

**Universidad de las Ciencias Informáticas**

**Facultad 7**



**Título: Propuesta de actividades del proceso de aseguramiento de la calidad del flujo de trabajo de pruebas del área temática APS**

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

**Autoras:** Darlenys Guzmán Ruiz  
Yaira Caridad Lima Ojeda

**Tutores:** Lic. Lucy Cruz Águila  
Ing. Yoan Manuel Cabrerías Arribas

**“Ciudad de la Habana, Julio, 2008”**

**“Año 50 de la Revolución”**

## DECLARACIÓN DE AUTORÍA

Declaramos ser autoras de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 10 días del mes de julio del año 2008.

---

Darlenys Guzmán Ruiz.

Firma

---

Yaira Caridad Lima Ojeda.

Firma

---

Lic. Claribel Lucy Cruz Águila.

Firma

---

Ing. Yoan Manuel Cabrera Arribas.

Firma

*"Cuando se es joven, se crea. Cuando se es inteligente,  
se produce. No se adapta, se innova: la medianía  
copia; la originalidad se atreve".*

*José Martí*

## DATOS DE CONTACTO

**Tutor:** Lic. Lucy Cruz Águila

Graduada de Licenciatura en Cibernética-Matemática en el año 1988.

Ha desarrollado diferentes proyectos de gestión en la esfera de la salud, turismo, empresarial, ministerial, ha dirigido técnicamente proyectos de gestión en estas áreas.

Ha participado en diferentes eventos científicos-técnicos en temas de calidad del desarrollo de software.

Ha recibido diferentes cursos de post-gradados en ingeniería de software, programación y calidad.

Se desarrolló como especialista de la dirección de Desarrollo de la Empresa SOFTEL.

En la actualidad se desempeña como vicepresidenta de la empresa DESOFT.

Correo electrónico: [lucy.cruz@desoft.cu](mailto:lucy.cruz@desoft.cu)

Ubicación: DESOFT, Cuba

**Tutor.** Ing. Yoan Manuel Cabrera Arribas

Graduado de Ingeniero en Ciencias Informáticas, Profesor Recién graduado de Práctica Profesional de la Facultad 7, es el líder de proyecto Registro de Actividades Diarias. Ha presentado trabajo en el Evento Uciencia 2007 y en la Serie Interna de la UCI.

Correo electrónico: [ycabreraar@uci.cu](mailto:ycabreraar@uci.cu)

Ubicación: UCI, Cuba

## AGRADECIMIENTOS

*A todas las personas que, de una forma u otra, nos apoyaron en la realización de nuestro Trabajo de Diploma.*

*A nuestro Comandante en Jefe y a la Revolución, por permitirnos soñar y hacer realidad nuestros sueños de estudiar y convertirnos en profesionales comprometidos.*

*A la Universidad, por poner a nuestro alcance tantos recursos para nuestra formación y darnos la posibilidad de crecer como personas y formar parte de este gran "Ejército de Luz".*

*A nuestras compañeras de estudio más cercanas: Daylen, Yaney, Maybelis, Mayrenis, Ayme, Ariuska, Lisandra, Orisbel, Yakelin y Yenisbel; convertidas a lo largo de tantos años en amigas y hermanas para nosotras, gracias por comprender todas nuestras diferencias, aceptarnos en sus vidas y permitirnos grabarlas para siempre en nuestros corazones.*

*A nuestros tutores Lucy y Yoan por su apoyo durante toda esta etapa.*

*A nuestras familias, especialmente a nuestros padres, por la fe depositada en nosotras y por ser nuestra fuerza e inspiración para lograr cualquier cosa que nos propongamos.*

## DEDICATORIA

*A mis padres, en especial a mi mamá Clara y a mi abuela Teresa, las personas que más amo en el mundo, gracias a ustedes me he convertido en lo que soy, gracias por todo.*

*A mis hermanos, Raúl, Ernesto, Leidy y en especial Daniela, quien estoy segura que será capaz de alcanzar todo lo que yo he logrado y mucho más.*

*A mi novio, José Antonio, por su paciencia y su apoyo cuando más lo necesitaba.*

*A mis tíos, Maiquel, Rafael, Carlos .E, Cristina, Raiza y Karenia, quienes han sabido ayudarme todos estos años.*

*A mis amigas de la Universidad, Daylen, Yaney, Mayrenis, Maybelis, Yasnalla y Ayme, quienes han sido incondicionales todos estos años y a quienes recordaré por siempre.*

*A mi compañera de tesis, Yaira, por su apoyo en la realización del presente trabajo.*

*A todas las personas que me apoyaron, dedico este trabajo.*

*Darlenys Guzmán Ruíz.*

*En especial dedico esta tesis a mis padres, a mi mamá María y a mi papá Roberto, quienes me han guiado por el camino correcto y quienes me han dado todo el apoyo del mundo para estudiar en esta universidad y que han sacrificado todo por verme hecha lo que hoy soy, gracias por ser lo más grande para mí.*

*A mi hermana Rosi, y a mi cuñado Danler, que es mi otro hermano, a ellos que han sido incondicionales conmigo también les dedico esta tesis.*

*A mi novio, Raciél, por su paciencia y su apoyo cuando más lo necesitaba.*

*A mi familia, en general, por confiar y siempre esperar lo mejor de mí.*

*A mis amigas Orisbel, Yaquelín, Yenisbel, Ariuska y Lisandra por acompañarme durante estos años de carrera, y apoyarme en todo.*

*A mi compañera de tesis, Darlenys, por el buen trabajo en equipo y su plena disposición para la realización de esta tesis.*

*A nuestro Comandante en jefe Fidel Castro, por poner en nuestras manos una universidad de excelencia que nos ha forjado en formarnos como buenos ingenieros y con las mejores condiciones para formar parte de jóvenes del futuro.*

*Yaira Caridad Lima Ojeda.*

## RESUMEN

La presente investigación tiene como objetivo proponer un procedimiento que guíe el proceso de aseguramiento de la calidad del flujo de prueba de las aplicaciones desarrolladas en el área temática APS.

Para el desarrollo del procedimiento se hizo necesario el análisis de diferentes modelos de calidad, como es CMMI, las normas ISO conjuntamente con las buenas prácticas de PSP y TSP. Así como, un estudio del flujo de trabajo de prueba utilizado en el proyecto. Asimismo se analizó el estado de la calidad en el área temática, fundamentalmente en este flujo de trabajo, lográndose la definición de una propuesta de procedimiento que cumple con los objetivos para los cuales fue elaborada.

La aplicación de este procedimiento, conjuntamente con el compromiso y responsabilidad de cada participante permitirán obtener un producto con la calidad requerida en APS, ofreciéndole a quienes lo van a aplicar una guía para su realización.

**PALABRAS CLAVE:** Calidad, aseguramiento de la calidad, plan de aseguramiento de la calidad, proceso de prueba, flujo de trabajo de prueba.

**TABLA DE CONTENIDO**

**INTRODUCCIÓN ..... 1**

**CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA ..... 4**

1.1 Calidad de Software ..... 4

    1.1.1 Concepto de Calidad de Software ..... 4

    1.1.2 Aseguramiento de la Calidad ..... 4

    1.1.3 Plan de Calidad ..... 4

    1.1.4 Gestión de la Calidad ..... 5

    1.1.5 Política de Calidad ..... 5

    1.1.6 Control de la Calidad ..... 5

    1.1.7 Garantía de Calidad ..... 5

    1.1.8 Auditoría de Calidad ..... 6

    1.1.9 Factores que determinan la calidad del software ..... 6

1.2 PSP (Personal Software Process) ..... 7

1.3 TSP (Team Software Process) ..... 9

1.4 CMMI (Modelo de Madurez y Capacidad Integrado) ..... 10

    1.4.1 Áreas de procesos de CMMI ..... 12

    1.4.2 Áreas de proceso en relación a la investigación ..... 13

1.5 ISO (Organización Internacional de Normalización) ..... 15

    1.5.1 ISO 15504(SPICE, Software Process Improvement and Capability Determination) ..... 15

    1.5.2 ISO 9001 ..... 15

    1.5.3 ISO 9000 3 ..... 16

    1.5.4 ISO 12207 ..... 18

    1.5.5 ISO 9126 ..... 22

**CAPÍTULO 2 ESTADO DEL ARTE ..... 23**

2.1 Flujo de Trabajo de Prueba ..... 23

    2.1.1 Actividades del Flujo de trabajo de Prueba ..... 24

    2.1.2 Trabajadores ..... 27

2.2 Estado actual de las pruebas y la calidad en el proyecto ..... 28

**CAPÍTULO 3 PROPUESTA DE PROCEDIMIENTO Y HERRAMIENTAS ..... 33**

3.1 Procedimiento para el Aseguramiento de la calidad en el flujo de trabajo de prueba ..... 33

    3.1.1 Objetivos ..... 34



3.1.2 Alcance.....	34
3.1.3 Definiciones.....	34
3.1.4 Descripción.....	34
3.2 Herramientas para la administración y ejecución de las pruebas.....	48
3.2.1 JMeter .....	48
3.2.2 Rational Robot.....	55
<b>CONCLUSIONES .....</b>	<b>61</b>
<b>RECOMENDACIONES .....</b>	<b>62</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>63</b>
<b>BIBLIOGRAFÍA .....</b>	<b>65</b>
<b>ANEXOS.....</b>	<b>68</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>107</b>

## INTRODUCCIÓN

La informatización de la sociedad se define en Cuba como el proceso de utilización ordenada y masiva de las tecnologías de la información y las comunicaciones (TIC) para satisfacer las necesidades de información y conocimiento de todas las personas y esferas de la sociedad, como la cultura, la educación, el turismo y uno de los más importantes la salud entre otros.

Este proceso busca lograr más eficacia, eficiencia y hacer más sustentable el aumento sistemático de la calidad de vida de los ciudadanos. La informatización se basa en los principios y valores fundamentales del sistema social, con una marcada tendencia a defender las conquistas que se han logrado. [1]

Para mayor desarrollo de la informatización en las diferentes esferas sociales, surge la Industria Cubana del Software (ICSW), llamada a convertirse en una significativa fuente de ingreso nacional, como resultado del correcto aprovechamiento de las ventajas del considerable capital humano disponible en el país. Debido a la insuficiencia de informáticos, con conocimientos estratégicos para el proceso de informatización de la sociedad cubana se impone la necesidad de que las universidades garanticen la formación de profesionales con habilidades básicas para el desarrollo de software. Así es como surge la Universidad de las Ciencias Informáticas, con el objetivo de que los estudiantes tuvieran la mayor participación en esta industria.

La Universidad de las Ciencias Informáticas (UCI) surge al calor de la batalla de ideas, con la visión de realizar producciones intelectuales que serán el sustento fundamental de Cuba, la idea es convertir la informática en una de las ramas más productivas y contribuidoras de recursos para la nación cubana. Y como bien dice la Dra. Alina Ruiz Jhones Vicerrectora Primera de la UCI, en III Taller de Calidad, en el marco de la Informática 2007 expresa “Esta es una industria de jóvenes, una tecnología de jóvenes quienes tienen todos los recursos y le dedican todas sus energías al desarrollo de los proyectos”. [2]

Una de las ramas más importantes para ser informatizada, es la salud, existe mucha información relacionada con los pacientes y población en general que necesita mantenerse localizada. Por esto se hace necesaria la informatización de la Atención Primaria de la Salud (APS) viéndola como la aplicación gradual e integral de las nuevas tecnologías de la información y las comunicaciones en la gerencia de los procesos relacionados con las funciones del Sistema Nacional de Salud a este nivel de atención.

Una propiedad importante de los productos informáticos es la calidad. “Se tiene que calidad no es más, que es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”. [3]

El desarrollo de software de alta calidad está determinado por factores muy importantes, gracias a eso, los productos salen con la eficacia requerida. En dichos factores se debe estar seguros de la real necesidad del cliente, las cuales pueden ser modificadas, probadas y adaptables a cualquier entorno de trabajo.

Para la obtención de un buen producto se hace necesario poner en práctica el aseguramiento de la calidad, esta no es más que aquellas acciones que hacen que un producto o servicio cumpla con unos determinados requisitos de calidad. Si estos requisitos de calidad reflejan completamente las necesidades de los clientes se podrá decir que se cumple el aseguramiento de la calidad.

La función de aseguramiento de la calidad del software se debe basar en un planificado y sistemático diseño de acciones y métodos, requeridos para garantizar la calidad del mismo. Viendo el aseguramiento de la calidad dentro de la ingeniería de software es importante analizar dentro de los flujos trabajos de la ingeniería el flujo de trabajo de prueba, uno de los flujos más importantes pues de eso depende la calidad del producto final, una prueba de software no es más que ejecutar el software con determinados datos de entrada y producir resultados que luego serán comparados con los teóricos. [4]

Durante las investigaciones realizadas y encuestas individuales hechas al personal del área temática APS se obtuvieron resultados como: no está definido correctamente un plan de calidad que ayude a controlar las tareas del flujo de trabajo de prueba, un aseguramiento de la calidad, una gestión de calidad, políticas de calidad y un control de la calidad para dicho proceso, por tanto el flujo de trabajo de prueba no se está llevando a cabo como está establecido y con una correcta documentación de los resultados de los artefactos analizados, causando la no obtención de productos con la calidad requerida.

Para que las pruebas tengan el éxito requerido, todo el esfuerzo estará dirigido a resolver el siguiente **problema científico**: ¿Cómo minimizar las limitaciones del aseguramiento de la calidad en el flujo de trabajo de prueba, en el área temática Atención Primaria de la Salud (APS)?.

Para lograr una solución a este problema se ha trazado el siguiente **objetivo**: Proponer un procedimiento que guíe el proceso de aseguramiento de la calidad del flujo de prueba de las aplicaciones desarrolladas en el área temática APS.

Se identificó como **objeto de estudio**: Proceso de aseguramiento de la calidad de software para aplicaciones de Salud en el área temática de APS. Como **campo de acción** se tiene: Proceso de desarrollo de software en el flujo de trabajo de prueba.

Para darle cumplimiento al objetivo general del trabajo se proponen las siguientes **tareas**:

- Analizar flujo de trabajo de pruebas utilizado en el área temática de APS, así como las buenas y malas prácticas de este flujo relacionadas con el aseguramiento de la calidad.
- Investigar sobre PSP y TSP vinculado con el aseguramiento de la calidad del proceso de pruebas del desarrollo de software, así como atributos de calidad de productos de software.
- Conocer de las actividades de aseguramiento de la calidad de la ingeniería de software y de área de proceso de aseguramiento de la calidad del modelo CMMI.
- Analizar las normas de calidad del producto y del proceso de pruebas definidas por la Universidad.
- Realizar encuestas y entrevistas sobre temas de aseguramiento de la calidad del flujo de prueba.
- Definir una propuesta de qué actividades de aseguramiento de la calidad deben desarrollarse en el flujo de trabajo de prueba para los proyectos del área temática de APS.
- Mostrar una propuesta de qué herramientas y cómo deben ejecutarse en el proceso de prueba.

El trabajo está estructurado en 3 capítulos:

## Capítulo I

Se refiere a conceptos fundamentales de acuerdo a las problemáticas planteadas, haciendo un estudio de los principales aspectos relacionados al aseguramiento de la calidad. Además del estudio de modelos de calidad como CMMI, y las normas ISO, PSP y TSP haciendo referencia al uso de las mejores prácticas de estos para lograr una correcta propuesta del aseguramiento de la calidad para el flujo de prueba del área temática APS.

## Capítulo II

Muestra las características y especificaciones del flujo de trabajo de prueba llevado a cabo en el área temática de APS, haciendo además un análisis de la situación actual de este proceso, así como del estado del aseguramiento de la calidad en el desarrollo de aplicaciones del área temática APS.

## Capítulo III

Se dispone de una propuesta de actividades del proceso de aseguramiento de la calidad del flujo de trabajo de prueba, generalizables en el área temática APS, llevándose a cabo la descripción de estas actividades, mediante métodos, procesos, herramientas y basándose en un enfoque de calidad.

# CAPÍTULO

## FUNDAMENTACIÓN TEÓRICA

### 1.1 Calidad de Software

#### 1.1.1 Concepto de Calidad de Software

La calidad de software es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente. [5]

En otras bibliografías la calidad de software se define como “El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”. [6]

En general, la calidad debe ser mensurable, predecible y los factores deben ser: ausencia de defectos, satisfacción del usuario y conformidad con los requerimientos.

#### 1.1.2 Aseguramiento de la Calidad

Son todas aquellas acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisface los requerimientos de calidad establecidos. Esto se resume en la realización de pruebas formales o metodológicas e incorporar un sistema de calidad dentro del proceso de desarrollo del software. [7]

Para ser efectivo, el aseguramiento de la calidad requiere, generalmente, una evaluación permanente de aquellos factores que influyen en la adecuación del diseño y de las especificaciones según las aplicaciones previstas, así como también verificaciones y auditorías a las operaciones de producción, instalación e inspección. Dentro de una organización, el aseguramiento de la calidad sirve como una herramienta de la gestión de la calidad.

El objetivo del aseguramiento de la calidad es proporcionar la gestión para informar de los datos necesarios sobre la calidad del producto, adquiriéndose una visión más profunda y segura de que la calidad del producto está cumpliendo sus objetivos.

#### 1.1.3 Plan de Calidad

Es un documento que establece las prácticas específicas de calidad, recursos y secuencia de actividades relativas a un producto, servicio, contrato o proyecto en particular. [8]

RUP ha definido un Plan de Aseguramiento de la Calidad, el cual es un artefacto que provee una vista clara de cómo la calidad del producto, artefactos y procesos serán aseguradas. Contiene un plan de revisión y auditoría y un número de referencias de otros artefactos desarrollados durante la fase de inicio.

### **1.1.4 Gestión de la Calidad**

Conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades y se implanta por medios tales como la planificación de la calidad, el control de la calidad, el aseguramiento (garantía) de la calidad y la mejora de la calidad, en el marco del sistema de calidad. [9]

Es aquel aspecto de función general de la gestión de una organización que define y aplica la política de calidad. La gestión de la calidad incluye la planificación, asignaciones de recursos y otras actividades sistemáticas, tales como los planes de calidad. [10]

### **1.1.5 Política de Calidad**

Se define como política de calidad a las Intenciones globales y orientación de una organización relativas a la calidad tal como se expresan formalmente por la alta dirección. [11]

Se define además como orientaciones y objetivos generales de una organización en relación con la calidad, expresadas formalmente por la dirección superior. [12]

### **1.1.6 Control de la Calidad**

Es una serie de inspecciones, revisiones y pruebas utilizadas a lo largo del proceso del software para asegurar que cada producto cumple con los requisitos que le han sido asignados. El control de calidad incluye un bucle de realimentación del proceso que creó el producto. La combinación de medición y realimentación permite afinar el proceso cuando los productos de trabajo creados fallan al cumplir sus especificaciones. Este enfoque ve el control de calidad como parte del proceso de fabricación. [13]

Técnicas y actividades de carácter operativo utilizadas para satisfacer los requisitos relativos a la calidad, centradas en dos objetivos fundamentales: mantener bajo control un proceso y eliminar las causas de defectos en las diferentes fases del ciclo de vida. [14]

### **1.1.7 Garantía de Calidad**

Se especializa en la aplicación de procesos de calidad a lo largo del proyecto de software. Consiste en la auditoría y las funciones de información de la gestión. Por supuesto, si los datos proporcionados mediante la garantía de calidad identifican problemas, es responsabilidad de la gestión afrontar los problemas y aplicar los recursos necesarios para resolver aspectos de calidad.

Su misión no se limita a actividades de verificación, sino que además asume un rol de liderazgo en la gestión de la calidad durante el proceso de creación y diseño del producto. La garantía de calidad (QA) acentúa la prevención de defectos y la adición de la calidad a través del ciclo de vida del desarrollo del software.

La garantía de calidad se asegura de lo siguiente: [15]

- Se usa la metodología de desarrollo apropiada.
- Las actividades de desarrollo han sido debidamente planeadas.
- Se han definido estándares y procedimientos para el proyecto.
- El personal ha sido debidamente entrenado en los procesos de calidad aplicables.
- Se llevan a cabo regularmente revisiones y auditorías independientes.
- El desarrollo es documentado adecuadamente para facilitar la mantención y la reutilización.
- La documentación se produce oportunamente y no después que el desarrollo ha sido completado.
- Los cambios introducidos han sido debidamente controlados.
- Las pruebas efectuadas son eficaces para detectar defectos, especialmente en aquellas áreas de mayor riesgo.
- Las actividades se llevan a cabo de acuerdo a los plazos y en los términos planeados.
- Las desviaciones a los estándares se identifican rápidamente.
- El proyecto está en condiciones para ser sometido a auditorías externas, si corresponde.
- La calidad es verificada con respecto a criterios preestablecidos.
- La gerencia es oportunamente informada de problemas y riesgos relativos a la calidad.
- Los problemas de calidad se analizan y las causas se comunican al proyecto para tomar medidas preventivas que eviten su repetición.

### 1.1.8 Auditoría de Calidad

Es un examen sistemático e independiente para determinar si las actividades y los resultados relativos a la calidad cumplen con las disposiciones previamente establecidas, si estas se han aplicado efectivamente y son adecuadas para lograr los objetivos. Existe auditoría de calidad del proceso, del producto y de los servicios. Uno de los propósitos de una auditoría de calidad es evaluar la necesidad de acciones correctivas o mejoramiento. [16]

### 1.1.9 Factores que determinan la calidad del software [17]

#### ✓ Operaciones del producto: características operativas

- Corrección: ¿Hace lo que se le pide? : Grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente.

- **Fiabilidad:** ¿Lo hace de forma fiable todo el tiempo?: Grado que se puede esperar de que una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida.
- **Eficiencia:** ¿Qué recursos hardware y software necesito?: Cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.
- **Integridad:** ¿Se puede controlar su uso?: Grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.
- **Facilidad de uso:** ¿Es fácil y cómodo de manejar?: Esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.

### ✓ **Revisión del producto: capacidad para soportar cambios**

- **Facilidad de mantenimiento:** ¿Se puede localizar los fallos?: Esfuerzo requerido para localizar y reparar errores.
- **Flexibilidad:** ¿Se puede añadir nuevas opciones?: Esfuerzo requerido para modificar una aplicación en funcionamiento.
- **Facilidad de prueba:** ¿Se puede probar todas las opciones?: Esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos.

### ✓ **Transición del producto: adaptabilidad a nuevos entornos**

- **Portabilidad:** ¿Se puede usar en otra máquina?: Esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo.
- **Reusabilidad:** ¿Se puede utilizar alguna parte del software en otra aplicación?: Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones.
- **Interoperabilidad:** ¿Puede comunicarse con otras aplicaciones o sistemas informáticos?: Esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos.

## **1.2 PSP (Personal Software Process)**

El Proceso Software Personal (PSP) fue diseñado para ayudar a los ingenieros del software a hacer bien su trabajo. Muestra cómo aplicar métodos avanzados de ingeniería a sus tareas diarias. Proporciona métodos detallados de planificación y estimación, muestra a los ingenieros cómo controlar su rendimiento frente a esos planes y explica como los procesos definidos guían su trabajo.

Este modelo es una continuación de la CMM (Capability Maturity Model) ya que al igual que éste, trata de demostrar que es más productivo trabajar con prácticas de ingeniería de software y también es benéfico para su mantenimiento.



La disciplina del PSP proporciona un marco de trabajo estructurado para desarrollar las habilidades personales y los métodos que necesitarás como ingeniero del software.

El PSP puede ser utilizado en:

- Desarrollo de pequeños programas.
- En pruebas de los sistemas utilizados.
- Mantenimiento de sistemas.
- Mantenimiento de software en equipos grandes.
- Verificación y validación de técnicas.

### ✓ **Defectos**

El PSP puede ayudar a hacer productos de calidad, mostrando cómo utilizar métodos efectivos de calidad. La calidad del software afecta a los costes de desarrollo, programación de entregas y la satisfacción del usuario, puesto que esta es imprescindible. Algo muy importante para garantizar que los productos poseen la calidad requerida, es que estos deben satisfacer tanto las necesidades funcionales de los usuarios, como hacer de una forma segura y consistente el trabajo de los mismos. Entonces la primera prioridad es entender los defectos que introduces y prevenirlos como puedas, pues los defectos son una de las principales causas de los problemas del software.

La calidad de un sistema de software también está regida por cualquier error trivial en los programas de soporte causando efectos devastadores. Para producir sistemas software de calidad, cada ingeniero debe aprender a hacer el trabajo con calidad.

### ✓ **Calidad del producto**

Con el PSP, los ingenieros se centran en hacer programas limpios y libres de defectos desde el principio. La razón para esta estrategia, es que una vez que el ingeniero hace un producto poco sólido, no hay herramienta que pueda dejarlo libre de defecto.

#### • **Las pruebas**

Cuanto más complejo es el producto, las pruebas consumen más tiempo y son más caras. Una de las dificultades es la de probar, aún en programas pequeños. Las pruebas son un gran problema, pues los programas no se comportan de la forma que se piensa que lo haría.

Siguiendo métodos y revisiones disciplinados, y probando los módulos en pequeñas porciones será mucho mejor encontrar defectos en los programas. Esto se debe a que nadie mejor que el que elabora el producto para conocer la lógica del programa y que haría. Si se hace todo lo que se puede para asegurar que los módulos del programa no tengan defectos, consigues eliminar en gran parte el coste y el tiempo requerido para anular los defectos posteriormente.

De esta forma al menos se puede asegurar que los productos estén esencialmente libres de defectos. Es importante tener en cuenta que una suposición es probable que introduzca un defecto.

### ✓ **Calidad del proceso**

El proceso que lleve a cabo cada ingeniero determina la calidad de los programas que se están desarrollando, determinada por la forma de trabajar. Para hacerlo más eficiente, se necesita saber cómo se está realizando el proceso individualmente y si es de la forma correcta.

Se puede determinar el rendimiento real de cada cual y como se puede cambiar para obtener mejores resultados en el futuro, midiéndose cada uno según el volumen de productos que ha desarrollado, el tiempo y los recursos requeridos para realizar el trabajo.

### ✓ **Un compromiso de calidad**

Como ingeniero de software, la calidad de los proyectos que se produzcan tendrá una importancia crítica para los jefes y clientes. Para producir un software de calidad primeramente es necesario saber la importancia de la calidad de un producto. La cuestión no es si puedes hacer un trabajo libre de defectos, sino, si se es cuidadoso para continuar haciéndolo. Así se consigue más satisfacción por su trabajo.

Como PSP plantea, está enfocado en tareas como la predicción, la verificación de calidad y productividad; las cuales están diseccionadas como procesos individuales, ya que al principio solo tiene apuntado hacia una meta, pero para que esto realmente funcione en un proceso grande, se necesita un conjunto de este tipo de actividades, lo cual lleva de la mano a la explicación de el Team Software Process (TSP). [18]

### **1.3 TSP (Team Software Process)**

El proceso del equipo de software (TSP, Team Software Process) fue diseñado para ayudar a controlar, administrar y mejorar la forma en que trabaja un equipo de software. Cuando se trabaja en equipo en el proyecto, se puede concluir mucho más rápido los productos, pues existe mayor comunicación entre los integrantes del equipo de desarrollo. Además se puede asignar

responsabilidades a una persona externa, que no sea miembro del equipo, para que revise lo existente hasta el momento, este puede identificar defectos introducidos que no están a la vista del equipo de desarrollo.

Es importante establecer los roles de los miembros del equipo, ponerse de acuerdo en las metas del equipo y funciones de cada uno de sus miembros, ayudando así a proveer un marco de trabajo para un equipo de proyecto basado en los principios de PSP. Por otra parte ayuda a las partes del grupo a ser cooperativos y a lograr un trabajo personal disciplinado, de forma tal que planifique y le dé seguimiento a su propio trabajo y sobre la base producir productos con calidad; siempre orientados a alcanzar las metas del equipo. [19]

Los siguientes puntos son objetivos que persigue el TSP:

- Generar un marco basado en PSP.
- Desarrollar productos en varios ciclos.
- Establecer estándares para medir la calidad y el comportamiento.
- Proporcionar métricas para equipos.
- Evaluar roles y equipos.
- Guías para solución de problemas en equipos.
- Resumiendo su propósito es maximizar la calidad del software y minimizar los costos.

### **1.4 CMMI (Modelo de Madurez y Capacidad Integrado)**

El CMMI es un modelo de calidad utilizado mundialmente en la industria del software. Los modelos de calidad son herramientas que guían a las organizaciones a la mejora continua y la competitividad dándoles especificaciones de qué tipo de requisitos debe implementar para poder brindar productos y servicios de alto nivel.

El Modelo de Madurez y Capacidad Integrado (CMMI), del Instituto de Ingeniería de Software (SEI), tiene el propósito de proveer una única guía unificada para la mejora de múltiples disciplinas tales como ingeniería de sistemas(SE), ingeniería de software(SW) y el desarrollo integrado del producto y el proceso(IPP). Es un marco de trabajo que describe cómo evolucionar de un proceso de desarrollo de software empírico e inmaduro a un proceso efectivo, maduro y disciplinado. [20]

Existen varias representaciones del CMMI atendiendo a diversas necesidades. La representación organizada hace especial énfasis en el grado de madurez de los procesos, mientras que la representación continua hace hincapié en la capacidad de ciertas áreas para realizar adecuadamente sus actividades.

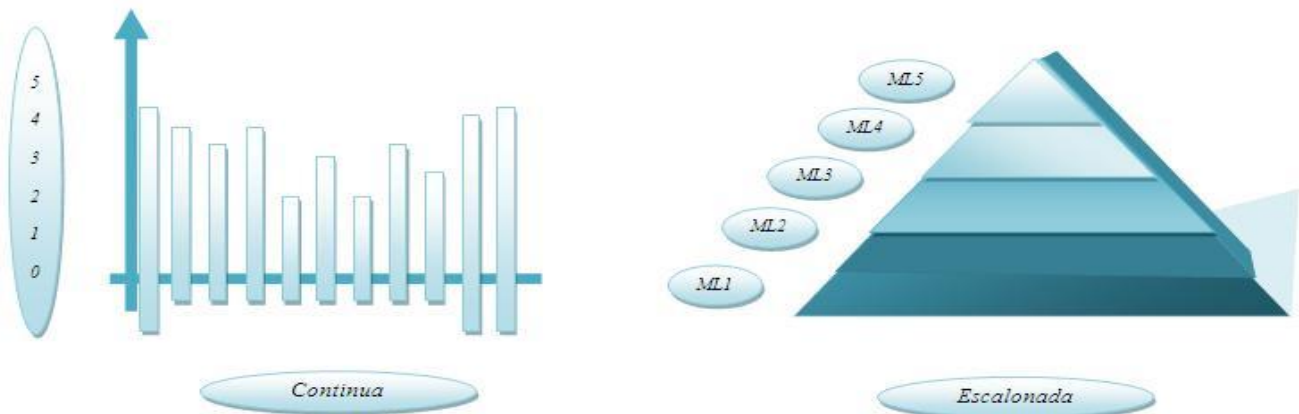


Figura 1.1 Representaciones del modelo CMMI.

El CMMI está caracterizado por 28 áreas de proceso para las cuatro disciplinas que cubre actualmente, es decir: Ingeniería de sistemas (SE), Ingeniería del software (SW), Desarrollo integrado del Producto y del Proceso (IPPD) y la Gestión de Compras y Control de Proveedores (A).

- **Ingeniería de Sistemas (SE):** Cubre el desarrollo de sistemas completos, que pueden o no incluir software. Esta disciplina se centra en la transformación de las necesidades, expectativas y restricciones del cliente en soluciones y en apoyar estas soluciones a través de la vida del producto.
- **Ingeniería de Software (SW):** Cubre el desarrollo de los sistemas de software. Esta disciplina se enfoca en la aplicación de métodos sistemáticos, disciplinados, y cuantificables del desarrollo, la operación y mantenimiento del software.
- **Desarrollo Integrado de Productos y Procesos (IPPD):** El desarrollo integrado de productos y procesos es un método sistemático que logra la colaboración oportuna de los diferentes grupos de interés durante el ciclo de vida de un producto para satisfacer las necesidades, expectativas y requisitos del cliente.
- **Administración de Proveedores:** A medida que los esfuerzos de trabajo se vuelvan más complejos, las organizaciones pueden utilizar a los proveedores para realizar funciones o para agregar modificaciones a los productos.

### 1.4.1 Áreas de procesos de CMMI

Estas áreas vistas desde la representación continua del modelo, se agrupan en 4 categorías según su finalidad: Gestión de proyectos, Ingeniería, Gestión de procesos y Soporte a las otras categorías. Vistas desde la representación escalonada, se clasifican en los 5 niveles de madurez. Al nivel de madurez 2 pertenecen las áreas de proceso cuyos objetivos deben lograr la organización para alcanzarlo.

<b>Áreas de proceso de CMMI (Capability Maturity Model Integration)</b>		
<b>Área de proceso</b>	<b>Categoría</b>	<b>Nivel de madurez</b>
Análisis y resolución de problemas	Soporte	5
Gestión de la configuración	Soporte	2
Análisis y resolución de decisiones	Soporte	3
Gestión integral de proyecto	Gestión de proyectos	3
Gestión integral de proveedores	Gestión de proyectos	3
Gestión de equipos	Gestión de proyectos	3
Medición y análisis	Soporte	2
Entorno organizativo para integración	Soporte	3
Innovación y desarrollo	Gestión de procesos	5
Definición de procesos	Gestión de procesos	3
Procesos orientados a la organización	Gestión de procesos	3
Rendimiento de los procesos de la organización.	Gestión de procesos	4
Formación	Gestión de procesos	3
Integración de producto	Ingeniería	3
Monitorización y control de proyecto	Gestión de proyectos	2
Planificación de proyecto	Gestión de proyectos	2
Gestión calidad procesos y productos	Soporte	2
Gestión cuantitativa de proyectos	Gestión de proyectos	4
Desarrollo de requisitos	Ingeniería	3
Gestión de requisitos	Ingeniería	2
Gestión de riesgos	Gestión de proyectos	3
Gestión y acuerdo con proveedores	Gestión de proyectos	2
Solución técnica	Ingeniería	3
Validación	Ingeniería	3
Verificación	Ingeniería	3

Tabla 1.1 Áreas de proceso del modelo CMMI.

Un **área de proceso** es un conjunto de prácticas relacionadas en un área que al ponerse en ejecución colectivamente, satisfacen un sistema de metas consideradas importantes para llevar a cabo la mejora en esta área.

Para dar cumplimiento a un área de proceso es necesario cumplir con metas específicas y genéricas planteadas en el área y a su vez para cumplir estas metas deben realizarse prácticas específicas y genéricas asociadas a las metas.

**Metas específicas:** Describen las características únicas que deben ser presentadas para satisfacer el área de proceso, son componentes requeridos y se utilizan para determinar si el área de proceso fue satisfecha.

**Metas genéricas:** Son llamadas metas genéricas ya que la misma meta se aplica a múltiples áreas de procesos, describiendo las características que deben estar presentes para institucionalizar los procesos puestos en ejecución de un área de proceso determinada, son componentes requeridos y se utilizan para determinar si el área de proceso fue satisfecha.

**Prácticas específicas:** Describen las actividades consideradas importantes en la realización de las metas específicas asociadas, así como las actividades que se espera que den lugar al logro de las metas específicas de un área de proceso. Son componentes esperados.

**Prácticas genéricas:** Son se llamadas prácticas genéricas porque la misma práctica se aplica a múltiples áreas de proceso, describen una actividad considerada importante en la realización de las metas genéricas asociadas y son componentes esperados. [21]

### 1.4.2 Áreas de proceso en relación a la investigación

Para la investigación, las áreas de proceso relacionadas están ubicadas en el nivel 2 de la escala de los niveles de madurez de CMMI, pues con este se consigue que en los proyectos haya una gestión de los requisitos y que los procesos (formas de hacer las cosas) estén planeados, ejecutados, medidos y controlados, así se logra que los proyectos se ejecuten y gestionen de acuerdo con los planes de proyecto, los elementos de trabajo se revisen con las personas involucradas y estos satisfagan las especificaciones, estándares y objetivos.

Estas ideas se materializan en las siguientes áreas de proceso:

✓ **Gestión de Requisitos o Requerimientos.**

Con la gestión de requisitos de los elementos del proyecto y sus componentes se logra identificar inconsistencias entre estos, el plan de proyectos y los elementos de trabajo, gestionándose tantos requisitos funcionales como no funcionales, y se podrán revisar con las personas encargadas de estas revisiones.

### ✓ **Planificación de proyectos.**

Esta área de proceso permite la planificación de proyectos para establecer y mantener planes que definan las actividades del proyecto.

Las tareas que conlleva la planificación de proyectos son:

- Desarrollar un plan inicial del proyecto.
- Establecer una relación adecuada con todas las personas involucradas en el proyecto.
- Obtener compromiso con el plan.
- Mantener el plan durante el desarrollo del proyecto.

El plan de proyectos es un herramienta de trabajo viva que se debe de actualizar con mucha frecuencia ya que los requisitos cambiarán, habrá que re estimar, habrá riesgos que desaparezcan y otros que surjan nuevos, habrá que tomar acciones correctivas. [22]

### ✓ **Medición y Análisis**

El objetivo de la medición y el análisis es desarrollar y sostener una capacidad de medición, usada para ayudar a las necesidades de información de la actividad en gestión. Los datos tomados para la medición deben estar alineados con los objetivos del proyecto para proporcionar información útil.

### ✓ **Aseguramiento de la calidad**

Con esta área de proceso se va a proporcionar personas y gestión de actividades de aseguramiento de la calidad con el objetivo de que los procesos y los elementos del flujo de prueba se hagan con la calidad requerida.

Este aseguramiento de la calidad se va a lograr evaluando objetivamente la ejecución de los procesos, los elementos de trabajo contra las descripciones de procesos, estándares y procedimientos, además se va a identificar y documentar los elementos no conformes, proporcionando información a las personas que están usando los procesos, de los resultados de las actividades del aseguramiento de la calidad, asegurándose que los elementos no conformes sean arreglados. Esta es un área de proceso clave, que a veces no se le da la suficiente importancia, pero que sin ella no será posible lograr la calidad necesaria en el flujo de prueba.

### ✓ **Gestión de la configuración**

Con la gestión de la configuración se establece y mantiene la integridad de los artefactos de trabajo identificados, controlando y auditando dichos artefactos. Se logra mantener una organización de todos estos elementos, facilitándose el control y gestión de los mismos.

### **1.5 ISO (Organización Internacional de Normalización)**

ISO es una federación mundial de organismos nacionales de normalización, donde se prepara el trabajo de las normas internacionales mediante sus comités técnicos. Las series de ISO 9000 son un grupo de 5 individuales, pero relacionadas, estándares internacionales de administración de la calidad y aseguramiento de calidad. Estos estándares fueron desarrollados para documentar efectivamente los elementos de calidad que son instrumentados para mantener un sistema eficiente de calidad en un proyecto. [23]

#### **1.5.1 ISO 15504(SPICE, Software Process Improvement and Capability Determination)**

ISO/IEC 15504 es un emergente estándar internacional de evaluación y determinación de la capacidad y mejora continua de procesos de ingeniería del software, con la filosofía de desarrollar un conjunto de medidas de capacidad estructuradas para todos los procesos del ciclo de vida y para todos los participantes. SPICE es una norma que trata los procesos de ingeniería, gestión, relación cliente-proveedor, de la organización y del soporte. [24]

Mediante este modelo se puede valorar los procesos de software, auto-evaluándose y ofreciendo un mecanismo que brinde confianza en los resultados de la valoración, permitiendo determinar la capacidad de los procesos, así como su adecuación para cumplir un requisito determinado o una clase de requisitos. Con esta norma se puede asegurar que productos y procesos de un proyecto cumplen con sus requerimientos específicos según lo planificado.

#### **1.5.2 ISO 9001**

Básicamente, la norma ISO 9001, son un conjunto de reglas de carácter social y organizativo para mejorar y potenciar las relaciones entre los miembros de una organización. Cuyo último resultado, es mejorar las capacidades y rendimiento de la organización, y conseguir un aumento por este procedimiento de la calidad final del producto. [25]

Con esta norma se provee además que el aseguramiento de la calidad del producto pretende aumentar la satisfacción del cliente. El proyecto debe establecer, documentar, implementar y mantener un sistema de gestión de la calidad y mejorar continuamente su eficacia de acuerdo con los requisitos de esta Norma Internacional.

El proyecto debe:

- Identificar los procesos necesarios para el sistema de gestión de la calidad y su aplicación.
- Determinar la secuencia e interacción de estos procesos.



- Determinar los métodos necesarios para asegurarse de que la ejecución y control de estos procesos sean eficaces.
- Asegurar la disponibilidad de recursos e información necesaria para apoyar la ejecución y seguimiento de estos procesos, realizando el análisis de los mismos.
- Implementar las actividades necesarias para alcanzar los resultados planificados y la mejora continua de estos procesos.

La organización debe gestionar estos procesos de acuerdo con los requisitos de esta Norma Internacional. [26]

### 1.5.3 ISO 9000 3

El objetivo de la ISO 9000-3 es proveer cómo aplicar la ISO 9001 al desarrollo del software. Se incluyen algunos temas que no se encuentran en las normas ISO 9000 genéricas, tales como Administración de la Configuración o Planeación de Proyectos. Sería poco probable lograr resultados de calidad en un proyecto de desarrollo software de tamaño mediano, sin haber tomado las provisiones necesarias para el control de configuración. Esto implica que para ciertos productos o servicios, la especificación de requerimientos contenida en las normas genéricas ISO 9000 no es suficiente para asegurar la calidad, y esto justifica la necesidad de otras normas o guías más específicas. [27]

La norma ISO 9000-3 se usa en el desarrollo de software, para cubrir las expectativas de los clientes, obteniéndose beneficios de calidad y como estrategia para reducir los costos de producción.

Dentro de los beneficios que se obtienen de la certificación ISO 9000-3, se encuentran:

- Mejor documentación en el sistema.
- Incremento en la eficiencia y productividad.
- Mayor percepción de calidad.
- Se amplía la satisfacción del cliente.
- Se reducen las auditorías de calidad de los clientes.
- Agiliza el tiempo de desarrollo del sistema.

#### 1.5.3.1 Secciones de la Norma

##### ✓ **Responsabilidades de la dirección**

La dirección del proyecto debe definir y documentar su política y sus objetivos con respecto a la calidad. Debe asegurarse que esta política es conocida, entendida e implementada en todos los niveles del área temática.

Las responsabilidades y relaciones entre todo el personal, cuyo trabajo afecte la calidad del producto, deben ser definidas.

### ✓ **Sistemas de Calidad**

El proyecto debe establecer y mantener un procedimiento de calidad documentado (un manual interior como guía de actividades de aseguramiento de la calidad) como medio de asegurar que los productos cumplen con los requerimientos especificados y la aplicación efectiva de este procedimiento y de las instrucciones documentadas del aseguramiento de la calidad.

### ✓ **Inspección y pruebas**

El proyecto debe asegurar que los productos no se utilicen o procesen hasta que sean inspeccionados o verificados que cumplen con los requerimientos específicos. Las verificaciones deben estar de acuerdo con el plan de calidad y los procedimientos documentados.

Cuando los productos son aplicados sin ser antes inspeccionados, éstos deben identificarse y registrarse para que en caso de no conformidad sean rápidamente reconocidos y reemplazados. El proyecto debe establecer o mantener registros que contengan el criterio de aceptación del producto.

### ✓ **Equipos de Inspección, medición y pruebas**

El proyecto debe controlar y mantener el equipo de inspección, medición y pruebas para verificar la conformidad del producto con los requerimientos especificados. El equipo debe ser usado de una manera que asegure que la incertidumbre de medición sea conocida y que esté dentro de la capacidad de medición requerida. El proyecto debe:

- Precisar las mediciones a efectuar, con la exactitud requerida y además, seleccionar el equipo adecuado de inspección y pruebas.
- Establecer, documentar y mantener los procedimientos de medición que incluyan detalles del equipo en cuanto a frecuencia de verificación, criterios de aceptación y las acciones a tomar cuando los resultados no sean satisfactorios.
- Asegurarse de que el equipo de inspección, medición y pruebas registra la exactitud, el error y la precisión requerida.
- Mantener registros de evaluación del equipo de inspección, medición y pruebas.
- Auditar y documentar la validez de los resultados de las inspecciones y pruebas cuando los equipos de medición, inspección y pruebas sean encontrados sin medición.
- Asegurar los equipos de inspección, medición y pruebas para evitar ajustes que invaliden la evaluación.

### ✓ **Estado de Inspección y pruebas**

La identificación del estado y pruebas debe ser mantenida en el proceso de producción e instalación del producto para asegurar que sólo los que hayan pasado las pruebas e inspecciones requeridas sean entregados al cliente.

### ✓ **Control de producto no conforme**

El proyecto debe mantener y controlar los procedimientos que aseguren que los productos que no cumplan los requerimientos especificados, no sean usados o instalados inadvertidamente. Se deben controlar las actividades de identificación, documentación, evaluación (cuando sea práctico) y desecho de productos no-conformes.

### ✓ **Control de registros de calidad**

El proyecto debe establecer y mantener los registros de calidad. Todos los registros deben ser legibles e identificables con el producto del que se trate. El tiempo que deberán mantenerse esos registros debe ser definido y registrado.

### ✓ **Auditorías internas de calidad**

El proyecto debe llevar un sistema de auditorías internas de calidad, planeado y documentado, para verificar que las actividades de calidad cumplan con lo planeado y que determine la efectividad del aseguramiento de a calidad del flujo de pruebas. Las auditorías deben programarse de acuerdo con la importancia de la actividad. La auditoría y el seguimiento deben llevarse a cabo de acuerdo a los procedimientos documentados. El resultado de las auditorías debe ser documentado y mostrado al personal que tenga responsabilidad en el área auditada. El personal administrador responsable del área debe tomar acciones correctivas sobre las deficiencias encontradas por la auditoría.

### ✓ **Capacitación**

El proyecto debe establecer y mantener procedimientos para identificar las necesidades de capacitación y proveer entrenamiento a todo el personal que realice tareas específicas debe ser calificado con base en su educación, entrenamiento y/o experiencia. Se deben mantener registros apropiados de capacitación. [28]

#### **1.5.4 ISO 12207**

Se establece un proceso de ciclo de vida para el software que incluye procesos y actividades. Este estándar tiene como objetivo principal proporcionar una estructura para que se use un lenguaje común, de manera que pueda ser adaptada a las necesidades de cualquiera que lo use.

Para conseguirlo, el estándar se basa en dos principios fundamentales: modularidad y responsabilidad. Con la modularidad se pretende conseguir procesos con un mínimo acoplamiento y una máxima cohesión. En cuanto a la responsabilidad, se busca establecer un responsable para cada proceso, facilitando la aplicación del estándar en proyectos en los que pueden existir distintas personas involucradas. [29]

### Procesos del ciclo de vida

#### Procesos de Apoyo

**1. Aseguramiento de calidad:** Define las actividades para asegurar, de una manera objetiva, que los productos software y los procesos son conforme a sus requerimientos especificados y se ajustan a sus planes establecidos. Revisión Conjunta, Auditoría, verificación y Validación pueden ser utilizados como técnicas de Aseguramiento de la Calidad. [30]

✓ **Actividades del proceso:**

Con el proceso de aseguramiento de la calidad se asegura que los productos y los procesos empleados para proporcionar dichos productos, cumplan con los requisitos establecidos, y estén coordinados con los procesos de verificación, validación, revisión conjunta y auditoría. Además, se debe preparar, documentar, implementar y mantener un plan para llevar a cabo las actividades de aseguramiento de la calidad, este plan debe incluir:

- Metodologías, procedimientos y herramientas para llevar a cabo las actividades de aseguramiento de la calidad.
- Recopilación, rellenado, mantenimiento y eliminación de registros de calidad.
- Recursos, cronogramas y responsabilidades para llevar a cabo las actividades del aseguramiento de la calidad.

Cuando se detecten problemas o no conformidades con los requerimientos del contrato se deben documentar. Se deben preparar y mantener registros de estas actividades y tareas, de su ejecución, de los problemas y sus soluciones.

**2. Validación:** Se definen las actividades para validar los productos software del proyecto. Consta de las siguientes tareas:

- Se preparan los requerimientos de prueba, casos de prueba y especificaciones de prueba seleccionados para analizar los resultados de las pruebas.
- Se asegura que estos requerimientos de prueba, casos de prueba y especificaciones de prueba reflejan los requerimientos particulares para el uso específico previsto.
- Llevar a cabo pruebas como:
  - Pruebas con sobrecarga, límites y entradas excepcionales.
  - Pruebas del producto software respecto a su habilidad para aislar y minimizar el efecto de errores.

- Prueba de usuarios representativos que pueden llevar a cabo con éxito sus tareas previstas usando el producto software.
- Validar que el producto software satisface su uso previsto.
- Probar el producto software cuando sea apropiado en áreas seleccionadas del entorno de destino.

**3. Revisión conjunta:** Se definen las actividades para evaluar el estado y productos de un elemento. Este proceso puede ser empleado por cualquiera de las dos partes, donde una de las partes (la revisora) revisa a la otra parte (la parte revisada), de una manera conjunta. Este proceso consta de las siguientes actividades:

✓ **Implementación del proyecto**

Se deben llevar a cabo revisiones periódicas tal como se especifica en los planes del proyecto, registrando los problemas detectados durante las revisiones, así como se debe documentar y distribuir los resultados de estas revisiones.

**4. Auditoría:** Se definen actividades para determinar la conformidad con los requerimientos, planes y contrato. Se debe llevar a cabo auditorías tal como se especifica en los planes del proyecto, los auditores no deben tener responsabilidad directa sobre los productos y actividades que auditen, y deben registrar los problemas detectados durante las auditorías, documentando y proporcionándola a la parte auditada.

### Procesos de Gestión

**1. Gestión:** El líder de proyecto es responsable de la gestión del producto, gestión del proyecto y gestión de las tareas de los procesos aplicables.

✓ **Revisión y evaluación**

El líder de proyecto deberá asegurar que los productos software y los planes se evalúan con relación a la satisfacción de los requerimientos, analizando los resultados de la evaluación de los productos software, actividades y tareas completadas durante la ejecución del proceso, en relación al cumplimiento de los objetivos y los planes.

✓ **Finalización**

El líder de proyecto deberá comprobar que los resultados y registros de los productos software, actividades y tareas empleadas se han completado y se deberá archivar estos resultados y registros en un entorno adecuado.

**2. Mejora:** Define las actividades básicas que una organización (adquiriente, proveedor, desarrollador, operador, responsable de mantenimiento o gestor de otro proceso) lleva a cabo para establecer, medir, controlar y mejorar sus procesos del ciclo de vida.

✓ **Establecimiento del proceso**

Se debería documentar los procesos y su aplicación en casos específicos y establecer un mecanismo de control del proceso para desarrollar, hacer seguimientos, controlar y mejorar los procesos.

✓ **Evaluación del proceso**

Se deben guardar y mantener registros de las evaluaciones. El proyecto debe planificar y llevar a cabo revisiones de los procesos que asegure su continua adecuación y efectividad, según los resultados de las evaluaciones.

✓ **Mejora del proceso de mejora**

El proyecto debe efectuar en sus procesos las mejoras que se consideran necesarias como resultado de las evaluaciones y revisiones de los procesos. Se debe actualizar la documentación del proceso para reflejar las mejoras en los procesos de la organización.

**3. Formación:** Se define cómo conseguir personal adecuadamente capacitado. Contiene las siguientes actividades:

✓ **Implementación del proceso**

Se debe determinar los tipos y niveles de capacitación y categorías del personal que la necesita. Se deberá preparar y documentar un plan de capacitación que tenga en cuenta los plazos de implementación, necesidad de recursos y necesidades de formación.

✓ **Implementación del plan de formación**

Se deberá implementar el plan de capacitación para proporcionar la formación al personal, manteniendo registros de formación. Se deberá asegurar que el personal esté adecuadamente capacitado y esté disponible en el momento preciso para las actividades y tareas planificadas.

### 1.5.5 ISO 9126

Esta Norma se utiliza en la certificación de productos. Simplemente, certifica si el software sirve para lo que dice que sirve. Sintetiza una serie de características que deben reunir los programas para que sean considerados de calidad. El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterística de la siguiente manera: [31]

**Funcionalidad** - Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen lo indicado o implica necesidades.

**Fiabilidad** - Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período de tiempo establecido.

**Usabilidad** - Un conjuntos de atributos relacionados con el esfuerzo necesitado para el uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

**Eficiencia** - Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.

**Mantenimiento** - Conjunto de atributos relacionados con el esfuerzo necesitado para modificar las especificaciones.

**Movilidad** - Conjunto de atributos relacionados con la habilidad del software para ser transferido desde un entorno a otro.

Mediante este estándar se puede proveer un entorno para definir en el proyecto un aseguramiento de la calidad para el flujo de prueba.

Se puede distinguir entre fallos y no conformidad, siendo un fallo el no cumplimiento de los requisitos previos, mientras que la no conformidad afecta a los requisitos especificados. Una distinción similar es hecha entre la validación y la verificación. [32]

### Lineamientos de Calidad

Los productos desarrollados en el área temática de APS, deben cumplir con los lineamientos de calidad establecidos por la Universidad ver **Anexo 14**.

A lo largo de este capítulo se trataron temas relacionados con la calidad, afines con las buenas prácticas de CMMI (Modelo de Madurez y Capacidad Integrado), de las diferentes normas ISO (Organización Internacional de Normalización) y además un estudio según lo planteado por PSP y TSP en relación con la calidad.

**CAPÍTULO** **2**  
**ESTADO DEL ARTE****2.1 Flujo de Trabajo de Prueba**

La figura 2.1 muestra el flujo de trabajo de prueba utilizado en cada iteración del desarrollo de los componentes o módulos del área temática APS.

Este flujo cuenta con las siguientes actividades: Primero se planifican las pruebas, después se analizan y diseñan, se implementan, se ejecutan y se evalúan, donde si el resultado de la evaluación no es positivo se debe regresar a la actividad planificar pruebas, con el fin de volver a repetir el ciclo de vida del flujo de trabajo; de ser el resultado de la evaluación positivo, se pasa a la última actividad: aceptar las pruebas, donde quedarán cumplidos los objetivos de prueba de la iteración pasando al proceso de liberación correspondiente por parte del grupo de desarrollo de conjunto con su líder de proyecto, al grupo de desarrollo del área temática APS, donde se realizarán las pruebas de aceptación según las estrategias definidas por el proyecto.

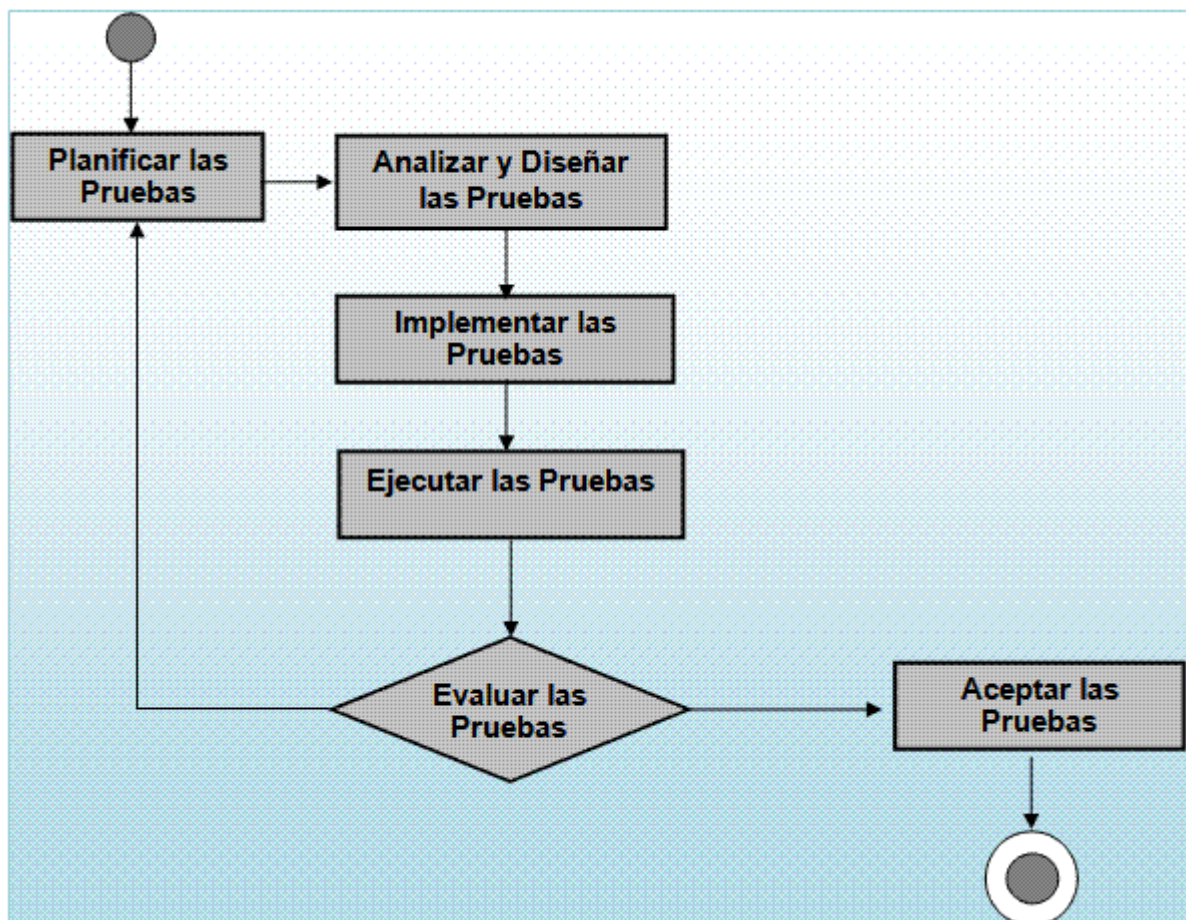


Figura 2.1 Flujo de trabajo de prueba del Área Temática de APS.



### 2.1.1 Actividades del Flujo de trabajo de Prueba

- Planificar las Pruebas.
- Analizar y Diseñar las Pruebas.
- Implementar las Pruebas.
- Ejecutar las Pruebas.
- Evaluar las Pruebas.
- Aceptar las Pruebas.

#### ✓ Planificar pruebas

En la actividad planificar pruebas la función fundamental es describir la estrategia de prueba de la iteración definiendo claramente el método apropiado para ejecutarlas, los objetivos, las posibles pruebas a realizar, las pautas a seguir y el enfoque de las mismas. El objetivo de esta actividad es realizar el Plan de prueba que describe los recursos y la planificación de las mismas.

Los trabajadores: administrador de prueba, analista de prueba y diseñador de prueba, mediante las actividades que realizan en la iteración participan en la elaboración de este plan, siendo el administrador de prueba el principal responsable de elaborar el Plan de prueba, que es el artefacto de salida de esta actividad y debe poseer como artefactos de entrada, el modelo de análisis, modelo de diseño, modelo de caso de uso, descripción de los casos de uso, descripción de la Arquitectura y documento visión. Además puede tener una lista de ideas de prueba, documentos o artefactos que le faciliten la confección del plan.

#### ✓ Analizar y Diseñar las pruebas

La actividad diseñar prueba fluye a partir de las principales ideas de las pruebas que se van a realizar, al llevar a cabo un diseño de las pruebas y los diferentes casos de pruebas a implementar.

Esta actividad verifica los enfoques de la pruebas, por ejemplo: si estas pueden brindar resultados precisos y es apropiado para los recursos disponibles. Además su objetivo es lograr una solución de implementación apropiada de cada técnica de pruebas o encontrar técnicas alternativas que puedan usarse; es decir se analizan, se identifican y se construyen los procedimientos de prueba que especifican como realizar uno o varios casos de prueba, sirven de guía para que un individuo sepa cómo realizar casos de prueba manualmente o una especificación de cómo interactuar con una herramienta para ejecutar una prueba.

Los casos de pruebas de la iteración, también se coleccionan y se administran los datos de prueba generándose los scripts de datos de prueba, así como se realizan las descripciones de los casos de pruebas los cuales describen la forma de probar un sistema incluyendo las entradas, salidas y resultados esperados.

Los casos de prueba describen bajo qué condiciones debe probarse el sistema y ellos pueden derivarse de las descripciones de los casos de uso del diseño. Siendo los artefactos de salida de esta actividad los casos de prueba, los scripts de datos de prueba y los procedimientos de prueba.

En esta actividad intervienen los trabajadores analista de prueba y diseñador de prueba y tiene como artefactos de entrada, el modelo de implementación, el modelo de caso de uso, la descripción de los casos de uso, el plan de prueba, la descripción de la arquitectura, el documento visión y el modelo de diseño.

Al describir los casos de pruebas deben quedar definidos:

- Descripción General.
- Nombre de la Prueba.
- Descripción.
- Condiciones de ejecución.
- Entrada.
- Resultado esperado.
- Evaluación de la Prueba.

### ✓ **Implementar las Pruebas**

El propósito de la actividad implementar prueba consiste en automatizar uno o varios procedimientos de pruebas, pues esto ahorraría tiempo y esfuerzo a la hora de ejecutar las pruebas, así como a la hora de llevar a cabo pruebas de regresión sería bueno tener una herramienta que lo hiciese.

Existen herramientas que cuando se ejecuta una prueba con ella, esta graba los pasos de la prueba y los guarda, lo cual es algo muy útil a la hora de ejecutar pruebas de regresión. Esta actividad la deben ejecutar los trabajadores analista de prueba y diseñador de pruebas, los mismos son responsables de actividades que tributan al objetivo de la actividad. Los artefactos de entrada son los procedimientos de prueba y los artefactos de salida los componentes de prueba

### ✓ **Ejecutar las Pruebas**

La actividad ejecutar pruebas tiene como objetivo realizar correctamente las pruebas planificadas y diseñadas. Se van a realizar las pruebas unitarias, de integración y de sistema según los objetivos de la iteración.

Es decir se centra en la ejecución del artefacto casos de pruebas, a partir de las descripciones de los casos de prueba, apoyándose de los componentes de pruebas que automatizan uno o varios procedimientos de prueba proporcionando entradas de pruebas, controlando y monitoreando los componentes a probar. El uso de herramientas de prueba facilita y mejora el proceso de ejecución de las pruebas, además se ejecutarán algunas pruebas manualmente, siempre guiadas por los epígrafes del Plan de pruebas:

- Estrategia de prueba.
- Tipos de pruebas.
- Recursos.

En la actividad, interviene el trabajador probador y la misma tiene como entrada los artefactos casos de prueba, procedimiento de prueba, componente de prueba, plan de prueba y como artefacto de salida la lista de defectos, que no son más que las anomalías del sistema, que se detectan durante la ejecución de las pruebas.

### ✓ **Evaluación de las Pruebas**

Esta actividad se centra en evaluar los esfuerzos de las pruebas en una iteración. Para evaluar los resultados de pruebas estos deben ser comparados con los objetivos esbozados en el Plan de prueba. Esta actividad reúne fundamentalmente mantener y mejorar la calidad de las pruebas, pues permite re-usar las anteriores actividades del actual ciclo de prueba, en ciclo de pruebas posteriores. La misma presenta como entrada los artefactos Plan de prueba, modelo de prueba, los defectos y como artefacto de salida la evaluación de las pruebas. En esta actividad intervienen todos los trabajadores del flujo de prueba: administrador de prueba, analista de prueba, diseñador de prueba y probador.

### ✓ **Aceptar las Pruebas**

Esta actividad se centra en la aceptación o no de las pruebas a nivel de proyecto por el líder de proyecto teniendo en cuenta la evaluación de las mismas, empezando entonces los procesos de gestión de cambios y/o liberación del grupo de proyecto de Softel. En esta actividad intervienen todos los trabajadores del flujo de trabajo de prueba: administrador de prueba, analista de prueba, diseñador de prueba y probador.

### 2.1.2 Trabajadores

- Administrador de Prueba.
- Analista de Prueba.
- Diseñador de Prueba.
- Probador.

#### ✓ **Administrador de Prueba**

El administrador de prueba, es responsable del éxito global de las pruebas, dirige y organiza el proceso.

#### **Subactividades que debe cumplir:**

- Identificar los elementos que motiven o faciliten las pruebas.
- Evaluar y auditar la calidad.
- Evaluar y mejorar las pruebas.
- Informar al líder de proyecto de los resultados del proceso de aceptación de la pruebas.
- Evaluar los resultados de las pruebas.

#### ✓ **Analista de Prueba**

Es responsable de identificar y definir las pruebas requeridas en una iteración, supervisando el progreso de las pruebas, así como el resultado para cada ciclo de prueba. El rol de manera general tiene la responsabilidad de representar las necesidades de los clientes que no se relacionan directamente con el proyecto. Este rol es una especialización del administrador de prueba.

#### **Subactividades que debe cumplir:**

- Identificar los objetivos de las pruebas.
- Identificar las ideas de las pruebas.
- Generar los scripts de datos de prueba.

#### ✓ **Diseñador de Prueba**

El diseñador de prueba, se centra en identificar las técnicas apropiadas, herramientas y pautas para llevar a cabo las pruebas y usar los recursos correspondientes para el esfuerzo de las mismas.

### Subactividades que debe cumplir:

- Definir enfoque de las pruebas.
- Diseñar los casos de prueba.
- Describir los casos de prueba
- Identificar herramientas apropiadas para la ejecución de las pruebas.
- Estructurar la implementación de las pruebas.
- Identificar y definir técnicas apropiadas de prueba.

### ✓ **Probador**

El probador es encargado de la actividad de mayor esfuerzo del flujo de trabajo de prueba, las pruebas se realizan ejecutando los casos de pruebas a partir de sus procedimientos. Estas pruebas se pueden ejecutar de forma manual, o a través de componentes de pruebas o herramientas de pruebas, las cuales ayudan a aminorar el esfuerzo de las mismas para una iteración así como facilitan ejecutar pruebas de sistema, como son las de carga, rendimiento, stress entre otras.

### Subactividades que debe cumplir:

- Analizar y evaluar fallos como resultado de las pruebas.
- Identificar y listar los defectos encontrados.

## 2.2 Estado actual de las pruebas y la calidad en el proyecto

Con el objetivo de conocer y profundizar sobre el estado de las pruebas de Software, como parte importante en la producción de los mismos, se llevó a cabo encuestas a 25 integrantes, incluyendo estudiantes y profesores del área temática APS, con el objetivo de conocer el estado actual de las pruebas, la plantilla utilizada se encuentra en el **Anexo 13**. Los resultados de las entrevistas realizadas mostraron los siguientes datos:

El 76% de los encuestados no tienen conocimiento de la existencia de un plan de prueba, solo el 24% aseguran que se planifican las pruebas como está establecido.

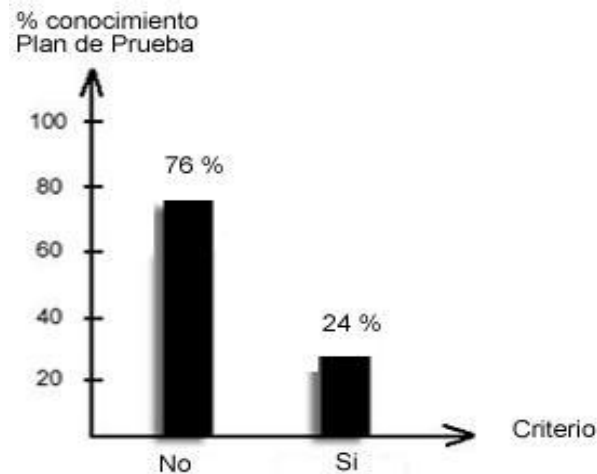


Figura 2.2 Conocimiento del Plan de Prueba.

Un 80% de los programadores prueban lo que están desarrollando, para verificar que cumplen con lo establecido pues no es factible esperar al final, así van verificando la calidad del trabajo que están realizando y solo un 20 % de ellos no lo hacen dejando los defectos para el final.

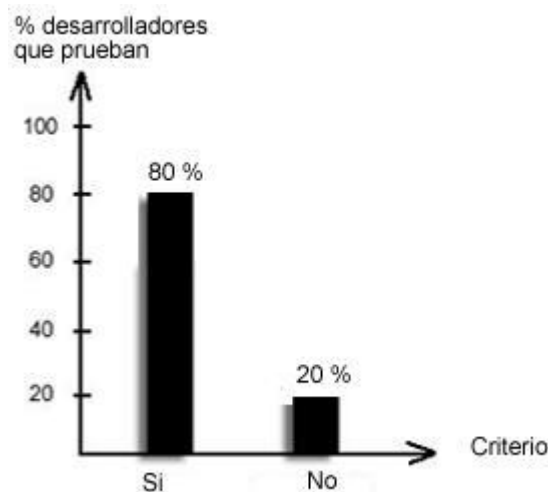


Figura 2.3 Porcentaje de pruebas realizadas.

En la realización de las actividades del flujo de prueba que se lleva a cabo en el proyecto, se tiene que se identifican los objetivos de la planificación de las pruebas al 80 %. En cuanto a diseñar las pruebas que fluye a partir de las ideas de pruebas, se lleva a cabo el diseño de casos de prueba a un 59 % así como su descripción, ya que se llevan a la par dichas actividades; mientras que solo un 36% identifica y define las técnicas de prueba a utilizar.

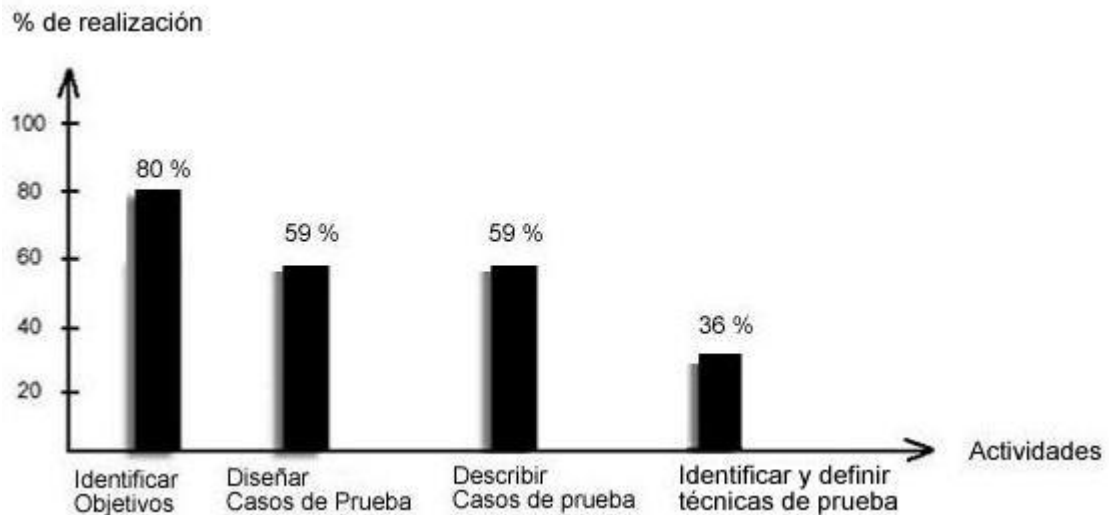


Figura 2.4 Actividades del flujo de trabajo de prueba.

De las restantes actividades analizan y evalúan los fallos el 80 % de los ingenieros involucrados. Un 63 % realizan la identificación de las listas de defectos y un 87 % están capacitados para estas actividades del flujo.

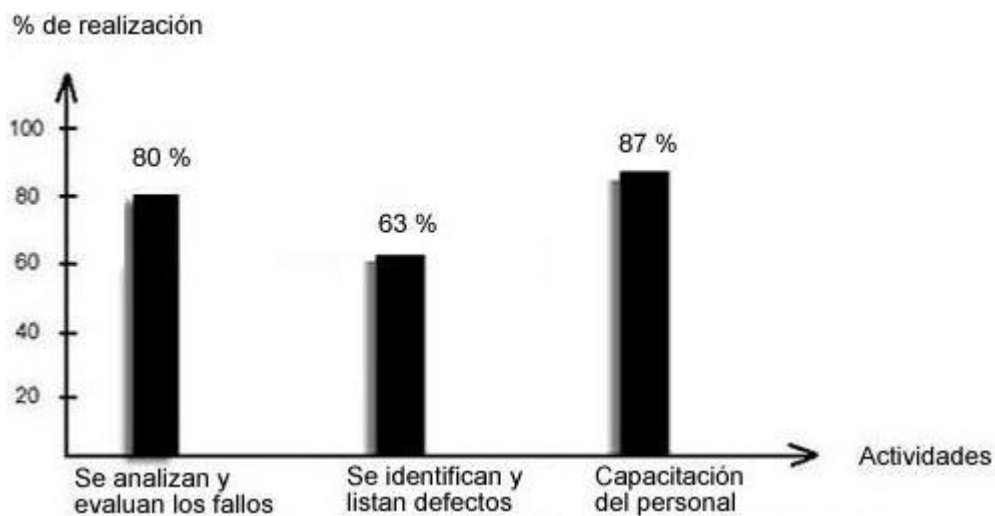


Figura 2.5 Técnicas de registros.

Las técnicas de registro de PSP no se aplican en la gran mayoría de los casos, pues solo un 20% lleva el registro de tiempo por el cual se rige, y un 30% de los integrantes del equipo planifica las actividades que realiza, así como el 63% identifica los defectos del producto. Las técnicas TSP son más utilizadas pues existe cierta comunicación entre los integrantes del equipo, teniendo una organización entre los mismos que ayuda a controlar la forma en que se trabaja, aunque no se lleva a un 100% pues no se cumple con los objetivos que persigue TSP(Ver Anexo 13).

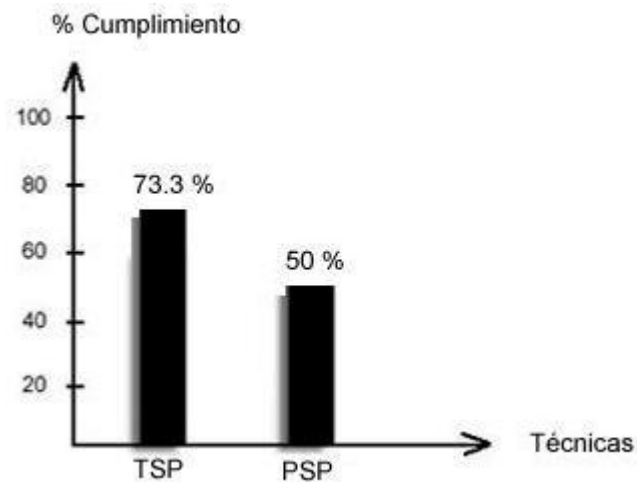


Figura 2.6 Buenas Prácticas.

Para lograr un control de la calidad se hacen técnicas de inspecciones, revisiones, pruebas y auditorías en un 43, 77, 83 y 36 % respectivamente, concluyendo que no se hacen al 100 % estas actividades para lograr una buena calidad de los productos que se están desarrollando.

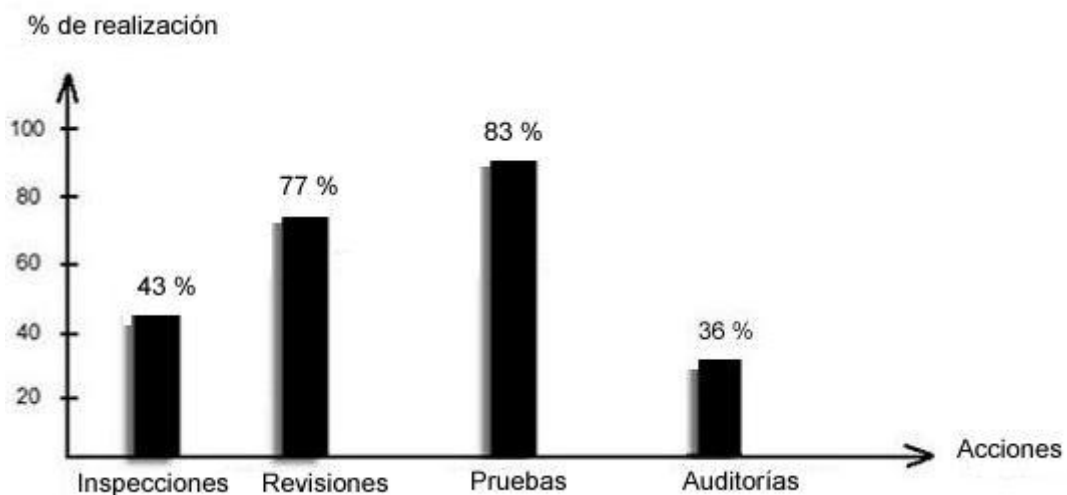


Figura 2.7 Actividades de control de calidad.

En cuanto a los elementos principales que debe tener un plan de aseguramiento de calidad (SQA) solo el 28 % de los encuestados tienen conocimiento de la existencia de un plan de SQA en el proyecto, el 42 % sigue las políticas de calidad requeridas para el logro de los objetivos, el 55% hace auditorías de los procesos involucrados y no se llevan a cabo evaluaciones de calidad.



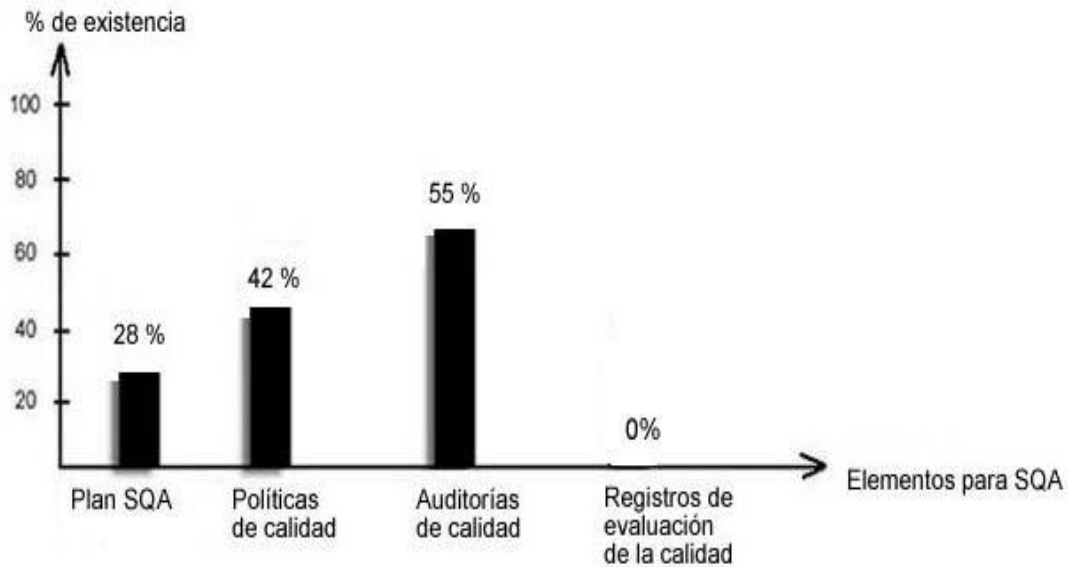


Figura 2.8 Elementos de calidad.

Concluyendo se plantea que los requisitos a tener en cuenta para lograr un producto de alto nivel con la calidad necesaria, no se están cumpliendo de la forma más adecuada, ni siquiera a un 90 %. Actualmente no se está siguiendo pautas para un aseguramiento de la calidad de los productos que se están realizando en el proyecto. Estos factores demuestran que es necesaria la creación de un procedimiento que guíe al grupo SQA para realizar las actividades y que garanticen que se cumplan con los aspectos que se requieren a la hora de realizarle pruebas al software y finalmente que satisfagan las necesidades del cliente.

En el presente capítulo se hizo un estudio del flujo de prueba utilizado en el área temática de APS, sus actividades y artefactos. Además se hace un análisis del estado de las pruebas y la calidad en el área temática de APS, mediante unas encuestas realizadas a estudiantes y profesores de dicha área.

## PROPUESTA DE PROCEDIMIENTO Y HERRAMIENTAS **CAPÍTULO 3**

### 3.1 Procedimiento para el Aseguramiento de la calidad en el flujo de trabajo de prueba

Teniendo en cuenta que la Ingeniería de Software es una tecnología multicapa, que se apoya sobre un compromiso de organización de calidad, se realiza la definición del procedimiento a proponer, el cual pretende recorrer todas estas capas (ver figura 3.1) en la parte correspondiente al flujo de trabajo de prueba.



Figura 3.1 Capas de la Ingeniería de Software para el flujo de trabajo de Prueba.

El procedimiento se basa en las mejores prácticas del uso de CMMI e ISO, descritas previamente en el capítulo 1. El modelo de calidad CMMI brinda una guía del trabajo a desarrollar por parte del grupo SQA y mediante los estándares ISO se pueden documentar efectivamente los elementos para el aseguramiento de calidad que son instrumentados para garantizar un aseguramiento eficiente de calidad en el proyecto.

Basado en la ISO 15 504, que establece valorar los procesos y asegurar que cumplen con sus requerimientos específicos. Con la ISO 9001 el procedimiento propone la estructura del grupo de aseguramiento de la calidad, responsabilidades, procesos y métodos, y se apoya en la ISO 90003 que provee como aplicar la ISO 9001 al proceso de prueba agregando la administración de la configuración y planeación de aseguramiento de la calidad. Además la norma ISO 12207 ayuda a definir las actividades para asegurar, de una manera objetiva, así como con la ISO 9126 se pueda comprobar si el producto sirve para lo que realmente se necesita. Esto se comprobará mediante Revisiones y Auditorías que serán utilizadas como técnicas de Aseguramiento de la Calidad.

A continuación se describen los epígrafes del procedimiento propuesto:

## 3.1.1 Objetivos

Regular las actividades que definen el aseguramiento de la calidad del flujo de trabajo de prueba.

## 3.1.2 Alcance

Para el área temática Atención Primaria de la Salud (APS).

## 3.1.3 Definiciones

SQA: Aseguramiento de la Calidad del Software.

RTF: Revisiones Técnicas Formales.

APS: Atención Primaria de la Salud.

Desviación: Son violaciones cometidas y que no están acorde con lo planteado en el Plan SQA del proceso de prueba o Plan de Prueba del proyecto.

RTF: Revisiones técnicas formales (RTF) son métodos formales para asegurar la calidad objetivamente, teniendo como objetivos descubrir errores antes que el producto esté en manos del cliente, encontrar desviaciones, documentarlas e informarlas al líder del proyecto, de acuerdo a los requerimientos y estándares establecidos.

## 3.1.4 Descripción

Las **actividades propuestas** son las siguientes:

1. Conformar el grupo de aseguramiento de la calidad (SQA).
2. Establecimiento de un Plan de SQA del proceso de prueba para el proyecto.
3. Participación en el desarrollo de la descripción del proceso de prueba.
4. Revisión de las actividades de ingeniería de software para verificar su ajuste al proceso de prueba definido.
5. Auditorías de los procesos de software designados para verificar el ajuste con los definidos como parte del proceso de prueba.
6. Asegurar que las desviaciones del trabajo y los productos de software se documentan y se manejan de acuerdo con un procedimiento establecido.
7. Medir los procesos.
8. Revisión de las actividades del grupo SQA del proyecto.
9. Entrenamiento del grupo SQA del proyecto.
10. Evaluar los procesos del grupo SQA del proyecto.
11. Gestión de Configuración de los artefactos generados por el grupo SQA del proyecto.

### 3.1.4.1 Conformar el grupo de aseguramiento de la calidad (SQA).

Las reglas del grupo de SQA tratan de ayudar al equipo de ingeniería del software en la obtención de un producto final de alta calidad.

El grupo SQA a conformar debe llevar a cabo un proceso con calidad, además se deben poner en práctica un grupo de actividades a lo largo de todo el proceso de desarrollo, teniendo en cuenta que las mismas se desarrollan paralelamente, sin darle a unas mayor o menor importancia. Debe participar en el establecimiento de los planes, procesos, estándares y procedimientos definidos en el Plan del Proyecto, donde cada uno debe realizar las tareas que le corresponde con la calidad requerida y así asegurar la calidad del proceso y el producto. Por tanto se proponen los siguientes integrantes del grupo SQA.

1 Jefe del grupo de aseguramiento de la calidad.

1 Jefe de revisión.

10 Revisores.

3 Auditores.

### 3.1.4.2 Establecimiento de un Plan SQA del proceso de prueba para el proyecto.

Todos los proyectos de desarrollo de software del área temática de APS deben tener definido un Plan SQA del proceso de prueba, el cual debe guiar las actividades realizadas por el grupo SQA, así como a los ingenieros de software implicados en el proyecto.

Este plan se desarrolla durante la planificación del proyecto, en la fase de Inicio y es revisado por todas las partes interesadas. Este plan debe mantenerse en constante mejora, siempre que sea necesario ser modificado. La descripción de este acápite va teniendo ajustes en la medida que se va refinando el alcance del Plan de Desarrollo de Software del Proyecto, originándose tantas revisiones como sean necesarias.

Se propone utilizar el Plan SQA definido por la Dirección de Calidad de la Universidad (**Anexo 1 Plan SQA**).

A continuación se describen los acápites más significativos del Plan SQA, del proceso de prueba, propuesto.

#### 1. Propósito

*Se expone el propósito del Plan SQA del proceso de prueba para el flujo de trabajo de prueba, donde se definen las actividades de aseguramiento de la calidad del flujo de trabajo de prueba de auditoría, revisión y evaluación de los procesos en el desarrollo.*

### 2. Alcance.

*Se define el alcance del Plan SQA del proceso de prueba según el alcance del Plan de Desarrollo de Software del Proyecto aprobado.*

### 3. Objetivos de Calidad.

*Se incluyen los requerimientos del proyecto, donde exista claridad, completitud y consistencia en los artefactos desarrollados en el proyecto, alineados con los requerimientos de calidad siguientes.*

- Lograr que el módulo cumpla con los RF y RNF que se definieron con el cliente.
- Lograr la calidad requerida de los artefactos del desarrollo de software descrito en el expediente de proyecto.
- Lograr llevar a cabo una correcta administración y ejecución de las pruebas en todo el ciclo de vida del software.
- Lograr una correcta revisión de los artefactos generados en el desarrollo del software basado en los estándares definidos en el proyecto.

### Requisitos No Funcionales del proyecto APS.

- **Usabilidad**

Cada usuario que se autentique sólo tendrá acceso a la información que le corresponde, ya sea editor o visualizador a nivel nacional, provincial, municipal o de unidad de salud.

- **Rendimiento**

**RNF1** El sistema debe soportar la conexión simultánea de más de 60 000 usuarios.

**RNF 2** El sistema debe soportar un tiempo de respuesta menor o igual a 30 segundos.

**RNF 3** El sistema debe soportar respuestas SOAP que no deben exceder los 50 Kbyte en el tiempo de respuesta al usuario.

- **Soporte**

**RNF 4** El personal que trabaja con el módulo debe contar con el nivel técnico requerido mediante adiestramiento de servicio.

- **Portabilidad**

**RNF 5** Permitir que el sistema se ejecute sobre el Sistema Operativo Windows 98 o superior o Linux.

- **Confidencialidad**

**RNF 6** Disponer de un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoría (AAA).

**RNF 7** La autenticación será la primera acción del usuario en el sistema y consistirá en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica.

**RNF 8** Si el usuario autenticado no se encuentra registrado se debe reportar un error de acceso.

**RNF 9** Si el usuario autenticado se encuentra registrado se autoriza su acceso y se crea un certificado digital.

**RNF 10** El certificado digital constará de:

- Un identificador único (token) de 32 caracteres generado de manera aleatoria en el momento de la autorización.
- El identificador del usuario.
- El nivel de acceso del usuario (Nacional, Provincial, Municipal o Unidad de Salud).
- El identificador del nivel de acceso.
- Un listado de los módulos que el usuario tiene acceso y el tipo de acceso en cada uno de ellos (Editor o Visualizador).

**RNF 10.1** Deberá registrarse el token, el identificador del usuario presente en el certificado digital y el día, mes, año, hora, minuto, segundo de la creación del certificado.

**RNF 10.2** Para cada petición que se haga al sistema, debe enviarse el token recibido en el certificado digital el cual será validado con la lista de tokens autorizados que el sistema ha registrado.

**RNF 10.3** Si el token se encuentra registrado y la petición se corresponde con los derechos del usuario se procederá a ejecutar la petición solicitada.

**RNF 10.4** Si el token no se encuentra registrado o la petición no se corresponde con los derechos del usuario se debe reportar un error de acceso.

**RNF 11** Cada Petición de usuario, autorizada o no, será registrada, así como el día, mes, año, hora, minuto, segundo en que se registra y si fue o no autorizada.

- **Integridad**

**RNF 12** Prevenir posibles fallos y recuperarse ante ellos.

**RNF 13** Deberán existir mecanismos de chequeo de integridad.

**RNF 14** Deberá existir una estrategia de replicación que permita, de manera transparente para el usuario final, balancear la carga de acceso entre múltiples servidores aumentando los tiempos de respuesta y facilitar la recuperación inmediata del sistema si falla uno de ellos.

**RNF 15** Se permitirá la creación de copias de respaldo que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información.

- **Disponibilidad**

**RNF 17** La seguridad no implicará lentitud o retaso en la respuesta dada por el sistema, por lo que se debe minimizar y reducir el tiempo de respuesta, así como optimizar el código.

#### 4. Organización.

*El grupo de SQA está compuesto por catorce estudiantes encargados del aseguramiento de la calidad y de las pruebas y un jefe del grupo de SQA.*

<b>Rol</b>	<b>Cantidad</b>
Jefe del grupo de SQA.	1
Audidores	3
Revisores	10
Jefe de revisión.	1

Tabla 3.1 Grupo de SQA.

#### 5. Tareas y Responsabilidades.

*Se muestra una tabla que registra la asignación de responsables y participantes por tareas del procedimiento.*

<b>Tareas</b>	<b>Participantes</b>	<b>Responsable</b>
Revisión de artefactos generados en el flujo de pruebas.	Jefe de revisión y revisores.	Revisores.
Auditoría a los procesos y artefactos que están guardados en el repositorio.	Audidores, auditor(es) de la facultad, auditor(es) externo.	Audidores.
Verificar documentación de las desviaciones y su seguimiento.	Líder de proyecto, grupo de SQA.	Jefe del grupo SQA.
Medición de procesos.	Líder de proyecto, grupo de SQA de la facultad, personas externas independientes al proyecto APS y de la facultad.	Jefe del Grupo SQA

Entrenamiento del grupo SQA del proyecto APS.	Grupo de SQA, Líder del proyecto.	Jefe del grupo de SQA.
Evaluación de los procesos.	Grupo de SQA, Líder de proyecto.	Líder de proyecto.
Gestión de configuración.	Líder de proyecto, grupo de SQA, comité de control de cambios.	Jefe del grupo de SQA.

Tabla 3.2 Tareas y Responsables.

### 6. Métricas.

*En este epígrafe se describen las métricas a utilizar. Estas son detalladas en la sección 3.1.4.6 del procedimiento.*

### 7. Estándares y Guías.

*Lista de los estándares y guías utilizadas por el Plan SQA del proceso de prueba, acorde a los estándares del proyecto APS que se decidirán en la descripción del proceso de prueba, reflejados en el Plan de desarrollo del software.*

Estándar	Ubicación
Estándares de codificación.	Subversion ( <a href="http://10.31.7.35:5901/repositorio/trunk/Estándares/Estándares_Negocio">http://10.31.7.35:5901/repositorio/trunk/Estándares/Estándares_Negocio</a> )
Estándar para elaboración de documentos.	Subversion ( <a href="http://10.31.7.35:5901/repositorio/trunk/Estándares/Estándares_Negocio">http://10.31.7.35:5901/repositorio/trunk/Estándares/Estándares_Negocio</a> )
Interfaz de Usuario.	Subversion ( <a href="http://10.31.7.35:5901/repositorio/trunk/EXPEDIENTE_DE_PROYECTO/Expediente_del_proyecto/Ingeniería/Requisitos/">http://10.31.7.35:5901/repositorio/trunk/EXPEDIENTE_DE_PROYECTO/Expediente_del_proyecto/Ingeniería/Requisitos/</a> )
Pruebas: <ul style="list-style-type: none"> <li>- Plan de prueba.</li> <li>- Casos de prueba.</li> <li>- Componentes de prueba.</li> <li>- Procedimientos de prueba.</li> <li>- Lista de defectos.</li> </ul>	Subversion ( <a href="http://10.31.7.35:5901/repositorio/trunk/Expediente_del_proyecto/Ingeniería/Implementacion_y_pruebas">http://10.31.7.35:5901/repositorio/trunk/Expediente_del_proyecto/Ingeniería/Implementacion_y_pruebas</a> )



- Evaluación de las pruebas.	
Manuales de usuario.	Subversion <a href="http://10.31.7.35:5901/repositorio/trunk/EXPEDIENTE_DE_PROYECTO/Expediente_del_proyecto/Ingeniería/Implementación_y_pruebas/">http://10.31.7.35:5901/repositorio/trunk/EXPEDIENTE_DE_PROYECTO/Expediente_del_proyecto/Ingeniería/Implementación_y_pruebas/</a>
Otros estándares.	Subversion ( <a href="http://10.31.7.35:5901/repositorio/trunk/Estándares">http://10.31.7.35:5901/repositorio/trunk/Estándares</a> )

Tabla 3.3 Estándares y Guías.

### 8. Tareas generales de las Revisiones y Auditorías.

Estas revisiones estarán guiadas por el Plan de Aseguramiento de la Calidad del proyecto, el jefe del grupo de SQA será el responsable de ordenar el cronograma de aseguramiento de calidad.

De aquí entonces se auditarán y revisarán todos los artefactos generados en el proceso de pruebas, por medio de los estándares, listas de chequeos, y el proceso de pruebas en general.

*Las revisiones son referenciadas en la actividad 3.1.4.3 del procedimiento.*

*Las tareas generales a desarrollar son las siguientes:*

*Definir el cronograma de revisión.*

*Ejecutar el proceso de revisión.*

*Realizar conclusiones del proceso de revisión.*

*En los planes SQA del proceso de prueba y Desarrollo del Software deben aparecer los procesos y artefactos del flujo de trabajo de prueba que serán revisados.*

*Las auditorías son descritas en la actividad 3.1.4.4 del procedimiento.*

*Las tareas generales a desarrollar son las siguientes:*

*Definir el cronograma de auditoría.*

*Ejecutar el proceso de auditoría.*

*Realizar conclusiones del proceso de auditoría.*

*En los planes SQA del proceso de prueba y Desarrollo del Software deben aparecer los artefactos del flujo de trabajo de Prueba que serán auditados.*

### 9. Cronograma, Organización, Responsabilidades de las Revisiones y Auditorías.

De este cronograma se podrá planificar por cada miembro del equipo las actividades de revisión y auditoría. Se debe detallar el cronograma de todas las tareas específicas para lograr la correcta ejecución de todos los artefactos del flujo de pruebas haciendo uso de la siguiente tabla:

Tarea	Responsable	Fecha Inicio	Fecha Fin
Revisión del Plan de pruebas.	Jefe de revisión y Revisores.		
Revisión de los Casos de prueba, Script de pruebas y procedimientos de prueba.	Jefe de revisión y Revisores.		
Revisión de la Lista de defectos y evaluación de las pruebas.	Jefe de revisión y Revisores		
Auditoría a los artefactos generados en el flujo de pruebas.	Auditores.		

Tabla 3.4 Tareas y Responsables de Revisión y Auditoría.

### 10. Técnicas y Metodologías

En este acápite se describen las técnicas y metodologías específicas que serán usadas para llevar a cabo las actividades de revisión y auditoría (expuestas en las actividades 3.1.4.3 y 3.1.4.4 respectivamente) identificadas en este plan.

### 11. Herramientas

Listar las herramientas utilizadas en las actividades del Plan SQA del proceso de prueba en dicho flujo.

Se propone el uso de la herramienta DotProject para el seguimiento del cronograma del Plan SQA del proceso de prueba.

Se propone el uso de la herramienta JMeter y Rational Robot, para la ejecución de las pruebas de estrés y pruebas de regresión respectivamente.

Repositorio (Subversion). Para garantizar la gestión de la configuración.

Listas de chequeo de Plan de Pruebas.

Listas de chequeo de Casos de Prueba.

Listas de chequeo de procedimientos de prueba.

Listas de chequeo de componente de prueba.

Listas de chequeo de defecto.

Listas de chequeo de evaluación de las pruebas.

### **12. Gestión de Configuración.**

*Se hace referencia en la actividad 3.1.4.10 del procedimiento.*

### **13. Registros de Calidad.**

*Se describen los Registros de Calidad utilizados en las actividades de revisión y auditorías en el flujo de trabajo de prueba y dónde será guardado cada registro (Subversion).*

- Acta de revisión. Ver **Anexo 2**.
- Registro de revisión. Ver **Anexo 3**.
- Registro de auditorías. Ver **Anexo 4**.
- Lista de desviaciones. Ver **Anexo 5**.
- Listado de defectos. Ver **Anexo 6**.

Estos registros se guardarán en el Subversion, y formarán parte del expediente de proyecto.

### **14. Entrenamiento**

*Se explica en la actividad 3.1.4.8 del procedimiento.*

#### **3.1.4.3 Participación en el desarrollo de la descripción del proceso de prueba.**

- ✓ **Los integrantes del grupo SQA, junto a probadores y demás integrantes deben participar en la descripción del proceso de prueba.**

En la definición del proyecto deben quedar establecidos los estándares, técnicas y metodologías que serán aplicadas en el mismo, en el flujo de trabajo de prueba, quedando reflejadas en el punto 10 del Plan de SQA.

- ✓ **El grupo SQA tiene la responsabilidad de establecer la concordancia del proceso definido con las políticas de calidad del proyecto.**

Para el desarrollo del flujo de trabajo de prueba se propone cumplir con determinadas actividades y artefactos, considerando este aspecto como política del proyecto, teniendo en cuenta los objetivos de calidad que deben quedar reflejados en el punto 3 del Plan de SQA, relacionados con los RF y RNF identificados para el proyecto que se analiza.

<b>Actividades</b>	<b>Artefactos</b>
1. Planificar las Pruebas.	1. Plan de Prueba
2. Analizar y Diseñar las Pruebas.	2. Casos de Prueba 3. Procedimientos de Prueba 4. Script de datos de Prueba
3. Implementar las Pruebas.	5. Componente de prueba 6. Lista de defectos
4. Ejecutar las Pruebas.	7. Evaluación de las Pruebas
5. Evaluar las Pruebas.	
6. Aceptar las Pruebas.	

Tabla 3.5 Artefactos de salida del proceso de prueba.

### 3.1.4.4 Revisión de las actividades de ingeniería de software para verificar su ajuste al proceso de prueba definido.

- ✓ **Deben realizarse Revisiones Técnicas Formales (RTF) en el flujo de prueba, planificándose estas y quedando reflejadas en el Plan SQA del proceso de prueba.**

Se deben llevar a cabo mediante una reunión previamente planificada, quedando reflejadas las técnicas y metodologías a utilizar en el Plan SQA del proceso de prueba, descritas en el punto 5 del epígrafe 3.1.4.2.

Los artefactos de prueba deben revisarse en pequeñas porciones para encontrar el mayor número de errores y lograr un buen desarrollo del mismo.

Las tareas generales que deben desarrollarse en las RTF son las siguientes:

1. Se define el cronograma de revisión según lo planificado en el plan de SQA, y se asigna la responsabilidad a los encargados de revisión por parte del grupo de SQA (Jefe de revisión y los demás revisores).
2. Individualmente los revisores se preparan por adelantado, revisando el producto y tomando las notas pertinentes. Se harán revisiones a los artefactos generados por el flujo de trabajo de

prueba como, el plan de prueba, casos de prueba, procedimientos de prueba, componentes de prueba y a la evaluación de las pruebas, para verificar el ajuste de los mismos con los requisitos esperados. En este equipo debe participar además el desarrollador pues este será quien debe estar dispuesto para informar a los integrantes del equipo de revisión sobre cualquier duda existente en los documentos que deben ser revisados. El Jefe de revisión también cumple con el rol de revisor.

Luego se convoca a una reunión, donde participan todos los involucrados en la revisión, el jefe de la revisión, el desarrollador y los revisores. Se hace un análisis conjunto de las revisiones y se toman notas de los errores detectados, los resultados del proceso de revisión quedan anexados al **Anexo 2**, ubicándose estas bajo la Administración de Control de Cambio, de forma tal que los interesados tengan fácil acceso a las mismas.

3. Se concluye que si se rechaza el artefacto debido a los errores encontrados se propone para una próxima revisión. Una vez tomada la decisión, se establece la fecha de la próxima revisión, esto queda reflejado en el epígrafe conclusiones del Acta de Revisión.

Para la revisión de los artefactos mencionados en la tabla 3.5 del epígrafe 3.1.4.3, se proponen utilizar las siguientes listas de chequeo correspondientes a cada uno de estos y presentados como anexos.

Lista de Chequeo Plan de Prueba. **Anexo 7.**

Lista de Chequeo Casos de Prueba. **Anexo 8.**

Lista de Chequeo Procedimientos de Prueba. **Anexo 11.**

Lista de Chequeo Componente de prueba. **Anexo 10.**

Lista de Chequeo Lista de defectos. **Anexo 9.**

Lista de Chequeo Evaluación de las Pruebas. **Anexo 12.**

### **3.1.4.5 Auditorías de los procesos de software designados para verificar el ajuste con los definidos como parte del proceso de prueba.**

- ✓ **Auditar los procesos de prueba definidos en el Plan SQA.**

Para las auditorías se utilizará un acta de revisión igual a la de las RTF, diferenciándose solo en los objetivos de cada actividad. En el Registro de Auditorías (**Anexo 4**) se describen los puntos correspondientes a esta plantilla.

Las tareas generales a desarrollar en las auditorías son las siguientes:

1. Se define el cronograma de auditoría para los procesos definidos en el Plan de SQA, y se asigna la responsabilidad a los auditores del grupo de SQA.

2. Se ejecuta el proceso de auditoría a cada uno de los procesos definidos como auditables en el Plan SQA del proceso de prueba y según la política del proyecto, los auditores deben auditar que dichos procesos se han desarrollado objetivamente, y detectar las desviaciones e informarlas al jefe del grupo SQA, quien se encarga de informarlo al líder del proyecto. En la auditoría puede participar además un auditor de la facultad, que es integrante del grupo de SQA de la facultad, y un auditor externo, que puede ser ajeno al proyecto y a la facultad, para garantizar una mejor visión en el proceso de auditoría.

A la hora de realizar las auditorías no es necesaria la presencia de todos a la vez, pueden participar de forma aleatoria, nunca sin la presencia de los auditores del grupo de SQA del proyecto.

3. Al finalizar la auditoría de los procesos seleccionados, debe detallarse si el proceso es aprobado o no, quedando reflejadas en el Acta de Revisión (**Anexo 2**) y en el Registro de Auditorías (**Anexo 4**) para de esta forma darles seguimiento.

#### **3.1.4.6 Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.**

Las desviaciones son detectadas en las RTF y Auditorías, explicadas en las actividades 3.1.4.3 y 3.1.4.4 respectivamente, quedando reflejadas en una Lista de Desviaciones (**Anexo 5**) y además siendo informadas al jefe del grupo de SQA. Estas desviaciones pueden ser: Desviaciones en los atributos de calidad o desviaciones en el cronograma del proyecto

Las desviaciones presentadas al jefe del grupo de SQA deben revisarse periódicamente hasta que se solucionen. La documentación de estas desviaciones debe ser controlada y administrada por un integrante del grupo de SQA, en este caso es el revisor asignado por el jefe del grupo SQA, para cumplir con esta función.

#### **3.1.4.7 Medición de los procesos.**

- ✓ **Se deben medir los procesos de prueba realizados para tener la certeza de que estén desarrollándose correctamente.**

El jefe del grupo de SQA será el encargado de medir los procesos.

Para medir la ejecución del proceso de prueba se define la métrica siguiente:

$$Expp = Cpe/Cpt$$

Donde:

Expp: Éxito del proceso de prueba.

Cpe: Casos de pruebas exitosos.

Cpt: Casos de prueba total.

El proceso de prueba medido será exitoso si el valor de Éxito del proceso de prueba (Expp) está en el rango de 0.8 a 1, este es el umbral de la métrica.

$$0.8 < \text{Expp} \leq 1$$

### 3.1.4.8 Revisión de las actividades del grupo SQA del proyecto.

Las actividades del grupo SQA deben ser revisadas por terceros

**Líder del proyecto:** Debe revisar periódicamente cómo se desarrollan las actividades del grupo SQA.

**Grupo SQA de la facultad:** Debe revisar periódicamente las actividades planificadas y llevadas a cabo por el grupo SQA del proyecto, así como fechas y duración de las mismas. Además puede gestionar la revisión del proceso de aseguramiento de la calidad por personas independientes al proyecto y a la facultad.

### 3.1.4.9 Entrenamiento del grupo de SQA del proyecto.

- ✓ **Se debe entrenar a las personas que realicen el proceso de aseguramiento de la calidad en el proyecto.**

Los integrantes del proyecto APS varían de un año a otro, por lo que es necesario entrenar a los que ingresan al proyecto y que pertenezcan al grupo SQA. Para llevar a cabo esta actividad se debe definir un plan de entrenamiento para el proyecto. El personal encargado de las actividades de aseguramiento de la calidad debe tener amplio conocimiento en diferentes materias, por lo que se proponen algunos temas en los que deben ser entrenados:

- Qué tipo de arquitectura de software se usa en el proyecto.
- Cómo llevar a cabo los procesos de Revisiones, Revisiones Técnicas Formales y Auditorías.
- Cómo identificar las Desviaciones resultantes de los procesos de Revisiones, Revisiones Técnicas Formales y Auditorías.
- Cómo medir los procesos a través de métricas.
- Cómo evaluar los procesos a través de métodos de evaluación.
- Cursos relacionados con el flujo de prueba llevado a cabo en el proyecto APS.
- Las responsabilidades que tiene una persona a la hora de ocupar un rol determinado, como parte del grupo SQA del proyecto.

- Actividades específicas del grupo SQA (procedimiento de SQA).

Se deben realizar reuniones técnicas todas las semanas para evacuar dudas y tratar temas de aseguramiento de la calidad de software.

### **3.1.5.10 Evaluación del proceso del grupo SQA del proyecto.**

- ✓ **Evaluar el proceso realizado por parte del grupo SQA del proyecto contra lo que está descrito en el Plan SQA.**

Para evaluar el proceso de aseguramiento de la calidad se debe valorar si es necesario mejorar o no, no basta con la opinión del líder del proyecto, se debe justificar con datos reales la necesidad de llevar a cabo una mejora. Se deben realizar encuestas y entrevistas de manera que se detecten debilidades y proponer como mejorar las mismas.

### **3.1.5.11 Gestión de Configuración de los artefactos generados por el grupo SQA del proyecto.**

- ✓ **Se debe guardar en el repositorio del proyecto los elementos de configuración de las actividades del grupo SQA que se pongan bajo la administración de configuración.**

Elementos que se deben guardar en el Repositorio:

- Acta de Revisión.
- Lista de Defectos.
- Lista de Desviaciones.
- Registros de Auditorías.
- Registros de Evaluación.
- Registro de Revisión.
- Listas de Chequeos.
- Plan de Aseguramiento de la Calidad del proceso de prueba.
- Plan de Capacitación

El procedimiento propuesto desarrolla un grupo de actividades para asegurar la calidad en el flujo de trabajo de prueba del área temática APS. Para esto definen las tareas a desarrollar por el grupo SQA del área temática APS, así como el vínculo que debe existir entre este grupo SQA del proyecto y el grupo SQA de la facultad.



## 3.2 Herramientas para la administración y ejecución de las pruebas.

A continuación se exponen algunas de las herramientas que son las más utilizadas en la actualidad, sus principales utilidades y beneficios. Algunas, se utilizan en la UCI y en otras, su uso es aún incipiente. La automatización de las pruebas dentro de un proceso de software trae consigo que se pueda agilizar y optimizar este proceso.

El objetivo es mostrar algunas de las características fundamentales de estas, así como su uso y explotación en aras de mejorar el proceso de pruebas.

### 3.2.1 JMeter

Dándole continuidad a la recomendación hecha en la tesis del curso 2006-2007, de Yoan Manuel y Yosbel Ernesto, de investigar sobre el trabajo con la herramienta JMeter para flujo de prueba en el área temática de APS, se muestra en este epígrafe las funciones de la herramienta, así como las ventajas que trae su uso. Se muestra además un ejemplo de una prueba realizada en la herramienta y en los anexos una guía más completa de su uso.

El JMeter es una herramienta libre, además es una herramienta Java, que permite realizar pruebas de Rendimiento y pruebas Funcionales sobre Aplicaciones Web. Es una herramienta de carga para llevar acabo simulaciones sobre cualquier recurso de Software.

JMeter una herramienta Java dentro del proyecto de Jakarta, que permite realizar pruebas de rendimiento y pruebas funcionales sobre aplicaciones web y bases de datos. Existe un gran número de herramientas para realizar pruebas gratuitas y de pago (LoadRunner), pero JMeter se destaca por su versatilidad, estabilidad, y por ser de uso gratuito. JMeter permite realizar pruebas web clásicas, pero también permite realizar test de FTP, JDBC, JNDI, LDAP, SOAP/XML-RPC y Web Service (en Beta). También permite la ejecución de pruebas distribuidas entre distintos ordenadores para realizar pruebas de rendimiento.

Esta herramienta se basa en el concepto de "Elemento" (Element), y en una estructura en "Árbol" (Tree). Cualquier parte o rama del árbol, puede ser guardada de forma independiente, para ser reutilizada en otras pruebas. Los "elementos" van a permitir reutilizar y definir el plan de pruebas.

Una característica importante del JMeter es que es extensible y ofrece la posibilidad de que el propio usuario desarrolle en Java un controlador ("Controller") a medida, cumpliendo una interfaz Java y depositando el .jar correspondiente al desarrollo en el directorio "lib" de JMeter. Además JMeter permite realizar pruebas distribuidas en distintos ordenadores que actúan como clientes. Puede simular una gran carga en el servidor, HTTP y FTP testing y bases de datos mediante JDBC, multitarea y con grandes facilidades para extender su funcionalidad mediante plugins.

El JMeter muestra los resultados de las pruebas en una amplia variedad de informes y gráficas. Además facilita a una rápida detección de los cuellos de botella existentes debido al tiempo de respuesta excesivo. Todas estas herramientas pueden ser usadas para hacer las pruebas de eficiencia bajo carga intensiva, sin embargo hay algunas que poseen ventajas con respecto a las demás, por lo que son más óptimas a utilizar durante las pruebas en cuestión.

En una aplicación es de vital importancia emplear algo de tiempo a preparar pruebas de eficiencia bajo carga y stress, antes de ser entregada. El tiempo invertido es recuperado con creces, ya que se detectarían los posibles efectos laterales y se podrá comprobar si esa nueva funcionalidad soporta la cantidad de usuarios concurrentes que se especificaban en los requisitos.

Las pruebas de eficiencia bajo carga intensiva, no por ser poco conocidas y aplicadas son de menos importancia que las demás. Tanto a nivel mundial como en Cuba, aún no existe el conocimiento suficiente sobre dichas pruebas en conjunto. Existen muchas empresas que conocen de pruebas de stress y pruebas de carga, pero no de pruebas para medir los tiempos de respuesta de las aplicaciones web bajo stress.

✓ **Estos son algunos objetivos específicos que se pueden tener:**

- Verificar que el sistema esté ajustado para soportar la máxima carga de trabajo posible usando la infraestructura actual.
- Asegurar que, ante una carga de trabajo determinada, las páginas a las que se accede responden en menos del intervalo de tiempo especificado por el grupo de diseñadores.
- Determinar el tiempo medio de respuesta que obtendrá el usuario.
- Determinar el número máximo de usuarios concurrentes que pueden acceder a una página específica, o transacciones por segundo que la aplicación es capaz de soportar.
- Identificar las páginas que responden más lentas y las más rápidas.
- Identificar las páginas con una mayor desviación típica en sus tiempos de respuesta. [33]

Las pruebas de eficiencia bajo carga intensiva deben de ejecutarse idealmente sobre un entorno estable, lo más similar posible al entorno final de producción, siguiendo los objetivos antes mencionados, siempre pudiendo ser agregados otros en dependencia del interés que se persiga.

✓ **Ventajas de la herramienta JMeter**

- De las herramientas gratis, es la más completa y útil para el tipo de pruebas en cuestión.
- Es una herramienta que sirve para realizar pruebas funcionales, pero también sirve para realizar pruebas de regresión en aplicaciones web, algo, que a veces es verdaderamente complicado,

según la aplicación, pero que es casi imprescindible en el mantenimiento y evolución de las aplicaciones, si se quiere asegurar un nivel de capacidad adecuado en la entrega del producto.

- Tiene una estructura en árbol que le da potencia, permitiendo que sea la imaginación de quien la use la que ponga los límites a la hora de diseñar el plan de prueba. Y brinda mayor cantidad de variantes para recoger los resultados obtenidos, que el resto de las herramientas gratis, lo que permiten hacer un análisis exhaustivo de las pruebas realizadas.

### 3.2.1.1 Ejecución de prueba de rendimiento al Registro de Actividades Diarias (RAD)

El sistema de gestión de la información de los Modelos de Actividades Diarias (RAD) forma parte en la actualidad del conjunto de aplicaciones desarrolladas para el Sistema de Información para la Salud (SISalud) en el área temática de APS. Dentro de los requisitos No Funcionales que rigen al desarrollo de la aplicación, se tiene por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc. Todos estos requisitos deben ser medibles. Debe indicarse en por ciento (%) relacionado con su tiempo.

**RNF 1:** El sistema debe soportar la conexión simultánea de más de 60 000 usuarios.

**RNF 2:** El sistema debe soportar un tiempo de respuesta menor o igual a 30 segundos.

**RNF 3:** El sistema debe soportar respuestas SOAP que no deben exceder los 50 Kbyte en el tiempo de respuesta al usuario.

La prueba fue llevada a cabo en una de las opciones del Registro de Actividades Diarias, Entrega de Guardia, donde va a ser objeto de prueba el reporte Casos Vistos Según Categoría de Pacientes.

El caso de uso comienza cuando el Personal de la Unidad de Salud se interesa por la cantidad de casos vistos según Indicadores de Salud en una determinada fecha.

### Descripción Textual.

<b>Obtener Reporte Casos Vistos Según Categoría de Pacientes</b>	
<b>Actores:</b>	Personal Unidad Salud
<b>Breve Descripción:</b>	El Caso de Uso comienza cuando el Personal de la Unidad de Salud se interesa por la cantidad de casos vistos según Categoría de Pacientes en una determinada fecha.
<b>Precondiciones:</b>	
<b>Poscondiciones:</b>	El sistema retorna los datos según los casos vistos por categorías existentes hasta esa fecha.

Tabla 3.6 Descripción textual Obtener Reporte Casos Vistos Según Categoría de Pacientes.

✓ **Prueba 1: Caso de prueba Casos Vistos Según Categoría de Pacientes.**

**Nombre del proyecto a medir:** Registro de Actividades Diarias.

**Nombre del caso de prueba:** Casos Vistos Según Categoría de Pacientes.

**Nombre del caso de uso a medir:** Casos Vistos Según Categoría de Pacientes.

**Número de la prueba:** 1

**Nivel de concurrencia o grupo de usuarios:** 5

**Ciclos de trabajo:** 10

**Opción a probar:** Se va a probar 1 opción del módulo Registro de Actividades Diarias. (Obtener Reporte Casos Vistos Según Categoría de Pacientes).

**Resultados obtenidos en prueba 1:**

<b>El tiempo máximo de respuesta de la opción Obtener Reporte Casos Vistos Según Categoría de Pacientes:</b>	2.1 segundos
<b>Tiempo promedio de obtención de resultados:</b>	0.4 segundos
<b>% de error:</b>	50 % en la opción probada.
<b>Respuestas en bytes:</b>	9.2 Kb/segundos

Tabla 3.7 Resultados de la prueba 1.

✓ **Prueba 2: Caso de prueba Casos Vistos Según Categoría de Pacientes.**

**Nombre del proyecto a medir:** Registro de Actividades Diarias.

**Nombre del caso de prueba:** Casos Vistos Según Categoría de Pacientes.

**Nombre del caso de uso a medir:** Casos Vistos Según Categoría de Pacientes.

**Número de la prueba:** 2

**Nivel de concurrencia o grupo de usuarios:** 10

**Ciclos de trabajo:** 10

**Opción a probar:** Se va a probar 1 opción del módulo Registro de Actividades Diarias. (Obtener Reporte Casos Vistos Según Categoría de Pacientes).

**Resultados obtenidos en prueba 2:**

<b>El tiempo máximo de respuesta de la opción Obtener Reporte Casos Vistos Según Categoría de Pacientes:</b>	4.2 segundos
<b>Tiempo promedio de obtención de resultados:</b>	0.6 segundos
<b>% de error:</b>	50 % en la opción probada.
<b>Respuestas en bytes:</b>	9.2 Kb/segundos

Tabla 3.8 Resultados de la Prueba 2.

✓ **Prueba 3: Caso de prueba Casos Vistos Según Categoría de Pacientes.**

**Nombre del proyecto a medir:** Registro de Actividades Diarias.

**Nombre del caso de prueba:** Casos Vistos Según Categoría de Pacientes.

**Nombre del caso de uso a medir:** Casos Vistos Según Categoría de Pacientes.

**Número de la prueba:** 2

**Nivel de concurrencia o grupo de usuarios:** 15

**Ciclos de trabajo:** 10

**Opción a probar:** Se va a probar 1 opción del módulo Registro de Actividades Diarias. (Obtener Reporte Casos Vistos Según Categoría de Pacientes).

### Resultados obtenidos en prueba 3:

<b>El tiempo máximo de respuesta de la opción Obtener Reporte Casos Vistos Según Categoría de Pacientes:</b>	5.1 segundos
<b>Tiempo promedio de obtención de resultados:</b>	0.9 segundos
<b>% de error:</b>	50 % en la opción probada.
<b>Respuestas en bytes:</b>	9.2 Kb/segundos

Tabla 3.9 Resultados de la Prueba 3.

Las pruebas fue llevada a cabo simulando una carga de 5, 10 y 15 usuarios solicitando Reportes de Casos Vistos Según Categoría de Pacientes del Registro de Actividades Diarias. La prueba con más de 15 usuarios no se puede llevar a cabo en estas condiciones, es decir en una sola máquina, ya que requiere de más de una máquina probando la aplicación. Haciéndola en una sola máquina, el máximo de usuarios permitidos es de 15, ya que el porcentaje de utilización de la CPU tiene que ser menor que 90%, si la prueba consume más de ese porcentaje, los resultados obtenidos son falsos, y no son los reales.

Una vez definidas las características del grupo de usuarios que se desea simular (Test Plan > Thread Group) para ver la carga generada por la aplicación, se crea un HTTP Request Defaults (Thread Group > Config Element > HTTP Request Defaults), se especifican parámetros requeridos como por ejemplo numero ip del servidor, puerto etc. Seguidamente se adiciona un Proxy Server (WorkBench > Non- Test Elements > http Proxy Server), donde se especifica donde van a ser guardadas las muestras. Se pone a correr la herramienta, y se comienza a trabajar en la aplicación, paso a paso para obtener el reporte Casos Vistos Según Categoría de Pacientes, a medida que se visite una página nueva de la aplicación, las muestras obtenidas de dicha página, se agregan en un Transaction Controller (Insert Parent > Logic Controller > Transaction Controller).

Una vez terminada la navegación en la aplicación, se agregan dos reportes uno para ver los tiempos de respuesta y otro para poder ver errores encontrados en dicha prueba (Thread Group > add > Listener > Summary Report and View Results Tree respectivamente).

The screenshot shows the 'Summary Report' window in JMeter. It includes a menu bar (File, Edit, Run, Options, Help) and a sidebar with a tree view of the test plan. The main area displays the report details and a data table.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Pagina Prin...	150	500	47	3327	593.54	0.00%	2.6/sec	21.80	8668.0
Pagina Pric...	150	1646	891	5155	776.70	100.00%	2.5/sec	27.63	11163.0
Pagina Men...	150	601	79	1812	214.62	0.00%	2.7/sec	45.87	17336.0
Seleccion E...	150	696	93	2062	264.53	100.00%	2.7/sec	24.49	9281.0
Seleccion r...	150	562	110	1531	214.34	0.00%	2.7/sec	0.00	.0
Listado de ...	150	1829	359	3109	497.94	100.00%	2.7/sec	24.35	9281.0
TOTAL	900	972	47	5155	725.61	50.00%	14.9/sec	134.85	9288.2

Tabla 3.2 Reporte de JMeter Summary Report.

Con este reporte se obtiene el tiempo de respuesta de la aplicación bajo la carga de 15 usuarios, se tiene que el tiempo de respuesta promedio de la aplicación es de 2 segundos, el tiempo mínimo se tardó de 0.047 segundo y el tiempo máximo fue de 5,1 segundo. El porcentaje de error fue del 50% que en análisis del reporte de la Tabla 3.2, se hace referencia a dichos errores. También muestra el rendimiento de la aplicación durante la navegación que fue de 14.9 segundos y una respuesta de 134 KB/segundos.

The screenshot shows the 'View Results Tree' window in JMeter. It displays a tree view of test results on the left and a detailed view of a selected 'Sampler result' on the right. The selected result shows a 404 error for a CSS file.

**Sampler result**

Thread Name: Registro de Actividades Diarias 1-4  
 Sample Start: 2008-06-23 07:42:19 ACT  
 Load time: 16  
 Size in bytes: 284  
 Response code: 404  
 Response message: Not Found

**Response headers:**  
 HTTP/1.1 404 Not Found  
 Date: Mon, 23 Jun 2008 11:47:55 GMT  
 Server: Apache/1.3.29 (Unix) PHP/4.3.4  
 Keep-Alive: timeout=15, max=44  
 Connection: Keep-Alive  
 Content-Type: text/html; charset=iso-8859-1

Tabla 3.3 Reporte de JMeter View Results Tree.

Con este reporte (View Results Tree) se puede obtener los errores encontrados por cada página visitada, por ejemplo En la página principal del RAD, se encontraron dos errores (/general/style.css y /general/allstyles\_aps.css) que según el reporte dice que no se encuentra dichos ficheros, al solicitar ésta página existe una demora, al hacer la solicitud de dichos ficheros, que no existen.

En el **Anexo 15** será descrita su instalación, así como una serie de pruebas para ejemplificar su uso.

### 3.2.2 Rational Robot

Rational Robot permite a los equipos de pruebas ('testers') automatizar el testing de regresión, funcional y de configuración para aplicaciones e-commerce, cliente/servidor y ERP. Se usa para testear aplicaciones basadas en una gran variedad de tecnologías de interfaz de usuario y está integrado a la solución Rational TestManager para proporcionar soporte para administrar todas las actividades de testing. Es la única solución de automatización que ofrece reutilización y portabilidad de grabación de pruebas a través de plataformas Windows para usar una grabación que se reproduce en todas las plataformas Windows.

Rational Robot está diseñado para ayudarlo a cerrar la brecha de calidad en sus aplicaciones e-business, ERP, y cliente-servidor – la mejora crucial que debe tener una aplicación luego de haberse decidido, planeado y codificado las características, pero antes de ser liberado. Esta mejora solo se logra al encontrar defectos mediante pruebas, y luego removiéndolos. Un gran número de defectos se reflejan en una brecha de calidad más grande; y una brecha de calidad grande implica un gran riesgo de liberación de producto. En cualquier situación, las dispendiosas reparaciones serán costosas y demoradas, y puede resultar en la pérdida de tanto el cliente como de la credibilidad. [34]. **Ver Anexo 16.**

#### ✓ **Rational Robot le permite:**

- Automáticamente encontrar defectos tempranamente.
- Graba y reproduce scripts de pruebas que reconocen los objetos en sus aplicaciones Web, ERP o cliente-servidor
- Rastrea, reporta y grafica toda la información sobre procesos de aseguramiento de calidad en pruebas

#### ✓ **Características y Beneficios Claves**

- Integración con TestManager – Rational TestManager es una poderosa aplicación dentro de la administración del ciclo de vida que mejora la productividad del equipo. Usándolo con Rational Robot le da el control y acceso a todas las actividades y activos de la planeación de pruebas, y unifica a los probadores con el resto del equipo.



- Soporta IE y Netscape – Rational Robot gradua pruebas funcionales sobre browsers Netscape y los reproduce ya sea en el mismo Netscape o en Microsoft Internet Explorer.
- Disminuye el Impacto de los Cambios en el Ambiente – Rational Robot soporta una gran variedad de ambientes y lenguajes, incluyendo HTML, y DHTML, Java, Microsoft Visual Basic y Visual C++, Oracle Developer / 2000, PeopleSoft, SAP, y Sybase PowerBuilder.
- Alcance Todo el Poder de un Ambiente de Programación Integrado – Rational Robot genera scripts de pruebas en SQABasic, un ambiente de encriptamiento MDI integrado que le permite ver y editar sus scripts de pruebas mientras usted lo está grabando.
- Reduzca la Curva de Aprendizaje – Rational Robot usa tecnología de grabación orientada a objetos para proporcionar la creación de pruebas de la manera más fácil e intuitiva con las curvas de aprendizaje más cortas.
- Analice los Problemas Rápidamente – El Rational Robot automáticamente ingresa los resultados de pruebas al repositorio integrado Rational, los cuales están codificados por colores, para lograr un análisis visual rápido. Al hacer doble clic en un ingreso, Rational Robot lo lleva directamente a la línea correspondiente en su script, por lo tanto asegurando, análisis y correcciones de errores.
- Optimiza WebSites – Entregado como parte del Rational Robot, el Rational SiteCheck proporciona un test funcional de un solo clic para todo su WebSites, el cual permite detectar y reparar problemas, analizar desempeño, y maximizar los beneficios para los visitantes de su WebSites.
- Reducir Costos – Rational Robot asegura que el mismo script puede probar, sin modificación, una aplicación en Microsoft Windows 2000, Windows 98, Windows 95, o Windows NT, por lo tanto eliminando costos innecesarios.
- Personalización de Controles y Objetos – Rational Robot le permite probar cada componente de la aplicación bajo varias condiciones y proporciona casos de pruebas para menús, listas, caracteres alfanuméricos, bitmaps, y muchos otros objetos.
- Tecnología de Grabación Orientada a los Objetos – Proporciona la creación de pruebas más intuitiva y rápida con la curva de aprendizaje más corta.

## ✓ Sistemas Operativos y Plataformas de Hardware Apropriadas

- Windows 2000
- Windows 95/98
- Windows NT
- Windows XP

### 3.2.2.1 Ejecución de prueba de regresión al Registro de Actividades Diarias (RAD)

**Ejemplo:** RAD, Registro de Actividades Diarias.

Este ejemplo está grabado sobre la funcionalidad Nueva Actividad, dentro de la opción Otras actividades del EBS.

**Nombre del caso de estudio:** CP\_Tesis\_Rad\_NuevaActv\_EBT.

Se hizo la prueba sobre la opción Seleccionar, donde aparece la ventana de seleccionar los datos del personal de la salud asociado, luego de ser seleccionado con los datos No Registro: 38489, Nombre del médico: Maikel, Primer Apellido: Suárez, Segundo Apellido: Corrales, estos datos deben haberse insertado en los campos asociados requeridos para la nueva actividad.

Opciones de Menú	
Actividades Asistenciales	
Grupo Básico de Trabajo	
Cuerpo de Guardia	
Otras Actividades EBS	
Actividades	
Reportes	
Reunión Mensual GBT	
Entrega de Guardia	

Nuevo - Otras actividades EBS	
Nº Registro	38489 * Seleccionar
Nombre del Médico	*
Primer Apellido	*
Segundo Apellido	Corrales *
Fecha	2008-06-05 *
Actividad	Reuniones *
Lugar	docente 1 *
Tiempo	8 * (horas)
Observaciones	

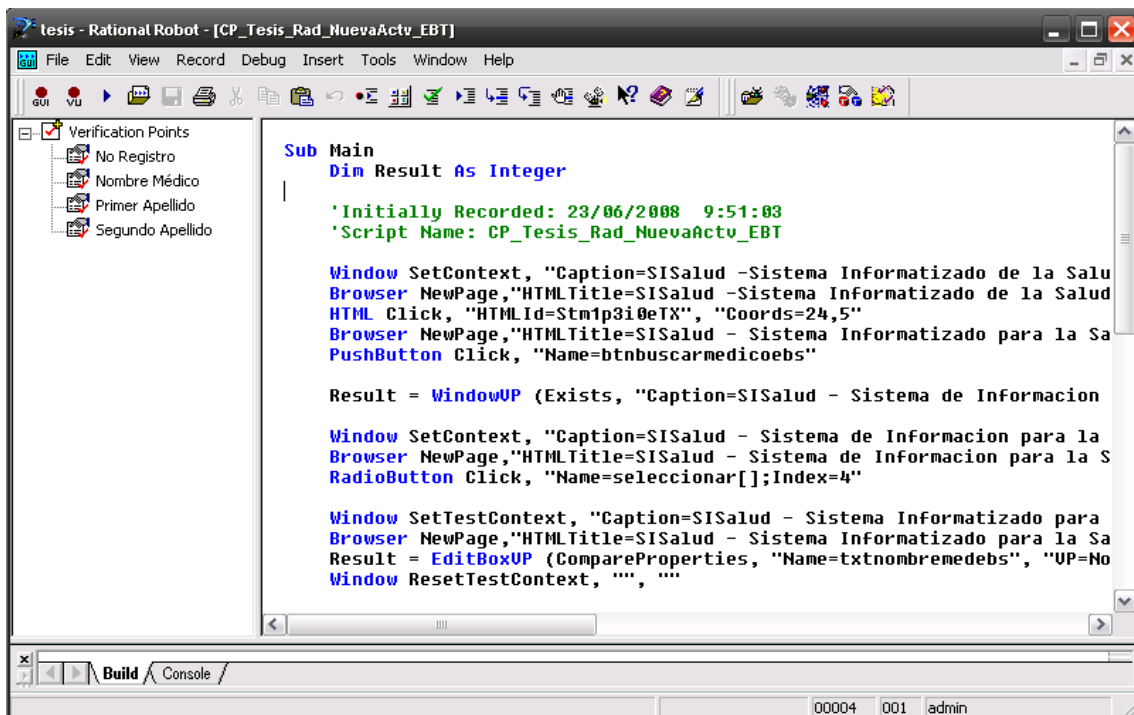
(\*) Los campos señalados son de entrada obligatoria.

Aceptar Cancelar

Tabla 3.4 Pantalla de la aplicación Registro de Actividades Diarias.

## Propuesta de Procedimiento y Herramientas

Al correr la aplicación en la grabación, se han insertado puntos de verificación como Windows Existence, para comprobar que al marcar la opción Seleccionar de la aplicación, sale la ventana de Personal de la Salud. Además se insertaron puntos de verificación para comprobar si son insertados en los campos No registro, Nombre del Médico, Primer Apellido y Segundo Apellido, los datos seleccionados en la tabla de Personal de la salud. Luego se ha generado el script de la prueba, apareciendo en el Rational Robot el script de todas las acciones realizadas sobre la aplicación en prueba.



```
Sub Main
  Dim Result As Integer

  'Initially Recorded: 23/06/2008 9:51:03
  'Script Name: CP_Tesis_Rad_NuevaActv_EBT

  Window SetContext, "Caption=SISalud -Sistema Informatizado de la Salu
  Browser NewPage,"HTMLTitle=SISalud -Sistema Informatizado de la Salud
  HTML Click, "HTMLId=Stm1p3i0eTX", "Coords=24,5"
  Browser NewPage,"HTMLTitle=SISalud - Sistema Informatizado para la Sa
  PushButton Click, "Name=btbuscarmedicoebs"

  Result = WindowUP (Exists, "Caption=SISalud - Sistema de Informacion

  Window SetContext, "Caption=SISalud - Sistema de Informacion para la
  Browser NewPage,"HTMLTitle=SISalud - Sistema de Informacion para la S
  RadioButton Click, "Name=seleccionar[];Index=4"

  Window SetTestContext, "Caption=SISalud - Sistema Informatizado para
  Browser NewPage,"HTMLTitle=SISalud - Sistema Informatizado para la Sa
  Result = EditBoxUP (CompareProperties, "Name=txtnombredebs", "UP=No
  Window ResetTestContext, "", ""
```

Tabla 3.5 Rational Robot, script de prueba grabada.

Cuando se realizaron cambios en la aplicación, se corrió el script para verificar si se había afectado la función de Nueva Actividad, generándose en el Test Manager, los siguientes resultados.

Event Type	Result	Date & Time	Failure
Computer Start	Fail	23/06/2008 9:59:30	
Script Start (CP_Tesis_Rad_NuevaActv_EBT)	Fail	23/06/2008 9:59:30	
Verification Point (Window Existence - Window Existence)	Pass	23/06/2008 9:59:35	
Verification Point (Nombre Médico - Object Properties)	Fail	23/06/2008 9:59:37	
Verification Point (Primer Apellido - Object Properties)	Fail	23/06/2008 9:59:37	
Verification Point (Segundo Apellido - Object Properties)	Pass	23/06/2008 9:59:38	
Verification Point (No Registro - Object Properties)	Pass	23/06/2008 9:59:39	
Script End (CP_Tesis_Rad_NuevaActv_EBT)	Fail	23/06/2008 9:59:44	
Computer End	Fail	23/06/2008 9:59:44	

Tabla 3.6 Test Manager, resultados de la reproducción del script.

Los resultados de script han sido fallidos, lo que significa que han ocurrido variaciones en la aplicación que han producido que algunos de los puntos de verificación no hayan coincidido. Los puntos de verificación cuyo resultado ha sido Fail, se analizarán pues es donde se han proporcionado los errores. El punto de verificación escogido para la comparación fue el Object Properties, que permite comparar campos como editBox. Se puede citar el caso del punto de verificación Primer Apellido, donde debería aparecer en el campo citado, Suárez, que es el primer apellido del personal seleccionado, pero como ha fallado, entonces este campo ha salido vacío. Este resultado puede ser observado, dando clic derecho sobre el punto de verificación primer apellido, y seleccionando la opción View Verification Point, donde se mostrará el punto de partida de la comparación y los resultados actuales.

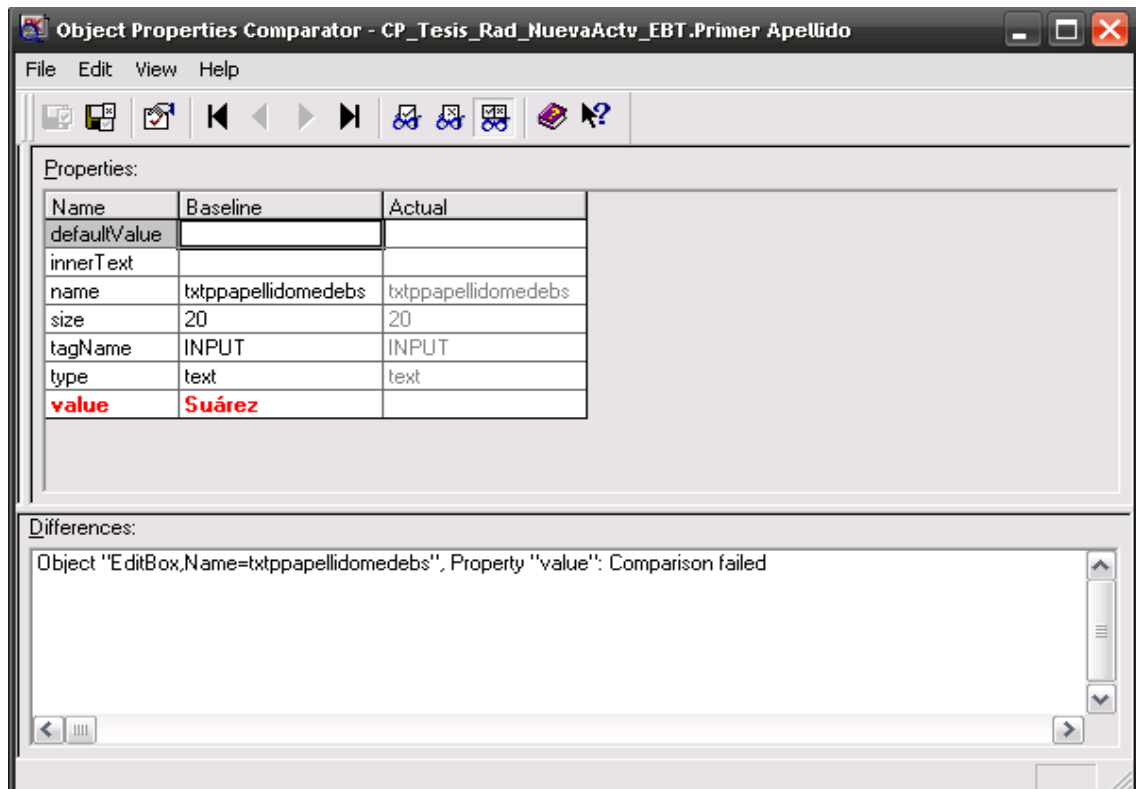


Tabla 3.7 Comparador de propiedades del objeto analizado.

En esta ventana se puede se muestra los datos del punto de verificación, mostrando las propiedades del campo analizado y comparando los resultados obtenidos, donde se puede observar que en el campo Primer Apellido, no ha sido insertado el dato seleccionado, lo que ha provocado que el punto de verificación no haya sido pasado como exitoso.

En el presente capítulo se ha descrito la propuesta de actividades a seguir para asegurar la calidad del flujo de trabajo de prueba en el área temática APS. Se proponen 11 actividades que se deben llevar a cabo, para la obtención de productos con calidad. Además se hace un resumen de dos herramientas muy importantes para el flujo de prueba, como son JMeter y Robot, así como su uso.

### CONCLUSIONES

Luego de la investigación realizada sobre normas y modelos de calidad existentes y en aras de apoyar el aseguramiento de la calidad del proceso de prueba en el área temática de APS, así como el cumplimiento de los objetivos y tareas proyectados, se concluye que:

- Se realizó una investigación detallada acerca de los modelos y normas de calidad existentes hasta el momento, exponiendo los principales conceptos asociados a dicho tema y las mejores prácticas de calidad de dichos modelos.
- Se investigó sobre el flujo de trabajo de prueba que se aplica en el área temática de APS, así como el estado de funcionamiento de este flujo en el proyecto y la calidad con que es desarrollado el proceso por sus trabajadores.
- Se desarrolló un procedimiento, que cuenta con un grupo de actividades para asegurar la calidad del flujo de trabajo de prueba del área temática de APS, basado en las mejores prácticas de CMMI e ISO.
- Se cuenta con una propuesta de herramientas como el JMeter y el Rational Robot para pruebas de stress y de regresión respectivamente, así como listas de chequeo para los artefactos de salida del flujo de prueba estudiado.

### RECOMENDACIONES

Una vez cumplido con el objetivo general propuesto, se recomienda:

- Desarrollar un procedimiento de aseguramiento de la calidad para los demás flujos de trabajo del ciclo de desarrollo del software.
- Estudiar utilidades de la herramienta JMeter para pruebas a Bases de Datos.
- Estudiar herramientas asociadas al Rational Robot e investigar sobre el Test Manager.
- Se recomienda para el uso de Rational Robot que se cree una plantilla donde se registre los nombres de los Script de casos de prueba con fácil acceso a la hora de trabajar con esta herramienta.
- Se sugiere aplicar este procedimiento y evaluarlo usando métricas de evaluación de procesos.
- Se propone que el procedimiento sea publicado, aprobado y fácilmente accesible y el proceso definido debe ser usado en las tareas habituales llevadas a cabo por el área temática Atención Primaria de la Salud de la Facultad 7, para tener un seguimiento y constancia en el mantenimiento de la calidad de los procesos de prueba que se están desarrollando en el mismo.
- Se recomienda que se pongan a funcionar en el proyecto 4 o más máquinas, para llevar a cabo las pruebas de stress a los proyectos desarrollados en el área temática de APS, donde se puedan simular mayor cantidad de usuarios.

### REFERENCIAS BIBLIOGRÁFICAS

- [1]. **Autores, Colectivo de. 2005.** Mirar a Cuba. *www.cubaminrex.cu*. [En línea] 2005. [Citado el: 20 de enero de 2008.]  
[http://www.cubaminrex.cu/Mirar\\_Cuba/Ciencia/ct\\_Programa%20sobre%20la%20informatizaci%F3n\\_tc.htm](http://www.cubaminrex.cu/Mirar_Cuba/Ciencia/ct_Programa%20sobre%20la%20informatizaci%F3n_tc.htm)
- [2]. **Autores, Colectivo de. 2007.** Informatica Evento Virtual. *www.informaticahabana.com*. [En línea] 2007. [Citado el: 20 de enero de 2008.] [http://www.informaticahabana.com/evento\\_virtual/?q=node/583](http://www.informaticahabana.com/evento_virtual/?q=node/583)
- [3]. **Peña, Carlos Martínez. 2002.** *Departamento de Ciencias Matemáticas e Informáticas*. [En línea] 2002. [Citado el: 26 de enero de 2008.] <http://dmi.uib.es/~bbuades/calidad/sld014.htm>
- [4]. **Dorado, Carlos. 2003.** Ingeniería de Software Educativo. *www.fi.net.ar*. [En línea] 2003. [Citado el: 23 de enero de 2008.] <http://www.fi.net.ar/laboratorios/lsi/c-icie99-ingenieriasoftwareeducativo.pdf>
- [5]. **Pressman, Roger. 2002.** *Ingeniería de Software. Un enfoque Práctico*. 5ta Edición. s.l. : McGraw-Hill, 2002.
- [6]. **Juan Manuel Cueva Lovelle. 1999.** <http://gidis.ing.unlpam.edu.ar>. [En línea] 21 de Octubre de 1999. [Citado el: 26 de Enero de 2008.] [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF)
- [7]. Ídem a la Referencia 3.
- [8]. **Autores Colectivo de. 2004.** Departamento de Educación y Cultura. *SCIENCE*. [En línea] <http://www.science.oas.org>, 2004. [Citado el: 19 de Enero de 2008.]  
[http://www.science.oas.org/OEA\\_GTZ/LIBROS/CTM/cap2\\_ctm.htm](http://www.science.oas.org/OEA_GTZ/LIBROS/CTM/cap2_ctm.htm)
- [9]. Ídem a la referencia 4.
- [10]. Ídem a la referencia 5.
- [11]. **Autores, Colectivo de. 2006.** *UNAD*. [En línea] <http://calidad.unad.org>, 2006. [Citado el: 20 de enero de 2008.] <http://calidad.unad.org/asesoramiento/definicion.html>
- [12]. Ídem a la referencia 5.
- [13]. **Navarro, Antonio. 2006.** [En línea] 13 de enero de 2006. [Citado el: 26 de enero de 2008.]  
[http://www.fdi.ucm.es/profesor/anavarro/9\\_Garantia\\_de\\_calidad\\_del\\_software.pdf](http://www.fdi.ucm.es/profesor/anavarro/9_Garantia_de_calidad_del_software.pdf)
- [14]. **Autores, Colectivo de. 1999.** *eumet.net*. [En línea] 1999. [Citado el: 26 de enero de 2008.]  
<http://www.eumed.net/libros/2008a/351/Problemas%20y%20costos%20del%20Aseguramiento%20de%20la%20Calidad%20en%20el%20Software.htm>
- [15]. **Guerrero, Luciano. 1999.** *www.geocities.com*. [En línea] 1999. [Citado el: 26 de enero de 2008.]  
<http://209.85.215.104/search?q=cache:uYCNmTtfRh4J:www.geocities.com/SiliconValley/Lab/3629/garantia.htm+La+garant%C3%ADa+de+calidad+se+asegura+de+lo+siguiente:&hl=es&ct=clnk&cd=1&gl=cu>
- [16]. Ídem a la referencia 5.



- [17]. Ídem a la referencia 4.
- [18]. Ídem a la referencia 5.
- [19]. **Humphrey, Watts S. 2000.** *Introduction to the Team Software Process*. s.l. : Addison Wesley, 2000.
- [20]. **Rico, Ariel Edgar Serrano. 2003.** [En línea] 2003. [Citado el: 30 de enero de 2008.]  
[http://is.ls.fi.upm.es/doctorado/Trabajos20032004/Ariel\\_Serrano\\_Resumen\(Esp\).pdf](http://is.ls.fi.upm.es/doctorado/Trabajos20032004/Ariel_Serrano_Resumen(Esp).pdf)
- [21]. **2005, Colectivo de Autores. 2005.** *Navegapolis.net*. [En línea] 1 de mayo de 2005. [Citado el: 12 de febrero de 2008.] <http://www.navegapolis.net/content/view/330/60/>
- [22]. **Gracia, Joaquin. 2005.** *www.ingenierossoftware.com*. [En línea] noviembre 26, 2005. [Citado el: 13 de febrero de 2008.] <http://www.ingenierossoftware.com>  
<http://209.85.165.104/search?q=cache:thLkCU5r6t8J:www.ingenierossoftware.com/calidad/cmm-cmmi-nivel-2.php+relacion+cmmi+calidad&hl=es&ct=clnk&cd=3&gl=cu>
- [23]. **Autores, Colectivo de. 2001.** [En línea] 2001. [Citado el: 18 de febrero de 2008.] <http://www.rsc-chile.cl/modules/smartpartner/partner.php?id=39>
- [24]. **de la Villa, Manuel, Ruiz, Mercedes y Ramos, Isabel.** [En línea] [Citado el: 18 de febrero de 2008.]  
[www.sc.ehu.es/jiwdocoj/remis/docs/DelaVillaadis2004.doc](http://www.sc.ehu.es/jiwdocoj/remis/docs/DelaVillaadis2004.doc)
- [25]. **Pereda, Héctor Fernández. 2004.** [En línea] 2004. [Citado el: 20 de febrero de 2008.]  
[http://www.buscarportal.com/articulos/iso\\_9001\\_gestion\\_calidad.html](http://www.buscarportal.com/articulos/iso_9001_gestion_calidad.html)
- [26]. Ídem a la referencia 23.
- [27]. Ídem a la referencia 23.
- [28]. **Autores, Colectivo de . 2006.** *FortuneCity*. [En línea] 2006. [Citado el: 21 de febrero de 2008.]  
<http://campus.fortunecity.com/defiant/114/iso9000.htm>
- [29]. Ídem a la referencia 26.
- [30]. **Autores, Colectivo de . 2006.** *www.wikipedia.org*. [En línea] 12 de marzo de 2006. [Citado el: 23 de febrero de 2008.]  
[http://64.233.167.104/search?q=cache:wmvEz\\_1lZBcJ:es.wikipedia.org/wiki/ISO/IEC\\_12207+objetivos+ISO+12207&hl=es&ct=clnk&cd=1&gl=cu](http://64.233.167.104/search?q=cache:wmvEz_1lZBcJ:es.wikipedia.org/wiki/ISO/IEC_12207+objetivos+ISO+12207&hl=es&ct=clnk&cd=1&gl=cu)
- [31]. **Autores, Colectivo de . 2006.** *http://www.bvindicopi.gob.pe*. [En línea] 13 de julio de 2006. [Citado el: 24 de febrero de 2008.] <http://www.bvindicopi.gob.pe/normas/isoiec12207.pdf>
- [32]. **Autores, Colectivo de . 2007.** *www.wikipedia.org*. [En línea] 3 de abril de 2007. [Citado el: 26 de febrero de 2008.]  
[http://64.233.167.104/search?q=cache:tpgKEYQ3NScJ:es.wikipedia.org/wiki/ISO\\_9126+objetivos+ISO+9126&hl=es&ct=clnk&cd=1&gl=cu&lr=lang\\_es](http://64.233.167.104/search?q=cache:tpgKEYQ3NScJ:es.wikipedia.org/wiki/ISO_9126+objetivos+ISO+9126&hl=es&ct=clnk&cd=1&gl=cu&lr=lang_es)
- [33]. **Ramírez, Yilen Pons. 2007.** *Procedimiento para pruebas de eficiencia bajo carga intensiva en aplicaciones web de salud*. Ciudad de la Habana : UCI, 2007.
- [34]. **Autores, Colectivo de. 2006.** *GSINOVA. www.rational.com.ar*. [En línea] 2006. [Citado el: 10 de mayo de 2008.] <http://www.rational.com.ar/cursos/essentialsoffunctionaltestingwithrationalrobot.html>

## BIBLIOGRAFÍA

1. **Aguilar, José Alfonso. 2007. MYGNET.** [En línea] 8 de febrero de 2007. [Citado el: 10 de febrero de 2008.]  
[http://www.mygnet.net/articulos/software/capability\\_maturity\\_model\\_integration\\_cmimi.984](http://www.mygnet.net/articulos/software/capability_maturity_model_integration_cmimi.984)
2. **Autores, Colectivo de. 1998.** Soluciones en Informática. *www.indudata.com*. [En línea] 1998. [Citado el: 25 de abril de 2008.] [http://www.indudata.com/1rational\\_robot.htm](http://www.indudata.com/1rational_robot.htm)
3. **Autores, Colectivo de. 1999.** The Apache Jakarta Project. *http://jakarta.apache.org*. [En línea] 1999. [Citado el: 20 de abril de 2008.] <http://jakarta.apache.org/jmeter/>
4. **Autores, Colectivo de. 1999.** *eumet.net*. [En línea] 1999. [Citado el: 26 de enero de 2008.] <http://www.eumed.net/libros/2008a/351/Problemas%20y%20costos%20del%20Aseguramiento%20de%20la%20Calidad%20en%20el%20Software.htm>
5. **Autores, Colectivo de. 2002.** International Organization Standardization. *www.iso.org*. [En línea] 2002. [Citado el: 2 de marzo de 2008.] <http://www.iso.org/iso/home.htm>
6. **Autores, Colectivo de . 2004.** *Designing processes to your needs*. [En línea] 2004. [Citado el: 18 de febrero de 2008.] [www.synspace.com/ES/Seminars/spice.html](http://www.synspace.com/ES/Seminars/spice.html)
7. **Autores, Colectivo de. 2005.** Osmosis Latina. *www.osmosislatina.com*. [En línea] 20 de octubre de 2005. [Citado el: 12 de abril de 2008.] <http://www.osmosislatina.com/jmeter/basico.htm>
8. **Autores, Colectivo de . 2006.** Dirección de Calidad de Software. *http://calidadsoft.prod.uci.cu*. [En línea] 2006. [Citado el: 1 de marzo de 2008.] <http://calidadsoft.prod.uci.cu>
9. **Brualla, Cecilia Rigoni. 2004.** *www.mityc.es*. [En línea] 2004. [Citado el: 6 de marzo de 2008.] <http://www.mityc.es/NR/ronlyres/A570B90C-B41A-46E2-BD39-4A31D18BB7FD/0/s01CeciliaRigoni.pdf>
10. **Cabreras Arribas, Yoan Manuel y González Alonso, Yosbel Ernesto. 2007.** *Proceso de Pruebas del Registro de Áreas de Salud de la Atención Primaria del Sistema de Información para la Salud*. Universidad de las Ciencias Informáticas. Ciudad Habana : s.n., 2007.
11. **Cedeño González, Fabián Felipe y Carralero Mulet, Raúl. 2007.** *Propuesta de proceso de pruebas basado en RUP para proyectos de Gestión*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.
12. **Dorado, Carlos. 2003.** Ingeniería de Software Educativo. *www.fi.net.ar*. [En línea] 2003. [Citado el: 23 de enero de 2008.] <http://www.fi.net.ar/laboratorios/lsi/c-icie99-ingenieriasoftwareeducativo.pdf>
13. **Ema, Ernesto. 2002.** Soluziona. *www.soluziona.com*. [En línea] 2002. [Citado el: 22 de marzo de 2008.]  
[http://www.soluziona.com/htdocs/areas/cyma/de\\_interes/articulos/gestion\\_convencional.shtml](http://www.soluziona.com/htdocs/areas/cyma/de_interes/articulos/gestion_convencional.shtml)

14. **Figuroa, María Antonieta Abud. 2004.** Calidad en la Industria de Software. *www.revistaupiicsa.20m.com*. [En línea] 2004. [Citado el: 14 de abril de 2008.] <http://www.revistaupiicsa.20m.com/Emilia/RevEneAbr04/Antonieta1.pdf>
15. **González, Carlos D. 2007.** *www.usabilidadweb.com.ar*. [En línea] junio de 2007. [Citado el: 26 de marzo de 2008.] [http://www.usabilidadweb.com.ar/metodos\\_eval\\_calidad\\_web.php](http://www.usabilidadweb.com.ar/metodos_eval_calidad_web.php)
16. **Gracia, Joaquin. 2005.** *www.ingenierossoftware.com*. [En línea] noviembre 26, 2005. [Citado el: 13 de febrero de 2008.] <http://www.ingenierossoftware.com>
17. **Guerrero, Luciano. 1999.** *www.geocities.com*. [En línea] 1999. [Citado el: 26 de enero de 2008.] <http://209.85.215.104/search?q=cache:uYCNmTfRh4J:www.geocities.com/SiliconValley/Lab/3629/garantia.htm+La+garant%C3%ADa+de+calidad+se+asegura+de+lo+siguiente:&hl=es&ct=clnk&cd=1&gl=cu>
18. **Hernando, Sergio. 2004.** Seguridad de la Información y Auditorías de Sistemas. *www.sahw.com*. [En línea] 11 de febrero de 2004. [Citado el: 16 de marzo de 2008.] <http://www.sahw.com/wp/archivos/2004/07/07/iso-900032004-calidad-iso-90012000-en-la-ingeniera-del-software/>
19. **Humphrey, Watts S. 2000.** *Introduction to the Team Software Process*. s.l. : Addison Wesley, 2000.
20. **Juan Manuel Cueva Lovelle. 1999.** <http://gidis.ing.unlpam.edu.ar>. [En línea] 21 de Octubre de 1999. [Citado el: 26 de Enero de 2008.]
21. **Navarro, Antonio. 2006.** [En línea] 13 de enero de 2006. [Citado el: 26 de enero de 2008.] [http://www.fdi.ucm.es/profesor/anavarro/9\\_Garantia\\_de\\_calidad\\_del\\_software.pdf](http://www.fdi.ucm.es/profesor/anavarro/9_Garantia_de_calidad_del_software.pdf)  
[http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF)
22. **Peña, Carlos Martínez. 2002.** *Departamento de Ciencias Matemáticas e Informáticas*. [En línea] 2002. [Citado el: 26 de enero de 2008.] <http://dmi.uib.es/~bbuades/calidad/sld014.htm>
23. **Pereda, Héctor Fernández. 2004.** [En línea] 2004. [Citado el: 20 de febrero de 2008.] [http://www.buscarportal.com/articulos/iso\\_9001\\_gestion\\_calidad.html](http://www.buscarportal.com/articulos/iso_9001_gestion_calidad.html)
24. **Pressman, Roger. 2002.** *Ingeniería de Software. Un enfoque Práctico*. 5ta Edición. s.l. : McGraw-Hill, 2002.
25. **Ramírez, Eduardo. 2004.** BoletínTress. *www.tress.com.mx*. [En línea] agosto de 2004. [Citado el: 28 de febrero de 2008.] <http://www.tress.com.mx/boletin/agosto2004/calidad.htm>
26. **Ramírez, Yilen Pons. 2007.** *Procedimiento para pruebas de eficiencia bajo carga intensiva en aplicaciones web de salud*. Ciudad de la Habana : UCI, 2007.
27. **Rico, Ariel Edgar Serrano. 2003.** [En línea] 2003. [Citado el: 30 de enero de 2008.] [http://is.ls.fi.upm.es/doctorado/Trabajos20032004/Ariel\\_Serrano\\_Resumen\(Esp\).pdf](http://is.ls.fi.upm.es/doctorado/Trabajos20032004/Ariel_Serrano_Resumen(Esp).pdf)
28. **Robbins, Jason. 2003.** *readysset.tigris.org*. [En línea] 2003. [Citado el: 25 de abril de 2008.] <http://readysset.tigris.org/nonav/es/templates/qa-plan.html>

29. **Tabasco Reyes, Perla Mailen y Marrero Machín, Yadira. 2007.** *Procedimiento para el control y aseguramiento de la calidad en los flujos trabajo Modelación del Negocio y Requerimientos de los proyectos de software de la Facultad 2 de la Universidad de las Ciencias Informáticas.* Universidad de las Ciencias Informaticas. Ciudad Habana : s.n., 2007.
30. **Ulloa, Juan Pablo Cruz. 2005.** *pgi.umsa.bo.* [En línea] 14 de agosto de 2005. [Citado el: 23 de febrero de 2008.]  
[http://pgi.umsa.bo/enlaces/investigacion/pdf/INGSW3\\_60.pdf?PHPSESSID=f22bb43f2cfe70ddbfcfa5ff6ad987c0](http://pgi.umsa.bo/enlaces/investigacion/pdf/INGSW3_60.pdf?PHPSESSID=f22bb43f2cfe70ddbfcfa5ff6ad987c0)

## ANEXOS

### Anexo 1 Plan de Aseguramiento de la Calidad

#### 1. Introducción

##### 1.1 Propósito

*[Resumen del propósito del Plan de Calidad]*

##### 1.2 Alcance

*[Proyectos con los que se involucra el Plan]*

##### 1.3 Definiciones, Acrónimos y Abreviaturas

*[Definiciones, acrónimos y abreviaturas utilizadas en el documento]*

##### 1.4 Referencias

*[Lista de documentos a los que se hace referencia en el Plan de Aseguramiento de la Calidad.]*

Título	Fecha	Autor	Ubicación (anexo, documento, etc.)
<i>Plan de pruebas</i>			
<i>Plan de desarrollo del proyecto</i>			
<i>Plan de Gestión de Configuración</i>			
<i>Plan de gestión de riesgos</i>			

##### 1.5 Resumen

*[Resumen de los aspectos del plan]*

##### 1. Objetivos de Calidad

*[Incluir los requerimientos de proyecto que están alineados con los requerimientos de calidad]*

##### 2. Gestión

##### 3.1 Organización

*[Se describe la estructura de la organización. Especificar cada uno de los responsables de la calidad]*

### 3.2 Tareas y responsabilidades

<b>Tarea de Aseguramiento de calidad</b>	<b>Precondición</b> <i>Al finalizar la fase:</i>	<b>Pos condición</b> <i>Antes de la fase:</i>	<b>Responsable</b>	<b>Comentarios</b>
<i>Prueba1</i>				
<i>Auditoría 1</i>			<i>Dirección de Calidad</i>	
<i>Revisión 1</i>			<i>Dirección de Calidad</i>	
<i>Revisión 2</i>			<i>Dirección de Calidad</i>	
<i>...</i>			<i>Dirección de Calidad</i>	
<i>Certificación final del producto</i>	<i>Prueba</i>	<i>Implantación</i>	<i>Dirección de Calidad</i>	
<i>Certificación por parte del Cliente</i>	<i>Certificación final del producto</i>			
<i>Evaluación de la satisfacción del cliente</i>	<i>Un mes después de la implantación del producto</i>			

### 3. Documentación

*[Lista de los documentos utilizados por el Plan de Calidad]*

### 4. Métricas

*[En esta sección se describe todo el proceso de métricas que se realizará como producto del monitoreo del trabajo]*

### 5. Estándares y Guías

*[Lista de los estándares y guías utilizados por el Plan de Calidad]*

<b>Estándar</b>	<b>Ubicación</b>	<b>Comentarios</b>
<i>Codificación en j2ee</i>		
<i>Codificación en .NET</i>		
<i>Modelo del Negocio</i>		
<i>Interfaz de Usuario</i>		
<i>Pruebas</i>		
<i>Manuales</i>		

## **6. Plan de Revisiones y Auditorías**

### **7.1 Tareas generales de Revisiones y Auditorías**

*[Describa brevemente cada tipo de revisión y auditoría que se llevará a cabo en el proyecto. Para cada tipo, identifique los artefactos del proyecto que serán el asunto de la revisión o auditoría. Éstos pueden incluir Revisiones Técnicas y de Gestión Conjuntas entre Cliente y Desarrollador, Revisiones y Auditorías de Proceso, Auditorías de Cliente, Revisiones Internas, Técnicas y de Gestión.]*

### **7.2 Cronograma**

*[Detalle aquí el cronograma para las revisiones y auditorías. Este debe incluir las revisiones y auditorías programadas en las fechas principales del proyecto, así como revisiones que son provocadas por la entrega de artefactos del proyecto. Esta subsección puede referenciar el proyecto o el plan de iteración.]*

### **7.3 Organización y Responsabilidades**

*[Liste aquí los grupos específicos o individuos a ser involucrados en cada una de las actividades de revisión y de auditoría identificadas. Describa brevemente las tareas y responsabilidades de cada uno. También, liste cualquier agencia externa que se espera que apruebe o regule cualquier producto del proyecto.]*

### **7.4 Resolución de problemas y actividades de corrección**

*[Esta subsección describe los procedimientos para informar y manejar problemas identificados durante las revisiones y auditorías del proyecto. El Plan de Resolución de Problemas puede ser referenciado]*

### **7.5 Herramientas, técnicas y Metodologías**

*[Describa aquí las herramientas, técnicas o metodologías específicas que serán usadas para llevar a cabo las actividades de revisión y de auditoría identificadas en este plan. Usted debe describir el proceso explícito a ser seguido para cada tipo de revisión o auditoría. Su organización puede tener un Manual de Procedimientos de Revisión y de Auditoría estándar que puede ser referenciado. Estas descripciones de los procedimientos también deben dirigir la recolección, almacenamiento y archivado de los Registros de Revisión del proyecto. Deben describirse las listas de chequeo a utilizar en cada revisión y los atributos de calidad que serán abordados en cada una de ellas.]*

NOTAS:

*[Un grupo de revisiones y auditorías sugeridas a usar como base para la planificación son:*

*Revisión de los Requerimientos (se corresponde con la tradicional Revisión de las Especificaciones del Software)*

- *Revisión de la Arquitectura (se corresponde con la tradicional Revisión del Diseño Preliminar)*
- *Revisión del Diseño (se corresponde con la tradicional Revisión del Diseño Crítico)*
- *Auditoría de la configuración funcional (para verificar que todos los requerimientos han sido cumplidos)*

- Auditoría de la configuración física (para verificar que el software y su documentación están completos y listos para entregar)
- Auditoría del Proceso
- Revisión del Proceso
- Revisión Administrativa (Revisión de Aprobación del Proyecto, Revisión de la Planificación del Proyecto, Revisión del Plan de Iteración)
- Revisiones Post-mortem (Revisión de Aceptación de la Iteración, Revisión de Aceptación del Proyecto).

Pueden utilizar la siguiente tabla para hacer corresponder las actividades de las revisiones con el plan de desarrollo del proyecto

No. de Revisión	Tipo	Objetivos	Descripción (Iteración)	Fase del Proyecto	Responsable

## 7. Pruebas y Evaluación

[Se hace referencia al plan de pruebas]

## 8. Herramientas, Técnicas y Metodologías

[Lista de todas las herramientas, técnicas y metodologías utilizadas en las actividades del Plan de Calidad]

## 9. Resolución de Problema y Acción Correctiva

[Esta sección referencia el Plan de Resolución de Problema.]

## 10. Gestión de Configuración

Referencia al Plan de Gestión de Configuración

## 11. Registros de Calidad

[Descripciones de varios registros de calidad que se mantendrán durante el proyecto, incluyendo cómo y dónde cada tipo de registro se guardará y por cuánto tiempo.]

## 12. Entrenamiento

[Listado de las actividades de entrenamiento necesarias para que el equipo de proyecto ejecute las actividades del Plan de Aseguramiento de la Calidad]



**Anexo 2 Acta de Revisión**

**Título: Revisión**

*[Nota: La siguiente plantilla se ha desarrollado para la revisión que realizan el Grupo SQA de cada proyecto de la facultad. El texto que se encuentra entre corchetes y de color azul se ha incluido para proporcionar una guía para el autor y se debería borrar antes de la entrega del documento.]*

*[ En el caso de referirse a una Revisión Técnica Formal al inicio del documento debe sustituirse Tipo de Revisión por las siglas RTF , para otra revisión debe eliminarse este texto]*

<b>REDACTADO POR : (*)</b>	{*}	[*]	
<b>REVISADO POR : (*)</b>	{*}	[*]	
<b>APROBADO POR: (*)</b>	{*}	[*]	

(\*) Aquí se pone el nombre de la persona encargada de cada trabajo, {\*} firma de la persona, [\*] fecha de elaboración.

**Relación de las Revisiones**

*[Se debe llevar el control de cada revisión hasta que se eliminen todos los defectos encontrados]*

REV.	FECHA	ARTEFACTOS AFECTADOS	REDACTADO	REVISADO	APROBADO
0					
1					

## Índice

### Artefacto revisado

#### 1.1. Nombre y versión del Producto revisado

*[Se debe especificar:*

*Nombre y versión (si corresponde) del artefacto revisado*

*Fase:*

*Iteración:*

*Que corresponda]*

#### 1.2. Participantes de la revisión

*[Nombre y rol de los participantes de la revisión.]*

#### 1.3. Técnica utilizada

*[Forma en que se hizo la revisión, por Ej. Lista de chequeo definida en el Plan de SQA para la revisión de productos, entradas, salidas y descripción de actividades.]*

**1.4. Auditado:** SI:  NO:

*[Se debe marcar con una X la casilla SI, si la revisión corresponde a una tarea de auditoría del proceso de desarrollo, en caso contrario se debe marcar con una X la casilla NO.]*

### 2. Objetivos de la Revisión

*[Se plantean brevemente los aspectos del artefacto que serán revisados, que propiedades de la calidad se buscará que cumpla y en que grado, que principios y estándares de calidad se le aplican al artefacto, si ya hubo revisión de otra versión del artefacto que correcciones quedaron pendientes de realizar. De la misma forma se establecen los aspectos del Proceso - en lo que tiene que ver con la elaboración del artefacto - que serán revisados con los participantes de la Revisión. Esta información debe coincidir con la que se incluyó en la convocatoria para la realización de la Revisión enviada a los participantes.]*

### 3. Problemas detectados

*[Se describen en forma detallada los problemas detectados tanto en el artefacto como en el proceso de elaboración del mismo, y se plantean las sugerencias de corrección para la próxima versión del Producto.]*

#### 3.1. Problema detectado 1

*[Se describe el problema detectado estableciendo sus características, gravedad y especificando su ubicación.]*

##### 3.1.1. Causa

*[Se detalla la causa del problema o defecto detectado]*

**3.1.2. Sugerencia de corrección**

*[Se sugieren las correcciones a realizar para que el artefacto cumpla con los principios y estándares de calidad establecidos y los procedimientos definidos.]*

**3.2. Problema detectado 2**

**4. Conclusiones.**

**4.1. Evaluación.**

*[Se realiza una evaluación global del artefacto revisado de acuerdo a los Objetivos planteados y los problemas detectados.]*

**4.2. Estado actual del artefacto.**

*[Se describe el estado actual del artefacto por Ej. se debe rehacer o corregir, se puede entregar.]*

**4.3. Acciones correctivas**

*[Se detallan las acciones a tomar para que el artefacto sea corregido y revisado en una nueva versión (en caso de ser requerido).]*

**4.4. Próxima Revisión del Artefacto**

*[Se establece la próxima revisión de acuerdo a los puntos detallados anteriormente, en términos de fechas o Fase e iteración.]*

**Anexo 3 Registro de Revisión**

Proyecto: \_\_\_\_\_ Código: \_\_\_\_\_

No.	RTF	Fase	Iteración	Fecha	Producto	Observaciones

**Anexo 4 Registro de Auditorías**

No	Proyecto	Cód.	Fecha	Responsable	Tipo Aud.		Producto	Observaciones	Evaluación	No Ref.
					RTF	Rev.				

**Anexo 5 Lista de Desviaciones**

Proyecto: \_\_\_\_\_

No	Desviación	Observaciones
<i>Número según la desviación encontrada</i>	<i>Desviaciones encontradas durante las actividades de revisiones y auditorías</i>	<i>Observaciones sobre las desviaciones expuestas.</i>

**Anexo 6 Lista de defectos.**

Error	Nombre Caso Prueba	Descripción
<i>Se pone el error encontrado.</i>	<i>Se especifica el caso de prueba en el que se ha detectado el error.</i>	<i>Se describe las especificaciones del error.</i>

**Anexo 7 Lista de chequeo Plan de prueba.**

Nivel	Criterio de evaluación	Evaluación		N.P	Observaciones	%cumplido
		SI	NO			
	¿El plan de prueba identifica el alcance del esfuerzo de prueba?					
	¿El plan de prueba identifica los artefactos a generar?					
	¿Cada requisito de proyecto tiene asociado (como decir en los casos de uso) tiene por lo menos un requisito asociado para la prueba o una declaración que justifica por qué no es un requisito para la prueba?					

	¿Todos los requisitos para la prueba han sido identificados y priorizados para cada uno de las diferentes clases de pruebas a ser implementados y ejecutados?					
	¿Existe una estrategia de prueba clara y concisa documentada para cada tipo de prueba a ser implementado y ejecutado?					
	¿Todos los recursos necesarios para implementar y ejecutar la prueba con éxito han sido identificados? Incluyendo equipo físico, software, y personal.					
	El plan de prueba contiene un programa o la lista de acontecimientos que identifican el proyecto y las actividades de prueba descritas (principio y fechas de final).					

**Anexo 8 Lista de chequeo Casos de prueba.**

Nivel	Criterio de evaluación	Evaluación		N.P	Observaciones	%cumplido
		SI	NO			
	¿Se ha hecho una descripción del caso de uso, objetivo de prueba, o la condición que está					

<p>valorado ha sido dicha evidentemente para cada prueba?</p>							
<p>¿Cada prueba dice el resultado esperado y el método de valorar el resultado?</p>							
<p>¿Todos los nombres de pruebas y/o identificaciones son compatibles con la convención para dar nombre del artefacto de prueba?</p>							
<p>¿Las pruebas han sido identificadas para ejecutar todos los comportamientos de los requisitos del producto?</p> <ul style="list-style-type: none"> <li>- Función.</li> <li>- Validación de datos.</li> <li>- Implementación y reglas del negocio.</li> <li>- Flujo de trabajo o control de las metas de prueba.</li> <li>- Flujo de datos.</li> <li>- Estado de objetos.</li> <li>- El rendimiento (incluyendo trabajo, configuración, y la tensión).</li> <li>- Seguridad y accesibilidad.</li> </ul>							
<p>¿Cada caso de prueba o grupo relacionado de pruebas, identifica el objetivo inicial de la prueba y el estado actual de los datos de prueba?</p>							
<p>¿Cada caso de prueba tiene dos tipos de pruebas una para</p>							

	verificar el comportamiento esperado y otra para condiciones inesperadas?					
--	---	--	--	--	--	--

**Anexo 9 Lista de chequeo Defectos.**

Nivel	Criterio de evaluación	Evaluación		N.P	Observaciones	%cumplido
		SI	NO			
	¿Se han identificado correctamente los defectos encontrados?					
	¿Se hace una correcta descripción de cada error encontrado?					
	¿Los defectos encontrados se corresponden con el caso de prueba especificado?					

**Anexo 10 Lista de chequeo Componente de pruebas.**

Nivel	Criterio de evaluación	Evaluación		N.P	Observaciones	%cumplido
		SI	NO			
	¿Se ha automatizado uno o varios de los procedimientos de prueba?					
	¿Se han utilizado herramientas para automatizar las pruebas?					

**Anexo 11 Lista de chequeo Procedimiento de Prueba (artefacto de salida, del Análisis y Diseño de pruebas).**

Nivel	Criterio de evaluación	Evaluación		N.P	Observaciones	%cumplido
		SI	NO			
	¿Existen para cada requisito funcional del sistema uno o más casos de uso asociados?					
	¿Se han expuesto todos los escenarios de casos de prueba en el diseño de casos de prueba?					
	¿Se han implementado diseños de prueba para requisitos no funcionales?					

**Anexo 12 Lista de chequeo Evaluación de las pruebas.**

Nivel	Criterio de evaluación	Evaluación		N.P	Observaciones	%cumplido
		SI	NO			
	¿Se evalúan los esfuerzos de las pruebas en una iteración, según la comparación de los resultados de las pruebas con lo esbozado en el Plan de prueba?					
	¿Esta evaluación permite reutilizar las actividades del actual ciclo de pruebas en ciclos posteriores?					

**Anexo 13 Entrevista realizada a estudiantes y profesores del proyecto APS (Atención Primaria de la Salud).**

La presente entrevista pretende conocer el estado de la calidad del software en el flujo de trabajo de prueba.

Nombre \_\_\_\_\_

Ocupación: \_\_\_\_\_



**Calidad del software**

1. ¿Los integrantes del proyecto ponen en práctica lo planteado por el PSP?

Si \_\_\_ No \_\_\_

¿Cómo? \_\_\_\_\_

2. ¿Tiene definido algún Plan de Aseguramiento de la Calidad?

Si \_\_\_ No \_\_\_

3. ¿Utiliza algún estándar para este Plan de Aseguramiento de la Calidad?

Si \_\_\_ No \_\_\_

¿Cuál?

CMMI \_\_\_ nivel \_\_\_

SPICE \_\_\_

CMM \_\_\_ nivel \_\_\_

ISO \_\_\_ # \_\_\_

4. ¿Cuándo desarrolla el Plan de Aseguramiento de Calidad del software, antes, durante o después de desarrollar el producto?

Antes \_\_\_ Durante \_\_\_ Después \_\_\_

¿Por qué? \_\_\_\_\_

5. ¿Cada que período de tiempo desarrolla una Revisión Técnica Formal? \_\_\_\_\_

6. ¿Tiene definida alguna Política de Calidad?

Si \_\_\_ No \_\_\_

7. ¿Evalúa objetivamente los procesos?

Si \_\_\_ No \_\_\_

8. ¿Se comunica y se garantiza la resolución de las no-conformidades (fallas o defectos)?

Si \_\_\_ No \_\_\_

9. ¿Mediante que acciones controla la calidad?

\_\_\_ **Inspecciones** (Es una técnica de revisión que se lleva a cabo antes de que se implemente el programa)

\_\_\_ **Revisiones** (analizan a los objetos de forma grupal para que no recaiga la responsabilidad de la calidad del objeto sobre el desarrollador).

\_\_\_ **Pruebas** (proceso de ejecutar un programa con el fin de encontrar errores)

\_\_\_ **Auditorías** (sirve para determinar el grado de cumplimiento y la adecuación de los procedimientos, instrucciones, especificaciones, códigos, estándares u otros.)

10. ¿El personal implicado en el proyecto es capacitado?

Si \_\_\_ No \_\_\_

11. ¿Hay Registros de Evaluación de la Calidad establecidos?

Si \_\_\_ No \_\_\_

12. ¿El equipo del proyecto pone en práctica lo que plantea TSP?

Si \_\_\_\_\_ No \_\_\_\_\_

¿Qué técnicas se usan?

\_\_\_\_\_ Se desarrollan productos en varios ciclos.

\_\_\_\_\_ Se establecen estándares para medir la calidad y el comportamiento.

\_\_\_\_\_ Se proporcionan métricas para equipos.

\_\_\_\_\_ Se evalúan los roles y equipos.

\_\_\_\_\_ Se usan guía para soluciones de problemas en los equipos.

### Flujo de prueba:

1. ¿Utiliza alguna metodología de desarrollo en el proyecto?

Si \_\_\_\_\_ No \_\_\_\_\_

En caso de utilizar alguna diga cuál: \_\_\_\_\_

2. ¿Tienen definido en el proyecto un flujo de pruebas?

Si \_\_\_\_\_ No \_\_\_\_\_

3. ¿Los desarrolladores prueban lo que ellos hacen?

Si \_\_\_\_\_ No \_\_\_\_\_

¿Cómo? \_\_\_\_\_

¿Por qué? \_\_\_\_\_

4. ¿Tienen en el proyecto definidos roles de pruebas?

Si \_\_\_\_\_ No \_\_\_\_\_

¿Cuáles? \_\_\_\_\_

5. ¿Qué funciones hacen estos roles?

6. ¿Se elabora el plan de prueba?

Si \_\_\_\_\_ No \_\_\_\_\_

7. ¿Cómo se diseñan las pruebas en el proyecto?

8. ¿Se Identifican los objetivos de las pruebas?

Si \_\_\_\_\_ No \_\_\_\_\_

9. ¿Se diseñan casos de prueba?

Si \_\_\_\_\_ No \_\_\_\_\_

¿Cómo? \_\_\_\_\_

10. ¿Se describen los casos de prueba?

Si \_\_\_\_\_ No \_\_\_\_\_

11. ¿Se identifican y definen las técnicas apropiadas de prueba?

Si \_\_\_\_\_ No \_\_\_\_\_

12. ¿Se lleva a cabo la implementación y ejecución de las pruebas?

Si \_\_\_\_\_ No \_\_\_\_\_

13. ¿Se evalúan las pruebas?

Si\_\_\_ No\_\_\_

14. ¿Se analizan y evalúan los fallos encontrados?

Si\_\_\_ No\_\_\_

¿Por qué?\_\_\_\_\_

15. ¿Se identifican y listan los defectos encontrados?

Si\_\_\_ No\_\_\_

¿Por qué?\_\_\_\_\_

16. ¿Se le informa al líder del proyecto de los resultados del proceso de aceptación de prueba?

Si\_\_\_ No\_\_\_

¿Por qué?\_\_\_\_\_

17. ¿Se lleva a cabo el proceso de aceptación de prueba?

Si\_\_\_ No\_\_\_

18. ¿El personal del grupo de prueba cuenta con la capacitación suficiente?

Si\_\_\_ No\_\_\_

#### Anexo 14 Lineamientos de Calidad

Los presentes tópicos resumen un pequeño conjunto de prácticas básicas para orientar la producción de software, mas no garantizan con su aplicación productos de alta calidad, pues se trata de un pequeño grupo de elementos para empezar. Estos lineamientos deben ser ajustados a cada caso particular, aplicables a cualquier proyecto de desarrollo de software, permitiendo la orientación y organización del trabajo dentro y fuera del proyecto.

##### ✓ Proyecto

- Las primeras actividades planificadas deben estar encaminadas a la motivación con el proyecto.
- El líder de proyecto debe contar con conocimientos básicos del proceso de desarrollo de software.
- Los miembros del equipo deben recibir superación antes o durante del proceso que les permita enfrentarse al problema en cuestión con una visión amplia y con un número mayor de herramientas.

##### ✓ Trabajo en Equipo

- Debe quedar definida y documentada una estructura organizacional que permita identificar y agrupar las tareas requeridas, seleccionar y establecer las estructuras organizativas, definir roles y relaciones de autoridad en el proyecto.

- Debe realizarse un estudio de los roles que serán desempeñados en el proyecto y de las personas a las cuales serán asignados dichos roles, teniendo en cuenta responsabilidades y objetivos que debe cumplir este rol así como la formación y los conocimientos que posee la persona.
- Debe definir y asignar cada rol a los miembros del equipo del proyecto [documentando dicho proceso de asignación y el método empleado], Debe quedar explícito como medir su desempeño para con dicho rol, debe ser evaluado de forma sistemática contraponiendo los resultados obtenidos a los objetivos definidos con anterioridad y midiendo el cumplimiento de las tareas en cada hito definido.
- En el proyecto no deben olvidarse definir roles comúnmente obviados como es el caso de: Planificador, Integrador, Ingeniero en Componentes, Diseñador de pruebas y Probador.
- Las reuniones de equipo deben realizarse de forma sistemática y quedar reflejadas en el cronograma de trabajo como parte del proceso normal de desarrollo.
- Deben estar definidos los espacios en que los miembros del equipo ventilen dudas e inquietudes, podrían definirse talleres de discusión de soluciones técnicas que formarían parte también del cronograma como actividad planificada.
- Debe realizarse un control sistemático, marcado por los diferentes hitos definidos en el cronograma del avance del trabajo que realiza cada integrante del equipo. Una buena práctica es también el chequeo de cada tarea o subtarea al 50% de su realización y una vez concluida la misma debe evaluarse su cumplimiento de manera integral.
- La realización de las tareas en todas sus fases debe ser registrada de forma medible en función del tiempo y del volumen de trabajo (el estimado contra el real) y definir métricas sobre el tiempo que dedica un desarrollador a cada tarea específica.
- En cada fase del proceso de desarrollo deben ser registrados los defectos detectados, obtener datos sobre estos, analizarlos y determinar métricas que permitan detectar la mejor forma de reducirlos.

### ✓ **Planificación**

- Deben ser planificadas las fases y las iteraciones del ciclo de vida del proyecto y quedar registrada dicha planificación en el cronograma, definiéndose los entregables que evidencien que se ha vencido una iteración o una fase.
- Deben establecerse puntos de chequeo que permitan verificar el estado del proceso.
- Deben establecerse objetivos medibles y claros para las fases y las iteraciones.
- Las tareas deben ser atómicas y bien delimitado su tamaño para poder precisar su avance.
- Definir y documentar en cada etapa los artefactos que deben producirse y una vez obtenidos deben ser evaluados.

- La planificación debe realizarse en base a los datos obtenidos del control de tiempo y esfuerzo de los miembros del equipo.
- ✓ **Trabajo Personal**
  - Deben registrarse el tiempo personal de trabajo de los miembros de equipo.
  - Deben registrarse de forma individual los defectos encontrados.
  - Debe llevarse un control personal de los compromisos.
- ✓ **Modelo de Calidad**
  - Debe existir conocimiento por parte de los miembros del equipo de los modelos de calidad existentes y regirse por alguno de ellos que estará bien definido y documentado.
  - Si no se utiliza ningún modelo de calidad debe existir, al menos, definido en el proyecto un plan de calidad.
- ✓ **Requisitos**
  - Una vez capturados los requisitos deben ser verificados y validados. [32]

A lo largo de este capítulo se trataron temas relacionados con la calidad, afines con las buenas prácticas de CMMI (Modelo de Madurez y Capacidad Integrado), de las diferentes normas ISO (Organización Internacional de Normalización) y además un estudio según lo planteado por PSP y TSP en relación con la calidad.

## **Anexo 15 Herramienta JMeter**

### **Proceso de Instalación:**

Se necesita tener una Máquina Virtual Java 1.3 o superior instalada en el sistema operativo (Windows en este caso).

Primeramente se necesita instalar una máquina virtual de Java. Se puede encontrar en la siguiente dirección `ftp://10.0.0.22/Development/Java/JVM/` y se ejecuta el fichero con el nombre `jre-6u2-windows-i586-p-s.exe`.

Una vez instalada, se puede entonces instalar la herramienta JMeter.

- Copiar JMeter.
- Descomprimir el archivo Tar de JMeter en un directorio temporal (tmp por lo general) a través del comando: `jakarta-jmeter-<número_deversión>`.
- El paso anterior genera un directorio por nombre `jakarta-jmeter-<numero_de_version>`, dentro del cual se encuentran los diversos componentes de JMeter, descritos a continuación.

## Estructura

Bin: Este directorio contiene los ejecutables utilizados por JMeter, tanto para ambientes Linux (JMeter) así como Windows (jmeter.bat).

Docs: Contiene documentación acerca de JMeter

Printable\_docs: Contiene documentación -- en modalidad de impresión -- acerca de JMeter

Lib: Este directorio contiene los archivos JAR's empleados por JMeter requeridos en cualquier modalidad.

## Ejecución:

La ejecución de JMeter es llevada a cabo del directorio bin, que contiene las siguientes opciones:

JMeter | jmeter.bat: Corresponde a los ejecutables de la interface principal para plataformas Linux y Windows, respectivamente.

JMeter-server | jmeter-server.bat: Representan los ejecutables para el emulador de Servidor JMeter para plataformas Linux y Windows, respectivamente.

JMeter.properties: Contiene propiedades de arranque para JMeter que son utilizadas por cualquiera de sus ejecutables.

jmeter.log: Representa los registros ("logs") generados al ejecutar JMeter.

users.xml | users.dtd: Un archivo XML y su correspondiente DTD, empleados para definir características de usuarios que serán simulados por JMeter.

ApacheJMeter.jar: Archivo JAR que contiene las principales clases de JMeter.

El componente principal de JMeter es el Plan de Prueba, a continuación se ilustra paso a paso un Plan de Prueba utilizado para simular una carga de 50 usuarios solicitando la página principal en determinado sitio.

- Estando en la interface principal de JMeter en la columna izquierda hacer clic en **Test Plan**, en la ventana derecha aparecerán varias opciones, aquella encontrada en la parte superior permite asignar un nombre a su plan de prueba, se define a criterio deseado.
- Definido el nombre, colóquese nuevamente en el icono Test Plan y hacer clic derecho, del menú generado seleccione la opción grupo de usuarios **Add -- Thread Group**.
- Seleccionar el icono **Thread Group** recientemente creado. Al llevar a cabo este paso la ventana derecha mostrará la siguiente serie de opciones:

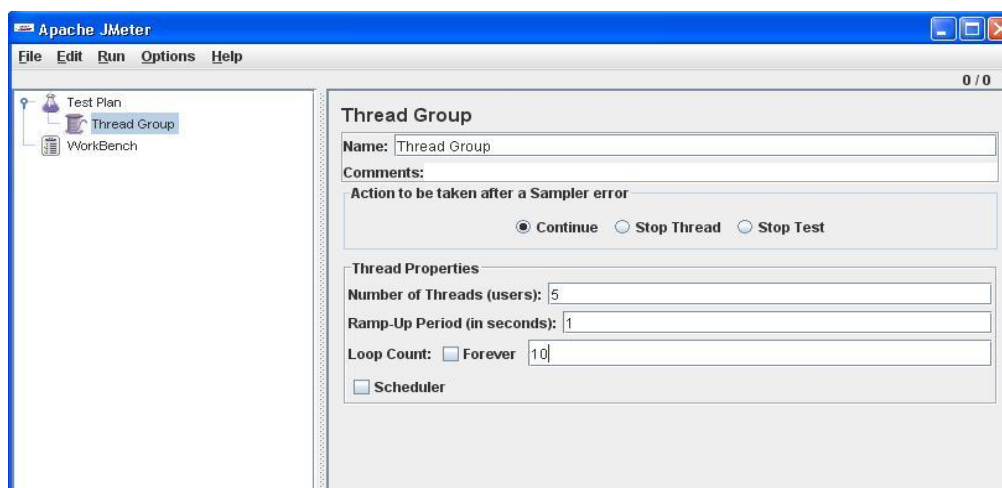
Name: Utilizado para definir un nombre más descriptivo sobre el grupo de usuarios

Number of threads: Equivale al número de usuarios que se desean simular, en este caso se simularán 50 usuarios.

Ramp-up period: Es el lapso de tiempo en segundos que se desea tener entre cada grupo de usuarios, en este caso 15 segundos.

Forever: Utilizado para indicar si la simulación para grupos será llevada a cabo infinitamente, esto es, si selecciona esta opción indica que desea simular 5 usuarios, esperar 1 segundos (Ramp-up period), simular otros 5 usuarios y así sucesivamente. Para esta prueba es recomendable defina 10 ciclos para simular un total de 50 usuarios en cuestión de 10 segundos.

Scheduler: Finalmente, esta opción permite definir tiempos de arranque para determinados grupos de usuarios, para efectos prácticos de esta simulación no será seleccionada esta opción y se iniciará la prueba a discreción.

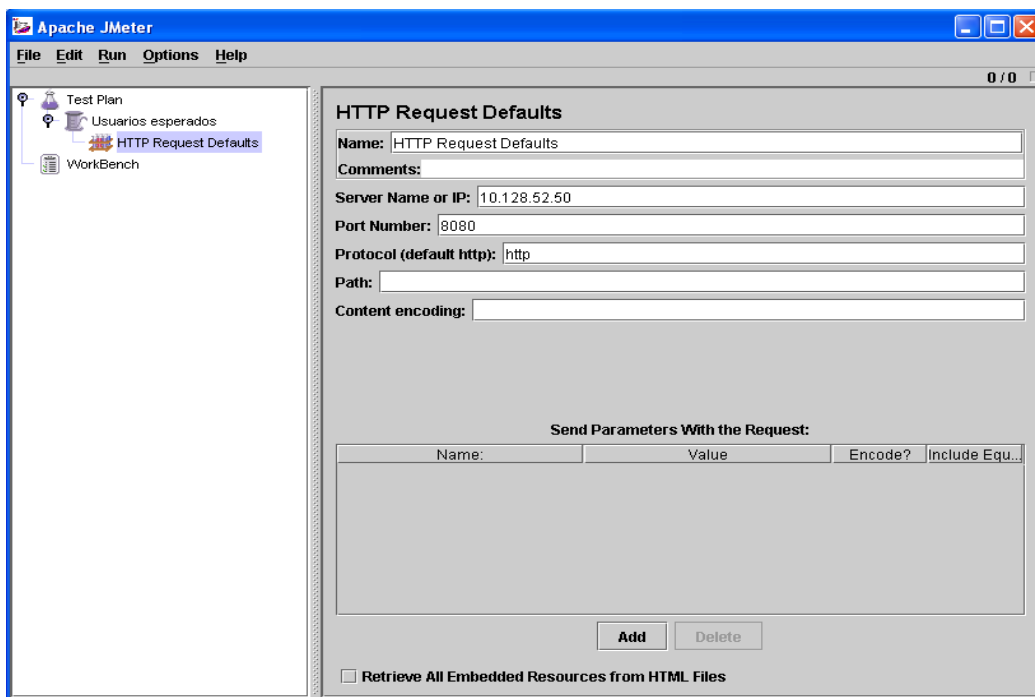


Una vez definidas las características del grupo de usuarios, colóquese encima del ícono Usuarios esperados y ahora seleccione la opción Add > Config Element > HTTP Request Defaults. Lo anterior genera el ícono HTTP Request Defaults (se especifican parámetros requeridos).

Server Name or IP: Empleado para definir la dirección IP o nombre del servidor donde será llevada a cabo la prueba de carga, se utilizará 10.128.52.50 para indicar un servidor.

Port number: Define el puerto TCP de operación del servidor, será empleado 80, que es el "default" para Servidores de Páginas.

Protocol (default http): Define el protocolo que en este caso va a ser http.



Seguidamente en el ícono WorkBench o Banco de Trabajo, se selecciona Add < Non-test Element < http Proxy Server. Dentro de la configuración se especifican los siguientes parámetros:

Port: El puerto en este caso es el 8080.

Test Plan Content:

Target Controller: Se selecciona Thread Group que es donde van a ser guardadas las muestras.

En excluir e incluir se ponen las extensiones que se necesitan o no descargar junto con la petición.

.\* - all

.\*\png – png images

.\*\gif – gif images

.\*\jpg – jpeg images

.\*\php

.\*\jsp

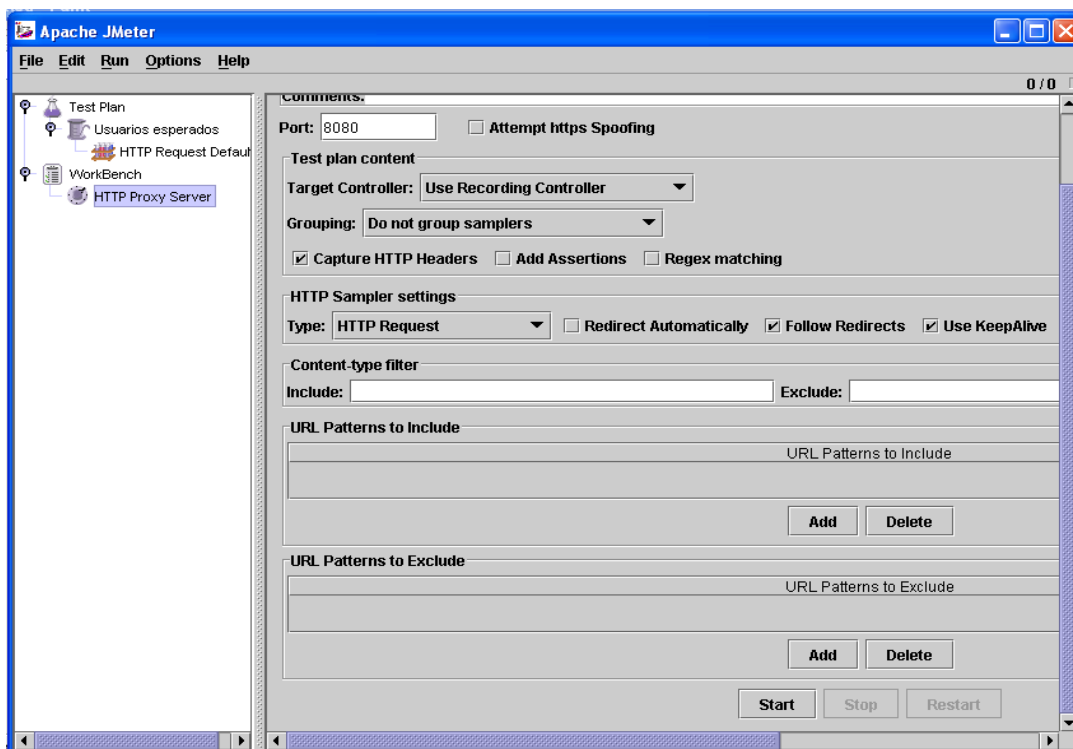
.\*\html

.\*\htm

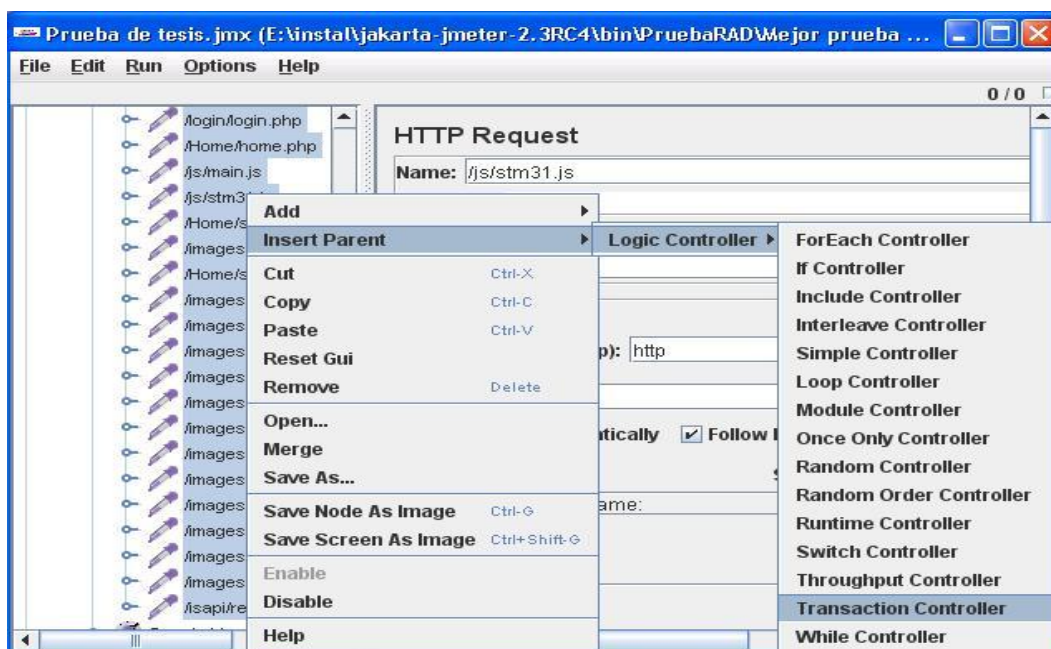
.\*\js

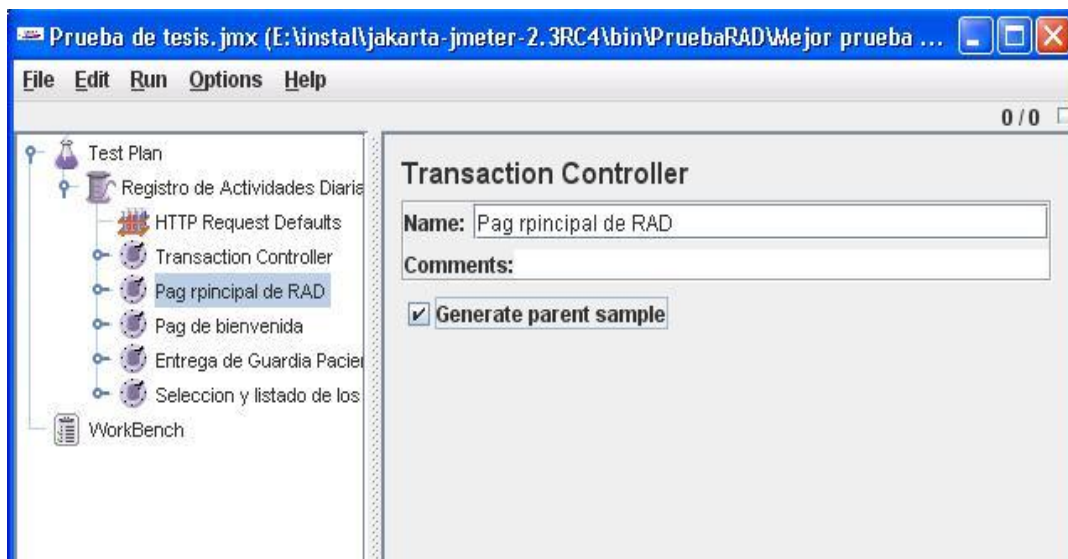
Se pone a correr la Herramienta, haciendo clic en comenzar.





Una vez comenzada la navegación por la aplicación, y a medida que se vaya visitando una página nueva, se agrupan las muestras obtenidas por cada página en un Transaction Controller, para mayor organización. Para agruparlos, se selecciona todas las muestras de la página y se va a: Insert Parent < Logic Controller < Transaction Controller, donde se selecciona la opción Generate Parent Sample.





Seguidamente en el ícono Thread Group o usuarios esperados, seleccionando Add < Listener < y ahí se selecciona la forma en que se quieren ver los resultados.

Un Listener es un componente que muestra los resultados de las muestras. Los resultados pueden mostrarse en un árbol, tablas, gráficos o absolutamente escrito en un archivo. Los listeners despliegan la información de la contestación de maneras diferentes. Sin embargo, todos ellos escriben los mismos datos del rendimiento del archivo.

A continuación una breve descripción de la funcionalidad de 2 de los Listener (los más importantes).

#### ✓ **Summary Report**

El Summary Report o reporte de resumen crea una fila de la tabla diferente para cada uno nombró la demanda en su prueba.

Label ---- La etiqueta de la muestra.

# Muestras ---- El número de muestras para la URL.

Average ---- El tiempo medio de obtención de resultados (tiempo promedio de atención a las solicitudes).

Min ---- El tiempo más bajo para las muestras de la URL dada.

Max ---- El tiempo más largo para las muestras de la URL dada.

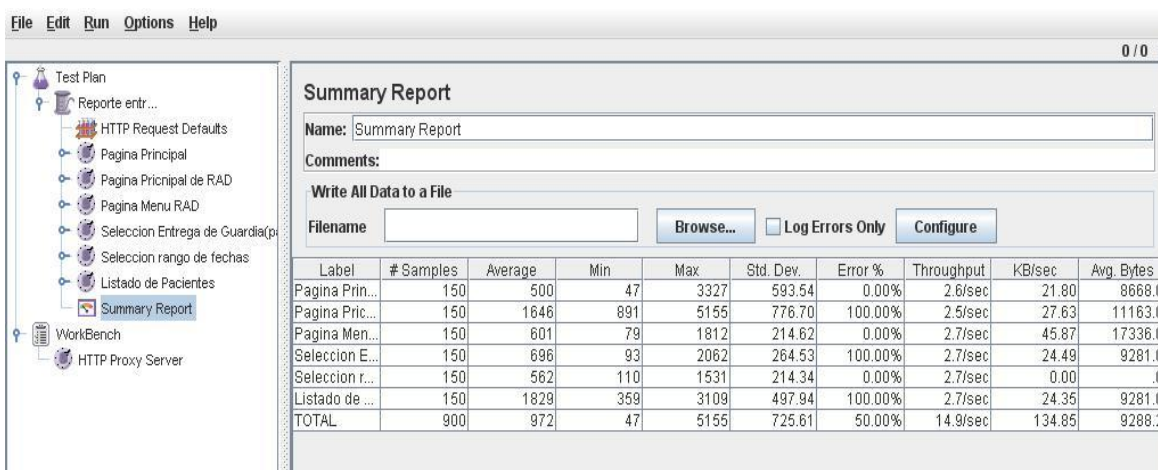
Std.Dev ----- La desviación normal de la muestra (es una medida de dispersión de los datos con respecto al valor promedio).

Error % ---- Porcentaje de solicitudes con errores.

Throughput---- Rendimiento que midió en las solicitudes por second/minute/tour (número de solicitudes por minutos que el servidor ha procesado).

KB/sec ---- Rendimiento que midió en Kilobytes por segundo.

Avg.Bytes ---- El tamaño medio de la contestación de la muestra en bytes.



**Summary Report**

Name: Summary Report

Comments:

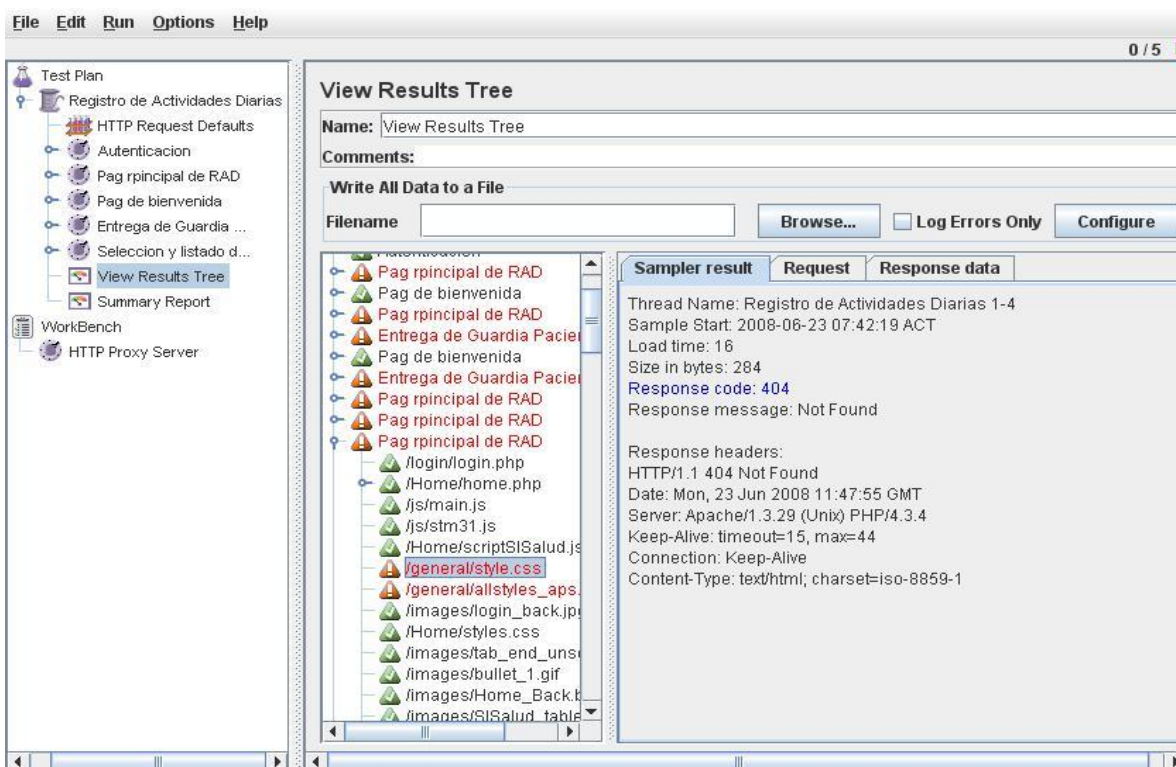
Write All Data to a File

Filename:    Log Errors Only

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Pagina Prin...	150	500	47	3327	593.54	0.00%	2.6/sec	21.80	8668.0
Pagina Pric...	150	1646	891	5155	776.70	100.00%	2.5/sec	27.63	11163.0
Pagina Men...	150	601	79	1812	214.62	0.00%	2.7/sec	45.87	17336.0
Seleccion E...	150	696	93	2062	264.53	100.00%	2.7/sec	24.49	9281.0
Seleccion r...	150	562	110	1531	214.34	0.00%	2.7/sec	0.00	.0
Listado de ...	150	1829	359	3109	497.94	100.00%	2.7/sec	24.35	9281.0
TOTAL	900	972	47	5155	725.61	50.00%	14.9/sec	134.85	9288.2

### ✓ View Result Tree

Este reporte muestra los errores encontrados, localización y tipo de error.



**View Results Tree**

Name: View Results Tree

Comments:

Write All Data to a File

Filename:    Log Errors Only

Sampler result Request Response data

Thread Name: Registro de Actividades Diarias 1-4  
Sample Start: 2008-06-23 07:42:19 ACT  
Load time: 16  
Size in bytes: 284  
Response code: 404  
Response message: Not Found

Response headers:  
HTTP/1.1 404 Not Found  
Date: Mon, 23 Jun 2008 11:47:55 GMT  
Server: Apache/1.3.29 (Unix) PHP/4.3.4  
Keep-Alive: timeout=15, max=44  
Connection: Keep-Alive  
Content-Type: text/html; charset=iso-8859-1

Existen más Listener, pero se explican los más importantes. Se consta con un documento en el expediente del proyecto que explica los demás Listener (reportes). Pero se creyó necesario que fueran puestos en este anexo.

## **Anexo 16 Herramienta Rational Robot.**

### **Requerimientos de Sistema y Software:**

Rational Robot debe usarse sobre los sistemas operativos Windows 2000, 95/98, NT o XP, que son los sistemas compatibles para el uso de esta herramienta.

### **Precondiciones de Instalación:**

Verifique que su sistema cubra los requisitos mínimos y esté funcionando en un sistema operativo compatible.

Haga una copia de seguridad de su registro y directorios de sistema.

Si usted tiene productos de Rational de IBM sobre su sistema de uno release previo, este proceso de instalación los desinstala. Haga una copia de seguridad de cualquier archivo local y datos relacionados con esas versiones.

Detenga todas las aplicaciones que estén ejecutándose. Por ejemplo si usted tiene software de protección de virus, usted debe apagarlo. Estos programas corren a menudo «en background» y obstruyen el rendimiento del proceso de instalación porque la protección de virus verifica cada archivo que es instalado.

Si usted ha especificado una unidad de disco alternativa para la instalación. Estos archivos pueden requerir entre 5 y 15 megabytes del espacio de disco temporal sobre su disco duro.

### **Instalación:**

Entre al sistema como un usuario con los privilegios de administrador en la computadora local.

Dirija la instalación de la siguiente forma:

En una ventana de comando, navegue al directorio raíz del área de release. Luego entre en el setup.exe de mandato. En el Explorer de Windows, trace un mapa de una unidad de disco hacia donde el área de release está ubicada. En ese directorio ejecutar el comando setup.exe del Rational. Luego de siguiente hasta que aparezca la siguiente ventana.

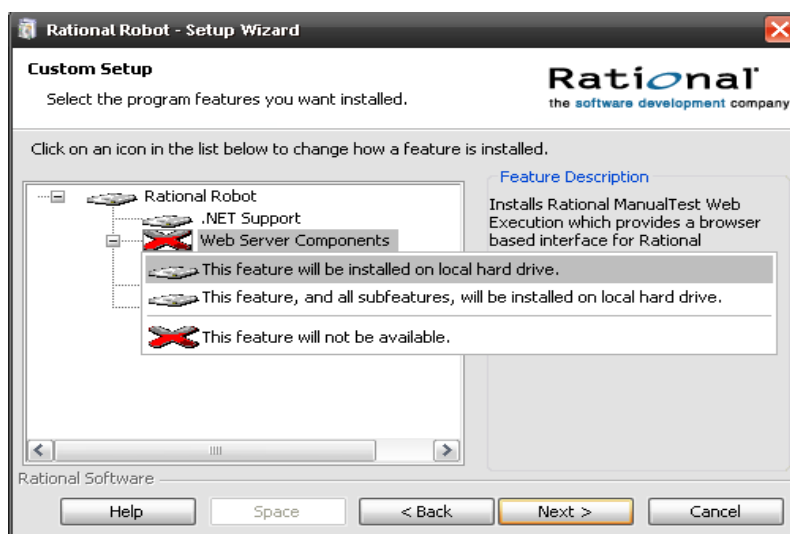
Al salir la siguiente ventana se va a mostrar una lista de los diferentes programas que se pueden instalar desde el Rational, de ahí se selecciona la herramienta que se quiere instalar, en este caso, el *Rational Robot*, y se le da siguiente.

Luego de este paso ya aparece la instalación de la herramienta. Ahora se selecciona la opción *Desktop Installation from Cd image*, pues previamente se tiene el programa guardado en una carpeta local de la computadora donde se instalará. Dar siguiente.



Luego de *Next* y acepte los términos y condiciones para su uso.

En la ventana siguiente se debe marcar en la cruz roja y cuando salga la opción marcar *This feature will be installed on the local hard drive*. Esto permitirá una mejor ubicación para el directorio usado por el software en instalación. Presione el botón *Next*.



AL comenzar la instalación cuando salga esta ventana marcar *Next* y cuando se finalice la instalación presionar *Finish*.

### Usando el Rational Robot.

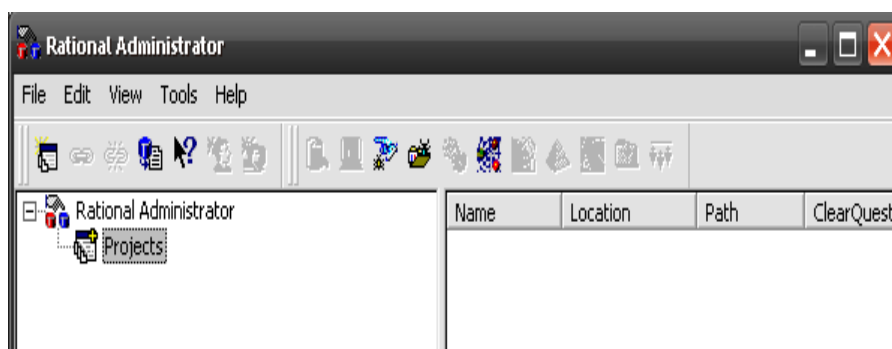
Antes de comenzar usando el Robot:

- Debe haber instalado el Rational Robot.
- Debe tener creado un Proyecto Rational.

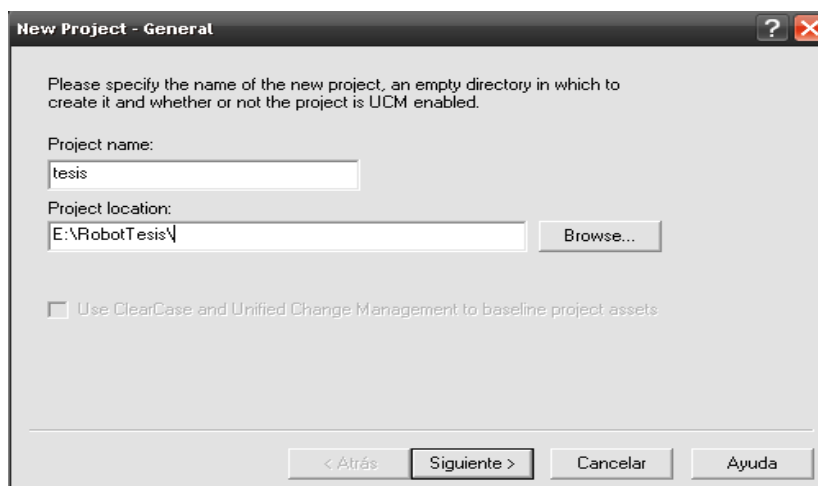
Si usted no ha creado un proyecto para usar el robot debe hacerlo para poder vincular las pruebas a este proyecto y hacer mejor la funcionalidad.

### Pasos para crear Proyecto Rational:

Para crear un proyecto usted debe ir a *Inicio/Todos los programas/ Rational Software/ Rational Administrator*.



Cuando abra el Rational Administrator, *File/ New Project*.



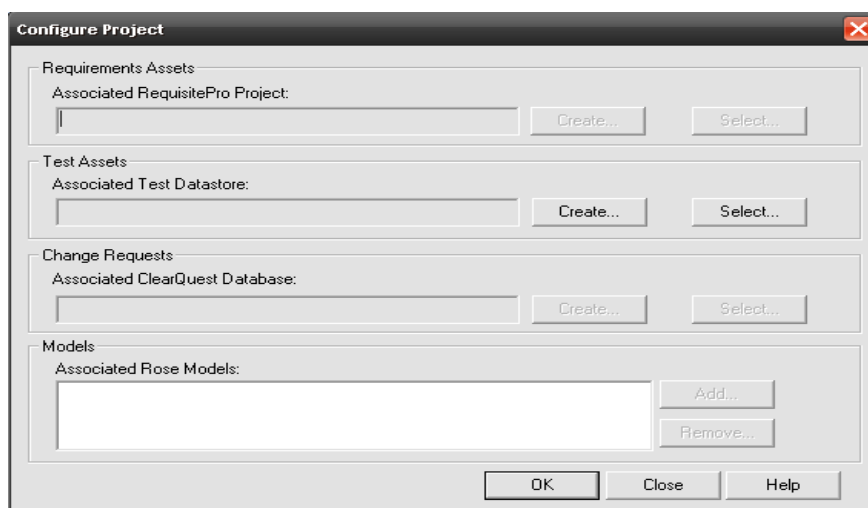
En el Project name usted debe especificar un nombre para este proyecto que se va a crear, y en Project location usted debe dar clic en Browse y buscar un directorio donde se guardará el proyecto para su futura utilización. La carpeta especificada para la ubicación debe ser una carpeta que esté compartida. Dar siguiente.

Ahora debe establecer una contraseña para su proyecto (Opcional). Luego dar siguiente.



Ahora tendrá una ventana donde se hará un resumen de la configuración que se ha establecido, se debe marcar la opción *Configure Project Now*, y marcar *Finalizar*.

Cuando se ha finalizado la creación y configuración del proyecto es necesario vincularlo a un Test Datastore donde se va a guardar todo cuando se trabaja con este proyecto. Para asociarlo debe crear el Test Datastore de la siguiente forma. Debe marcar en *Test Assets Create*.

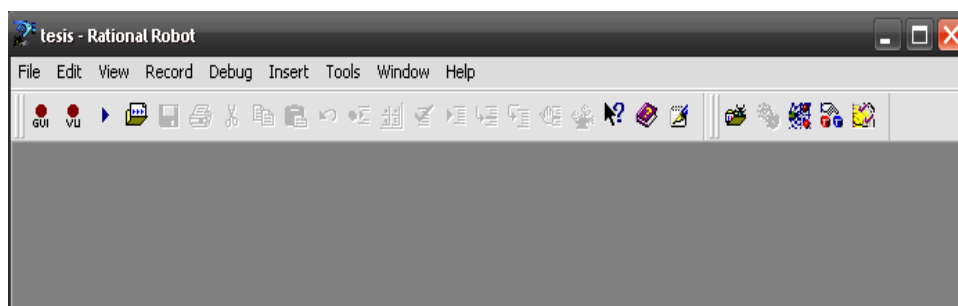
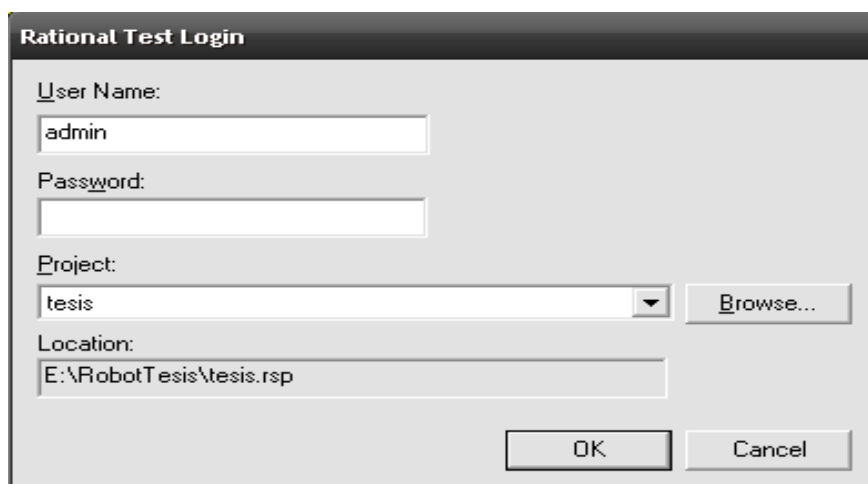


Para crear el Test Datastore al salir la próxima ventana usted debe marcar *Microsoft Access* y marcar siguiente. Es en este lugar donde se va a guardar todo lo relacionado con los datos que se obtengan de los resultados de los script generados.

Luego dar *Siguiente* en todas las ventanas hasta llegar a la venta de *Finalizar*. A partir de ahí se creará el Test Datastore asociado y es el paso final.

### Trabajo con Rational Robot. Generación de scripts.

Para iniciar la aplicación se busca en *Inicio/Todos los Programas/Rational Software/Rational Robot*. De ahí se especifican los permisos con el usuario y contraseña para acceder al proyecto que va a ser usado para el proceso y se da OK. Luego de esto se va a abrir la ventana del Rational Robot.



Cuando usted graba un script de GUI en robot, se graban todas sus acciones durante la ejecución de la aplicación bajo prueba. Estas acciones de usuario incluyen las pulsaciones y los clics de ratón que lo ayudan navegan a través de la aplicación.

Debe considerar las siguientes líneas guías antes de comenzar con la grabación:

- El comienzo y fin de la grabación debe tener un punto común, pues los script pueden ser corridos otra vez en cualquier orden.
- Establecer un ambiente de pruebas.


Cualquier ventana que está abierta cuando se comience la grabación debe ser cerrada, si está activa o visualizada si no se requiere que esté en la grabación. Incluyendo otras aplicaciones como explorador de Windows, correo electrónico u otra. Pues se graba el tamaño y la posición de aquellas ventanas que estén visualizadas cuando comience la grabación. En general usted debe cerrar las ventanas abiertas antes de comenzar a grabar.

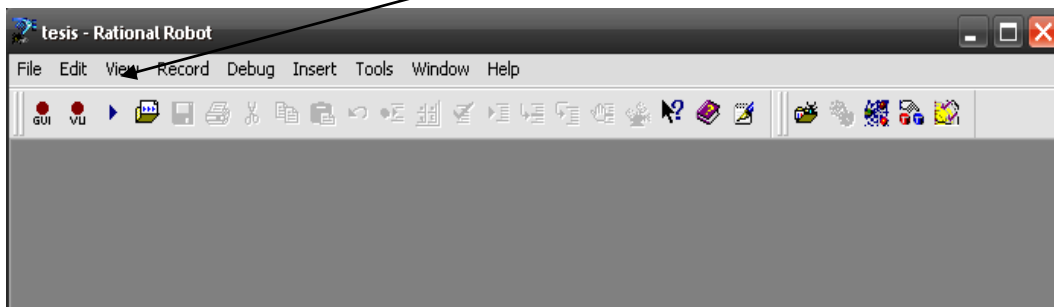
### **Pasos para grabación de scripts.**

Existen dos tipos de grabaciones de scripts. Unas son las GUI script, que son para pruebas funcionales y las VU script que son para pruebas de rendimiento.

Grabando un GUI script.



Dar Clic al botón Record GUI Script  en la barra de herramientas para abrir el cuadro de dialogo Record GUI.




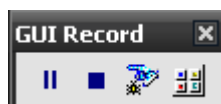
Escribir un nombre nuevo o seleccionar un script de la lista.

Para cambiar las opciones de grabación marque Options, y luego de Ok.


Clic Ok para comenzar la grabación. Luego aparecerán los siguientes eventos:

- Robot es minimizado por defecto.
- La barra de herramientas de GUI aparece flotando en la pantalla. Usted puede usar esta barra de herramientas para pausar o parar la grabación, la visualización de robot, e insertar especiales en un script.

Para comenzar la aplicación bajo prueba, de clic en la ventana flotante en el botón .

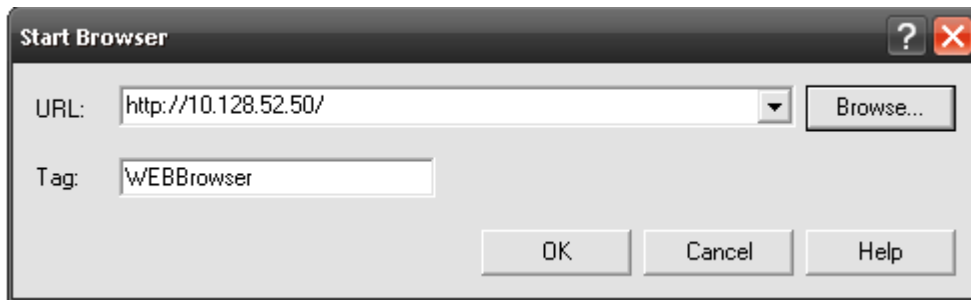


Si ya usted tiene abierta y visualizada la aplicación que se desea testear solo debe realizar las acciones que quiere grabar sobre la misma. Y seguir en el paso

De clic en el botón Start Browser , en la ventana de inserción de la GUI, pues se va a realizar script de pruebas en páginas HTML.




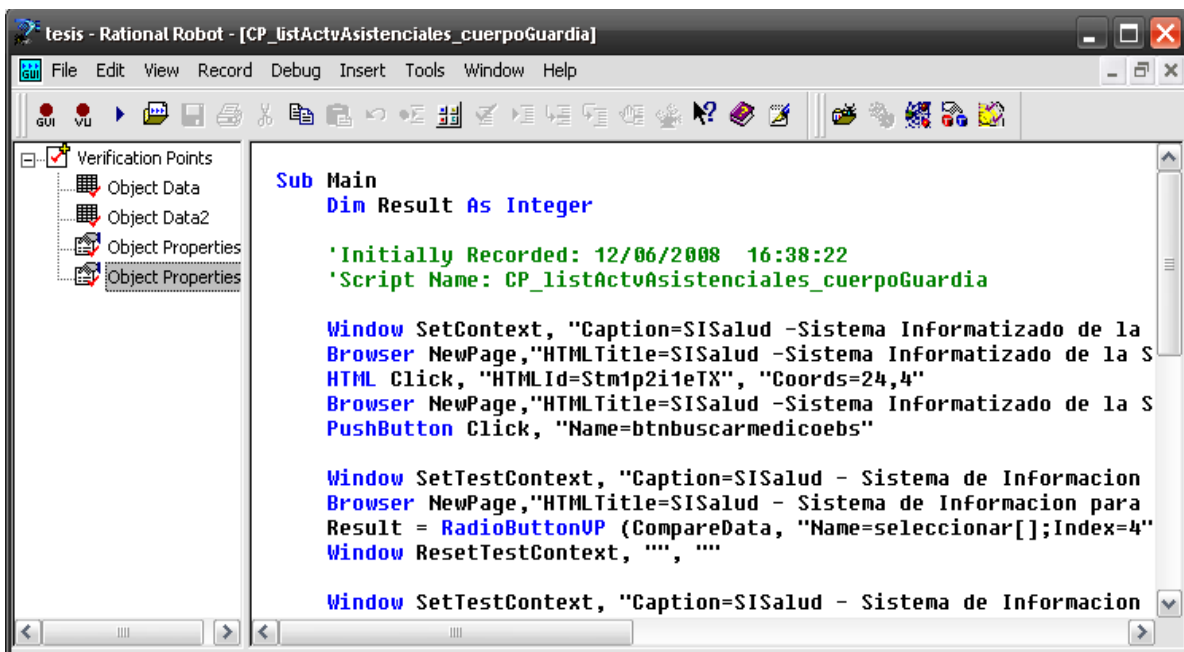
Dar clic en la caja de diálogo y cuando salga la siguiente ventana especificar la URL a donde se va a conectar y presionar Ok, esto es en el caso que se quiera comenzar a grabar desde la autenticación en el módulo.



Para asegurar de que se hayan usado las mismas coordenadas en este tiempo el script es corrido de vuelta, usted debe maximizar el explorador de Windows si es necesario. En este punto usted debe performar acciones como sean necesarias para navegar a través de la aplicación.

También usted puede insertar otros aspectos como puntos de verificación y comentarios.

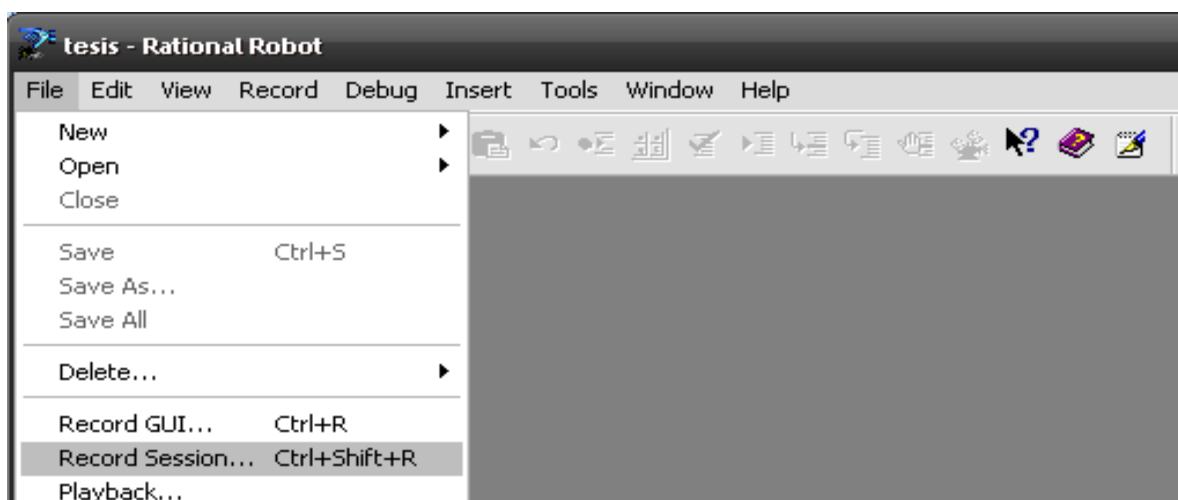
Cuando haya terminado presione el botón Stop .



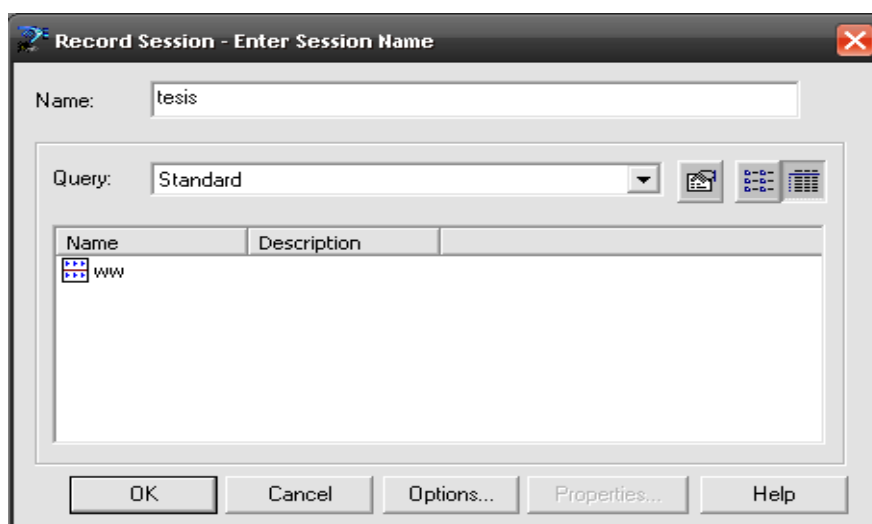
En esta ventana aparece el objetivo de su trabajo, que es el script grabado en la ventana del menú del Robot.

### Grabando un VU script o sesión de script.

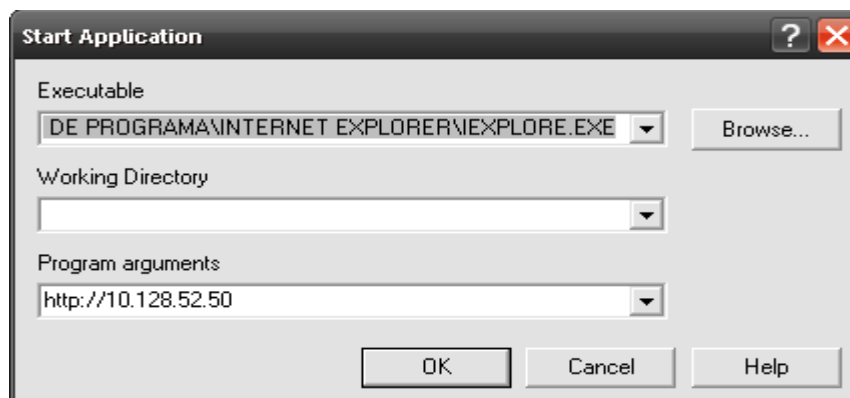
Cuando tenga la ventana de Rational Robot abierta seleccionar en *File/ Record Session*.



Se crea una sesión, y se especifica un nombre para la misma. Luego se da clic en Ok.

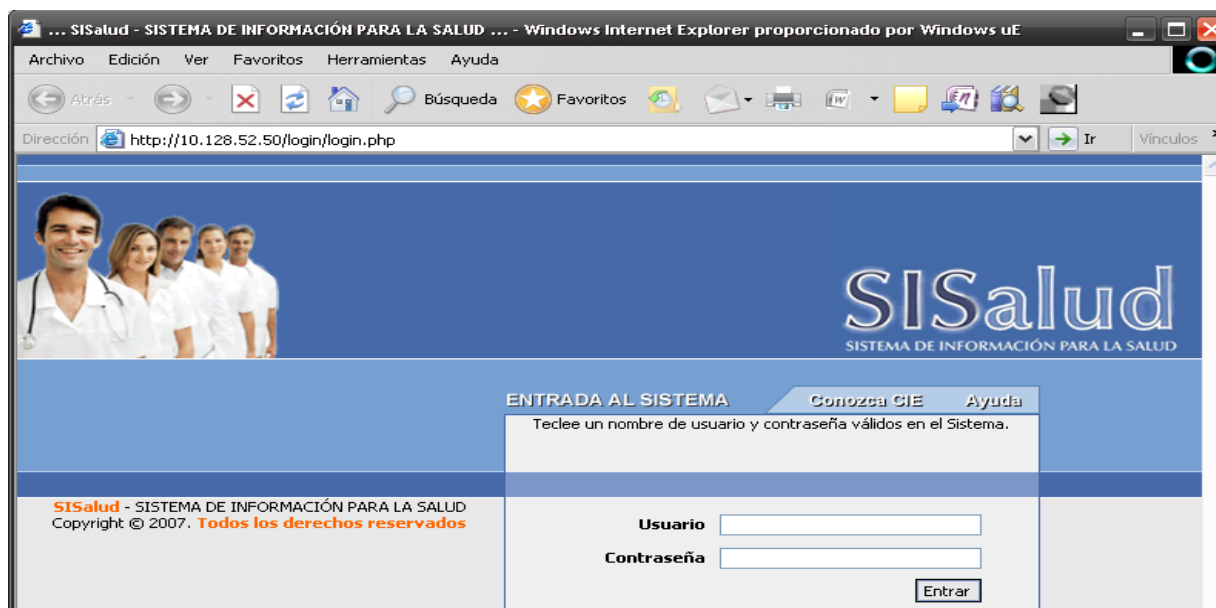


Se especifica el navegador que va a ser usado, así como la dirección Web o Desktop. Luego presionar Ok.



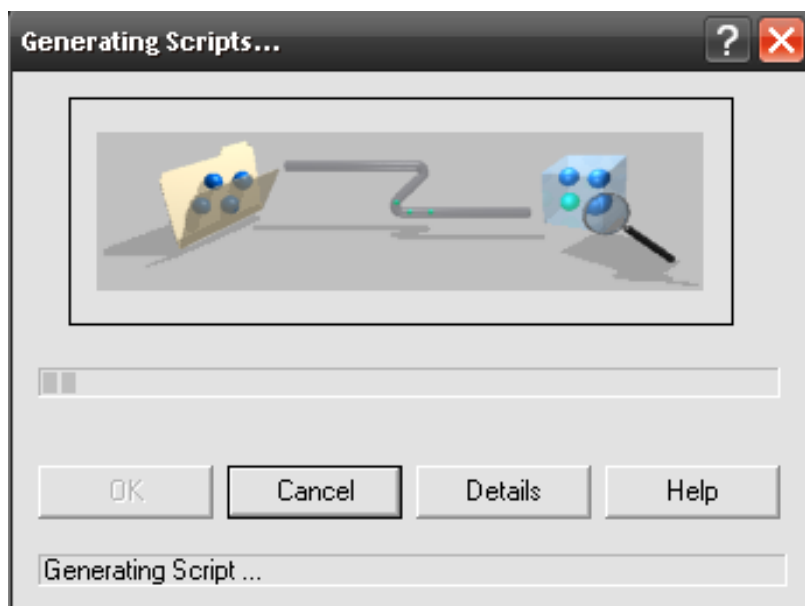
Se van a realizar acciones sobre el caso de prueba actual.

Por ejemplo registrarse en la página principal.

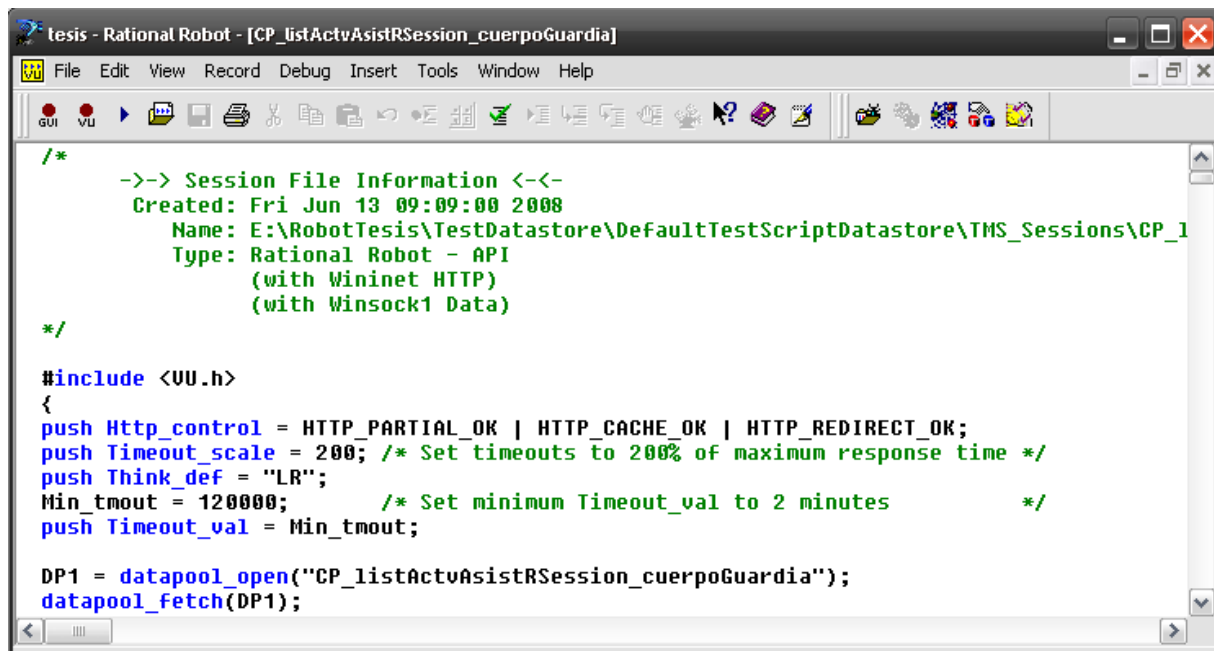


Luego que se ha entrado a la aplicación, se realizarán tantas acciones como sea necesario en la misma, estas serán grabadas, para medir el rendimiento de las mismas y los tiempos de respuesta y pedidos al servidor.

En este paso se va a generar el script del caso de estudio probado.



En el menú del Robot aparecerá el script generado.



```

/*
  ->-> Session File Information <-<-
  Created: Fri Jun 13 09:09:00 2008
  Name: E:\RobotTesis\TestDatastore\DefaultTestScriptDatastore\TMS_Sessions\CP_1
  Type: Rational Robot - API
        (with Wininet HTTP)
        (with Winsock1 Data)
*/


#include <UU.h>
{
push Http_control = HTTP_PARTIAL_OK | HTTP_CACHE_OK | HTTP_REDIRECT_OK;
push Timeout_scale = 200; /* Set timeouts to 200% of maximum response time */
push Think_def = "LR";
Min_tmout = 120000; /* Set minimum Timeout_val to 2 minutes */
push Timeout_val = Min_tmout;

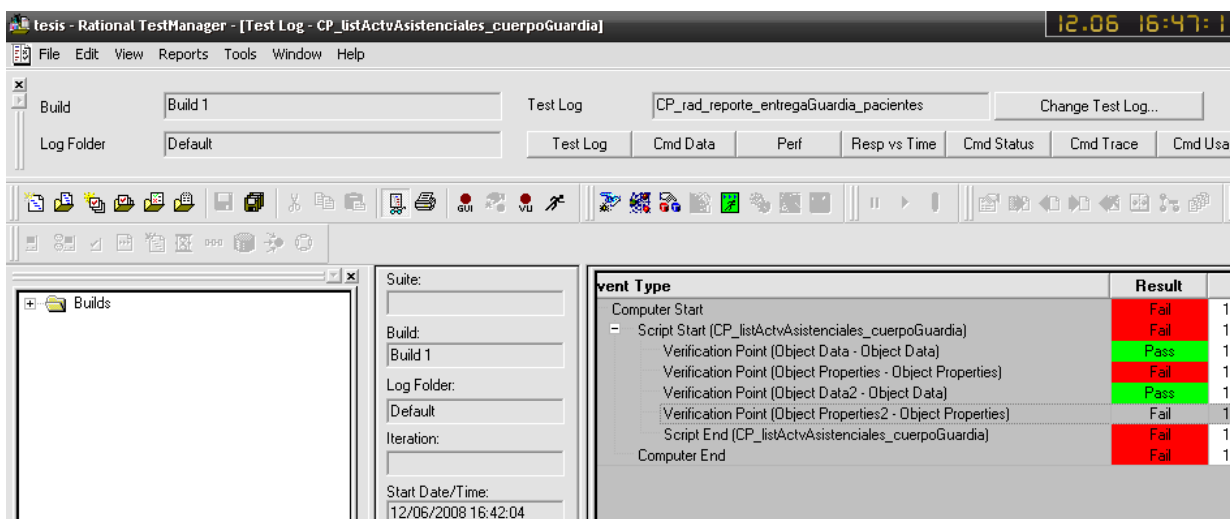
DP1 = datapool_open("CP_listActvAsistRSession_cuerpoGuardia");
datapool_fetch(DP1);

```

### Trabajo con Rational Robot. Análisis de resultados de las grabaciones de GUI script.

En ambos casos se usará el módulo Registro de Actividades Diarias (RAD) para hacerle una prueba a uno de sus casos de uso como caso de estudio.

Siguiendo el script generado en el punto anterior de GUI script, se da clic en el botón *Playback*  y se obtienen los siguientes resultados en el Test Manager.



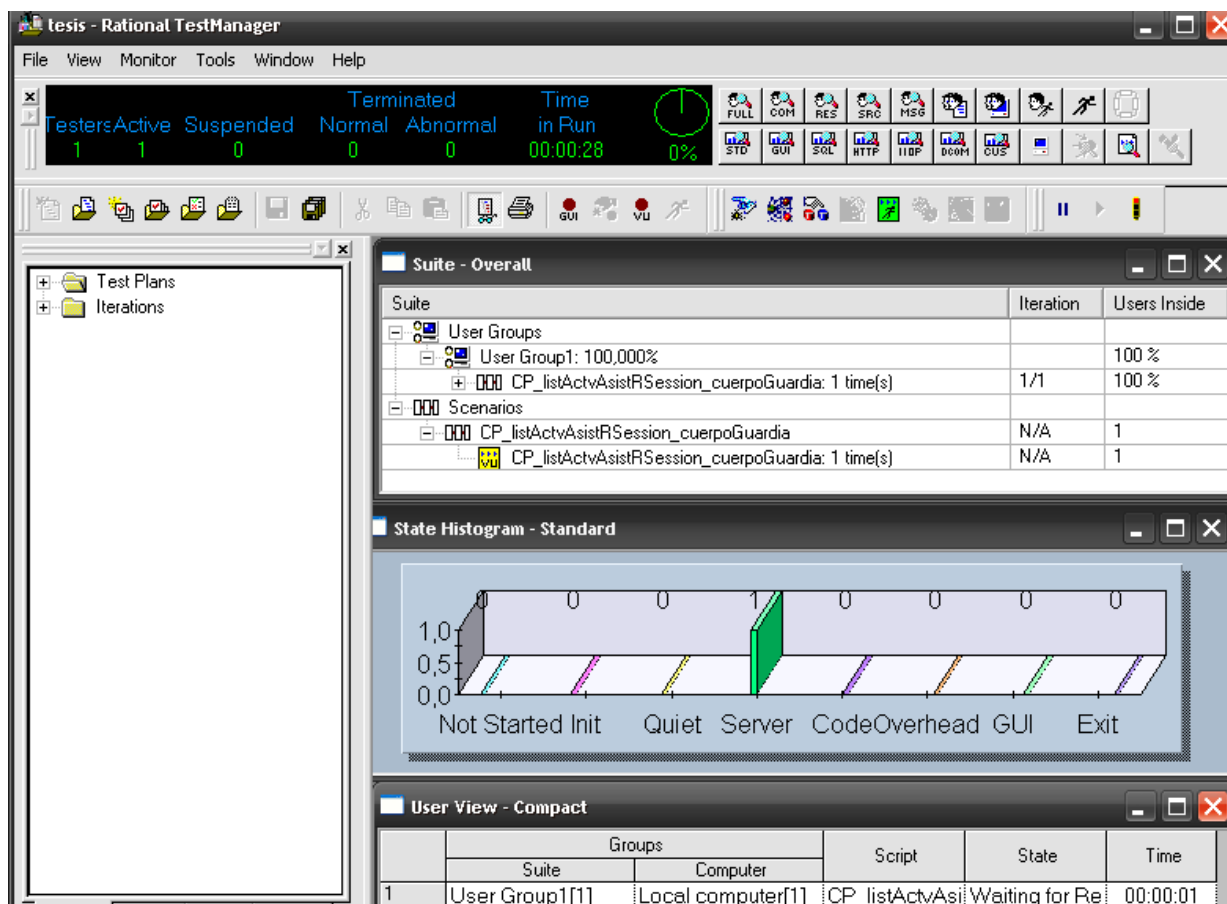
Event Type	Result	Time
Computer Start	Fail	12:06:16
Script Start (CP_listActvAsistenciales_cuerpoGuardia)	Fail	12:06:16
Verification Point (Object Data - Object Data)	Pass	12:06:16
Verification Point (Object Properties - Object Properties)	Fail	12:06:16
Verification Point (Object Data2 - Object Data)	Pass	12:06:16
Verification Point (Object Properties2 - Object Properties)	Fail	12:06:16
Script End (CP_listActvAsistenciales_cuerpoGuardia)	Fail	12:06:16
Computer End	Fail	12:06:16

Al fallar uno de los puntos de verificación insertados durante la grabación, falla la reproducción del script por completo, mostrándose en *Result/Fail*, pues no se han obtenido los resultados esperados según el script generado. Entonces se analizarán estos resultados según los puntos de verificación, tales como Alphanumeric, Object Data, Object Properties, Region image, Windows image. Si no falla ninguno de los

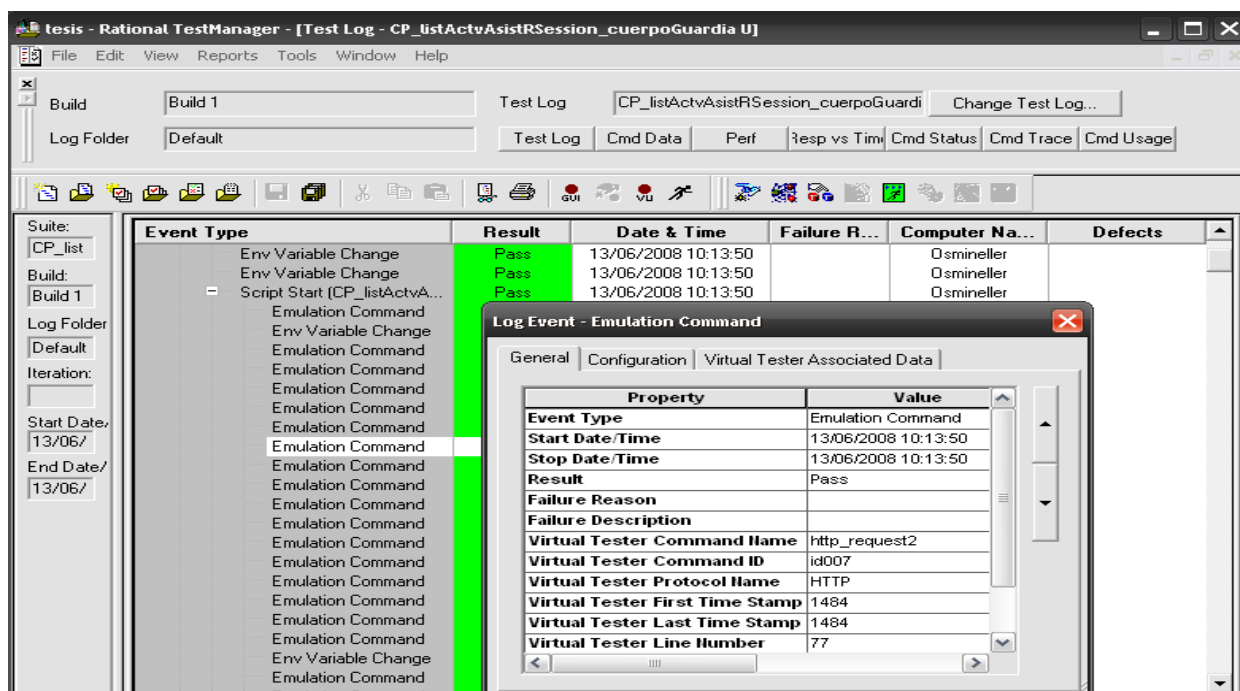
puntos insertados, y los resultados que aparecen son en *Result/Pass*, esto significa que no han ocurrido problemas durante la prueba de regresión y se ha terminado con la misma.

### Trabajo con Rational Robot. Análisis de resultados de las grabaciones de VU script

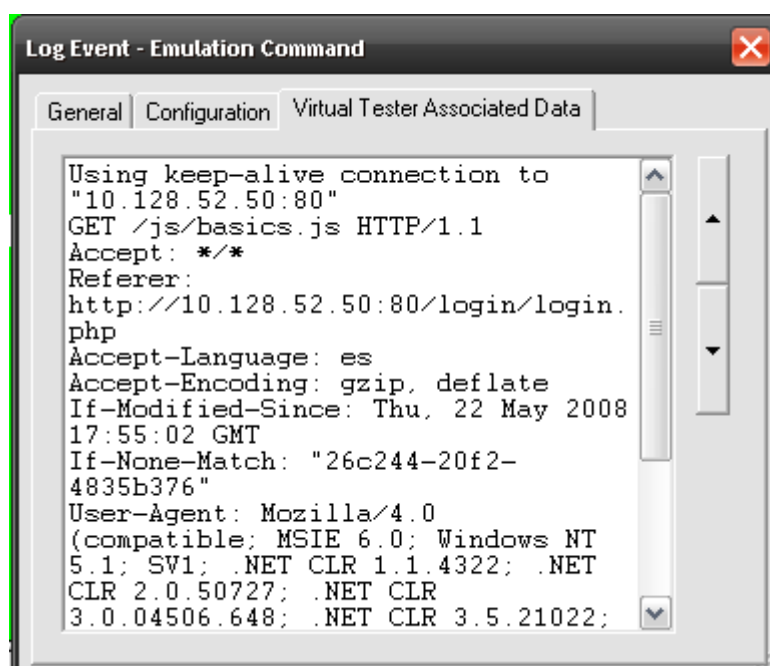
Al reproducir el script o mediante el Test Manager/ Run Suite se muestran los siguientes resultados. Se muestra el histograma de la prueba, así como otros datos.



Al terminar la ejecución de la suite entonces se muestran los siguientes resultados de los diferentes eventos ocurridos.

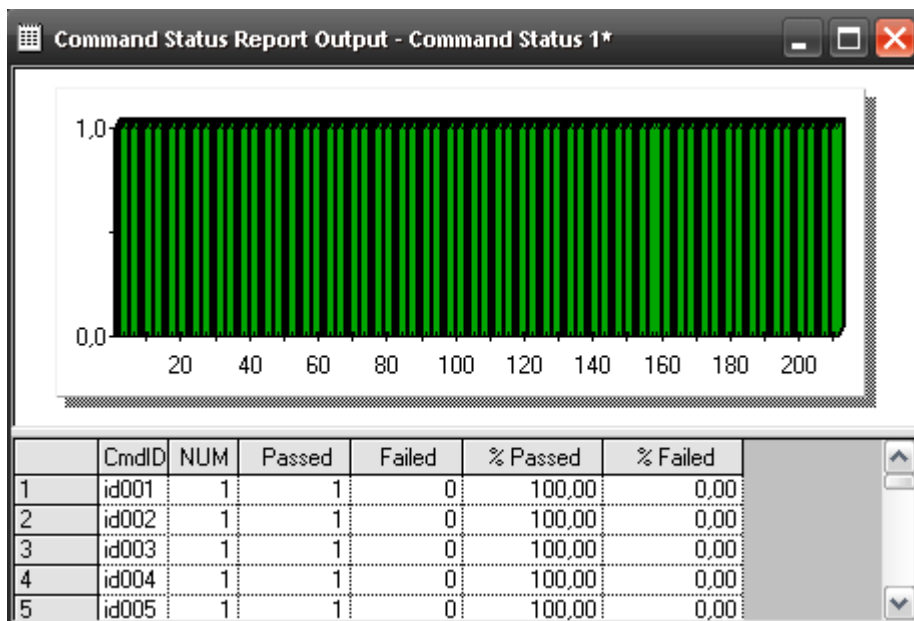


En esta ventana se muestran diferentes datos como fecha y tiempo en que se ejecuto la suite. También se puede ver los registros de un evento, dando clic derecho sobre el mismo (se muestra la ventana *Log Event*). En esta ventana también puede verse algunos detalles de la conexión, esto mediante *Virtual Tester Associated Data*, en la ventana de *Log Event*.

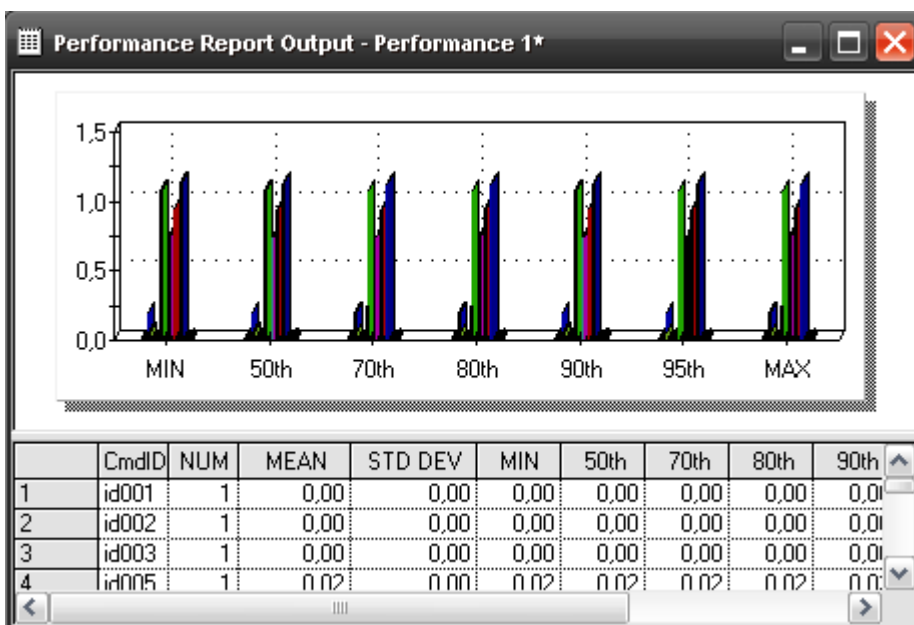


**Otros resultados.**

También se muestran otras tablas con reportes específicos como:



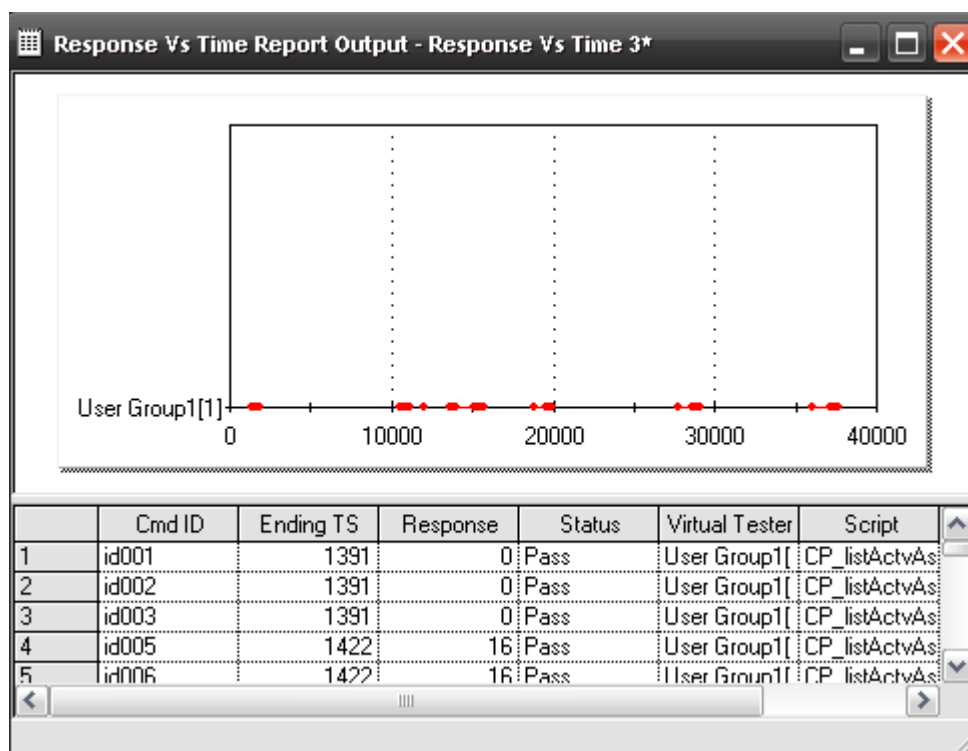
Esta tabla muestra el estado de los diferentes comandos ejecutados durante la grabación. El número de veces que este comando es respondido. Así como el estado si ha pasado, o si ha fallado en la ejecución del mismo.



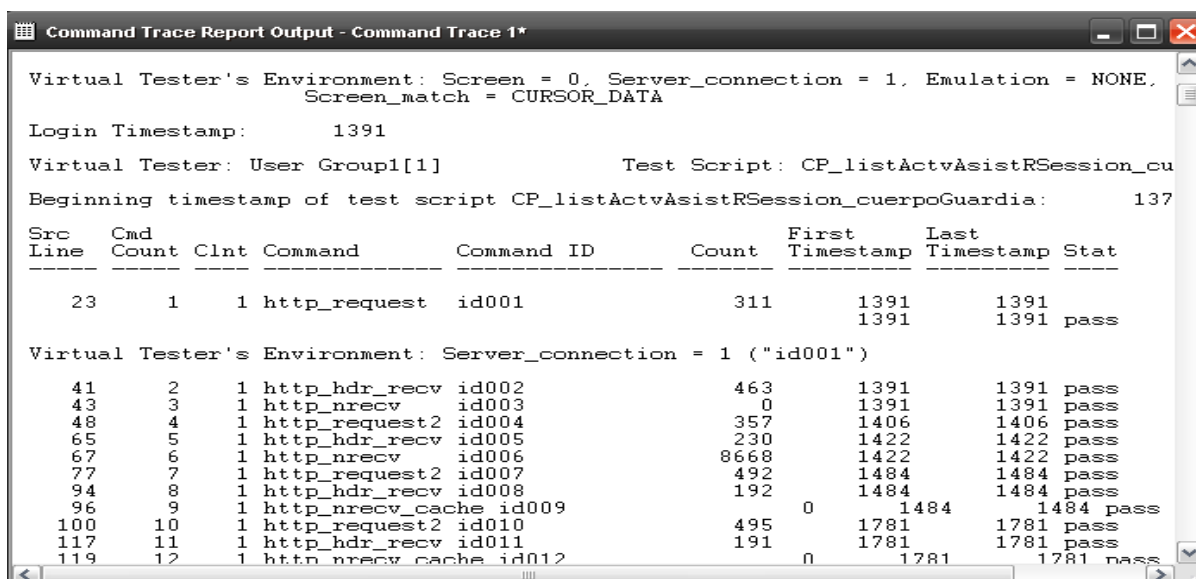
Esta tabla muestra un reporte de rendimiento de los comandos ejecutados. El número de respuestas para este comando, o sea el número de tiempo que aparece en el registro de prueba. También muestra la media, no es más que la media de veces de respuesta para un comando con la media de todas las



respuestas en general. También la desviación típica para cada comando, así como el tiempo máximo y mínimo de respuesta para cada comando y sus porcentajes.



Esta tabla es las respuestas contra el reporte de tiempo. En esta se muestra la cantidad de respuestas individuales. Verifica la tendencia en el tiempo de respuesta. Si la tendencia varía durante varias iteraciones es porque se ha excluido el registro de entrada y el tiempo de configuración de los resultados, por tanto se hace necesario diseñar nuevamente las pruebas. Se muestra la fecha fin de lectura para cada comando así como el tiempo de respuesta, en milisegundos.



En este reporte se muestran resultados de los reportes en general como es el tiempo en milésimas desde que comenzó la ejecución del script.

```

Command Usage Report Output - Command Usage 1*
Time of Emulation Session: Fri Jun 13 2008 10:13

Command Usage Report

Cumulative Statistics for All Selected Virtual Testers and Scripts

Active Time:          36.3 secs   Inactive Time:          1.4 secs
Passed Commands:     95          Average Throughput:    157.11 cmds/min
Failed Commands:     0          Average Throughput:     0.00 cmds/min
Passed Responses:    95          Average Throughput:    157.11 resps/min
Failed Responses:    0          Average Throughput:     0.00 resps/min

Time Spent Waiting:   4.5 seconds => 12.4%
Time Spent in Input:  0.0 seconds =>  0.0%
Time Spent Thinking: 31.8 seconds => 87.6%

Passed HTTP Connections: 4          Passed Connections/Min: 6.6
Failed HTTP Connections: 0          Avg Connect Setup Time: 4 ms
HTTP Sent Kbytes:      49          HTTP Sent KBytes/Sec: 1.3
HTTP Received Kbytes:  115         HTTP Recv KBytes/Sec: 3.2
Sent Kbytes/Connection: 12.1       Recv Kbytes/Connection: 28.7

Summary Statistics

Duration of Run:       36.3 seconds (00:00:36.3)

Summary Statistics for Selected Virtual Testers and Scripts:

```

En este reporte se muestran resultados de los reportes en general como es el tiempo total en milésimas, en que se mantuvo en ejecución el script desde el inicio. Se muestran datos generales como fecha en que se hizo la ejecución del script, así como la cantidad de usuarios con que emuló. En este reporte se puede observar datos de la ejecución de los diferentes comandos en que línea del código se puede encontrar el comando, así como un contador de las veces que ha sido ejecutado un comando según el tiempo en que ha sido corrido el script.

El id del comando y si ha sido ejecutado para un pedido o para una respuesta del servidor. El Count da la cantidad de bytes que ha requerido este comando para su tarea, y además, para comandos de emulación este es una clave que puede ser 0 si (ni recibió ni transmitió), 1 si la memoria intermedia envía pero no recibe y 2 si la memoria intermedia envía y recibe. Por último muestra el tiempo en milésimas de segundos, en que el comando comenzó a ser ejecutado desde el inicio de la corrida del script y el tiempo final en que se ha terminado con su uso así como el estado, por ejemplo si ha sido pasado o si ha fallado en la conexión.

```

Command Usage Report Output - Command Usage 2*
Command Usage Report
Cumulative Statistics for All Selected Virtual Testers and Scripts

Active Time:          36.3 secs  Inactive Time:          1.4 secs

Passed Commands:      95          Average Throughput: 157.11 cmds/min
Failed Commands:      0          Average Throughput:  0.00 cmds/min
Passed Responses:     95          Average Throughput: 157.11 resps/min
Failed Responses:     0          Average Throughput:  0.00 resps/min

Time Spent Waiting:   4.5 seconds => 12.4%
Time Spent in Input:  0.0 seconds =>  0.0%
Time Spent Thinking: 31.8 seconds => 87.6%

Passed HTTP Connections: 4          Passed Connections/Min: 6.6
Failed HTTP Connections: 0          Avg Connect Setup Time: 4 ms
HTTP Sent Kbytes:       49          HTTP Sent KBytes/Sec: 1.3
HTTP Received Kbytes:   115         HTTP Recv KBytes/Sec: 3.2
Sent Kbytes/Connection: 12.1        Recv Kbytes/Connection: 28.7

Summary Statistics

Duration of Run:       36.3 seconds (00:00:36.3)

Summary Statistics for Selected Virtual Testers and Scripts:

Passed Commands:      95          Total Throughput: 157.11 cmds/min
Failed Commands:      0          Total Throughput:  0.00 cmds/min
Passed Responses:     95          Total Throughput: 157.11 resps/min
Failed Responses:     0          Total Throughput:  0.00 resps/min

Number of Virtual Testers: 1          Number of stop_time Cmds: 0
Number of Completed Scripts: 1        Number of Uncompleted Scripts: 0

Average Number of Scripts Completed per Virtual Tester: 1.0 scripts
Average Duration for Completed Scripts: 36.3 seconds
Script Throughput for Completed Scripts: 99.2 scripts/hour

```

Esta tabla de reporte del uso de comandos, muestra estadísticas acumulativas de todas las pruebas virtuales seleccionadas y los script corridos. Da el tiempo activo en que estuvo el script en ejecución así como el tiempo en que estuvo inactivo, la cantidad de comandos pasados y fallidos y el porciento de comandos por minuto. También brinda el tiempo de espera y el tiempo de procesamiento de los comandos y las conexiones pasadas o fallidas por minuto. Brinda estadísticas generales de Kbytes recibidos y enviados durante la conexión, mostrando resultados de rendimiento en general.

### GLOSARIO DE TÉRMINOS

**Actor:** Alguien o algo, fuera del sistema o negocio que interactúa con el sistema o negocio.

**Artefactos:** Pedazo de información producido, modificado o usado por un proceso. Son resultados tangibles del proyecto, las cosas que se van creando y usando hasta obtener el producto final.

**Auditoría:** Inspección independiente a un producto de trabajo o conjunto de productos de trabajo para evaluar si está acorde con las especificaciones, estándares, acuerdos contratados u otros criterios.

**Casos de prueba:** Conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

**Ciclo de vida del software:** Ciclo que cubre cuatro fases en el siguiente orden: inicio, elaboración, construcción y transición.

**Ciente:** Una persona u organización, interna o externa a la organización productora que toma responsabilidad financiera por el sistema. El cliente es el último destinatario del producto desarrollado y sus artefactos.

**Cuello de botella:** Límite en la capacidad de transferencia de información de un sistema que puede reducir el tráfico en condiciones de sobrecarga. Suele producir una baja del rendimiento y la velocidad general tanto en un sistema como en una conexión. Los cuellos botella (llamados restricciones) condicionan la salida de toda la producción.

**Defecto:** Anomalía del sistema, por ejemplo un síntoma de error en el software descubierto durante las pruebas, o un problema descubierto durante una reunión de revisión.

**Empresa:** En términos estrictamente económico, es una unidad económica que reúne una serie de factores de producción: recursos naturales, humanos, tecnológicos (o de capital) y financieros (que posibilitan la adquisición de los anteriores), y los utiliza para producir bienes y/o servicios, que vende a personas individuales, a otras empresas y/ o a las administraciones públicas.

**Fase:** Periodo de tiempo entre dos hitos principales de un proceso de desarrollo.

**Flujo de trabajo:** Realización de un caso de uso de negocio parte de él. Puede describirse en términos de diagrama de actividad, que incluye a los trabajadores participantes, las actividades que realizan y los artefactos que producen.

**IEEE:** (Institute of Electrical and Electronics Engineers, Instituto de Ingenieros Eléctricos y Electrónicos) Asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, ingenieros en sistemas e ingenieros en telecomunicación.

**Proceso:** Secuencia de actividades invocadas para producir un producto de software.

**Procedimiento de pruebas:** Especificación de cómo llevar a cabo uno o varios casos de prueba o parte de ellos.

**Pruebas:** Flujo de trabajo fundamental cuyo propósito esencial es comprobar el resultado de la implementación mediante pruebas de cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema que van a ser entregadas a terceras partes.

**Proyecto:** Esfuerzo de desarrollo para llevar un sistema a lo largo del ciclo de vida.

**Registro de Calidad:** Son los documentos o archivos en los cuales se identifica, agrupa, codifica, conserva y dispone todo lo referente a los productos elaborados o servicios prestados. Los registros de calidad se deben conservar para demostrar que se ha logrado la calidad requerida y la operación efectiva del sistema de calidad.

**Requerimiento:** Condición o capacidad que debe cumplir un sistema.

**Rol:** Papel, cometido o función que tiene o desempeña que interpreta un actor.

**Software:** Palabra en inglés utilizada para indicar a los programas de computadoras, a las aplicaciones.

**Usuario:** Persona que usa ordinariamente una cosa.