

Universidad de las Ciencias Informáticas

Facultad 4



**Título: Propuesta de un Plan de Verificación y
Validación para el proyecto SIGEP**

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

Autores: Brisey López Bello

Javier Arza Valdes

Tutor: MSc. Michael González Jorrín

Co-tutor: Ing. Geiser Arcio Pérez Rivas

Consultante: Ing. Lizandra Arza Pérez

Junio, 2008

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los 25 días del mes de junio del año 2008

Autor: Brisey López Bello

Autor: Javier Arza Valdes

Tutor: Msc. Michael González Jorrín

DATOS DE CONTACTO

Tutor: MSc. Michael González Jorrín

Graduado en el 2000 en el Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), de Ingeniero Informático.

Máster en Gestión de Proyectos Informáticos.

Categoría Docente de Asistente.

Con 8 años de experiencia laborar vinculado al desarrollo de software y la docencia universitaria.

Con 5 años de experiencia en el trabajo relacionado con la Calidad de Software.

Ha realizado varias publicaciones en eventos y revistas nacionales relacionadas con el tema de la Calidad de Software.

Se desempeña como especialista superior en la Dirección de Calidad de Software de la UCI.

Co-tutor: Ing. Geiser Arcio Pérez Rivas

Graduado en la Primera Graduación de la UCI en el curso 2006-2007.

Profesor del Dpto. de Ingeniería de Software de la Facultad 4 de la UCI.

Especialista de la Dirección de Calidad de Software de la Universidad.

Consultante: Ing. Lizandra Arza Pérez

Profesor Instructor Graduado

Graduada en el 2002 de Ingeniero Informático del Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), Título de Oro

Ha ocupado los cargos de, Directora de Formación para la Producción, Directora de Calidad de Software y Directora de Producción de la UCI.

Se desempeña como Decana de la Facultad 4 de la UCI

Jurado de la Feria de Informática 2003

Miembro del Comité Organizador del Evento de Calidad en la Feria Internacional Informática 2004 y 2005

Comité Organizador del Primer Evento de Ingeniería de Software y Calidad en el Evento UCiencia.

Miembro del Grupo Nacional de Calidad.

“Y vegetar no es vivir. Vivir es tener algo que hacer. Vivir es tener una meta, un objetivo, una tarea, una obra a la cual dedicar el tiempo, la energía, y dedicar la vida...”

Fidel Castro Ruz

AGRADECIMIENTOS

A la Revolución, por darnos la oportunidad de servirle desde el corazón y el trabajo.

A la UCI, por habernos hecho crecer como personas y como profesionales, por formarnos, por ser nuestra casa todos estos años.

A la FEU y la UJC, por hacernos mejores personas, mejores cubanos, por comprometernos cada día más.

A nuestros profes, por todas las enseñanzas y exigencias que trazaron el camino hasta el día de hoy.

A tío Reynold y tía Cecy, por ser nuestros padres y brindarnos un hogar durante estos cinco años.

A Yude, por su cariño y apoyo incondicional desde el primer día en la universidad.

A Liza, por su apoyo en estos años y toda su ayuda en la realización de este trabajo.

A Michael, por ser nuestro guía y soporte en la realización de este trabajo.

A Geiser, por su colaboración.

A todos los miembros del proyecto SIGEP, por los años compartidos y por su gran ayuda en la realización de este trabajo.

A todos los amigos, por compartir alegrías, tristezas, por ser parte de nuestra familia en la universidad

Brisey y Javier

A machy, por darme la vida, por su amor, ejemplo y entrega, te adoro mamita...gracias por existir.

A papi, que a pesar del tiempo y la distancia siempre ha estado cerca, con su cariño, ejemplo y ayuda.

A mi abuela Elba, por todo su amor y entrega desde el primer día.

A mi abuela Lelé, por todo su cariño.

A mis hermanos Isa y Joaco, por quererme tanto, por ser como son, por imponerme el reto de ser su ejemplo.

A mis hermanitas Jessi y Tina, por su ternura, por acordarse siempre de mí.

A mi neni, mi Javi, por regalarme estos años, por ser amigo, compañero, consejero, por ayudarme tanto y por ser mi amor.

A toda mi familia, los quiero mucho, gracias por tenerme siempre presente.

A Grise, por todo su cariño y dedicación en estos largos años.

A Luis, por todo su apoyo y cariño.

A todos mis amigos, a los viejos y a los nuevos, gracias por comprenderme, acompañarme y enseñarme.

Gracias a todos, por todo, por ser parte de mi vida....

Brisey

A mami por estar cada día, por hacer de mí el hombre que soy, por ser una gran mujer, por ser única, por ser mi vida.

A papi, por su ejemplo, su cariño, su ayuda incondicional, por hacerme sentir tan orgulloso.

A tato, por su amor, por ser tan especial, por guiarme toda la vida.

A abuela Cristina por su amor, por estar siempre aquí, aunque no te pueda ver.

A abuela Amparo por hacerme tan feliz, por darle solo alegrías a mi vida.

A tío Agustín por ser como un padre para mí.

A tía Miriam por su cariño sin límites durante todos estos años.

A mis primos Liza, Leane, Laura, Agu, Yiyo, Ranyi, Raude, por su apoyo y cariño desde el primer día.

A toda mi familia, por quererme y ayudarme en todo momento.

A mi nena, a mi Brise, por ser mi amiga, mi soporte, por entregarme su vida todos estos años, gracias mi amor.

A machy por ser especial, gracias por tu amor.

A Isa por su ternura, por darme tanto cariño.

A mis amigos, los de ahora, y los de siempre, gracias por no olvidar el verdadero sentido de la amistad.

Gracias a todos, siempre los llevo conmigo.

Javier

DEDICATORIA

A machy y a papi

A Isa, Joaco, Jessi y Tina

A mis abuelas Elba y Lele

A mi neni

Brisey

A mami y papi

A Tato

A Cristina y Amparo

A mi nena

Javier

RESUMEN

El Aseguramiento de la Calidad es una necesidad de toda empresa productora de software en el afán de implantar soluciones propias en medio de la creciente competencia global, y con esto obtener los beneficios económicos y sociales correspondientes. Resulta imprescindible el uso de buenas prácticas durante el desarrollo que conduzcan hacia productos terminados con calidad. Se requiere además de procesos que contribuyan a verificar, validar y controlar que dichas prácticas sean realizadas correctamente.

La presente investigación se centra en el estudio de los procesos de Verificación y Validación (V&V) en el proyecto Sistema de Gestión Penitenciaria (SIGEP) como parte del Aseguramiento de la Calidad del mismo. El objetivo general de la investigación es elaborar un Plan V&V que facilite la introducción de buenas prácticas en los procesos de Verificación y Validación además de la estandarización de estos en el entorno productivo citado. Basado en los estudios realizados se decide que la propuesta estará sustentada por el estándar internacional IEEE 1012-1998, y se enfocará al Proceso de Desarrollo del SIGEP, incluyendo desde las actividades relacionadas con el modelado del negocio hasta la instalación y aceptación del producto final.

PALABRAS CLAVES

Aseguramiento de la Calidad, Verificación, Validación, SIGEP, Plan V&V, Proceso de Desarrollo.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA	8
1.1. Introducción.....	8
1.2. Ciclo de vida del software.....	8
1.3. Calidad del software.....	10
1.4. El Aseguramiento de la Calidad.....	11
1.5. Verificación y Validación.....	12
1.5.1. Definiciones.....	12
1.5.2. Objetivos del proceso de Verificación y Validación.....	13
1.5.3. V&V Independiente.....	13
1.6. Métodos V&V.....	15
1.6.1. Controles estáticos manuales informales.....	16
1.6.2. Controles estáticos manuales disciplinados.....	16
1.6.3. Pruebas.....	18
1.7. V&V en la industria del software.....	25
1.7.1. V&V en CMMI.....	25
1.7.2. Planificación de la V&V.....	28
1.8. Modelos del ciclo de vida del software.....	30
1.8.1. Sobre estándar ISO/IEC 12207.....	33
1.9. Conclusiones parciales.....	35
CAPÍTULO II. SOLUCIÓN PROPUESTA	36
2.1. Introducción.....	36
2.2. Características del proyecto SIGEP.....	36
2.3. Guía de adopción del estándar IEEE 1012-1998.....	37
2.3.1. Nivel de integridad del software.....	38

2.3.2.	Mínimo de tareas recomendadas para el nivel de integridad del software	39
2.3.3.	Intensidad y rigor de las tareas V&V	39
2.3.4.	Criterios que se emplearán en las tareas V&V.....	39
2.3.5.	Características del esquema productivo del proyecto SIGEP	40
2.4.	El Plan V&V SIGEP	41
2.4.1.	Estructura	41
2.4.2.	La Actividad V&V	46
2.4.3.	La Tarea V&V	48
2.4.4.	Métodos y procedimientos aplicados	54
2.5.	Conclusiones parciales.....	57
CAPÍTULO III: VALIDACIÓN DE LA PROPUESTA		59
3.1.	Introducción.....	59
3.2.	El Método Delphi	59
3.3.	Aplicación del método.....	60
3.3.1.	Proceso de selección de los Expertos.....	60
3.3.2.	Elaboración del cuestionario	62
3.4.	Resultados del proceso de aplicación del método	63
3.5.	Otros resultados que validan la propuesta.....	69
3.6.	Conclusiones parciales.....	70
CONCLUSIONES		71
RECOMENDACIONES		72
REFERENCIAS BIBLIOGRÁFICAS		73
GLOSARIO.....		76
ANEXOS		80

ÍNDICE DE FIGURAS

Figura 1. Situación de los proyectos de sistemas 1994 - 2004. Fuente:(GROUP 2004).....	1
Figura 2. Actividades de control. Fuente: (ANTONIO 1999).....	15
Figura 3. Modelo de pruebas clásico. Fuente: (KYNETIA 2007)	22
Figura 4. Modelo de pruebas en V. Fuente: (KYNETIA 2007).....	23
Figura 5. Modelo de pruebas en W. Fuente: (KYNETIA 2007).....	24
Figura 6. Flujo de trabajo de pruebas de RUP. Fuente: (BOOCH et al. 2000).....	25
Figura 7. Marco de Trabajo V&V. Jerarquía de procesos, actividades y tareas. Fuente:(IEEE 1998b).	29
Figura 8. Distribución de roles del proyecto SIGEP.....	37
Figura 9. Experiencia de trabajo de los miembros del proyecto	40
Figura 10. Vista general del Plan V&V SIGEP	41
Figura 11. Estructura del Plan V&V SIGEP	41
Figura 12. Cronograma tipo	43
Figura 13. Proceso de Desarrollo V&V.....	46
Figura 14. Conceptualización V&V.....	47
Figura 15. Estructura de la Tarea V&V.....	48
Figura 16. Vida de las pruebas a través de las Actividades V&V SIGEP	56
Figura 17. Diagrama del procedimiento del Método Delphi. Fuente: (TORRES <i>et al.</i> 2005).....	60
Figura 18. Evaluaciones de los expertos: Objetivo General #1	64
Figura 19. Evaluaciones de los expertos: Objetivo General #2	65
Figura 20. Evaluaciones de los expertos: Objetivo General #3	66
Figura 21. Evaluaciones de los expertos: Objetivo General #4	67
Figura 22. Evaluaciones de los expertos: Objetivo General #5	68

ÍNDICE DE TABLAS

Tabla 1. Esquema de niveles de integridad.....	38
Tabla 2. Objetivos de los tipos de preguntas.....	62
Tabla 3. Evaluaciones positivas y negativas del Objetivo General # 1	63
Tabla 4. Evaluaciones positivas y negativas del Objetivo General # 2	64
Tabla 5. Evaluaciones positivas y negativas del Objetivo General # 3	65
Tabla 6. Evaluaciones positivas y negativas del Objetivo General # 4	66
Tabla 7. Evaluaciones positivas y negativas del Objetivo General # 5	67

INTRODUCCIÓN

Los proyectos de desarrollo de software han padecido tradicionalmente problemas de calidad, tanto en el propio proceso de desarrollo como en los productos que entregan. Esta problemática tiene su origen en las habituales desviaciones de plazos y esfuerzo sobre los valores previstos y en la frecuente aparición de fallos durante la implantación y operación de los productos resultantes.

Los principales problemas son de naturaleza diferente: por un lado las empresas no dedican recursos ni la atención necesaria para que la actividad tenga éxito y por otro que los esfuerzos que se hacen en su mayoría son reactivos, es decir, cuando ya detectaron un problema en el producto final. Para muchas organizaciones, la información y la tecnología que la soporta representan los activos más valiosos de la empresa; es más, en el competitivo, rápido y cambiante ambiente actual, la gerencia ha incrementado sus expectativas relacionadas con la entrega del servicio y la calidad en cada uno de los sistemas de información. (LEBRÚN *et al.* 2008)

Estudios realizados por diferentes autores demuestran que un alto porcentaje del éxito o fracaso del proyecto está en la tecnología disponible y en el conocimiento que se tenga de ella, además, en la forma en que el proyecto lleva un control de calidad durante su desarrollo. A escala mundial, en el año 2004, el desempeño de los proyectos de sistemas fue: 29% de ellos exitosos, un 53% proyectos cuestionables y un 18% proyectos fallidos, arrojando una cifra que supera los 90 mil millones de dólares de desperdicio. Casi el 25% de los proyectos de software fueron cancelados por atraso o por ejecutarse con un presupuesto mayor que el pactado, por tener una calidad deficiente o por experimentar alguna combinación de estos factores. (GROUP 2004)

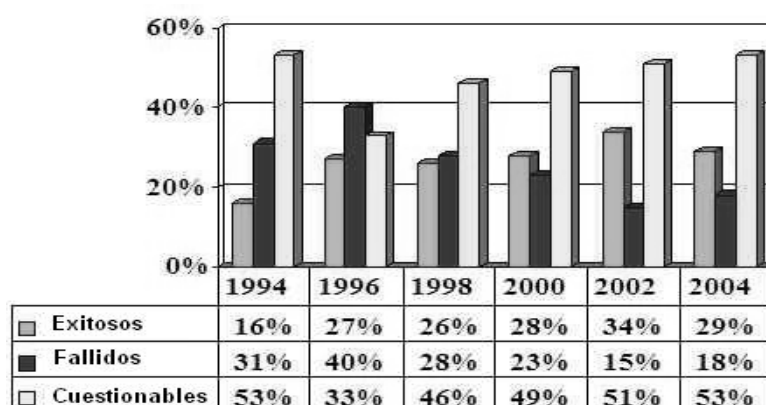


Figura 1. Situación de los proyectos de sistemas 1994 - 2004. Fuente:(GROUP 2004)

Estos datos reflejan una situación que aún hoy afecta a la mayoría de las empresas de software, incluso las más prestigiosas, y se agrava mucho más si se tiene en cuenta que cada vez se trabaja con

una orientación mayor hacia procesos críticos en estas empresas. Los software de gestión están llamados a ser, según las tendencias actuales, la premisa fundamental de la industria de desarrollo de software a nivel global, precisamente por su implicación en la rama industrial y de los servicios, modelando y automatizando la mayoría de los procesos que se desarrollan en la empresa actual: gestión de recursos humanos, contabilidad y finanzas, gestión de inventarios, etcétera.

El aseguramiento de calidad en el software, no es un lema, es una necesidad. Al igual que la Ingeniería de Software, que sugiere la integración de actividades de ingeniería en el desarrollo de software, también es necesario el uso de prácticas que conduzcan hacia productos terminados con calidad. De igual manera, se requiere de procesos que contribuyan a verificar, validar y controlar que dichas prácticas sean realizadas correctamente.

Se han definido varios modelos basados en las experiencias exitosas de la Ingeniería de Software que sirven de guía para las mejoras y unifican los criterios de evaluación de las empresas.

Numerosos estudios relativos al beneficio obtenido a raíz de procesos de mejora, reportan que la implantación de un proceso formal de *Verificación y Validación* supone importantes ventajas para las organizaciones productoras de software (GUZMÁN *et al.* 2006):

- Se produce un incremento de la satisfacción del cliente al utilizar un software con una cantidad de errores inferior.
- Se incrementa la eficiencia del proceso de desarrollo.
- Se facilita la definición y cumplimiento de los objetivos de calidad.
- Se incrementa la satisfacción de los trabajadores debido a que se proporcionan herramientas y recursos apropiados para la realización eficiente del trabajo.

Además, las empresas participantes en los mencionados estudios han reportado la consecución de beneficios económicos asignables a las actividades de mejora e introducción de buenas prácticas en el proceso de *Verificación y Validación* de software. Algunos ejemplos son los siguientes (GUZMÁN *et al.* 2006):

- Reducción de un 20% en los errores en el software entregado al cliente (BKIN Software).
- El esfuerzo en pruebas de software se redujo desde el 25% al 20% del esfuerzo total del proyecto (BKIN Software).
- El número de errores detectados en las pruebas de aceptación es menor del 12% de los errores detectados en las pruebas de integración. El valor anterior era del 37%. (BKIN Software)

- Los errores informados por el cliente/usuario se han reducido en un 77% (Archetypon).
- 30% de reducción en los costes de ejecución (IMB SEMEA SUD).
- Reducción en el tiempo de entrega e incremento en la eficiencia de las pruebas (Nokia Network Management Systems).

En el año 2007 Guillermo de la Cuesta en un artículo titulado *Comunicación eficiente: parte esencial en los cambios empresariales* hizo referencia a la existencia de un problema subjetivo en la calidad de software que da al traste con la necesidad de estas transformaciones: la resistencia al cambio. Esto sucede cuando no están especificados los sistemas o subsistemas objetos de transformación, apareciendo dos tendencias: quienes se resisten y los defensores. Estos últimos despliegan una sensata lucha mediante apreciaciones que contribuyen a ratificar o rectificar algunos de los aspectos que facilitan el desarrollo del trabajo. El convencimiento es la forma válida para lograr el apoyo mayoritario a los cambios propuestos, de modo que los trabajadores varíen también la forma de pensar y actúen con criterios propios, responsablemente y con un comportamiento estable. Solo la comunicación efectiva puede lograr esos resultados. (CUESTA 2007)

Mario Saffirio en su artículo *Mi experiencia en la Ejecución de Proyectos en las Empresas*, expone una conclusión que resume la importancia de entender la necesidad de mejorar los procesos que se realizan sistemáticamente: “La oposición al cambio he visto, más de una vez, que ha conducido al fracaso de un proyecto. Por eso es importante considerarla y aprender cómo abordarla”.(SAFFIRIO 2005)

En medio de toda esta coyuntura, Cuba busca abrirse paso con sus producciones intelectuales, que tendrán que ser inevitablemente el sustento económico de la nación. El Comandante en Jefe lo señalaba: “...nuestra sociedad será cada vez más una sociedad de técnicos y de hombres de ciencia”

La Industria Cubana del Software (INCUSOFT) está llamada a convertirse en una significativa fuente de ingresos para el país, como resultado del correcto aprovechamiento de las ventajas del alto capital humano disponible.

La promoción de INCUSOFT en el ámbito internacional ha tenido, como línea estratégica, aprovechar la enorme credibilidad que tiene Cuba en sectores tales como la salud, la educación y el deporte.

La Universidad de las Ciencias Informáticas (UCI) es la primera universidad creada en el marco de la Batalla de Ideas que libra el país. Ha sido fundada con una nueva concepción: una universidad productiva. Una de sus misiones es la producción de software y servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación, logrando una fuerte relación Universidad-

Empresa. Esta universidad que convierte a la producción en sustento económico, político y social; es productora de soluciones informáticas y está comprometida a ser la vanguardia del desarrollo de software en Cuba, llevando la informatización a todos los sectores de la sociedad, de manera tal que convierta a la industria de software en un renglón fundamental de la economía del país, en cumplimiento de los objetivos que se propone.

Para lograr que la UCI se adueñe de una posición ventajosa en el competitivo mercado mundial del software, es imprescindible crear productos de calidad que se distingan por ser confiables, mantenibles y flexibles, donde los costos de mantenimiento y perfeccionamiento durante el tiempo de explotación sean cada vez más factibles.

Uno de los proyectos que desarrolla la universidad es el Sistema de Gestión Penitenciaria (SIGEP), que constituye la solución de software que incluye el proyecto de Humanización Penitenciaria Venezolano y tiene el objetivo de dar respuesta a las necesidades de gestión, información y apoyo a la toma de decisiones de la Dirección General de Custodia y Rehabilitación del Recluso (DGCRR) de la República Bolivariana de Venezuela, en sus tres niveles. Con este sistema se busca:

- Aumentar la eficacia, profesionalismo y la equidad en el sistema penitenciario venezolano, con el objetivo de lograr confianza en el sistema penitenciario en general.
- Generar y diseminar información vital para el funcionamiento de los establecimientos penitenciarios. Esto permitirá generar estadísticas confiables y actualizadas sobre la situación jurídica de los privados de libertad, condiciones de vida y salud, actividades de rehabilitación y reinserción, la situación operativa, la actividad administrativa, entre otras.
- Permitir la comunicación en línea con tribunales, sistemas de identificación y antecedentes penales, que complementan la información necesaria para la gestión de los procesos vinculados a los privados de libertad.

En este proyecto los procesos de Verificación y Validación son atendidos desde etapas tempranas. Pero sucede que las tareas que se relacionan con estos se llevan a cabo muchas veces sin tener en cuenta el impacto real que tienen para la V&V. Existen áreas dentro del desarrollo que no reciben la atención requerida. Muchos de los mecanismos empleados no son los más óptimos y no se genera la evidencia (documentación) suficiente para cada tarea concerniente a la V&V.

Se tiene, por ejemplo, que los artefactos relacionados con la documentación del usuario (manuales de usuario, ayuda en línea) no son sometidos a revisiones internas previas a las pruebas de aceptación

con el cliente. Además de que para aspectos como la integración del sistema no se cuenta con un espacio planificado y normado dentro de la actividad de pruebas del software.

La presencia de asuntos como los mencionados se constató en entrevistas realizadas a diferentes miembros del equipo de desarrollo (Anexo C), y como resultado de la observación de los autores de la investigación; de lo que se pudo concluir además, que la base fundamental de la problemática planteada está en que no existe un documento formal que estandarice y guíe los procesos de Verificación y Validación para el proyecto.

El análisis de esta situación trae consigo la definición de que el **problema científico** de la investigación sea *¿cómo elaborar un Plan V&V que facilite la introducción de buenas prácticas en los procesos de Verificación y Validación además de la estandarización de estos en el proyecto SIGEP?*

Partiendo del problema planteado se tiene que el **objeto de estudio** se enfocará al *Aseguramiento de la Calidad en los proyectos de software de gestión.*

El **campo de acción** de la investigación lo constituyen *los procesos de Verificación y Validación en el proyecto SIGEP.*

Para resolver el problema planteado con anterioridad se propone como **objetivo general**: *Elaborar un Plan V&V que facilite la introducción de buenas prácticas en los procesos de Verificación y Validación además de la estandarización de estos en el proyecto SIGEP.*

El citado objetivo se desglosó en otros más **específicos**:

1. Identificar las actividades de Verificación y Validación que se realizan en el proceso de desarrollo del SIGEP, así como las posibles mejoras a estas.
2. Identificar en la literatura y normas internacionales nuevas actividades y tareas de Verificación y Validación que se deben incorporar a las existentes en el SIGEP.
3. Conformar una propuesta de Plan V&V que incluya buenas prácticas de Verificación y Validación ajustado a las características del proyecto.
4. Validar la propuesta de solución.

Para satisfacer los objetivos planteados se realizaron las siguientes **tareas de la investigación**:

1. Identificar a partir del estudio del proceso de desarrollo del proyecto SIGEP, qué actividades de Verificación y Validación se realizan y cómo se ejecutan.

2. Estudiar a profundidad los procesos de Verificación y Validación y las principales definiciones dadas por diferentes autores.
3. Identificar los principales estándares y modelos que servirán de soporte a la propuesta de solución.
4. Estudiar métodos V&V que se aplican a lo largo del ciclo de vida de un software y que se puedan ajustar al entorno del proyecto.
5. Confeccionar sobre la base de las buenas prácticas de Verificación y Validación la propuesta de Plan V&V.
6. Validar el Plan V&V mediante el Criterio de Expertos.

En la investigación cualitativa se utilizaron métodos teóricos y empíricos. Entre los métodos teóricos se emplearon el analítico-sintético para sintetizar y obtener los elementos más importantes de la información contenida en todos los documentos analizados. Además se utilizó el análisis histórico-lógico para conocer la evolución y desarrollo de los estándares y modelos para la mejora de procesos, la Verificación y Validación como parte esencial de estos y las buenas prácticas en su utilización. Se usa también la modelación al tratar de reproducir simplificada y subjetivamente la parte de la realidad objetiva, proporcionando explicaciones sobre sus características como guía para propuestas teóricas que fueron utilizadas para elaborar el plan. Estos métodos ayudaron a desarrollar el contenido principal de la investigación. Los métodos empíricos permitieron efectuar el análisis preliminar de la información, así como verificar y comprobar las concepciones teóricas acerca de la problemática existente en el ámbito de los procesos de Verificación y Validación en el entorno del proyecto. Dentro de estos se utilizó la observación, para obtener un registro visual de las principales características del proceso de desarrollo del SIGEP y como primera identificación de las actividades V&V que se realizan en el mismo. Además se realizaron entrevistas individuales a diferentes miembros del proyecto que permitieron constatar muchos de los resultados obtenidos como parte de la observación e incorporar nuevos elementos que resultaron imprescindibles para conformar la propuesta de solución.

La investigación está estructurada en tres capítulos:

En el Capítulo I se tratarán los temas que proporcionan la fundamentación teórica de la investigación donde se enuncian definiciones relativas al ciclo de vida del software, calidad del software y aseguramiento de la calidad. También se abordan los contenidos relacionados directamente con los procesos de Verificación y Validación, además de estudiarse los modelos de ciclo de vida del software,

profundizando en el que establece el estándar ISO/IEC 12207 por el impacto que tiene el mismo para la propuesta de solución.

En el Capítulo II se describe la solución propuesta, donde partiendo de la caracterización general del proyecto SIGEP, se explican todos los aspectos concernientes a la creación del Plan de Verificación y Validación. Se detallan los elementos relacionados con su estructura, el papel que juegan actividades y tareas incluyendo los métodos y procedimientos que en estas se aplican.

En el Capítulo III se muestra el proceso de validación de la propuesta utilizando el Método Delphi, detallando la manera en que se realizó la selección de los expertos y elaboración del cuestionario aplicado. Se explica además la tabulación de los datos obtenidos de la aplicación de este cuestionario al Panel de Expertos.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

El presente capítulo tratará diferentes epígrafes que servirán como base de conocimiento para entender la propuesta de solución a la problemática que trata esta investigación. Al lector se le propone, para una mejor lectura y entendimiento de los contenidos, dividir el estudio del capítulo en tres partes. Inicialmente (primera parte) encontrará definiciones relativas al ciclo de vida del software, calidad del software y aseguramiento de la calidad. Esto sobre la base de lo planteado por varios autores, que son referenciados por la importancia que tienen sus estudios para este trabajo. Para una segunda parte de la lectura, los contenidos relacionados a los procesos de Verificación y Validación, detallando diferentes aspectos como son la V&V en la industria del software (CMMI¹ y la Planificación de la V&V), y los métodos de Verificación y Validación como son las revisiones y las pruebas al software. La última parte se centra en el estudio de los modelos de ciclo de vida, profundizando en el que establece el estándar ISO/IEC 12207² por el impacto que tiene el mismo para la propuesta de solución.

1.2. Ciclo de vida del software

Cuando se trabaja para construir un producto o un sistema, es importante seguir una serie de pasos predecibles –un mapa de carreteras que le ayude a obtener el resultado oportuno de calidad-. El mapa de carreteras a seguir es llamado proceso del software.(PRESSMAN 2002)

Un proceso define *quien* está haciendo *qué*, *cuándo*, y *cómo* alcanzar un determinado objetivo. En la Ingeniería de Software el objetivo es construir un producto software o mejorar uno existente. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad. Un proceso de desarrollo de software debería también ser capaz de evolucionar durante muchos años. Durante esta evolución debería limitar su alcance, en un momento del tiempo dado, a las realidades que permitan las tecnologías, herramientas, personas y patrones de la organización.(QUISPE-OTAZU 2007)

La relación indisoluble entre el proceso de desarrollo de software y la calidad del mismo se encuentra en una definición bien completa dada por Roger S. Pressman en su libro *Ingeniería del Software. Un*

¹ Siglas en inglés del Modelo Integrado de Madurez de la Capacidad

² Norma Internacional ISO/IEC 12207 Procesos del Ciclo de Vida del Software

enfoque Práctico: “definimos un proceso de software como un marco de trabajo de las tareas que se requieren para construir software de alta calidad”.(PRESSMAN 2002)

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software(ZAVALA-RUIZ 2008).

Algunas definiciones sobre el ciclo de vida de un software plantean lo siguiente:

Es el período de tiempo que comienza cuando un producto de software es concebido y termina cuando ya no está disponible para su uso. El ciclo de vida del software incluye las fases de Conceptualización, Requerimientos, Diseño, Implementación, Prueba, Instalación y Chequeo, Operación y Mantenimiento, y en ocasiones la fase de la retirada del software. (IEEE 1990)

Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso.(IEEE/EIA 1997)

El ciclo de vida del software consta de cuatro fases: inicio, elaboración, construcción y transición, con cada fase subdividida en iteraciones. Cada ciclo concluye con una versión de producto para los clientes.(BOOCH *et al.* 2000)

Booch, Jacobson y Rumbaugh, los padres de la metodología RUP³, plantean que la vida de un producto consta de ciclos desde su nacimiento hasta su muerte.(BOOCH *et al.* 2000)

Estas referencias son ejemplo de las diferentes concepciones y puntos de vista en cuanto al ciclo de vida de un software, pero todas transmiten la idea de que solo es posible producir software de calidad de manera organizada y planificada. La definición de un ciclo de vida facilita el control sobre los tiempos en que es necesario aplicar recursos de todo tipo (personal, equipos, suministros, etcétera.) al proyecto.(GARZON 2007).

Esta definición es irrelevante si se abandonan los esfuerzos para cumplir con la premisa de construir un producto que satisface al cliente, lo cual solo se puede lograr si se comprende qué significa alcanzar esa gran meta: la calidad del software.

³ Siglas en inglés de Proceso Unificado de Desarrollo

1.3. Calidad del software

La IEEE⁴ en su estándar 610.12⁵ del año 1990 define calidad como el grado con el que un sistema, componente, o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. (IEEE 1990)

Pressman la define como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente. (PRESSMAN 2002)

Hay dos definiciones que aunque apuntan en direcciones diferentes son igualmente válidas para el desarrollo de esta investigación (SÁNCHEZ 2006):

- En el enfoque del cliente, calidad del software es el grado en que un cliente y/o usuario percibe que el producto software satisface sus necesidades.
- Enfocado en la condición industrial del producto, calidad del software es la habilidad de un producto software de satisfacer su especificación de requerimientos.

La calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de cero fallas; un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad.

Existen dos enfoques principales: la calidad del proceso y la calidad del producto.

Medir la calidad del producto de software es un aspecto primordial, pero: ¿el producto software puede ser de calidad si el proceso no tiene calidad?

El gran problema radica en que no se puede asegurar. Por eso es que se habla de aseguramiento de la calidad y no simplemente de calidad del software, pues esta última en ambientes no controlados solo podría darse en casos fortuitos. (SÁNCHEZ 2006)

La calidad del producto software depende de tareas realizadas durante todo el proceso: detectar errores en forma temprana ahorra esfuerzos, tiempo y recursos.

⁴Siglas en inglés del Instituto de Ingenieros Eléctricos y Electrónicos

⁵ Norma IEEE 610.12 Glosario de Terminologías de Ingeniería de Software

Existe otro enfoque no menos importante, sobre todo cuando el escenario de los nuevos productos y de los ciclos de producción muestra un patrón común: complejidad creciente y evolución constante de la tecnología, la *calidad a nivel de servicio*.

Este enfoque está muy relacionado con las corrientes actuales del mercadeo y con los servicios vía web que ofertan muchas empresas.

Generalmente si los consumidores están satisfechos con el producto o servicio, lo comprarán y usarán probablemente en mayor cantidad y comentarán a otros de su favorable experiencia con dicho producto o servicio. Si están insatisfechos, probablemente lo cambiarán y se quejarán a los fabricantes, a los vendedores u otros consumidores, lo que podría ser perjudicial para la empresa en términos económicos de imagen y publicidad.

Para la investigación, y los objetivos de este estudio teórico, todas las definiciones referenciadas anteriormente sobre la calidad del software tienen significación y como tal serán usadas.

La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad y es esencial para el desarrollo exitoso de un software que esta se asegure y controle.

1.4. El Aseguramiento de la Calidad

La Garantía de Calidad del Software (SQA, Software Quality Assurance, GCS, Gestión de la Calidad de Software) es una actividad de protección que se aplica a cada paso del proceso del software. Comprende procedimientos para la aplicación efectiva de métodos y herramientas, revisiones técnicas formales, técnicas y estrategias de pruebas, procedimientos de garantía de ajuste a los estándares y mecanismos de medida e información. El aseguramiento de la calidad del software, no es más que un patrón de acciones planificado y sistemático que se requiere para asegurar la calidad del software. (PRESSMAN 2002)

La IEEE lo define como un patrón planificado y sistemático de todas las acciones necesarias para proveer confianza adecuada de que un componente o producto cumple con los requerimientos técnicos establecidos.(IEEE 1998a)

El aseguramiento de la calidad es la aplicación de actividades planificadas y sistemáticas relativas a la calidad, para asegurar que el proyecto emplee todos los procesos necesarios para cumplir con los requisitos. (PMBOK 2005)

Plantea Pressman que el objetivo de SQA es proporcionar personas y gestión con el objetivo de que los elementos de trabajo cumplan los procesos. Esto se consigue mediante diferentes actividades. (1)

Evaluar objetivamente la ejecución de los procesos, los elementos de trabajo y servicios contra las descripciones de procesos, estándares y procedimientos. (2) Identificar y documentar los elementos no conformes. (3) Proporcionar información a las personas que están usando los procesos y a los gestores, de los resultados de las actividades del aseguramiento de la calidad. (4) Asegurar que los elementos no conformes son arreglados.(PRESSMAN 2002)

Aunque los diferentes autores referenciados tienen puntos en común a la hora de tratar el aseguramiento de la calidad, para esta investigación se tomará con mayor énfasis la definición dada por PMBOK⁶.

Para asegurar la calidad del software se realizan un grupo de actividades, de estas, las que se enfocan en garantizar que son realizadas las funciones correctas se agrupan como actividades de Validación, las que van dirigidas a garantizar el funcionamiento correcto o fiable de estas funciones se agrupan como actividades de Verificación.(TIAN 2005)

1.5. Verificación y Validación

Cualquiera que sea el ciclo de vida, metodología y herramientas aplicadas, se requiere de condiciones mínimas de gestión al proyecto y a los procesos, por ello toma fuerza la generación de estrategias que permitan apoyar los procesos de Verificación y Validación en cada una de las etapas del desarrollo de software.(CARRILLO and RENDÓN 2007)

1.5.1. Definiciones

La norma 610.12-1990 entre sus terminologías para la Ingeniería de Software trata la V&V como: El proceso de determinar si los requisitos para un sistema o componente son completos y correctos, los productos de cada fase de desarrollo cumplen con los requisitos o condiciones impuestas por la fase anterior, y el sistema o componente final cumple con los requisitos especificados. (IEEE 1990)

(PRESSMAN 2002) citando a Barry Bohem plantea dos interrogantes que identifican claramente a la Verificación y a la Validación:

Verificación: ¿Estamos construyendo el producto correctamente?

Validación: ¿Estamos construyendo el producto correcto?

⁶ Siglas en inglés de Fundamentos de la Gestión de Proyectos

De la misma forma define que la Verificación se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica y la Validación se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente.

1.5.2. Objetivos del proceso de Verificación y Validación

Gabriela Bermúdez y Marcelo Corona en su artículo *Diseño de un estándar de Verificación y Validación para asegurar la calidad del software*, exponen dos objetivos principales:

1. Verificar que los productos obtenidos en cada fase del ciclo de vida:
 - Cumplan con los requerimientos de la fase anterior.
 - Satisfagan los estándares, prácticas y convenciones de la fase actual.
 - Establezcan las bases apropiadas para iniciar la siguiente fase del ciclo de vida.
2. Validar que el producto final cumpla con los requerimientos del software establecidos. (BERMUDEZ and CORONA 2002)

La (IEEE 1998b) también define los objetivos de este proceso, los cuales guardan relación con los primeros expuestos:

1. Facilitar la temprana detección y corrección de errores en el software.
2. Mejorar la gestión de los riesgos del proceso y el producto.
3. Apoyo a los procesos del ciclo de vida del software para asegurar el cumplimiento con la ejecución del programa, cronograma, y presupuesto.

En síntesis y de manera muy concreta la norma IEEE 1012-1998⁷ indica que: El propósito de V&V es ayudar a la organización del desarrollo de software, para construir con calidad durante el ciclo de vida de software. Esta definición es de mucha relevancia para la investigación actual.(IEEE 1998b)

Para darle cumplimiento a los objetivos V&V dentro de una organización determinada es de vital importancia tener en cuenta el nivel y tipo de independencia utilizados para desarrollar estos procesos.

1.5.3. V&V Independiente

La independencia, según lo trata el Estándar para la Verificación y Validación de Software, viene definida por tres parámetros(IEEE 1998b):

⁷ Estándar para la Verificación y Validación de Software

– Independencia Técnica

Requiere de los esfuerzos de personal que no está involucrado en el desarrollo del software. Es un importante método para detectar sutiles errores pasados por alto por aquellos demasiado cerca de la solución.

– Independencia de Gestión

La responsabilidad de los esfuerzos V&V recaen en una organización separada del desarrollo y del programa de gestión de la que desarrolla. Se debe permitir que se presente sin ningún tipo de restricciones (por ejemplo, sin necesidad de previa aprobación de los grupos de desarrollo) o presiones adversas indebidas, ya directas o indirectas, de los grupos de desarrollo.

– Independencia Financiera

El control del presupuesto de las actividades lo lleva una organización independiente al desarrollo. Esta independencia impide situaciones en que no se pueden completar los esfuerzos V&V o entregar resultados en el momento oportuno porque los fondos se han desviado o adversas presiones financieras se han ejercido.

Tipos de Independencia

La IV&V⁸ se clasifica en cuatro tipos, en función del rigor con el que se realiza: clásica, modificada, interna y doméstica. (IEEE 1998b)

- *Clásica*: Contiene los tres parámetros de independencia, la responsabilidad recae en una organización independiente a la organización que desarrolla. Utiliza una estrecha relación de trabajo con la entidad que desarrolla para garantizar que las recomendaciones son integradas rápidamente dentro del proceso de desarrollo.
- *Modificada*: El desarrollo y la V&V lo realizan organizaciones diferentes, pero la responsabilidad de la gestión en el proyecto es única, y es la que recibe la información de ambas partes. No obstante, los presupuestos y el personal técnico están separados.
- *Interna*: Se emplea cuando el equipo de V&V pertenece a la misma organización desarrolladora, pero en la forma de una entidad diferenciada del grupo de desarrollo del proyecto.

⁸ Siglas en inglés de la Verificación y Validación Independiente

- *Doméstica*: Se emplea personal de la organización desarrolladora para realizar la V&V. El personal de desarrollo y de Verificación y Validación trabaja conjuntamente. No se puede garantizar la independencia técnica, y la gestión y el presupuesto son únicos para el desarrollo y V&V.

Cualquiera que sea el tipo de independencia que posea una organización determinada, los objetivos V&V a alcanzar constituyen la base principal de los esfuerzos a realizar, y en función de estos, la aplicación correcta de métodos de Verificación y Validación es un factor imprescindible.

1.6. Métodos V&V

Para dar cumplimiento a los objetivos de la Verificación y la Validación se utilizan los métodos o técnicas de control, que se dividen en dos grandes grupos: los controles estáticos y los controles dinámicos.

Los métodos estáticos tienen que ver con el análisis y control de las representaciones del sistema, es decir de los diferentes modelos construidos durante el proceso de desarrollo de software, tales como documentos de requerimientos, diagramas de análisis y diseño, y código fuente. En esta categoría se encuentran las revisiones técnicas formales o inspecciones de programas, la Verificación formal (basada en el uso de notaciones formales con base matemática) y herramientas de análisis estático, muchas de ellas provistas hoy en día como parte del compilador, como por ejemplo la detección de variables no utilizadas, código inalcanzable, etcétera.

Los métodos dinámicos, también conocidos como pruebas, se basan en ejercitar una implementación. Por lo tanto, sólo pueden ser aplicados si existe una versión operativa o ejecutable del producto.

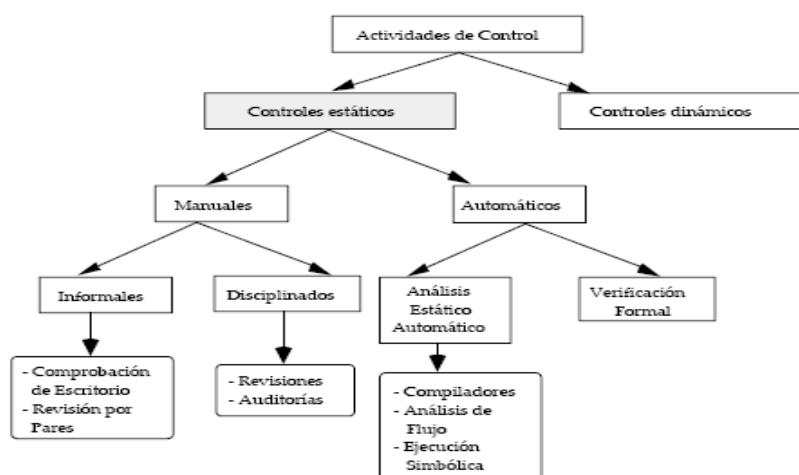


Figura 2. Actividades de control. Fuente: (ANTONIO 1999)

1.6.1. Controles estáticos manuales informales

Estas actividades las realizan los propios autores de los objetos a comprobar, o personas de su misma categoría y ocupación.

La comprobación de escritorio (desk checking por su terminología en inglés) consiste en examinar a mano e individualmente el objeto que se acaba de desarrollar. Es el método tradicional para analizar un programa. Se debe aplicar a los requisitos, especificaciones de diseño y código según se van desarrollando. Debe ser cuidadoso y concienzudo para que sea efectivo. Es más efectivo si se hace intercambiando el objeto a examinar con otro compañero. (ANTONIO 1999)

La revisión por pares o iguales (peer review)

Desde el punto de vista científico, peer review es la evaluación científica en la búsqueda o propósitos para la competencia y originalidad por expertos calificados que buscan crear productos de la misma forma.(MARTÍNEZ 2004)

Consiste en la revisión del código de un programador por otros programadores (sus pares). Se puede poner en práctica creando un panel que se encarga de revisar periódicamente muestras de código.(ANTONIO 1999)

Este método V&V se continuará citando por su terminología en inglés debido a que mundialmente se utiliza de esa manera. Para la investigación no solo es válido para aplicar sobre el código a través del trabajo entre los programadores, sino que puede ser usado para labores relacionadas con cualquier tipo de artefacto cuyas características requieran de la participación de más de una persona, cualquiera que sea su rol dentro del proyecto.

1.6.2. Controles estáticos manuales disciplinados

La misión principal de este tipo de control es que el control de la calidad no recaiga solo sobre el propio desarrollador.

Auditorías

La norma IEEE 1028-1997⁹ plantea que la auditoría es el examen independiente de un producto de trabajo o grupo de productos de trabajo para evaluar el cumplimiento con especificaciones, estándares, acuerdos contractuales u otro criterio.

⁹ Estándar para Revisiones al Software

Se pueden considerar tres tipos de auditorías:

- *Auditoría del producto*: El objetivo es cuantificar el grado de conformidad del producto con las características requeridas. Las auditorías del producto software más comunes son la Auditoría Funcional y la Auditoría Física.
- *Auditoría del proceso*: El objetivo es evaluar el proceso de desarrollo o de gestión en cuanto a completitud y efectividad, determinando dónde se puede mejorar. En el desarrollo de software se suelen realizar dos tipos de auditorías del proceso:
 - ✓ Auditorías de proyecto: evaluar la productividad y eficacia del equipo que trabaja en un proyecto, así como la efectividad de los métodos y herramientas utilizados.
 - ✓ Auditorías de gestión de proyecto: evaluar la efectividad de las prácticas de gestión realizadas y la organización del proyecto.
- *Auditoría del sistema de calidad*: El objetivo es evaluar la completitud y efectividad del propio sistema de calidad establecido. (ANTONIO 1999)

Revisiones técnicas

Una revisión es un proceso o una reunión durante la cual un producto de software es presentado al personal de proyectos, gerentes, usuarios, clientes, representantes de los usuarios, u otras partes interesadas para que formulen sus observaciones o lo aprueben.

Una revisión técnica es una evaluación sistemática de un producto de software por un equipo de personal calificado, que examina la idoneidad del producto de software para su uso e identifica discrepancias con las especificaciones y estándares. Las revisiones técnicas también pueden proporcionar recomendaciones de alternativas y el examen de las distintas alternativas. (IEEE 1997)

Uno de los objetivos fundamentales de las revisiones técnicas es ofrecer a los gestores información fiable acerca de los aspectos técnicos del proceso de desarrollo de software, de la misma forma que les llega información fiable acerca de los costes y la programación del trabajo, para que puedan tomar decisiones adecuadas para dirigir con éxito el proyecto.

Con las revisiones se consigue que el peso de la evaluación técnica no recaiga sobre las mismas personas involucradas en la producción del software, que por la posición que ocupan no pueden ser totalmente objetivas, sino en otras personas técnicamente competentes y objetivas. (ANTONIO 1999)

Inspecciones

Una inspección es un examen visual de un producto de software para detectar e identificar anomalías, incluyendo errores y desviaciones de los estándares y especificaciones. Las inspecciones son exámenes por pares dirigidos por facilitadores imparciales que están capacitados en técnicas de inspección. La determinación de una acción de reparación o investigación para una anomalía es un elemento obligatorio de una inspección de software, aunque la solución no debe ser determinada en la reunión de inspección. (IEEE 1997)

Walk-through

El walk-through es una técnica de análisis estático en el cual el diseñador o programador guía a miembros del equipo de desarrollo y otras partes interesadas a través del producto de software, los participantes realizan preguntas y comentarios acerca de posibles errores, violación de los estándares de desarrollo y otros problemas. (IEEE 1997)

Las diferencias entre las inspecciones y los walk-through se basan esencialmente en los siguientes aspectos:

- Los walk-through están planteados como una medida de ayuda al desarrollador, mientras que las inspecciones están planteadas como una medida de ayuda al gestor.
- En los walk-through el objetivo fundamental es incrementar el entendimiento, comprender mejor el objeto, mientras que en las inspecciones el objetivo es detectar defectos.
- En las inspecciones el proceso está guiado por la lista de comprobación, y en los walk-through está guiado por la estructura del producto revisado.
- Las inspecciones se planifican y procesan de una manera mucho más formal que los walk-through. Se usan para asegurar la satisfacción de los criterios de salida establecidos entre diferentes etapas del desarrollo (revisiones de fase).

1.6.3. Pruebas

La calidad de un sistema software es algo subjetivo que depende del contexto y del objeto que se pretenda conseguir. Para determinar dicho nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

Al decir de Pressman: las pruebas de software son un elemento crítico para la garantía de la calidad del software, y representa una revisión final de las especificaciones, del diseño y de la codificación.(PRESSMAN 2002)

En la cadena de valor del desarrollo de un software específico, el proceso de prueba es clave a la hora de detectar errores o fallas. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan a la calidad de un producto bien desarrollado. Las aplicaciones de software han crecido en complejidad y tamaño, y por consiguiente también en costos. Hoy en día es crucial verificar y evaluar la calidad de lo construido de modo de minimizar el costo de su reparación. Mientras antes se detecte una falla, más barato es su corrección.(FOUNDATION 2008)

Objetivos de las pruebas

Citando a Glen Myers, (PRESSMAN 2002) plantea una serie de normas que al decir del propio autor, pueden servir acertadamente como objetivos de las pruebas:

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

Métodos de pruebas

El software debe probarse desde dos perspectivas diferentes: (1) la lógica interna del programa se comprueba utilizando técnicas de diseño de casos de prueba de *caja blanca*. (2) Los requisitos del software se comprueban usando técnicas de diseño de casos de prueba de *caja negra*. En ambos casos, se intenta encontrar el mayor número de errores con la menor cantidad de esfuerzo y tiempo.(PRESSMAN 2002)

Prueba de caja blanca

Las pruebas de caja blanca (también llamada caja de cristal) ocupan un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para realizar sus casos de prueba.(LEON 2006)

Según Pressman, mediante las pruebas de caja blanca se pueden obtener casos de prueba que:

- Garantizan que se ejecuten al menos una vez todos los caminos independientes de cada módulo.

- Ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.(PRESSMAN 2002)

Prueba de caja negra

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.(MAÑAS 1994)

Varios autores caracterizan este método de pruebas en tres aspectos fundamentales:

- Los métodos de pruebas de caja negra se centran en los requisitos funcionales del software.
- Con estas pruebas se obtienen conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.
- Las pruebas de caja negra no son una alternativa a las de caja blanca, sino que son un complemento para tratar de descubrir otros errores.

Niveles de pruebas

Alfredo Weitzenfeld en su libro *Ingeniería de software orientada a objetos con Java e Internet* argumenta que existen tres niveles principales para aplicar las diferentes técnicas de pruebas (WEITZENFELD 2006):

- *Prueba de unidad*: Mediante esta prueba solo una unidad es probada como tal, como una clase, un paquete de servicio o un subsistema.
- *Prueba de integración*: En ella se verifica que las unidades trabajen juntas correctamente. Ambas pueden ser realizadas mediante casos de uso de pruebas, los cuales pueden ser aplicados a clases, paquetes de servicio, subsistema o el sistema completo.
- *Prueba de sistema*: Verifica el sistema completo o su aplicación como tal. Se toma el punto de vista del usuario final y los casos de uso de prueba ejecutan acciones típicas del usuario.

El mismo autor explica una serie de técnicas de pruebas:

- *Prueba de regresión*: Tiene el propósito de verificar el sistema luego de haberle introducido cambios, por ejemplo después de corregir algún error, de manera que se mantenga la funcionalidad especificada originalmente.
- *Prueba de operación*: Su objetivo es verificar el sistema en operación por un largo período bajo condiciones normales de uso.
- *Prueba de escala completa*: Trata de verificar el sistema en su carga máxima mediante la asignación de los parámetros a su valor límite y la interconexión del sistema con un máximo de equipos y usuarios simultáneos. Su máxima expresión es la *prueba de estrés*, que significa que se prueba el sistema en los límites extremos para determinar su tolerancia si ocurre algún tipo de falla.
- *Prueba de rendimiento o de capacidad*: Tiene como propósito medir la capacidad de procesamiento del sistema bajo diferentes cargas, incluyendo espacio de almacenamiento y utilización de la unidad de procesamiento.
- *Prueba de sobrecarga*: Pretende observar cómo se comporta el sistema cuando se le aplica una sobrecarga, más allá de las pruebas de escala completa y rendimiento.
- *Prueba basada en requisitos o prueba de casos de uso*: Intenta llevar a cabo pruebas basadas directamente en la especificación de requisitos. Se trata de verificar que el sistema final cumple con las especificaciones funcionales descritas para los caso de uso originales.
- *Pruebas ergonómicas*: Tienen como propósito probar los aspectos ergonómicos del sistema, en otras palabras, las interfaces hombre-máquina en el caso de que estas existan.
- *Prueba de documentación del usuario*: Tiene como propósito probar la documentación de usuario, incluyendo el manual de este y la documentación de mantenimiento y servicio. Se prueba que los manuales y el comportamiento del sistema sean congruentes entre sí, que sean legibles, con una buena redacción y, en general, que sean comprensibles.
- *Prueba de aceptación de validación*: Pretende lograr una revisión final por parte de la organización que solicitó el sistema, lo cual, a menudo significa validación del sistema. El sistema se prueba en su ambiente real por un período extenso. Cuando se termina la prueba se toma la decisión de aceptar o no el producto. Este tipo de pruebas es a veces conocida como

prueba alfa. A menudo se hace una *prueba beta*, lo cual implica que el producto es probado por clientes seleccionados que utilizan el sistema y reportan fallas encontradas.

Modelos de Pruebas

El proceso de pruebas también debe adaptarse a las necesidades del proyecto y en cada caso se pueden emplear de diversa forma las diferentes técnicas de pruebas.

La definición del proceso de pruebas para un proyecto en concreto se recoge en el Plan de Pruebas y consta esencialmente de las siguientes etapas(KYNETIA 2007):

- Análisis de pruebas
- Diseño de las pruebas
- Ejecución de las pruebas
- Evaluación de resultados e introducción de mejoras

Los procesos de pruebas que se aplican por lo general se basan (ya sea adaptando o combinando) en los diferentes modelos de pruebas que existen, los cuales han ido evolucionando desde los que ubicaban la etapa de pruebas en una sola fase del desarrollo (casi siempre aplicando las pruebas al producto ya terminado) hasta los que verifican todo el proceso de desarrollo en sus diferentes fases.

A continuación se muestran tres modelos que guardan estrecha relación con el proceso de V&V.

El *Modelo en V* es el modelo de pruebas más comúnmente extendido, por su simplicidad y fácil entendimiento. La V indica también Verificación y Validación.

Por el orden de las actividades en secuencia de tiempo y con la abstracción de niveles, la conexión entre desarrollo y actividades de pruebas se vuelve más clara.

A diferencia de los modelos clásicos, extiende las pruebas a lo largo de todo el ciclo de vida del software.

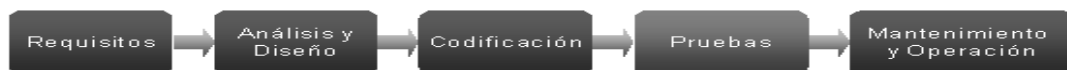


Figura 3. Modelo de pruebas clásico. Fuente: (KYNETIA 2007)

Este modelo deriva directamente de la aplicación de pruebas de Verificación y Validación a un ciclo de vida de desarrollo en cascada.

Es fácilmente extensible a modelos de desarrollo iterativos si se considera la V como una iteración del proyecto completo.

En consecuencia, un plan de pruebas, inmerso en el proceso de desarrollo iterativo, irá evolucionando conforme el proyecto va transcurriendo por sus sucesivas iteraciones.



Figura 4. Modelo de pruebas en V. Fuente: (KYNETIA 2007)

El Modelo en W surge como refinamiento del *Modelo en V*. Refleja mejor la interdependencia que existe entre equipo de desarrollo y el equipo de pruebas a lo largo de todo el proceso de desarrollo del sistema, destacando los siguientes dos puntos:

- En las primeras etapas se consideran labores de prueba que no aparecen reflejadas en el modelo original como son la revisión de los requisitos, la revisión de la especificación del sistema, la revisión de la arquitectura y del diseño, y las revisiones de código.
- En las etapas finales también se distingue del modelo original en V porque se desglosan las tareas de pruebas propiamente dichas y las labores de depuración y corrección de los errores detectados, que serán llevadas a cabo por desarrolladores. (KYNETIA 2007)

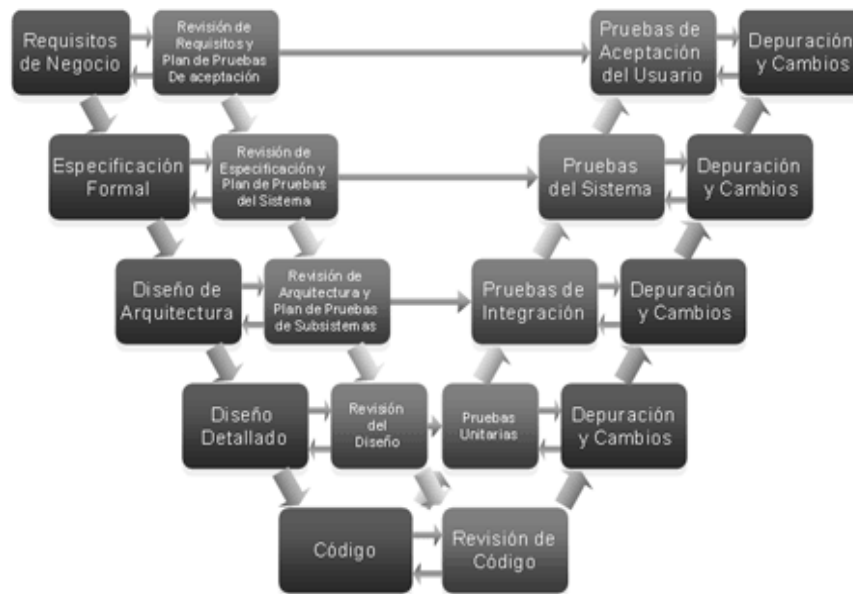


Figura 5. Modelo de pruebas en W. Fuente: (KYNETIA 2007)

Por su parte la *disciplina de pruebas de RUP* plantea que los objetivos de la prueba son:

- Planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de integración y las pruebas de sistema. Las pruebas de integración son necesarias para cada construcción dentro de la iteración, mientras que las pruebas de sistema son necesarias solo al final de la iteración.
- Diseñar e implementar las pruebas creando los casos de prueba que especifican qué probar, creando los procedimientos de prueba ejecutables para automatizar las pruebas.
- Realizar las diferentes pruebas y manejar los resultados de cada prueba sistemáticamente. Las construcciones en las que se detectan son probadas de nuevo y posiblemente devueltas a otro flujo de trabajo, como diseño o implementación, de forma que los defectos importantes puedan ser arreglados.(BOOCH et al. 2000)

Además define un flujo de trabajo para las pruebas centrado en actividades como: planificar la prueba, diseñar prueba, implementar prueba, realizar pruebas de integración, realizar pruebas de sistema y evaluar prueba, que están asignadas a roles bien definidos en la metodología.

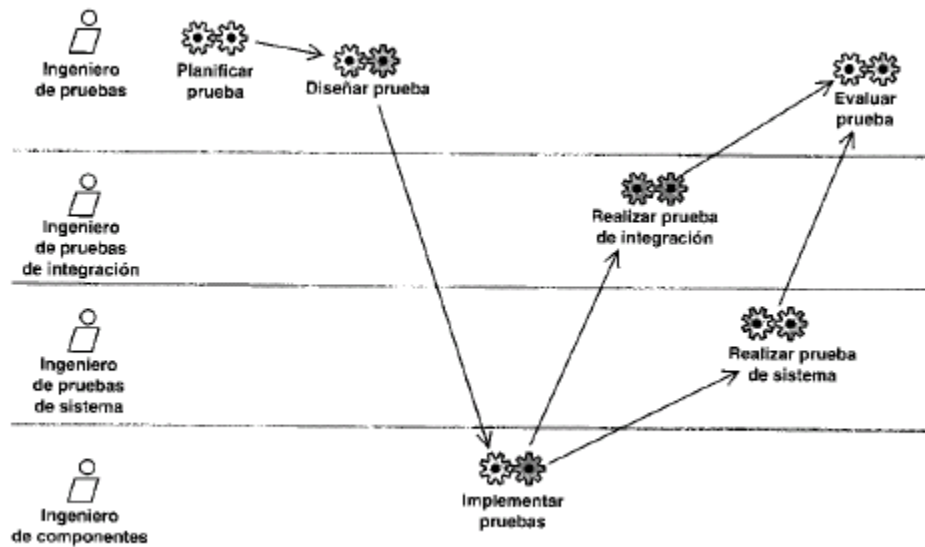


Figura 6. Flujo de trabajo de pruebas de RUP. Fuente: (BOOCH et al. 2000)

La introducción de buenas prácticas de Verificación y Validación con la aplicación de métodos estáticos y dinámicos, han convertido a estos procesos en focos de atención para las empresas que buscan un lugar cimero en el competitivo entorno de la industria de software.

1.7. V&V en la industria del software

Los requerimientos del mercado internacional y los proyectos a escala mundial, han hecho reflexionar a muchos sobre la necesidad de iniciar programas de mejora de procesos y perfeccionar su competitividad y desempeño actual. Para esto se apoyan en modelos y estándares internacionales de gran prestigio que pueden ser adoptados e implementados acorde a las características de cada organización. En estos modelos y estándares, las buenas prácticas y la planificación de los procesos de Verificación y Validación, tienen un lugar importante por su implicación directa en la mejora de los procesos de desarrollo de software.

1.7.1. V&V en CMMI

CMMI es un modelo de mejora del proceso de madurez para el desarrollo de productos y servicios. Se compone de las mejores prácticas que se ocupan de actividades de desarrollo y mantenimiento que cubren el ciclo de vida del producto desde su concepción hasta su entrega y mantenimiento. (INSTITUTE 2006)

Provee orientación para diseñar procesos efectivos (tiempo y coste), en distintos dominios (desarrollo de productos y servicios, adquisiciones y mantenimiento), dentro del ámbito de una organización, cuya

principal premisa es: La calidad de un producto es determinada en gran medida por la calidad del proceso utilizado para desarrollarlo y mantenerlo.(ALS 2007)

CMMI tiene 22 áreas de procesos clasificadas en cuatro categorías: Ingeniería, Gestión de Proyecto, Gestión de Procesos, Soporte, las cuales componen las dos representaciones que el modelo propone: la representación escalonada (*madurez*) donde se definen un conjunto de áreas por nivel de madurez y la representación continua (*capacidad*) en la cual se seleccionan las áreas a mejorar.

Entre estas áreas de proceso se encuentran la *Verificación y Validación*.

La Validación demuestra que el producto cumplirá con el uso que tiene definido, mientras que la Verificación está encaminada a que los productos de trabajo reflejen los requisitos especificados. En otras palabras, la Verificación asegura que se está construyendo correctamente y la Validación asegura que se está construyendo lo correcto. Las actividades de Validación usan enfoques similares a las de Verificación (pruebas, análisis, inspección, etcétera.). Frecuentemente los usuarios finales y otros interesados directos se involucran en las actividades de Validación. Ambas, Verificación y Validación con frecuencia fluyen concurrentemente y pueden usar porciones del mismo entorno.(INSTITUTE 2006)

Por la densidad de la explicación sobre cada una de estas áreas que el modelo propone solo se mostrarán algunos de los aspectos esenciales que trata (INSTITUTE 2006):

Validación

El propósito de la Validación es demostrar que un producto o componente de un producto cumple con su uso previsto cuando es puesto en práctica en el entorno previsto.

El modelo incluye un conjunto de *prácticas específicas* para esta área de proceso:

- La práctica específica de *Seleccionar los productos para la validación* permite la identificación del producto o componente de producto para ser validado y los métodos que serán usados para llevar a cabo la Validación.
- La práctica específica de *Establecer el entorno de validación* permite la determinación del ambiente que será usado para llevar a cabo la Validación.
- La práctica específica de *Establecer los procedimientos y criterios de validación* permite el desarrollo de los procedimientos y criterios de validación que están alineados con las características de los productos seleccionados, con las restricciones del cliente en la validación, con los métodos y el entorno de validación.

- La práctica específica de *Ejecutar la Validación* permite la ejecución de la Validación de acuerdo con los métodos, procedimientos y criterios.

Verificación

El propósito de la Verificación es asegurar que el producto de trabajo seleccionado o el producto realizado es compatible o responde a sus requisitos específicos.

La Verificación es sin dudas un proceso incremental, porque está presente en todo el desarrollo del producto y productos de trabajo.(INSTITUTE 2006)

Las *prácticas específicas* de esta área de proceso son realizadas cada una de la siguiente manera:

- La *Selección del Producto de Trabajo para la Verificación*, permite la identificación del producto de trabajo para ser verificado, los métodos que se usan para realizar la Verificación y los requerimientos que deben ser satisfechos (o que debe cumplir) por cada producto de trabajo seleccionado.
- El *Establecimiento del Entorno de Verificación* permite la determinación del ambiente que se usará para llevar a cabo la Verificación.
- El *Establecimiento de los Procedimientos y Criterios de Verificación* permite el desarrollo de los procedimientos y criterios de verificación en línea con los productos de trabajo seleccionados, requerimientos, métodos, y las características del entorno de verificación.
- La *Ejecución de la Verificación* conduce la Verificación acorde a los métodos permitidos, los procedimientos y criterios.

Ambas áreas de proceso se encuentran en el nivel 3 de madurez según la representación escalonada del modelo, algo que demuestra la importancia que le conceden dentro de la mejora de los procesos de desarrollo.

En el listado de evaluaciones de empresas que publica el SEI¹⁰, con actualización de junio del 2008, aparecen más de 700 empresas e instituciones evaluadas con el nivel 3 de madurez (no se ha incluido en este estudio las de nivel 4 y 5), lo que por supuesto implica un desarrollo ostensible en la aplicación de la V&V. Algunos ejemplos son los siguientes(SEI 2008):

- AAI Services Corp Training & Simulation

¹⁰ Siglas en inglés del Instituto de Ingeniería de Software

- Argentina Delivery Center
- Communications & High Tech Italy, Greece, Emerging Markets, India
- Information Industry Business unit (IIB) and R&D Department of ACS
- AYRTON Technology Nantes
- Beijing Philisence Technology Co. Ltd

Es imposible cumplir con lo que establecen modelos como CMMI para la Verificación y la Validación, si no se tiene en cuenta que el éxito en la implementación de estos procesos depende en gran medida de una correcta planificación para el entorno en el cual serán aplicados.

1.7.2. Planificación de la V&V

Se define un plan como todo proyecto, esquema o preparación de obra o proceder. (OSSORIO 2004)

Los planes describen la forma en que un proyecto concreto será realizado: cómo, cuándo y qué coste tendrá. También se puede planificar tareas individuales. (HUMPHREY 2001)

El propósito de planificar la V&V es asegurar que cada recurso, rol y responsabilidad esté claramente asignado, el plan V&V que se obtiene de este proceso documenta y describe los diferentes recursos, sus roles y actividades, así como las técnicas y herramientas a ser utilizadas. La comprensión de los diferentes propósitos de cada actividad V&V ayudará en la planificación cuidadosa de las técnicas y recursos necesarios para el cumplimiento de estos. El plan también dirige la gestión, la documentación, políticas y procedimientos de las actividades V&V y su interacción, así como el reporte de defectos y requerimientos de documentación. (ABRAN and MOORE 2004)

La norma IEEE 1012-1998 es un estándar que abarca todos los procesos del ciclo de vida del software, incluyendo la Adquisición, Suministro, Desarrollo, Operación y Mantenimiento.

Planificar la Verificación y Validación de un sistema informático ajustándose a este estándar implica la comprensión obligatoria de todos los planteamientos y conceptos que en él se reflejan:

- Mediante el proceso de V&V del software se determina si los productos del desarrollo de una determinada actividad se ajustan a los requisitos de esa actividad, y si el software cumple con el uso previsto y las necesidades de los usuarios. La determinación incluye la evaluación, el análisis, revisión, inspección, y las pruebas a los procesos y productos del software. La V&V se realiza en paralelo con el desarrollo de software, no en la conclusión de este.

- Los resultados de V&V permiten a la organización de desarrollo de software modificar los productos de manera oportuna y reducir el costo total del proyecto y el impacto en el cronograma. Sin un enfoque dinámico, las anomalías y los asociados cambios en el software suelen ser retrasados para más adelante en el cronograma programado, lo que resulta en un mayor costo del programa y retrasos en el calendario.(IEEE 1998b)

Dos de los conceptos principales introducidos son: *Niveles de Integridad del Software* y *Tareas V&V mínimas para cada nivel de integridad*. El primero define cuatro niveles que describen la *criticidad* del software variando desde una alta integridad hasta el nivel más bajo. El segundo concepto describe el mínimo de tareas V&V requeridas para cada uno de los niveles de integridad.

La siguiente figura muestra el marco de trabajo de V&V definido por el estándar, con la jerarquía de Procesos V&V → Actividades V&V → Tareas V&V.

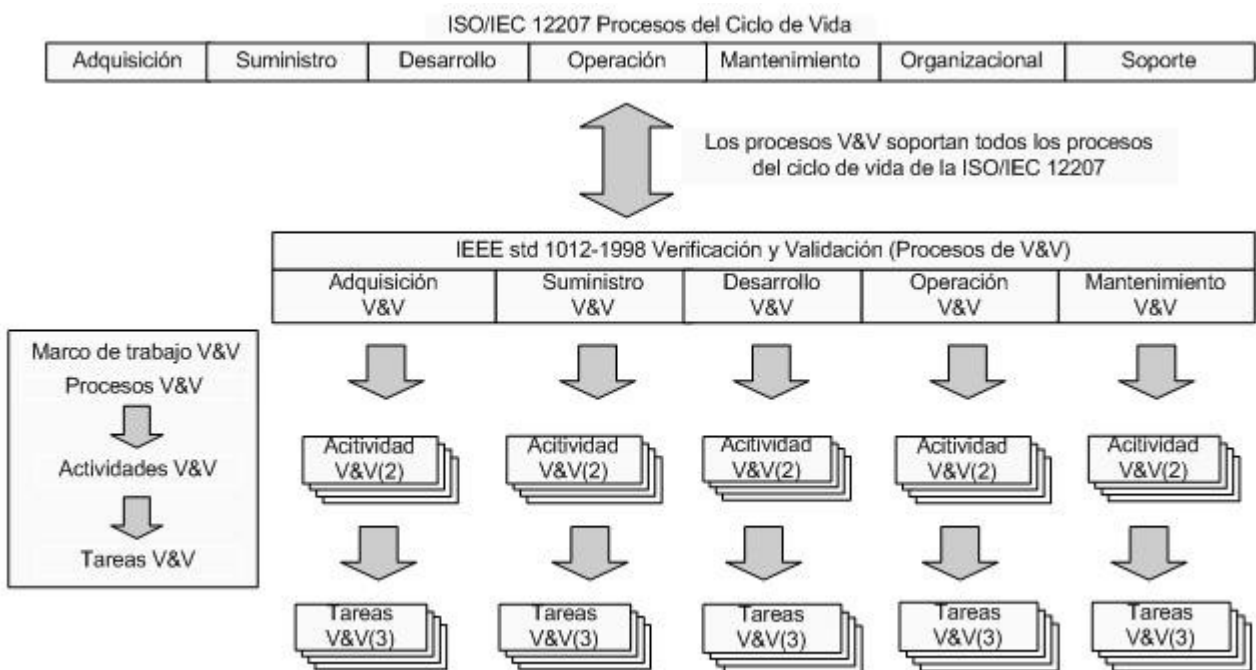


Figura 7. Marco de Trabajo V&V. Jerarquía de procesos, actividades y tareas. Fuente:(IEEE 1998b)

La IEEE 1012-1998 facilita a quien persigue el objetivo de planificar la Verificación y la Validación de un software:

- El establecimiento de una base teórica completa para la comprensión de todo lo que concierne a la V&V con conceptos claros y aplicables de ambos procesos.

- Un esquema de niveles de integridad que puede ser adoptado para definir a partir de la criticidad, el nivel de integridad de cualquier software.
- La definición de los niveles y tipos de IV&V.
- La definición de los contenidos requeridos para un Plan de Verificación y Validación de Software, soportado por procesos V&V, donde cada proceso está conformado por actividades V&V y estas a su vez por tareas V&V a ejecutar durante todo el ciclo de vida.
- La estructura para un plan de este tipo con la explicación detallada de cada una de sus secciones.

Teniendo en cuenta esto, y luego del análisis exhaustivo que permitió conocer no solo los aspectos reflejados anteriormente, sino otros de importante significación para este estudio, se decidió adoptar la IEEE 1012-1998 *Estándar para la Verificación y Validación de Software* para cumplir el objetivo general de la investigación, por lo que la propuesta de solución estará basada en el mencionado estándar.

El marco de trabajo que desarrolla la norma seleccionada requiere del estudio global de los modelos de ciclo de vida de software, profundizando en el establecido por el estándar ISO/IEC 12207 que define los procesos del ciclo de vida, sobre los cuales la IEEE 1012-1998 organiza los procesos V&V.

1.8. Modelos del ciclo de vida del software

Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software. Intenta determinar el orden de las etapas involucradas y los criterios de transición asociadas entre estas etapas.

A partir de las consultas realizadas a los principales autores de la materia, se puede decir que un modelo de ciclo de vida:

- Describe las fases principales de desarrollo de software.
- Define las fases primarias esperadas de ser ejecutadas durante esas fases.
- Ayuda a administrar el progreso del desarrollo.
- Provee un espacio de trabajo para la definición de un detallado proceso de desarrollo de software.

El estudio del tema en diferentes fuentes bibliográficas permite señalar que las diferencias entre modelos de ciclo de vida están divididas en tres grandes aspectos:

- El *alcance del ciclo de vida*, que depende de hasta dónde se quiere llegar con el proyecto: saber si es viable el desarrollo de un producto, el desarrollo completo o el desarrollo completo más las actualizaciones y el mantenimiento.
- La *cualidad y cantidad de las etapas* en que se dividirá el ciclo de vida: según el ciclo de vida que se adopte y el proyecto para el cual se está adoptando.
- La *estructura y sucesión de las etapas*: si hay comunicación entre ellas y si es factible repetirlas (iterar).

Entre los modelos más conocidos se pueden citar (CANTONE 2006):

- *Ciclo de vida lineal*: Es el más sencillo de todos los modelos. Consiste en descomponer la actividad global del proyecto en etapas separadas que son realizadas de manera lineal, es decir, cada etapa se realiza una sola vez, a continuación de la etapa anterior y antes de la etapa siguiente. Con un ciclo de vida lineal es muy fácil dividir las tareas, y prever los tiempos (sumando linealmente los de cada etapa).
- *Ciclo de vida en cascada puro*: Es un ciclo de vida que admite iteraciones, contrariamente a la creencia de que es secuencial como el lineal. Después de cada etapa se realiza una o varias iteraciones para comprobar si se puede pasar a la siguiente. Es un modelo rígido, poco flexible, y con muchas restricciones, aunque fue uno de los primeros y sirvió de base para el resto de los modelos de ciclo de vida.
- *Ciclo de vida con componentes*: Es un ciclo adecuado para los proyectos en los que se dispone de todos los requerimientos al comienzo, para el desarrollo de un producto con funcionalidades conocidas o para proyectos, que aun siendo muy complejos se entienden perfectamente desde el principio. Es un modelo puramente teórico, ya que el usuario rara vez mantiene los requerimientos iniciales y existen muchas posibilidades de que se deba volver a alguna etapa anterior.
- *Ciclo de vida iterativo*: Derivado del ciclo de vida en cascada puro, este modelo busca reducir el riesgo que surgen entre las necesidades del usuario y el producto final por malos entendimientos durante la etapa de requerimientos. Es la iteración de varios ciclos en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto.
- *Ciclo de vida por prototipos*: En la práctica los prototipos se utilizan para validar los requerimientos de los usuarios en cualquier ciclo de vida. En este modelo el objetivo es lograr

un producto intermedio, antes de realizar el producto final, para conocer mediante el prototipo como responderán las funcionalidades previstas para el producto final.

- *Ciclo de vida evolutivo*: Este modelo acepta que los requerimientos del usuario pueden cambiar en cualquier momento. La práctica demuestra que obtener todos los requerimientos al inicio es muy difícil, no solo por la dificultad del cliente de transmitir su idea, sino porque estos requerimientos evolucionan durante el desarrollo, y de esta manera surgen otros nuevos. Resulta un modelo muy útil cuando se desconocen la mayoría de los requerimientos iniciales o estos no están completos.
- *Ciclo de vida incremental*: Se basa en la filosofía de construir incrementando las funcionalidades del programa. Se realiza construyendo por módulos que realizan las distintas funciones del sistema. Esto permite ir aumentando gradualmente las capacidades del software. Se puede usar este modelo para casi cualquier proyecto, y especialmente cuando el usuario necesite entregas rápidas aunque sean parciales.
- *Ciclo de vida en espiral*: Puede considerarse una variación del prototipado. Se basa en una serie de ciclos repetitivos para ir ganando madurez en el producto final. La ventaja más notoria de este modelo es que se puede comenzar el proyecto con un alto grado de incertidumbre, así como el bajo riesgo de retraso en caso de detectar errores, porque se pueden solucionar en la próxima rama del espiral.
- *Ciclo de vida definido en RUP*: Es una implementación del desarrollo en espiral. Divide el proceso de desarrollo en ciclos y estos son la clave que tiene el modelo para crear productos de buena calidad. Organiza las tareas en fases e iteraciones. Pretende implementar las mejores prácticas en la Ingeniería del Software. Se caracteriza por ser iterativo e incremental, centrado en la arquitectura y guiado por casos de uso. Se ha convertido en uno de los ciclos de vida más usados en el mundo del software, precisamente por ser parte de una de las metodologías de desarrollo más utilizadas.
- *Ciclo de vida del estándar ISO/IEC 12207-1995*

1.8.1. Sobre estándar ISO/IEC 12207

Esta norma provee un marco de trabajo para el desarrollo y la gestión de software.(IEEE/EIA 1996)

El marco cubre el ciclo de vida del software desde la conceptualización hasta la retirada del software y consta de los procesos para adquirir y suministrar productos y servicios de software. Además proporciona el marco para controlar y mejorar estos procesos.(IEEE/EIA 1996)

Contiene procesos, actividades y tareas que se deben aplicar durante la adquisición de un sistema que contiene software, un producto independiente de software, y servicio de software, y durante el suministro, desarrollo, operación, y mantenimiento de los productos de software.(IEEE/EIA 1996)

El estándar describe la arquitectura de los procesos del ciclo de vida del software, pero no especifica los detalles de cómo implementar o ejecutar las actividades y tareas incluidas en los procesos.(IEEE/EIA 1996)

La norma organiza las actividades que se pueden realizar durante el ciclo de vida del software. Son agrupadas en: cinco procesos primarios, ocho procesos de soporte y cuatro procesos organizacionales. Además cada actividad se divide en un conjunto de tareas que le dan cumplimiento.

A continuación se detalla cada grupo de procesos:

Procesos primarios

- *Proceso de adquisición*: Define las actividades del adquiridor, la organización que adquiere un sistema, producto de software o servicio de software.
- *Proceso de suministro*: Define las actividades del proveedor, la organización que proporciona el sistema, producto de software o servicio de software para el adquirente.
- *Proceso de desarrollo*: Define las actividades del desarrollador, la organización que define y desarrolla el producto de software.
- *Proceso de operación*: Define las actividades del operador, la organización que presta el servicio de operación de un sistema informático en su entorno para sus usuarios.
- *Proceso de mantenimiento*: Define las actividades del mantenedor, la organización que presta el servicio de mantenimiento del producto de software, es decir, la gestión de las modificaciones al software para que esté actualizado y en buen estado operacional. Este proceso incluye la migración y retirada del producto de software.

Procesos de soporte

- *Proceso de documentación*: Define las actividades para recoger la información producida durante los procesos del ciclo de vida.
- *Proceso de Gestión de la Configuración*: Define las actividades de la gestión de configuración.
- *Proceso de Aseguramiento de la Calidad*: Define las actividades para asegurar que los productos de software y los procesos están en conformidad con sus requisitos especificados y se adhieren a sus planes establecidos. Las revisiones conjuntas, auditorías, Verificación y Validación, pueden utilizarse como técnicas de Aseguramiento de la Calidad.
- *Proceso de Verificación*: Define las actividades para la verificación de los productos y servicios de software.
- *Proceso de Validación*: Define las actividades para la validación del producto de software.
- *Proceso de revisión conjunta*: Define las actividades para evaluar el estado y productos de una actividad.
- *Proceso de auditoría*: Define las actividades para determinar el cumplimiento de los requisitos, planes y contrato.
- *Proceso de resolución de problemas*: Define un proceso para analizar y eliminar los problemas (incluyendo las no conformidades) cualquiera sea su origen.

Procesos organizacionales

- *Proceso de Gestión*: Define las actividades básicas de gestión, incluyendo la gestión del proyecto, relativas a la ejecución de los procesos del ciclo de vida.
- *Proceso de infraestructura*: Define las actividades básicas para el establecimiento de la estructura de un proceso del ciclo de vida.
- *Proceso de mejora*: Define las actividades básicas que se deben llevar a cabo para la creación, medición, control y mejora del ciclo de vida.
- *Proceso de entrenamiento*: Define las actividades para proveer un adecuado entrenamiento al personal.

1.9. Conclusiones parciales

Los procesos de Verificación y Validación son claves para asegurar la calidad de un producto informático, algo que se puede concluir del estudio efectuado sobre estos en el capítulo: definiciones, objetivos, su utilización en la industria del software. Para darle cumplimiento a sus objetivos deben ser planificados y ejecutados durante todo el ciclo de vida del software.

A partir de todo el estudio teórico realizado se determinó que la propuesta de Plan de Verificación y Validación para el proyecto SIGEP estará sustentada por la norma IEEE 1012-1998. De este se obtuvo un cúmulo significativo de información relacionada con la puesta en práctica de los procesos de Verificación y Validación a lo largo de todo el ciclo de vida del software, así como la planificación de los mismos. Esto permitió identificar nuevas actividades y tareas V&V a aplicar en el SIGEP y determinar los métodos que se pueden implementar en el proyecto para dar cumplimiento a cada tarea, analizando la importancia de utilizar tanto mecanismos de control estático, como dinámicos, y no solamente pruebas al software.

CAPÍTULO II. SOLUCIÓN PROPUESTA

2.1. Introducción

Con el objetivo de ayudar a la comprensión del entorno sobre el cual se trabaja en la investigación, se comienza el capítulo realizando una caracterización general del proyecto SIGEP. Seguidamente se explican los aspectos que se tuvieron en cuenta para adoptar la norma IEEE 1012-1998 a las peculiaridades del esquema productivo con el que se interactúa. Posteriormente, se expone todo lo que concierne directamente al Plan de Verificación y Validación, su estructura, actividades y tareas, además de los métodos y procedimientos que en él se aplican.

Estudiando el capítulo se podrá notar el uso reiterativo de vocablos como: proceso, desarrollo, actividad, tarea, verificación, validación y la abreviatura V&V, que por el significado que tienen para la propuesta de solución no pueden ser sustituidos por ningún otro término.

2.2. Características del proyecto SIGEP

El proyecto SIGEP para su desarrollo está dividido en ocho subsistemas y cuarenta y nueve módulos, además de un subsistema esencial para la toma de decisiones (Sala Situacional). La cantidad de miembros es de ochenta y nueve (71 estudiantes y 18 profesores) los cuales están organizados en los diferentes equipos de trabajo: un equipo de arquitectura, un equipo de bases de datos, siete equipos de desarrollo, un equipo de analistas y un equipo de calidad interno. Los equipos de desarrollo están dirigidos por un diseñador y lo componen además, el diseñador gráfico y los programadores de las distintas capas de la aplicación (presentación, lógica de negocio, acceso a datos). La distribución de roles es realizada como se muestra en la Figura 8. Cada rol tiene definido el flujo de trabajo que debe seguir para realizar sus responsabilidades. La preparación previa al desarrollo permitió que cada programador domine en detalle la arquitectura definida para el proyecto, algo muy importante a la hora de implementar las funcionalidades de cada módulo. Existe una comunicación constante entre el diseñador y los programadores de su equipo de desarrollo, de igual manera con el equipo de arquitectura para la toma de decisiones que impliquen cambios arquitectónicos. Los analistas son un eslabón imprescindible para el completo entendimiento del negocio por parte del equipo de desarrollo encargado de implementar las funcionalidades relativas a este.

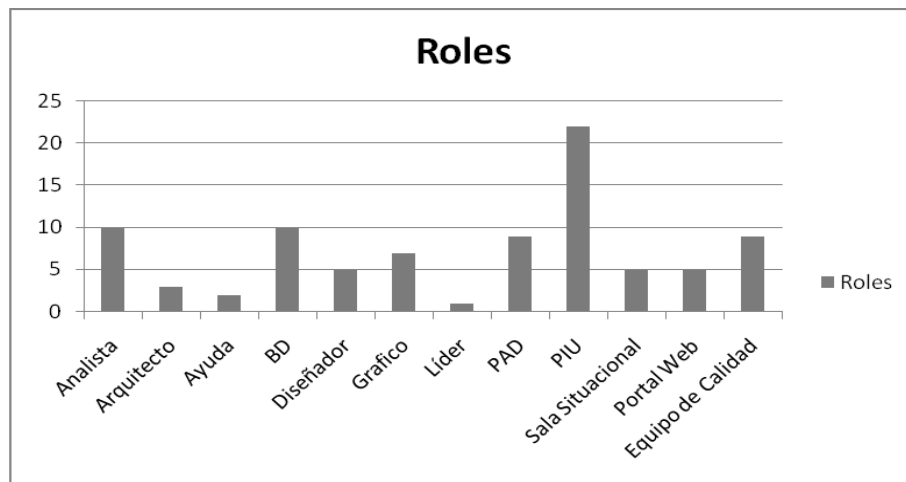


Figura 8. Distribución de roles del proyecto SIGEP

2.3. Guía de adopción del estándar IEEE 1012-1998

Para adecuar el estándar IEEE 1012-1998 a las características particulares del proyecto SIGEP, se tuvieron en cuenta cinco aspectos fundamentales:

- El *nivel de integridad del software*: es un parámetro que mide la criticidad del software y que después de definido será el punto fundamental a tener en cuenta para tomar las decisiones precisas sobre los próximos tres aspectos.
- *Mínimo de tareas recomendadas para el nivel de integridad del software*: el estándar define las tareas mínimas que deben ser realizadas para cada nivel de integridad. En dependencia de las condiciones internas del proyecto se podrá prescindir de alguna de ellas o agregar otras que se consideren necesarias.
- *Intensidad y rigor de las tareas V&V*: en dependencia del nivel de integridad definido se aplicarán con más o menos rigor estas tareas, y se decidirá si son desarrolladas por el propio personal de desarrollo, un equipo diferenciado dentro del proyecto o una organización externa (auditora).
- *Los criterios que se emplearán en las tareas V&V* para establecer los parámetros mínimos de corrección, completitud, consistencia, exactitud, legibilidad y comprobabilidad.
- *Las características del esquema productivo del proyecto SIGEP*, como conocimiento obligatorio del entorno para el que se genera el plan y criterio que rige la elaboración del mismo.

2.3.1. Nivel de integridad del software

Para determinar el nivel de integridad del SIGEP se realizó un análisis de la criticidad del software, que implica directamente el análisis de los riesgos y daños que existen ante el fallo del sistema o de algunas de sus funciones para cuantificar su posible impacto, y de esta manera determinar el nivel de integridad, teniendo en cuenta las posibilidades de mitigar estos riesgos y daños.

Para este análisis se adoptó el esquema de niveles de integridad que propone la IEEE 1012-1998.

Nivel	Dimensión del daño por fallo del software	Mitigación posible
4	<ul style="list-style-type: none"> – Pérdidas de vidas humanas – Pérdida de la seguridad del sistema – Graves pérdidas económicas o sociales 	No es posible mitigar los daños producidos
3	<ul style="list-style-type: none"> – El sistema no completa el objetivo que tiene previsto. – Graves pérdidas económicas o sociales 	Es posible una mitigación parcial de los daños
2	<ul style="list-style-type: none"> – El sistema incumple parcialmente con el objetivo que tiene previsto. – Pérdidas sociales o económicas importantes. 	Se pueden mitigar los daños producidos
1	<ul style="list-style-type: none"> – Una determinada función no se realiza. – Consecuencias mínimas 	No es necesario mitigar los daños.

Tabla 1. Esquema de niveles de integridad

El citado análisis se efectuó en una reunión de trabajo con el analista principal y el líder de proyecto (Anexo B), los cuales son especialistas en el negocio, algo reconocido por la parte cliente. Se examinó cada módulo y subsistema según el objetivo y el impacto que tiene dentro del sistema y se llegó a la siguiente conclusión:

Aunque los módulos y subsistemas del proyecto tienen diferentes objetivos y el impacto de cada uno para el sistema es distinto, tomando como base el esquema propuesto por la norma IEEE 1012-1998, pueden clasificarse todos en el nivel 2 de integridad.

Tomando esto en cuenta, todos los aspectos que dependen del nivel de integridad del software, se tratarán atendiendo al nivel 2, y sobre esta base se organizarán los esfuerzos V&V.

2.3.2. Mínimo de tareas recomendadas para el nivel de integridad del software

Las tareas seleccionadas para darle cumplimiento dentro del Plan de Verificación y Validación tienen como base el conjunto de tareas mínimas que establece el estándar para el nivel de integridad 2, las cuales fueron ajustadas a las particularidades del proyecto.

2.3.3. Intensidad y rigor de las tareas V&V

Las tareas serán aplicadas con el rigor e intensidad que requieren las condiciones del ambiente de desarrollo en el cual se realizarán, y conforme a esto se detallan en la propuesta. Se contará con el Equipo de Calidad que existe dentro del proyecto (IV&V Interna) y se contemplarán dentro del plan las labores V&V que debe llevar a cabo el propio desarrollador en función de garantizar la calidad de su trabajo (IV&V Doméstica).

Se cuenta además con los procesos de validación realizados cada siete semanas con el cliente, previo a los cuales el Grupo Central de Calidad de la UCI realiza las pruebas de liberación (IV&V Modificada).

2.3.4. Criterios que se emplearán en las tareas V&V

Las tareas del plan están enfocadas a medir parámetros como consistencia, corrección, completitud, exactitud, legibilidad o comprobabilidad de un producto de trabajo determinado. Como conceptos globales estos términos se refieren a:

Consistencia: El grado de uniformidad, estandarización, y libertad de contradicciones entre la documentación o parte de un sistema o componente.

Corrección: (1) El grado en que un sistema o componente está libre de fallas en su especificación, diseño e implementación.

(2) El grado en que el software, documentación u otro producto cumplen con sus requerimientos específicos.

Completitud: (1) Nivel o grado en que algo está completo.

(2) El estado de estar completo y entero.

Exactitud: (1) Una evaluación cualitativa de la corrección, de la libertad de errores.

(2) Una medida cuantitativa de la magnitud del error.

Legibilidad: Calidad de la escritura que hace que sea fácil de leer y entender

Comprobabilidad: (1) El grado en que un sistema o componente facilita el establecimiento de criterios de prueba y la ejecución de las pruebas para determinar si esos criterios han sido satisfechos.

(2) El grado en que un requerimiento es indicado en términos de permitir el establecimiento de criterios de prueba y la ejecución de las pruebas para determinar si esos criterios han sido satisfechos.

En el plan se detallan los criterios específicos a evaluar en cada parámetro dentro de las tareas. Asegurar que un producto de trabajo satisface esos criterios, es un aporte significativo (desde ese nivel) a la calidad con que se desarrolla el producto.

2.3.5. Características del esquema productivo del proyecto SIGEP

A partir de las características explicadas en el inicio del capítulo y otros elementos obtenidos a raíz de las entrevistas realizadas y de la observación de los autores, se definieron los principales aspectos a tener en cuenta para la confección del plan:

- La *distribución de roles del proyecto* facilita la asignación de responsabilidades a estos dentro de cada tarea, ajustando los fondos de tiempo para emplear sus esfuerzos en diferentes funciones que dependan del rol.
- La *organización del trabajo en equipos con objetivos bien definidos*, que permite que cada cual cumpla las funciones establecidas en constante comunicación con los demás miembros.
- La *estrategia de desarrollo utilizada* donde a partir del modelado del negocio y la captura de requisitos se desarrolla iterativamente y se valida con el cliente en un rango aproximado de siete semanas.
- Las *peculiaridades en la realización de las actividades dentro del proyecto*, que son muy específicas de este entorno (por ejemplo, el trabajo conjunto de analistas, diseñadores, programadores para el entendimiento, análisis e implementación de las soluciones).
- La *experiencia de trabajo de los miembros del proyecto* que brinda una garantía adicional para la correcta realización de las tareas.

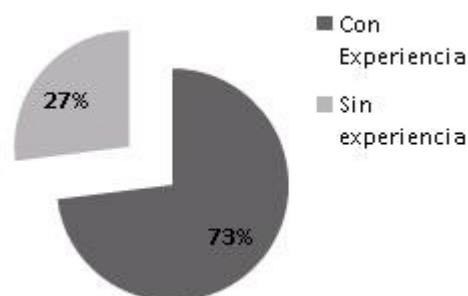


Figura 9. Experiencia de trabajo de los miembros del proyecto

2.4. El Plan V&V SIGEP

A partir de las características del proyecto SIGEP, con la base teórica estudiada, y con la definición de adoptar la norma IEEE 1012-1998, se crea el Plan de Verificación y Validación (Anexo A), constituido por seis actividades que agrupan veinticinco tareas.

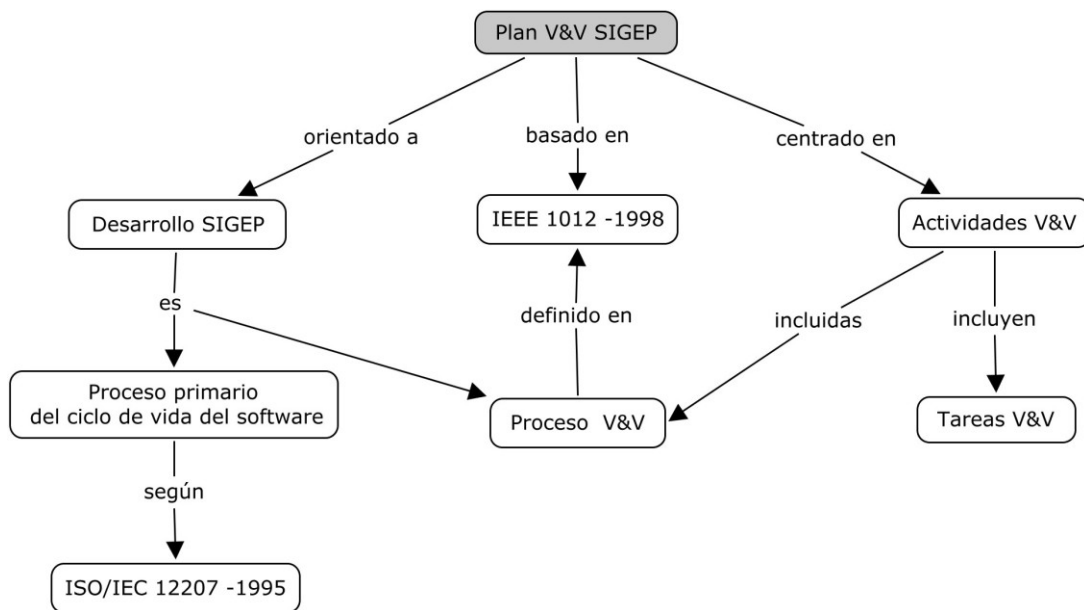


Figura 10. Vista general del Plan V&V SIGEP

2.4.1. Estructura

La estructura está sustentada por la que propone la norma adoptada, con las variaciones necesarias para un mejor entendimiento de su contenido y facilidad de aplicación en el entorno del proyecto.

Plan V&V SIGEP	
1.	Propósito
2.	Documentos referenciados
3.	Definiciones y Abreviaturas
4.	Vista general V&V
4.1.	Cronograma general
4.2.	Esquema de niveles de integridad
4.3.	Recursos
4.4.	Responsabilidades
5.	Procesos V&V
5.1.	Proceso de Desarrollo
5.1.1.	Actividad: Conceptualización V&V
5.1.2.	Actividad: Requerimientos V&V
5.1.3.	Actividad: Diseño V&V
5.1.4.	Actividad: Implementación V&V
5.1.5.	Actividad: Pruebas V&V
5.1.6.	Actividad: Instalación y Chequeo V&V

Figura 11. Estructura del Plan V&V SIGEP

A continuación se detallan sus elementos principales:

2.4.1.1. Propósito

En el propósito se especifican los objetivos que se persiguen con la planificación de la Verificación y la Validación para el proyecto, puntualizando el (los) procesos del ciclo del vida del software para el cual se prepara el plan. (Anexo A – Sección 1)

2.4.1.2. Documentos referenciados

Se especifican los documentos referenciados dentro del Plan V&V. (Anexo A – Sección 2)

2.4.1.3. Definiciones y abreviaturas

Se definen todos los términos necesarios para una buena comprensión del plan además de las abreviaturas utilizadas dentro del mismo. (Anexo A – Sección 3)

2.4.1.4. Cronograma general

Se propone un cronograma tipo, para lograr un plan más flexible y aplicable a las iteraciones del proyecto, sin restricciones a fechas exactas, que especifica los siguientes elementos:

- El momento dentro del desarrollo en que se ejecuta la actividad, ilustrando la correspondencia entre las actividades V&V y las del desarrollo del SIGEP.
- El orden de ejecución de las actividades.
- Las actividades que componen una iteración (Diseño V&V, Implementación V&V, Pruebas V&V, Instalación y Chequeo V&V) en concordancia con las características del proyecto.
- El orden de ejecución de las tareas, que está determinado por las relaciones de dependencia que existen entre ellas, y es lo que permite definir cuáles se pueden hacer concurrentemente y cuáles deben esperar a la culminación de una anterior.

Los recursos humanos (roles) necesarios para cumplir una actividad no se precisan en el cronograma, estos son descritos para cada tarea en específico.

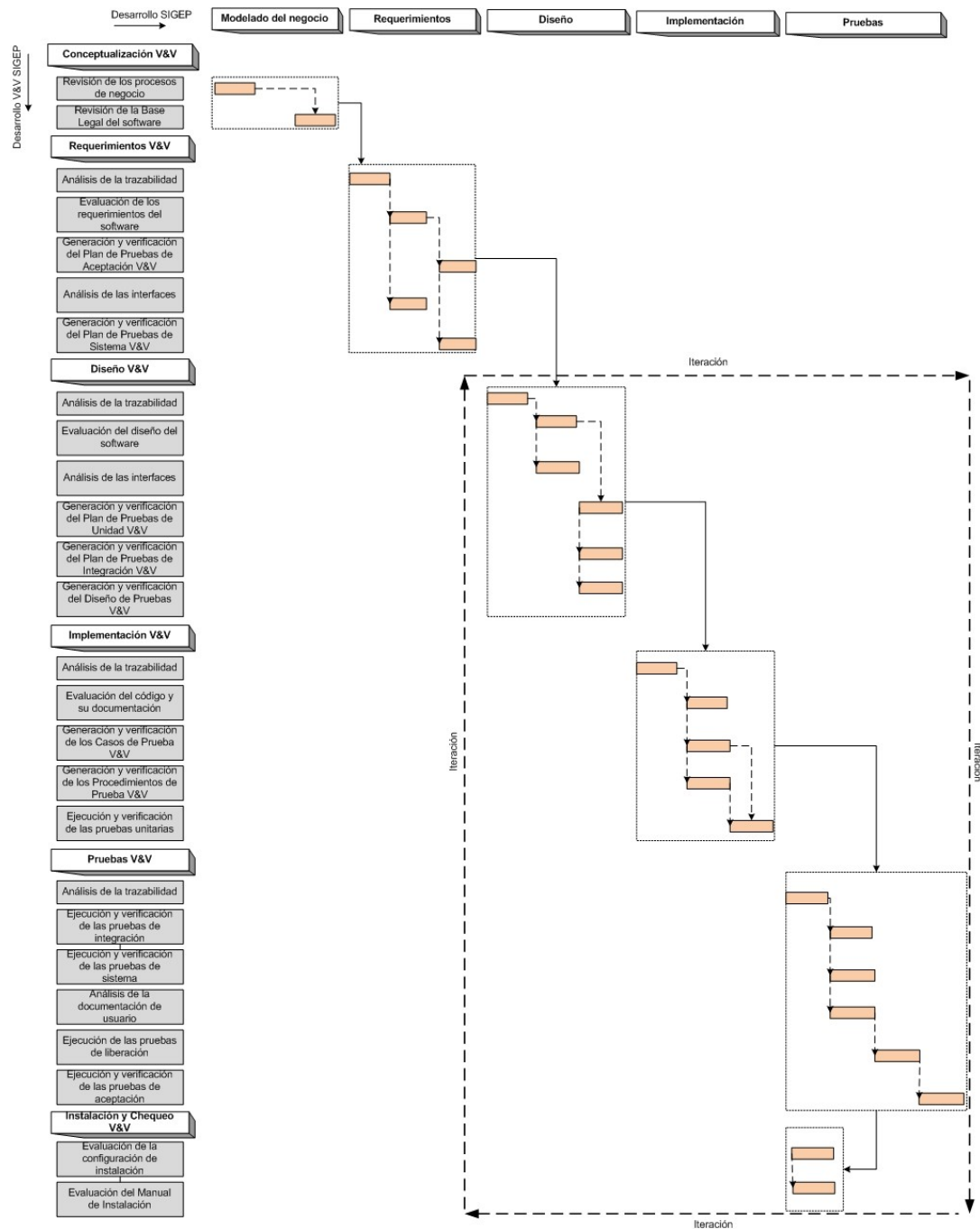


Figura 12. Cronograma tipo

2.4.1.5. Esquema de niveles de integridad

Se muestra el esquema de niveles de integridad adoptado (Anexo A – Sección 4.2) para determinar el nivel de cada módulo y subsistema del proyecto.

2.4.1.6. Recursos

En esta sección se hace un resumen de los recursos humanos (distribuidos por roles) y tecnológicos (Anexo A – Sección 4.3) con los que cuenta el proyecto para la puesta en práctica del plan.

Existen roles que aunque no forman parte del equipo de desarrollo, juegan un papel importante para la realización de algunas tareas y por eso se especifican, es el caso del *Especialista de procesos* y el *Cliente*. Este último a diferencia de otros, no tiene una cantidad fija, puesto que varía en dependencia de las responsabilidades que tiene en las tareas en las cuales se implica, y de las decisiones de la gerencia de la parte cliente, por ejemplo, su participación en las pruebas de aceptación.

Los recursos tecnológicos son básicamente computadoras asignadas a los miembros del proyecto y otros dispositivos como escáner de huellas y cámaras digitales. Las computadoras están distribuidas en los tres laboratorios asignados al SIGEP de manera tal que cada integrante tiene su puesto y horario de trabajo, constituyendo de esta forma el recurso más importante para la realización de las tareas V&V.

2.4.1.7. Responsabilidades

La asignación de responsabilidades se hace teniendo en cuenta que el primer esfuerzo V&V lo realiza el desarrollador (analistas, diseñadores, documentadores, programadores, en dependencia de la actividad que se esté realizando), que debe garantizar la calidad del proceso que ejecuta y del producto que se obtiene del mismo. El equipo de calidad interna del proyecto, en funciones de revisor, diseñador de pruebas, probador, está presente en todas las actividades del desarrollo y jugando su papel de contraparte obtiene un nivel de conocimientos vital para ejecutar de la manera más efectiva las pruebas y revisiones planificadas. La descripción detallada de las funciones precisas de cada rol con fines V&V se realiza dentro de cada tarea.

Un mismo rol puede asumir diferentes responsabilidades, dependiendo de la tarea en la cual participa.

Ejemplo de esto es el Analista de negocio, y se ilustra en las dos tareas siguientes:

Tarea 1: Revisión de los procesos de negocio (Conceptualización V&V)

Analista de negocio: Lleva a cabo la comprensión de todos los procesos de negocio y los describe en detalle, especificando el flujo de actividades de cada uno de ellos. De esto se obtienen los Diagramas de Actividades del Negocio y la Descripción Textual del Negocio (DTN).

Tarea 2: Evaluación de los requerimientos de software (Requerimientos V&V)

Analista de negocio: Identifica y describe las funcionalidades de sus módulos verificando constantemente su trabajo en cuanto a los criterios plasmados en la descripción de la tarea.

El Anexo D representa una matriz de tres dimensiones que muestra los métodos que aplican los roles para cumplir cada tarea del plan.

2.4.1.8. Procesos V&V

Se especifican todos los procesos para los cuales se está planificando la Verificación y la Validación, describiendo las actividades y tareas que incluyen.

La norma IEEE 1012-1998 define cinco procesos de Verificación y Validación que cubren todo el ciclo de vida del software: Adquisición V&V, Suministro V&V, Desarrollo V&V, Operación V&V, Mantenimiento V&V.

Adquisición V&V: Este proceso comienza con la definición de la necesidad de adquirir un software, un producto o servicio de software. Continúa con la preparación y emisión de una solicitud de oferta (oferta de precio que somete el usuario al proveedor), selección de un suministrador y la gestión del proceso de adquisición a través de la aceptación del software, producto o servicio de software. Los esfuerzos V&V en este proceso están dirigidos a determinar el alcance inicial de la Verificación y Validación, planificar las interacciones entre el adquiridor y el suministrador, y revisar el diseño inicial de los requerimientos del software contenidos en la solicitud de oferta.

Suministro V&V: El proceso es iniciado por cualquier decisión de responder a la solicitud de oferta del adquiridor o por firmar y acceder al contrato con este para suministrar el software, producto o servicio de software. Continúa con la determinación de los procedimientos y recursos necesarios para administrar el proyecto, incluyendo el plan del proyecto y su ejecución a través del desarrollo del software, producto o servicio de software para el adquiridor. Los esfuerzos V&V están dirigidos a verificar que el diseño inicial de los requerimientos del software contenidos en la solicitud de oferta y los requerimientos del contrato son consistentes y satisfacen las necesidades del usuario.

Desarrollo V&V: El proceso de desarrollo contiene las actividades y tareas del desarrollador. Se incluyen las actividades referentes a la modelación del negocio, análisis de requerimientos, diseño, codificación, pruebas, instalación y aceptación de los productos de software. Estas actividades están organizadas en: Conceptualización V&V, Requerimientos V&V, Diseño V&V, Implementación V&V, Pruebas V&V e Instalación y Chequeo V&V.

Operación V&V: Este proceso cubre la operación del producto de software y el soporte operacional para los usuarios.

Mantenimiento V&V: El proceso es activado cuando el producto de software sufre modificaciones al código y la documentación asociada, causado por un problema o necesidad de mejora o adaptación.

Los procesos de Adquisición V&V y Suministro V&V no se encuentran en la propuesta, puesto que son etapas ya realizadas en el proyecto y no es necesario dedicar esfuerzos a ellos.

El plan contiene las actividades y tareas del Desarrollo V&V, que es el proceso más extenso de todo el ciclo de vida y al cual se dedican la mayoría de los esfuerzos.

Los procesos de Operación V&V y Mantenimiento V&V tampoco se incluyen en la propuesta, pero son objeto de estudio para su posterior planificación. El Mantenimiento guarda estrecha relación con el proceso de Desarrollo, puesto que las modificaciones que se realicen al software tienen que someterse a gran parte de las tareas planificadas para este último, iterando las que se consideren necesarias.

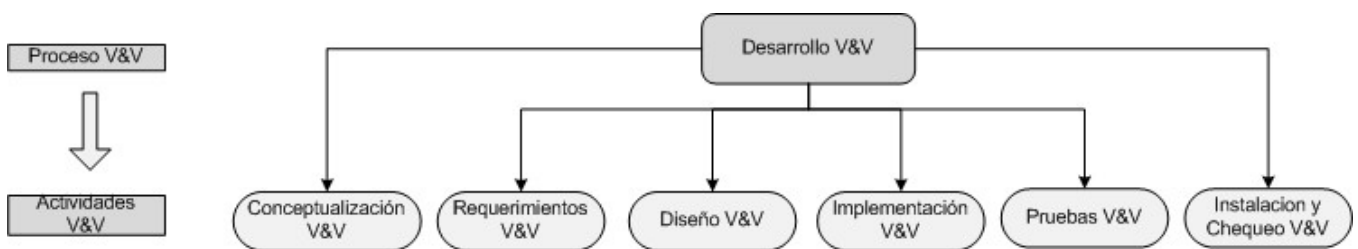


Figura 13. Proceso de Desarrollo V&V

2.4.2. La Actividad V&V

La actividad es el elemento principal para organizar los esfuerzos V&V. Cada una se dirige a una etapa diferente dentro del desarrollo del software, constituyendo los puntos de relación directa con el entorno de trabajo que se evalúa, por ejemplo la correspondencia entre Implementación V&V y la fase de Implementación del SIGEP. Para la culminación exitosa de una actividad se requiere del cumplimiento de todas las tareas que la componen.

En función de los aspectos que se tuvieron en cuenta para la adopción de la norma, el Proceso de Desarrollo V&V del plan tendrá las mismas actividades que plantea la IEEE 1012-1998, ajustándolas a las particularidades del proyecto y redefiniendo los aspectos necesarios de cada una de ellas. De esta manera se describen:

Conceptualización V&V

Se sientan las bases de todo el respaldo legal que debe tener el software, se identifican los procesos de negocio de la organización que se describen textualmente para una mejor comprensión de los mismos. Los esfuerzos V&V están encaminados fundamentalmente a verificar y validar que los procesos de negocio identificados reflejan correctamente el funcionamiento de la organización y que fueron documentados adecuadamente.

La Figura 14 muestra todas las tareas de la actividad y los roles que las ejecutan, además de las entradas y salidas requeridas.

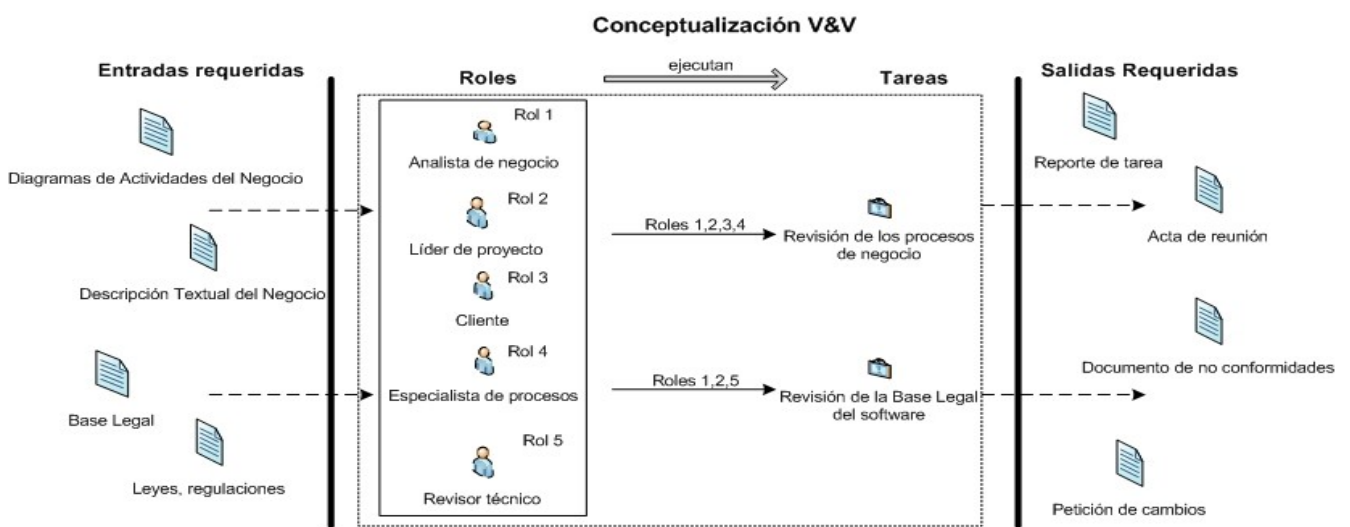


Figura 14. Conceptualización V&V

Requerimientos V&V

Se definen los requerimientos funcionales y no funcionales del software, las interfaces, y se construye la primera versión de la base de datos. Dirige el análisis de los requerimientos del software. El objetivo fundamental de los esfuerzos V&V es asegurar la corrección, completitud, exactitud, comprobabilidad y consistencia de los requerimientos.

Diseño V&V

En esta actividad los requerimientos del software son transformados en arquitectura y diseño detallado para cada componente del software. El diseño incluye bases de datos (entidades de dominio), interfaces de acceso a datos, interfaces de negocio y además la interfaz gráfica. Esta actividad guía el análisis y diseño de la solución. El objetivo fundamental de la V&V es demostrar que el diseño es una correcta, precisa y completa transformación de los requerimientos del software y que no se han introducido errores.

Implementación V&V

En la actividad de implementación se transforma el diseño en código, estructuras de bases de datos y las representaciones ejecutables relacionadas. Dirige la codificación del software. Los objetivos de la V&V son verificar y validar que estas transformaciones son correctas, precisas y completas.

Pruebas V&V

Esta actividad cubre las pruebas del software. El objetivo fundamental es asegurar que los requerimientos del software son satisfechos con la ejecución de pruebas de sistema, integración y aceptación, además de realizar una revisión completa a la documentación de usuario que se genera en el proyecto.

Instalación y Chequeo V&V

Esta actividad es la instalación del producto de software en el entorno previsto para su uso, así como la aceptación por parte del que lo adquiere. El objetivo fundamental es verificar y validar la corrección de la instalación del software en al ambiente correspondiente.

2.4.3. La Tarea V&V

La tarea es el componente más pequeño del Plan V&V. Está orientada a evaluar un producto de trabajo determinado dentro del desarrollo. La estructura que se propone cumple con lo establecido en la IEEE 1012-1998 y se muestra en el siguiente gráfico:

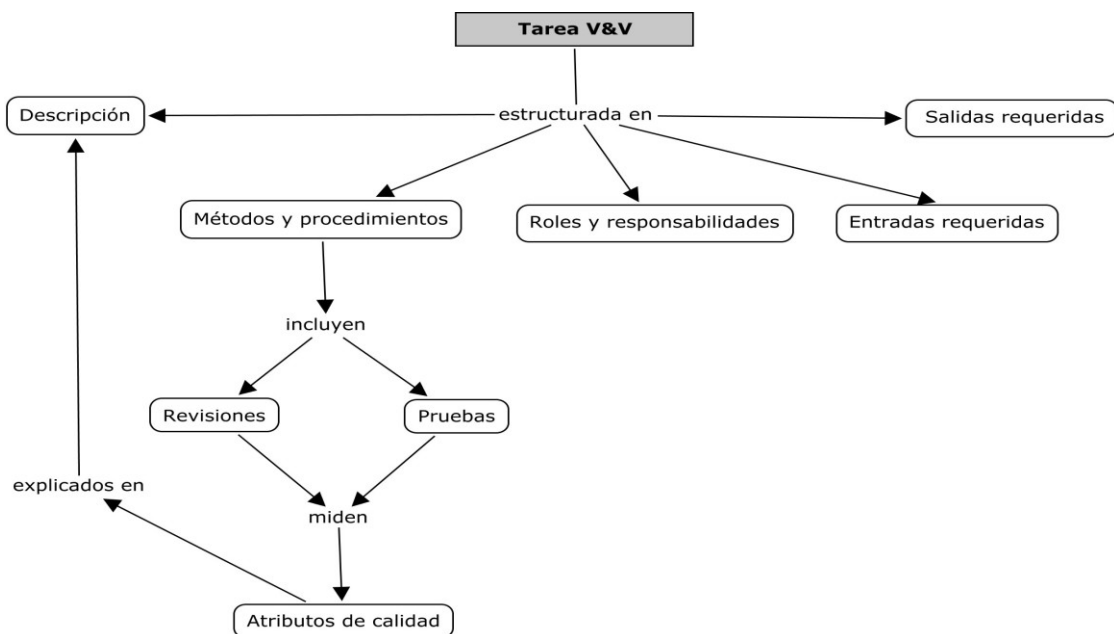


Figura 15. Estructura de la Tarea V&V

En la *descripción* se especifica claramente la manera en que se evaluará el producto de trabajo, atendiendo a los parámetros de corrección, completitud, consistencia, exactitud, legibilidad y comprobabilidad, para los cuales son definidos los criterios a tener en cuenta.

En los *métodos y procedimientos* se especifica el(los) método(s) que se utilizará(n) para darle cumplimiento a la tarea, así como el modo de aplicación (procedimiento) dentro del proyecto. Para la selección de los mismos se tuvieron en cuenta los conceptos que aporta la teoría estudiada, pero generalmente se aplican con algunas peculiaridades que derivan de las características y necesidades del ambiente productivo en el que se llevan a cabo.

Las *entradas requeridas* son aquellos artefactos necesarios para la ejecución de la tarea. Pueden constituir el objeto de revisión o prueba, o un material de apoyo vital para ello.

Las *salidas requeridas* son artefactos que se obtienen como resultado de la ejecución de la tarea. Las exigidas dentro del plan están en correspondencia con la norma IEEE 1012-1998, con las respectivas adaptaciones a la documentación generada en cada actividad del SIGEP.

Se tienen artefactos como el Reporte de Tarea, donde se registra el resultado de la ejecución de cada tarea del plan, pues este constituye una evidencia fundamental de que se realizó, además de quién, cómo y cuándo la ejecutó y de la cantidad de errores detectados. Este documento es una herramienta fundamental para obtener datos estadísticos de la V&V. (Anexo I)

Para la selección de las tareas a incluir en el plan, se realizó un análisis de las tareas mínimas requeridas para el nivel de integridad 2 establecidas por la norma, a las cuales se le hicieron las variaciones requeridas para lograr una correcta adaptación a la realidad del SIGEP, teniendo en algunos casos que unificar tareas y en otros adicionar o quitar.

La Conceptualización V&V se adaptó completamente a la fase de Modelación del Negocio del proyecto. Para esta se entiende que realizando una revisión de los procesos de negocio y de la Base Legal del software se garantiza su culminación exitosa, y por este motivo se incluyen solamente dos tareas en el plan:

Tarea 1: Revisión de los procesos de negocio

Descripción

Se evalúa el modelado de negocio en cuanto a completitud y corrección

Completitud

Verificar y validar que los procesos identificados modelan la estructura y funcionamiento completo de la organización.

Corrección

Verificar que los procesos identificados están descritos correctamente y no se viola ningún paso dentro del flujo que describe la dinámica de los mismos.

Métodos y procedimientos.

Los diagramas de actividades del negocio obtenidos a partir del entendimiento del negocio que se realiza partiendo de los procesos identificados son sometidos a una *prueba de validación* con el cliente venezolano.

De la misma manera la Descripción Textual del Negocio se somete a una *prueba de validación* con el cliente, que valida y firma la propuesta quedando formalizado su total acuerdo con lo planteado en la misma.

Roles y Responsabilidades

Especialista de procesos: Identifica y describe muy brevemente los procesos de negocio.

Analista de negocio: Lleva a cabo la comprensión de todos los procesos de negocio y los describe en detalle, especificando el flujo de actividades de cada uno de ellos. De esto se obtienen los Diagramas de Actividades del Negocio y la DTN.

Cliente: Valida los Diagramas de Actividades del Negocio y la DTN.

Líder de proyecto: Participa en las pruebas de validación con el cliente como máximo responsable de la solución de software.

Entradas requeridas

Diagramas de Actividades del Negocio

DTN

Salidas requeridas

Reporte de tarea

Documento de no conformidades (DNC) o Documento de Petición de Cambios.

En esta tarea se implica el cliente venezolano, ya que se considera que es la figura más preparada para realizar la validación.

Tarea 2: Revisión de la base legal del software.

Descripción

Evaluar la Base Legal del software en cuanto a corrección, completitud y legibilidad.

Corrección.

Verificar que se exprese correctamente la correspondencia entre los procesos definidos y los artículos de las leyes, normativas y reglamentos del sistema penitenciario.

Completitud

Verificar que se explica detalladamente cómo se aplica la ley para cada proceso definido.

Legibilidad

Verificar que el documento ha sido escrito cumpliendo con el formato establecido para el proyecto (Anexo K), en un vocabulario adecuado y claro.

Métodos y procedimientos

Se aplica un *walk-through* que será desarrollado en una o varias reuniones de trabajo entre el equipo de analistas, un miembro del equipo de calidad y el líder de proyecto, donde realizarán la evaluación de la Base Legal en cuanto a corrección y completitud.

Se aplica una *revisión técnica* para evaluar la legibilidad.

Roles y responsabilidades

Analista: Realiza la exposición en el walk-through

Líder de proyecto: Participa en el walk-through

Revisor Técnico: Participa en el walk-through y realiza la revisión

Entradas requeridas

Leyes, normativas y reglamentos vigentes del Sistema Penitenciario

Base Legal

DTN

Salidas requeridas

Reporte de tarea

Acta de cada reunión realizada

La revisión de aspectos como este es de vital importancia para un proyecto como el SIGEP, que automatiza el comportamiento de una entidad donde cada proceso tiene un respaldo legal importante, y de no lograrse el total cumplimiento de las leyes no se estará construyendo el sistema idóneo.

Existen tareas a las que se le da seguimiento en casi todas las actividades dentro del proceso de desarrollo. Se tiene en este caso el *Análisis de la Trazabilidad*, donde siempre se evalúan las relaciones entre los componentes de una actividad particular y sus predecesores en actividades anteriores.

Se tiene así que en la actividad de *Requerimientos V&V* en el análisis de la trazabilidad se evalúan las relaciones entre las funcionalidades descritas en el Documento de Descripción de Funcionalidades (DDF) y los procesos de negocio detallados en la Descripción Textual del Negocio en cuanto a: corrección, consistencia, completitud. La *corrección* radicará en validar que las relaciones entre funcionalidades y procesos de negocio identificados son correctas, la *consistencia* en verificar que las relaciones entre cada funcionalidad con los procesos de negocio correspondientes están especificadas detalladamente, y la *completitud* en verificar que cada funcionalidad descrita es traceable con alguno de los procesos de negocios identificados.

En la actividad de *Pruebas V&V* el análisis de la trazabilidad se dedica a evaluar las relaciones entre el Plan de Pruebas, el Diseño de Prueba, los Casos de Prueba y los Procedimientos de Prueba en cuanto a corrección y completitud, donde para la *corrección* se verifica que las relaciones entre todos estos artefactos son válidas y para la *completitud* se verifica que todos los Procedimientos de Prueba son traceables al Plan de Pruebas.

Esta misma actividad contiene una tarea para realizar la verificación y validación interna de la documentación de usuario. Se decidió situarla aquí porque es el momento dentro del desarrollo en que se obtienen productos terminados de estos artefactos y se considera que se deben evaluar porque permite obtener una documentación perfeccionada y con mayores posibilidades de ser aprobada por el cliente en la prueba piloto. La tarea será desarrollada por revisores del equipo de calidad, analistas y documentadores, de la manera que se muestra a continuación:

Tarea: Análisis de la documentación de usuario

Descripción

Evaluar la documentación de usuario en cuanto a legibilidad, completitud y corrección.

Legibilidad

Verificar que los manuales de usuario y la ayuda en línea se han escrito en un vocabulario correcto y claro, con la estructura y formato adecuados.

Completitud

Revisar que estos contengan toda la información necesaria para que el usuario comprenda todas las funcionalidades del sistema, y en el caso de la ayuda, que permita realizar búsquedas de contenidos.

Corrección

Verificar que la ayuda funciona correctamente integrada al SIGEP.

Métodos y procedimientos

Se realiza un *peer review* entre el Analista de negocio que confecciona el manual de usuario y un revisor técnico, evaluando los criterios expuestos en la descripción de la tarea.

Se realiza un *peer review* entre el responsable de la ayuda y un revisor técnico, evaluando los criterios expuestos en la descripción de la tarea.

Roles y responsabilidades

Analista de negocio: Confecciona el manual de usuario y participa en el peer review.

Responsable de la ayuda: Confecciona la ayuda y participa en el peer review.

Revisor Técnico: Participa en el peer review y elabora el DNC en caso de ser necesario.

Entradas requeridas

Manuales de usuario

Ayuda del sistema

Salidas requeridas

Reporte de tarea

DNC

2.4.4. Métodos y procedimientos aplicados

La aplicación de los métodos y procedimientos en el plan está muy orientada a la singularidad con que se realizan las tareas del desarrollo SIGEP, insertadas en un cronograma muy estricto que deriva de importantes compromisos contractuales de obligatorio cumplimiento.

Las revisiones como método estático de V&V son aplicadas en todas sus variantes; se tiene la comprobación de escritorio realizada por el propio desarrollador, miembro del equipo de calidad interna, líder de proyecto o cualquier integrante del equipo de desarrollo con la preparación necesaria para jugar el papel de revisor en el caso que se necesite. También se hacen revisiones técnicas en equipos, ya sean reuniones orientadas a la revisión puramente o las que propicia el trabajo en conjunto de un determinado equipo del proyecto. Los walk-through guiados por analistas, diseñadores y probadores, son aplicados para la obtención y revisión de artefactos fundamentales en actividades como Conceptualización V&V, Diseño V&V y Pruebas V&V. Los peer review que requieren del trabajo obligatorio de dos personas, también juegan un papel muy importante dentro del plan. Con la siguiente tarea se ilustra la aplicación del *walk-through*:

Tarea: Evaluación del diseño del software (Diseño V&V)

Descripción

Se evalúa el diseño del software en cuanto a corrección, completitud y consistencia.

Corrección

Verificar y validar que el diseño del software se ajusta a las especificaciones de la arquitectura definida para el proyecto:

- Que cumpla con las convenciones de código establecidas.
- Que se ajuste a la estructura de paquetes.
- Que estén bien asignadas las responsabilidades de cada capa arquitectónica.

Completitud

Verificar que están definidas todas las funcionalidades necesarias para satisfacer cada caso de uso.

Consistencia

Verificar que las funcionalidades con objetivos similares que puedan ser parametrizadas en una sola no estén separadas.

Verificar que están definidas las funcionalidades comunes para que sean implementadas solamente en un módulo.

Verificar que se han identificado todas las dependencias entre los módulos.

Métodos y procedimientos

Se aplica un *walk-through* donde en una o varias sesiones de trabajo que tendrán lugar obligatoriamente con todos los diseñadores del proyecto, donde cada diseñador explica al resto la propuesta de solución que tiene para su módulo y en función de esto se toman todas las decisiones pertinentes para obtener un diseño correcto, completo y consistente.

Se aplica una *revisión técnica* por parte del arquitecto de software que debe indicar los errores y explicar la forma idónea de proceder para solucionar cada uno.

Roles y responsabilidades

Diseñador: Construye el diseño del software y participa en el walk-through

Arquitecto de software: lleva a cabo la revisión e interviene en el diseño de la solución de ser necesario.

La prueba como método dinámico de V&V tiene un peso importante en el plan, a ella se dedica una actividad completa dentro del proceso de desarrollo, como lo plantea la IEEE 1012-1998. La generación y verificación de los planes, diseño, procedimientos y casos de prueba se hace en diferentes momentos del desarrollo anteriores a la actividad de Pruebas V&V, que es donde se ponen en práctica, con la excepción de las pruebas unitarias que se ejecutan en la actividad de Implementación V&V.

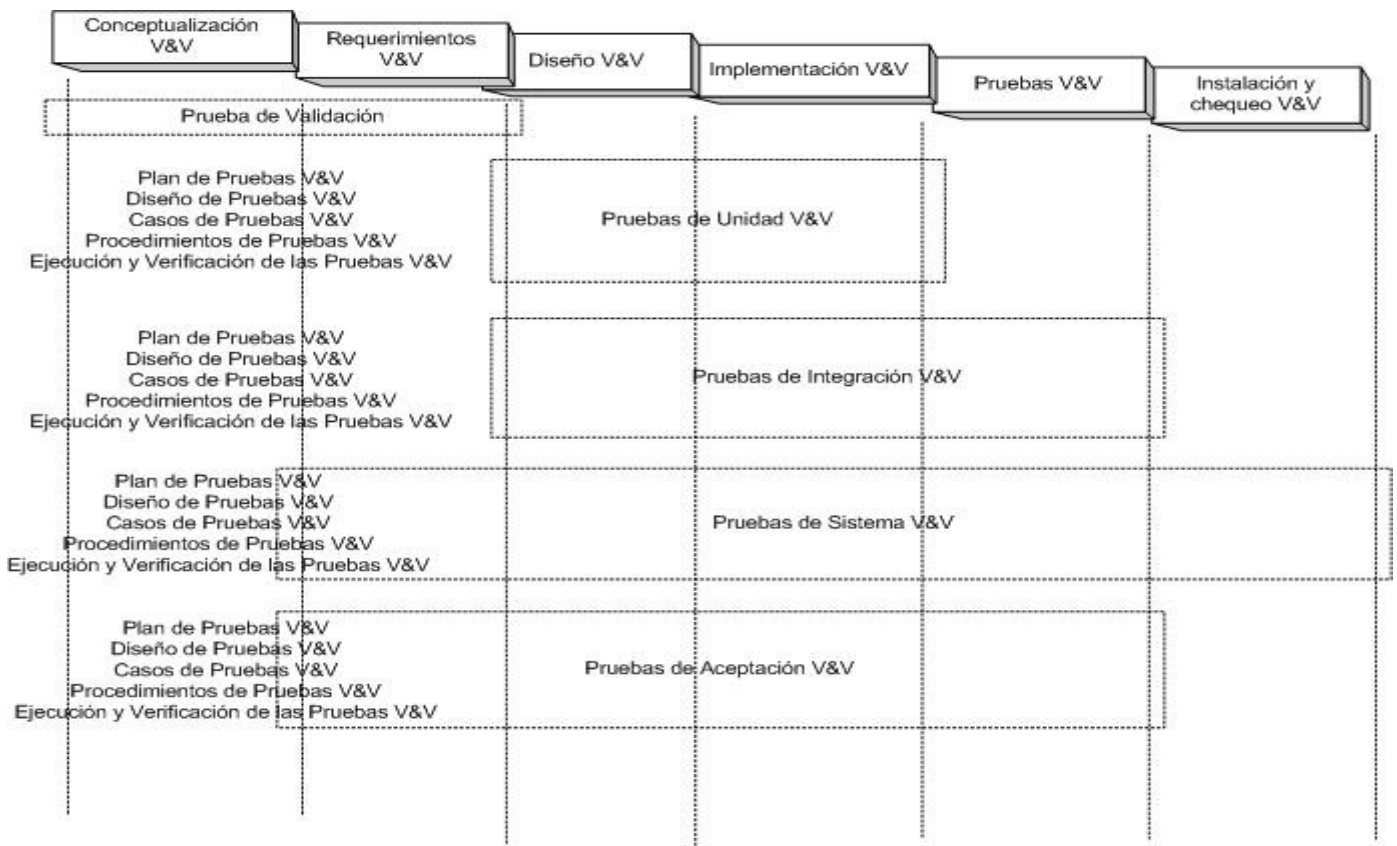


Figura 16. Vida de las pruebas a través de las Actividades V&V SIGEP

El plan incluye pruebas de unidad para la capa de acceso a datos y de negocio, pruebas de integración, pruebas de sistema y pruebas de aceptación. Para las pruebas de sistema se propone la realización de pruebas de funcionalidad, rendimiento, estrés y seguridad. La prueba de seguridad está orientada fundamentalmente al módulo de seguridad del sistema y a la administración del mismo.

Para la prueba de integración se especifica el enfoque que tiene su ejecución en el SIGEP, que al ser un sistema en el cual el diseño de la arquitectura obliga al desarrollo integrado, esta tiene el objetivo de verificar que al ejecutar una determinada acción, se actualicen consistentemente todas sus dependencias dentro del sistema. Es importante destacar el peso que tiene para la ejecución de la prueba de integración que las postcondiciones de cada caso de uso estén correctamente planteadas en la descripción de funcionalidades, pues estas constituyen una guía indispensable para el probador, porque detallan todas las implicaciones que tiene la ejecución de una determinada funcionalidad. Para ganar en claridad se muestra a continuación la especificación de la tarea correspondiente:

Tarea: Ejecución y verificación de las pruebas de integraciónDescripción

Se ejecutan las pruebas de integración que fueron planificadas en la actividad de Diseño V&V, con el objetivo de evaluar el comportamiento del software como un todo, teniendo en cuenta que de la correcta integración entre los módulos depende que no existan inconsistencias en los datos registrados en el sistema. Se verifican los resultados contra los criterios de aceptación para la prueba.

Métodos y procedimientos

Se realizan las pruebas de integración y se evalúan los resultados contra los criterios de aceptación previamente definidos.

Roles y responsabilidades

Diseñador: Ejecuta la prueba.

Probador: Participa en la ejecución de la prueba y confecciona el DNC.

Arquitecto de software: Supervisa la prueba

Entradas requeridas

Versión estable del software.

Casos de prueba.

Criterios de aceptación de la prueba

Salidas requeridas

Reporte de tarea

DNC (Anexo N)

También, con el objetivo de evaluar la corrección y completitud del instalador del SIGEP y la completitud del Manual de Instalación que forma parte de la documentación de usuario generada, se aplica una prueba de instalación en la actividad de Instalación y Chequeo V&V.

2.5. Conclusiones parciales

Se explicaron los aspectos que se tuvieron en cuenta para adaptar el estándar IEEE 1012-1998 a las condiciones del esquema productivo del proyecto SIGEP, realizando previamente una caracterización general del mismo.

Se desglosaron las partes que conforman el Plan V&V explicando cómo se aborda cada una dentro de la propuesta. Se explica la Actividad V&V y se exponen las características de todas las incluidas en el plan, además de esquematizar las tareas que las componen ilustrando los roles que intervienen en cada una, así como sus entradas y salidas requeridas. De la Tarea V&V se hace un desglose de los aspectos que definen su estructura revelando como se enfrentaron para confeccionar las tareas V&V SIGEP, haciendo énfasis en los métodos y procedimientos, por la variedad y especialización que tiene su aplicación dentro del plan.

Se obtuvo una propuesta capaz de introducir buenas prácticas y estandarizar los procesos de Verificación y Validación para el desarrollo del SIGEP.

CAPÍTULO III: VALIDACIÓN DE LA PROPUESTA

3.1. Introducción

Para validar la propuesta de Plan V&V que se definió en el capítulo anterior, se conformó un Panel de Expertos que emitió su criterio. El proceso de validación se realizó mediante el Método Delphi que según algunos autores como Jon Landeta en su artículo *Aplicación del Método Delphi en la elaboración de la tabla simétrica de las tablas input-output 2001 de Catalunya*:

- Es una técnica de investigación social que tiene como objeto la obtención de una opinión grupal fidedigna a partir de un grupo de expertos.
- Es un método de estructuración de la comunicación entre un grupo de personas que pueden aportar contribuciones valiosas para la resolución de un problema complejo. (LANDETA *et al.* 2003)

3.2. El Método Delphi

El Método Delphi, cuyo nombre se inspira en el antiguo oráculo de Delphos, fue ideado originalmente a comienzos de los años 50 en el seno del centro de investigación estadounidense RAND Corporation¹¹ por Olaf Helmer y Theodore J. Gordon, como un instrumento para realizar predicciones sobre un caso de catástrofe nuclear. Desde entonces, ha sido utilizado frecuentemente como sistema para obtener información sobre el futuro. (ASTIGARRAGA 2004)

Proporciona un medio para agregar sistemáticamente los juicios informados de un grupo de expertos dentro de su campo de conocimiento. Las opiniones de los participantes no se ven influenciadas por confrontaciones cara a cara. Además, el anonimato de los expertos permite expresar sus opiniones reales sin temor o influencias debidas al efecto autoridad. (ARAGÓN 2003)

Todo uso del Método Delphi asume el denominado Efecto McGregor que afirma que el juicio colectivo de un grupo de expertos es considerado como más fiable que sus juicios individuales y por tanto, más objetivo en sus conclusiones. (ARAGÓN 2003)

El anonimato de los panelistas evita presiones hacia la conformidad asociadas con reacciones ante las ideas de los otros y facilita la igualdad de los participantes evitando el efecto autoridad asociado a

¹¹ Corporación RAND: Tanque pensante norteamericano, formado en un primer momento para ofrecer investigación y análisis a las fuerzas armadas norteamericanas. Criticada por su marcado carácter militarista

panelistas muy relevantes que podrían ser percibidos como referentes por el resto del grupo. (ARAGÓN 2003)

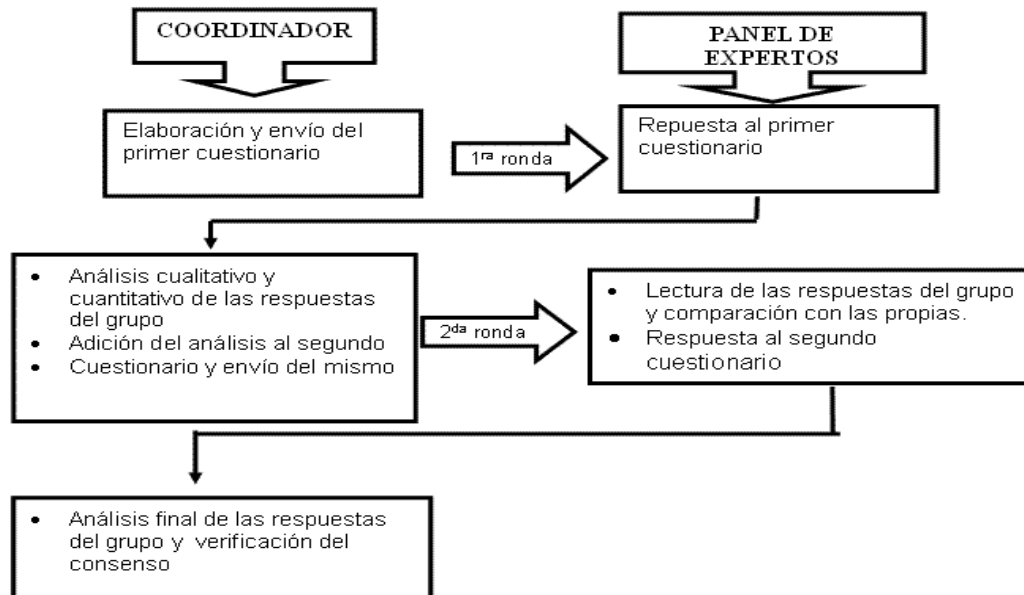


Figura 17. Diagrama del procedimiento del Método Delphi. Fuente: (TORRES et al. 2005)

3.3. Aplicación del método

Según los investigadores María de Lourdes Bravo y José Joaquín Arrieta en su artículo *El Método Delphi. Su implementación en una estrategia didáctica para la enseñanza de las demostraciones geométricas*, hay 2 aspectos básicos a tener en cuenta para la aplicación del método:

- La selección del grupo de expertos a encuestar: personas conocedoras, con reconocida competencia y con experiencia en el tema que garantice la confiabilidad de los resultados, creativos e interesados en participar.
- Elaboración de los cuestionarios: tener en cuenta la teoría de la comunicación, con mecanismos que reduzcan los sesgos en las respuestas, preguntas claras, precisas e independientes. Suelen ser preguntas cuantitativas para calcular medias y rangos, y cualitativas para la justificación de sus opiniones. (BRAVO and ARRIETA 2005)

3.3.1. Proceso de selección de los Expertos

Se entiende por experto, tanto al individuo en sí como a un grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones

respecto a sus momentos fundamentales con un máximo de competencia. (URIZARRI and FERNÁNDEZ 2005)

Se conformó un panel integrado por especialistas de la Dirección de Calidad de Software de la universidad y otros especialistas del área productiva con experiencia en temas relacionados con la investigación, así como directivos con un alto nivel de conocimiento en temas de calidad de software.

Por la limitación en tiempo y recursos se decidió que la cantidad de expertos no fuera muy numerosa, aunque se corresponde con el intervalo sugerido por los autores José Ignacio Ruiz Olabuénaga y María Antonia Ispizua Uribarri en su libro *La descodificación de la vida cotidiana: métodos de investigación cualitativa.* (OLABUÉNAGA and URIBARRI 1989)

Los expertos se seleccionaron según las características siguientes:

- Conocimientos acerca de los contenidos que sustentan la propuesta de solución:
 - ✓ Calidad de Software
 - ✓ Verificación y Validación
 - ✓ Métodos de Verificación y Validación (estáticos y dinámicos)
 - ✓ Proceso de Desarrollo del Software
 - ✓ Normas y estándares de calidad
 - ✓ Planificación
- Años de experiencia trabajando los temas que se señalaron anteriormente
- Prestigio en el colectivo de trabajo
- Capacidad de análisis y pensamiento lógico
- Integración a las actividades productivas

Elegir los expertos atendiendo a las características mencionadas propicia obtener resultados con calidad, junto a otras cualidades propias de estos como pueden ser: la seriedad, la honestidad, la sinceridad, la responsabilidad y otras en este sentido, que hacen que las opiniones brindadas sean confiables y válidas para el objetivo propuesto.

El número de expertos seleccionados fue nueve, de ellos tres con el título de Ingenieros Informáticos y seis con el título de Ingenieros en Ciencias Informáticas. Por la ocupación actual se desglosan en: seis

Especialistas de la Dirección de Calidad de la UCI, un Asesor de Calidad de una facultad, un directivo del Centro de Calidad, Seguridad y Estándares del UCID, y un Decano de una Facultad. (Anexo G)

Una vez confeccionado el panel se invitó cada experto de manera personal para que participaran en el proceso de validación y aceptación de la propuesta.

De esta manera se termina el primer paso correspondiente a la selección del personal que integra el Panel de Expertos, por lo que se procede a la elaboración y aplicación del cuestionario.

3.3.2. Elaboración del cuestionario

Para el procesamiento y análisis de la información se tiene en cuenta el tipo de pregunta si es abierta o cerrada y se valora desde lo cualitativo o cuantitativo respectivamente.

En el cuestionario elaborado se hizo énfasis en las preguntas de tipo cerradas, pero la mayoría de ellas incluye la posibilidad de que el experto emita su criterio sobre el tema tratado, algo característico de las preguntas abiertas. Esta forma de relacionar ambos tipos (abiertas y cerradas) permitió hacer en estos casos análisis cuantitativos y cualitativos sobre la misma pregunta. El siguiente cuadro detalla los objetivos que se persiguen con cada tipo de pregunta:

Objetivos	Tipo de pregunta
Búsqueda de concordancia entre los expertos	Cerrada
Búsqueda de elementos comunes y esenciales en las valoraciones de los expertos	Abiertas

Tabla 2. Objetivos de los tipos de preguntas

Para la elaboración de las preguntas se tuvieron en cuenta cinco objetivos generales:

1. Demostrar la importancia de la detección temprana de errores y su implicación en la garantía de la calidad de un software a través de la planificación V&V, combinando la aplicación de mecanismos de control estáticos y dinámicos. Preguntas 1, 2, 3 y 7
2. Demostrar que las actividades propuestas son necesarias y suficientes para darle cumplimiento a los objetivos de los procesos de Verificación y Validación en el proyecto SIGEP. Pregunta 6
3. Demostrar que el plan es el resultado de la correcta adecuación del estándar IEEE 1012-1998 a las características del SIGEP. Preguntas 4, 5, 8

4. Demostrar que se hace una distribución correcta de roles, responsabilidades, métodos y procedimientos para darle cumplimiento a las tareas del plan. Preguntas 9 y 10
5. Demostrar que la propuesta constituye un aporte a la mejora y estandarización de los procesos de Verificación y Validación en el proyecto SIGEP. (Este objetivo se hace explícito en la pregunta 11, pero se encuentra implícito en todas las demás)

Los expertos que conforman el panel recibieron un resumen de la propuesta de solución como documentación primaria para responder los temas encuestados, además del cuestionario con un total de once preguntas con las características explicadas anteriormente. Se realizó una sola ronda de preguntas y luego se prosiguió a analizar los resultados.

3.4. Resultados del proceso de aplicación del método

La tabulación de los resultados se efectuó por cada uno de los objetivos descritos anteriormente, sobre la base de las evaluaciones positivas y negativas emitidas por los expertos. Se tabularon en gráficos que muestran la concordancia entre los expertos, donde por el eje “x” se tiene la cantidad de expertos que dieron una u otra respuesta (positiva o negativa) y por el eje “y” los objetivos específicos relacionados. Para el cumplimiento de estos últimos, la respuesta a la pregunta asociada debe ser evaluada positivamente por más del 75% de los expertos. A continuación se detallan los resultados:

Objetivo General # 1: Demostrar la importancia de la detección temprana de errores y su implicación en la garantía de la calidad de un software a través de la planificación V&V combinando la aplicación de mecanismos de control estáticos y dinámicos.

Se desglosa en cuatro objetivos específicos asociados a las preguntas 1, 2, 3 y 7, para las cuales las evaluaciones positivas y negativas se toman de la forma en que se muestra en la Tabla 3.

Pregunta	Objetivo específico relacionado	Evaluación Positiva	Evaluación Negativa
1	Corroborar la importancia de la detección temprana de errores durante el desarrollo de un software	Ponderación en valores de 4 y 5	Ponderación en valores de 1 a 3
2	Ratificar la importancia de aplicar revisiones en todas sus variantes para garantizar la calidad de un software	Ponderación en valores de 4 y 5	Ponderación en valores de 1 a 3
3	Demostrar la importancia de planificar V&V dentro de un proyecto de desarrollo de software	Respuesta de Si	Respuesta de No
7	Reafirmar la insuficiencia de solo aplicar pruebas en funciones V&V.	Respuesta de No	Respuesta de Si

Tabla 3. Evaluaciones positivas y negativas del Objetivo General # 1

El resultado luego de la tabulación de las respuestas de cada experto y la evaluación según la tabla anterior se muestra en el siguiente gráfico:



Figura 18. Evaluaciones de los expertos: Objetivo General #1

Objetivo General # 2: *Demostrar que las actividades propuestas son necesarias y suficientes para darle cumplimiento a los objetivos de los procesos de Verificación y Validación en el proyecto SIGEP.*

Se desglosa en dos objetivos específicos asociados a la pregunta 6 para la cual las evaluaciones positivas y negativas se toman de la forma en que se muestra en la Tabla 4.

Pregunta	Objetivo específico relacionado	Evaluación Positiva	Evaluación Negativa
6	Demostrar que las actividades propuestas son necesarias para cumplir los objetivos V&V en el SIGEP	Todas las actividades están ponderadas de 4 y 5	Al menos una actividad está ponderada de 1 a 3
6-a	Demostrar que las actividades propuestas son suficientes para cumplir los objetivos V&V en el SIGEP	Respuesta de No	Respuesta de Sí

Tabla 4. Evaluaciones positivas y negativas del Objetivo General # 2

El resultado luego de la tabulación de las respuestas de cada experto y la evaluación según la tabla anterior se muestra en el siguiente gráfico:

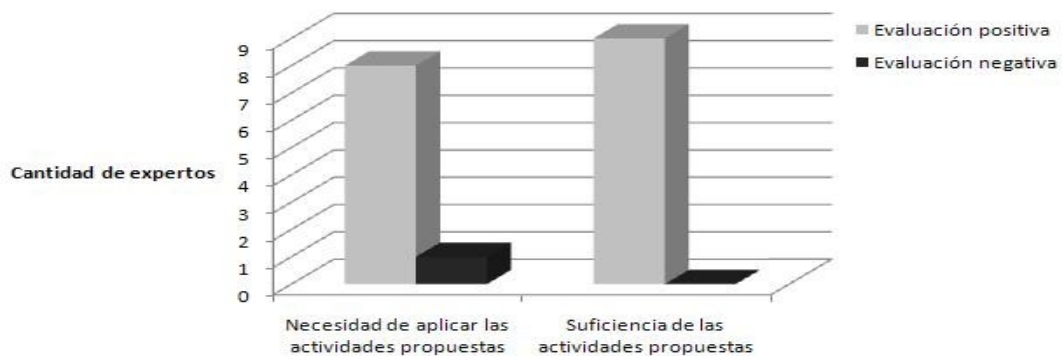


Figura 19. Evaluaciones de los expertos: Objetivo General #2

Objetivo General #3: *Demostrar que el plan es el resultado de la correcta adecuación del estándar IEEE 1012-1998 a las características del SIGEP*

Se desglosa en tres objetivos específicos asociados a las preguntas 4,5 y 8, para las cuales las evaluaciones positivas y negativas se toman de la forma en que se muestra en la Tabla 5

Pregunta	Objetivo específico relacionado	Evaluación Positiva	Evaluación Negativa
4	Demostrar que la guía de adopción de la norma es correcta	Respuesta de Adecuada	Respuesta de No Adecuada
5	Demostrar que la estructura del plan posibilita su correcta aplicación en el SIGEP	Respuesta de Sí	Respuesta de No
8	Demostrar que el cronograma tipo propuesto expresa claramente los aspectos necesarios de las actividades y tareas para la aplicación del plan en el SIGEP	Selección correcta de más de 2 aspectos	Selección de 2 o menos aspectos

Tabla 5. Evaluaciones positivas y negativas del Objetivo General # 3

El resultado luego de la tabulación de las respuestas de cada experto y la evaluación según la tabla anterior se muestra en el siguiente gráfico:

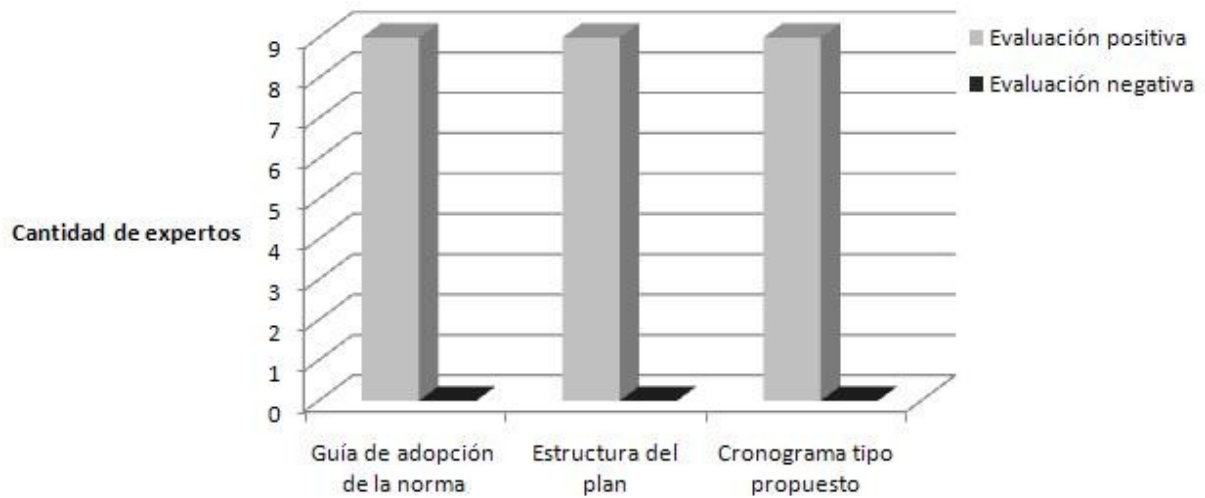


Figura 20. Evaluaciones de los expertos: Objetivo General #3

Objetivo General # 4: *Demostrar que se hace una distribución correcta de roles, responsabilidades, métodos y procedimientos para darle cumplimiento a las tareas del plan.*

Se desglosa en dos objetivos específicos asociados a las pregunta 9 y 10, para las cuales las evaluaciones positivas y negativas se toman de la forma en que se muestra en la Tabla 6.

Pregunta	Objetivo específico relacionado	Evaluación Positiva	Evaluación Negativa
9 y 10-b	Demostrar que la asignación de roles y responsabilidades es correcta para las tareas del plan.	Respuestas de Correcta y Aceptable	Respuesta de Incorrecta
10	Demostrar que la aplicación de métodos y procedimientos para darle cumplimiento a las tareas es correcta.	Respuestas de Correcta y Aceptable	Respuesta de Incorrecta

Tabla 6. Evaluaciones positivas y negativas del Objetivo General # 4

El resultado luego de la tabulación de las respuestas de cada experto y la evaluación según la tabla anterior se muestra en el siguiente gráfico:

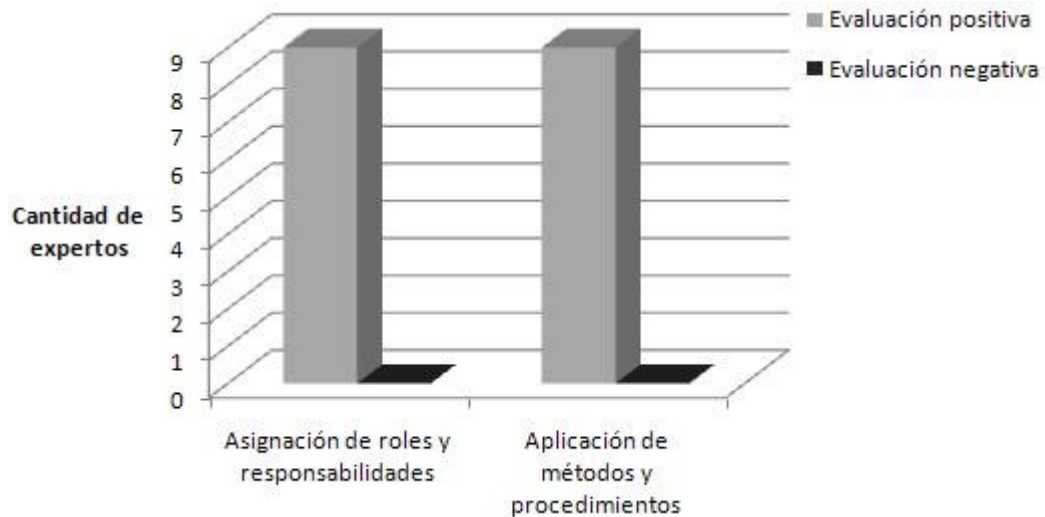


Figura 21. Evaluaciones de los expertos: Objetivo General #4

Objetivo General # 5: *Demostrar que la propuesta constituye un aporte a la mejora y estandarización de los procesos de Verificación y Validación en el proyecto SIGEP.*

Este objetivo como se explica anteriormente se hace explícito en la pregunta 11, pero tiene una dependencia parcial del cumplimiento de los cuatro objetivos generales que le preceden, la manera en que se toman las evaluaciones positivas y negativas se muestra en la Tabla 7.

Pregunta	Objetivo específico relacionado	Evaluación Positiva	Evaluación Negativa
11	Demostrar que la propuesta constituye un aporte a la mejora y estandarización de los procesos de Verificación y Validación en el proyecto SIGEP	Respuesta de Si	Respuesta de No

Tabla 7. Evaluaciones positivas y negativas del Objetivo General # 5

El resultado luego de la tabulación de las respuestas de cada experto y la evaluación según la tabla anterior se muestra en el siguiente gráfico:



Figura 22. Evaluaciones de los expertos: Objetivo General #5

Sobre los diferentes temas tratados en el cuestionario los expertos emitieron opiniones como las que se señalan a continuación:

- Es de suma importancia la detección de errores en etapas tempranas. Esto ayuda a que el producto termine con mejor calidad.
- La detección temprana de errores ahorra tiempo de corrección en otras etapas del desarrollo y permite que queden correctamente especificados los requisitos de acuerdo a las exigencias del cliente.
- Si se detectan los errores durante la etapa de desarrollo de un software se garantiza la calidad del producto final, así como menor tiempo y esfuerzo en la fase de mantenimiento.
- En la medida que se avanza en el proceso de desarrollo el costo de los errores aumenta de manera exponencial e implican inversión de tiempo en la solución del problema, que pudiera extenderse según el grado de complejidad del error, por lo que se retrasarían los cronogramas.
- Todas las actividades que se realizan en el desarrollo de un proyecto tienen que estar planificadas. Los procesos de Verificación y Validación han de estar planificados para que se realicen con la mejor calidad posible y no se afecte el cronograma del proyecto.
- Con una planificación objetiva y bien detallada se está asegurando un buen término y entrega en tiempo del producto a los clientes, y con la calidad requerida.
- La planificación de los procesos de Verificación y Validación garantiza obtener un producto final con calidad y con la cantidad mínima de no conformidades.

- Planificar la V&V es la forma de garantizar que el software que se está desarrollando tiene la calidad requerida durante todo el proceso de su desarrollo y no al final como ocurre en muchos casos.
- La adecuación de la norma IEEE 1012-1998 al proyecto es muy importante, ya que la IEEE es una organización que tiene bien definidos todos sus estándares y con gran reconocimiento a nivel internacional y esto crea confianza en la propuesta.
- La guía de adopción de la norma asegura la confección de un buen Plan V&V mediante la adecuación al proyecto del estándar IEEE 1012-1998.
- La estructura tiene un orden lógico y una secuencia que permite el entendimiento del plan.
- En la estructura del plan están desglosadas cada una de las actividades con sus tareas muy bien descritas y son especificados claramente los métodos que se utilizarán, las responsabilidades de los roles, así como las entradas que se requieren y las salidas esperadas.
- Resulta muy importante el cronograma tipo propuesto para conocer en qué momento del desarrollo se deben ejecutar las tareas, así como el orden de ejecución de las mismas.
- Siguiendo las tareas que se describen en el plan se asegura una buena Verificación y Validación del software.
- Las actividades propuestas están correctas, y se tendrá éxito si se implementan de forma adecuada.
- Si el plan se aplica al 100 % en el proyecto, el resultado debe ser satisfactorio y se recomienda que se extienda no solo a los proyectos de la facultad donde se desarrolla, sino a toda la universidad.
- Si se aplica el plan acorde a lo definido en esta estrategia, debe de cumplir su objetivo que es estandarizar los procesos de Verificación y Validación del proyecto productivo.
- A través de este plan el proyecto garantiza la calidad de su desarrollo y evita que el sistema y la documentación asociada llegue con errores a las pruebas de liberación y aceptación.

3.5. Otros resultados que validan la propuesta

La propuesta fue presentada en tres eventos científicos, en los cuales fue evaluada por jurados diferentes y se obtuvieron los siguientes resultados (Anexo H):

- Premio *Relevante* en la VI Edición de la Jornada Científica Estudiantil a nivel de Facultad.
- Premio *Relevante* en la VI Edición de la Jornada Científica Estudiantil a nivel de Universidad.
- Premio *Destacado* en la XVI Edición del FORUM de Ciencia y Técnica a nivel de Facultad.

3.6. Conclusiones parciales

La aplicación del cuestionario a los nueve expertos seleccionados arrojó resultados satisfactorios:

- Todas las preguntas fueron respondidas positivamente por más del 75% de los expertos.
- Los objetivos específicos asociados a cada pregunta fueron cumplidos.
- Los cinco objetivos generales a partir de los cuales se elaboraron las preguntas fueron cumplidos satisfactoriamente.

Se puede concluir que la propuesta fue validada por la totalidad de los miembros del panel, algo que corrobora las opiniones emitidas.

CONCLUSIONES

Como conclusiones generales de la investigación se muestran alcanzados los objetivos propuestos satisfactoriamente:

- *Identificar las actividades de Verificación y Validación que se realizan en el proceso de desarrollo del SIGEP, así como las posibles mejoras a estas.* Para este objetivo se realizó la tarea de identificar a partir de un estudio del proceso de desarrollo del proyecto SIGEP, qué actividades de Verificación y Validación se realizan y cómo se ejecutan. Las entrevistas realizadas, explicadas como parte de los métodos empíricos detallados en la Introducción del trabajo, así como la observación de los autores como miembros de experiencia en el proyecto, permitieron cumplir con este objetivo.
- *Identificar en la literatura y normas internacionales nuevas actividades y tareas de Verificación y Validación que se deben incorporar a las existentes en el SIGEP.* La tarea realizada para satisfacer este objetivo fue identificar los principales estándares y modelos que sirvieron de soporte a la propuesta de solución. Luego del estudio realizado y mostrado en el Capítulo I se definió basar la propuesta en el estándar 1012-1998, lo cual permitió identificar las nuevas actividades y tareas a incorporar según las características del proyecto.
- *Conformar una propuesta de Plan V&V que incluya buenas prácticas de Verificación y Validación ajustado a las características del proyecto.* Para darle cumplimiento a este objetivo, en el Capítulo I del trabajo investigativo se estudiaron a profundidad los procesos de Verificación y Validación y las principales definiciones dadas por diferentes autores, además de los métodos V&V que se aplican a lo largo del ciclo de vida de un software que se puedan ajustar al entorno del proyecto. Sobre la base de estos conocimientos y con el estudio del estándar seleccionado, en el Capítulo II se explicó la confección de la propuesta de Plan V&V.
- *Validar la propuesta de solución,* se realizó utilizando el Método Delphi donde a través de la tabulación de los resultados obtenidos de la aplicación del cuestionario al Panel de Expertos, proceso que se describe en el Capítulo III, se puede llegar a la conclusión de que fue cumplido satisfactoriamente el objetivo, por lo que se cuenta con un documento válido que estandariza y guía los procesos de Verificación y Validación en el proyecto.

Por lo anteriormente descrito se concluye el cumplimiento del objetivo general de la investigación de *elaborar un Plan V&V que facilite la introducción de buenas prácticas en los procesos de Verificación y Validación además de la estandarización de estos en el proyecto SIGEP.*

RECOMENDACIONES

No se puede olvidar que el objetivo fundamental de todo proyecto de software es entregar un producto de calidad que se ajuste a los requerimientos del cliente y que este quede satisfecho con la solución recibida. Tomando en cuenta esto y con la importancia que revisten los procesos de Verificación y Validación para el desarrollo de software, se recomienda lo siguiente:

- Completar el Plan para las etapas de Operación del Software y Mantenimiento del Software.
- Exigir el Plan V&V como documento obligatorio para todo proyecto productivo de la universidad.
- Realizar los ajustes necesarios al plan de manera que pueda ser aplicado en el resto de los proyectos de la facultad y la universidad.
- Capacitar a los equipos de calidad de los proyectos de la universidad sobre los temas de Verificación y validación de software.

REFERENCIAS BIBLIOGRÁFICAS

1. (ABRAN and MOORE 2004) ABRAN, A. and J. W. MOORE. *Guide to the Software Engineering Body of Knowledge*, 2004.
2. (ALS 2007) ALS, S. L. O. *Demostración a medida de una evaluación CMMI clase C*, 2007. [2008]. Disponible en: www.als-es.com/home.php?location=noticias_y_eventos/demo-evaluacion-cmmi
3. (ANTONIO 1999) ANTONIO, A. D. *Gestión, control y garantía de la calidad del software*, 1999.
4. (ARAGÓN 2003) ARAGÓN, S. *Método Delphi*, 2003. 2.
5. (ASTIGARRAGA 2004) ASTIGARRAGA, E. *El Método Delphi*, 2004. 14.
6. (BERMUDEZ and CORONA 2002) BERMUDEZ, G. S. and M. J. CORONA. *Diseño de un estándar de verificación y validación para asegurar la calidad del software*, 2002.
7. (BOOCH *et al.* 2000) BOOCH, G.; I. JACOBSON, *et al.* *El Proceso Unificado de desarrollo de software*. 2000. 458 p. 84-7829-036-2
8. (BRAVO and ARRIETA 2005) BRAVO, M. D. L. and J. J. ARRIETA. *El Método Delphi. Su implementación en una estrategia didáctica para la enseñanza de las demostraciones geométricas.: Revista Iberoamericana de Educación*, 2005. 11.
9. (CANTONE 2006) CANTONE, D. *La biblia del programador: Implementación y debugging: Claves, técnicas y herramientas para construir código sólido y confiable*. 2006. 320 p. 987-22995-7-9
10. (CARRILLO and RENDÓN 2007) CARRILLO, C. A. C. and J. P. G. RENDÓN. *Estrategia de apoyo a revisiones técnicas formales en el proceso de desarrollo de software*, 2007.
11. (CUESTA 2007) CUESTA, G. D. L. *Comunicación eficiente: parte esencial en los cambios empresariales*, 2007.
12. (FOUNDATION 2008) FOUNDATION, W. *Pruebas de Software*, 2008. [2008]. Disponible en: http://es.wikipedia.org/wiki/Pruebas_de_software
13. (GARZON 2007) GARZON, D. J. *Ciclo de vida del software*, 2007. [2008]. Disponible en: <http://darwinjimenezgarzon.blogspot.com>
14. (GROUP 2004) GROUP, S. *Chaos Report*, 2004.

15. (GUZMÁN *et al.* 2006) GUZMÁN, J. G.; A. D. A. SECO, *et al.* *TOP 10 de factores que obstaculizan la mejora de los procesos de verificación y validación en organizaciones intensivas en software*, 2006.
16. (HUMPHREY 2001) HUMPHREY, W. S. *Introducción al Proceso de Software Personal*, 2001.
17. (IEEE 1998a) IEEE. *Standard for Software Quality Assurance Plans. Std 730*, 1998a.
18. (IEEE 1997) ---. *Standard for Software Reviews. Std 1028*, 1997.
19. (IEEE 1998b) ---. *Standard for Software Verification and Validation. Std 1012*, 1998b.
20. (IEEE 1990) ---. *Standard Glossary of Software Engineering Terminology*, 1990.
21. (IEEE/EIA 1997) IEEE/EIA. *Industry Implementation of International Standard ISO/IEC 12207: 1995*, 1997.
22. (IEEE/EIA 1996) ---. *Industry Implementation of International Standard ISO/IEC 12207: 1995*, 1996.
23. (INSTITUTE 2006) INSTITUTE, S. E. *CMMI for Development, Version 1.2*. 2006. 573 p.
24. (KYNETIA 2007) KYNETIA. *Metodología de Pruebas*, 2007. [Disponible en: <http://www.kynetia.es/calidad/metodologia-de-pruebas.html>]
25. (LANDETA *et al.* 2003) LANDETA, J.; J. M. D. ANTONIO, *et al.* *Aplicación del Método Delphi en la elaboración de la tabla simétrica de las tablas input-output 2001 de Catalunya*, 2003. 28.
26. (LEBRÚN *et al.* 2008) LEBRÚN, C. V.; L. R. PRIETO, *et al.* *Mejores prácticas para el establecimiento y aseguramiento de la calidad del software*, 2008.
27. (LEON 2006) LEON, F. *Ingeniería del Software*, 2006. 22.
28. (MAÑAS 1994) MAÑAS, J. *Prueba de Programas*, 1994.
29. (MARTÍNEZ 2004) MARTÍNEZ, C. A. *La importancia de las Herramientas de V&V*, 2004.
30. (OLABUÉNAGA and URIBARRI 1989) OLABUÉNAGA, J. I. R. and M. A. I. URIBARRI. *La descodificación de la vida cotidiana: métodos de investigación cualitativa*. 1989. 241 p. 84-7485-126-2
31. (OSSORIO 2004) OSSORIO, M. *Diccionario de ciencias jurídicas, políticas y sociales*, 2004. 1008.

32. (PMBOK 2005) PMBOK. *Guía de los Fundamentos de la Dirección de Proyectos*, 2005.
33. (PRESSMAN 2002) PRESSMAN, R. S. *Ingeniería de Software. Un enfoque práctico*. 5ta. 2002. 614 p.
34. (QUISPE-OTAZU 2007) QUISPE-OTAZU, R. *Proceso de Desarrollo de Software*, 2007. [2008]. Disponible en: <http://www.rodolfoquispe.org/blog/proceso-de-desarrollo-de-software.php>
35. (SAFFIRIO 2005) SAFFIRIO, M. *Mi experiencia en la Ejecución de Proyectos en las Empresas* 2005.
36. (SÁNCHEZ 2006) SÁNCHEZ, P. F. *Aseguramiento de la Calidad del Software*, 2006. [2008]. Disponible en: <http://pfsanchez.blogspot.com/2006/08/aseguramiento-de-la-calidad-del.html>
37. (SEI 2008) SEI. *Published Appraisal Results*, Carnegie Mellon, 2008.
38. (TIAN 2005) TIAN, J. *Software Quality Engineering. Testing, Quality Assurance, and Quantifiable Improvement*, 2005.
39. (TORRES et al. 2005) TORRES, M.; C. DANDICOURT, et al. *Funciones de enfermería en la atención primaria de salud. Revista Cubana de Medicina General Integral*, 2005.
40. (URIZARRI and FERNÁNDEZ 2005) URIZARRI, L. A. R. and A. M. T. FERNÁNDEZ. *Algunas consideraciones acerca del método de evaluación utilizando el criterio de expertos*, 2005.
41. (WEITZENFELD 2006) WEITZENFELD, A. *Ingeniería de software orientada a objetos con Java e Internet*, 2006.
42. (ZAVALA-RUIZ 2008) ZAVALA-RUIZ. *Ingeniería de Software*, 2008.

GLOSARIO

A

Aseguramiento de la Calidad del Software :El aseguramiento de la calidad es la aplicación de actividades planificadas y sistemáticas relativas a la calidad, para asegurar que el proyecto emplee todos los procesos necesarios para cumplir con los requisitos, V

Auditoría: Es el examen independiente de un producto de trabajo o grupo de productos de trabajo para evaluar el cumplimiento con especificaciones, estándares, acuerdos contractuales u otro criterio., 18

C

Calidad: La capacidad de un grupo de características inherentes a un producto, componente o proceso de satisfacer los requerimientos del cliente., 1

Caso de prueba: Un conjunto de entradas de prueba, condiciones de ejecución, y los resultados esperados desarrollados para un objetivo particular., 21

Ciclo de vida del software: Es el período de tiempo que comienza cuando un producto de software es concebido y termina cuando ya no está disponible para su uso., 9

CMMI: Modelo Integrado de Madurez de la Capacidad., 8

Comprobación de escritorio (desk checking): Consiste en examinar a mano e individualmente el objeto que se acaba de desarrollar. Es el método tradicional para analizar un programa. Se debe aplicar a los requisitos, especificaciones de diseño y código según se van desarrollando., 17

Criterios de aceptación: Los criterios que un producto o componente del producto debe satisfacer a ser aceptado por un usuario, cliente, u otra entidad autorizada., 61

Criticidad: El grado de impacto que un requisito, módulo, error, fallo, u otro elemento tiene en el desarrollo o funcionamiento del sistema., 31

D

Diseño de Pruebas: Documentación que especifica los detalles del método de prueba para una función de software o una combinación de funciones de software y la identificación de las correspondientes pruebas., 106

E

Error: Acción humana que produce un resultado incorrecto. Por ejemplo, una acción incorrecta por parte de un programador u operador., 20

F

Falla: La incapacidad de un sistema o componente para desempeñar sus funciones requeridas dentro de los requerimientos de rendimiento específicos., 20

I

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos., V

IEEE 1012-1998: Estándar para la Verificación y Validación de Software., V

IEEE 1028-1997: Estándar para Revisiones al Software., 18

IEEE 610.12: Glosario de Terminologías de Ingeniería de Software., 10

Inspección: Es un examen visual de un producto de software para detectar e identificar anomalías, incluyendo errores y desviaciones de los estándares y especificaciones. Las inspecciones son exámenes por pares dirigido por facilitadores imparciales que están capacitados en técnicas de inspección., 19

ISO/IEC 12207: Procesos del Ciclo de Vida del Software., 8

IV&V: Verificación y Validación Independiente., 15

P

Plan V&V: Es un plan que guía el comportamiento de la verificación y la validación en un software., V

PMBOK: Fundamentos de la Gestión de Proyectos., 12

Práctica específica (CMMI): Un modelo de componentes que es considerado importante en el logro de la meta específica asociada. La práctica específica describe las actividades previstas en el logro de los objetivos específicos de un área de proceso., 28

Procedimientos de Prueba: Las instrucciones detalladas para la puesta en marcha, ejecución y evaluación de resultados para un determinado caso de prueba., 106

Proceso: Un proceso (del latín processuss) es un conjunto de actividades o eventos que se realizan o suceden con un determinado fin., 36

Proceso de Desarrollo: Es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto., V

Prueba de software: El proceso de analizar un software para detectar las diferencias entre las condiciones existentes y las requeridas además evaluar las características del sistema., 20

R

Revisión: Es un proceso o una reunión durante la cual un producto de software es presentado al personal de proyectos, gerentes, usuarios, clientes, representantes de los usuarios, u otras partes interesadas para que formulen sus observaciones o lo aprueben., 18

Revisión por pares o iguales (peer review): Es la evaluación científica en la búsqueda o propósitos para la competencia y originalidad por expertos calificados que buscan crear productos de la misma forma., 17

Revisión técnica: Es una evaluación sistemática de un producto de software por un equipo de personal calificado, que examina la idoneidad del producto de software para su uso e identifica discrepancias con las especificaciones y estándares. Las revisiones técnicas también pueden proporcionar recomendaciones de alternativas y el examen de las distintas alternativas., 19

RUP: Proceso Unificado de Desarrollo., 9

S

SEI: Instituto de Ingeniería de Software., 27

SIGEP: Sistema de Gestión Penitenciaria., V

V

Validación: El proceso de evaluar un sistema o componente durante o al final del proceso de desarrollo para determinar si este satisface los requerimientos especificados., V

Verificación: El proceso de evaluar un sistema o componente para determinar si los productos de una fase satisfacen las condiciones impuestas al inicio de la misma., V

Verificación y Validación (V&V): El proceso de determinar si los requisitos para un sistema o componente son completos y correctos, los productos de cada fase de desarrollo cumplen con los requisitos o condiciones impuestas por la fase anterior, y el sistema o componente final cumple con los requisitos especificados., 13

W

Walk-through: Es una técnica de análisis estático en el cual el diseñador o programador guía a miembros del equipo de desarrollo y otras partes interesadas a través del producto de software, los participantes realizan preguntas y comentarios acerca de posibles errores, violación de los estándares de desarrollo y otros problemas., 19

ANEXOS

Anexo A: Plan V&V SIGEP



Proyecto para la Humanización Penitenciaria

Sistema de Gestión Penitenciaria. Solución de Software.

PLAN DE VERIFICACIÓN Y VALIDACIÓN

1. Propósito

Organizar y estandarizar los procesos de Verificación y Validación (V&V) en el Proyecto Sistema de Gestión Penitenciaria (SIGEP), planificando las actividades y tareas V&V a realizar durante la fase de Desarrollo del ciclo de vida del software. Esta fase incluirá desde el modelado del negocio hasta la instalación y aceptación del producto final.

2. Documentos referenciados

No existen documentos referenciados.

3. Definiciones y Abreviaturas

Completitud: (1) Nivel o grado en que algo está completo. (2) El estado de estar completo y entero.

Comprobabilidad: (1) El grado en que un sistema o componente facilita es establecimiento de criterios de prueba y la ejecución de las pruebas para determinar si esos criterios han sido satisfechos. (2) El grado en que un requerimiento es indicado en términos de permitir el establecimiento de criterios de prueba y la ejecución de las pruebas para determinar si esos criterios han sido satisfechos.

Consistencia: El grado de uniformidad, estandarización, y libertad de contradicciones entre la documentación o parte de un sistema o componente.

Corrección: (1) El grado en que un sistema o componente está libre de fallas en su especificación, diseño e implementación. (2) El grado en que el software, documentación u otro producto cumplen con sus requerimientos específicos.

Exactitud: (1) Una evaluación cualitativa de la corrección, de la libertad de errores. (2) Una medida cuantitativa de la magnitud del error.

Legibilidad: La calidad de la escritura que hace que sea fácil de leer y entender

Trazabilidad: El grado en que una relación puede ser establecida entre dos o más productos del proceso de desarrollo, especialmente productos que tienen relaciones predecesor-sucesor o maestro subordinado de uno al otro; por ejemplo el grado en los requerimientos y el diseño u otro componente del software se relacionan.

DDF: Documento de Descripción de Funcionalidades

DNC: Documento de No Conformidades

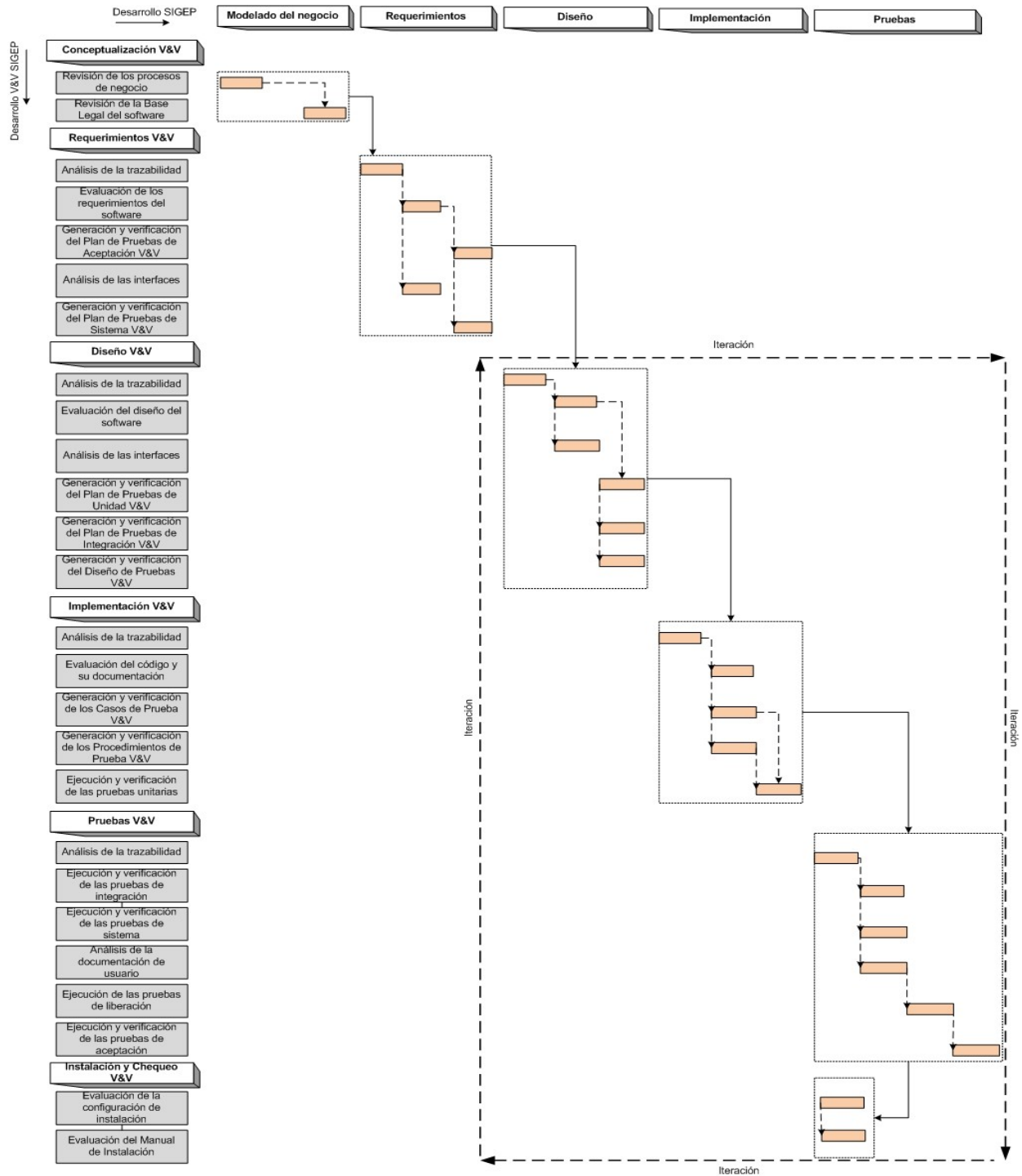
DP: Documento de Pruebas definido para el proyecto

DTN: Descripción Textual del Negocio

V&V: Verificación y Validación

4. Vista general de V&V

4.1. Cronograma general



4.2. Esquema de niveles de integridad

La tabla 1 muestra el esquema de niveles de integridad adoptado para determinar el nivel de integridad del proyecto.

Nivel	Dimensión del daño por fallo del software	Mitigación posible
4	<ul style="list-style-type: none"> – Pérdidas de vidas humanas – Pérdida de la seguridad del sistema – Graves pérdidas económicas o sociales 	No es posible mitigar los daños producidos
3	<ul style="list-style-type: none"> – El sistema no completa el objetivo que tiene previsto. – Graves pérdidas económicas o sociales 	Es posible una mitigación parcial de los daños
2	<ul style="list-style-type: none"> – El sistema incumple parcialmente con el objetivo que tiene previsto. – Pérdidas sociales o económicas importantes. 	Se pueden mitigar los daños producidos
1	<ul style="list-style-type: none"> – Una determinada función no se realiza. – Consecuencias mínimas 	No es necesario mitigar los daños.

Tabla 1. Niveles de integridad de software

Los niveles de integridad asociados a cada módulo y subsistema del proyecto se muestran en el Anexo B.

4.3. Recursos

Los recursos humanos con que cuenta el proyecto para realizar las tareas V&V se muestran en la tabla que se presenta a continuación:

Rol	Cantidad (Personas)
Analista	10
Arquitecto	3
Ayuda	2
Bases de Datos	10
Calidad	9
Diseñador	5
Diseñador gráfico	7

Líder de proyecto	1
Portal Web	5
Programador Acceso a Datos	9
Programador Interfaz de Usuario	22
Sala Situacional	5
Total	88

Tabla 2. Recursos humanos del proyecto SIGEP

Otros roles implicados en las tareas V&V son:

Rol	Cantidad (Personas)
Especialista de procesos	1
Cliente	Al menos 1 para cada tarea en que participa

Tabla 3. Otros roles implicados en las tareas de V&V

Los recursos tecnológicos del proyecto se agrupan en tres laboratorios de producción en los cuales se distribuye el total de personas.

	Capacidad (máquinas)	Ocupadas (máquinas)	Cantidad (Personas)
Laboratorio 1	30	30	39
Laboratorio 2	31	31	39
Laboratorio 3	20	10	10
	81	71	88

Tabla 4. Recursos tecnológicos del proyecto SIGEP

Recurso	Cantidad
Cámara fotográfica digital	1
Escáner de huellas	1
Impresora	1
Escáner	1

Tabla 5. Recursos tecnológicos adicionales

4.4. Responsabilidades

La asignación de responsabilidades se hace teniendo en cuenta que el primer esfuerzo V&V lo realiza el desarrollador (analistas, diseñadores, documentadores, programadores, en dependencia de la actividad que se esté realizando), que debe garantizar la calidad del proceso que ejecuta y del

producto que se obtiene del mismo. El equipo de calidad interna del proyecto, en funciones de revisor, diseñador de pruebas, probador, está presente en todas las actividades del desarrollo y jugando su papel de contraparte obtiene un nivel de conocimientos vital para ejecutar de la manera más efectiva las pruebas y revisiones planificadas. La descripción detallada de las funciones precisas de cada rol con fines V&V se realiza dentro de cada tarea.

5. Procesos V&V

5.1. Proceso de Desarrollo

El proceso de desarrollo contiene las actividades y tareas del desarrollador. Se incluyen las actividades referentes a la modelación del negocio, análisis de requerimientos, diseño, codificación, pruebas, instalación y aceptación de los productos de software. Estas actividades están organizadas en Conceptualización V&V, Requerimientos V&V, Diseño V&V, Implementación V&V, Pruebas V&V e Instalación y Chequeo V&V

5.1.1. Actividad: Conceptualización V&V

Durante la actividad de Conceptualización V&V se sientan las bases de todo el respaldo legal que debe tener el software, se identifican los procesos de negocio de la organización que se describen textualmente para una mejor comprensión de los mismos. Los esfuerzos V&V están encaminados fundamentalmente a verificar y validar que los procesos de negocio identificados reflejan correctamente el funcionamiento de la organización y que fueron documentados adecuadamente.

5.1.1.1. Tarea: Revisión de los procesos de negocio

Descripción

Se evalúa el modelado de negocio en cuanto a completitud y corrección

Completitud

Verificar y validar que los procesos identificados modelan la estructura y funcionamiento completo de la organización.

Corrección

Verificar que los procesos identificados están descritos correctamente y no se viola ningún paso dentro del flujo que describe la dinámica de los mismos.

Métodos y procedimientos.

Los diagramas de actividades del negocio, obtenidos a partir del entendimiento del negocio, que se realiza partiendo de los procesos identificados son sometidos a una *prueba de validación* con el cliente venezolano.

De la misma manera la DTN se somete a una *prueba de validación* con el cliente, que valida y firma la propuesta quedando formalizado su total acuerdo con lo planteado en la misma.

Roles y Responsabilidades

Especialista de procesos: Identifica y describe muy brevemente los procesos de negocio.

Analista de negocio: Lleva a cabo la comprensión de todos los procesos de negocio y los describe en detalle, especificando el flujo de actividades de cada uno de ellos. De esto se obtienen los Diagramas de Actividades del Negocio y la DTN.

Cliente: Valida los Diagramas de Actividades del Negocio y la DTN.

Líder de proyecto: Participa en las pruebas de validación con el cliente como máximo responsable de la solución de software.

Entradas requeridas

Diagramas de Actividades del Negocio

DTN

Salidas requeridas

Reporte de tarea

DNC o Documento de Petición de Cambios.

5.1.1.2. Tarea: Revisión de la Base Legal del software

Descripción

Evaluar la Base Legal del software en cuanto a corrección, completitud y legibilidad.

Corrección

Verificar que se exprese correctamente la correspondencia entre los procesos definidos y los artículos de las leyes, normativas y reglamentos del sistema penitenciario.

Completitud

Verificar que se explica detalladamente cómo se aplica la ley para cada proceso definido.

Legibilidad

Verificar que el documento ha sido escrito cumpliendo con el formato establecido para el proyecto, en un vocabulario adecuado y claro.

Métodos y procedimientos

Se aplica un *walk-through* que será desarrollado en una o varias reuniones de trabajo entre el equipo de analistas, un miembro del equipo de calidad y el Líder de proyecto, donde realizarán la evaluación de la Base Legal en cuanto a corrección y completitud.

Se aplica una *revisión técnica* para evaluar la legibilidad.

Roles y responsabilidades

Analista de negocio: Realiza la exposición en el walk-through.

Líder de proyecto: Participa en el walk-through.

Revisor Técnico: Participa en el walk-through y realiza la revisión.

Entradas requeridas

Leyes, normativas y reglamentos vigentes del Sistema Penitenciario

Base Legal

DTN

Salidas requeridas

Reporte de tarea

Acta de cada reunión realizada.

5.1.2. Actividad: Requerimientos V&V

En la actividad de Requerimientos V&V se definen los requerimientos funcionales y no funcionales del software, las interfaces, se construye la primera versión de la base de datos. La actividad de Requerimientos V&V dirige el análisis de requerimientos del software. El objetivo fundamental de los esfuerzos V&V es asegurar la corrección, completitud, exactitud, comprobabilidad y consistencia de los requerimientos.

5.1.2.1. Tarea: Análisis de la trazabilidad

Descripción

Evaluar las relaciones entre las funcionalidades descritas en el DDF y los procesos de negocio detallados en la DTN en cuanto a corrección, consistencia y completitud.

Corrección

Validar que las relaciones entre funcionalidades y procesos de negocio identificados son correctas.

Consistencia

Verificar que las relaciones entre cada funcionalidad con los procesos de negocio correspondientes están especificadas detalladamente.

Completitud

Verificar que cada funcionalidad descrita es traceable con alguno de los procesos de negocios identificados.

Métodos y procedimientos

Se aplica una *revisión técnica* en la cual se evalúan los criterios expuestos en la descripción de la tarea.

Roles y Responsabilidades

Revisor Técnico: Realiza la revisión

Analista de negocio: Participa en el proceso de revisión.

Entradas requeridas

DDF

DTN

Salidas requeridas

Reporte de tarea

Análisis de la trazabilidad.

5.1.2.2. Tarea: Evaluación de los requerimientos del software

Descripción

Evaluar los requerimientos del software en cuanto a corrección, consistencia, completitud, legibilidad y comprobabilidad.

Corrección

Verificar que todos los requerimientos plasmados en el *Documento Visión* estén incluidos en el DDF.

Verificar y validar que los requerimientos del software satisfacen las necesidades del usuario.

Verificar que los requerimientos cumplen con los estándares, referencias, regulaciones, políticas, leyes gubernamentales y reglas del negocio.

Consistencia

Verificar que todos los términos y conceptos son documentados con el suficiente nivel de detalle y claridad, sin entrar en contradicciones internas.

Completitud

Verificar que el DDF incluye todos los requerimientos correctamente descritos, teniendo en cuenta para cada funcionalidad:

- Precondiciones y postcondiciones
- Valores permitidos para cada campo
- Campos requeridos
- Prototipo de interfaz

Legibilidad

Verificar que la documentación es legible, entendible y sin ambigüedades para la audiencia requerida.

Verificar que la documentación define todas las siglas, abreviaturas, términos y símbolos.

Verificar que la documentación ha sido escrita cumpliendo con el formato para la documentación definido para el proyecto. (Anexo K)

Comprobabilidad

Verificar y validar que existe un proceso concreto y finito que permita comprobar si se cumple el requisito.

Métodos y procedimientos

Se aplica una *revisión técnica* para evaluar los criterios expuestos en la descripción de la tarea.

Se realiza una *prueba de validación* con el cliente para garantizar la corrección de los requisitos.

Es vital que prevalezca el trabajo en equipo entre todos los Analistas del negocio y el Líder de proyecto, para que se pueda verificar internamente la calidad de todas las funcionalidades identificadas.

Siempre que surjan no conformidades se llevará a cabo un proceso de refinamiento de los requisitos para que estos vuelvan a ser sometidos a la revisión y a la prueba de validación.

Roles y Responsabilidades

Analista de negocio: Identifica y describe las funcionalidades de sus módulos verificando constantemente su trabajo en cuanto a los criterios plasmados en la descripción de la tarea.

Cliente: Valida los requerimientos descritos en el DDF, verifica que satisfacen sus necesidades y que cumplen con los estándares, referencias, regulaciones, políticas, leyes gubernamentales y reglas del negocio

Líder de proyecto: Utilizando sus amplios conocimientos sobre el negocio verifica constantemente el proceso de identificación y descripción de las funcionalidades, participando en todas las decisiones que deben ser tomadas por el equipo de análisis en función de esto. Participa en la prueba de validación.

Revisor Técnico: Lleva a cabo la revisión técnica, entrega las no conformidades detectadas y repite la revisión a las funcionalidades refinadas.

Entradas requeridas

DDF

Documento Visión

Estándares, leyes, regulaciones y políticas relacionadas con el trabajo del proyecto

Reglas del negocio

Salidas requeridas

Reporte de tarea

DNC

5.1.2.3. Tarea: Generación y verificación del Plan de Pruebas de Aceptación V&V.

Descripción

Se genera el Plan de Pruebas de Aceptación y se verifica que este cumpla con el formato, propósito y contenido del Documento de Pruebas definido para el proyecto. Se valida que el plan satisface los siguientes criterios: 1) las pruebas programadas cubren los requerimientos del software, 2) el uso apropiado de estándares y métodos de prueba, 3) conformidad con los resultados esperados, 4) capacidad de ser implementado y mantenido en correspondencia con las necesidades del usuario.

Métodos y procedimientos

El responsable de pruebas de aceptación junto a otro miembro del Equipo de Calidad realiza un *peer review* durante la generación del Plan de Pruebas. Este trabajo permite que, en paralelo con la obtención del plan, se ejecute la revisión del mismo.

Roles y responsabilidades

Responsable de pruebas de aceptación (miembro del Equipo de Calidad): Participa en el peer review.

Revisor Técnico: Participa en el peer review.

Líder de proyecto: Aprueba el Plan

Entradas requeridas

DDF

DP

Salidas requeridas

Reporte de tarea

Plan de Pruebas de Aceptación V&V

5.1.2.4. Tarea: Análisis de las interfaces

Descripción

Evaluar las interfaces obtenidas a partir de la identificación de las funcionalidades en cuanto a corrección, completitud y exactitud.

Corrección

Verificar y validar que todos los requerimientos del software son reflejados adecuadamente en las pantallas definidas.

Compleitud

Verificar que cada interfaz es descrita e incluye el detalle de cada funcionalidad que contiene.

Exactitud

Verificar que cada interfaz provee la información correctamente organizada, clara y concisa para su posterior uso dentro del desarrollo y comprensión por parte del usuario.

Métodos y procedimientos

Es indispensable la verificación y validación interna en el proceso de obtención de las pantallas que es garantizada por el trabajo unido de Analistas y Líder del proyecto, por el hecho de que permite la obtención de un producto más acabado, adecuado al problema que se intenta resolver y por tanto con mayores posibilidades de ser aprobado exitosamente por la parte cliente.

Se realiza una *prueba de validación* con el cliente venezolano para el cual debe haber una explicación previa de cada pantalla y la entrega de la descripción textual de las mismas. El cliente debe revisar la corrección, completitud y exactitud de la pantalla y en función de eso validarla, expresar sus no conformidades y/o peticiones de cambio.

Roles y responsabilidades

Analista de negocio: Genera y describe las pantallas.

Líder de proyecto: Revisa internamente el proceso de obtención de las pantallas, así como su descripción.

Cliente: Revisa y valida el DDF que contiene las pantallas con el detalle correspondiente a cada una.

Entradas requeridas

DDF

Salidas requeridas

Reporte de tarea

DNC y/o Petición de cambios

5.1.2.5. Tarea: Generación y verificación del Plan de Pruebas de Sistema V&V

Descripción

Se genera el Plan de Pruebas de Sistema y se verifica que este cumpla con el formato, propósito y contenido del Documento de Pruebas definido para el proyecto. Se valida

que el plan satisface los siguientes criterios: 1) las pruebas programadas cubren los requerimientos del software, 2) el uso apropiado de estándares y métodos de prueba, 3) conformidad con los resultados esperados, 4) capacidad de ser implementado y mantenido.

Métodos y procedimientos

El responsable de pruebas de sistema, junto a otro miembro del Equipo de Calidad realiza un *peer review* durante la generación del Plan de Pruebas. Este trabajo permite que en paralelo con la obtención del Plan se ejecute la revisión del mismo.

Roles y responsabilidades

Responsable de pruebas de sistema (miembro del Equipo de Calidad): Participa en el peer review

Revisor Técnico: Participa en el peer review

Líder de Proyecto: Aprueba el Plan.

Entradas requeridas

DDF

DP

Salidas requeridas

Reporte de tareas

Plan de Pruebas de Sistema V&V

5.1.3. Actividad: Diseño V&V

En la actividad de diseño los requerimientos del software son transformados en arquitectura y diseño detallado para cada componente del software. El diseño incluye bases de datos (entidades de dominio), interfaces de acceso a datos, interfaces de negocio y además la interfaz gráfica. Esta actividad guía el análisis y diseño de la solución. El objetivo fundamental de la V&V es demostrar que el diseño es una correcta, precisa y completa transformación de los requerimientos del software y que no se han introducido errores.

5.1.3.1. Tarea: Análisis de la trazabilidad.

Descripción

Verificar que el diseño incluye todos los requerimientos del software, de manera tal que:

- Para todos los requerimientos que impliquen persistencia de datos se hayan diseñado las entidades pertinentes.

- Cada interfaz de negocio y acceso a datos diseñada esté en correspondencia con algún requerimiento del software.
- Cada componente gráfico diseñado está asociado a un requisito determinado.

Métodos y procedimientos

Es obligatorio que el análisis de la solución, que es el proceso donde se diseñan todas las interfaces de negocio y acceso a datos, donde se determinan las entidades de dominio y los componentes gráficos de cada pantalla, además de la navegación entre las pantallas, sea el resultado del trabajo conjunto del Analista de negocio, Diseñador del módulo, Diseñador de Base de Datos, con la presencia de un miembro del Equipo de Calidad. Esta forma de trabajo permite que se diseñe y revise la solución al mismo tiempo, siendo la mejor variante para garantizar la trazabilidad de los elementos de diseño a los requisitos y viceversa, además de la robustez del diseño planteado.

Roles y responsabilidades

Analista de negocio: Explica todas las funcionalidades del modulo, enuncia los requisitos funcionales, precisa cuales son los datos persistentes y como deben ser presentados en la interfaz de usuario

Diseñador: Diseña las clases, detecta las posibles clases del modelo objetual, sus posibles relaciones y cómo estas están representadas en la base de datos

Diseñador de Base de Datos: Presenta la base de datos ya elaborada, la representación relacional de los datos y precisa que tipo de valores soporta cada campo

Revisor: Participa en el análisis de la solución y confecciona el *Análisis de la trazabilidad*

Entradas requeridas

DDF

Base de datos

Modelo de datos relacional

Salidas requeridas

Reporte de tareas

Análisis de la trazabilidad

Propuesta de solución

5.1.3.2. Tarea: Evaluación del diseño del software

Descripción

Se evalúa el diseño del software en cuanto a corrección, completitud y consistencia.

Corrección

Verificar y validar que el diseño del software se ajusta a las especificaciones de la arquitectura definida para el proyecto:

- Que cumpla con las convenciones de código establecidas.
- Que se ajuste a la estructura de paquetes.
- Que estén bien asignadas las responsabilidades de cada capa arquitectónica.

Compleitud

Verificar que están definidas todas las funcionalidades necesarias para satisfacer cada caso de uso.

Consistencia

Verificar que las funcionalidades con objetivos similares que puedan ser parametrizadas en una sola no estén separadas.

Verificar que están definidas las funcionalidades comunes para que sean implementadas solamente en un módulo.

Verificar que se han identificado todas las dependencias entre los módulos.

Métodos y procedimientos

Se aplica un *walk-through* donde en una o varias sesiones de trabajo que tendrán lugar obligatoriamente con todos los diseñadores del proyecto donde cada diseñador explica al resto la propuesta de solución que tiene para su módulo y en función de esto se toman todas las decisiones pertinentes para obtener un diseño correcto, completo y consistente.

Se aplica una *revisión técnica* por parte del arquitecto de software que debe indicar los errores y explicar la forma idónea de proceder para solucionar cada uno.

Roles y responsabilidades

Diseñador: Construye el diseño del software y participa en el walk-through

Arquitecto de software: Lleva a cabo la revisión e interviene en el diseño de la solución de ser necesario.

Entradas requeridas

Resultado del Análisis de la solución.

Convenciones de código establecidas en el proyecto

Salidas requeridas

Reporte de tarea

Diseño correcto de la solución.

5.1.3.3. Tarea: Análisis de las interfaces.

Descripción

Verificar y validar que las interfaces han sido diseñadas ajustándose a las pautas de diseño establecidas por el proyecto. Tener en cuenta: color, tipo y tamaño de la letra.

Verificar que todos los mensajes lanzados por el sistema cumplen con el estándar definido por el proyecto según el sentido del mensaje.

Métodos y procedimientos

Se aplica una *Inspección* a las interfaces gráficas verificando los criterios que se exponen en la descripción, haciendo uso de una lista de chequeo definida con anterioridad. (Anexo J)

Roles y responsabilidades

Diseñador gráfico: Diseña todas las pantallas del software.

Revisor Técnico: Ejecuta la inspección a todas las interfaces.

Entradas requeridas

Interfaces gráficas

Lista de chequeo de funcionalidades

Salidas requeridas

Reporte de tarea

DNC

5.1.3.4. Tarea: Generación y verificación del Plan de Pruebas de Unidad V&V

Descripción

Se genera el Plan de Pruebas de Unidad V&V y se verifica que este cumpla con el formato, propósito y contenido del Documento de Pruebas definido para el proyecto. Se valida que el Plan satisface los siguientes criterios: 1) las pruebas programadas cubren las unidades definidas, 2) traceable a los requerimientos del software y el diseño, 3) consistencia externa con los requerimientos del software y el diseño 4) capacidad de ser implementado y mantenido acorde a las necesidades de los usuarios.

Métodos y procedimientos

El responsable de pruebas de unidad junto a otro miembro del Equipo de Calidad realiza un *peer review* durante la generación del Plan de Pruebas. Este trabajo permite que en paralelo con la obtención del Plan se ejecute la revisión del mismo.

Roles y responsabilidades

Responsable de pruebas de unidad (miembro del Equipo de Calidad): Participa en el peer review

Revisor Técnico: Participa en el peer review.

Líder de proyecto: Aprueba el Plan.

Entradas requeridas

DDF

DP

Salidas requeridas

Reporte de tarea

Plan de de Pruebas de Unidad V&V

5.1.3.5. Tarea: Generación y verificación del Plan de Pruebas de Integración V&V

Descripción

Se genera el Plan de Pruebas de Integración y se verifica que este cumpla con el formato, propósito y contenido del Documento de Pruebas definido para el proyecto. Se valida que el Plan satisface los siguientes criterios: 1) traceable a los requerimientos del software, 2) consistencia externa con los requerimientos del software, 3) consistencia interna, 4) las pruebas programadas cubren los requerimientos del software, 5) uso correcto de estándares y métodos de pruebas, 6) conformidad con los resultados esperados, 7) capacidad para ser implementado y mantenido acorde a las necesidades del usuario.

Métodos y procedimientos

El responsable de pruebas de integración junto a otro miembro del Equipo de Calidad realiza un *peer review* durante la generación del Plan de Pruebas. Este trabajo permite que en paralelo con la obtención del Plan se ejecute la revisión del mismo.

Roles y responsabilidades

Responsable de pruebas de integración (miembro del Equipo de Calidad): Participa en el peer review.

Revisor Técnico: Participa en el peer review.

Líder de Proyecto: Aprueba el Plan

Entradas requeridas

DDF

DP

Documento de descripción de la arquitectura

Salidas requeridas

Reporte de tarea

Plan de Pruebas de Integración V&V

5.1.3.6. Tarea: Generación y verificación del diseño de las pruebas V&V

Descripción

Se diseñan las pruebas y se verifica que cumplan con el formato, propósito y contenido del Documento de Pruebas definido para el proyecto. Se valida que el diseño satisface los siguientes criterios:

- Unidad
Criterios definidos en la Tarea 5.1.3.4
- Integración
Criterios definidos en la Tarea 5.1.3.5
- Sistema
Criterios definidos en la Tarea 5.1.2.5
- Aceptación.
Criterios definidos en la Tarea 5.1.2.3

Métodos y procedimientos

Se aplica una *revisión técnica* al diseño de las pruebas evaluando los criterios para cada uno de los niveles, en el orden en que aparecen.

Roles y responsabilidades

Diseñador de pruebas: Genera el diseño de las pruebas V&V

Jefe del Equipo de Calidad: Revisa y aprueba el diseño

Entradas requeridas

DP

Planes de Prueba

Salidas requeridas

Reporte de tarea

Diseño de las Pruebas de Unidad V&V

Diseño de las Pruebas de Integración V&V

Diseño de las Pruebas de Sistema V&V

Diseño de las Pruebas de Aceptación V&V

5.1.4. Actividad: Implementación V&V

En la actividad de implementación se transforma el diseño en código, estructuras de bases de datos y las representaciones ejecutables relacionadas. La actividad Implementación V&V dirige la codificación del software. Los objetivos de V&V son verificar y validar que estas transformaciones son correctas, precisas y completas.

5.1.4.1. Tarea: Análisis de la trazabilidad

Descripción

Tracear los componentes del código fuente a los elementos de diseño y los elementos de diseño hacia el código fuente. Analizar las relaciones identificadas en cuanto a corrección, consistencia y completitud.

Corrección

Validar las relaciones entre los componentes del código fuente y los elementos de diseño.

Consistencia

Verificar que las relaciones entre los componentes del código fuente y los elementos de diseño están especificadas a un nivel de detalle consistente.

Completitud

Verificar que todos los componentes de código fuente son traceables desde los elementos de diseño.

Verificar que todos los elementos de diseño son traceables desde los componentes de código fuente.

Métodos y procedimientos

Se aplica una *revisión técnica* para evaluar las relaciones entre los elementos de diseño y el código fuente según los criterios detallados en la descripción de la tarea.

Roles y responsabilidades

Diseñador: Ejecuta la revisión

Revisor Técnico: Participa en la revisión y elabora el Análisis de la trazabilidad.

Entradas requeridas

Elementos de Diseño

Código fuente

Salidas requeridas

Reporte de tarea

Análisis de la trazabilidad.

5.1.4.2. Tarea: Evaluación del código y su documentación

Descripción

Se evaluará el código (y la documentación relativa a este) generado en cuanto a corrección, consistencia, legibilidad y completitud

Corrección

Se verifica que los componentes del código fuente cumplen con los estándares, regulaciones y políticas establecidas en el proyecto (convenciones de nombres).

Verificar que se codifique respetando la estructura de paquetes global y para cada módulo.

Consistencia

Se verifica que exista la documentación adecuada para todo el código generado teniendo en cuenta las peculiaridades para cada componente:

- Interfaces (DAO, Manager, Fachadas)
- Recursos (XML y ficheros *.properties*)

Legibilidad

Verificar que la documentación es legible, entendible y sin ambigüedades.

Completitud

Verificar que en el código fuente están presentes elementos como: funcionalidad (algoritmos, validaciones, manejo de excepciones)

Métodos y procedimientos

Se realiza un *peer review* por parte del implementador del código que se revisa junto a otro programador de su equipo de desarrollo para evaluar los criterios de corrección, consistencia y completitud.

Se aplica una *inspección* al código por parte del arquitecto de software.

Se aplica una *revisión técnica* para evaluar el criterio de legibilidad.

Roles y responsabilidades

Programador: Genera el código y participa en el peer review.

Revisor Técnico: Ejecuta la revisión de la documentación.

Diseñador: Supervisa el peer review.

Arquitecto de software: Ejecuta la inspección.

Entradas requeridas

Código fuente

Estándares de codificación

Convenciones de nombres definidos en el proyecto (Anexo L)

Salidas requeridas

Reporte de tarea

DNC

5.1.4.3. Tarea: Generación y verificación de los Casos de Prueba V&V.

Descripción

Generar los casos de prueba teniendo en cuenta todos los escenarios de cada caso de uso y diseñando los juegos de datos correspondientes. Se verifica que estos cumplan con el formato, propósito y contenido del Documento de Pruebas definido para el proyecto. (Anexo M) Se valida que los casos de prueba satisfacen los siguientes criterios:

- Unidad
Criterios definidos en la Tarea 5.1.3.4
- Integración
Criterios definidos en la Tarea 5.1.3.5
- Sistema
Criterios definidos en la Tarea 5.1.2.5
- Aceptación.
Criterios definidos en la Tarea 5.1.2.3

Métodos y procedimientos

Se aplica una *revisión técnica* sobre los casos de prueba evaluando los criterios para cada uno de los niveles, en el orden en que aparecen.

Roles y responsabilidades

Diseñador de casos de pruebas: Genera los casos de pruebas

Jefe del Equipo de Calidad: Revisa y aprueba los casos de prueba V&V

Entradas requeridas

DDF

DP

Salidas requeridas

Reporte de tarea

Casos de Prueba de Unidad V&V

Casos de Prueba de Integración V&V

Casos de Prueba de Sistema V&V

Casos de Prueba de Aceptación V&V

5.1.4.4. Tarea: Generación y verificación de los Procedimientos de Prueba V&V

Descripción

Se generan los procedimientos de prueba. Se verifica que estos cumplan con el formato, propósito y contenido del Documento de Pruebas definido para el proyecto. Se valida que estos satisfacen los siguientes criterios:

- Unidad
Criterios definidos en la Tarea 5.1.3.4
- Integración
Criterios definidos en la Tarea 5.1.3.5
- Sistema
Criterios definidos en la Tarea 5.1.2.5
- Aceptación.
Criterios definidos en la Tarea 5.1.2.3

Métodos y procedimientos

Los miembros del equipo de calidad que son responsables de las pruebas de unidad, sistema, integración y aceptación, junto a otro revisor realizan un *peer review* durante la generación de los procedimientos de prueba, este trabajo permite que en paralelo con la obtención de los procedimientos se ejecute la revisión de los mismos.

Roles y responsabilidades

Responsable de pruebas: Participa en el peer review.

Revisor Técnico: Participa en el peer review.

Entradas requeridas

DDF

DP

Salidas requeridas

Reporte de tarea

Procedimientos de pruebas de Unidad V&V

Procedimientos de pruebas de Integración V&V

Procedimientos de pruebas de Sistema V&V

Procedimientos de pruebas de Aceptación V&V

5.1.4.5. Tarea: Ejecución y verificación de las pruebas unitarias

Descripción

Ejecutar las pruebas unitarias a nivel de acceso a datos y de negocio. Los resultados serán utilizados para validar que el software satisface los criterios de aceptación para la prueba.

Métodos y procedimientos

Se aplica una prueba unitaria a las unidades definidas y se evalúa el resultado contra los criterios de aceptación previamente definidos para esa prueba.

Roles y responsabilidades

Programador: Genera el código y realiza pruebas unitarias.

Revisor Técnico: Participa junto a los programadores en la aplicación de las pruebas. Evalúa la prueba atendiendo a los criterios de aceptación para la misma y elabora el DNC.

Entradas requeridas

Código fuente

Casos de prueba

Criterios de aceptación de la prueba

Salidas requeridas

Reporte de tarea

DNC

5.1.5. Actividad: Pruebas V&V

La actividad de pruebas V&V cubre las pruebas del software. El objetivo fundamental es asegurar que los requerimientos del software son satisfechos con la ejecución de pruebas de sistema, integración y aceptación; además de realizar una revisión completa a la documentación de usuario que se genera en el proyecto.

5.1.5.1. Tarea: Análisis de la trazabilidad

Descripción

Analizar las relaciones entre el Plan de Pruebas, el Diseño de Pruebas, los Casos de Pruebas y los Procedimientos de Prueba en cuanto a corrección y completitud.

Corrección

Verificar que las relaciones entre el Plan de Pruebas, Diseño de Pruebas, Casos de Pruebas y Procedimientos de Pruebas son válidas.

Completitud

Verificar que todos los Procedimientos de Pruebas son traceables al Plan de Pruebas.

Métodos y procedimientos

Se aplica un *walk-through*, donde en una o varias reuniones de trabajo los responsables de cada tipo de prueba explican a los demás miembros del equipo de calidad la forma en que se relacionan los planes de prueba, con los diseños, casos y procedimientos de prueba. El equipo realiza un análisis a partir de la explicación recibida, detecta anomalías y propone las posibles soluciones que deben ser ejecutadas de inmediato.

Roles y responsabilidades

Miembro del Equipo de Calidad: Participa en el walk-through

Entradas requeridas

Planes de prueba

Diseños de prueba

Casos de prueba

Procedimientos de prueba

Salidas requeridas

Reporte de tarea

Análisis de la trazabilidad.

5.1.5.2. Tarea: Ejecución y verificación de las pruebas de integración

Descripción

Se ejecutan las pruebas de integración que fueron planificadas en la actividad de Diseño V&V, con el objetivo de evaluar el comportamiento del software como un todo, teniendo en cuenta que de la correcta integración entre los módulos depende que no existan inconsistencias en los datos registrados en el sistema. Se verifican los resultados contra los criterios de aceptación para la prueba.

Métodos y procedimientos

Se realizan las pruebas de integración y se evalúan los resultados contra los criterios de aceptación previamente definidos.

Roles y responsabilidades

Diseñador: Ejecuta la prueba.

Probador: Participa en la ejecución de la prueba y confecciona el DNC.

Arquitecto de software: Supervisa la prueba

Entradas requeridas

Versión estable del software.

Casos de prueba.

Criterios de aceptación de la prueba

Salidas requeridas

Reporte de tarea

DNC (Anexo N)

5.1.5.3. Tarea: Ejecución y verificación de las pruebas de sistema

Descripción

Se ejecutan las pruebas de sistema planificadas en la actividad de Diseño V&V y se verifican los resultados contra los criterios de aceptación para la prueba.

Métodos y procedimientos

Se ejecutan las siguientes pruebas, y se evalúan los resultados contra los criterios de aceptación previamente definidos para cada una:

- Prueba de funcionalidad
- Prueba de rendimiento
- Prueba de estrés
- Prueba de seguridad

Roles y responsabilidades

Probador: Ejecuta las pruebas y confecciona los DNC

Entradas requeridas

Versión estable del software.

Casos de prueba.

Criterios de aceptación de la prueba

Salidas requeridas

Reporte de tarea

DNC

5.1.5.4. Tarea: Análisis de la documentación de usuario

Descripción

Evaluar la documentación de usuario en cuanto a legibilidad, completitud y corrección.

Legibilidad

Verificar que los manuales de usuario y la ayuda en línea se han escrito en un vocabulario correcto y claro, con la estructura y formato adecuados.

Completitud

Revisar que estos contengan toda la información necesaria para que el usuario comprenda todas las funcionalidades del sistema, y en el caso de la ayuda que permita realizar búsquedas de contenidos.

Corrección

Verificar que la ayuda funciona correctamente integrada al SIGEP.

Métodos y procedimientos

Se realiza un *peer review* entre el Analista de negocio que confecciona el manual de usuario y un revisor técnico, evaluando los criterios expuestos en la descripción de la tarea.

Se realiza un *peer review* entre el responsable de la ayuda y un revisor técnico, evaluando los criterios expuestos en la descripción de la tarea.

Roles y responsabilidades

Analista de negocio: Confecciona el manual de usuario y participa en el *peer review*.

Responsable de la ayuda: Confecciona la ayuda y participa en el *peer review*.

Revisor Técnico: Participa en el *peer review* y elabora el DNC en caso de ser necesario.

Entradas requeridas

Manuales de usuario

Ayuda del sistema

Salidas requeridas

Reporte de tarea

DNC

5.1.5.5. Tarea: Ejecución y verificación de las pruebas de liberación

Descripción

Se realizan las pruebas de liberación por la Dirección de Calidad de Software de la universidad para probar y validar las funcionalidades del producto que va a ser liberado.

Métodos y procedimientos

Se ejecutan las pruebas de liberación por personal ajeno al equipo de desarrollo del proyecto pertenecientes a la Dirección de Calidad de Software de la universidad.

Roles y responsabilidades

Especialista de la Dirección de Calidad: Dirige el proceso de pruebas de liberación

Jefe del equipo de calidad: Asesora el proceso

Probador-UCI: Realiza las pruebas

Entradas requeridas

Versión estable del software

Casos de prueba

Salidas requeridas

Reporte de tarea

DNC

Resultado de las pruebas de liberación

5.1.5.6. Tarea: Ejecución y verificación de las pruebas de aceptación

Descripción

Se realizan las pruebas de aceptación con el cliente para validar que el producto entregado cumple con las funcionalidades acordadas.

Métodos y procedimientos

Se ejecutan las pruebas de aceptación en el entorno del cliente que recibe el producto.

Roles y responsabilidades

Representante (cliente): Asegura las precondiciones correspondientes, controla el plan de pruebas de aceptación y coordina el trabajo de pruebas, representa los intereses de la parte venezolana.

Especialista funcional venezolano: Ejecuta las pruebas y realiza las peticiones de cambios

Representante cubano: Asegura las precondiciones correspondientes, valora la significación de una petición de cambios, supervisa el trabajo de pruebas.

Arquitecto: Asegura el correcto funcionamiento del equipamiento técnico del software, valora la significación de una petición de cambios, prepara a todo el equipo de prueba en el trabajo con el producto

Diseñador: Asegurar el correcto funcionamiento del equipamiento técnico y el software, valora la significación de una petición de cambios

Probador: Garantiza la correcta ejecución de las pruebas, asegura el entendimiento fluido entre la parte cliente y el equipo de desarrollo, recoge las no conformidades y solicitudes de cambios, y conforma el resumen de los resultados de las pruebas de aceptación

Analista de negocio: Apoya la correcta ejecución de las pruebas, valora la petición de un cambio, conforma el resultado de las pruebas de aceptación, y prepara al equipo de prueba en el trabajo con el producto.

Entradas requeridas

Versión estable del software

Casos de prueba

Salidas requeridas

Reporte de tarea

DNC

Documento de Petición de Cambios

Resultado de las pruebas de aceptación

5.1.6. Actividad: Instalación y Chequeo V&V

Esta actividad es la instalación del producto de software en el entorno previsto para su uso, así como la aceptación por parte del que lo adquiere. El objetivo fundamental es verificar y validar la corrección de la instalación del software en el ambiente correspondiente.

5.1.6.1. Tarea: Evaluación de la configuración de instalación.

Descripción

Se evalúa la configuración de instalación en cuanto a completitud y corrección.

Completitud

Verificar que todos los productos de software necesarios para instalar y operar correctamente el software están presentes en el paquete de instalación:

- Máquina Virtual de Java (JVM) en la versión recomendada.
- Apache TomCat en la versión recomendada
- Código (*.jar* ó *.war*)

Corrección

Verificar que la instalación se completa según los pasos definidos en el *asistente* de la instalación.

Verificar una vez completada la instalación que están presentes todos los componentes que debían instalarse.

Métodos y procedimientos

Se realiza una *revisión técnica* para evaluar la completitud de la configuración de instalación.

Se ejecuta *una prueba de instalación* para evaluar el criterio de corrección detallado en la descripción de la tarea.

Roles y responsabilidades

Desarrollador: Prepara el paquete de instalación.

Revisor Técnico: Revisa el paquete de instalación.

Probador: Realiza la prueba de instalación.

Entradas requeridas

Paquete de instalación del software

Salidas requeridas

Reporte de tarea

DNC

5.1.6.2. Tarea: Evaluación del Manual de Instalación

Descripción

Evaluar el manual de instalación en cuanto a legibilidad y completitud.

Legibilidad

Verificar que el Manual de Instalación se ha escrito en un vocabulario correcto y claro, con la estructura y formato adecuados.

Completitud

Verificar que contenga toda la información necesaria, para que el usuario siguiendo los pasos definidos para la instalación logre completarla con éxito.

Métodos y procedimientos

Revisión técnica para evaluar la legibilidad del manual.

La *prueba de instalación* descrita en la tarea anterior debe ser realizada con el apoyo del Manual de Instalación, para evaluar la completitud del mismo.

Roles y responsabilidades

Revisor Técnico: Ejecuta la revisión del Manual y confecciona el DNC

Probador: Ejecuta la prueba de instalación.

Entradas requeridas

Manual de Instalación

Salidas requeridas

Reporte de tarea

DNC

Anexo B: Reunión de trabajo para determinar los niveles de integridad en el SIGEP

Reunión de trabajo.

Fecha: 27-01-08

Lugar: Laboratorio 302, Docente 1

Objetivo: Definir el nivel de integridad de cada módulo del SIGEP.

Asistentes

Nombre	Responsabilidad
Ing. Arturo César Arias Orizondo	Líder de proyecto
Ing. Yanet Vega Miniet	Analista principal
Brisey López Bello	Tesista
Javier Arza Valdés	Tesista

Guía de la reunión

1. Analizar exhaustivamente las funcionalidades de cada módulo del sistema en correspondencia con los objetivos que tiene dentro del Sistema Penitenciario y su repercusión en caso de fallo.
2. Analizar si después de implantado el SIGEP en un penal, quedan procesos que solamente dependen del sistema.
 - a. Analizar el impacto que tendría el fallo de una funcionalidad que se corresponda con uno de esos procesos.
3. Revisar si a los procesos legales se les da seguimiento solo dentro del software o además se siguen utilizando las vías tradicionales.
4. Analizar que podría pasar en un penal si el software queda totalmente fuera de servicio.
5. Analizar el impacto en el Sistema Penitenciario en general si el software queda totalmente fuera de servicio.

Con el análisis de los puntos anteriormente expuestos se llegó a la siguiente conclusión.

Subsistema	Descripción	Nivel
Control Penal	Administrar los aspectos relacionados con la situación legal de los privados de libertad y la observancia del cumplimiento de la pena durante su tránsito por el sistema penitenciario.	2
Módulos del Subsistema		Nivel
Datos Personales		2
Ingreso		2
Egreso		2
Situación Jurídica		2
Solicitudes		2
Decisiones		2

Subsistema	Descripción	Nivel
Observación, Clasificación y Tratamiento	Controlar las actividades de reinserción de los privados de libertad.	2

Módulos del Subsistema **Nivel**

Control de Vínculos		2
Entrevista Inicial		2
Clasificación		2
Evaluaciones Técnicas		2
Educación		2
Trabajo		2
Deporte, Cultura y Recreación		2

Subsistema	Descripción	Nivel
------------	-------------	-------

Salud Integral	Controlar las actividades de la atención médica odontológica que reciben los privados de libertad.	2
----------------	--	---

Módulos del Subsistema **Nivel**

Interrogatorio (Anamnesis)		2
Odontología		2
Diagnóstico y Tratamiento		2
Exámenes Complementarios		2
Control de Maternidad		2
Vacunación		2
Vigilancia Epidemiológica		2

Subsistema	Descripción	Nivel
------------	-------------	-------

Custodia y Seguridad	Controlar las actividades de custodia a los reclusos.	2
----------------------	---	---

Módulos del Subsistema **Nivel**

Gestión de Cupo		2
Salidas Transitorias		2
Traslados Interpenales		2
Planificación Visitas Familiares		2

Ejecución Visitas Familiares		2
Planificación Visitas Institucionales		2
Ejecución Visitas Institucionales		2
Requisas y Decomisos		2
Registro de Pertenencias		2
Novedades y Contingencias		2
Medidas preventivas y Disciplinarias		2
Ubicación		2
Subsistema	Descripción	Nivel
Administración del establecimiento	Mantener un control del personal que labora en los establecimientos y la dotación que se le entrega a los privados de libertad.	2
Módulos del Subsistema		Nivel
Control Armamento		2
Dotación del Interno		2
Control de Personal		2
Control de Capacidades		2
Administración de Biblioteca		2
Administración Farmacia		2
Nutrición		2
Subsistema	Descripción	Nivel
Inspectoría de Asuntos Internos	Registrar los hechos y personas objetos de investigación por parte del departamento de asuntos internos.	2
Subsistema	Descripción	Nivel
Gestión de Información para el Análisis	Monitorear el comportamiento de los indicadores de riesgo del sistema penitenciario.	2
Módulos del Subsistema		Nivel
Información Operativa		2
Avisos		2

Configuración		2
Información Histórica		2
Subsistema	Descripción	Nivel
Administración del SIGEP	Herramientas que faciliten la administración y configuración de la solución de software.	2
Módulos del Subsistema		Nivel
Administración de Usuarios		2
Actualización Sistema		2
Auditoría		2
Réplica de Datos		2
Subsistema	Descripción	Nivel
Dirección Operativa (Sala Situacional)	Servir de soporte para la generación de reportes, informes, estadísticas, cuadros de análisis y gráficos que provean información diaria de la situación operativa. Apoyará la toma de decisiones de índole táctica y estratégico	2


Arturo Arias Orizondo

Líder de Proyecto


Janet Vega Pliniet

Analista Principal

Anexo C: Guía utilizada para entrevistar a miembros del proyecto SIGEP

Entrevistadores: Autores de la investigación.

Entrevistados: Líder de proyecto, Arquitecto principal, Diseñadores, Analistas, Programadores, Miembros del equipo de calidad, Documentador.

Objetivos:

- a) Determinar el nivel de conocimientos sobre V&V que tienen los miembros del equipo de desarrollo.
- b) Obtener todos los detalles concernientes a la estrategia que se utiliza para el desarrollo del SIGEP.
- c) Identificar en todas las actividades y tareas los puntos relacionados con la Verificación y la Validación.
- d) Comprender cómo se realizan estas actividades.
- e) Profundizar en las funciones específicas que desempeña cada rol dentro del proyecto.
- f) Teniendo en cuenta los puntos de vista del entrevistado, identificar las deficiencias reales que existen relacionadas con la V&V y encontrar posibles vías de solución para las mismas.
- g) Determinar la factibilidad de la aplicación de los métodos V&V en el proyecto.
- h) Dar los primeros pasos en función de reducir los efectos que pudiera tener la resistencia a la puesta en práctica del plan.

Preguntas

Sobre la base de la estructuración de los objetivos, los entrevistadores efectuarán preguntas a los entrevistados de manera que se obtenga toda la información requerida para cumplir con estos objetivos.

	Diseñador de BD, Revisor técnico								
Evaluación del diseño del software	Arquitecto de software	Diseñador							
Análisis de las interfaces				Revisor técnico					
Generación y verificación del Plan de Pruebas de Unidad V&V			Responsable de pruebas de unidad, Revisor técnico						
Generación y verificación del Plan de Pruebas de Integración V&V			Responsable de pruebas de integración, Revisor técnico						
Generación y verificación del Diseño de las Pruebas V&V	Jefe del equipo de calidad								
Implementación V&V									
Análisis la trazabilidad	Diseñador, Revisor técnico								
Evaluación del código y su documentación	Revisor técnico		Programador	Arquitecto de software					
Generación y verificación de los Casos de Prueba V&V	Jefe del equipo de calidad								
Generación y verificación de los Procedimientos de Prueba V&V			Responsables de prueba, Revisor técnico						
Ejecución y verificación de las pruebas unitarias						Programador Revisor			
Pruebas V&V									
Análisis la trazabilidad		Equipo de							

