



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**VICERRECTORADO DE INVESTIGACIÓN  
INNOVACIÓN Y TRANSFERENCIA TECNOLÓGICA**

**DIRECCIÓN DE POSGRADOS**

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
MAGÍSTER EN GERENCIA DE SISTEMAS**

**TEMA: ESTUDIO DE LA INCLUSIÓN DE PRUEBAS NO  
FUNCIONALES EN EL ASEGURAMIENTO DE LA CALIDAD DE  
SOFTWARE**

**AUTOR: FLORES RAMÍREZ, CARLOS JAVIER**

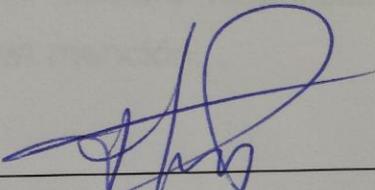
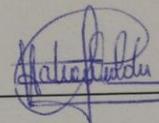
**DIRECTOR: ING. MBA FRANCIS SALAZAR PICO**

**SANGOLQUÍ**

**2014**

## CERTIFICADO

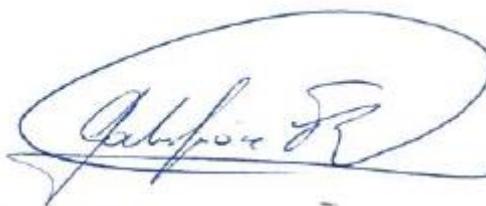
Certificamos que el presente proyecto titulado “Estudio de la Inclusión de pruebas no funcionales en el Aseguramiento de la Calidad del Software”, fue desarrollado en su totalidad por el Ing. Carlos Javier Flores Ramírez, bajo nuestra dirección.

  
\_\_\_\_\_  
Ing. Francis Salazar Pico, MBA  
\_\_\_\_\_  
Ing. Tatiana Gualotuña, MsC

## AUTORÍA DE RESPONSABILIDAD

El presente proyecto titulado “Estudio de la Inclusión de pruebas no funcionales en el Aseguramiento de la Calidad del Software”, ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado el derecho intelectual de terceros considerándolos en citas a pie de página y como fuentes en el registro bibliográfico.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance del proyecto en mención.

A handwritten signature in blue ink, appearing to read 'Carlos J. Flores Ramírez', enclosed within a large, stylized oval flourish.

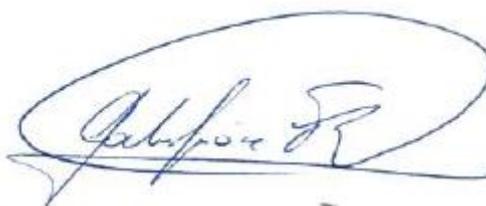
---

Ing. Carlos J. Flores Ramírez

## AUTORIZACIÓN

Yo, Carlos Javier Flores Ramírez, autorizo a la Universidad de las Fuerzas Armadas “ESPE” a publicar en la biblioteca virtual de la institución el presente trabajo “Estudio de la Inclusión de pruebas no funcionales en el Aseguramiento de la Calidad del Software”, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, julio del 2015

A handwritten signature in blue ink, appearing to read 'Carlos J. Flores Ramírez', enclosed within a large, loopy oval shape.

---

Ing. Carlos J. Flores Ramírez

## DEDICATORIA

El esfuerzo invertido en este proyecto académico quiero dedicarlo a mi familia que siempre ha sido mi motivación y fuerza para culminar con todos los proyectos emprendidos, quienes pacientemente han sabido entender mis momentos de ausencia con ellos, comprendiendo que han sido con un fin positivo.

A mi amada esposa Patricia y mi adorada hija Paula quiero decirles que este nuevo peldaño que se ha subido no hubiera sido posible sin su apoyo.

## **AGRADECIMIENTO**

Mi principal agradecimiento es a Dios por haberme permitido culminar este importante proyecto académico y poder aplicar los conocimientos adquiridos en mi labor profesional, después a los ingenieros Francis Salazar y Tatiana Gualotuña quienes con su guía han dado realce al contenido de este proyecto.

A la Universidad de las Fuerzas Armadas “ESPE” y sus colaboradores que han demostrado ser excelentes profesionales y poseer un alto espíritu de compromiso y apoyo al estudiante.

A mi familia por ser siempre un soporte y aliento en las decisiones que he tomado a lo largo de mi vida personal, académica y profesional. Mi esposa, mi hija, mis padres, mi hermano MUCHAS GRACIAS por su valiosa e incondicional ayuda en todo momento.

## Índice de contenidos

CERTIFICADO .....	i
AUTORÍA DE RESPONSABILIDAD .....	ii
AUTORIZACIÓN .....	iii
DEDICATORIA .....	iv
AGRADECIMIENTO .....	v
Índice de contenidos .....	vi
Índice de Ilustraciones .....	ix
Índice de Tablas .....	x
RESUMEN .....	xi
ABSTRACT .....	xii
1.                   CAPÍTULO I - INTRODUCCIÓN .....	1
1.1.                Antecedentes .....	1
1.2.                Justificación e Importancia.....	1
1.3.                Planteamiento del problema .....	4
1.3.1.             Formulación del problema a resolver .....	4
1.4.                Hipótesis.....	4
1.5.                Objetivo General .....	5
1.6.                Objetivos Específicos.....	5
2.                   CAPÍTULO II – MARCO TEÓRICO .....	6
2.1.                Evolución de la Calidad .....	6
2.1.1.             Inspección de la calidad .....	6
2.1.2.             Control de Calidad.....	7
2.1.3.             Aseguramiento de la Calidad .....	7
2.1.4.             Calidad Total y Mejora Continua.....	8
2.2.                Certificaciones y Normativas de la Calidad en Software .....	9
2.2.1.             ISO/IEC 25000 y sus divisiones.....	11
2.2.2.             Otras Normas .....	14
2.2.3.             Biblioteca de Infraestructura de Tecnologías de la Información (ITIL) .....	15
2.2.4.             ISTQB Mejores prácticas .....	17

2.2.4.1.	Importancia del proceso de pruebas .....	20
2.3.	Aspectos técnicos de Calidad del software .....	23
2.4.	Pruebas no funcionales .....	27
2.4.1.	Tipos de pruebas no funcionales .....	28
2.4.2.	Tipos de ambientes para ejecución de pruebas no funcionales .....	32
2.4.2.1.	Ejecución de pruebas no funcionales de forma local .....	32
2.4.2.2.	Ejecución de pruebas no funcionales en la nube .....	33
2.5.	Los servicios en la nube .....	35
2.5.1.	Tipos de Servicio en la nube .....	37
2.5.2.	Tipos de pruebas que pueden aplicarse .....	38
2.5.3.	Herramientas libres o propietarias .....	39
3.	CAPÍTULO III – INVESTIGACIÓN DE MERCADO .....	42
3.1.	Propuesta .....	42
3.1.1.	Metodología para ejecutar la propuesta .....	42
3.1.1.1.	Selección de la población, muestra y método de muestreo .....	43
3.1.1.2.	Selección de las técnicas de recopilación de la información, diseño de instrumentos .....	44
3.1.1.3.	Ejecución .....	46
3.1.1.4.	Análisis y reporte de resultados .....	46
3.1.2.	Exposición de la propuesta .....	47
3.1.2.1.	Las áreas de aseguramiento de calidad en las organizaciones .....	47
3.1.2.2.	Tipos de actividades dentro del aseguramiento de la calidad .....	49
3.1.2.3.	Sectores para los que se desarrolla software en nuestro medio .....	52
4.	CAPÍTULO IV .....	54
4.1.	Diseño Caso de Estudio .....	54
4.1.1.	Justificación .....	54
4.1.2.	Propósito .....	55
4.1.3.	Las unidades de caso de estudio .....	55
4.1.4.	Preguntas de investigación .....	55
4.1.5.	Proposiciones .....	56
4.1.6.	Conceptos y mediciones .....	56
4.1.7.	Métodos de colección de la información .....	57
4.1.8.	Métodos de análisis de datos .....	58

4.1.9.	Estrategia de selección de datos.....	58
4.1.10.	Estrategia de replicación.....	58
4.1.11.	Calidad del software, validez y confiabilidad.....	59
4.1.12.	Mejores prácticas a considerar de ISTQB.....	60
5.	CAPÍTULO V.....	62
5.1.	Conclusiones.....	62
5.2.	Recomendaciones.....	63
6.	BIBLIOGRAFÍA.....	64
7.	ANEXOS.....	66
7.1.	Encuesta.....	66
7.2.	Resumen de las respuestas en la encuesta.....	68
7.3.	Entrevista a experto.....	71
7.4.	Matriz de correlación.....	73

## Índice de Ilustraciones

Ilustración 1: Evolución de la Gestión de la Calidad Total .....	6
Ilustración 2: Aseguramiento de la Calidad .....	8
Ilustración 3: Las normativas de Calidad TIC en el software .....	9
Ilustración 4: La familia de normas ISO/IEC 25000 .....	12
Ilustración 5: Aspectos no funcionales de Calidad del software .....	24
Ilustración 6: Ambiente ejemplo de pruebas no funcionales de forma local.....	33
Ilustración 7: Ambiente ejemplo de pruebas no funcionales en la nube .....	35
Ilustración 8: Computación en la nube .....	36
Ilustración 9: Tamaños de las organizaciones por cantidad de empleados .....	47
Ilustración 10: Tamaños de los equipos de desarrollo de SW en organizaciones por número de empleados.....	48
Ilustración 11: Familiaridad del concepto de aseguramiento de la calidad en organizaciones de desarrollo de SW .....	48
Ilustración 12: Responsabilidad del aseguramiento de la calidad en las organizaciones de desarrollo de SW .....	49
Ilustración 13: Actividades en el ciclo de pruebas que realizan las organizaciones de desarrollo de SW en Ecuador.....	50
Ilustración 14: Organizaciones que realizan pruebas no funcionales.....	50
Ilustración 15: Pruebas no funcionales más comunes realizadas en las organizaciones de desarrollo de SW .....	51
Ilustración 16: Modo de ejecución de pruebas no funcionales en las organizaciones de desarrollo de SW .....	51
Ilustración 17: Sectores para los que se desarrolla software en Ecuador .....	53

## Índice de Tablas

Tabla 1 Normativas ISO .....	10
Tabla 2 Publicaciones de ITIL .....	15
Tabla 3 Aristas de los componentes técnicos de Calidad .....	24
Tabla 4 Metodología para ejecutar la propuesta .....	42
Tabla 5 Cuadro de valores Z.....	43
Tabla 6 Distribución de la población para enviar la encuesta .....	44
Tabla 7 Tipo de encuesta que se utilizará.....	45

## RESUMEN

El Aseguramiento de Calidad en las organizaciones de software se ha convertido en una preocupación constante, el mercado reclama productos de calidad que satisfagan sus necesidades incluso aún que las superen.

Esta constante presión del mercado obliga a que se regrese a ver los procesos que se ejecutan para elaborar un producto de software, entre estos procesos se encuentran las pruebas funcionales como las no funcionales, ambas necesarias para certificar un producto con normas mínimas de calidad. Las pruebas funcionales son las más conocidas por las organizaciones ya que lógicamente se debe probar la funcionalidad del requerimiento o cambio tecnológico, pero a su vez que pasa con las pruebas no funcionales o técnicas, cuál es su importancia para que sean consideradas dentro del proceso regular de pruebas.

Esto impulsa a realizar una investigación de mercado que ayude a conocer como se encuentran las organizaciones de nuestro medio con respecto a sus procesos de calidad en software, si tienen equipos de pruebas exclusivos, que tipos de pruebas realizan y donde las ejecutan. Es importante documentar si la calidad puede estar directamente relacionada con el tamaño de la organización para obtener aplicaciones de primer nivel.

Hoy en día existen varias alternativas para ejecutar pruebas no funcionales en las aplicaciones lo que hace falta es la habilidad de detectar problemas técnicos en los productos antes de su liberación a producción, ya que estas características del programa no siempre son fáciles de asimilar para los usuarios responsables del negocio.

### **PALABRAS CLAVES:**

**CALIDAD**

**SOFTWARE**

**PROCEDIMIENTO DE PRUEBAS**

**REQUERIMIENTOS**

**PRUEBAS NO FUNCIONALES**

## **ABSTRACT**

Quality Assurance in software organizations has become a constant concern, the market demands quality products that meets and exceeds their needs.

This constant pressure of the market obligates to look back at the processes involved to develop software products, among these processes are functional and non-functional tests, both necessary to certify a product with minimum quality standards. Functional tests are best known by organizations because logically the functionality of the requirement or technological change should be tested; on the other hand, what happen to the non-functional tests? What is its significance to be considered within the regular testing process?

This leads to perform a market research to know how organizations of the Ecuadorian Association of Software are working with their software quality processes, if they have an independent testing equipment, which types of tests are performed and where these tests are executed. It is important to document whether quality can be directly related to the size of the organization for top-level applications.

Today there are several alternatives for executing non-functional testing in software applications, but there's a lack of ability to detect technical problems in products before release them to production, as these features of the program are not always easy to understand for business responsible user.

### **KEY WORDS:**

**QUALITY**

**SOFTWARE**

**TESTING PROCEDURES**

**REQUIREMENTS**

**NON-FUNCTIONAL TESTING**

## **1. CAPÍTULO I - INTRODUCCIÓN**

### **1.1. Antecedentes**

En cualquier organización que realice desarrollo de software siempre será un costo adicional el mantener un equipo de personas dedicadas al aseguramiento de calidad que permitan certificar la calidad del software que se pretende liberar. Sin embargo, algo que se debe evaluar es el beneficio de obtener un software estable y con la menor cantidad de defectos posibles, para esto existen un sin número de pruebas, técnicas y procedimientos que se pueden ejecutar en diferentes etapas del ciclo de vida del desarrollo de un producto de software.

Para corroborar la importancia de las pruebas se puede hacer referencia a una de las mejores prácticas en gestión de tecnología del mercado, como lo es ITIL que en su etapa de Transición del Servicio presenta el proceso de Gestión de Validación y Pruebas, este proceso es una guía para poder ofrecer un mínimo de calidad en base a las expectativas acordadas con el cliente.

Uno de estos tipos de pruebas de software que se deben ejecutar son las pruebas no funcionales o técnicas, estas se encargan de complementar aspectos de calidad en el desarrollo de software y ayudan a garantizar características como: fiabilidad, robustez, usabilidad, mantenibilidad, portabilidad y desempeño.

El proyecto está enfocado en las pruebas no funcionales de software y como parte de esto se realizará un análisis de la forma en que algunas organizaciones de nuestro medio ejecutan pruebas técnicas como parte del aseguramiento de calidad de software.

### **1.2. Justificación e Importancia**

Dentro de cualquier modelo de desarrollo que se utilice para crear productos de software siempre se debe planificar el tiempo necesario para realizar pruebas, esto como un estándar de calidad, algo que se viene poniendo en práctica hace varios años. Ya sean modelos por etapas o modelos iterativos las pruebas de software se ajustan perfectamente en

cada uno de ellos; además cabe recalcar que hay pruebas para cada tipo de entregable que se crea durante el ciclo de vida de desarrollo de un producto de software, llámense estos documentos de requerimientos de usuario, modelos UML de casos de uso, liberaciones de productos de software, ambientes, etc.

Por lo general dentro de las pruebas que se planifican se da solo importancia a las pruebas funcionales con el usuario, éstas lo que permiten es revisar con el usuario final si los requerimientos que se detallaron en un inicio concuerdan con la funcionalidad que se tiene en el producto. Este tipo de pruebas que validan la funcionalidad también se las puede realizar con el equipo de pruebas previo a la cita con el usuario, ya que los analistas de pruebas conocen lo que debe hacer el producto de software y tienen los scripts de pruebas que certifican que lo que pidió el requirente está liberado en el sistema. Con esto se valida que el producto de software cumpla con las necesidades del cliente, sin embargo se está omitiendo otras pruebas que son tan importantes como las funcionales que son las pruebas no funcionales o técnicas.

Las pruebas no funcionales para el usuario final tal vez sean algo sin relevancia, ya que su principal interés es que el software desarrollado cumpla con sus necesidades funcionales, pero no considera que para que el software funcione correctamente y que en un futuro no muestre síntomas de degradación deben existir parámetros a ser evaluados previo a la liberación del producto al ambiente de producción. Aquí es donde el responsable técnico de las pruebas puede guiar al usuario para definir estos parámetros no funcionales, lo más probable es que el usuario si los conozca pero no sabe cómo comunicarlos. Entre estos parámetros por ejemplo se puede definir la concurrencia que tiene la aplicación, con esto se refiere a la cantidad de usuarios que pueden estar conectados a la vez en el sistema y esta información sí la conoce el usuario final; por lo tanto con el uso de herramientas se puede simular la conexión de esta "N" cantidad de usuarios y garantizar que cuando la aplicación ya se encuentre en ambiente de producción siempre cumpla con el rendimiento esperado.

Al respecto un ejemplo si sólo se valida la funcionalidad en un producto de software y se libera a producción. Al momento que empiecen a acceder al sistema varios usuarios el software puede empezar a presentar tiempos de respuesta extensos hasta llegar al punto de que no permita ingresar a más personas y el resto que ya se encuentra dentro del sistema no puede realizar ninguna actividad, seguramente desde el punto de vista del usuario se va a perder la credibilidad en la aplicación y esto repercutirá directamente en la calidad del producto entregado.

A partir de este escenario se considera que se pueden ejecutar las pruebas no funcionales de forma local con software especializado dentro de las organizaciones que realizan desarrollo de software o arrendar herramientas en la nube.

Por otro lado, como es de conocimiento general el uso del internet se ha masificado a nivel mundial y con ello la diversidad de servicios que se ofertan. Uno de estos servicios es la administración de negocios desde la nube, hecho que realmente ha bajado algunos costos en las organizaciones.

Un claro ejemplo de esto es el manejo de correo empresarial desde la nube, varias organizaciones a nivel local se han alineado a este modelo de servicios ya que les brinda un servicio robusto, seguro y con alta disponibilidad.

Para apoyar las actividades de pruebas existen parámetros de calidad y sus mediciones asociadas pueden ser útiles no solamente para evaluar el producto de software sino también para definir los requerimientos de calidad. Entre estos está la serie ISO<sup>1</sup>/IEC<sup>2</sup> 25000:2005 que es una guía para el uso de las nuevas series de estándares internacionales, llamados Requisitos y Evaluación de Calidad de Productos de Software(SQuaRE) y que reemplaza a dos estándares relacionados: ISO/IEC 9126 (*Software Product Quality*) e ISO/IEC 14598 (*Software Product Evaluation*).

---

<sup>1</sup> Organización Internacional de Normalización

<sup>2</sup> Comisión Electrotécnica Internacional

Otro de los estándares que serán utilizados en el desarrollo de este proyecto serán las mejores prácticas expuesta por ISTQB<sup>3</sup> sobre lo referente a pruebas en los productos de software.

### **1.3. Planteamiento del problema**

Dentro de cualquier modelo de desarrollo de software es obligación realizar procesos de aseguramiento de la calidad, entre otras cosas consiste en realizar pruebas funcionales y no funcionales; por lo tanto, se requiere investigar en nuestro medio el porcentaje de organizaciones que están familiarizadas con las pruebas no funcionales, si las ejecutan o no, conocer en qué ambiente y si cuentan con personal especializado para esto.

#### **1.3.1. Formulación del problema a resolver**

- ¿Las organizaciones que desarrollan software están familiarizadas con el concepto de calidad en el software?
- ¿Las organizaciones que desarrollan software tienen equipos de personas dedicadas exclusivamente al aseguramiento de la calidad de software?
- ¿Se puede considerar que las organizaciones que desarrollan software conocen sobre las pruebas no funcionales, su importancia y si las ejecutan?
- ¿Las pruebas no funcionales en nuestro entorno son ejecutadas en su mayoría de forma local o se las realiza en la nube?
- ¿El usuario final de un producto de software puede percibir la calidad en el servicio una vez que se han implementado pruebas no funcionales?
- ¿Cuáles son los beneficios e inconvenientes que puede tener una organización al ejecutar pruebas no funcionales en la nube?

### **1.4. Hipótesis**

No requiere hipótesis por ser una investigación exploratoria.

---

<sup>3</sup> Junta Internacional de Calificación de Pruebas de Software

### **1.5. Objetivo General**

Investigar la importancia y factibilidad de implementación que tienen las pruebas no funcionales en las organizaciones que cumplen con procesos de aseguramiento de calidad de software.

### **1.6. Objetivos Específicos**

- Documentar la cantidad de organizaciones de desarrollo de software que realizan pruebas de aseguramiento de calidad y entre ellas pruebas no funcionales.
- Identificar los diferentes tipos de pruebas no funcionales que realizan las organizaciones en nuestro medio.
- Evidenciar el grado de aceptación que tiene la nube como recurso para pruebas no funcionales en las empresas que realizan actividades de aseguramiento de la calidad en software.
- Valorar la incidencia que tiene en una organización el ejecutar pruebas no funcionales de software como parte de un proceso de validación y pruebas para certificar un producto de software.

## 2. CAPÍTULO II – MARCO TEÓRICO

### 2.1. Evolución de la Calidad

A continuación se presenta en la ilustración 1 como los procesos de calidad han ido evolucionando y acoplándose hasta conformar lo que hoy en día conocemos como Calidad Total y Mejora Continua:

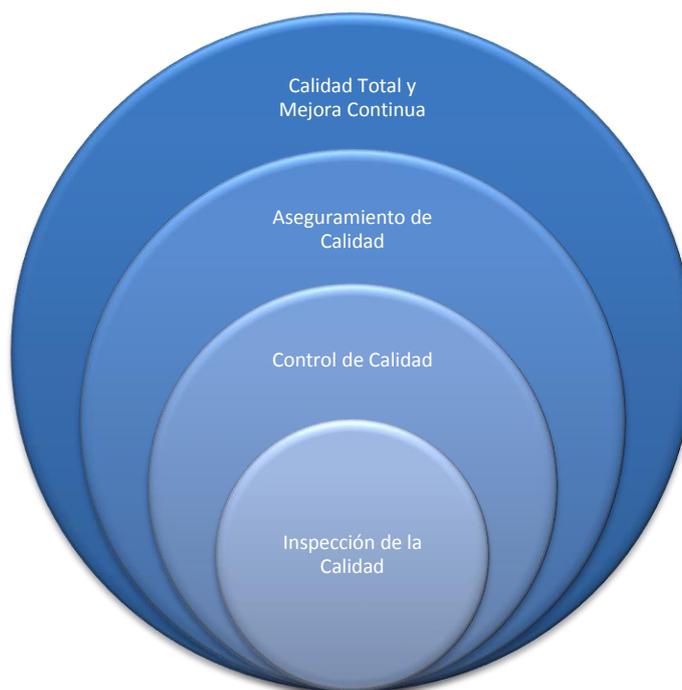


Ilustración 1: Evolución de la Gestión de la Calidad Total  
Fuente: Módulo Gestión de la Calidad y Productividad – MGS ESPE

#### 2.1.1. Inspección de la calidad

El proceso de inspección nació de la necesidad de las empresas que realizaban producción en masa, ellos requerían condiciones mínimas de calidad en sus productos y empezaron a identificar al final de su cadena de producción elementos con defectos o que no cumplan con todas las condiciones necesarias para la entrega al cliente final.

Esto dio como resultado la creación de oficinas de inspección que aporten con la identificación de los productos defectuosos y se los presente a los departamentos de producción.

La idea de calidad se fue entonces identificando con la ausencia de defectos y toda la actividad relacionada con la calidad era la inspección.

### **2.1.2. Control de Calidad**

Tomando como base el aporte de la inspección durante los procesos de producción, en esta etapa se toman nuevas consideraciones y se definen técnicas para mejorar el proceso de revisión de productos e identificación de defectos, como por ejemplo: técnicas de muestreo, comparación entre productos.

Las nuevas técnicas ayudan al departamento de producción a conocer la cantidad de defectos versus su producción total para ir identificando un índice de calidad en su línea de producción. Esto todavía no implica que se vaya a trabajar sobre las causas de los defectos.

### **2.1.3. Aseguramiento de la Calidad**

Esta etapa es fundamental porque los directivos de las empresas se dan cuenta que deben involucrar la solución de los defectos a todo el proceso de producción y no solo en la etapa final, esto permite identificar la causa de los defectos lo más pronto posible, lo que generará menos costos para su solución.

La calidad empieza a convertirse en algo más que una preocupación del productor y se crea la necesidad de incrementar su aplicación en los procesos de la organización.

En esta fase aparecen los sistemas y normas de calidad tales como: *Balance ScoreCard*<sup>4</sup>, ISO 9000.

El aseguramiento de la calidad como un concepto sencillo puede describirse como las actividades preventivas que se realizan para satisfacer las necesidades de un cliente. Se empezó a manejar este concepto desde que los productores buscaron cumplir con las necesidades o requisitos de sus clientes, y para hacer más fácil esta tarea empezaron a segmentar procedimientos de producción para poder realizar análisis y mediciones respecto a los resultados esperados, con esto se fue perfeccionando el

---

<sup>4</sup> Sistema de gestión que traduce la estrategia de negocios en indicadores

procedimiento para elaborar productos y al final obtener resultados que no solo cumplan con las necesidades del cliente sino que las sobrepasen.

Por lo tanto, el aseguramiento de la calidad en software envuelve a todo el proceso de desarrollo de software en sí, monitorea y mejora todos los procesos. Se asegura de que todas las normas y procedimientos acordados se sigan y asegura que los defectos detectados hayan sido resueltos.

Como se muestra en la ilustración 2, el aseguramiento de la calidad cubre todo el ciclo de vida de un proyecto a lo largo de sus etapas: Definición del alcance del proyecto, Especificación de funcionalidades y línea base, Construcción del producto, Entrega del producto al cliente.



Ilustración 2: Aseguramiento de la Calidad

Fuente: [http://www.cubicpharma.com/QA\\_Services/default.html](http://www.cubicpharma.com/QA_Services/default.html)

#### 2.1.4. Calidad Total y Mejora Continua

Esta filosofía de trabajo se fundamenta en que todas las personas que se involucran en el proceso son parte fundamental del mismo, todas las áreas de una organización deben estar alineadas para cumplir un objetivo común, para que este sea alcanzado con éxito y sobrepase los niveles de satisfacción del cliente.

Se lo interpreta también como una forma de gestionar eficazmente la calidad para minimizar errores y satisfacer a los clientes de una forma constante.

La idea de mejora continua lleva a pensar en ciclos de trabajo, esto se debe convertir en una cultura empresarial donde se analizan todas las

etapas de los procesos constantemente y se tiende a la reducción de costos y optimización de procesos con el fin de obtener un mejor producto o servicio para los clientes.

## 2.2. Certificaciones y Normativas de la Calidad en Software

Como se muestra en la ilustración 3, existen diversas normativas de calidad en el ámbito del software.

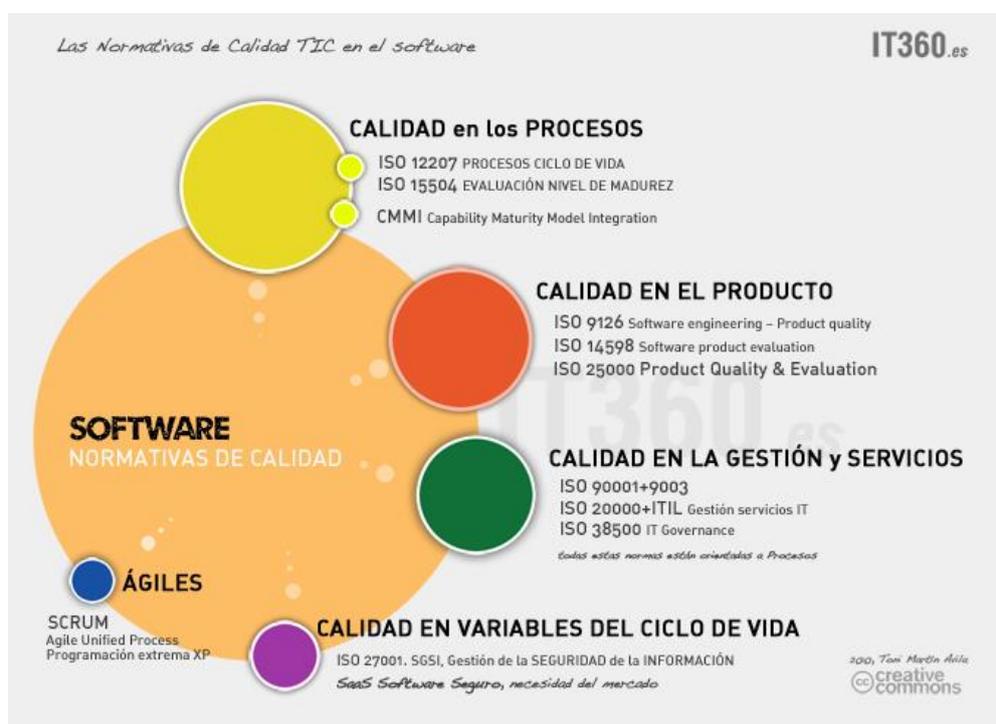


Ilustración 3: Las normativas de Calidad TIC en el software

Fuente: <http://www.it360.es/certificaciones-normativas-calidad-en-desarrollo-de-software.php>

Se los puede agrupar en sistemas de gestión, calidad en el producto de software y calidad en los procesos de software. Comúnmente las normativas son conjuntos de buenas prácticas que se aplican sobre el ciclo de vida de proyectos informáticos y que contribuyen a mejorar los factores de la calidad del software que se han expuesto con anterioridad. Existen multitud de modelos para la gestión de la calidad del software y otros sistemas y normas de gestión que se han aplicado sobre estos procesos, muchas de ellas con apéndices (normas específicas) para uno de los conceptos más importante

en el software como es la evaluación. En este sentido algunas empresas de desarrollo de software han implantado sistemas de gestión basados en las normas ISO 9001, ISO 27001 o ISO 20000 con alcances en los procesos de desarrollo y entrega, pero éstos quizás no son la mejor opción en el caso de que el corazón productivo de la organización sea únicamente el software.

Actualmente existen en el mercado diversas certificaciones y normativas relacionadas a normas ISO de las que se destacan:

Tabla 1 Normativas ISO

Norma	Detalle
<b>ISO 9001</b>	Establece los criterios para un sistema de gestión de la calidad y es el único estándar de la familia en el que pueden certificarse. Este puede ser utilizado por cualquier empresa, sea grande o pequeña, independiente además de su campo de trabajo.
<b>ISO/IEC 9003</b>	Ingeniería del software. Es una guía de aplicación de la ISO 9001:2000 al software, no es certificable. Es una norma de buenas prácticas para definir con más detalle los conceptos de software sobre los procesos de la organización.
<b>ISO/IEC 12207</b>	<i>Information Technology / Software Life Cycle Processes</i> , es el estándar para los procesos de ciclo de vida del software de la organización. Es la base para ISO 15504-SPICE.
<b>ISO/IEC 15504</b>	Conocida como SPICE - <i>Software Process Improvement And Assurance Standards Capability Determination</i> . Un conjunto de 7 normas para establecer y mejorar la capacidad y madurez de los procesos de las organizaciones, proporcionando los principios requeridos para realizar una evaluación de la calidad de los procesos. La definición de los procesos se realiza sobre ISO/IEC 12207.
	

Norma	Detalle
<b>ISO/IEC 9126</b>	Desarrolladas entre 1991 y 2001. <i>Software engineering – Product Quality</i> consta de 4 partes. Define las características de calidad del producto de software (parte 1), las métricas internas y externas (partes 2 y 3), y la calidad en uso, que explica cómo la calidad del producto está sujeta a las condiciones particulares de uso (parte 4).
<b>ISO/IEC 14598</b>	Desarrolladas entre 1999 y 2001. <i>Software product evaluation</i> , Evaluación del producto de software, la familia consta de 6 partes. Directamente relacionada con ISO 9126.
<b>ISO/IEC 25000</b>	La familia de normas 25000, conocida como SQuaRE, establece un modelo de calidad para el producto software además de definir la evaluación de la calidad del producto.

### 2.2.1. ISO/IEC 25000 y sus divisiones

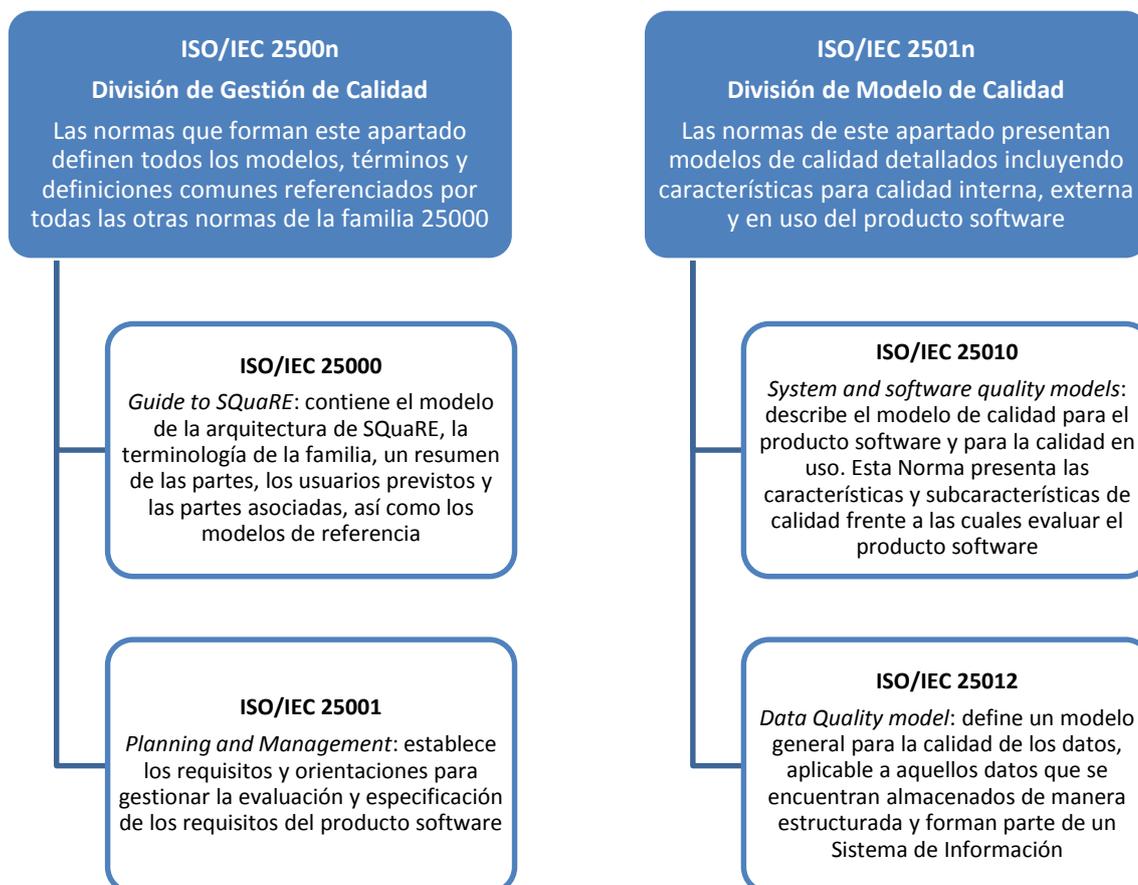
La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones como se muestra en la ilustración 4:

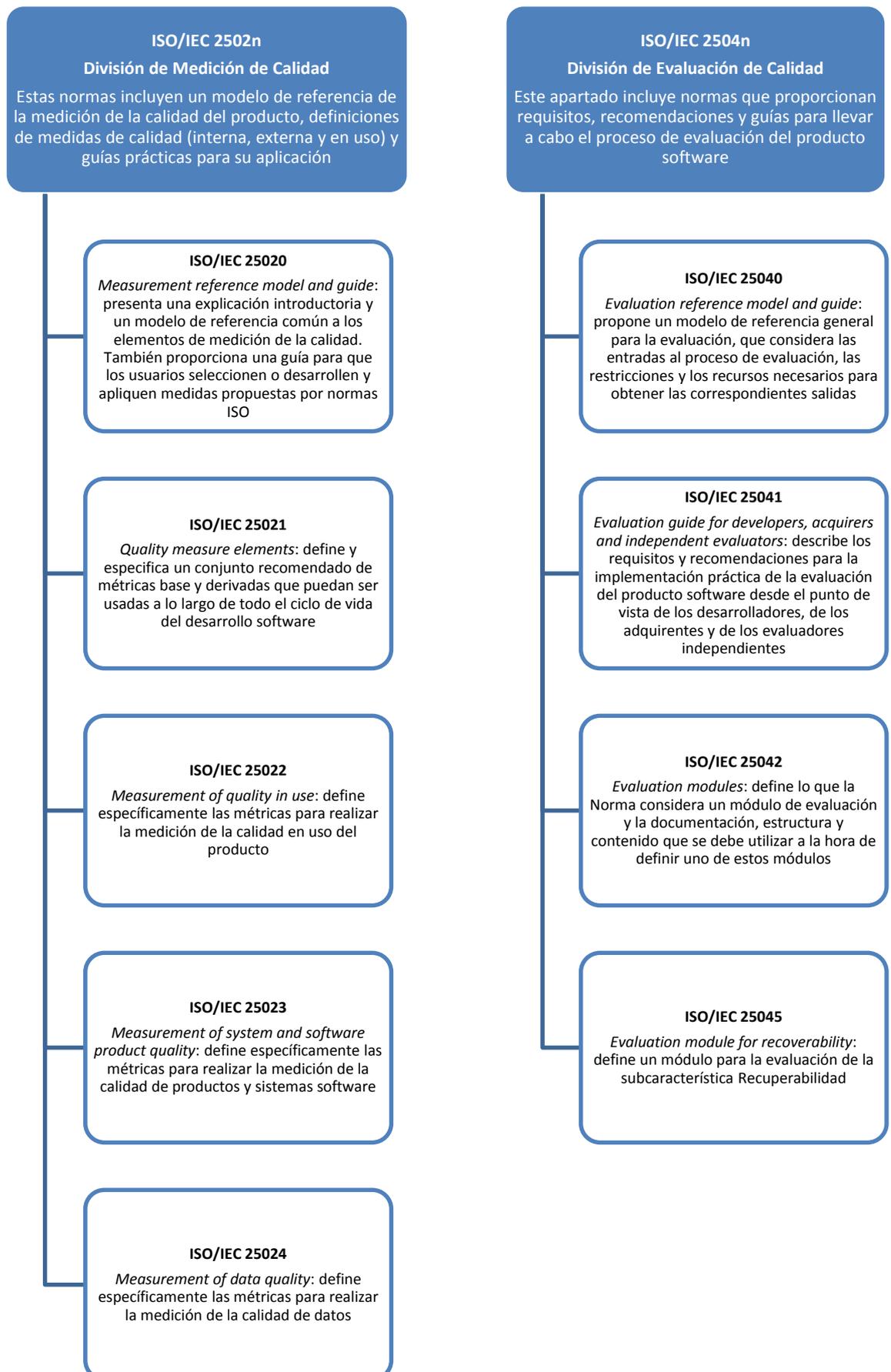


Ilustración 4: La familia de normas ISO/IEC 25000

Fuente: <http://www.iso25000.com>

A continuación el detalle de las divisiones de la norma ISO/IEC 25000:







### 2.2.2. Otras Normas

Adicional a las normas ISO tenemos otras normativas como las que se describen a continuación:

- **Capability Maturity Model Integration (CMMI)** se ha convertido mundialmente en un requisito para acceder a la exportación de servicios de software. La norma provee una guía para implementar una estrategia de calidad y mejorar los procesos de una organización que se dedica al desarrollo y/o mantenimiento de software. Dispone de un esquema de certificación creado sobre organismos privados (no normas ISO).
- **SCRUM** Un método sencillo y práctico para empezar a practicar calidad. Fabrica y gestiona el desarrollo en tres fases fundamentales: una breve fase de planificación, en la cual se realizan las labores básicas de una planificación: visión general del proyecto (estimación muy general, viabilidad del sistema) y construcción del *Product Backlog*<sup>5</sup> por un lado y por otro el desarrollo de la arquitectura al detalle; otra fase de desarrollo, en la cual tienen lugar los famosos

<sup>5</sup> Documento con las historias, requerimientos que necesita el usuario según la importancia que le asigna.

*Sprints*<sup>6</sup>, y una fase final de entrega y balance de los éxitos y fracasos logrados.

- **ITIL** es una Biblioteca de Infraestructura de Tecnologías de la Información, este marco de trabajo ha logrado una gran difusión debido al enfoque práctico para la gestión del servicio en las organizaciones. Los principales lineamientos tras el marco de trabajo aseguran que todos los esfuerzos tengan un objetivo común: entregar servicios TI que soporten los requerimientos del negocio entregando valor a la organización.

### 2.2.3. Biblioteca de Infraestructura de Tecnologías de la Información (ITIL)

ITIL ha definido cinco publicaciones del ciclo de vida del servicio. Las publicaciones exploran los procesos y conceptos de cada etapa del ciclo de vida y como estas interactúan con las otras. Cada etapa del ciclo de vida alimenta a las otras y con la etapa de Mejora Continua del Servicio todas interactúan. Cada una de las publicaciones provee lineamientos de un aspecto particular de la gestión del servicio y contribuye para formar un enfoque integrado.

Las cinco publicaciones de ITIL se desarrollan en el cuadro siguiente:

Tabla 2 Publicaciones de ITIL

Publicación	Detalle
<b>Estrategia del Servicio</b>	Esta cubre el núcleo del ciclo de vida, configura el enfoque estratégico para las actividades de la gestión del servicio
	

<sup>6</sup> Es un período de tiempo en el cuál se va a desarrollar el trabajo planificado, por lo general su duración es de dos a tres semanas y se lo puede ir ajustando. En el proceso de desarrollo van a existir varios Sprints.

Publicación	Detalle
<b>Diseño del Servicio</b>	Esta provee lineamientos sobre diseño y desarrollo de servicios acorde a los requerimientos del cliente y el enfoque estratégico
<b>Transición del Servicio</b>	Esta provee lineamientos para que los servicios definidos en la fase de Diseño del Servicio se integren en el entorno de producción y sean accesibles a los clientes y usuarios autorizados
<b>Operación del Servicio</b>	Esta cubre la gestión del día a día en la entrega del servicio, incluyendo una efectiva optimización y eficiencia
<b>Mejora Continua del Servicio</b>	Esta provee lineamientos sobre cómo responder a las cambiantes necesidades del negocio para ofrecer siempre servicios optimizados y con un mejor retorno de la inversión

Se profundizará en el **proceso de Validación y Pruebas**, proceso que permite asegurar que un servicio desarrollado no presente problemas al momento de liberarlo en producción, para esto las pruebas deben ser realizadas en un ambiente igual o similar al real. Los principales objetivos de este proceso son:

- Diseñar y mantener un ambiente, en lo posible igual al ambiente donde el servicio será puesto en producción.
- Conocer todas las funcionalidades que componen el servicio, así como mantener actualizada toda la documentación que requiera el servicio para su desarrollo.
- Validar que todos los acuerdos de calidad que se han acordado con el cliente siempre se mantengan a pesar de que el servicio cambie de versiones.

- Planificar y ejecutar una batería de pruebas que permita cubrir toda la funcionalidad del servicio.

Las ventajas que se obtienen al momento de implementar este proceso se resumen de la siguiente manera:

- Reducir la cantidad de incidentes registrados por problemas de incompatibilidad con software y hardware pre existente.
- Los requerimientos a la mesa de servicios se reducen a la par de la reducción de incidentes del servicio.
- Los errores y problemas conocidos pueden ser detectados y corregidos de una manera más fácil en el ambiente de pruebas que en el de producción.
- Se reducen los costos de solución de incidentes, ya que mientras más pronto se encuentre y solucione un error será más barato.
- El proceso de validación y pruebas no solo permite realizar el aseguramiento de la calidad del servicio que se está liberando, sino que también permite conocer la opinión de los usuarios respecto a nuevas funcionalidades del servicio y su usabilidad.

#### **2.2.4. ISTQB Mejores prácticas**

De manera general en las pruebas no funcionales el analista técnico de pruebas se centra en probar cómo funciona el producto de software a diferencia de su par funcional que se enfoca en probar qué hace la aplicación. Cabe recalcar que las pruebas técnicas pueden implementarse en cualquier momento durante el desarrollo de la aplicación, mientras más pronto mejor.

La falla en los aspectos no funcionales de un proyecto puede poner en riesgo la salida a producción de cualquier producto de software, por lo tanto es recomendable tratar de identificar la mayor cantidad de riesgos e irlos documentando en el plan maestro de pruebas.

Algunos de los aspectos en los que se debe poner mayor atención al momento de valorar las pruebas no funcionales son los siguientes:

- **Requerimiento de los interesados**, a menudo estos son escasos o inexistentes. En la mayoría de las ocasiones los clientes no comprenden que deben definir como requerimiento técnico y para esto es muy útil la ayuda del encargado de las pruebas técnicas ya que guía o interpreta las necesidades de los clientes.  
Para la definición del alcance de las pruebas no funcionales es recomendable involucrar a responsables de varios equipos como: gente del negocio, analistas de pruebas, personal de operaciones, personal de mantenimiento para analizar sus puntos de vista e ir especificando el documento final de requerimientos.
- **Adquisición y entrenamiento en herramientas requeridas**, para realizar las pruebas no funcionales es necesario que el analista de pruebas identifique las herramientas que va a utilizar, en caso de una compra debe incluir el tiempo de aprendizaje de la misma dentro del cronograma del proyecto para que no exista problemas con el avance de las actividades.
- **Requerimiento del ambiente de pruebas**, algunas de las pruebas técnicas requieren de un ambiente de pruebas similar al de producción para recopilar métricas que sirvan para analizar si los resultados reales son similares a los resultados esperados. Este hecho es algo complejo ya que depende del tamaño del ambiente en producción para poder replicarlo en ambiente de pruebas.  
En caso de no poseer un ambiente similar al de producción para pruebas lo que se debe hacer es tener un ambiente representativo que permita obtener resultados y estos sean proyectados en base a la diferencia de ambientes.
- **Consideraciones de la organización**, en algunas ocasiones las pruebas se deben realizar con integración de componentes (aplicaciones, servidores, bases de datos, etc.) fuera de la organización y para ello se depende de ciertos horarios para

ejecutar las pruebas, esto debe contemplarse en la planificación de las mismas.

- **Consideraciones de seguridad de los datos**, desde la etapa de planificación se debe definir las métricas de seguridad para los datos que van a ser utilizados, como por ejemplo la encriptación de datos para información sensible, sin embargo hay que tomar en cuenta hasta qué nivel se debe encriptar porque esto puede producir dificultades en las pruebas.

Políticas y leyes de protección de datos pueden imposibilitar la generación de la data de pruebas desde producción, por lo tanto crear los datos para las pruebas no es una tarea simple y requiere una actividad planificada como parte de la construcción de las pruebas.

El Aseguramiento de la Calidad de software es un proceso planificado que permite identificar y evaluar los defectos que puede tener un producto de software, no existe una etapa exclusiva donde se realice este conjunto de tareas, más bien este proceso debe existir durante todo el ciclo del desarrollo. Mientras más temprano se identifique los defectos su solución tendrá menos costo en recursos y esfuerzo.

El papel del aseguramiento de la calidad es ir supervisando las tareas que se realizan en el desarrollo para ir cumpliendo las necesidades de los clientes.

Existen tres aspectos muy importantes con relación al aseguramiento de la calidad del software según Karl Wieggers<sup>7</sup>:

- La calidad no se puede probar, se construye.
- El aseguramiento de la calidad del software no es una tarea que se realiza en una fase particular del ciclo de vida de desarrollo.

---

<sup>7</sup> Ingeniero de Software, autor de varios libros principalmente enfocados en requerimientos de software.

- Las actividades asociadas con el aseguramiento de la calidad del software deben ser realizadas por personas que no estén directamente involucradas en el esfuerzo de desarrollo.

Las actividades del aseguramiento de la calidad pueden partir desde la revisión de documentos, modelos UML, hasta la validación de los requerimientos del usuario en un producto de software estable.

Un proceso que afecta directamente al aseguramiento de la calidad del software es el control de cambios. Esto se puede presentar no solamente durante las fases de desarrollo sino que una vez el producto de software este puesto en producción puede tener modificaciones que representen cambios.

Las actividades del control de cambios son:

- Identificar el cambio
- Controlar el cambio
- Garantizar que el cambio se implemente adecuadamente
- Informar del cambio a todos aquellos que les afecte

Una herramienta externa que aporta a la calidad de un producto son las auditorías, ya que estas no solo evalúan a las actividades de desarrollo sino también las actividades de aseguramiento de la calidad.

Algo que es muy importante de aclarar es que la calidad no se refiere exclusivamente a un producto sin errores, además como se indica en uno de los manifiestos de ISTQB, no existe un producto con defectos cero. La especificación de la calidad debe ser más detallada y para esto debe existir una formalización mediante un proceso de pruebas maduro.

#### **2.2.4.1. Importancia del proceso de pruebas**

El tener un proceso ya establecido quiere decir que se referencia a buenas prácticas reconocidas en la gestión del desarrollo del software y cuyo resultado debe ser obtener un producto de alta calidad y con la menor cantidad de defectos posible.

En esto ISTQB menciona que las pruebas son un proceso más allá que una simple actividad durante el desarrollo de un sistema. Este proceso

empieza desde la planeación, diseño de casos de prueba, preparación para la ejecución y evaluación y finalmente las tareas de cierre. Se divide el proceso en las siguientes actividades: Planificación y Control, Análisis y Diseño, Implementación y Ejecución, Evaluación de criterios de salida y Reportes y Actividades de cierre.

- Planificación y Control
  - En la Planificación se mencionan las siguientes tareas:
    - Determinar el alcance y riesgos
    - Identificar los objetivos de las pruebas y sus criterios de salida
    - Determinar el enfoque, por ejemplo: técnicas de pruebas, cobertura de pruebas, equipos de pruebas.
    - Implementar el método o la estrategia de pruebas, planificación del período de tiempo para el desarrollo de las actividades a seguir.
    - Adquirir y programar recursos necesarios para las pruebas.
  - En Control de tienen las siguientes tareas:
    - Es una actividad continua que influye en la planificación. El plan de pruebas debe ser modificado en función de la información adquirida a partir del control.
    - El estado del proceso de pruebas se determina comparando el progreso logrado respecto al plan de pruebas.
    - Se miden y analizan resultados.
    - La evolución de las pruebas, la cobertura y el cumplimiento de los criterios de salida son objeto de seguimiento y documentación.
    - Se inician medidas correctivas.
    - Se preparan y toman decisiones.
- Análisis y Diseño, se indican las siguientes tareas:
  - Revisar las bases de pruebas. Las bases de prueba son la información que se necesita para iniciar el análisis y crear los casos de prueba. Básicamente esta es la documentación en que se basan los casos de prueba, como los requerimientos, diseño de especificaciones, análisis de riesgo del producto, arquitectura e interfaces.

- Identificar las condiciones de prueba.
- Diseñar las pruebas.
- Evaluar la posibilidad de pruebas de los requerimientos y el sistema.
- Diseñar el entorno de pruebas.
- Probar la infraestructura y herramientas en caso de ser necesario.
- Implementación y ejecución, se tienen las siguientes tareas:
  - Finalizar, implementar y priorizar los casos de prueba.
  - Desarrollar y priorizar procedimientos de prueba.
  - Verificar el entorno de pruebas
  - Verificar y actualizar la trazabilidad.
  - Ejecutar las pruebas de forma manual o automática.
  - Registrar resultados de prueba y análisis.
  - Informar y analizar incidencias con el objeto de establecer las causas.
  - Repetir actividades de prueba para confirmar la corrección.
  - Ejecutar pruebas de regresión, permite asegurar que los cambios no han revelado otros defectos o introducido nuevos defectos.
- Evaluación de criterios de salida y Reportes, se identifican las siguientes tareas:
  - Evaluar la ejecución de pruebas con respecto a objetivos definidos.
  - Evaluar los registros de pruebas como: resumen de las actividades de pruebas, resultados de prueba, comunicar criterio de salida.
  - Proporcionar información con el objeto de dar lugar a la decisión de si llevar a cabo pruebas adicionales.
  - Hacer resúmenes de pruebas para las personas interesadas en el proyecto.
- Actividades de cierre, se mencionan las siguientes actividades:
  - Recopilar datos de las actividades del proceso finalizado con el fin de consolidar experiencias, producto de soporte, hechos y números en general.
  - Cerrar informes de incidencias o generación de solicitudes de cambio para cualquier punto que permanezca abierto.

- Comprobar que entregables planificados han sido liberados y probados.
- Documentar la aceptación del sistema.
- Finalizar y archivar los productos de soporte de prueba, el entorno e infraestructura de prueba para un uso posterior, transferencia a operaciones.
- Analizar las lecciones aprendidas para futuros proyectos.
- Utilizar la información recopilada para mejorar la madurez del proceso de pruebas.

Adicional, dentro de varios marcos referenciales de mejores prácticas para el desarrollo de software se encuentra la etapa de validación y pruebas, esta etapa permite garantizar que los requerimientos iniciales del usuario se vean reflejados en el producto de software, teniendo al final un grado de satisfacción del usuario sobre la aplicación desarrollada lo que se puede traducir como calidad en un producto final.

Las pruebas no funcionales de software, hacen referencia al tipo de pruebas que complementan los procesos de calidad de un desarrollo de software, estas no tienen que ver con las definiciones de los requerimientos funcionales sino con las características de la aplicación como por ejemplo la robustez, la concurrencia, la mantenibilidad, el desempeño; además permiten validar que el software pueda cubrir necesidades intrínsecas del usuario.

### **2.3. Aspectos técnicos de Calidad del software**

De acuerdo a la norma ISO/IEC 9126 para evaluar la calidad de un producto de software se deben certificar las siguientes aristas: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

En la ilustración 5 se puede apreciar cada una de las aristas mencionadas anteriormente con cada uno de sus componentes para visualizar más a detalle que aspectos se deben validar en un producto de software:

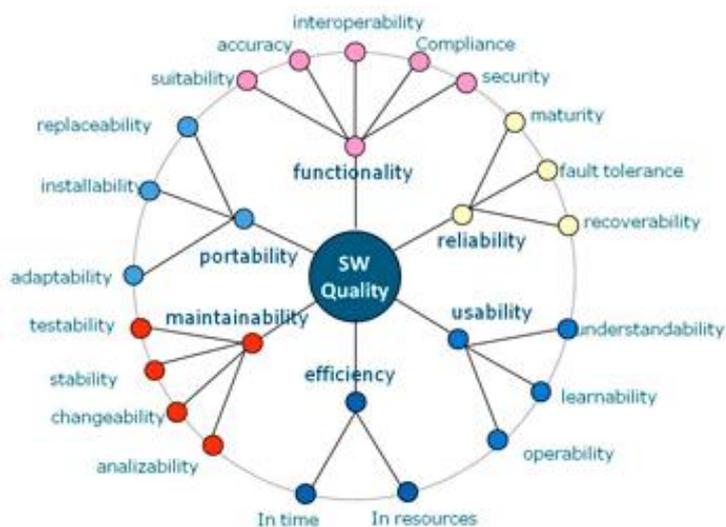


Ilustración 5: Aspectos no funcionales de Calidad del software

Fuente: <http://blog.optimyth.com/es/2012/12>

Tabla 3 Aristas de los componentes técnicos de Calidad

Arista	Detalle	Atributos
<b>Confiabilidad</b>	<p>En determinadas condiciones, el software mantendrá su capacidad o funcionalidad a lo largo de un período de tiempo.</p> <p><b><i>confiabilidad = calidad / tiempo</i></b></p>	<p><b>Madurez</b>, el software ha sido construido bajo metodologías probadas y ha pasado por varios procesos de pruebas.</p> <p><b>Tolerancia a fallos</b>, maneja controladores de errores.</p> <p><b>Recuperabilidad</b>, los datos mantienen su integridad para cuando la aplicación se restaura y no presenta errores de inconsistencias.</p>
<div style="border: 1px solid black; padding: 5px; display: inline-block; background-color: #4a7ebb; color: white;">                     Continúa  </div>		

Arista	Detalle	Atributos
<b>Usabilidad</b>	<p>Es importante entender porque los usuarios podrían tener dificultad usando el sistema. Para obtener este entendimiento es necesario primero conocer el término usuario, esto puede referirse a diferentes tipos de personas, como expertos en tecnología, niños y personas con alguna discapacidad.</p> <p>Algunas instituciones recomiendan que las páginas web sean accesibles para discapacitados, ciegos, deficientes visuales, personas con movilidad reducida, sordos, usuarios con discapacidad cognitiva. Es importante validar que las aplicaciones y sitios web sean usables para todos los usuarios.</p>	<p><b>Comprensibilidad</b>, atributo que afecta al esfuerzo requerido por el usuario para entender los conceptos lógicos y su aplicabilidad.</p> <p><b>Fácil aprendizaje</b>, atributo que se refiere al esfuerzo requerido por el usuario para aprender una aplicación.</p> <p><b>Operabilidad</b>, atributo que indica el esfuerzo requerido por el usuario para llevar a cabo tareas con eficacia y eficiencia.</p> <p><b>Atractivo</b>, es la capacidad del software de que sea llamativo para el usuario.</p>
<b>Eficiencia</b>	<p>Se refiere al comportamiento del sistema: funcionalidad y respuesta temporal.</p> <p>Características: El sistema requiere la utilización de un mínimo de recursos (por ejemplo tiempo de CPU) para ejecutar una tarea determinada.</p>	<p><b>En tiempo</b>, la aplicación responde en un tiempo corto cumpliendo resultados esperados.</p> <p><b>En recursos</b>, se ejecutan las transacciones utilizando sólo los recursos necesarios.</p>
		

Arista	Detalle	Atributos
<b>Mantenibilidad</b>	<p>Que permita ser analizable, facilidad de realizar modificaciones, estabilidad, facilidad de realizar pruebas.</p> <p>Características: Medida del esfuerzo requerido para realizar cambios en los componentes de un sistema.</p>	<p><b>Facilidad de pruebas</b>, no contiene componentes fuera de la arquitectura planteada.</p> <p><b>Estabilidad</b>, se ha certificado la parte funcional exitosamente.</p> <p><b>Facilidad de cambios</b>, el código es lo suficientemente claro y documentado como para realizar una modificación.</p> <p><b>Facilidad de análisis</b>, las estructuras de código son entendibles y están construidas de acuerdo a su diseño.</p>
<b>Portabilidad</b>	<p>Capacidad del software de ser transferido a un nuevo entorno (software, hardware, organización)</p> <p>Características: Fácil de instalar y desinstalar, parámetros.</p>	<p><b>Reemplazabilidad</b>, facilidad de reemplazo.</p> <p><b>Instalabilidad</b>, facilidad de instalación en cualquier ambiente definido.</p> <p><b>Adaptabilidad</b>, facilidad de coexistencia.</p>
		

Arista	Detalle	Atributos
<b>Funcionalidad</b>	Características adicionales a las definidas por el usuario en su documento de especificaciones funcionales.	<p><b>Idoneidad</b>, las funciones implementadas son adecuadas para el uso esperado.</p> <p><b>Exactitud</b>, las funciones desarrolladas presentan los resultados correctos.</p> <p><b>Interoperabilidad</b>, las interacciones con el entorno del sistema no presentan problemas.</p> <p><b>Conformidad</b>, el sistema cumple con todos los estándares y normas que se indicaron en la planificación.</p> <p><b>Seguridad</b>, las aplicaciones y los datos están protegidos contra accesos no deseados o daños involuntarios.</p>

## 2.4. Pruebas no funcionales

Básicamente cuando se realizan pruebas no funcionales el analista de pruebas se centra en cómo debe trabajar el producto de software más que en la funcionalidad pretendida. Estas pruebas se ejecutan en cualquier parte del ciclo de desarrollo, desde componentes pequeños a módulos que se deben integrar.

Hay pruebas muy importantes que consideran combinaciones entre hardware y software que deben ser realizadas cuando el producto de software se encuentra estable funcionalmente hablando, caso contrario se mete ruido en los resultados pretendidos para estas pruebas. Se toma en cuenta todos los componentes del ambiente para planificar las pruebas, servidores de aplicación, servidores de base de datos, anchos de banda en

red, clientes conectados, etc. Independiente del nivel de pruebas, estas pueden tener prioridades por riesgo o recursos disponibles en el proyecto.

Como se indicó anteriormente existen normas que aportan una guía para realizar pruebas no funcionales como la serie ISO 25000 que ha ido reemplazando a la ISO 9126, estas normas describen varias características que pueden ser de utilidad.

Es muy importante definir y documentar una estrategia de pruebas donde se irán identificando los riesgos típicos que tienen las características de calidad que se van a validar en el proyecto. Para estas características de calidad durante todo el ciclo de vida del proyecto se analizarán las herramientas con las que se debe hacer las pruebas, si requiere software adicional instalado, la documentación existente y los conocimientos técnicos del o los analistas de pruebas. Sin una planificación de la estrategia, el analista de pruebas no puede tener una adecuada gestión en la ejecución de las pruebas, tanto en efectividad de las mismas como en uso de recursos.

Algunas de estas pruebas, como por ejemplo, las pruebas de rendimiento necesitan una amplia planificación, varios recursos técnicos como herramientas y equipos dedicados, habilidades técnicas para manejar las herramientas, todo esto enmarca una gran cantidad de tiempo y esfuerzo.

#### 2.4.1. Tipos de pruebas no funcionales

Entre las principales pruebas no funcionales se describen las siguientes:

**Prueba de carga**, permite verificar el comportamiento del producto de software bajo carga mínima e incremental. Estas pruebas son realizadas para lograr objetivos de desempeño que por lo general están definidos en acuerdos de servicio.

Las pruebas de carga permiten medir tiempos de respuesta, tasas de rendimiento y niveles de utilización de recursos, además ayudan a identificar los puntos de quiebre de una aplicación. Estos puntos de quiebre deben estar bajo las condiciones de carga pico.

Los beneficios que pueden obtenerse son:

- Determinar el desempeño requerido para soportar las cargas pico que existirán en el ambiente de producción.
- Definir las características del hardware requerido.
- Evaluar la necesidad de un balanceador de carga.
- Detectar incidentes de concurrencia.
- Detectar errores de funcionalidad bajo carga.
- Recoger datos para escalabilidad y propósitos de planeación de capacidad.
- Ayudar a determinar la cantidad de usuarios ideal hasta que la aplicación empiece a degradarse.
- Apoya en definir la capacidad del hardware hasta que la aplicación empieza a degradarse.

**Prueba de rendimiento**, se considera a la rapidez con la cual un sistema o aplicación ejecuta una determinada función, también ayuda a medir la escalabilidad de una aplicación.

La prueba de rendimiento es una investigación técnica realizada para determinar o validar la capacidad de respuesta y características como la velocidad del producto que está siendo probado.

Los beneficios que pueden obtenerse son:

- Determinar si el usuario está de acuerdo con las características de rendimiento de la aplicación.
- Identificar diferencias entre el rendimiento esperado y la realidad de la aplicación.
- Soporta afinamientos, planificación de la capacidad y esfuerzos de optimización.

**Prueba de volumen**, es un subgrupo de la prueba de carga donde se evalúa el comportamiento de la aplicación con el procesamiento de grandes cantidades de datos o ficheros.

Este tipo de pruebas permiten calcular métricas como: Tiempos promedios entre fallos, Tiempos promedios para fallos.

**Prueba de estrés**, reacción de un producto de software a la sobrecarga o recuperación tras el retorno a una carga normal, su objetivo es revelar defectos en la aplicación sometida a condiciones de altas cargas.

Estos defectos pueden tener incluidos aspectos como incidentes de sincronización, condiciones de carrera o fugas de memoria. Las pruebas de estrés permiten identificar los puntos débiles en las aplicaciones y mostrar su comportamiento bajo extremas condiciones de carga.

Se tienen los siguientes beneficios:

- Determinar si la data puede corromperse debido a la sobrecarga de la aplicación.
- Proveer una estimación de cuanto más allá del límite de la carga puede soportar una aplicación antes de presentar errores y fallas aparte de mostrar lentitud.
- Permite establecer factores de monitoreo que ayudan a alertar sobre fallos inminentes.
- Asegurar que las vulnerabilidades de seguridad no aparezcan en condiciones de estrés.
- Ayudar a determinar qué tipos de fallas son las más importantes para planificar su solución.
- Determinar los efectos secundarios de hardware común o fallos de las aplicaciones de soporte.

**Prueba de estabilidad**, rendimiento en modo de operación continua, o por períodos prolongados de tiempo.

**Prueba de robustez**, reacción a entradas erróneas o datos no especificados. También la reacción a fallos de hardware o recuperación ante situaciones de desastre.

**Pruebas de cumplimiento**, cumplir normas y reglamentos tanto internos como externos.

**Pruebas de usabilidad**, el producto de software debe ser claramente estructurado, comprensible, fácil de aprender por parte del usuario.

**Pruebas de operabilidad**, esta es una amplia categoría de pruebas que mide el comportamiento del sistema bajo una variedad de condiciones. Un típico ejemplo son las pruebas bajo condiciones límite en las cuales un sistema es sometido a insumos altamente inesperados fuera del rango funcional para asegurar que la disponibilidad no está comprometida.

El reto para este tipo de pruebas es identificar los elementos críticos que deberían ser probados. El número de permutaciones de casos de prueba es muy alto y su alcance debe ser definido muy minuciosamente, por lo tanto deben ser casos definidos con una gran justificación.

Durante el diseño de la aplicación se debe conocer el ambiente en el cual será desplegada y se mantendrá operativa, por lo tanto el diseño debería aportar con patrones para que la aplicación no tenga problemas de operabilidad en cualquier ambiente que sea implementada. El diseño de la aplicación debería siempre contemplar los siguientes principios:

- Los datos y las transacciones nunca deben perderse o corromperse.
- Las condiciones de excepción deben ser capturadas y reportadas de una manera coherente.
- La aplicación debe recuperar de forma automatizada las excepciones una vez que estas hayan sido removidas.
- La aplicación debe mostrar la disponibilidad y salud de sus componentes de forma visible para poder solucionar los incidentes a tiempo.

**Pruebas de seguridad**, es un tipo de pruebas que permiten conocer si la aplicación es segura o no. Se valida si la aplicación es vulnerable a ataques, si alguien puede irrumpir en el sistema o ingresar sin autorización.

Este es un proceso para determinar que un sistema de información protege los datos y mantiene la funcionalidad de la forma prevista.

Las pruebas de seguridad además son ejecutadas para verificar si hay alguna fuga de información y respecto a esto se realizan encriptaciones en la aplicación y también se utilizan firewalls de software y hardware.

Los seis conceptos básicos de seguridad que necesitan ser cubiertos son: confidencialidad, integridad, autenticación, disponibilidad, autorización y no repudio.

#### **2.4.2. Tipos de ambientes para ejecución de pruebas no funcionales**

En el presente proyecto se consideran dos tipos de ambientes para la ejecución de pruebas no funcionales.

El primero es un ambiente de ejecución local, esto hace referencia a que las herramientas están instaladas en los servidores de la organización y el segundo es un ambiente de ejecución en la nube, se utiliza las herramientas que se encuentran alojadas en internet.

##### **2.4.2.1. Ejecución de pruebas no funcionales de forma local**

La ilustración 6 muestra como sería un ambiente de pruebas creado para ejecución de forma local o dentro de una organización.

Esto es, dentro de una misma red, ya sea LAN<sup>8</sup> o WAN<sup>9</sup>, se encuentran todos los servidores y equipos clientes que serán parte de la prueba, se tiene un servidor exclusivo para la o las herramientas de pruebas y estas a su vez, según el ejemplo, podrán conectarse a los respectivos servidores web, de aplicación, de base de datos, de correo, etc. según la infraestructura propia de cada organización.

Adicional a los servidores, están los equipos clientes que permiten hacer una simulación de conectividad de varios usuarios, ya que, cada equipo

---

<sup>8</sup> Local Area Network

<sup>9</sup> Wide Area Network

físico puede convertirse en N cantidad de usuarios virtuales si se requiere por ejemplo realizar una prueba de concurrencia.

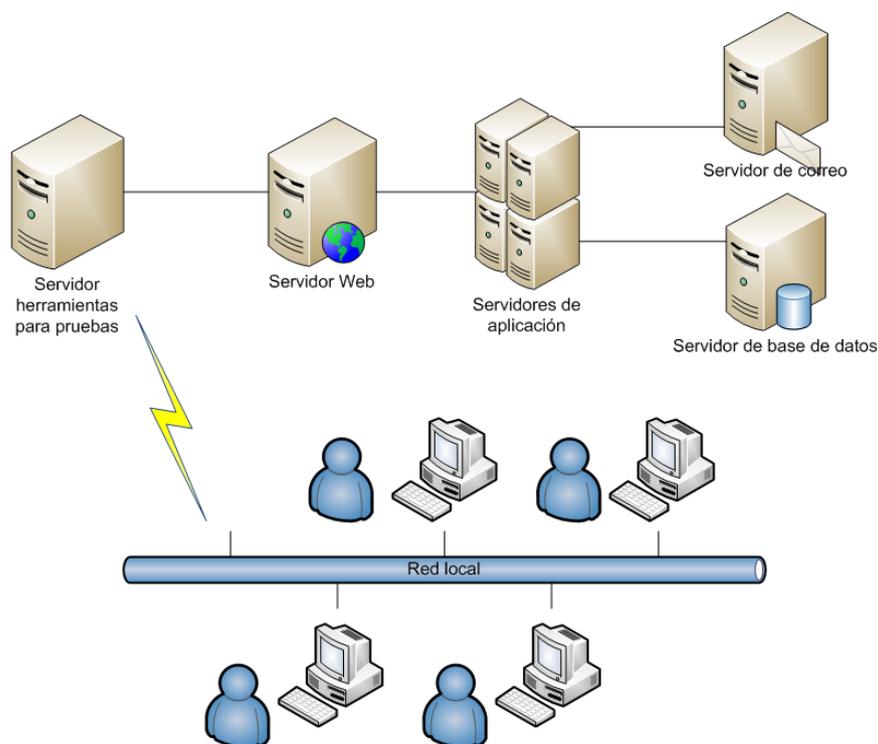


Ilustración 6: Ambiente ejemplo de pruebas no funcionales de forma local

#### 2.4.2.2. Ejecución de pruebas no funcionales en la nube

Computación en la nube, es un sistema gigante alojado en internet con centros de datos remotos donde se puede gestionar varios servicios o aplicaciones. La ventaja de esto es que las organizaciones ya no requieren instalar programas en sus ordenadores para utilizar determinado servicio, sino que por medio de un navegador y conexión a Internet puede acceder al servicio requerido.

Con esto se ofrece un uso eficiente de recursos como almacenamiento, uso de memoria, uso de procesador, ancho de banda, ya que solo se utilizan en el momento en que el usuario está conectado al servicio.

La computación en la nube ha tomado una gran importancia debido a la forma que se utilizan los servicios y como éstos son consumidos por los

clientes. Varias empresas de todo tamaño han migrado sus soluciones a la nube.

Existen varias herramientas alojadas en internet que permiten ejecutar pruebas funcionales y requerimientos técnicos del negocio. Pero no solo es suficiente contar con las herramientas en la nube ya que se requiere de cierto conocimiento técnico para poder utilizar dichas herramientas y ejecutar las pruebas.

Para ejecutar los servicios en la nube se debe tener presente los riesgos pre existentes y evaluar los pros y contras de migrar las pruebas a la nube.

Según un análisis hecho por la revista "*Testing Experience*", desde la perspectiva del proveedor de soluciones de software, independientemente si la organización tiene o no un equipo definido para aseguramiento de la calidad, existen claros beneficios económicos como de rendimiento al usar pruebas basadas en la nube.

Las compañías dedicadas a la realización de pruebas están obteniendo grandes ventajas del uso de la nube, especialmente para esos servicios que requieren una gran infraestructura para ejecutar actividades de pruebas, particularmente para pruebas no funcionales.

El siguiente diagrama ejemplifica cómo sería la ejecución de pruebas en la nube, por un lado están los servidores web, de aplicación, de base de datos, de correo, etc. del lado de la organización, mientras que desde un equipo cliente se accede a la herramienta que se haya contratado en la nube y se otorgan ciertos privilegios para que pueda acceder a los servidores que despliegan la aplicación que se requiere probar.

Una de las ventajas que se puede observar en el gráfico es que se obtiene mayor cantidad de usuarios virtuales que permitan realizar, como en el ejemplo anterior, alguna prueba de concurrencia, esto ayuda para no instalar clientes virtuales en los equipos clientes de la organización y que a su vez estos consuman recursos de la organización.

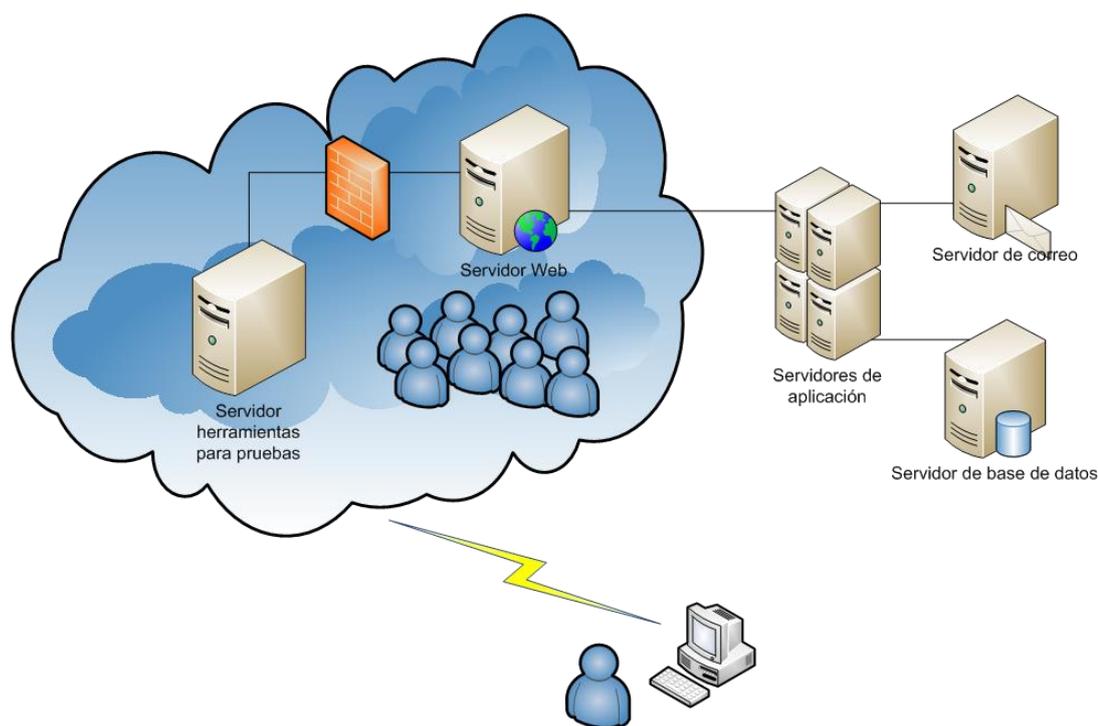


Ilustración 7: Ambiente ejemplo de pruebas no funcionales en la nube

Las pruebas no funcionales que se puedan realizar en cualquiera de los ambientes antes descritos sólo dependen de la suite de herramientas seleccionadas según la necesidad de cada usuario y del tipo de aplicación que se está validando, por lo tanto es un gran abanico que se abre para la adquisición de herramientas en una organización.

Existe gran variedad de herramientas licenciadas o gratis que permiten cubrir el total de las características de pruebas no funcionales anteriormente mencionadas, por lo que sería parte de otro análisis el identificar la mejor suite de herramientas. Para este caso puntual se trabajará con herramientas que pertenezcan a la organización y ayuden a solventar los objetivos planteados en el proyecto.

## 2.5. Los servicios en la nube

El modelo de la arquitectura de los servicios en la nube permite aprovechar todos los recursos disponibles en el internet, recursos que para

muchas organizaciones sería muy complicado adquirir para trabajar de forma local.



Ilustración 8: Computación en la nube

Fuente: <http://asdfgexc.blogspot.com/>

Este es un punto muy importante a considerar, ya que la utilización de estas herramientas es muy flexible porque depende de la etapa en la que se encuentre el desarrollo del producto de software, la periodicidad con la que se deban hacer las pruebas y el tipo de transaccionalidad de las mismas, convirtiéndose en un servicio de pruebas no funcionales bajo demanda.

La idea de utilizar servicios bajo demanda en organizaciones donde su negocio no es desarrollar software pero lo hacen para cubrir propias necesidades es vital porque no necesitan gastar cantidades excesivas en capacitar a su personal, ni adquirir licencias de herramientas, ni adquirir nueva infraestructura que van a ser utilizadas muy eventualmente.

Algo que también es importante considerar es que el ambiente que se utilice en la nube debe tener una configuración y características similares al ambiente local de la organización donde se pretende implementar el software, por ejemplo, el hecho de que la aplicación soporte pruebas de carga en la nube no garantiza que tendrá los mismos resultados en el

ambiente local que puede tener servidores con recursos más limitados y da una falsa expectativa de la calidad del producto.

Las modalidades más utilizadas de la computación en la nube son las siguientes:

- **Infraestructure as a Service (IaaS)**, se refiere a la contratación de procesamiento (CPU) y almacenamiento de disco en la nube, la parte de la administración de los recursos corre por cuenta del proveedor, como por ejemplo EC2 de Amazon.
- **Platform as a Service (PaaS)**, con referencia al modelo anterior este va un escalón más allá, ya que incluye la contratación de servidores de aplicación, base de datos que permitan la administración del propio contratante según sus necesidades, como por ejemplo Google App Engine.
- **Software as a Service (SaaS)**, se refiere a la aplicación que recibe el usuario final para su uso sin necesidad de ejecutar instaladores en su computador, como por ejemplo Google Docs. Este modelo es el que más acogida ha tenido ya que pequeños negocios o personas se han inclinado por este servicio ya que no necesitan adquirir infraestructura nueva para suplir sus necesidades tecnológicas.

Grandes empresas como LAN han preferido utilizar los servicios de correo de Google (GMAIL) para su necesidad de correo electrónico empresarial.

### 2.5.1. Tipos de Servicio en la nube

**Nube Pública**, es el servicio que se ofrece por medio de internet, este servicio ofrece la solución de infraestructura para las empresas sin embargo no tienen acceso a la administración de dicha infraestructura, existen divisiones lógicas que permiten que varias empresas puedan utilizar la misma infraestructura y solamente los usuarios autorizados ingresan a cada de su correspondiente aplicación alojada en la nube.

Una de las principales ventajas de utilizar la nube pública es que permite flexibilidad en infraestructura a costos relativamente bajos, entre estos

costos se puede indicar el mantenimiento, como la infraestructura le pertenece a un tercero él será quién se responsabilice por mantener siempre afinados los equipos de su nube.

**Nube Privada**, se refiere a la creación de una nube en el centro de cómputo de la organización, por lo tanto está asegurada por un firewall para que nadie del exterior sin autorización pueda acceder, se la conoce también como nube interna o nube de empresa.

Una de las herramientas que permite la creación de nubes privadas es la virtualización, esto ayuda disponer de infraestructura flexible en base a la infraestructura real de la organización otorgando el control total de su administración a la propia empresa.

Entre sus mayores ventajas se encuentra que por utilizar hardware propio se puede configurar niveles altos de seguridad y además se puede personalizar de tal manera que se pueda tener mejor desempeño de sus componentes.

**Nube Híbrida**, este tipo de servicio es la mezcla de los dos tipos mencionados antes, conjuga lo mejor de la nube pública y la nube privada.

Básicamente de la forma que se maneja una nube híbrida es la siguiente: la organización puede seleccionar sus aplicaciones críticas que requieren altos niveles de seguridad por tener información confidencial y las alojan en su nube privada y por otro lado existen las aplicaciones menos relevantes y estas son alojadas en una nube pública.

El problema de mantener las aplicaciones separadas en dos nubes radica en que la organización deberá estar pendiente de las operaciones que se realizan en cada una de las plataformas para garantizar la disponibilidad y funcionamiento de los servicios.

### **2.5.2. Tipos de pruebas que pueden aplicarse**

A nivel general en el mercado existen varias herramientas que permiten ejecutar pruebas no funcionales tanto en ambientes locales como en la nube, estas a su vez pueden ser gratuitas o licenciadas.

Dependiendo de los recursos económicos que posea la organización podrá alinearse a alguna de estas dos opciones de herramientas, sin embargo existen otros parámetros que se deben considerar como son: el soporte técnico, disponibilidad, frecuencia de actualización, innovación, interoperabilidad; por ejemplo: si van por el lado del software libre y se presenta algún problema con la herramienta y no se tiene el soporte necesario no se podrá avanzar con las pruebas planificadas con lo que se verá afectado la planificación del proyecto, o por otro lado, si van por la adquisición de herramientas propietarias y estas no tiene una actualización constante para soportar la creciente evolución de las arquitecturas, lo que se tiene es una herramienta inservible que no permitirá hacer ninguna prueba válida.

Antes de alinearse con alguna suite de herramientas es necesario realizar un análisis previo de costo beneficio para poder utilizar una herramienta que aporte a la organización y no se convierta en un riesgo para la ejecución de pruebas en el proyecto.

### **2.5.3. Herramientas libres o propietarias**

En el mercado se encuentra un gran número de herramientas que permiten la ejecución de pruebas no funcionales en ambientes locales como en la nube, algunas de estas son licenciadas sin embargo ponen a disposición del usuario versiones trial para que se interesen en consumir dichos productos, por otro lado existen herramientas de código abierto que en la mayoría de los casos poseen poderosas soluciones pero sin soporte y esto es algo necesario en el proyecto ya que cualquier incidente se necesita resolverlo a la brevedad posible.

Por lo antes expuesto se preferiría utilizar las herramientas propietarias tanto para las pruebas locales como para las pruebas en la nube.

Para las pruebas de carga que se ejecutarán de forma local se utilizará una de las suites de pruebas con más trayectoria en el mercado como lo es Silk de Borland, para las pruebas de seguridad se utilizará IBM Security AppScan. Para la ejecución de pruebas de carga en la nube se utilizará la

versión gratuita de la herramienta LoadStorm y para las pruebas de seguridad en la nube se utilizará HP Fortify on Demand.

No se realizará un profundo estudio sobre las herramientas propietarias u open source ya que no es el objetivo de este proyecto, por lo que nos basaremos solo en el procedimiento de la ejecución de cada una de ellas y los resultados.

A continuación una breve descripción de las herramientas que se utilizarán en el proyecto:

- **SilkPerformer**, es una herramienta propietaria con instaladores de la casa Borland que permite crear pruebas de carga robustas y reales en una amplia variedad de plataformas, entre ellas Oracle Forms, .Net, C#, Web Services, Aplicaciones Java, AJAX, Applets Java, entre otras, incluyendo las últimas tecnologías para web y móviles. Tiene una configuración relativamente sencilla, lo importante es adquirir o alquilar la cantidad necesaria de usuarios virtuales que se necesiten para la ejecución correcta de las pruebas.

Un componente adicional de esta herramienta es la suite para realizar análisis de resultados de una manera gráfica.

- **LoadStorm**, es una herramienta propietaria que maneja el modelo SaaS y sirve para realizar pruebas de carga desde la nube donde se puede simular un tráfico real en las aplicaciones, su configuración es sencilla no requiere mayor conocimiento de lenguajes de programación para la grabación de los escenarios de prueba. Posee una alta escalabilidad en el manejo de usuarios virtuales, ya que utiliza el cloud AMAZON EC2<sup>10</sup> con los data centers ubicados en varias ciudades alrededor del mundo.

Tiene un reportador versátil que permite revisar métricas de las pruebas de una manera dinámica.

- **HP Fortify on Demand**, es una herramienta de pruebas bajo el modelo SaaS que permite a las empresas realizar pruebas de seguridad en el software de una manera rápida, exacta y asequible, y sobre todo sin la

---

<sup>10</sup> Amazon Elastic Compute Cloud (EC2)

necesidad de instalar o administrar ningún software. Este servicio automatizado bajo demanda ayuda a las empresas de dos formas: 1) Garantizar la seguridad de las aplicaciones que se construyen, 2) Incrementar la velocidad y eficiencia en la construcción de software seguro durante su ciclo de vida.

La manera de utilizar la herramienta es muy sencilla ya que se sube el código de la aplicación a evaluar o se ingresa la dirección URL de la aplicación. Además la herramienta puede llevar a cabo pruebas estáticas o dinámicas y presentar informes con los resultados obtenidos e incluso incidentes relacionados.

- **IBM Security AppScan**, es una herramienta propietaria de la empresa IBM que permite a las organizaciones evaluar la seguridad de aplicaciones móviles y web, reforzar la gestión de programas de seguridad de las aplicaciones y alcanzar el cumplimiento normativo mediante la identificación de vulnerabilidades y la generación de informes con recomendaciones sobre arreglos inteligentes para facilitar la corrección.

### 3. CAPÍTULO III – INVESTIGACIÓN DE MERCADO

#### 3.1. Propuesta

Análisis de la implementación de pruebas no funcionales en las organizaciones de desarrollo de software del Ecuador afiliadas a la Asociación Ecuatoriana de Software (AESOFT).

##### 3.1.1. Metodología para ejecutar la propuesta

Al ser esta una investigación exploratoria, se utilizará el muestreo determinístico, ya que en base a la información obtenida con las técnicas recopilación se dará la orientación de la investigación.

#### Definición del Objetivo

Se requiere investigar el nivel de conocimiento y elaboración de pruebas no funcionales de software que existe en las organizaciones a nivel nacional que se dedican total o parcialmente al desarrollo de productos de software.

El propósito es netamente académico, sin fines de divulgar la información obtenida por cada organización consultada.

Tabla 4 Metodología para ejecutar la propuesta

Etapas	Métodos	Técnicas	Resultados
Fundamentación Teórica	Analítico sintético Inductivo deductivo Sistémico	Revisión bibliográfica Internet	Bases teóricas para conocer sobre pruebas no funcionales y sus ambientes de ejecución
Diagnóstico	Muestreo determinístico	Entrevistas Encuestas	Conocer sobre el desarrollo e implementación de pruebas no funcionales
Propuesta	Analítico sintético Inductivo deductivo	Tabulación de resultados	Análisis de los datos obtenidos en las encuestas y la entrevista

### 3.1.1.1. Selección de la población, muestra y método de muestreo

La población seleccionada son las organizaciones a nivel nacional que realicen desarrollo de productos de software y se encuentran afiliadas a la AESOFT. Al ser una población finita la muestra se calculará de la siguiente manera:

$$n = \frac{Z^2 * P * Q * N}{E^2 (N - 1) + Z^2 * P * Q}$$

Donde:

n= es el número de elementos de la muestra

N= número de elementos de la población o universo

P/Q= probabilidades con las que se presenta el fenómeno. Q=1-P (La encuesta abarca diferentes aspectos en los que estos valores pueden ser desiguales, se tomará el caso más adecuado, es decir, aquel que necesite el máximo tamaño de la muestra y eso ocurre cuando P y Q son 50.)

Z= valor crítico correspondiente al nivel de confianza elegido

E= margen de error permitido

Tabla 5 Cuadro de valores Z

Intervalo de confianza	Z	Nivel de significancia E
70%	1,04	30%
75%	1,15	25%
80%	1,28	20%
85%	1,44	15%
90%	1,64	10%
95%	1,96	5%
96%	2,05	4%
99%	2,58	1%

En base a la información obtenida:

Tabla 6 Distribución de la población para enviar la encuesta

Ciudad	Afiliados	Porcentaje
Quito	95	73,64%
Guayaquil	24	18,60%
Loja	4	3,10%
Manta	2	1,55%
Ambato	2	1,55%
Latacunga	1	0,78%
Portoviejo	1	0,78%
<b>Totales</b>	<b>129</b>	<b>100%</b>

$$N = 129$$

$$P=Q = 50$$

$$E = 10\%$$

$$Z = 1,64$$

$$n = \frac{1,64^2 * 50 * 50 * 129}{10^2(129-1) + 1,64^2 * 50 * 50} = 44,43$$

**Muestra = 44 encuestas**

El canal de comunicación con las organizaciones será la Asociación Ecuatoriana de Software (AESOFT).

No se define un método de muestreo debido a que se tiene acceso a la población entera y esta no es muy extensa (129 empresas afiliadas) por lo que se dirigirán las técnicas de recopilación de datos a todos los individuos.

### 3.1.1.2. Selección de las técnicas de recopilación de la información, diseño de instrumentos

## Encuesta

Esta técnica de recopilación permite conseguir un buen detalle de información sobre el objeto de la investigación en grandes poblaciones, permite conocer ambientes y detectar niveles de implementación de las pruebas en las organizaciones.

Las encuestas serán enviadas a los correos de los contactos registrados en la AESOFT de cada una de las organizaciones afiliadas.

Las respuestas de las encuestas se almacenarán automáticamente en la nube para el procesamiento de dicha información y posterior análisis.

Consideraciones importantes para realizar la encuesta:

- La población está ubicada en diferentes ubicaciones geográficas lo que dificultaría realizar varias entrevistas.
- Como se realiza un estudio preliminar se desea medir el conocimiento general antes de que se determine el rumbo del siguiente proyecto.
- Como herramienta adicional se ejecutará una entrevista con un experto para profundizar en ciertos temas.
- En la encuesta se manejarán preguntas abiertas y cerradas.

A continuación se indican ventajas e inconvenientes para el uso de la encuesta electrónica, dicha encuesta es la que será ejecutada.

Tabla 7 Tipo de encuesta que se utilizará

Tipo de encuesta	Ventajas	Inconvenientes
Por correo electrónico	<ul style="list-style-type: none"> <li>• Bajo costo</li> <li>• No requiere de la presencia del encuestador</li> <li>• Mayor dispersión de la muestra</li> <li>• Intimidación del encuestado</li> </ul>	Muchos cuestionarios no se llenarán

## **Entrevista**

Esta técnica dirigida permite obtener información más precisa y con más nivel de detalle que la encuesta y es realizada en un formato de preguntas y respuestas, uno de sus objetivos debe ser buscar la satisfacción del entrevistado. Hay que permitir la comodidad del entrevistado sin perder el control ni la objetividad de la entrevista.

Consideraciones importantes para realizar la entrevista:

- Seleccionar a la persona que se desea entrevistar, debe ser un experto que conozca bien del tema que se desea abordar.
- Preparar al entrevistado, esto es muy importante ya que la persona puede preparar mejor la información que desea compartir en la entrevista.
- Decidir el tipo de preguntas y su estructura, las preguntas deben perseguir un objetivo, ser claras y bien estructuradas para que la respuesta del entrevistado sea precisa.

### **3.1.1.3. Ejecución**

La ejecución de las técnicas de recopilación se la realizará en el siguiente orden: primero se difundirá por medio de canales electrónicos la encuesta para conocer un poco más de la población objetiva y una vez tabulados los datos se procederá con una entrevista a un experto para profundizar los temas que se requieran.

### **3.1.1.4. Análisis y reporte de resultados**

El análisis y reporte de los resultados principalmente comprende la descripción de los datos recopilados de las encuestas en la nube en base a una tabulación realizada.

Adicional se realiza una medición de las variables incluidas en la investigación presentándolas en forma gráfica para que genere una mejor percepción para el lector como también incluidas en un cuadro de forma numérica.

No se realiza una comparación de las relaciones observadas versus las relaciones teóricas debido a que la teoría indica que las pruebas técnicas

deben realizarse siempre independiente del tamaño de organización o del tipo de software, y con esta investigación lo que se pretende conocer es el nivel de aceptación de dichas pruebas en las organizaciones de nuestro medio.

### 3.1.2. Exposición de la propuesta

#### 3.1.2.1. Las áreas de aseguramiento de calidad en las organizaciones

Independiente del tamaño de las organizaciones el concepto de aseguramiento de la calidad debe ser promovido para todos los que realizan desarrollos de software, como se puede ver en la ilustración 9, los resultados de la encuesta muestran que dentro de todas las organizaciones encuestadas existe una diversidad respecto a la cantidad de personal que manejan. En su mayoría se encuentran las organizaciones con grupos de personas de 11 a 49 y las de 250 en adelante.

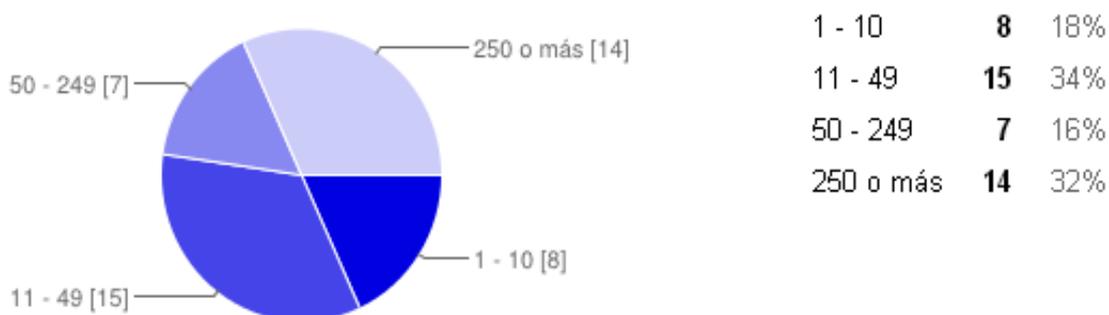


Ilustración 9: Tamaños de las organizaciones por cantidad de empleados

Y a su vez estas organizaciones poseen diferentes cantidades de personas realizando actividades de desarrollo de software como se muestra en la ilustración 10:

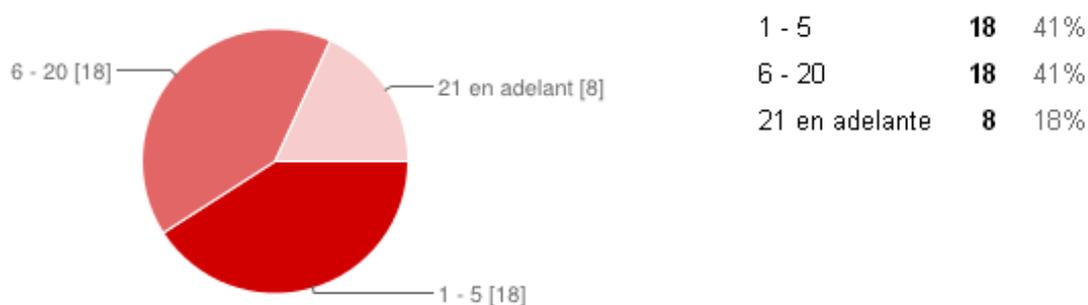


Ilustración 10: Tamaños de los equipos de desarrollo de SW en organizaciones por número de empleados

En base a la encuesta electrónica realizada a las empresas de la Asociación Ecuatoriana de Software se determina que la mayoría de organizaciones está familiarizada con el concepto de aseguramiento de la calidad del software como se muestra en la ilustración 11:

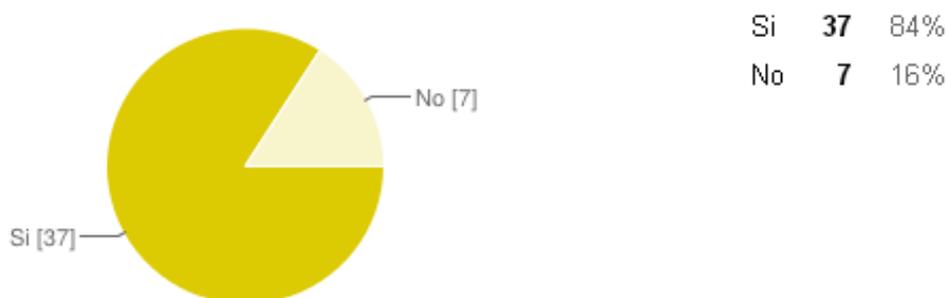
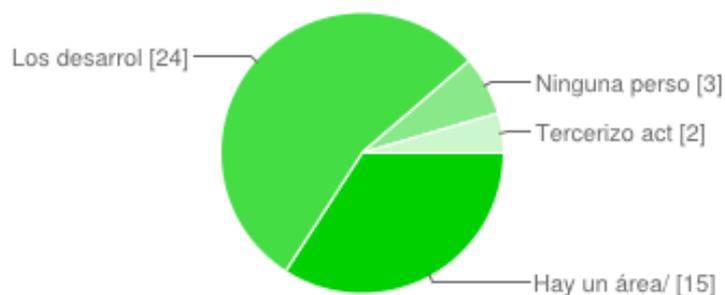


Ilustración 11: Familiaridad del concepto de aseguramiento de la calidad en organizaciones de desarrollo de SW

Sin embargo de todas las organizaciones consultadas solo un 34% de estas indica que posee un área de aseguramiento de la calidad claramente definida, y un 55% menciona que los mismos desarrolladores realiza actividades de aseguramiento de la calidad (Ilustración 5), hecho que no es muy recomendable, ya que como lo indica uno de los manifiestos de ISTQB, un buen analista de pruebas debe ser una persona independiente al desarrollo para tener más objetividad en la evaluación del producto de

software, ya que por lo general las personas tienden a pasar por alto sus propios defectos.



Hay un área/equipo de aseguramiento de calidad claramente definido	<b>15</b>	34%
Los desarrolladores también hacen actividades de aseguramiento de calidad	<b>24</b>	55%
Ninguna persona hace actividades de aseguramiento de calidad	<b>3</b>	7%
Tercerizo actividades de aseguramiento de calidad	<b>2</b>	5%

Ilustración 12: Responsabilidad del aseguramiento de la calidad en las organizaciones de desarrollo de SW

### 3.1.2.2. Tipos de actividades dentro del aseguramiento de la calidad

Dentro de las actividades en los procesos de pruebas que realizan las organizaciones de la Asociación Ecuatoriana de Software, las más comunes que se evidenciaron son las siguientes:

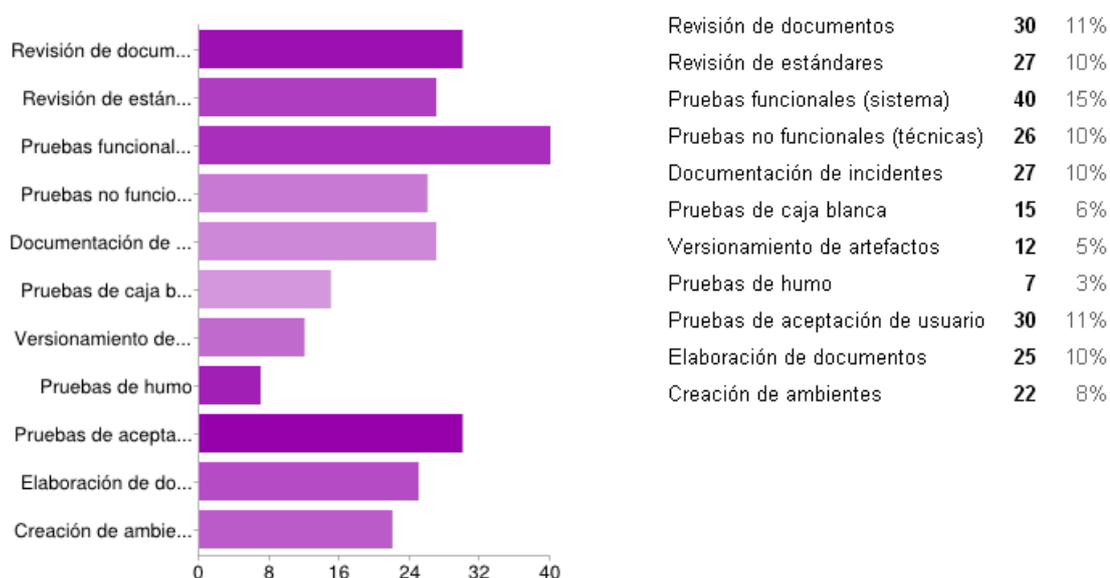


Ilustración 13: Actividades en el ciclo de pruebas que realizan las organizaciones de desarrollo de SW en Ecuador

En este estudio se muestra que de las 44 organizaciones encuestadas solo 26 realizan pruebas no funcionales durante el desarrollo del software a manera de un proceso formal, como se muestra en la ilustración 14:

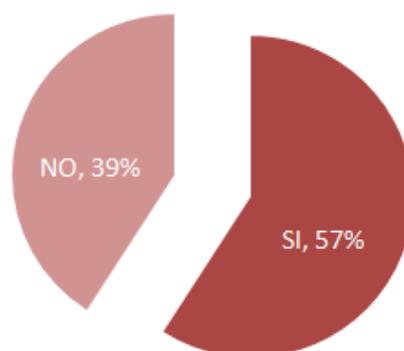


Ilustración 14: Organizaciones que realizan pruebas no funcionales

Sin embargo se conoce que algunas organizaciones hacen pruebas no funcionales de forma esporádica o bajo pedido, entre las pruebas más comunes están las que se muestran en la ilustración 15:

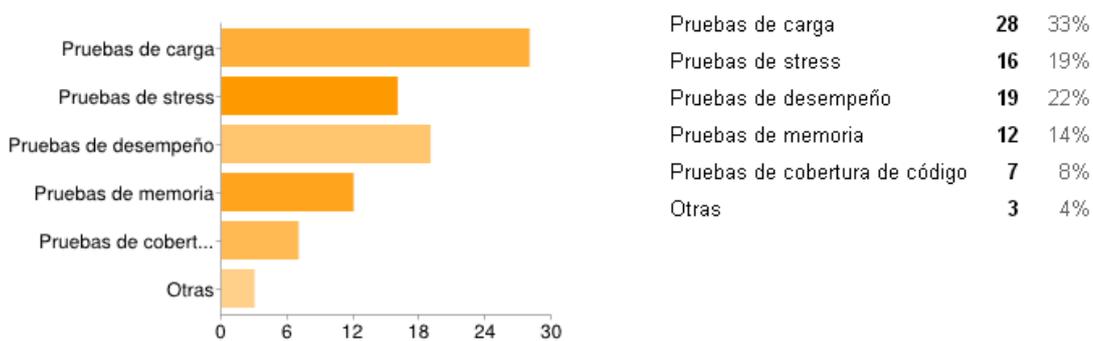


Ilustración 15: Pruebas no funcionales más comunes realizadas en las organizaciones de desarrollo de SW

Y de las pruebas no funcionales indicadas en la anterior ilustración en su mayoría están realizadas en medios locales, esto quiere decir con herramientas instaladas en las propias organizaciones, sean estas licenciadas o no. En la ilustración 16 se muestran los ambientes utilizados:

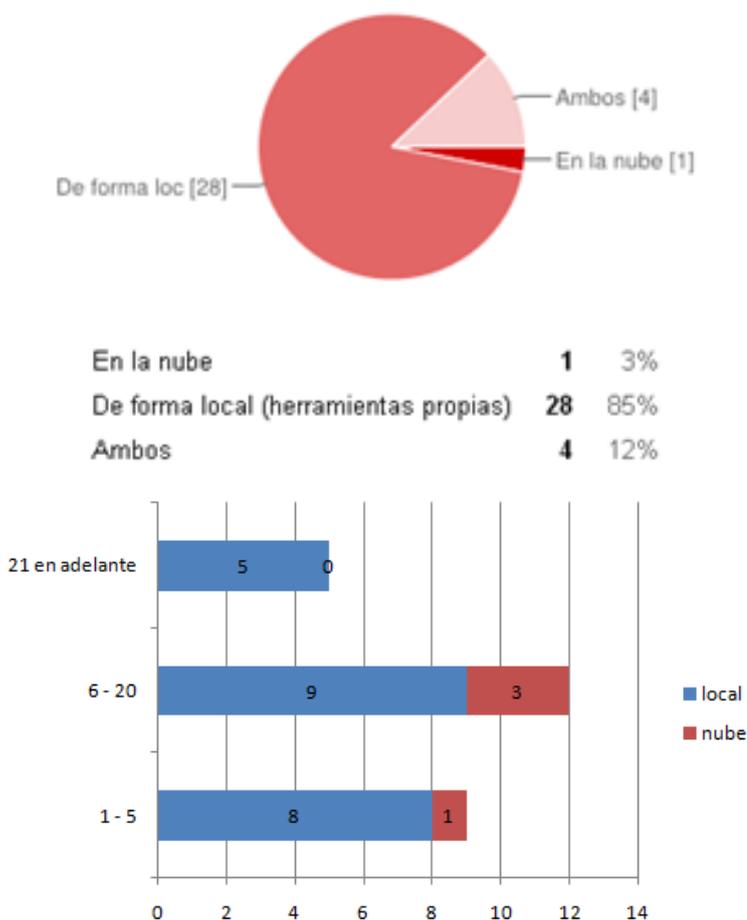


Ilustración 16: Modo de ejecución de pruebas no funcionales en las organizaciones de desarrollo de SW

Referente a la infraestructura que utilizan para las pruebas no funcionales se identificó lo siguiente:

- Empresas con 21 o más recursos en desarrollo, realizan pruebas solo de forma local con una infraestructura bien establecida con servidores independientes para: aplicaciones, base de datos, herramientas de pruebas, equipos de red para restringir ancho de banda.
- En empresas de 6 a 20 recursos de desarrollo, 9 realizan pruebas de forma local, 4 de ellas tienen una infraestructura bien definida con servidores independientes, mientras que las 5 restantes tienen las herramientas instaladas en los equipos propios de desarrollo. Respecto a la infraestructura de las 3 empresas que hacen pruebas en la nube, 1 empresa posee servidores independientes mientras que las 2 restantes tienen las aplicaciones distribuidas en equipos clientes.
- En empresas de 1 a 5 recursos de desarrollo, todas indican que su infraestructura es bastante limitada y tienen sus herramientas instaladas en equipos propios de desarrollo o computadoras clientes independientes para pruebas.

### **3.1.2.3. Sectores para los que se desarrolla software en nuestro medio**

Tomando la información publicada por la Asociación Ecuatoriana de Software, en nuestro medio existen los siguientes sectores para los que se desarrolla software, ilustración 17:

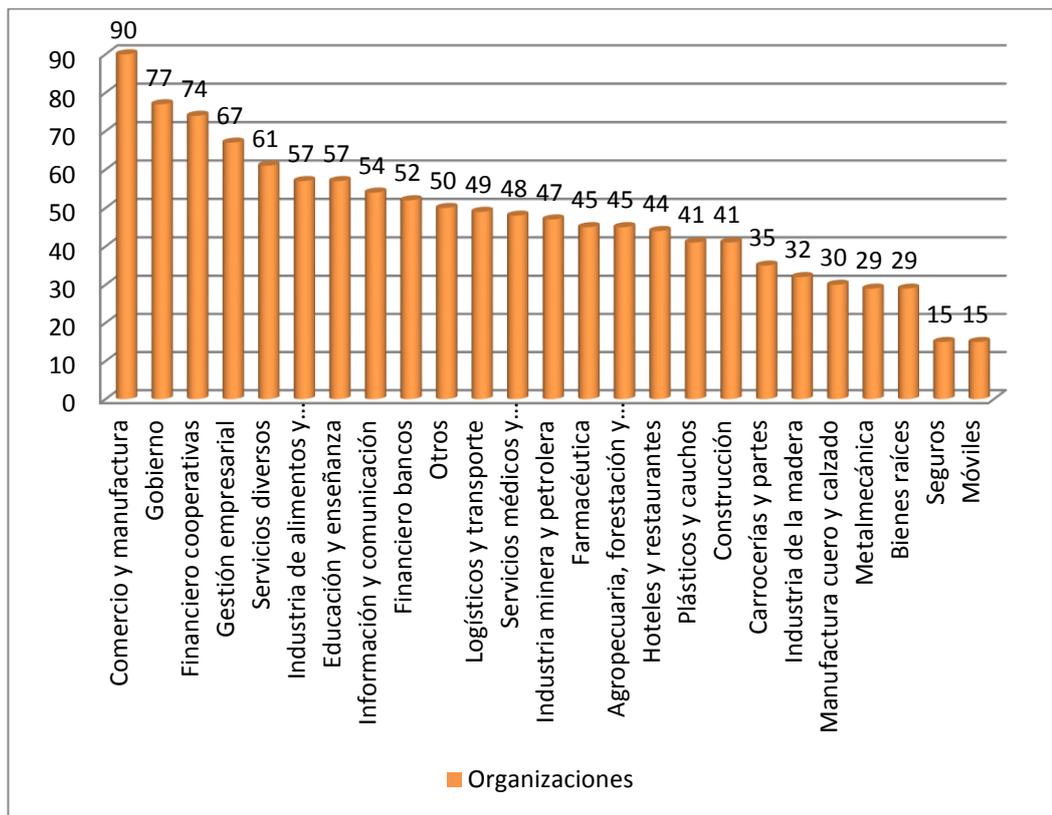


Ilustración 17: Sectores para los que se desarrolla software en Ecuador

Fuente: AESOFT

## **4. CAPÍTULO IV**

### **4.1. Diseño Caso de Estudio**

Una vez que se ha obtenido la información del Capítulo III se procede a seleccionar tres productos de software para el caso de estudio:

- Aplicación intranet para gestión de impuesto a los vehículos motorizados desarrollada en Java y desplegada sobre un servidor de aplicaciones JBoss 4.3
- Aplicación internet para consultas públicas de impuestos a los vehículos motorizados, desarrollada en Java y desplegada sobre un servidor de aplicaciones JBoss 4.3, y
- Aplicación móvil para consultas tributarias públicas, desarrollada con el framework PhoneGap y desplegada sobre un servidor de aplicaciones JBoss 6.2

A las aplicaciones anteriormente mencionadas se les aplicará dos tipos de pruebas no funcionales: de carga y de seguridad. La infraestructura y demás información será detallada en el segundo proyecto donde se ejecutarán las pruebas en entorno local como en la nube, los resultados que arrojen las herramientas se utilizarán para hacer comparaciones y poder determinar informes de costo beneficio de las pruebas no funcionales y su aplicación.

#### **4.1.1. Justificación**

Para las organizaciones es prioridad obtener mejores resultados con la menor cantidad de recursos invertidos, los mejores resultados van atados a brindar productos de calidad sean estos productos tangibles o intangibles. Los productos de software día a día son consumidos por cientos o miles de personas por lo tanto deben ofrecer ciertas características como fiabilidad, robustez, disponibilidad a sus consumidores, aquí radica la importancia de evaluar un software con pruebas técnicas o también conocidas como no funcionales.

Para que no exista un abanico muy grande de pruebas este proyecto se centrará en ejecutar pruebas de carga y de seguridad como un caso de estudio, ya que estas son las más conocidas en el medio.

#### **4.1.2. Propósito**

Evaluar los resultados que aporten las herramientas durante la ejecución de las pruebas técnicas en la nube como de forma local para sugerir a las organizaciones en qué ambiente se recomienda realizar las pruebas técnicas para su beneficio.

#### **4.1.3. Las unidades de caso de estudio**

Para tener una mejor comprensión del caso de estudio planteado se ha fraccionado su estructura en las siguientes unidades de análisis:

- Configuración de la aplicación intranet y herramientas, y ejecución de las pruebas en el ambiente propio de la organización.
- Configuración de la aplicación internet y herramientas, y ejecución de las pruebas en el ambiente propio de la organización.
- Configuración de la aplicación móvil y herramientas, y ejecución de las pruebas en el ambiente propio de la organización.
- Configuración de la aplicación intranet y herramientas, y ejecución de las pruebas en la nube.
- Configuración de la aplicación internet y herramientas, y ejecución de las pruebas en la nube.
- Configuración de la aplicación móvil y herramientas, y ejecución de las pruebas en la nube.
- Análisis de los resultados obtenidos.

#### **4.1.4. Preguntas de investigación**

En base a la justificación planteada para el caso de estudio a continuación se detallan las preguntas de investigación:

¿Las organizaciones que realizan pruebas no funcionales en la nube requieren una infraestructura amplia para realizar las pruebas?

¿Las organizaciones que realizan pruebas no funcionales en la nube requieren de personal técnico especializado para ejecutar dichas pruebas?

¿Es recomendable alquilar una herramienta en la nube versus adquirir la licencia de una herramienta para ejecutar pruebas de forma local?

¿Existe alguna brecha de seguridad al subir información sensible a la nube?

¿Cuál es la disponibilidad de las herramientas en la nube versus las herramientas instaladas en las organizaciones?

¿Los resultados de las pruebas son fáciles de obtener y son entendibles en cada uno de los tipos de ambientes en los que se ejecuta?

#### **4.1.5. Proposiciones**

En caso de que la ejecución de pruebas no funcionales en la nube no requiera de infraestructura extra en la organización sería atractivo para las mismas la utilización de la nube.

Si las herramientas disponibles en la nube tienen información técnica clara y precisa sobre su funcionamiento las organizaciones no requerirán personal técnico especializado en pruebas no funcionales.

Las organizaciones tienen la responsabilidad de garantizar la confidencialidad de la información de sus clientes y por ello requieren máxima seguridad con la información que están subiendo a la nube para que no sea utilizada por terceros.

Al tener resultados en la nube estos deberían estar al alcance de las organizaciones a cualquier momento, o cumplir con los acuerdos establecidos.

La información que se obtiene de las herramientas es entendible por el personal técnico que las utiliza y permite la interpretación a sus clientes.

#### **4.1.6. Conceptos y mediciones**

Para las pruebas de carga se ha definido las siguientes métricas:

- Tiempo de respuesta, es el tiempo que utiliza el servidor en responder a una solicitud. Esto es muy importante en el éxito de una aplicación.

- Velocidad de transferencia de la solicitud, esto se refiere a la cantidad de datos que pueden ser transferidos por la aplicación en cada segundo. Esta métrica afecta directamente al desempeño de la aplicación.

- Utilización de recursos, para analizar el desempeño de una aplicación, es esencial controlar el uso de recursos como procesador, memoria, uso de red, uso de disco en el servidor. El uso de los recursos debe estar bajo el umbral de los recursos de la aplicación.

- Métricas específicas del servidor, para identificar potenciales problemas que puedan limitar el desempeño de la aplicación se necesita monitorear algunas métricas específicas de cada servidor, esto dependiendo de la arquitectura de las aplicaciones. Para este caso se monitoreará los servidores de aplicación JBoss y el servidor de base de datos Oracle.

Para las pruebas de seguridad se utilizarán las siguientes métricas:

- Número y criticidad de las vulnerabilidades
- Cantidad de líneas de código analizado
- Complejidad ciclomática por función

#### **4.1.7. Métodos de colección de la información**

Para la colección de la información se mantendrán los tres principios propuestos por Verner:

- Utilizar varios orígenes de datos, en este caso de estudio puntual los orígenes dependerán de los ambientes donde se ejecuten las diversas pruebas.

- Crear un repositorio del caso de estudio, esto servirá para realizar las comparaciones de los resultados obtenidos de cada prueba.

- Validar los datos e identificar los cambios, lo que permitirá conocer las variaciones que existen entre los tipos de pruebas ejecutados y si los datos son consistentes.

La información de los resultados de las pruebas será obtenida por medio de reportes que proporcionan las propias herramientas a utilizar en cada uno de los ambientes, tanto local como en la nube, sobre los sujetos a examinar.

Estos datos serán tabulados en una hoja electrónica para posterior análisis de los datos y entregar un lineamiento a las organizaciones.

#### **4.1.8. Métodos de análisis de datos**

El análisis que se realizará es el estadístico descriptivo, esto permite por medio de gráficas tener un conocimiento de los datos que han sido recolectados de los diferentes ambientes.

Se organizarán los datos recolectados en base al perfil de prueba ejecutada para realizar la evaluación de los resultados de las herramientas.

Dependiendo de la cantidad y tipo de información que se pueda obtener, ya que se desconoce el resultado de algunas herramientas, se propondrá otros métodos para ir mejorando el análisis de los datos.

#### **4.1.9. Estrategia de selección de datos**

La selección de datos de los resultados en este proyecto de pruebas no funcionales será total ya que se debe comparar toda la información que arroje cada herramienta para demostrar que independiente del ambiente en que se trabaje se pueden obtener los mismos resultados técnicos.

Los datos de ingreso que se utilizarán en la ejecución de las pruebas serán categorizados según los lineamientos más comunes del negocio y se tomará una muestra de ellos para realizar la configuración del grupo de datos que corresponde a los scripts de las pruebas.

#### **4.1.10. Estrategia de replicación**

Debido a la necesidad de informar sobre las mejores prácticas y hallazgos obtenidos en la elaboración del proyecto se realizarán las siguientes tareas como estrategia de la replicación:

- Identificar la audiencia objetivo a la que le debe llegar la información de los resultados obtenidos. Por ejemplo si va dirigido a los directivos de la organización, al gobierno, a personas integrantes de alguna área técnica o de negocio, a comunidades educativas.

- En base a la audiencia identificada se debe seleccionar que tipo de información se debe presentar. Los reportes estarán estructurados en base a las mejoras prácticas y recomendaciones de los marcos de trabajo.

- Cuando presentar la información, para este caso particular se lo hará una vez finalizado el proyecto.

- Se entregará como parte del segundo proyecto un informe con el detalle de los resultados obtenidos del caso de estudio.

En este punto cabe mencionar que se mantendrá estricta confidencialidad y ética profesional en el manejo de la información de la organización donde se va a ejecutar las pruebas no funcionales.

#### **4.1.11. Calidad del software, validez y confiabilidad**

Los productos de software seleccionados han pasado primero por varias etapas de pruebas funcionales, tanto en pruebas de sistema como pruebas de certificación con el usuario final para garantizar que es un producto válido y limpio para arrancar con las pruebas técnicas, con esto se espera que no se generen incidentes fuera del esquema planteado.

Para las pruebas no funcionales se tiene planificado ejecutar una metodología en cascada para avanzar etapa por etapa con documentación requerida y otras tareas necesarias como las siguientes:

- En análisis, se levanta el plan de pruebas que se implementará donde se define a detalle el alcance, el tipo de pruebas a ejecutar, los ambientes establecidos, las herramientas a utilizar, roles requeridos, tareas no contempladas, y un cronograma de trabajo.

- En diseño, se escribirán los casos de prueba que serán utilizados a lo largo de las pruebas no funcionales.

- En la construcción, se configurarán los ambientes necesarios para las pruebas y se crearán los scripts para las herramientas.

- En la etapa de pruebas, se ejecutarán los scripts de las pruebas en cada uno de los ambientes y en cada uno de los productos seleccionados. Además se utilizará una herramienta para registrar los incidentes que aparezcan durante la etapa de pruebas.

- En el cierre, se emitirá el informe de pruebas concerniente al proceso de pruebas realizado.

Al culminar esta etapa de pruebas no funcionales se estará culminando con una etapa más grande que corresponde al cierre de la certificación de un producto de software donde se puede garantizar que una aplicación puede liberarse a producción.

A partir de este punto, y según el informe emitido, queda recomendar el monitoreo de la aplicación durante una etapa de estabilización en producción, donde la aplicación se encuentra en una instancia sola para su posterior paso a convivencia con otras aplicaciones.

#### **4.1.12. Mejores prácticas a considerar de ISTQB**

En el documento plan maestro de pruebas se planificarán las actividades que se llevarán a cabo a lo largo del segundo proyecto, donde se indicará que pruebas se ejecutarán sobre los productos de software seleccionados.

Como puntos importantes se realizarán las siguientes actividades:

- Mantener una reunión con los usuarios responsables de las aplicaciones, personal de operaciones, personal de mantenimiento para comprender de mejor manera el alcance que se le debe dar a las pruebas y las restricciones que se tengan en los diferentes ambientes.
- Configurar las herramientas seleccionadas previamente para las pruebas en cada uno de los ambientes.
- Preparar los ambientes locales y en la nube que se utilizarán para la ejecución de las pruebas, validar que los ambientes de la nube permitan ejecutar el mismo tipo de pruebas que de forma local.
- Asegurar que no exista ninguna limitante respecto a integraciones con componentes externos, para que se pueda tener un mejor control de los tiempos y los recursos de los servidores en la ejecución de las pruebas.

- Garantizar que el software y datos que se carguen en la nube no tengan problemas de vulnerabilidad o puedan ser accedidos por personas sin autorización.
- Realizar el informe final donde se analizarán los resultados reales de cada prueba con su respectivo ambiente y se obtendrán las conclusiones para validar el segundo proyecto.

Acerca del registro de incidentes, estos no serán ingresados en ninguna herramienta o documento ya que el alcance del proyecto no indica la corrección de ninguno de los incidentes funcionales o técnicos que aparezcan.

## 5. CAPÍTULO V

### 5.1. Conclusiones

- En nuestro medio las organizaciones que desarrollan software si conocen o si están familiarizadas con el concepto de aseguramiento de la calidad en software y el concepto de pruebas no funcionales.
- Para ciertas organizaciones como el sector público, no siempre es factible alquilar herramientas en la nube debido al manejo de su presupuesto, ya que esto se lo hace anual y no se puede solicitar el dinero para alquiler de las herramientas en cualquier ocasión.
- Independiente del tamaño de la organización, la mayoría de ellas realiza pruebas no funcionales, esto indica la gran responsabilidad de cada organización por elaborar productos de software apegados a estándares de calidad.
- En un alto porcentaje los desarrolladores en las organizaciones realizan el trabajo propio de desarrollo y además tareas del probador, esto funciona si el desarrollador es muy objetivo en su trabajo, sin embargo en este caso es preferible que se realicen revisiones cruzadas con otros desarrolladores.
- No existe nada que asocie el tamaño de las organizaciones con el conocimiento o implementación de actividades de aseguramiento de calidad, ya que de las empresas que contestaron que no están familiarizadas con este concepto, se evidencio que están distribuidas entre pequeñas, medianas y grandes.
- Los resultados de las pruebas no funcionales no siempre son fáciles de identificar por parte de los usuarios finales de las aplicaciones, a medida que el tiempo transcurre puede ir conociendo la robustez de un producto de software.
- Las organizaciones medianas y grandes son aquellas que pueden tener equipos de personas dedicadas exclusivamente a tareas de aseguramiento de la calidad.

## 5.2. Recomendaciones

- Difundir la importancia de las pruebas no funcionales de software para que más organizaciones en el país puedan realizarlas. Si el impedimento es la parte económica por el costo de las licencias, existen herramientas de código libre que se pueden utilizar.
- Realizar un modelamiento para la implementación de pruebas no funcionales desde la nube tomando un tipo de prueba como ejemplo aplicada a un producto de software. Esto es una alternativa para las organizaciones que prefieran realizar este tipo de pruebas en la nube.
- Analizar si en realidad es más conveniente económicamente realizar las pruebas no funcionales en la nube o de manera local, ya que como se registró en la entrevista la suite de herramientas para ejecutar de forma local puede sobrepasar los \$100.000 dólares y este valor no muchas organizaciones pueden invertirlo en pruebas de software.
- Como parte de la calidad de un producto de software se asocia las características no funcionales, por ello debe existir la necesidad de la planificación de dichas pruebas en las etapas tempranas del desarrollo del software.
- Para definir qué se debe probar dentro de las pruebas no funcionales, el usuario experto o responsable del negocio debería contar con la ayuda de un experto técnico para que estos requerimientos no sean vagos y que el software pueda cumplir con las necesidades de los usuarios.

## 6. BIBLIOGRAFÍA

- Osiatis S.A. (s.f.). *ITIL V3 Gestión de servicios TI: Validación y Pruebas*. Recuperado de: [http://itilv3.osiatis.es/transicion\\_servicios\\_TI/validacion\\_pruebas.php](http://itilv3.osiatis.es/transicion_servicios_TI/validacion_pruebas.php)
- International Software Testing Qualifications Board. (2011). *Certified Tester Foundation Level (V1.10)*. Bélgica.
- Van Veenendaal, E., Graham, D., (2012). *Foundations of Software Testing ISTQB Certification*, Cheriton House, North Way, Andover, Hampshire, United Kingdom.
- Martín Ávila, T., (2010). *Certificaciones y Normativas de calidad en Software*. Recuperado de: <http://www.it360.es/certificaciones-normativas-calidad-en-desarrollo-de-software.php>
- Podelko, A. (s.f.). *Performance Testing in the Cloud: Look Beyond the Word*. *Testing Experience, Diciembre 2012*, 7-8.
- Forno, M. (s.f.). *Software Testing: Adapt and Grow with the Cloud*. *Testing Experience, Diciembre 2012*, 19-20.
- Chittanai, K. (s.f.). *Moving Testing to the Cloud – An Exploration*. *Testing Experience, Diciembre 2012*, 27-29.
- Computerworld. (2011). *La nube computacional un modelo de aprovechamiento de infraestructura que avanza para quedarse*. (Ediworld, Ed.) Computerworld. Ecuador(230), 19-21.
- Manilla, J. y Torres, H. Scribd, *Técnicas de recopilación de información*. Recuperado de: <http://es.scribd.com/doc/23659550/TECNICAS-DE-RECOPIACION-DE-INFORMACION>
- Robert, Y. *Investigación sobre Estudio de Casos*. Sage Publications. Recuperado de: <http://www.polipub.org/documentos/YIN%20ROBERT%20.pdf>
- Barrio, I., González, J., Padín, L., Peral, P., Sánchez, I., Tarín, E. *El Estudio de Casos*. Universidad Autónoma de Madrid. Recuperado de: [http://www.uam.es/personal\\_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Est\\_Casos\\_doc.pdf](http://www.uam.es/personal_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Est_Casos_doc.pdf)

- Yacuzzi, E. El estudio de caso como metodología de la investigación: Teoría, Mecanismos Causales, Validación. Recuperado de: <http://www.ucema.edu.ar/publicaciones/download/documentos/296.pdf>
- Gallagher, L. y Morris H., ITIL Foundation Exam Study Guide, Agosto del 2012
- Runeson, P., Host, M., Rainer, A., Regnell, B. (2012). Case Study Research in Software Engineering: Guidelines and Examples. New Jersey, Estados Unidos: John Wiley & Sons, Inc.
- Aerohive Networks, Nube pública o privada: La decisión es suya. Recuperado de: [http://www.aerohive.com/pdfs/international/Aerohive\\_Whitepaper\\_Public-or-Private-Cloud\\_Spanish.pdf](http://www.aerohive.com/pdfs/international/Aerohive_Whitepaper_Public-or-Private-Cloud_Spanish.pdf)
- Meier, J. D., Farre, C., Bansode, P., Barber, S., Rea, D. (2007). Performance Testing Guidance for Web Applications.
- Ford, C., Gileadi, I., Purba, S., Moerman, M., (2007). Patterns for Performance and Operability: Building and Testing Enterprise. Auerbach Publications, 11-13.
- LOADUIWEB. SmartBear. Análisis de Reportes, métricas más importantes, Recuperado de: <http://loaduiweb.org/docs/testing/results/analyzing-the-report.html>
- ISO25000 (2014). La familia de normas ISO/IEC 25000. Recuperado de: <http://iso25000.com/index.php/normas-iso-25000?start=4>

## 7. ANEXOS

### 7.1. Encuesta

A continuación se adjunta la encuesta que se realizó a las empresas afiliadas a la AESOFT por medio electrónico.

#### Aseguramiento de la Calidad del Software

IMPORTANTE: La siguiente encuesta está orientada a empresas que realizan desarrollo de software.

**1. Seleccione la cantidad de personas que trabajan en su organización \***

- 1 - 10
- 11 - 49
- 50 - 249
- 250 o más

**2. Qué tipo de software desarrolla? \***

- Comercio y manufactura
- Gobierno
- Financiero
- Gestión empresarial
- Servicios diversos
- Alimentos y bebidas
- Educación
- Información y comunicación
- Servicios médicos
- Móviles
- Otros

**3. Cuántas personas trabajan en su equipo de desarrollo? \***

- 1 - 5
- 6 - 20
- 21 en adelante

**4. Está familiarizado con el concepto de ASEGURAMIENTO DE LA CALIDAD (QA) en software? \***

- Sí
- No

**5. Seleccione lo que se apega a su negocio \***

- Hay un área/equipo de aseguramiento de calidad claramente definido
- Los desarrolladores también hacen actividades de aseguramiento de calidad
- Ninguna persona hace actividades de aseguramiento de calidad
- Tercerizo actividades de aseguramiento de calidad

**6. Si en la pregunta anterior contestó que "Ninguna persona hace actividades de aseguramiento de calidad" indique un motivo**

**7. Si realiza tareas de aseguramiento de calidad en su empresa o externamente, cuáles de las siguientes actividades realiza?**

- Revisión de documentos
- Revisión de estándares
- Pruebas funcionales (sistema)
- Pruebas no funcionales (técnicas)
- Documentación de incidentes
- Pruebas de caja blanca
- Versionamiento de artefactos
- Pruebas de humo
- Pruebas de aceptación de usuario
- Elaboración de documentos
- Creación de ambientes

**8. Está familiarizado con el concepto de PRUEBAS NO FUNCIONALES de software? \***

- Sí
- No

**9. Si realiza PRUEBAS NO FUNCIONALES, cuál de las siguientes realiza:**

- Pruebas de carga
- Pruebas de stress
- Pruebas de desempeño
- Pruebas de memoria
- Pruebas de cobertura de código
- Otras

**10. Si realiza PRUEBAS NO FUNCIONALES en el software, las realiza:**

**11. Si en la pregunta anterior contestó que hace PRUEBAS NO FUNCIONALES en la nube, indique el motivo**

**12. Su empresa tiene alguna CERTIFICACIÓN respecto a la calidad del software? \***

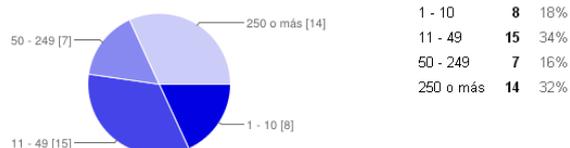
- Sí
- No

**13. Si en la anterior pregunta contestó que SI, indique cuál es la CERTIFICACIÓN**

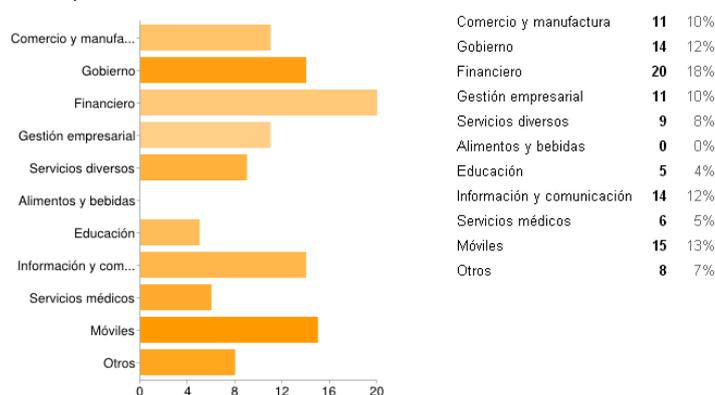
## 7.2. Resumen de las respuestas en la encuesta

A continuación se adjunta el resumen de las respuestas obtenidas por medio de la encuesta realizada por correo electrónico a las empresas afiliadas a la AESOFT.

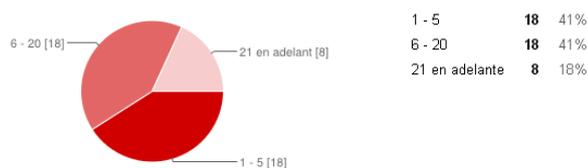
### 1. Seleccione la cantidad de personas que trabajan en su organización



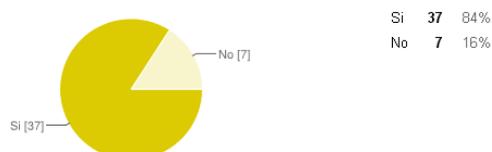
### 2. Qué tipo de software desarrolla?



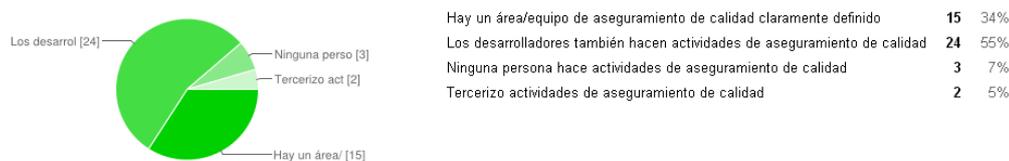
### 3. Cuántas personas trabajan en su equipo de desarrollo?



### 4. Está familiarizado con el concepto de ASEGURAMIENTO DE LA CALIDAD (QA) en software?



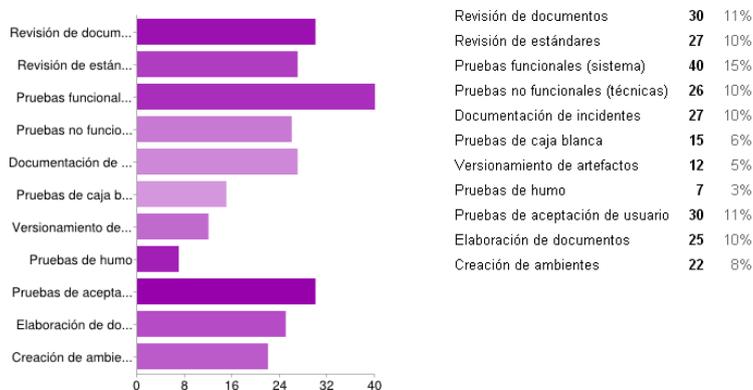
### 5. Seleccione lo que se apega a su negocio



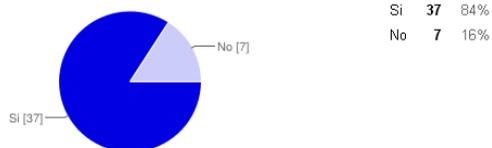
**6. Si en la pregunta anterior contestó que "Ninguna persona hace actividades de aseguramiento de calidad" indique un motivo**

Desconocimiento | ignorancia, falta de presupuesto, falta de interes por el usuario.

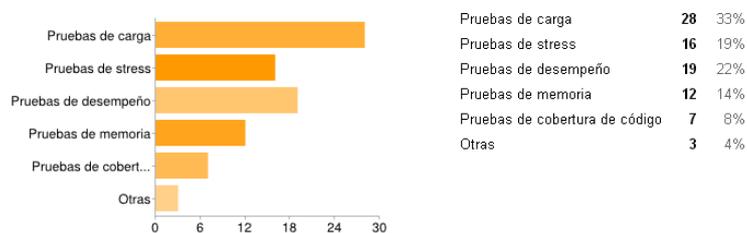
**7. Si realiza tareas de aseguramiento de calidad en su empresa o externamente, cuáles de las siguientes actividades realiza?**



**8. Está familiarizado con el concepto de PRUEBAS NO FUNCIONALES de software?**



**9. Si realiza PRUEBAS NO FUNCIONALES, cuál de las siguientes realiza:**



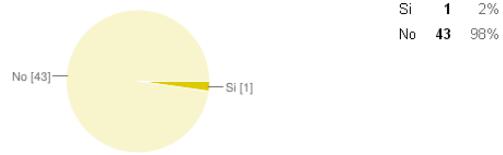
**10. Si realiza PRUEBAS NO FUNCIONALES en el software, las realiza:**



**11. Si en la pregunta anterior contestó que hace PRUEBAS NO FUNCIONALES en la nube, indique el motivo**

Porque son aplicaciones Web | Hay servicios que cubren más fácilmente las pruebas de carga | Estoy en el proceso de hacerlo | Depende del ambiente que tenga el cliente, hacer un ambiente muy similar.

12. Su empresa tiene alguna CERTIFICACIÓN respecto a la calidad del software?



13. Si en la anterior pregunta contestó que SI, indique cuál es la CERTIFICACIÓN

ISO, CMMi

### 7.3. Entrevista a experto

Entrevista a la Ingeniera Susana Samaniego.

#### Acerca de la empresa

El Servicio de Rentas Internas (SRI) es una entidad técnica y autónoma que tiene la responsabilidad de recaudar los tributos internos establecidos por Ley mediante la aplicación de la normativa vigente. Su finalidad es la de consolidar la cultura tributaria en el país a efectos de incrementar sostenidamente el cumplimiento voluntario de las obligaciones tributarias por parte de los contribuyentes.

#### Acerca del entrevistado

La Ingeniera Susana Samaniego ha trabajado alrededor de 5 años en el Servicio de Rentas Internas en el área de Control de Calidad, durante este tiempo ha participado de la madurez del área ya que se han venido implementando varias metodologías para ir puliendo los procesos existentes en ella.

FICHA DE ENTREVISTA	
<b>Nombre del entrevistado:</b>	Ing. Susana Samaniego
<b>Cargo:</b>	Jefe Nacional del área de Control de Calidad
<b>Organización:</b>	Servicio de Rentas Internas
<b>Personas a cargo:</b>	16
<b>Fecha:</b>	25 de junio del 2013
<b>Información de la entrevista</b>	
<b>¿Cuál es la importancia que tienen las pruebas no funcionales en los procesos de aseguramiento de la calidad del software?</b>	
<p>Dependiendo el software que se desarrolle y su cliente objetivo radica la importancia de las pruebas no funcionales o técnicas, por ejemplo si es un software pequeño para Pymes tal vez no sea necesario invertir tantos recursos en las pruebas, mientras que si hacemos un portal para el estado donde se pueden</p>	

conectar miles de personas al día en cambio debe ser una obligación realizar pruebas no funcionales para saber si el software soportará o no la concurrencia.

**¿Existe un documento donde se planifiquen las pruebas que se van a realizar en el proceso de pruebas de software?**

Si, se utiliza el plan de pruebas

**¿Qué pruebas no funcionales ejecuta en las pruebas de software?**

Pruebas de carga, stress, desempeño, análisis de memoria, cobertura de código

**¿Cuántos ambientes tiene para ejecutar estas pruebas antes de salir a producción?**

Dos de los que corresponden al área, uno de *testing* y otro de certificación

**¿Realiza pruebas no funcionales post producción?**

No como parte del procedimiento de pruebas, solo bajo pedido, cuando alguna aplicación empieza a presentar algún defecto no esperado.

**¿En qué medio realiza las pruebas no funcionales, cuán factible es su implementación?**

Las pruebas no funcionales se las ejecuta en nuestros propios ambientes, debido a que tenemos comprado varias licencias para las herramientas.

Respecto a la factibilidad, ya no hay excusas para no hacer pruebas no funcionales de pronto la única limitante sería tener ambientes aislados para hacer las pruebas porque ahora tenemos herramientas gratis en la nube.

**¿Cuál es la principal razón para ejecutar en este medio las pruebas no funcionales?**

Al no tener la flexibilidad de disponer de recursos económicos en el momento que se desee por manejar un estricto presupuesto anual por pertenecer al sector público, nos vemos en la obligación de adquirir las herramientas con licencia, ya que se ha realizado un análisis previo y no se ha podido conseguir herramientas *open source* que cubran todas nuestras necesidades. Además, aquí se desarrolla software en gran medida y prácticamente siempre estamos utilizando las herramientas.

**¿Cuál es el costo aproximado por mes del uso de las herramientas para pruebas no funcionales?**

En nuestro caso la suite de herramientas Silk (SilkPerformer, SilkCentral, SilkTest, SilkMobile, DevPartner) tienen un costo aproximado de \$180.000, cada año se contrata un servicio de soporte de alrededor de \$40.000

**¿A su parecer considera que el usuario final percibe el valor agregado cuando se aplican las pruebas no funcionales?**

El usuario final básicamente lo que requiere es que el producto de software desarrollado cumpla con sus necesidades funcionales, así que desde un inicio no percibe el valor agregado que dan este tipo de pruebas, conforme va pasando el tiempo y va creciendo por ejemplo el número de usuarios es cuando el usuario final se da cuenta de la robustez del software.

**¿Cuál es la incidencia que tiene en una organización ejecutar pruebas no funcionales?**

Si partimos desde el hecho de que hay que tener un equipo enfocado a realizar actividades de calidad, algo que es costoso para muchas organizaciones, ya estamos agregando valor al producto final que es la aplicación para el usuario. Así que implementando estas pruebas dentro del proceso de aseguramiento de la calidad estamos aportando con más valor a nuestro producto final y a pesar de que nuestro giro de negocio no es comercial ofrecemos a los contribuyentes servicios de calidad por medio de las aplicaciones y esto si tiene incidencia en la imagen de la organización.

**¿Recomendaría ejecutar pruebas no funcionales en la nube?**

Si, al parecer es menos costoso, esto lo digo con referencia a la suite de Office, es más económico adquirir las nuevas licencias del software en la nube.

Algo que si es importante recalcar es que la infraestructura de la nube puede permitir que las pruebas no sean tan limitadas en recursos.

**En el caso de subir datos a la nube para hacer pruebas no funcionales ¿Confía en que sus datos seguirán siendo confidenciales?**

Se entendería que parte del servicio que ofrecen las empresas que realizan estas pruebas en la nube sería la seguridad de mantener la confidencialidad de los datos. En nuestro caso tenemos información muy sensible de personas como de empresas que debe ser protegida.

Hoy en día existen bastantes parámetros de seguridad, existe contratos con acuerdos de penalización, lo importante de esto es trasladar el riesgo a los proveedores del servicio.

#### **7.4. Matriz de correlación**

Información cruzada de la encuesta realizada a las organizaciones de la AESOFT versus la entrevista realizada a un experto de Control de Calidad.

Encuesta	Entrevista	Comentario
¿Qué pruebas no funcionales ejecuta en su organización?	¿Qué pruebas no funcionales se ejecutan en las pruebas de software?	La mayoría concuerda con pruebas de carga, stress, desempeño, cobertura de código, análisis de memoria.
Si realiza pruebas no funcionales, ¿dónde las realiza?	¿En qué medio se realizan las pruebas no funcionales?	En la encuesta se evidencia que la mayoría de organizaciones realiza pruebas de forma local, en tanto que nuestra entrevistada nos da la razón principal del por qué las realiza de forma local, debido a razones presupuestarias del sector público.
	¿Cuál es el costo aproximado por mes del uso de las herramientas para pruebas no funcionales?	Esta pregunta en la entrevista indica una muy probable desventaja de adquirir software para pruebas no funcionales, el costo es alto.
Si realiza pruebas no funcionales en la nube, indique el motivo	¿Recomendaría ejecutar pruebas no funcionales en la nube?	Ambas partes indican de manera puntal que una de las ventajas de hacer pruebas de este tipo en la nube es debido a su infraestructura.