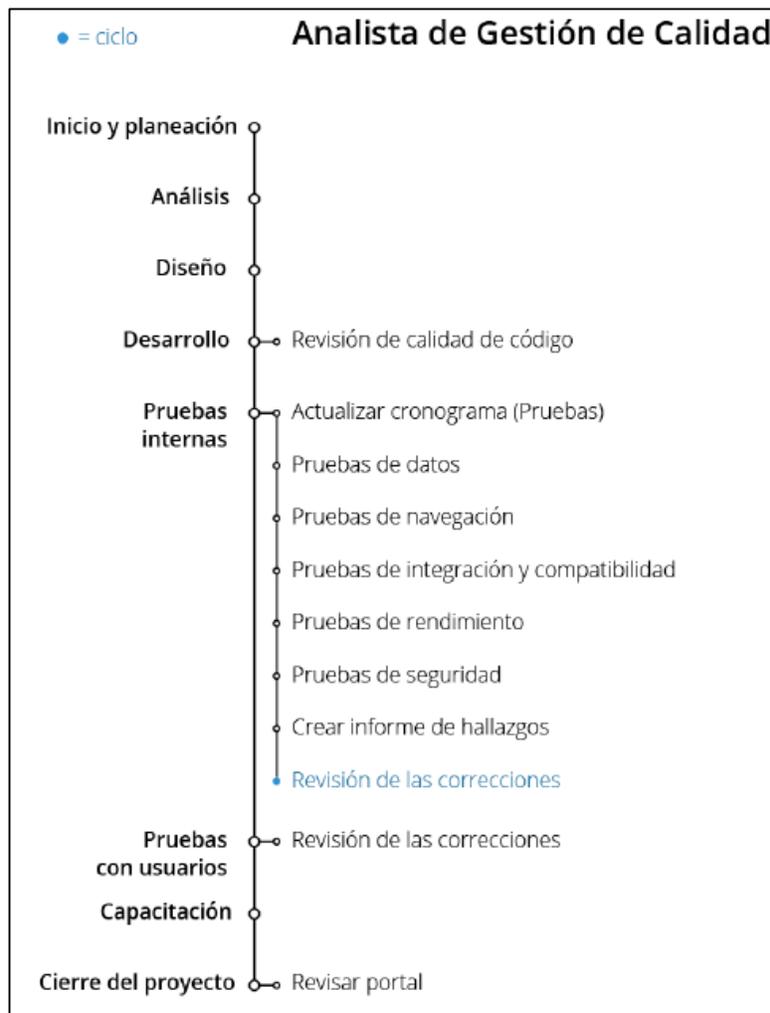
#### **4.2.3.5. Metodología de trabajo**

En la metodología de trabajo se describen las actividades y los roles que deben integrarse a través de los distintos procesos y actividades para el desarrollo de los proyectos de software de la Unidad. En la Figura 17, se visualizan las actividades que realiza el “Analista de Gestión de Calidad”, rol establecido en este documento para representar al colaborador encargado de Gestión de la Calidad del TD, quien ejecuta el proceso de control de la calidad, desde una perspectiva gráfica.

Este documento que contiene la metodología de trabajo no ha sido actualizado desde el momento de su creación en el 2014 y tampoco ha recibido una revisión.

Además, de acuerdo con las respuestas de los colaboradores entrevistados de la Unidad, el 40% de estos no tiene conocimiento de la existencia del mismo.

Figura 17. Actividades del “Analista de Gestión de Calidad” del TEC Digital



Fuente: Elaboración del TEC Digital.

#### 4.2.3.6. Plan de aseguramiento de la calidad

Este documento contiene la primera iteración del Plan de Aseguramiento de la Calidad que elaboró el licenciado Luis Alexander Calvo Valverde, como proyecto final de graduación dentro de su licenciatura.

Se indica primera iteración; pues el mismo autor afirma que dicho trabajo de investigación se perfilaba a crear el ambiente necesario para la ejecución del

aseguramiento de la calidad del software en todas las etapas del ciclo de desarrollo del mismo; sin embargo, sucede que, tal y como se explica dentro del mismo documento, al analizar la situación actual del TD, se determinó que la prioridad era utilizar el estándar IEEE 730-2014 para elaborar el registro y trazabilidad de los requerimientos de los proyectos de software del TD.

Cabe aclarar que, dicha iteración del Plan no se ha implementado en la Unidad. Por lo tanto, la herramienta para el registro y la trazabilidad de los requerimientos (detallada en el apartado 1.4.1.3.) que desarrolló el licenciado Calvo o el plan como tal no han sido ejecutados de forma práctica en la realidad actual de gestión de la calidad. De acuerdo con las razones desarrolladas en el apartado que se especificó en el presente párrafo.

#### **4.2.3.7. Plantilla de Control de Calidad**

Esta plantilla busca documentar el detalle y la evaluación de los planes de prueba de los proyectos de software desarrollados en el TD, los mismos equipos desarrolladores elaboran dichos planes y se gestiona a través de una hoja de cálculo (ver anexo A).

La plantilla mencionada contiene las siguientes secciones (que se presentan a modo de pestañas en el documento): generalidades, plan de pruebas, hallazgos, portal TD, compatibilidad, calidad de los paquetes de software, interfaz/navegación y sección de seguridad.

En el 2014, otra colaboradora que se encargaba del área de Gestión de Calidad en el TD, realizó una actualización de esta Plantilla y le agregó unas escalas de medición a las categorías y preguntas de evaluación que se habían establecido para la autocalificación del Equipo Desarrollador.

En esta plantilla se realiza entonces, el registro digital de los datos y la información resultantes de las revisiones de calidad sobre el software; por esto, la identificación y la interpretación de las variables implícitas que se realizan sobre el documento lleva a pensar que los estados de calidad externa (según el reporte técnico

ISO/IEC/TR 9126-3 elaborado por la ISO y el IEEE [2003b]) de un proyecto de software y sus productos pueden ser definidos mediante la revisión de la misma.

Estas variables que se indican, tal y como se muestra en la propuesta de solución, podrían brindar la capacidad de evaluar la calidad de los proyectos de software a través de implementar el modelo de calidad de la ISO/IEC 9126, respecto a la calidad externa.

#### **4.2.4. Análisis y descripción de problemática actual**

Para la etapa del establecimiento y el desarrollo de la problemática que presenta la calidad de los proyectos de software del TD, de forma específica, dentro del área de gestión de la misma, se ejecutaron dos actividades que permitieron concluir dicha descripción con un profundo detalle y resultados triangulados, según los demás criterios y análisis recabados y elaborados, respectivamente, tanto en el capítulo dos del presente trabajo final desarrollado, como en el análisis de resultados presente en los sub apartados anteriores del actual capítulo.

Estas actividades son: la descripción de la metodología de desarrollo de software de la Unidad, junto al modelado BPMN 2.0 del proceso de control de la calidad del software y la construcción de un diagrama de Ishikawa junto a sus causas profundizadas de forma posterior; los cuales fueron descritos a partir de la observación participativa de cuatro instancias del proceso de control de calidad del software y del desarrollo de los proyectos en general. Estas actividades se documentan seguidamente.

##### **4.2.4.1. Descripción de metodología y proceso**

El TD aparte de ser la unidad encargada del mantenimiento de la plataforma de aprendizaje electrónico de la comunidad TEC, también desarrolla software enfocado en ofrecer soporte a los procesos de enseñanza y aprendizaje del TEC.

Este software se elabora por medio de una metodología de desarrollo que se creó por la misma Unidad en el 2013 (recibió una actualización en el 2014) y en la cual se encuentra el ciclo de vida de desarrollo de un proyecto de este tipo en el TD.

La metodología mencionada se visualiza de forma resumida en la Figura 18. La misma, se elaboró para estandarizar la ejecución de los proyectos y actividades de desarrollo de software, para de esta forma, buscar una mayor calidad de los productos resultantes. De seguido, se describe el modo de aplicación actual de la misma y su proceso de control de la calidad.

El ciclo de construcción de los proyectos de software del TD cuenta con los siguientes roles o equipos involucrados dentro de cada proyecto (se definen también de acuerdo con el organigrama de la Unidad mostrado en la Figura 2 del apartado 1.3.): coordinador de la Unidad, comunicación visual, equipo de desarrollo, gestión de la calidad, capacitación, arquitectura y servidores. Dichos equipos se conforman por 39 colaboradores actualmente.

Además, tal y como se muestra en la Figura 18 posterior, este ciclo de vida de desarrollo de los proyectos de software se compone de siete etapas (resaltadas en color rojo oscuro): uno) Inicio y planeación, dos) Análisis, tres) Diseño, cuatro) Desarrollo, cinco) Pruebas, seis) Capacitación y, por último, siete) Cierre del proyecto.

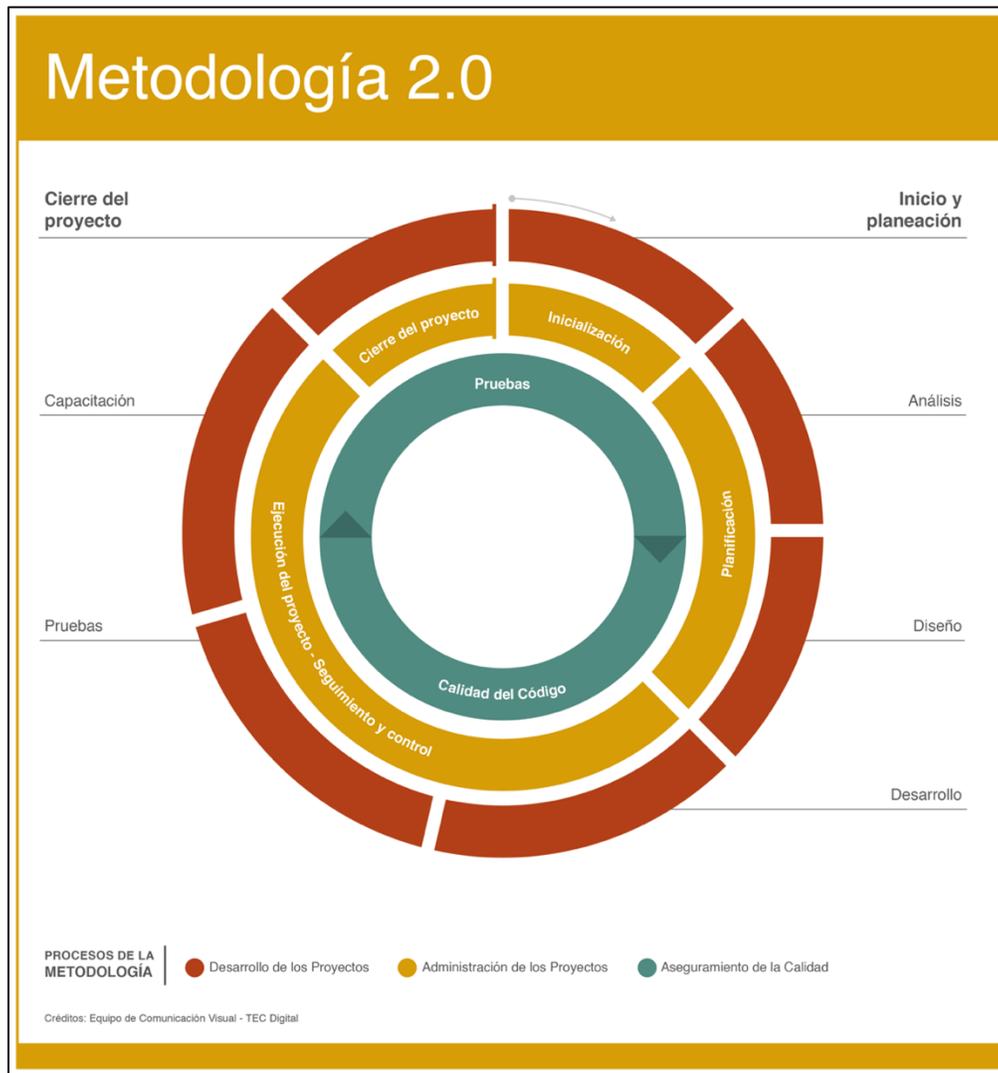
En la etapa uno, se gesta la idea del proyecto por parte del coordinador del TD, el mismo identifica una necesidad de la Comunidad TEC y la expresa con el equipo desarrollador o bien, decide específicamente, quién elabora la resolución respectiva a esa necesidad.

Después, en la etapa dos, el equipo desarrollador designado para el proyecto, en teoría, debe analizar y documentar de forma general el software que se construirá por medio del inicio del documento que contendrá todo lo relacionado al proyecto, conocido como el Documento de Arquitectura o Arquitectura de la Aplicación (explicado en el apartado 4.2.3., ver anexo E). En la práctica esta documentación se elabora hasta días previos a la revisión de la calidad en el proceso que se explica más adelante.

En la etapa tres, según dicha metodología, el equipo desarrollador debe especificar de forma más detallada y concisa, en el Documento de Arquitectura, algunos componentes mínimos de diseño de software y, cuando aplique, las interfaces

gráficas que llevará el software. Por otro lado, en la ejecución de esta etapa (como sucede actualmente), se encuentran tres situaciones según lo observado: los equipos desarrolladores no completan ese documento, no es actualizado o se completa de forma rápida y con imprecisiones de contenido.

Figura 18. Metodología resumida para el desarrollo de los proyectos de software del TEC Digital



Fuente: Elaboración de Comunicación Visual del TEC Digital.

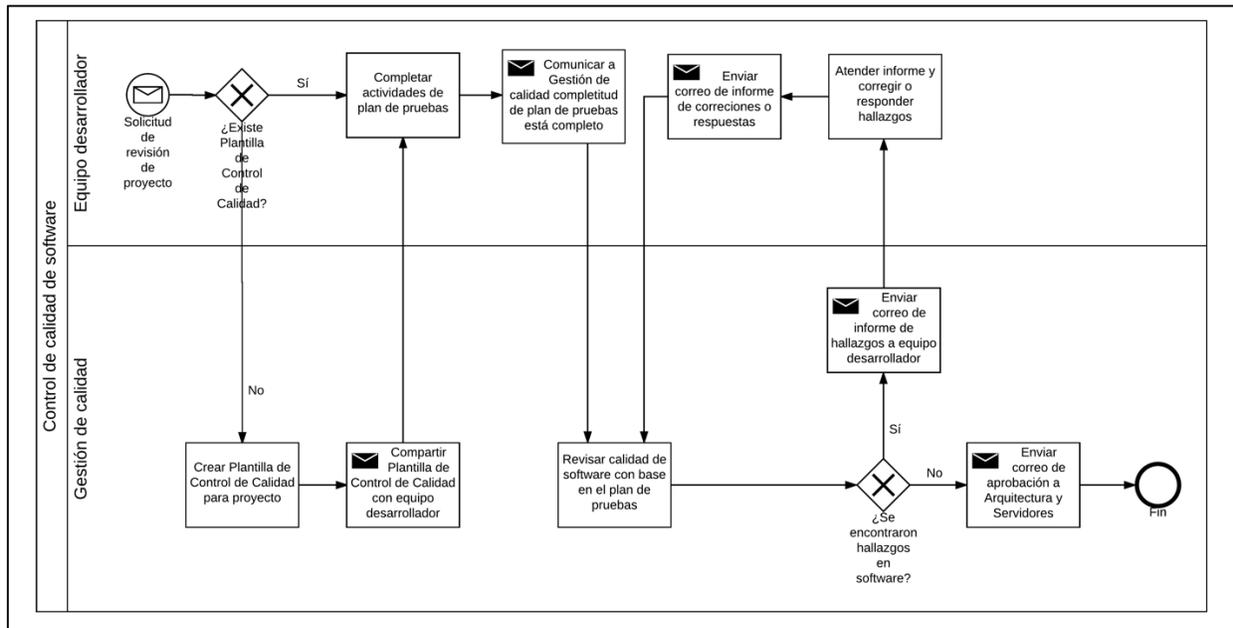
Seguidamente, en la etapa cuatro de desarrollo del software se construye todo lo formulado mediante los diagramas de clases, especificación de casos de uso, diseño

de interfaces gráficas (en caso que aplique) y demás componentes técnicos. De forma explícita, se realiza la programación computacional del producto de software que se requiere. En esta etapa, se presenta un estándar a seguir por los programadores: la Guía de Programación. Este documento no es conocido o utilizado por todo el equipo, según las respuestas recibidas en las entrevistas a los colaboradores del apartado 4.2.2.

Luego, se ejecuta la etapa cinco donde se inician las pruebas, una vez desarrollado el paquete de software e instalado debidamente, en el ambiente correspondiente. Estas actividades de pruebas las ejecuta el colaborador de Gestión de la Calidad. Como se mencionó en el apartado 1.5., el TD cuenta con tres ambientes diferentes para el desarrollo y la liberación de los proyectos de software, los cuales resultan en una arquitectura de servidores, dichos ambientes son: Desarrollo, Pruebas y Producción. En teoría (según el manual de procedimiento), estas actividades se deberían iniciar en Desarrollo; sin embargo, en la actualidad, se empieza en el ambiente Pruebas y luego se migra al ambiente de Producción.

Este proceso iterativo de pruebas se conoce en la entidad como aseguramiento de la calidad del software, así está dispuesto en el Manual de Procedimiento detallado en el apartado 4.2.3.4. A pesar de dicha definición, con base a la metodología de Avella y Gómez (2011) y el criterio mostrado por los profesionales expertos en sus entrevistas respectivas, las actividades que se realizan dentro de ese proceso corresponden a un control de la calidad de software. Este proceso se visualiza en la Figura 19. Más adelante, en este mismo apartado, el mismo se explicará de forma extendida.

Figura 19. Modelación del proceso de control de la calidad de software del TEC Digital



Fuente: Elaboración propia.

Posteriormente, en la penúltima etapa seis, se debe realizar un plan de capacitación en conjunto con el equipo de dicha área, antes de lanzarse el software al ambiente de Producción, según lo estipulado en la metodología de trabajo. Esto significa que resulta necesario el involucramiento por parte de los equipos desarrolladores hacia esa área del TD; sin embargo, de acuerdo con lo observado, en ningún caso sucede así; sino que se libera primero el software y luego se realiza el manual de capacitación.

Una vez que el software se encuentra en el ambiente de Producción, se concluye el proyecto, de modo que se realiza la etapa siete: el cierre del proyecto. Para esta etapa no se realiza ningún tipo de documento o entregable específico. Además, aunque con esa etapa se finaliza el proyecto, luego vienen otras que no son descritas en la Figura 18, pero sí se mencionan en la ISO/IEC 12207 IEEE Std 12207-2008 elaborada por la ISO y el IEEE (2011), especificadas en el apartado 2.6. como las de soporte y reutilización del software.

En dichas etapas posteriores a la liberación del software, se analizan y se implementan cambios que surgen principalmente, debido a correcciones del funcionamiento del software u oportunidades de mejora, las cuales son canalizadas mediante sugerencias de los usuarios finales del software que surgen por lo general, en actividades de capacitación, o bien, por los mismos miembros del equipo desarrollador. Estos cambios son vistos como nuevos proyectos o mejoras, según el alcance de trabajo requerido y el criterio del coordinador de la Unidad.

Por otra parte, al retomar lo que se mencionó anteriormente, los equipos que participan en un proyecto de desarrollo de software interactúan en el ciclo de vida de estos proyectos, según la etapa en la cual se encuentran los mismos.

De acuerdo con la metodología de la Unidad, en el caso del equipo de Gestión de la Calidad (en adelante GC), su participación inicia desde la etapa que la Unidad define propiamente, como “Desarrollo” y sobre el ambiente que lleva el mismo nombre. Esto se define en la teoría; pues en la práctica se inicia en el ambiente de Pruebas.

Una característica importante por destacar es que, la etapa de “Desarrollo” se presenta después del “Inicio y Planeación”, después del “Análisis” (engloba la especificación de requerimientos del software) y más importante aún, después del “Diseño”. En ninguno de estos procesos o los anteriores los equipos de desarrollo interactúan con GC o tienen algún tipo de revisión o control de la calidad.

Como se mencionó previamente, la interacción entre el equipo desarrollador y GC inicia en la etapa de Desarrollo, por medio de la solicitud de creación de la Plantilla de Control de Calidad para el proyecto. Esta solicitud se realiza a través del envío de un correo electrónico a GC, el colaborador de GC crea el documento en la herramienta *Google Drive* y lo comparte al equipo desarrollador respectivo, por correo electrónico y, al mismo tiempo, otorga los permisos correspondientes sobre el archivo. Sin embargo, la descripción del método para realizar esa solicitud no se encuentra dentro del manual de este procedimiento.

Esta Plantilla de Control de Calidad, es la herramienta con la cual se ejecuta el proceso de control de la calidad en los proyectos de software del TD, el mismo que en términos prácticos, se basa en el plan de pruebas que detalla el equipo desarrollador sobre esta plantilla; es decir, los desarrolladores ingresan las pruebas mínimas que se deben ejecutar sobre el producto final de software. Acción que, de acuerdo con el entrevistado dos, no es conveniente para este tipo de revisiones, ya que este plan debería ser elaborado junto a GC. Las pruebas que contiene el plan desarrollado por los equipos desarrolladores permiten la verificación de aspectos funcionales mayoritariamente.

Por otro lado, un requisito previo, para que GC realice las actividades del control de la calidad es que los paquetes de software de cada proyecto estén actualizados. Estos paquetes deben ser instalados en el servidor donde serán revisados y probados.

Ahora bien, en el ambiente de Pruebas, donde se realizan las actividades de control de calidad, se inician las revisiones orientadas principalmente, hacia un aspecto funcional; es decir, se corrobora que el software funcione de acuerdo con lo descrito en el plan de pruebas de la Plantilla.

Dentro de este plan de pruebas que se encuentra en la Plantilla de Control de la Calidad, actualmente, se están realizando entonces solo revisiones de tipo funcional; por lo tanto, otros aspectos o características de calidad que permiten determinar el nivel de cumplimiento del software, como: seguridad, rendimiento, usabilidad, facilidad, portabilidad, mantenibilidad, entre otros puntos, quedan por fuera. Estas métricas, según la ISO/IEC 9126 deben ser evaluadas para definir con una perspectiva formal y objetiva la calidad de un software.

Es importante aclarar que, en esta Plantilla de Control de Calidad si se encuentran otras pestañas adicionales al plan de pruebas que contienen preguntas, las cuales buscan, medir estos criterios de calidad a partir de cuestionamientos o parámetros que se desarrollaron en la Unidad. Sin embargo, estas verificaciones no siguen un orden lógico de evaluación y 2 se encuentran obsoletas, porque ya no se utilizan sus componentes de programación. Además, se debe revisar cada una de

estas pestañas por cada una de las actividades que se describen dentro del plan de pruebas; por lo tanto, la revisión de calidad se vuelve y aquí se retoman las palabras de un colaborador entrevistado del TD, un “cuello de botella”.

Adicionalmente, en el 2015, para estas verificaciones de la Plantilla de Control de Calidad se agregó una serie de escalas de evaluación (de uno a tres), un ejemplo de estas se muestra en la Figura 20. Sin embargo, estas no muestran algún tipo de medición objetiva y clara; por lo tanto, su resultado y calificación se establecen según la “perspectiva” del colaborador encargado de GC que realice la revisión respectiva.

Figura 20. Sección de seguridad y rendimiento de la Plantilla de Control de Calidad

tecDigital		Aspectos y seguridad del proyecto:	
<b>Seguridad</b>			
Criterios	Descripción	Evaluación (1-3)	
Datos sensibles por URL	¿Se envían los datos por POST o algún método		
SQL injection	¿Los campos de texto son seguros?		
Requiere o no inicio de sesión	Probar entrar sin iniciar sesión, algunas aplicaciones requieren obligatoriamente esto.		
<b>Rendimiento</b>			
Criterios	Descripción	Evaluación (1-5)	Proceso interno
Tiempos de respuesta	Uso de Robots    Coordinar con los coordinadores un tiempo aceptable de respuesta. (según el proceso interno)		
Pruebas de estrés	Se debe doblar el número de usuarios hasta que produzca un error.		
Pruebas de carga	Carga de datos		

Fuente: Elaboración del TEC Digital.

Otra característica importante de la Plantilla de Control de Calidad y pieza clave del proceso de control de calidad es la pestaña de hallazgos (ver Figura 21); esta representa uno de los puntos principales de la revisión funcional de la calidad del software que desarrolla el TD.



“Estado” en la sección de hallazgos; dichos estados son: reportado, pendiente, resuelto y seguimiento.

El primer estado de un hallazgo se genera cuando el mismo es creado. El segundo se ingresa en caso de que se realice una revisión de corrección y el hallazgo aún esté presente; es decir, no ha sido resuelto. El tercero se muestra cuando el hallazgo se resuelve debidamente, por parte del equipo desarrollador. Y el cuarto estado de seguimiento se presenta cuando un hallazgo exponga un comportamiento intermitente (en ciertos casos sucede de prueba y en otros no).

Una vez que GC concluye la revisión de calidad –esta actividad del proceso de control de la calidad del software finaliza cuando se chequea de forma completa el plan de pruebas; es decir, todas las actividades descritas en el mismo–, el equipo desarrollador efectúa una consulta sobre los resultados de esta revisión y documenta anotaciones de acuerdo con lo que corresponde en cada caso.

Existen tres opciones para estas documentaciones de respuesta ante el hallazgo: el equipo desarrollador reconoce el hallazgo y realiza la acción correspondiente para resolverlo; el equipo desarrollador no comprende la descripción del hallazgo, por lo tanto, solicita más información y otra opción es que, el equipo desarrollador refuta el hallazgo y GC acepta que no era un hallazgo pertinente.

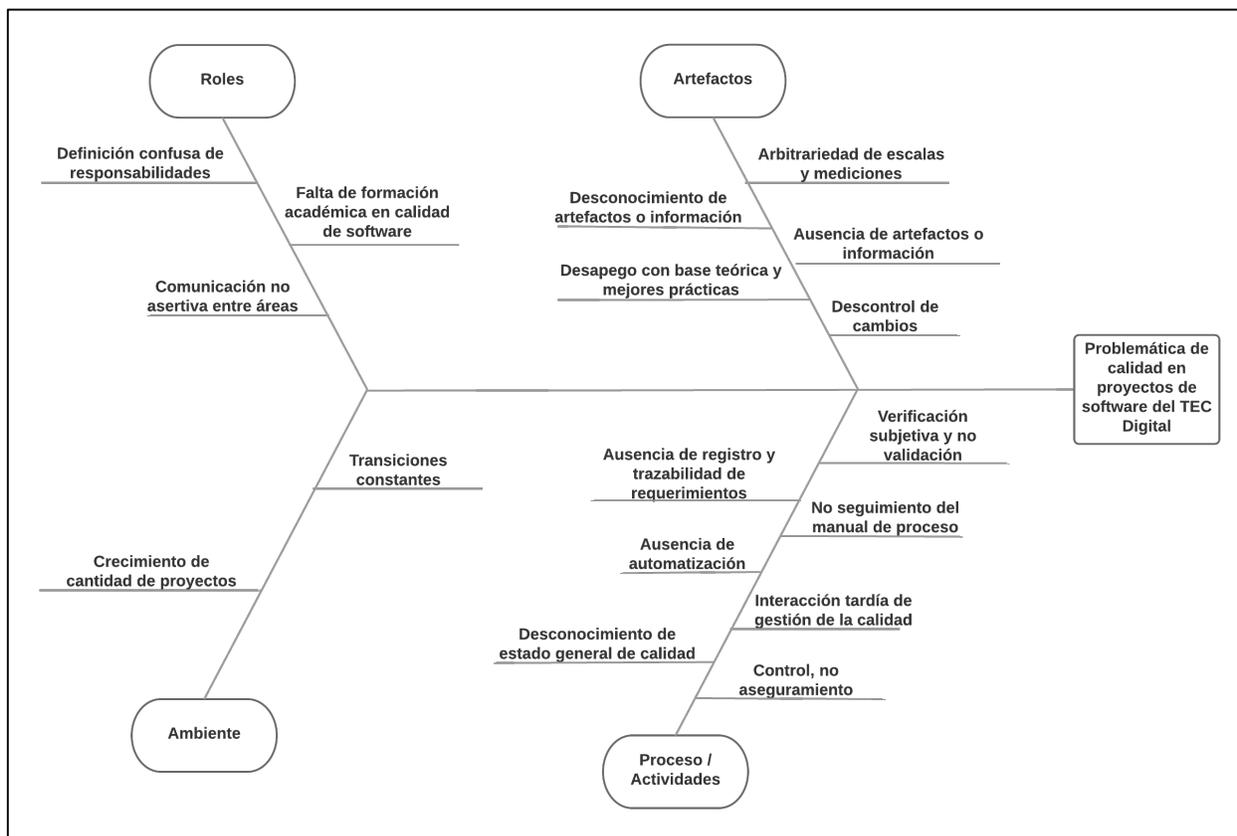
Este proceso de interacción entre los equipos desarrolladores y GC, como se mencionó anteriormente, es iterativo. Finaliza cuando la totalidad de hallazgos tipo error ha sido resuelta. Pueden existir eso sí, hallazgos de tipo mejora y duda sin resolver.

La aprobación de la migración del software revisado queda a cargo de GC; por lo tanto, se envía un correo electrónico al área de arquitectura y servidores del TD para aprobar dicha acción. Y, de esta forma, finaliza una instancia del proceso de control de la calidad de software.

#### 4.2.4.2. Diagrama de Ishikawa

A partir de los resultados obtenidos como aplicación de los métodos, las técnicas y los instrumentos investigativos utilizados para identificar los componentes de la problemática actual de la calidad de software, se formula como herramienta de sintetización de dichos resultados, un diagrama de Ishikawa, conocido de forma común como diagrama de causa y efecto, el cual, se visualiza en la Figura 22.

Figura 22. Diagrama de Ishikawa de la problemática de calidad en los proyectos de software en el TEC Digital



Fuente: Elaboración propia.

Para la elaboración de este diagrama de Ishikawa, se dividió mediante una categorización las causas de la problemática mencionada, la cual fue construida a partir del criterio emitido por el entrevistado dos. Este criterio enfatiza construir un sistema de gestión de la calidad basado en la “tríada de calidad”, así lo detalla el entrevistado dos en el apartado 4.1.2. Esta tríada está compuesta por: roles,

actividades (o procesos) y artefactos necesarios para realizarlas o que surgen a partir de su ejecución.

Aparte de esa triada como elementos de categorización que agrupan las causas, además se define, una cuarta categoría: el ambiente. Esto, pues según Turner (2008), citado por Stoshikj, Kryvinska y Strauss (2013) en el apartado 2.3. del presente trabajo final de graduación, el papel que representa la entidad dentro del desarrollo de un proyecto resulta fundamental. Además, durante la recopilación de causas basadas en los análisis de resultados, surgieron las que se muestran en la Figura 22 para esta categoría de ambiente, las mismas comparten características provenientes de esta arista de análisis, por lo cual se consideró relevante sumarla a lo recomendado por el entrevistado dos.

#### **4.2.4.3. Estudio analítico de las causas de la problemática de calidad en los proyectos de software del TEC Digital**

A raíz de la elaboración del diagrama de Ishikawa se obtuvieron las causas que generan la problemática actual de calidad en los proyectos de software del TD, estas causas se describen y analizan de una forma más extendida de continuo, divididas por categorías de agrupación.

##### **4.2.4.3.1. Roles**

Tabla 3. Descripción y análisis de las causas de la categoría “Roles”

<b>Causa</b>	<b>Descripción y análisis</b>
Definición confusa de responsabilidades	<p>Según respuestas recibidas de los colaboradores entrevistados del TD en el apartado 4.2.2., en la organización no se tiene una definición clara de las responsabilidades específicas dentro de la metodología de desarrollo de los proyectos de Software de la Unidad.</p> <p>A pesar de que existe dicha metodología, tres de los colaboradores de la Unidad entrevistados afirman que, a nivel de actividades y tareas (aún más puntuales), se desconoce a quién corresponde cada una de estas. Un colaborador brinda el siguiente ejemplo: a varios desarrolladores (los nuevos</p>

Causa	Descripción y análisis
	<p>principalmente) les surgen dudas una vez que gestión de la calidad aprueba la migración de un software al ambiente de Producción, tales como: ¿Qué corresponde? ¿Quién migra el mismo? ¿Qué pasos debe realizar el equipo desarrollador?</p> <p>Esta situación, según el criterio aportado por el entrevistado dos en el apartado 4.1.2., genera una problemática; pues la integración de los roles, así como las responsabilidades y las actividades asignadas para cada uno de estos, es lo que define correctamente, un sistema para la gestión de la calidad de software (tanto para el aseguramiento como control de calidad de software).</p>
<p>Falta de formación académica en calidad de software</p>	<p>De acuerdo con el criterio, mostrado en el apartado 4.2.2., de dos de los colaboradores del TD entrevistados, esta Unidad, cuenta con un recurso humano que no tiene la formación académica suficiente para el desarrollo de las responsabilidades y las tareas de dicha área; aconsejan que se debería tener algún profesional con experiencia en el tema de calidad de software.</p> <p>Por otra parte, de acuerdo con el entrevistado dos en el apartado 4.1.2., actualmente, es comprensible no contar con profesionales (a nivel costarricense) que presenten una formación académica en el tema de calidad de software, según su criterio emitido; pues a nivel curricular, las instituciones de educación superior, como el TEC, apenas están en proceso de iniciar la introducción de esta temática dentro de sus programas.</p>
<p>Comunicación no asertiva entre áreas</p>	<p>Cuatro de los colaboradores entrevistados mencionan que el desarrollo de la comunicación en el TD no se ejecuta de forma asertiva, tendencia que se afecta de forma muy relevante en los flujos de información durante el desarrollo de un proyecto de software.</p> <p>Esto es apoyado también por el entrevistado uno en el apartado 4.1.1.; pues menciona que encontró esa situación en el periodo en el cual desarrolló el trabajo de investigación descrito en el apartado 1.4.1.3.</p> <p>Esta provoca, de acuerdo con lo que presenta la ISO (2015)</p>

Causa	Descripción y análisis
	<p>en el apartado 2.9. sobre la norma ISO 9000, una desventaja en la Unidad; pues la “gestión de las relaciones” (así se le llama en esa norma) es uno de los principios de la calidad, que una organización debe tratar de reforzar a nivel interno de la misma, siempre y cuando busque el mejoramiento en sus procesos y productos.</p>

Fuente: Elaboración propia.

#### 4.2.4.3.2. Proceso / Actividades

Tabla 4. Descripción y análisis de las causas de la categoría “Proceso / Actividades”

Causa	Descripción y análisis
<p>Verificación subjetiva y no validación</p>	<p>El entrevistado uno afirma en el apartado 4.1.1. que, durante la realización de su trabajo de investigación en el TD, identificó que el proceso de control de la calidad del software no tiene escalas de medición concretas y objetivas. Esto genera el riesgo, en el caso que dicho proceso lo realicen distintos colaboradores o haciendo la suposición de cambiar el encargado de gestión de la calidad, la medición de la “calidad interna” del software se podría variar y no se mantendría consistencia ni en el proceso ni en sus productos.</p> <p>Así también, los colaboradores entrevistados de la Unidad mencionan en el apartado 4.2.2. que la calidad de software muchas veces en esta organización, se convierte en un resultado de “perspectiva” del personal de Gestión de la Calidad.</p> <p>Lo anterior, muestra una diferencia con lo propuesto por Wagner et al. (2011) donde se especifica que la calidad del software debe ser determinada en función de las metas de los usuarios finales, lo cual, en la actualidad no sucede así en esta organización.</p> <p>De igual forma, mediante el conocimiento general de las actividades y los esfuerzos del TD en la calidad del software, el profesional experto de la segunda entrevista comenta que en dicha instancia se realiza una verificación de la calidad, más no validación; pues, se verifica que cumpla con ciertas características de calidad, sin embargo, no se valida que esas</p>

Causa	Descripción y análisis
	<p>características satisfagan las necesidades verdaderas de los clientes.</p>
<p>Ausencia de registro y trazabilidad de requerimientos</p>	<p>Un hallazgo del entrevistado uno, descrito en el apartado 4.1.1. y encontrado durante su investigación en el TD fue la debilidad encontrada en la gestión y trazabilidad de los requerimientos del software, los cuales, según la observación del proceso de control de la calidad (ver apartado 4.2.4.1.), en ocasiones no son documentados y mucho menos analizados a través del ciclo de vida de desarrollo de la Unidad.</p> <p>Para el entrevistado dos es una problemática; pues los requerimientos representan las necesidades de los clientes y al no contar con una correcta gestión de estos se complica la verificación de la calidad, según la definición formal (consultar el apartado 2.10.) de la misma.</p> <p>También, se crea una problemática respecto a lo que dice Arias (2007) en el apartado 2.7., donde indica que un requerimiento de software debe ser especificado, verificable, conciso, completo, consistente y no ambiguo. Estas características, al no ser registradas, no se presentan en los requerimientos del software de la mayoría de proyectos del TD actualmente, según el entrevistado uno.</p>
<p>No seguimiento del manual del proceso</p>	<p>Esta causa se identificó al realizar el análisis de documentos, específicamente, por medio del apartado 4.2.3.4. donde se identifican con claridad las actividades, según el manual de procedimiento para el proceso de control de la calidad. Sin embargo, durante la observación participativa del mismo, se identificó que la ejecución de este proceso no sigue lo estipulado en dicho manual, lo cual contradice lo propuesto por el entrevistado dos, quien afirma que para lograr un aseguramiento de la calidad, se debe primero sistematizar y ejecutar una ruta planificada de las acciones que se realizarán para brindar propiciar dicha calidad de software. En el caso del TD sí se planificó, más no se ejecutó de esa forma.</p> <p>Además, se encuentra una discrepancia entre lo que efectúa</p>

Causa	Descripción y análisis
	<p>la Unidad y lo que se menciona sobre Pressman (2010) en el apartado 2.11.; pues el autor señala con el objetivo de proveer dicho aseguramiento se debe tener un proceso establecido para el mismo, actualmente, se ejecuta un proceso, el cual es muy diferente al documento, de acuerdo con la revisión del apartado 4.2.3.4 y lo documentado sobre la descripción del proceso en el apartado 4.2.4.1.</p>
<p>Ausencia de automatización</p>	<p>Según los colaboradores de la Unidad, existen tareas en el control de la calidad del software que podrían ser automatizadas; sin embargo, no se han realizado esfuerzos para lograr esa meta. Esto provoca una situación, descrita por un colaborador entrevistado, de “cuello de botella” dentro del ciclo de vida de desarrollo de los proyectos de software, lo cual, según el criterio de los dos entrevistados que se muestra en el apartado 4.1.3., es el paso a seguir, una vez que se logre instaurar un sistema o proceso de forma repetida y consistente.</p>
<p>Interacción tardía de gestión de la calidad</p>	<p>Para siete de los colaboradores entrevistados de la Unidad, es necesario que el aseguramiento y el control de la calidad del software del TD sean realizados desde el inicio del proyecto, situación que según lo recabado en la observación del proceso de control de calidad, no se ejecuta de esa forma; pues de acuerdo con lo mencionado por el entrevistado uno, en su trabajo de investigación en la Unidad halló que dicha organización realiza lo que él caracteriza como una revisión de calidad post-desarrollo, la cual crea un riesgo asociado al costo de calidad (término mencionado igualmente, por el entrevistado dos).</p> <p>Esta situación es contraria a la propuesta por Avella y Gómez (2011) en el apartado 2.11. donde se muestra como buena práctica iniciar con actividades de procesos de calidad desde la etapa de análisis del ciclo de vida de desarrollo del software. De igual forma, se contrapone con lo dispuesto por Pressman (2011) en el mismo apartado, donde señala que una realización de este tipo de actividades desde etapas iniciales en proyectos de</p>

<b>Causa</b>	<b>Descripción y análisis</b>
	software, reduce los costos finales y post-implementación (cambios o mejoras) de una forma cuantiosa, o como el autor también lo señala: el costo de la calidad.
Desconocimiento de estado general de calidad	<p>Esta causa se identificó al realizar el análisis cuantitativo desarrollado en el apartado 4.2.1., donde se resume de forma concisa, datos e información estadística de cuatro instancias del proceso de control de la calidad del software sobre el proyecto Evaluaciones. Este resultado refleja que actualmente, el TD desconoce de esa información, la cual resulta vital para la toma de decisiones basada en actividades, como la evaluación mediante métricas de calidad, las cuales se pueden encontrar en la ISO/IEC 9126. El modelo fundamentado en esa norma, es un buen abordaje, según el entrevistado dos (apartado 4.1.2.) hacia la evaluación objetiva y concreta de la calidad del software.</p>
Control, no aseguramiento	<p>De acuerdo con el resultado obtenido mediante la observación participativa en el proceso de calidad del TD (descrita en el apartado 4.2.4.1.) y la revisión documental (desarrollada en el apartado 4.2.3.), se encuentra un contraste en referencia a lo que múltiples autores detallados en el apartado 2.11. definen como aseguramiento de la calidad del software.</p> <p>Asimismo, entre lo establecido como aseguramiento de la calidad del software, tanto por el entrevistado uno como por el dos, respecto a lo tratado por la Unidad como “aseguramiento”, se encuentra una discrepancia total. Esto se denota de forma más clara en el Manual de procedimiento (ver apartado 4.2.3.4.) contrastado con las definiciones en el apartado 2.11.</p>

Fuente: Elaboración propia.

### 4.2.4.3.3. Artefactos

Tabla 5. Descripción y análisis de las causas de la categoría “Artefactos”

Causa	Descripción y análisis
Arbitrariedad de escalas y mediciones	<p>Este hallazgo se identifica mediante la descripción del proceso de control de la calidad, mostrado en el apartado 4.2.4.1., donde se reconocen los atributos de los hallazgos como los tipos y las prioridades; sin embargo, la definición y priorización de estos queda a criterio del colaborador encargado de Gestión de la Calidad.</p> <p>Dicha situación, de acuerdo con Dutil, Rose, Suryan y Thimot (2010), autores citados en el apartado 2.12., se puede abordar mediante la utilización de un modelo de calidad para la construcción del software.</p>
Desconocimiento de artefactos o información	<p>En las respuestas mostradas por los colaboradores del TD (ver Figura 16), se identifica que al menos un 40% de esta organización no tiene idea de la existencia de artefactos o información que propicie el aseguramiento y control de la calidad del software de la Unidad.</p> <p>Esta afirmación se puede asociar a una incorrecta “gestión de las relaciones” o a la ausencia de un “compromiso de los colaboradores”, características que son establecidas por la ISO (2015) en su norma 9000 como principios fundamentales en la búsqueda hacia el mejoramiento de la calidad de una organización y sus productos o servicios.</p>
Ausencia de artefactos o información	<p>Mediante la revisión documental mostrada en el apartado 4.2.3., se identifica que existe sólo un artefacto (Manual de procedimiento) que propicia el control de la calidad; lo cual genera una diferencia respecto a la metodología de Avella y Gómez (2011) en el apartado 2.11., o también respecto a lo resumido por Pressman (2010), donde se especifica una serie de requisitos a seguir para lograr una correcta ejecución del aseguramiento de la calidad del software.</p>
Desapego con	

<b>Causa</b>	<b>Descripción y análisis</b>
base teórica y mejores prácticas	<p>A través de la revisión documental descrita en el apartado 4.2.3., se reconoce la ausencia de registro sobre la base teórica, estándares internacionales, plantillas, mejores prácticas o demás cuerpos de conocimiento de las cuales se sustentó la realización de los artefactos de conocimiento con los que cuenta el TD actualmente.</p> <p>Este hallazgo indica un posible desapego entre los artefactos actuales y lo que indica Pressman (2010) en el apartado 2.11., donde el autor señala que se debe contar con métodos y herramientas establecidos, de tal forma que se ejecuten actividades y prácticas eficaces de aseguramiento de calidad.</p> <p>El entrevistado uno por su parte, comentó que en su trabajo de investigación realizado en la Unidad identificó este hallazgo; de tal forma que, por esa razón, elaboró uno de sus productos finales con base en la plantilla IEEE 730-2014.</p>
Descontrol de cambios	<p>El profesional experto entrevistado número uno afirma que, dentro de su trabajo de investigación en el TD, identificó que no se realiza un control de cambios sobre los documentos y las herramientas utilizados en la metodología de desarrollo de software de esta organización (consultar el apartado 4.1.1.).</p> <p>Dicho hallazgo se confirma con lo encontrado mediante el análisis documental resumido en la Tabla 2 del apartado 4.2.3.; pues se reconoció que solo los artefactos actualizados en el 2016, contienen una tabla de control y seguimiento de los cambios que han recibido.</p> <p>Lo mencionado de manera anterior, genera una problemática respecto a lo que indica el entrevistado uno, al comentar que una clave para el aseguramiento de la calidad del software es la repetición y constancia en la ejecución de procesos y actividades con la utilización de estándares de artefactos o herramientas (dicha afirmación se puede revisar en el apartado 4.1.1.), los cuales actualmente, no son controlados por la Unidad desde la óptica de gestión de cambios.</p>

Fuente: Elaboración propia.

#### 4.2.4.3.4. Ambiente

Tabla 6. Descripción y análisis de las causas de la categoría “Ambiente”

Causa	Descripción y análisis
Transiciones constantes	Si se toma como base las respuestas de los colaboradores de la Unidad en el cuestionario, cabe indicar que se ha reconocido a nivel interno de esta entidad, específicamente, en el área de Gestión de la Calidad, que hay rotación y esto ha generado que no se mantenga un estándar de definiciones, debido al establecimiento tácito de las métricas del proceso de control de la calidad y revisiones asociadas a la calidad del software en el TD.
Crecimiento de cantidad de proyectos	A través de los datos recabados y mostrados en la Tabla 1, cabe indicar que la cantidad de proyectos en el ambiente del TD ha venido creciendo de forma muy rápida, con casi un 78% de porcentaje de aumento anual, lo cual influye de forma directa en la cantidad de instancias de los procesos de calidad que serán requeridos en la Unidad conforme avance el tiempo. Esto, a su vez, implica en un aumento de carga de trabajo para el colaborador encargado de Gestión de la Calidad de dicha organización.

Fuente: Elaboración propia.

#### 4.2.5. Síntesis de problemática de calidad del TEC Digital

En el TD se encuentra una problemática en la calidad de los proyectos desarrollados de software en esta Unidad, la cual surge desde la contradicción que la mayoría de los colaboradores reconocen el concepto de calidad de software como satisfacción de los requerimientos del usuario final, en este caso, los requerimientos del software producido; sin embargo, a los usuarios finales del software desarrollado no se les está tomando en cuenta de forma constante dentro de la metodología y actividades de desarrollo de estos proyectos, principalmente, en las etapas de liberación y despliegue de los mismos.

Además, el único proceso que se realiza en la actualidad, respecto a la calidad de software, es el control de la misma y de una forma muy tardía dentro del ciclo de vida de desarrollo de los proyectos de software de la TD, lo cual eleva el riesgo sobre el costo de la calidad. A esto se agrega, que no se cuenta con una ruta sistematizada de actividades que velen por el cumplimiento de requisitos de calidad, lo cual representaría un aseguramiento de la calidad en estos proyectos.

Dentro del proceso de control de la calidad del software, no se tiene definido de forma clara, las responsabilidades y las actividades correspondientes para cada rol o colaborador del Unidad. Por lo tanto, la gestión comunicativa entre las áreas y roles que ejecutan el mismo, no se realiza de forma asertiva.

De igual manera, a pesar de que el proceso cuenta con su documentación respectiva (la cual no se utiliza para la ejecución del mismo), no posee métricas de evaluación, esto ocasiona que exista subjetividad en dicha evaluación de la calidad del software en los proyectos de ese tipo de la Unidad. En otras palabras, el efecto que tiene esta situación es que se relaciona el término de calidad de software con un asunto de “perspectiva”.

Para concluir la síntesis, la falta de estandarización y ausencia de métricas objetivas de calidad de software, generan un proceso “cuello de botella”, el cual no provee una noción clara del estado de calidad de los proyectos de software del TD; por lo tanto, no cumple sus objetivos de ejecución. Además, de los resultados obtenidos, no se provee estadísticas, indicadores o información pertinentes para la toma de decisiones a la organización.

#### **4.3. Elaboración de diseño del sistema para el aseguramiento y el control de la calidad en los proyectos de software del TEC Digital**

Para elaborar el diseño del sistema, se utilizó como base de referencia de sus elementos de eje principal, lo propuesto por el entrevistado dos en el apartado 4.1.2., donde se indica la “tríada de calidad”, donde se hace una representación de la interacción entre los roles que ejecutan las actividades y, a su vez, producen o utilizan artefactos.

La composición de cada uno de estos elementos se definió según lo expuesto a continuación:

- Roles: se definen con base en las áreas que contiene el TD y que muestran relación directa o indirecta en el ciclo de vida de desarrollo de los proyectos de software de esta entidad.
- Actividades: se describen al tomar como punto de referencia principal, lo expuesto en las etapas de la metodología propuesta por Avella y Gómez (2011) en el apartado 2.11. y según las actividades mencionadas por Pressman (2010) en ese mismo apartado.
- Artefactos: se definieron según lo propuesto por Avella y Gómez (2011) y de acuerdo con los que presenta la Unidad y fueron analizados en el apartado 4.2.3.

Por otro lado, el diseño de las interacciones (mostradas adelante en el apartado 5.2.) propuestas para el sistema, se dividen según las etapas del ciclo de vida de desarrollo de los proyectos de software de la Unidad, mostrado en la descripción realizada en el apartado 4.2.4.1. anterior. Además, se agregó una etapa al ciclo de elaboración: “post-liberación”.

Este añadido de etapa surgió como medida para dar respuesta a la causa de la problemática que actualmente, el TD no efectúa una práctica de validación de calidad de software; pues en ningún paso, etapa, o actividad de la metodología de desarrollo, se toma en cuenta la percepción de calidad según los usuarios finales, una vez que el software es liberado en el ambiente de Producción.

#### **4.3.1. Descripción del diseño del sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital**

De forma puntual, se resume la descripción del diseño propuesto para el aseguramiento y el control de la calidad en los proyectos de software de la Unidad en la elaboración de la Figura 24 (mostrada más adelante en el apartado 5.5.), la cual plasma gráficamente, el producto final del trabajo final de graduación.

Esta síntesis gráfica, permite identificar varios niveles que se plantean para abordar el sistema: el primero (de abajo hacia arriba), el nivel del cuerpo de conocimiento que muestra un modelo de calidad de software, el cual se propone integrar en todo el ciclo de vida de desarrollo de los proyectos de la Unidad a través de la asociación de las ópticas de la calidad, las cuales a su vez presentan la definición de los atributos de calidad para cada uno de los productos de software de las etapas de dicho ciclo y, finalmente, para todos esos atributos se presentan al menos dos o tres métricas que se podrían medir de forma objetiva y estandarizada.

El segundo nivel se asoció con las etapas presentes del ciclo de vida de desarrollo de los proyectos de software del TD, las cuales son siete actualmente, más la propuesta de agregación de la “post - liberación” descrita en el apartado anterior. Esto permitió establecer la sistematización de la ruta a lo largo de estas etapas, para las actividades del aseguramiento y control de la calidad del software.

Finalmente, en la Figura 24, se presenta la interacción principal y los tipos de relaciones entre los tres elementos que fundamentan el sistema propuesta, el cual se debería aplicar a todos los proyectos de software de la Unidad y está en constante funcionamiento. Este nivel además presenta la integración que se propone con herramientas de software (se ahondará esta propuesta en el siguiente capítulo) para la ejecución de las actividades presentes de las interacciones del sistema.

#### **4.3.2. Propuesta de integración de herramientas de software**

Gracias a la entrevista que se llevó a cabo al primer profesional experto en calidad de software (consultar apartado 4.1.1.), se reconoció la existencia de herramientas de software (desarrolladas previamente) que se pueden utilizar para ejecutar actividades referentes al aseguramiento y control de la calidad del software, con el fin de automatizar el soporte de dichas actividades.

Debido a esto, se propuso la búsqueda, comparación analítica y selección de varias herramientas de software que se pudieran integrar dentro de la ejecución de las actividades e interacciones del sistema, para el aseguramiento y el control de la calidad en los proyectos de software del TD.

Para esto, primeramente, se efectuó la especificación, junto a la colaboradora encargada de la asesoría del presente trabajo final de graduación dentro de la Unidad (ver apéndice H), de las funcionalidades requeridas para estas herramientas de software.

A continuación, mediante la Tabla 7, se describen estos requerimientos funcionales mínimos.

Tabla 7. Requerimientos de funcionalidades para herramienta de software del sistema

<b>No. de funcionalidad</b>	<b>Descripción de requerimiento</b>
R01	Gestionar documentos y distintos de activos de conocimiento del sistema (artefactos).
R02	Manejar listas de chequeo o rúbricas para la evaluación de las métricas de calidad en las distintas etapas del ciclo de vida de desarrollo de los proyectos de software de la Unidad, basadas en la ISO/IEC 9126 debido a que esta es la propuesta del trabajo final de graduación, la cual se describe en el apartado siguiente.
R03	Gestionar planes de prueba (funcionales y totales).
R04	Gestionar hallazgos (registro y clasificación por tipo y estado) sobre la ejecución de los planes de prueba y control automatizado de las fechas y horas de registro de realización de los mismos.
R05	Asociar hallazgos con funcionalidades y, a su vez, estas con casos de prueba que representan los casos de uso de un requerimiento de software.
R06	Reportar visualmente los resultados sobre ejecución de planes de prueba y gestión de sus hallazgos correspondientes.
R07	Notificar mediante algún medio (correo electrónico o similar), sobre conclusión de evaluación de alguna actividad o pruebas.
R08	Gestionar de forma automática el informe (reportes gráficos) sobre la ejecución de los planes de prueba.

<b>No. de funcionalidad</b>	<b>Descripción de requerimiento</b>
R09	Manejar tipos de usuario (al menos dos como mínimo) para el detalle de pruebas y evaluación de las mismas.

Fuente: Elaboración propia.

En esta elaboración colaborativa de la especificación de requerimientos de funcionalidades, también se documentaron las categorías o atributos de las herramientas de software, las cuales se debían recabar con el objetivo de efectuar el posterior análisis comparativo y selección de la misma. Dichas categorías se especifican a continuación:

- Código abierto: si es código abierto o no, para efectos de modificación de la herramienta y adaptación a la Unidad.
- Software libre: si es software con licencia privativa o libre, para efectos de valoración de costos.
- Plataforma operativa: plataforma de sistema operativo en donde se debe instalar y ejecutar sus funcionalidades.
- Funcionalidades: cumplimiento de las funcionalidades que presenta cada herramienta de software, con respecto a las que se definieron como requisitos, mostradas en la Tabla 7.
- Comunidad: existencia de alguna comunidad o foros para soporte de la herramienta de software.
- Ambiente: las tecnologías (de arquitectura principalmente) o lenguajes de programación en las cuales se fundamenta la herramienta.
- Funcionalidades adicionales: estas no fueron contempladas dentro de la Tabla 7 y pueden significar algún valor para el TD.

De esta forma, de acuerdo con la base de los requerimientos mínimos de funcionalidades y las categorías de evaluación, se realizó una búsqueda y un análisis

comparativo de las herramientas de software para la ejecución de las actividades del sistema propuesto, cuatro de estas herramientas fueron sugeridas por el entrevistado uno (Consultar apartado 4.1.1). Dicho análisis se desarrolla en la Tabla 8 que se muestra seguidamente.

Tabla 8. Análisis comparativo de herramientas de software para el sistema

<b>Nombre de herramienta</b> <b>Categorías de análisis comparativo</b>	<i>Tarantula</i> <sup>9</sup>	<i>Test Link</i> <sup>10</sup>	<i>rth-turbo</i> <sup>11</sup>	<i>Testitools</i> <sup>12</sup>
<b>Código abierto</b>	Sí	Sí	Sí	Sí
<b>Software libre</b>	Sí	Sí	Sí	Sí
<b>Plataforma operativa</b>	<i>Linux</i>	<i>Linux</i>	<i>Windows</i>	<i>Windows</i>
<b>Funcionalidades que cumplen</b>	Todas excepto 01 y 02.	Todas excepto 01 y 02.	Todas excepto 01 y 02.	Todas excepto 01 y 02.
<b>Comunidad</b>	Sí	Sí	No	Sí
<b>Ambiente</b>	<i>Ruby-on-Rails, APACHE y MySQL.</i>	<i>PHP, APACHE y MySQL.</i>	<i>PHP, APACHE y MySQL.</i>	<i>PHP, APACHE y MySQL.</i>
<b>Funcionalidades adicionales</b>	Gestión de requerimientos	Gestión de requerimientos	Envío de correos	Envío de correos
	Integración con <i>Jira</i> y <i>Bugzilla</i>	Integración con <i>Bugzilla</i>		
	Máquina virtual con ambiente configurado	Envío de correos		

<sup>9</sup> *Testia* (2017).

<sup>10</sup> *TestLink Development Team* (2014).

<sup>11</sup> *Google Code* (2016).

<sup>12</sup> *QaTestingTools* (2003).

Nombre de herramienta	<i>Tarantula</i> <sup>9</sup>	<i>Test Link</i> <sup>10</sup>	<i>rth-turbo</i> <sup>11</sup>	<i>Testitools</i> <sup>12</sup>
Categorías de análisis comparativo				
	Envío de correos			

Fuente: Elaboración propia con información consultada de páginas web de cada herramienta de software.

Con el resultado del análisis, se propone la selección de *Tarantula* como herramienta de software para la ejecución de las actividades propuesta en el sistema, para el aseguramiento y el control de la calidad en los proyectos de software del TD, específicamente, para las actividades que se desarrollan en la etapa de Pruebas del ciclo de vida de desarrollo de los proyectos y el sistema planteado.

Esta selección obedece a que dicha herramienta presenta características similares a las demás opciones comparadas; sin embargo, destaca en las funcionalidades adicionales, las cuales podrían brindar los siguientes beneficios a la Unidad: reducción de tiempos de implementación (debido a la máquina virtual que contiene el ambiente preconfigurado) y también podría crear la capacidad a futuro de efectuar actividades de automatización de pruebas (mediante la integración con *Jira* y *Bugzilla*), junto con el enlace de otras herramientas; acciones que son propuestas como trabajos futuros en el apartado 4.4. posterior.

Por otro lado, a raíz del incumplimiento de requerimientos que presenta *Tarantula* con dos funcionalidades específicas requeridas (01 y 02 según la Tabla 7), se propone el uso de cinco herramientas (las cuales se integran entre sí) que contiene el TD en su plataforma de aprendizaje electrónico para la ejecución y la gestión de las actividades del sistema requeridas a lo largo de las etapas del ciclo de vida de desarrollo de los proyectos de software. Estas herramientas de software son descritas y se amplia su conexión respecto al sistema propuesto en el apartado de solución 5.3.

#### **4.3.3. Propuesta de integración de modelo de calidad de software**

Otra integración que se planteó para el diseño del sistema especificado, es el modelo de calidad de software basado en la ISO/IEC 9126 propuesta por la ISO (2001) y desarrollado ampliamente, en el apartado 2.12. del presente trabajo final.

Este modelo cimienta la base de verificación, validación y evaluación de la calidad de software que responde a las causas de problemáticas asociadas a este tipo de situación en los apartados 4.2.4.2. y el 4.2.4.3., descritos anteriormente.

La utilización de este modelo se deberá abordar luego, para la construcción de los artefactos y la definición específica del modo de ejecución de las actividades donde será utilizado, principalmente, para las actividades de evaluación por medio de rúbricas que se plantean en cada etapa de las interacciones del sistema en el apartado 5.2. del capítulo siguiente al presente.

La incorporación y el uso de dicho modelo de calidad de software es avalado y apoyado por múltiples estudios de autores en el mencionado apartado 2.12. del capítulo dos, además fue recomendado por los entrevistados profesionales expertos en el tema de calidad de software, lo cual se denota en el apartado 4.1.3. del presente capítulo.

#### **4.3.4. Propuesta de plan de implementación**

Con el fin de ejecutar la propuesta de solución, formulada más adelante en el capítulo cinco, se elaboró un plan de implementación, el cual contiene una estimación de los recursos necesarios por actividad para implantar este abordaje de solución. Este plan fue presentado y aprobado por colaboradores representantes de cada área del TD, tal y como se muestra en el apéndice I.

El plan de implementación se elaboró por medio de una propuesta de etapas secuenciales, las cuales contienen el detalle en tablas de las actividades por realizar, los colaboradores encargados de la Unidad para ejecutarlas (definidos con la notación y descripción de los roles que se muestra más adelante en el apartado 5.1.1.) y una estimación (en horas) del esfuerzo necesario por rol. De acuerdo con K. M. Gómez

(comunicación personal, 30 de mayo, 2017), estas etapas se deben realizar de forma gradual para cada fase del ciclo de vida de desarrollo de los proyectos de software del TD. Seguidamente, se describen dichas etapas de implementación.

#### 4.3.4.1. Etapa uno: inventario de artefactos

En esta etapa se plantean las actividades necesarias para hacer un recuento de los recursos con los cuales cuenta el TD actualmente, para utilizarlos en la construcción y la documentación requeridas de los elementos principales del sistema.

Tabla 9. Etapa uno del plan de implementación: inventario de artefactos

No. actividad	Actividad	Rol	Esfuerzo
01	Revisar artefactos por área que sean útiles para la definición del sistema: documentación de procesos y actividades y documentos (plantillas, estándares, guías, entre otros).	CA	3 horas
		CV	3 horas
		CTD	3 horas
		ED	3 horas
		GC	3 horas
02	Elaborar documento de resumen de inventario de artefactos, debe incluir como mínimo: nombre, fecha de creación, fecha de actualización, objetivo y descripción.	GC	10 horas

Fuente: Elaboración propia.

#### 4.3.4.2. Etapa dos: preparación de artefactos

Una vez realizado el inventario de los recursos disponibles en la Unidad, se procedería a la preparación de los artefactos requeridos para la ejecución y resultado de las actividades del sistema. Esta preparación se puede desarrollar de dos formas: si existe algún recurso equivalente a un artefacto propuesto del sistema, se realiza una adaptación a lo propuesto de referencia en el apartado 5.1. más adelante; la segunda se presenta cuando no hay ningún recurso disponible que funcione como base de referencia para la construcción del artefacto; por lo tanto, se debe elaborar desde “cero”, según lo propuesto en el mismo apartado detallado de forma anterior.

Tabla 10. Etapa dos del plan de implementación: preparación de artefactos

No. actividad	Actividad	Rol	Esfuerzo
01	Adaptar inventario de artefactos existentes a propuesta de sistema.	CA	5 horas
		CV	3 horas
		CTD	2 horas
		ED	15 horas
		GC	10 horas
02	Elaborar artefactos de inventario faltantes para sistema.	CA	7 horas
		CV	10 horas
		CTD	5 horas
		ED	12 horas
		GC	15 horas

Fuente: Elaboración propia.

#### 4.3.4.3. Etapa tres: pruebas y sistematización

Para esta etapa se propone realizar la ejecución de pruebas para el modo de desarrollo del sistema propuesto, además de la sistematización de sus resultados y, en caso de requerirse, su debido ajuste de acuerdo con estos.

Tabla 11. Etapa tres del plan de implementación: pruebas y sistematización

No. actividad	Actividad	Rol	Esfuerzo
01	Probar modo de ejecución del sistema en al menos tres proyectos de software.	GC	30 horas
02	Ajustar sistema según los resultados de las pruebas del mismo.	CA	6 horas
		CV	5 horas
		CTD	3 horas
		ED	15 horas
		GC	20 horas

Fuente: Elaboración propia.

#### 4.3.4.4. Etapa cuatro: lineamientos

Esta etapa se puede desarrollar en paralelo con la anterior o bien, de forma secuencial; pues dentro de esta se propone la creación de lineamientos que dicten el modo de ejecución y características del sistema, el cual debería seguir con base en las pruebas y la sistematización de las mismas dentro de la Unidad.

Tabla 12. Etapa cuatro del plan de implementación: lineamientos

No. actividad	Actividad	Rol	Esfuerzo
01	Elaborar política de cada área y rol dentro del sistema para el aseguramiento y control de la calidad en los proyectos de software del TD.	GC y CA	5 horas (para cada rol)
		GC y CV	5 horas (para cada rol)
		GC y CDT	5 horas (para cada rol)
		GC y ED	5 horas (para cada rol)
		GC	7 horas

Fuente: Elaboración propia.

#### 4.3.4.5. Etapa cinco: capacitación del TEC Digital

La etapa final que se propone es para capacitar al TD en cuanto a todo el uso y la ejecución del sistema, en relación con todas las áreas, roles, actividades y artefactos dentro del mismo.

Tabla 13. Etapa cinco del plan de implementación: capacitación del TEC Digital

No. actividad	Actividad	Rol	Esfuerzo
01	Preparación de capacitación por área y rol.	GC y CA	3 horas
		GC y CV	3 horas
		GC y CDT	3 horas
		GC y ED	3 horas
		GC	3 horas
02	Ejecución de capacitación.	GC y CA	1.5 horas
		GC y CV	1.5 horas

No. actividad	Actividad	Rol	Esfuerzo
		GC y CDT	1.5 horas
		GC y ED	1.5 horas
		GC	3 horas

Fuente: Elaboración propia.

#### 4.4. Trabajos futuros

Como resultado del análisis consiguiente de la metodología y el desarrollo en general del presente trabajo final, se detallan tres trabajos futuros que podrían complementarse, el diseño del sistema para el aseguramiento y control de la calidad en los proyectos de software del TD que se propone. A continuación, se realiza dicha especificación:

- Automatización de actividades del sistema:
  - Revisión automática de código en las etapas de desarrollo y pruebas del sistema propuesto. Esto se puede realizar mediante la integración de herramientas de software libre que presentan esta funcionalidad, las cuales fueron recomendadas por el entrevistado uno. Un ejemplo de estas herramientas: *Selenium*, la cual es un conjunto de herramientas de software que automatizan tareas como la descrita y otras más, relacionadas a las pruebas de software, según SeleniumHQ (2017).
  - De acuerdo con M. Chacón (comunicación personal, 01 de febrero, 2017), se podría implementar una revisión automática de los artefactos que se proponen como elementos del sistema, esto mediante su constitución en un formato como el que maneja la herramienta de software libre *LaTeX*. Dicho formato se puede utilizar para el arreglo de texto por medio de etiquetas y partes específicas, según The LATEX Project (2017), las cuales podrían ser utilizadas como variables de comprobación de extensión mínima en los artefactos mediante alguna herramienta de software.

- Al contar con la implementación del sistema planteado en el presente trabajo final, se debería implementar el plan de aseguramiento de la calidad propuesto por el entrevistado uno; pues se propiciará la instauración de actividades que el trabajo de investigación realizado por dicho entrevistado suponía como necesarias para implementar su propuesta de solución.
- Desarrollo de los procesos y los procedimientos que implican cada una de las actividades del sistema, lo cual a su vez supondrá la creación y la instauración de lineamientos más específicos con los cuales se ejecutará el sistema.

#### **4.5. Cambio en el trabajo final**

Además, cabe destacar en los resultados, que se realizó un cambio sobre el contenido planteado en el anteproyecto del presente trabajo, específicamente, para la propuesta de solución. Este cambio fue aprobado por la Coordinación de Trabajos Finales de Graduación (consultar apéndice K) y también por los principales involucrados del proyecto por parte del TD (consultar apéndice L).

De seguido, se especifica un resumen de la información correspondiente para el cambio:

- Justificación del cambio:
  - El cambio surge como medida accionaria ante la restricción que se menciona en el apartado 1.12.
  - El cambio se genera también, como respuesta ante la sugerencia técnica que plantea el entrevistado uno, descrita de forma previa en el apartado 4.1.1. Este experto entrevistado realizó la observación de consultar herramientas de software, las cuales provean los requerimientos de funcionalidades para una necesidad como la que presenta la Unidad antes de diseñar y desarrollar software desde el inicio.

- Alcance del cambio:
  - El cambio no tuvo implicación en los objetivos del presente trabajo final fijados en el anteproyecto; pues los mismos se definieron en torno al diseño de un sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD.
  - El cambio modificó uno de los entregables de producto; pues de forma inicial se tenía completado realizar el diseño de software de una herramienta para integrarla al sistema; sin embargo, se hizo el intercambio por un análisis comparativo entre herramientas de software libre y una propuesta de selección de estas, según los requisitos de la organización (tal y como se mostró en el apartado 4.3.2.).

## **CAPÍTULO V: PROPUESTA DE SOLUCIÓN**

La propuesta de solución a la problemática desarrollada en el capítulo anterior, se especifica en función de un eje principal: el diseño del sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD.

Seguidamente, mediante los sub apartados se derivan: aspectos generales del sistema, las interacciones que se generan a través del mismo y las propuestas de integración de componentes externos a este sistema (herramientas de software y el modelo de calidad).

De manera posterior, se describe la interrelación entre los elementos descritos por medio de una figura que explica de forma gráfica, la consistencia y la ejecución del sistema dentro del ciclo de desarrollo de los proyectos de software de la Unidad y finalmente, se proponen lineamientos mínimos respecto al sistema y un plan de capacitación sobre calidad de software y el mencionado eje principal.

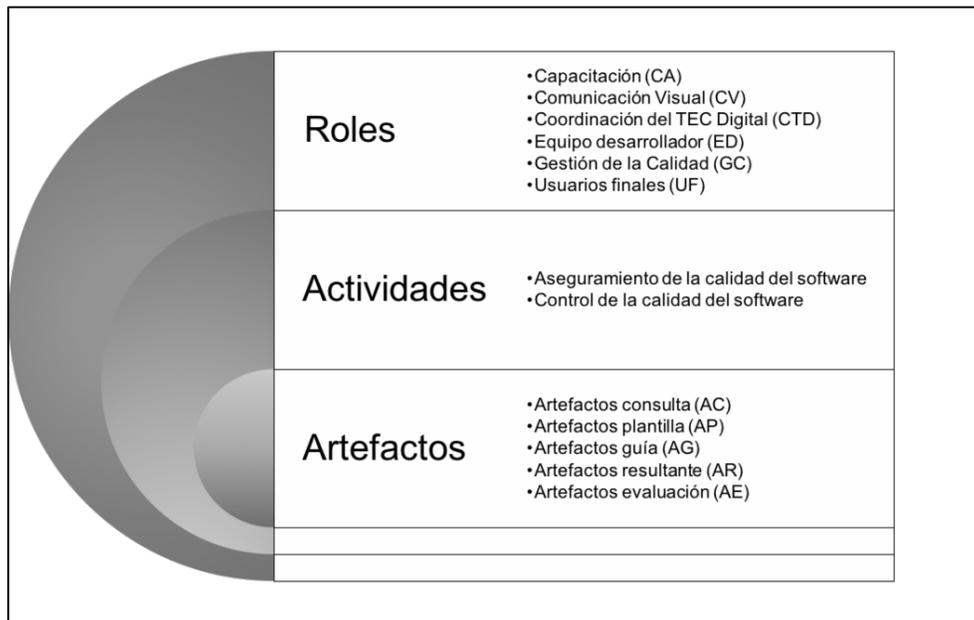
### **5.1. Aspectos generales del sistema**

El eje de la propuesta de solución se compone de tres elementos principales, los cuales se plantean para relacionarse entre sí, tal y como se muestra en la Figura 23, a través del ambiente de interacción entre Gestión de la Calidad y las demás áreas del TD. Estos elementos se derivan de acuerdo con lo propuesto por el entrevistado dos en el apartado 4.1.2.

El ordenamiento de las actividades y los artefactos de cada etapa, se proponen como medio para alcanzar lo planteado por Pressman (2010), por la ISO y el IEEE (2011), por Avella y Gómez (2011) y por los mismos expertos profesionales entrevistados, todos estos criterios describen, de forma resumida, que el aseguramiento de la calidad del software se consigue al planear y sistematizar una ruta de actividades y otros elementos, con el fin de crear esfuerzos puntuales, los cuales soporten acciones de evaluación de esta calidad sobre el ciclo de vida de desarrollo de los proyectos.

Los elementos planteados para el sistema son: los roles (1) que ejecutan las actividades (2), al utilizar y producir con esto artefactos (3). Estos elementos se definen principalmente, en función de lo que expone el entrevistado dos mediante el término de “tríada de calidad”.

Figura 23. Elementos del sistema para el aseguramiento y el control de la calidad en los proyectos de software del TEC Digital



Fuente: Elaboración propia.

A modo de introducción del diseño del sistema, se presentan y describen los tres elementos que lo componen.

### 5.1.1. Roles

Los roles del sistema corresponden a la representación de los colaboradores de las áreas con las cuales cuenta la Unidad y tienen relación directa con el ciclo de vida de desarrollo del software en esta entidad; para esto, se describe de seguido, cada uno de estos roles:

- Capacitación (CA): alude al colaborador que coordina dicha área, o bien, al colaborador o colaboradores que se deleguen, para la ejecución de las actividades e interacción dentro del sistema, en representación de esta área.

- Comunicación Visual (CV): representa al colaborador o colaboradores relacionados al ciclo de vida de desarrollo en cuanto al diseño de interfaces gráficas de usuarios.
- Coordinación del TEC Digital (CTD): se refiere al colaborador que coordina la Unidad.
- Equipo desarrollador (ED): se compone del colaborador o colaboradores que forman parte del equipo desarrollador del proyecto de software.
- Gestión de la Calidad (GC): es el colaborador o colaboradores encargados de esta área y quienes ejecutan la mayor parte del sistema, en especial las actividades de revisión y evaluación de la calidad.
- Usuarios finales (UF): es uno o varios de los usuarios finales de algún proyecto de software, pueden ser constituidos por cualquier tipo de usuario de la comunidad TEC.

### **5.1.2. Actividades**

Las actividades son las acciones que realiza cada uno de los roles, ya sea en forma individual o conjunta, para la ejecución concreta del sistema, dentro de cada etapa del proyecto de software, de acuerdo con las presentes en la metodología 2.0 de la Unidad.

De esta forma, según lo que expone Pressman (2010), brindan el ambiente adecuado para el aseguramiento y el control de la calidad del ciclo de vida de desarrollo de todos los proyectos de software del TD, puntualmente, para la definición de tareas específicas de los procesos de calidad del software, con el fin de lograr una correcta ejecución de los proyectos de este tipo.

### **5.1.3. Artefactos**

Los artefactos son los activos del sistema (entre otros: documentos, plantillas, estándares y políticas) que utilizan los roles para la ejecución de las actividades o, de igual forma, son el resultado de la ejecución de estas; a estos, la ISO y y el IEEE

(2011) los define como productos de software. En el sistema, se plantea la gestión de cinco tipos de artefactos, los cuales se exponen a continuación:

- Artefacto consulta (AC): se refiere a un producto resultante del mismo sistema y del desarrollo del proyecto de software, el cual se utiliza como punto de consulta y referencia, con el fin de emitir un criterio o elaborar otro artefacto.
- Artefacto plantilla (AP): significa un activo del sistema que funge como base o plantilla para la elaboración de un artefacto resultante.
- Artefacto guía (AG): representa un activo de conocimiento que se ha elaborado previamente, el cual establece un conjunto de guías o lineamientos; por lo tanto, se utiliza como referencia para elaborar un artefacto resultante bajo ciertos parámetros definidos de forma anticipada.
- Artefacto resultante (AR): es el activo de producto de alguna actividad del sistema, puede ser un documento de consulta para una etapa o actividad posterior, o bien, representar un activo para ser evaluado.
- Artefacto evaluación (AE): es el artefacto de herramienta que genera la evaluación y calificación de la calidad de algún otro artefacto o componente del sistema, la mayoría de estos se especifican mediante rúbricas de evaluación o listas de cotejo.

Además, para la constitución específica de los artefactos requeridos en el ciclo de vida de desarrollo de los proyectos de software del TD (en el siguiente apartado serán detallados), se propone una base de referencia por las etapas de dicho ciclo, la cual se muestra en la Tabla 14, recabada según la base teórica que se construyó.

Tabla 14. Etapas del sistema y referencias teóricas para la preparación de sus artefactos

Etapa del ciclo	Referencia de preparación para sus artefactos
Iniciación y planeación	-ISO/IEC 9126-1 para la evaluación general por medio del modelo calidad de software.
Análisis	-ISO/IEC 9126-1 para la evaluación general por medio del

Etapa del ciclo	Referencia de preparación para sus artefactos
	<p>modelo calidad de software.</p> <p>-ISO/IEC/TR 9126-3 para la evaluación de las métricas de calidad internas del software.</p> <p>-IEEE std 830-1998 para la definición de las plantillas y las rúbricas de evaluación sobre especificaciones de requerimientos de software. (Propuesto por Avella y Gómez [2011] y por el entrevistado uno).</p> <p>-Documento de Arquitectura con el que cuenta el TD.</p>
Diseño	<p>-ISO/IEC 9126-1 para la evaluación general por medio del modelo calidad de software.</p> <p>-ISO/IEC/TR 9126-3 para la evaluación de las métricas de calidad internas del software.</p> <p>-ISO/IEC/TR 9126-2 para la evaluación de las métricas de calidad externas del software.</p> <p>-IEEE 1016-2009 para la descripción del diseño del software y la elaboración de rúbricas de evaluación de la misma. (Propuesto por el entrevistado uno en el apartado 4.1.1.).</p> <p>-Documento de Arquitectura con el que cuenta el TD.</p>
Desarrollo	<p>-ISO/IEC 9126-1 para la evaluación general por medio del modelo calidad de software.</p> <p>-ISO/IEC/TR 9126-3 para la evaluación de las métricas de calidad internas del software.</p> <p>-ISO/IEC/TR 9126-2 para la evaluación de las métricas de calidad externas del software.</p> <p>-Manual y políticas de Git que presenta la Unidad.</p> <p>-Guía de programación con la que cuenta el TD.</p>
Pruebas	<p>-ISO/IEC 9126-1 para la evaluación general por medio del modelo calidad de software.</p> <p>-IEEE 829-2008 para la descripción de planes de pruebas</p>

Etapa del ciclo	Referencia de preparación para sus artefactos
	del software de la Unidad. (propuesto por Avella y Gómez [2011] y el entrevistado uno en el apartado 4.1.1.). -Plantilla de Control de Calidad que desarrolló el TD.
Capacitación	-ISO/IEC 9126-1 para la evaluación general por medio del modelo calidad de software. -ISO/IEC/TR 9126-4 para la evaluación de las métricas de calidad en uso del software.
Post-liberación	-ISO/IEC 9126-1 para la evaluación general por medio del modelo calidad de software. -ISO/IEC/TR 9126-3 para la evaluación de las métricas de calidad internas del software. -ISO/IEC/TR 9126-2 para la evaluación de las métricas de calidad externas del software. -ISO/IEC/TR 9126-4 para la evaluación de las métricas de calidad en uso del software. -IEEE std 1028-2008 para la revisión y auditoría de la calidad del proyecto de software después de su liberación. (Propuesto por Avella y Gómez [2011]).

Fuente: Elaboración propia.

Cabe destacar, que todas las propuestas de uso de la ISO/IEC 9126 como referencia para la construcción de los artefactos de cada etapa, así como de sus reportes técnicos, las recomiendan Avella y Gómez (2011) y los dos profesionales expertos en calidad de software que se entrevistaron. De igual forma, este planteamiento se respalda por múltiples autores citados en el apartado 2.12. donde se exponen las ventajas y forma de empleo de dicho modelo de calidad de software.

Por otro lado, se agrega como referencia de utilización, para la consistencia de los artefactos, documentos que ya se han desarrollado en la Unidad, los cuales serán de utilidad para la elaboración de los elementos. Para ello, se deberá realizar un

proceso de análisis y ajuste de cada uno de estos, según las referencias desarrolladas en la Tabla 14.

## 5.2. Interacciones de los elementos principales del sistema

El diseño del sistema propuesto y su futura implementación generará interacciones entre los tres elementos principales que se desarrollaron anteriormente (roles, actividades y artefactos). Estas interacciones se desglosan y se ordenan mediante una mezcla estratégica, que se desarrolla, entre las etapas que conforman la metodología actual del TD para el ciclo de vida de desarrollo de los proyectos de software (consultar Figura 18 en el capítulo 4) y las propuestas por Avella y Gómez (2011). Las mismas se especifican a continuación:

### 5.2.1. Iniciación y planeación

En esta etapa se realizará todo lo referente al aseguramiento y el control de la calidad de los proyectos de la Unidad desde el momento de su inicio; es decir, en la planeación de sus recursos, los requerimientos a nivel macro del proyecto y otros elementos que se identifican en la Tabla 15.

Tabla 15. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de iniciación y planeación

Roles	Actividades	Artefactos
CV y ED	Elaborar acta constitutiva del proyecto de software.	AP: plantilla de documento de acta constitutiva.
		AR: documento de acta constitutiva.
GC	Evaluar calidad de acta constitutiva del proyecto de software.	AE: rúbrica de evaluación de acta constitutiva.
		AR: rúbrica aplicada de evaluación de acta constitutiva.
CTD, CV y ED	Aprobar acta constitutiva del proyecto de software.	AR: acta constitutiva aprobada por CTD, CV y ED.

Fuente: Elaboración propia.

### 5.2.2. Análisis

En esta etapa resulta clave el aseguramiento y el control de la calidad para sus actividades y artefactos; pues para el TD es vital mejorar en lo referente al desarrollo de los requerimientos y su calidad, esto de acuerdo con el entrevistado uno. Para ello, Arias (2007) propone una serie de características con el objetivo de realizar con efectividad el proceso de análisis de requerimientos, las cuales serán tomadas en cuenta para el sistema, específicamente, en las actividades que se muestran en la Tabla 16.

Tabla 16. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de análisis

Roles	Actividades	Artefactos
ED y UF	Elaborar documento de especificación de requerimientos de software.	AP: plantilla de documento de especificación de requerimientos de software.
		AR: documento de especificación de requerimientos de software.
GC	Evaluar calidad de documento de especificación de requerimientos de software.	AE: rúbrica de evaluación de documento de especificación de requerimientos de software.
		AR: rúbrica aplicada de evaluación de documento de especificación de requerimientos del proyecto de software.
ED y UF	Elaborar plan de pruebas funcionales.	AP: plantilla de documento de plan de pruebas funcionales de software.
		AR: documento de plan de pruebas funcionales.
GC	Evaluar calidad de plan de pruebas funcionales.	AE: rúbrica de evaluación de plan de pruebas funcionales de software.
		AR: rúbrica aplicada de evaluación requerimientos de software.
CTD y ED	Aprobar documento de especificación de requerimientos software.	AR: documento de especificación de requerimientos de software aprobado por CTD y ED.

Fuente: Elaboración propia.

### 5.2.3. Diseño

Para esta etapa del diseño, por medio de sus actividades será importante realizar la evaluación de la trazabilidad que se presenta con respecto a la etapa anterior de análisis. Las interacciones para esta etapa se muestran en la Tabla 17.

Tabla 17. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de diseño

Roles	Actividades	Artefactos
ED y CV	Elaborar documento de arquitectura y diseño de software.	AG: guía de programación del TEC Digital.
		AC: documento de especificación de requerimientos de software.
		AP: plantilla de documento de arquitectura y diseño de software.
		AR: documento de arquitectura y diseño de software.
GC	Evaluar calidad de documento de arquitectura y diseño de software.	AE: rúbrica de evaluación de documento de arquitectura y diseño de software.
		AR: rúbrica aplicada de evaluación de documento de arquitectura y diseño de software.
CTD, CV y ED	Aprobar documento de arquitectura y diseño de software.	AR: documento de arquitectura y diseño de software aprobado por CTD, CV y ED.

Fuente: Elaboración propia.

### 5.2.4. Desarrollo

Para la etapa de desarrollo se presentan las actividades que giran en torno, principalmente, al código fuente del proyecto de software; sin embargo, cabe destacar la propuesta que se realiza para iniciar desde esta etapa, con la ejecución de planes de pruebas funcionales, definidos según los requerimientos prácticos de etapas anteriores.

Tabla 18. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de desarrollo

Roles	Actividades	Artefactos
ED	Elaborar código fuente de software.	AG: guía de programación del TEC Digital.
		AG: políticas de gestión del GIT del TEC Digital.
		AC: documento de especificación de requerimientos software.
		AC: documento de arquitectura y diseño de software.
		AR: código fuente de software almacenado en GIT.
GC	Evaluar calidad de código fuente de software.	AE: rúbrica de evaluación de código fuente de software.
		AR: rúbrica aplicada de evaluación de código fuente de software.
GC	Ejecutar plan de pruebas funcionales.	AC: documento de plan de pruebas funcionales.
		AP: plantilla de documento de reporte de ejecución de plan de pruebas funcionales.
		AR: documento de reporte de plan de pruebas funcionales.
GC	Aprobar migración a ambiente etapa de pruebas.	AP: plantilla de documento de aprobación de migración a etapa de pruebas.
		AR: documento de aprobación por parte de GC de migración a etapa de pruebas.

Fuente: Elaboración propia.

### 5.2.5. Pruebas

Esta es la etapa en donde más se realizarán interacciones entre el rol de GC y el resto de colaboradores de la Unidad. Representa una elevada carga de trabajo para dicha área; por lo tanto será conveniente la automatización al máximo (propuesta en el apartado posterior) de sus procesos y actividades, las cuales son mostradas en la Tabla 19.

Tabla 19. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de pruebas

Roles	Actividades	Artefactos
ED y GC	Elaborar plan de pruebas totales.	AC: acta constitutiva.
		AC: documento de especificación de requerimientos software.
		AC: documento de arquitectura y diseño de software.
		AC: documento de reporte de plan de pruebas funcionales.
		AP: plantilla de documento de plan de pruebas totales de software.
		AR: documento de plan de pruebas totales de software.
GC	Ejecutar plan de pruebas totales.	AC: documento de plan de pruebas totales de software.
		AP: plantilla de documento de reporte de ejecución de plan de pruebas.
		AR: documento de reporte de ejecución de plan de pruebas totales.
GC	Evaluar calidad de software, según documento de reporte de ejecución de plan de pruebas totales.	AE: rúbrica de evaluación de documento de reporte de ejecución de plan de pruebas totales.
		AR: rúbrica aplicada de evaluación de documento de reporte de ejecución de plan de pruebas totales.
ED, GC y CA	Elaborar plan piloto de pruebas con usuarios finales.	AC: documento de plan de pruebas totales de software.
		AC: documento de reporte de ejecución de plan de pruebas totales.
		AP: plantilla de documento de plan pilotos de pruebas con usuarios finales.

Roles	Actividades	Artefactos
		AR: documento de plan piloto de pruebas con usuarios finales.
ED, GC, CA y UF	Ejecución de plan piloto de pruebas con usuarios finales.	AC: documento de plan piloto de pruebas con usuarios finales.
		AP: plantilla de documento de reporte de ejecución de plan piloto de pruebas con usuarios finales.
		AR: documento de reporte de ejecución de plan piloto de pruebas con usuarios finales.
GC	Evaluar resultado de plan piloto de pruebas con usuarios finales.	AE: rúbrica de evaluación de documento de reporte de ejecución de plan piloto de pruebas con usuarios finales.
		AR: rúbrica aplicada de evaluación de documento de reporte de ejecución de plan piloto de pruebas con usuarios finales.
GC	Aprobar paso a etapa de capacitación.	AP: plantilla de documento de aprobación de paso a capacitación.
		AR: documento de aprobación de paso a capacitación.

Fuente: Elaboración propia.

### 5.2.6. Capacitación

Esta etapa ya existe dentro de la metodología de desarrollo del TD; sin embargo, de acuerdo con la observación efectuada del proceso de control de calidad, no se estaba realizando. Cabe resaltar la cooperación que se muestra en la primera actividad de la Tabla 20; pues se requiere que el ED tenga comunicación directa con CA, lo cual podría ocasionar una mayor retroalimentación sobre aspectos de calidad en uso, según la ISO (2001).

Tabla 20. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de capacitación

Roles	Actividades	Artefactos
ED y CA	Elaborar manual de capacitación de software.	AC: documento de reporte de ejecución de plan piloto de pruebas con usuarios finales.
		AP: plantilla de documento de manual de capacitación de software.
		AR: documento de manual de capacitación de software.
GC	Evaluar calidad de manual de capacitación de software.	AE: rúbrica de evaluación de documento de manual de capacitación de software.
		AR: rúbrica aplicada de evaluación de documento de manual de capacitación de software.

Fuente: Elaboración propia.

### 5.2.7. Cierre del proyecto

Esta es una etapa fundamental dentro del ciclo de vida de desarrollo de los proyectos; pues es cuando se realiza la mayor consulta de artefactos, lo cual ocasiona que se evalúe la calidad de los productos obtenidos mediante múltiples criterios de revisión y, de esa forma, se decida si el software se libera o no al ambiente de Producción.

Dicha etapa no se presenta en lo planteado por Avella y Gómez (2011); sin embargo, a través de las interacciones propuestas en la Tabla 21, se pretende abordar la fase que el TD contiene en su metodología de desarrollo de los proyectos de software.

Tabla 21. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de cierre de proyecto

Roles	Actividades	Artefactos
ED	Elaborar documento de cierre de proyecto.	AC: acta constitutiva.
		AC: rúbrica aplicada de evaluación de acta constitutiva.

Roles	Actividades	Artefactos
		<p>AC: documento de especificación de requerimientos software.</p> <p>AC: rúbrica aplicada de evaluación de documento de especificación de requerimientos de software.</p> <p>AC: documento de arquitectura y diseño de software.</p> <p>AC: rúbrica aplicada de evaluación de documento de arquitectura y diseño de software.</p> <p>AC: código fuente de software almacenado en GIT.</p> <p>AC: rúbrica aplicada de evaluación de código fuente de software.</p> <p>AC: documento de reporte de ejecución de plan de pruebas totales.</p> <p>AC: rúbrica aplicada de evaluación de documento de reporte de ejecución de plan de pruebas totales.</p> <p>AC: documento de reporte de ejecución de plan piloto de pruebas con usuarios finales.</p> <p>AC: rúbrica aplicada de evaluación de documento de reporte de ejecución de plan piloto de pruebas con usuarios finales.</p> <p>AC: documento de manual de capacitación de software.</p> <p>AC: rúbrica aplicada de evaluación de documento de manual de capacitación de software.</p> <p>AP: plantilla de documento de cierre de proyecto.</p> <p>AR: documento de cierre de proyecto.</p>
GC	Evaluar calidad de documento de cierre de proyecto.	<p>AE: rúbrica de evaluación de documento de cierre de proyecto.</p> <p>AR: rúbrica aplicada de evaluación de documento de cierre de proyecto.</p>
CTD	Aprobar liberación de	AC: documento de cierre de proyecto.

Roles	Actividades	Artefactos
	proyecto de software.	AC: rúbrica aplicada de evaluación de documento de cierre de proyecto.
		AP: plantilla de documento de aprobación de liberación de software.
		AR: documento firmado de aprobación de liberación de software.

Fuente: Elaboración propia.

### 5.2.8. Post-liberación del proyecto

Esta etapa no existe dentro de la metodología del ciclo de vida de los proyectos del TD, ni en la propuesta hecha por Avella y Gómez (2011). Sin embargo, se introduce como un elemento que apunta hacia la evaluación de la calidad de dichos proyectos, para identificar los niveles de cumplimiento que se tienen respecto a los requerimientos y las necesidades de los usuarios finales; es decir, lo que la ISO (2001) define como calidad en uso. Las interacciones de esta etapa se muestran en la Tabla 22.

Tabla 22. Diseño de sistema para aseguramiento y control de la calidad de software del TEC Digital: etapa de post-liberación del proyecto

Roles	Actividades	Artefactos
ED	Elaborar cuestionario de satisfacción de usuarios finales.	AP: plantilla de cuestionario de satisfacción de usuarios finales.
	Aplicar cuestionario a usuarios finales.	AR: cuestionario de satisfacción de usuarios finales.
GC	Evaluar resultado de cuestionario a usuarios finales.	AE: rúbrica de evaluación de resultados de cuestionario a usuarios finales.
		AR: rúbrica aplicada de evaluación de resultados de cuestionario a usuarios finales.
GC y CTD	Evaluar calidad total del proyecto de software.	AE: rúbrica de evaluación de calidad total de proyecto de software.
		AR: rúbrica aplicada de evaluación de calidad total de proyecto de software.

Fuente: Elaboración propia.

### **5.3. Integración de software en el sistema**

En la ejecución del sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD y, específicamente, para las actividades que ejecutará el rol de Gestión de la Calidad (GC), se plantea la integración de una serie de herramientas que permitirán automatizar parte de estas actividades.

Las herramientas planteadas para integrarse dentro del sistema, se dividen en dos tipos: las que son internas de la Unidad (ya existentes dentro de la plataforma de aprendizaje electrónico) y la que es externa, *Tarantula*. Estas integraciones propuestas, se describen de seguido.

#### **5.3.1. Herramientas internas del TEC Digital**

La Unidad cuenta en su plataforma de aprendizaje electrónico con una serie de herramientas de software, las cuales se pueden aplicar para la ejecución de las actividades y el manejo de los artefactos propuestos en el sistema.

Dentro de estas, existe una carpeta en línea (se le conoce como “Sección de Documentos”), donde se puede albergar todo tipo de documentos y archivos, lo cual se identifica como una ventaja para contener todos los artefactos propuestos para el sistema e incluso, es posible manejar un modelo de gestión de sub carpetas en donde se divida por etapa del ciclo de vida de desarrollo de los proyectos de software.

Además, dicha plataforma presenta un conjunto de herramientas, las cuales permiten la calificación de criterios de evaluación, esto incluye además, la ventaja de asociar rúbricas de evaluación (o en su defecto, listas de cotejo) a estas calificaciones, lo cual es óptimo para las actividades de evaluación que se plantean realizar dentro del sistema. Las rúbricas de evaluación o listas de cotejo se utilizarán como herramientas para determinar el nivel de cumplimiento que se tiene en algún proceso o resultado, respecto a criterios de evaluación, establecidos previamente, según la integración del modelo de calidad que se detalla de forma posterior.

Este conjunto de herramientas también permite el reporte automático de forma gráfica, según la selección de rúbrica evaluativa que se haya calificado. Esta

funcionalidad permitirá identificar datos y estadísticas importantes dentro del sistema, para todos los proyectos que han entrado al mismo.

Cuando se evalúa algún criterio de calificación, existe otra herramienta que notifica de forma automática (tanto por correo como en la misma plataforma) esa acción; por lo tanto, se podría hacer uso de esa integración y así resolver la problemática comunicativa no asertiva que presenta el TD entre las áreas involucradas del ciclo de vida de los proyectos.

Por último, se propone el uso de una herramienta, la cual permite copiar el modelo de evaluación y de la carpeta en línea de determinado curso o comunidad (sitios donde se ubican todas estas herramientas dentro de la plataforma del TD) y migrar a otro, lo cual haría posible la estandarización, sugerida por los entrevistados expertos en calidad, del modelo de gestión de artefactos y evaluaciones para todos los proyectos de la Unidad.

### **5.3.2. Herramienta externa del TEC Digital**

Para efectos de la ejecución de las actividades de la etapa de pruebas (totales de funcionalidad, de integración, de rendimiento, de seguridad, entre otras) en el sistema, se plantea la utilización de la herramienta de software *Tarantula*<sup>13</sup>; esta propuesta se realiza con base en el cuadro comparativo del apartado 4.3.2.

*Tarantula* presenta una serie de funcionalidades que la hacen aplicable a las condiciones del TD y satisfacen las necesidades propuestas para el sistema (mostradas en la Tabla 7 del apartado 4.3.2.).

A dicha herramienta, se le debe realizar una parametrización de los criterios requeridos para evaluar la calidad del software, según las métricas de calidad contenidas en la ISO/IEC 9126, especialmente, para la calidad externa del software, la cual es medida en la etapa de Pruebas.

---

<sup>13</sup> Para mayor información se puede consultar Testia (2017).

#### 5.4. Integración de modelo de calidad de software

Como mecanismo para medir y evaluar la calidad de los proyectos de software del TD, en una forma transversal, para todas sus etapas del ciclo de desarrollo, se propone la integración del modelo de calidad de software propuesto por la ISO (2001) en la norma ISO/IEC 9126-1 y las métricas de sus reportes técnicos correspondientes.

Esta propuesta se realiza de acuerdo con lo descrito (presente en el apartado 2.12. del capítulo dos) por Avella y Gómez (2011), Lochmann y Goeb (2011), St-Louis y Suryan (2012), Al-Qutaish y de igual forma, recomendado por el entrevistado dos (se puede consultar el apartado 4.1.2.).

Esta integración además, es posible plantearla, considerando la preparación (mencionada en el apartado 4.3.4. del capítulo anterior) de los artefactos inmersos en el sistema para el aseguramiento y control de la calidad en los proyectos de software de la Unidad. Seguidamente, mediante la Tabla 23, se realiza un mapeo específico de los tipos de calidad y su reporte técnico específico, expuestos en la ISO/IEC 9126-1 por la ISO (2001) y sus demás partes, que se plantean recabar y medir a través de cada una de las etapas propuestas en el sistema.

Tabla 23. Propuesta de integración del modelo de calidad de la ISO/IEC 9126 en el sistema

Tipo de calidad	Etapas del sistema
Calidad interna ISO/IEC/TR 9126-3	-Iniciación y planeación -Análisis -Diseño -Desarrollo
Calidad externa ISO/IEC/TR 9126-2	-Desarrollo -Pruebas -Capacitación
Calidad en uso ISO/IEC/TR 9126-4	-Capacitación -Cierre del proyecto -Post-liberación

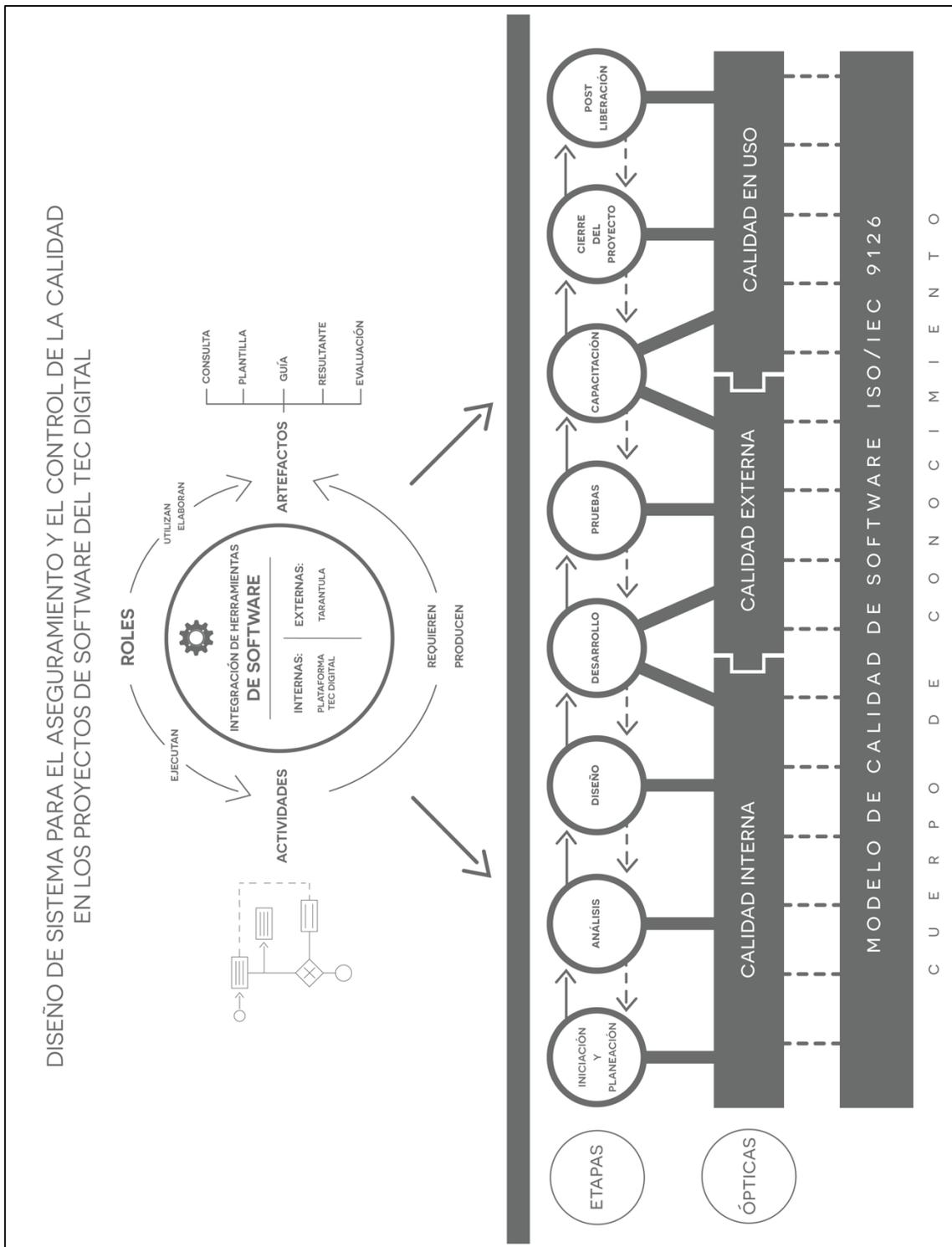
Fuente: Elaboración propia.

Cabe indicar que, según lo mostrado en el apartado 2.12. del presente trabajo, existe una relación directa entre los tipos de calidad propuestos en el modelo de calidad de software de la ISO/IEC 9126-1. Por este motivo, resulta imperante que dentro de las actividades y los artefactos propuestos del sistema, se realice un enfoque de calidad transversal durante la ejecución de cada una de las etapas del ciclo de vida de desarrollo de los proyectos de software del TD.

### **5.5. Diseño del sistema**

Seguidamente, se denota en la Figura 24, el diseño del sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD. Este diseño muestra, de manera integrada, todos los elementos que han sido desarrollados de forma anterior al presente apartado.

Figura 24. Sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital



Fuente: Elaboración propia.

En la Figura 24 se muestra los componentes integrados del sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD: 1) la base de conocimiento para la calidad de software, 2) las ópticas asociadas a este (contienen las métricas respectivas), 3) las etapas específicas del desarrollo de los proyectos de la Unidad donde serán integradas y 4) las interrelaciones que se dan entre los roles, actividades y artefactos, con el fin de que el sistema sea ejecutado en dicha entidad.

Con base en la ISO y el IEEE (2011), se podría afirmar entonces que, por medio del sistema propuesto en la Figura 24 y el planteamiento de todos sus elementos especificados anteriormente, se implementará el aseguramiento de la calidad del software como en el ambiente del TD, ya que, se elabora la planeación y la sistematización de las acciones necesarias para proveer un nivel de confianza respecto al cumplimiento de los requerimientos de un proceso o producto de software.

Precisamente, eso se busca a través del sistema propuesto, proveer una sistematización para aplicar las actividades dentro de cada proyecto de software del TD y para cada una de sus etapas, de modo que se realice la verificación, la validación y la evaluación de la calidad del software mediante los propuesto por la ISO y el IEEE (2011) en el modelo de calidad de software de la ISO/IEC 9126.

Este modelo que desarrolló la ISO y el IEEE (2011), representa el nivel del cuerpo de conocimiento que se muestra en la Figura 24, necesario para sentar la verificación, la validación y la evaluación mencionadas en el párrafo anterior, lo cual se realiza por medio de las ópticas de calidad presentes en los distintos reportes técnicos de dicho modelo: calidad interna, externa y de uso. Con la utilización de estas ópticas de calidad se logra ofrecer un enfoque integral de calidad, lo cual es un principio clave para implantar un sistema de calidad en una organización, según lo explicado por la ISO (2015) en la norma ISO 9000.

Además, de forma puntual, la introducción de estas ópticas de calidad, las cuales se describieron en el apartado 2.12. del capítulo dos y también, recomendadas tanto por Avella y Gómez (2011) así como por los dos entrevistados expertos en calidad de software, funciona como medio de respuesta a varias causas de la

problemática del apartado 4.2.4.2. que se detallan a continuación: verificación subjetiva y no validación, arbitrariedad de escalas y mediciones, desapego con base teórica y mejores prácticas.

En cuanto a lo descrito por Pressman (2010) en el apartado 2.10., por medio del sistema, se estarían generando los mecanismos de medición objetiva y reporte de sus resultados, a través de la integración transversal de la ISO/IEC 9126 y sus ópticas de calidades aunada a la integración de las herramientas de software propuestas en el apartado 5.3.

Por otra parte, según Pressman (2010), mediante el nivel de las etapas del ciclo de vida de desarrollo de software de la Unidad de la Figura 24, se está creando un proceso establecido con las tareas (actividades) y las herramientas (artefactos) específicas, las cuales permitirán la ejecución eficaz del aseguramiento y el control de la calidad a lo largo de las etapas de este ciclo. De esta forma, según el autor, se puede reafirmar que el proceso se está abordando realmente, al existir en el sistema entradas, con las cuales se ejecutan actividades específicas y, a su vez, estas producen salidas, las cuales se convertirán en insumos de las otras etapas, esto se denota en el apartado 5.2. anterior.

Al mismo tiempo, se está controlando los productos de software, según lo recomendado por Pressman (2010), que se utilizan y resultan a través del ciclo de desarrollo del software en dicha organización. Para esa finalidad en específico, se estableció un lineamiento de control de cambios, que se detalla más adelante en el apartado 5.6. Esta acción responde a la causa presentada en la problemática del 4.2.4.2. respecto al descontrol de cambios que existe actualmente, en el TD.

Además, en el sistema diseñado se propone una integración de todos los componentes descritos con herramientas de software (tanto internas como externa a la Unidad), las cuales soportarían las actividades planteadas dentro del sistema en el apartado 5.2. y reduce de esta forma, el “costo de la calidad” que mencionan Pressman (2010) y el entrevistado dos.

Cabe destacar que, el proceso sistematizado visualizado en la Figura 24, mediante las relaciones del eje principal y su presencia en todas las etapas del ciclo de desarrollo de los proyectos de software, genera una de las principales características del aseguramiento y control de la calidad de software, señalada por Pressman (2010) como “actividad sombrilla”. Esto significa que las actividades que se derivan del aseguramiento y el control de la calidad de software, se proponen ejecutar a lo largo del ciclo de vida de desarrollo de los proyectos de la Unidad, para cubrir de esta forma el total de etapas, esto actualmente, no se está realizando de acuerdo con lo reconocido en las causas del apartado 4.2.4. Por lo tanto, se estaría dando respuesta a la causa mostrada sobre una interacción tardía de gestión de la calidad.

Por otra parte, se puede sustentar que, efectivamente, uno de los productos del presente trabajo, detallado en la Figura 24, representa el diseño de un sistema para el aseguramiento y el control de la calidad de software, esto de acuerdo con la unificación de las definiciones dadas por Chiavenato (2007) para sistema y, por otro lado, Pressman (2010), Avella y Gómez (2011) y la ISO y el IEEE (2011) sobre aseguramiento y control de la calidad de software (estas definiciones se desarrollaron de forma amplia en párrafos anteriores).

Chiavenato (2007) expone que un sistema se establece como *“un conjunto de elementos dinámicamente relacionados en interacción (formando una red de comunicación cuyos elementos son interdependientes) que realizan una actividad, operación o proceso para alcanzar un objeto o propósito [...]”*. Esta definición se engrana dentro del sistema propuesto por medio de sus elementos constitutivos, el eje principal (roles, actividades y artefactos) y las propuestas de integración (tanto de las herramientas de software como de la base de conocimiento que presenta el modelo de calidad de software) para el mismo.

De esta forma, es posible afirmar que, el abordaje de solución constituye un sistema, debido a que muestra las características descritas por Chiavenato (2007); las mismas son detalladas de seguido:

- Es dinámico, porque sus elementos están en constante interacción por medio de los diferentes proyectos de software que ejecuta el TD de forma concurrente y en sus múltiples etapas propuestas.
- Es interdependiente; pues cada uno de sus componentes por sí mismos no generan un aporte a la Unidad, se necesita la integración de todas sus partes, de tal forma que se genere una sinergia en el desarrollo del sistema; con el fin de lograr así, que la suma de estas piezas impacte de mayor forma, que cada una de estas partes de forma separada.
- Apunta a un objetivo: asegurar y controlar la calidad en los proyectos de software del TD, por medio de la definición de las interacciones, la integración de herramientas de software y la inclusión del modelo de calidad de software para verificar, validar y evaluar dicha calidad.

## **5.6. Lineamientos y características del sistema**

Cabe indicar seis propuestas de lineamientos y características mínimas, que se realizan para el sistema planteado, por medio de sus elementos y en el modo de ejecución del mismo.

Esta propuesta se realiza como medida de seguimiento a lo expuesto por Pressman (2010) en el apartado 2.10., donde se define que se debe contar con un proceso establecido de aseguramiento de la calidad del software, para buscar la excelente calidad en los proyectos de este tipo.

Además, se plantea como respuesta a la causa de problemática identificada en los apartados 4.2.4.2. y 4.2.4.3., donde se describe por ejemplo, que para la categoría de artefactos existe un descontrol de los cambios que se realizan sobre estos.

A continuación, se especifican los lineamientos que se acompañan la propuesta:

- Cuando se requiera realizar algún cambio sobre un artefacto resultante ya evaluado y aprobado, primeramente se deberá elaborar una solicitud de cambio que contenga una fundamentación del cambio. Así también, deberá existir un

registro (puede ser una minuta de reunión), donde se evidencie la aprobación por parte de los roles involucrados. Un ejemplo de este accionar se presentaría si se requiere un cambio en el diseño de la interfaz gráfica de algún proyecto de software de la Unidad, de esta forma, se debe elaborar una solicitud de cambio por parte de los colaboradores que proponen el mismo y aprobarse por el rol o roles de CV determinados para ese proyecto en específico.

- Se plantea el uso de minutas (el formato queda a definir por parte del TD) cuando se desarrollen reuniones y ejecuciones de actividades de forma colaborativa entre distintos roles del sistema. Dichas minutas de reunión, deberán almacenarse en el mismo lugar donde se encuentran los artefactos relacionados a cada etapa del ciclo de vida de desarrollo de los proyectos de software de la Unidad.
- El sistema deberá aplicar para todos los proyectos de software, sin excepciones; con el fin de crear constancia y repetibilidad (clave según los expertos entrevistados) en el aseguramiento y el control de la calidad de software. Por esta razón, es un sistema y no una metodología; pues no tendrá un inicio o un fin, se realizará de forma constante a través de todos los proyectos de software de la Unidad.
- El modo de gestión, almacenamiento y aprobación de los artefactos resultantes puede ser formal impreso (con firma, escaneado y almacenado en un hospedaje en línea centralizado), formal mediante formato digital (aprobado por el encargado respectivo mediante firma digital y almacenado en un solo lugar) o bien, comunicado mediante correo electrónico y con la respuesta de los roles encargados de aprobación al correo inicial.
- Se excluye del planteamiento de procedimiento, la definición de los procedimientos específicos para cada una de las actividades propuestas de aseguramiento y control de la calidad.

- Se propone agregar una etapa (mostrada en las Tabla 22) dentro de la metodología del ciclo de vida de desarrollo de los proyectos del TD, la cual se le nombró “post-liberación”. La misma busca la retroalimentación de los usuarios finales del software elaborado en esta organización, para propiciar de esta forma, la medición y la evaluación de la calidad de los resultados destacada por la ISO y el IEEE (2004) como calidad en uso, la cual representa el nivel de conformidad de los usuarios finales respecto a los productos de software de un proyecto de este tipo.

### **5.7. Plan de capacitación sobre calidad de software**

Se propone la elaboración de un plan de capacitación sobre calidad de software y con el objetivo también de culturizar al TD en el sistema propuesto para esta organización. Esta iniciativa de propuesta de capacitación surge, a raíz de lo formulado por la ISO (2015) en su norma ISO 9000, donde se explica que, para conseguir un enfoque de calidad dentro de una organización, se recomienda tomar en cuenta siete principios (descritos en el apartado 2.9.), dentro de los cuales cabe señalar: enfoque en el cliente, liderazgo, compromiso de los colaboradores, acercamiento al proceso y la gestión de las relaciones (en términos de comunicación).

Enfáticamente, este plan de capacitación va dirigido hacia el equipo de desarrollo de software del TD y para los colaboradores de otras áreas que desempeñan los roles específicos, detallados en el apartado 5.1.1.

El propósito de este plan de capacitación es dotar a los colaboradores involucrados en los proyectos de desarrollo de software del TD, de las herramientas de conocimiento y enfoque de calidad, para que desempeñen sus funciones y actividades de acuerdo con este énfasis, especialmente, las que se incluyen en las interacciones (se puede consultar el apartado 5.2.) del sistema propuesto.

Los temas por sesión a desarrollar en este plan de capacitación serían:

1. Primera sesión:
  - a. Calidad (enfoque general)

- i. Definición.
    - ii. Principios de calidad de la ISO 9000, de acuerdo con ISO (2015).
      - 1. Definiciones.
      - 2. Beneficios de cada principio.
  - b. Calidad de software
    - i. Definición.
    - ii. Beneficios.
    - iii. Aseguramiento de la calidad del software.
      - 1. Definición.
      - 2. Actividades.
      - 3. Resultados.
    - iv. Control de la calidad del software.
      - 1. Definición.
      - 2. Actividades.
      - 3. Resultados.
2. Segunda sesión:
- a. Situación actual de calidad de software de la Unidad.
  - b. Sistema para el aseguramiento y control de la calidad en los proyectos de software del TD:
    - i. Roles.
    - ii. Actividades.
    - iii. Artefactos.

Este plan de capacitación, responde a la problemática presente en el TD (resumida en los apartados 4.2.4.2. y 4.2.4.3.), donde se identificó existe una definición confusa de responsabilidades, una comunicación no asertiva en las áreas involucradas en la construcción del software (descritos en el sistema como roles) y una falta de formación en calidad de software.

## CAPÍTULO VI: CONCLUSIONES

A través del desarrollo del presente trabajo final de graduación y con base en el cumplimiento de los objetivos propuestos, se desarrollaron las siguientes conclusiones:

- **Objetivo específico uno:** examinar las mejores prácticas, normas, marcos de referencia y metodologías de aseguramiento y control de la calidad de software, para la adquisición de una base teórica de gestión de la calidad en proyectos de desarrollo de software.
  - Se documentó una base teórica sobre calidad de software, aseguramiento y control de la misma. Esta se compuso principalmente, de los siguientes aportes:
    - ISO (2001)
    - Pressman (2010).
    - ISO y el IEEE (2011).
    - ISO (2015).
    - Avella y Gómez (2011).
  - La consulta realizada a los expertos profesionales en calidad de software, se caracterizó como una técnica efectiva para determinar el nivel de pertinencia de la base teórica adquirida.
  - Por lo tanto, el estudio y formulación de la base teórica, fue clave para el análisis de los resultados de la metodología y principalmente, para la construcción de la propuesta solución mediante el diseño del sistema para el aseguramiento y el control de la calidad en los proyectos de software de la Unidad.
- **Objetivo específico dos:** analizar la brecha existente entre la base teórica adquirida de gestión de la calidad respecto a la ejecución de los procesos de

aseguramiento y control de la calidad en los proyectos de desarrollo de software del TD.

- Al incluir la actividad de pruebas de forma más temprana en el ciclo de desarrollo de los proyectos de la Unidad, se aumentó en promedio un 21.67% la cantidad de hallazgos por número de tareas revisadas.
- En el aumento de hallazgos que se presentó en las mediciones del proceso de control de la calidad, se mostró un aumento proporcional de hallazgos tipo error de un 46%. No se puede determinar la incidencia en la disminución de errores una vez que los proyectos son liberados, sin embargo, se establece como base para futuras mediciones.
- El 90% de los colaboradores entrevistados del TD considera que existe una situación negativa en cuanto a la calidad de software presente en los proyectos que desarrolla esta entidad.
- El TD no realiza actualmente, procesos o actividades de aseguramiento de la calidad en sus proyectos de software, lo que realiza es un esfuerzo puntual del control de calidad de software, mediante la ejecución de la actividad de pruebas dentro de su metodología de desarrollo de software.
- El TD no presenta una base objetiva y medible para la verificación, la validación y la evaluación de la calidad de los proyectos de software que se desarrollan en la Unidad.
- El TD no cuenta con el resumen de datos, información o estadísticas que permitan reconocer el estado actual de la calidad de los proyectos de software que han sido desarrollados o están en ese proceso de ejecución. Los datos que se utilizaron para el análisis cuantitativo fueron extraídos y tratados estadísticamente, en el proyecto.
- En la metodología de desarrollo de los proyectos de software del TD, no se ejecutan procesos o actividades para medir el nivel de calidad

percibido por parte de los usuarios finales de dichos proyectos, respecto a las características y funcionalidades de los productos de los mismos.

- **Objetivo específico tres:** determinar el alcance de solución aplicable a la situación que se presenta actualmente, en la gestión de la calidad de los proyectos de software del TD.
  - Los objetivos específicos uno y dos permitieron determinar el alcance de la solución, el cual consta de los siguientes componentes:
    - Elementos del sistema (roles, actividades y artefactos).
    - Interacciones entre sus elementos (divididas por etapas de la metodología de los proyectos de software de la Unidad).
    - Integraciones al sistema (de herramientas de software y de un modelo de calidad de software).
    - Lineamientos y características del sistema.
    - Plan de implementación del sistema diseñado, el cual se divide en cinco etapas.
- **Objetivo general:** diseñar un sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD.
  - La propuesta de solución elaborada representa el diseño de un sistema, ya que, se compone de elementos que se integran entre sí de forma dinámica, son interdependientes y tienen los objetivos de asegurar y controlar la calidad de los proyectos de software del TD.
  - El sistema diseñado propicia el aseguramiento y control de la calidad de los proyectos de software de la Unidad al proponer interacciones y uso de sus integraciones a través de todas las etapas de la metodología de desarrollo del TD, lo que representa una ruta sistematizada y planificada de esfuerzos explícitos para asegurar y controlar esa calidad.

- El sistema brinda una base objetiva y medible para la verificación, la validación y la evaluación de la calidad de los proyectos de software que se desarrollan en la Unidad, mediante la integración del modelo de calidad de software propuesto por la ISO (2011) en la ISO/IEC 9126.
- El sistema permite la automatización de sus actividades por medio de la integración de las herramientas de software que son señaladas en la propuesta de solución. Además, esta integración permitirá dotar a dicha entidad, de datos, información y estadísticas relacionadas el estado general de calidad de los proyectos, propiciando de esta forma, una efectiva toma de decisiones.
- El sistema cubre ampliamente, el concepto de calidad en los proyectos de software del TD desde las tres ópticas de la ISO/IEC 9126 a lo largo de las etapas de la metodología de desarrollo de la Unidad.
- Se incluye la etapa de “post-liberación” dentro de la metodología de desarrollo de los proyectos de software del TD, la cual, actualmente, no está presente en dicha metodología; esta etapa, permitirá medir el nivel de calidad percibido por los usuarios finales, una vez que se liberan los productos de software para la comunidad TEC.

## CAPÍTULO VII: RECOMENDACIONES

Con el desarrollo del presente trabajo final de graduación y como enlace con las conclusiones elaboradas anteriormente, se proponen las siguientes recomendaciones:

- **Objetivo específico uno:** examinar las mejores prácticas, normas, marcos de referencia y metodologías de aseguramiento y control de la calidad de software, para la adquisición de una base teórica de gestión de la calidad en proyectos de desarrollo de software.
  - La base teórica adquirida debe ser transmitida y compartida con todos los colaboradores de la Unidad. De modo que, en un futuro, al proponer trabajos o esfuerzos relacionados a calidad de software en la organización, se consulte de forma inicial dicha base.
  - Para nuevos trabajos investigativos en el TD, se debe consultar con profesionales expertos en el área o áreas que se vaya a desarrollar, de tal forma que orienten en el proceso de consulta y recopilación de la teoría necesaria para la realización de dichos trabajos.
  - Realizar un plan de aseguramiento de la calidad que integre el criterio técnico obtenido, el cual sea resumido, puntual, y dictamine los procesos y actividades específicas para gestionar la calidad de los proyectos de software de la Unidad.
- **Objetivo específico dos:** analizar la brecha existente entre la base teórica adquirida de gestión de la calidad respecto a la ejecución de los procesos de aseguramiento y control de la calidad en los proyectos de desarrollo de software del TD.
  - Presentar el análisis y síntesis de la problemática presentes en trabajo final a todos los colaboradores del TD para que se conozca a nivel organización las causas específicas del problema.

- Documentar los procedimientos y los procesos específicos para cada una de las actividades propuestas en el sistema, mediante la modelación de procesos de negocio con BPMN 2.0.
  - Definir las métricas precisas para cada actividad de verificación, validación y evaluación de la calidad en los proyectos de software de la Unidad.
  - Hacer uso de las herramientas propuestas de la plataforma del TD y de *Tarantula* para generar los datos, la información y las estadísticas que permitan reconocer el estado de calidad de los proyectos de software de la Unidad.
  - Emplear la medición de la óptica de calidad en uso (presente en el modelo de calidad de software de la ISO/IEC 9126), para los productos de software de los proyectos de la Unidad. Esta medición, sentará la correcta definición del nivel de calidad alcanzado en los proyectos de software del TD.
- **Objetivo específico tres:** determinar el alcance de solución aplicable a la situación que se presenta actualmente, en la gestión de la calidad de los proyectos de software del TD.
    - Abordar el plan de implementación del diseño del sistema de forma gradual e iterativa; es decir, realizar cada las etapas del plan, progresivamente, para cada una de las etapas definidas del propio sistema.
    - Capacitar a los colaboradores de la Unidad en el rol que desempeñan cada uno en las etapas del sistema, dentro de la metodología de desarrollo de los proyectos de software de esta entidad.
  - **Objetivo general:** diseñar un sistema para el aseguramiento y el control de la calidad en los proyectos de software del TD.

- Implementar cada uno de los elementos propuestos para el sistema, para el aseguramiento y el control de la calidad en los proyectos de software de la Unidad. Primero, desarrollar el plan de implementación, segundo, realizar la integración del modelo de calidad de la ISO/IEC 9126 y tercero, integrar las herramientas de software.
- Actualizar los manuales de procedimientos que existen en el TD, de forma tal, que se incluya la ruta sistematizada y planificada de las actividades propuestas en el sistema dentro de cada uno de estos.
- Hacer un análisis de las métricas que puedan ser implementadas en las etapas del sistema y realizar iteraciones de dicha integración tanto en las actividades y artefactos, como en las herramientas de software planteadas.
- Realizar un análisis bimensual en la Unidad, de los resultados obtenidos con la producción de los datos, la información y las estadísticas mediante la implementación de las herramientas de software que soportarán las actividades del sistema.
- Utilizar el enfoque de las ópticas de calidad para el desarrollo de más esfuerzos puntuales que busquen el cumplimiento de los requerimientos y de las necesidades de los usuarios finales que presentan los productos de software elaborados en el TD.
- Introducir, de forma inmediata, la etapa de “post-liberación” en la metodología de desarrollo de los proyectos de software de la Unidad.
- Certificar al TD en la norma ISO 9001 con el ente costarricense respectivo, sobre la gestión de la calidad en la elaboración de los proyectos de software, cuando el sistema sea implementado en su totalidad.

## GLOSARIO

### Acrónimos

ACM	<i>Association for Computing Machinery.</i>
GC	Equipo de Gestión de la Calidad.
HASTQB	<i>Hispanic America Testing Quality Board.</i>
IEEE	<i>Institute of Electrical and Electronics Engineers.</i>
ISO	<i>International Organization for Standardization.</i>
LRN™	<i>Learn, Research, Network.™</i>
SWEBOK	<i>The Guide to the Software Engineering Body of Knowledge.</i>
TD	TEC Digital.
TEC	Instituto Tecnológico de Costa Rica.
TIC	Tecnologías de información y comunicación.

### Abreviaturas

Comunidad TEC	Comunidad del Instituto Tecnológico de Costa Rica conformada por estudiantes, profesores y demás funcionarios.
Procesos de calidad	Procesos correspondientes al aseguramiento y control de la calidad de software del TEC Digital.

## ANEXOS

### Anexo A. Plantilla de Control de Calidad

La siguiente imagen muestra la pestaña de generalidades de la Plantilla de Control de Calidad que utiliza el TEC Digital para la ejecución de su proceso de control de la calidad de software.

The image shows a spreadsheet titled 'QA-Plantilla' with a menu bar (Archivo, Editar, Ver, Insertar, Formato, Datos, Herramientas, Complementos, Ayuda) and a toolbar. The spreadsheet content is as follows:

tecDigital				
<b>Generalidades del proyecto</b>				
3	Título del proyecto:			
4	Propósito del proyecto:			
5	Público meta del proyecto:			
6	Coordinador (es):			
7	Equipo desarrollador:			
8	Revisado por (nombre/nickname): Michael Sánchez Soto (MichSS)			
9				
10	<b>Requisito/Característica de proyecto</b>	<b>Cumplimiento de requisito/característica</b>		<b>Observaciones</b>
11		Sí	No	
12	Todos los integrantes del proyecto entienden los objetivos del proyecto debe realizar.			
13	Se cuenta con la información suficiente.			
14	Requiere investigación.			
15	Tiene Paper.			
16				
17	<b>Estado del proyecto</b>			
18	<b>Criterio</b>	<b>Estado actual</b>		
19	Portal			
20	Calidad Paquetes			
21	Interfaz			
22	Seguridad y rendimiento			
23	Integridad			
24	Compatibilidad-Integración			
25	Usabilidad			
26	Plan de Capacitación			
27	<b>NOTA:</b>			

At the bottom of the spreadsheet, there is a navigation bar with the following tabs: Generalidades, Bitácora, Plan de Pruebas, Hallazgos, Portal TD, Compatibilidad, Calidad Paquetes, Seguridad/Rendimiento, Integridad, and Interfaz/Navegación.

QA-Plantilla

Archivo Editar Ver Insertar Formato Datos Herramientas Complementos Ayuda

Solo comentarios

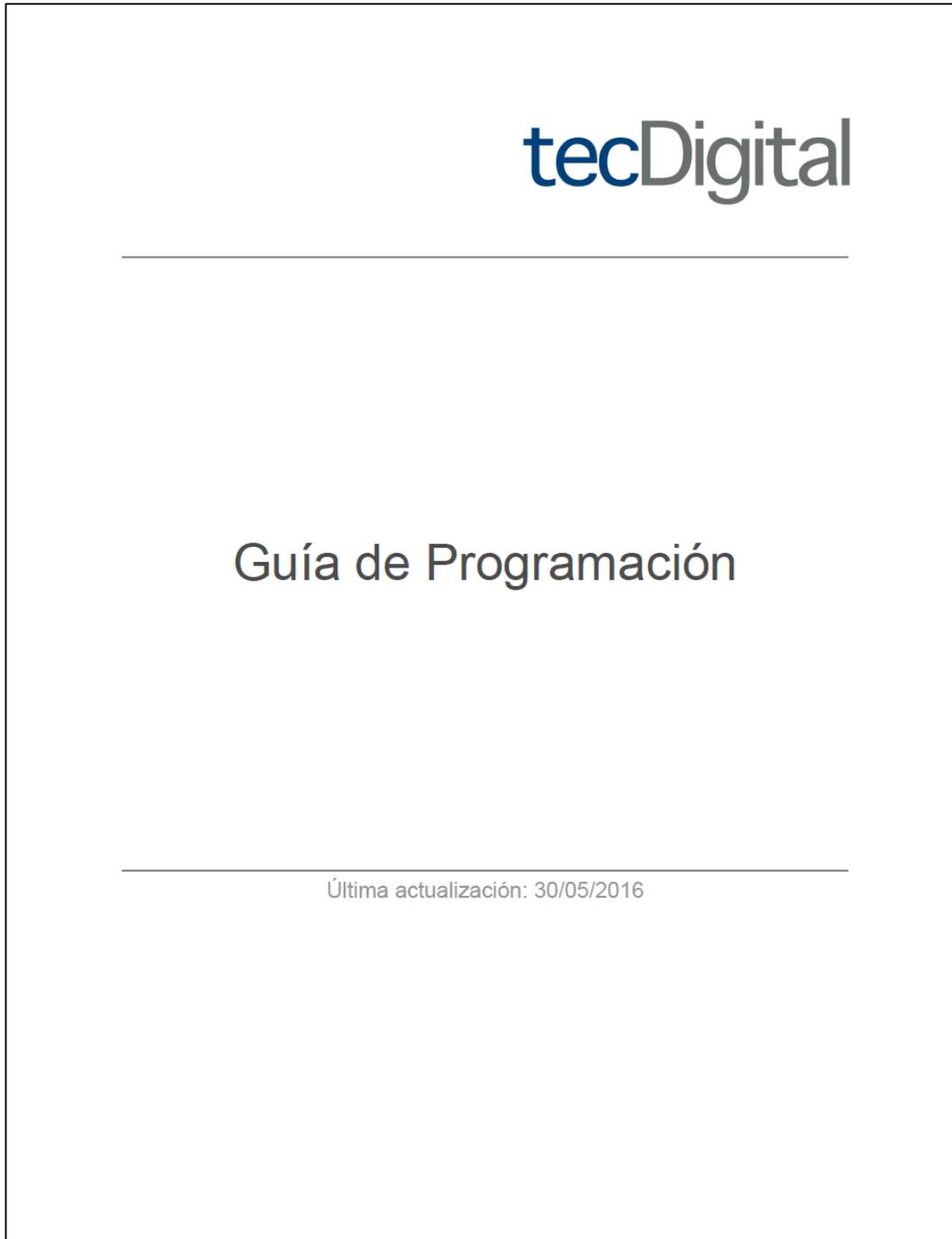
=Generalidades!B3

	A	B	C	D	E	F	G	H	I
1	tecDigital	Código y revisión de paquetes de proyecto:							
2									
3	Nombre del paquete: tda-ga								
4	Definición			Evaluación		Observaciones			
5				Lo aplica	No lo aplica				
6	<b>Generales</b>								
7	La distribución de las carpetas debe estar según el estándar.								
8	Existe el archivo de especificación .info, y debe indicar de que paquetes depende, callbacks, versión								
9	Tiene lenguajes/aplicaciones necesarias que sean distintos a tcl-adp-js-css (Indicar versiones) esto sólo puede ser revisado si es autorizado.								
10	Debe tener un nombre significativo (tda/tds + nombre)								
11	No se debe hacer copia de archivos sin cambiar el nombre a un nombre significativo de este paquete.								
12	Debe existir la documentación en el código (archivos de la carpeta entradas salidas descripción, fecha de creación, fecha de modificación)								
13	Debe haber un responsable de la elaboración del paquete con la fecha del release/update *								
14	Se debe considerar la internacionalización en el paquete (Existe carpeta catalog en la raíz)								
15	No tiene puts ni alerts dentro del código de la aplicación.								
16	<b>Capa de Base de Datos (archivos .sql y xql)</b>								
17	Se debe usar un esquema propio para el paquete, en inglés. Según estándar ( <a href="https://drive.google.com/file/d/0B14FVIBunmGYQ0ZyUm9DQ0hpN2s/view?usp=sharing">https://drive.google.com/file/d/0B14FVIBunmGYQ0ZyUm9DQ0hpN2s/view?usp=sharing</a> ).								
18	Existen archivos para base de datos postgresql. Según estándar.								

Generalidades Bitácora Plan de Pruebas Hallazgos Portal TD Compatibilidad Calidad Paquetes Seguridad/Rendimiento Integridad

## **Anexo B. Guía de Programación**

El siguiente documento presenta la Guía de Programación que utilizan los equipos desarrolladores de software del TEC Digital para elaborar el código fuente de sus proyectos.



tD

---

## Control de cambios

### Documento original

Versión 1.0	Fecha desconocida	Jenny Manrique
Descripción:	Elaborar una guía para la programación en el TEC Digital.	

### Cambios realizados

Versión 1.1	12/03/2015	Fabricio Soto Mejías
Descripción:	Revisión y actualización del documento.	

Versión 2.0	28/03/2016	Isaac Alpízar Chacón
Descripción:	Revisión y actualización del documento. Se agregan secciones de Internacionalización, Interfaz gráfica y Versionamiento. Se modificó la sección 5 para eliminar los tipos de datos en el nombre de la variable.	

Versión 2.1	30/05/2016	Michael Sánchez Soto
Descripción:	Revisión y actualización del documento de acuerdo al estándar de documentos del TEC Digital (formato).	

tD

---

## Tabla de contenido

- [1. Objetivo](#)
- [2. Estructura de Paquetes](#)
  - [2.1 Carpeta SQL](#)
  - [2.2 Carpeta TCL](#)
  - [2.3 Carpeta www](#)
  - [2.4 Archivo .info](#)
- [3. Idioma de Programación](#)
- [4. Nombres de Archivos](#)
- [5. Documentación Interna de Archivos](#)
  - [5.1 Encabezado](#)
  - [5.2 Namespaces](#)
  - [5.3 Ejecución de un Procedimiento](#)
- [6. Variables](#)
- [7. Tablas de la Base de Datos](#)
- [8. Nombres de Constraints](#)
- [9. Programación de interfaz gráfica](#)
- [10. Internacionalización](#)
- [11. Versionamiento](#)

tD

---

## 1. Objetivo

Presentar una guía de programación a seguir por los desarrolladores encargados de cualquier proyecto del TEC Digital.

Su objetivo es estandarizar el contenido de los paquetes de las diversas aplicaciones, así como nombres de variables, procedimientos y demás aspectos importantes a considerar durante la etapa de desarrollo, con el fin de facilitar la lectura del código y agilizar su comprensión.

## 2. Estructura de Paquetes

Con el fin de estandarizar la estructura de paquetes, se detalla las buenas prácticas para realizar la conformación del mismo. De manera general, en el nombre del paquete debe incluir el prefijo “tda”, si hace referencia a un paquete de aplicación, ó “tds”, si este corresponde a un servicio, el cual será acompañado por un “-”, seguido por el nombre del paquete; a continuación se detalla un ejemplo:

Si el paquete es de tipo aplicación y posee como nombre “registration”, el resultado será: “tda-re”.

Además, dentro del paquete se deben incluir 4 carpetas básicas y un archivo: *sql*, *tcl*, *www*, *catalog* y el archivo *.info*.

Conforme avance el documento, se detalla cada una de las carpetas. En la siguiente página se detalla la estructura de un paquete y sus elementos.

## tD

A continuación se muestra la estructura general de un paquete, y se detallan sus componentes:

```
+ tda-paquete {1}
|
| +-- tda-paquete.info {2}
|
| +-- lib/ {3}
| | +-- *.tcl
| | +-- *.adp
|
| +-- sql/
| | +-- postgres/ {4}
| | | +-- tda-paquete-create.sql {5}
| | | +-- tda-paquete-drop.sql {6}
| | | +-- upgrade/ {7}
| | | | +-- upgrade-0.1d-2.1d.sql
| |
| | +-- oracle/ {8}
| | | +-- tda-paquete-create.sql
| | | +-- tda-paquete-drop.sql
| | | +-- upgrade/
| | | | +-- upgrade-0.1d-2.1d.sql
|
| +-- tcl/ {9}
| | +-- tda-paquete-procs.tcl {10}
| | +-- tda-paquete-procs.xql {11}
| | +-- tda-paquete-init.tcl {12}
| | +-- tda-paquete-init.xql {13}
| | +-- ...
|
| +-- www/ {14}
| | +-- admin/ {15}
| | | +-- index.tcl
| | | +-- index.adp
| | | +-- ...
|
| | +-- doc/ {16}
| | | +-- docs/ {17}
| | | | +-- *.odt
| | | | +-- bitacora_cambios.txt {18}
| | | +-- index.tcl
| | | +-- index.adp
|
| | +-- resources/ {19}
| | | +-- js/ {20}
| | | | +-- lib/ {21}
| | | | | +-- *.js
| | | +-- css/ {22}
| | | | +-- *.css
| | | +-- images/ {23}
| | | | +-- *.*
|
| +-- ajax/ {24}
| | +-- *.tcl
| | +-- *.adp
```

En la siguiente página se describe cada uno de estos componentes detallados.

tD

Elemento	Descripción
1	Nombre del paquete, según el estándar, tda si es aplicación, tds si es servicio. (tec digital aplicación/servicio)
2	Archivo de especificación, debe indicarse de qué paquetes dependen, sus callbacks (en la mayoría de los casos no se edita)
3	Incluye los templates, utilizado cuando se hacen portlets.
4	Archivos para una base de datos en postgres.
5	Archivo de creación en base de datos de todo lo relacionado con esto (tablas, funciones, restricciones...). Útil para instalar un paquete.
6	Archivo que elimina todo lo que se crea en el archivo {5}. Útil para desinstalar un paquete.
7	Carpeta para hacer actualizaciones en la base de datos. Útil para cambios de versión.
8	Igual que el {4} pero en oracle.
9	Carpeta de la lógica del paquete, además es la única parte donde se puede acceder a la base de datos.
10	Los "-procs" es donde se definen los procedimientos del paquete.
11	Son los queries de los procedimientos del archivo con el mismo nombre.
12	{sin fuente que lo confirme} Los "-init" son archivos que están en el ambiente y proporcionan un nivel más bajo, a diferencia de los "-procs", estos siempre se ejecutan al iniciar el servidor.
13	Son los queries de los procedimientos del archivo con el mismo nombre.
14	Es la carpeta de presentación, va todo lo relacionado con interfaz gráfica, no debe existir lógica del paquete.
15	Carpeta que contiene las funciones del administrador del paquete, sólo el que sea definido como administrador desde la plataforma podrá ver su contenido.

tD

16	Carpeta de documentación, debe contener un index que muestre links a su contenido.
17	Contiene todos los archivos de la metodología de desarrollo.
18	Bitácora con todo cambio realizado al paquete, una vez puesto en producción.
19	Carpeta que contiene los recursos del paquete.
20	Carpeta que contiene todos los javascript, jquerys o relacionados.
21	Carpeta que contiene librerías de javascript, jquery o relacionados.
22	Carpeta que contiene todos los css del paquete.
23	Contiene todas las imágenes utilizadas en el paquete.
24	Contiene todos los tcl y adp que se utilizan en llamadas asincrónicas, estos archivos no contienen ninguna lógica, únicamente llaman a un procedimiento en {9} y se ajustan según su presentación (agregarle tags html, agregarles algún atributo, unir datos recogidos en otros procedimientos, entre otros.)

La estructura y notación de las carpetas y archivo del paquete se detallan a continuación:

## 2.1 Carpeta SQL

Incluye dos posibles carpetas:

- Postgresql: Incluye el archivo de creación y borrado (create y drop) con extensión .sql. Se debe desarrollar un "SCHEMA", de la siguiente manera:
  - **tda +\_+ package\_name**
  - Package\_name: Corresponde al nombre de la aplicación escrito en minúscula, el cual debe de encontrarse escrito en el idioma inglés.

Dentro del esquema se crearán las tablas correspondientes, en las cuales el nombre queda a criterio del desarrollador.

El nombre de los archivos de creación y borrado deben de seguir el siguiente estándar:

*[package]-create.sql*  
*[package]-drop.sql*

tD

## 2.2 Carpeta TCL

Incluye dos archivos '-procs', contiene los procedimientos almacenados y funciones que se utilizarán por el paquete. Sus archivos son:

- [package]-procs.xml: Debe identificar la versión. Ejemplo de su estructura:

```
<?xml version="1.0"?>
<queryset>
  <fullquery
    name="namespace::procedimiento1.nombreQuery">
<querytext>
      Código del query deseado...
    </querytext>
  </fullquery>
</queryset>
```

- [package]-procs.tcl: Incluye la fecha de creación y autor(es) responsables dentro del método `ad_library`, adicional incluye la ruta dentro del directorio. El método `ad_library`, se usa luego del comentario de ruta de archivo, y emplea cadenas de documentación similar al método `ad_page_contract`, incluyendo una descripción de la funcionalidad y líneas descriptivas de los elementos principales. Ej:

```
#packages/tda-prematricula/tcl/tda-prematricula-procs.tcl
ad_library {
  [Resumen general del propósito del documento]
  @author [Nombre del programador (correo@tec.ac.cr)]
  @creation-date [13/06/2014]
  @cvs-id $Id
}
```

Donde lo que está entre “[ ]” es lo que se modifica según cada archivo de procedimientos.

Los procedimientos deben ser declarados usando `ad_proc` y deben estar dentro del `namespace` del paquete (namespace debe ser igual al nombre del paquete), los namespace se emplean para agrupar procedimientos de un mismo propósito. Los parámetros deben ser nombrados todos en minúscula y si el nombre contiene más de una palabra se deben separar por medio de underscore (“\_”), en los parámetros se debe especificar cuáles son requeridos, opcionales (incluye default) y booleanos, así como cuando se envían datos según el nombre

tD

---

de la variable (se declara con “ - “, por ejemplo si declaro el nombre con “ - “, sería -nombre Agustín -apellido Francesa). Ejemplo de declaración del procedimiento y tipos de parámetros:

```
namespace eval name {}
ad_proc -public name::procedure_name {
    id:required
    name:required
    apellido:required
    activo:boolean
}
Usar para documentar (Ej. Process a web service that shows ...)
@param
}
...proc body
}
```

## 2.3 Carpeta www

Son los primeros archivos que se cargan al entrar en el paquete, es la raíz que se toma al acceder a los diferentes archivos. Contiene una carpeta de “Recursos” dividida en una carpeta para archivos .css, otra para .js y otra de imágenes de la aplicación. Fuera de la carpeta de “Resources” se debe incluir un archivo “index.adp” e “index.tcl”, por cada uno de estos archivos .adp y .tcl debe existir un archivo .css y otro .js con el mismo nombre (estos últimos ubicados en sus respectivas carpetas dentro de la carpeta “Resources”).

- Archivo tcl: Se codifica posterior al encabezado del archivo. En el primer argumento se crean las cadenas de documentación del método, iniciando con una descripción general de la funcionalidad y posteriormente líneas descriptivas de los elementos principales. Ej:

```
ad_page_contract {
    Purpose/Reference: XXX
    @author: FirstNameLastName (correo@itcr.ac.cr)
    @creation_date: 20/09/2013
    @modification_date: 06/05/2014, por: NameLastaname
}
```

## tD

- Archivo adp: Comentar el código, e incluir nombre del autor, fechas de creación y modificación (control de cambios). Ej:

```
<comment>
    Author: NameLastname
    creation Date: 20/01/2014
    Modification Date: 05/05/2014, por: NameLastName
</comment>
```

### 2.4 Archivo .info

Este es el archivo de especificación y en él se debe incluir el nombre del autor o autores responsables de la aplicación. También se debe incluir la versión que corresponde. En este archivo se incluyen callbacks, packages y provides. El archivo se crea automáticamente si se genera el paquete desde la plataforma. Ejemplo de su estructura:

```
<?xml version="1.0"?>
<version name="0.1d"
url="http://openacs.org/repository/download/apm/tda-XXX-0.1d.apm">
    <owner url="mailto:correo@tec.ac.cr">NombreApellido</owner>
    <summary>XXX</summary>
    <description format="text/html">XXX</description>
    <release-date>2013-11-25</release-date>
    <update-date>2013-11-25</release-date>
    <maturity>0</maturity>
    <provides url="tda-XXX" version="0.1d"/>
    <requires url="tda-XXX" version="0.1d"/>
    <callbacks>
    </callbacks>
    <parameters>
    <!-- No version parameters -->
    </parameters>
</version>
```

## 3. Idioma de Programación

Para los nombres de variables, procedimientos, nombres de archivos, paquetes, y en general, se utilizará el idioma inglés.

tD

---

Se debe de incluir como requisito mínimo dentro de las características de internacionalización el idioma **inglés y español** para el paquete.

## 4. Nombres de Archivos

Los archivos se nombran en minúscula e inglés y las palabras se separan con un guión ("-"). Para nombrar los archivos que ejecutan una acción específica sobre un objeto, se emplea el siguiente formato:

*drop-record.tcl*

Se deben utilizar nombres **significativos** y no muy extensos (no más de 30 caracteres) para nombrar los archivos a emplear.

## 5. Documentación Interna de Archivos

### 5.1 Encabezado

Todos los scripts o archivos deben incluir encabezado. La primera línea debe ser un comentario especificando la ruta del archivo relativa al directorio raíz de dotlrn, por ejemplo:

*#!/www/index.tcl*

ó

*#!/tcl/module-defs.tcl*

Seguidamente se debe de especificar:

-Nombre del paquete al que pertenece: Package Name

-Breve descripción de su objetivo/funcionalidad: Description

-Fecha de creación: Creation Date

-Observaciones (Que faciliten el entendimiento de cualquier otro desarrollador):

Notes/comments

Estos puntos son de suma importancia, ya que cumple con las características de un buen diseño.

**Además del encabezado, cualquier procedimiento/función/método que sea desarrollada dentro del paquete que utilice javascript, php o cualquier otro lenguaje, tiene que encontrarse comentada (documentación interna), como cualquier elemento que así lo requiera, utilizando el estándar de documentación correspondiente (Creation Date – Short description - Comments).**

tD

---

## 5.2 Namespaces

Para los nuevos paquetes se debe definir un *namespace* e incluir todos los procedimientos dentro de éste.

El *namespace* debe ser igual al nombre del paquete sin el prefijo “tda” o “tds”, pero con underscore (“\_”) para separar las palabras.

Ejemplo de su declaración:

```
namespace eval name {}
```

## 5.3 Ejecución de un Procedimiento

Para ejecutar un procedimiento se realiza lo siguiente:

```
nameNamespace::ProcedureName parameters
```

Ejemplo:

```
math::adittion 5 10 ----> 15
```

Si el procedimiento retorna algún valor, se deben utilizar los “[ ]”. Ej:

```
set result [nameNamespace::ProcedureName parameters]
```

## 6. Variables

Las variables deben ser nombradas todas en minúscula, y si el nombre contiene más de una palabra se deben separar por medio de underscore (“\_”). Ej: *user\_id*

## 7. Tablas de la Base de Datos

El nombre de las tablas debe definirse tomando en cuenta las siguientes consideraciones:

- El nombre de la tabla se define en minúscula y en plural.
- Escoja nombres cortos, no ambiguos y usando no más de dos palabras. En caso de requerir más de una palabra, estas se deben separar con underscore (“\_”). Ej. *professor\_profiles*.
- Si el nombre requiere de más de dos palabras para su definición se debe utilizar la primera palabra para incluir un prefijo o iniciales que amplíen la descripción.
- El nombre de la tabla no debe superar los 30 caracteres, de otra forma se debe truncar.
- El nombre de los campos en cada tabla se definen en minúscula y no deben exceder los 30 caracteres, de otra forma se deben truncar.

tD

---

## 8. Nombres de Constraints

Para el nombre de los constraints se debe utilizar el siguiente formato:

- **Llaves primarias:**

*table\_pk*

*table* = la tabla sobre la que se construye el índice.

- **Llaves únicas:**

*table\_column\_pk*

*table* = la tabla sobre el que se construye el índice

*column* = nombre de la columna que tiene la característica de ser única.

- **Llaves foráneas:**

*tfather\_tdaughter\_fk*

*tfather* = nombre de la tabla padre

*tdaughter* = nombre de la tabla hija

## 9. Programación de interfaz gráfica

Para las aplicaciones con Interfaz Gráfica para el usuario se debe seguir el Manual de interfaces 2014 de Comunicación Visual (ver documento 'Mensajes\_interfaces\_v4.pdf'). Para seguir este manual y realizar la programación de la Interfaz, se deben utilizar solamente la Standar Library del TEC Digital (ver: <http://tecdigital.tec.ac.cr/tds-lib/#/>).

## 10. Internacionalización

Para un tutorial de como utilizar la internacionalización de OpenACS el usuario debe visitar: <http://openacs.org/doc/i18n-introduction.html>

[Rafael Oliver Murrillo, 16/10/2015] Para mostrar texto del lado del cliente utilizando archivos javascript y que este debe ser traducido según el lenguaje configurado en el servidor por parte del usuario, se propone la siguiente solución:

## tD

---

1. Se debe contar con un archivo tcl y adp, en algún lugar dentro del proyecto. Se sugiere que sea en /www/ajax/internacionalizacion. El archivo debe contar con una estructura similar a la siguiente:

```
ad_page_contract {
    Propósito: Cargar variables según idioma
    @create_date 06/05/2015
    @author Rafael Oliver
} {
    nombre_archivo_js
}
switch $nombre_archivo_js {

    gestion {
        set lista_variables [list "txt_levels" #tda-ga.txt_levels#
                                "txt_components" #tda-ga.txt_components# "txt_choose_level"
                                #tda-ga.txt_choose_level# "txt_write_description_level"
                                #tda-ga.txt_write_description_level#]

        set variables [util::json::gen [util::json::object::create
                                        $lista_variables]]
    }
    asignacion {
        set lista_variables [list "lb_attributes" #tda-ga.lb_attributes#
                                "txt_priority" #tda-ga.txt_priority# "txt_loading_graphs"
                                #tda-ga.txt_loading_graphs# "err_loading_graphs"
                                #tda-ga.err_loading_graphs#]

        set variables [util::json::gen [util::json::object::create
                                        $lista_variables]]
    }
}
```

## tD

---

Dentro del switch se cargan las variables que se requieren para el archivo javascript, esto para no cargar siempre todas las variables.

2. Una vez creada la estructura tcl, dentro de cada uno de los archivos javascript que van a mostrar las variables se debe incluir lo siguiente:

var variables ; → **Toma en cuenta que debe estar disponible dentro del scope en el que se requiere acceder (por aquello de ser utilizado en otros .js)**

```
$.ajax({
    async:false,
    type: "POST",
    url: '/tda-ga/ajax/internacionalizacion',
    data: { nombre_archivo_js:"asignacion" },
    success: function(data) {
        variables = JSON.parse(data);
    },
    error: function(xhr, ajaxOptions, thrownError){
        /*mostrarError("Ha ocurrido un error al cargar los mensajes
de usuario.");*/
    }
});
```

Esto cargará en la variable “variables” que se escogieron en el archivo tcl.

3. Una vez cargadas las variables, sólo se debe acceder el objeto creado de la siguiente forma:

```
$( '<p>', {html: variables.lb_attributes});
```

tD

---

De esta forma se creará una estructura HTML de tipo p con el contenido que corresponde a lb\_attributes (“**Atributos**”, en este caso), que en el fondo se está accediendo a la variable de internacionalización tda-ga.lb\_attributes.

## 11. Versionamiento

Para el versionamiento de los sistemas desarrollados se debe seguir las políticas para el versionamiento y almacenamiento de los sistemas desarrollados en el TEC Digital (ver documento ‘Políticas para el manejo de versiones.pdf’ en la carpeta de Git dentro de Conocimiento del TEC Digital, sección de Desarrollo).

## Anexo C. Actualización sobre Plantilla de Control de Calidad

En la siguiente imagen se muestra la adición (en un cuadro rojo) de escalas de evaluación que se realizó a la Plantilla de Control de Calidad de software del TEC Digital.

tecDigital					Asepectos de compatibilidad e integración del proyecto:					
1										
2	Criterios	Descripción	Evaluación (1-5)		Observaciones					
3	<b>Compatibilidad</b>									
4	<b>Multi-navegador</b>									
5	Mozilla Firefox	El sistema puede ser visto y ejecutado en este navegador de manera correcta.								
6	Chrome	El sistema puede ser visto y ejecutado en este navegador de manera correcta.								
7	Internet Explorer	El sistema puede ser visto y ejecutado en este navegador de manera correcta.								
8	Safari	El sistema puede ser visto y ejecutado en este navegador de manera correcta.								
9	<b>Multi-plataforma</b>									
10	Linux	El sistema puede ser ejecutado en el sistema operativo.								
11	Mac OS	El sistema puede ser ejecutado en el sistema operativo.								
12	Windows	El sistema puede ser ejecutado en el sistema operativo.								
13	Movil	El sistema se ejecuta de manera correcta en los dispositivos móviles								
14	<b>Integración</b>									
15	Las interconexiones con otros paquetes se mantienen	Las interconexiones con otros paquetes se mantienen, aún en condiciones de error.								
16	Integración acordada	Las interconexiones con otros paquetes se generan cuando fueron previamente acordadas								

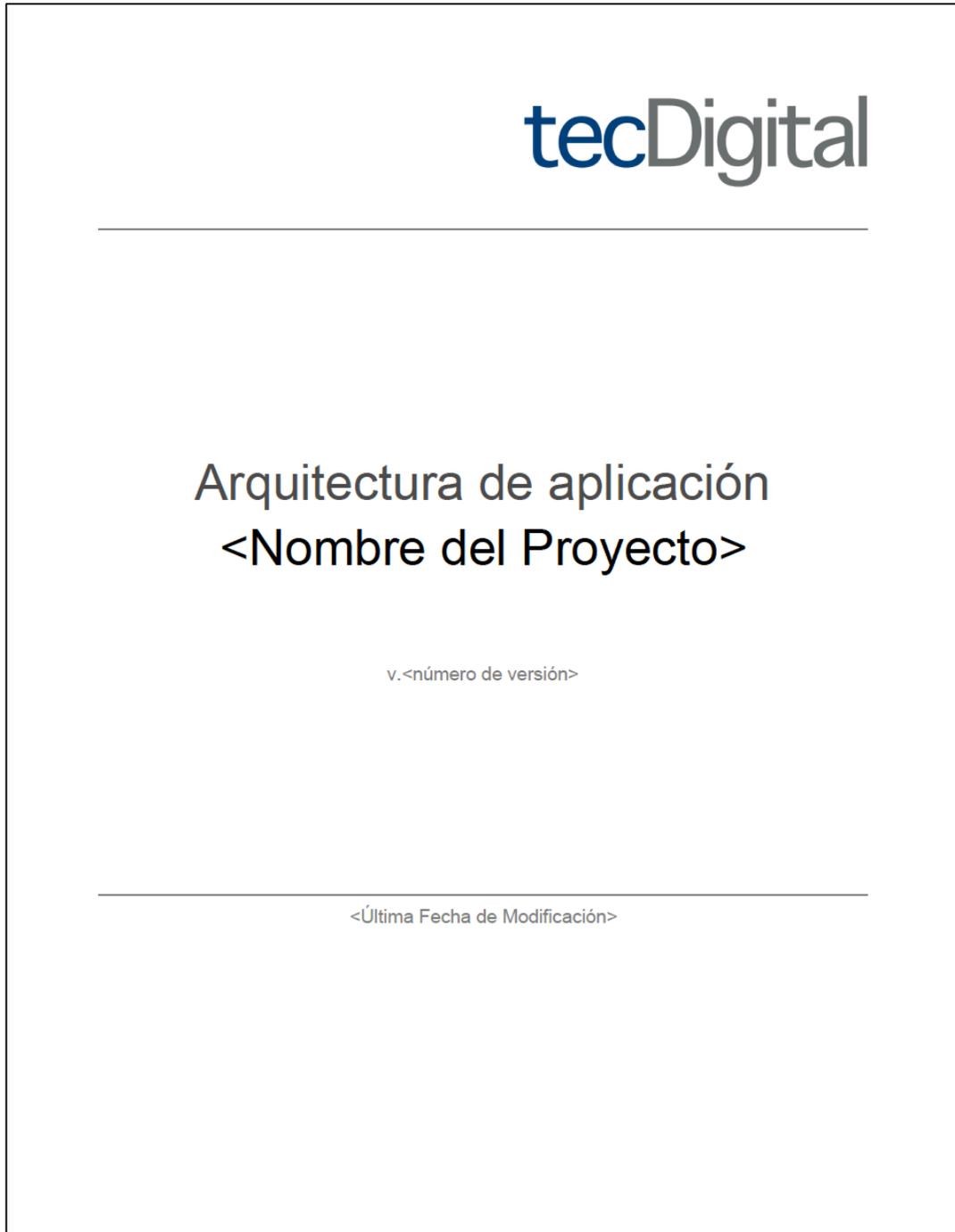
## Anexo D. Plantilla de autoevaluación

En la siguiente imagen se muestra la plantilla que se elaboró para que los equipos desarrolladores de software del TEC Digital, se evaluaran a sí mismos.

	A	C	D	E	F	G	H
1	<b>Criterios actuales de evaluación</b>	<b>Programa de curso</b>	<b>Diseño Instruccional</b>	<b>Evaluaciones</b>	<b>Rúbricas</b>	<b>Plan Estudios Gráfico</b>	<b>Expediente Es</b>
2	<b>Portal</b>						
3	Estructura de pestañas						
4	Descripción						
5	Video Demo						
6	Justificación						
7	Obj. General						
8	Obj. Especificos						
9	Estado Proyecto						
10	Fecha cierre						
11	Involucrados						
12	Modelo Conceptual						
13	Arquitectura Sist.						
14	Diagrama BD						
15	Requerimientos						
16	Documentos						
17	Admin						
18	<b>Calidad de Paquetes</b>						
19	Distribución carpetas						
20	.info						
21	Nombre significativo						
22	Archivos /www						
23	Responsable						

## Anexo E. Documento de Arquitectura

En el siguiente documento se muestran los elementos requeridos para el análisis de requerimientos, el diseño y la arquitectura de un proyecto de software del TEC Digital.



tD

---

## Control de cambios

### Documento original

Versión 1.0	Fecha	Persona que elaboró documento
Descripción:	Descripción inicial del documento.	

### Cambios realizados

Versión ...	Fecha	Persona que cambió documento
Descripción:	Descripción del cambio, elaborar una tabla de estas por cada cambio que se considere necesario.	

tD

---

## Tabla de Contenidos

### Tabla de Contenidos

1. Especificación de Requerimientos

2. Arquitectura de la Aplicación

3. Modelo Conceptual

4. Diagrama General de Casos de Uso

5. Especificación de Casos de Uso (Opcional)

<Nombre del Caso de Uso>

6. Diagrama de Clases(Opcional)

7. Diseño de la Base de datos

Fuentes de Datos

Modelo Entidad-Relación

Diccionario de datos

8. Seguridad (Opcional)

9. Modificaciones a la Plataforma (Opcional)

10. Instalación y configuración (Opcional)

tD

---

## 1. Especificación de Requerimientos

<Se presenta la lista de requisitos funcionales que se deben satisfacer en el sistema. Estos determinan el alcance del producto.

Es necesario que estos requerimientos sean puntualizados en forma detallada y precisa, y se encuentren agrupados por producto o subproducto, es decir se debe identificar el producto y luego organizar sus requisitos como parte de este.

El resultado de los mismos debe reflejar la lista de entregables definidos en el charter del proyecto.

Los requisitos deberán presentarse de una manera ordenada, numerada, procurando su ubicación según módulo del sistema.>

### **Entregable <Nombre o Identificador>**

Los requisitos funcionales son:

1. Descripción del requerimiento funcional
2. ..

tD

---

## 2. Arquitectura de la Aplicación

<Esta sección presenta una primera aproximación del diseño del software, involucrando aspectos de diseño o distribución de hardware y software de terceros requeridos para la implantación o puesta en marcha del proyecto.

Se presenta una descripción textual de los elementos técnicos que involucra el producto, se deberán indicar los conceptos tecnológicos involucrados, por ejemplo si el software se realiza en n-capas o cliente-servidor, etc.

Finalmente se presenta un diagrama con el diseño tecnológico involucrado en el proyecto.>

tD

---

### 3. Modelo Conceptual

<Esta sección presenta en una gráfica los flujos de información que se dan entre el sistema, como un todo, y los actores externos al sistema propiamente. Se presenta un diagrama general, sin entrar en detalles de formatos ni orden cronológico.>

tD

---

## 4. Diagrama General de Casos de Uso

<Esta sección presenta los principales procesos o actividades que se dan entre los actores externos al sistema y el sistema mismo. Se presenta mayor nivel de detalle, en término que se deben especificar los nombres de los procesos o actividades que se realizan con el sistema.

Los flujos de datos que se presentan en las gráficas no presentan ningún orden cronológico ni de secuencia alguna. Se emplean para clarificar solamente los tipos de proceso o relación que tienen los usuarios y otros sistemas con el software a desarrollar.

Esta sección es realmente importante de tener clara y precisa, debido a que estas actividades enumeradas acá evolucionarán prácticamente en los módulos y/u opciones del sistema. Ver ejemplo en el Anexo al documento.>

tD

## 5. Especificación de Casos de Uso (Opcional)

<Será la base primordial para la estimación que se desarrollará en el Plan de Proyecto.

En esta sección se especifican los casos de uso de las diferentes funcionalidades del proyecto. Identificando los principales actores, el flujo normal de eventos, así como los flujos alternativos. Se determina la prioridad de los diferentes casos de uso así como requerimientos especiales y suposiciones.

Un punto muy importante de ésta sección son las Reglas del Negocio, en el cual se debe especificar, como su nombre lo indica, reglas importantes del negocio que pueden afectar la funcionalidad que se desea modelar.

Algunas de las secciones del apartado serán desarrolladas en el transcurso de la ejecución del proyecto, de manera que al finalizar el mismo, se refleje la información actualizada del mismo y sea referencia para el cierre del proyecto.>

En esta sección se describen los casos de uso de la aplicación <Nombre de aplicación>, cada uno de ellos presenta la siguiente información:

- *Prioridad - Frecuencia*: Indica la prioridad del caso de uso para su futura clasificación, así como la frecuencia de uso del mismo.
- *Actores*: Indicando puntualmente los actores que participan
- *Descripción general del caso de uso*: Breve descripción que establece el propósito del caso de uso.
- *Especificaciones físicas*: Decisiones de diseño.
- *Condiciones previas*: Pre-condiciones
- *Condiciones posteriores*: Pos-condiciones
- *Fórmulas de cálculo*: Si aplican, de lo contrario indicar **No Aplica**.
- *Flujo básico de eventos*: Secuencia básica numerada de los pasos que se consideran al ejecutar cada caso de uso
- *Flujos alternos*: Secuencia secundaria numerada de eventos. Comprende variantes del flujo básico y casos especiales
- *Excepciones*: Listado de las posibles excepciones de la ejecución.
- *Reglas del Negocio*: Reglas del negocio importantes que puedan afectar la funcionalidad que se desea modelar.

A continuación el detalle de los casos de uso:

<Nombre del Caso de Uso>

Prioridad - Frecuencia

<Número de prioridad> - <Frecuencia de utilización>

Actores

<Desarrollo de la información>

Descripción General

<Desarrollo de la información>

tD

---

Especificaciones físicas

<Desarrollo de la información>

Condiciones previas

<Desarrollo de la información>

Condiciones Posteriores

<Desarrollo de la información>

Fórmulas de Cálculo

<Desarrollo de la información>

Flujo Básico de Eventos

<Desarrollo de la información. De gran importancia indicar el código SQL necesario para satisfacer cada uno de los pasos que involucra agregar, obtener, eliminar, actualizar información, así como observaciones importantes para el mejor entendimiento del mismo.>

Flujos Alternos

<Desarrollo de la información. De gran importancia indicar el código SQL necesario para satisfacer cada uno de los pasos que involucra agregar, obtener, eliminar, actualizar información, así como observaciones importantes para el mejor entendimiento del mismo.>

Excepciones

<Desarrollo de la información>

Reglas del Negocio

<Desarrollo de la información>

tD

---

## 6. Diagrama de Clases(Opcional)

<Esta sección presenta un diseño de las clases u objetos que se requieren para la implementación del software. Este diagrama es muy útil en el desarrollo de aplicaciones basadas en tecnologías o lenguajes orientados a objetos.

En caso de emplear lenguajes no orientados a objetos, estos diagramas permiten mayor entendimiento del sistema, su funcionalidad y sus relaciones. >

tD

---

## 7. Diseño de la Base de datos

### Fuentes de Datos

<En la primer versión de este documento, esta sección deberá indicar la leyenda Por definir ya que se desarrollará al iniciar la fase de Ejecución y Control del proyecto.

En esta sección se muestra de donde provienen los datos que se utilizan, esto es: las bases de datos, los servicios web y otros. Ver ejemplo en Fuentes de Datos. Ver ejemplo en el Anexo al documento.>

### Modelo Entidad-Relación

<En la primer versión de este documento, esta sección deberá indicar la leyenda Por definir ya que se desarrollará al iniciar la fase de Ejecución y Control del proyecto.

Esta sección presenta el diseño del modelo entidad relación de la base de datos. Se presenta el diseño y preferiblemente el modelo físico. >

### Diccionario de datos

Las siguientes tablas especifican el diccionario de la base de datos utilizada en el proyecto <Nombre de la aplicación>.

Cada tabla incluye el nombre de la tabla, sus columnas, el tipo de dato, tamaño, descripción y propiedades establecidas a cada columna como los son las restricciones de nulidad, llave primaria y llave foránea. (Nota: tanto las tablas como las columnas deben tener orden alfabético)

<Nombre de la tabla>						
Nombre	Tipo	Tamaño	Not Null	P K	F K	Descripción
<Nombre Columna>	ej. varchar	ej. 4	ej. X	ej .X	ej .X	<Descripción>

tD

---

## 8. Seguridad (Opcional)

<En la primer versión de este documento, esta sección deberá indicar la leyenda Por definir ya que se desarrollará al iniciar la fase de Ejecución y Control del proyecto.

En esta sección se especifica cómo se manejan los permisos de acceso a la aplicación, esto es, si se hace utilizando funciones de seguridad de la plataforma o si se hace por otro método. Además presenta qué actores tienen permisos.>

tD

---

## 9. Modificaciones a la Plataforma (Opcional)

<En la primer versión de este documento, esta sección deberá indicar la leyenda Por definir ya que se desarrollará al iniciar la fase de Ejecución y Control del proyecto.

En esta sección se especifican los paquetes que debieron ser modificados para desarrollar la aplicación, así como cuáles fueron los cambios.>

tD

---

## 10. Instalación y configuración (Opcional)

<En esta sección se especifica las diferentes formas de instalar la aplicación desarrollada dependiendo del ambiente de uso.>

### NOTA IMPORTANTE

Si alguna de las secciones presentadas anteriormente no aplica para un proyecto en particular debe excluirla, indique **No aplica** y especifique razón por la cual no aplica.

## APÉNDICES

### Apéndice A. Plantilla de minuta de reunión del trabajo final de graduación

La siguiente plantilla fue la base de elaboración de todas las minutas que se desarrollaron en el trabajo final de graduación.

			
<b>Minuta de Trabajo Final de Graduación (TFG)</b>			
Minuta No.: aaaammdd		Fecha de minuta: aaaa/mm/dd	
<b>1. Información de Reunión</b>			
Fecha	Hora de inicio	Hora de Finalización	Lugar / Medio
aaaa/mm/dd			
<b>Agenda</b>			
1.			
<b>2. Datos de promotor(a) de la reunión</b>			
Área Promotora	Nombre de Promotor(a)	Correo / Teléfono	
<b>3. Participantes</b>			
Nombre	Correo/Teléfono	Rol	Iniciales
<b>4. Puntos tratados</b>			
☐			
<b>5. Acuerdos / Consultas</b>			
Id	Descripción	Responsables (Iniciales)	Fecha de cumplimiento
01			
<b>6. Puntos Pendientes</b>			
1.			
<b>7. Firman de conformidad los participantes</b>			
Participante		Firma	
Julia Espinoza Guzmán			
Krissia Gómez Román			
Sonia Mora González			
Michael Sánchez Soto			
<b>Nota:</b> 48 horas después de ser enviada la minuta a los participantes, no se recibirán comentarios, por lo que se dará por hecho la aprobación de la misma.			
Minuta de TFG		Pág. 1 / 1	

## Apéndice B. Minuta número uno del trabajo final de graduación

			
<b>Minuta de Trabajo Final de Graduación (TFG)</b>			
Minuta No.. 20170214		Fecha de minuta: 2017/02/14	
<b>1 Información de Reunión</b>			
Fecha	Hora de inicio	Hora de Finalización	Lugar / Medio
2017/02/14	3:04 pm	4:27 pm	TEC Digital, sala de reuniones
<b>Agenda</b>			
<ol style="list-style-type: none"> <li>1 Resumen de reunión con Sonia Mora González.</li> <li>2. Definición de compromisos:             <ol style="list-style-type: none"> <li>a. Plan de Trabajo para el TFG.</li> <li>b. Entrega de avance corregido.</li> </ol> </li> <li>3. Enfoque metodológico.</li> </ol>			
<b>2. Datos de promotor(a) de la reunión</b>			
Área Promotora	Nombre de Promotor(a)	Correo / Teléfono	
Profesora asesora del TFG	Julia Espinoza Guzmán	juliespinoza@gmail.com	
<b>3. Participantes</b>			
Nombre	Correo/Teléfono	Rol	Iniciales
Julia Espinoza Guzmán	juliespinoza@gmail.com	Profesora Asesora del TFG.	JE
Krissia Gómez Román	krissiagomez@gmail.com	Persona encargada de la organización del TFG.	KG
Michael Sánchez Soto	sanchez.soto.michael@gmail.com	Estudiante cursando TFG.	MS
<b>4. Puntos tratados</b>			
<b>Resumen de reunión con Sonia Mora González.</b>			
<ul style="list-style-type: none"> <li><input type="checkbox"/> MS expone el resultado de la reunión con Sonia Mora González (SMG).</li> <li><input type="checkbox"/> MS detalla que la preocupación de SMG era sobre la ausencia que existe en el TFG de MS del tema de "diseño de software" específicamente dentro del marco teórico del TFG. SMG también apuntó que hay errores subsanables de forma que se deben corregir pero nada de gran impacto.</li> <li><input type="checkbox"/> MS informa que SMG también preguntó sobre la utilización de la ISO 9126 como norma para el modelo de calidad. MS comenta que Mario Chacón Rivas aconsejó dicha norma.</li> <li><input type="checkbox"/> JE informa que consulte mediante un correo al profesor Ignacio Trejos para preguntar sobre el aseguramiento y control de la calidad.</li> </ul>			
<b>Definición de compromisos</b>			
<ul style="list-style-type: none"> <li><input type="checkbox"/> MS se compromete a enviar plan de trabajo para el viernes y arreglos para el lunes siguiente.</li> <li><input type="checkbox"/> JE menciona que se puede consultar a Luis Javier, Ignacio Trejos e inclusive a Luis Alexander Calvo, además se debe buscar en bibliografía según APA como citar la "comunicación personal" con dichos profesionales. Buscar algún anexo o presentaciones de los mismos.</li> <li><input type="checkbox"/> JE comenta que los puntos detallados en la página número 1 se deben citar según la memoria del TEC Digital.</li> <li><input type="checkbox"/> JE menciona que se debe utilizar referencias cruzadas desde el inicio del TFG.</li> </ul>			
Minuta de TFG		Pág. 1 / 3	



**Enfoque metodológico.**

- MS expone que el enfoque metodológico presente es cualitativo, pero JE expone que considera que sería más adecuado utilizar un enfoque mixto. JE explica que la integración de los dos enfoques dotaría al TFG de un sustento más integral y con una base más sólida. Debido a que un análisis cualitativo permitirá identificar evidencias o hipótesis del por qué existe la problemática actual y el análisis cuantitativo confirmará o negará dichas hipótesis, inclusive permitirá identificar tendencias o comportamientos de los procesos que no se tengan presentes.
- JE consulta sobre el estatus del TFG anterior
- MS responde que se realizó hasta el marco metodológico, pero de forma incompleta.
- JE responde que se introduzca el tema de diseño de software en el marco teórico a un nivel general.
- JE se compromete a pasar un artículo sobre técnicas de "Focus Group" para recopilar información o para confirmar hallazgos. Explica que dicha herramienta debe tener una planeación sumamente elaborada para que sea bien desarrollada.
- JE comenta que se debe:
  - Redefinir el cronograma, se cuenta con un límite de tiempo menor
  - Definir metodología pero con un enfoque mixto.
- JE comenta que rastreo de la información para el enfoque mixto se debe describir dentro del marco metodológico.
- JE aconseja que se debe tener como una bitácora de decisiones, hasta de diálogos o conversaciones para que se tenga el registro de todos los pasos del TFG.
- JE explica que se puede hacer un análisis documental para el marco metodológico.
- KG indica que con este análisis se podrían identificar patrones o tendencias dentro de las revisiones de calidad del TEC Digital.
- JE indica que se debe incluir la población que se utilizará, todo esto dentro del marco metodológico. MS tiene que definir y seleccionar cuántos sujetos se van a utilizar de la población, es decir, la muestra que se utilizará.
- MS pregunta a JE cómo determinar el porcentaje de la consulta para la población. JE explica que se puede detallar según la conveniencia del proyecto, pero justificándose el porqué de la decisión.
- JE recomienda que se debe hacer esquema para el marco metodológico.
- MS menciona sobre esquema que se encuentra en un paper colombiano, del 2012, JE solicita si se puede tener algo más reciente sería mejor
- JE aconseja incluir dentro de la población a estudiantes y profesores. Trabajar mediante un tipo de formulario.
- MS consulta que si se puede recibir una aprobación de los capítulos iniciales una vez que sean corregidos.
- JE se compromete a buscar el libro de Fundamentos de Investigación Educativa.
- MS apunta que se puede usar los datos e información de las revisiones de calidad para incluir el enfoque cuantitativo dentro del marco metodológico.
- KG apunta que se debe buscar la forma en cómo se organizará la información de las revisiones, pensar en el esquema de resumen para ver los datos e información que se recopilarán.
- JE aconseja buscar más libros de metodologías de la investigación.

Minuta de TFGPág. 2 / 3




---

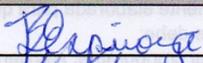
**5. Acuerdos / Consultas**

Id	Descripción	Responsables (Iniciales)	Fecha de cumplimiento
01	Compromiso para entregar plan de trabajo del TFG.	MS	2017/02/17
02	Avance corregido según la revisión del TFG por parte de SMG.	MS	2017/02/20
03	Facilitar el libro de Fundamentos de Investigación Educativa y el documento sobre elaboración de "focus groups"	JE	Ninguna.

**6. Puntos Pendientes**

Ninguno.

**7. Firman de conformidad los participantes**

Participante	Firma
Julia Espinoza Guzmán	
Krissia Gómez Román	
Michael Sánchez Soto	

**Nota:** 48 horas después de ser enviada la minuta a los participantes, no se recibirán comentarios, por lo que se dará por hecho la aprobación de la misma.

---

Minuta de TFG Pág. 3 / 3

## Apéndice C. Minuta número dos del trabajo final de graduación

			
<b>Minuta de Trabajo Final de Graduación (TFG)</b>			
Minuta No.. 20170223		Fecha de minuta: 2017/02/24	
<b>1 Información de Reunión</b>			
Fecha	Hora de inicio	Hora de Finalización	Lugar / Medio
2017/02/23	03:31 pm	04:17 pm	TEC Digital, sala de reuniones.
<b>Agenda</b>			
<ol style="list-style-type: none"> <li>1. Revisión de plan de trabajo.</li> <li>2. Revisión de avance del TFG.             <ol style="list-style-type: none"> <li>a. Título del TFG.</li> <li>b. Cambio de la introducción.</li> <li>c. Cambios en los objetivos.</li> </ol> </li> <li>3. Consulta sobre tiempos verbales:             <ol style="list-style-type: none"> <li>a. Tiempos verbales para capítulos:                 <ol style="list-style-type: none"> <li>i. ¿Tiempo futuro para metodología?</li> </ol> </li> <li>b. Abreviaciones para el TEC Digital (TD).</li> </ol> </li> </ol>			
<b>2. Datos de promotor(a) de la reunión</b>			
Área Promotora	Nombre de Promotor(a)	Correo / Teléfono	
ATI	Michael Sánchez Soto	sanchez.soto.michael@gmail.com	
<b>3. Participantes</b>			
Nombre	Correo/Teléfono	Rol	Iniciales
Julia Espinoza Guzmán	juliespinoza@gmail.com	Profesora Asesora del TFG.	JE
Krissia Gómez Román	krissiagomez@gmail.com	Persona encargada de la organización del TFG.	KG
Michael Sánchez Soto	sanchez.soto.michael@gmail.com	Estudiante cursando TFG.	MS
<b>4. Puntos tratados</b>			
<b>Revisión de plan de trabajo</b>			
<input type="checkbox"/> MS presenta el plan de trabajo creado para el desarrollo del TFG. El cual está implementado a modo de diagrama de Gantt en un documento de hoja de cálculo en línea.			
<input type="checkbox"/> JE explica que la fecha de entrega máxima (informe escrito) no puede sobrepasar el 2 de junio.			
<input type="checkbox"/> MS comenta que según el cronograma si se cumplirá con dicha fecha límite.			
<b>Revisión de avance del TFG</b>			
<b>Título del TFG</b>			
<input type="checkbox"/> MS presenta el nombre del nuevo título a KG, el cual elaboró mediante la guía de JE el 21 de febrero anterior			
<input type="checkbox"/> JE y KG aceptan el nuevo título del TFG.			
<input type="checkbox"/> MS se compromete a cambiar el título así como la redacción necesaria dentro del TFG para adaptarlo al nuevo título.			
<b>Cambio de la introducción</b>			
<input type="checkbox"/> MS muestra la nueva introducción a JE y KG.			
<input type="checkbox"/> MS se compromete a enviar la nueva introducción por correo electrónico para la revisión correspondiente.			
Minuta de TFG		Pág. 1 / 3	




---

**Cambios de los objetivos**

- MS muestra el nuevo objetivo principal a JE y KG.
- JE y KG comentan que consideran que dicho objetivo principal es correcto, queda aprobado.
- JE detalla que se deben revisar los específicos, principalmente el segundo. Se debe consultar la taxonomía de Bloom y encontrar una forma de redactarlos de acuerdo con los niveles de dicha taxonomía. JE explica que el TFG no puede presentar niveles inferiores y luego cumplir un objetivo del nivel superior
- JE considera que el tercer específico no es correcto. En su lugar se debe agregar el análisis para los requerimientos del sistema.
- MS se compromete a realizar cambios sobre los objetivos y enviarlos a JE para su debida revisión.

**Consultas**

**Tiempos verbales para capítulos**

- MS hace la consulta sobre los tiempos verbales.
- JE contesta que los mostrados en la guía suministrada por la Coordinadora del TFG son correctos.
- La mayoría de tiempos que se utilizarán serán en presente y en su mayoría en pasado.

**¿Tiempo futuro para metodología?**

- MS hace la consulta sobre el tiempo verbal de futuro para la metodología.
- JE comenta que piense en el momento que se entrega el informe final del TFG, será todo en pasado. Además hace la salvedad que aunque en el anteproyecto sea en futuro una vez en el informe final ya se tuvo a haber realizado la metodología, por lo cual es en presente y pasado.

**Abreviaciones para el TEC Digital**

- MS hace la consulta sobre abreviar el nombre del TEC Digital dentro del documento (mejora propuesta por la Coordinadora del TFG). JE responde que considera pertinente dicha abreviación, se debe hacer la aclaración dentro del documento y agregar en el glosario la misma.

**Varios**

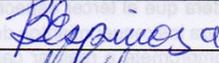
- MS consulta sobre la plantilla de informe semana. JE solicita que se pregunte con la Coordinadora del TFG.

**5. Acuerdos / Consultas**

Id	Descripción	Responsables (Iniciales)	Fecha de cumplimiento
01	Acuerdo: MS se compromete a cambiar el título así como la redacción necesaria dentro del TFG para adaptarlo al nuevo título.	MS	Próxima entrega de avance.
02	Acuerdo: MS muestra la nueva introducción a JE y KG. Se compromete a enviar la nueva introducción por correo electrónico para la revisión correspondiente.	MS	2017/02/24
03	Acuerdo: MS se compromete a realizar cambios sobre los objetivos y enviarlos a JE para su debida revisión.	MS	2017/02/25

Minuta de TFG

Pág. 2 / 3

 																			
04	Consulta: preguntar a la Coordinadora del TFG sobre el uso de los reportes semanales y el seguimiento de la plantilla propuesta.	MS	2017/02/24																
<b>6. Puntos Pendientes</b> Ninguno.																			
<b>7. Firman de conformidad los participantes</b>																			
Participante		Firma																	
Julia Espinoza Guzmán																			
Kressia Gómez Román																			
Michael Sánchez Soto																			
<p><b>Nota:</b> 48 horas después de ser enviada la minuta a los participantes, no se recibirán comentarios, por lo que se dará por hecho la aprobación de la misma.</p>																			
<p>5. Acuerdos y Consultas</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Descripción</th> <th>Responsable</th> <th>Fecha de entrega</th> </tr> </thead> <tbody> <tr> <td>01</td> <td>Acuerdo: MS se compromete a cambiar el título del TFG para como la redacción necesaria dentro del TFG para <i>según el nuevo título</i>.</td> <td>MS</td> <td>Próxima entrega de semana.</td> </tr> <tr> <td>02</td> <td>Acuerdo: MS muestra la nueva introducción a JE y KG. Se compromete a enviar la nueva introducción por correo electrónico para la revisión correspondiente.</td> <td>MS</td> <td>2017/02/24</td> </tr> <tr> <td>03</td> <td>Acuerdo: MS se compromete a realizar cambios sobre los objetivos y envíos a JE para su debida <i>revisión</i>.</td> <td>MS</td> <td>2017/02/28</td> </tr> </tbody> </table>				ID	Descripción	Responsable	Fecha de entrega	01	Acuerdo: MS se compromete a cambiar el título del TFG para como la redacción necesaria dentro del TFG para <i>según el nuevo título</i> .	MS	Próxima entrega de semana.	02	Acuerdo: MS muestra la nueva introducción a JE y KG. Se compromete a enviar la nueva introducción por correo electrónico para la revisión correspondiente.	MS	2017/02/24	03	Acuerdo: MS se compromete a realizar cambios sobre los objetivos y envíos a JE para su debida <i>revisión</i> .	MS	2017/02/28
ID	Descripción	Responsable	Fecha de entrega																
01	Acuerdo: MS se compromete a cambiar el título del TFG para como la redacción necesaria dentro del TFG para <i>según el nuevo título</i> .	MS	Próxima entrega de semana.																
02	Acuerdo: MS muestra la nueva introducción a JE y KG. Se compromete a enviar la nueva introducción por correo electrónico para la revisión correspondiente.	MS	2017/02/24																
03	Acuerdo: MS se compromete a realizar cambios sobre los objetivos y envíos a JE para su debida <i>revisión</i> .	MS	2017/02/28																
Minuta de TFG		Pág. 3 / 3																	

## Apéndice D. Minuta número tres del trabajo final de graduación

			
<b>Minuta de Trabajo Final de Graduación (TFG)</b>			
Minuta No.. 20170214		Fecha de minuta: 2017/03/03	
<b>1. Información de Reunión</b>			
Fecha	Hora de inicio	Hora de Finalización	Lugar / Medio
2017/03/03	04:00 pm	04:32 pm	TEC Digital
<b>Agenda</b>			
1. Aprobación de forma para los informes.			
2. Informe de avance de la semana.			
<b>2. Datos de promotor(a) de la reunión</b>			
Área Promotora	Nombre de Promotor(a)	Correo / Teléfono	
ATI	Michael Sánchez Soto	sanchez.soto.michael@gmail.com	
<b>3. Participantes</b>			
Nombre	Correo/Teléfono	Rol	Iniciales
Julia Espinoza Guzmán	juliespinoza@gmail.com	Profesora Asesora del TFG.	JE
Michael Sánchez Soto	sanchez.soto.michael@gmail.com	Estudiante cursando TFG.	MS
<b>4. Puntos tratados</b>			
<b>Aprobación de método para los informes.</b>			
<input type="checkbox"/> MS consulta a JE que para efectos del TFG, si es posible realizar el detalle de los informes de trabajo implícitamente, en las minutas de reunión mediante la descripción de los puntos tratados.			
<input type="checkbox"/> JE aprueba dicha forma de informe de trabajo.			
<b>Informe de avance de la semana.</b>			
<input type="checkbox"/> MS expone el trabajo colaborativo que se ha realizado bajo la guía de JE en la semana de trabajo actual (semana 4: 27 de febrero a 05 de marzo):			
<input type="checkbox"/> Capítulo 1			
<input type="checkbox"/> Se realizó la consulta sobre la numeración y contenido específico de dicho capítulo a la Coordinadora de los TFG de ATI, ya que el día 25 de febrero anterior se realizaron cambios sobre la misma y eso varió la estructura y profundidad de desarrollo del capítulo, la Coordinadora explica y confirma la variación.			
<input type="checkbox"/> Se completó el capítulo corregido según indicaciones de JE y se subió al portal del TEC Digital; sin embargo, se deben realizar los cambios producto de la variación en la numeración.			
<input type="checkbox"/> MS se compromete a tener corregida dicha enumeración para el 08 de marzo. Se está a la espera de la revisión final sobre las correcciones realizadas.			
<input type="checkbox"/> JE aprueba los objetivos presentes en la versión subida al TEC Digital del capítulo 1 los cuales fueron corregidos en la semana de trabajo actual y se detallan a continuación:			
<input type="checkbox"/> Objetivo general: Diseñar un sistema para el aseguramiento y control de la calidad en los proyectos de software del TD.			
<input type="checkbox"/> Objetivo específico 1 examinar las mejores prácticas, normas, marcos de referencia y metodologías de aseguramiento y control de la calidad de software para			
Minuta de TFG		Pág. 1 / 2	



- adquirir una base teórica de gestión de la calidad en proyectos de desarrollo de software.
- ❑ Objetivo específico 2: Analizar la brecha existente entre la base teórica adquirida de gestión de la calidad respecto a la ejecución de los procesos de aseguramiento y control de la calidad en los proyectos de desarrollo de software del TD.
  - ❑ Objetivo específico 3: Determinar el alcance de solución aplicable a la situación que se presenta actualmente, en la gestión de la calidad de los proyectos de software del TD.
  - ❑ MS se compromete a subir el avance del capítulo 2 al TEC Digital, el día 07 de marzo.
  - ❑ JE comenta sobre la creación de la bitácora de gestión de documentos, la cual compartió el 03 de marzo. Esta bitácora mantiene el estatus de los documentos y una forma de control sobre el proceso de seguimiento y revisión sobre el trabajo realizado. MS la completó con la información requerida por parte de la profesora asesora.
  - ❑ JE solicita un compromiso a MS de tener el anteproyecto ajustado para el miércoles 08 de marzo de acuerdo a las observaciones.
  - ❑ JE se compromete a enviar el instrumento de evaluación a la organización la próxima semana.
  - ❑ JE comenta sobre el marco metodológico, recomienda que se utilice la base de Sampieri y tomar en cuenta unas dos o tres referencias más (puede ser Maxwell y Creswell), además de ir analizando la estrategia de análisis de los resultados sobre el enfoque cuantitativo y cualitativo que se realizará, es decir se utilizará un enfoque mixto.

#### 5. Acuerdos / Consultas

Id	Descripción	Responsables (Iniciales)	Fecha de cumplimiento
01	Acuerdo: MS se compromete a tener corregida la enumeración del capítulo 1.	MS	2017/03/08
02	Acuerdo: MS se compromete a subir el avance del capítulo 2 al tecDigital.	MS	2017/03/07
03	Acuerdo: JE solicita un compromiso a MS de tener el anteproyecto ajustado de acuerdo con las observaciones.	MS	2017/03/07
04	Acuerdo: JE se compromete a enviar la rúbrica de evaluación a la organización.	JE	2017/03/06 al 2017/03/12

#### 6. Puntos Pendientes

Ninguno.

#### 7. Firman de conformidad los participantes

Participante	Firma
Julia Espinoza Guzmán	
Michael Sánchez Soto	

**Nota:** 48 horas después de ser enviada la minuta a los participantes, no se recibirán comentarios, por lo que se dará por hecho la aprobación de la misma.

## Apéndice E. Minuta número cuatro del trabajo final de graduación

			
<b>Minuta de Trabajo Final de Graduación (TFG)</b>			
Minuta No.. 20170314		Fecha de minuta: 2017/03/14	
<b>1. Información de Reunión</b>			
Fecha	Hora de inicio	Hora de Finalización	Lugar / Medio
2017/03/14	04:00 pm	04:46 pm	TEC Digital
<b>Agenda</b>			
<ol style="list-style-type: none"> <li>1 Informe de avance</li> <li>2. Consulta y comunicación sobre formato en redacción:             <ol style="list-style-type: none"> <li>a. ¿Sangría?</li> <li>b. Citado de las figuras.</li> </ol> </li> <li>3. Varios.</li> </ol>			
<b>2. Datos de promotor(a) de la reunión</b>			
Área Promotora	Nombre de Promotor(a)	Correo / Teléfono	
ATI	Michael Sánchez Soto	sanchez.soto.michael@gmail.com	
<b>3. Participantes</b>			
Nombre	Correo/Teléfono	Rol	Iniciales
Julia Espinoza Guzmán	juliespinoza@gmail.com	Profesora Asesora del TFG.	JE
Michael Sánchez Soto	sanchez.soto.michael@gmail.com	Estudiante cursando TFG.	MS
<b>4. Puntos tratados</b>			
<b>Informe de avance:</b>			
<input type="checkbox"/> MS comunica que se subió el anteproyecto corregido al portal del TEC Digital. <input type="checkbox"/> JE revisó el anteproyecto y queda aprobado como versión final. <input type="checkbox"/> MS comunica que 07 de marzo se subió la primera versión del capítulo 2 (Marco Teórico) al portal del TEC Digital. <input type="checkbox"/> JE se compromete a revisar los capítulos 1 y 2 del TFG.			
<b>Consulta y comunicación sobre formato en redacción:</b>			
<input type="checkbox"/> ¿Sangría? <input type="checkbox"/> MS consulta sobre este uso. <input type="checkbox"/> Citado de las figuras y tablas. <input type="checkbox"/> JE se compromete a realizar la consulta a la Coordinadora Sonia Mora sobre algún tipo de reglamento de formato y estilos para el TFG.			
<b>Varios:</b>			
<input type="checkbox"/> MS se compromete a finalizar los capítulos 1 y 2 a más tardar el 26 de marzo, si se cuenta con una revisión de los mismos. <input type="checkbox"/> JE aborda el marco metodológico del TFG: <ul style="list-style-type: none"> <li><input type="checkbox"/> Cada fase de la metodología debe estar asociada a un objetivo del TFG.</li> <li><input type="checkbox"/> Se debe explicar detalladamente cada fase de la metodología, ejemplo:                 <ul style="list-style-type: none"> <li><input type="checkbox"/> Aplicación de algún instrumento:                     <ul style="list-style-type: none"> <li><input type="checkbox"/> Preparación.</li> <li><input type="checkbox"/> Validación (prueba piloto).</li> <li><input type="checkbox"/> Aplicación.</li> <li><input type="checkbox"/> Análisis de resultados.</li> </ul> </li> </ul> </li> <li><input type="checkbox"/> JE y MS proponen las fases metodológicas:</li> </ul>			
Minuta de TFG		Pág. 1 / 2	




Tecnología de Costa Rica  


---

- Elaboración de la base teórica sobre aseguramiento y control de la calidad en proyectos de software.
- Descripción del estado actual del aseguramiento y control de la calidad en el TEC.
- Análisis de brechas entre base teórica y estado actual de aseguramiento y control de la calidad del TEC Digital.
- Definición de alcance de solución.
- Diseño del sistema de aseguramiento y control de la calidad de proyectos de software.
- JE comenta que se debe definir los instrumentos de investigación que se utilizarán muy cuidadosamente, y prepararlos con detalle, evitar la aplicación de entrevistas o cuestionarios abiertos.
- JE agrega que una posible estrategia para un instrumento de evaluación sería definición de la prioridad de un componente de la solución mediante dos variables, pueden ser la importancia y la factibilidad.
- JE aporta que una posible segregación de resultados de los instrumentos entre todo el equipo desarrollador y el comité técnico, podría arrojar resultados interesantes.
- JE comenta que el alcance de la solución se podría definir según los instrumentos de investigación que se apliquen.
- JE aconseja a MS que mientras se realizará la corrección de los capítulos 1 y 2, se inicie con el capítulo 3 (marco metodológico).

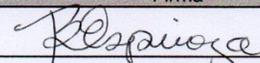
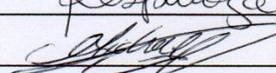
**5. Acuerdos / Consultas**

Id	Descripción	Responsables (Iniciales)	Fecha de cumplimiento
01	Acuerdo: JE se compromete a realizar la revisión del capítulo 1 y 2 del TFG de MS.	JE	No se especifica
02	Acuerdo: JE se compromete a realizar la consulta a la Coordinadora Sonia Mora sobre estilos y formato de redacción del TFG.	JE	2017/03/14
03	Acuerdo: MS se compromete a terminar de corregir los capítulos 1 y 2.	MS	2017/03/26

**6. Puntos Pendientes**

Ninguno.

**7 Firman de conformidad los participantes**

Participante	Firma
Julia Espinoza Guzmán	
Michael Sánchez Soto	

**Nota:** 48 horas después de ser enviada la minuta a los participantes, no se recibirán comentarios, por lo que se dará por hecho la aprobación de la misma.

---

Minuta de TFG

Pág. 2 / 2

## Apéndice F. Minuta número cinco del trabajo final de graduación

			
<b>Minuta de Trabajo Final de Graduación (TFG)</b>			
Minuta No.. 20170329		Fecha de minuta: 2017/03/30	
<b>1. Información de Reunión</b>			
Fecha	Hora de inicio	Hora de Finalización	Lugar / Medio
2017/03/29	2:30 pm	3:15 pm	TEC Digital
<b>Agenda</b>			
<ol style="list-style-type: none"> <li>1. Consulta sobre revisión de capítulos.</li> <li>2. Informe de avance de los capítulos.</li> <li>3. Consulta sobre temas:             <ol style="list-style-type: none"> <li>a. Diseño y arquitectura.</li> <li>b. BPMN.</li> </ol> </li> </ol>			
<b>2. Datos de promotor(a) de la reunión</b>			
Área Promotora	Nombre de Promotor(a)	Correo / Teléfono	
ATI	Michael Sánchez Soto	sanchez.soto.michael@gmail.com	
<b>3. Participantes</b>			
Nombre	Correo/Teléfono	Rol	Iniciales
Julia Espinoza Guzmán	juliespinoza@gmail.com	Profesora Asesora del TFG.	JE
Krissia Gómez Román	krissiagomez@gmail.com	Persona encargada de la organización del TFG.	KG
Michael Sánchez Soto	sanchez.soto.michael@gmail.com	Estudiante cursando TFG.	MS
<b>4. Puntos tratados</b>			
<b>Consulta sobre revisión de capítulos:</b>			
<input type="checkbox"/> MS consulta a JE sobre la revisión de los capítulos uno y dos. <input type="checkbox"/> JE se compromete a revisar completamente los cap 1 y 2 para el 06 de abril.			
<b>Informe de avance de los capítulos:</b>			
<input type="checkbox"/> MS comenta que realizó una revisión de los capítulos uno y dos sobre aspectos de redacción y mejoró la misma en el documento de informe final. <input type="checkbox"/> MS explica los temas que se desarrollaron en el marco teórico del trabajo final de graduación. <input type="checkbox"/> MS explica que se inició la elaboración del marco metodológico. <input type="checkbox"/> MS comenta que aplicará un instrumento de investigación en la semana del 03 al 07 de abril. <input type="checkbox"/> MS explica además algunos de los instrumentos que se han pensado aplicar para desarrollar el marco metodológico: <ul style="list-style-type: none"> <li><input type="checkbox"/> Entrevistas.</li> <li><input type="checkbox"/> Encuestas.</li> <li><input type="checkbox"/> Focus group.</li> <li><input type="checkbox"/> Análisis documental.</li> <li><input type="checkbox"/> Análisis cuantitativo de datos de las revisiones de calidad sobre los proyectos de software del TEC Digital.</li> <li><input type="checkbox"/> Análisis de brechas.</li> <li><input type="checkbox"/> Diagrama Ishikawa.</li> </ul> <input type="checkbox"/> JE solicita a MS una hoja de ruta de implementación de estos instrumentos.			
Minuta de TFG		Pág. 1 / 2	




---

- MS se compromete a tener esta hoja de ruta de los instrumentos que serán aplicados para más tardar el lunes 03 de abril.
- JE recomienda consultar las opciones de análisis de información según Sampieri para definir la utilización y hoja de ruta de aplicación de dichos instrumentos.

**Consulta sobre temas:**

- Diseño y arquitectura:**
  - MS hace la consulta sobre la posibilidad de cambiar el tema dentro del marco teórico de diseño de software por diseño arquitectónico, ya que dentro del diseño del sistema para los procesos de calidad del TEC Digital, lo que se realizará será más bien un detalle del diseño arquitectónico, aparte del levantamiento de requerimientos de un software, utilizando la plantilla de arquitectura de la Unidad.
  - JE y KG aprueban dicho cambio.
- BPMN:**
  - MS hace la consulta si debe introducir la notación de administración de procesos de negocio dentro del marco teórico, ya que se utilizará para describir procesos dentro del análisis de resultados.
  - JE afirma que sí debe introducir dicho tema.

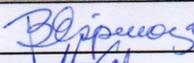
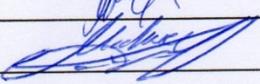
**5. Acuerdos / Consultas**

Id	Descripción	Responsables (Iniciales)	Fecha de cumplimiento
01	Acuerdo: JE se compromete a revisar los cap 1 y 2.	JE	2017/04/06
02	Acuerdo: MS se compromete a elaborar una hoja de ruta de los instrumentos que serán aplicados para el marco metodológico.	MS	2017/04/03

**6. Puntos Pendientes**

Ninguno.

**7. Firman de conformidad los participantes**

Participante	Firma
Julia Espinoza Guzmán	
Krissia Gómez Román	
Michael Sánchez Soto	

**Nota:** 48 horas después de ser enviada la minuta a los participantes, no se recibirán comentarios, por lo que se dará por hecho la aprobación de la misma.

Minuta de TFG

Pág. 2 / 2

## Apéndice G. Minuta número seis del trabajo final de graduación

			
<b>Minuta de Trabajo Final de Graduación (TFG)</b>			
Minuta No.. 20170420		Fecha de minuta: 2017/04/21	
<b>1. Información de Reunión</b>			
Fecha	Hora de inicio	Hora de Finalización	Lugar / Medio
2017/04/20	3:30 pm	5:20 pm	TEC Digital
<b>Agenda</b>			
1 Cronograma restante del Trabajo Final de Graduación (TFG).			
2. Revisión de marco metodológico.			
<b>2. Datos de promotor(a) de la reunión</b>			
Área Promotora	Nombre de Promotor(a)	Correo / Teléfono	
ATI	Julia Espinoza Guzmán	juliespinoza@gmail.com	
<b>3. Participantes</b>			
Nombre	Correo/Teléfono	Rol	Iniciales
Julia Espinoza Guzmán	juliespinoza@gmail.com	Profesora Asesora del TFG.	JE
Michael Sánchez Soto	sanchez.soto.michael@gmail.com	Estudiante cursando TFG.	MS
<b>4. Puntos tratados</b>			
<b>Cronograma restante del TFG</b>			
<ul style="list-style-type: none"> <li><input type="checkbox"/> MS muestra el cronograma del TFG.</li> <li><input type="checkbox"/> JE explica que debe agregar el tiempo para revisión final de parte de ella.</li> <li><input type="checkbox"/> JE adiciona que el cronograma de lo restante del TFG tiene fechas establecidas muy ajustadas en tiempo, por este motivo se deben cumplir con mucha precisión y en los periodos establecidos.</li> </ul>			
<b>Revisión del marco metodológico del TFG</b>			
<ul style="list-style-type: none"> <li><input type="checkbox"/> JE aconseja agregar las ISO que se puedan encontrar sobre las diferentes áreas relacionadas a la calidad del software, para introducirlas en las entrevistas individuales a colaboradores del TEC Digital.</li> <li><input type="checkbox"/> JE consulta sobre la entrevista con don Ignacio Trejos, MS comenta que no se había tenido respuesta aún al correo electrónico enviado. JE procede a contactar a don Ignacio Trejos y se acordó cita para el martes 25 de abril, esto con el fin de aplicar una entrevista individual al segundo experto en calidad de software.</li> <li><input type="checkbox"/> JE aconseja realizar un tipo de entrevista semi-estructurada con ideas básicas y selección única (u otros componentes cuantitativos), luego que sugieran los otros elementos; esto con el objetivo de procesar los resultados de una forma más provechosa.</li> <li><input type="checkbox"/> JE recomienda cambiar objetivo del grupo focal al siguiente: resumir resultados de las entrevistas individuales y llegar a un consenso sobre el estado actual de calidad de software del TEC Digital.</li> <li><input type="checkbox"/> JE aconseja definir los objetivos y resultados que se esperan de las entrevistas de forma muy clara, procesar los resultados por medio de categorías (como lo indica Hernández Sampieri y demás autores), y producir un resultado de análisis por la aplicación de cada técnica de la metodología.</li> <li><input type="checkbox"/> JE aconseja triangulación de datos (descrita por Hernández Sampieri y demás autores) para el análisis de resultados general.</li> </ul>			
Minuta de TFG		Pág. 1 / 2	



- JE y MS revisan de forma colaborativa, lo restante de la metodología del TFG.
- JE guía a MS mediante la elaboración de un diagrama ("a mano alzada") sobre posibles etapas o pasos que se deben de seguir para la aplicación exitosa de los demás elementos de la metodología propuesta para el TFG.

**5. Acuerdos / Consultas**

Id	Descripción	Responsables (Iniciales)	Fecha de cumplimiento
01	Acuerdo: MS se compromete a cargar en el TEC Digital el capítulo 3 (marco metodológico) del TFG con las debidas correcciones para el domingo siguiente, así como todos los instrumentos de investigación que se emplearán.	MS	2017/04/23

**6. Puntos Pendientes**

Ninguno.

**7. Firman de conformidad los participantes**

Participante	Firma
Julia Espinoza Guzmán	
Michael Sánchez Soto	

**Nota:** 48 horas después de ser enviada la minuta a los participantes, no se recibirán comentarios, por lo que se dará por hecho la aprobación de la misma.

## Apéndice H. Minuta número siete del trabajo final de graduación

			
<b>Minuta de Trabajo Final de Graduación (TFG)</b>			
Minuta No.. 20170511		Fecha de minuta: 2017/05/11	
<b>1. Información de Reunión</b>			
Fecha	Hora de inicio	Hora de Finalización	Lugar / Medio
2017/05/11	9:30 am	11:50 am	TEC Digital
<b>Agenda</b>			
<ol style="list-style-type: none"> <li>1. Recopilación de los requerimientos de funcionalidades para una herramienta de software que se integre al sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital.</li> <li>2. Recopilación de categorías de análisis para las herramientas de software que se integren al sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital.</li> </ol>			
<b>2. Datos de promotor(a) de la reunión</b>			
Área Promotora	Nombre de Promotor(a)	Correo / Teléfono	
Estudiante cursando TFG.	Michael Sánchez Soto	sanchez.soto.michael@gmail.com	
<b>3. Participantes</b>			
Nombre	Correo/Teléfono	Rol	Iniciales
Krissia Gómez Román	krissiagomez@gmail.com	Persona encargada de la organización del TFG.	KG
Michael Sánchez Soto	sanchez.soto.michael@gmail.com	Estudiante cursando TFG.	MS
<b>4. Puntos tratados</b>			
<ul style="list-style-type: none"> <li><input type="checkbox"/> MS explica el contenido de la propuesta de solución: diseño del sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital.</li> <li><input type="checkbox"/> MS detalla las funcionalidades que se buscan para la integración a dicho sistema.</li> <li><input type="checkbox"/> KG define entonces junto a MS los requerimientos de funcionalidades que se deben buscar en una herramienta de software:             <ul style="list-style-type: none"> <li><input type="checkbox"/> Capacidad de gestionar documentos que serán los artefactos para la ejecución del sistema.</li> <li><input type="checkbox"/> Funcionalidad para la evaluación de listas de chequeo o rúbricas.</li> <li><input type="checkbox"/> Capacidad para gestionar planes de prueba.</li> <li><input type="checkbox"/> Capacidad para documentar los hallazgos resultantes de la ejecución de los planes de prueba.</li> <li><input type="checkbox"/> Capacidad para enlazar casos de prueba a casos de uso.</li> <li><input type="checkbox"/> Envío de notificaciones para para ciertas actividades del proceso.</li> <li><input type="checkbox"/> Informes gráficos sobre resultados de ejecución de planes de prueba.</li> <li><input type="checkbox"/> Capacidad para varios tipos de usuarios.</li> </ul> </li> <li><input type="checkbox"/> Además, KG indica que a raíz de la forma de trabajo del TEC Digital y la búsqueda de utilización de software libre y con código abierto, se recomienda que las herramientas que se vayan a analizar cumplan esas características de libertad de uso en licenciamiento y apertura de su código fuente.</li> <li><input type="checkbox"/> KG solicita además, que las categorías para el cuadro de análisis de las herramientas de software sean las siguientes:             <ul style="list-style-type: none"> <li><input type="checkbox"/> Si es código abierto o no.</li> <li><input type="checkbox"/> Si es software libre o no.</li> </ul> </li> </ul>			
Minuta de TFG		Pág. 1 / 2	



- Los sistemas operativos en los que puede instalarse.
- El cumplimiento de las funcionalidades definidas.
- Si tienen o no comunidad de soporte.
- El tipo de ambiente y tecnologías de desarrollo.
- Mencionar funcionalidades adicionales que sean vinculantes al sistema propuesto.
- MS se compromete a realizar la descripción de las funcionalidades requeridas y el análisis comparativo de las herramientas de software mediante las categorías detalladas.

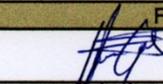
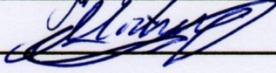
#### 5. Acuerdos / Consultas

Id	Descripción	Responsables (Iniciales)
01	Acuerdo: MS se compromete a realizar la descripción de las funcionalidades requeridas y el análisis comparativo de las herramientas de software mediante las categorías detalladas.	MS

#### 6. Puntos Pendientes

Ninguno.

#### 7. Firman de conformidad los participantes

Participante	Firma
Krissia Gómez Román	
Michael Sánchez Soto	

**Nota:** 48 horas después de ser enviada la minuta a los participantes, no se recibirán comentarios, por lo que se dará por hecho la aprobación de la misma.

## Apéndice I. Minuta número ocho del trabajo final de graduación



---

**Minuta de Trabajo Final de Graduación (TFG)**

Minuta No.. 20170517	Fecha de minuta: 2017/05/19
----------------------	-----------------------------

**1. Información de Reunión**

Fecha	Hora de inicio	Hora de Finalización	Lugar / Medio
2017/05/17	2:04 pm	3:07 pm	TEC Digital

**Agenda**

<p>1 Presentación de productos de TFG:</p> <ul style="list-style-type: none"> <li>a. Definiciones relacionadas a calidad de software.</li> <li>b. Diseño de sistema para el aseguramiento y control de la calidad de software en el TEC Digital.</li> <li>c. Plan de implementación de la propuesta de solución.</li> </ul>
---

**2. Datos de promotor(a) de la reunión**

Área Promotora	Nombre de Promotor(a)	Correo / Teléfono
Gestión de la calidad	Michael Sánchez Soto	sanchez.soto.michael@gmail.com

**3. Participantes**

Nombre	Correo/Teléfono	Rol	Iniciales
Andrea Calvo	acalvoelizando@gmail.com	Comunicación Visual	AC
Cristian Campos	ccampos04@gmail.com	Administración de proyectos	CC
Krissia Gómez	krissiagomez@gmail.com	Asesoría de TFG en el TEC Digital	KG
Pedro Leiva	pedro.leiva.ch@gmail.com	Desarrollo de proyectos	PL
Michael Sánchez	sanchez.soto.michael@gmail.com	Estudiante cursando TFG.	MS

**4. Puntos tratados**

<ul style="list-style-type: none"> <li><input type="checkbox"/> MS presenta explica la agenda y motivo de la presentación.</li> <li><input type="checkbox"/> MS inicia desarrollando las definiciones principales vinculantes al TFG:             <ul style="list-style-type: none"> <li><input type="checkbox"/> Calidad</li> <li><input type="checkbox"/> Aseguramiento de la calidad</li> <li><input type="checkbox"/> Control de la calidad</li> <li><input type="checkbox"/> Sistema</li> </ul> </li> <li><input type="checkbox"/> MS expone diseño de sistema para el aseguramiento y control de la calidad de software en el TEC Digital, por medio de los tres elementos principales:             <ul style="list-style-type: none"> <li><input type="checkbox"/> Roles</li> <li><input type="checkbox"/> Actividades</li> <li><input type="checkbox"/> Artefactos</li> </ul> </li> <li><input type="checkbox"/> MS explica la integración de las herramientas de software para la ejecución del sistema.             <ul style="list-style-type: none"> <li><input type="checkbox"/> Tarantula</li> <li><input type="checkbox"/> Aplicaciones del TEC Digital:                 <ul style="list-style-type: none"> <li><input type="checkbox"/> Documentos</li> <li><input type="checkbox"/> Evaluaciones</li> <li><input type="checkbox"/> Rúbricas</li> <li><input type="checkbox"/> Notificaciones</li> <li><input type="checkbox"/> Copiado de contenido</li> </ul> </li> </ul> </li> <li><input type="checkbox"/> PL consulta si se está tomando en cuenta el uso del Trello del TEC Digital.</li> </ul>
---

---

Minuta de TFG Pág. 1 / 2

- MS responde que no, que en el caso del TFG no se agregó el tema de gestión del tiempo.
- CC pregunta si Tarantula está dentro de las herramientas que recomendó Luis Alexander Calvo en la entrevista.
- MS responde que sí, Tarantula y otras herramientas más que fueron analizadas en el TFG mediante un cuadro comparativo.
- MS realiza una visita al sitio web de la aplicación Tarantula.
- Se aprueba por los colaboradores presentes el resultado del TFG en la organización.

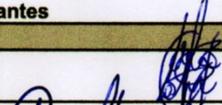
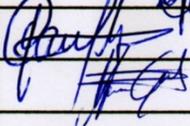
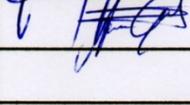
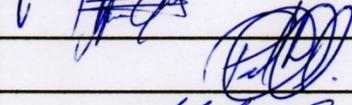
**5. Acuerdos / Consultas**

Id	Descripción	Responsables (Iniciales)
01	Acuerdo: se da el visto bueno al resultado del TFG por parte de los colaboradores presentes.	AC, CC, KG y PL

**6. Puntos Pendientes**

Ninguno.

**7. Firman de conformidad los participantes**

Participante	Firma
Andrea Calvo	
Cristian Campos	
Krissia Gómez	
Pedro Leiva	
Michael Sánchez	

**Nota:** 48 horas después de ser enviada la minuta a los participantes, no se recibirán comentarios, por lo que se dará por hecho la aprobación de la misma.

## Apéndice J. Minuta número nueve del trabajo final de graduación

			
<b>Minuta de Trabajo Final de Graduación (TFG)</b>			
Minuta No.: 20170530		Fecha de minuta: 2017/05/30	
<b>1. Información de Reunión</b>			
Fecha	Hora de inicio	Hora de Finalización	Lugar / Medio
2017/05/30	03:45 pm	04:15 pm	TEC Digital
<b>Agenda</b>			
<ol style="list-style-type: none"> <li>1. Entrega de trabajo final de graduación.</li> <li>2. Revisión de aspecto en capítulos finales.</li> </ol>			
<b>2. Datos de promotor(a) de la reunión</b>			
Área Promotora	Nombre de Promotor(a)	Correo / Teléfono	
ATI	Julia Espinoza Guzmán	juliespinoza@gmail.com	
<b>3. Participantes</b>			
Nombre	Correo/Teléfono	Rol	Iniciales
Julia Espinoza Guzmán	juliespinoza@gmail.com	Profesora Asesora del TFG.	JE
Krissia Gómez Román	krissiagomez@gmail.com	Persona encargada de la organización del TFG.	KG
Michael Sánchez Soto	sanchez.soto.michael@gmail.com	Estudiante cursando TFG.	MS
<b>4. Puntos tratados</b>			
<b>Entrega de trabajo final de graduación</b>			
<input type="checkbox"/> JE menciona que es la última reunión en conjunto con KG, la encargada por parte del TEC Digital.			
<input type="checkbox"/> JE le comenta a KG que le enviará la última evaluación en los días siguientes.			
<b>Revisión de aspecto en capítulos finales</b>			
<input type="checkbox"/> JE y MS le consultan a KG sobre el resultado del TFG.			
<input type="checkbox"/> KG menciona que se integró muy bien la teoría con la propuesta de solución.			
<input type="checkbox"/> JE expone que eso se buscó con el desarrollo de la metodología.			
<input type="checkbox"/> KG menciona que se debe agregar que la implementación debe ser un proceso gradual.			
<input type="checkbox"/> KG expone que la inclusión de las etapas de post-liberación es algo muy positivo. JE menciona a MS que se debe incluir esto en las conclusiones.			
<input type="checkbox"/> MS consulta sobre cómo mostrar el gráfico del diseño del sistema.			
<input type="checkbox"/> JE menciona que se debe desplegar la figura en una sola hoja, separar del texto.			
<b>5. Acuerdos / Consultas</b>			
Id	Descripción	Responsables (Iniciales)	Fecha de cumplimiento
01	Acuerdo: JE enviará la hoja de evaluación del trabajo desarrollado.	JE	31 de mayo
<b>6. Puntos Pendientes</b>			
Ninguno			
Minuta de TFG		Pág. 1 / 2	



**7. Firman de conformidad los participantes**

Participante	Firma
Julia Espinoza Guzmán	
Krissia Gómez Román	
Michael Sánchez Soto	

**Nota:** 48 horas después de ser enviada la minuta a los participantes, no se recibirán comentarios, por lo que se dará por hecho la aprobación de la misma.

## Apéndice K. Minuta de solicitud de cambio a la Coordinación de Trabajos Finales de Graduación

 <p>Diseño de sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital</p>			
<b>Trabajo Final de Graduación (TFG)</b> <b>Solicitud de cambio</b>			
Solicitud de cambio número: 01		Fecha de solicitud: 2017/05/10	
<b>1. Información de reunión de solicitud de cambio</b>			
Fecha	Hora de inicio	Hora de finalización	Lugar
2017/05/10	6:00 pm	6:55 pm	Centro de Investigación en Computación
<b>2. Persona promotora del cambio</b>			
Área promotora	Nombre de promotor	Correo	
Estudiante cursando TFG	Michael Sánchez Soto	sanchez.soto.michael@gmail.com	
<b>3. Personas involucradas del TFG</b>			
Nombre	Correo/Teléfono	Rol	Iniciales
Julia Espinoza Guzmán	juliespinoza@gmail.com	Profesora Asesora del TFG.	JE
Sonia Mora González	soniamora0407@gmail.com	Coordinadora de los TFG.	SM
Michael Sánchez Soto	sanchez.soto.michael@gmail.com	Estudiante cursando TFG.	MS
<b>4. Descripción de solicitud de cambio</b>			
<ul style="list-style-type: none"> <li><input type="checkbox"/> Cambiar la propuesta de un diseño de software (que sería desarrollado de forma posterior al TFG), para la integración sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital, por una propuesta de herramientas de software libre y con código abierto ya existentes que pueden realizar las mismas funcionalidades que se tenían previstas para dicha integración.</li> <li><input type="checkbox"/> La propuesta de herramientas de software se realizaría mediante un cuadro comparativo de las herramientas existentes y con base en categorías de comparación definidas colaborativamente, junto a la persona encargada del TEC Digital para la asesoría del TFG.</li> </ul>			
<b>5. Justificación de solicitud de cambio</b>			
<ul style="list-style-type: none"> <li><input type="checkbox"/> La propuesta de cambio surge a raíz de las entrevistas que se aplicaron a profesionales expertos en calidad de software, ellos proponen un abordaje de propuesta de solución distinto al que se tenía planteado al inicio del TFG. Estas entrevistas fueron realizadas en la semana 09 y 11, por esta razón, dicho abordaje no se había tomado en cuenta en el primer alcance de propuesta de solución.</li> <li><input type="checkbox"/> La implementación del cambio haría posible una alineación con la forma de trabajo del TEC Digital, al ser una Unidad que busca la utilización de software libre y de código abierto para la implementación de las tecnologías de la información en sus soluciones de software.</li> <li><input type="checkbox"/> Se plantea el cambio con el objetivo de seguir los lineamientos aprendidos durante el desarrollo de la carrera de Administración de Tecnología de Información, donde se enseñó mediante cursos como "Sistemas de Información Empresariales" o "Formulación y Evaluación de Proyectos de TI" que el desarrollo de nuevo software no siempre es la solución a una problemática, sino que se puede dar un abordaje distinto y resolver mediante herramientas de software ya existentes o evaluación de proceso y sistemas dicha problemática.</li> </ul>			
Trabajo Final de Graduación			Pág.1/2



Diseño de sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital

**6. Implicaciones de solicitud de cambio**

**Tiempo**

- 21 horas adicionales para la investigación y análisis de comparación de las herramientas de software que cumplan los requerimientos de funcionalidades para el sistema del TFG.

**Costo**

- Se suma 86.650,00 colones (costo estimado según horas adicionales y salario del estudiante) al desarrollo del TFG.
- Se reducirán 2.150.000,00 colones de tiempo de desarrollo para el TEC Digital, según estimaciones de desarrollo de un programador para tres meses y medio.
- Se requerirán 495.000,00 (equivalentes a tres semanas de trabajo) colones para el tiempo de configuración, parametrización y adaptación de las herramientas de software que serán instaladas en el TEC Digital.

**Alcance**

- El cambio no tiene implicación alguna en los objetivos del TFG, debido a que los mismos están definidos en términos del diseño de un sistema para el aseguramiento y control de la calidad de los proyectos de software del TEC Digital.
- El cambio tiene implicación en las actividades y productos finales que se realizarán para la elaboración de la propuesta de solución, debido a que, no se elaborará el diseño de una herramienta de software, sino una propuesta mediante un análisis comparativo de herramientas de software libre y código abierto que ya cuentan con funcionalidades para ser integradas en el sistema diseñado de aseguramiento y control de la calidad de los proyectos de software del TEC Digital.

**Calidad**

- Formulación de una solución de mayor calidad para el mercado laboral que se ajuste a las necesidades y características del mismo.
- Disminución de tiempos de implementación de la herramienta de software en el TEC Digital que se integrará al sistema para el aseguramiento y control de la calidad del software.
- Seguimiento de la línea de trabajo del TEC Digital sobre búsqueda, análisis y utilización de software libre y con código abierto.

**7. Acuerdos**

Id	Descripción	Responsables (Iniciales)
01	Acuerdo: se aprueba el cambio solicitado por el estudiante cursando el TFG.	SM y JE

**8. Firman de conformidad las personas involucradas**

Participante	Firma
Julia Espinoza Guzmán	
Sonia Mora González	
Michael Sánchez Soto	

## Apéndice L. Minuta de solicitud de cambio a los involucrados en el TEC Digital del trabajo final de graduación

ati		tecDigital	
<b>Minuta de Trabajo Final de Graduación (TFG)</b>			
Minuta No.. 20170510		Fecha de minuta: 2017/05/11	
<b>1. Información de consultas independientes</b>			
Fecha	Lugar / Medio		
2017/05/10	TEC Digital		
Estructura de consultas independientes para cada persona			
1. Descripción de propuesta de cambio. 2. Consulta sobre aprobación. 3. Criterio de los colaboradores involucrados.			
<b>2. Persona consultante</b>			
Área Promotora	Nombre de Promotor(a)	Correo / Teléfono	
ATI	Michael Sánchez Soto	sanchez.soto.michael@gmail.com	
<b>3. Personas consultadas</b>			
Nombre	Correo/Teléfono	Rol	Iniciales
Mario Chacón Rivas	mario.chacon.rivas@gmail.com	Profesional de TI del TEC Digital	MC
Julia Espinoza Guzmán	juliespinoza@gmail.com	Profesora Asesora del TFG.	JE
Agustín Francesa Alfaro	agustin.francesa@gmail.com	Coordinador del TEC Digital	AF
Krissia Gómez Román	krissiagomez@gmail.com	Persona encargada de la organización del TFG.	KG
Michael Sánchez Soto	sanchez.soto.michael@gmail.com	Estudiante cursando TFG.	MS
<b>4. Descripción de consultas realizadas</b>			
<input type="checkbox"/> MS acude al TEC Digital y consulta (de forma independiente entre sí, a cada uno de los colaboradores involucrados) sobre un cambio propuesto: <ul style="list-style-type: none"> <li><input type="checkbox"/> Se podría variar el alcance del proyecto, ya que, a raíz de dos entrevistas que se realizaron a profesionales expertos en calidad de software, se encuentra factible la búsqueda y análisis comparativo de herramientas de software para proponer las opciones de implementación de las mismas, de forma tal que sea integrada al sistema de gestión de la calidad.</li> <li><input type="checkbox"/> Este cambio en el alcance aportaría beneficios como:               <ul style="list-style-type: none"> <li><input type="checkbox"/> Disminución de tiempos de implementación de esta herramienta de software en la organización.</li> <li><input type="checkbox"/> Seguimiento de la línea de trabajo del TEC Digital sobre utilización de software libre y con código abierto (estos serían requisitos fundamentales para la comparación de las herramientas).</li> </ul> </li> </ul>			
<b>Consulta uno a JE:</b>			
<input type="checkbox"/> Horas de consulta: 8:15 am a 8:40am <input type="checkbox"/> MS comunica la propuesta de cambio. <input type="checkbox"/> JE comenta que considera prudente el cambio. <input type="checkbox"/> JE detalla referencias literarias para sustentar la implementación de dicho cambio: Memoria del TEC Digital y tesis doctoral de MC.			
Minuta de TFG		Pág. 1 / 2	




---

- JE recomienda que la propuesta de cambio sea consultado, primero, a los colaboradores del TEC Digital con vinculación al Trabajo Final de Graduación (TFG) y de igual forma con Sonia Mora González, Coordinadora de los TFG de ATI.
- JE aprueba el cambio, siempre y cuando se consulte con la Coordinadora de los TFG y ella lo apruebe.

**Consulta dos a MC:**

- Horas de consulta: 8:50 am a 9:20am
- MS comunica la propuesta de cambio.
- MC comenta que sería una buena decisión, además detalla documentación que puede dar respaldo a la utilización de una herramienta de software libre.
- MC aprueba el cambio propuesto.

**Consulta tres a KG:**

- Horas de consulta: 9:30 am a 9:50am
- MS comunica la propuesta de cambio.
- KG ve de forma positiva este cambio, pensando en términos prácticos de la implementación de la propuesta de solución.
- KG aprueba el cambio propuesto.

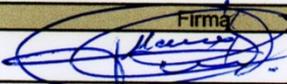
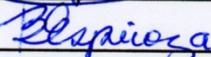
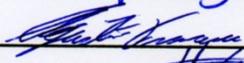
**Consulta cuatro a AF:**

- Horas de consulta: 4:01 pm a 4:47 pm
- MS comunica la propuesta de cambio.
- AF considera que no hay ningún problema en dicho cambio.
- AF aprueba el cambio.

**5. Acuerdos**

Id	Descripción	Responsables (Iniciales)
01	Acuerdo: se aprueba por parte del TEC Digital el cambio propuesto en el alcance. Se hará la consulta a la Coordinadora de los TFG, si ella lo aprueba, entonces se implementará el mismo.	MC, JE, AF y KG.

**6. Firman de conformidad las personas consultadas**

Participante	Firma
Mario Chacón Rivas	
Julia Espinoza Guzmán	
Agustín Francesa Alfaro	
Krissia Gómez Román	
Michael Sánchez Soto	

Minuta de TFG

Pág. 2 / 2

### Apéndice M. Cronograma del trabajo final de graduación.

En la siguiente imagen, se muestra el cronograma desarrollado para el Trabajo Final de Graduación.

Actividades	CRONOGRAMA DE TRABAJO FINAL DE GRADUACIÓN													
	Febrero		Marzo			Abril				Mayo			Junio	
	06 al 12	13 al 19	20 al 26	27 al 02	03 al 09	10 al 16	17 al 23	24 al 30	01 al 07	08 al 14	15 al 20	21 al 27	28 al 03	
Desarrollo del Capítulo I: Introducción														
Desarrollo del Capítulo II: Marco teórico														
Desarrollo del Capítulo III: Desarrollo metodológico														
Desarrollo del Capítulo IV: Análisis de resultados														
Desarrollo del Capítulo V: Propuesta de solución														

## Apéndice N. Mediciones

Los siguientes datos resumen las mediciones efectuadas para cuatro instancias del proceso de control de la calidad del software del proyecto Evaluaciones del TEC Digital.

Información de Evaluaciones		Instancias de control de calidad sobre Evaluaciones			
Mes de revisión		marzo	mayo	junio	setiembre
Coordinador		Isaac Alpizar Chacón			
Desarrolladora del proyecto		Ana Lucia Villavicencio Rosales			
Encargado de calidad		Michael Sánchez Soto			
Cantidad total de tareas		128	161	186	199
Número de tareas revisadas		128	33	25	13
<b>Número de tareas revisadas correctas e incorrectas</b>					
Correctas		123	29	21	11
Incorrectas		5	4	4	2
<b>Número hallazgos segmentados por errores, mejoras o dudas</b>					
Error		24	14	19	9
Mejora		23	5	1	0
Duda		5	1	1	0
Cantidad de hallazgos en general:		52	20	21	9
<b>Cantidad de hallazgos por tipo y por prioridad</b>					
Errores	Alta	8	7	8	4
	Medio	9	3	3	1
	Bajo	7	4	8	4
	Total:	24	14	19	9
Mejoras	Alta	18	5	1	0
	Medio	3	0	0	0
	Bajo	2	0	0	0
	Total:	23	5	1	0
Dudas	Alta	4	1	1	0
	Medio	1	0	0	0
	Bajo	0	0	0	0
	Total:	5	1	1	0
<b>Cantidad de hallazgos por sección</b>					
Promedio del días de respuesta		10.26	3.29	4.72	6.71
<b>Promedio de tiempo (días) en que se resuelven los hallazgos segmentados por:</b>					
Error		28.78	8.54	11.16	34.33
Mejora		26.53	3	0	0
Duda		26.40	1	9	0
Promedio total:		27.67	7.12	11.05	34.33

## Apéndice Ñ. Entrevista elaborada al profesional experto número uno

A continuación, se muestra la descripción de la entrevista aplicada al experto en calidad de software número uno.



---

**Entrevista para Trabajo Final de Graduación**  
**Diseño de sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital**

Fecha: 04 de abril de 2017.  
Entrevistado: Luis Alexander Calvo Valverde  
Hora inicio: 10:00 am.  
Hora fin: 10:36 am.  
Duración: 36 minutos.

**Objetivos de la entrevista**

1. Consultar conceptos relacionados a la calidad de software.
2. Conocer sobre el proyecto que realizó el entrevistado en el TEC Digital.
3. Consultar el estado de gestión de la calidad de software del TEC Digital de acuerdo con el entrevistado.
4. Presentar trabajo final de graduación, sus resultados esperados y consultar sobre los mismos.
5. Recolectar información adicional sobre aseguramiento y control de la calidad de software.

**Preguntas**

- Objetivo 1: conceptos relacionados a calidad de software.
  - 1) Defina los siguientes conceptos relacionados a la calidad de software:
    - Calidad de software.
    - Aseguramiento de la calidad de software.
    - Control de la calidad del software.
    - Modelo de calidad de software.
- Objetivo 2: proyecto realizado en el TEC Digital sobre calidad de software.
  - 1) ¿Cuál fue el proyecto que usted realizó en el TEC Digital relacionado a la calidad de software?
  - 2) ¿Cuál era el objetivo del proyecto relacionado a la calidad de software?
  - 3) ¿Cuál fue el resultado del proyecto relacionado a la calidad de software?

1



- Objetivo 3: estado de gestión de la calidad de software del TEC Digital.
  - 1) ¿Cuál es el estado actual del TEC Digital, según su criterio, en cuanto a gestión de la calidad de software?
  - 2) ¿Qué aspectos, procesos y actividades se pueden mejorar o implementar para proveer al TEC Digital de un estado de mayor madurez en procesos de calidad de software?
  
- Objetivo 4: presentación de trabajo final de graduación y sus resultados esperados.
  - 1) ¿Qué consideraciones tiene respecto a los resultados esperados?
    - Implementación de aseguramiento de la calidad dentro de la metodología de desarrollo del software del TEC Digital.
    - Alineación de metodología de desarrollo de software con base teórica: metodología de Avella y Gómez, modelo de calidad de software de la ISO 9126.
    - Automatización del aseguramiento y control de la calidad mediante herramienta de software.
    - Conocimiento de estado de calidad de los proyectos de software desde una perspectiva integral.
  - 2) ¿Son pertinentes al TEC Digital los resultados esperados?
  - 3) ¿Modificaría algo de los componentes o resultados esperados?
  
- Objetivo 5: información adicional sobre aseguramiento y control de la calidad.
  - 1) ¿Conoce alguna herramienta de software que gestione actividades o procesos de aseguramiento y control de la calidad de software?
  - 2) ¿Conoce alguna metodología o marco de referencia adicional para actividades o procesos de aseguramiento y control de la calidad de software?

## Apéndice O. Entrevista elaborada al profesional experto número dos

A continuación, se muestra la descripción de la entrevista aplicada al experto en calidad de software número dos.



---

**Entrevista para Trabajo Final de Graduación**  
**Diseño de sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital**

Fecha: 25 de abril de 2017.  
Entrevistado: Ignacio Trejos Zelaya.  
Hora inicio: 3:25 pm.  
Hora fin: 4:40 pm.  
Duración: 75 minutos.

**Contexto del trabajo final de graduación**

- Organización y Unidad donde se desarrolla: Instituto Tecnológico de Costa Rica, TEC Digital.
- Tema: aseguramiento y control de la calidad del software en la Unidad.
- Nombre: diseño de sistema para el aseguramiento y control de la calidad en los proyectos de software del TEC Digital.
- Resultados esperados:
  - 1) Diseño y planeación para la implementación del proceso de aseguramiento de la calidad y cambio en la metodología de desarrollo de software del TEC Digital.
  - 2) Automatización de actividades de aseguramiento y control de la calidad mediante el diseño de un software.
  - 3) Conocimiento de estado de calidad de los proyectos de software de la Unidad.

**Objetivos de la entrevista**

1. Consultar y confirmar conceptos relacionados a la calidad de software.
2. Conocer sobre los principales cuerpos de conocimiento respecto al aseguramiento y control de la calidad de software que existen actualmente.
3. Presentar trabajo final de graduación, sus resultados esperados y consultar sobre los mismos.
4. Recolectar información adicional sobre aseguramiento y control de la calidad de software.

**Preguntas**

- Objetivo 1: conceptos relacionados a calidad de software.
  - 1) ¿Para usted qué significan los siguientes conceptos?
    - Calidad de software.
    - Aseguramiento de la calidad de software.
    - Control de la calidad del software.

1



- Modelo de calidad de software.
  
- Objetivo 2: cuerpos de conocimiento sobre calidad de software.
  - 1) ¿Qué estándares, mejores prácticas, metodologías, modelos y demás cuerpos de conocimiento reconoce usted relacionados a la calidad de software?
  - 2) ¿Estos cuerpos de conocimiento sobre calidad de software tienen algún énfasis? Es decir: ¿Están enfocados en fases propias, en documentos, entregables del ciclo de vida de desarrollo del software?
  
- Objetivo 3: presentación de trabajo final de graduación y sus resultados esperados.
  - 1) ¿Qué consideraciones tiene respecto a los siguientes resultados esperados?
    - Implementación de aseguramiento de la calidad dentro de la metodología de desarrollo del software del TEC Digital.
    - Alineación de metodología de desarrollo de software con base teórica: metodología de Avella y Gómez, modelo de calidad de software de la ISO 9126 y otros estándares para la gestión de la calidad del software.
    - Automatización del aseguramiento y control de la calidad mediante herramienta de software.
    - Conocimiento de estado de calidad de los proyectos de software desde una perspectiva integral.
  - 2) ¿Son pertinentes al TEC Digital los resultados esperados?
  - 3) ¿Modificaría algo de los componentes o resultados esperados?
  - 4) ¿Agregaría algo a los componentes o resultados esperados?
  
- Objetivo 4: información adicional sobre aseguramiento y control de la calidad.
  - 1) ¿Conoce alguna herramienta de software que gestione actividades o procesos de aseguramiento y control de la calidad de software?
  - 2) ¿Qué recomendaciones adicionales tiene usted para una organización que se dedique al desarrollo de software y que presente el objetivo de mejorar su calidad de software?
  - 3) En su criterio académico y profesional: ¿Cuál es la clave o claves para que una organización dedicada al desarrollo de software mejore la calidad de sus proyectos y productos finales de los mismos?

## Apéndice P. Cuestionario aplicado a colaboradores del TEC Digital

El siguiente formulario se aplicó a diez colaboradores del TEC Digital para la actividad de cuestionario dentro de la metodología del proyecto.



---

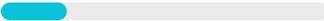
# Diseño de sistema para el aseguramiento y control de la calidad de software del TEC Digital.

El siguiente formulario tiene como fines conocer y recopilar datos, información y criterios para la elaboración de un sistema para el aseguramiento y control de la calidad de software en el TEC Digital. Los datos y la información que se consignan son confidenciales y para estricto uso en la elaboración de dicho sistema, que es parte del Trabajo Final de Graduación de Michael Sánchez Soto en la carrera de Administración de Tecnología de Información del TEC. Por favor desarrolle sus respuestas de forma específica y precisa.

**\*Obligatorio**

Escriba su nombre completo. \*

Tu respuesta

**SIGUIENTE**  Página 1 de 5

Nunca envíes contraseñas a través de Formularios de Google.



## Diseño de sistema para el aseguramiento y control de la calidad de software del TEC Digital.

\*Obligatorio

### Sección 1: contexto del término calidad.

Según su criterio académico y profesional, responda las siguientes preguntas.

#### 1.1. Defina calidad. \*

Tu respuesta

---

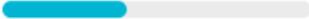
1.2. Defina calidad para su área específica de trabajo y desempeño (puede relacionarse al desarrollo de software, a diseño gráfico, a la administración de proyectos, a la capacitación, a la coordinación del TEC Digital, entre otras áreas). \*

Tu respuesta

---

ATRÁS

SIGUIENTE

 Página 2 de 5

Nunca envíes contraseñas a través de Formularios de Google.



---

## Diseño de sistema para el aseguramiento y control de la calidad de software del TEC Digital.

\*Obligatorio

### Sección 2: estado actual de la calidad del software del TEC Digital.

Complete las siguientes preguntas tomando como base la siguiente definición de calidad de software: el nivel de cumplimiento de los requerimientos y necesidades de sus usuarios finales, así como el nivel de aprovechamiento y efectividad de los recursos necesarios para su desarrollo.

**2.1. ¿Cuál categoría describe mejor, según su criterio, el estado de la calidad del software que es desarrollado en el TEC Digital? \***

- Mala.
- Regular.
- Buena.
- Excelente.

**2.2. Justifique su respuesta anterior. \***

Tu respuesta

---

**2.3. ¿Considera que el TEC Digital presenta una situación negativa o algún problema en cuanto a la calidad del software que desarrolla o la gestión de la misma? \***

Si marca la opción "Sí" responda la pregunta siguiente (2.4.), de lo contrario diríjase a la pregunta 2.5.

- Sí.
- No.

**2.4. ¿Qué problema o problemas encuentra usted respecto a la calidad del software que es desarrollado en el TEC Digital o a la gestión de la misma?**

Tu respuesta

---

**2.5. ¿Considera que la calidad con la que cuenta el software que es desarrollado en el TEC Digital o su gestión se pueden mejorar? \***

Si marca la opción "Sí" responda la pregunta siguiente (2.6.), de lo contrario diríjase a la siguiente sección.

- Sí.
- No.

**2.6. ¿De qué forma, tipo de propuesta o acciones considera usted necesarias implementar para mejorar la calidad del software que se desarrolla en el TEC Digital?**

Tu respuesta

---

[ATRÁS](#) [SIGUIENTE](#)

Página 3 de 5

Nunca envíes contraseñas a través de Formularios de Google.



---

## Diseño de sistema para el aseguramiento y control de la calidad de software del TEC Digital.

\*Obligatorio

### Sección 3: mejoramiento de la calidad de software del TEC Digital y su gestión, desde la perspectiva de su área.

Tomando en consideración su experiencia académica y profesional en su área y conocimiento de los activos (documentos, manuales, estándares, plantillas, etc.) del TEC Digital respecto a la calidad del software, responda las siguientes preguntas.

**3.1. ¿Conoce usted de algún activo del TEC Digital que brinde las condiciones u oportunidades para generar una excelente calidad para el software que se desarrolla en la Unidad o para su gestión? \***

Si marca la opción "Sí" responda la pregunta siguiente (3.2.), de lo contrario diríjase a la pregunta 3.3.

Sí.

No.

**3.2. Mencione dichos activos que brindan las condiciones u oportunidades para generar una excelente calidad para el software que se desarrolla en el TEC Digital o su gestión. (Ejemplos: plantilla para las minutas, manual de interfaces, entre otros)**

Tu respuesta

---

**3.3. Aparte de los activos del TEC Digital ¿Que otra herramientas, técnica, marco de referencia, metodología u otro elemento considera usted que podría mejorar la calidad del software desde la perspectiva de su área o labores de desempeño? (Ejemplo hipotético: la utilización de un ISO para el diseño de interfaces). \***

Si tiene conocimientos sobre software, por favor extienda su respuesta desde un punto de vista por etapas del ciclo de vida de desarrollo de software. Por ejemplo: para los requerimientos sería positivo utilizar un estándar específico de la IEEE que usted conozca o para el diseño del software sería buena práctica utilizar el estándar UML 2.0 únicamente, entre otros.

Tu respuesta

---

 Página 4 de 5

Nunca envíes contraseñas a través de Formularios de Google.



# tecDigital

## Diseño de sistema para el aseguramiento y control de la calidad de software del TEC Digital.

### Sección 4: consideraciones adicionales.

Describe alguna consideración o comentario adicionales que pueda tener.

4.1. Por favor describa alguna consideración o comentario adicional sobre la calidad del software del TEC Digital y su gestión, que de acuerdo con su punto de vista pueda ser tomado en cuenta.

Tu respuesta

ATRÁS

ENVIAR

Página 5 de 5

Nunca envíes contraseñas a través de Formularios de Google.

## REFERENCIAS BIBLIOGRÁFICAS

- Abran, A., Al-Qutaish, R., Desharnais, J. y Habra, N. (s.f.). *ISO-Based models to measure software product quality*.
- Al-Elaimat, A. y Al-Ghuwairi, A. (2015). Procedural Assessment Process of Software Quality Models Using Agility. *Actas de la International Conference on Intelligent Information Processing, Security and Advanced Communication, Estados Unidos de América*, 61. doi: 10.1145/2816839.2816884
- Al-Qutaish, R. (2009). *An Investigation of the Weaknesses of the ISO 9126 International Standard*. Trabajo presentado en la Second International Conference on Computer and Electrical Engineering. Dubai, Emiratos Árabes Unidos. doi: 10.1109/ICCEE.2009.83
- Arias, M. (2007). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *Revista Intersedes*, 6(10), 1-13.
- Asociación Española de Normalización y Certificación, International Organization for Standardization y Institute of Electrical and Electronics Engineers. (2005). *UNE/ISO/IEC 90003 Ingeniería del software: Guía de aplicación de la ISO 9001:2000 al software*. España: AENOR.
- Avella, C. y Gómez, J. (2011). Metodología integrada al proceso de construcción de software para aplicar inspecciones y pruebas. *Revista Virtual Universidad Católica del Norte*, 34, 1-18.
- Bianchi, T., Soares, D., y Romero, K. (2015). Quality Attributes of Systems-of-Systems: A Systematic Literature Review. *Actas de la Third International Workshop on Software Engineering for Systems-of-Systems, Italia*, 23-30. doi: 10.1109/SESoS.2015.12

- Bryman, A. (2008). Of methods and methodology. *Qualitative Research in Organizations and Management: An International Journal*, 3(2), 159-168. doi: 10.1108/17465640810900568
- Chacón, M. (2015). *Desarrollo de una metodología para el análisis, diseño, implantación y evaluación de soluciones e-learning de software libre en instituciones de educación superior*. (Tesis doctoral). Universidad de Alicante, España.
- Chacón, M. y Garita, C. (2010). *Desarrollo del Proyecto TEC Digital como Plataforma para la Integración de las TIC en la Docencia Académica*. Instituto Tecnológico de Costa Rica.
- Chawla, M. y Chhabra, I. (2015). SQMMA: Software Quality Model for Maintainability. *Actas de la 8th Annual ACM India Conference, India*, 9-17. doi: 10.1145/2835043.2835062
- Chiavenato, I. (2007). *Administración de recursos humanos: El capital humano de las organizaciones*. (8ª ed.). México: McGraw- Hill.
- Crosby, P. (1987). *La calidad no cuesta: El arte de cerciorarse de la calidad*. México: McGraw- Hill.
- Deming, W. (2000). *Out of the crisis*. Estados Unidos de América: MIT Press.
- Dutil, D., Rose, J., Suryan, W. y Thimot, B. (2010). Software Quality Engineering in the new ISO standard: ISO/IEC 24748 – Systems and software engineering Guide for life cycle management. *Actas de la Third C\* Conference on Computer Science and Software Engineering, Canadá*, 89-96. doi: 10.1145/1822327.1822339

Ejaz, R., Nazmeen, M. y Zafar, M. (2010). A Quality Assurance Model for Analysis Phase. *Actas de la National Software Engineering Conference, Pakistan, 1*, 1-4. doi: 10.1145/1890810.1890811

Elberzhager, F., Munch, J., Rombach, D. y Freimut., B. (2011). Optimizing Cost and Quality by Integrating Inspection and Test Processes. *Actas de la International Conference on Software and Systems Process*, 3-12. doi: 10.1145/1987875.1987880

Escobar, J. y Bonilla, F. (2009). Grupos focales: una guía conceptual y metodológica. *Cuadernos Hispanoamericanos de Psicología*, 1(9), 51-67.

Freund, J., Rücker, B. y Hitpass, B. (2014). *BPMN 2.0: Manual de referencia y Guía Práctica*. (4a ed.). Chile: BPMCenter.

Google. (2017a). *Formularios*. Recuperado de <https://www.google.com/intl/es/forms/about/>

Google. (2017b). *Google Drive*. Recuperado de [https://www.google.com/intl/es\\_ALL/drive/](https://www.google.com/intl/es_ALL/drive/)

Google Code. (2016). Rth-turbo. Recuperado de <https://code.google.com/archive/p/rth-turbo/>

Greene, J. (2007). *Mixed Methods in Social Inquiry*. Estados Unidos de América: Jossey Bass.

Grinell, R. y Unrau, Y. (2007). *Social Work research and evaluation: Foundations of evidence-based practice*. (8a ed.). Estados Unidos de América: Oxford University Press.

Gustavson, K. L. y Branch, R. (1997). *Survey of instructional development models*. (3ª ed.). Estados Unidos de América: Eric Clearinghouse on Information.

Hampp, T. (2012). A cost-benefit model for Software Quality Assurance Activities. *Actas de la 8<sup>th</sup> International Conference on Predictive Models in Software Engineering, Suecia*, 99- 108. doi: 10.1145/2365324.2365337

Hernández, M. (2007). *Cómo escribir una tesis*. Cuba: Escuela Nacional de Salud Pública.

Hernández, R., Fernández, C. y Baptista, M. (2014). *Metodología de la investigación*. (6ª ed.). México: McGraw-Hill.

Hernández, R. y Mendoza, C. (2008). *El matrimonio cuantitativo cualitativo: el paradigma mixto*. Trabajo presentado en el Sexto Congreso de Investigación en Sexología del Instituto Mexicano de Sexología, A. C. y Universidad Juárez Autónoma de Tabasco. México.

Hispanic America Software Testing Qualifications Board. (2017). *Bienvenidos al HASTQB*. Recuperado de <http://hastqb.org/>

Institute of Electrical and Electronics Engineers. (2017). *About IEEE*. Recuperado de <https://www.ieee.org/about/index.html>

International Organization for Standardization. (2015). *9000: Quality management systems-Fundamentals and vocabulary*. Suiza: ISO.

International Organization for Standardization. (2017). *About ISO*. Recuperado de <https://www.iso.org/about-us.html>

International Organization for Standardization y Institute of Electrical and Electronics Engineers. (2011). *ISO/IEC 12207 IEEE Std 12207-2008 Systems and software engineering – Software life cycle processes*. Suiza: ISO.

International Organization for Standardization y Institute of Electrical and Electronics Engineers. (2001). *ISO/IEC 9126-1 Software engineering – Product Quality Part 1: Quality model*. Suiza: ISO.

International Organization for Standardization y Institute of Electrical and Electronics Engineers. (2003a). *ISO/IEC 9126-2 Software engineering – Product Quality Part 2: External metrics*. Suiza: ISO.

International Organization for Standardization y Institute of Electrical and Electronics Engineers. (2003b). *ISO/IEC 9126-3 Software engineering – Product Quality – Part 3: Internal metrics*. Suiza: ISO.

International Organization for Standardization y Institute of Electrical and Electronics Engineers. (2004). *ISO/IEC 9126-4 Software engineering – Product Quality Part 4: Quality in use metrics*. Suiza: ISO.

International Organization for Standardization y Institute of Electrical and Electronics Engineers. (2011). *ISO/IEC/IEEE 24765 Systems and software engineering Vocabulary*. Suiza: ISO.

Kerlinger, F. (1983). *Investigación del Comportamiento: Técnicas y Metodología*. México: Interamericana.

Lavallée, M. y Robillard, P. (setiembre, 2011). *Do Software Process Improvements Lead to ISO 9126 Architectural Quality Factor Improvement?*. Trabajo presentado en la International Conference on Software Engineering. Szeged, Hungría. doi: 10.1145/2024587.2024592

Learn, Research, Network. (2017). *About.LRN*. Recuperado de <http://www.dotlrn.org/about/>

Lochmann, K., Ramadani, J. y Wagner, S. (2013). Are Comprehensive Quality Models Necessary for Evaluating Software Quality?. *Actas de la 9th International Conference on Predictive Models in Software Engineering, Estados Unidos de América*, 3. doi: 10.1145/2499393.2499404

Lochmann, K. y Goeb, A. (2011). A Unifying Model for Software Quality. *Actas de la 8th international workshop on Software quality, Hungría*, 3-10. doi: 10.1145/2024587.2024591

Madison, D. (2005). *Process Mapping, Process Improvement and Process Management*. Estados Unidos de América: Paton Press.

Miranda, J. (2014). *Propuesta de Plan Estratégico de Tecnologías de Información y Balanced Scorecard para la unidad TEC Digital del Instituto Tecnológico de Costa Rica* (Tesis de maestría). Instituto Tecnológico de Costa Rica, Cartago, Costa Rica.

Mehdi, M. y Aharpour, M. (2011). Issues in e-learning quality assurance. *Actas de la Second Kuwait Conference on e-Services and e-Systems, Kuwait*, 1, 1-5. doi: 10.1145/2107556.2107557

Moreno, I., Rodríguez, A., Snoeck, M., Moreno, R. Casas, G. y González, L. (2014). Directrices prácticas y métricas de calidad en la modelación de procesos de negocio: un caso de estudio. *Revista Cubana de Ciencias Informáticas*, 8(2), 1-18. Recuperado de <http://scielo.sld.cu/pdf/rcci/v8n2/rcci01214.pdf>

Nandakumar, R., Lal, A. y Parmar, R. (2014). *State of the Art in Software Quality Assurance*. *ACM SIGSOFT Software Engineering Notes*, 39(3), 1-6. doi: 10.1145/2597716.2597724

Nazir, S. (2005). Why Quality? ISO 9126 Software Quality Metrics (Functionality) Support by UML Suite. *ACM SIGSOFT Software Engineering Notes*, 30(2), 1-5.

Object Management Group. (2011). *Business Process Model and Notation (BPMB) Version 2.0*. Estados Unidos de América: OMG.

Object Management Group. (2017). *About OMG*. Recuperado de <http://www.omg.org/gettingstarted/gettingstartedindex.htm>

Pressman, R. (2010). *Ingeniería del software: Un enfoque práctico*. (7a ed.). México: McGraw-Hill.

Project Management Institute. (2013). *Guía de los fundamentos para la dirección de proyectos (guía del PMBOK)*. (5a ed.). Estados Unidos de América: Project Management Institute.

Project Management Institute y IEEE Computer Society. (2013). *Software Extension to the PMBOK Guide*. (5a ed.). Estados Unidos de América: Project Management Institute.

QaTestingTools. (2003). *Testitools*. Recuperado de <http://www.qatestingtools.com/testing-tool/testitools>

Rodríguez, D. y Valdeoriola, J. (2009). *Metodología de la investigación*. España: Eureka Media.

Ruiz, R. (2017). *Historia de la ciencia y el método científico: el análisis y la síntesis*. Recuperado de <http://www.eumed.net/libros-gratis/2007b/283/45.htm>

Saif, S., Khan, A. y Arif, F. (2010). An analysis of a comprehensive Planning framework for customizing SQA. *Actas de la National Software Engineering Conference*, 2. doi: 10.1145/1890810.1890812

- Sartori, G. (1984). *La política, lógica y método en las ciencias sociales*. México: Fondo de Cultura Económico.
- SeleniumHQ. (2017). *What is Selenium?*. Recuperado de <http://www.seleniumhq.org/>
- Shewhart, W. A. (1931). *Economic Control of Quality of Manufactured Product*. Estados Unidos de América: American Society for Quality Control.
- Sommerville, I. (2011). *Ingeniería del software*. (9a ed.). México: Pearson Educación.
- Stoshikj, M., Kryvinska, N. y Strauss, C. (2013). Project Management as a Service. *Actas de la International Conference on Information Integration and Web-based Applications & Services, Austria*. doi: 10.1145/2539150.2539171
- St-Louis, D. y Suryan, W. (octubre, 2012). *Enhancing ISO/IEC 25021 Quality Measure Elements for wider application within ISO 25000 series*. Trabajo presentado en la 38th Annual Conference on IEEE Industrial Electronics Society. Montreal, Canadá. doi: 10.1109/IECON.2012.6389400
- Silvestrini, M. y Vargas, J. (2017). *Fuentes de información primarias, secundarias y terciarias*. Recuperado de <http://ponce.inter.edu/cai/manuales/FUENTES PRIMARIA.pdf>
- Teddlie, C. y Tashakkori, A. (2009). *Foundations of Mixed Methods Research: Integrating Quantitative and Qualitative Approaches in the Social and Behavioral Sciences*. Estados Unidos de América: SAGE Publications.
- Teddlie, C. y Yu F. (2007). Methods sampling: A typology with examples. *Journal of Mixed Methods Research*, 1(1), 77-100.

- TestLink Development Team. (2014). *Testlink*. Recuperado de <http://forum.testlink.org/viewforum.php?f=11&sid=2ef504c5905fbcfd07015772eb65acf4>
- Testia. (2017). *Tarantula Agile Test Management*. Recuperado de <http://www.testiatarantula.com/>
- The LATEX Project. (2017). *LaTeX – A document preparation system*. Recuperado de <https://www.latex-project.org/>
- Tonon, G. (2011). La utilización del método comparativo en estudios cualitativos en ciencia política y ciencias sociales: diseño y desarrollo de una tesis doctoral. *Kairos Revista de Temas Sociales*, (27), 1-12.
- Turner, J. (2008). *The handbook of project-based management*. (3a ed.). Estados Unidos de América: McGraw-Hill Companies.
- Vicerrectoría de Docencia. Instituto Tecnológico de Costa Rica. (2010). *Memoria TEC Digital 2007-2010*. Instituto Tecnológico de Costa Rica: Cartago, Costa Rica.
- Wagner, S., Chulani, S. y Wong, B. (setiembre, 2011). *8<sup>th</sup> International Workshop on Software Quality (WoSQ)*. Trabajo presentado en la International Conference on Software Engineering. Szeged, Hungría. doi: 10.1145/2024587.2024591