

PLAN DE NEGOCIO PARA LA CREACIÓN DE UNA EMPRESA DE SERVICIO Y ASESORÍA DE PRUEBAS DE SOFTWARE

**JOHANA ANDREA MORALES MEJÍA
ANA MARÍA SIERRA GONZALEZ**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS: ELÉCTRICA, ELECTRÓNICA,
FÍSICA Y CIENCIAS DE LA COMPUTACIÓN
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
2010**

**PLAN DE NEGOCIO PARA LA CREACIÓN DE UNA EMPRESA DE
SERVICIO Y ASESORÍA DE PRUEBAS DE SOFTWARE**

**JOHANA ANDREA MORALES MEJÍA
ANA MARÍA SIERRA GONZALEZ**

**Trabajo de grado para obtener el título de Ingeniero de Sistemas y
Computación**

**DIRECTOR DEL PROYECTO:
Luz Stella Valencia Ayala
Ingeniera Industrial**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS: ELÉCTRICA, ELECTRÓNICA, FÍSICA Y
CIENCIAS DE LA COMPUTACIÓN
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA
2010**

Nota de aceptación

Firma del jurado

Pereira, 01 de junio de 2010

TABLA DE CONTENIDO

| | Pág. |
|--|------|
| INTRODUCCIÓN | 15 |
| DEFINICIÓN DEL PROBLEMA | 16 |
| OBJETIVO GENERAL | 17 |
| OBJETIVOS ESPECÍFICOS..... | 17 |
| 1 MARCO CONCEPTUAL | 19 |
| 1.1 FUNDAMENTOS DE LAS PRUEBAS DE SOFTWARE | 19 |
| 1.1.1 Objetivos de las pruebas..... | 19 |
| 1.1.2 Principios de las pruebas. | 19 |
| 1.2 DISEÑO DE CASOS DE PRUEBA..... | 20 |
| 1.3 PRUEBA DE CAJA BLANCA..... | 21 |
| 1.4 PRUEBA DEL CAMINO BÁSICO | 21 |
| 1.4.1 Notación del grafo de flujo..... | 22 |
| 1.4.2 Complejidad ciclomática..... | 22 |
| 1.5 PRUEBA DE LA ESTRUCTURA DE CONTROL..... | 23 |
| 1.5.1 Prueba de condición..... | 23 |
| 1.5.2 Prueba del flujo de datos..... | 24 |
| 1.5.3 Prueba de bucles. | 24 |
| 1.6 PRUEBA DE CAJA NEGRA | 25 |
| 1.6.1 Métodos prueba basados en grafos..... | 25 |
| 1.6.2 Participación equivalente. | 26 |
| 1.6.3 Análisis de Valores Límite. | 27 |
| 1.6.4 Prueba de comparación. | 27 |
| 1.6.5 Prueba de la tabla ortogonal. | 28 |

| | | |
|--------|--|----|
| 1.7 | PRUEBAS DE ENTORNOS ESPECIALIZADOS, ARQUITECTURAS Y APLICACIONES | 28 |
| 1.8 | ESTRATEGIAS DE PRUEBAS DE SOFTWARE | 30 |
| 1.8.1 | Validación y verificación..... | 30 |
| 1.8.2 | Organización para las pruebas de software..... | 31 |
| 1.8.3 | Una estrategia de prueba de software. | 31 |
| 1.8.4 | Aspectos estratégicos. | 32 |
| 1.9 | PRUEBA DE UNIDAD | 33 |
| 1.10 | PRUEBAS DE INTEGRACIÓN | 35 |
| 1.10.1 | Integración descendente..... | 35 |
| 1.10.2 | Integración ascendente..... | 37 |
| 1.10.3 | Prueba de regresión. | 37 |
| 1.10.4 | Prueba de humo. | 38 |
| 1.11 | PRUEBAS DE VALIDACIÓN | 39 |
| 1.12 | PRUEBAS DE SISTEMA | 40 |
| 1.12.1 | Prueba de recuperación..... | 40 |
| 1.12.2 | Prueba de seguridad..... | 40 |
| 1.12.3 | Prueba de resistencia (stress). | 41 |
| 1.12.4 | Pruebas de rendimiento..... | 41 |
| 1.13 | METODOLOGÍAS DE PRUEBAS DE SOFTWARE | 41 |
| 1.13.1 | Metodología en V..... | 41 |
| 1.13.2 | Metodología en W..... | 42 |
| 1.13.3 | TDD desarrollo basado en pruebas. | 43 |
| 1.14 | PRUEBAS SEGÚN EL MODELO DE DESARROLLO..... | 44 |
| 1.14.1 | Actores..... | 45 |
| 1.14.2 | Metodologías convencionales..... | 45 |
| 1.14.3 | Metodologías ágiles. | 46 |
| 1.15 | HERRAMIENTAS | 46 |
| 1.15.1 | Bugzilla. | 46 |
| 1.15.2 | Rational ClearQuest..... | 47 |
| 1.15.3 | DevTrack..... | 47 |
| 1.15.4 | Census Bug Tracking and Defect. | 48 |

| | | |
|---------|---|----|
| 1.15.5 | Mantis | 49 |
| 1.15.6 | Rational Performance Tester | 49 |
| 1.15.7 | Jmeter | 50 |
| 1.15.8 | Load Runner Product Family | 50 |
| 1.15.9 | QaTraQ Profesional test management | 51 |
| 1.15.10 | TestDirector | 51 |
| 1.15.11 | TestGen4J | 52 |
| 1.15.12 | .Test | 53 |
| 1.15.13 | SoapUI | 53 |
| 1.15.14 | Rational Robot | 54 |
| 2 | RESUMEN EJECUTIVO | 56 |
| 2.1 | CONCEPTO DEL NEGOCIO | 57 |
| 2.2 | POTENCIAL DEL MERCADO EN CIFRAS | 57 |
| 2.3 | VENTAJAS COMPETITIVAS Y PROPUESTA DE VALOR | 57 |
| 2.4 | RESUMEN DE LAS INVERSIONES REQUERIDAS | 58 |
| 2.5 | PROYECCIONES DE VENTAS Y RENTABILIDAD | 58 |
| 2.6 | CONCLUSIONES FINANCIERAS Y EVALUACIÓN DE VIABILIDAD | 58 |
| 3 | MODULO MERCADO | 60 |
| 3.1 | ANÁLISIS DEL SECTOR | 61 |
| 3.1.1 | Industria de software | 61 |
| 3.1.2 | Cadena de valor de la industria del software | 61 |
| 3.1.3 | Actores que participan en la industria del software | 63 |
| 3.1.4 | Industria del software en el mundo | 64 |
| 3.1.4.1 | Estados Unidos | 64 |
| 3.1.4.2 | India | 66 |
| 3.1.4.3 | Irlanda | 67 |
| 3.1.4.4 | Israel | 67 |
| 3.1.5 | Industria del software en Latinoamérica | 68 |
| 3.1.5.1 | Brasil | 70 |
| 3.1.5.2 | Argentina | 71 |
| 3.1.5.3 | México | 71 |
| 3.1.6 | Industria del Software en Colombia | 72 |
| 3.2 | ANÁLISIS DEL MERCADO | 77 |

| | | |
|-------|---|----|
| 3.2.1 | Mercado Objetivo. | 77 |
| 3.2.2 | Justificación. | 77 |
| 3.2.3 | Estimación del mercado potencial. | 78 |
| 3.2.4 | Perfil del consumidor. | 81 |
| 3.3 | ANÁLISIS DE LA COMPETENCIA | 81 |
| 3.3.1 | Galia Technologies..... | 81 |
| 3.3.2 | Choucair Testing S.A..... | 81 |
| 3.3.3 | Indudata Ltda..... | 82 |
| 3.3.4 | Q-Vision Technologies S.A..... | 82 |
| 3.3.5 | GreenSQA..... | 82 |
| 3.3.6 | Posición frente a la competencia..... | 83 |
| 3.3.7 | Análisis del costo..... | 83 |
| 3.3.8 | Análisis precio de venta..... | 84 |
| 3.4 | CONCEPTO DEL SERVICIO | 85 |
| 3.5 | ESTRATEGIAS DE MERCADEO. | 86 |
| 3.6 | ESTRATEGIAS DE PROMOCIÓN | 87 |
| 3.7 | ESTRATEGIAS DE COMUNICACIÓN. | 87 |
| 3.8 | ESTRATEGIAS DE SERVICIO..... | 87 |
| 3.9 | PROYECCIONES DE VENTAS..... | 88 |
| 4 | MODULO OPERACIÓN | 90 |
| 4.1 | FICHA TÉCNICA DEL SERVICIO | 91 |
| 4.1.1 | Prueba de requerimientos. | 91 |
| 4.1.2 | Prueba de diseño. | 92 |
| 4.1.3 | Prueba de integración. | 92 |
| 4.1.4 | Pruebas funcionales. | 92 |
| 4.1.5 | Prueba de aceptación..... | 93 |
| 4.1.6 | Prueba de seguridad. | 93 |
| 4.1.7 | Prueba de stress. | 93 |
| 4.1.8 | Pruebas de rendimiento..... | 94 |
| 4.1.9 | Herramientas de automatización de pruebas. | 95 |
| 4.2 | DESCRIPCIÓN DEL PROCESO DE PRUEBAS. | 95 |
| 4.2.1 | Planificación de las pruebas..... | 96 |
| 4.2.2 | Análisis de las pruebas..... | 98 |

| | | |
|---------|--|-----|
| 4.2.3 | Diseño de las pruebas..... | 100 |
| 4.2.4 | Ejecución de las pruebas..... | 103 |
| 4.2.5 | Evaluación de resultados..... | 104 |
| 4.3 | NECESIDADES Y REQUERIMIENTOS | 105 |
| 4.3.1 | Hardware requerido..... | 105 |
| 4.3.1.1 | Computadores. | 105 |
| 4.3.1.2 | Servidor..... | 106 |
| 4.3.1.3 | Impresora..... | 106 |
| 4.3.2 | Software requerido. | 107 |
| 4.3.2.1 | Sistema operativo. | 107 |
| 4.3.2.2 | Herramientas de automatización de pruebas..... | 107 |
| 4.3.3 | Localización..... | 107 |
| 4.3.4 | Presupuesto requerido. | 108 |
| 5 | MODULO ORGANIZACIÓN..... | 109 |
| 5.1 | MISIÓN..... | 110 |
| 5.2 | VISIÓN..... | 110 |
| 5.3 | VALORES..... | 110 |
| 5.4 | GRUPO EMPRENDEDOR | 111 |
| 5.5 | ANÁLISIS DOFA..... | 111 |
| 5.6 | ESTRUCTURA ORGANIZACIONAL | 113 |
| 5.6.1 | Organigrama..... | 113 |
| 5.6.2 | Descripción de cargos. | 113 |
| 5.7 | ASPECTOS LEGALES..... | 115 |
| 5.8 | GASTOS DE ARRANQUE..... | 116 |
| 5.9 | GASTOS DE PERSONAL | 117 |
| 6 | MODULO FINANCIERO | 118 |
| 6.1 | EGRESOS..... | 119 |
| 6.1.1 | Inversión inicial. | 119 |
| 6.1.2 | Mano de obra. | 120 |
| 6.1.3 | Costos de operación..... | 121 |

| | | |
|-------|-----------------------------------|-----|
| 6.1.4 | Depreciaciones..... | 121 |
| 6.2 | INGRESOS POR VENTAS..... | 122 |
| 6.3 | ESTADOS DE RESULTADOS | 124 |
| 6.4 | BALANCE GENERAL..... | 125 |
| 6.5 | FLUJO DE CAJA | 126 |
| 7 | MODULO IMPACTO DEL PROYECTO | 127 |
| | CONCLUSIONES | 129 |
| | GLOSARIO | 130 |
| | BIBLIOGRAFIA..... | 131 |
| | ANEXOS..... | 133 |

ÍNDICE DE TABLAS

| | Pág. |
|--|------|
| Tabla 1. Costo inversión inicial | 58 |
| Tabla 2. Proyección de ventas..... | 58 |
| Tabla 3. Costo del servicio..... | 84 |
| Tabla 4. Precio del servicio..... | 85 |
| Tabla 5. Unidades vendidas primer año | 88 |
| Tabla 6. Ventas primer año..... | 88 |
| Tabla 7. Presupuesto requerido..... | 108 |
| Tabla 8. Gastos legales | 116 |
| Tabla 9. Salarios básicos..... | 117 |
| Tabla 10. Costo total empleado por mes | 117 |
| Tabla 11. Herramientas técnicas | 119 |
| Tabla 12. Herramientas administrativas..... | 119 |
| Tabla 13. Gastos legales | 119 |
| Tabla 14. Costos inversión inicial..... | 120 |
| Tabla 15. Mano de obra..... | 120 |
| Tabla 16. Costo mano de obra..... | 120 |
| Tabla 17. Proyección costos de mano de obra..... | 121 |
| Tabla 18. Costos de operación | 121 |
| Tabla 19. Depreciación | 122 |

| | |
|--|-----|
| Tabla 20. Precio del servicio | 122 |
| Tabla 21. Proyección unidades vendidas | 122 |
| Tabla 22. Proyección de ventas..... | 123 |

ÍNDICE DE ILUSTRACIONES

| | Pág. |
|---|------|
| Ilustración 1. Notación del grafo de flujo | 22 |
| Ilustración 2. Integración descendente | 36 |
| Ilustración 3. Integración ascendente | 37 |
| Ilustración 4. Metodología en V..... | 42 |
| Ilustración 5. Metodología en W..... | 43 |
| Ilustración 6. Ciclos TDD | 44 |
| Ilustración 7. Mapa procesos | 62 |
| Ilustración 8. Relación oferta y demanda..... | 63 |
| Ilustración 9. Participación de las Tic en el PIB de Estados Unidos | 64 |
| Ilustración 10. Industria de las TIC en el PIB de Estados Unidos | 65 |
| Ilustración 11. Mercado latinoamericano | 69 |
| Ilustración 12. Perfil de las empresas brasileñas de software | 70 |
| Ilustración 13. Evolución del mercado TI en Colombia | 73 |
| Ilustración 14. Disponibilidad mano de obra calificada en Colombia | 74 |
| Ilustración 15. Universidades registradas en Colombia | 74 |
| Ilustración 16. Número graduados universitarios América Latina | 75 |
| Ilustración 17. Tipo de producto o servicio..... | 79 |
| Ilustración 18. Tipo de mercado..... | 79 |
| Ilustración 19. Ventas anuales en millones de pesos | 80 |

| | |
|--|-----|
| Ilustración 20. Quien realiza las pruebas en la parte de desarrollo..... | 80 |
| Ilustración 21. Diagrama del proceso..... | 96 |
| Ilustración 22. Actividades etapa de planificación de pruebas..... | 97 |
| Ilustración 23. Actividades etapa de análisis de pruebas..... | 98 |
| Ilustración 24. Actividades etapa diseño de pruebas..... | 101 |
| Ilustración 25. Actividades etapa de ejecución de pruebas..... | 103 |
| Ilustración 26. Actividades etapa de evaluación de resultados..... | 104 |
| Ilustración 27. Organigrama..... | 113 |

ANEXOS

| | Pág. |
|--|------|
| Anexo A. Informe de viabilidad y alcance. | 134 |
| Anexo B. Informe de planificación..... | 135 |
| Anexo C. Plan de pruebas. | 136 |
| Anexo D. Casos de prueba..... | 138 |
| Anexo E. Matriz de trazabilidad. | 139 |
| Anexo F. Informe de incidencias..... | 140 |
| Anexo G. Histórico de pruebas. | 141 |
| Anexo H. Informe final de pruebas..... | 142 |
| Anexo I. Ficha técnica de la encuesta | 143 |

INTRODUCCIÓN

En este proyecto se llevo a cabo el proceso para la creación de una empresa de servicio de pruebas de software, con el fin de que las empresas desarrolladoras alcancen competitividad en la industria del software, por medio de productos de excelente calidad a nivel funcional, lo cual se puede conseguir por medio de la aplicación de un proceso metodológico de pruebas, aplicado en las etapas del ciclo de vida del desarrollo de software.

Este proceso se realizo basados en el formato de plan de negocio del SENA - Fondo emprender, el cual consiste en lo siguiente:

- En el modulo mercado se definió el mercado objetivo y la actual competencia, como también la proyección de ventas para el primer año de funcionamiento de QualityTest S.A.S
- En el modulo operación se estableció las características principales del servicio a prestar en cuanto a pruebas y herramientas a utilizar en cada etapa del plan de pruebas
- En el modulo organización se definió bajo que aspectos legales y organizacionales trabaja QualityTest S.A.S., además de la estimación de presupuestos.
- En el modulo financiero se determinaron el valor de los egresos e ingresos dando como resultado el balance general y el flujo de caja proyectado a cinco años.

DEFINICIÓN DEL PROBLEMA

En la actualidad la ciudad de Pereira presenta un alto crecimiento de empresas desarrolladoras de software, las cuales deben competir en un mercado nacional e internacional, requiriendo desarrollar productos de una alta calidad para sobresalir en dichos mercados, por esto, se hace necesario que estas empresas brinden mayor atención a todos sus procesos y en especial al de pruebas, el cual garantiza el cumplimiento de los requerimientos del cliente.

Un gran número de empresas desarrolladoras de software de la ciudad no poseen la cultura de aplicar metodologías de ingeniería de software desde el inicio de su proceso de desarrollo, lo cual trae como consecuencia un análisis de requerimientos deficiente, proceso básico para que las pruebas puedan cumplir con sus objetivos, ya que son estas las que se encargan de verificar el cumplimiento de los requerimientos. También, se presenta problemas en el proceso de pruebas, porque dichas empresas hacen pruebas sobre sus productos informalmente, ya que no cuentan con personal especializados en estos procesos, normalmente usan a los mismos desarrolladores como analistas de pruebas, los cuales evitan realizar pruebas más exhaustivas y las harán solo en los módulos que ellos consideran necesario; o utilizan a los futuros usuarios los cuales tienen poca experiencia en la detección de errores.

La mayoría de inconsistencias del software se descubren en la etapa de implementación, incrementando costo y tiempo tanto para la empresa desarrolladora como para el cliente y creando una imagen desfavorable.

Finalmente el problema es la baja capacidad que exhiben las empresas de desarrollo de software de la ciudad de Pereira en el proceso de pruebas, que se refleja en la calidad del producto final.

OBJETIVO GENERAL

Desarrollar un plan de negocio para la creación de una empresa de servicio y asesoría de pruebas de software.

OBJETIVOS ESPECÍFICOS

- Realizar análisis de mercado para recopilar información con la cual analizar el consumo y la competencia actualmente, y desarrollar estrategias de mercadeo.
- Desarrollar análisis operacional en el cual se describirá las características técnicas del servicio a ofrecer.
- Realizar análisis organizacional para definir la estructura de la empresa y definir los aspectos legales que la regirán.
- Desarrollar análisis financiero con el cual se deducirá cuanto se necesita para llevar a cabo la creación de la empresa.
- Definir el impacto del proyecto a nivel económico, regional, social y ambiental que puede generar el proyecto.

DISEÑO METODOLÓGICO

HIPÓTESIS

¿Es posible a través de una empresa de servicios y asesorías pruebas de software, reducir los problemas de calidad de los productos de software que se presentan en las empresas desarrolladoras de la ciudad de Pereira?

TIPO DE INVESTIGACIÓN

Descriptiva.

POBLACIÓN

20 empresas.

MUESTRA

Dado el tamaño de la población se realizó un censo.

VARIABLES

- Número de empresas desarrolladoras de software
- Número de empresas que tienen buenas prácticas de desarrollo de software
- Número de empresas que están dispuesta a contratar un servicio de pruebas.

1 MARCO CONCEPTUAL

La información que aparece a continuación fue tomada del libro: Ingeniería de software. Un enfoque práctico, Roger S. Pressman

1.1 FUNDAMENTOS DE LAS PRUEBAS DE SOFTWARE

Las pruebas de software se definen como el proceso que permite evaluar el desarrollo, manual o automáticamente, para verificar los requerimientos o para identificar los resultados esperados y los que produce el sistema.

Las pruebas de software son un elemento crítico para la calidad del software, sus ventajas están asociadas a los costos que traen los errores en etapas avanzadas del ciclo de vida del software.

1.1.1 Objetivos de las pruebas.

El principal objetivo de las pruebas de software es encontrar defectos y/o errores desde el inicio del proceso de desarrollo de software. Algunos otros objetivos son:

- Mejorar la calidad de los productos de software
- Reducir esfuerzo, tiempo y costos en los procesos del desarrollo de software, en especial en la etapa de mantenimiento
- Mejorar la satisfacción de los usuarios y de los clientes.

1.1.2 Principios de las pruebas.

A continuación encontraremos los principios básicos que guían las pruebas de software:

- Al momento de descubrir un error la prueba debe permitir identificar el requerimiento del usuario que se esté incumpliendo.

- Las pruebas deben planificarse mucho antes de que empiecen. La planificación de las pruebas puede empezar tan pronto como esté completo el levantamiento de requerimientos y la definición detallada de los casos de prueba cuando se tenga consolidado el modelo de diseño.
- Las pruebas deben empezar por «lo pequeño» y progresar hacia «lo grande». Las primeras pruebas planeadas y ejecutadas se centran generalmente en módulos individuales del programa. A medida que se avanza en éstas, el enfoque de las pruebas cambia en un intento de encontrar nuevos errores relacionados con la integración de éstos módulos y finalmente con la interacción del sistema completo.
- No son posibles las pruebas exhaustivas. El número de permutaciones de caminos para incluso un programa de tamaño moderado es demasiado grande, por lo cual, es imposible ejecutar todas las combinaciones de caminos durante las pruebas. Es posible, sin embargo elegir y ejecutar una serie de caminos lógicos importantes que permitan probar adecuadamente el software.
- Para ser más eficaces, las pruebas deberían ser realizadas por un equipo independiente. El ingeniero de software que creó el sistema no es el más indicado para realizar las pruebas debido a que consciente o inconscientemente puede omitir casos de prueba importantes que conlleven a descubrir nuevos errores. Por consiguiente, es recomendable organizar un grupo de trabajo independiente para las pruebas que suministre una visión más objetiva del software.

1.2 DISEÑO DE CASOS DE PRUEBA

Los casos de prueba deben ser diseñados con la mayor probabilidad de encontrar un gran número de errores, con la mínima cantidad de tiempo y esfuerzo posible. Al diseñar los casos de prueba se debe indicar cuáles son los procedimientos, las técnicas y medidas que se debe aplicar, como también los datos de prueba que se usarán y los resultados esperados que se producirán. Esto se hace con el objetivo de demostrar que la prueba cumple con la verificación y validación del requisito.

Los productos de ingeniería se pueden probar con una de estas dos formas:

- Conociendo la función específica del producto. Se ejecutan pruebas que demuestren que cada función es completamente operativa, y al mismo tiempo se busca errores en cada función; este enfoque de prueba se llama prueba de caja negra.
- Conociendo el funcionamiento del producto. Se desarrollan pruebas que aseguren que la operación interna se ajusta a las especificaciones, comprobando de forma adecuada todos los componentes internos; este enfoque de prueba se llama prueba de caja blanca.

La prueba de caja negra se lleva a cabo sobre la interfaz de software. Este tipo de prueba demuestra que las funciones del software son operativas, que la entrada se acepta adecuadamente produciendo un resultado correcto. La prueba de caja blanca examina minuciosamente los detalles procedimentales, comprobando los caminos lógicos del software y ejercitando conjuntos específicos de condiciones.

1.3 PRUEBA DE CAJA BLANCA

La prueba de caja blanca es también llamada prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Se pueden obtener casos de prueba que cumplan con siguientes actividades:

- Garantizar que se ejecuta cada camino independiente de todos los módulos por lo menos una vez
- Ejercitar todas las decisiones lógicas
- Ejecutar en sus límites y con sus límites operacionales
- Ejercitar las estructuras internas de datos.

Aunque las pruebas de caja blanca se pueden aplicar en diferentes estrategias de pruebas (unitaria, integración y sistema), fundamentalmente se aplica en el ámbito de las pruebas unitarias.

1.4 PRUEBA DEL CAMINO BÁSICO

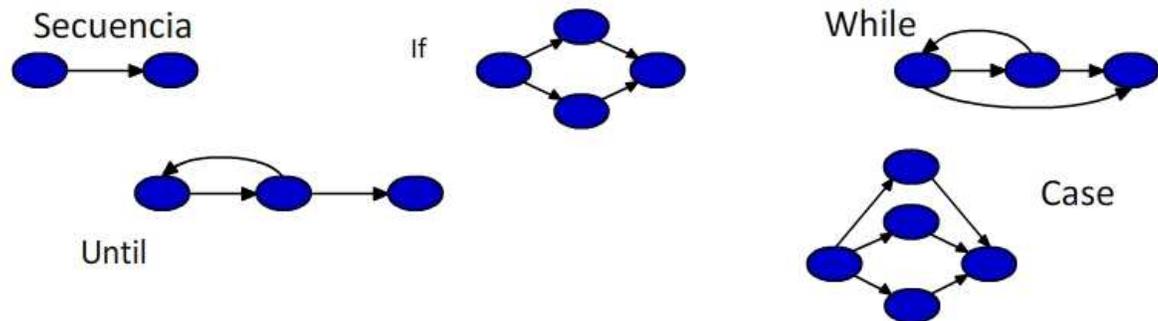
La prueba de camino básico es un método de caja blanca propuesto por McCabe, que permite tener una medida de la complejidad del diseño

procedimental y utilizar esta medida para la definición de caminos básicos de ejecución, que se utilizan para diseñar los casos de prueba y aseguran que cada camino sea ejecutado al menos una vez.

1.4.1 Notación del grafo de flujo.

El grafo de flujo o grafo del programa representa el flujo de control lógico y utiliza la siguiente notación:

Ilustración 1. Notación del grafo de flujo



Fuente: Pressman.

1.4.2 Complejidad ciclomática.

La complejidad ciclomática es una métrica que proporciona una medición de la complejidad lógica del software. Cuando se usa esta métrica en el método del camino básico, el valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y nos da el número límite de pruebas que se deben ejecutar para garantizar que todas las sentencias sean ejecutadas al menos una vez.

Un camino independiente es aquel camino del programa que introduce un nuevo conjunto de instrucciones o una nueva condición.

1.5 PRUEBA DE LA ESTRUCTURA DE CONTROL

La prueba de camino básico descrita anteriormente, es una de las muchas técnicas para la prueba de la estructura de control. Estas técnicas mejoran la calidad y amplían la cobertura de la prueba de caja blanca.

1.5.1 Prueba de condición.

La prueba de condición es un método de diseño de casos de prueba que ejecuta las condiciones lógicas que existen en el módulo de un programa. Se basa en probar cada una de las condiciones del programa, buscando detectar errores no solo en las condiciones, sino también otro tipo de errores en el programa.

Una condición simple es una expresión relacional o una variable lógica, algunas veces precedida de un operador NOT. Una condición compuesta está formada por dos o más condiciones simples, paréntesis y operadores lógicos (NOT, OR, AND). Una expresión lógica es una condición sin expresiones relacionales.

Las estrategias de pruebas de condiciones, tiene como ventaja que la cobertura de la prueba de condición es más sencilla y sirve de guía para generar pruebas adicionales del programa. La estrategia de prueba de condiciones más sencilla es, la prueba de ramificación y consiste en ejecutar al menos una vez la rama verdadera y falsa y cada condición simple de la condición compuesta. La prueba del domino consiste en realizar tres o cuatro pruebas a una expresión racional. La técnica BRO¹ (Prueba del Operador Relacional y de Ramificación), detecta los errores de ramificaciones y operadores relacionales en una condición, en las cuales, todos los operadores relacionales y las variables lógicas de la condición, aparecen solo una vez y no tienen variables en común.

¹ En Ingles, BranLh arid Relational Operator

1.5.2 Prueba del flujo de datos.

Este método selecciona caminos de prueba, según el uso de las variables y la ubicación de las definiciones del programa.

1.5.3 Prueba de bucles.

La prueba de bucles es una técnica de la prueba de caja blanca, la cual se centra en validar la construcción de los bucles. Estos se pueden definir en cuatro clases diferentes:

Bucles simples. A este tipo de bucles se les debe aplicar el siguiente conjunto de pruebas, donde n es el número máximo de pasos permitidos en el bucle:

- Pasar completamente por alto el bucle
- Pasar por el bucle una sola vez
- Pasar por el bucle dos veces
- Hacer m pasos por el bucle con $m < n$
- Hacer $n-1$, n y $n+1$ pasos por el bucle.

Bucles anidados. En este tipo de bucles, el número de pruebas aumentaría geométricamente a medida que aumenta el nivel de anidamiento provocando un número impracticable de pruebas. Para reducir este número de pruebas, se debe aplicar el siguiente enfoque:

- Comenzar con el bucle más interior y configurar los demás bucles con sus valores mínimos
- Realizar las pruebas de bucles simples al bucle más interior y también realizar una prueba con valores fuera del rango permitido
- Progresar hacia afuera, realizando pruebas en los bucles siguientes
- Continuar hasta probar todos los bucles.

Bucles concatenados. Los bucles concatenados se pueden probar con el enfoque definido para los bucles simples, siempre y cuando estos bucles sean independientes del resto.

Bucles no estructurados. Esta clase de bucles se deben rediseñar para que se ajusten a las construcciones de programación estructurada, siempre que sea posible.

1.6 PRUEBA DE CAJA NEGRA

Las pruebas de caja negra son también llamadas pruebas de comportamiento, ya que se centra en los requisitos funcionales del software. Esta prueba trata de descubrir diferentes tipos de errores a los que se encuentra por medio de los métodos de caja blanca. Estos errores se clasifican en las siguientes categorías:

- Funciones ausentes e incorrectas
- Errores de interfaz
- Errores en estructuras de datos o en accesos a bases de datos externas
- Errores de rendimiento
- Errores de inicialización y de terminación.

1.6.1 Métodos prueba basados en grafos.

El primer paso en la prueba de caja negra, es entender los objetos que se modelan en el software y las relaciones que hay entre estos. La prueba de software crea un grafo de objetos con sus relaciones y luego diseña una serie de pruebas que cubran todo el grafo de tal manera que todos los objetos y sus relaciones se ejerciten para descubrir los errores.

El ingeniero de software inicia creando un grafo -colección de nodos que representan objetos; enlaces que representan las relaciones entre los objetos; pesos de nodos que describen las propiedades de un nodo y pesos de enlaces que describen alguna característica de un enlace-. Los nodos se representan como círculos conectados por enlaces que toman diferentes formas como por ejemplo:

- Enlace dirigido: este tipo de enlace se representa por una flecha e indica que una relación se mueve sólo en una dirección.

- Enlace bidireccional: también se denomina enlace simétrico, el cual indica que la relación se aplica en ambos sentidos.
- Enlace paralelo: este tipo de enlace se usa cuando entre los nodos del grafo se establecen diferentes relaciones.

Las pruebas basadas en grafos inician con la definición de todos los nodos y sus respectivos pesos, es decir, se identifican los objetos y los atributos. Una vez identificado los nodos, se deben establecer los enlaces y sus pesos. Algunas veces, el modelo de grafos puede contener bucles, lo cual significa que el grafo posee un camino en donde se encuentra uno o más nodos más de una vez.

1.6.2 Participación equivalente.

Este método de prueba de caja negra, consiste en dividir el campo de entrada de un programa en clases de datos, de los cuales surgen casos de prueba que descubren clases de errores, reduciendo el número total de casos de prueba que hay que desarrollar.

Para la partición equivalente, los casos de prueba se basan en evaluar en una condición de entrada las clases de equivalencia, las cuales representan un conjunto de estados válidos o no válidos para estas condiciones. Una condición de entrada puede ser un rango de valores, un valor numérico específico, una condición lógica o un conjunto de valores relacionados.

Las clases de equivalencias se pueden definir así:

- Si la condición de entrada es un rango, se define una clase de equivalencia válida y dos no válidas.
- Si la condición de entrada es un valor específico, se define una clase de equivalencia válida y dos no válidas.
- Si la condición de entrada hace parte de un conjunto, se define una clase de equivalencia válida y una no válida.

- Si la condición de entrada es una condición lógica, se define una clase de equivalencia válida y una no válida.

1.6.3 Análisis de Valores Límite.

Los errores tienden a presentarse más en los límites del campo de entrada que en el centro, por esta razón, se ha desarrollado la técnica de prueba Análisis de Valores Límites (AVL), la cual lleva a elegir casos de prueba que ejerciten los valores límite. AVL, es una técnica de diseño de casos de prueba, la cual complementa a la partición equivalente ya que esta técnica elige casos de prueba en los extremos de la clase y obtiene también casos de prueba para el campo de salida.

Directrices de AVL:

- Si la condición de entrada es un rango delimitado por los valores a y b, se diseñan casos de prueba para a y b y los valores por encima y por debajo de a y b respectivamente.
- Si la condición de entrada es un número de valores, se diseñan casos de prueba para los valores máximo y mínimo y también para probar los valores por encima y por debajo de estos.
- Las directrices 1 y 2 aplicadas a las condiciones de salida, diseñando casos de prueba que produzcan el máximo y el mínimo número permitido de entradas.
- Cuando las estructuras de datos internas, poseen un límite determinado, se debe diseñar un caso de prueba que ejercite la estructura de datos en sus límites.

1.6.4 Prueba de comparación.

La prueba de comparación o también llamada prueba mano a mano, es una técnica de prueba de caja negra, la cual se basa en el desarrollo de versiones independientes.

Cuando se producen diferentes implementaciones con las mismas especificaciones, se le proporciona como entrada a cada versión los casos de prueba diseñados mediante alguna de las técnicas de caja negra. Si las salidas producidas de las diferentes versiones son idénticas, se concluye que son correctas todas las implementaciones; si son diferentes, se realiza una investigación en todas las aplicaciones para buscar el defecto que produjo la diferencia en las versiones.

1.6.5 Prueba de la tabla ortogonal.

El método de prueba de la tabla ortogonal, permite diseñar casos de prueba con una cobertura más amplia y con un número de casos de prueba razonable. Este tipo de prueba, se podría aplicar cuando crece el número de valores de entrada y cuando incrementa el número de valores diferentes para cada elemento de dato

1.7 PRUEBAS DE ENTORNOS ESPECIALIZADOS, ARQUITECTURAS Y APLICACIONES

Los ingenieros de software pueden encontrarse entornos, arquitecturas y aplicaciones especializadas para las cuales los métodos de caja blanca y caja negra no son aplicables, necesitando pruebas con enfoques únicos.

- Prueba de Interfaces Gráficas de Usuarios (IGUs): la complejidad de las IGUs ha aumentado, dificultando el diseño y la ejecución de los casos de prueba. En los últimos años, una amplia lista de herramientas automáticas de prueba de IGU han aparecido en el mercado.
- Prueba de Arquitectura Cliente/Servidor.
- Prueba de la documentación y facilidades de ayuda: esta prueba se divide en dos etapas. La primera etapa es "la revisión e inspección", en la cual se examina el documento para comprobar la claridad editorial. La segunda etapa es "la prueba en vivo", en la cual se utiliza el programa real junto con la documentación.

- Pruebas de sistemas de tiempo real: el diseñador de los casos de prueba, no solo debe considerar los casos de prueba de caja negra y caja blanca si no también los casos de prueba que tratan los sucesos, como lo es una interrupción. Algunas veces, cuando se proporciona un dato de prueba al sistema real, estando en un determinado estado, dará un proceso correcto y si lo proporcionamos en otro estado dará un error. Existe una estrategia para diseñar casos de prueba para sistemas de tiempo real:
 - ✓ Prueba de tareas: se diseñan casos de prueba de caja negra y caja blanca y para cada tarea se ejecutan independientemente, probándose cada una de ellas. Por medio de estas pruebas se descubren errores en el funcionamiento y en la lógica pero no los errores de comportamiento ni de temporización.
 - ✓ Prueba de comportamiento: consiste en probar cada suceso (interrupciones del usuario, interrupciones mecánicas, interrupciones del sistema, modos de fallo) independientemente, observando el comportamiento del sistema y así, detectar errores producidos por el proceso asociado a esos sucesos. Luego, se le presentan sucesos al sistema, con una frecuencia y orden aleatorio, y se detectan los errores de comportamiento del software.
 - ✓ Pruebas intertareas: detectados los errores en las tareas individuales y en el comportamiento del sistema, la prueba se dirige hacia los errores relacionados con el tiempo. Para determinar errores de sincronización entre las tareas, se prueban las tareas asíncronas que se comunican con otras, con diferentes cargas de proceso y tasas de datos. Para detectar errores en el tamaño de las áreas de almacenamiento de datos, se prueban las tareas que se comunican mediante almacenes de datos o colas de mensajes.
 - ✓ Prueba del sistema: este tipo de pruebas, trata de descubrir errores en la interfaz software/hardware. El responsable de esta prueba desarrolla una lista de las posibles interrupciones y de la consecuencia de esta interrupción.

1.8 ESTRATEGIAS DE PRUEBAS DE SOFTWARE

Una estrategia de prueba da como resultado una correcta construcción del software, al unir las técnicas de diseño de casos de prueba en una serie de pasos planificados. Una estrategia básica de prueba debe incorporar: la planificación de la prueba, el diseño de casos de prueba, la ejecución de las pruebas y la agrupación y evaluación de los datos resultantes.

En la ingeniería de software, se debe definir una plantilla para las pruebas de software, en las cuales se deben planificar y llevar a cabo un conjunto de actividades sistemáticamente. Las plantillas, generalmente tienen las siguientes características:

- Las pruebas inician a nivel de modulo y trabajan hacia afuera, logrando la integración de todo el sistema.
- La prueba es llevada a cabo por el responsable del desarrollo del software y para grandes proyectos, un grupo de pruebas independiente.
- A pesar de que la prueba y la depuración son actividades diferentes, la depuración debe ir incluida en cualquier estrategia de prueba.

Las estrategias de pruebas deben incluir pruebas de bajo nivel, las cuales verifican que todos los pequeños segmentos de código se hayan implementado correctamente y también, pruebas de alto nivel, las cuales validan las principales funciones del sistema frente a los requerimientos del cliente.

1.8.1 Validación y verificación.

La validación, se refiere a un conjunto de diferentes actividades, las cuales aseguran que el software desarrollado, cumple con los requisitos del cliente. La verificación, se refiere al conjunto de actividades, las cuales aseguran que el software implementa una función específica, correctamente.

La validación y la verificación hacen parte de la lista de actividades para garantizar la calidad del software, como también lo son: auditorías de calidad, simulación, revisión de la documentación, pruebas de desarrollo, validación e instalación. La calidad en el software se incorpora durante el proceso de ingeniería de software, aplicando adecuadamente los métodos y herramientas, y confirmándose durante el proceso de pruebas.

1.8.2 Organización para las pruebas de software.

En los proyectos de software, se pide a los desarrolladores que prueben el software que ellos construyeron. Estos programadores tienen un gran interés en demostrar que el programa se encuentra libre de errores y que cumple con los requerimientos del cliente. Desde el punto de vista del constructor, las pruebas tratan de romper lo que el ingeniero ha construido, por lo cual este diseña y ejecuta pruebas que demuestren que el programa funciona correctamente, en lugar de detectar errores, errores que encontrara el cliente.

El desarrollador, siempre es el responsable de probar los módulos del programa individualmente, asegurándose de que cada uno realice su función para la cual fue diseñada. En muchos casos, también es el encargado de la prueba de integración, llegando a la construcción total de la estructura del sistema.

La función principal del Grupo Independiente de Prueba (GIP) es eliminar los problemas asociados al hecho de que el desarrollador pruebe lo que ha construido, eliminado así el conflicto de intereses mediante una prueba independiente. El responsable del desarrollo de software, trabaja estrechamente con el GIP a lo largo del proyecto, para asegurar que se realizan pruebas exhaustivas.

1.8.3 Una estrategia de prueba de software.

La estrategia de prueba del software, se puede ver como una espiral, comenzando con la prueba de unidad en el vértice de la espiral, centrándose en cada unidad de software. Avanzando hacia afuera, se llega a la prueba de integración, donde la atención se centra en el diseño y la construcción de la arquitectura del software. Siguiendo hacia afuera, en la otra vuelta de la espiral encontramos la prueba de validación, donde los requisitos establecidos en el análisis, son validados comparándolos con el sistema que

se ha construido. Finalmente, se llega a la prueba del sistema, en la que el software se prueba como un todo.

Desde el contexto de la ingeniería de software, la estrategia de prueba son cuatro pasos secuenciales:

- Prueba de unidad, se centra en cada módulo individualmente para asegurar que funcionan correctamente. En esta prueba se usan las técnicas de caja blanca, ejercitando caminos específicos para asegurar un completo alcance y una máxima detección de errores.
- Prueba de integración, mediante la cual prueba se ensamblan los módulos para formar completamente el paquete de software. Durante esta integración, se utilizan los métodos de caja negra para el diseño de casos de pruebas, también llevando a cabo algunas pruebas de caja blanca para asegurar que se cubran los principales caminos de control.
- Pruebas de validación, donde se comprueban los criterios de validación establecidos en el análisis de requisitos, asegurando que el software satisface los requisitos de comportamiento, funcionales y de rendimiento. Durante esta prueba, solo se usan técnicas de caja negra.
- Prueba del sistema, donde se combina el software con otros elementos del sistema como lo son el hardware, base de datos y gente. Esta prueba verifica que cada elemento encaja adecuadamente, alcanzado la funcionalidad y el rendimiento del sistema total.

1.8.4 Aspectos estratégicos.

Si se desea implementar una estrategia de prueba del software con éxito, se deben abordar los siguientes puntos:

- Antes de comenzar las pruebas, se debe especificar de manera cuantificable los requisitos del producto: características de calidad como la portabilidad, facilidad de mantenimiento y de uso, deben ser

evaluadas en la estrategia de prueba, especificándose de manera medible para que los resultados de la prueba no sean ambiguos.

- Los objetivos de la prueba se deben establecer de manera explícita: dentro de la planificación de la prueba se deben establecer en términos medibles, objetivos específicos de la prueba como por ejemplo la efectividad, la cobertura, el costo para encontrar y arreglar errores, tiempo medio o frecuencia de ocurrencia de fallo y horas de trabajo por prueba.
- Desarrollar un perfil para cada categoría de usuario que va a manejar el software.
- Desarrollar un plan de prueba de ciclo rápido: un equipo de ingeniería de software debe aprender a probar en ciclos rápidos (2 por 100 del esfuerzo del proyecto); utilizándose para controlar los niveles de calidad y las correspondientes estrategias de prueba.
- Construir un software diseñado para probarse a sí mismo: el software deberá ser capaz de detectar ciertas clases de errores, diseñándose de manera que use técnicas de depuración, además de incluir pruebas de regresión y automatizadas.
- Realizar revisiones técnicas formales antes de la prueba: las técnicas formales pueden ser efectivas como las pruebas a la hora de descubrir errores, reduciendo el esfuerzo de prueba necesaria para producir software de alta calidad.
- Evaluar la estrategia y casos de prueba, realizando revisiones técnicas formales: las revisiones técnicas formales, ahorran tiempo y mejoran la calidad del producto mediante la detección de inconsistencias, errores y omisiones.
- Enfoque de mejora continua al proceso de prueba.

1.9 PRUEBA DE UNIDAD

Las pruebas de unidad se encargan de probar cada uno de los módulos por separados, sin tener en cuenta la interrelación con otros módulos. Estos

módulos deben ser unidades estructurales encargados de una tarea específica, en programación procedural u orientada a objetos estas unidades son métodos o funciones definidas.

Las pruebas unitarias son una buena práctica que garantiza la obtención de software robusto y más fácilmente sostenible. Dentro de las pruebas de unidad se realizan tareas que permitan determinar que cada modulo se ejecuta correctamente. Se prueba la interfaz del modulo para asegurar que la información fluye de forma adecuada, que los datos ingresen bien, de lo contrario las demás pruebas no tendrían sentido. Se prueba la estructura de los datos, para asegurar que los datos locales no pierdan integridad durante la ejecución del modulo y de ser posible el impacto de los datos globales sobre el modulo. Se prueban las condiciones limites para determinar que el modulo funciona de acuerdo a las restricciones que se han fijado. Se ejercitan los caminos independientes para garantizar que todas las sentencias sean ejecutada por lo menos una vez.

Entre los errores que se presentan con más frecuentes en las pruebas de unidad, están: (1)precedencia aritmética incorrecta o mal interpretada; (2) operaciones mezcladas; (3) inicializaciones incorrectas; (4) falta de precisión; (5) incorrecta representación simbólica de una expresión. Los casos de prueba deben descubrir errores como: (1) comparaciones ente tipos de datos distintos; (2) operadores lógicos o de precedencia incorrectos; (3) igualdad esperada cuando los errores de precisión la hacen poco probable; (4) variables o comparaciones incorrectas; (5) terminación de bucles inapropiada o inexistente; (6) fallo de salida cuando se encuentra una iteración divergente; (7) variables de bucles modificadas de forma inapropiada

Procedimientos de prueba de unidad.

Para llevar a cabo el proceso de pruebas de unidad se debe tener en cuenta que un modulo no es un programa independiente, por tal razón es necesario crea un software que controle, es decir que acepte los datos del caso de prueba, los pase al modulo que está siendo probado e imprime los resultados importantes.

Algunos módulos, debido a que no es posible desarrollar el controlador deben esperar para ser probados en la etapa de pruebas de integración.

La prueba de unidad se simplifica cuando se diseña un módulo con un alto grado de cohesión. Cuando un módulo sólo realiza una función, se reduce el

número de casos de prueba y los errores se pueden predecir y descubrir más fácilmente.

1.10 PRUEBAS DE INTEGRACIÓN

Las pruebas de integración se realizan con el fin de detectar errores asociados con la interacción de los módulos que están relacionados entre sí y que han sido probados en la etapa anterior mediante pruebas de unidad. Con el uso de estas pruebas, se va formando el programa a medida que se comprueba como los distintos componentes interaccionan y se comunican libres de errores.

Los errores más comunes que se encuentran a la hora de aplicar pruebas de integración son:

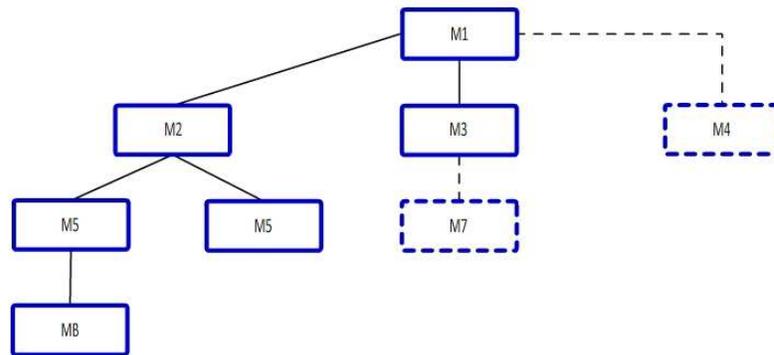
- Comunicación a través de la interface
- Efectos colaterales perniciosos
- Acumulación notable de errores de cálculo
- Acceso incoherente a estructuras de datos globales
- Tiempos de respuesta.

1.10.1 Integración descendente.

La integración descendente es un método para la construcción de la estructura de programas, el cual consiste en integrar los módulos dirigiéndose hacia abajo de la jerarquía de control. Al modulo control principal se van incorporando los módulos subordinados, por medio de los enfoques de integración primero en profundidad o primero en anchura.

- Primero en profundidad: integra todos los módulos de una ruta de control principal de la estructura del programa. Por ejemplo, si se elige el camino de la izquierda, se integrarían primero los módulos M1, M2 y M5, seguido de M8 y si es necesario para M2 se integra M6.
- Primero en anchura: incorpora todos los componentes directamente subordinados en cada nivel, desplazándose horizontalmente por la estructura. Según la figura, se integran primero los módulos M2, M3 y M4, seguidos de M5 y M6, etc.

Ilustración 2. Integración descendente



Fuente: Pressman.

El proceso de integración descendente se realiza en una serie de cinco pasos:

1. Se usa el módulo de control principal como controlador de la prueba, disponiendo de resguardos para todos los módulos directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración elegido (es decir, primero en profundidad o primero en anchura) se van sustituyendo uno a uno los resguardos subordinados por los módulos reales.
3. Cada vez que se integra un nuevo módulo se llevan a cabo pruebas.
4. Tras terminar cada conjunto de pruebas, se reemplaza otro resguardo con el módulo real.
5. Se hace la prueba de regresión para asegurarse de que no se han introducido errores nuevos.

El proceso continúa desde el paso dos hasta que se haya construido la estructura del programa entero.

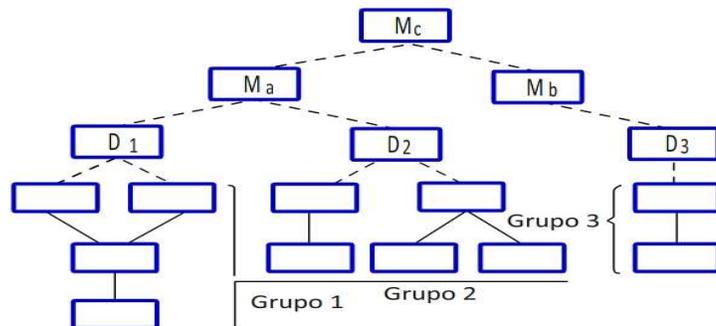
1.10.2 Integración ascendente.

En la integración ascendente empieza la construcción y la prueba desde los módulos de los niveles más bajos de la estructura de control. Dado que los módulos se integran de abajo hacia arriba, el proceso requerido de los módulos subordinados siempre está disponible y se elimina la necesidad de resguardos.

Una estrategia de integración ascendente, se puede implementar mediante los siguientes pasos:

1. Se combinan en grupos los módulos de bajo nivel, que realicen una subfunción específica del software.
2. Se escribe un programa de control de la prueba para coordinar la entrada y la salida de los casos de prueba.
3. Se prueba el grupo.
4. Se eliminan los controladores y se combinan los grupos moviéndose hacia arriba por la estructura del programa.

Ilustración 3. Integración ascendente



Fuente: Pressman.

1.10.3 Prueba de regresión.

Las pruebas de regresión son aquellas que intentan descubrir las causas de nuevos errores, carencias de funcionalidad, o divergencias funcionales con

respecto al comportamiento esperado del software, que son producidos por cambios recientes en el software.

El conjunto de pruebas de regresión contiene tres clases diferentes de casos de prueba:

- Una muestra representativa de pruebas que ejercite todas las funciones del software
- Pruebas adicionales que se centran en las funciones del software que se van a ver probablemente afectadas por el cambio
- Pruebas que se centran en los componentes del software que han cambiado.

A medida que el proceso de pruebas avanza, las pruebas de regresión serán cada vez más numerosas, por esta razón se hace necesario que se haga un buen diseño y así realizar solo las pruebas que traten una o más clases de errores en cada una de las funciones principales del programa.

1.10.4 Prueba de humo.

Este tipo de pruebas de integración se utilizan cuando se está desarrollando un software empaquetado, para proyectos que cuenta con poco tiempo. Las pruebas de humo comprenden las siguientes actividades:

- Integrar en una construcción los componentes de software que han sido traducidos a código
- Diseñar una serie de pruebas para descubrir errores que impiden a la construcción realizar su función adecuadamente
- Es habitual en la prueba de humo que la construcción se integre con otras construcciones y que se aplique una prueba de humo al producto completo.

Beneficios de las pruebas de humo en proyectos de ingeniería de software complejos y críticos por su duración:

- Se minimizan los riesgos de integración. Errores bloqueantes e incompatibilidades son encontrados rápidamente, debido a la frecuencia de las pruebas de humo.
- Se perfecciona la calidad del producto final. Se pueden descubrir errores funcionales y defectos de diseño a nivel de componente y de arquitectura.
- Se simplifican el diagnóstico y la corrección de errores.
- El progreso es fácil de observar. Cada día que pasa, se integra más software y se demuestra que funciona.

1.11 PRUEBAS DE VALIDACIÓN

Las pruebas de validación consisten en demostrar la conformidad de los requisitos del cliente. Este proceso se desarrolla mediante pruebas de caja negra.

El plan de pruebas para las pruebas de validación tiene como objetivo asegurar el cumplimiento de los requisitos funcionales y de rendimiento, y que la documentación este correcta y clara. Con los errores encontrados en esta prueba, se hace una lista para llegar a un acuerdo con el cliente y resolver las diferencias.

Pruebas alfa y beta.

Estas pruebas son desarrolladas para descubrir errores que en apariencia solo el usuario final podría encontrar.

Pruebas alfas: son llevadas a cabo por un cliente en el lugar del desarrollo, en donde el desarrollador observa los errores.

Pruebas beta: son llevadas a cabo por los clientes en sus lugares de trabajo sin presencia de desarrolladores; estos son informado de los errores que

encuentran los clientes en informes periódicos, de esta manera los desarrolladores hacen modificaciones al software y prepara la versión para ser entrega.

1.12 PRUEBAS DE SISTEMA

Las pruebas de sistema son una serie de pruebas, todas con propósitos diferentes, pero con el mismo fin de verificar la integración del software desarrollado con todo el sistema (hardware, sistema operativo e información) y la realización de las funciones esperadas, con base a los requerimientos. Las pruebas de sistemas son ejecutadas sobre el software terminado, quedando restringido el acceso de los desarrolladores para hacer cambios en el código.

1.12.1 Prueba de recuperación.

La prueba de recuperación, fuerza el fallo del software de varias maneras y verifica que la recuperación se realice adecuadamente sin cesar el funcionamiento de todo el sistema. La recuperación puede ser llevada a cabo por el propio sistema (automática), evaluando los mecanismos de recuperación del estado del sistema, de los datos y del proceso de arranque. La recuperación podría requerir la intervención humana, teniendo en cuenta que los Tiempos Medios de Reparación (TMR) se encuentren dentro de unos límites aceptables.

1.12.2 Prueba de seguridad.

La prueba de seguridad, verifica que los mecanismos de protección implementados en el sistema lo protejan de entradas impropias e ilegales. Durante esta prueba, el responsable debe tratar de entrar en el sistema a toda cabalidad intentando conseguir las claves de acceso, atacando el sistema con software diseñado para romper la defensa, bloqueando el sistema para que otras personas no puedan acceder al servicio, produciendo errores en al sistema a propósito, accediendo durante la recuperación. El diseñador del sistema, debe hacer costar más la entrada ilegal al sistema que la información obtenida.

1.12.3 Prueba de resistencia (stress).

En la prueba de resistencia, se ejecuta el sistema de forma que demande en gran cantidad de recursos, frecuencia o volúmenes anormales; esto se puede conseguir realizando pruebas que generen los siguientes casos:

- Generar 10 interrupciones por segundo mediante una prueba
- Incrementar la frecuencia de datos de entrada para corroborar como responden las funciones de entrada
- Ejecutar pruebas que requieran el máximo de memoria y de otros recursos
- Causar problemas en un sistema operativo virtual
- Producir excesivas búsquedas de datos residentes en disco.

La prueba de sensibilidad, es una técnica de la prueba de resistencia la cual intenta descubrir combinaciones de datos como entrada válida, que produzca inestabilidad o un proceso incorrecto.

1.12.4 Pruebas de rendimiento.

Las pruebas de rendimiento son realizadas con el fin de determinar cómo responde el sistema ante ciertas cargas y validar atributos de calidad, además que cumpla con los criterios de rendimientos, puede medir que partes del sistema o que carga hacen que el sistema rinda de forma incorrecta.

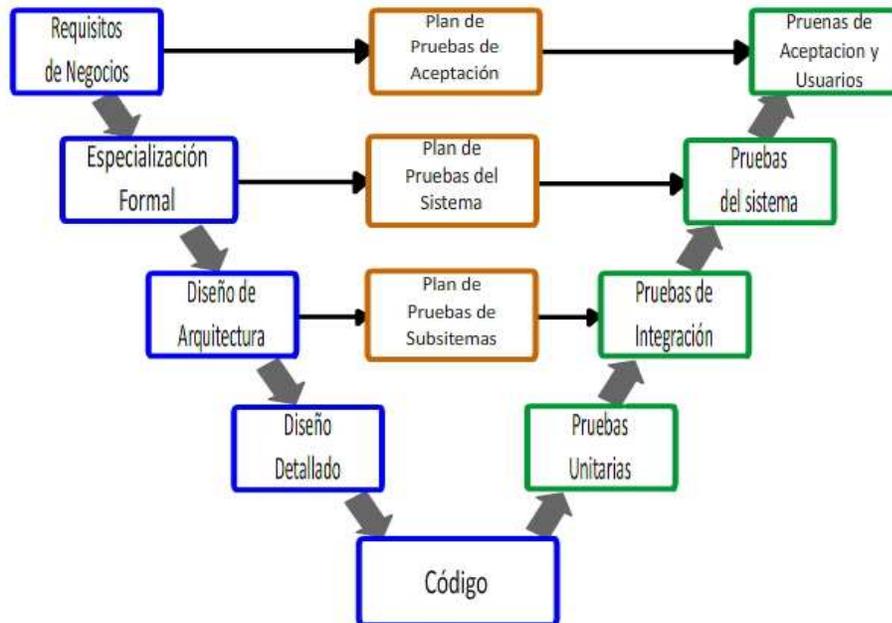
1.13 METODOLOGÍAS DE PRUEBAS DE SOFTWARE

Las metodologías de pruebas muestran las diferentes relaciones entre las actividades del ciclo de vida del desarrollo de software y el proceso de pruebas.

1.13.1 Metodología en V.

En esta metodología se considera que el proceso de pruebas de software se desarrolla en paralelo con el análisis y el desarrollo.

Ilustración 4. Metodología en V



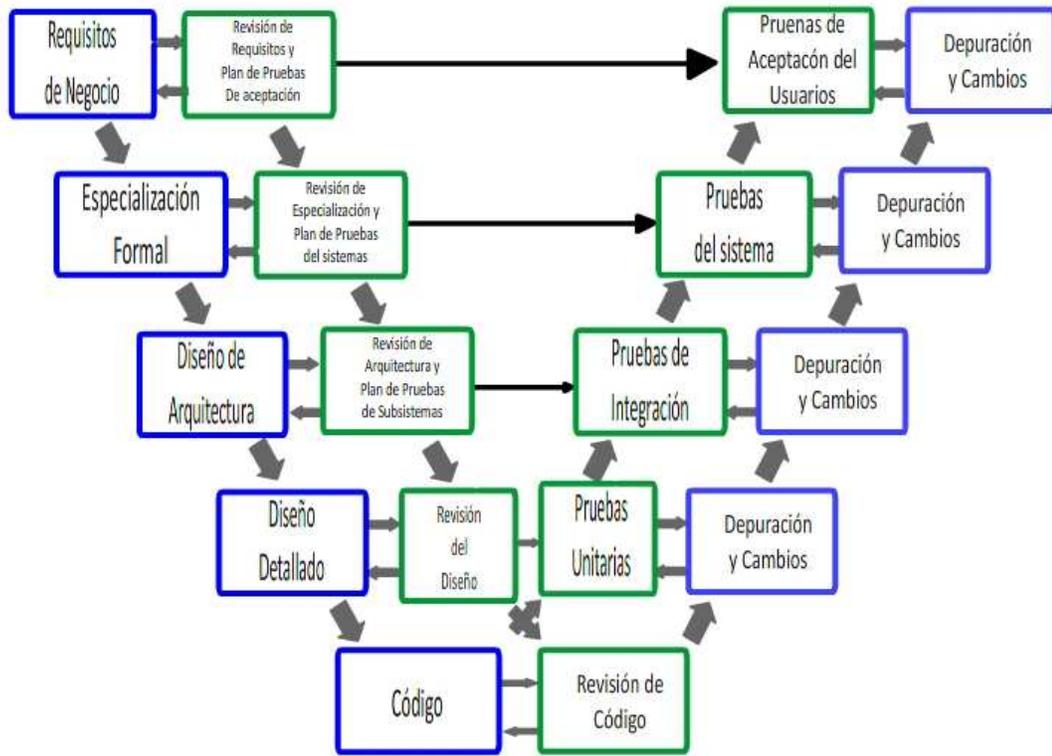
Fuente: Kynetia.

El lado izquierdo de la V representa las fases del desarrollo de software y la creación de especificaciones del sistema. El lado derecho de la V representa los niveles correspondientes a la pruebas. Los niveles superiores están relacionados con pruebas de caja negra y los niveles inferiores en caja blanca. Este modelo posee gran cantidad de iteraciones y es muy rígido.

1.13.2 Metodología en W.

La metodología en W surge como una variación de la metodología en V, mejorando la interdependencia de los equipos de pruebas y los equipos de desarrollo, e incluyendo actividades como son, revisión de requerimientos, la revisión de la arquitectura y del diseño de detalle y las revisiones de código, en las primeras etapas del desarrollo y tareas de pruebas, depuración y corrección de errores en etapas finales.

Ilustración 5. Metodología en W



Fuente: Kynetia.

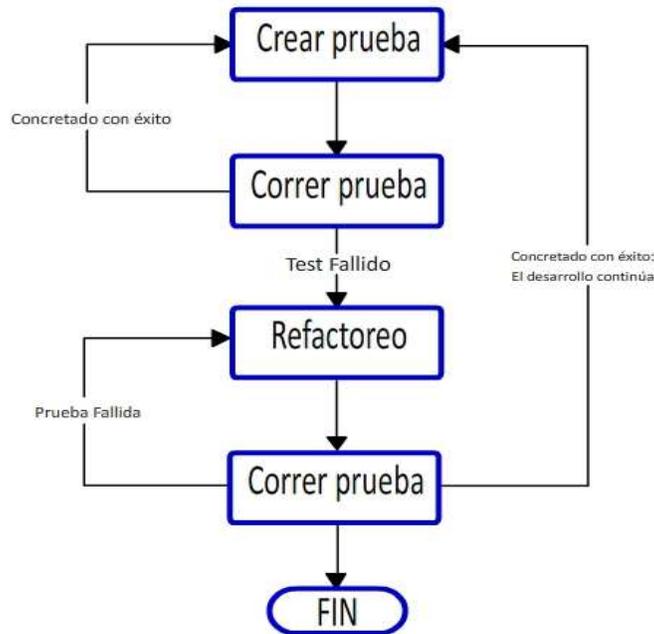
Los bloques azules representan las actividades del ciclo de vida del desarrollo de software, los bloques verdes las labores de pruebas de software.

1.13.3 TDD desarrollo basado en pruebas (Test-Driven Development).

TDD es una metodología que basa su área de trabajo en las pruebas de software, consiste en diseñar las pruebas antes de empezar a codificar y luego ir superándolas.

Ciclos TDD:

Ilustración 6. Ciclos TDD



Fuente: Kynetia.

Crear una prueba: consiste en la codificación que representa una prueba para un requerimiento a superar.

Creación de código y refactorio: en esta parte se modela la aplicación, pensando en que debe superar las pruebas ya diseñadas.

Ejecutar la prueba: se corren las pruebas para verificar que no existan fallos, en caso contrario debemos refactorizar y volver a ejecutar las pruebas.

Crear nuevas pruebas: se desarrolla otras pruebas que dan inicio el ciclo TDD y permite seguir con el desarrollo.

1.14 PRUEBAS SEGÚN EL MODELO DE DESARROLLO

Las pruebas han adquirido relevancia en el desarrollo de software, teniendo objetivos diferentes según el modelo de desarrollo. En las metodologías convencionales se usan como validación del producto desarrollado y en las

metodologías ágiles se utilizan en sustitución de las especificaciones de requisitos y como guía para el desarrollo de software

1.14.1 Actores.

En las metodologías ágiles, los desarrolladores son los encargados de ejecutar las pruebas de caja blanca, ya que están basadas en código fuente, y son ellos los que trabajan a este nivel y conocen la estructura del software, por lo tanto pueden definir los casos de prueba adecuados para probar el código. Si se produce un error, conocen la línea de código afectada por el caso de prueba que falló. En las metodologías convencionales son los ingenieros de prueba los encargados de definir los casos de prueba y los responsables de ejecutarlos, son los probadores.

En las pruebas de caja negra se debe tener una visión externa del producto, por esto, estas pruebas son llevadas a cabo principalmente por los probadores en las metodologías convencionales, pero también colaboran los desarrolladores, ya que ellos son los que conocen los tipos de datos de cada uno de los parámetros

1.14.2 Metodologías convencionales.

Cuando hablamos de pruebas en una metodología convencional, hablamos principalmente de estas cuatro: pruebas unitarias, pruebas de integración, pruebas de aceptación y pruebas de sistema. En esta metodología, el proceso de desarrollo se dirige desde las especificaciones de los requisitos hacia el código, y luego del código hasta las pruebas, entendiéndose que el proceso de validación es un complemento en todo el proceso de desarrollo.

En las metodologías convencionales, las pruebas no se ejecutarán hasta que el código esté terminado, aunque puedan diseñarse antes de la codificación y las actividades se aplican en secuencia, lo cual significa que las pruebas de integración no se pueden llevar a cabo hasta no terminar todas las pruebas unitarias.

El enfoque de las pruebas en esta metodología se basa en la definición de los casos de prueba, los cuales dependen de las estrategias de prueba a aplicar y su representación mediante los lenguajes adecuados.

1.14.3 Metodologías ágiles.

Las pruebas son el centro de estas metodologías, por lo tanto el proceso de desarrollo es dirigido por ellas. Estas metodologías no manejan el desarrollo como un conjunto de fases en las que las pruebas es una de ellas, sino que plantean que las prácticas y el desarrollo deben estar integrados completamente. Las pruebas se escriben antes comenzar la codificación y solo se escribe el código necesario para superar las pruebas y guían el proceso de desarrollo

Las pruebas en las metodologías ágiles no se consideran como un conjunto de niveles que se deben alcanzar para la validación final del sistema que se está; desarrollando. Estas metodologías utilizan las pruebas unitarias y de aceptación como principales herramientas para dar soporte a los enfoques de pruebas. Las pruebas de integración y las pruebas de sistema están implícitas en el proceso de desarrollo, ya que cada nueva parte que se escribe del código se integra con el sistema completo, por lo tanto, no es necesario diseñar pruebas de integración ya que estas se encuentran incorporadas en las otras pruebas que se han desarrollado.

1.15 HERRAMIENTAS

La utilización de herramientas de automatización de pruebas de software, genera beneficios para mejorar el proceso de pruebas. La decisión del tipo de herramientas a utilizar, se debe tomar basada en el proyecto en particular.

1.15.1 Bugzilla.

Bugzilla es un sistema de seguimiento y notificación de errores, está escrito en Perl y fue desarrollado por el proyecto Mozilla. Puede ser aplicado a prácticamente cualquier desarrollo de software, funciona mediante un esquema de anotaciones y notificaciones automáticas.

El funcionamiento de Bugzilla es muy sencillo, ya que solo es necesario contar con un navegador Web y una cuenta de correo electrónico para recibir las notificaciones.

Características:

- Esquema de seguridad integrado
- Dependencias entre bugs y gráficos
- Capacidades avanzadas de informes
- Un back-end estable y robusto basado en bases de datos relacionales
- Altamente configurable
- Protocolo de resolución de fallos, fruto de un proceso de comprensión y pensamiento natural
- APIs de correo electrónico, XML, consola y HTTP
- Permite la integración con sistemas automatizados de gestión de configuración de software.

1.15.2 Rational ClearQuest.

Rational ClearQuest es un sistema flexible para el seguimiento de errores, independiente del equipo de desarrollo, la ubicación geográfica que tengan y la plataforma que utilicen. Con Rational ClearQuest se puede dar seguimiento a los defectos y solicitudes de mejoras, asignar actividades y evaluar el estado real a través de todo el ciclo de vida de los proyectos. Los cambios realizados se implementan una vez y se propagan automáticamente a las interfaces de los clientes.

Características

- Seguimiento basado en actividad de cambios y defectos
- Soporte robusto y flexible para flujos de trabajo, que incluye notificaciones por correo electrónico y opciones de envío
- Fácil personalización
- Soporte completo para consultas con generación de informes y gráficos
- Interfaz web para acceder fácilmente desde cualquier navegador web
- Integración transparente con Rational ClearCase.

1.15.3 DevTrack.

DevTrack es una solución comercial para el seguimiento de errores, es usada por equipos de desarrollo de software para el aseguramiento de la

calidad ya sea en grandes o pequeños proyectos, facilitando la colaboración efectiva entre los usuarios finales, los clientes, y los miembros del equipo de desarrollo. Provee automatización de flujos de trabajo y procesos, búsqueda avanzada y reportes.

Características

- Definir un número ilimitado de roles para garantizar que la información sea segura
- Los administradores y gerentes del proyecto pueden controlar el acceso de usuarios a la información
- Un portal web donde los clientes pueden enviar, rastrear y actualizar temas. También pueden crear y asignar subtemas.

1.15.4 Census Bug Tracking and Defect.

Census Bug Tracking and Defect es una herramienta para el seguimiento de errores basada en Web, que facilita la gestión de problemas relacionados con proyectos de software, como son errores de programación, defectos, peticiones de funcionalidades, tareas de desarrollo, llamadas de soporte.

Características

- Ofrece una tecnología de flujo de trabajo que se puede personalizar para adaptarse a las necesidades específicas del negocio.
- Notificaciones por correo electrónico que mantiene informados a los miembros del equipo de desarrollo, a los usuarios finales, a los clientes y a las demás personas que están involucradas en el desarrollo del proyecto.
- Mediante informes y estadísticas permite la evaluación del estado del proyecto, su progreso y el cumplimiento de los objetivos.
- Permite una fácil personalización de las características, de acuerdo a la terminología y los procesos de cada proyecto.

- Permite la coordinación y seguimiento de versiones directamente desde la herramienta de seguimiento de errores de programación.

1.15.5 Mantis.

Mantis es una aplicación OpenSource, desarrollada en mysql y php, de fácil instalación y configuración. Se pueden configurar diferentes estados para cada tarea y los perfiles que sean necesarios. Desde la herramienta se puede configurar el flujo de trabajo de acuerdo con los roles, es decir, que solo los probadores pueden abrir problemas, que solo los coordinadores pueden analizarlos y solo los programadores atenderlos. Mantis incluye filtros, un sistema de búsqueda, tiene soporte para varios idiomas y también informa por e-mail de la resolución de los errores de los que se ha informado.

Características

- Permisos de usuario por roles
- Interfaz gráfica sencilla
- Permite crear categorías para manejar los distintos tipos de actividades, en cada proyecto
- Anuncios globales.

1.15.6 Rational Performance Tester.

Rational Performance Tester es una herramienta para realizar pruebas de rendimiento, pruebas funcionales, pruebas de regresión, pruebas de interfaces y basadas en los datos, que permite verificar la presencia y la causa de los fallos que afectan al rendimiento de los sistemas.

Características

- Crea rápidamente pruebas sin usar código y sin conocimientos de programación
- Ofrece un completo editor de pruebas a modo de árbol que proporciona vistas de alto nivel y detalladas de las pruebas
- Automatiza la variación de datos de pruebas y permite la inserción de código Java a medida para la personalización flexible de las pruebas

- Automatiza la identificación y la gestión de respuestas de servidor dinámico
- Proporciona modelos y emulaciones flexibles de diversas poblaciones de usuarios
- Permite la ejecución de pruebas basadas en tecnología Windows, Linux y mainframe
- Genera informes en tiempo real para permitir el reconocimiento inmediato de problemas de rendimiento y presenta una vista de páginas web a modo de navegador de HTML
- Recopila e integra datos de recursos de servidores con datos de rendimiento de aplicaciones en tiempo real
- Disponible con interfaz de usuario basada en software Windows y Linux.

1.15.7 Jmeter.

Jmeter es una herramienta Java que permite realizar diferentes pruebas, principalmente pruebas de carga y rendimiento sobre aplicaciones web.

JMeter puede ser utilizado para probar: conexiones JDBC, FTP, LDAP, Web Service, JMS, conexiones HTTP, conexiones TCP genéricas.

Características

- Puede probar el rendimiento en diferentes tipos de servidores: de aplicaciones, SGBD, Web Services, Colas de Mensajería y otro tipo de sistemas mediante el uso de JAVA.
- Dispone de una interfaz visual amigable y su estructura modular permite la personalización de las numerosas herramientas de las que dispone.

1.15.8 Load Runner Product Family.

Load Runner Product Family es una herramienta desarrollada por Mercury Interactive, que le permite determinar la escalabilidad, comportamiento y rendimiento de aplicaciones emulando ciento y hasta miles de usuarios virtuales. Muestra el estado del sistema bajo carga por medio de monitores

en tiempo real, que obtienen datos desde los distintos componentes de la infraestructura, permitiendo trazar los resultados de la prueba.

Características

- Descubre cuellos de botella en infraestructuras complejas
- Asegura que los requerimientos de rendimiento sean cumplidos en la producción
- Soporta los entornos empresariales más difundidos como son: ERP/CRM, Web, J2EE, .NET, Wireless y Streaming Media
- Posee una integración completa con todos los productos de Mercury.

1.15.9 QaTraq Profesional test management.

QaTraq es una herramienta gratuita para la gestión de pruebas de software, diseñada para pequeñas y medianas empresas. Basado en LAMP (Linux, Apache, Mysql y PHP) por lo que es de fácil instalación en cualquier plataforma.

QaTraq proporciona una solida base para que los equipos de desarrollo puedan crear y actualizar scrips de pruebas, casos de pruebas y resultados de las pruebas, así como también pueden tener acceso a una variedad de informes.

Características

- Generación de informes gráficos
- Informes profesionales sobre aéreas claves: el seguimiento de scripts de prueba, el seguimiento de los resultados de pruebas, la identificación de cobertura de las pruebas y la determinación del progreso de prueba
- Los casos de prueba pueden ser importados desde archivos de Excel o XML
- Los casos de prueba y los resultados de las pruebas pueden ser exportados a archivos PDF, Excel o XML.

1.15.10 TestDirector.

TestDirector es una solución basada en Web que cubre todos los aspectos de la gestión de calidad del ciclo de vida del software: administración de requerimientos, planificación de las pruebas, ejecución de pruebas,

seguimiento de defectos, gráficos y reportes. Facilita la comunicación entre los diferentes equipos de trabajo involucrados en el desarrollo del proceso de calidad, permitiéndoles el acceso a la información desde cualquier lugar geográfico.

Características

- El administrador de requerimientos relaciona los casos de prueba a los requerimientos funcionales, asegurando trazabilidad a través del proceso de pruebas.
- De acuerdo con los requerimientos, se construye el plan de pruebas y se diseñan las pruebas, manteniendo la información en un repositorio, para que se puedan acceder a planes de pruebas en futuros desarrollos fácilmente.
- El administrador de ejecuciones "TestLab" permite ejecuciones espontáneas o programadas. los grupos de pruebas pueden emular en forma real los procesos de negocio de la vida real.
- El "Defect Manager" abarca el ciclo completo de los defectos, desde la detección del problema hasta la solución y la verificación. Asiste a la gerencia en la toma de decisiones para las aplicaciones. posee un sistema de alertas y flujo de trabajo flexible y la posibilidad de identificar en la base de datos defectos similares, minimizando la probabilidad de tener defectos duplicados.
- El modulo de "Analysis" genera gráficos y reportes con información actualizada en tiempo real, asistiendo a los usuarios en el análisis de datos y en la toma de decisiones.

1.15.11 TestGen4J.

TestGen4J es una herramienta libre para la generación automática de casos pruebas de unidad. Utiliza reglas, escritas en un formato de archivo XML configurable, que define las condiciones límite para los tipos de datos. El código de prueba está separado de los datos de prueba con la ayuda de JTestCase. El objetivo principal de esta herramienta es ejercitar la prueba del

valor límite de los argumentos pasados al método, utilizando un motor de reglas que el usuario configura desde un archivo XML.

El código de prueba es generado de forma jerárquica. Un conjunto de pruebas principal genera conjuntos de pruebas que invoca el de las clases individuales. El conjunto de pruebas individuales de clase está formado por la colección de métodos de ensayo de esa clase.

1.15.12 .Test.

.Test es una herramienta desarrollada por Parasoft para la automatización de pruebas de unidad y conformidad de código para entornos Microsoft .Net. Con esta herramienta los desarrolladores no necesitan realizar casos de pruebas.

Características

- Pruebas unitarias o pruebas a nivel en la aplicación
- Análisis de los estándares del código empleando conjuntos configurables de reglas
- Creación gráfica o automatizada de reglas definidas por el usuario, específicas para una aplicación, equipo o estándar de organización
- Conjunto de reglas avanzadas de seguridad diseñado para identificar vulnerabilidades comunes y críticas de la seguridad de la aplicación .NET
- Generación automática de casos de pruebas extensibles de NUnit
- Soporte y cobertura para casos de pruebas NUnit
- Modos de pruebas preconfiguradas y personalizadas
- Cobertura detallada de análisis del código
- Desarrollo y gestión de informes HTML configurables
- Compartir en todo el equipo las configuraciones de las pruebas y ficheros.
- IGU y modo de línea de comando
- Integración con Microsoft Visual Studio .NET.

1.15.13 SoapUI.

SoapUI es una herramienta grafica de software libre basada en Java, que permite realizar pruebas de servicio Web.

SoapUI permite hacer pruebas a servicios web de forma fácil, ver los resultados. Además, permite facilitar el uso de herramientas comunes para la generación de clientes, como Axis.

Trabajando con servicio web, y sin interface gráfica en la aplicación, esta herramienta nos permite automatizar fácilmente las pruebas funcionales y así asegurar la calidad en los proyectos.

Las pruebas funcionales de los servicios Web pueden usarse para realizar:

- Pruebas unitarias: para validar que cada operación de los servicios funciona como se definió
- Prueba de aceptación: para validar que el servicio retorna resultados aceptables según los requerimientos
- Pruebas de proceso: para validar que una sucesión de invocaciones del servicio cumple con el proceso de negocio definido
- Pruebas de manejo de datos: para validar el comportamiento con las entradas de datos externos al sistema
- Pruebas de regresión: para validar el comportamiento luego de realizar cambios.

1.15.14 Rational Robot.

Rational es una herramienta para la automatización de pruebas funcionales, de regresión y de configuración de aplicaciones e-commerce, cliente/servidor y ERP. Proporciona pruebas a objetos como menús, listas, pruebas especializados para objetos específicos del ambiente de desarrollo. Da soporte a múltiples tecnologías de interfaz de usuario para diversos entornos. Incluye un componente de gestión de pruebas y se integra con las herramientas de Rational Team Unifying Platform para realizar el seguimiento de defectos, gestión de cambios y rastreo de requerimientos.

Características

- Simplifica la configuración de la pruebas de software: puede usarse para distribuir las pruebas funcionales entre varias máquinas, cada una de ellas configurada en diferente forma. Se pueden correr simultáneamente las mismas pruebas funcionales, reduciendo el

tiempo necesario para identificar los problemas con configuraciones específicas.

- Prueba muchos tipos de aplicaciones: realiza pruebas de configuración, regresión y funcionales, soporta una amplia gama de ambientes y lenguajes.
- Asegura la Profundidad de las pruebas: realiza pruebas más allá de la interfaz de usuario de la aplicación.
- Ayuda a analizar rápidamente los problemas: registra automáticamente los resultados de la prueba en el Rational Repository.
- Facilita la transición de pruebas manuales a pruebas automatizadas.

RESUMEN EJECUTIVO

2.1 CONCEPTO DEL NEGOCIO

QualityTest ofrece el servicio de acompañamiento en el desarrollo de una aplicación, evaluándola desde un punto de vista crítico por medio del proceso de pruebas, principalmente a pequeñas y medianas empresas desarrolladoras de software de la ciudad de Pereira, utilizando nuestro conocimiento y algunas herramientas, para ayudar a reducir el costo del desarrollo, ya que se previenen o detectan errores y defectos en etapas tempranas haciendo una corrección oportuna.

2.2 POTENCIAL DEL MERCADO EN CIFRAS

La estimación del mercado potencial para el servicio que ofrece QualityTest S.A.S es principalmente para las 20 empresas desarrolladoras de software ubicadas en la ciudad de Pereira.

2.3 VENTAJAS COMPETITIVAS Y PROPUESTA DE VALOR

- La principal ventaja que tiene QualityTest S.A.S es su ubicación geográfica, ya que la región no cuenta con alguna empresa que preste el servicio de pruebas, permitiéndoles esto prestar un servicio más rápido. La ubicación les permite también estar cerca de parques tecnológicos de software los cuales funcionan como clústers y donde se encuentra gran cantidad de empresas desarrolladoras.
- La región cuenta con universidades con carrera de Ingeniería de Sistemas, las cuales da personal calificado, apto para laborar en QualityTest S.A.S y así, asegurar la prestación de un servicio de muy buena calidad.
- Presencia de medianas y grandes empresas en la ciudad de Pereira, las cuales cuentan su propio departamento de sistemas.

2.4 RESUMEN DE LAS INVERSIONES REQUERIDAS

Tabla 1. Costo inversión inicial

| Inversión inicial | Valor |
|------------------------------|----------------------|
| Herramientas técnicas | 7.410.000 |
| Herramientas administrativas | 4.080.000 |
| Costos legales | 179.600 |
| Total | \$ 11.669.600 |

Cada uno de los socios contribuirá con 6 millones de pesos, para un total de 12 millones de pesos en aportes, los cuales se utilizarán para pagar los costos iniciales.

2.5 PROYECCIONES DE VENTAS Y RENTABILIDAD

Tabla 2. Proyección de ventas

| Servicio | Año 1 | Año 2 | Año 3 | Año 4 | Año 5 |
|---------------------------|---------------------|---------------------|----------------------|----------------------|----------------------|
| Pruebas de requerimientos | 4.500.000 | 4.725.000 | 4.961.250 | 8.682.188 | 9.116.297 |
| Pruebas de diseño | 4.500.000 | 4.725.000 | 4.961.250 | 8.682.188 | 9.116.297 |
| Pruebas de integración | 12.000.000 | 12.600.000 | 22.050.000 | 23.152.500 | 29.172.150 |
| Pruebas funcionales | 16.000.000 | 16.800.000 | 22.050.000 | 27.783.000 | 29.172.150 |
| Pruebas de aceptación | 9.000.000 | 9.450.000 | 13.230.000 | 17.364.375 | 18.232.594 |
| Pruebas de seguridad | 9.000.000 | 9.450.000 | 9.922.500 | 17.364.375 | 18.232.594 |
| Pruebas de stress | 9.000.000 | 9.450.000 | 9.922.500 | 17.364.375 | 18.232.594 |
| Pruebas de rendimiento | 9.000.000 | 12.600.000 | 13.230.000 | 20.837.250 | 21.879.113 |
| Total | \$73.000.000 | \$79.800.000 | \$100.327.500 | \$141.230.250 | \$153.153.788 |

2.6 CONCLUSIONES FINANCIERAS Y EVALUACIÓN DE VIABILIDAD

- Al evaluar la inversión inicial, los costos y la obligaciones financieras de la empresa QualityTest S.A.S, se obtuvo Tasa de Retorno Interna

(TIR) de 10% y un Valor Presente Neto (VPN) de \$6.469.028, datos que demuestran la viabilidad del proyecto.

- Basados en el porcentaje de rendimiento del Depósito a Término Fijo (DTF) de 3.91% y el porcentaje de la TIR obtenido, resulta más rentable invertir en la empresa QualityTest S.A.S.

MODULO MERCADO

3.1 ANÁLISIS DEL SECTOR

3.1.1 Industria de software.

El inicio de esta industria fue a mediados de la década del 50, cuando las empresas entregaban el software como un producto asociado al hardware; a partir de los años 60 comienza a verse el software como un producto independiente, pero solo hasta finales de los 70 y comienzos de los 80 se crean empresas dedicadas a la fabricación de software².

Entre 1990 y 2002 se reportaron ventas superiores a 397 miles de millones de dólares en la industria de software, de los cuales el 51% provienen de firmas pioneras en Estados Unidos, Alemania y Japón y el 49% fue originado en países de economía emergentes los cuales son India, Irlanda, Israel, China y Brasil³

El movimiento de esta industria tiene impacto en todos los sectores de la economía y el desarrollo de un país y a pesar de los problemas de calidad, confiabilidad y cumplimiento de tiempos que todavía se presentan en el desarrollo de software, ha mantenido una importante participación en el crecimiento de la economía de los países. Esto ha llevado a crear nuevos modelos e implementar nuevas técnicas y herramientas de programación para la producción de software.

3.1.2 Cadena de valor de la industria del software.

La cadena de valor es un componente del proceso de negocio de cualquier organización, en especial de las pertenecientes a la industria del software.

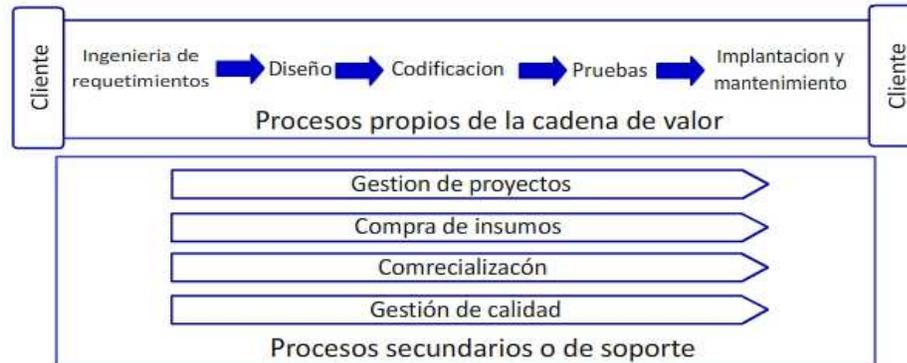
En la industria del software, se tiene una cadena de valor que recibe como entrada los requerimientos del cliente y obtiene como salida el producto o servicio de software solicitado. También es importante tener en cuenta los

² HOCH D, Roeding C, Purkert G y Lindner S. Secrets of Software Success. Managements Insights from 100 software firms around the world, Harvard Business School, Boston 1999

³ ARORA, A. and Gambardella, A. The Rise and Growth of the Software Industry in Brazil, China, India, Ireland and Israel. Oxford University Press 2005

procesos secundarios o de apoyo, los cuales no hacen parte de la cadena de valor, como lo son: la gestión de proyectos, la compra de insumos, la comercialización de productos y la gestión de calidad.

Ilustración 7. Mapa procesos



Fuente: Intersoftware.

En cada uno de los procesos de la cadena de valor encontramos otras actividades:

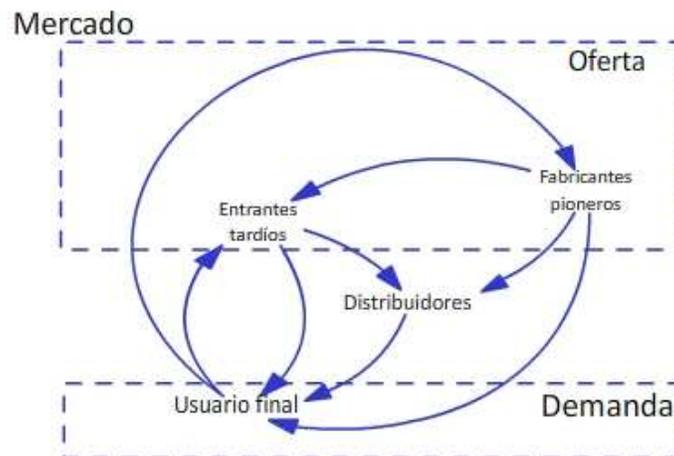
- Ingeniería de requerimientos: determinación, análisis, especificación y validación de requerimientos
- Diseño: construcción de arquitectura básica, construcción de diagramas de diseño (actividades, interfaces, clases, entre otros), definición de estructuras de datos y elaboración de algoritmos
- Codificación: generación de código, selección de componentes reusables, documentación
- Pruebas: pruebas de unidad, pruebas de integración, pruebas de sistema, pruebas de aceptación, pruebas de implantación, documentación
- Implantación y mantenimiento: instalación, soporte, capacitación, análisis de impacto, reingeniería.

Los procesos de soporte, pueden marcar ventajas competitivas y hacer que el cliente se incline más hacia la compra de productos ofrecidos por una u otra empresa.

3.1.3 Actores que participan en la industria del software.

A continuación se presenta una descripción del mercado de software y de los actores que participan en él y la relación entre la oferta (fabricantes pioneros y entrantes tardíos) y la demanda (usuario final).

Ilustración 8. Relación oferta y demanda



Fuente: Intersoftware.

Los fabricantes pioneros son aquellas primeras empresas que iniciaron en el mercado del software y se dedican a la fabricación de software estandarizado y/o servicios de software de forma masiva. Estas empresas comercializan el producto directamente con los grandes clientes o usan canales de distribución que llevan el producto hasta el usuario final. El usuario final no puede hacer modificaciones sobre el producto ya que son estandarizados, pero se admite hacer a los fabricantes pioneros sugerencias o mejoras deseables para las nuevas versiones de los productos.

Por otro lado los entrantes tardíos, son emprendedores con conocimiento y destrezas obtenidas por experiencias anteriores, los cuales no cuentan con la capacidad de competir con productos masivos estandarizados, pero su principal mercado son aquellos clientes que tienen necesidades particulares y específicas de software. Ellos comercializan su producto directamente con el usuario final, recibiendo también sugerencias acerca de las mejoras para los nuevos productos y cuando van adquiriendo posicionamiento en el mercado, se ven en la necesidad de usar canales de distribución para comercializar sus productos.

3.1.4 Industria del software en el mundo.

La industria del software en el mundo ha tenido desde sus inicios una gran importancia para el desarrollo de las naciones que han apostado a este sector de la economía, permitiendo que las empresas accedan a los recursos que necesitan, para ser cada vez más competitivas y ofrecer servicios y productos de mejor calidad.

La industria del software es parte fundamental en la economía actual del mundo y crea grandes oportunidades de trabajo. Su desarrollo aumenta cada día, para mejorar actividades y facilitar el uso de artefactos que hacen parte de la vida moderna, además de su incorporación a todos los sectores de la economía y de la sociedad.

Países como India, Irlanda, Corea y Chile, han desarrollado políticas que permiten el fortalecimiento y el crecimiento del sector del software, promoviendo la formación de profesionales capacitados para la construcción de software, estableciendo beneficios tributarios, desarrollando estrategias de comercio internacional y fortaleciendo la infraestructura de transmisión de datos para las zonas de producción de software y de asistencia remota. Israel se concentró en mejorar la calidad del software ofrecido comercialmente y en obtener certificaciones de mayor reconocimiento a nivel internacional, para conquistar segmentos especializados del mercado.

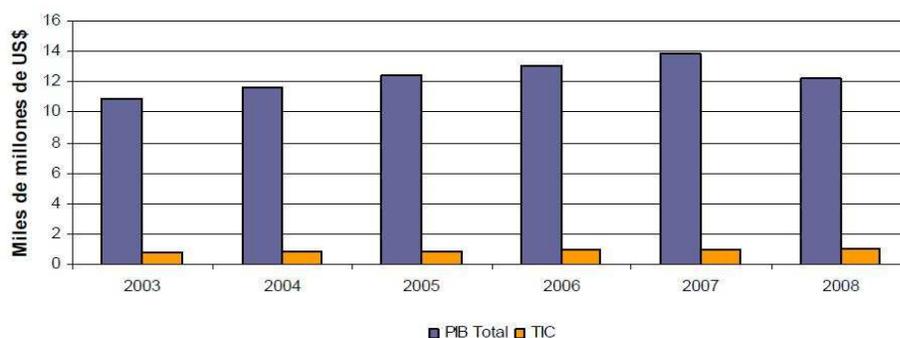
3.1.4.1 Estados Unidos.

Estados Unidos ha sido uno de los países más importantes a nivel de software, por ser uno de los mayores productores y consumidores. El sector del software en Estados Unidos ha sido uno de los de mayor crecimiento en la historia reciente de este país, por sus ventas millonarias en productos y servicios y por el crecimiento de las empresas pertenecientes a este sector, encontrando las más grandes a nivel mundial.

La industria de las TIC aporta cada vez más al PIB de los Estados Unidos de América, alcanzando un 7.22%, y demostrando una dinámica creciente y cuya tendencia persistirá en los próximos años⁴.

Ilustración 9. Participación de las Tic en el PIB de Estados Unidos

⁴ DURAN, Martin y SAAVEDRA, Isis A. Informe mensual junio de 2009. CEPREDE, Madrid 2009.

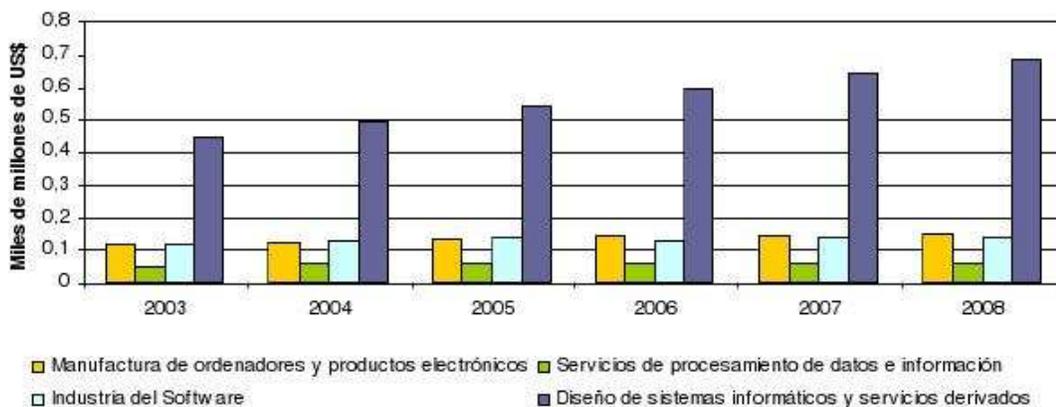


Fuente: CEPREDE.

El crecimiento de la industria del software en Estados Unidos se ha dado principalmente por factores como: movimientos del tipo de cambio generan un incremento en la competencia, permitiendo a las empresas extranjeras ingresar al mercado, la facilidad de adquirir computadores personales influyen en la compra y actualización del software, la protección de la propiedad intelectual, por ser un mercado muy competitivo y consciente de los precios no existen barreras en la información.

El mayor peso se encuentra en la categoría de diseño de sistemas informáticos y servicios derivados con una tendencia creciente, seguido por manufactura de computadores y equipos electrónicos, así como la industria del software con una tendencia permanente.

Ilustración 10. Industria de las TIC en el PIB de Estados Unidos



Fuente: CEPREDE.

3.1.4.2 India.

La industria de software brinda la posibilidad a los países menos industrializados a insertarse en mercados internacionales, como es el caso de India, que paso de ser un país que solo ofrecía trabajo de programación a bajo costo a ser hoy en día uno de los países que más exportan servicios y productos de software. Bangalore, la tercera ciudad con más habitantes en la India, es reconocida a nivel internacional como una de las ciudades símbolo en el desarrollo de la industria del software.

Según Nasscom⁵, los ingresos generados por esta industria se estiman en US\$ 52 mil millones para el 2008, de los cuales el 77.5% constituyen exportaciones hacia EE.UU. y el Reino Unido (61.4% y 17.8%, respectivamente). Dicha industria, que incluye el desarrollo de software, outsourcing y servicios, ha crecido un 33% anual en los últimos cinco años y constituye alrededor del 6% del PIB indio.

Los factores fundamentales para que la industria de software en India haya tenido el crecimiento y fortalecimiento que hasta la actualidad ha tenido, son: el dominio del idioma ingles que le permite mayor facilidad para competir en el mercado estadounidense y británico, la inversión privada al desarrollo de centros de investigación y tecnología, el apoyo del gobierno, al adoptar medidas que favorezcan el crecimiento de la industria, como la fundación del Instituto Nacional de la Tecnología del Software, que sirvió para elaboración e implementación de políticas gubernamentales en el campo de la tecnología, la creación de subsidios, creación de programas de crédito y de financiamiento, protección de los derechos de propiedad intelectual, reducción de impuestos y eliminación de barreras de entrada y salida, entre otras.

Otro aspecto fundamental para la participación de India en el mercado internacional del software, es que las empresas en este país se han esforzado en obtener certificaciones CMMI nivel cinco y contar con este tipo de certificaciones es importante para competir en el mercado mundial.

⁵ NASSCOM: Asociación nacional de las compañías del software y de servicios. India <http://www.nasscom.in/>

3.1.4.3 Irlanda.

Irlanda estableció como principal estrategia para el desarrollo de la industria de software la creación de condiciones favorables para la instalación de empresas internacionales. Su ubicación estratégica y su condición de ser miembro de la Unión Europea le permiten el acceso a un mercado de 400 millones de personas, también su excelente infraestructura de transporte, y sus favorables condiciones de inversión son características fundamentales que han permitido el establecimiento de empresas extranjeras de software y el desarrollo de proyectos nacionales.

Irlanda logro consolidar su economía y hacerla una de las de mayor crecimiento en los últimos años, con tasas promedio de crecimiento del 8,66% entre 1994 y 2001⁶, alcanzando altos niveles de desarrollo humano e ingresos per cápita. Empresas multinacionales han realizado uniones con empresas locales, debido a los bajos costos de operación e impuestos y los incentivos que se ofrecen para inversionistas extranjeros.

La industria del software en Irlanda se ve beneficiada gracias a las facilidades con que cuentan las empresas extranjeras que quieren invertir, además de contar con uno de los mejores servicios de telecomunicaciones de Europa, con una red de conectividad con el mundo ilimitada.

La educación ha sido un elemento importante; Irlanda tiene alrededor de 30 universidades y centros de capacitación, lo que le permite contar con servicios profesionales de primer nivel, creando políticas educativas, con el fin de mejorar la competitividad y la interacción con las nuevas tecnologías. Además cuenta con cerca de dos mil graduados por año en una población de tres millones de habitantes.

3.1.4.4 Israel.

Israel en las últimas décadas ha logrado hacer un cambio importante en cuanto a la diversificación de la economía, incluyendo además de sus actividades económicas principales (cultivos de cítricos y algodón), actividades relacionadas con los sectores de alta tecnología, obteniendo un repunte en las exportaciones de software, con tres mil 500 millones de

⁶ http://www.prochile.cl/documentos/pdf/irlanda_software_2003.pdf

dólares al año y en la industria de las telecomunicaciones digitales con 10 mil 900 millones de dólares al año.

Las políticas creadas por el gobierno israelí (incentivos fiscales, préstamos bancarios para las nuevas empresas de informática) buscan favorecer la consolidación de la industria del software y el sector tecnológico. Israel también desarrolló centros de investigación e incubadoras de negocios para apoyar las industrias tecnológicas.

Grandes empresas a nivel mundial como Microsoft, IBM, SAP, HP, Intel, entre otras, han creado centros de investigación y desarrollo en ciudades como Haifa y Tel Aviv. Israel tiene una gran concentración de empresas informáticas, ocupando el segundo lugar, solo superado por el Valle del Silicio en Estados Unidos.

India, Irlanda e Israel, a pesar de tener características tan diferentes en su población, su idiosincrasia, cultura y religión, han aprovechado la participación en una nueva economía basada en el conocimiento. Los gobiernos de estas naciones han dado un gran apoyo a la educación tecnológica, al impulso de las pequeñas, medianas y grandes empresas del sector informático mediante financiamiento y capacitación. Un aspecto fundamental para estas tres naciones ha sido su alto nivel de especialización en el sector tecnológico y su dominio del idioma inglés.

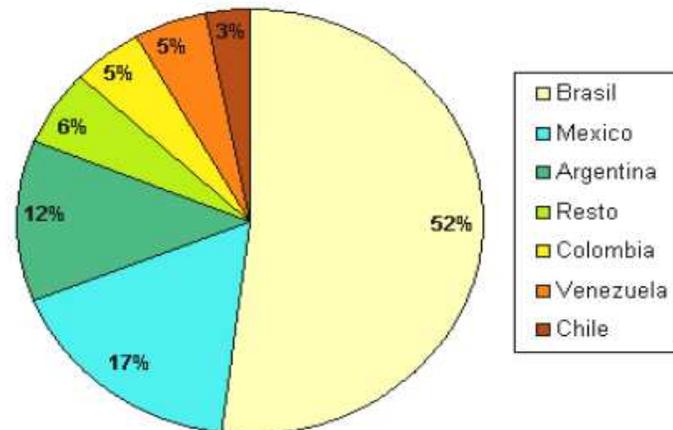
3.1.5 Industria del software en Latinoamérica.

En los últimos años, el software tomó un protagonismo primario en los países de Latinoamérica, donde la tecnología de información y el software son tomados como sectores estratégicos ya que tiene un gran aporte en el crecimiento y dinamismo económico, como también por su elevado consumo de recursos humanos calificados.

En América Latina, la industria del software cuenta con asociaciones de cooperación, las cuales tiene como objetivo mejorar los mercados y las cadenas de distribución, propiciar políticas, ayudar a sus asociados a mejorar sus capacidades competitivas y busca el desarrollo de programas conjuntos, beneficiándose mutuamente. La Federación de Asociaciones de Latinoamérica, es un ejemplo de estas asociaciones.

El mercado del software en Latinoamérica tiene una participación del 1,8% en relación al mercado mundial⁷. Brasil, México y Argentina se destacan como los países en Latinoamérica con más actividades en la producción y comercialización de software y servicios informáticos. A continuación se muestra como esa distribuida la participación de la industria del software en Latinoamérica.

Ilustración 11. Mercado latinoamericano



Fuente: WITSA.

Debido a su gran aporte para el crecimiento y dinamismo económico, los países de Latinoamérica han tomado como actividad principal el sector del software, ya que si analizamos el sector económico de cada país, encontramos que los productos de software y servicios relacionados, generan una gran demanda y oferta, por consiguiente se ve la necesidad de estimular este sector por medio de programas de financiamiento los cuales han implementado algunos países de Latinoamérica para favorecer el desarrollo de software.

Países como Brasil, México, Argentina, Uruguay y Costa Rica, han comenzado a evolucionar en los últimos años en la producción de software, algunos en el nivel interno del país como lo son Brasil, Argentina y México y otros en el nivel externo como lo son Uruguay y Costa Rica

⁷ La industria del software y los servicios informáticos en Argentina.pdf. Página 17

3.1.5.1 Brasil.

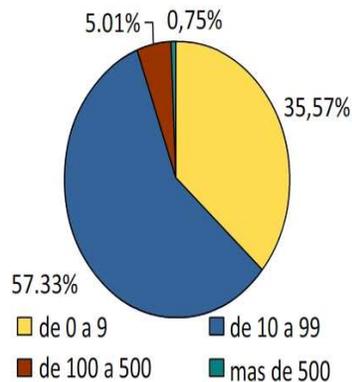
El mercado de software en Brasil ocupa el 12° lugar en el mercado mundial. En el 2005 fueron facturados 7.410 millones de dólares americanos (1.2 % del PIB brasileño), los cuales corresponde al comercio del software (2.720 millones de dólares) y a servicios relacionados con software (4.690 millones de dólares)⁸.

Diversas empresas que desarrollan y producen software en este país, trabajan para el mercado nacional, esto se debe a varios factores: Brasil ocupa el 5to lugar a nivel mundial con respecto a su número de habitantes (191.480.630), por lo tanto no existe la posibilidad de que el mercado nacional se sature. Tampoco existe la necesidad económica de salir al exterior para aumentar sus ventas, por lo tanto se dedican al desarrollo de software aplicativo para cubrir las necesidades locales.

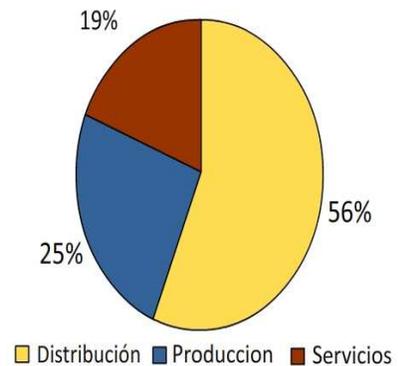
El sector de software en este país está compuesto por más de 7.000 empresas, de las cuales la mayoría son pequeñas empresas de 10 a 99 empleados y microempresas de menos de 10 empleados. El 81% de estas empresas se dedican al desarrollo y distribución del software y el 19% restante trabaja en el área de servicios.

Ilustración 12. Perfil de las empresas brasileñas de software

Número de empleados en las empresas de desarrollo y producción



Área de actuación de las empresas



Fuente: ABES.

⁸ ABES: Asociación Brasileña de Empresas de Software

Las empresas brasileñas productoras de software están ubicadas donde el sector está más desarrollado. En la región del Sudeste están localizadas el 64% de las empresas brasileñas, más específicamente en Sao Paulo, donde se encuentra el 41% de estas empresas.

En lo que a empresas extranjeras se refiere, las empresas estadounidenses dominan el sector. Microsoft, IBM, HP y Oracle, son las empresas que más facturan en el mercado brasileño.

3.1.5.2 Argentina.

Argentina, se consolida dentro de América Latina como uno de los grandes en la industria del software. Según un informe realizado por la World Information Technology & Services Alliance (WITSA), Argentina se ubica entre los países con las tasas más altas de crecimiento en el sector del software.

Después de la crisis del 2002, la industria del software en Argentina muestra un fuerte crecimiento; según datos de la Cámara de Empresas de Software y Servicios Informáticos (CESSI), en el 2007 registro ventas de 5.800 millones de dólares, mostrando un buen desempeño en 2006 y 2007, con crecimientos reales del 11.5% y 10.6% respectivamente (estimado anual).

Un gran número de empresas pertenecientes al sector del software, son de capital nacional y emplean menos de 50 personas, pero el fuerte de la facturación, proviene de las pocas empresas que poseen más de 50 empleados y de las firmas extranjeras. Del total de las empresas, el 54% son pequeñas, 41% medianas y el 5% grandes⁹.

Las exportaciones en este sector crecen rápidamente, destinándose fundamentalmente a Ibero América y en poca cantidad a Estados Unidos.

3.1.5.3 México.

El mercado mexicano, posee un fuerte crecimiento, su provisión interna está desarrollada medianamente y la competencia externa se encuentra liderada

⁹CESSI: Cámara de Empresas de Software y Servicios Informáticos

por Estados Unidos. La demanda es muy fuerte en los sectores privados de mayor volumen y crecimiento y también desde el estado, que se encuentra incorporando tecnología en todos los ámbitos del gobierno. En el 2007, el mercado de software en México se evaluó en 4.130 millones de dólares, creciendo un 15% con respecto a 2006¹⁰.

El sector de software en México, está integrado por 250 empresas aproximadamente, donde el 30% son microempresas que cuentan con menos de 15 empleados, el 64% Pymes y solo el 5% son grandes empresas que cuentan con más de 250 empleados¹¹.

México, actualmente trabaja con el Programa para el Desarrollo de la Industria del Software (PROSOFT), el cual abarca: exportaciones, inversiones, capital humano, marco legal, financiamiento, incubadoras, mercado interno, compras de gobierno, calidad y agrupamientos empresariales. Este programa tiene como meta para el 2013, exportar 5.000 millones de dólares, ser el líder a nivel de Latinoamérica de soporte y desarrollo de servicios basados en tecnologías de información.

3.1.6 Industria del Software en Colombia.

La industria del software en Colombia ha tenido un crecimiento relativamente rápido y ha llegado a ser una parte importante para el crecimiento económico del país y a estar al nivel de las tendencias de la economía mundial.

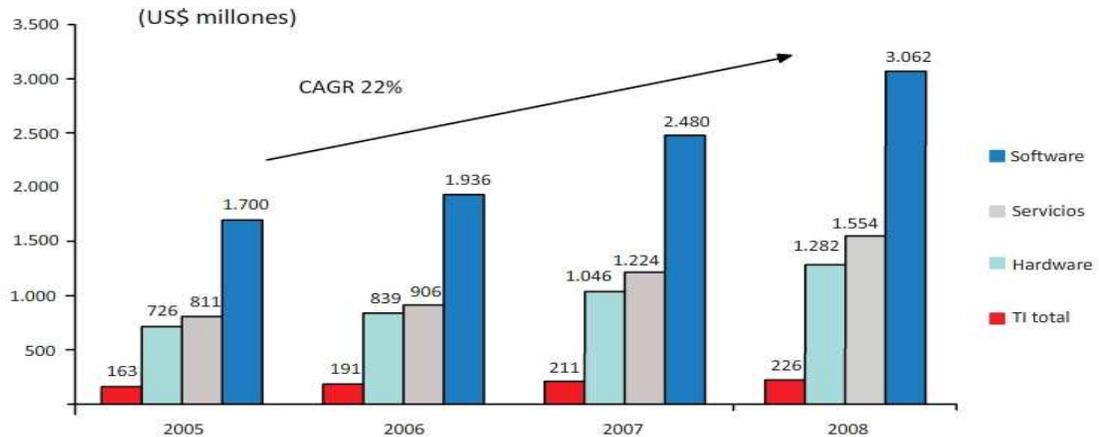
La producción de software en Colombia ha surgido y crecido con relativa rapidez, y las regiones lo consideran como un sector promisorio que promovería su desarrollo. Las bondades que significa el desarrollo de la industria del software para el aparato productivo nacional están relacionadas con las tendencias de la economía mundial. Por una parte, la utilización de software en procesos productivos permite su modernización y conlleva a la producción de bienes con mayor valor agregado. Por otra, el software es una industria “basada en el conocimiento” y por lo tanto, propicia el desarrollo de habilidades laborales sofisticadas y la innovación tecnológica, generando empleos bien remunerados. Así mismo, es una industria que no contamina y que no requiere de grandes inversiones de capital para comenzar.

¹⁰ ProMéxico: Oportunidades en el exterior. Página 5

¹¹ Asociación Mexicana de la Industria de Tecnología de Información, 2001

Colombia es el segundo sector en tecnología de la información de América Latina¹², debido al comportamiento de este sector que cuenta con más de 4000 empresas (850 dedicadas al desarrollo de software), con más 32000 trabajadores, y generan US\$ 270 millones al año. Adicionalmente, el país es el tercer mercado en crecimiento después de Brasil y México.

Ilustración 13. Evolución del mercado TI en Colombia



Fuente: IDC.

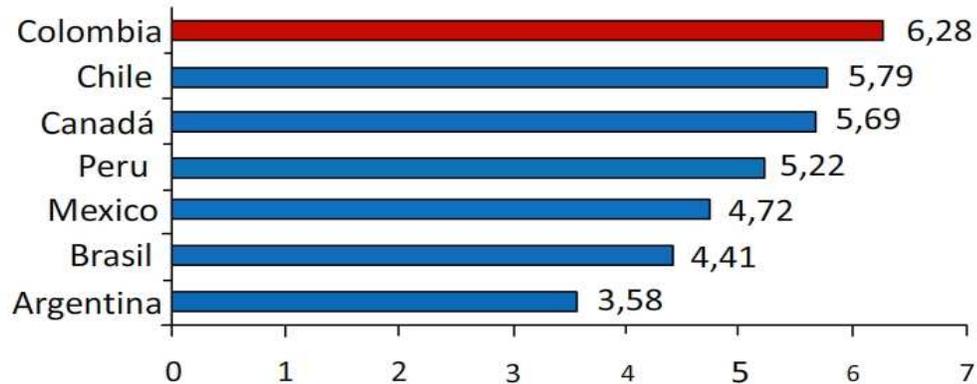
Ventaja del sector de software colombiano.

Factores como el de recursos humanos hace que el mercado del software en Colombia sea competitivo en relación a países de América Latina que ya tienen un reconocimiento a nivel internacional.

Colombia posee el índice más alto en disponibilidad de mano de obra calificada.

¹² Business Software Alliance, 2005

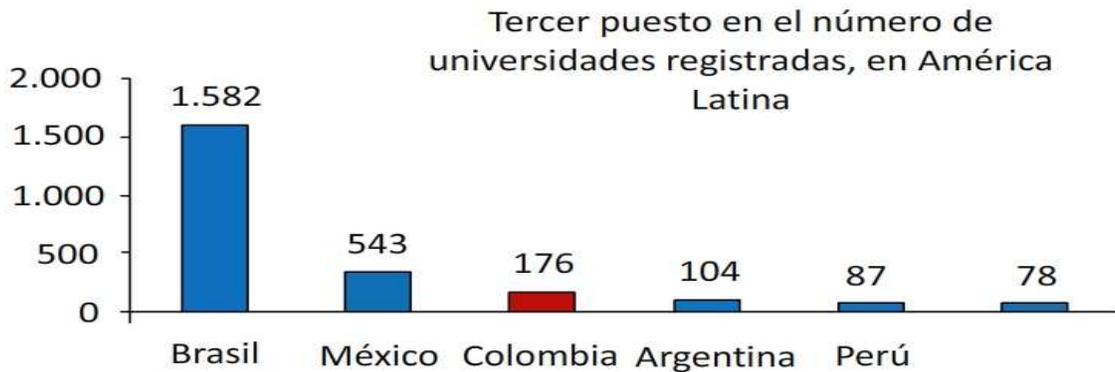
Ilustración 14. Disponibilidad mano de obra calificada en Colombia



Fuente: the world competitiveness yearbook 2008.

Colombia ocupa el tercer lugar en cuanto al número de Universidades registradas.

Ilustración 15. Universidades registradas en Colombia



Fuente: CSIC

El número de graduados universitarios tanto en carreras administrativas, como en ingenierías de información, es alto y ocupa el tercer lugar en América Latina.

Ilustración 16. Número graduados universitarios América Latina

Tercer puesto en el número de graduados universitarios en los principales países de América latina 2007

| | # de Graduados Carreras Administrativas | # de Graduados Ingeniería de TI |
|-----------|---|---------------------------------|
| Brasil | 51.181 | 10.766 |
| México | 61.813 | 13.005 |
| Colombia | 21.340 | 6.824 |
| Argentina | 18.640 | 4.047 |
| Chile | 8.353 | 9.205 |
| Uruguay | 1.031 | 235 |

Fuente: Proexport.

En términos generales, Colombia ha logrado consolidar ventajas competitivas frente al mercado Latinoamericano en varios aspectos:

- La mitad de las empresas que hacen presencia en el país hacen trabajos para desarrolladores extranjeros con altos estándares de calidad.
- Recurso humano altamente capacitado para el desarrollo de productos innovadores.
- Cuenta con tres importantes clúster desarrolladores de productos de Software: ParqueSoft, InterSoft y SinerTic.

El desarrollo de la industria de software en Colombia, se ha fortalecido por medio de los parques tecnológicos, ya que es el único país en Latinoamérica que cuenta con una red de clústeres tecnológicos especializados en el desarrollo de software, en los cuales se llevan a cabo proyectos de tecnología muy innovadores. Las compañías que trabajan para los clúster se dedican exclusivamente al desarrollo de software para diferentes sectores económicos.

Hoy en día, existen tecnológicos en diferentes departamentos del país e incubadoras de empresas que apoyan el desarrollo de la industria; sin embargo los parques tecnológicos de Cali y Popayán han tenido un mayor desarrollo respecto a los del resto del país y buenos resultados en la incubación de empresas de software y en asegurar su éxito en el mercado.

Colombia cuenta con tres importantes clústers de productos de software:

ParqueSoft: es un clúster de emprendimiento de ciencia y tecnología informática, que nace en 1999 en la ciudad de Cali, con el objetivo de encubar empresas de software creadas por jóvenes emprendedores. Parquesoft en la actualidad está integrado por más de 14 sedes en el territorio nacional, que dan vida a más de 400 emprendimientos alrededor de la tecnología.

InterSoft: es un empresa colombiana del sector informático conformado por profesionales de altamente especializados en diversas plataformas, con sede principal en la ciudad de Medellín y con una sucursal en la ciudad de Bogotá. Su objetivo es brindar servicios informáticos de consultoría, desarrollo, mantenimiento de software y outsourcing informático.

SinerTic: es un modelo empresarial resultado de la asociación de empresas colombianas del sector de la industria de las TICs que nace como un proyecto de integración y cooperación empresarial, el cual es soportado sobre la plataforma de productos y servicios especializados que ofrecen cada una de las empresas asociadas.

Colombia ha creado normas que benefician la exportación.

- Tratamiento preferencial para exportación de servicios: permite que los servicios intermedios de la producción se presten a sociedades de comercialización internacional sean exentos del impuesto sobre las ventas.
- Se encuentran exentos del IVA la venta de servicios que sean prestados en el país, y que se utilicen exclusivamente en el exterior por empresas o personas sin negocios o actividades en Colombia.

- La producción de software está exenta del pago del impuesto sobre la renta, para aplicar a este beneficio es necesario que Colciencias certifique que se cuenta con alto contenido de investigación científica y tecnológica. (del 1 de enero de 2003 hasta el 31 de diciembre de 2012.)
- Los desarrollos de nuevos software elaborados en Colombia, con alto grado de investigación científica y tecnológica están exentos del impuesto a la renta por diez años a partir del 2003.

3.2 ANÁLISIS DEL MERCADO

3.2.1 Mercado Objetivo.

El servicio que presta la empresa QualityTest S.A.S es llevar a cabo el proceso de pruebas en las empresas desarrolladoras de software por medio de un acompañamiento en las etapas del ciclo de vida del desarrollo para así evitar el incumplimiento de los requisitos del cliente, definiendo un proceso uniforme y estándar.

Los servicios ofrecidos están orientados a empresas de la ciudad de Pereira dedicadas al desarrollo de software, las cuales requieren mejorar la calidad de sus productos de software, aplicando una metodología para la ejecución de pruebas, y así lograr un nivel competitivo en la industria del software, generando productos de excelente calidad basados en la disminución de fallas y/o errores a nivel funcional, adquiriendo un posicionamiento a nivel regional y nacional.

El servicio aplica para cualquier empresa desarrolladora de software, independientemente de que plataforma o que metodología usa en su desarrollo.

3.2.2 Justificación.

Las empresas desarrolladoras de software de Pereira presentan fallas al aplicar los procesos de pruebas, ya que solo llevan a cabo pruebas

funcionales realizadas en su mayoría por los desarrolladores y/o clientes¹³. Estas empresas requieren del servicio para desarrollar un producto de mejor calidad, garantizando el cumplimiento de los requerimientos funcionales y no funcionales del cliente. Otra razón por lo cual las empresas necesitan del servicio, es para ayudar a detectar y corregir errores en etapas tempranas del proceso, evitando el reproceso, disminuyendo costos y tiempo en el desarrollo.

La industria del software tiene una ventaja frente a otros mercados, como es la de permitir la expansión fácil y rápida de sus productos y servicios, permitiendo a las empresas incursionar en mercados nuevos, exportar sus servicios y darse a conocer internacionalmente, razón por la cual es necesario reevaluar todos los procesos, principalmente aquellos que garantizan la calidad del producto, como es el proceso de pruebas, ya que si el software se somete a un adecuado y completo proceso, es posible garantizar la disminución significativa de la cantidad de errores que encontrarán cuando se pone en marcha.

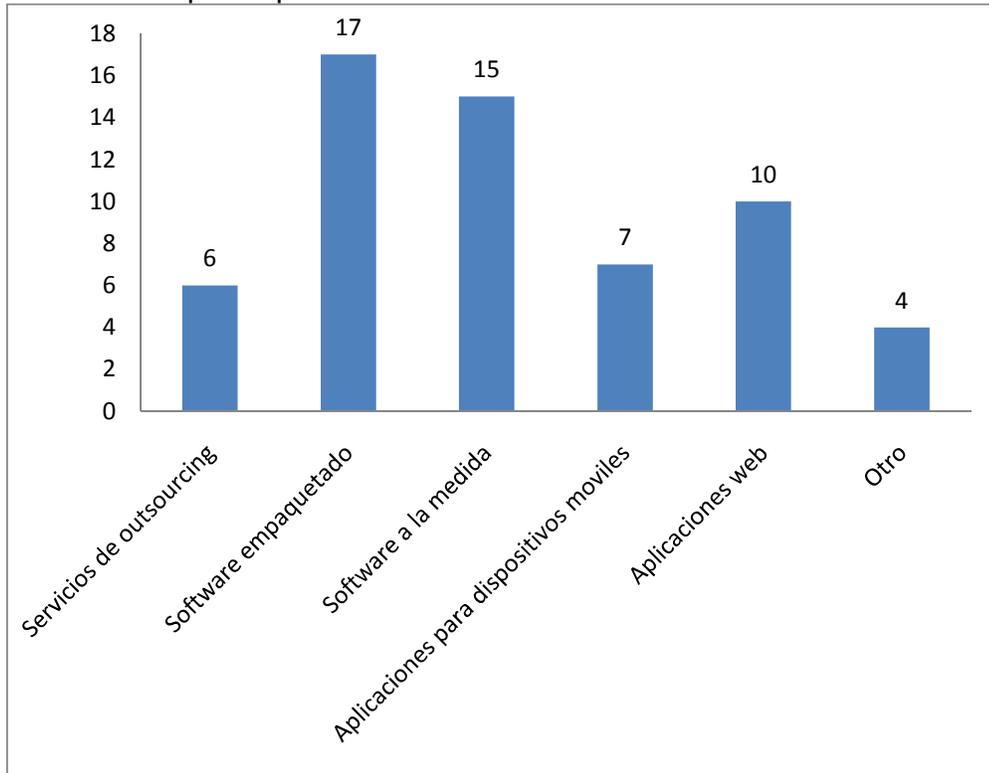
Las empresas desarrolladoras requieren contratar el servicio de pruebas en sus proyectos, ya que para llevar a cabo este proceso es necesario contar con profesional especializado y enfocado solo en esta actividad, permitiendo de esta manera invertir el tiempo de los desarrolladores del proyecto en su tarea principal. De igual forma resulta más eficiente dejar el proceso de pruebas en manos expertas, ya que los desarrolladores llevan a cabo pruebas informales y poco exhaustivas, basados en que su trabajo posee pocos errores y funciona correctamente, además de no contar con el tiempo necesario para aplicar el proceso de pruebas completo.

3.2.3 Estimación del mercado potencial.

La estimación del mercado potencial para nuestro servicio, está basada en la encuesta realizada en el marco del proyecto laboratorio de calidad de software 2009, de la Universidad Tecnológica De Pereira (Anexo I). La cual muestra que en actualidad existen 20 empresas dedicadas al desarrollo de software como actividad principal y sus principales características.

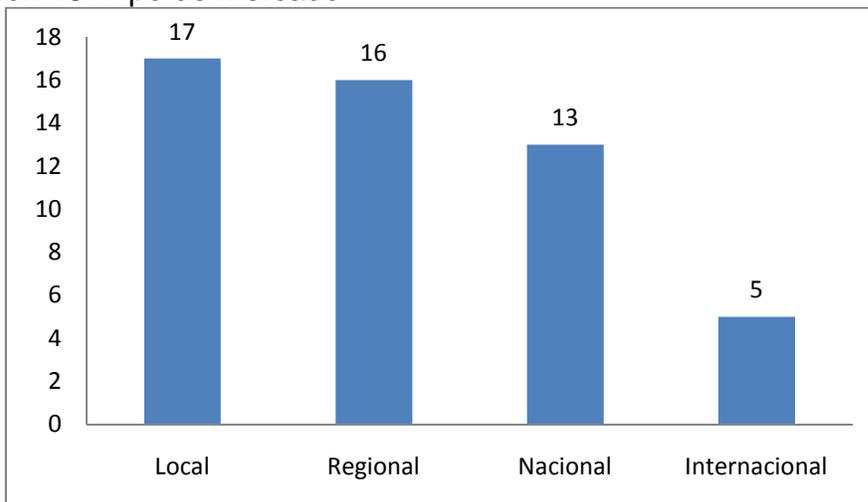
¹³ Encuesta realizada por el grupo de Investigación GRANDE, de la Universidad Tecnológica de Pereira, pregunta 9 y 10.

Ilustración 17. Tipo de producto o servicio



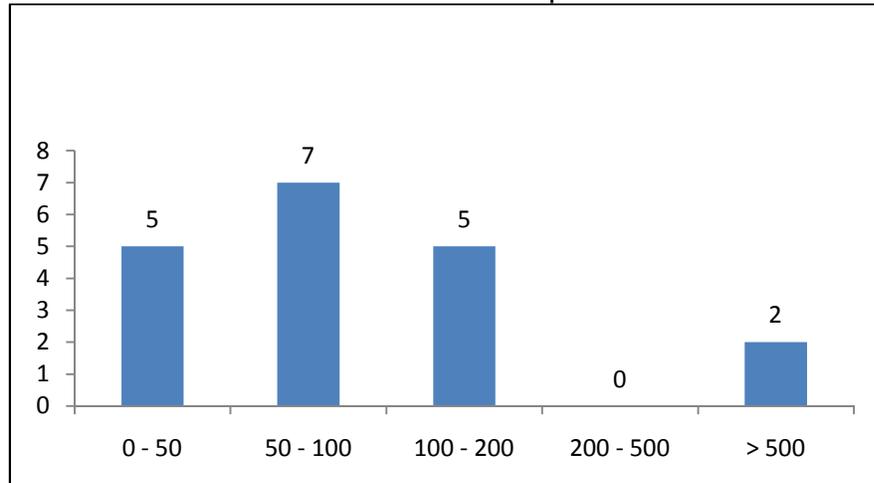
Fuente: Encuesta realizada en el marco del proyecto laboratorio de calidad de software 2009.

Ilustración 18. Tipo de mercado



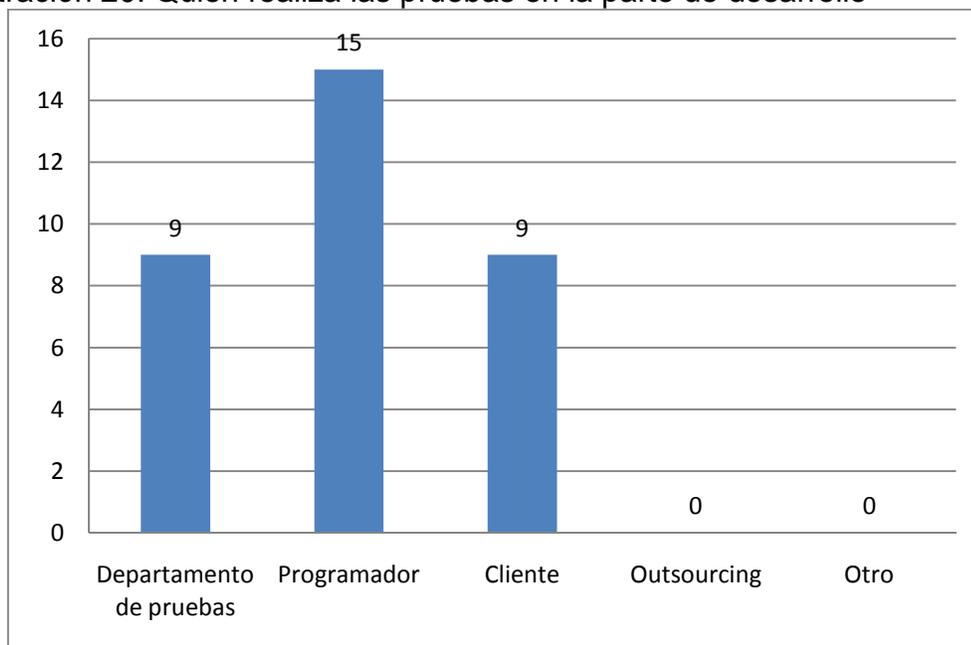
Fuente: Encuesta realizada en el marco del proyecto laboratorio de calidad de software 2009.

Ilustración 19. Ventas anuales en millones de pesos



Fuente: Encuesta realizada en el marco del proyecto laboratorio de calidad de software 2009.

Ilustración 20. Quien realiza las pruebas en la parte de desarrollo



Fuente: Encuesta realizada en el marco del proyecto laboratorio de calidad de software 2009.

Estas ilustraciones muestran las características de las empresas desarrolladoras de software de Pereira, lo cual nos permite conocer el tipo de producto y/o servicio que desarrolla, el mercado en el cual lo están comercializando, el valor de las ventas que obtienen anualmente y quiénes

son los encargados de realizar las pruebas. Siendo este último, el dato más relevante para determinar que las empresas requieren del servicio de pruebas de software.

3.2.4 Perfil del consumidor.

Las empresas que requieren del servicio de QualityTest S.A.S, son las desarrolladoras de software que han percibido que la calidad de sus productos puede mejorar para así ingresar a nuevos mercados y dar satisfacción a sus clientes.

Otras empresas que requieren del servicio son las que se encuentran en la situación de adquirir un producto de software del cual necesita garantizar que se integre completamente a su sistema de información, además de que sea un producto de excelente calidad.

3.3 ANÁLISIS DE LA COMPETENCIA

En la actualidad la presencia de empresas que ofrecen un servicio como el nuestro, están ubicadas en las ciudades más grandes del país (Bogotá, Cali y Medellín), siendo limitada la oferta que puedan tener en regiones como la nuestra, pero de igual manera debido a que en la industria del software esta limitación es poco importante, se deben tener como referencia y tenerlas en cuenta como posible competencia.

3.3.1 Galia Technologies: servicios de calidad de software.

Empresa colombiana con sede en la ciudad de Bogotá, que brinda servicios de ingeniería de requerimientos, control de versiones y configuración y pruebas (testing), pruebas funcionales, pruebas técnicas, pruebas de seguridad y pruebas de desempeño.

3.3.2 Choucair Testing S.A.: pruebas de software (Software Testing) para aplicativos de ruta crítica.

Empresa colombiana creada desde 1999 en la ciudad de Medellín, su actividad principal es el servicio de pruebas de software.

Choucair ha participado en más de 3.200 proyectos con resultados en la disminución de los tiempos improductivos, reprocesos, afectación de usuarios y costos innecesarios. De esta forma Choucair interviene en cada una de las etapas del proceso de desarrollo de Software para que los clientes se concentren en su negocio.

Choucair tiene clientes en los sectores de: finanzas, manufactura, servicios públicos y desarrollo de software.

3.3.3 Indudata Ltda.: asesoría, consultoría, outsourcing y capacitación en aseguramiento de calidad de software.

Indudata es una empresa bogotana que presta servicio de outsourcing en pruebas de software, brinda asesoría para el montaje de oficinas de gestión de calidad de software y personalización del proceso de calidad en definición, administración, automatización y ejecución de pruebas de desempeño y de carga.

3.3.4 Q-Vision Technologies S.A: servicios de aseguramiento y control de calidad de software.

Compañía bogotana con trayectoria internacional compuesta por profesionales en tecnología de información e ingeniería de software que se especializa en la prestación de servicios de aseguramiento y control de calidad de software en organizaciones que utilizan y/o desarrollan software como soporte a la operación de su negocio.

Q-Vision ha logrado alcanzar altos niveles de madurez y capacidad en sus procesos de pruebas. Con la valoración y adopción de CMMI® como modelo para definición e implementación de los procesos de testing, y con la certificación bajo la norma internacional ISO 9001:2000

3.3.5 GreenSQA.

GreenSQA es una empresa que pertenece a Parquesoft Cali y está constituida desde el año 2003. En la actualidad brinda sus servicios a través de dos modalidades principalmente:

- In-house: cuando el equipo humano se alojan en instalaciones del cliente y se integran temporalmente en su organización.
- Outsourcing: cuando el servicio es realizado desde las instalaciones de GreenSQA.

GreenSQA brinda servicios en calidad de software, calidad de procesos y entrenamiento y cuentan con una plataforma integrada GreenVolution®, suite que provee herramientas para la gestión de los entregables y registros asociados a las diferentes etapas del proceso de desarrollo de software, tales como los requerimientos, los planes y requerimientos de pruebas, las no conformidades, las notas de release, el banco de preguntas frecuentes y las medidas e indicadores.

GreenSQA otorga un sello de calidad a las empresas que cumplen con su ciclo de pruebas.

3.3.6 Posición frente a la competencia.

Las empresas que prestan un servicio similar al ofrecido por QualityTest S.A.S, son empresas con amplio reconocimiento en el sector, debido a su trayectoria y número de casos de éxito, con los cuales QualityTest S.A.S no cuenta, lo que conlleva a tener un bajo posicionamiento en el mercado de las pruebas de software.

3.3.7 Análisis del costo.

El costo del servicio de pruebas de software varía según aspectos como:

- Alcance del servicio: se define el tipo de pruebas a realizar y en que etapas del ciclo de vida del desarrollo las vamos a llevar a cabo.
- Tiempo del servicio: es el tiempo necesario para invertir en el proceso de pruebas de software. Se define de acuerdo al tiempo total del proyecto y al nivel de madurez del ciclo de vida de desarrollo.

- Equipo de desarrollo: se tiene en cuenta el nivel de calidad del equipo de desarrollo con el cual interactúa nuestro equipo de pruebas.
- Costos de personal: especialistas en proceso de pruebas.
- Costos de infraestructura: equipos, locaciones y herramientas, que sean necesarios para llevar a cabo el proceso de pruebas.

El costo de los servicios ofrecidos por QualityTest S.A.S. se calculo teniendo en cuenta el tiempo que requiere realizar cada tipo de prueba por parte del equipo de pruebas (director operativo y analista de pruebas), y los costos de los salarios administrativos (director ejecutivo y asesor contable) y de ventas (Asesor comercial).

Tabla 3. Costo del servicio.

| Prueba | Tiempo (meses) | Costos mano de obra | Costos salarios admón. | Costos salarios ventas | Total |
|---------------------------|----------------|---------------------|------------------------|------------------------|---------------------|
| Pruebas de requerimientos | 0.5 | 600.000 | 126.388 | 60.788 | 762.175 |
| Pruebas de diseño | 0.5 | 600.000 | 126.388 | 60.788 | 762.175 |
| Pruebas de integración | 1.5 | 1.800.000 | 379.163 | 182.363 | 2.286.525 |
| Pruebas funcionales | 1.5 | 1.800.000 | 379.163 | 182.363 | 2.286.525 |
| Pruebas de aceptación | 1 | 1.200.000 | 252.775 | 121.575 | 1.524.350 |
| Pruebas de seguridad | 1 | 1.200.000 | 252.775 | 121.575 | 1.524.350 |
| Pruebas de stress | 1 | 1.200.000 | 252.775 | 121.575 | 1.524.350 |
| Pruebas de rendimiento | 1 | 1.200.000 | 252.775 | 121.575 | 1.524.350 |
| Total | | | | | \$12.594.800 |

3.3.8 Análisis precio de venta

El precio de los servicios que ofrece la empresa QualityTest S.A.S. es determinado teniendo en cuenta factores como los gastos administrativos y la ganancia que se espera obtener por la venta de los servicios de pruebas.

Tabla 4. Precio del servicio

| Prueba | Precio |
|---------------------------|---------------------|
| Pruebas de requerimientos | 1.500.000 |
| Pruebas de diseño | 1.500.000 |
| Pruebas de integración | 4.000.000 |
| Pruebas funcionales | 4.000.000 |
| Pruebas de aceptación | 3.000.000 |
| Pruebas de seguridad | 3.000.000 |
| Pruebas de stress | 3.000.000 |
| Pruebas de rendimiento | 3.000.000 |
| Total | \$23.000.000 |

3.4 CONCEPTO DEL SERVICIO

QualityTest S.A.S ofrece el servicio de acompañamiento en el desarrollo de una aplicación, evaluándola desde un punto de vista crítico por medio del proceso de pruebas, principalmente a desarrolladoras de software de Risaralda, utilizando el conocimiento y herramientas, para así ayudar a reducir el costo del desarrollo, ya que se previenen o detectan errores y defectos en etapas tempranas haciendo una corrección oportuna.

El servicio de pruebas de QualityTest S.A.S, se realiza en las instalaciones del cliente y comienza en la fase donde se lleva a cabo la definición de los requerimientos, terminando con la aplicación de pruebas al sistema completo. El acompañamiento se hace en las siguientes etapas del ciclo de vida del desarrollo:

- **Requerimientos:** en esta etapa del proyecto, se verifica si los requisitos se encuentran completos y que sean consistentes, rastreables, probables (testeable). Esta verificación se realiza mediante pruebas de requerimientos, las cuales son realizadas en paralelo a esta etapa del desarrollo.
- **Diseño:** en esta etapa del proyecto se realiza la prueba de diseño, la cual busca detectar e identificar no conformidades en el diseño del producto de software antes de realizar la codificación.
- **Construcción:** en esta etapa se verifica si el programa es consistente con el diseño, mediante la ejecución de pruebas de integración y

funcionales. Las pruebas de aceptación, de seguridad, de stress y de rendimiento se realizan para probar requisitos no funcionales del sistema.

El trabajo que realiza QualityTest S.A.S es bajo una clausula de confidencialidad, por medio de la cual se compromete con el cliente en tener absoluta reserva con el código que se está manejando y a su vez, respetar la propiedad intelectual del software, ya que sabemos que el mayor activo con el cuenta la empresas desarrolladoras de software es código de sus productos.

3.5 ESTRATEGIAS DE MERCADEO.

Generalmente, las empresas esperan hasta el final del ciclo del proyecto para probar el software y a veces limitan las pruebas por falta de tiempo. Esto puede repercutir en costos extras, incumplimiento de plazos e incluso insatisfacción de clientes y usuarios.

- Crear en las empresas desarrolladoras de software la necesidad de adoptar el proceso de pruebas como parte importante del ciclo de desarrollo de software para así asegurar el incremento de la calidad del producto, detectando las fallas en etapas tempranas y a la vez disminuyendo costos de desarrollo.
- Dictar seminarios gratuitos a los ingenieros desarrolladores de las empresas de software del departamento, con el fin de crear un cambio cultural en el proceso de desarrollar proyectos, mediante la aplicación de una metodología de ingeniería de software desde el inicio de su proceso de desarrollo, realizándolo completamente y llevando a cabo las pruebas necesarias para asegurar el funcionamiento esperado del producto.
- Crear blogs para interactuar con personas interesadas en el tema de pruebas y así darnos a conocer en este gremio, compartiendo tips que nos retroalimenten.
- Participación en ferias, eventos y congresos donde se exponga la importancia del proceso de pruebas en el ciclo de vida del desarrollo de software.

3.6 ESTRATEGIAS DE PROMOCIÓN

Para lograr que QualityTest S.A.S. se posicione y sea reconocida a nivel regional, inicialmente se aprovechara la condición del sector al cual se va a ingresar, ya que es un sector pequeño y esto permitirá prestar el servicio de una manera muy personalizada, conociendo las necesidades particulares de cada empresa desarrolladora de software.

- Se da a conocer el servicio de QualityTest S.A.S, con la participación en las ferias comerciales más importantes que se llevan cabo en el país
- Participando en el Congreso Colombiano de Computación y en el Campus Party que se lleva a cabo cada año.
- Patrocinando eventos de programación en las universidades de la región.
- Participando en los eventos programados por la Asociación Colombiana de Ingenieros de Sistemas, dictando conferencias por ejemplo.

3.7 ESTRATEGIAS DE COMUNICACIÓN.

Se realiza publicidad de la empresa QualityTest S.A.S. por medio de revistas, páginas amarillas (sección de programación y diseño de software), y en el portal de catálogo de software La empresa también cuenta con sitio web, en el cual se va a publicar información relacionada con el servicio e interés para el sector de la industria del software. QualityTest S.A.S se asociara a gremios y promocionara el servicio por medio de clústeres.

3.8 ESTRATEGIAS DE SERVICIO.

QualityTest S.A.S presta asesoría de servicio, realizando unas capacitaciones sobre aspectos a tener en cuenta en el transcurso del

desarrollo. También da un servicio muy personalizado enfocado a garantizar el aseguramiento de la calidad del producto del cliente.

3.9 PROYECCIONES DE VENTAS.

Las proyecciones de ventas se realizan con el objetivo de encontrar un punto de equilibrio, basados en el número de empresas desarrolladoras de software de la ciudad de Pereira y en las expectativas y metas de ventas de la empresa, lo cual permita su funcionamiento los primeros años.

Para calcular estas proyecciones suponemos que en cada una de las empresas de la ciudad de Pereira se realiza un proyecto nuevo por año, tres de ellas contratan el servicio de pruebas completo en el primer año. Adicionalmente las empresas que cuentan con productos ya desarrollados, contratan el servicio de pruebas como: de integración, funcionales, aceptación y rendimiento, para nuevas versiones, que producirán en los años posteriores.

Tabla 5. Unidades vendidas primer año

| Prueba | Unidades vendidas |
|---------------------------|--------------------------|
| Pruebas de requerimientos | 3 |
| Pruebas de diseño | 3 |
| Pruebas de integración | 3 |
| Pruebas funcionales | 4 |
| Pruebas de aceptación | 3 |
| Pruebas de seguridad | 3 |
| Pruebas de stress | 3 |
| Pruebas de rendimiento | 3 |
| Total | 25 |

Tabla 6. Ventas primer año

| Prueba | Ventas |
|---------------------------|----------------------|
| Pruebas de requerimientos | 4.500.000 |
| Pruebas de diseño | 4.500.000 |
| Pruebas de integración | 12.000.000 |
| Pruebas funcionales | 16.000.000 |
| Pruebas de aceptación | 9.000.000 |
| Pruebas de seguridad | 9.000.000 |
| Pruebas de stress | 9.000.000 |
| Pruebas de rendimiento | 9.000.000 |
| Total | \$ 73.000.000 |

Para el segundo y tercer año se espera contar con la contratación de tres servicios de pruebas completos como mínimo, y para el cuarto y quinto año con cinco. También se espera contar con la venta adicional de las pruebas: integración, funcionales, aceptación y rendimiento, las cuales son contratadas para desarrollos terminados, a los cuales se les hacen modificaciones en su funcionalidad dando como resultados nuevas versiones.

MODULO OPERACIÓN

4.1 FICHA TÉCNICA DEL SERVICIO

El proceso de pruebas en un desarrollo de software es una de las etapas más costosas, ya que se debe realizar pruebas a cada una de las partes del sistema. Para describir el servicio, se debe tener en cuenta que cada proyecto es diferente y cada uno necesita una determinada arquitectura, según su objetivo, necesitamos una metodología y plan específico para cada uno de ellos.

QualityTest S.A.S ofrece un el servicio de pruebas en las instalaciones del cliente, aplicables en tres etapas del ciclo de vida del desarrollo, requerimientos, diseño y construcción.

4.1.1 Prueba de requerimientos.

El objetivo de la prueba de requerimientos es revisar el grado de madurez de la definición, especificación y administración de los requisitos de software. Estas revisiones, aseguran que en la fase de análisis de requerimiento, se establecieron los requerimientos del cliente con el grupo de desarrollo, los cuales serán aplicados en el proyecto de software. Se deben tener en cuenta los siguientes criterios:

- Identificación de requerimientos funcionales y no funcionales
- Calidad en la especificación y definición de requerimientos
- Gestión de los requerimientos en el ciclo del desarrollo
- Administración de controles de cambio
- Mecanismos de medición de requerimientos.

Con la prueba de requerimientos se detectaran errores para garantizar las siguientes características de los requisitos: carencia de ambigüedad, especificación completa y clara del problema, y consistencia para asegurar que no hayan requisitos contradictorios. Esta prueba se realiza en paralelo a la etapa de requerimiento de desarrollo de software.

4.1.2 Prueba de diseño.

Por medio de esta prueba se revisa el diseño de los productos de software con el objetivo de detectar e identificar no conformidades en el diseño antes de pasar a la codificación. Se verifica la arquitectura y se da apoyo a la modelación de los requisitos del cliente.

Esta prueba se realiza en paralelo a la etapa de diseño de desarrollo de software.

4.1.3 Prueba de integración.

La prueba de integración se ejecuta con el objetivo de comprobar que la comunicación entre los diferentes módulos de un desarrollo integrado sea correcta. Esta prueba se lleva a cabo en la etapa de construcción del desarrollo de software.

QualityTest S.A.S posee en su metodología de integración, dos estrategias, las cuales son aplicadas según las necesidades del proyecto de integración:

- De arriba a abajo: consiste en empezar por los módulos que se encuentran en los niveles superiores, e integrar incrementalmente los niveles inferiores, verificando que el llamado se haga de manera correcta.
- De abajo a arriba: consiste en empezar por los módulos que se encuentran en los niveles inferiores, e integrar incrementalmente en los niveles superiores, verificando que el llamado se haga de manera correcta.

4.1.4 Pruebas funcionales.

Las pruebas funcionales validan que la funcionalidad del aplicativo cumpla con los requisitos y el diseño establecidos por el cliente, como también su comportamiento frente a entradas no esperadas. QualityTest S.A.S. utiliza la técnica de caja negra para realizar esta prueba en la etapa de construcción del ciclo de vida del desarrollo.

4.1.5 Prueba de aceptación.

La prueba de aceptación valida que el sistema cumpla con el funcionamiento esperado y permite que el usuario de dicho sistema determine su aceptación, desde el punto de vista de su rendimiento y funcionalidad.

Mediante la realización de pruebas de caja negra se logra la validación del sistema, ya que se demuestra la conformidad de los requerimientos, los cuales se recogen en el plan de pruebas. Dicho plan asegura que todos los requisitos funcionales especificados por el usuario se satisfacen, teniendo en cuenta también los requisitos no funcionales los cuales están relacionados con el rendimiento y la seguridad de acceso al sistema, a los datos y procesos.

4.1.6 Prueba de seguridad.

Antes de la liberación del sistema, se verifica por medio de la prueba de seguridad el correcto funcionamiento de los mecanismos de protección que posee el sistema para asegurar la integridad y confidencialidad de los datos. Con la realización de esta prueba se detecta la existencia de vulnerabilidades frente accesos y manipulaciones no autorizadas, como también defectos de seguridad para eliminarlos y así evitar el riesgo y costo que ellos ocasionarían en el ambiente final de operación del sistema. Dicha prueba se lleva a cabo mediante las siguientes actividades:

- Identificar cada tipo de usuario con sus respectivas funciones y datos a los que se debe autorizar.
- Crear pruebas para cada tipo de usuario y verificar cada permiso mediante transacciones específicas para cada uno de ellos.
- Modificar tipos de usuarios y realizar nuevamente las pruebas.

4.1.7 Prueba de stress.

La prueba de stress se encarga de verificar que el sistema funcione correctamente bajo condiciones que sobrecargan recursos como son: memoria baja o no disponible en el servidor, máximo número de usuarios

conectados, múltiples usuarios realizando la misma transacción. Aunque las condiciones en que se desarrollan estas pruebas pueden no presentarse en la realidad, es importante realizarlas porque se deben esperar los mismos resultados al aproximarse a las condiciones anormales.

La ejecución de las pruebas de stress se efectúan con el fin de probar recursos limitados, las pruebas se corren en un servidor con configuración reducida o limitada y utilizando múltiples clientes.

4.1.8 Pruebas de rendimiento.

El objetivo de las pruebas de rendimiento es validar los tiempos de respuesta, índices de procesamiento de transacciones y demás requisitos sensibles al tiempo. Este tipo de pruebas se ejecutan varias veces con diferentes cargas al sistema, llevando a cabo las siguientes actividades:

- Comparar el desempeño del sistema actual con los requisitos
- Poner a punto el sistema para mejorar las métricas de desempeño y proyectar la capacidad futura de carga del sistema.

El desempeño de los módulos individuales se evalúa cuando se realiza la prueba de caja blanca, pero el rendimiento del sistema no se puede determinar hasta que no estén integrados todos los elementos.

QualityTest S.A.S realiza las siguientes pruebas de rendimiento:

- Prueba normal: se realiza con solo un usuario conectado a la aplicación, permite conocer el tiempo de respuesta medio y establecer una referencia para posibles comparaciones y análisis.
- Prueba con número mínimo de usuarios. Las pruebas de los sistemas son realizadas con el número de usuarios mínimo establecido.
- Prueba con número máximo de usuarios. Las pruebas de los sistemas son realizadas con el número de usuarios máximo establecido.

- Prueba de número máximo soportado de usuarios. Se busca encontrar cuál es el límite del sistema.

4.1.9 Herramientas de automatización de pruebas.

La selección de las herramientas de automatización que vamos a usar en el proceso de pruebas depende de los lenguajes en los que se realice el desarrollo, el tamaño del sistema y el tipo de prueba que se va a realizar.

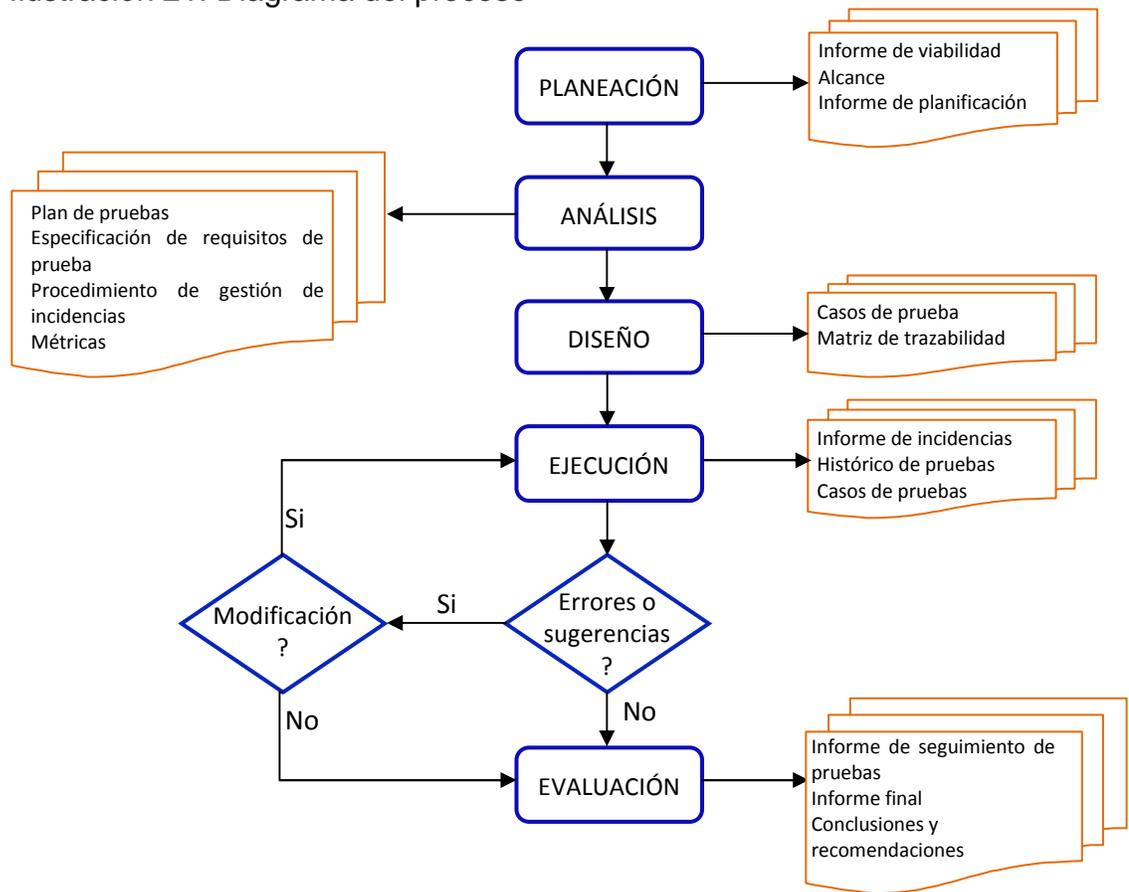
Algunas candidatas son:

- Herramienta de gestión a usar será mantis.
- Pruebas de aceptación: fitnessse, Avignon.
- Pruebas funcionales: HTTPUnit, SAHL.
- Pruebas de stress: Jmeter.
- Pruebas de rendimiento: Jmeter, grinder.

4.2 DESCRIPCIÓN DEL PROCESO DE PRUEBAS.

El proceso de pruebas que aplica QualityTest S.A.S. en un proyecto consta de las siguientes etapas:

Ilustración 21. Diagrama del proceso

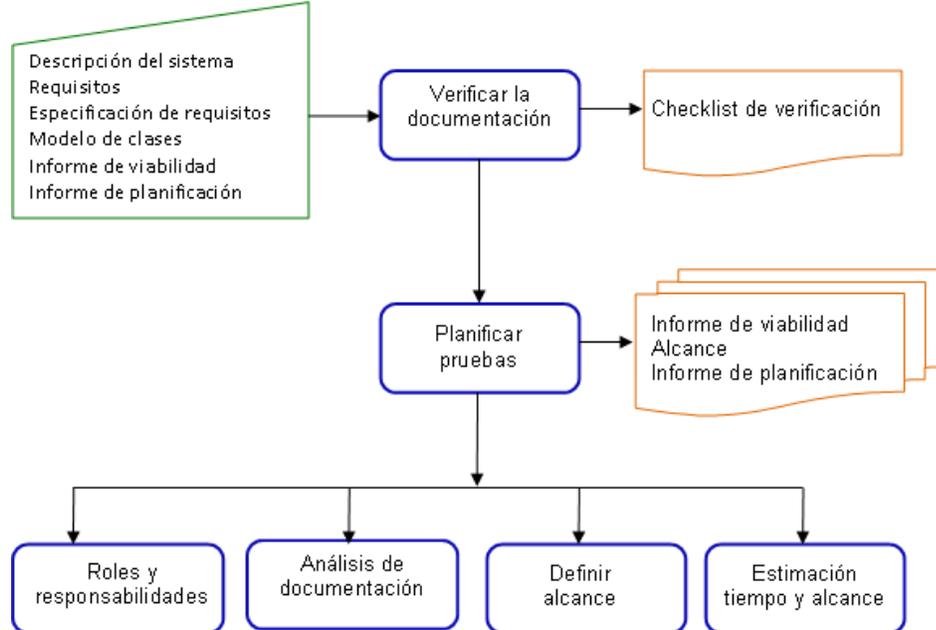


Fuente: autor.

4.2.1 Planificación de las pruebas.

La planificación de las pruebas permite asegurar el cumplimiento de los plazos establecidos dentro del proyecto. El objetivo de esta etapa es realizar la planificación de las actividades a realizar en el ciclo de vida del proyecto para evitar desviaciones en él y especialmente en las actividades de la etapa de pruebas.

Ilustración 22. Actividades etapa de planificación de pruebas



Fuente: autor.

- Verificar la documentación: se verifica que el contenido de los documentos entregados satisfacen los mínimos controles de calidad. Esta verificación se realiza por medio de un check list.
- Planificar las pruebas: se determina la viabilidad, el alcance y los tipos de pruebas que se van a realizar, como también la estimación de tiempo y recursos necesarios para llevarlas a cabo. Dentro de esta fase se llevan a cabo las siguientes actividades:
 - Roles y responsabilidades: se definen los equipos involucrados en la realización de las pruebas y se reparten las responsabilidades entre los diferentes participantes.
 - Análisis de documentación: se estudia la documentación recibida y se analizan los requisitos, identificándolos como funcionales y no funcionales.
 - Definir alcance: se define el alcance de las pruebas y los tipos de prueba que vamos a realizar

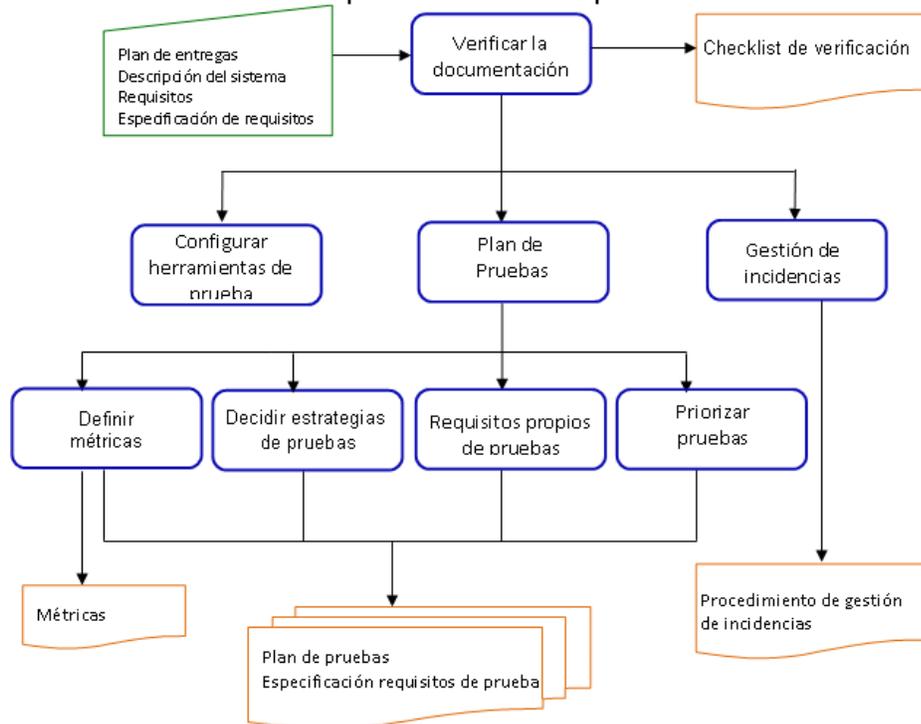
- Estimación de tiempo y esfuerzo de las pruebas: se realiza en base a los módulos a probar, los tipos de prueba y la herramienta de gestión de prueba establecida

Como resultado de la fase de planificar las pruebas, se obtienen dos informes: informe de viabilidad y alcance (Anexo A), y el informe de planificación con las actividades a realizar y la estimación de tiempos y recursos necesarios para llevar a cabo las pruebas (Anexo B).

4.2.2 Análisis de las pruebas.

Una vez realizada la planificación de las pruebas, se realiza la revisión detallada del sistema, comprobando que los requisitos y el diseño del mismo sean coherentes.

Ilustración 23. Actividades etapa de análisis de pruebas



Fuente: autor.

- Verificar la documentación: se revisa que el contenido de los documentos satisfagan los controles de calidad establecidos; aunque algunos documentos de entrada ya fueron verificados en la etapa de

planificación, deben ser revisados nuevamente si sufrieron algún cambio. Esta verificación se hace por medio de un check list.

- Configurar herramientas de pruebas: en esta fase se crea la estructura del proyecto de prueba en la herramienta de gestión la cual almacenará los datos y la documentación del proyecto.

Desde la etapa de planificación se debe utilizar una herramienta de gestión de pruebas, la cual brinda las siguientes ventajas para la realización de cada una de las actividades del proyecto:

- Garantiza la definición completa del plan de pruebas
- Crea un repositorio de casos de prueba reutilizables
- Facilita el diseño de casos de prueba
- Facilita la trazabilidad de los casos de prueba
- Facilita la obtención métricas y generación de informes.

- Plan de pruebas: se definen las actividades para la ejecución de las pruebas que verifiquen el correcto funcionamiento de sistema.

El plan de pruebas, a lo largo del proyecto, puede tener modificaciones, ya que puede ocurrir que durante la ejecución de las pruebas se encuentren situaciones no contempladas inicialmente (Anexo C).

- Definición de métricas: la definición de métricas e indicadores facilitan el control sobre la evaluación de las pruebas. Estas métricas deben permitir realizar un seguimiento del grado de avance del plan de pruebas, resúmenes de ejecución de pruebas, indicadores de progreso y productividad, seguimiento y control de incidencias y defectos.
- Definir estrategias de pruebas: se termina de detallar la estrategia a seguir a lo largo del proyecto, dependiendo del alcance, de los tipos de prueba y de los requisitos establecidos.

La estrategia de prueba definida debe contener: identificación de las herramientas a utilizar, definición del alcance de

automatización, las métricas a utilizar y las posibles configuraciones de hardware a utilizar en las pruebas, así como las necesidades software y/o combinaciones de ambos. Para los elementos que se han determinado que no se pueden probar, se debe determinar cómo serán tratados.

- Requisitos de pruebas: se deben identificar los requisitos funcionales y no funcionales que correspondan al módulo para los cuales se especificaran los casos de prueba que deben ejecutarse.

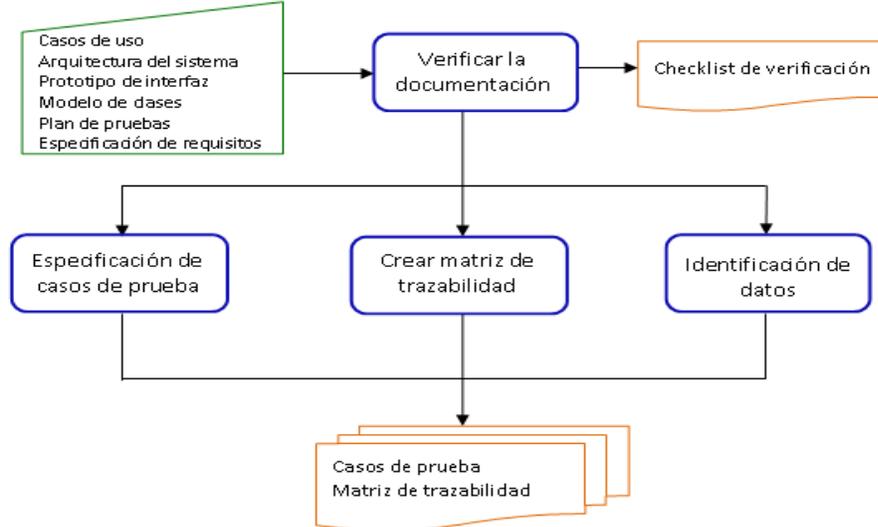
 - Priorización de las pruebas: a cada requerimiento se le asigna un nivel de complejidad, utilizando la siguiente escala: 3 - Alto, 2 - Medio, 1 – Bajo, los cuales son acordados con el usuario.
- Gestión de incidencias: se definen las actividades a realizar para cualquier anomalía o modificación que puedan afectar el desarrollo y funcionamiento del sistema.

El propósito del procedimiento de gestión de incidencias es: identificar los estados de la incidencia desde su detección hasta su fin e identificar sus posibles orígenes.

4.2.3 Diseño de las pruebas.

Durante la etapa de diseño de pruebas se elaboran e implementan los casos de prueba detalladamente, los cuales deben cumplir los requisitos documentados funcionales y no funcionales del sistema, también se debe indicar los pasos y las técnicas a aplicar. Los casos de prueba describe el conjunto de entradas, las condiciones de ejecución y los resultados esperados,

Ilustración 24. Actividades etapa diseño de pruebas



Fuente: autor

- Verificar la documentación: se verifica que el contenido de los documentos entregados en la etapa de diseño sean correctos y cumplan con las metodologías de desarrollo utilizadas en el proyecto, aunque hay documentos de entrada que fueron verificados en las etapas anteriores del proceso de pruebas, si estos han sufrido cambios, deben ser verificados nuevamente por medio de un check list.
- Especificación de casos de prueba: para generar los casos de prueba, se debe establecer una jerarquía que permita organizar los requisitos de negocio a probar y para cada conjunto de requisitos funcionales o no funcionales, existirá una serie de situaciones que someterá al sistema a diferentes estímulos frente a los cuales se obtendrá una determinada respuesta. Dentro de este proceso se llevan a cabo las siguientes actividades:
 - Definir los escenarios de negocio: los escenarios de negocio son agrupaciones de actividades que garantizan una función particular del negocio del cliente.
 - Definir las reglas del negocio: las reglas del negocio son las actividades que componen los escenarios del negocio.

- Especificar los casos de prueba: los casos de prueba define las funcionalidades a probar dentro del plan de pruebas.
- Definir los test scripts: especificación de pasos a seguir para automatizar un caso de prueba.
- Automatizar los test scripts: se realiza la automatización de los test scripts para los casos de pruebas que se haya determinado automatizar.

En la fase de especificación de casos de prueba obtiene el documento de especificación de casos de prueba, el cual especifica formalmente las condiciones necesarias para llevar a cabo las pruebas de software. Contiene información como las estradas, los resultados esperados, condiciones de ejecución y restricciones (Anexo D).

- Matriz de trazabilidad: esta matriz permite asociar los casos de prueba y controlar la cobertura de los requerimientos a medida que se va avanzando en la ejecución de pruebas. Cada caso de uso debe estar asociado como mínimo con un caso de prueba (Anexo E). En este proceso se realizan las siguientes actividades:
 - Asociar los casos de prueba a los requisitos
 - Actualizar el plan de pruebas con la matriz de trazabilidad.
- Identificación de datos: se identifican los datos de entrada y salida que son necesarios para la ejecución de los casos de prueba. Se deben tener en cuenta los datos que serán proporcionados por sistemas externos. La identificación de los datos requiere las siguientes actividades:
 - Identificar los datos requeridos cada tipo de prueba y su formato.
 - Gestionar la solicitud y recepción de los datos por parte de los sistemas externos, garantizando la confidencialidad e integridad referencial de los datos.

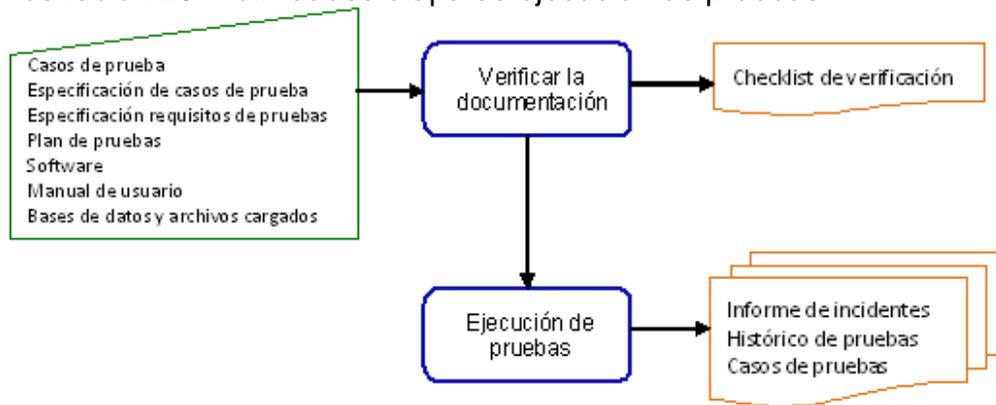
- Generar datos no proporcionados.

La información generada de la fase identificación de datos, complementa el documento de especificación de casos de prueba.

4.2.4 Ejecución de las pruebas.

En la etapa de ejecución de las pruebas se realizan e implementan los casos de prueba que se definieron anteriormente.

Ilustración 25. Actividades etapa de ejecución de pruebas



Fuente: autor

- Verificar la documentación: se verifica que el contenido de los documentos entregados en la etapa de ejecución de pruebas sean correctos y satisface unos mínimos controles de calidad, aunque hay documentos de entrada que fueron verificados en las etapas anteriores del proceso de pruebas, si estos han sufrido cambios, deben ser verificados nuevamente por medio de un check list.
- Ejecución de pruebas: se realiza la ejecución manual y de scripts automáticos. El orden de ejecución será por prioridad decreciente, ejecutando en primer lugar los casos de alta prioridad de acuerdo con la planificación definida. Este proceso se realiza siguiendo las actividades:

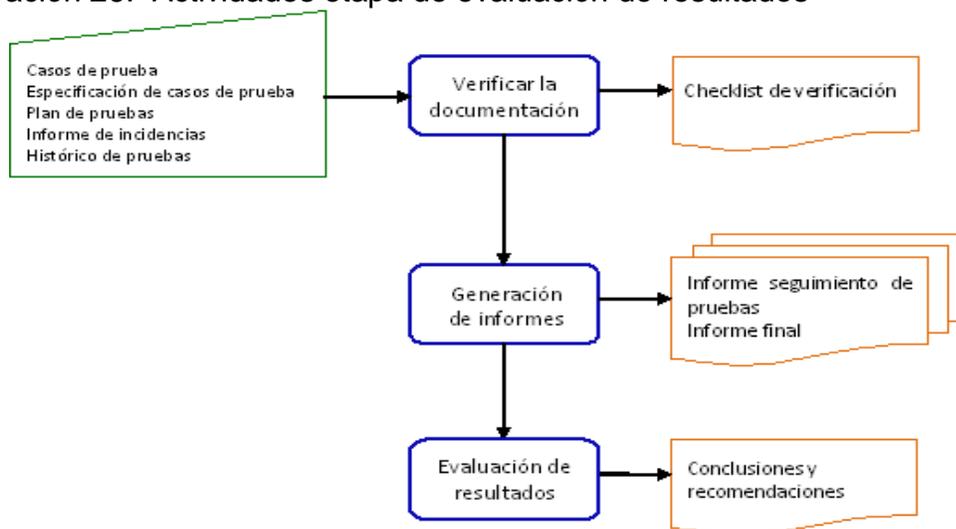
- Ejecutar los casos de prueba: se ejecutan los casos de prueba de acuerdo si son manuales o automatizadas.
- Realizar la gestión de incidencias: esta gestión se realiza de acuerdo a lo expresado en el documento procedimiento de gestión de incidencias.
- Registrar los resultados en la herramienta de pruebas: De esta manera se tiene un inventario de todos los casos de pruebas ejecutados, sus estados, las incidencias asociadas, así como el histórico de resolución de las mismas.

Los documentos generados en la etapa de ejecución de pruebas son: el informe de incidencias (Anexo F), histórico de pruebas (Anexo G) y se actualiza el documento de especificación de casos de pruebas con los resultados obtenidos durante la ejecución de los casos de prueba.

4.2.5 Evaluación de resultados.

El objetivo de esta etapa es evaluar los resultados obtenidos en el informe para obtener una visión del estado del proyecto y así tomar la decisión de realizar una nueva iteración de pruebas o no.

Ilustración 26. Actividades etapa de evaluación de resultados



Fuente: autor.

- Verificar la documentación: se revisa que el contenido de los documentos de entrada cumplan con los controles de calidad establecidos; aunque algunos documentos de entrada ya fueron verificados en etapas anteriores, deben ser revisados nuevamente si sufrieron algún cambio. Esta verificación se hace por medio de un check list.

- Generación de informes: por medio de los informes se determina el nivel de calidad obtenido tras el proceso de ejecución, extrayendo los resultados de las pruebas.
 - Informe de seguimiento: contiene un resumen de las actividades realizadas, en donde se detalla el estado del proyecto, las tareas realizadas y pendientes a realizar, principales problemas y desviaciones en la planificación

 - Informe final: se especifican los casos de prueba ejecutado con el resultado obtenido. Contiene una evaluación objetiva del estado del sistema con base a los datos obtenidos de las diferentes métricas e indicadores (Anexo H).

- Evaluación de resultados: se evalúa la información obtenida en los informes, con el objetivo de obtener conclusiones y recomendaciones.

4.3 NECESIDADES Y REQUERIMIENTOS

4.3.1 Hardware requerido.

Para la operación de la empresa QualityTest S.A.S es necesario contar con elementos de hardware que permitan el desarrollo de sus actividades con el fin de prestar un buen servicio.

4.3.1.1 Computadores.

Se requiere contar con dos computadores que permitan llevar a cabo el desarrollo del proceso de pruebas y que soporten las herramientas de automatización.

Características

Procesador: CORE 2 DUO
Disco Duro: 500 GB
Memoria RAM: 4 GB
Monitor: 19" LCD
Unidad: DVD – quemador
Multilector de tarjetas

4.3.1.2 Servidor.

Para garantizar la ejecución del proceso de pruebas completo y eficiente, se requiere adquirir un servidor que permita desarrollar establemente las actividades de dicho proceso.

Características

Procesador: Intel® Xeon® X3430 (8MB Caché, 2.40 GHz, Turbo, HT)
Disco Duro: RAID 1 - PERC S100 (SATA Software RAID Integrado)
soporta 2 Disco Duros
Disco Óptico Interno: DVD-ROM
2da Unidad Óptica: DVD Drive, Internal
Memoria: 4GB (4X1GB), 1066Mhz, Single Ranked UDIMM
Sistema Operativo: Windows Server® 2008 R2, Foundation
Adaptador de Red: Integrado de puerto doble, Gigabit Ethernet
Multiple selección de Discos Duro: 160GB 7.2k RPM Serial ATA 3Gbps 3.5-in
Cabled Hard Drive

4.3.1.3 Impresora.

QualityTest S.A.S necesita contar con esta herramienta para desarrollar actividades administrativas y generar la documentación del proceso de pruebas.

Características

Conectividad: 1 USB

Ciclo de trabajo: 1000 páginas
Velocidad en B/N: 30 ppm
Velocidad en color: 23 ppm
Resolución de impresión B/N: 1200 dpi renderizado negro
Resolución de impresión Color: 4800 x 1200 dpi optimizados y 1200 dpi de entrada
Resolución de escaneado por HW: 1200 x 2400 dpi

4.3.2 Software requerido.

Para el desarrollo del proceso de pruebas es necesario contar con las licencias del sistema operativo de los computadores, además de las herramientas de automatización de pruebas.

4.3.2.1 Sistema operativo.

El sistema operativo van a tener instalados los computadores es Windows 7.

4.3.2.2 Herramientas de automatización de pruebas.

Las herramientas que QualityTest S.A.S emplea no tienen ningún costo en su licencia de funcionamiento.

La herramienta que se utiliza para gestión es mantis.

4.3.3 Localización.

La empresa está ubicada en la ciudad de Pereira, capital del departamento de Risaralda. El lugar de trabajo consta de una oficina de 25 m², la cual está ubicada en la zona céntrica de la ciudad en el edificio Diario del Otún (calle 19 con carrera 10). El valor del alquiler del local es de \$500.000 pesos mensuales (incluida administración).

4.3.4 Presupuesto requerido.

Para la adquisición de los elementos de hardware y software se necesita de un presupuesto, que se discriminara a continuación:

Tabla 7. Presupuesto requerido

| Item | Cantidad | V. Unitario | V. Total |
|---------------------|----------|-------------|---------------------|
| Computador | 2 | 1.400.000 | 2.800.000 |
| Servidor | 1 | 2.500.000 | 2.500.000 |
| Switch | 1 | 150.000 | 150.000 |
| UPS | 1 | 270.000 | 270.000 |
| Licencias Windows 7 | 2 | 810.000 | 1.620.000 |
| Estabilizadores | 2 | 35.000 | 70.000 |
| Total | | | \$ 7.410.000 |

MODULO ORGANIZACIÓN

5.1 MISIÓN

Somos una empresa que presta el servicio de pruebas de software a empresas desarrolladoras de la región, haciendo un acompañamiento a nuestros clientes, mediante las técnicas y estrategias adecuadas para asegurar la calidad de sus productos a través del cumplimiento de los requerimientos del cliente.

5.2 VISIÓN

QualityTest S.A.S. será consolidada como una empresa pionera a nivel regional en la prestación del servicio de pruebas de software. Seremos una empresa con presencia nacional, brindando servicio y soluciones integrales de calidad a las empresas desarrolladoras.

5.3 VALORES

Calidad.

Es el requisito indispensable con el que debe contar toda actividad realizada, para garantizar la satisfacción de los clientes, a través de un servicio óptimo.

Responsabilidad.

Es el compromiso de cumplir con las obligaciones adquiridas con una persona o empresa y así dar respuesta a lo que se espera, asumiendo las consecuencias de las decisiones tomadas.

Confidencialidad.

Es la obligación de garantizar la confidencialidad de la información de nuestros clientes, sin ser revelada a ningún tercero.

Trabajo en equipo.

Es un valor fundamental para generar un ambiente de confianza, uniendo talentos y esfuerzos, para lograr los objetivos comunes.

5.4 GRUPO EMPRENDEDOR

El grupo emprendedor está conformado por Ana María Sierra G. y Johana Andrea Morales M., estudiantes de último semestre de Ingeniería de sistemas y computación de la Universidad Tecnológica de Pereira.

Perfiles.

Ana María Sierra González.

Persona responsable, disciplinada y organizada, con metas y propósitos concretos en los cuales utiliza sus principales habilidades para alcanzarlos, actuando por iniciativa propia. Desarrolla sus actividades con mucha vitalidad, controlando la calidad del trabajo y buscando continuamente el desarrollo profesional y personal; le gusta trabajar en equipo, investigar y preguntar.

Johana Andrea Morales Mejía.

Persona creativa, emprendedora, capaz para proponer y participar en proyectos multidisciplinarios y dar soluciones de tipo tecnológico, desempeñando labores y tomando decisiones bajo presión, basada en principios y valores éticos. Con capacidad de análisis, para identificar y solucionar problemas.

5.5 ANÁLISIS DOFA

Debilidades.

- Poco capacidad de inversión

- Sin suficiente personal calificado
- Tasa de retorno prolongado
- Falta de cultura en la utilización de metodologías de ingeniería de software
- Falta de confianza por parte de los clientes.

Fortalezas.

- Compromiso del equipo de trabajo
- Calidad del servicio
- Cumplimiento en el servicio
- Ampliación de servicios
- Costos reducidos al utilizar herramientas libres.

Oportunidades.

- Crecimiento del mercado
- Poca competencia
- Existencia de parques tecnológicos en la región
- Ley Quimbaya: descuentos y exenciones en pago de impuestos para empresas tecnológicas y de software de la región
- Alta demanda de soluciones tecnológicas.

Amenazas.

- Falta de cultura del proceso de pruebas por parte las empresas desarrolladoras
- Automatización de pruebas
- Empresas competidoras de otras ciudades creen sucursales en nuestra región.

Estrategias.

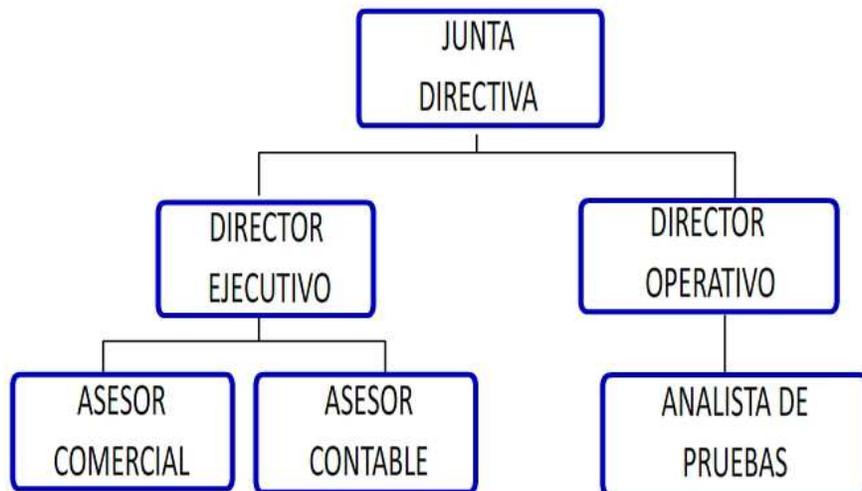
- Dar a conocer las ventajas y beneficios que generan a una empresa desarrolladora, la contratación de un servicio de pruebas externo.

- Documentos y contratos en los cuales se incluyen cláusulas de confidencialidad.
- Como principal política de inversión tendremos la permanente capacitación del equipo de pruebas, logrando obtener conocimiento que se verá reflejado en la calidad del servicio y en la satisfacción de nuestros clientes.
- Los servicios ofrecidos estarán en evolución constante de acuerdo con las necesidades de los clientes y adaptándose a los cambios tecnológicos.

5.6 ESTRUCTURA ORGANIZACIONAL

5.6.1 Organigrama.

Ilustración 27. Organigrama



Fuente: autor.

5.6.2 Descripción de cargos.

Director ejecutivo (Ingeniero de sistemas/ Ingeniero industrial/ Administrador de empresas).

Perfil.

Persona capacitada para planear, supervisar, coordinar todas las actividades relacionadas con la administración de recursos humanos, financieros, contables y legales. Con experiencia en toma de decisiones, con el fin de garantizar el cumplimiento de los objetivos y metas propuestas por la empresa.

Director operativo (Ingeniero de sistemas).

Perfil.

Profesional con capacidad de planificar el proyecto, dimensionar recursos, controlar el avance de las actividades, comunicar resultados, liderar el diseño y la ejecución de los planes de pruebas y reportar resultados, analizar la documentación técnica, definir requerimientos, fijar objetivos de las pruebas.

Analista de pruebas (Ingeniero de sistemas).

Perfil.

Persona con conocimientos en el desarrollo de procesos de prueba, diseño y ejecución de planes de prueba y casos de prueba. Manejo de herramientas para la gestión de pruebas de software, diseño y ejecución de diferentes tipos de pruebas, manejo de herramientas para la automatización de pruebas. Con capacidad para trabajar en equipo y bajo presión.

Asesor Contable (Contador).

Perfil.

Persona con capacidad de proponer soluciones a aspectos como la liquidez, los índices de rentabilidad. Se debe encargar del registro de los movimientos contables del negocio, registro de ventas, registro de compras, tramites tributarios e informes contables.

5.7 ASPECTOS LEGALES

QualityTest S.A.S. se constituye como una Sociedad por Acciones Simplificada, conformada por dos accionistas, los cuales trabajan activamente en la empresa y limitan sus responsabilidades hasta el monto de sus aportes. La principal actividad de la empresa es prestar el servicio de pruebas a empresas desarrolladoras de la región.

Pasos para la constitución.

1. Redactar el contrato o acto unilateral constitutivo de las SAS. Este documento debe contener al menos la siguiente información de la sociedad:
 - Nombre, documento de identidad y domicilio de los accionistas
 - Razón Social, seguida de las palabras "Sociedad por Acciones Simplificada" o de las letras S.A.S.
 - El domicilio principal de la sociedad
 - El capital pagado, la clase, número y valor nominal de las acciones representativas del capital y la forma y términos en que éstas deberán pagarse
 - El nombre, documento de identidad y forma de administración. Deberá designarse al menos un representante legal.
2. Las personas que suscriben el documento de constitución deberán autenticar sus firmas antes que éste sea inscrito en el registro mercantil. La autenticación se podrá hacer directamente o a través de apoderado, en la Cámara de Comercio de la ciudad donde la sociedad establezca su domicilio.
3. El documento privado debe ser inscrito en el Registro Mercantil de la Cámara de Comercio del lugar en que la sociedad establezca su domicilio. También, se diligencia el Registro Único Empresarial (RUE), el formulario de inscripción en el RUT y se realiza el pago de la matrícula mercantil, impuesto de registro y derechos de inscripción.

Beneficios de las SAS.

- Permite fijar las reglas que van a regir el funcionamiento de la sociedad
- El proceso de constitución y reforma es más fácil y eficiente
- Las responsabilidades de los socios se limita a sus aportes
- Es posible crear diversas clases y series de acciones
- Es un buen vehículo de negocios que facilita el desarrollo de inversiones extranjeras
- No se requiere establecer una duración determinada
- El objeto social puede ser indeterminado
- El pago del capital puede diferirse hasta por dos años
- Permite los acuerdos de accionistas sobre cualquier asunto lícito
- Por regla general no se exige revisor fiscal
- Se establecen disposiciones que facilitan la operación y administración de las SAS
- Mayor flexibilidad en la regulación de los Derechos patrimoniales y políticos de los accionistas en las SAS
- Se establece un mayor tiempo para enervar la causal de disolución por perdidas
- El trámite de liquidación es más ágil
- Mayor agilidad para la solución de conflictos
- Se consagra un reproche expreso a conductas abusivas por parte de los socios de las SAS.

5.8 GASTOS DE ARRANQUE

Los costos al momento de crear la empresa varían dependiendo de los activos a declarar y de la inversión inicial presupuestada.

Tabla 8. Gastos legales

| Gastos legales | Valor |
|----------------------------------|-------------------|
| Formulario único | 3.600 |
| Registro mercantil ¹⁴ | 139.000 |
| Inscripción libro de comercio | 27.000 |
| Autenticación firmas | 10.000 |
| Total | \$ 179.600 |

¹⁴ Tarifas 2010, para trámites de registro público mercantil. www.camarapereira.org.co/

5.9 GASTOS DE PERSONAL

Para el cálculo del valor de la nómina se tuvieron en cuenta todo los porcentajes parafiscales, de seguridad social y cargas prestacionales por parte de QualityTest S.A.S. También se tuvieron en cuenta el valor del salario mínimo y el auxilio de transporte establecido para el presente año.

A continuación se muestra el valor del salario básico a pagar según su cargo.

Tabla 9. Salarios básicos

| Cargo | Salario base |
|---------------------|---------------------|
| Director operativo | 1.200.000 |
| Analista de pruebas | 1.100.000 |
| Director ejecutivo | 1.200.000 |
| Asesor contable | 200.000 |
| Asesor comercial | 600.000 |
| Total | \$ 4.300.000 |

Nota: El asesor contable tiene una vinculación a la empresa por prestación de servicios

Tabla 10. Costo total empleado por mes

| Cargo | | Director operativo | Analista de pruebas | Director ejecutivo | Asesor comercial | Asesor contable |
|--------------------|----------|---------------------|---------------------|--------------------|-------------------|-------------------|
| Sueldo básico | | 1.200.000 | 1.100.000 | 1.200.000 | 600.000 | 200.000 |
| ICBF | 3,00% | 33.000 | 36.000 | 36.000 | 18.000 | 0 |
| Comfamiliar | 4,00% | 44.000 | 48.000 | 48.000 | 24.000 | 0 |
| Sena | 2,00% | 22.000 | 24.000 | 24.000 | 12.000 | 0 |
| ARP | 0,52% | 5.720 | 6.240 | 6.240 | 3.120 | 0 |
| Pensión | 12,00% | 132.000 | 144.000 | 144.000 | 72.000 | 0 |
| Salud | 8,50% | 93.500 | 102.000 | 102.000 | 51.000 | 0 |
| Prima de servicios | 8,33% | 91.630 | 99.960 | 99.960 | 49.980 | 0 |
| Cesantías | 8,33% | 91.630 | 99.960 | 99.960 | 49.980 | 0 |
| Interés cesantías | 1,00% | 11.000 | 12.000 | 12.000 | 6.000 | 0 |
| Vacaciones | 4,17% | 45.870 | 50.040 | 50.040 | 25.020 | 0 |
| Aux. transporte | < 2SMMLV | | 0 | 0 | 61.500 | 0 |
| Total | | \$ 1.822.200 | \$ 1.670.350 | \$1.822.200 | \$ 972.600 | \$ 200.000 |

MODULO FINANCIERO

6.1 EGRESOS

6.1.1 Inversión inicial.

La inversión inicial que requiere QualityTest S.A.S. para empezar a operar se muestra en las siguientes tablas y se divide en tres categorías: herramientas operativas, herramientas administrativas y gastos legales.

Tabla 11. Herramientas técnicas

| Herramientas técnicas | Cantidad | V. unitario | V. total |
|-----------------------|----------|-------------|---------------------|
| Computador | 2 | 1.400.000 | 2.800.000 |
| Servidor | 1 | 2.500.000 | 2.500.000 |
| Switch | 1 | 150.000 | 150.000 |
| UPS | 1 | 270.000 | 270.000 |
| Licencias Windows 7 | 2 | 810.000 | 1.620.000 |
| Estabilizadores | 2 | 35.000 | 70.000 |
| Total | | | \$ 7.410.000 |

Tabla 12. Herramientas administrativas

| Herramientas administrativas | Cantidad | V. unitario | V. total |
|------------------------------|----------|-------------|---------------------|
| Computador | 2 | 1.400.00 | 2.800.000 |
| Escritorios modulares | 4 | 250.000 | 1.000.000 |
| Sillas | 5 | 70.000 | 280.000 |
| Total | | | \$ 4.080.000 |

Tabla 13. Gastos legales

| Gastos legales | Valor |
|-------------------------------|-------------------|
| Formulario único | 3.600 |
| Registro mercantil | 139.000 |
| Inscripción libro de comercio | 27.000 |
| Autenticación firmas | 10.000 |
| Total | \$ 179.600 |

Tabla 14. Costos inversión inicial

| Inversión inicial | Valor |
|------------------------------|----------------------|
| Herramientas técnicas | 7.410.000 |
| Herramientas administrativas | 4.080.000 |
| Costos legales | 179.600 |
| Total | \$ 11.669.600 |

6.1.2 Mano de obra.

Según el organigrama de QualityTest S.A.S, el costo de mano de obra que se debe pagar a cada uno de los empleados de la empresa corresponde a la siguiente nómina.

Tabla 15. Mano de obra

| Cargo | Salario base |
|---------------------|---------------------|
| Director operativo | 1.200.000 |
| Analista de pruebas | 1.100.000 |
| Director ejecutivo | 1.200.000 |
| Asesor contable | 200.000 |
| Asesor comercial | 600.000 |
| Total | \$ 4.300.000 |

Tabla 16. Costo mano de obra

| Cargo | | Director operativo | Analista de pruebas | Director ejecutivo | Asesor comercial | Asesor contable |
|--------------------|----------|---------------------------|----------------------------|---------------------------|-------------------------|------------------------|
| Sueldo básico | | 1.200.000 | 1.100.000 | 1.200.000 | 600.000 | 200.000 |
| ICBF | 3,00% | 33.000 | 36.000 | 36.000 | 18.000 | 0 |
| Comfamiliar | 4,00% | 44.000 | 48.000 | 48.000 | 24.000 | 0 |
| Sena | 2,00% | 22.000 | 24.000 | 24.000 | 12.000 | 0 |
| ARP | 0,52% | 5.720 | 6.240 | 6.240 | 3.120 | 0 |
| Pensión | 12,00% | 132.000 | 144.000 | 144.000 | 72.000 | 0 |
| Salud | 8,50% | 93.500 | 102.000 | 102.000 | 51.000 | 0 |
| Prima de servicios | 8,33% | 91.630 | 99.960 | 99.960 | 49.980 | 0 |
| Cesantías | 8,33% | 91.630 | 99.960 | 99.960 | 49.980 | 0 |
| Interés cesantías | 1,00% | 11.000 | 12.000 | 12.000 | 6.000 | 0 |
| Vacaciones | 4,17% | 45.870 | 50.040 | 50.040 | 25.020 | 0 |
| Aux. transporte | < 2SMMLV | | 0 | 0 | 61.500 | 0 |
| Total | | \$ 1.822.200 | \$ 1.670.350 | \$1.822.200 | \$ 972.600 | \$ 200.000 |

En la tabla anterior, se muestra el costo total de la nómina el cual se calculo teniendo en cuenta los pagos por concepto de parafiscales, seguridad social, prestaciones y auxilio de transporte para quienes devengan menos de dos (2) SMMLV. El salario del Auxiliar Contable no tendrá esos pagos ya que su vinculación con la empresa es por prestación de servicios. El total de la nomina mensual es de \$ 6.487.350. Teniendo en cuenta el aumento de las ventas, se contratará un analista de pruebas en el cuarto año. A continuación se observa la proyección del costo de mano de obra para los primeros cinco años.

Tabla 17. Proyección costos de mano de obra

| Año1 | Año 2 | Año3 | Año 4 | Año 5 |
|------------|------------|------------|-------------|-------------|
| 77.848.200 | 80.962.128 | 83.390.992 | 108.008.209 | 111.248.455 |

La proyección de los costos de mano de obra se hará de 4% en el 2010 y de 3% para 2011, 2012, 2013 y 2014¹⁵.

6.1.3 Costos de operación.

Tabla 18. Costos de operación

| Item | Valor/mes | AÑO 1 | AÑO 2 | AÑO 3 | AÑO 4 | AÑO 5 |
|---------------------------------------|------------------|--------------------|--------------------|---------------------|---------------------|---------------------|
| Arrendamiento | 500.000 | 6.000.000 | 6.240.000 | 6.427.200 | 6.620.016 | 6.818.616 |
| Servicios públicos | 150.000 | 1.800.000 | 1.872.000 | 1.928.160 | 1.986.005 | 2.045.585 |
| Internet | 130.000 | 1.560.000 | 1.622.400 | 1.671.072 | 1.721.204 | 1.772.840 |
| Total gastos de administración | \$780.000 | \$9.360.000 | \$9.734.400 | \$10.026.432 | \$10.327.225 | \$10.637.042 |

6.1.4 Depreciaciones.

La depreciación de los equipos de cómputo se realiza por el método de línea recta a 5 años y los muebles y enseres se deprecian a 10 años.

¹⁵ Proyección del IPC, según estudios de Inflación del Banco de la República.

Tabla 19. Depreciación

| Item | Precio de compra | AÑO 1 | AÑO 2 | AÑO 3 | AÑO 4 | AÑO 5 |
|---------------------|------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Maquinaria y equipo | 7.410.000 | 1.482.000 | 1.482.000 | 1.482.000 | 1.482.000 | 1.482.000 |
| Muebles y enceres | 1.280.000 | 128.000 | 128.000 | 128.000 | 128.000 | 128.000 |
| Equipos de oficina | 2.800.000 | 560.000 | 560.000 | 560.000 | 560.000 | 560.000 |
| Total | | \$2.170.000 | \$2.170.000 | \$2.170.000 | \$2.170.000 | \$2.170.000 |

6.2 INGRESOS POR VENTAS

Los ingresos adquiridos por QualityTest S.A.S. son por la venta del servicio de pruebas, el cual esta discriminado en ocho tipos de pruebas y cuyos precios se muestran en la siguiente tabla.

Tabla 20. Precio del servicio

| Servicio | Precio |
|---------------------------|---------------------|
| Pruebas de requerimientos | 1.500.000 |
| Pruebas de unidad | 1.500.000 |
| Pruebas de Integración | 4.000.000 |
| Pruebas funcionales | 4.000.000 |
| Pruebas de aceptación | 3.000.000 |
| Pruebas de seguridad | 3.000.000 |
| Pruebas de stress | 3.000.000 |
| Pruebas de rendimiento | 3.000.000 |
| Total | \$23.000.000 |

Tabla 21. Proyección unidades vendidas

| Servicio | Año 1 | Año 2 | Año 3 | Año 4 | Año 5 |
|---------------------------|-----------|-----------|-----------|-----------|-----------|
| Pruebas de requerimientos | 3 | 3 | 3 | 5 | 5 |
| Pruebas de diseño | 3 | 3 | 3 | 5 | 5 |
| Pruebas de integración | 3 | 3 | 5 | 5 | 6 |
| Pruebas funcionales | 4 | 4 | 5 | 6 | 6 |
| Pruebas de aceptación | 3 | 3 | 4 | 5 | 5 |
| Pruebas de seguridad | 3 | 3 | 3 | 5 | 5 |
| Pruebas de stress | 3 | 3 | 3 | 5 | 5 |
| Pruebas de rendimiento | 3 | 4 | 4 | 6 | 6 |
| Total | 25 | 26 | 30 | 42 | 43 |

A continuación se muestra la proyección de las ventas en los cinco primeros años de QualityTest S.A.S. El incremento del precio fue del 5% anual.

Tabla 22. Proyección de ventas

| Servicio | Año 1 | Año 2 | Año 3 | Año 4 | Año 5 |
|---------------------------|---------------------|---------------------|----------------------|----------------------|----------------------|
| Pruebas de requerimientos | 4.500.000 | 4.725.000 | 4.961.250 | 8.682.188 | 9.116.297 |
| Pruebas de diseño | 4.500.000 | 4.725.000 | 4.961.250 | 8.682.188 | 9.116.297 |
| Pruebas de integración | 12.000.000 | 12.600.000 | 22.050.000 | 23.152.500 | 29.172.150 |
| Pruebas funcionales | 16.000.000 | 16.800.000 | 22.050.000 | 27.783.000 | 29.172.150 |
| Pruebas de aceptación | 9.000.000 | 9.450.000 | 13.230.000 | 17.364.375 | 18.232.594 |
| Pruebas de seguridad | 9.000.000 | 9.450.000 | 9.922.500 | 17.364.375 | 18.232.594 |
| Pruebas de stress | 9.000.000 | 9.450.000 | 9.922.500 | 17.364.375 | 18.232.594 |
| Pruebas de rendimiento | 9.000.000 | 12.600.000 | 13.230.000 | 20.837.250 | 21.879.113 |
| Total | \$73.000.000 | \$79.800.000 | \$100.327.500 | \$141.230.250 | \$153.153.788 |

6.3 ESTADOS DE RESULTADOS

| | Año 1 | Año 2 | Año 3 | Año 4 | Año 5 |
|------------------------------------|--------------------|--------------------|-------------------|-------------------|-------------------|
| Ventas | 73.000.000 | 79.800.000 | 100.327.500 | 141.230.250 | 153.153.788 |
| Devoluciones y rebajas en ventas | 0 | 0 | 0 | 0 | 0 |
| Materia prima, mano de obra | 41.910.600 | 43.587.024 | 44.894.635 | 68.356.961 | 70.407.670 |
| Depreciación | 3.531.333 | 3.531.333 | 3.531.333 | 695.794 | 695.794 |
| Agotamiento | 0 | 0 | 0 | 0 | 0 |
| Otros costos | 6.420.000 | 6.676.800 | 6.877.104 | 8.937.022 | 9.205.132 |
| Utilidad bruta | 21.138.067 | 26.004.843 | 45.024.428 | 63.240.473 | 72.845.191 |
| Gasto de ventas | 11.671.200 | 12.138.048 | 12.502.189 | 12.877.255 | 13.263.573 |
| Gastos de administración | 26.606.400 | 27.670.656 | 28.500.776 | 29.355.799 | 30.236.473 |
| Provisiones | 0 | 0 | 0 | 0 | 0 |
| Amortización gastos | 35.920 | 35.920 | 35.920 | 35.920 | 0 |
| Utilidad operativa | -17.175.453 | -13.839.781 | 3.985.543 | 20.971.499 | 29.345.146 |
| Otros ingresos y egresos | 0 | 0 | 0 | 0 | 0 |
| Total corrección monetaria | 0 | 0 | 55.582 | 0 | 58.967 |
| Utilidad antes de impuestos | -17.175.453 | -13.839.781 | 3.985.543 | 21.028.749 | 29.345.146 |
| Impuestos (35%) | 122.400 | 0 | 1.355.085 | 7.149.775 | 9.977.350 |
| Utilidad neta final | -17.297.853 | -13.839.781 | 2.630.458 | 13.878.974 | 19.367.796 |

6.4 BALANCE GENERAL

| | Año 0 | Año 1 | Año 2 | Año 3 | Año 4 | Año 5 |
|---------------------------------------|-------------------|--------------------|--------------------|--------------------|-------------------|-------------------|
| Activo Corriente | | | | | | |
| Efectivo | 366.320 | -13.241.880 | -23.636.808 | -16.084.012 | 2.355.787 | 25.246.952 |
| Gastos anticipados neto | 143.680 | 107.760 | 71.840 | 35.920 | 0 | 0 |
| Total activo corriente: | 510.000 | -13.134.120 | -23.564.968 | -16.048.092 | 2.355.787 | 25.246.952 |
| Maquinaria y equipo de operación neto | 7.410.000 | 4.940.000 | 2.470.000 | 0 | 1.060.674 | 530.337 |
| Muebles y enseres neto | 1.280.000 | 1.152.000 | 1.024.000 | 896.000 | 1.105.112 | 939.655 |
| Equipo de oficina neto | 2.800.000 | 1.866.667 | 933.333 | 0 | 0 | 0 |
| Total activos fijos: | 11.490.000 | 7.958.667 | 4.427.333 | 896.000 | 2.165.786 | 1.469.992 |
| Total otros activos fijos | 0 | 0 | 0 | 0 | 0 | 0 |
| Total activo | 12.000.000 | -5.175.453 | -19.137.635 | -15.152.092 | 4.521.573 | 26.716.944 |
| Pasivo | | | | | | |
| Cuentas x pagar proveedores | 0 | 0 | 0 | 0 | 0 | 0 |
| Impuestos x pagar | 0 | 122.400 | 0 | 1.355.085 | 7.149.775 | 9.977.350 |
| Total pasivo | 0 | 122.400 | 0 | 1.355.085 | 7.149.775 | 9.977.350 |
| Patrimonio | | | | | | |
| Capital social | 12.000.000 | 12.000.000 | 12.000.000 | 12.000.000 | 12.000.000 | 12.000.000 |
| Reserva legal acumulada | 0 | 0 | 0 | 0 | 263.046 | 1.650.943 |
| Utilidades retenidas | 0 | 0 | -17.297.853 | -31.137.635 | -28.770.222 | -16.279.145 |
| Utilidades del ejercicio | 0 | -17.297.853 | -13.839.781 | 2.630.458 | 13.878.974 | 19.367.796 |
| Revalorización patrimonio | 0 | 0 | 0 | 0 | 0 | 0 |
| TOTAL PATRIMONIO | 12.000.000 | -5.297.853 | -19.137.635 | -16.507.176 | -2.628.202 | 16.739.594 |
| TOTAL PAS + PAT | 12.000.000 | -5.175.453 | -19.137.635 | -15.152.092 | 4.521.573 | 26.716.944 |

6.5 FLUJO DE CAJA

| | Año 0 | Año 1 | Año 2 | Año 3 | Año 4 | Año 5 |
|--|--------------------|--------------------|--------------------|--------------------|--------------------|-------------------|
| Flujo de Caja Operativo | | | | | | |
| Utilidad operacional | | -17.175.453 | -13.839.781 | 3.985.543 | 20.971.499 | 29.345.146 |
| Depreciaciones | | 3.531.333 | 3.531.333 | 3.531.333 | 695.794 | 695.794 |
| Amortización gastos | | 35.920 | 35.920 | 35.920 | 35.920 | 0 |
| Impuestos | | 0 | -122.400 | 0 | -1.355.085 | -7.149.775 |
| Neto flujo de caja operativo | | -13.608.200 | -10.394.928 | 7.552.796 | 20.348.129 | 22.891.165 |
| Inversión activos fijos | -11.490.000 | 0 | 0 | 0 | -1.908.330 | 0 |
| Neto flujo de caja inversión | -11.490.000 | 0 | 0 | 0 | -1.908.330 | 0 |
| Flujo de caja financiamiento | | | | | | |
| Capital | 12.000.000 | 0 | 0 | 0 | 0 | 0 |
| Neto flujo de caja financiamiento | 12.000.000 | 0 | 0 | 0 | 0 | 0 |
| Neto periodo | -12.000.000 | -13.608.200 | -10.394.928 | 7.552.796 | 18.439.799 | 22.891.165 |
| Saldo anterior | | 366.320 | -13.241.880 | -23.636.808 | -16.084.012 | 2.355.787 |
| Saldo siguiente | -12.000.000 | -13.241.880 | -23.636.808 | -16.084.012 | 2.355.787 | 25.246.952 |

| | |
|-----|-----------------|
| VPN | \$ 6.469.028,15 |
| TIR | 10% |

En el cuarto año es necesario invertir en equipos de cómputo y muebles de oficina, debido a la contratación del nuevo ingeniero.

MODULO IMPACTO DEL PROYECTO

IMPACTO DEL PROYECTO

Las empresas desarrolladoras tendrán la posibilidad de contar con mecanismos que ayuden a que sus proyectos mejoren la calidad, mejorando el proceso de pruebas de software, para así reducir el número de errores que se encuentran en las etapas avanzadas del ciclo de vida del desarrollo.

QualityTest S.A.S. genera inicialmente cinco empleos directos y en el transcurso de sus actividades incrementará este número, al requerir personal calificado para la prestación del servicio de pruebas de software.

Al estar la empresa QualityTest S.A.S. legalmente constituida genera el pago de impuestos y salarios, asegurando el bienestar de sus empleados y generando recursos para la región.

CONCLUSIONES

- Se realizó un marco conceptual donde se definieron los principales conceptos relacionados con pruebas de software.
- Al realizar el análisis del mercado, se detectaron las empresas de la ciudad de Pereira que requieren servicios relacionados con pruebas de software, y la ausencia de empresas que prestan este servicio en la región.
- Se planteó el proceso para la prestación del servicio de pruebas, determinando los requerimientos tecnológicos y humanos necesarios.
- Se estableció la estructura organizacional y legal necesaria para funcionamiento de la empresa.
- Al evaluar la inversión inicial, los costos y las obligaciones financieras de la empresa QualityTest S.A.S, se obtuvo Tasa de Retorno Interna (TIR) de 10% y un Valor Presente Neto (VPN) de \$6.469.028, datos que demuestran la viabilidad del proyecto.
- Finalmente se concluye que la puesta en marcha de la empresa QualityTest S.A.S, dadas las características expuestas en este documento, es una solución factible a los problemas de calidad que presentan las empresas desarrolladoras de la ciudad de Pereira.

GLOSARIO

ABES: Asociación Brasileira de Ingeniería sanitaria y ambiental

ARTEFACTO: cualquier obra manual realizada con un propósito o función técnica específica. Es sinónimo de aparato y de máquina

CEPREDE: Centro de Predicción Económica

CESSI: Cámara de Empresas de Software y Servicios Informáticos

CLUSTER: grupo de compañías y asociaciones interconectadas, las cuales están geográficamente cerca, se desempeñan en un sector de industria similar, y están unidas por una serie de características comunes y complementarias

CSIC: Consejo Superior de Investigaciones Científicas

JDBC (Java Database Connectivity): la conectividad de la base de datos de Java

JMS (Java Message Service): servicio de mensajes Java

LDAP (Lightweight Directory Access Protocol): protocolo ligero de acceso a directorios

PROBADOR: persona que realiza las pruebas de software. En inglés “tester”

PROSOFT: Programa para el Desarrollo de la Industria del Software

REFACTORIZACIÓN: modificar la estructura interna del software con el objeto de que sea más fácil de entender y de modificar a futuro, tal que el comportamiento observable del software al ejecutarse no se vea afectado

REQUERIMIENTOS: son las descripciones que hace el usuario de los deseos o necesidades que tiene frente a un producto a los ingenieros o desarrolladores de software

REQUISITOS: lo que se necesita para cumplir las peticiones del usuario

WEB SERVICE: conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones

WITSA: World Information Technology and Services Alliance

BIBLIOGRAFIA

- PRESSMAN, Roger S. Ingeniería del software: un enfoque práctico, 5ª Edición. McGraw Hill, 2002.
- ARORA, A. y GAMBARDELLA, A. The rise and growth of the Software industry in Brazil, China, India, Ireland and Israel. Oxford University Press, 2005
- HOCH D, Roeding C, Purkert G y Lindner S. Secrets of Software Success. Managements Insights from 100 software firms around the world. Harvard Business School, Boston 1999.
- MAÑAS, José A. Prueba de programas [en línea]. <<http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>> [consultada enero 2010].
- IBM. Rational performance tester [en línea]. <<http://www-142.ibm.com/software/products/es/es/performance>> [consultada enero 2010].
- APACHE SOFTWARE FOUNDATION. Apache JMeter [en línea]. <<http://jakarta.apache.org/jmeter/>> [consultada enero 2010].
- QTRAQ SOFTWARE. 3.2.15.9 QaTraq profesional test management [en línea]. <<http://www.testmanagement.com/>> [consultada enero 2010].
- BUSTOS, Ligia Stella. Caracterización de las empresas del sector de las TICS (Tecnología de la Información y la Comunicación) en Risaralda, 2007. Pag. 16.
- IBM. Rational robot [en línea]. <<http://www-01.ibm.com/software/awdtools/tester/robot/>> [consultada enero 2010].

- APPLICATION LIFECYCLE SOLUTIONS. JTest [en línea]. <<http://www.als-es.com/home.php?location=herramientas/entorno-desarrollo/jtest>> [consultada enero 2010].
- APPLICATION LIFECYCLE SOLUTIONS. QADirector [en línea]. <<http://www.als-es.com/home.php?location=herramientas/entorno-pruebas/qadirector>> [consultada enero 2010].
- MANTIS. Mantis bug tracker system [en línea] <<http://www.mantisbt.org/>> [consultada enero 2010]
- LOPEZ, A. Estudios de competitividad sistémica, componente B: la sociedad de la información, servicios informáticos, servicios de alto valor agregado y software. CENIT. Argentina, 2003
- GLENFORD J. Myers. The art of software testing, 5ta Edición. John Wiley & Sons, Inc., Hoboken, New Jersey
- LERMA, Héctor Daniel. Presentación de informes: el documento final de investigación, 2da Edición. ECCOE EDICIONES. 2005
- ICONTEC. Tesis y otros trabajos de grado, edición actualizada. 2005 – 2006
- FITNESSSE. fitnessse [en línea] <<http://fitnessse.org/>> [consultada abril 2010]
- NOLA COMPUTER SERVICES. Avignon [en línea] <<http://www.nolacom.com/avignon/index.asp>> [consultada abril 2010].
- HTTPUNIT. Httpunit [en línea] <<http://httpunit.sourceforge.net/index.html>> [consultada abril 2010].
- TYTO SOFTWARE. Sahi. [en línea] <<http://sahi.co.in/w/>> [consultada abril 2010].

ANEXOS

ANEXO A

Informe de viabilidad y alcance.

| |
|--|
| Informe de viabilidad y alcance |
| Resumen de la documentación analizada |
| <i>Contenido de la documentación recibida</i> |
| Viabilidad de las funcionalidades a probar |
| <i>Viabilidad de las diferentes pruebas</i> |
| Alcance de las pruebas |
| <i>Tipos de prueba a realizar según su funcionalidad</i> |
| Definición entorno de pruebas |
| <i>Recursos hardware y software necesarios</i> |
| Roles y responsabilidades |
| <i>Diferentes roles existentes y sus responsabilidades</i> |

ANEXO B

Informe de planificación.

| Informe de planificación |
|---|
| Actividades |
| <i>Detalle de todas las actividades a realizar en las siguientes etapas del proceso de pruebas.</i> |
| Alcance de las pruebas |
| <i>Que se va a probar y tipo de pruebas a realizar.</i> |
| Definición entorno de pruebas |
| <i>A imagen del entorno final de producción.</i> |
| Estimaciones de tiempo |
| <i>El tiempo para realizar todas las actividades y responsabilidades.</i> |
| Recursos |
| <i>Para ejecutar cada una de las actividades planificadas.</i> |
| Herramienta gestión de pruebas |
| <i>Definición de herramienta para gestión de pruebas</i> |

ANEXO C

Plan de pruebas.

| Plan de pruebas | |
|--|----------|
| Identificador | |
| Alcance | |
| <i>Estructura de la prueba</i> | |
| Estrategia | |
| | |
| Prioridades | |
| <i>Por características de prioridades o por prioridad de casos de uso</i> | |
| Criterios de aceptación / fallo | |
| <i>Porcentaje mínimo de éxito de caso de prueba</i> | |
| Criterios de suspensión | |
| <i>Problemas y/o limitaciones que puedan darse a lo largo del proyecto</i> | |
| Complejidad | |
| <i>Alta, media y baja</i> | |
| Funciones y responsabilidades | |
| | |
| Recursos necesarios | |
| Entorno | Personal |

| | |
|---|--|
| | |
| Métricas | |
| <i>Se aplican a lo largo del ciclo de vida del proyecto</i> | |
| Herramientas de gestión | |
| | |
| Aprobaciones | |
| | |

ANEXO D

Casos de prueba.

| Caso de prueba | | |
|--|--|-----------------------------|
| Identificador | | |
| Responsable | | |
| Elementos de Software | | |
| <i>Definir los elementos que se van a probar y las características que ejercerá este caso.</i> | | |
| Necesidades de entorno | | |
| <i>Hardware, software, personal</i> | | |
| Entradas | Salidas | Resultados |
| <i>Especificaciones de cada entrada requerida para ejecutar cada caso de prueba, incluyendo las relaciones entre las diversas entradas</i> | <i>Especificaciones de todas las salidas y las características requeridas para los elementos que se van a probar</i> | <i>Resultados esperados</i> |
| Observaciones | | |
| | | |

ANEXO E

Matriz de trazabilidad.

| Caso de prueba Caso de uso | ID caso de prueba 1 | ID caso de prueba 2 | ID caso de prueba 3 | ... |
|---|---------------------|---------------------|---------------------|-----|
| Id Caso de uso 1 | | | | |
| Id Caso de uso 2 | | | | |
| Id Caso de uso3 | | | | |
| Id Caso de uso 4 | | | | |
| . | | | | |
| . | | | | |
| . | | | | |

ANEXO F

Informe de incidencias.

| Informe de incidencias | |
|--|--|
| Identificador | |
| Resumen del incidencias. | |
| | |
| Descripción de datos objetivos | |
| <i>Fecha/hora, entradas, resultados.</i> | |
| Impacto que tendrá sobre las pruebas | |
| | |

ANEXO G

Histórico de pruebas.

| Histórico de pruebas | |
|--|--|
| Identificador | |
| Descripción de la prueba | |
| | |
| Anotación de datos sobre cada hecho ocurrido | |
| <ul style="list-style-type: none">• <i>Fecha y hora</i>• <i>Identificador de informe de incidente</i> | |
| Pasos del Procedimiento | |
| <i>Resultados e Incidente</i> | |
| Secuencia de acciones | |
| | |
| Observaciones | |
| <i>Para iniciar la ejecución y durante la ejecución</i> | |

ANEXO H

Informe final de pruebas.

| Informe final de pruebas | |
|--|--|
| Identificador | |
| Resumen de la evaluación de los elementos probados | |
| | |
| Variaciones del software | |
| <i>Variaciones con respecto a su especificación de diseño, así como las variaciones en las pruebas</i> | |
| Valoración de la extensión de la prueba | |
| <i>Cobertura lógica, funcional, de requisitos.</i> | |
| Resumen de los resultados obtenidos. | |
| | |
| Evaluación de cada elemento software sometido a prueba | |
| <i>Evaluación general del software incluyendo las limitaciones del mismo</i> | |
| Aprobaciones | |
| | |

ANEXO I

Ficha técnica de la encuesta

Título: caracterización de las empresas desarrolladoras de software de Pereira año 2009.

Realizado por: Paola Ramírez A.
Carolina Ramírez A.

Tipo de encuesta: censo a empresas desarrolladoras de software de la ciudad de Pereira.

Método de recolección de datos: entrevista personal.

Universo de representación: empresas que se dedican al desarrollo de software.

Objetivo: Conocer las características relacionadas con aseguramiento de calidad de las empresas desarrolladoras de software de la ciudad de Pereira.

Tamaño de la muestra:

| POBLACIÓN TOTAL | | |
|----------------------|-------------------------|---|
| 23 EMPRESAS | | |
| EMPRESAS ENCUESTADAS | EMPRESAS NO ENCUESTADAS | |
| 20 | No atendió | 2 |
| | Imposible de localizar | 1 |
| | | 3 |

Fecha de realización: septiembre - octubre de 2009