

Universidad de las Ciencias Informáticas

Facultad 4



Título: Estrategia de Aseguramiento de la Calidad para el Proyecto SIGEP Fase IV.



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

Autores:

Anabel Fernández Martínez

Yamisel Cruz Sánchez

Tutores:

Ing. Anielis Rodríguez Sotolongo

Ing. Hermes M. Velázquez Domínguez

Ciudad de la Habana. Junio, 2009.

DECLARACIÓN DE AUTORÍA.

Declaramos ser autores de la presente tesis y reconocemos al proyecto Sistema de Gestión Penitenciario (SIGEP) y a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yamisel Crus Sánchez

Firma del Autor

Anabel Fernández Martínez

Firma del Autor

Ing. Anielis Rodríguez Sotolongo

Firma del Tutor

Hermes M. Velázquez Domínguez

Firma del Tutor



"Calidad es una característica de fortaleza y estabilidad que es reconocida por un proceso inerte. Debido a que las definiciones son producto de un pensamiento formal y rígido, la calidad no puede ser definida."

Robert M. Dilts.

AGRADECIMIENTOS.

Anabel.

- ✓ A mami y papi, por ser siempre los que comprenden, los que apoyan, los que regañan, los que aconsejan, los que aman. En fin, por ser los que se siempre se quedan, los que nunca dan la espalda. Son ustedes lo que más amo en este mundo. Gracias.
- ✓ A mis abuelos Zena y Nene, por ser mis segundos padres, la segunda mayor influencia de mi vida, los pilares de mis pilares. Los que dan la fuerza, los que dicen "Adelante".
- ✓ A Ana Lilian por convertirme, más que en mi prima, en mi hermana. Por oírme, por soportarme, por quererme, por admirarme. Eres una de las principales razones por las que quiero ser mejor.
- ✓ A mis tíos Fefita y Albertico por estar ahí para mí, por brindarme muchos de mis buenos momentos y por formar parte de mis mejores recuerdos.
- ✓ A Mercedes y Bertha por su infinita paciencia y por ser las tías más lindas y buenas de este mundo.
- ✓ A Yami, mi amiga del alma y compañera de tesis. Eres lo mejor que me quedará de esta escuela. Confío en que estaremos en contacto a pesar de las distancias.
- ✓ A Leonelvys, mi amigo del alma. Tú aportaste mucho a mi vida. Lo único que lamento es que no hayas llegado antes, muchas cosas habrían sido más sencillas.
- ✓ A Maelis y Mara por soportar mis llantos, mis risas, mis chistes y por haberme permitido darles lo mejor de mí.
- ✓ A mi prima Haydée por haberme apoyado, por haberme introducido aquí en la UCI y haberme mostrado muchas de sus cosas buenas y más que todo por haber compartido tus amistades conmigo.

- ✓ A mis tíos abuelos Andrea y Pedro, Pepe y Rosa, Felo y Lourdes, Flora y Felipe, por haberme brindado el mejor de los ambientes familiares en estos 5 años que no estuve en casa.
- ✓ A mis primos Maibel, Maikel, Anay y Viedma. Por sus locos consejos y por ayudarme siempre que busqué sus palabras.
- ✓ A mis tutores: Ann y Hermes, por la guía en este trabajo que fue la oportunidad de mostrar que con esfuerzo todo se logra.
- ✓ A todos los profesores que durante mi carrera compartieron conmigo su sabiduría.
- ✓ A la Revolución y a Fidel por haber creado la sociedad y la escuela que forjaron mi persona.
- ✓ A todos los que han aportado alegrías a mi vida, por ser reales, leales y especiales conmigo. Por cada abrazo gracias, por cada beso gracias. Mil veces gracias por sus consejos y esfuerzos.

Pamisel.

- ✓ A mi mamá, por ser la luz de mi vida, mi guía incondicional a través de estos 5 años, mi orgullo personal y mi razón de ser cada día. Eres la persona que más quiero en este mundo mamita.
- ✓ A mi papá por ser mi héroe y mi principal razón de esforzarme todos los días de mi vida. Te quiero un montón papito.
- ✓ A mi abuela Iris, por ser más que mi abuela, ser mi madre y mi apoyo a través de las más grandes dificultades. Por ayudarme en cada paso de mi vida y estar siempre presente. Adoro tus peleas mi abuela linda.
- ✓ A mi abuelo Nene, por ser la persona que más respeto en mi vida y por estar siempre ahí para su nietecita. Te quiero mucho papi.
- ✓ A mis hermanos, los adoro a los tres.
- ✓ A Tati, por ser más que mi prima, eres mi personita y mi ídolo en muchas de las cosas de mi vida. Te quiero tati.
- ✓ A Tata, por estar siempre ahí para mí y ayudarme siempre. Muchas gracias en todo.
- ✓ A Yosbel, por ser mi primo, mi hermano y mi mejor amigo. Te quiero y siempre te querré.
- ✓ A Jorge e Ileana, por ser ambos mis segundos padres.
- ✓ A mi tía China, por ser la tía más genial del mundo.
- ✓ A todos mis tíos y tías, por darme su apoyo incondicional.
- ✓ A mis tutores, por representar una parte satisfactoria del desarrollo de este trabajo.

- ✓ A Leosvel, por mantener su fiel amistad a pesar de las adversidades y apoyarme en todas las facetas de mi vida a través de los años. Eres de esas personas que llegan y nos marcan para siempre.
- ✓ A Leonelbys, por ser más que mi amigo en esta universidad, ser mi hermano y tener esa fe en mí que nunca ha decaído. Te quiero una pila, ojalá lo mismo tú, me convertí en un bombillito que no prende sin tu luz.
- ✓ A Anabel, por ser mi mejor amiga durante estos años y estar siempre en las buenas y en las malas para mí. Espero que con la distancia no acabe la buena amistad.
- ✓ A Rubén, por haber sido mi guía desde mi primer día en la universidad.
- ✓ A todas esas personas que han significado para mí algo más que una amistad por lo que me han ofrecido y ayudado y que saben que estarán siempre en ese espacio tan especial que les tengo reservado.
- ✓ A todos mis amigos y compañeros que nunca los olvidaré donde quiera que me encuentre.
- ✓ A los profes que han hecho de mí una mejor persona en estos años.

Los quiero a todos muchísimo y mil gracias por lo que soy hoy.

DEDICATORIA

Anabel.

A mami y papi, mis dos amores, los dueños de mi corazón.

A mis abuelos, la base de la familia que amo tanto, y mía por supuesto.

Espero ser lo suficientemente buena como para merecerlos.

Este triunfo es para ustedes.

Famisel

A mi mamá, por ser mi guía a través de todos mis años de universidad y dedicarme su vida entera.

A mi papá, por ser el hombre más bueno del mundo.

A mis abuelos, por ser mi inspiración diaria.

A mis hermanos, por darme su apoyo siempre.

A toda mi familia en general por su apoyo incondicional a través de estos 5 años.

Cada línea de este trabajo este trabajo es para ustedes que son el orgullo de mi vida.

RESUMEN

El proyecto informático **Sistema de Gestión Penitenciaria (SIGEP)** para la **Dirección Nacional de Servicios Penitenciarios (DNSP)** se encuentra en su fase IV de desarrollo. Con la experiencia acumulada hasta el momento, se hace necesario rediseñar la estrategia de aseguramiento de la calidad existente, para darle mejor soporte al proceso de desarrollo de software basado en las características del mismo. En este trabajo se propone una nueva estrategia necesaria para un proyecto de esta escala.

Para la elaboración de este trabajo se estudiaron los diferentes elementos que intervienen en el aseguramiento de la calidad en un proyecto de software, además de los procesos determinados para el SIGEP. Se adoptaron las plantillas y normas de calidad que establece la Universidad de las Ciencias Informáticas y las metodologías y prácticas de calidad establecidas a nivel mundial.

PALABRAS CLAVES

Calidad, Calidad de Software, Aseguramiento de la Calidad de Procesos y Productos, Proceso de Desarrollo de Software, Plan de Aseguramiento de la Calidad, SIGEP, Pruebas de Software, Plan de Pruebas, Revisiones Técnicas.

ÍNDICE

RESUMEN.....	9
INTRODUCCIÓN.....	13
CAPÍTULO 1: FUNDAMENTACION TEORICA.....	16
1.1. Introducción.....	16
1.2. Calidad.....	16
1.3. Calidad de Software.....	19
1.4. Gestión de la Calidad de Software.....	21
1.5. Aseguramiento de la Calidad de Software.....	23
1.6. Las Pruebas.....	27
1.6.1. Las Pruebas Unitarias.....	28
1.6.2. Pruebas de Integración.....	29
1.6.3. Pruebas de Aceptación.....	30
1.6.4. Pruebas de Funcionalidad.....	29
1.7. Revisiones Técnicas Formales.....	31
1.8. Tareas de Calidad en RUP.....	32
1.9. Normas, Modelos y Estándares.....	35
1.9.1. Normas ISO 9000.....	35
1.9.2. IEEE 730.....	36
1.9.3. CMMI.....	37
1.9.4. Garantía de la Calidad de Procesos y de Productos (PPQA).....	38
1.10. CONCLUSIONES:.....	45
CAPITULO 2: ANALISIS AL PLAN DE ASEGURAMIENTO DE LA CALIDAD SIGEP-2007..	46
2.1. Introducción.....	46
2.2. Plan de Aseguramiento de la Calidad.....	46

2.3. Análisis de los puntos del PAC SIGEP – 2007.....	47
2.3.1. Propósito.....	47
2.3.2. Objetivos de Calidad.....	47
2.3.3. Gestión y Organización.....	48
2.3.4. Tareas y Responsabilidades.....	48
2.3.5. Documentación.....	49
2.3.6. Métricas.....	49
2.3.7. Estándares y Guías.....	49
2.3.8. Plan de Revisiones y Auditorías.....	50
2.3.8.1. Tareas generales de Revisiones y Auditorías.....	50
2.3.8.2. Cronograma.....	50
2.3.8.3. Resolución de problemas y actividades de corrección.....	51
2.3.8.4. Proceso de las Revisiones y Auditorías.....	51
2.3.9. Pruebas.....	51
2.3.10. Herramientas, Técnicas y Metodologías.....	52
2.3.11. Gestión de Configuración.....	52
2.3.12. Registros de Calidad.....	52
2.3.13. Entrenamiento.....	53
2.4. CONCLUSIONES.....	54
CAPITULO 3: ESTRATEGIA DEL PAC – SIGEP FASE IV.....	55
3.1. Introducción.....	55
3.2. Estrategia de Calidad.....	55
3.2.1. Organización de los Roles en el Proyecto.....	56
3.2.2. Tareas y Responsabilidades.....	59
A. Tareas de Prevención.....	59

B. Tareas de Evaluación.	60
C. Tareas de Control de Fallos Internos.	68
3.2.3. Métricas.	74
3.2.4. Estándares y Guías.	75
3.2.5. Resolución de Problemas y Acciones de Corrección.....	75
3.2.6. Herramientas, Técnicas y Metodologías.	76
3.2.7. Gestión de Configuración.	77
3.2.8. Registros de Calidad.	78
3.2.9. Entrenamiento.....	79
3.3. CONCLUSIONES.....	81
CONCLUSIONES GENERALES.....	82
RECOMENDACIONES	83
BIBLIOGRAFÍA	84
Anexo 1: Plan Aseguramiento de la Calidad SIGEP – 2009	85
Anexo 2: Plan de Pruebas	122
GLOSARIO DE TERMINOS	144

INTRODUCCIÓN.

Hoy en día las compañías de todo el mundo industrializado reconocen que la calidad del producto se traduce en aumento de las ganancias y permite mayor competitividad en el mercado creciente. La calidad implica costos, pero son más sus beneficios.

La industria del software no es la excepción, por lo que en los últimos años se han realizado intensos trabajos para aplicar los conceptos de calidad en este ámbito. La calidad del software implica la necesidad de contar con parámetros que permitan establecer los niveles mínimos que un producto de este tipo debe alcanzar.

En los últimos años, Cuba ha alcanzado resultados considerables en la implantación de soluciones informáticas para el país. Aparejado a esto se ha logrado concretar esfuerzos conjuntos de universidades, empresas y centros de investigación en la estandarización y mejora de los procesos de ingeniería de software y sistemas. La estandarización es un elemento importante para insertarse en el mercado y lograr productos de mayor calidad y más competitivos.

La calidad debe aplicarse a todos los niveles de la organización, sin embargo es necesaria que sea adoptada una estructura organizacional, la cual ayudará a evitar el desperdicio de esfuerzo y de recursos. La UCI como la mayor entidad productora de software de Cuba tiene que luchar para lograr que sus productos de software tengan la calidad requerida por el cliente y las reglas de calidad establecidas.

A partir de la creación del sistema informático que gestiona los procesos penitenciarios, Sistema de Gestión Penitenciaria (SIGEP), el cual se diseñó con el objetivo de satisfacer las necesidades de gestión de información y apoyo a la toma de decisiones de la Dirección Nacional de Servicios Penitenciarios (DNSP) de la República Bolivariana de Venezuela, el sistema ha pasado por otras versiones y en estos momentos se está trabajando en su cuarta fase de desarrollo.

Cuando se realiza una Reunión de la Comisión Mixta Cuba-Venezuela, ésta tiene un número consecutivo (I, II, III...). A raíz de la VI Comisión Mixta se firmaron contratos con diversos

Ministerios. En el caso del SIGEP, se firmó con el MPPRIJ, constituyendo está la Fase I de todos los subproyectos contratados, incluyendo el desarrollo de Software.

Al firmarse la VII Mixta, se contrató más desarrollo correspondiente a la Fase II. En el caso de la VIII Mixta solo se aprobó presupuesto para la parte venezolana, por lo que la parte cubana no ejecutó ninguna tarea. En el caso de la IX Mixta se contrató la Fase IV del desarrollo.

Como la Fase III la ejecutó la parte venezolana solamente, para evitar que la nomenclatura de la documentación no coincidiera, se determinó que la actual es la Fase IV del SIGEP y no la Fase III.

El SIGEP, en sus fases anteriores, poseía un plan de aseguramiento de la calidad que no se adecuaba a los lineamientos mínimos de calidad de la Universidad y no estaba enmarcado en el proceso de desarrollo del mismo. En un proyecto de esta envergadura, se hace necesario crear una estrategia con el fin de organizar las actividades de aseguramiento de la calidad, aplicar las experiencias adquiridas hasta el momento y mantener bajo control el proceso de desarrollo de software eliminando las causas de los defectos en las diferentes fases del ciclo de vida del producto.

Teniendo en cuenta la situación problemática anteriormente planteada se traza el siguiente **Problema a Resolver**: ¿Cómo asegurar la calidad del proceso de desarrollo de software en el proyecto SIGEP Fase IV?

Basado en lo antes expuesto se ha definido como **Objeto de Estudio** la Gestión de la Calidad de Software. Derivado de este, el **Campo de Acción** será el Aseguramiento de la Calidad de Software en el proyecto SIGEP Fase IV.

Para darle solución al problema planteado se propone como **Objetivo General**: Elaborar una estrategia para asegurar la calidad en el proceso de desarrollo de software del proyecto SIGEP Fase IV a través de la definición de un Plan de Aseguramiento de la Calidad.

Para darle solución al problema y cumplir el objetivo trazado se proponen las siguientes **Tareas de la Investigación**:

- ✓ Diseñar el estado del arte sobre la Gestión de la Calidad del Software.

- ✓ Estudiar estándares, normas y modelos para el aseguramiento de la calidad de software.
- ✓ Analizar el Plan de Aseguramiento de la Calidad de las fases anteriores del SIGEP.
- ✓ Elaborar una estrategia de aseguramiento de la calidad para el proyecto SIGEP Fase IV.
- ✓ Documentar la Estrategia de Aseguramiento de la Calidad propuesta en el Plan de Aseguramiento de la Calidad del proyecto SIGEP Fase IV.

Estructura de la tesis.

La tesis está conformada por tres capítulos. El Capítulo I se refiere al marco teórico relacionado a los conceptos de Calidad, Calidad de Software y Aseguramiento de la Calidad de Software además de estudiar los elementos necesarios para elaborar un Plan de Aseguramiento. El Capítulo II está dirigido a evaluar el plan que ha asegurado la calidad de las fases anteriores del SIGEP. En el Capítulo III se propone la Estrategia de Aseguramiento de la Calidad para la Fase IV del proyecto.

CAPÍTULO 1: FUNDAMENTACION TEORICA.

1.1. Introducción.

El impacto del software en nuestra sociedad y en la cultura continúa siendo profundo. Al mismo tiempo que crece su importancia, la comunidad del software trata continuamente de desarrollar tecnologías que hagan más sencillo, rápido y menos costosa la construcción de programas de computadoras de alta calidad. **(Pressman 1998)**

En este capítulo se muestra un acercamiento al tema de la calidad del proceso de desarrollo de software en el mundo y al mismo tiempo se abordan diferentes metodologías, prácticas y otros procedimientos que contribuyen a asegurar la calidad en el proyecto de software.

Se establecen diferentes definiciones y conceptos que son necesarios conocer para comprender el propósito de la nueva estrategia. Se dan a conocer conceptos y definiciones tales como: calidad, vista desde diferentes perspectivas, calidad de software, gestión de la calidad de software, aseguramiento de la calidad de software, métricas y estándares, pruebas y revisiones.

1.2. Calidad.

"La Calidad es satisfacción y precios competitivos para el cliente, y rentabilidad y sostenibilidad para la empresa".

(Lema)

Calidad es un concepto manejado con bastante frecuencia en la actualidad, pero a su vez, su significado es percibido de distintas maneras, sigue siendo ambiguo y muchas veces su uso depende de la comprensión que cada cual tenga en dependencia del marco en el que se encuentre, por lo cual es importante analizar diferentes definiciones.

Muchos han sido los autores que han aportado sus ideas sobre la calidad y sobre cómo la misma debe de ser aplicada. La calidad ha tomado un auge importante en los últimos años por

el papel que representa en el proceso de desarrollo de un proyecto y más aún cuando las Tecnologías de la Informática y las Comunicaciones (TIC) van creciendo vertiginosamente.

Uno de los muchos conceptos que han sido redactados referentes al tema lo otorga la **Real Academia de la Lengua Española (RAE 2006)** que define la calidad como: **“Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor. Condición o requisito que se pone en un contrato”**.

La norma **ISO 9000:2000**, plantea que la calidad es el **“Grado en que un conjunto de características inherentes cumple con unos requisitos” (ISO)**. También la **ISO/IEC 9126** define la calidad como: **“El conjunto total de características de una entidad (producto, proceso o servicio) que le confieren la capacidad de satisfacer las necesidades establecidas y las necesidades implícitas”**. (ISO/IEC 1991)

Definiciones dadas por otros autores:

- ✓ "El control de Calidad no significa alcanzar la perfección. Significa conseguir una eficiente producción con la calidad que espera obtener en el mercado". **(Deming)**
- ✓ "Adecuación al uso". **(Juran)**
- ✓ "Conformidad con los requisitos". **(Crosby)**
- ✓ "La composición total de las características de los productos y servicios de marketing, ingeniería, fabricación y mantenimiento, a través de los cuales los productos y los servicios es unos cumplirán las expectativas de los clientes". **(Feigenbaum)**
- ✓ "Calidad es la menor pérdida posible para la sociedad". **(Taguchi)**
- ✓ "Calidad es satisfacción del cliente". **(Deming)**
- ✓ "La calidad como resultado de la interacción de dos dimensiones: dimensión subjetiva (lo que el cliente quiere) y dimensión objetiva (lo que se ofrece). **(Shewhart)**

Además de este conjunto de definiciones, la calidad es vista desde diferentes perspectivas en dependencia del escenario en el que se encuentre.

Definiciones desde una perspectiva de producto.

La calidad de un producto está dada por la percepción del cliente objetivo del mismo, en función del conjunto de características que ese usuario final evalúa para el producto en cuestión, y del nivel significativo que cada una de ellas posee.

Definiciones desde una perspectiva de usuario.

La calidad implica la capacidad de satisfacer los deseos de las personas dentro de su estilo de vida, esto involucra un equilibrio entre lo objetivo/tangible y lo subjetivo/intangible, ofrecer características beneficiosas y saludables para las personas y su entorno. La calidad de un producto depende de cómo éste responda a las preferencias y a las necesidades de los clientes, por lo que se dice que la calidad es adecuación al uso de sí mismo en la actualización de los roles presentados a un consumidor.

Definiciones desde una perspectiva de las tecnologías de la información o calidad de datos.

La calidad de datos implica que los datos capturados, procesados, almacenados y entregados son un fiel reflejo de la realidad que se desea tratar mediante sistemas informáticos. Esto supone que los datos no contengan errores, sean veraces y estén actualizados.

Definiciones desde una perspectiva de producción.

La calidad puede definirse como el grado en que un producto cumple las especificaciones del diseño y la conformidad relativa con las especificaciones de producción.

Desde una perspectiva de valor.

La calidad significa aportar valor al cliente, esto es, ofrecer unas condiciones de uso del producto o servicio superiores a las que el cliente espera recibir y a un precio accesible.

También, la calidad se refiere a minimizar las pérdidas que un producto pueda causar a la sociedad humana mostrando cierto interés por parte de la empresa a mantener la satisfacción del cliente.

Con todo lo analizando se concluye que **la calidad es la aptitud de un producto o servicio para satisfacer las necesidades del usuario y es la cualidad de todos los productos, no solamente de equipos sino también de programas.**

La calidad de un producto o servicio es la percepción que el cliente tiene del mismo, es una fijación mental del consumidor que asume conformidad con dicho producto o servicio y la capacidad del mismo para satisfacer sus necesidades.

1.3. Calidad de Software.

Sin un buen proceso de desarrollo es casi imposible obtener un buen producto, por lo que es importante que la calidad del producto sea medida a lo largo de todo el ciclo de desarrollo del software. Sin embargo la **calidad del software** puede medirse después de elaborado el producto, pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

La obtención de un software con calidad implica la utilización de metodologías o procedimientos y estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

Ahora bien, enfoquémonos en el significado de la **calidad del software**. De cierto modo todos podemos hacernos una idea de este concepto, dado que es muy manejado por la alta demanda que presenta en estos último tiempos, ya que existen muchos productos de software a nivel mundial.

Una definición encontrada en **Internet (SEDNA 2006)** plantea que la calidad del software es el **“Nivel de fiabilidad, robustez y eficiencia del software referido a todo su comportamiento en todo el período de vigencia”**.

Enfocándose en el cliente, **calidad del software podría ser el grado en que un cliente y/o usuario percibe que el producto software satisface sus necesidades y expectativas (LÓPEZ 2002)**; pero en la condición industrial del producto, **calidad del software es la habilidad de un producto software de satisfacer su especificación de requerimientos**.

El profesor **Roger S. Pressman** da el siguiente concepto: **“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” (Pressman 1993)**.

La **ISO 8402 (ISO 1994)** expresa que la calidad de software es **“El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”**. Es importante destacar que en estos conceptos falta agregar el factor **tiempo** que es muy importante a la hora de desarrollar un producto de software porque hay que tener siempre presente los compromisos de entrega. Por lo tanto el primer compromiso que se debe de trazar un equipo de desarrollo de software es producir un software de máxima calidad y en el tiempo establecido.

La creación de un software con los atributos con los que fue planificado es una preocupación a la que se dedican muchos esfuerzos. Sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software con la mayor calidad, que cumpla, y si es posible, supere las expectativas de los usuarios. La **calidad del software** ha pasado de una simple inspección y detección de errores a un cuidado total en su proceso de fabricación, desarrollo y mantenimiento. El correcto funcionamiento de este, es fundamental para el óptimo comportamiento de los sistemas informáticos de las empresas.

Un software de calidad es aquel que debiera cumplir con los requerimientos funcionales y de rendimiento, además de ser mantenible, confiable y aceptable.

Se describen a continuación las principales características que crean a un software de calidad.

- ✓ **Mantenibilidad:** el software debe ser diseñado de tal manera, que permita ajustarlo a los cambios en los requerimientos del cliente. Esta característica es crucial, debido al inevitable cambio del contexto en el que se desempeña un software.
- ✓ **Confiabilidad:** incluye varias características además de la confiabilidad, como la seguridad, control de fallos, etc.
- ✓ **Eficiencia:** se refiere al uso eficiente de los recursos que necesita un sistema para su funcionamiento.
- ✓ **Usabilidad:** el software debiera ser utilizado sin un gran esfuerzo por los usuarios para los que fue diseñado, documentado, etc.

De lo anteriormente expuesto se concluye que la **calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia.**

1.4. Gestión de la Calidad de Software.

La gestión de la calidad se define de diversas maneras según diferentes autores. Como ejemplo podemos citar a **Carlos López (LÓPEZ 2002)** que plantea que: “...la **gestión de la calidad incluye la planificación estratégica, la asignación de recursos y otras actividades sistemáticas, tales como la planificación, las operaciones y las evaluaciones relativas a la calidad**”.

En el **Manual de Gestión de la Calidad Total a la Medida (MALEVSKI 1995)** se plantea que: “...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, las asignaciones de recursos y otras actividades sistemáticas, tales como los planes de calidad”.

La **ISO 9000 (ISO)** define: “**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.**”

Además de estos conceptos, la **ISO 9000:2000 (ISO)** (Sistemas de Gestión de la Calidad. Fundamentos y vocabulario) propone **ocho principios** de la gestión **de la calidad**, identificados para lograr los objetivos de la calidad. Estos ocho principios de gestión de la calidad constituyen la base de las normas de sistemas de gestión de la calidad de la familia de **Normas ISO 9000 (ISO)**.

- ✓ **Enfoque al cliente.** Las organizaciones dependen de sus clientes y por lo tanto deberían comprender las necesidades actuales y futuras de los clientes, satisfacer los requisitos de los clientes y esforzarse en exceder las expectativas de los clientes.
- ✓ **Liderazgo.** Los líderes establecen la unidad de propósito y la orientación de la organización. Ellos deberían crear y mantener un ambiente interno, en el cual el personal pueda llegar a involucrarse totalmente en el logro de los objetivos de la organización.
- ✓ **Participación del personal.** El personal, a todos los niveles, es la esencia de una organización y su total compromiso posibilita que sus habilidades sean usadas para el beneficio de la organización.
- ✓ **Enfoque basado en procesos.** Un resultado deseado se alcanza más eficientemente cuando las actividades y los recursos relacionados se gestionan como un proceso.
- ✓ **Enfoque de sistema hacia la gestión.** Identificar, entender y gestionar los procesos interrelacionados como un sistema, contribuye a la eficacia y eficiencia de una organización en el logro de sus objetivos.
- ✓ **Mejora continua.** La mejora continua del desempeño global de la organización debería ser un objetivo permanente de ésta.
- ✓ **Enfoque basado en hechos para la toma de decisiones.** Las decisiones eficaces se basan en el análisis de los datos y la información.

- ✓ **Relación mutuamente beneficiosa con el proveedor.** Una organización y sus proveedores son interdependientes, y una relación mutuamente beneficiosa aumenta la capacidad de ambos para crear valor.

De esta manera se concluye que **la Gestión de la Calidad no es más que todas las actividades de la función gerencial realizadas de forma periódica que determinan la política de calidad, los objetivos y las responsabilidades y que los ponen en práctica por medios tales como la planificación, el aseguramiento, el control y el mejoramiento de la calidad, dentro del sistema de calidad.**

1.5. Aseguramiento de la Calidad de Software.

Para comenzar a abordar sobre el **Aseguramiento de la Calidad**, (también conocido como **SQA** dado el origen anglosajón de la terminología: **Software Quality Assurance**) se debe conocer que es uno de los elementos de la ingeniería del software que tiene como principal objetivo el dar confianza al usuario de un software de que satisface los requisitos del cliente acordados con el Equipo de Desarrollo y que está libre de defectos.

A partir de este planteamiento es necesario profundizar en aspectos de vital importancia, tales como:

¿En qué consiste el Aseguramiento de la Calidad de Software y dónde está presente?

El desarrollo o mantenimiento de software, es un proceso que por su naturaleza abstracta y carente de visibilidad física, se puede volver complejo. El aseguramiento de calidad del software se diseña para cada aplicación antes de comenzar a desarrollarla. Hay quienes prefieren decir garantía de calidad en vez de aseguramiento. La garantía, puede confundirse con garantía de productos, mientras que el aseguramiento pretende dar confianza en que el producto tiene calidad. Por eso se debe conocer con profundidad el concepto de aseguramiento de la calidad para que pueda ser implantar con éxito.

Aseguramiento de la calidad de software: Un planeado y sistemático patrón de todas las acciones necesarias para proveer la adecuada confianza de las conformidades de los artículos o productos para establecer los requerimientos técnicos. **(IEEE 1998)**

El aseguramiento de calidad del software está presente en:

- ✓ Métodos y herramientas de análisis, diseño, programación y prueba.
- ✓ Inspecciones técnicas formales en todos los pasos del proceso de desarrollo del software.
- ✓ Estrategias de prueba.
- ✓ Control de la documentación del software y de los cambios realizados.
- ✓ Procedimientos para ajustarse a los estándares (y dejar claro cuando se está fuera de ellos).
- ✓ El Aseguramiento de la Calidad se realiza mediante diferentes **actividades** a lo largo del ciclo de desarrollo o mantenimiento. Se intercalan actividades y tareas específicas a lo largo del desarrollo o mantenimiento de una aplicación, con la intención de reducir errores.

¿Qué actividades conforman el Aseguramiento de la Calidad del Software?

Es conveniente destacar que hay dos tipos de **acciones** importantes: Las **prácticas** y las **acciones**.

Entre las **prácticas** que ayudan a desarrollar una aplicación sin defectos se destacan:

- ✓ La administración de requerimientos.
- ✓ La administración de las versiones de los diferentes elementos que forman el software (documentos, programas, etc.).
- ✓ Administración y autorización de cambios a las especificaciones iniciales del cliente.

- ✓ Administración del proyecto.
- ✓ Ciclo de desarrollo.
- ✓ Administración de riesgos.

Y entre las **acciones** específicas para asegurar la calidad se encuentran:

- ✓ Verificación de la elaboración de los productos intermedios o entregables de cada actividad o tarea.
- ✓ Inspección de los productos elaborados por un compañero en busca de defectos.
- ✓ Desarrollo de un prototipo.
- ✓ Pruebas de caja negra, de funcionalidad, de integración y de unidad.
- ✓ Pruebas de volumen y stress.

Estas **actividades** se puntualizan a través de:

- ✓ Métricas de software para el control del proyecto.
- ✓ Verificación y validación del software a lo largo del ciclo de vida (Incluye las pruebas y los procesos de revisión e inspección).
- ✓ La gestión de la configuración del software.
- ✓ Los elementos anteriores no garantizan el desarrollo o mantenimiento de una aplicación libre de defectos, sin embargo, reducen notablemente el riesgo de que ocurran.

Además de las acciones que han sido descritas anteriormente, el aseguramiento de la calidad define funciones y objetivos. Según la bibliografía consultada (**Pressman 1998**) los tres principales objetivos del SQA son:

- ✓ **Perfeccionar** la calidad del software monitoreando debidamente tanto los productos de software como el proceso de desarrollo que los genera.
- ✓ **Asegurar** el cumplimiento de los estándares y procedimientos establecidos para el software y el proceso de software establecidos.
- ✓ **Asegurar** que cualquier desviación en el producto, el proceso, o los estándares son elevados a la dirección del proyecto para poder resolverlas.

Aparte de estos objetivos se plantean **funciones** muy importantes que no pueden faltar en el **aseguramiento de la calidad** y que son aplicables para cualquier proyecto de software.

- ✓ *Evaluación de la planificación del proyecto de software:* si no se planifican prácticas de calidad adecuadas desde el inicio y sincronizadas con el plan del proyecto, luego no serán implementadas.
- ✓ *Evaluación de los requerimientos:* como es extremadamente inusual que se desarrollen productos de alta calidad a partir de requerimientos de baja calidad, los requerimientos iniciales deben ser revisados contra los estándares de calidad establecidos y pactados con anterioridad.
- ✓ *Evaluación del proceso de diseño:* se definen los medios para asegurar que el diseño siga las metodologías planificadas, que implemente los requerimientos y que la calidad del diseño propiamente dicha sea revisada independientemente.
- ✓ *Evaluación de las prácticas de codificación:* prácticas apropiadas de codificación deben ser establecidas y usadas.
- ✓ *Evaluación del proceso de integración y testeado del software:* se establece un programa de testing de calidad.
- ✓ *Evaluación del uso del proceso de control y gestión del proyecto:* asegurando que los procesos de gestión están funcionando.
- ✓ *Gestión de la Configuración del Software:* asegurando la validez de todo producto obtenido durante cualquiera de las etapas del desarrollo del software, a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad

constante de una versión estable de cada elemento para toda persona involucrada en el citado desarrollo. La gestión de la configuración se realiza durante todas las fases del ciclo de desarrollo del producto, incluyendo el mantenimiento y control de cambios, una vez empezada la fase de elaboración.

- ✓ *Evaluación de la aplicación de métricas* para evaluar y predecir la calidad del software mediante la aplicación de métricas de software.

Como conclusión de los planteamientos anteriores se puede afirmar que el **aseguramiento de la calidad** aborda principalmente un enfoque de gestión de la calidad; métricas del software; verificación y validación a lo largo del ciclo de vida del software, incluyendo pruebas y procesos de revisión y auditorías; gestión de configuración del software; y un procedimiento que asegure los ajustes a los estándares en el proceso de desarrollo de software siempre que esto sea posible.

1.6. Las Pruebas.

"La prueba puede probar la presencia de errores pero no la ausencia de ellos"

(Dijkstra)

Las pruebas de software se integran dentro de las diferentes fases del ciclo de creación del software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir qué errores tiene. Son los procesos que permiten verificar y revelar la calidad de un producto software.

Las pruebas de software es un proceso centrado en el objetivo de encontrar defectos a un software; puede ser por razones de depuración al software o de aceptación del mismo. Las pruebas son un elemento esencial en el desarrollo de software y ayudan a mejorar la calidad del sistema. No mejoran directamente la calidad del mismo, pero sí lo hacen indirectamente previendo un panorama claro de las debilidades del software observadas y de los riesgos asociados al proyecto.

Existe una regla universal, la cual indica que es imposible encontrar todos los defectos y que nunca hay suficiente tiempo, personal o dinero para probar todo. Se deben tomar decisiones inteligentes de cómo distribuir los recursos disponibles. **(Rivas Junio 2007)**

Los **objetivos más específicos** que persiguen las pruebas dada la bibliografía consultada son los siguientes **(Rivas Junio 2007)**

- ✓ La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
- ✓ Un buen caso de prueba es aquel donde se tiene una alta probabilidad de encontrar un error, no descubierto.
- ✓ Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Todas las pruebas no son aplicables a todas las ramas de construcción del software. Existen diferentes tipos de pruebas identificadas por el alcance que tienen y por los resultados que deben arrojar al ponerse en aplicación.

1.6.1. Las Pruebas Unitarias.

Las pruebas unitarias constituyen un proceso para validar que una porción del código del sistema funciona apropiadamente de manera aislada. Las pruebas unitarias proveen la capacidad para probar un módulo del software de forma reproducible, eficiente y automatizada.

Las pruebas unitarias fueron definidas como el nivel más fino de granularidad en las pruebas por Rob Johnson pues verifican una simple unidad de funcionalidad y deben probar que cada método de una clase satisface su contrato documentado. **(Duharte Junio de 2008).**

William E. Lewis dice: **“Prueba unitaria es la más micro escala de las pruebas; para probar funciones particulares o módulos de código. Típicamente hechas por desarrolladores y no por probadores. (...).”**(Duharte Junio de 2008).

Las pruebas de **Caja Blanca** y de **Caja Negra** son métodos de **Pruebas Unitarias**.

Pruebas de caja negra: estas pruebas solo consideran las interfaces públicas de las clases bajo prueba. No se basa en el conocimiento de los detalles de implementación.

Pruebas de caja blanca: las pruebas son conscientes del contenido interno de las clases bajo prueba. No sólo prueba lo que es requerido hacer por la clase, sino también cómo lo hace. Estas pruebas en ocasiones son llamadas pruebas de caja de cristal.

1.6.2. Pruebas de Integración.

La prueba de integración es una técnica sistemática que permite construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es juntar los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. Se combinan todos los módulos por anticipado. Se prueba todo el programa y se encuentra un gran conjunto de errores. La corrección se hace difícil, ya que es complicado aislar las causas al tener el programa entero en toda su extensión. Una vez que se corrigen esos errores aparecen otros nuevos y el proceso continúa su ciclo. Estas pruebas se pueden llevar a cabo mediante dos métodos: top-down o bottom up, o sea: de arriba hacia abajo que empieza por lo general, y de abajo hacia arriba que va uniendo pequeños fragmentos.

1.6.3. Pruebas de Funcionalidad.

Una vez que se ha desarrollado una aplicación, los desarrolladores necesitan verificar que está libre de anomalías y que se ha logrado el objetivo de su diseño. Las pruebas de funcionalidad determinan la extensión en la que la aplicación satisface los requisitos funcionales esperados. Este proceso simulará varios escenarios para confirmar que todos los resultados satisfacen las expectativas establecidas.

Además de estos objetivos, las pruebas de funcionalidad se trazan metas específicas:

- ✓ Verificar el procesamiento, recuperación e implementación adecuada de las reglas del negocio.
- ✓ Verificar la apropiada aceptación de datos.

Para la implementación y desarrollo de estas pruebas se siguen los métodos de caja negra donde se ejecuta cada caso de uso, flujo de caso de uso, o función, usando datos válidos e inválidos, para verificar lo siguiente:

- ✓ Que se aplique apropiadamente cada regla de negocio.
- ✓ Que los resultados esperados ocurran cuando se usen datos válidos.
- ✓ Que sean desplegados los mensajes apropiados de error y precaución cuando se usan datos inválidos.

1.6.4. Pruebas de Aceptación.

Cuando se construye software a la medida para un cliente, se llevan a cabo una serie de pruebas de aceptación para permitir que el cliente valide y verifique todos los requisitos que han sido establecidos. Estas pruebas las realiza el usuario final en lugar del responsable del desarrollo del sistema. Las pruebas de aceptación, conducidas por el cliente, verifican que el sistema satisface los requerimientos y son similares a las pruebas del sistema, pues se basan fundamentalmente en pruebas de funcionalidad.

Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable al cliente; sino que se realizan sobre el producto terminado e integrado, o pudiera ser una versión del producto o una iteración funcional pactada previamente con el cliente. Estas pruebas tienen una gran importancia, ya que definen el paso hacia nuevas fases del proyecto, como el despliegue y mantenimiento.

El proceso de pruebas de aceptación está basado en nuestra universidad en la presencia de un tercero confiable, en este caso el Laboratorio de Pruebas y Certificación, que está integrado por el grupo de calidad de la universidad, el cual se encarga de diseñar, guiar y dirigir las pruebas de aceptación.

1.7. Revisiones Técnicas Formales.

Existen diversos motivos para aplicar las **Revisiones Técnicas Formales (RTF)** en un proyecto de software, pero el fundamental, es que el trabajo técnico necesita ser revisado y existen errores que son percibidos más fácilmente por otras personas que por los creadores.

Es un grupo de personas que permiten (1) Señalar la necesidad de mejoras, (2) Señalar que NO hay que mejorar y (3) Conseguir un trabajo técnico más homogéneo. **(Quesada)**

Las **RTF** son un filtro que permite purificar las actividades de ingeniería de software. Se aplican en diversos momentos del desarrollo para detectar defectos. **(Alonso)**

Entre sus **objetivos** fundamentales se encuentran:

- ✓ Descubrir errores en la función, lógica o implementación de cualquier representación del software.
- ✓ Verificar el cumplimiento de los requisitos
- ✓ Garantizar el cumplimiento de los estándares.
- ✓ Conseguir un desarrollo uniforme del software.
- ✓ Obtener proyectos que hagan más sencillo los trabajos técnicos (análisis que permitan buenos diseños, diseños que permitan implementaciones sencillas, estrategias de pruebas que faciliten éstas,...)

Las revisiones formales tienen tres **elementos** (1) Informe escrito del estado del producto revisado, (2) La participación activa y abierta de todos los del grupo de revisión y (3) Total responsabilidad de todos los participantes en la calidad de la revisión. **(Quesada)**

Las **RTF** sirven para promover la seguridad y continuidad, ya que varias personas se familiarizarán con partes del sistema de información, que de otro modo, no hubieran visto. Es una clase de revisión que incluye recorridos, inspecciones, torneo de revisiones y otras tareas de revisión técnica de los sistemas.

La orientación de la **RTF** debe ser muy específica, en el sentido que cada **RTF** se debe centrar en una parte muy bien delimitada del software total. Por ejemplo, en lugar de intentar revisar un diseño completo, se hacen inspecciones para cada módulo o pequeño grupo de módulos. Al limitar el centro de atención de la **RTF** la probabilidad de descubrir errores es mayor.

Las **RTF** pueden llevarse a cabo durante las siguientes fases:

- ✓ Después del desarrollo de las especificaciones.
- ✓ Después del análisis y diseño del programa.
- ✓ Después del test de prueba.

A modo de conclusión se plantea que es importante aplicar las **RTF** en el proyecto porque está calculado que estas son un 75% efectivas, además (1) reducen sustancialmente el coste del software, (2) tienen gran valor educativo para los participantes, (3) sirven para comunicar la información técnica y (4) fomentan la seguridad y la continuidad.

1.8. Tareas de Calidad en RUP.

RUP propone una serie de aspectos sobre las revisiones que son importantes y hay que tener siempre en cuenta a la hora de realizar las mismas. A modo general plantea que se realicen las revisiones en forma de reunión y que se supervise continuamente la calidad durante las tareas de proceso para evitar grandes números de defectos hasta las revisiones (**Teams**).

El número de revisores debe ser aproximadamente siete, o menos. Más revisores reducen la calidad de la revisión, alargando las reuniones, dificultando la participación e inyectando

problemas laterales y discusión en la revisión. Menos de cuatro revisores aumenta el riesgo de miopía en la revisión, ya que se reduce la diversidad de la preocupación.

Los revisores deben tener experiencia en el área que se va a revisar, en caso de que haya algún revisor no la tenga, puede participar y aprender mediante ella, pero contribuirán poco en la revisión y su presencia puede crear distracciones. Además, se debe mantener el grupo reducido, más de tres personas y menos de siete. Menos revisores ponen en peligro la calidad de la revisión, y más revisores evitan la discusión interactiva esencial para alcanzar resultados de calidad.

Además de estos puntos, existen varios aspectos claves para realizar una revisión satisfactoria(**Teams**):

- ✓ Comprensión del proceso de revisión.
- ✓ Asegurarse de que los revisores comprenden sus roles.
- ✓ Tener un moderador.
- ✓ Mantener la brevedad de la revisión y seguir el orden del día.
- ✓ Identificar los problemas, no solucionarlos.
- ✓ No solo estas cuestiones son aplicables, hay que saber que la revisión no tiene valor si no se consigue nada de esta, al final de la revisión.
- ✓ Dar prioridad a la lista de problemas.
- ✓ Crear defectos para realizar el seguimiento de los problemas y de su resolución.
- ✓ Si es necesaria una investigación adicional, asigne un pequeño equipo para que investigue el problema (no para solucionarlo).
- ✓ Para problemas que se pueden resolver en la iteración actual, asigne a una persona o equipo para solucionar el problema.
- ✓ Rellene la lista de problemas no resueltos para esfuerzos de planificación de iteración futuros.

En cuanto a las **pruebas**, RUP plantea una serie de cuestiones importantes.

Las pruebas de software informático implican mucho más que la simple evaluación de las funciones, la interfaz y las características de tiempo de respuesta de un destino de la prueba. Las pruebas deben centrarse en características y atributos, como el destino de la prueba.

- ✓ integridad (resistencia a los errores).
- ✓ capacidad para instalarse y ejecutarse en plataformas diferentes.
- ✓ capacidad para manejar varias solicitudes al mismo tiempo.

Para conseguir esto, deben implementarse y ejecutarse muchos tipos diferentes de pruebas. Cada tipo de prueba tiene un objetivo específico y una técnica de soporte. Cada técnica se centra en la prueba de uno o varios atributos o características del destino de la prueba.

Las pruebas se aplican a diferentes tipos de destinos, en fases o niveles diferentes de esfuerzo de trabajo. Estos niveles suelen distinguirlos los roles que están más capacitados para diseñar y dirigir las pruebas, donde las técnicas son más adecuadas para realizar la prueba en cada nivel. Es importante asegurarse de hay un equilibrio entre los diferentes esfuerzos de trabajo.

Es necesario determinar adecuadamente la secuencia y el tiempo de los niveles de prueba. Normalmente, se cree que las pruebas de unidad se implementan al principio de la iteración, como primera fase de pruebas: es necesario que se aprueben todas las unidades antes de que pasar a las fases posteriores. Sin embargo, en un proceso de desarrollo repetitivo, este enfoque es, por regla general, inadecuado. Es más apropiado identificar las pruebas de unidad, de integración y del sistema que ofrecen más potencial para detectar errores y, a continuación, implementarlas y ejecutarlas en función de una combinación de un entorno de mayor riesgo y soporte.

1.9. Normas, Modelos y Estándares.

La obtención de un software con calidad implica la utilización de metodologías o procedimientos, estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

Los estándares de calidad son aquellos que permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Los estándares suministran los medios para que todos los procesos se realicen de la misma forma y son una guía para lograr la productividad y la calidad. **(Piattini 2003)**.

Los **Modelos de Calidad** son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad. **(Piattini 2003)**

Implantar modelos o estándares de calidad tiene como objetivo principal que se desarrollen sistemáticamente productos, bienes y servicios de mejor calidad y que estos cumplan con las necesidades y deseos de los clientes. Para esto, se requiere de un modelo / estándar que permita tener procesos y procedimientos ágiles y comprensibles para todos los involucrados, pasando por las etapas de desarrollo, prueba, producción y satisfacción del cliente.

Varias de las normas y estándares de calidad que constituyen puntera en los procesos de aseguramiento de la calidad de un proyecto de software son:

1.9.1. Normas ISO 9000.

Las normas ISO 9000, son un conjunto de enunciados que conforman un modelo para desarrollar un sistema de gestión de la calidad en una empresa. En este caso ha sido enmarcada en lo que nos ocupa, que es el proceso de desarrollo de un proyecto de software.

La familia de estándares ISO 9000, creada por ISO (Organización Internacional para la Estandarización) establece los lineamientos y guía el aseguramiento de calidad de los productos, servicios y relaciones con el cliente.

Cabe señalar que las normas ISO 9000 se refieren al sistema de gestión de calidad y no a las especificaciones para la elaboración de un producto. Es decir, lo que se logra es homologar sistemas de calidad para los productos.

Según lo expresado en las normas, la participación de recursos para transformar requisitos de las partes interesadas en productos de acuerdo a sus demandas representa un enfoque basado en procesos.

1.9.2. IEEE 730.

El estándar IEEE 730 define un conjunto de requerimientos para llevar a cabo un Plan de Aseguramiento de Calidad asociado a un Proyecto de Software y establece una relación entre la Gestión de la Calidad y la Ingeniería del Software. Es importante señalar que el IEEE- 730 es aplicado a un único proyecto dentro de esa organización lo que lo diferencia de otros estándares como el ISO 9000-3 que está pensado para ser aplicado en toda una organización.

El IEEE 730 mantiene relaciones significativas con otros estándares como el ISO 9003 y el ISO 9001. Dicha relación queda representada en la siguiente figura:

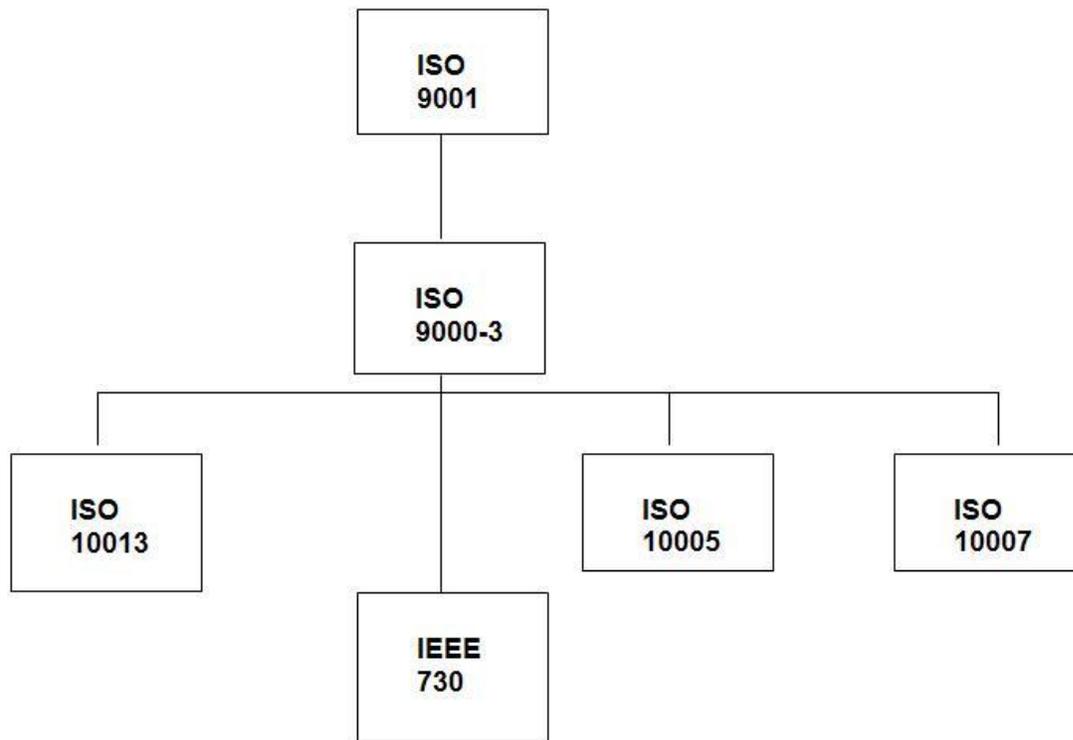


Figura 1: Relación de la IEEE 730 con otros estándares.

1.9.3. CMMI.

CMMI son las siglas que identifican al **Capability Maturity Model Integration**, es un modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software (**CMMI for Development August 2006; CMMI August 2006**). Consiste en las mejores prácticas que aborden el desarrollo y las actividades de mantenimiento aplicadas a los productos y servicios.

El **CMMI**, exige un enfoque riguroso de Gestión de la Calidad, donde las organizaciones deben tener sus procesos definidos, documentados y controlados en todas sus partes, además de que los procesos para cada proyecto están basados en el proceso organizacional, mediante ajustes a las necesidades del proyecto y estos procesos puedan ser medidos para permitir un mejoramiento continuo de la calidad. (**VALERIS Mayo 2007**).

El glosario de **CMMI** define un proceso como actividades que pueden ser reconocidas como implementaciones de prácticas en un modelo. Estas prácticas pueden mapearse a una o más prácticas de las áreas de procesos de CMMI para permitir la utilidad de un modelo en la mejora y evaluación de los procesos (**CMMI for Development August 2006**).

El modelo **CMMI** está formado por dos representaciones, el **continuo** y el **escalonado**. Para la representación continua se utiliza los términos nivel de capacidad o **áreas de procesos** de capacidad. Para la representación escalonada se utiliza los términos **nivel de madurez** o madurez organizacional. CMMI cuenta con 5 niveles de capacidad y 22 áreas de procesos. Estas áreas de procesos están divididas en **categorías** (Gestión de Procesos, Gestión de Proyectos, Ingeniería y Soporte). En la categoría de **Soporte** se encuentra el área de proceso que nos ocupa, **Garantía de la Calidad del Procesos y del Productos (PPQA)**. Esta área de proceso se encuentra ubicada en el nivel 2 de madurez, en el cual el proceso está administrado.

1.9.4. Garantía de la Calidad de Procesos y de Productos (PPQA).

El **propósito** que persigue la **Garantía de la Calidad del Proceso y del Producto (PPQA)** es proporcionar a los equipos de trabajo y a la dirección, visibilidad objetiva de los procesos y de sus productos asociados (**CMMI for Development August 2006**).

- ✓ **Evaluar objetivamente procesos y productos:** se evalúa objetivamente la adherencia de los procesos implantados y de sus productos y servicios asociados contra las descripciones del proceso, los estándares y los procedimientos aplicables.
- ✓ **Proporcionar visibilidad objetiva:** los incumplimientos o no-conformidades se supervisan con objetividad, son comunicados y se asegura su resolución.

Por estas razones cuando no existe un procesos de **PPQA** o cuando no es correctamente implementado: (1) No se sabe qué procesos se siguen en la organización, (2) No se sabe si se respetan los estándares definidos para los productos, (3) Los mismos productos se entregan con niveles diferentes de calidad, (4) Los buenos productos no pueden garantizar buenos

procesos organizacionales y (5) Se paga un alto costo financiero y de posicionamiento en el mercado por la falta de calidad. **(CMMI for Development August 2006)**

El **contexto** de **PPQA** está determinado por las **metas y las prácticas específicas** que agrupan los objetivos fundamentales del mismo a los cuales se les da cumplimiento.

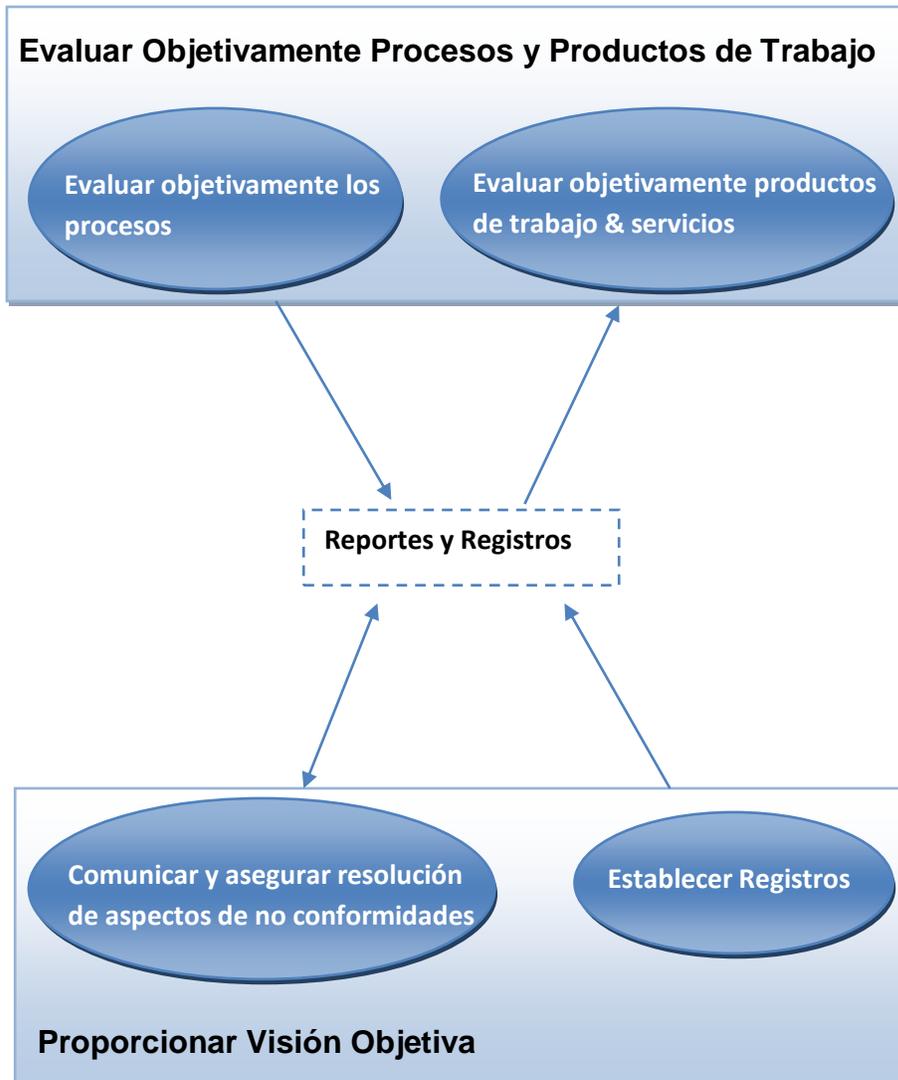


Figura 2: Contexto de PPQA

Metas específicas (CMMI for Development August 2006)

SG1: Evaluar objetivamente los procesos y los productos de trabajo

Evaluar objetivamente la adherencia de los procesos, los productos de trabajo asociados y los servicios, contra la descripción de dichos procesos, los estándares y los procedimientos.

SG1 Evaluar objetivamente los procesos y los productos de trabajo

SP 1.1 Evaluar objetivamente los procesos

SP 1.2 Evaluar objetivamente los productos de trabajo y los servicios

SP1.1 Evaluar objetivamente los procesos.

Evaluar objetivamente una selección de procesos ejecutados contra las descripciones de los procesos, estándares y procedimientos aplicables.

Objetivamente la evaluación del aseguramiento de la calidad es crucial para el éxito del proyecto.

Subprácticas:

- ✓ Promover un ambiente (creado como parte de la administración del proyecto) que motive la participación de los empleados en identificar y reportar asuntos de calidad
- ✓ Establecer y mantener criterios claros para las evaluaciones
 - Qué se evaluará
 - Cuándo y con qué frecuencia se harán las evaluaciones
 - Cómo se llevarán a cabo las evaluaciones
 - Quién debe participar
- ✓ Usar los criterios establecidos para evaluar la adherencia de procesos ejecutados contra las descripciones de los procesos, estándares y procedimientos
- ✓ Identificar cada no-conformidad encontrada durante la evaluación

- ✓ Identificar lecciones aprendidas que pueden mejorar los procesos para futuros productos y servicios

SP1.2 Evaluar objetivamente los productos de trabajo y los servicios.

Evaluar objetivamente los productos de trabajo y servicios seleccionados contra la descripción del proceso, estándares y procedimientos aplicables.

Subprácticas:

- ✓ Seleccionar los productos de trabajo a ser evaluados, basado en un criterio documentado de muestreo si se usan muestras.
- ✓ Establecer y mantener criterios claros para las evaluaciones de los productos de trabajo
 - Qué se evaluará
 - Cuándo y con qué frecuencia se harán las evaluaciones
 - Cómo se llevarán a cabo las evaluaciones
 - Quién debe participar
- ✓ Usar los criterios establecidos durante las evaluaciones de los productos de trabajo.
- ✓ Evaluar los productos de trabajo antes de que sean entregados al cliente.
- ✓ Evaluar los productos de trabajo en los hitos seleccionados durante su desarrollo.
- ✓ Realizar evaluaciones durante su progreso o incrementales de los productos de trabajo y de los servicios contra las descripciones de procesos, estándares y procedimientos.
- ✓ Identificar cada caso de no-conformidad encontrado durante las evaluaciones
- ✓ Identificar lecciones aprendidas que pueden mejorar los procesos para futuros productos y servicios

La **objetividad** de **PPQA** es crucial para los proyectos y es obligatoria, no así la independencia que es deseable y que varía según las organizaciones. La objetividad se logra de dos maneras:

- Evaluando con base en criterios definidos
- Evaluando de manera independiente

SG2: Proveer una percepción clara y objetiva

Monitorear y comunicar objetivamente las no-conformidades encontradas durante las evaluaciones hasta garantizar su resolución.

SP2.1 Comunicar y asegurar la resolución de las no-conformidades

Comunicar los asuntos de calidad encontrados y asegurar la resolución de los asuntos de no-conformidades con la gerencia y con el equipo de trabajo.

Las no conformidades son problemas identificados en evaluaciones que reflexionan una necesidad de adherencia para estándares aplicables, descripciones de procesos o procedimientos.

Subprácticas:

- ✓ Resolver cada no-conformidad con los miembros del equipo apropiados cuando sea posible
- ✓ Documentar los asuntos de las no conformidades cuando no puedan ser resueltos dentro del proyecto
- ✓ Escalar los asuntos de las no conformidades que no puedan resolverse dentro del proyecto, al nivel de administración apropiado y designado para recibir y actuar en asuntos de no conformidades
- ✓ Analizar los asuntos de no-conformidades para ver si existe alguna tendencia de calidad que pueda ser identificada y atendida
- ✓ Asegurar que los agentes relevantes están al tanto oportunamente de los resultados de las evaluaciones y de las tendencias de calidad
- ✓ Revisar periódicamente los asuntos abiertos de no-conformidades y las tendencias con el administrador designado para recibir y actuar en asuntos de asuntos de no-conformidades
- ✓ Rastrear los asuntos de no-conformidades hasta su cierre

Para resolver las **no-conformidades** se corrige directamente la causa de la no conformidad, modificar la descripción del proceso, los estándares o procedimientos para mejorarlos en función del análisis de la no-conformidad. También es necesario obtener y documentar el permiso para evadir la no conformidad y escalar la no-conformidad para que la alta gerencia opine y apoye la solución de la no-conformidad.

SP2.2 Establecer registros

Establecer y mantener registros de las actividades del aseguramiento de la calidad.

Subprácticas:

- ✓ Registrar los procesos y las actividades del aseguramiento de calidad del producto, en detalle suficiente que el estatus y los resultados sean conocidos
- ✓ Revisar el estatus y la historia de las actividades del aseguramiento de calidad como sea necesario

PPQA es un área de proceso clave, que a veces no se le da la importancia que requiere, sin embargo, con ella se tiene la posibilidad de detectar errores en el desarrollo de software antes de que se produzcan y así lograr un modelo de calidad.

La **UCI**, como la mayor productora de software del país, define un conjunto de pautas para lograr lo antes expuesto. Dentro de estas, se encuentran los Lineamientos de Calidad. Los mismos son elaborados por la DCS y han pasado por varias versiones hasta llegar a la V 2.0.

Estos lineamientos son: (1) Planificar capacitación para el personal del proyecto, (2) Establecer registro de resultados de investigación del proyecto, (3) Definir roles y responsabilidades, (4) Definir las competencias, (5) Definir equipos de proyecto, (6) Cuidar los bienes del cliente, (7) Ingeniería, (8) Gestión de proyectos y (9) Soporte.

Dentro de **Ingeniería** se encuentran: (1) Gestionar requerimientos, (2) Desarrollar requisitos, (3) Definir arquitectura de software, (4) Definir arquitectura de información, (5) Definir modelo de diseño, (6) Definir estándares para el desarrollo del proyecto, (7) Planificar y ejecutar pruebas del proyecto, (8) Efectuar seguimiento de las No Conformidades y (9) Diseñar artefactos de prueba.

Dentro de **Gestión de Proyectos** se encuentran: (1) Definir proyecto técnico para el proyecto, (2) Definir visión del proyecto, (3) Estimar costo y esfuerzo del proyecto, (4) Desarrollar plan del proyecto, (5) Definir plan de resultados del proyecto, (6) Establecer horarios de trabajo de los miembros del proyecto, (7) Identificar riesgos del proyecto y establecer plan de mitigación de los mismos, (8) Gestión de recursos y (9) Registrar los acuerdos de trabajo y las minutas de las reuniones.

Dentro de **Soporte** se encuentran: (1) Plan de aseguramiento de la calidad, (2) Definir Plan de Gestión de la Configuración de Software, (3) Definir configuración de la metodología a utilizar y (4) Utilizar herramienta CASE.

Estos lineamientos de calidad serán aplicables a todos los proyectos productivos de la Universidad.

1.10. CONCLUSIONES:

La calidad es un elemento imprescindible en la creación de un proyecto de software, por esta razón los esfuerzos que se realizan para conseguirla nunca son suficientes y el equipo no puede quedarse conforme, siempre hay que hacer más con mayor calidad.

Los desarrolladores deben trabajar y progresar apoyados y guiados por un equipo designado para garantizar la calidad y tratar que se adopte una filosofía de trabajo para llegar a un entendimiento, comenzando primeramente con el establecimiento de un propósito común para todos. Debe ser establecida una metodología de trabajo de calidad que organice todas las actividades que se siguen en el proceso de garantía de la calidad del software.

De forma general el software será considerado de calidad cuando:

Los procesos de calidad queden establecidos a partir de metodologías de calidad probadas y aprobadas. Sean realizadas pruebas al software para erradicar la mayor cantidad posible de errores y revisiones para supervisar el trabajo técnico. Se hayan establecido Normas, Modelos y Estándares de calidad que rijan la uniformidad en la filosofía de trabajo. Se garantice la calidad de los procesos y de los productos. Todo esto asegurado por una buena coordinación del trabajo realizado por todo el equipo de desarrollo del software.

CAPITULO 2: ANALISIS AL PLAN DE ASEGURAMIENTO DE LA CALIDAD SIGEP-2007.

2.1. Introducción.

Con el objetivo de identificar los puntos débiles del Plan de Aseguramiento de la Calidad de las fases I, II y III del SIGEP y erradicar los mismos, este fue sometido a un profundo análisis. Se realizaron además entrevistas a diferentes miembros del proyecto y se tomaron en cuenta las experiencias de las autoras de este trabajo.

2.2. Plan de Aseguramiento de la Calidad.

Un Plan de Aseguramiento de la Calidad (PAC), enmarcado en un proyecto, en este caso el SIGEP, es definir y describir todos aquellos requisitos que este ha de cumplir, para desarrollar con eficacia y corrección las tareas de aseguramiento de la calidad.

El desarrollo de un PAC permite definir métodos concretos de control y gestión de calidad en un proyecto, entregando herramientas de planificación, gestión y ejecución del aseguramiento de la calidad en todos los procesos, permitiendo trabajar con método y visión de mejora continua.

En el proyecto SIGEP se estableció un plan de calidad, el cual a lo largo de este capítulo llamaremos: **PAC SIGEP – 2007**. Dicho plan no contempla en su totalidad los puntos contenidos en la plantilla incluida en el Expediente de Proyecto definida en ese momento, ni aplica los lineamientos de calidad establecidos por la Universidad.

De manera general se evidencia que el PAC SIGEP – 2007 es un documento deficiente por no contener todas las secciones establecidas por la IEEE 730. Dichas secciones son: Propósito y Alcance del documento, Gestión, Documentación, Estándares y Guías, Revisiones, Auditorías y Pruebas. Se identifican además los métodos y las herramientas que soportan las actividades y tareas del aseguramiento de la calidad.

De estas secciones, las que se encuentran documentadas en el PAC SIGEP - 2007 son pobres en cuanto a la explicación de su contenido, siendo poco entendible para aquellas personas que

vayan a hacer uso del mismo. Todo lo anterior evidencia que se seguía una estrategia de calidad, pero que no era eficiente ni se adaptaba completamente al proceso de desarrollo.

Para usar los elementos correctamente desarrollados y mejorar los deficientes, se procede a analizar más detalladamente los puntos que conforman el PAC SIGEP - 2007.

2.3. Análisis de los puntos del PAC SIGEP – 2007.

2.3.1. Propósito.

Todo plan de calidad se traza con un propósito específico. Dicho propósito debe quedar plasmado en el mismo para que el equipo de desarrollo o cualquier personal ajeno que lo lea comprenda claramente lo que se recoge en este.

En el PAC SIGEP – 2007 se establece un propósito que es hasta cierto punto verdadero, aunque ambiguo, faltan ideas y la manera de comentar las existentes no es apropiada.

2.3.2. Objetivos de Calidad.

Los objetivos de calidad presentes en cada PAC son dependientes del proyecto para el cual ha sido elaborado el plan. Este aspecto el PAC SIGEP – 2007 lo tiene desarrollado satisfactoriamente, tiene elaborado los objetivos en dependencia de los que han sido trazados para el proyecto.

Con la nueva versión del PAC SIGEP-Fase IV se pretende continuar dando cumplimiento a estos objetivos. Serán agregados nuevos objetivos, debido a que esta nueva fase tiene definidos requisitos distintos que los clientes esperan que sean cumplidos y respetados, por tanto son metas para el equipo de calidad que estos requerimientos sean tenidos en cuenta y tratados como objetivos de calidad para continuar asegurando su cumplimiento.

2.3.3. Gestión y Organización.

Los roles involucrados en el aseguramiento de la calidad deben estar plasmados en el plan con una explicación de sus responsabilidades y los conocimientos básicos que cada uno de ellos debe tener.

El PAC SIGEP – 2007 tiene incluidos los roles encargados del aseguramiento de la calidad del software bien especificados y detallados. Esto facilita que en el momento de ubicar a cada individuo en el rol determinado para él, se esté haciendo una elección correcta, además garantiza una organización más eficiente y evita que se queden tareas sin responsable.

2.3.4. Tareas y Responsabilidades.

Uno de los elementos que tienen que estar incluidos en este documento son las tareas a realizar a lo largo del ciclo de desarrollo del software y el encargado de las mismas.

El PAC SIGEP – 2007 comprende muchas de las actividades necesarias para el aseguramiento de la calidad del software aunque se hace necesario incluir otras ausentes en el mismo hasta el momento.

A las tareas que se encuentran presentes les faltan detalles como los siguientes:

- ✓ Deberían estar reflejadas en un orden específico de cómo se hace cada una, dado que en ocasiones, para poder comenzar una tarea se hace necesario que ya esté completa una tarea anterior.
- ✓ Deben quedar reflejadas las condiciones previas a cada una. Es decir, qué tarea tiene que haber sido completada cuyos artefactos de salida deben estar en manos del responsable de calidad, además del nombre del rol responsable de dicha tarea.

- ✓ Se deben plasmar las post condiciones a partir del cumplimiento de cada una de las tareas, además de los artefactos generados en su realización, garantizando que el responsable de la tarea obtenga los resultados plasmados en el plan de calidad.
- ✓ Cada tarea debe estar acompañada de una descripción que explique lo que se hace en la misma y las herramientas usadas en su realización.

2.3.5. Documentación.

Todo plan de calidad tiene que sustentarse en documentos donde se van reflejando los resultados alcanzados a lo largo del proceso de calidad. El PAC SIGEP – 2007 no tiene especificados estos artefactos que lo conforman. Los mismos tienen que formar parte del documento, especificando el rol del responsable y la ubicación de este en el repositorio. El hecho de que estos no estén en el PAC SIGEP – 2007 trae consigo incongruencias y pérdida de información en los procesos de la calidad del software.

2.3.6. Métricas.

En el PAC SIGEP – 2007 no son reflejadas las métricas usadas en el ciclo de desarrollo del software. Debe estar descrito el tipo de métrica usada y las especificaciones de cada una de ellas, puede ser a manera de tabla o en forma de párrafo.

2.3.7. Estándares y Guías.

Los estándares usados en el PAC SIGEP – 2007 han sido nombrados en el documento aunque se dan pocos detalles sobre los mismos.

Es de gran utilidad saber en qué etapa del desarrollo del software se aplica cada uno y un breve comentario sobre las utilidades de este, así como cualquier especificación que se crea

necesario plasmar con el fin de hacer más claro para los desarrolladores y el equipo de calidad el por qué del uso de un determinado estándar.

2.3.8. Plan de Revisiones y Auditorías.

2.3.8.1. Tareas generales de Revisiones y Auditorías.

A lo largo del ciclo de desarrollo del software es necesario que se realicen revisiones, y se actualicen una serie de artefactos que son generados en cada paso que se da en la fabricación del mismo.

En el PAC SIGEP – 2007 no se mencionan estas revisiones, lo que no significa que no se realizaran. En el proyecto se realizó una sola revisión y fue a la captura de requisitos, llevada a cabo por el equipo de calidad. Una sola revisión no asegura la calidad eficientemente, por lo que serán agregadas las revisiones necesarias.

Las revisiones deben estar en el plan de aseguramiento de la calidad de una forma detallada poniendo una descripción de la misma y la lista de los artefactos involucrados.

Respecto a las auditorías, el PAC SIGEP-2007 no contiene detalles de las mismas, lo que nos demuestra que si fueron realizadas, no fueron documentadas.

2.3.8.2. Cronograma.

El PAC SIGEP – 2007 no tiene comprendido un cronograma de las fechas específicas en las que realizan las diferentes revisiones y auditorías del proyecto. Se hace necesario que estas actividades estén programadas para una eficiente ejecución de las mismas y que sean asignadas al miembro del equipo de calidad responsable de efectuarlas.

2.3.8.3. Resolución de problemas y actividades de corrección.

Este punto se encuentra desarrollado de manera eficiente en el PAC SIGEP-2007. Es una manera eficaz de darle solución a los problemas que han sido encontrados producto de las diferentes actividades de aseguramiento de la calidad.

El poseer una vía de solución garantiza que las dificultades que atentan contra la integridad del proceso de desarrollo del proyecto sean tratadas y erradicadas.

2.3.8.4. Proceso de las Revisiones y Auditorías.

El PAC SIGEP-2007 no muestra esta sección en la cual deben de estar descritos los pasos que se persiguen para realizar el proceso de las distintas actividades de revisiones y auditorías en el proyecto. Debe de estar presente además una descripción de las listas de chequeo a utilizar en cada una de estas actividades y los atributos de calidad que serán abordados en cada una de ellas.

Dado que no existen documentos que indiquen que alguna vez estas actividades fueron hechas y de qué modo se llevaron a cabo, no se garantiza un proceso eficiente de las mismas.

2.3.9. Pruebas.

El PAC SIGEP-2007 no contiene desarrollada esta sección, ni se encuentra referenciada la documentación existente sobre las pruebas. Esto no significa que las mismas no fueran realizadas, el equipo de calidad efectuaba las pruebas de funcionalidad, pero para garantizar un eficiente proceso de desarrollo no es suficiente.

Sobre estas pruebas de funcionalidad se contaba con poca capacitación, se tenía una metodología de cómo redactar casos de prueba a partir de casos de uso. Esta no era lo suficientemente explicativa para aplicarla de manera adecuada. La lista de chequeo en la mayoría de las ocasiones estaba incompleta pues no trataba todos los errores que se

encontraban y las no conformidades no se documentaban en una plantilla, como se establece por parte de la Universidad.

2.3.10. Herramientas, Técnicas y Metodologías.

La sección que especifica las herramientas con las cuales se trabajaba en el proyecto y mediante qué técnicas y metodologías se regía el mismo, no se encuentra documentada de manera eficiente en el PAC SIGEP-2007.

Es importante que se enumeren todas las herramientas que fueron utilizadas, porque además de las explicadas, en el proyecto se utilizan otras, de las cuales es necesario que se conozcan los detalles para un uso eficiente de ellas.

2.3.11. Gestión de Configuración.

Dada la importancia que tiene la gestión de la configuración para el proyecto es necesario que el equipo de calidad esté al tanto de la misma. En esta sección se hace una referencia al Plan de Gestión de Configuración que se encuentra en el Expediente de Proyecto del SIGEP.

En el PAC SIGEP-2007 no está referenciado este plan y no se le da el tratamiento que necesita por parte de los miembros de calidad del proyecto.

2.3.12. Registros de Calidad.

El PAC SIGEP-2007 contiene la mayoría de los documentos que se guardarán en este registro. En el proyecto se encuentran otros de importancia que deben estar especificados y no vienen contenidos en el registro que se encuentra actualmente.

2.3.13. Entrenamiento.

El PAC SIGEP-2007 no contiene esta sección, razón por la cual se desconocen las actividades de entrenamiento al equipo de calidad. Dada la experiencia de las autoras de este trabajo como miembros de este equipo, se asegura que no se les impartieron cursos de entrenamiento a los integrantes del equipo de calidad.

2.4. CONCLUSIONES

Se han analizado todos los puntos necesarios que debe comprender un PAC eficiente. Es válido aclarar que una de las razones por las cuales se desarrolla esta trabajo de diploma, es para evitar el desconocimiento que se ha evidenciado a través del análisis anterior sobre los detalles de los diferentes elementos y actividades que forman parte del proceso de aseguramiento de la calidad del SIGEP.

Como hasta el momento se ha evidenciado, el plan con el que contaba el SIGEP, no era el más adecuado para un proyecto de esta envergadura.

CAPITULO 3: ESTRATEGIA DEL PAC – SIGEP FASE IV

3.1. Introducción.

“La Calidad es habilidad, talento, profesionalismo. La Calidad es un Arte.”

Anónimo.

El objetivo del presente capítulo es proponer una estrategia, la cual se va a documentar a través de un nuevo PAC, que solucione los elementos que no recibieron el tratamiento adecuado hasta el momento, que aporte otros que serán propuestos y que aproveche además los ya descritos de manera correcta en el PAC SIGEP-2007.

El PAC documenta la estrategia descrita en este capítulo y será referenciado a partir de este momento como: **Plan de Aseguramiento de la Calidad para SIGEP – Fase IV (PAC SIGEP – Fase IV).**

3.2. Estrategia de Calidad.

La nueva estrategia se encuentra plasmada en el PAC SIGEP-Fase IV, la cual sigue la plantilla de Aseguramiento de la Calidad en su versión 2.0. La misma viene implícita dentro del Expediente de Proyecto V 2.0 definido y aprobado por la Dirección de Calidad de Software (DCS) de la Universidad.

El proceso de aseguramiento de la calidad se mantendrá a lo largo de todo el ciclo de desarrollo del proyecto, dándole un tratamiento especial a las No Conformidades encontradas en cada una de las actividades.

En el PAC SIGEP-Fase IV se incluyen íntegramente los objetivos de calidad determinados en el PAC SIGEP-2007, que son los propuestos en el proyecto SIGEP. Este aspecto es el único que no sufrió cambios en la elaboración de la nueva estrategia. Otros, por su parte, sufrieron

modificaciones buscando su optimización como: (1) Gestión del Equipo de Calidad, (2) Métricas, (3) Estándares y Guías, (4) Tareas generales de revisiones y auditorías, (5) Resolución de problemas y actividades de corrección, (6) Herramientas, técnicas y metodologías y (7) Registros de calidad.

A modo general esta estrategia propone nuevas pautas a seguir para garantizar la calidad en el proyecto SIGEP. Las mismas están basadas en adicionar elementos como: (1) Cambios en la estructura del equipo de calidad, (2) Tareas de Prevención, Evaluación y Control de Fallos Internos regidas por nuevas metodologías, (3) Uso de nuevos estándares y guías de calidad, (4) Uso de nuevas métricas, (5) Referencia a documentos imprescindibles en las actividades de aseguramiento de la calidad y (6) Entrenamiento al Equipo de Calidad.

3.2.1. Organización de los Roles en el Proyecto.

Es necesario determinar cómo va a estar estructurado el grupo de personas que aplicarán el plan de aseguramiento de la calidad.

El equipo de calidad estará estructurado por roles específicos, con los conocimientos necesarios para realizar las actividades que se propondrán para cada uno.

Todo el equipo de desarrollo responde por la calidad de las tareas que ejecuta, cada integrante del proyecto es responsable de la calidad del proyecto y contribuye con la misma.

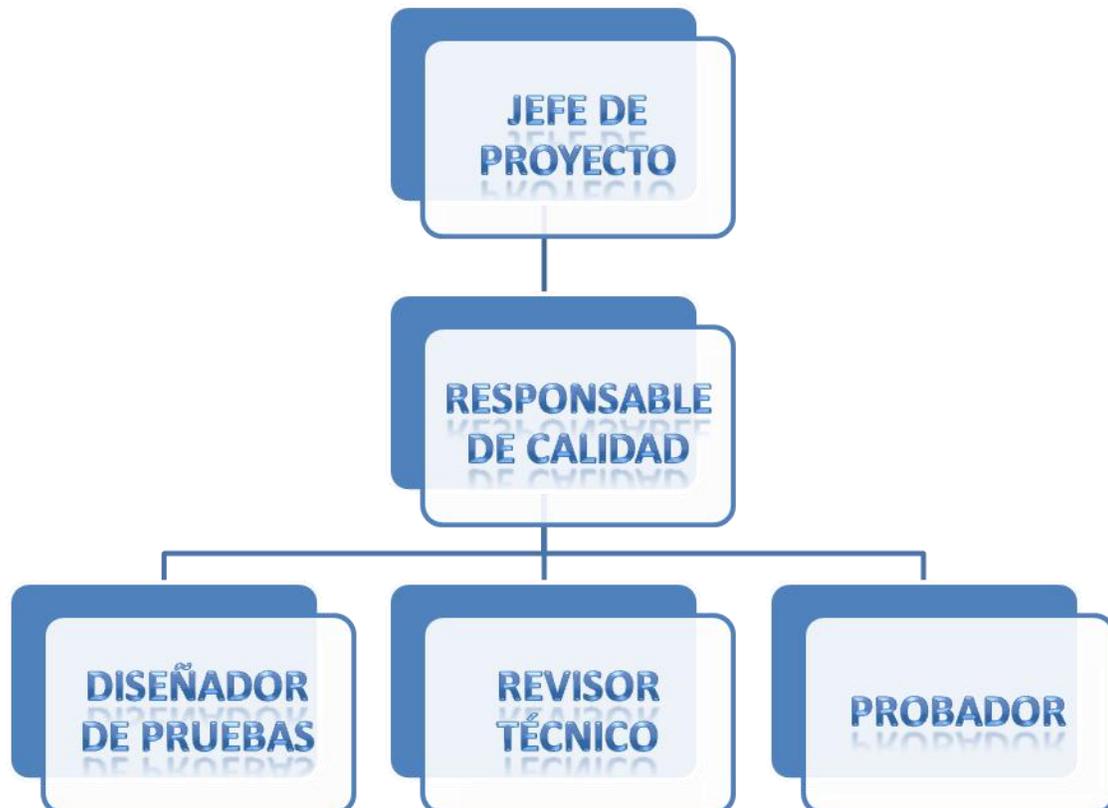


Figura 3: Roles responsables del Equipo de Calidad del SIGEP.

Esta organización propone que el aseguramiento de la calidad sea responsabilidad de todo el equipo de desarrollo del proyecto, siendo el **Jefe de Proyecto** el máximo responsable en el proceso de Aseguramiento de la Calidad. Sus responsabilidades serán: (1) Aprobar las tecnologías a usar en el desarrollo, (2) Aprobar la metodología que se va a aplicar, (3) Supervisar y apoyar todas las tareas de aseguramiento de la calidad y (4) Gestionar los recursos y materiales necesarios para el proyecto.

El **Responsable de Calidad**: (1) Organiza y coordina todas las tareas de aseguramiento de la calidad (2) Asegura que la aplicación producida se ajuste a las especificaciones y esté libre de errores, (3) Proporciona una metodología para realizar las pruebas, (4) Coordina las pruebas de calidad hechas al sistema, las pruebas de aceptación del cliente y las pruebas piloto y (5) Evalúa los resultados que se obtienen en las pruebas de calidad.

De su trabajo se generan los siguientes artefactos:

- ✓ Plan de pruebas.
- ✓ Listado de datos de pruebas.
- ✓ Casos de prueba.
- ✓ Resultados de prueba.
- ✓ Plan de aseguramiento de la calidad
- ✓ Plan de iteración.
- ✓ Evaluación de iteración.
- ✓ Proceso específico de proyecto.
- ✓ Herramientas.
- ✓ Plantillas específicas de proyecto.

Subordinado al Responsable de Calidad están otros tres roles: (1) Diseñador de pruebas, (2) Probador y (3) Revisor Técnico.

El **Diseñador de pruebas**: (1) Diseña de los casos de prueba, (2) Evalúa y documenta el resultado de las pruebas realizadas al software y (3) Define las listas de chequeo.

De su trabajo en el equipo de calidad se obtienen los siguientes artefactos:

- ✓ Plan de pruebas.
- ✓ Listado de datos de pruebas.
- ✓ Casos de prueba.

El **Revisor Técnico**: Verifica que los artefactos generados concuerdan con las normas instituidas para su confección.

El **Probador**: (1) Ejecuta las pruebas diseñadas y (2) Anota los resultados obtenidos con el fin de documentarlos para los procesos venideros.

El artefacto generado de su trabajo son los informes de no conformidades.

De esta manera ha quedado definida la estructura del Equipo de Calidad con cada uno de sus integrantes, sus responsabilidades y tareas dentro de la organización.

3.2.2. Tareas y Responsabilidades.

Estas tareas y responsabilidades estarán divididas en tres grupos: (1) Tareas de prevención, (2) Tareas de evaluación y (3) Tareas de control de fallos internos.

Las Tareas de Prevención estarán destinadas a establecer la infraestructura para la calidad del software. Las de Evaluación consisten en detectar errores en los proyectos en épocas tempranas del proceso de desarrollo, antes de que esos errores sean críticos y con el objetivo de que las desviaciones sean menores. Por su parte las de Control de Fallos Internos, se encargan también de identificar errores encontrados antes de entregar el software, basándose específicamente en los procesos de prueba, además de los detectados por los mismos después de la instalación del sistema.

A. Tareas de Prevención.

Dentro de estas tareas se encuentran: (1) Capacitación, (2) Certificación del Personal, (3) Definición de estándares y guías y (4) Registro de Mejoras del Proyecto.

Tareas de Capacitación: se propone que se realicen actividades de capacitación al equipo de desarrollo del proyecto, con el objetivo de garantizar que en el transcurso de su trabajo se comentan la menor cantidad de errores posibles. Estas actividades se realizarán previas al comienzo del desarrollo del software, sin desechar la posibilidad de que sean enriquecidas en

el transcurso del mismo. El jefe de capacitación será el responsable del cumplimiento de esta tarea.

Estas tareas estarán conformadas por cursos sobre metodologías, herramientas, estándares y buenas prácticas de desarrollo. Se pondrá a disposición de los miembros del proyecto toda la documentación posible de los temas sobre los cuales se estén preparando.

Tareas de Certificación del Personal: para evaluar el nivel de conocimiento mínimo que posee cada miembro del proyecto en el rol que juega se propone que se les apliquen evaluaciones, garantizando con esto, que cada cual cumpla correctamente su trabajo. Esta tarea se efectuará después de estructurados los roles en el proyecto y antes de asignar responsabilidades específicas.

Se propone que esta tarea sea llevada a cabo mediante una prueba de nivel al personal del proyecto en concordancia con el trabajo que realizará.

Tareas de Definición de Estándares y Guías: se propone que el equipo de calidad forme parte del proceso de definición de los estándares y guías que serán usados durante todo el ciclo de desarrollo del proyecto, con el objetivo de garantizar que dicho proceso se realice con la calidad requerida. Esta tarea se realizará cuando se estén sentando las bases del proyecto. El jefe de proyecto es el responsable de garantizar la presencia del equipo de calidad en este proceso.

Registro de Mejoras al Proyecto: se propone la creación de un registro donde cada integrante del proyecto irá documentando las experiencias adquiridas a medida que desarrolle su trabajo con el objetivo de que el mismo esté disponible para futuras consultas de todo el equipo del proyecto.

B. Tareas de Evaluación.

Dentro de estas tareas se encuentran: (1) Revisiones, (2) Monitoreo al proyecto y (3) Auditoría a la Gestión de la Configuración.

Revisiones: se proponen que sean realizadas las que a continuación se describen:

Revisión de la captura de requisitos: mediante la aplicación de listas de chequeo se comprobarán las especificaciones de los requisitos de software que han sido elaborados para el proyecto. Con esta tarea se revisa que el proceso y los productos de la captura de requisitos. El responsable será el Revisor Técnico, el que tiene como responsabilidad verificar que se cumpla la lista de chequeo correspondiente. Las no conformidades son documentadas en el informe de No Conformidades, dándole seguimiento hasta su resolución.

Los artefactos involucrados en este proceso son:

- ✓ Informe de Especificación de requisitos.
- ✓ Informe de Plan de gestión de requisitos.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión de los Diseños Gráficos: antes de aplicar estas revisiones deben haber quedado definidas las pautas del diseño gráfico en el proyecto, las cuales serán probadas mediante la lista de chequeo correspondiente. El responsable de esta tarea será el Revisor Técnico, su meta es revisar los modelos de diseño gráfico para que cumplan con las pautas establecidas. Las no conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución.

Los artefactos involucrados en este proceso serán:

- ✓ Documento de pautas establecidas para el diseño gráfico.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión de la base de datos: se verifica el diseño de la base de datos, obtenido por una herramienta de modelado, mediante la lista de chequeo correspondiente. El responsable será

el Revisor Técnico, su meta es revisar los modelos de diseño gráfico para que cumplan con las pautas establecidas. Las no conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución.

Los artefactos involucrados en este proceso son:

- ✓ Script de la base de datos.
- ✓ Modelo de datos.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión de la Implementación: mediante la ejecución de la lista de chequeo correspondiente se prueba el código fuente, con el objetivo de encontrar errores de codificación según los estándares establecidos. El responsable será el Revisor Técnico, quien revisará el código fuente con el objetivo de encontrar errores de codificación de acuerdo a las pautas establecidas. Las no conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución.

Los artefactos involucrados en este proceso son:

- ✓ Código fuente.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión de la documentación: los documentos definidos en el expediente de proyecto deben haber sido confeccionados para asegurar el cumplimiento de las listas de chequeo elaboradas. El responsable de esta revisión será el Equipo de Calidad, garantizando que todos los artefactos sean documentados y verificando el cumplimiento de las listas de chequeo. Las no

conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución.

Los artefactos involucrados con este proceso son:

- ✓ Manual de usuario.
- ✓ Plantilla DCS Glosario de términos.
- ✓ Todas las plantillas del expediente de proyecto.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión de la gestión de configuración: se garantiza mediante la aplicación de la lista de chequeo que se mantenga la calidad durante los cambios en el proyecto y que se documente la gestión de la configuración. El responsable de esta revisión será el Revisor Técnico, garantizando que se mantenga la calidad durante todos los procesos de cambios que se efectúen en el proyecto. Las no conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución.

Los artefactos involucrados con este proceso son:

- ✓ Plan Gestión de Configuración.
- ✓ Informe de Pedido de cambio.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de Solicitud de cambio.
- ✓ Informe de No Conformidades correspondiente.

Revisión a la capacitación del personal: con esta revisión se asegura que se estén impartiendo los cursos de capacitación al personal del proyecto. El responsable de esta

revisión será el Revisor Técnico, garantizando que sean impartidos los cursos de capacitación al personal del proyecto con la calidad requerida. Las no conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución.

Los artefactos involucrados con este proceso son:

- ✓ Registro de notas del personal.
- ✓ Documentos de capacitación.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión de los registros y mejoras: se asegura que se documenten las experiencias de trabajo de cada integrante del proyecto en el repositorio del mismo. El responsable será el Revisor Técnico quien asegura que se realicen los registros de mejoras y experiencias del personal y que se mantenga actualizado. Las no conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución.

Los artefactos involucrados con este proceso son:

- ✓ Informe del registro de mejoras.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisiones por pares: se propone como una técnica mediante la cual, a través la lista de chequeo, se verifica que el código fuente cumpla con las pautas y los estándares que han sido anteriormente definidos. El responsable será el Revisor Técnico, asegurando el cumplimiento de lo planteado anteriormente. Las no conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución. Los artefactos involucrados con este proceso son:

- ✓ Módulos terminados completamente.
- ✓ Informe de No Conformidades correspondiente.
- ✓ Lista de chequeo correspondiente.

El conjunto de todas las revisiones que han sido descritas anteriormente, definen el proceso de las revisiones técnicas formales que se realizarán en el proyecto.

El conjunto de pasos a seguir para las actividades de revisión se describen en el siguiente diagrama.

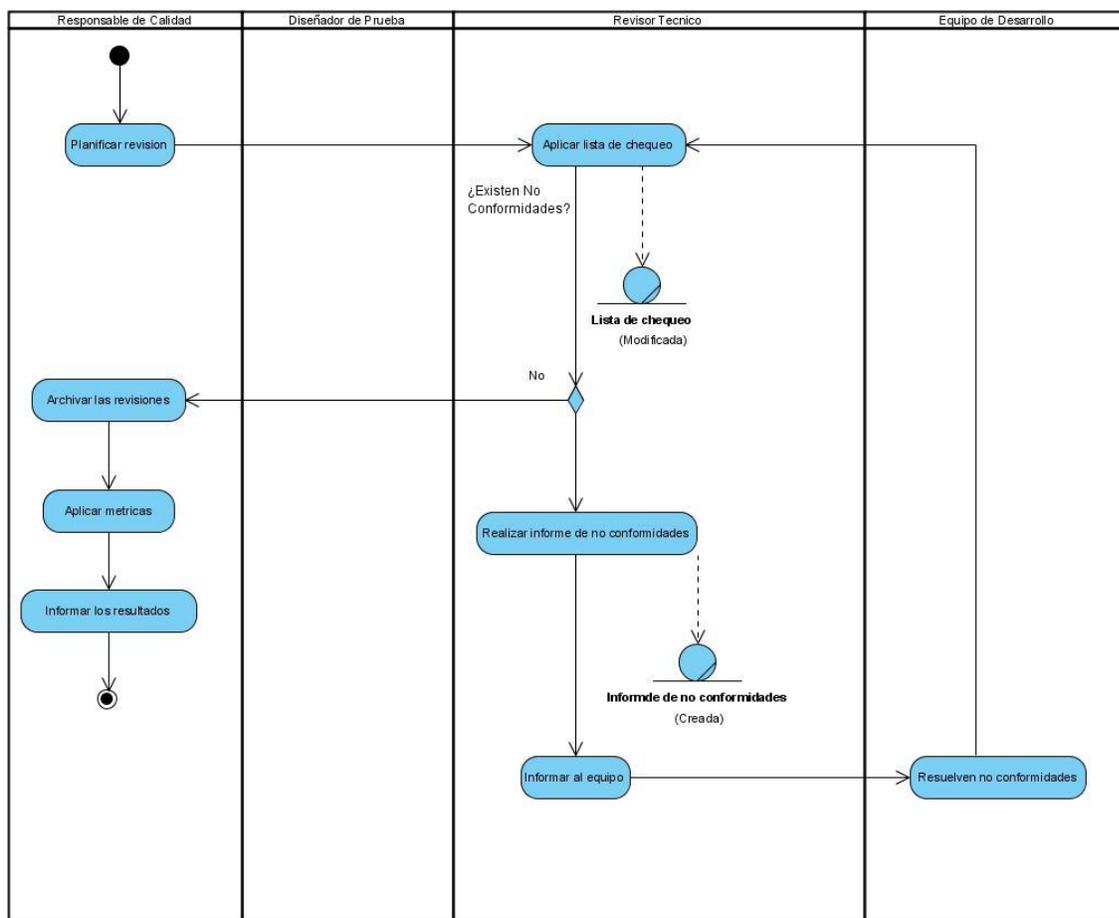


Figura 4: Flujo de trabajo para las actividades de revisión.

Durante la planificación de los procesos de revisión se crea un tiempo para estabilizar, en el cual se encuentra el tiempo para resolver las no conformidades detectadas y para realizar la planificación de otras iteraciones de la revisión.

Monitoreo al proyecto.

El monitoreo al proyecto persigue el mismo objetivo de las auditorías convencionales que realiza el Grupo de Auditoría de la DCS, sin aportar evaluaciones. El Equipo de Calidad es el responsable de realizar esta tarea aplicando la lista de chequeo correspondiente y determinará el momento exacto para ejecutar la misma. El monitoreo revisa que no existan desviaciones en el cronograma del proyecto y que la documentación esté actualizada.

Los artefactos involucrados en este proceso son:

- ✓ Informe de No Conformidades correspondiente.
- ✓ Lista de chequeo correspondiente.

Auditoría de la Gestión de la Configuración.

Se propone que en el proyecto se realice una auditoría a la Gestión de Configuración dada la importancia que representa para el desarrollo. Será llevada a cabo por parte del Equipo de Calidad, quien aplicará la lista de chequeo correspondiente y determinará el momento exacto para ejecutarla. Mediante esta auditoría se asegurará que se mantenga la calidad durante todos los cambios que se realicen y se emitirá una evaluación sobre el estado del proyecto.

Los artefactos involucrados en este proceso son:

- ✓ Informe de Plan Gestión de Configuración.
- ✓ Informe de Pedido de cambio.
- ✓ Informe de No Conformidades.
- ✓ Plantilla de Acciones Correctivas o de mejoras.

- ✓ Lista de Chequeo correspondiente.
- ✓ Plantilla Informe Final de Auditoría.

El conjunto de pasos a seguir para la auditoría de gestión de configuración y el monitoreo al proyecto se describen en el siguiente diagrama.

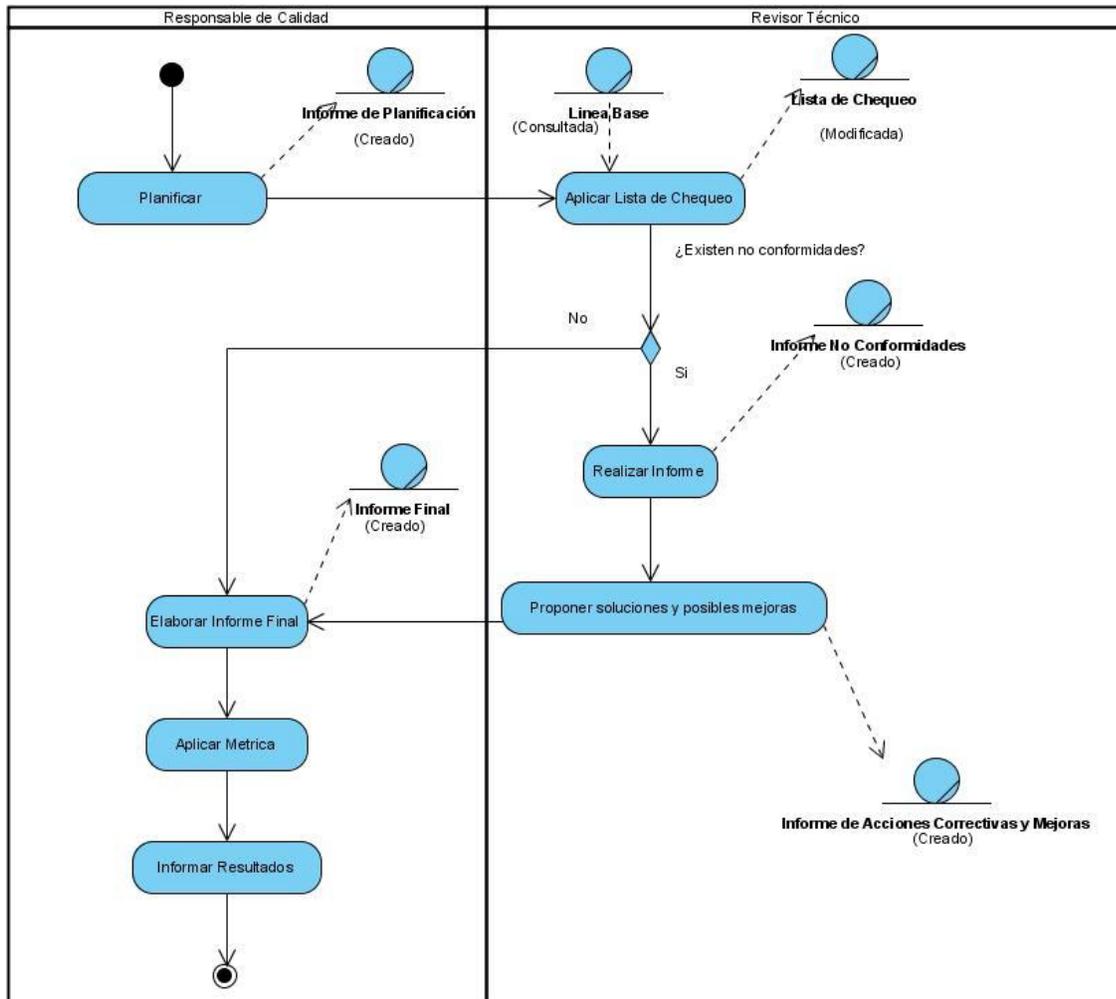


Figura 5: Flujo de trabajo para la auditoría de gestión de configuración y monitoreo del proyecto.

El monitoreo al proyecto y la auditoría a la gestión de configuración son actividades que se incluyen en el cronograma del proyecto, debido que deben ser planificadas y afectan el cumplimiento de otras tareas.

Para conocer detalladamente el procedimiento a seguir de estas actividades que han sido descritas hasta el momento, se propone que se consulte el **Manual de Revisiones y Auditorías** del SIGEP. Además de este informe, está el **Registro de Revisiones**, en el que se encuentran las listas de chequeo correspondientes a estas actividades y los atributos de calidad que se aborda en cada una. En este registro estará documentada cada versión de las distintas revisiones y auditorías realizadas y los artefactos generados de cada una de ellas con los responsables de las mismas.

C. Tareas de Control de Fallos Internos.

Para el control de fallos internos se propone que se realicen: (1) Pruebas Unitarias, (2) Pruebas de Funcionalidad, (3) Pruebas de Integración, (4) Pruebas de Aceptación del Cliente, (5) Pruebas de Seguridad y (6) Pruebas de Rendimiento.

Para conocer las condiciones necesarias que deben cumplirse para realizar los procesos de pruebas en el desarrollo de un proyecto de software, el escenario idóneo para realizarlas, los modelos de despliegue del sistema en el escenario real y los requisitos de hardware y de software necesarios, se propone consultar el **Plan de Pruebas** anexo a este trabajo.

El grupo de Calidad de la Universidad lleva a cabo las actividades de Prueba de Caja Negra, basadas en la búsqueda de los escenarios de los casos de uso para el control de la calidad de los productos de software.

Pruebas Unitarias: Se propone que sean realizadas por el programador del módulo en cuestión, en coordinación con el diseñador del equipo de trabajo, quien elabora los casos de prueba con juegos de datos usando el framework JUnit. El Revisor Técnico será el responsable de velar por el cumplimiento y calidad de esta tarea. Esta nueva estrategia ayuda a mitigar errores encontrados, dado que los programadores conocen sobre las técnicas que emplean y los lenguajes en que trabajan, mientras que el diseñador comprueba que los resultados

arrojados sean los correctos, dado que fue él quien diseñó lo que está siendo probado. Las no conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución. Se usará la metodología para pruebas unitarias de software en la plataforma J2EE (Duharte Junio de 2008).

El conjunto de pasos a seguir para las pruebas unitarias se describen en el siguiente diagrama.

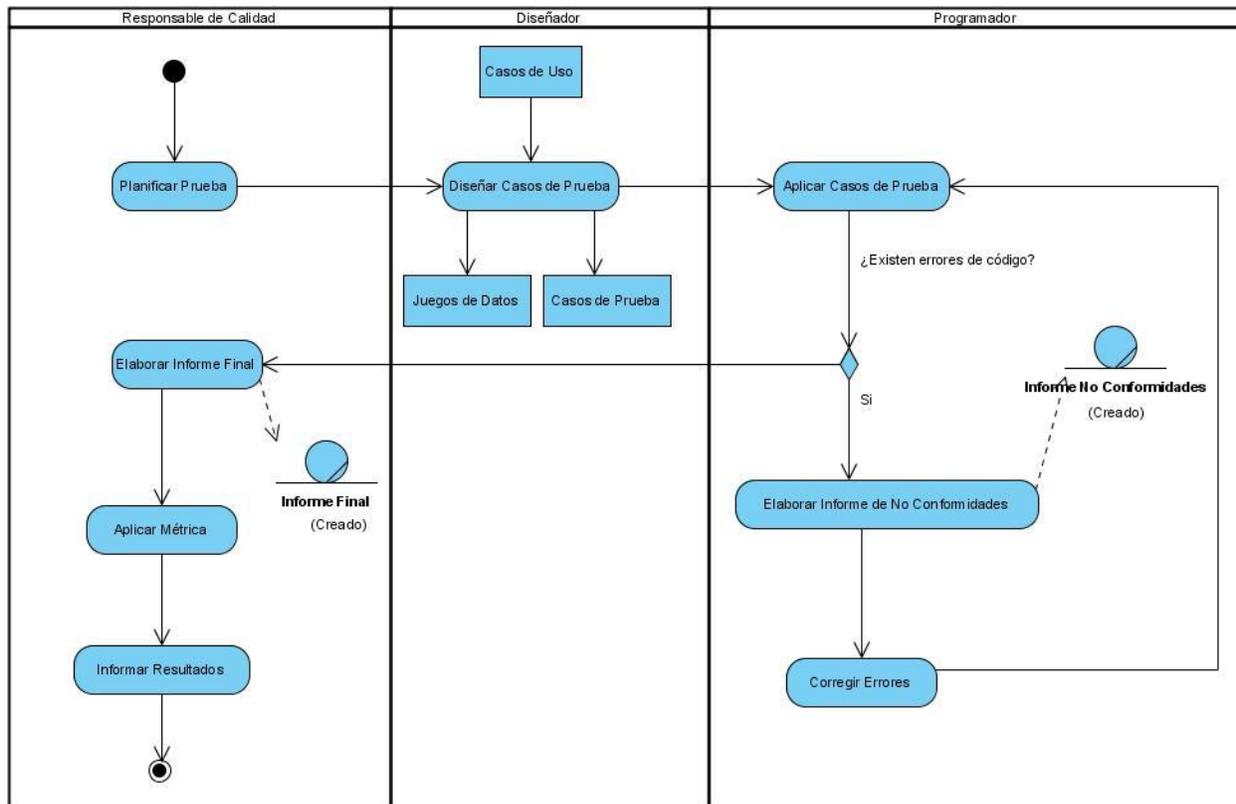


Figura 6: Flujo de trabajo para las Pruebas Unitarias.

Pruebas de Integración: se propone que sean ejecutadas por el Arquitecto, quien integrará todos los módulos del sistema desarrollados de forma independiente en una unidad para probar su integridad y funcionalidad. En el proyecto se utilizará el método **bottom up** para estas pruebas.

Para la realización de estas pruebas, el arquitecto se guía por el **Plan de Integración** para integrar los componentes, y al mismo tiempo va probando si se realiza correctamente con la técnica utilizada. Este plan contiene las dependencias entre los módulos, de ahí se genera el orden en que se pueden integrar. El resultado obtenido lo analiza el arquitecto, buscando si la causa del problema viene de la integración incorrecta o de la programación del módulo. En el primero de los casos, si el problema es de su trabajo lo resuelve, si es de programación, lo resuelve el desarrollador. Este proceso lleva un análisis de los tipos de errores que se generan y sus posibles soluciones. Las no conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución.

El conjunto de pasos a seguir para las pruebas de integración se describen en el siguiente diagrama.

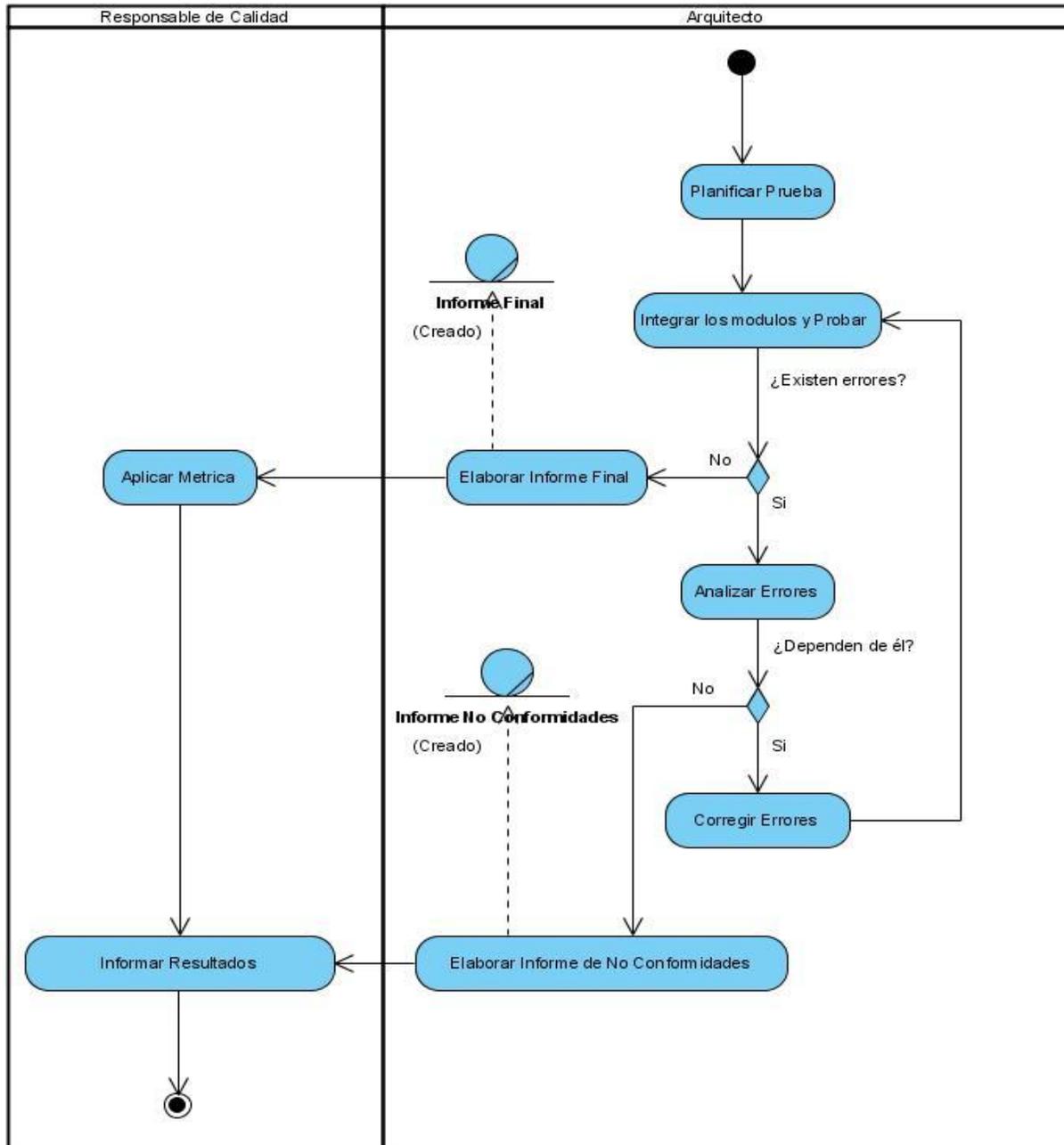


Figura 7: Flujo de trabajo de las Pruebas de Integración.

Pruebas de Funcionalidad: De estas pruebas se encargará el equipo de calidad utilizando listas de chequeo para comprobar la veracidad de la interfaz del sistema y documentando las no conformidades en el informe correspondiente. Serán aplicadas para ellas la técnica de caja

negra, ejecutando los casos de prueba a los cuales se les aplicarán datos válidos e inválidos. Se propone que estos casos de prueba sean realizados por los analistas en conjunto con el Revisor Técnico, asegurando la calidad de los mismos. Para estas pruebas se va a utilizar la misma metodología que se usaba en las fases anteriores del SIGEP.

El conjunto de pasos a seguir para las pruebas de funcionalidad se describen en el siguiente diagrama.

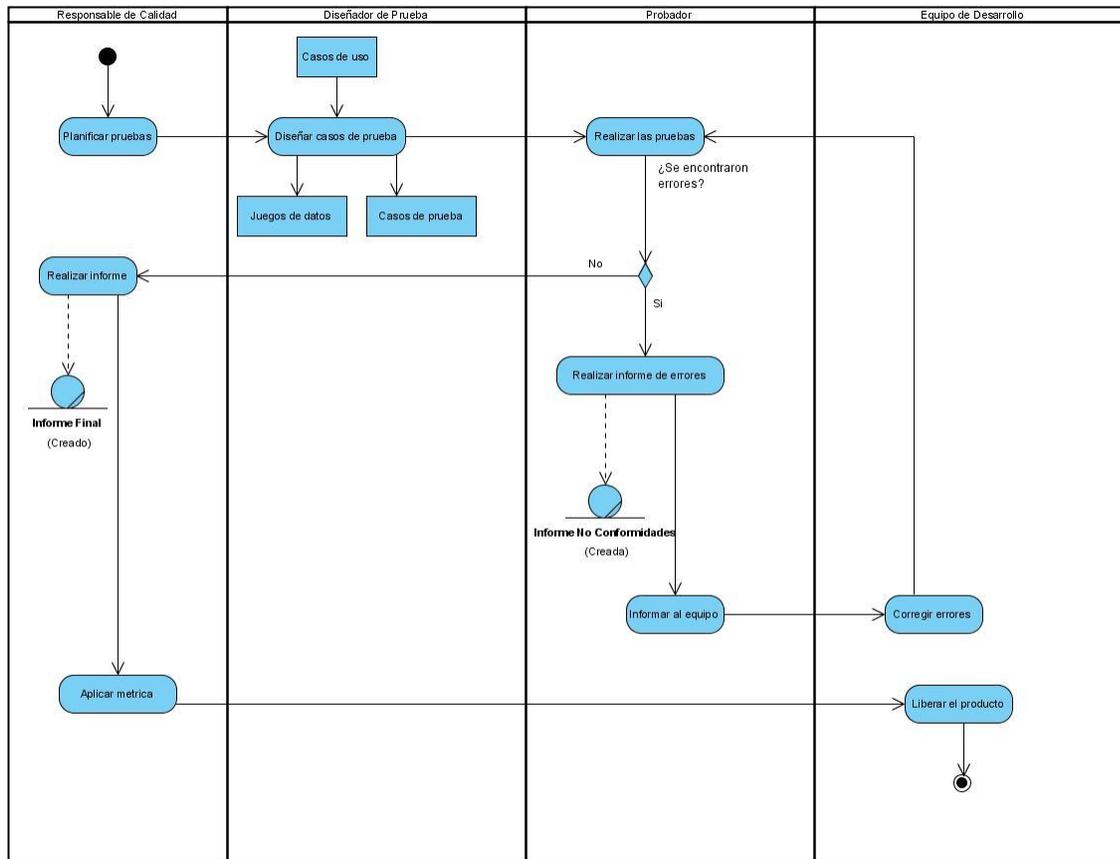


Figura 8: Flujo de trabajo de las Pruebas de Funcionalidad.

Pruebas de Seguridad: estas pruebas están siendo determinadas a nivel de Universidad, se conoce que está en proceso de desarrollo y de prueba el flujo de su ejecución. No se ha establecido una política concreta para llevar a cabo la misma en los proyectos productivos. Su estructura está aún por determinar en La UCI y por tanto, en el proyecto SIGEP. Esto no

significa que estas pruebas no van a ser realizadas, esto constituye un punto importante, por lo que no serán pasadas por alto.

Pruebas de Rendimiento: Estas pruebas serán realizadas cuando el software esté concluido. El mismo será puesto en funcionamiento bajo condiciones que simulan la realidad y que lo sobrecarguen, con el fin de comprobar cuanto es capaz de soportar el sistema funcionando correctamente. El encargado de llevar a cabo esta prueba será el Revisor Técnico. Las no conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución. Para realizar estas pruebas se usará la metodología que está siendo definida en estos momentos por la facultad y que está siendo usada por el proyecto SUA.

Pruebas de aceptación del cliente: Estas pruebas son realizadas por el cliente en conjunto con el Equipo de Calidad de la Universidad. Mediante ellas el cliente comprueba si el producto cumple con los requisitos establecidos y determina si el mismo es aceptado. Son llevadas a cabo cuando finaliza el desarrollo software o una parte de este. Las no conformidades serán documentadas en el informe correspondiente, dándole seguimiento hasta su posterior resolución.

El conjunto de pasos a seguir para las pruebas de aceptación del cliente se describen en el siguiente diagrama.

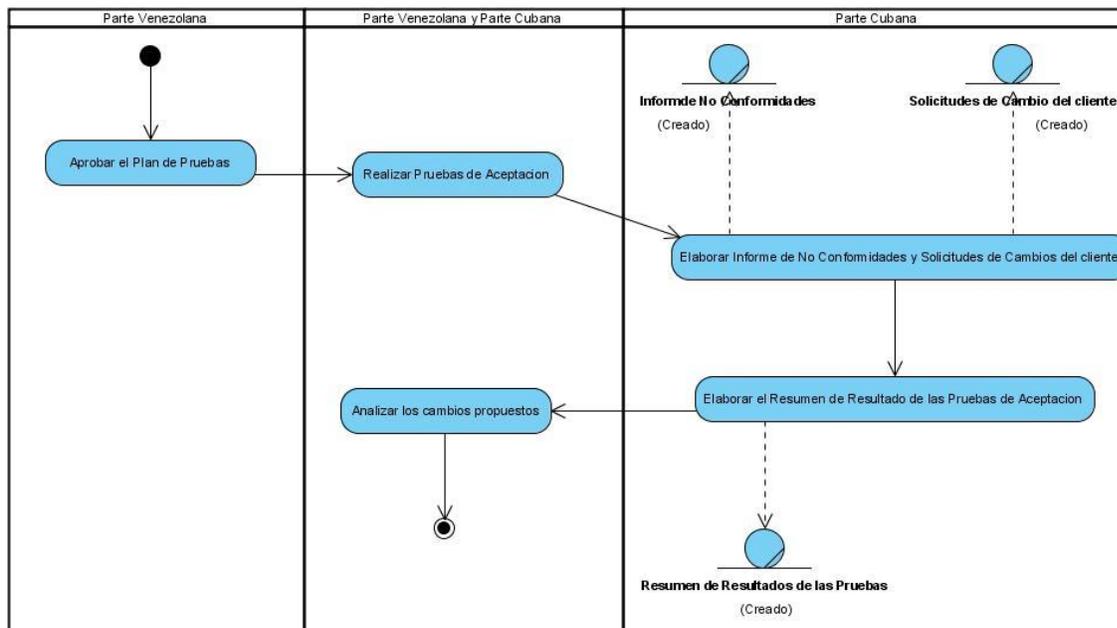


Figura 9: Flujo de trabajo de las Pruebas de Aceptación del Cliente

Las tareas que han sido abordadas hasta el momento se encuentran planificadas con fechas en el cronograma del **PAC SIGEP – Fase IV**. El mismo es un aproximado de posibles fechas en las cuales se realizarán estas actividades, dado que las fechas exactas serán planificadas después que comience el desarrollo de la nueva fase. Se propone que para cada tarea se realicen dos revisiones como una medida preventiva. Para más detalle se propone consultar el anexo que corresponde al **PAC SIGEP – Fase IV**.

3.2.3. Métricas.

Anteriormente en SIGEP se empleaba la métrica para el estado de avance del flujo de implementación, la cual continuará en uso para esta nueva fase. Esta métrica debe ser calculada con periodicidad durante todo el flujo de implementación, en el caso del SIGEP se utiliza para emitir los partes semanales a los directivos interesados en el desarrollo del software. La métrica propuesta tiene entre sus objetivos establecer el por ciento de avance de la implementación de un módulo.

Se proponen además nuevas métricas que serán usadas en la nueva fase del proyecto.

Métricas de Calidad:

Categoría	Conceptos a Medir	Métrica
Defectos	No conformidades	
Defectos	Eficiencia en la eliminación de no conformidades.	$EENC = N_{Ci} / (N_{Ci} + N_{Ce})$
Tiempo	Tiempo	
Tiempo	% del tiempo total dedicado a las fases del ciclo de vida.	$Tiempo(i) / Tiempo\ total$

Tiempo	Varianza en la planificación.	$VP = CTD \ CTP$
Tiempo	Índice de desarrollo del coste	$IDC = CTP / CTR$
Tiempo	Varianza del coste	$VC = CTP \ CTR$
Tiempo	Costo de la calidad	$Tiempo(sq) / Tiempo$

La meta que persigue la métrica de **Defectos** es evaluar la eficiencia del grupo de aseguramiento de la calidad del proyecto e indicar la calidad del producto. La métrica de **Tiempo** ayuda a justificar el uso de nuevas herramientas utilizadas en el proyecto y la formación adicional que hoy se está recibiendo. Evalúa la productividad y eficiencia de las personas que desarrollan el producto y tracea el funcionamiento real contra planes y objetivos establecidos. Esta métrica se obtendrá a través de la herramienta Process Dashboard con soporte para equipo.

Se propone que se consulte el **Plan de Mediciones** del SIGEP para conocer con detalles como se calculan estas métricas, en qué consisten y los resultados que se esperan de las mismas.

3.2.4. Estándares y Guías.

En el proyecto se continuarán utilizando los estándares y guías que estaban en uso en etapas anteriores. Los distintos estándares se aplican durante las diferentes etapas del proceso de desarrollo. Para conocer más detalladamente estos estándares se propone consultar el PAC SIGEP - Fase IV.

3.2.5. Resolución de Problemas y Acciones de Corrección.

A través del desempeño de las diferentes actividades y tareas que se llevan a cabo durante el proceso de desarrollo, se documentan las no conformidades que han sido encontradas.

Referente a estos errores se proponen que sean documentados por escrito con detalles en informes de No Conformidades. Dichos documentos se harán llegar al responsable de atenuar o eliminar los problemas, una copia se le pasará al líder del proyecto y otra al historial manejado dentro del grupo de Calidad. Se propone que la **Lista de Chequeo** sea elaborada de forma tal que se refiera a aspectos de interés según el artefacto a evaluar.

Se propone que las No Conformidades sean clasificadas de acuerdo a los siguientes niveles de impacto:

- ✓ **Bajo:** No impacta los objetivos del proyecto, sólo se registra y se pide que se cumpla para la siguiente ocasión.
- ✓ **Medio:** Puede impactar los objetivos del proyecto y se pide que se realice una acción correctiva.
- ✓ **Crítico:** Impacta el cumplimiento de los objetivos, se pide que se detenga el proyecto y que se realice una acción correctiva.

En caso del nivel de impacto con el que sea clasificado, el equipo puede elevar la no conformidad a la alta gerencia del proyecto para que apoyen en su solución y darle un continuo seguimiento y monitoreo hasta haberle dado solución a las mismas.

En el proyecto estará definido el **Plan de Resolución de Problemas**. En el mismo estarán definidos los pasos a seguir y las pautas a aplicar para darle solución a las dificultades que se presenten derivadas de las actividades de aseguramiento de la calidad.

3.2.6. Herramientas, Técnicas y Metodologías.

Para el apoyo a las actividades de aseguramiento de la calidad se proponen un conjunto de herramientas, técnicas y metodologías.

Listas de chequeo: las mismas serán elaboradas en concordancia con la plantilla establecida en el EP 2.0. No existe una lista de chequeo única, serán elaboradas en dependencia de para

qué actividad serán usadas y los artefactos que serán evaluados con las pautas que deben cumplir.

Casos de prueba: serán elaborados a partir de los diferentes casos de uso. Los analistas son los que llevarán a cabo esta tarea, dado que son ellos los que conocen con profundidad el módulo en cuestión y los parámetros con los que debe de cumplir. Esta tarea será realizada en conjunto con el revisor técnico para asegurar la calidad de los mismos.

Plantillas del EP 2.0: estas plantillas serán usadas para registrar toda la documentación del proyecto y mantenerla de esta forma organizada y cumplir además con la DCS de la universidad.

Lineamientos de calidad 2.0: el PAC SIGEP- Fase IV cumplirá con los lineamientos de calidad que establece la universidad para sus proyectos productivos.

Normas y Estándares: serán aplicados diferentes normas y estándares para la elaboración del PAC. IEEE 730, ISO 9000, CMMI.

Redmine: es una herramienta de gestión de proyectos de software con interfaz web. Será usada en el proyecto para la asignación de tareas a través de los diferentes roles que han sido establecidos.

Subversion: es el sistema de control de versiones que se continuará utilizando en el proyecto para el manejo de las distintas versiones de los módulos implementados y los cambios.

JUNIT: se utilizaba anteriormente en el proyecto y se continuará usando para la ejecución de las pruebas unitarias, dado la gran variedad de patrones en los que se abstrae para facilitar su comprensión y, con ello, su uso.

3.2.7. Gestión de Configuración.

La gestión de configuración es un aspecto de vital importancia dentro del proyecto por lo que el Equipo de Calidad estará todo el tiempo pendiente de este proceso. Con un proceso de calidad de la gestión de configuración, se logra mantener la integridad de los productos que se

obtienen a lo largo del desarrollo de los sistemas de información. Se garantiza de esta manera que no se realizan cambios incontrolados y que todos los participantes en el desarrollo del sistema, dispongan de la versión adecuada de los productos que se manejan. Para conocer con más detalle los pasos que se siguen para lograr lo anteriormente expuesto se propone consultar el **Plan de Gestión de Configuración** del EP.

3.2.8. Registros de Calidad.

Los registros de calidad del proyecto serán guardados en un repositorio que se mantendrá durante todo el ciclo de vida del proyecto. Estos artefactos serán guardados de acuerdo a los roles que los han generado y son los elementos de configuración del área de calidad.

Elementos que se guardarán en el Registro de Calidad

- ✓ Plan de Aseguramiento de la Calidad.
- ✓ Informe de No Conformidades.
- ✓ Registro de Revisiones y Auditorías.
- ✓ Registro de Pruebas.
- ✓ Registro de mejoras.
- ✓ Registro de capacitaciones y certificaciones.
- ✓ Casos de Pruebas.
- ✓ Plan de Pruebas.
- ✓ Listas de Chequeos.
- ✓ Informe de Gestión de Configuración.
- ✓ Informe de RTF.
- ✓ Actas de Reunión efectuadas.

- ✓ Resúmenes de Resultados.
- ✓ Correos de trabajo y comunicación de intercambio de trabajo.

3.2.9. Entrenamiento.

Es importante definir un conjunto de actividades para capacitar al personal que va a hacer uso del PAC SIGEP-Fase IV, con el objetivo de que cumplan sus funciones apropiadamente para así ganar en tiempo de desarrollo. Para ello se propone una serie de cursos.

- ✓ **Curso de capacitación sobre los políticas y procesos de aseguramiento de la calidad.**

Con este curso se pretende que el equipo de trabajo obtenga mayores conocimientos del tema en general y así lograr una nivelación entre todos los integrantes de este equipo.

- ✓ **Investigación sobre las normas nacionales e internacionales de calidad.**

Se propone con este trabajo elevar los conocimientos del equipo al respecto y garantizar que todos los integrantes adquieran familiarización con las normas seleccionadas, con el fin de trabajar sobre elementos sobre los cuales se ha probado su eficiencia y efectividad.

- ✓ **Investigación detallada de los roles de RUP.**

El objetivo de esta tarea es identificar, según cada rol, las tareas que se deben realizar, así se podría distribuir mejor el trabajo y no sobrecargar a una sola persona con exceso de responsabilidades. Incluso hay actividades de los roles de calidad que deberán ser combinados con los esfuerzos de otros roles pertenecientes a otras ramas de desarrollo del Sistema.

✓ **Curso de capacitación para la confección de las listas de chequeos.**

Se procura que se vayan creando las herramientas y conocimientos necesarios para la realización de este tipo de trabajo.

✓ **Curso de capacitación para mejorar el trabajo con las herramientas propuestas anteriormente en este documento.**

Con el objetivo de emplearlas para contribuir al aseguramiento de la calidad del producto software desarrollado.

✓ **Curso de capacitación para el diseño de casos de prueba.**

La meta de esta actividad es proporcionar los conocimientos indispensables para realizar de forma satisfactoria este tipo de trabajo.

✓ **Curso de auditoría.**

De esta forma se confirma la estrategia trazada para el desarrollo de las auditorías en el plan, acorde con la establecida por la universidad.

De esta manera ha quedado definida la estrategia para asegurar la calidad del SIGEP en su fase IV a través de la elaboración del **PAC SIGEP – Fase IV**. Para profundizar en los detalles de los elementos que en este capítulo han sido expuestos se propone consultar el **PAC** que es una anexo de este trabajo.

3.3. CONCLUSIONES

Con este capítulo ha quedado definida la estrategia de aseguramiento de la calidad para SIGEP en su Fase IV. Han quedado plasmados con detalles cada uno de los puntos y la importancia de ellos para el proyecto.

La aplicación de un PAC, debidamente estructurado y basado en las peculiaridades de un proyecto de software, ayuda a determinar una metodología a seguir donde se controle la calidad de cada uno de los procesos del mismo. Es un conjunto de métodos que hay que saber aplicar para que no se vea afectado el proceso que se sigue y el objetivo que se propone alcanzar con el proyecto. Con esto se demuestra la importancia de aplicar el plan propuesto en este trabajo de diploma para la nueva fase de desarrollo del SIGEP.

CONCLUSIONES GENERALES

Con el desarrollo del presente trabajo de diploma se han considerado todos los elementos que demuestran la necesidad de planificar el aseguramiento de la calidad en el proceso de desarrollo de software. Un Plan de Aseguramiento de la Calidad es un elemento básico para lograr que los objetivos del proyecto sean alcanzados. Dicho documento, debe tener una estructura determinada que comprenda los elementos relacionados con la calidad y que se deben contemplar para alcanzar la máxima expresión de la misma.

Mediante este trabajo se ha implementado una estrategia que servirá de apoyo y de guía para la futura fase del proyecto SIGEP. Será una manera eficiente de desarrollar y recoger un trabajo orientado a la calidad. Ha permitido además aplicar los conocimientos adquiridos de experiencias anteriores. Se trata de un plan de calidad estructurado bajo las normas requeridas por la Universidad de las Ciencias Informáticas y fundamentado en prácticas y estrategias de calidad reconocidas mundialmente. Esta estrategia será implantada en la nueva versión del SIGEP y en las que surgirán a partir de ella.

RECOMENDACIONES

Después de la elaboración y ejecución del PAC SIGEP – Fase IV, se recomienda que este plan sirva de guía y de apoyo para otros proyectos productivos de la Universidad que necesiten implantar un Plan de Aseguramiento de la Calidad en su sistema.

BIBLIOGRAFÍA

- Alonso, E. B.
CMMI for Development, V., CMMI-DEV, V1.2 (August 2006). "Improving processes for better products."
CMMI, V. (August 2006). "Improving processes for better products."
Crosby, P.
Deming, W. E.
Dijkstra, E. W.
Duharte, F. R. (Junio de 2008). "Pruebas Unitarias de Software en la Plataforma J2EE."
Feigenbaum, A. V.
IEEE, -. (1998). "IEEE Standard for Software Quality Assurance Plans."
ISO (1994). "Gestión de calidad y garantía de calidad. Vocabulario."
ISO, B. "Importancia de asimilar el concepto de calidad y beneficios de implementar un sistema de gestión de la calidad en la Empresa."
ISO/IEC (1991). "Tecnología de la información. Evaluación del producto software. Características y directrices de calidad para su uso."
Juran, J. M.
Lema, M. V.
LÓPEZ, C. (2002).
LÓPEZ, C. (2002). "Calidad."
MALEVSKI, Y. (1995). "Manual de Gestión de la Calidad Total a la Medida ".
Piattini, G. (2003). "Calidad en el desarrollo y mantenimiento del software."
Pressman, R. S. (1993). "Ingeniería del Software. Un enfoque práctico. Tercera Edición."
Pressman, R. S. (1998). "Ingeniería de software. Un enfoque práctico. Cuarta Edición."
Quesada, J. A. L. "Fundamentos de Ingeniería del Software."
RAE (2006). "Calidad, 2006."
Rivas, N. S. C. y. G. A. P. (Junio 2007). "Pruebas de Aceptación Parciales del Cliente."
SEDNA, P. (2006). "La Revolución Industrial."
Shewhart, W. A.
Taguchi, G.
Teams, R. U. P. B. P. f. S. D. "Rational Unified Process Best Practices for Software Development Teams."
VALERIS, I. H. (Mayo 2007). "Propuesta de Control para la Gestión de la Calidad del Software en el Proceso Productivo UCI."

Anexo 1: Plan Aseguramiento de la Calidad SIGEP – 2009

1. Introducción.

1.1. Propósito.

Con este documento se pretende crear una guía para todo el equipo de desarrollo del proyecto para asegurar eficientemente todas sus actividades a lo largo del ciclo de vida del mismo. Además es el encargado de describir la visión de cómo se asegurará la calidad del producto de forma general, los artefactos, herramientas y procesos a través de los cuales se planifica lograr dicho propósito. Se pretende dejar constancia de todo el estilo de trabajo, herramientas, técnicas y metodologías a asumir, flujos de trabajo, artefactos, además de los niveles de responsabilidades dentro de cada rol en el proyecto. Conjuntamente con esto se propone el uso de un conjunto de documentos que se incluyen como anexos de este trabajo.

1.2. Alcance.

Este Plan es elaborado a partir de la plantilla establecida por el expediente de Proyecto elaborado por la DCS de la universidad sobre cómo se elabora un Plan de Aseguramiento de la calidad, teniendo como punto de referencia el propuesto por el Proceso Unificado Racional (RUP) y adaptado a las condiciones específicas del proyecto de forma general. También se pretende llevar a la par conjuntamente las especificaciones requeridas por el estándar IEEE 730 – 1998. Por tanto, su uso es muy particular del proyecto SIGEP Fase IV.

1.3. Definiciones, Acrónimos y Abreviaturas.

- ✓ DSC: Dirección de Calidad de Software.
- ✓ RUP: Proceso Unificado Racional
- ✓ PPQA: Aseguramiento de la Calidad de Procesos y Productos.
- ✓ SIGEP: Sistema de Gestión Penitenciara.

- ✓ IEEE: The Institute of Electrical and Electronics Engineers o Instituto de Ingenieros Eléctricos y Electrónicos. (Asociación técnico-profesional mundial dedicada a la estandarización).
- ✓ ISO: Organización Internacional para la Estandarización.

1.4. Referencias.

Código	Título
1	Lista de Chequeo
2	Informe de Requerimientos
3	Plan de Prueba
4	Informe de No Conformidades
5	Plan de Iteración
6	Plan de Resolución de Problemas
7	Manual de Revisiones y Auditorías
8	Registro de Revisiones
9	Plan de Integración
10	Plan de Mediciones
11	Plan de Gestión de Configuración

1.5. Resumen.

En las primeras secciones del presente documento se hace alusión al Propósito, el Alcance y los Objetivos de Calidad del Proyecto SIGEP en su Fase IV. Se hace una descripción de los miembros del equipo de calidad y de sus tareas y responsabilidades correspondientes. Se plantean las propuestas de estándares y guías a usar en las diferentes etapas del proyecto. Se abordan las estrategias, técnicas y herramientas a utilizar en las Revisiones, Auditorías y Pruebas para cada proceso de desarrollo del software definido por los miembros del proyecto. Se identifican las herramientas y métodos que soportan las

actividades y tareas de aseguramiento de la calidad de forma general, además de actitudes a tomar en el proceso de garantía de la calidad.

2. Objetivos de Calidad.

Los objetivos de calidad plasmados son los objetivos básicos del proyecto, asegurando su cumplimiento. Específicamente se propone lograr:

- ✓ Asegurar la calidad del trabajo, a través de la vigilancia, prevención, comprobación y valoración sistemática en el proyecto, a lo largo del ciclo de vida del mismo, velando por que el producto software cumpla con los requerimientos establecidos por el cliente.
- ✓ Identificar posibles errores antes de que se conviertan en puntos fatales o problemáticos.
- ✓ Garantizar un elevado nivel de idoneidad e interoperabilidad en el producto a desarrollar; asegurando, a través de las revisiones y pruebas, que se logre de una manera flexible y configurable todas las operaciones que se desarrollan en las diferentes cárceles de Venezuela.
- ✓ Mantener el trabajo sobre la base de los diferentes estándares y normas internacionales existentes, así como la metodología establecida por RUP y los estándares definidos para el proyecto.
- ✓ Velar y asegurar el desarrollo e implantación de un sistema informático que soporte las decisiones estratégicas del Ministerio del Interior y Justicia y de la Dirección General de Custodia y Rehabilitación del Recluso.
- ✓ Asegurar que el producto a desarrollar gane un alto grado de comprensibilidad y cognoscibilidad, según las especificaciones establecidas; aplicando para ello, revisiones y pruebas durante todo el ciclo de desarrollo del software y pretendiendo, de esta forma,

contar con una documentación bien detallada de todos los procesos del negocio penitenciario, además de la documentación propia del sistema.

- ✓ Lograr un software que contribuya a garantizar el respeto a los derechos de los internos, su actividad de rehabilitación y reinserción en la sociedad.

- ✓ Corresponder y hacer cumplir con los lineamientos de calidad establecidos por Calisoft para los proyectos productivos de la UCI.

- ✓ Asegurar y contribuir al alcance de los objetivos trazados para la Solución Tecnológica del SIGEP en el Documento Técnico del Proyecto.

- ✓ Lograr que el equipo de calidad cuente con el personal capacitado con el conocimiento y las habilidades necesarias para las realizar las tareas y actividades encaminadas a lograr la calidad del proyecto.

- ✓ Colaborar con que la Gestión de Configuración sea desarrollada de tal manera que satisfaga las necesidades de la evolución del producto software y los procesos de producción del mismo.

3. Gestión.

3.1. Organización.

A continuación se describen las responsabilidades de cada uno de los roles que responden por los procesos de aseguramientos de la Calidad en el proyecto SIGEP, así como los conocimientos básicos que deben poseer para desempeñar sus funciones.

Rol	Descripción	Conocimientos mínimos
Jefe de proyecto	Es el encargado de aprobar las tecnologías a usar en el desarrollo de software, aprueba la metodología a emplear, es además el responsable de supervisar y apoyar todas las tareas a ejecutar y es el que gestiona los recursos y materiales necesarios para el proyecto.	RUP Tecnologías de desarrollo de software. Gestión de proyectos. Ingeniería de Software. Facilidades para la comunicación.
Responsable de calidad	Es una persona bien orientada al detalle, su objetivo principal se centra en asegurarse que la aplicación se ajusta a los requerimientos y mantenerse lo más lejano posible de errores. Provee una metodología para realizar las pruebas. Evalúa los resultados que se obtienen al realizar las pruebas de calidad.	Metodología RUP Lenguaje UML Calidad del software Ingeniería de software
Diseñador de pruebas	Diseña los casos de pruebas. Valora y documenta el efecto de las pruebas realizadas al producto.	Metodología RUP Lenguaje UML Pruebas de software

	Define las listas de chequeo.	
Revisor técnico	Verifica que los artefactos generados concuerdan con las normas instituidas para su confección.	Metodología RUP Lenguaje UML
Probador	Ejecuta las pruebas previamente diseñadas. Registra los resultados obtenidos.	Conocimientos del negocio. Conocimientos de requisitos de software.

3.2. Tareas y Responsabilidades.

En esta sección se describen detalladamente las revisiones de calidad que se efectuarán en el proyecto. Las precondiciones y las poscondiciones que deben cumplir cada una de ellas, así como el responsable de ejecutar las mismas.

Tarea	de	Precondición	Pos condición	Responsable	Comentarios
Aseguramiento de la Calidad		Al finalizar la fase	Antes de la fase		
Revisión de la Captura de Requisitos	de la	Se ha efectuado la captura de requisitos.	Se documentado la revisión y en caso de encontrar Conformidades, se habrán solucionado.	Revisor Técnico	Verificar del cumplimiento de la lista de chequeo elaborada.

Revisión de los Diseños Gráficos	Se han establecido las pautas del diseño gráfico para el proyecto.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Revisor Técnico	Confirmar los modelos de diseños gráficos por las pautas predefinidas.
Revisión de la Base de Datos	Se ha obtenido un modelo del diseño de la base de datos a través de una herramienta de modelado y establecido un estándar para diseñar la BD.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Revisor Técnico	Verificar la base de datos por las pautas establecidas.
Revisión de la Implementación	Se ha establecido el estándar de codificación para el proyecto.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Revisor Técnico	Hacer una revisión del código fuente con el objetivo de encontrar posibles errores de codificación según el estándar establecido.
Revisión de la Documentación	Todos los documentos establecidos en el expediente de proyecto han sido confeccionados.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán	Equipo de Calidad	Verificar el cumplimiento de las listas de chequeo elaboradas.

solucionado.					
Revisión de la Gestión de Configuración	Se ha documentado la Gestión de Configuración, así como que se esté ejecutando.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Revisor Técnico		Asegurar que se mantiene la calidad durante la realización de los cambios. Los informes de estado proporcionan información sobre cada cambio a aquellos que tienen que estar informados.
Revisión de las actividades de Capacitación del Personal.	Se han realizado y documentado las actividades de certificación pertinentes y se han impartidos cursos al personal del proyecto.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Revisor Técnico		Asegurar que se impartan los cursos al personal con la calidad requerida.
Revisión al Registro de Mejoras y Experiencias.	Se han documentado todos los errores y experiencias de cada miembro del proyecto dependiendo de su rol.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Revisor Técnico		Asegurar que en proyecto exista un registro de mejoras y experiencias para facilitar el trabajo del personal.
Revisiones por	Se han realizado las pruebas	Se ha documentado	ha la		Asegurar que el código fuente cumpla

Pares.	unitarias pares.	por	revisión y en caso de encontrar Conformidades, se habrán solucionado.	Revisor Técnico	con las pautas establecidas.
Auditoría de la gestión de configuración.	Se ha efectuado la revisión planificada para la Gestión de Configuración.	Se	ha documentado la auditoría y en caso de encontrarse No Conformidades estas se habrán solucionado.	Revisor Técnico	Validar el funcionamiento de la gestión de configuración y aplicar listas de chequeo para verificar que se cumplan las pautas establecidas.
Monitoreo al Proyecto	Se ha efectuado el monitoreo a todos los módulos del proyecto y a los artefactos generados durante este.	Se	ha documentado el monitoreo y en caso de encontrarse No Conformidades estas se habrán solucionado.	Equipo de Calidad	Revisar la gestión del proyecto en el momento en que el equipo de calidad crea necesario.
Pruebas de Funcionalidad	Se han documentado los requisitos del software.	Se	han documentado los casos de prueba y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad.	Comprobar la adecuada funcionalidad del software según los requerimientos pautados.

Pruebas de Unidad	Se programado el módulo correspondiente.	ha el	Se documentado en caso de encontrar Conformidades, se habrán solucionado.	han los casos de prueba y No	Programador y diseñador	Comprobar el funcionamiento del código fuente con el objetivo de encontrar errores en los módulos programados. Serán realizadas por el programador y por el diseñador del equipo de trabajo. El revisor técnico velará por la calidad de esta tarea.
Pruebas de Integración	Se realizados pruebas unitarias a los módulos se han subido a las mismas al repositorio del proyecto.	han se	Se documentado en caso de encontrar Conformidades, se habrán solucionado.	han los casos de prueba y No	Arquitecto	El arquitecto integra los branches subidos al repositorio en un solo sistema. Comprobar la correcta integración entre los módulos definidos.
Pruebas de Seguridad	Se determinado requisitos de seguridad del software y que se encuentre funcionando su totalidad.	han los de del en caso de encontrar Conformidades, se habrán solucionado.	Se documentado en caso de encontrar Conformidades, se habrán solucionado.	han los casos de prueba y No	Revisor Técnico	Comprobar que el producto final mantenga altos niveles de fiabilidad.

Pruebas de Aceptación del Cliente	Se ha cumplido con el ciclo de vida del software y que éste se encuentre en condiciones de ser entregado al cliente.	Se han documentado los casos de prueba y en caso de encontrar No Conformidades, se habrán solucionado.	Cliente y Equipo de Calidad	Pruebas a través de las cuales el cliente determina si acepta o no el producto entregado.
Pruebas de Rendimiento	Se ha completado el funcionamiento del software	Se han documentado los casos de prueba y en caso de encontrar No Conformidades, se habrán solucionado.	Revisor Técnico	Pruebas para comprobar el completo y correcto funcionamiento del software.

4. Documentación.

Código	Título
1	Informe del Proyecto Técnico
2	Informe de No Conformidades
3	Plan de Gestión de Configuración
4	Documento Visión del Proyecto

5. Métricas.

Se empleará la métrica para el estado de avance del flujo de implementación. La misma debe ser calculada con periodicidad durante todo el flujo de implementación, en el caso del SIGEP, se utiliza para emitir los partes semanales a los directivos interesados en el desarrollo del software. Esta métrica tiene entre sus objetivos establecer el por ciento de avance de la implementación de un módulo. Para calcular este por ciento se divide la medición por casos de uso y se basa en los factores de complejidad técnica (alta, media, baja). Para conocer el por ciento de avance de la implementación de un módulo, antes es necesario determinar el por ciento de avance de la implementación de cada caso de uso que lo compone.

Se proponen además de esta métrica, otras que serán usadas en la nueva fase del proyecto.

Categoría	Conceptos a Medir	Métrica
Defectos	No conformidades	
Defectos	Eficiencia en la eliminación de no conformidades.	$EENC = N_{Ci} / (N_{Ci} + N_{Ce})$
Tiempo	Tiempo	
Tiempo	% del tiempo total dedicado a las fases del ciclo de vida.	$Tiempo(i) / Tiempo\ total$
Tiempo	Varianza en la planificación.	$VP = CTD / CTP$
Tiempo	Índice de desarrollo del coste	$IDC = CTP / CTR$
Tiempo	Varianza del coste	$VC = CTP / CTR$
Tiempo	Costo de la calidad	$Tiempo(sq) / Tiempo$

La meta que persigue la métrica de **Defectos** es evaluar la eficiencia del grupo de aseguramiento de la calidad del proyecto e indicar la calidad del producto. La métrica de **Tiempo** a su vez ayuda a justificar el uso de nuevas herramientas utilizadas en el proyecto y la formación adicional que hoy se está recibiendo, evalúa la productividad y eficiencia de las

personas que desarrollan el producto y tracea el funcionamiento real contra planes y objetivos establecidos. Esta métrica se obtendrá a través de la herramienta Process Dashboard con soporte para equipo.

Se propone que se consulte el **Plan de Mediciones** del SIGEP para conocer con detalles como se calculan estas métricas, en qué consisten y los resultados que se esperan de las mismas.

6. Estándares y Guías.

En esta sección se listan los estándares y guías que han sido utilizados por el plan de calidad.

Estándar	Etapa a aplicar	Comentarios
IEEE 830	Captura de Requisitos	
ISO 12207		
IEEE 1233		
IEEE 1471	Arquitectura	
IEEE 1016	Diseño gráfico	
IEEE 730	Pruebas y Métricas	
ISO 9000		
CMMI		
ISO 12119	Documentación	
IEEE 829		
IEEE 1063		
Lineamientos de Calidad de la Universidad	Al proyecto de forma general	
RUP	Durante todo el ciclo de vida	

7. Plan de Revisiones y Auditorías

7.1. Tareas Generales de Revisiones y Auditorías.

Serán descritas en esta sección cada una de las actividades de revisión y auditoría que se llevarán a cabo en el proyecto, así como los artefactos que serán objeto de estas revisiones.

Revisiones de la captura de requisitos: esta revisión se corresponde con la tradicional Revisión de las Especificaciones del Software. A los artefactos involucrados se les aplica la lista de chequeo para comprobar las especificaciones de requisitos. Las no conformidades encontradas serán recogidas en un documento a las cuales se les dará solución y seguimiento.

Artefactos involucrados:

- ✓ Informe de Especificación de requisitos.
- ✓ Informe de Plan de gestión de requisitos.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión del diseño gráfico: para realizar este tipo de revisión deben de haberse definido anteriormente las pautas del diseño gráfico para el proyecto. Posteriormente se aplica la lista de chequeo correspondiente al diseño, las no conformidades son documentadas para darle seguimiento y solución.

Artefactos involucrados:

- ✓ Documento de pautas establecidas para el diseño gráfico.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión de la Base de Datos: esta revisión se encarga de aplicar la lista de chequeo correspondiente para verificar el diseño de la base de dato que ha sido posteriormente obtenido por una herramienta de modelado. Se documentan las no conformidades para darle seguimiento.

Artefactos involucrados:

- ✓ Script de la base de datos.
- ✓ Modelo de datos.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión de la Implementación: a través de esta revisión se verifica el código fuente con el objetivo de encontrar errores de codificación según los estándares establecidos aplicando la lista de chequeo correspondiente. Las no conformidades son documentadas para su posterior solución y seguimiento. Esta revisión garantiza que se realicen pruebas de caja blanca a cada caso de uso implementado, pruebas de caja negra a cada módulo concluido, pruebas de integración al sistema y pruebas de seguridad.

Artefactos involucrados:

- ✓ Código fuente.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión de la Documentación: todos los documentos que han sido anteriormente definidos en el expediente de proyecto deben de haber sido confeccionados para poder asegurar con esta revisión el cumplimiento de las listas de chequeo elaboradas.

Artefactos involucrados:

- ✓ Manual de usuario.
- ✓ Plantilla DCS Glosario de términos.
- ✓ Todas las plantillas del expediente de proyecto.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión de la Gestión de Configuración: con esta revisión se asegura mantener la calidad durante los cambios en el proyecto y se asegura además que se haya documentado la gestión de configuración. Después de aplicada la lista de chequeo se documentan las no conformidades para darle seguimiento y solución.

Artefactos involucrados:

- ✓ Plan Gestión de Configuración.
- ✓ Informe de Pedido de cambio.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de Solicitud de cambio.
- ✓ Informe de No Conformidades correspondiente.

Revisión de las actividades de certificación al personal: mediante esta revisión se asegura que todos los miembros del proyecto estén capacitados para desarrollar el trabajo que les corresponde de acuerdo al rol que desempeñan.

Artefactos involucrados:

- ✓ Registro de notas del personal.

- ✓ Documentos de capacitación.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisión al registro de mejoras y experiencias: se garantiza que estén recogidas las experiencias del trabajo de cada uno de los integrantes del proyecto en el repositorio del mismo.

Artefactos involucrados:

- ✓ Informe de errores por roles.
- ✓ Informe del registro de experiencias.
- ✓ Informe del registro de mejoras.
- ✓ Lista de Chequeo correspondiente.
- ✓ Informe de No Conformidades correspondiente.

Revisiones por pares: se garantiza con esta técnica de revisión que el código que ha sido probado esté en concordancia con los estándares y pautas establecidas.

Artefactos involucrados:

- ✓ Módulos terminados completamente.
- ✓ Informe de No Conformidades correspondiente.
- ✓ Lista de chequeo correspondiente.

Monitoreo al proyecto: se garantiza con este monitoreo revisar toda la gestión del proyecto, será realizado cuando el equipo de calidad crea que es necesario.

Artefactos involucrados:

- ✓ Módulos terminados.
- ✓ Informe de requerimientos.
- ✓ Informe de estándares y pautas establecidas.
- ✓ Informe de No Conformidades correspondiente.
- ✓ Lista de chequeo correspondiente.

Auditoría de la gestión de configuración: con esta auditoría se garantiza que mediante la aplicación de listas de chequeo se valide el correcto funcionamiento de la gestión de configuración cumpliendo las pautas establecidas anteriormente.

Artefactos involucrados:

- ✓ Informe de Plan Gestión de Configuración.
- ✓ Informe de Pedido de cambio.
- ✓ Plantilla DCS Solicitud de cambio.
- ✓ Informe de No Conformidades.
- ✓ Plantilla de Acciones Correctivas o de mejoras.
- ✓ Lista de Chequeo correspondiente.
- ✓ Plantilla Informe Final de Auditoría.

No. de Revisión	Tipo	Objetivos	Descripción (Iteración)	Fase del Proyecto	Responsable
1	Requerimientos.	Asegurar la especificación de los requisitos.	Al finalizar la captura de los requisitos.	Inicio	Revisor Técnico
2	Diseño gráfico.	Confirmar que se cumplan los modelos del diseño gráfico según las pautas establecidas.	Al finalizar la fase de elaboración	Elaboración	Revisor Técnico
3	Base de datos.	Confirmar que se haya realizado un modelo del diseño de la base de datos mediante una herramienta de modelado y verificar mediante las pautas establecidas.	Al finalizar la fase de Elaboración	Elaboración	Revisor Técnico
4	Implementación	Revisar el código fuente	Al finalizar la fase de	Construcción	Revisor Técnico

		para encontrar errores de codificación según el estándar que ha sido definido anteriormente	Construcción				
5	Documentación	Se revisa el cumplimiento de las listas de chequeo elaboradas.	Durante todo el ciclo de vida del software	Todas las fases	Equipo de calidad		
6	Gestión de configuración	Se garantiza que se mantenga la calidad durante todos los procesos de cambio.	Durante todo el ciclo de vida del software	Todas las fases	Revisor Técnico		
7	Certificación personal.	Impartirle cursos de preparación personal para garantizar que tengan los conocimientos necesarios para desarrollar	Antes de comenzar el desarrollo del proceso de software	Inicio	Revisor Técnico		

			eficientemente su trabajo.				
8	Registro de experiencias y mejoras.	de	Asegurar que se mantenga actualizado el registro de mejoras y experiencias en el repositorio.	Durante todo el ciclo de vida del software	Todas las fases	las	Revisor Técnico
9	Revisión por pares	por	Garantizar que las pruebas unitarias sean realizadas por pares de programadores	Durante la fase de Construcción	la	Construcción	Revisor Técnico
10	Monitoreo al proyecto	al	El correcto y completo funcionamiento de todos los módulos del sistema	Durante todo el ciclo de vida del software	Todas las fases	las	Equipo de calidad
11	Auditoría de la gestión de configuración	de la	Validar el funcionamiento de la gestión de configuración aplicando listas de chequeo.	Durante todo el ciclo de vida del software	Todas las fases	las	Revisor técnico

7.3. Cronograma.

En esta sección se describe un aproximado de las fechas según el avance del proyecto en las que serán realizadas las actividades de revisiones y auditorías. Para conocer con exactitud las fechas correspondientes se propone consultar el Plan de Iteración del proyecto.

		Fecha según el avance del proyecto.	
Revisiones y Auditorías.		Inicio	Duración
		Revisión de los requerimientos.	1ra Revisión: 10 días después de haber iniciado la captura de requisitos 2da Revisión: 7 días antes de concluir el modelado del negocio.
Revisión de los diseños gráficos.	1ra Revisión: 9 días después de haber iniciado el diseño gráfico. 2da Revisión: 5 días antes de concluir el diseño gráfico.	1 semana	
Revisión de la base de datos.	1ra Revisión: 15 días después de haber iniciado el diseño de la base de datos. 2da Revisión: 7 días antes de concluir el diseño de la base de datos.	1 semana	
Revisión de la implementación.	1ra Revisión: 20 días después de haber iniciado la implementación del módulo.		

	2da Revisión: 5 días antes de concluir la implementación del módulo.	2 semanas
Revisión de la documentación.	1ra Revisión: Al concluir la implementación del módulo desarrollado correspondiente. 2da Revisión: 10 días después de haber concluido la primera revisión a la documentación.	1 semana
Revisión de la Gestión de Configuración.	1ra Revisión: 10 días después de haber iniciado la captura de requisitos. 2da Revisión: 5 días antes de concluir la captura de requisitos.	1 semana
Revisión de las actividades de Certificación Personal.	1ra Revisión: 7 días después de iniciado el curso 2da Revisión: 3 días después de terminado el curso. Realizar una prueba para comprobar los conocimientos adquiridos.	1 semana
Revisión al registro de mejoras y experiencias.	Revisiones semanales al repositorio.	2 días
Revisiones por pares.	1ra Revisión: 7 días después de haber comenzado con la prueba. 2da Revisión: 5 días antes de terminar la ejecución de la prueba.	1 semana
Monitoreo al Proyecto.	1ra Revisión: 1 mes antes de finalizar el desarrollo total del software 2da Revisión: al instante de terminado el desarrollo del software.	2 semanas

Auditoría de la gestión de configuración.	<p>1ra Revisión: 7 días después de haber iniciado la captura de requisitos.</p> <p>2da Revisión: 7 días antes de concluir la captura de requisitos.</p>	1 semana
---	---	----------

7.4. Organización y Responsabilidades

Se enumeran en esta sección los roles que intervienen en cada una de las anteriores actividades de revisión y auditoría y las responsabilidades que desempeñan cada uno de ellos.

Revisiones y Auditorías	Roles de Calidad Involucrados	Responsabilidades
Revisión de la captura de requisitos.	Revisor Técnico	Verificar que se cumpla la lista de chequeo elaborada.
Revisión del diseño gráfico.	Revisor Técnico	Revisar los modelos de diseño gráfico para que los mismos cumplan con las pautas establecidas.
Revisión de la implementación.	Revisor Técnico	Revisar el código fuente con el objetivo de encontrar errores de codificación de acuerdo a los estándares definidos en el proyecto.

Revisión de la Base de datos.	Revisor Técnico	Verificar el funcionamiento de la base de datos de acuerdo a las pautas establecidas.
Revisión de la documentación.	Equipo de calidad	Verifica el cumplimiento de las listas de chequeo que han sido elaboradas y que los diferentes artefactos estén siendo documentados.
Revisión de la gestión de configuración	Revisor Técnico	Asegura que se mantenga la calidad durante todos los cambios realizados.
Revisión de la certificación al personal	Revisor Técnico	Garantiza que se impartan los cursos de capacitación en el proyecto.
Revisión al registro de mejoras y experiencias	Revisor Técnico	Asegura que exista y que se mantenga actualizado el registro de experiencias y mejoras en el proyecto.

Revisiones por pares	Revisor Técnico	Asegura que las pruebas unitarias estén siendo realizadas por pares de programadores
Auditoría a la Gestión de Configuración.	Equipo de calidad	Valida el correcto funcionamiento de la gestión de configuración en el proyecto.
Monitoreo al proyecto	Equipo de Calidad	Valida el funcionamiento de los módulos del proyecto según las pautas establecidas.

7.5. Resolución de problemas y actividades de corrección.

Durante el proceso de desarrollo del sistema, ya sea a través de la creación del software, como a través de pruebas, auditorías o revisiones, pueden ser detectados errores, problemas o incongruencias. En ese caso se irá actualizando un documento titulado **Informe de No Conformidades**, donde será relacionada la no conformidad encontrada.

Luego de cada proceso de auditoría o revisión, se deberán registrar los errores o problemas encontrados tanto en la **Lista de Chequeo** como en el **Informe de No Conformidades** correspondiente. Dichos documentos se harán llegar al responsable de atenuar o eliminar los problemas, una copia se le pasará al líder del proyecto y otra al historial manejado dentro del grupo de Calidad. Proponemos que la **Lista de Chequeo** sea elaborada de forma tal que se

refiera a aspectos de interés según el artefacto a evaluar, con el fin de encontrar errores que luego serán documentados de forma más específica en el **Informe de No Conformidades**.

Para conocer con más detalle las actividades específicas que se realizarán para corregir los problemas encontrados se propone consultar el **Plan de Resolución de Problemas** con que cuenta el proyecto.

7.6. Proceso de las revisiones y auditorías

Se propone consultar el **Manual de Revisiones y Auditorías** del proyecto para conocer detalladamente el procedimiento a seguir para cada una de las actividades de revisiones y auditorías. Además el proyecto cuenta con un **Registro de Revisiones** en el que se encuentran las listas de chequeo correspondientes a estas actividades y los atributos de calidad que se aborda en cada una de estas listas. Se propone consultar todos estos documentos para conocer con profundidad dicho proceso.

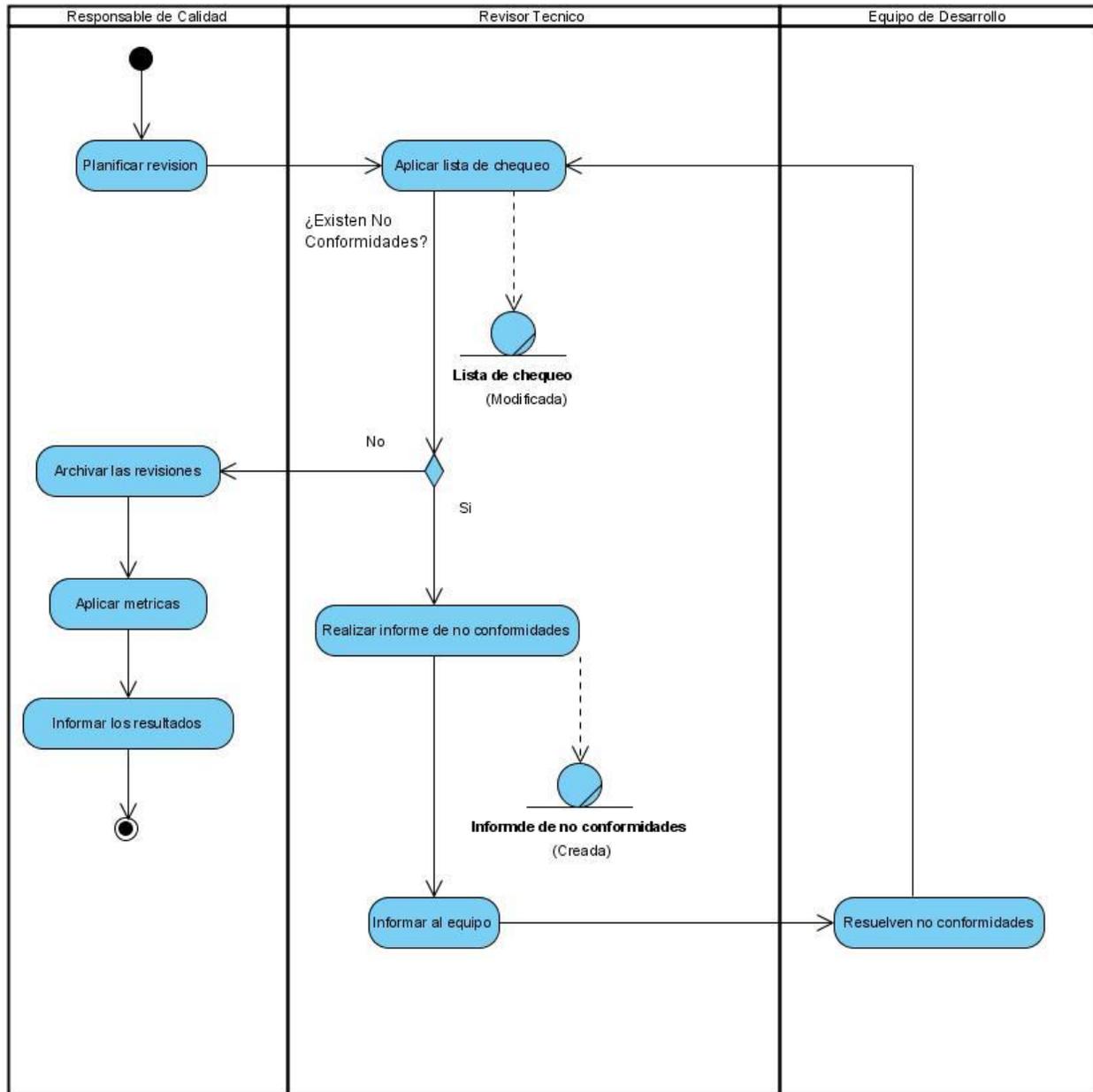
8. Pruebas.

Se propone para esta sección que se consulte el Plan de Prueba del proyecto.

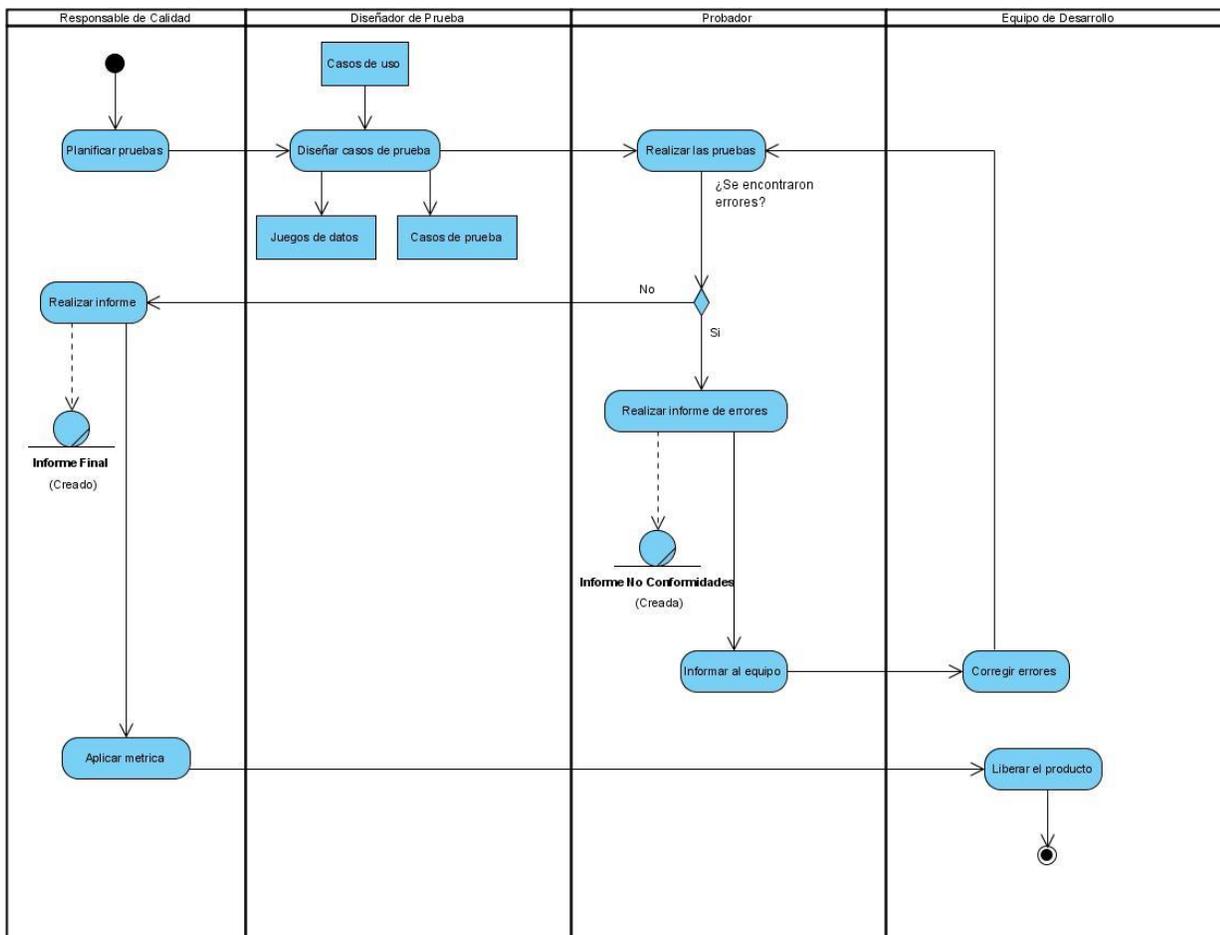
Flujos de Trabajo

Se muestran a continuación los flujos de trabajo descritos para los diferentes tipos de prueba.

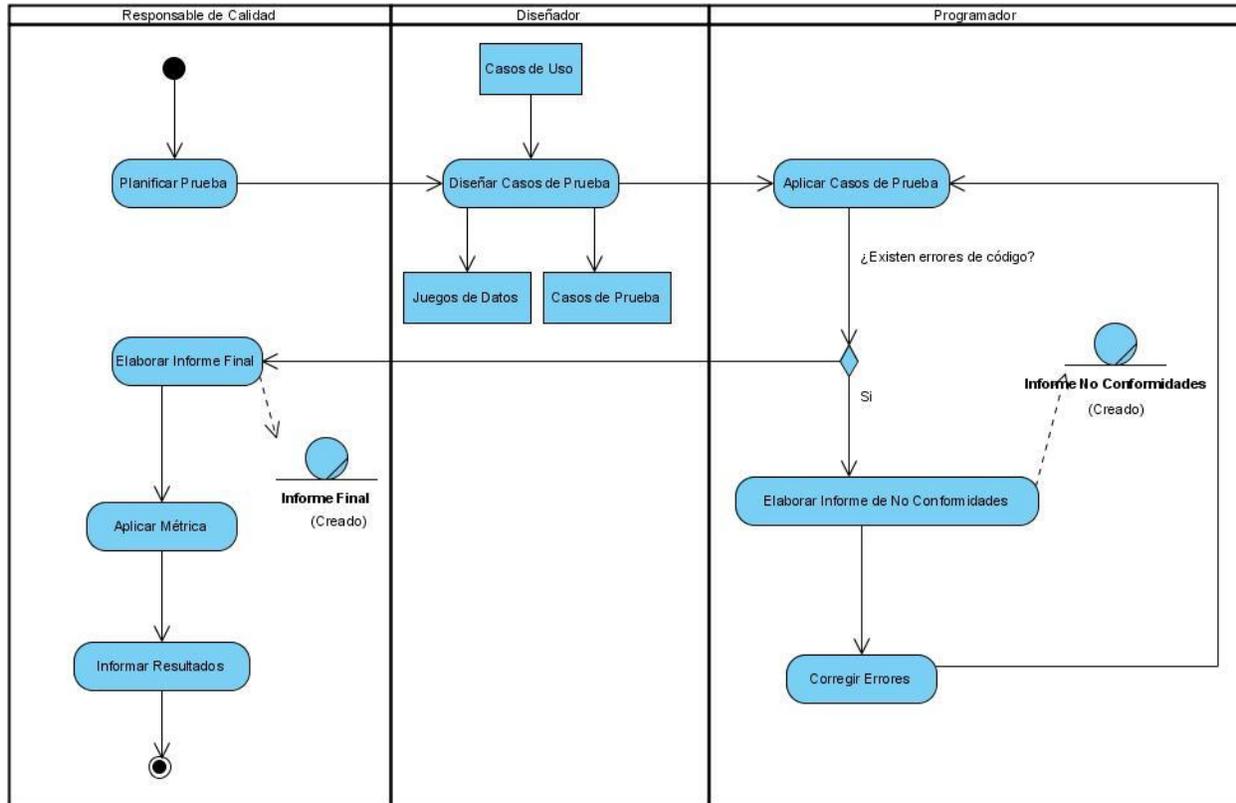
Flujo de trabajo para las actividades de revisión.



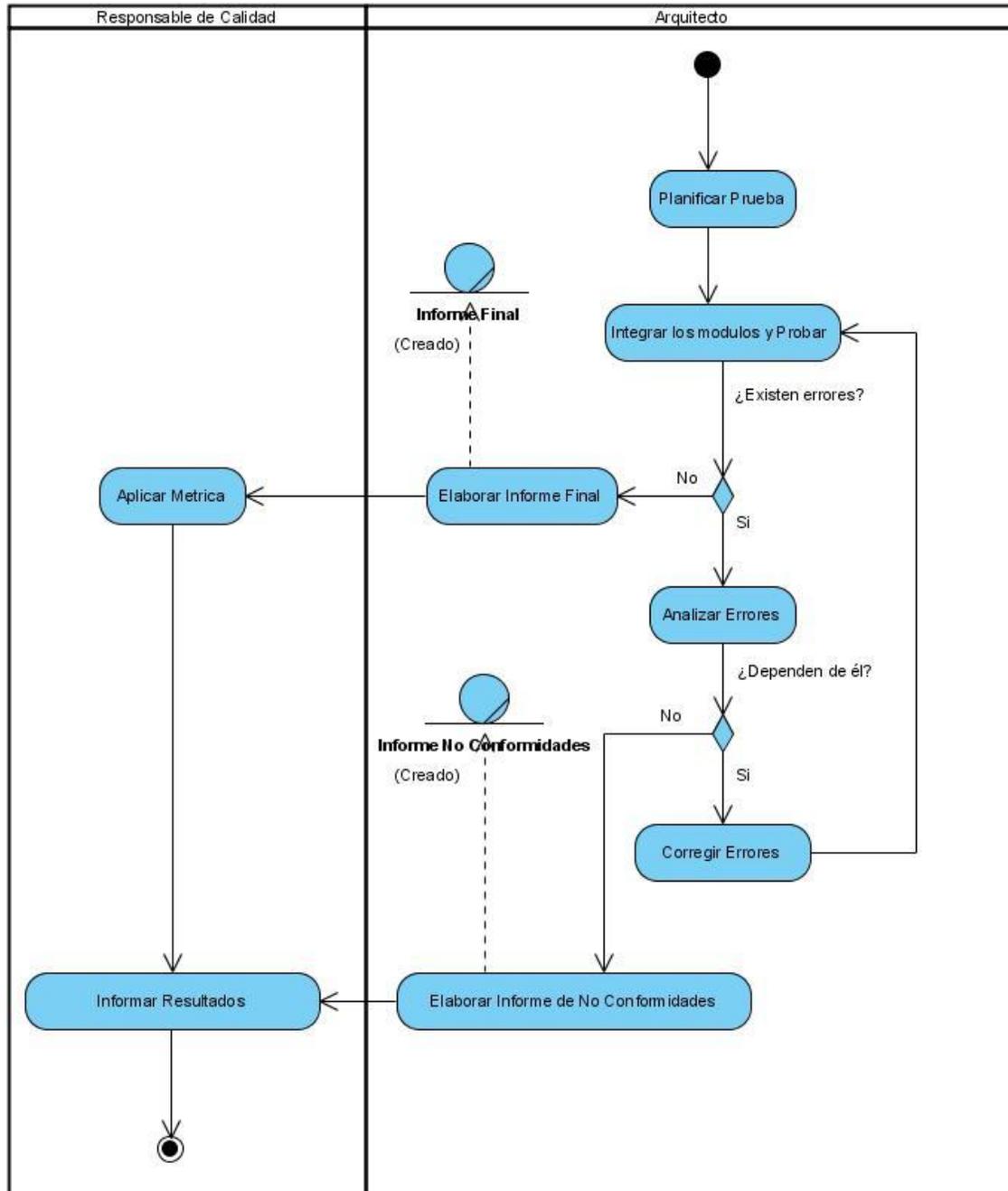
Flujo de trabajo para las pruebas de funcionalidad.



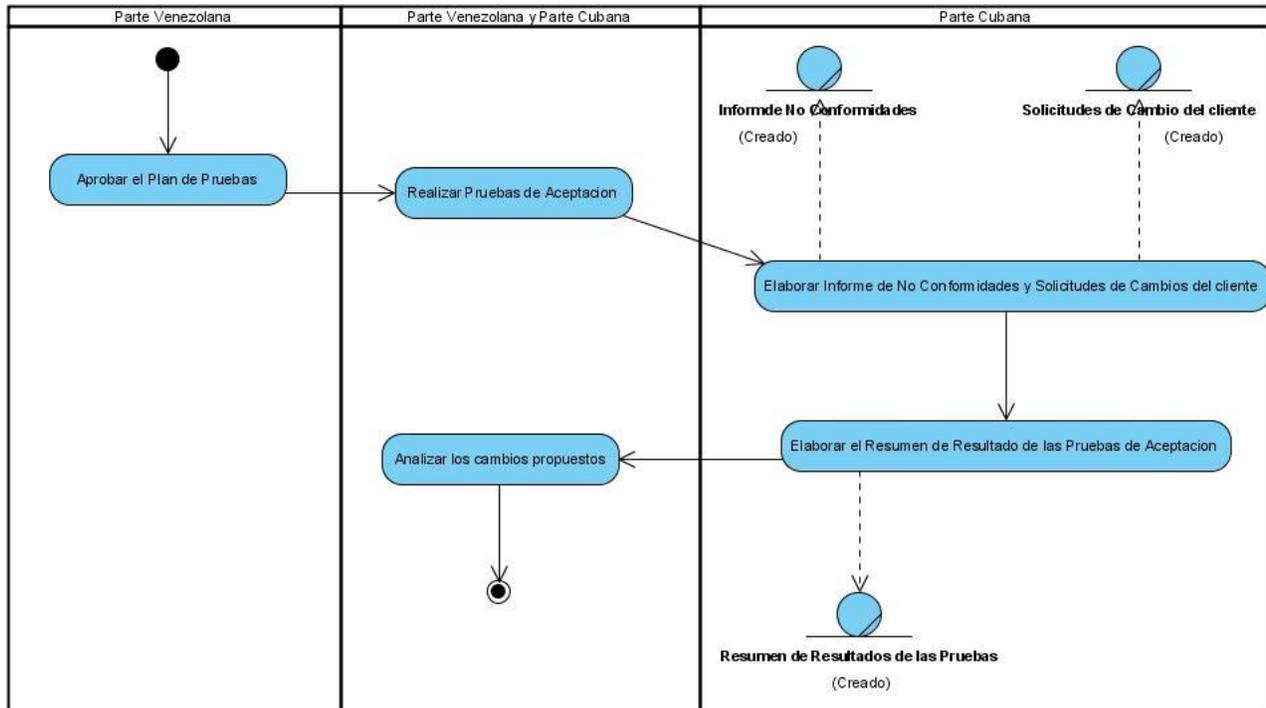
Flujo de Trabajo para las Pruebas de Unidad.



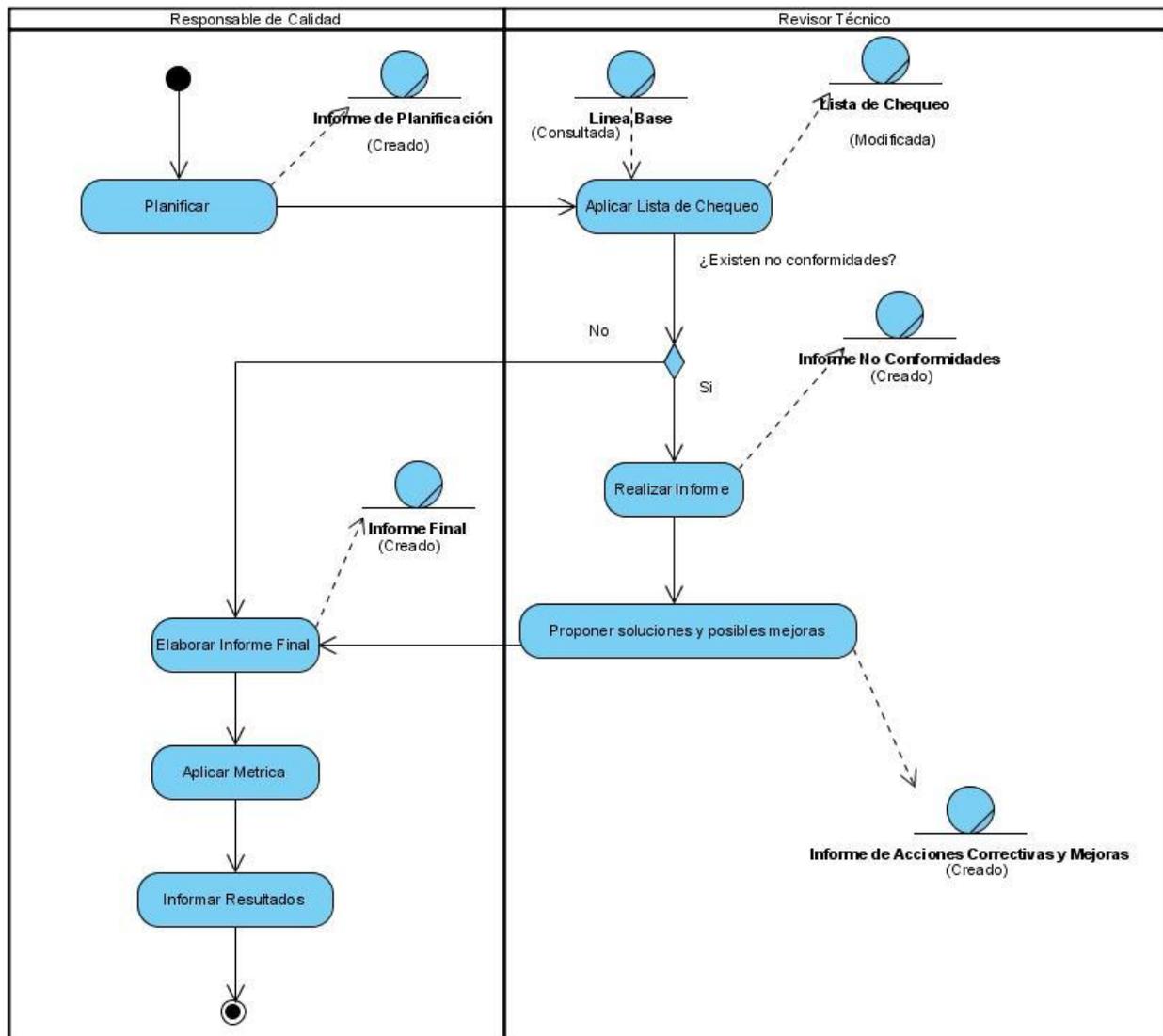
Flujo de Trabajo para las Pruebas de Integración.



Flujo de Trabajo para las Pruebas de Aceptación del cliente.



Flujo de Trabajo para la Auditoría de Gestión de Configuración.



9. Herramientas, Técnicas y Metodologías.

A continuación se describen las diferentes herramientas, técnicas y metodologías que sirvieron de apoyo para la realización de las diferentes secciones del plan.

- ✓ De forma general se propone el uso de Listas de Chequeo para ser usadas durante las actividades de revisiones y auditorías.

- ✓ Generación de casos de prueba a partir de casos de uso
- ✓ Plantillas del expediente de proyecto establecidas por la DCS para registrar la documentación del proyecto.
- ✓ Aplicación de los Lineamientos mínimos de calidad (LMC versión 2.0) de la Universidad.
- ✓ Evaluación bajo las pautas antes establecidas por los responsables de cada área de trabajo.
- ✓ Aplicación de Normas internacionales (IEEE, ISO).
- ✓ Utilización de los conceptos de RUP.
- ✓ Se propone JUNIT para la realización de las pruebas unitarias.
- ✓ Subversión, una herramienta de control de versiones para las pruebas de integración.
- ✓ Redmine para la asignación de tareas.

10. Gestión de Configuración.

Se propone para esta sección que se consulte el **Plan de Gestión de Configuración** del proyecto.

11. Registros de Calidad.

Los registros de calidad del proyecto serán guardados en un repositorio que se mantendrá durante todo el ciclo de vida del proyecto. Estos artefactos serán guardados de acuerdo a los roles que los han generado, por ejemplo, los informes de arquitectura generados por los arquitectos, serán guardados en la carpeta arquitectura.

Elementos que se guardarán en el Registro de Calidad

- ✓ Informe de No Conformidades.
- ✓ Registro de Revisiones y Auditorías
- ✓ Registro de Pruebas.
- ✓ Registro de mejoras.
- ✓ Registro de capacitaciones y certificaciones.
- ✓ Casos de Pruebas.
- ✓ Plan de Pruebas.
- ✓ Listas de Chequeos.
- ✓ Informe de Gestión de Configuración.
- ✓ Informe de RTF
- ✓ Actas de Reunión efectuadas
- ✓ Resúmenes de Resultados
- ✓ Correos de trabajo y comunicación de intercambio de trabajo

12. Entrenamiento.

En vías de garantizar que el equipo de calidad cumpla completamente sus funciones proponemos seguidamente un conjunto de actividades para capacitar al mismo, practicas que deberán aplicarse cuanto antes para que no se atrase el proceso de creación del sistema:

- ✓ **Curso de capacitación sobre los políticas y procesos de aseguramiento de la calidad.**

Con este curso se pretende que el equipo de trabajo obtenga mayores conocimientos del tema en general y así lograr una nivelación entre todos los integrantes de este equipo.

- ✓ **Investigación sobre las normas nacionales e internacionales de calidad.**

Se propone con este trabajo elevar los conocimientos del equipo al respecto y garantizar que todos los integrantes adquieran familiarización con las normas seleccionadas con el fin de trabajar sobre elementos sobre los cuales se ha probado su eficiencia y efectividad.

- ✓ **Investigación detallada de los roles de RUP.**

El objetivo de esta tarea es identificar según cada rol las tareas que se deben realizar, así se podría distribuir mejor el trabajo y no sobrecargar a una sola persona con exceso de responsabilidades. Incluso hay actividades de los roles de calidad que deberán ser combinados con los esfuerzos de otros roles pertenecientes a otras ramas de desarrollo del Sistema.

- ✓ **Curso de capacitación para la confección de las listas de chequeos.**

Se procura que se vayan creando las herramientas y conocimientos necesarios para la realización de este tipo de trabajo.

- ✓ **Curso de capacitación para mejorar el trabajo con las herramientas propuestas anteriormente en este documento.**

Con el objetivo de emplearlas para contribuir al aseguramiento de la calidad del producto software desarrollado.

- ✓ **Curso de capacitación para el diseño de casos de prueba.**

La meta de esta actividad es proporcionar los conocimientos indispensables para realizar de forma satisfactoria este tipo de trabajo.

✓ **Curso de auditoría.**

De esta forma se confirma la estrategia trazada para el desarrollo de las auditorías en el plan, acorde con la establecida por la universidad.

Anexo 2: Plan de Pruebas

1. Introducción.

El presente documento está estructurado para explicar todas las condiciones necesarias para realizar los procesos de pruebas que se harán a través del desarrollo de un proyecto de software.

En el mismo están comprendidos los roles involucrados en los procesos de prueba, así como las responsabilidades que debe atender cada uno de los mismos, se describe el escenario idóneo para realizar las mismas, los modelos de despliegue del sistema en el escenario real de este además de los requisitos de hardware y de software necesarios.

Los requerimientos del sistema que serán probados, además de la estrategia de las pruebas de aceptación y la evaluación de las mismas son los otros elementos comprendidos dentro de este documento que tiene como objetivo planificar todo el proceso de pruebas que se harán al sistema.

1.1. Alcance.

El único proyecto con el que se involucra el presente Plan de Pruebas es el proyecto **SIGEP** en su **Fase IV**.

1.2. Definiciones, Acrónimos y Abreviaturas.

- ✓ UCI: Universidad de las Ciencias Informáticas.
- ✓ SIGEP: Sistema de Gestión Penitenciaria.
- ✓ IEEE: The Institute of Electrical and Electronics Engineers o Instituto de Ingenieros Eléctricos y Electrónicos. (Asociación técnico-profesional mundial dedicada a la estandarización).
- ✓ ISO: Organización Internacional para la Estandarización.
- ✓ RUP: Rational Unified Process.
- ✓ DSC: Dirección de Calidad de Software.

1.3. Referencias.

Código	Título
1	Lista de Chequeo
2	Informe del Proyecto Técnico
3	Informe de Requerimientos
4	Plan de Aseguramiento de la Calidad

2. Roles y responsabilidades.

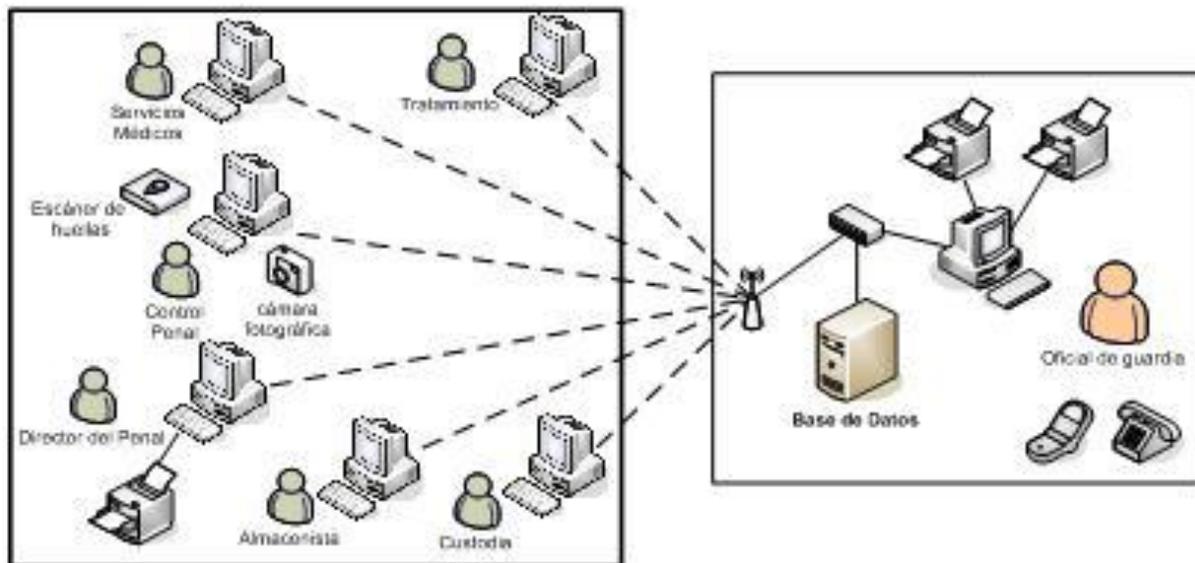
Los probadores que conforman el equipo de calidad de **SIGEP** son los probadores unitarios, los probadores de integración y los probadores de funcionalidad. Los mismos poseen los conocimientos mínimos indispensables para elaborar la tarea que se le es asignada. En la siguiente tabla se muestra la responsabilidad de cada uno de los probadores.

Rol Involucrado	Responsabilidad durante el proceso de pruebas
Probador unitario	<ul style="list-style-type: none"> ✓ Realizar las pruebas Unitarias. ✓ Registrar todos los defectos encontrados durante las pruebas unitarias. ✓ Establecer un seguimiento a los defectos encontrados en las pruebas Unitarias hasta su resolución.
Probador de Integración	<ul style="list-style-type: none"> ✓ Realizar las pruebas de Integración. ✓ Registrar todos los defectos encontrados durante las pruebas de Integración. ✓ Establecer un seguimiento a los defectos encontrados en las pruebas de Integración hasta su resolución.
Probador de Funcionalidad	<ul style="list-style-type: none"> ✓ Realizar las pruebas de Funcionalidad. ✓ Registrar todos los defectos encontrados durante las pruebas de Funcionalidad. ✓ Establecer un seguimiento a los defectos encontrados en las pruebas

de Funcionalidad hasta su resolución.

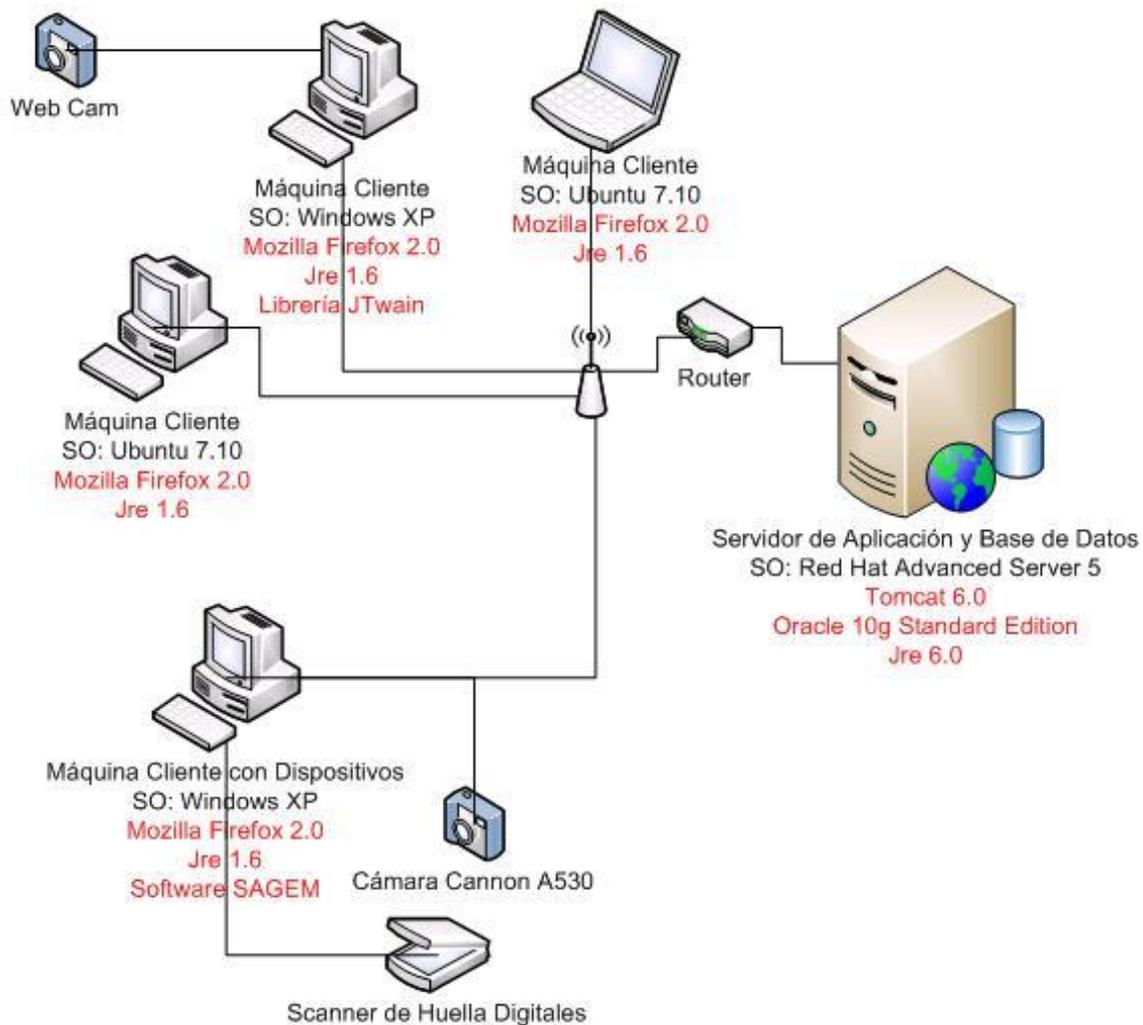
3. Escenario de Pruebas.

Durante la realización de las pruebas se tratará de llegar a simular el ambiente más real posible; garantizando un buen nivel de verificación de las funcionalidades de la aplicación. El siguiente gráfico describe la arquitectura organizacional.



4. Despliegue del sistema

En esta sección se representa el modelo de despliegue del sistema. El mismo es un diagrama con los principales componentes que conformarán el **SIGEP** en un establecimiento penitenciario.



5. Recursos del Sistema.

5.1. Servidores.

Servidores	
<u>Servidor Local 1</u>	
Descripción	Servidor de Base de Datos Central (1G de RAM)
Software Base	Red Hat Advanced Server 5.0
Servicios	
Nombre	Descripción
Oracle 10G Enterprise Edition	Conjunto de plataformas para aplicaciones (APS) que ofrece una solución integral para desarrollar, integrar e implementar todas las aplicaciones, portales y sitios Web de un proyecto.
<u>Servidor Local 2</u>	
Descripción	Servidor de Base de Datos para las Pruebas (1G de RAM)
Software Base	Red Hat Advanced Server 5.0
Servicios	
Nombre	Descripción
Oracle 10G Standard Edition	Ofrece varias características y funcionalidades implementadas en la versión Enterprise. Opciones de Base de Datos como Data Guard, Particionamiento, Spatial, no vienen incluidas con esta versión estándar.
<u>Servidor Local 3</u>	
Descripción	Servidor de Control de Versiones y FTP (1G de RAM)
Software Base	Gentoo

Servicios	
Nombre	Descripción
SVN 1.4	Servicio para el control de versiones
ProFtp	Servicio para servidor de FTP.
<u>Servidor Local 4</u>	
Descripción	Servidor de Base de Datos para Réplica (1G de RAM)
Software Base	Windows XP
Servicios	
Nombre	Descripción
Oracle 10G Standard Edition	Ofrece varias características y funcionalidades implementadas en la versión Enterprise. Opciones de Base de Datos como Data Guard, Particionamiento, Spatial, no vienen incluidas con esta versión estándar.
<u>Servidor Local 5</u>	
Descripción	Servidor para el portal Web (1G de RAM)
Software Base	Red Hat Advanced Server 5.0
Servicios	
Nombre	Descripción
Apache	Servidor web con plugin para PHP
MySql	sistema de gestión de base de datos relacional, multihilo y multiusuario

5.2. PC Clientes.

PC Cliente 1	
Descripción	Máquina de desarrollo para un programador de acceso a datos (1G de RAM)
Software Base	Windows XP o Linux
Servicios	
Nombre	Descripción
Eclipse 3.2 + WTP + Spring IDE + Hibernate Tools.	Herramientas de desarrollo
Toad for Oracle, PLSQL o Tora for Linux.	Herramientas de base de datos
Jdk 1.6	Software que provee herramientas de desarrollo para la creación de programas en java.
Apache Tomcat 6.0	Servidor web con soporte de servlets y JSPs.
Mozilla Firefox 2.0 o superior	Navegador web
PC Cliente 2	
Descripción	Máquina de desarrollo para un programador de Interfaz de Usuario (1G de RAM)
Software Base	Windows XP o Linux
Servicios	
Nombre	Descripción
Eclipse 3.2 + WTP + Spring IDE + Aptana.	Herramientas de desarrollo

Mozilla Firefox 2.0 o superior + Firebug	Navegador web.
Apache Tomcat 6.0	Servidor web con soporte de servlets y JSPs.
Jdk 1.6	Software que provee herramientas de desarrollo para la creación de programas en java.
<u>PC Cliente 3</u>	
Descripción	Máquina de desarrollo para un diseñador (1G de RAM)
Software Base	Windows XP o Linux
Servicios	
Nombre	Descripción
Eclipse 3.2 + WTP + Spring IDE + Aptana.	Herramientas de desarrollo
Mozilla Firefox 2.0 o superior + Firebug	Navegador web.
Apache Tomcat 6.0	Servidor web con soporte de servlets y JSPs.
Jdk 1.6	Software que provee herramientas de desarrollo para la creación de programas en java.
Toad for Oracle, PLSQL o Tora for Linux.	Sistema de gestión de base de datos
<u>PC Cliente 4</u>	
Descripción	Máquina de desarrollo para un diseñador gráfico (1G de RAM)

Software Base	Windows XP o Linux
Servicios	
Nombre	Descripción
Eclipse 3.2 + WTP + Spring IDE + Aptana.	Herramientas de desarrollo
Mozilla Firefox 2.0 o superior + Firebug	Navegador web.
Apache Tomcat 6.0	Servidor web con soporte de servlets y JSPs.
Jdk 1.6	Software que provee herramientas de desarrollo para la creación de programas en java.
Macromedia Dreamweaver	Editor de páginas web
<u>PC Cliente 5</u>	
Descripción	Máquina de Analista (756m de RAM)
Software Base	Windows XP o Linux
Servicios	
Nombre	Descripción
Visual Paradigm 3.0	Herramienta de modelado
<u>PC Cliente 6</u>	
Descripción	Máquina para el rol de Prueba (512m de RAM)
Software Base	Windows XP o Linux
Servicios	
Nombre	Descripción

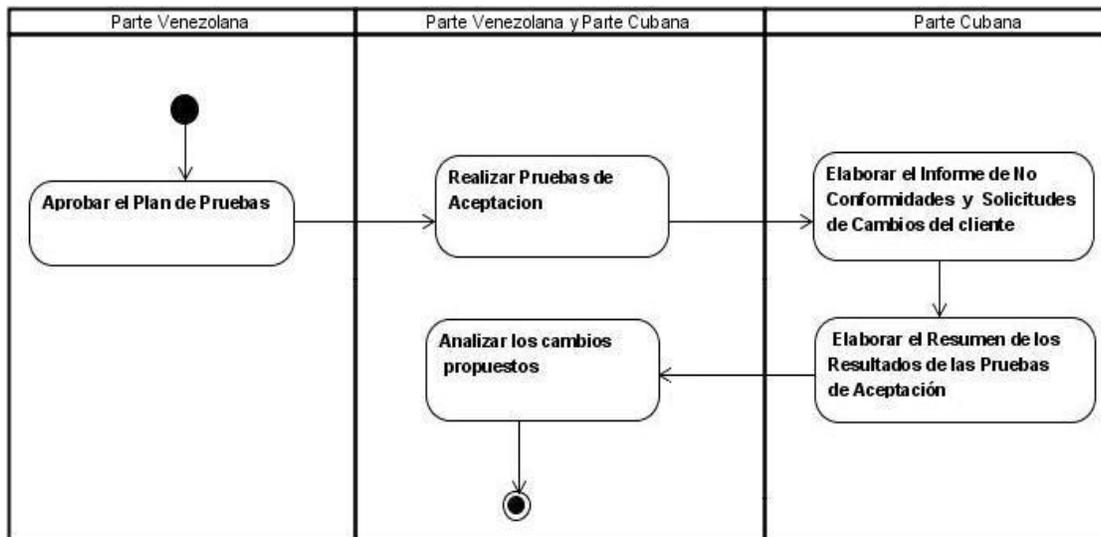
Mozilla Firefox 2.0 o superior + Firebug	Navegador web
Jre 1.6	Conjunto de utilidades que permite la ejecución de programas java sobre todas las plataformas soportadas.

6. Requerimientos a Probar.

En esta sección se hace referencia al documento de requerimientos del proyecto **SIGEP** que se encuentra ubicado en el repositorio del mismo.

7. Estrategia de Pruebas de Aceptación.

El flujo de trabajo que se sigue para realizar las pruebas parciales de aceptación del cliente es el que se describe a continuación.



Descripción del flujo de trabajo.

Las pruebas de aceptación parcial del cliente se inician con la aprobación y firma del Plan de Pruebas (que incluye el plan de trabajo detallado, la relación de los datos del personal que intervendrán en las pruebas y la selección de los módulos a probar). De manera opcional se puede asumir el paso donde se introduce a los especialistas funcionales al uso de las funcionalidades de los módulos a probar.

Garantizadas todas las condiciones técnicas y verificadas por la dirección representante de la parte proveedora, se puede dar inicio al desarrollo de las pruebas de los módulos planificados.

Para ello, intervienen el Especialista funcional cliente correspondiente y el Probador de forma directa con el software y como apoyo, el resto del equipo de desarrollo, esclareciendo las dudas de las funcionalidades que lo requieran.

Durante las pruebas se verificará el cumplimiento de los Procesos Elementales del Negocio, los Casos de Uso y los prototipos correspondientes validados en la etapa de Captura de Requisitos, así como las funcionalidades de cada uno de los módulos planificados.

Las no conformidades del cliente que surjan durante las pruebas serán recogidas en un Informe de No Conformidades, las cuales serán analizadas con posterioridad por el equipo de desarrollo, donde se clasificará y valorará su respectiva respuesta.

Las solicitudes de cambios que aparezcan serán recolectadas en su respectivo Informe de Solicitudes de Cambio, las cuales serán analizadas y llevadas a la mesa de negociaciones entre ambas partes; las aceptadas y pactadas pasarán a ser nuevos requisitos. Los resultados serán contemplados en el Resumen de los Resultados de las Pruebas y discutidos con las partes interesadas llegando a un acuerdo de aprobación y firma de dicho documento. De esta forma concluye el ciclo de las pruebas de aceptación parcial del cliente, genérico para cualquier caso que se aplique.

8. Evaluación de las Pruebas.

Durante el período que se desarrollen las pruebas surgen una serie de inquietudes por parte del cliente, las cuales se clasificarán en No Conformidades y Solicitudes de Cambios. Las No Conformidades se definen como todos aquellos problemas o inquietudes que surjan debido al incumplimiento, ya sea parcial o total, de la implementación de un requisito o parte de él, así como de un elemento o parte de él; en cualquiera de los casos debe encontrarse previamente pactado.

Las Solicitudes de Cambios son todas aquellas inquietudes que surgen y no se encuentran como requisitos o elementos pactados. En este caso, se valorará su inclusión o no dentro de la solución; siempre entre ambas partes, ya que se requiere de una completa negociación.

9. Criterios de aceptación.

Durante el desarrollo de cada tipo de pruebas se debe tener en cuenta cuáles serán los criterios límites de aceptación. En la siguiente tabla los definimos.

Tipo de Pruebas de Aceptación	Criterio de aceptación para las No Conformidades	Criterio de aceptación para las Solicitudes de Cambios
Parcial	<p>Ilimitada:</p> <p>El principal objetivo es detectar el mayor número en este tipo de pruebas.</p>	<p>Ilimitada:</p> <p>El cliente puede emitir todas las que desee, luego serán negociadas por ambas partes.</p>
Total	<p>Si su solución no es inmediata, no deben excederse de 5.</p>	<p>Ilimitada:</p> <p>El cliente puede emitir todas las que desee, luego serán negociadas por ambas partes.</p>

10. Clasificaciones de las No Conformidades.

Como resultado de las pruebas de aceptación, cada No Conformidad, deberá ser analizado en aras de ser clasificada para lograr mejores análisis y control del estado de cada módulo.

Las clasificaciones propuestas se han agrupado en *básicas* y *operativas*: las primeras se utilizarán siempre en todas las pruebas a celebrar, las segundas serán utilizadas según las condiciones de las pruebas lo ameriten.

El grupo de las *básicas* son: Criterio valorativo, Tipo de No Conformidad y Alcance de la No Conformidad. Las *operativas* son: Estatus, Nivel crítico y Nivel de dificultad de la solución.

Criterio Valorativo.

La clasificación hace referencia a la categorización que le puede conferir el equipo de desarrollo a cada No Conformidad emitida por parte del cliente.

Criterio valorativo	Descripción
Procede	Inconformidad encontrada y es positiva a los efectos del equipo de desarrollo.
No Procede	Inconformidad que es considerada no positiva a los efectos del equipo de desarrollo.

Tipo de No Conformidad.

La presente clasificación agrupa las No Conformidades por la naturaleza de su surgimiento, ya sea porque el requisito o elemento no se tuvo en cuenta, no se realizó de forma completa o se desarrolló de forma equivocada.

Tipo de No Conformidad	Descripción
Incumplido	Referido al hecho de que no se realizó nada de lo pactado con respecto a lo señalado, ya sea un requisito o un elemento.

Incompleto	Referido al hecho que la realización de lo señalado ha sido parcial, ya sea un requisito o un elemento.
Erróneo	Referido al hecho de que supuestamente se realizó lo pactado, pero lo conformado no responde a ninguna de las necesidades del cliente, siendo equivocada la solución presentada, puede referirse a un requisito o elemento.

Alcance de la No Conformidad.

La No Conformidad puede estarse referenciándose a un requisito o a un elemento en particular, entonces es importante tener bien definido el alcance de cada No Conformidad, causa por la que aparece la presente clasificación.

Alcance de la No Conformidad	Descripción
Requisito Mal Planteado	No conformidad encontrada que demuestra que no se ha planteado de forma correcta un requisito determinado.
Elementos con Problemas	No conformidad encontrada que demuestra la existencia de un elemento específico dentro de la solución, que no posee las características apropiadas e influye en lo pactado; puede ser de funcionalidad, diseño, etc.

Estatus.

La No Conformidad puede poseer una serie de estados por las que transitan y debe ser clasificada dado el momento que se analicen.

Estatus	Descripción
No Resuelta	Inconformidad No resuelta.
Resuelta	Inconformidad Resuelta.
Pendiente	Inconformidad Pendiente por parte del cliente.
Reincidencia	Inconformidad Reincidente en la falla.

Nivel crítico.

La presente clasificación se referencia al grado de impacto que posee la No Conformidad para el cliente cuando aparece en el producto, así como lo reincidente que puede ser.

Nivel crítico	Descripción
Baja	No conformidad leve y sencilla, que aparece en un reducido número de veces y no influye en nada de las funcionalidades de la solución. Errores ortográficos, desacuerdo en mensajes, colores, etc.
Media	No conformidad de un impacto medio dentro de las funcionalidades de la solución o de reincidente aparición. La presente no interrumpe el funcionamiento del módulo ni impacta en el proceso, pero si podría afectarlo de alguna forma. Validaciones de campos, generación de reportes, etc.

Alta

No conformidad de un considerable impacto en la solución o muy reincidente. Afecta el proceso del negocio, o impide el correcto comportamiento de la funcionalidad.

Nivel de dificultad de la solución.

La presente clasificación se referencia al grado de complejidad que representa para el equipo de desarrollo su posible solución.

Nivel de dificultad	Descripción	Ejemplo
Bajo	Su solución es ligeramente sencilla. Efectos no significativos en la funcionalidad y usabilidad del sistema. Errores de forma. (Amigabilidad del sistema).	No conformidad de algún aspecto de documentación o diseño; algo superficial del sistema. Palabra mal escrita o desacuerdo en el lenguaje, falta de datos en nomencladores.
Media	Su solución posee algún tipo de complejidad. La funcionalidad del sistema no está afectada hasta tal punto que el sistema no puede emplearse.	Validaciones de campos, fechas, etc. Errores en los reportes.

Alto	Su solución es compleja, implicando gran cantidad de tiempo y esfuerzo. El sistema se detiene y no puede avanzar.	Genera error al guardar una transacción de negocio.
------	---	---

10.1. Clasificaciones de las Solicitudes de Cambios.

Durante las pruebas de aceptación surgen un número considerable de Solicitudes de Cambios, ellas deben ser analizadas de conjunto por ambas partes para poder determinar cuales se asumen para la solución software. Es de vital importancia su clasificación para poder valorar con mayor certeza lo que implica para ambas partes su aceptación.

Criterio Valorativo.

La clasificación hace referencia a la categorización que le puede conferir el equipo de desarrollo a cada Solicitud de Cambio emitida por parte del cliente en dependencia de las negociaciones que se realicen. Es válido aclarar que una solicitud podría cambiar de estado en dependencia de las decisiones que se tomen.

Criterio valorativo	Descripción
Procede	Solicitud de cambio emitida por el cliente y aceptada por parte del equipo de desarrollo.
No Procede	Solicitud de cambio emitida por el cliente y rechazada por parte

del equipo de desarrollo.

Alcance de la Solicitud de Cambio.

La presente clasificación se plantea sobre el campo que hace referencia la solicitud, ya que puede estar relacionada a un requisito, un elemento de funcionalidad o de interfaz.

Alcance de la Solicitud de Cambio	Descripción
Nuevo Requisito	Solicitud que implica plantearse un nuevo requisito en la solución pactada.
Modificación de Requisito	Solicitud que implica modificar un requisito en la solución pactada.
Nuevo Elemento	Solicitud que implica plantearse el desarrollo de algún nuevo elemento. Ejemplo: datos de nomencladores, mensajes.
Modificación de Elementos	Solicitud que implica modificar un elemento. Ejemplo: validaciones realizadas que se solicitan cambiar y viceversa.
Modificación de Elementos de Interfaz	Solicitud que implica modificación de algunos

elementos del diseño de la interfaz de usuario:
mensajes, texto, color, posición.

Nivel de dificultad de solución.

La presente clasificación se refiere al grado de complejidad que representa para el equipo de desarrollo su posible solución.

Nivel de dificultad	Descripción	Ejemplo
Bajo	Su solución es ligeramente sencilla. Efectos no significativos en la funcionalidad y usabilidad del sistema. Errores de forma. (Amigabilidad del sistema).	No conformidad de algún aspecto de documentación o diseño; algo superficial del sistema. Palabra mal escrita o desacuerdo en el lenguaje, falta de datos en nomencladores.
Media	Su solución posee algún tipo de complejidad. La funcionalidad del sistema no está afectada hasta tal punto que el sistema no puede emplearse.	Validaciones de campos, fechas, etc. Errores en los reportes.

Alto	Su solución es compleja, implicando gran cantidad de tiempo y esfuerzo. El sistema se detiene y no puede avanzar.	Genera error al guardar una transacción de negocio.
------	---	---

11. Evaluación de las pruebas.

Una vez concluido el periodo de pruebas se realizará la evaluación de la misma siendo aprobada si se cumplen los siguientes criterios:

1. Asistencia y puntualidad por parte del equipo de probadores en un 80 por ciento. Para lograr esto se chequeara diariamente la asistencia mediante el registro de asistencia al laboratorio.
2. Cumplimiento del plan de pruebas de aceptación y en especial del flujo de trabajo descrito

La evaluación quedará reflejada en el documento “Resumen de los Resultados de las Pruebas de Aceptación” y el mismo será elaborado y aprobado por los representantes del proyecto SIGEP de conjunto con la parte venezolana.

12. Cronograma.

No.	Tarea	Fecha	Responsable	Participantes	Observaciones
1	Pruebas de unidad	Al terminar la programación de los módulos correspondiente.	Equipo de calidad	Programadores de los módulos en cuestión que serán probados.	Este tipo de prueba será realizado en pares. Un programador de un módulo le revisará el trabajo a otro

						programador de otro módulo.
2	Pruebas de integración	Al estar subidos al repositorio los módulos correctamente terminados y probados.	Equipo de calidad	de	Arquitecto	El arquitecto se encarga de integrar los módulos en un solo sistema y probar que este funcione correctamente.
3	Pruebas de funcionalidad	Después de estar terminado el módulo en cuestión a probar.	Equipo de calidad	de	Equipo de calidad	de Con estas pruebas se verifica la aceptación de los datos, el proceso, la recuperación y la implementación correcta de las reglas del negocio
4	Pruebas de seguridad	Después de estar terminado y el sistema y que el mismo esté	Equipo de calidad	de	Revisor técnico	Con estas pruebas se garantiza que el producto final tenga altos

		funcionando.				niveles de fiabilidad.
5	Pruebas de aceptación del cliente	Cuando se haya completado el ciclo de vida del software y esté en condiciones de ser entregado al cliente.	Equipo de calidad	de	Equipo de calidad clientes.	de Con estas pruebas el cliente puede determinar si acepta o no el producto que fue elaborado.
6	Pruebas de rendimiento	Después que se ha completado el funcionamiento del software	Equipo de calidad	de	Revisor Técnico	Mediante estas pruebas se puede medir que tanto rendimiento soporta el software, comprobando de esta manera su correcto funcionamiento.

GLOSARIO DE TERMINOS

SIGEP: Sistema de Gestión Penitenciaria.

Fase de Desarrollo: en el marco del SIGEP está determinado por aquellos momentos en los que se libera una porción del producto o el sistema completo.

UCI: Universidad de las Ciencias Informáticas.

DSC: Dirección de Calidad de Software.

PAC: Plan de Aseguramiento de la Calidad.

EP V2.0: Expediente de Proyecto versión 2.0.

RTF: Revisiones Técnicas Formales.

RUP: Rational Unified Process. Proceso unificado de desarrollo.

SQA: Software Quality Assurance. Aseguramiento de la Calidad de Software.

PPQA: Garantía de la Calidad de Procesos y de Productos. Área de procesos de CMMI.

CMMI: Capability Maturity Model Integration. Es un modelo para la mejora de procesos que proporciona a las organizaciones los elementos esenciales para procesos eficaces.

UML: Unified Modeling Language. Lenguaje Unificado de Modelado. Lenguaje de modelado de sistemas de software.

DNSP: Dirección Nacional de Servicios Penitenciarios.

Estrategia: Plan que permite la mejor distribución de los recursos y medios disponibles a efectos de poder obtener aquellos objetivos deseados mediante una secuencia de pasos establecidos.

SUA: Sistema Único de Aduana

MPPRIJ: Ministerio del Poder Popular para las Relaciones Interiores y Justicia.