

Universidad de las Ciencias Informáticas
Facultad 8



**Estrategia para el aseguramiento de la calidad en el
proyecto “Contenidos Educativos Digitales”.**

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

Autores: Maidelis Ayala Girón

Israel Acosta Ercia

Tutora: Ing. Maylen Cepero Madruga

Co-tutor: Ing. Yudisbel Pérez Moreno

Ciudad de la Habana

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 firmamos la presente a los ____ días del mes de _____ del año _____.

Autor: Maidelis Ayala Girón

Autor: Israel Acosta Ercia

Tutor(a): Maylen Cepero Madruga

Co – tutor: Yudisbel Pérez Moreno

Dedicatoria

A mis padres por estar siempre a mi lado, porque todo lo que soy se lo debo a ellos.

A mi hermanita, por aguantarme todas mis malcriadeces y ayudarme en los momentos más difíciles de mi vida.

A todos aquellos que tuvieron confianza en mí y que nunca dudaron, a los que las palabras y consejos de aliento nunca les faltaron para conmigo, aquí les va este logro compartido.

Israel Acosta Ercia

Dedicatoria

Dedico este trabajo con todo el amor del mundo a mis padres, por su sacrificio y dedicación, por escucharme siempre y saber comprenderme, gracias por ser los mejores padres del mundo y por toda su confianza. A mis abuelos por estar siempre pendiente de mí y regalarme tanto cariño. A todos ellos les debo lo que soy. Y a mis amigos que han estado para brindarme su ayuda incondicional.

Maidelis Ayala Girón

Agradecimientos

Agradezco en primer lugar a mis padres, a los que quiero y admiro. Mamá, papá: gracias por su apoyo y sabios consejos; un millón de gracias por darme la oportunidad de estar aquí en estos momentos, por hacerme ver que con esfuerzo, mucha fe y dedicación todos salimos adelante, por estar siempre a mi lado brindándome su cariño en los momentos felices y tristes, gracias por saberme mostrar el camino correcto, por su educación y por ser ejemplos de padres.

A mi mamá: por saberse engrandecer en todo momento frente a los problemas, y llevarnos a mi hermana y a mí cada día por el camino correcto... gracias y mil veces gracias mamita...

A mi papá: por el apoyo incondicional que me ha dado siempre, por su ternura, comprensión, su infinito amor y sobre todas las cosas, por la inmensa confianza que siempre ha depositado en mí, gracias por cuidar de este loco que te ama con todo el corazón.

A mi hermana: Por ser el más grande regalo que he recibido en la vida y por asumir con tanta decisión y valentía los malos momentos que nos ha impuesto la vida. Desde bien pequeña te convertiste en mujer. Gracias por esa sobrina que desde ya me hace la persona más feliz del mundo.

A mi tía: Por ser la otra madre que siempre me aconseja, me ayuda, me regaña. Por ayudarme y quererme tanto y sobre todo cuidarme como tu hijo, por darme esos consejos y esas conversaciones preparándome para la vida. A partir de hoy, vas a tener que buscar otro que te ayude en la casa y te rompa la computadora porque ya soy ingeniero...

A mi abuelita que siempre se ha preocupado porque yo sea un gran profesional.

A mis primos y primas, en especial Liz Mercedes por alegrarme la vida, por ser tan linda conmigo... cómo nos vamos a extrañar muchachita!!

A mi otra familia, mis amigos y amigas, que nunca pensé que fueran tan maravillosos, a todos los que durante estos años me hicieron felices que son demasiados para ponerlos en este papel, pero hay algunos que no puedo dejar de mencionar: Carlos, Carlitos, Raúl, Arianna, Claritza, Daylenis, Madelaine, Anita la gorda, Anita la de Jorge, Jose, Leydis, las chicas de mi aula, mi piquete de 5to año, y muchos otros que más que amigos son mis hermanos, a ellos por soportar mis buenos y malos

días, por darme confianza, por criticar y valorar mis acciones, por estar siempre a mi lado, más en los malos que en los buenos momentos, por ser otra de las cosas que más quiero.

A una persona que a pesar de haber conocido hace poco tiempo me ayudó muchísimo en todo momento con mis noches llenas de estrés y que siempre estuvo para escucharme, gracias Yádimir.

A la UCI por recibirme como un niño y despedirme hoy como todo un profesional.

A los que creyeron en mí; y a los que no lo hicieron esta es una forma de demostrarles cuán equivocados estaban.

Israel Acosta Ercia

Agradecimientos

Hay momentos en la vida que son especiales por sí solos. Compartirlos con las personas que quieres los convierte en momentos inolvidables. Cuando la gratitud es tan absoluta las palabras sobran y precisamente en este momento las palabras se me agotan para agradecer a tantas personas que han tenido que ver con que suceda este momento tan especial en mi vida.

Agradezco especialmente a mis padres, pues no tengo forma de agradecerles por todo su amor, comprensión, por su maravilloso ejemplo, por el todo el apoyo que me han brindado, por confiar siempre en mí y porque son las personas más especial en mi vida, gracias.

A mis abuelos, por su cariño en todo momento, pues son tan lindos conmigo y siempre están orgullosos de su nieta. A Wilson que ha sido parte de mi familia, gracias por tu cariño y apoyo incondicional y por soportarme todos estos años.

A mi prima Nayibis que ha sido una gran amiga en los años de mi carrera y me ha apoyado en todo momento.

A Oneida amiga mía por ser mi gran consejera.

A mi prima Maile por ser mi amiga y compartir conmigo momentos especiales en los primeros años de mi carrera.

A mis tías, tíos y primos que nunca me han dado la espalda cuando los he necesitado, especialmente mi tía Choly.

A todos los maravillosos vecinos que he tenido Tata, Mimi, Lizy, Yao, Maribel, Jose por soportarme cada día, Manuel, Gladis, Consuelo, a mi prima Anita, Jorgito, Juancito y familia, Luis Perez, Conchita, Darella, Danella, Erit y mi tío querido Juan Escobedo que aunque ya no esté presente fue una persona muy especial en mi vida, y a todos los que de alguna forma han sido mis compañeros, más que vecinos han sido una familia gigante, gracias por guiarme y poder contar siempre con su ayuda.

A Eri que a pesar de la distancia me ha apoyado mucho, gracias por todo amiga.

A Edgar que fue siempre de los hombres mi mejor compañero, brindándome su apoyo incondicional y estando siempre ahí para mí.

A todos los amigos de mi papa que son también mi familia, Suarez, Rjo, Odalis, Pedro, la neuro, Isidro, Idelsis, mi primo Reinier y a todos los que de alguna manera me han apoyado, gracias.

A mis amigas de siempre Yaima, Yuneisis, Marlen y Ailín que han estado ahí para apoyarme, gracias por poder confiar en ustedes.

A mis compañeros de siempre Lili, Street, Julio, Mario, Anita y en especial a Motojo por molestarlo tanto, gracias por su ayuda.

A mi compañero de tesis Israel que sin su entrega constante y su paciencia para conmigo no hubiera podido realizar este trabajo.

A Emilio Rubén por compartir conmigo momentos y locuras inolvidables.

A todos aquellos que me han brindado su ayuda en mis años de estudio en La Habana, Cuqui, Panchito, Jorge Cano, Chuchi, a mi tía María Ayala, Umbelina y María.

A mis compañeras de apartamento por ser mi familia aquí, especialmente Merlin, Yailín y Lili por permitirme compartir el cuarto con ellas y por ser mis amigas en este último año.

A mi compañero Jesus por brindarme su apoyo en todo momento, gracias por ser tan especial conmigo y por lograr refrescar mi alma con tu alegría.

A mis compañeros de grupo y a todos los amigos tan especiales que he conocido en esta universidad, gracias por su amistad.

A todos quienes compartían las noches de insomnio y las buenas y malas noticias de estos años a mi lado, mil gracias.

A mi tutora Maylen y al tribunal de tesis, gracias.

A la UCI y a mis profesores por contribuir en nuestra preparación profesional.

A todos los que aunque sus nombres no estén han sido parte de mi vida. En fin a todos los de alguna manera contribuyeron a que hoy este haciendo estos agradecimientos que no son más que la memoria de mi corazón.

A nuestro invencible Comandante en Jefe por ser el principal precursor de la Universidad de las Ciencias Informáticas y por permitirnos formar parte de este proyecto futuro.

Maidelis Ayala Girón

Resumen

El presente trabajo lleva por título “Estrategia para el aseguramiento de la calidad en el proyecto Contenidos Educativos Digitales” cuenta con un estudio del estado del arte enmarcado en elementos relacionados con la calidad del software, puntualizándose temas como el aseguramiento y estándares de calidad, además de pruebas, revisiones y métricas de software. Se realizó también una descripción detallada de la situación actual del aseguramiento de la calidad en el proyecto Contenidos Educativos Digitales, especificando los principales señalamientos detectados por los autores de este trabajo.

Todo esto se realiza con el objetivo de definir una estrategia para el aseguramiento de la calidad para el proyecto Contenidos Educativos Digitales que posibilite lograr un nivel de calidad en los diferentes módulos del producto a obtener. Esta propuesta está sustentada por el Plan de Aseguramiento de la Calidad como artefacto fundamental, además de una redefinición adecuada del equipo de calidad existente en dicho proyecto, y se describen los procesos de revisiones y pruebas en el proyecto. Finalmente se realizó la validación de la propuesta a partir de los resultados obtenidos luego de la aplicación de la misma dentro del proyecto Contenidos Educativos Digitales y a partir de los criterios reflejados por parte del equipo de desarrollo del producto, obteniéndose resultados satisfactorios que aprueban que la propuesta favorece el incremento de la eficiencia dentro del mismo.

Palabras Clave: Aseguramiento de la Calidad, Estrategia, Plan de Aseguramiento de la Calidad

Índice

Capítulo 1. Fundamentación Teórica	16
1.1 Introducción.....	16
1.2 Estudio sobre la calidad	16
1.2.1 Evolución histórica de la calidad.....	17
1.2.2 Concepto de calidad	18
1.2.3 ¿Qué es calidad del software?.....	19
1.2.4 Calidad a nivel de proceso de software.....	20
1.2.5 Calidad a nivel del producto	21
1.3 Factores que determinan la calidad del software	22
1.4 Metodología Proceso Unificado de Rational.....	24
1.5 Métricas de software	26
1.6 Revisiones del software	27
1.7 Revisiones Técnicas Formales (RTF).....	28
1.8 Pruebas de software	31
1.10 Enfoques del aseguramiento de la calidad.....	33
1.10.1 El Aseguramiento de la calidad según el modelo ISO 9000-3	33
1.10.2 El Aseguramiento de la calidad según el modelo Tick-IT	34
1.10.3 El Aseguramiento de la calidad según el modelo ISO-SPICE	35
1.10.4 El Aseguramiento de la calidad según el modelo CMMI	36
1.10.5 Análisis comparativo del aseguramiento de la calidad según los enfoques de los modelos estudiados ...	37
1.11 Estado actual del uso de los modelos de calidad.....	38
1.11.1 A nivel internacional.....	38
1.11.2 En Cuba	40
1.11.3 En la Universidad de la Ciencias Informáticas (UCI)	41

1.12 Estudio sobre modelos de Calidad.....	42
1.12.1 ¿Qué son los modelos de calidad?	42
1.12.2 Ventajas y necesidades del uso de los modelos de calidad	43
1.12.3 Características de los modelos de calidad	43
Características de Modelo Integrado de Capacidad y Madurez (CMMI):	43
1.13 Conclusiones del capítulo.....	45
Capítulo 2. Análisis de la Situación Actual	47
2.1 Introducción	47
2.2 Contenidos Educativos Digitales. Antecedentes	47
2.3 Descripción.....	48
2.4 Estructura y Composición	49
2.4.1 Introducción	49
2.4.2 Roles y responsabilidades definidos.....	49
2.4.3 Recursos humanos por roles	50
2.4.4 Soporte Tecnológico	51
2.5 Situación Real.....	51
2.6 Resultados Esperados	52
Capítulo 3. Propuesta y Descripción de la Solución	54
3.1 Introducción	54
3.2 Características de la Estrategia.....	54
3.3 Definición del Equipo de Aseguramiento de la Calidad en el Proyecto Contenidos Educativos Digitales.....	55
3.4 Definición del Plan de Aseguramiento de la Calidad del Software.....	56
3.4.1 Introducción	56
3.4.1.2 Propósito	56
3.4.1.3 Alcance	56
3.4.1.4 Definiciones, Acrónimos y Abreviaturas.....	57
3.4.1.5 Referencias	57

3.4.1.6 Resumen	57
3.4.2 Objetivos de Calidad.....	57
3.4.3.1 Organización	58
3.4.3.2 Tareas y Responsabilidades.....	59
3.4.4 Documentación	60
3.4.5 Métricas.....	61
3.4.6 Estándares y Guías.....	67
3.4.7 Revisiones y Auditorías.....	68
3.4.7.5 Cronograma.....	71
3.4.7.7 Resolución de problemas y actividades de corrección.....	73
3.4.8 Pruebas.....	74
3.4.10 Gestión de Configuración	77
3.4.11 Registros de Calidad	78
3.4.12 Entrenamiento.....	79
3.5 Actividades para el chequeo del Plan de Aseguramiento de la Calidad.....	81
3.6 Conclusiones del capítulo.....	84
Capítulo 4. Validación de la Propuesta	85
4.1 Introducción	85
4.2 Resultados de la Aplicación de la Propuesta de Solución	85
4.2.1 Entrenamiento del Equipo de Desarrollo	85
4.2.3 Revisiones	88
4.3 Conclusiones del capítulo	88
Conclusiones.....	90
Recomendaciones	91

Referencias Bibliográficas	92
Bibliografía	94
Glosario de Términos	96
Anexos	99
Anexo 1. Factores. Satisfacción del Cliente	99
Anexo 2. Roles y responsabilidades	99
Anexo 3. Políticas y Pautas de documentación	101
Anexo 4 Tabla de Métricas propuestas para la calidad del software educativo.	113
Anexo 5 Lista de Chequeo para el Plan de Gestión de Configuración de Software	115
Anexo 6 Registro de No Conformidades	117
Anexo 7 Lista de Chequeo para el Plan de Desarrollo de Software	118
Anexo 8 Lista de Chequeo de los Documentos Ingenieriles	118
Anexo 9 Diagrama de Actividades de la Revisión del Plan de Gestión de Configuración de Software	122
Anexo 10 Diagrama de Actividades de la Revisión del Plan de Desarrollo de Software	123
Anexo 11 Diagrama de Actividades de la Rev. de la Documentación Ingenieril del CED	124
Anexo 12 Diagrama de Actividades del Proceso de Pruebas	125

Introducción

En la actualidad el avance tecnológico es un elemento que está indisolublemente ligado al desarrollo de cualquier país del mundo. La tecnología y su adecuado uso son temas de primer orden en el ámbito mundial. Un gran número de naciones han destinado cuantiosos esfuerzos a la instrucción de personal calificado en el desarrollo de software, con el fin de que estos desarrollen múltiples productos que pueden ser de gran utilidad tanto para el propio progreso del país, como para exportarlos bajo contratos firmados con todo tipo de clientes que lo requieran, obteniendo a cambio grandes ingresos.

Es por esto que la industria del software en nuestros días ha alcanzado un vertiginoso auge, evolucionando aceleradamente con el fin de obtener cada vez mayores ganancias. Existe una gran competencia entre distintas empresas desarrolladoras de software, teniendo unas mucho más demandas que otras, por tal motivo, los clientes deben decidir cuál será su mejor opción, fijándose en el tiempo que tomará terminar su producto, su costo y principalmente en la calidad que tendrá el mismo. La interrogante será si estos productos realmente llegarán a manos del cliente con las funcionalidades requeridas para que cubran totalmente las necesidades del cliente. La tendencia actual consiste en que el software no cubre las expectativas reales del consumidor, como consecuencia del procedimiento escogido para construirlo.

El aseguramiento de la calidad y la estrategia que se decida para alcanzarla son puntos vitales que nunca deben ser ignorados por ningún equipo de desarrollo, no establecer dicha estrategia puede traer consigo grandes gastos tanto de tiempo como de recursos, así como el descontento del cliente.

La Universidad de las Ciencias Informáticas juega un papel fundamental en el desarrollo de software en nuestro país con sus múltiples proyectos productivos. Desde su surgimiento, la Facultad 8 se ha especializado principalmente en la producción de Multimedia y Software Educativos. Actualmente se encuentra en desarrollo el proyecto “Contenidos Educativos Digitales”, el cual constituye un medio informativo, formativo e interactivo de fácil manejo que enseña al usuario la Historia de Venezuela y lo familiariza con las herramientas multimedia al mismo tiempo.

En este proyecto se han detectado varios problemas que conspiran contra su exitoso desarrollo y con los cuales es imposible lograr un producto con niveles adecuados de calidad. Los integrantes del equipo de

desarrollo presentan faltas de conocimientos y de experiencia en el uso de las herramientas y metodologías seleccionadas para la realización del software. Además un problema típico es la necesidad de realizar cambios en procesos ya terminados debido a no haber comprobado previamente que dicho proceso no tenía la calidad requerida, esta situación se debe al hecho de no llevar un control de calidad en todas las etapas del ciclo de vida del proyecto. Se ha verificado también que hay errores en la documentación del expediente de proyecto, en algunos casos las plantillas han sido conformadas incorrectamente y en otros se encuentran incompletas. Aunque existen varios modelos y estándares de software, en sentido general son muy genéricos por lo que se torna en ocasiones difícil adaptarlo a las necesidades de un producto específico, en “Contenidos Educativos Digitales” no se ha seleccionado ningún estándar al cual adherir el proceso de desarrollo. En múltiples ocasiones no se ha cumplido cabalmente con el cronograma de trabajo, experimentándose retrasos significativos que atentan seriamente con los plazos de entrega, lo cual trae consigo el descontento por parte del cliente.

No asegurar la calidad en dicho proyecto no solo es una mala práctica, sino que trae varios problemas como la posibilidad de errores en el producto. Además no existe nadie encargado de llevar un control con los errores detectados y realizar una documentación con todos estos para ganar experiencia en la realización de productos futuros.

El proyecto “Contenidos Educativos Digitales” no cuenta actualmente con un método definido para hacer frente a todos estos problemas, y sin este no será posible lograr de forma efectiva una organización de manera que todos estén comprometidos seriamente con la realización correcta del producto que se está desarrollando.

A partir de esta **situación problemática** se ha planteado como **problema a resolver**: ¿Cómo lograr un proceso de desarrollo exitoso en el proyecto “Contenidos Educativos Digitales” y un producto libre de errores?

El **objeto de estudio** es: El proceso de aseguramiento de la calidad de software y el **campo de acción** específico a analizar es: El proceso de aseguramiento de la calidad en el proyecto “Contenidos Educativos Digitales”.

Por tal motivo se ha propuesto como **objetivo general**: Definir una estrategia para el aseguramiento de la calidad en el proyecto “Contenidos Educativos Digitales”, y como **objetivos específicos** los siguientes:

- Realizar un análisis de los modelos y estándares que existen internacionalmente vinculados a los procesos de aseguramiento de la calidad de un software.
- Definir las métricas a utilizar en el modelo de evaluación de la calidad.
- Proponer un entrenamiento para nivelar los conocimientos básicos del equipo de desarrollo.
- Proponer el plan de aseguramiento de la calidad: artefacto fundamental de la estrategia propuesta.
- Realizar la validación de la propuesta.

Como **idea a defender** el trabajo plantea que: Con la definición correcta de una estrategia para el aseguramiento de la calidad de software se espera lograr el cumplimiento de las metas trazadas, así como la entrega del proyecto “Contenidos Educativos Digitales” en tiempo y con los niveles de calidad requeridos.

Para darle cumplimiento a los objetivos propuestos se han trazado un conjunto de **tareas** que contribuirán a que la investigación se efectúe de una forma óptima. Estas son:

- Estudiar los modelos y estándares que existen internacionalmente para el proceso de aseguramiento de la calidad de un software.
- Estudiar las métricas que se utilizan en el proceso de evaluación del software y definir cuáles se adaptan a nuestro proyecto.
- Realizar un plan de aseguramiento de la calidad: artefacto fundamental de la estrategia propuesta.
- Realizar una propuesta de entrenamiento para el equipo de desarrollo del proyecto “Contenidos Educativos Digitales”.
- Organizar correctamente el trabajo y definir las responsabilidades de cada rol.
- Resolver los problemas existentes y corregir las actividades.
- Validar la propuesta.

Capítulo



Capítulo 1. Fundamentación Teórica

1.1 Introducción

En este capítulo se abordan diferentes conceptos y definiciones referentes al tema de la calidad de software, con el fin de ofrecer una visión más detallada y profunda sobre el contenido a tratar en el presente Trabajo de Diploma. Además se recogen los aspectos más significativos relacionados con la temática, abordados tanto en distintas fuentes bibliográficas como en criterios emitidos por especialistas. Se hará un estudio del estado del arte del tema referenciado, donde se realiza un análisis de los enfoques del aseguramiento de la calidad de algunos de los principales Estándares de Calidad de Software existentes a nivel mundial, y luego se efectúa una comparación de los mismos con el objetivo de escoger un estándar por el cual regirse.

1.2 Estudio sobre la calidad

La Organización Internacional para la Estandarización ISO define calidad como “*Conjunto de propiedades y de características de un producto o servicio, que le confieren aptitud para satisfacer necesidades explícitas o implícitas*” [1].

1.2.1 Evolución histórica de la calidad

- Hasta los años 40: Se realizaba una inspección/detección de los errores.
 - Inicialmente comenzó durante el trabajo artesanal, cuando se realizaba el control individual de cada tarea.
 - En 1918 en la Ford Motor Company (Primera cadena de montaje).
 - En 1930 en los Laboratorios Bell.
- Hasta los años 80: Control de la calidad (control estadístico).
 - Existía un mercado poco competitivo. El precio de venta era fijado por el fabricante, y siempre en función del costo del producto.
 - Se impedía que cualquier producto defectuoso llegara a las manos del cliente.
 - Se luchaba por lograr uniformidad en el servicio.
 - Se logra controlar la calidad del departamento de producción utilizando técnicas estadísticas.
- A partir de los años 80: Garantía de la calidad.
 - Existencia de un mercado competitivo y de oferta.
 - Los precios de venta son fijados por el mercado.
 - Aparece la planificación y medida de la calidad. Modelos de calidad.
 - La calidad es un factor que afecta a todos los departamentos.
- Hoy: Gestión de la calidad.
 - La gestión de la calidad tiene un gran impacto estratégico en el mercado. La empresa que gestione la calidad tiene la oportunidad de tener más ventaja en la competencia.
 - La gestión de la calidad comprende la planificación, la fijación de objetivos, la coordinación, la formación y la adaptación de toda la organización.
 - Afecta a la organización en general: los directivos, los trabajadores y los clientes.
 - Hace su aparición la norma ISO 9001:2000 [2].

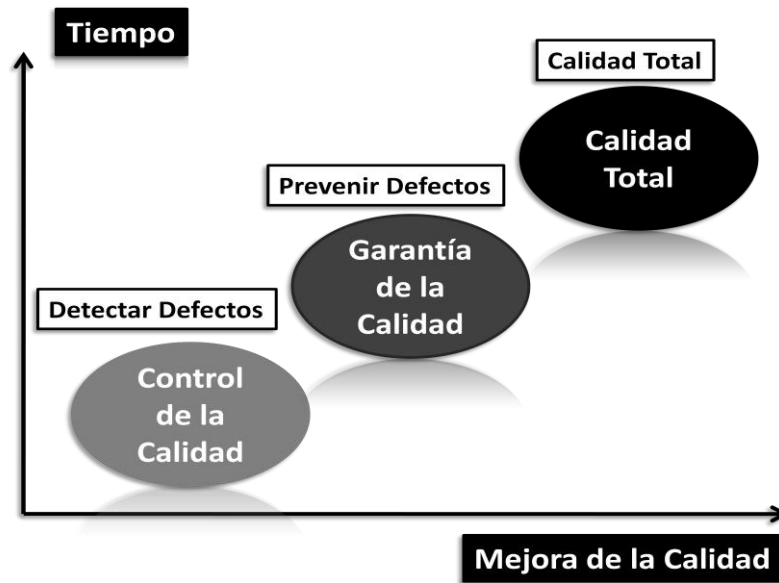


Fig. 1 Evolución Histórica de la Calidad.

1.2.2 Concepto de calidad

La Real Academia Española de la Lengua define el término Calidad como: “*el conjunto de cualidades que constituyen la manera de ser de una persona o cosa*” [3] y se pueden encontrar hoy en día en múltiples contextos. Por tanto el concepto técnico de calidad representa una forma de hacer las cosas en las que, fundamentalmente, predominan la preocupación por satisfacer al cliente y por mejorar, día a día, procesos y resultados.

Se dice que un producto o servicio es de calidad cuando satisface las necesidades y expectativas del cliente o usuario, en función de parámetros como:

- Seguridad que el servicio o producto confieren al cliente.
- Fiabilidad o capacidad que tiene el producto o servicio para cumplir con los requerimientos especificados, sin ningún tipo de fallos o por un período determinado de tiempo.
- Servicio o medida en que el fabricante o distribuidor responden en caso de fallo del producto o servicio.

La Sociedad Americana para el Control de la Calidad define la calidad como: *“el conjunto de características de un producto, proceso o servicio que le confieren su aptitud para satisfacer las necesidades del usuario o cliente.”* [4]

“Calidad es el conjunto de características de una entidad, que le confieren la aptitud para satisfacer las necesidades establecidas y las implícitas.” [5]

“Calidad es el grado en el que un conjunto de características inherentes cumple con los requerimientos.” [6]

Existen varios autores calificados como “Gurús de la Calidad Total” que definen calidad como sigue:

- *Adecuación (del producto) al uso.* [7]
- *Conformidad con requerimientos y confiabilidad en el funcionamiento.* [8]
- *Cero defectos.* [9]

Se considera como calidad, la capacidad que adquiere dicho producto elaborado, de satisfacer las necesidades y exigencias requeridas por el cliente.

1.2.3 ¿Qué es calidad del software?

Debido a la importancia de la calidad del software, en los estudios realizados distintos autores la han descrito de varias maneras:

- *“La Calidad del Software es el conjunto de cualidades que la caracterizan y que determinan su utilidad y existencia, plantea un adecuado balanceo de eficiencia, confiabilidad, facilidad de mantenimiento, portabilidad, facilidad de uso, seguridad e integridad.”* [10]
- También Pressman define la calidad del software como: *“(…) la concordancia con los requerimientos 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.”* [11]

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible

tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

A partir de analizar varios conceptos en especial estos dos que definen lo que es calidad y más específicamente la calidad del software, se llega a la conclusión de que la base de las medidas de calidad es cumplir con los requerimientos solicitados, con faltas de concordancia entre el software y los requerimientos es suficiente para una falta de calidad. También se destaca que si no se sigue ninguna metodología siempre faltará calidad ya que estos estándares y metodologías son los que guían el desarrollo del software basándose en un conjunto de criterios de desarrollo definidos. Se debe tener siempre presente que el control de la calidad debe ser construido desde el principio, no es factible hacer un control de calidad al final del desarrollo.

1.2.4 Calidad a nivel de proceso de software

El Instituto de Ingeniería de Software (SEI) en su modelo CMMI (“**C**apability **M**adurity **M**odel **I**ntegration” traducción al español “**M**odelo **I**ntegrado de **C**apacidad y **M**adurez”) define la calidad como:

- *“El grado en el cual un sistema, componente o proceso cumple con los requerimientos especificados.”*
- *“El grado en el cual el sistema, componente o proceso cumple con las expectativas del cliente o usuario.” [12]*

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. Es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

Para alcanzarla, es necesario la satisfacción por parte de los elementos que intervienen en el proceso: La satisfacción de la alta dirección, la satisfacción del personal involucrado en el desarrollo del sistema y la satisfacción del usuario final.

Para realizar el control de la calidad de un sistema, no basta solamente con lograr un sistema libre de errores una vez elaborado, sino que es de vital importancia que cada uno de los procesos que forman parte de su desarrollo sean llevados a cabo con el control requerido; es decir que eficacia de un sistema dependerá de la calidad del proceso de desarrollo, puesto que si posteriormente son detectados errores o

fallas en él, quiere decir que el proceso mediante el cual se desarrolló no fue el idóneo. La Calidad del Software se diseña conjuntamente con el sistema, nunca al final.

Se considera que este es un tema de suma importancia puesto que un producto desarrollado correctamente significa menos gastos de mantenimiento futuros, lo que proporciona satisfacción tanto para la empresa que desarrolla el producto como para el cliente.

Por esta razón se hace tan importante que el personal esté constantemente actualizado con respecto a temas referentes a la calidad del software, ya que de esto dependerá en gran medida que se logre un cliente contento con el trabajo del equipo de desarrollo.

1.2.5 Calidad a nivel del producto

Calidad del producto: *“Es en muchos aspectos, una característica intangible. La calidad la establece esencialmente el cliente, y se procura que el diseño y la fabricación del producto para la venta, satisfaga estos requerimientos.”* [13]

La calidad de un producto software debe evaluarse usando un modelo de calidad que tenga en cuenta criterios para satisfacer las necesidades de los desarrolladores, mantenedores, adquiridores y usuarios finales.

Se opina que para la obtención de un producto con la calidad requerida es necesaria la utilización de procedimientos o metodologías durante todo el proceso de desarrollo del proyecto. La adopción de una buena política contribuye en gran medida a lograr la calidad, pero no la asegura. Para el aseguramiento se hace necesario su control o evaluación.

A la hora de realizar un software hay que tener en cuenta que este debe cumplir una serie de características que determinan si el producto quedó con la calidad requerida, estos son:

- **Funcionalidad:** La capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el software se usa bajo las condiciones especificadas.
- **Confiabilidad:** La capacidad del software para mantener su nivel de ejecución cuando se usa bajo las condiciones especificadas.
- **Usabilidad:** La capacidad del software de ser comprendido, aprendido, utilizado y de ser amigable para el usuario, cuando se emplee bajo las condiciones especificadas.

- **Eficiencia:** La capacidad del software para proporcionar la requerida ejecución, en relación con la cantidad de recursos usados, bajo las condiciones declaradas.
- **Portabilidad:** La capacidad de software ser transferido de un ambiente a otro.
- **Mantenibilidad:** La capacidad del software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptación del software a los cambios en el ambiente, y en los requerimientos y las especificaciones funcionales.

1.3 Factores que determinan la calidad del software

Para lograr un producto con un nivel de calidad elevado, es necesario analizar una serie de factores que contribuyan de manera positiva en su proceso de desarrollo y que determinen el grado de aceptación que el cliente necesita, o sea, que tenga la máxima calidad.

Según Pressman estos factores se pueden clasificar en dos grandes grupos y son conocidos como:

- Factores que pueden ser medidos directamente (defectos por puntos de función).
- Factores que solo pueden ser medidos indirectamente (facilidad de uso o mantenimiento).

Se centran en tres aspectos fundamentales de un producto software según McCall:

- 1) Características operativas.
- 2) Capacidad de soportar los cambios.
- 3) Adaptabilidad a nuevos entornos.

• **Operaciones del producto:** Características operativas

- **Corrección** (¿Hace lo que se le pide?): El grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente.
- **Fiabilidad** (¿Lo hace de forma fiable todo el tiempo?): El grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida.
- **Eficiencia** (¿Qué recursos hardware y software necesito?): La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.

- Integridad (¿Puedo controlar su uso?): El grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.
 - Facilidad de uso (¿Es fácil y cómodo de manejar?): El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.
- **Revisión del producto**: Capacidad para soportar cambios
 - Facilidad de mantenimiento (¿Puedo localizar los fallos?): El esfuerzo requerido para localizar y reparar errores.
 - Flexibilidad (¿Puedo añadir nuevas opciones?): El esfuerzo requerido para modificar una aplicación en funcionamiento.
 - Facilidad de prueba (¿Puedo probar todas las opciones?): El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requerimientos.
- **Transición del producto**: Adaptabilidad a nuevos entornos
 - Portabilidad (¿Podré usarlo en otra máquina?): El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo.
 - Reusabilidad (¿Podré utilizar alguna parte del software en otra aplicación?): Grado en que las partes de una aplicación pueden utilizarse en otras aplicaciones.
 - Interoperabilidad (¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?): El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas Informáticos.
- [14]

Se definen también tres factores fundamentales que influyen directamente en la satisfacción del cliente, y considerando el hecho de que la calidad y la satisfacción del cliente guardan una estrecha relación, se considera importante reflejarlos ([Ver Anexo 1](#)). Por consiguiente, es importante que la organización recoja información de los clientes en dos momentos diferentes: A priori, sus necesidades y a posteriori, el grado en que ha conseguido satisfacerlas.

Uno de los factores principales por los cuales se ve afectada la calidad en los productos de software, es el factor humano. El personal que forma parte del proceso de desarrollo del software tiene que estar comprometido con el mismo, ya que un descuido o un mal trabajo de algún factor (miembro del equipo)

pueden comprometer, de manera considerable, la calidad final del producto en cuestión. Por ello estar involucrados, formar parte del proceso y estar vinculados a cada detalle relacionado con este, nos hace grandes responsables de la calidad, y por supuesto de la satisfacción del cliente.

Estos factores que inciden positivamente sobre cualquier producto hacen que el mismo tenga una gran ventaja en el mercado, pues según Deming “*La calidad aumenta la productividad*”. Todo esto conlleva a una reacción en cadena que permite producir más, vender más y complacer cada vez mejor las necesidades del cliente.

1.4 Metodología Proceso Unificado de Rational

El Proceso Unificado de Rational (RUP) es una metodología que se utiliza para realizar la documentación de productos de software, tiene como objetivo asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. Es dirigido por casos de uso, centrado en la arquitectura, iterativo (mini-proyectos) e incremental (versiones). Se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos. RUP pretende implementar las mejores prácticas actuales en ingeniería de software. [15]

Es un proceso de ingeniería de software propuesto por Rational Software Corporation para la construcción completa del ciclo de ingeniería de software. Permite la productividad en equipo y la realización de mejores prácticas de software a través de plantillas y herramientas que lo guían en todas las actividades de desarrollo del software. Es un producto que unifica las disciplinas en lo que a desarrollo de software se refiere, incluyendo modelado de negocio, manejo de requerimientos, componentes de desarrollo, ingeniería de datos, manejo y configuración de cambios, y pruebas, cubriendo todo el ciclo de vida de los proyectos basado en la construcción de componentes y maximizando el uso del Lenguaje Unificado de Modelado (UML). [16]

Los aspectos definitorios del Proceso Unificado de Rational pueden resumirse en tres características claves:

- Dirigido por Casos de Uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

Esto es lo que hace único al proceso unificado. Para hacer que estas ideas funcionen, se necesita un proceso polifacético, que tenga en cuenta ciclos, fases, flujos de trabajo, Gestión de Riesgos, Control de la Calidad, Gestión de Proyectos y Gestión de Configuración. [17]

Es preciso destacar que de acuerdo a la filosofía de RUP, todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al terminar el proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Por las características propias del proyecto, no se trabaja en todos los flujos de trabajo que propone la metodología RUP, específicamente no se trabaja en el Modelamiento del Negocio, en Requerimientos, así como tampoco se trabaja en Despliegue, por lo que los artefactos correspondientes a estos flujos de trabajo no constituyen entregables del proyecto. Conjuntamente con esta metodología de desarrollo se utiliza ApEM-L como lenguaje de modelado, por lo que aparecen nuevos artefactos definidos por este lenguaje y que también constituyen entregables del proyecto.

RUP para el Aseguramiento de la Calidad:

Existen buenas prácticas de aseguramiento de la calidad definidas por RUP donde se incluye la verificación continua de la calidad.

RUP provee planillas para la documentación durante todo el proceso de desarrollo del software, se divide en cuatro fases y describe varios flujos de trabajo de desarrollo y de soporte. Propone que los defectos detectados en las revisiones y formalizados en una Solicitud de Cambio durante el flujo de trabajo de Prueba, tendrán un seguimiento para asegurar su calidad. Además plantea que deben evaluarse todos los artefactos y las actividades que los producen. En particular, cuando el software ejecutable se produce, debe sujetarse a la demostración y prueba de guiones, importantes en cada iteración que proporciona un entendiendo más tangible de intercambios del plan y la eliminación más temprana de defectos arquitectónicos. Para ayudar en la evaluación del proceso y calidad del producto, RUP ha definido:

- **Actividades:** Una descripción de la actividad que debe ser realizada y los pasos para realizarla.
- **Pautas:** Son las técnicas y consejos prácticos útiles por realizar la actividad.
- **Pautas del artefacto y Puntos de control:** Cómo desarrollar y usar el artefacto, y comprobar si es el adecuado.

- **Plantillas:** Modelos o prototipos del artefacto que mantiene estructura y guía el volumen. [18]

El Proceso Unificado de Rational define los roles de trabajo así como las responsabilidades de cada uno y la creación de planes para asegurar la calidad en el desarrollo del software. Entre los roles que permitirán desarrollar las actividades de calidad dentro de un proyecto de Software se pueden destacar como los más importantes según RUP. [19]

- Diseñador de Pruebas
- Revisor Técnico
- Probador

Todas las actividades encaminadas al aseguramiento de la calidad desarrolladas por estos roles son planificadas en el plan del proyecto y es responsabilidad del Administrador del Proyecto. Se considera necesario estos roles planteados por la Metodología, como parte del equipo de calidad de "Contenidos Educativos Digitales".

1.5 Métricas de software

Paul Goodman en su libro *"Métricas del Software, Las Mejores Prácticas para el Éxito en la Gestión"* (Software Metrics-Best Practices for Successful IT Management) define las métricas de software como: *"La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo, así el desarrollador junto al empleo de estas técnicas mejorará el proceso y sus productos. Las métricas de software proveen la información necesaria para la toma de decisiones técnicas."* [20] Por su parte, el Dr. Andújar agrega que *"(...) el propósito de dichas métricas es permitir evaluaciones a través del ciclo de vida del software para asegurar que se alcanzan los requerimientos del producto y de la organización"*. [20]

Las métricas del software es un término que se asigna a un amplio rango de actividades diversas, por ejemplo:

- Medidas y modelos de estimación de coste y esfuerzo.
- Modelos y medidas de productividad.

- Aseguramiento y control de calidad.
- Recogida de datos.
- Medidas y modelos de calidad.
- Modelos de fiabilidad.
- Modelos y evaluación de ejecución.
- Complejidad computacional o algorítmica. [21]

En general, la medición persigue tres objetivos fundamentales:

- Entender qué ocurre durante el desarrollo y el mantenimiento.
- Controlar qué es lo que ocurre en nuestros proyectos.
- Mejorar nuestros procesos y nuestros productos.

Por esta razón es necesario dedicarle la mayor atención. La medición es parte de algo más amplio, es un enfoque de la organización hacia la mejora, es una actividad que requiere esfuerzo y preparación. Pero a pesar de los beneficios que pueda brindar la aplicación de las métricas no indican que todo esté bien, pues las mismas no son absolutas, simplemente proporcionan comprensión del proceso de software y no se puede identificar, explicar o predecir todo. Una vez aplicadas las mediciones no se debe estar conforme en el resultado que se obtiene a lo largo del desarrollo del software, para eso se necesita de una actividad de revisión y control sistemático y minucioso conocida como Revisión Técnica Formal.

1.6 Revisiones del software

Las revisiones de software son un “filtro” para el proceso de Ingeniería del Software y son aplicadas en varios momentos del desarrollo del software. Sirven para detectar errores y defectos que pueden ser eliminados, para “purificar” las actividades de la Ingeniería del Software que suceden como resultado del análisis, diseño y codificación, redundan en una mejora de la calidad del objeto que se examina, provoca indirectamente una mejora de la calidad del proceso de desarrollo al facilitar la comunicación entre los miembros del equipo de desarrollo, además facilitan el control del coste y el tiempo.

Dentro de las revisiones se encuentran las inspecciones que es donde los participantes van leyendo el documento, paso a paso, guiados por el autor, comprobando en cada paso el cumplimiento de los criterios

de una Lista de Chequeo (Checklist) o Lista de comprobación como también se le conoce. La lista de chequeo es una técnica que se utiliza para realizar las inspecciones, requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en la propia lista de chequeo. Esta lista será cumplimentada por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista. Al intentar dar respuesta a las preguntas formuladas deben salir a la luz los problemas que puedan existir. Cada revisor debe complementar la lista y anotar cualquier tipo de pregunta o defecto detectado. Existen diferentes tipos de revisiones del software, una de las más importantes es la Revisión Técnica Formal (RTF).

1.7 Revisiones Técnicas Formales (RTF)

Es el filtro más efectivo desde el punto de vista del aseguramiento de la calidad y es un medio eficaz para mejorar la calidad del software. [17] El objetivo principal de las RTF es encontrar errores durante el proceso, de forma que no se conviertan en defectos después de la entrega del software. El beneficio de las RTF es el descubrimiento de errores al principio para que no se propaguen al paso siguiente del proceso de software, otros de sus objetivos son:

- Descubrir errores en la función, la lógica o la implementación de cualquier representación del software.
- Verificar que el software bajo revisión alcance sus requerimientos.
- Asegurar que el software haya sido representado de acuerdo con ciertos estándares predefinidos.
- Obtener un software desarrollado de forma uniforme.
- Hacer que los proyectos sean más manejables.

Las RTF son una actividad colectiva de garantía de la calidad del software lo que permite ampliar la mirada sobre lo que se revisa. Estas agrupan en sí diversos mecanismos de revisión y permiten establecer un marco común para la definición de distintas etapas de revisión en el ciclo de vida del software, este mecanismo no sólo está pensado para las etapas tempranas del ciclo de vida sino que también puede y debe ser utilizado en períodos como el de prueba y el de mantenimiento.

La orientación de la RTF debe ser muy específica, en el sentido que cada cual se debe centrar en una parte muy bien delimitada del software total. Por ejemplo, en lugar de intentar revisar un diseño completo, se hacen inspecciones para cada módulo o pequeño grupo de módulos. Al limitar el centro de atención de la RTF, la probabilidad de descubrir errores es mayor. Las revisiones técnicas pueden llevarse a cabo después del desarrollo de las especificaciones, del análisis y diseño y del Test de prueba.

Las revisiones se basan en verificación y validación, esto implica la tarea de asegurar que el sistema está conforme a especificaciones y acorde a las necesidades del cliente. Cuando se verifica, se hace contra especificaciones del software, o sea, determinar si se ha construido de forma correcta. Cuando se valida, es contra requerimientos de usuario, aquí se revisa si se ha construido el producto correcto.

Las técnicas de revisión pueden ser estáticas, aquí se analizan y chequean los documentos de requerimientos, los diagramas de diseño, código fuente, entre otros y también pueden ser dinámicas, que son pruebas que se aplican sobre implementación real (solo pueden usarse cuando ya se tiene código ejecutable).

Cada RTF se lleva a cabo mediante una reunión y solo tendrá éxito si es bien planificada, controlada y atendida, independientemente de cualquier formato que se elija, dicha reunión debe acogerse a las siguientes restricciones:

- Deben convocarse normalmente entre 3 y 5 personas.
- Se debe preparar por adelantado, pero sin que requiera más de 2 horas de trabajo a cada persona.
- La duración de la reunión debe ser menor de 2 horas.

Como todo procedimiento, siempre se necesita de pasos específicos para llevar a cabo ciertas actividades. En primer lugar se seleccionan los revisores de forma que se cubran todos los ámbitos (futuro mantenimiento, ajuste a los estándares de la organización, usuarios y corrección y calidad del producto).

A partir de aquí, los pasos que se siguen son:

- 1) El productor del software solicita una revisión al jefe del proyecto.
- 2) El jefe de proyecto contacta con un jefe de revisión que:
 - Evalúa la disponibilidad del producto, genera copias del material del producto y las distribuye a dos o tres revisores.

- 3) Los revisores y el jefe de revisión revisan el producto (una o dos horas). El jefe de revisión elabora una agenda para la reunión.
- 4) Se reúnen el jefe de revisión, los revisores (uno actuará como secretario) y el productor.
 - La RTF comienza con una explicación de la agenda (jefe de revisión) y una breve introducción (productor).
 - Posteriormente se discuten las “pegas” que cada revisor ha encontrado. El secretario tomará las notas oportunas.
 - La duración de la reunión debe ser inferior a dos horas.
- 5) Al final decidirán si:
 - Aceptan el producto (no se hacen modificaciones).
 - Rechazan el producto (existen errores graves).
 - Aceptan el producto modificándolo (corregir errores).
- 6) Todos los revisores firman la conformidad de los resultados de la revisión.

Ventajas:

- Más efectivo que las pruebas para encontrar errores.
- Encuentra la causa del error en lugar del síntoma.
- Los programadores saben que sus programas serán revisados.
- Programas más fáciles de entender y programadores más cuidadosos.
- Anula el efecto de “puntos ciegos” (el programador pasa reiteradamente sobre el error sin verlo).
- Es posible imponer estándares de codificación con facilidad.
- Explicar el código hace que el programador lo entienda cada vez mejor.
- Reduce dramáticamente tiempo de pruebas.
- Sirven para promover la seguridad y la continuidad, ya que varias personas se familiarizarían con partes del software que de otro modo no hubieran visto nunca.
- Sirven como campo de entrenamiento, permitiendo que los ingenieros más jóvenes, puedan observar los diferentes enfoques de análisis, diseño e implementación del software.

Desventajas:

- Problemas de personalidad.
- Personas con buenas ideas no se expresan.
- Personas con malas ideas se expresan mucho.
- Algunas personas odian discutir o estar en desacuerdo.
- Fácil de perderse en cosas triviales (punto y coma).
- Es agotador (pierde efectividad después de un par de horas o menos).
- Dificultades para convencer a gerentes de beneficios económicos.

Para que se realice una revisión con calidad hay que establecer una serie de directrices que conducirán a los revisores para llegar a un consenso final. Es por eso que; se debe revisar el producto y no al productor, o sea, se deben destacar los errores educadamente y con un tono constructivo. Fijar una agenda y mantenerla es otro punto importante pues es conveniente seguir un plan de trabajo concreto y sin divagaciones.

Está establecido que una revisión no es una sesión de resolución de problemas, es por eso que se deben enunciar áreas de problemas pero no intentar resolverlos. El tomar notas por escrito puede ayudar que las declaraciones o la asignación de prioridades puedan ser comprobadas por el resto de los revisores.

Otras directrices que influyen en la realización de una buena revisión técnica formal son: desarrollar una lista de comprobación para cada producto que haya de ser revisado, disponer recursos y una agenda para las RTF, llevar a cabo un buen entrenamiento de todos los revisores, así como, repasar las revisiones que se van realizando, para descubrir problemas en el propio proceso de revisión.

1.8 Pruebas de software

La prueba de software es un elemento crítico e imprescindible para la garantía de la calidad, y de ahí la necesidad de aplicarla. Se llama prueba de software al *“proceso en el que se ejecuta un sistema con el objetivo de detectar fallos. Es un proceso destructivo que determina el diseño de los casos de prueba y la asignación de responsabilidades.”* [22]

Las pruebas de software se pueden traducir como una revisión del sistema con el objetivo de encontrar problemas antes que éstos sean encontrados por el cliente final. El éxito de las pruebas puede mejorar la

percepción de calidad del usuario final, evitando que un defecto grave se entregue conjuntamente con el sistema.

La expresión “calidad” está muy relacionada a la percepción de los involucrados. Es por eso que solemos observar la calidad a través de un conjunto de dimensiones. Son ejemplos de las dimensiones de la calidad:

- **Facilidad de Uso:** Por ejemplo: Poner a prueba la aplicación desde el punto de vista de la conveniencia para el usuario.
- **Funcionalidad:** Por ejemplo: Poner a prueba el comportamiento de la aplicación en escenarios definidos.
- **Fiabilidad:** Por ejemplo: Poner a prueba si la aplicación se porta de forma consistente y previsible.
- **Facilidad de Mantenimiento:** Por ejemplo: Poner a prueba la capacidad de mantener y proporcionar soporte a la aplicación en el entorno de producción.
- **Desempeño:** Por ejemplo: Poner a prueba el tiempo de respuesta en una carga normal y en el escenario de pico de utilización.

La prueba de software es un mecanismo para evaluar la calidad de acuerdo a las dimensiones descritas anteriormente. [23]

Todo proceso de desarrollo de software se encuentra sujeto a un mal entendimiento de los requerimientos, a una mala comunicación por parte de los usuarios y a equivocaciones por parte de sus desarrolladores por lo que se hace necesario controlar lo que se realiza desde que se empieza a elaborar un producto. Esto se logra utilizando las pruebas, las cuales sirven para combatir estos errores y lograr calidad en lo que se está realizando por parte de un determinado equipo de trabajo.

1.9 Mejora de la Calidad

“La mejora de la calidad es la parte de la gestión de la calidad orientada a aumentar la capacidad de cumplir con los requisitos de la calidad. Los requisitos pueden estar relacionados con cualquier aspecto tal como la eficacia, la eficiencia o la trazabilidad.” [24] El objetivo de la mejora continua del sistema de gestión de la calidad es incrementar la probabilidad de aumentar la satisfacción de los clientes y de otras partes interesadas. Las siguientes son acciones destinadas a la mejora:

- El análisis y la evaluación de la situación existente para identificar áreas para la mejora.
- El establecimiento de los objetivos para la mejora.
- La búsqueda de posibles soluciones para lograr los objetivos.
- La evaluación de dichas soluciones y su selección.
- La implementación de la solución seleccionada.
- La medición, verificación, análisis y evaluación de los resultados de la implementación para determinar que se han alcanzado los objetivos.
- La formalización de los cambios.

Los resultados se revisan, cuando es necesario, para determinar oportunidades adicionales de mejora. De esta manera, la mejora es una actividad continua. La información proveniente de los clientes y otras partes interesadas, las auditorías, y la revisión del sistema de gestión de la calidad pueden, asimismo, utilizarse para identificar oportunidades para la mejora. [25]

Mejora continua: La organización debe mejorar continuamente la eficacia del sistema de gestión de la calidad por medio de la utilización de la política de la calidad, objetivos de la calidad, resultados de las auditorías, análisis de datos, acciones correctivas y preventivas y la revisión por la dirección.

1.10 Enfoques del aseguramiento de la calidad

1.10.1 El Aseguramiento de la calidad según el modelo ISO 9000-3

Según la norma ISO 9000-3 el propósito del aseguramiento de la calidad es proporcionar una adecuada seguridad de que los productos de software y los procesos en el ciclo de vida del proyecto están conformes con sus requerimientos específicos y se ajustan a sus planes establecidos. El aseguramiento de la calidad puede ser interno o externo dependiendo de si la evidencia de la calidad del producto o del proceso se demuestra a la gerencia del proveedor o del cliente. Este proceso implementa varias actividades como:

- **Implementación del proceso:** Esta actividad cuenta con varias tareas como: establecer un proceso de aseguramiento de la calidad ajustado al proyecto. Elaborar, documentar, implementar y mantener actualizado un plan de ejecución de las actividades y tareas del proceso. Implementar

normas, metodologías, procedimientos y herramientas para asegurar la calidad. Proporcionar al cliente los registros de las actividades y tareas de aseguramiento de la calidad.

- **Aseguramiento del producto:** Esta actividad cuenta con varias tareas como: asegurar que todos los planes requeridos por el contrato estén documentados. Asegurar que todos los productos de software usados y la documentación relacionada con ellos cumplan lo establecido. En la preparación de la entrega de los productos, deberá ser asegurado que ellos satisfacen totalmente sus requerimientos contractuales y que son aceptables para el cliente.
- **Aseguramiento del proceso:** Esta actividad cuenta con tareas como: asegurar que los procesos del ciclo de vida del producto cumplen con lo establecido y se ajustan a los planes. Asegurar que las prácticas internas del diseño del producto, del ambiente de desarrollo y del ambiente de prueba cumplen lo establecido en el contrato.
- **Aseguramiento del sistema de la calidad:** Esta actividad cuenta con tareas como: asegurar actividades adicionales de gestión de la calidad de acuerdo con las cláusulas de la norma ISO 9001.

1.10.2 El Aseguramiento de la calidad según el modelo Tick-IT

Para este modelo el aseguramiento de la calidad es nombrado: Soporte y Aseguramiento de la Calidad. Al igual que los demás modelos Tick-IT incluye en esta área una serie de procesos a realizar con el fin de asegurar la calidad del proyecto. Uno de estos procesos es:

Aseguramiento de la Calidad:

- Establecer políticas y objetivos de calidad generales de la organización que sirvan para alinearla en todas sus actividades, procedimientos y políticas específicas.
- Implantar y mantener un sistema de aseguramiento de calidad.
- Auditorías, revisiones y acciones correctivas al sistema de calidad que aseguren que el sistema cumple con los requerimientos, es utilizado y que es efectivo en el logro de resultados.
- Definir, recolectar y analizar datos de calidad para evaluar la efectividad del sistema de calidad e identificar mejoras potenciales.
- Administración de la organización y los proyectos de tal forma que facilite los resultados de calidad.

- Administración de la configuración que identifique y controle, de manera continua, las partes constituyentes y sus versiones, de cada instancia de un sistema o subsistema.
- Respaldos, seguridad y almacenamiento que protejan contra cualquier pérdida o corrupción.
- Sistema de control de registros y documentación para todas las actividades de aseguramiento de calidad, de los proyectos y de soporte, incluyendo procedimientos y registros.
- Especificación y control del proceso de desarrollo incluyendo técnicas, prácticas, convenciones, estándares, mediciones y estadísticas.
- Proceso de compras, incluyendo identificación, selección, adquisición y aceptación que asegure que los bienes y servicios adquiridos sean como se requiere y de calidad aceptable.
- Control de productos incluidos, equipo y herramientas utilizadas: Hardware o Software, adquiridos o suministrados por el cliente, incluyendo utilización, configuración, seguridad.
- Entrenamiento, reclutamiento y desarrollo de personal que asegure su competencia y motivación, y disminuya su rotación.

1.10.3 El Aseguramiento de la calidad según el modelo ISO-SPICE

El aseguramiento de la calidad es tratado dentro de la categoría de proceso de soporte. Los procesos de soporte pueden ser empleados en varios puntos del ciclo de vida y pueden ser realizados por la organización que los emplea, el cliente o por una organización independiente.

El propósito de realizar el aseguramiento de la calidad es asegurar que los productos y actividades estén de acuerdo con los estándares, procedimientos y requerimientos aplicables. El requerimiento clave aquí es que se determine e informe una visión objetiva de la calidad del proceso y los productos. El aseguramiento de la calidad puede implementarse de diferentes formas: realizar un grupo independiente para el aseguramiento de la calidad o con los mismos integrantes del proyecto. En este modelo existe otro proceso con un nombre similar, "Gestionar la calidad", que se centra en identificar lo que necesita hacerse para construir calidad en los productos y establecer los controles de gestión para asegurar que está hecho; mientras que el aseguramiento de la calidad se centra más en un enfoque de auditoría y revisión y en el aseguramiento del cumplimiento.

Actividades realizadas como parte del aseguramiento de la calidad según este modelo:

- **Seleccionar los estándares del proyecto:** Contribuir a los planes software del proyecto, evaluando la completitud de los planes y ayudando a seleccionar los estándares y procedimientos que se utilizarán en el proyecto.
- **Revisar las actividades de ingeniería del software:** Revisar las actividades de ingeniería del software frente a los planos, estándares y procedimientos seleccionados.
- **Auditar los productos:** Auditar los productos software frente a los estándares y procedimientos seleccionados.
- **Informar de los resultados:** Informar los resultados de las actividades anteriores, en particular, de las desviaciones, a los niveles apropiados de dirección y plantilla.
- **Tratar las desviaciones:** Las desviaciones se tratan en el nivel de gestión apropiado, escalando el siguiente nivel, cuando sea necesario, hasta resolverlas.

1.10.4 El Aseguramiento de la calidad según el modelo CMMI

CMMI nombra esta área como Aseguramiento de la Calidad de los procesos y los productos del trabajo. El objetivo de dicho modelo en esta área de procesos es proporcionar personas y gestión, con el objetivo de que los procesos y los elementos de trabajo cumplan los procesos establecidos. Esto se consigue mediante las siguientes actividades:

- Evaluar objetivamente la ejecución de los procesos, los elementos de trabajo y servicios contra las descripciones de procesos, estándares y procedimientos.
- Identificar y documentar los elementos no conformes.
- 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.
- Asegurar que los elementos no conformes sean arreglados.

El modelo CMMI para asegurar la calidad exige 2 objetivos específicos, estos a su vez incluyen dos prácticas específicas las cuales están compuestas por sub-prácticas que en conjunto conforman el área de procesos.

Objetivo Específico 1: Evaluar objetivamente los procesos y los productos del trabajo

- Evaluar objetivamente los procesos.
- Evaluar objetivamente los productos y los servicios del trabajo.

Objetivo Específico 2: Proporcionar un entendimiento objetivo

- Comunicar y asegurar la resolución de las no conformidades emitidas.
- Establecer los registros.

Esta es un área de proceso clave, que a veces no se le da la suficiente importancia, pero que sin ella es imposible implantar un modelo de calidad.

1.10.5 Análisis comparativo del aseguramiento de la calidad según los enfoques de los modelos estudiados

Todos los modelos investigados presentan de forma general las mismas características generales en el área de aseguramiento de la calidad. Sin embargo presentan diferencias como por ejemplo: el aseguramiento de la calidad de la norma ISO 9000-3 proporciona confianza a los clientes ya que integra varias normas de estándares, además evita rectificar la calidad según los estándares locales o particulares de la empresa, pero muchas veces se hace de manera impuesta para los trabajadores del proyecto. Por otra parte no es indicativa de la calidad de los productos, procesos o servicios de la empresa. Esta norma da un enfoque del aseguramiento de la calidad muy general ya que en su principio fue concebida para cualquier tipo de empresa. En esta área se presta mucha atención al cliente y no es del todo eficaz en el proceso de producción. En este modelo se requiere mucho de auditorías externas.

Por otra parte el modelo Tick IT propone implantar un sistema de control de calidad pero no de forma que todos los trabajadores del proyecto se vean involucrados y responsables de la calidad del proyecto en sus procesos, sus productos y sus servicios. Es importante resaltar que este modelo tiene bien presente el almacenamiento y la seguridad de todos los registros. Aquí el aseguramiento de la calidad se trata de una manera genérica en cuanto a las evaluaciones, no se tiene en cuenta qué hacer para las evaluaciones.

El modelo ISO-SPICE tiene varias ventajas sobre los modelos ISO 9000-3 y el modelo Tick-IT en cuanto al tratamiento que le dan al aseguramiento de la calidad aunque el mismo hasta el cambio del nivel 4 para

el nivel 5 no propone actividades concretas para un mejoramiento continuo de la calidad, lo que provoca que la empresa se demore más en alcanzar los 5 niveles expuestos por dicho modelo. Al igual que el modelo ISO 9000-3, esta norma depende de otras normas y esto puede provocar malas interpretaciones a la hora de aplicarlo ya que son en algunos casos genéricas y no siempre son específicas para la producción de software.

El Modelo Integrado de Capacidad y Madurez fue concebido para la producción de software y tiene técnicas para ser independiente e incremental en su avance, lo cual motiva al personal, a diferencia del modelo ISO 9000-3 que prácticamente el único reto que tiene es el logro de la certificación. Este modelo se enfoca al proceso de producción de forma más específica y por tanto más eficiente. Además mueve al sistema de aseguramiento de la calidad de la empresa en una dirección de mejora continua en cualquier nivel y puede ser flexible en su implantación y además le aporta técnicas de auto evaluación de la calidad. Como en los demás modelos y las otras áreas de este mismo modelo aquí se indican qué actividades realizar pero no cómo hacerlas. En caso de proyectos muy pequeños la cantidad de documentación que se exige puede ser un atraso de tiempo aunque este modelo es flexible en su adaptación al proyecto. Los niveles de madurez no garantizan del todo el éxito ya que puede haber un proyecto de nivel 1 y ser exitoso, pero sí da una medida del atributo madurez de la calidad de la empresa.

1.11 Estado actual del uso de los modelos de calidad

1.11.1 A nivel internacional

En la economía moderna, la industria del software es la piedra angular de toda innovación tecnológica, ya que impacta en forma directa o indirecta sobre todas las actividades económicas. La demanda en el mundo entero de software y servicios informáticos se ha incrementado significativamente a medida que las computadoras y la Internet penetran cada vez más en todos los aspectos de la sociedad. Los países desarrollados y en desarrollo están en un acelerado proceso de informatización cada vez más amplio, que exige la más variada selección de soluciones de alta calidad.

La industria de software y servicios informáticos ha sido una de las más dinámicas a escala global en los últimos años que juega un papel clave dentro de las llamadas Tecnologías de la Información y las Comunicaciones (TICs).

La creciente necesidad, sumada a décadas de promesas incumplidas en cuanto a calidad, confiabilidad, costos y cumplimiento en el desarrollo de software condujo a muchos países a involucrar modelos de calidad en la industria del software. Si bien el grueso de la producción de software se concentra en los Estados Unidos, Japón y en los países más avanzados del continente europeo, algunas naciones en desarrollo de Asia y América Latina, así como otras de la periferia europea, han hecho significativos avances dentro del sector. La siguiente imagen muestra la producción de los países de entrada tardía para el año 2001. [26]

	Ventas	Exportaciones	Coefficiente X/Ventas	Empleo	Nº empresas
India	5.700	4.000	70%	410.000	1.250
Irlanda	6.245	5.907	94%	18.300	679
Israel	1.500	700	47%	20.000	300
Brasil	8.038	40	<1%	s.d.	2.500
Uruguay	180	60	33%	2.500-3.000	150
Argentina	1.340	35	<3%	15.000	500
Costa Rica	s.d.	50	s.d.	3.500-4.000	150
Chile	125	15	12%	s.d.	s.d.
Singapur	1.660	476	29%	s.d.	s.d.
China	3.000	s.d.	s.d.	100.000	2.000
Corea	6.000	96	<2%	s.d.	s.d.

Fig. 2 Producción de software de países de entrada tardía al mercado mundial [26]

La industria de software en los países en desarrollo se caracteriza por un mayor peso del sector servicios en relación al de productos. Entre los componentes clave para una estrategia exitosa de avance del sector, se menciona a la capacidad de competir vía costos, buen marketing, y mecanismos de networking con otras firmas de software y con clientes, inversores, etc. tanto del país como del exterior.

La evolución del software apunta al fortalecimiento del monopolio de los países desarrollados en el ámbito de la informática, sobre todo mediante la ampliación de la patentabilidad del software. Los países en mejores condiciones de alentar la investigación y desarrollo en informática son beneficiarios casi exclusivos del sistema internacional de patentes. Bajo la sombrilla protectora del *software* por la vía del derecho de autor, los creadores de programas de computación han obtenido decisiones jurisprudenciales e instrumentos contractuales que los protegen, al tiempo que desalientan en los países en desarrollo la voluntad de realizar nuevas e importantes inversiones para la creación de nuevos programas de computación.[26]

La industria del software argentino, activada por la devaluación durante la crisis económica de 2001, ha logrado cosechar en los últimos años importantísimos logros. El aumento en el número de empresas y de puestos de trabajo, así como el crecimiento de las exportaciones, entre otros aspectos, dan cuenta de este fenómeno. Creatividad, rapidez, capital humano y tecnológico son los principales secretos por los que el software argentino aumentó sus ventas un 20% entre 2002 y 2003, incorporó un 33% más de empleados en el mismo período y repuntó sus exportaciones ni más ni menos que un 42%, según datos oficiales de la Cámara de Empresas de Tecnología de la Información de Argentina. [27]

1.11.2 En Cuba

En Cuba, el Ministerio de la Informática y las Comunicaciones (MIC), es el organismo encargado de regular, dirigir, supervisar y controlar la política del Estado y del Gobierno en cuanto a las actividades de tecnologías informáticas, telecomunicaciones, redes de infocomunicaciones, entre otras. El cual, entre sus principales funciones tiene el establecimiento, regulación y control de la política para el desarrollo, producción y comercialización de la Industria Nacional de Software.

La Industria Cubana del Software está llamada a convertirse en una significativa fuente de ingreso para el país, como resultado del correcto aprovechamiento de las ventajas del alto capital humano disponible. La promoción de la industria cubana del software 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. El continuar la producción sostenida de software de alta calidad en prestaciones, imágenes y soporte, para satisfacer las necesidades nacionales en estos sectores, tendrá una positiva repercusión en el incremento de la exportación.

La calidad en el software es algo que está naciendo recientemente aunque hay muy buenas perspectivas. Hace un tiempo ya se conocía la necesidad de controlar de alguna forma la calidad de los productos que iban saliendo aunque eran pocos, pero se hacía poco por esto y como resultado se tenía que al final los productos no tenían calidad o al menos no la más óptima. Actualmente opera el Comité Técnico CTN 18 "Tecnología de la Información", que preside el Ministerio de la Informática y las Comunicaciones, entre otros aspectos, conduce investigaciones llamadas a promover el empleo de las normas internacionales y la aplicación de métodos de evaluación de la conformidad, tanto por medio de pruebas de calificación de software, como en los procesos de un laboratorio independiente para la evaluación de la conformidad y certificación del software. [28]

Producto de su acción, han ocurrido avances en la definición de las características de la calidad al nivel del producto aunque para lograr un alto nivel de gestión de la calidad en las empresas es necesario que estas tengan como objetivo lograr una mayor eficiencia y no solo ser certificadas. Una de las empresas punteras en este sector es Desoft, su interés por lograr la calidad crece de forma continua a medida que los clientes se vuelven más selectivos y comienzan a rechazar los productos poco fiables o que realmente no dan respuesta a sus necesidades, se rigen por las normas cubanas ISO 9000:2005, 90003:2004 y 12207:1995, pudiendo así cumplir con las expectativas de sus clientes. [29]

Otra empresa de gran prestigio lo es sin duda el Centro de Calidad de Software (Calisoft), surgida en el año 2004 con el objetivo de diseñar sistemas informáticos de alto nivel, sus principales objetivos son: entregar productos de gran calidad con estricta puntualidad, para lograrlos cuentan con un grupo de profesionales altamente capacitados y con amplia experiencia garantizando así la exitosa entrega del proyecto. Contribuye en gran medida al desarrollo de la industria de software en el país incrementando su competitividad mercado internacional. Además se conoce del apoyo de muchas organizaciones que antes no tenían un papel relevante en este sentido en nuestro país y se han visto obligadas a utilizar de una forma u otra alguna metodología que les pueda certificar sus productos. [30]

1.11.3 En la Universidad de la Ciencias Informáticas (UCI)

La Universidad de las Ciencias Informáticas (UCI) es un centro productivo, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación. Es considerada la mayor organización productora de software en el país.

En la actualidad el centro está acometiendo un proyecto de mejora de sus procesos basado en el modelo CMMI (Capability Maturity Model Integration) bajo los servicios del SIE Center (Software Industry Excellence Center) del Tecnológico de Monterrey.

El proceso de mejora está encaminado a que la universidad alcance en el 2010 una certificación internacional del nivel 2 del modelo CMMI. Hecho que la convertiría en la primera empresa cubana certificada con este modelo y una de las pocas en el área del Caribe.

CMMI es un modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de Administración como de Ingeniería en Sistemas y Software. Con su instauración se espera alcanzar beneficios como:

- Calendarios y presupuestos predecibles en los proyectos.
- Mejora dentro del ciclo de vida dentro del desarrollo de software.
- Mayor productividad.
- Mayor calidad de los productos y servicios que ofrece la universidad a sus clientes y por ende la satisfacción de los mismos.
- Mejorar la moral del personal que labora en el centro.

El servicio que ofrece el SIE Center permite:

- Ayudar a la UCI a revisar su estrategia de mejora de procesos de software, para asegurar que su organización esté basada en procesos y con un programa de mejora continua alineado con sus objetivos de negocio.
- Ayudar a la UCI a establecer las bases y fundamentos para seguir mejorando sus procesos y fortalecer su cultura de calidad en el desarrollo de software.
- Alinear los procesos de desarrollo de software con los principios y requerimientos del modelo CMMI, estableciendo planes de mejora con los que la organización oriente sus procesos hacia la consecución de sus metas. [31]

1.12 Estudio sobre modelos de Calidad

1.12.1 ¿Qué son los modelos de calidad?

Los modelos de calidad no son más que las técnicas, herramientas y metodologías que le facilitan a las empresas que se encargan de la fabricación de software guiar por un camino único el avance de dicho proceso y así lograr que se cumplan con los requerimientos iniciales pedidos por el cliente ya que esa es la base de la calidad de un producto. Existen varias metodologías y herramientas a seguir para la realización de un proyecto pero el objetivo de todas ellas es organizar el proceso para poder realizar un software con mayor calidad. Estas herramientas son las que pretenden dar solución a los problemas de calidad que se han presentado a lo largo de la historia del desarrollo de software por la falta de una guía en el proceso de desarrollo. Se debe tener presente que la aplicación de estas metodologías puede cambiar de un proyecto a otro y que no se puede decir con seguridad cual es la mejor, ya que esto

depende de las necesidades del proyecto, una metodología puede ser la ideal para un proyecto pero para otro distinto puede resultar poco abarcadora y puede que controle áreas de procesos que no están implícitas en dicho proyecto. Toda buena metodología contiene al menos buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos. [32]

1.12.2 Ventajas y necesidades del uso de los modelos de calidad

La aplicación de un modelo de calidad en la realización de un proyecto aporta varias ventajas, primeramente ayuda a asegurar la calidad del software, con esto se garantiza la confianza en los resultados, además es la mejor manera de asegurarse de que se cumplan los requerimientos iniciales que pidió el cliente consiguiendo una mayor satisfacción del mismo; también se reducen los errores en explotación logrando así una mayor fiabilidad del software; con la aplicación del modelo en todas las etapas de vida se evitan los riesgos de que aparezcan errores en la etapa de pruebas finales que pueden estar vinculados con el diseño y resulta fundamental la ayuda en cuanto a la organización del flujo de trabajo en el proyecto y a la seguridad de que se esté trabajando por buen camino.

Estos modelos convierten la calidad del software en algo que se puede medir, definir y planificar. Ayudan a mejorar la comunicación entre usuarios, dirección y técnicos. Al cliente le place mucho más saber que el software que va a adquirir está certificado por alguna metodología que controle la calidad en todas las etapas de vida de un software, esto da una mayor confianza en el resultado del trabajo esperado.

1.12.3 Características de los modelos de calidad

Para escoger un modelo para la realización de un proyecto se debe hacer un estudio de los modelos que sean adaptables al proyecto y de ellos escoger el que asegure mejor la calidad atendiendo al costo y el rendimiento del producto entre otros parámetros que se estimen importantes.

Después de haber hecho un estudio de los modelos de calidad, se presenta una descripción de las principales características del modelo escogido el cual fue CMMI.

Características de Modelo Integrado de Capacidad y Madurez (CMMI):

El CMM - CMMI es un modelo de calidad del software que clasifica las empresas en niveles de madurez. Estos niveles sirven para conocer la madurez de los procesos que se realizan para producir software.

Surge como resultado de la integración de varios CMM que se aplicaban de forma simultánea. CMM-SW, SE-CMM, IPD-CMM, son los principales modelos que se encuentran integrados en CMMI. Este modelo mantiene el mismo principio de CMM acerca de que la calidad de un producto o sistema es en su mayor parte consecuencia de la calidad de los procesos empleados en su desarrollado y mantenimiento.

Para el Modelo Integrado de Capacidad y Madurez (CMMI) la madurez es el atributo de las organizaciones que desarrollan software. En la medida que estas llevan a cabo su trabajo siguiendo procesos, en la forma que una empresa mejore y tenga mejores resultados tendrá un nivel de madurez mayor. También tiene Capacidad como atributo de los procesos. El nivel de capacidad indica si solo se ejecuta o si también se planifica, además si se encuentra formalmente definido, se mide y se mejora de forma sistemática. El Modelo Integrado de Capacidad y Madurez (CMMI) plantea 6 niveles para medir la capacidad de los procesos, 0-Incompleto, 1- Ejecutado, 2- Gestionado, 3-Definido, 4-Cuantitativamente gestionado, 5-Optimizado.

Existen 2 enfoques: Continuo y Escalonado. El Enfoque Continuo hace hincapié en la capacidad de ciertas áreas para realizar sus actividades de manera adecuada. El Enfoque Escalonado hace especial énfasis en el grado de madurez de los procesos.

El Modelo Integrado de Capacidad y Madurez (CMMI) identifica 25 áreas de procesos en sus 5 niveles. En el nivel 2 identifica las áreas de procesos: Gestión de configuración, Medición y Análisis, Monitoreo y Control, Planificación, Aseguramiento de la Calidad, Gestión de requerimientos, Gestión de acuerdo con proveedores. En cada área CMMI identifica objetivos generales y prácticas específicas, así como Sub-prácticas que serían las actividades necesarias para lograr los objetivos de dicha área.

El Modelo Integrado de Capacidad y Madurez (CMMI) es además una certificación para valorar la madurez que puede tener una empresa en la organización y desarrollo de software y valorar la capacidad de la empresa. [33]

Ventajas del Modelo Integrado de Capacidad y Madurez (CMMI):

Primeramente asegura que todo el personal se vea vinculado con las tendencias de la calidad del producto. Insiste en el uso de métricas. Detalla las áreas de procesos relativas a la ingeniería. Exige que se tenga una documentación bien detallada del proyecto. Reduce costos en un 20 % promedio. Reduce tiempo en un 37 % promedio. Aumenta la productividad en un 62 % promedio. Aumenta la calidad en un 50 % promedio. Satisfacción del cliente en un 14 % promedio.

Además de la ventaja directa e indirecta en costos, la eficiencia y agilidad de compartir la cultura y conocer las prácticas empresariales locales, son algunas de las principales ventajas.

Conviene señalar que la sigla CMMI responde en inglés a “Modelo de Madurez de Capacidades Integrado”. Además de asegurarse de “estar haciendo bien las cosas”, una empresa que obtiene el nivel de madurez CMMI cuenta con una evidencia objetiva respecto de la efectividad de sus procesos de desarrollo, operación y mantenimiento de software, que es reconocida internacionalmente como una ventaja competitiva distintiva. CMMI se posiciona, entonces, como un componente fundamental para las empresas de desarrollo locales decididas a generar negocios sustentables a nivel internacional.

CMMI, por otra parte, es una gran recopilación de experiencias y soluciones a casos concretos compaginada como una colección estructurada de mejores prácticas. Cuenta con una base totalmente práctica y ha sido generada por la misma comunidad informática internacional. Durante la evaluación se verifica de qué manera la organización ha implantado las mejores prácticas contenidas en el modelo, y se otorga un nivel de madurez que avala los procesos utilizados.

CMMI destaca, en las empresas que desarrollan software, la voluntad de mejorar las cosas y reducir su dependencia de los “héroes”. ¿Qué quiere decir esto? Que los proyectos salen bien porque las cosas se hacen bien y no porque empleados ‘estrella’ resuelven las deficiencias de los demás. Por eso, un proveedor de software con nivel de madurez CMMI es muy importante para una empresa, gracias a su mayor predictibilidad y visibilidad dentro del desarrollo. Y en el mundo ya son muchísimas las empresas que lo exigen, y a nivel local cada vez serán más.

1.13 Conclusiones del capítulo

- La fórmula para llegar al éxito en la Industria del Software es precisamente desarrollando productos de alta calidad, quedando demostrado a través de las certificaciones que ofrecen los estándares de calidad estudiados en este capítulo, siendo estos una herramienta de credibilidad a nivel mundial.
- El Aseguramiento de la Calidad y la estrategia que se decida para alcanzarla son puntos vitales a tener en cuenta durante todo el proceso de desarrollo del software. Son de sumo interés para todo el equipo de trabajo, durante todas las etapas de desarrollo, y gracias a dicha estrategia será posible la máxima calidad y la satisfacción del cliente.

- Se estudiaron todos los factores que se relacionan con la calidad del producto, en vistas conocer las características que debe presentar un software para ser considerado con buenas condiciones de calidad.

Capítulo

2

Capítulo 2. Análisis de la Situación Actual

2.1 Introducción

Para la exitosa realización de la Estrategia para el Aseguramiento de la Calidad es indispensable antes realizar un estudio de la situación actual en la que se encuentra el proyecto, por lo que en este segundo capítulo se analizará el estado real que presenta el proyecto productivo “Contenidos Educativos Digitales”. Primeramente se abordarán los antecedentes que dieron lugar a que se concibiera el proyecto, para luego describir el surgimiento del mismo, su estructura interna y otros aspectos de interés para la presente investigación.

2.2 Contenidos Educativos Digitales. Antecedentes

La Revolución Bolivariana ha identificado como sus ejes estratégicos la educación, los valores y la cultura liberadores e integradores; por esta razón, en el año 2007 se impulsó el llamado “tercer motor”: la Educación con Valores Socialista, identificada como la “Gran Jornada Nacional Moral y Luces”. Ésta es una de las cinco líneas transformadoras del Proyecto Simón Bolívar que junto a la Ley Habilitante, la Reforma Constitucional, la Nueva Geometría del Poder y la Explosión del Poder Comunal, tienen como objetivo la transformación política, social, económica, militar, territorial y ética de la República Bolivariana de Venezuela.

Un pueblo sin historia ni conciencia, sigue, ciego, el camino hacia su autodestrucción. Es tarea fundamental, conocer e interpretar correctamente los procesos, los personajes que dieron forma y contenido a la esencia venezolana y latinoamericana luego de años de falsa educación impartida por las oligarquías herederas de una patria sin rumbo debido a la traición y prematura desaparición de nuestros grandes Libertadores.

En el marco del Convenio de Cooperación Educativa Cuba-Venezuela de Gestión y Producción de Contenidos Educativos Digitales, y en un esfuerzo por divulgar los valores socioculturales de ambas naciones, el Ministerio de Ciencia y Tecnología, a través del Centro Nacional de Innovación Tecnológica (CENIT) y las Cooperativas Venezolanas, decide desarrollar una serie de Contenidos Educativos Digitales, con el objetivo de dar a conocer a una mayor cantidad de personas dentro de la población venezolana información sobre la historia de su país.

2.3 Descripción

La Facultad 8 de la UCI desarrolla principalmente proyectos con propósitos educativos. Entre los que se encuentran actualmente en desarrollo está “Contenidos Educativos Digitales”, el cual constituye un medio informativo, formativo e interactivo de fácil manejo que enseña al usuario la Historia de Venezuela y lo familiariza con las herramientas multimedia al mismo tiempo. El proyecto se basa en digitalizar mediante una plataforma web un grupo de información que les servirá a los usuarios para conocer acerca del patrimonio cultural venezolano. Los objetivos generales del proyecto son:

- Divulgar información de utilidad práctica sobre temas diversos a las comunidades en edad escolar, docentes, psicopedagogos y el público en general.
- Proteger parte del patrimonio cultural nacional representado por los juegos, costumbres y tradiciones de Venezuela y de la América Latina.

El cliente firma contratos con la empresa SIS Copextel perteneciente al Ministerio de la Informática y las Comunicaciones, planteando como problema la necesidad de digitalizar, organizar y facilitar la divulgación de la información del patrimonio cultural venezolano para todo su pueblo; y se pacta que se entregarían paulatinamente los Guiones Técnicos y de Contenido para cada uno de los 70 módulos del proyecto.

SIS Copextel no cuenta con el suficiente personal calificado, y por este motivo a inicios del año 2008 se acerca a la Facultad 8 de la UCI en busca de mano de obra especializada para desarrollar el producto, el cual sería una aplicación Web que utilizando hojas de estilo y código JavaScript muestre la información de los Contenidos Educativos. En la medida que el cliente entregue los guiones y sean revisados formalmente por SIS Copextel, serán entregados a la UCI para se realice el análisis de los mismos. La Universidad plantea a SIS que actualmente todos sus diseñadores estaban sobrecargados de trabajo, por lo que no puede asumir el diseño de las medias a utilizar en todos los módulos del proyecto; se acuerda que se trabajaría teniendo en cuenta el tiempo disponible por parte del equipo de diseñadores y en medida de que fuera posible; este acuerdo es aceptado por SIS.

Una vez firmados los contratos entre SIS Copextel y la UCI, y ya pactados todos los acuerdos referentes al desarrollo del producto, se inicia el proceso de elaboración del proyecto “Contenidos Educativos Digitales”.

2.4 Estructura y Composición

2.4.1 Introducción

En el presente sub-epígrafe se abordarán temas sobre la estructura y composición del proyecto “Contenidos Educativos Digitales”, tales como los diferentes roles desempeñados dentro del proyecto y sus respectivas responsabilidades, así como el soporte tecnológico con que se cuenta para el trabajo en el proyecto.

2.4.2 Roles y responsabilidades definidos

En el proyecto Contenidos Educativos Digitales se encontraban inicialmente definidos los roles y responsabilidades ([Ver Anexo 2](#)).

Las actividades asociadas al rol Asegurador de la Calidad definido dentro del proyecto son las siguientes:

- Planificar el proceso de Aseguramiento de la Calidad en el proyecto (Plan de Calidad).
- Realizar el Plan de Pruebas, de Revisión y Auditoría para cada iteración.

- Coordinar el proceso de recopilación, análisis y reporte de las estadísticas de calidad (Plan de mediciones).
- Dar seguimiento a estos planes.
- Guiar las revisiones técnicas formales.
- Guiar las pruebas que se realicen.
- Manejar todo lo relacionado con los riesgos de calidad (Lista de Riesgos).
- Realizar el Resumen de Evaluación.
- Identificar técnicas apropiadas, herramientas e instrucciones para la implementación de las pruebas.
- Diseñar los casos de pruebas.
- Definir listas de chequeo para las pruebas.

Se considera que el rol de Asegurador de la Calidad definido no satisface las necesidades del mismo, en cuanto al tema de Aseguramiento de la Calidad, ya que se le asocian demasiadas actividades. Para ello se realiza una propuesta de cómo deben quedar definidos los roles encargados de asegurar la Calidad dentro de Contenidos Educativos Digitales, esperando aumentar la organización y eficiencia del equipo de calidad.

2.4.3 Recursos humanos por roles

Los recursos humanos para cada uno de los roles, de acuerdo a las exigencias del proyecto y la cantidad de personal disponible, se encuentra confeccionada de la siguiente forma:

✓ **Roles de profesores:**

- Líder de Proyecto: 1 persona
- Arquitecto: 1 persona

- Asegurador de la Calidad: 1 persona
- Jefe de Equipo de Desarrollo: 1 persona
- Analista Principal: 1 persona
- Analista: 1 persona

✓ **Roles de estudiantes:**

- Analista: 4 personas.
- Asegurador de la Calidad: 9 personas.
- Gestión de Configuración: 1 persona.
- Programadores: 9 personas
- Arquitectos: 14 personas

A modo de resumen, Contenidos Educativos Digitales cuenta con un total de 43 integrantes en su composición, divididos en 37 estudiantes y 6 profesores.

2.4.4 Soporte Tecnológico

Contenidos Educativos Digitales cuenta con 18 estaciones de trabajo formadas por 18 computadoras, todas se encuentran actualmente en explotación y en perfectas condiciones técnicas. El proyecto cuenta además con un repositorio donde se guarda toda la información, el mismo se encuentra dentro del servidor de la facultad perteneciente al proyecto “Soporte Técnico”.

2.5 Situación Real

Se considera importante precisar que en el momento en que los autores comienzan a ejercer su rol dentro del proyecto, ya el mismo había comenzado y no se contaba con personal calificado para desempeñar el rol Asegurador de Calidad. A partir de este contexto se decide evaluar y analizar la situación existente en el proyecto, para a partir de estos resultados poder implementar la estrategia de aseguramiento de la calidad.

Al comenzar la investigación se pudo constatar que en los guiones enviados al proyecto por la empresa SIS Copextel aparecían múltiples errores e incongruencias, estas entregas no pudieron ser revisadas a tiempo, debido a la situación existente con el equipo de calidad. A partir de ese momento se presentan problemas con el entendimiento de los guiones por parte de los analistas, retrasando la actividad de documentarlos. A esto se le suma el hecho de que ningún analista poseía conocimientos en el uso del Lenguaje de Modelación Orientado a Objetos para Aplicaciones Educativas Multimedia (ApEM_L), usado para la documentación dentro del proyecto, por lo que el analista principal, única persona dentro del equipo con experiencias en la utilización de ApEM_L imparte un curso con el fin de capacitar a los demás

analistas. Posteriormente se realizan en ocasiones reuniones dentro del grupo de analistas para cambiar pautas en la documentación, hasta que en enero del 2009 se elabora “Políticas y Pautas de Documentación” ([Ver Anexo 3](#)) en su versión 1.0, artefacto que guía todas las normas a tener en cuenta a la hora de documentar los guiones, y de obligado uso por parte de todos los analistas.

Para obtener información actualizada sobre el estado actual del proyecto se entrevista a la líder del mismo la Ing. Mailyn Cabrera Torres el día 10 de marzo del 2009, como resultado de esta entrevista se obtiene lo siguiente:

De los 70 módulos con que cuenta el proyecto, la empresa SIS Copextel ha enviado a la UCI 31 guiones, de los cuales se han documentado 25 hasta el momento, o sea, se han llevado a un estado entendible por los arquitectos y diseñadores y se han definido los sistemas y subsistemas de cada contenido. Como resultado de la actividad de documentación se han generado los siguientes artefactos:

- 1) Documentación Ingenieril del Contenido Educativo General (CED).
- 2) Documentación para Diseño – CED

Actualmente el equipo de calidad se encuentra enfrascado en el proceso de revisión del artefacto 1, guiándose por las políticas y procedimientos acordados para la documentación. Este artefacto es de suma importancia pues es capaz de proporcionar datos para la implementación del producto. Asimismo el diseño de las medias a usar en los módulos está retrasado, el equipo de arquitectos está en espera de esta entrega, a partir de que se efectúe se podrá comenzar con la confección de la plantilla a usarse en las páginas web del producto y posteriormente al montaje de toda la información en las plantillas. En el momento de la entrevista los diseñadores se encuentran trabajando en la confección de las medias para 6 módulos del proyecto, sin tener aún fecha de entrega fijada.

También se puede constatar que existe un grado considerable de falta de conocimientos necesarios para desarrollar el producto en varios equipos de trabajo, como son los de analistas, implementadores, arquitectos, probadores y aseguradores de la calidad. Este es un problema que afecta considerablemente la culminación del producto de acuerdo al cronograma de trabajo.

2.6 Resultados Esperados

De forma concreta y precisa con el producto desarrollado en “Contenidos Educativos Digitales” se espera:

Capítulo 2. Análisis de la Situación Actual

- Divulgar la Historia de Venezuela como valor socio-cultural y político latinoamericano para reforzar el sentimiento de identidad y pertenencia.
- Exponer la relación socio-cultural y política existente entre los pueblos latinoamericanos desde sus orígenes para avanzar en el proceso de integración.
- Demostrar la importancia y eficacia de la tecnología aplicada al aprendizaje en el proceso de transformación educativa.
- Promover la iniciativa del Estado de rescatar y preservar la historia, la cultura y los ideales venezolanos y latinoamericanos ante el proceso de homogeneización cultural mundial.

Capítulo

3

Capítulo 3. Propuesta y Descripción de la Solución

3.1 Introducción

Una de las principales fases dentro de la elaboración de un proyecto es el Aseguramiento de la Calidad del Software (SQA) y su importancia se debe a que es un modelo sistemático y planeado de todas las acciones necesarias para proveer la confianza adecuada, según los requerimientos técnicos establecidos, de cada producto e ítem del proyecto.

En este capítulo se describe y se confecciona el plan de aseguramiento de la calidad para el proyecto “Contenidos Educativos Digitales” como parte de la estrategia, planteando las actividades a realizar por el equipo de calidad.

3.2 Características de la Estrategia

La estrategia de aseguramiento de la calidad del proyecto productivo “Contenidos Educativos Digitales” que se define en el presente capítulo comienza con una definición adecuada de los roles que conformarán el equipo de calidad del software; luego se confecciona el plan de aseguramiento de la calidad, artefacto que suministra una visualización de cómo debe ser seguro el producto y la calidad del proceso. El plan de aseguramiento de la calidad del software define las actividades de ingeniería de software específicas que se llevarán a cabo para asegurar la calidad del proyecto Contenidos Educativos Digitales.

Dentro de estas actividades se encuentran las revisiones donde se especifica el momento en el que se realizarán, así como una breve explicación de cómo fluirá el proceso de revisiones durante el proceso de desarrollo. Otras actividades contempladas dentro de las revisiones serán por ejemplo diseñar las listas de chequeo, bajo las cuales se inspeccionarán los artefactos, así como registrar todas las no conformidades encontradas ya que a partir de estas se elabora el registro de respuestas a las no conformidades para darle seguimiento a las mismas. Además se determinarán los tipos de pruebas que se le aplicarán a cada uno de los módulos terminados del producto, especificando la fase en que se comenzará el proceso de diseño de los casos de prueba la ejecución de los mismos. Ante todos los errores detectados, y en vistas a eliminarlos de raíz de forma que no se vuelva a incurrir en ellos, se realizan una serie de actividades correctivas cuyo objetivo principal es que los problemas descubiertos no se repitan. También se realizará una propuesta de cursos optativos con el fin de enriquecer los conocimientos del equipo de desarrollo. A raíz de la confección de la estrategia de aseguramiento de la calidad se crearán estos documentos utilizando las plantillas propuestas por la Dirección de Calidad de Software de la UCI, aunque en muchas de estas surgirán modificaciones dadas las características peculiares del proyecto “Contenidos Educativos Digitales”.

3.3 Definición del Equipo de Aseguramiento de la Calidad en el Proyecto Contenidos Educativos Digitales.

Se propone la existencia de un equipo de calidad definido dentro del proyecto, este es el encargado de realizar la estrategia de aseguramiento de la calidad interna del mismo, teniendo en cuenta las necesidades de aseguramiento de la calidad que se tiene. Este equipo será organizado por roles siguiendo la metodología RUP, al mismo, se le añadirá el rol de administrador de la calidad, además se definirán las actividades que van a desempeñar cada uno y la base de conocimientos de la que deben partir. Los roles y actividades del equipo de aseguramiento de la calidad definidos se describen como parte del Plan de Aseguramiento de la Calidad.

Una vez que se han precisado los roles que intervendrán en la estrategia de aseguramiento de la calidad del proyecto, es necesario definir un plan de entrenamiento para lograr nivelar los conocimientos básicos para realizar las actividades de aseguramiento de la calidad. La planificación del entrenamiento debe ser desarrollada por el Administrador de la Calidad teniendo en cuenta los conocimientos básicos requeridos

por cada rol para que puedan desempeñar sus tareas de la mejor manera posible y así asegurar durante el desarrollo del proyecto, la calidad del proceso y el producto.

3.4 Definición del Plan de Aseguramiento de la Calidad del Software

Se elaboró el Plan de Aseguramiento de la Calidad como actividad fundamental, para asegurar durante el desarrollo del proyecto, la calidad del proceso y el producto. En este se definen las principales actividades a llevar a cabo por el equipo de calidad y con la participación activa del líder del proyecto como responsable también en el proceso de Aseguramiento.

3.4.1 Introducción

El aseguramiento de la calidad del software es un patrón a seguir por todo el equipo de desarrollo. Contiene la planificación y sistematización de las acciones que van dirigidas a lograr la calidad final del producto. Con este se desea lograr una organización efectiva de todas las actividades que se realizarán durante el proceso de desarrollo del software de forma tal que se obtengan resultados satisfactorios. Además se trata de que la planificación sea lo más real posible y que se encuentre acorde con las características específicas del proyecto, teniendo siempre en cuenta el aprovechamiento medido de los recursos y esfuerzos destinados para su desarrollo.

3.4.1.2 Propósito

Describir cómo se asegurará la calidad del producto de forma general, los artefactos, herramientas y procesos a través de los cuales se pretende lograr dicho propósito.

3.4.1.3 Alcance

Este plan se realiza a partir de la utilización de la plantilla propuesta por la Dirección de Calidad de Software de la UCI, pero ha sufrido modificaciones dadas las características propias del proyecto “Contenidos Educativos Digitales”, así que solo puede ser aplicada en él, en todos los módulos con los que el mismo cuenta.

3.4.1.4 Definiciones, Acrónimos y Abreviaturas

- SQA: Aseguramiento de la Calidad del Software.
- DCS: Dirección de Calidad de Software.
- RUP: Proceso Unificado Racional.
- CMMI: Modelo Integrado de Capacidad y Madurez.
- ECS: Elementos de Configuración de Software.
- Aseguramiento de la calidad de software: Conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisficará los requisitos dados de calidad.

3.4.1.5 Referencias

Tabla 1 Referencias

Código	Título
1	Lineamientos de Calidad UCI
2	Plan de Gestión de Configuración de Software
3	Plan de Desarrollo de Software
4	Plan de Pruebas

3.4.1.6 Resumen

El Plan de Aseguramiento de la Calidad es el documento rector donde se responsabilizan las partes y se organiza el trabajo del Grupo de Calidad del proyecto para lograr un mejor nivel de productividad. Presenta los Objetivos de Calidad que se rigen en el proyecto, la estructura organizativa del mismo y una descripción de las pruebas, revisiones y auditorías que se le realizarán al producto y al proceso durante el ciclo de vida del software.

3.4.2 Objetivos de Calidad

Los objetivos planteados para el área de calidad son los siguientes:

- Asegurar la calidad del trabajo, a través de la vigilancia, prevención, comprobación y valoración sistemática en el proyecto, a lo largo del ciclo de vida del mismo, velando porque el producto software cumpla con los requerimientos establecidos por el cliente.
- Identificar posibles errores antes de que se conviertan en puntos fatales o problemáticos.
- Mantener el trabajo sobre la base de los diferentes estándares, modelos y normas internacionales existentes, así como las metodologías establecidas por el proyecto.
- Velar y asegurar el desarrollo e implantación de un sistema informático que permita la recolección de la información proveniente de los distintos entes involucrados.
- Corresponder y hacer cumplir con los lineamientos de calidad establecidos por Calisoft para los proyectos productivos de la Universidad de Ciencias Informáticas (UCI).
- Lograr que el personal de calidad esté capacitado con el conocimiento y las habilidades necesarias para realizar las tareas y actividades encaminadas a lograr la calidad del proyecto.

3.4.3 Gestión

3.4.3.1 Organización

El proyecto Contenidos Educativos digitales está compuesto por profesores y estudiantes, todos pertenecientes a la Facultad 8. La definición de los roles a desempeñar en el proyecto se ha realizado según las necesidades inmediatas. La siguiente tabla muestra una lista de los roles y responsabilidades que formarán parte del proceso de aseguramiento de la calidad del proyecto, dicho equipo fue redefinido con el fin de lograr una mejor organización, quedando de la siguiente forma:

Tabla 2 Roles y Responsabilidades

Rol	Descripción
Administrador de calidad	<ul style="list-style-type: none">• Asegura que la aplicación producida se ajusta a las especificaciones y está libre de errores.• Proporciona una metodología para realizar las pruebas, y el cronograma de pruebas.

Capítulo 3. Propuesta y Descripción de la Solución

	<ul style="list-style-type: none"> • Coordina las pruebas de calidad internas, las pruebas de aceptación del cliente y pilotos. • Evalúa los resultados que se obtienen en las pruebas de calidad. • Orienta y dirige las actividades de aseguramiento de la calidad.
Diseñador de pruebas	<ul style="list-style-type: none"> • Diseña los casos de prueba. • Evalúa y documenta el resultado de las pruebas realizadas al software. • Confecciona listas de chequeo. • Realiza el Plan de Pruebas.
Revisor técnico	<ul style="list-style-type: none"> • Chequea que los artefactos generados se ajusten a las pautas y lineamientos establecidos para su confección.
Probador	<ul style="list-style-type: none"> • Ejecuta las pruebas diseñadas. • Registra los resultados obtenidos.

3.4.3.2 Tareas y Responsabilidades

Tabla 3 Tareas y Responsabilidades

Tarea de Aseguramiento de calidad	Pre condición al finalizar la fase de:	Pos condición antes de la fase de:	Responsable	Comentarios
Establecimiento del Plan de Aseguramiento de la calidad			Administrador de la Calidad	
Mantener actualizados los documentos del proyecto (Expediente de proyecto)			Administrador de la Calidad	
Revisar el Producto			Revisor técnico, Diseñador de Pruebas, Probadores	
Revisar la documentación del Proyecto			Revisor técnico	

Capítulo3. Propuesta y Descripción de la Solución

Establecer una estrategia de prueba para el proyecto			Diseñador de Pruebas	
Definir el Plan de Pruebas			Diseñador de Pruebas	
Diseñar los casos de pruebas.			Diseñador de Pruebas	
Realizar pruebas internas al Proyecto.	Implementación		Probadores	
Comprobar correcto uso de estándares.	Todas	Todas	Diseñador de Pruebas, Probadores	
Recolección de Métricas.			Administrador de la Calidad	
Gestión y análisis de los registros de calidad			Administrador de la Calidad y Líder de Proyecto	
Certificación final del producto.	Prueba	Implementación	Dirección de Calidad de Software	

3.4.4 Documentación

Los siguientes documentos fueron consultados para la elaboración de este plan, todos se encuentran en el repositorio del proyecto “Contenidos Educativos Digitales”:

- Documento Glosario de Términos.
- Documento Roles y Responsabilidades.
- Registro de No Conformidades.
- Lineamientos de Calidad UCI.
- Plan de Desarrollo de Software.
- Estándares y normas de calidad.

- Plan de Pruebas.
- Plan de Gestión de Configuración de Software.

3.4.5 Métricas

Las métricas mejoran la calidad del producto final y del trabajo realizado, evalúan la efectividad del proceso y además permiten descubrir y corregir problemas potenciales en etapas tempranas del software. Aportan argumentos en la toma de medidas preventivas, necesarias para mejorar el proceso de revisiones. A continuación se describen una serie de métricas usadas para el monitoreo del producto.

Para definir las métricas que se ajustan al proyecto Contenidos Educativos, se tomó como base la propuesta de métricas para software educativo de la Máster en Ciencias Ma. Antonieta Abud Figueroa, a partir del estándar de calidad de software ISO-9126, donde se incorporan aspectos orientados a evaluar las características propias de los productos de software educativos.

Los aspectos educativos y de software en general que se deben medir en el desarrollo del software educativo son:

- **Pedagógicos** (orientados a evaluar las características referentes a la enseñanza-aprendizaje).
- **Contenido** (profundizan en los aspectos relacionados con la información que se presenta a través del software)
- **Técnicos** (verifica las características del programa en cuanto a requerimientos para su funcionamiento (calidad de los entornos, facilidad de uso, confiabilidad, usabilidad, documentación, etc.))
- **Interfaz Humano-Computadora** (contemplan los puntos o características a evaluar en cuanto a la presentación del material)

Junto a estos aspectos se consideraron los factores que sugiere el estándar ISO-9126, de los cuales se tomaron aquellas características que se acoplan al software educativo, entre los que se encuentran:

- **Funcionalidad** (intenta evaluar si el software satisface las necesidades para las que fue creado, considerándose para el software educativo conveniencia, exactitud, seguridad y conformidad)

Capítulo 3. Propuesta y Descripción de la Solución

- **Confiabilidad** (la capacidad del software para mantener su nivel de funcionamiento bajo ciertas condiciones indicadas por un período de tiempo, para lo que se considera: exactitud, recuperación y tolerancia a fallas)
- **Facilidad de uso** (conjunto de atributos que se refieren al esfuerzo necesitado para el uso del software, tomándose para el software educativo: comprensibilidad, facilidad de aprendizaje, operatividad, atractivo, conformidad y novedad)
- **Eficiencia** (conjunto de atributos que se refieren al lazo entre el nivel de funcionamiento del software y la cantidad de recursos usados, aquí se tienen: tiempo de respuesta, utilización de recursos y conformidad)
- **Facilidad de mantenimiento** (conjunto de atributos que se refieren al esfuerzo necesitado para hacer modificaciones especificadas, en las que se ha tomado: facilidad de análisis, cambiabilidad, claridad, modularidad y conformidad)
- **Transportabilidad** (es el conjunto de atributos que se refieren a la capacidad del software de ser transferido a partir de un ambiente, para lo cual se establece: adaptabilidad, facilidad de instalación y conformidad)

Existen diferentes métricas que se pueden usar para evaluar el software educativo ([Ver Anexo 4](#)), pero según las características del proyecto Contenidos Educativos Digitales se escogieron las siguientes:

- **Interfaz H-C 36%**
 - Facilidad de uso
 - Son atractivos los elementos del software
 - Conformidad con el diseño del menú
 - Novedad en la interfaz
 - Funcionalidad
 - Imágenes claras y comprensibles
 - Música apropiada al tema
 - Videos adecuados al tema
 - Colores adecuados
 - Manejo adecuado de las características del texto

Capítulo 3. Propuesta y Descripción de la Solución

- El puntero del mouse maneja diferentes formas de acuerdo a las acciones a ejecutar
- Acceso directo a temas
- Acceso al tema anterior
- Organización adecuada de las pantallas

➤ Técnico 12%

- Facilidad de uso
 - Las funciones de cada módulo del software se comprenden fácilmente
 - Facilidad de aprendizaje del software
 - Hay ayuda disponible desde cualquier punto del software
- Funcionalidad
 - La ejecución del software se realiza de manera fácil
 - Se cuenta con un manual sencillo y completo
 - Se cuenta con un manual que indica la forma de instalación del software
- Eficiencia
 - La respuesta del software a cualquier acción del usuario es adecuada
 - El tiempo de respuesta del software es adecuado
- Confiabilidad
 - Facilidad de recuperación cuando ocurren errores
 - Permite grabar la sesión en forma automática
- Transportabilidad
 - Facilidad de instalación

La métrica Interfaz Humano-Computadora constituye en un proyecto de software educativo, el 36% de la calidad, el 40 % es el aspecto Pedagógico, que no se medirá en el proyecto pues no se tiene relación directa con el cliente, el otro por ciento es según los aspectos Técnicos y el de Contenidos los cuales aportan cada uno el 12% de la calidad del software, este último no se medirá pues no es relevante para el proyecto.

Para propósitos de evaluación de cada una de las características identificadas y debido a su naturaleza, se determinó manejar dos tipos de criterios: los binarios, en donde sólo se evalúa la presencia o ausencia

Capítulo 3. Propuesta y Descripción de la Solución

del atributo; y los de multinivel, donde es necesario establecer un rango de valores, que para nuestro caso se determinaron como se muestra en la siguiente tabla:

Tabla 4 Niveles de calidad para el software educativo

Nivel de Calidad	Puntaje	Equivalencia
0	$u_k=0$	Ausencia del atributo o mala calidad
1	$u_k=70$	Calidad regular
2	$u_k=90$	Calidad aceptable
3	$u_k=100$	Calidad excelente

Para calcular la calidad global del producto se utilizó un modelo de atributos múltiples, usando la función:

$$U = \left[\sum_{k=1}^n w_k u_k \right] / 100$$

Donde

U es el valor de calidad global,

w_k es el peso para el factor de calidad

u_k es el puntaje obtenido para la alternativa k .

De modo que

$$\left[\sum_{k=1}^n w_k = 100 \right]$$

Una vez establecidos los aspectos y factores relevantes para la evaluación del *software* educativo, se estableció una escala de aceptabilidad, basada también en los rangos de satisfacción que establece el estándar ISO-9126, tal como se muestra en la tabla:

Tabla 5 Nivel de aceptabilidad de los valores de preferencia de los atributos

Nivel de Calidad	Categoría	Puntaje
0	Mal	0
1	Regular	50
2	Bien	70
3	Excelente	90

Así para cada factor se deberá establecer una calificación, 0 ó 100 para criterios binarios, o bien 0, 70, 90 ó 100 para los casos de mala, regular, aceptable y excelente calidad respectivamente en los criterios multinivel. Cada valor (u_k) se multiplica por el peso de ese atributo (w_k) obteniendo una calificación por cada factor. Finalmente, se suman los resultados obteniéndose un valor que al dividirlo entre 100 dará el nivel de calidad global del producto, el cual puede consultarse en la tabla anterior.

A continuación se describen otras métricas usadas para analizar la efectividad de las revisiones realizadas.

1. **Efectividad de eliminar los defectos en una Revisión:**

$$EED = \frac{DE_i}{DL} * 100$$

$$DL = DE_i + DEP$$

$$DEP = \sum_{k=i+1}^n DE_k$$

DE: cantidad de defectos detectados durante la revisión.

DEP: cantidad de defectos encontrados posterior a la revisión, es decir, la cantidad de defectos encontrados en las n-i restantes revisiones que se indican en el registro de revisiones y auditorías del proyecto. También puede calcularse este valor considerando los defectos detectados en las revisiones

efectuadas hasta el momento ($k = m$, con $i < m < n$) en que se desea analizar la métrica, pero serán resultados parciales que pueden cambiar al finalizar el producto.

DL: cantidad total de defectos presentes en el producto, cuando éste ha sido terminado y se entrega al cliente para su operación.

Si se representan en un gráfico los valores resultantes de la métrica de efectividad se puede analizar qué revisiones de las realizadas al proyecto de software resultan poco efectivas y así valorar la posibilidad de incluirla o no en la misma fase del Desarrollo del Proyecto o en otros cuyas características sean similares al analizado o tomar cualquier otra decisión que contribuya al mejoramiento del Proceso de Revisiones.

Ahora bien, esta medida da una vista global de la efectividad de la revisión, pero en ocasiones no basta con esta información y es necesario profundizar para conocer cuáles tipos de errores no han sido detectados y que por tanto conspiran contra la efectividad de dicha revisión, para ello se propone la métrica siguiente.

2. Efectividad de eliminar los defectos de la fase j en la revisión i.

Permite a los directivos conocer la efectividad de las revisiones en cuanto a la cantidad de defectos que pertenecen a una fase y que son encontrados oportunamente. Por tanto, se puede analizar la cantidad de defectos, por ejemplo de la fase de requisitos, que se han propagado hasta la implantación del sistema o hasta cualquier otra fase de desarrollo del proyecto.

$$EED_{i,j} = \frac{DE_{i,j}}{DL_j} * 100$$

DE_{i,j}: cantidad de defectos detectados durante la revisión i, correspondientes a la fase j.

DL_j: cantidad total de defectos presentes en el producto correspondientes a la fase j.

El comportamiento de estas dos primeras métricas puede ser representado en un único modelo que de una idea global de la eficiencia de las revisiones y de cada una de las fases consideradas en ellas. Por ejemplo, se pudiera pilotear los valores de la efectividad de diferentes revisiones realizadas a un proyecto y compararlas entre sí.

3. Eficiencia en la eliminación de No Conformidades:

Esta métrica se deriva de la métrica primitiva No conformidades y se calcula:

$$EENC = \frac{NC_i}{NC_i + NC_e}$$

EENC: Eficiencia en la eliminación de no conformidades.

NC_i: Las no conformidades detectadas durante las actividades de validación y verificación del proyecto.

NC_e: No conformidades detectadas durante las pruebas de liberación, aceptación y revisiones de la DCS.

Los posibles valores de estas métricas oscilan entre [1.0]. Donde los valores cercanos a 1 reflejan una mejor eficiencia en la eliminación de no conformidades.

3.4.6 Estándares y Guías

Tabla 6 Estándares

Estándar	Ubicación	Comentarios
Lineamientos de Calidad de la UCI.	Repositorio del proyecto	Aplicar durante todo el proceso de desarrollo.
CMMI	Repositorio del proyecto	Para el aseguramiento de la calidad, y gestión de configuración.
ISO-9126	Repositorio del proyecto	Permite especificar y evaluar la calidad del producto de software desde perspectivas diferentes de aquellos asociados con la adquisición, regulación, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría del software basado en los atributos internos del software y el comportamiento externo del sistema. Estos tipos de métricas son aplicables al especificar requisitos de calidad y objetivos del diseño para los productos del software, incluso los productos intermedios.

3.4.7 Revisiones y Auditorías

3.4.7.1 Tareas generales de Revisiones y Auditorías

Se requiere como mínimo que se realicen revisiones al terminar cada etapa de desarrollo. Se debe aclarar que las revisiones se realizarán sobre artefactos terminados. Entre estos artefactos se encuentran documentos como: el Plan de Gestión de Configuración de Software, Plan de Aseguramiento de la Calidad, Plan de Desarrollo del Software, Documentación Ingenieril del CED, así como toda la documentación que se generará por fase, los diagramas, el código fuente y ejecutables. Las listas de chequeo que se utilizan para revisar son confeccionadas por el Diseñador de Pruebas, y las revisiones son llevadas a cabo por el Revisor Técnico.

3.4.7.2 Revisiones en la fase de Inicio

En esta fase se define el objetivo del proyecto, los recursos con los que se cuenta para el desarrollo del mismo, los riesgos fundamentales que afectarían su ejecución y el ambiente de desarrollo en general. Al final de la misma debe haberse desarrollado los siguientes planes: Plan de Desarrollo de Software, Plan de Gestión de Configuración de Software, Plan de Aseguramiento de la Calidad de Software; este último es producido por el personal de SQA y los demás también son revisados por los mismos, para comprobar que están de acuerdo a los estándares.

Revisión del Plan de Gestión de Configuración de Software:

En este documento se definen los procedimientos de control del proyecto antes de que comience la producción. El personal de SQA necesita chequear que los mismos están propiamente definidos y llevados a cabo, o sea, verificar que el desarrollo del software es correctamente controlado. Se debe verificar que el plan esté completo, que existan procedimientos de control de cambios, procedimientos de almacenamiento, mecanismos de seguimiento de los cambios, etc.

Varios son los documentos que son las primeras salidas de esta fase y estos deben ser lo suficientemente flexibles para ser usados en todo el proyecto. Los aspectos claves que deben ser dirigidos son:

- Identificación del Documento.

Capítulo 3. Propuesta y Descripción de la Solución

- Almacenamiento del Documento.
- Documento de Control de Cambios.

Toda la documentación relevante del proyecto debe ser identificada de manera única, ya sea por el nombre de la compañía, el nombre del proyecto, el tipo de documento, el nombre del documento y el número de versión del mismo. Cada página del documento debe ser marcada con un identificador. El personal de SQA debe verificar que todos los documentos del proyecto hayan sido correctamente redactados.

Durante la vida del proyecto se generarán múltiples documentos, por lo que SQA debe chequear mediante auditorías físicas que una copia de cada uno de ellos debe estar almacenada en una biblioteca de archivos.

La revisión detallada de este plan será llevada a cabo a partir de la Lista de Chequeo del Plan de Gestión de Configuración de Software ([Ver Anexo 5](#)), al aplicar la lista de chequeo los errores detectados se recogerán en un Registro de No Conformidades ([Ver Anexo 6](#)), elaborándose después un registro de respuesta a estas no conformidades encontradas.

Un diagrama de actividades que describe detalladamente este proceso de revisión está reflejado en los anexos ([Ver Anexo 9](#)).

Revisión del Plan de Desarrollo de Software:

Cada proyecto debe hacer un Plan de Desarrollo de Software, por lo que el personal de SQA debe cerciorarse que éste contenga bien declarado los objetivos del proyecto y chequear que todos los planes son consistentes con dichos objetivos. El personal de SQA verificará que se define el enfoque del ciclo de vida del proyecto apropiado y que ha sido correctamente ajustado al mismo. Se debe chequear también que el plan sea suficiente para minimizar los riesgos y que sobre todo sea un plan realista. Todos los planes tienen suposiciones viables, dependencias a eventos externos y están sujetos a limitaciones, de ahí que sea necesario que lo anteriormente expuesto esté bien direccionado en el análisis de los riesgos. Por último, las métricas son importantes para la evaluación del proyecto, por lo que se verificará que cada una sea apropiada para el mismo.

La revisión detallada de este plan será llevada a cabo a partir de la Lista de Chequeo para el Plan de Desarrollo de Software ([Ver Anexo 7](#)) y los errores detectados serán recogidos en un registro de no conformidades, elaborándose luego un registro de respuestas a las no conformidades encontradas.

Un diagrama de actividades que describe detalladamente este proceso de revisión está reflejado en los anexos ([Ver Anexo 10](#)).

3.4.7.3 Revisiones en la fase de Elaboración

La primera iteración correspondiente a la fase de elaboración tiene como objetivo modelar utilizando ApEM-L, los contenidos pertenecientes a la primera entrega realizada por SIS-Copextel al proyecto.

Revisión del artefacto “Documentación Ingenieril del CED”:

Dada la gran importancia del artefacto “Documentación Ingenieril del CED”, se ha hecho énfasis en la revisión del mismo, con el objetivo de lograr que tenga la calidad requerida, y sirva como entregable para SIS-Copextel y como punto de partida para la implementación. Para una óptima revisión se hace necesario especificar las condiciones de hardware y configuración bajo las cuales se deben realizar las pruebas. Dentro de la segunda iteración de la fase de elaboración se llevará a cabo una revisión de este artefacto, resultante de la modelación usando ApEM-L de los primeros guiones entregados. Para lograr que los documentos tengan calidad, los probadores serán los encargados de revisar la Documentación según las pautas definidas por el proyecto ([Ver Anexo 3](#)) y una lista de chequeo ([Ver Anexo 8](#)) que corresponde con los aspectos necesarios para medir la organización, estructura y contenido del mismo.

El proceso que se sigue, se basa en la revisión de los documentos mediante lo expuesto anteriormente. ¿Cómo se hace?, se llena la planilla de no conformidades con los errores encontrados según las pautas y la lista de chequeo, se les comunica a los analistas y estos una vez que aceptan o niegan las no conformidades encontradas, suben el documento refinado al repositorio y se informa nuevamente a los probadores, estos se encargan de revisar si se solucionaron debidamente los errores o si es necesario modificar algún aspecto, como se puede apreciar todo se basa en una interacción analistas-equipo de calidad, equipo de calidad-analistas.

Para la ejecución de la revisión de la primera entrega de la documentación de los guiones, están involucrados varios roles, primero el diseñador, que confecciona la lista de chequeo, y evalúa el cumplimiento y la calidad requerida del trabajo, el revisor técnico que revisa si los artefactos generados se ajustan a las pautas establecidas, el probador que ejecuta la prueba, y el administrador como orientador y máximo responsable de la calidad del producto final.

En fases posteriores hay que realizar casos de prueba en vista a futuras pruebas del producto una vez implementado y otros aspectos que son necesarios para medir la calidad.

Un diagrama de actividades que describe detalladamente este proceso de revisión está reflejado en los anexos ([Ver Anexo 11](#)).

3.4.7.4 Auditorías

Se realizan auditorías para la Gestión de la Configuración. La auditoría de configuración de software tiene un carácter complementario y se preocupa de si:

- Se ha hecho el cambio especificado en la UCI.
- Se han incorporado modificaciones adicionales.
- Se ha llevado a cabo o no una RTF.

Se realizan todas las auditorías que sean necesarias:

- Auditorías en proceso: son aleatorias y se envía un aviso con anticipación. Su propósito será revisar el trabajo actual que se realiza en el proyecto.
- Auditorías funcionales: se realizarán para verificar que el producto que se entrega satisface los requerimientos especificados.
- Auditorías físicas: se realizarán para verificar que realmente se entreguen el software físico y su documentación designados para ser entregados.

Se deberá realizar un plan de auditoría a la gestión de configuración de cada módulo, para comprobar que se hayan realizado todos los cambios que fueron planificados y su correcta documentación.

3.4.7.5 Cronograma

El cronograma de las tareas y actividades de calidad se define partiendo de la organización, definición de tareas y responsabilidades y contando con los Planes de Iteración y Desarrollo del equipo de software. A continuación se muestran algunas tareas que se están realizando y otras que próximamente comenzarán.

Capítulo 3. Propuesta y Descripción de la Solución

Tabla 7 Cronograma

Revisión	Inicio	Fin
1. Revisar la documentación de la primera entrega de los guiones.	20/03/09	22/04/09
1.1 Chequeo de cumplimiento.	Todos los lunes y viernes	--
2. Revisar la documentación de la segunda entrega de los guiones	22/04/09	15/05/09

3.4.7.6 Organización y responsabilidades

El proyecto cuenta con 10 integrantes que trabajan en el proceso de calidad, cada uno desempeña una tarea:

Tabla 8 Organización y Responsabilidades

Actividad	Responsable
Establecer la estrategia de prueba	Isyed de la C Rodríguez Trujillo
Verificar el cumplimiento de la revisión de los documentos ingenieriles de la primera entrega según las pautas definidas.	Isyed de la C Rodríguez Trujillo
Revisar que se está cumpliendo con la respuesta a las no conformidades una vez detectadas.	Isyed de la C Rodríguez Trujillo
Realizar las listas de chequeo.	Maidelis Aayala Girón Israel Acosta Ercia

Capítulo 3. Propuesta y Descripción de la Solución

Chequear que los artefactos generados se ajusten a las pautas y lineamientos establecidos para su confección, utilizando la lista de chequeo.	Patricia Caulineau Castro Maidelis Ayala Girón Indira Gracia Infante Mónica M. Manchón Ortega Israel Acosta Ercia Maylin Acosta Sánchez Indira Gracia Infante
Chequear el cumplimiento de la revisión de documentos.	Noidis Barroso Hidalgo Yudaily Gómez Romero

3.4.7.7 Resolución de problemas y actividades de corrección

Los errores o incongruencias que se detecten durante las revisiones se reportarán en un documento titulado Registro de No Conformidades habilitado para el caso ([ver Anexo 6](#)). Las no conformidades emitidas son problemas identificados en las evaluaciones que reflejan una carencia de la adherencia a los estándares indicados, a las descripciones de proceso o a los procedimientos que se habían indicado como normas a seguir en los procesos y productos del trabajo. El estado de las no conformidades emitidas proporciona una indicación de las tendencias de la calidad en el proyecto. El proceso para informar y manejar problemas en ellas sería el siguiente:

- Una vez terminada la reunión de revisión, el revisor debe emitir un documento de no conformidades, de existir alguna.
- Este documento debe ser enviado al jefe del grupo de trabajo o líder del proyecto en cuestión.
- El jefe del grupo de trabajo y los desarrolladores del proyecto u otros miembros implicados procederán a corregir cada una de las no conformidades encontradas.
- Una vez resueltas, el jefe del grupo de trabajo debe emitir un documento de respuesta a no conformidades para el líder del proyecto quien decidirá si es necesaria una nueva revisión.
- Los documentos de no conformidades y respuesta a las no conformidades deben almacenarse como registros de calidad en el servidor del proyecto.

Capítulo 3. Propuesta y Descripción de la Solución

Asumiendo que las auditorías son realizadas por la DCS (Dirección de Calidad de Software) el procedimiento para informar y manejar los problemas en las mismas será el propuesto por la DCS.

3.4.8 Pruebas

Las pruebas posibilitan no solo la detección de errores de la aplicación, verificando el cumplimiento de los requisitos funcionales y no funcionales, errores ortográficos, abreviaturas, la funcionalidad de vínculos, botones, la ayuda, sino también verifican la correspondencia de la documentación con la aplicación, dígame especificaciones de casos de uso del sistema, manuales de usuario y de instalación. Se verifica además, la lógica interna del programa, pues un buen resultado de la implementación no solo depende de la codificación sino también de las pruebas. La realización de un proceso adecuado facilita que no aumenten los costos y tiempo del proyecto, que disminuyan los riesgos asociados y además que se eleve el nivel de calidad del producto, de esta forma, una vez que el mismo llegue a manos del cliente, va a tener la menor cantidad de errores dando como resultado la satisfacción del usuario final.

Para las pruebas se realiza un Plan de Pruebas, en este documento se describe de forma detallada cómo el Equipo de Calidad realizará todas sus pruebas. Su planificación, flujos de trabajo, herramientas y funcionamiento de los roles, deben quedar muy bien definidos, se hace una reseña además de los posibles casos de pruebas y los métodos que se utilizarán para ello, así como también del proceso de ejecución y evaluación de las pruebas. Se detalla una descripción de los requerimientos, se indica brevemente en qué consiste cada caso de uso, se plantea la matriz de escenarios correspondiente a los mismos y se plantea la planilla de Condiciones, esto es las condiciones que causan que se ejecute un escenario específico dentro de los posibles escenarios identificados para los casos de uso y se realizan casos de pruebas para los mismos. Es elaborado a partir del propuesto por RUP, sólo que adaptado a las condiciones específicas del proyecto de forma general. Por tanto, su uso es muy particular del proyecto.

La plantilla para el Plan de Pruebas no se corresponde con las características del proyecto Contenidos Educativos Digitales, por tanto, será adaptado a las condiciones requeridas. En el proyecto Contenidos Educativos Digitales, no se evalúan los requerimientos, ni se tienen en cuenta en las pruebas, debido a que la gestión de requisitos no la realizan integrantes del proyecto. El proyecto constituye la mano de obra especializada para lograr el producto final, donde todo el proceso de negocio, al igual que el despliegue del producto es realizado por SIS-Copextel, por lo que solo se trabaja en los flujos de Análisis y Diseño, Implementación y Pruebas.

Capítulo 3. Propuesta y Descripción de la Solución

Por tanto del Plan de Pruebas no se tendrá en cuenta lo referente a la descripción de los Casos de Uso debido a que dicho proyecto no está dividido por casos de uso, sino por vistas, en el mismo se realizará una breve descripción de las vistas más significativas y se trabajará con respecto a las mismas.

Las etapas definidas en el proceso de pruebas serán las especificadas por RUP en el ciclo de desarrollo: Inicio, Elaboración y Construcción ya que la última etapa no se tendrá en cuenta por lo antes expuesto. A continuación se especifica que se deberá hacer en cada etapa de desarrollo.

En la primera etapa de desarrollo del mismo (**Inicio**) no se realizan pruebas.

En la etapa de **Elaboración** se especifican las condiciones de hardware y configuración bajo las cuales se deben realizar las pruebas y comienzan a diseñarse los casos de pruebas.

En la etapa de **Construcción** se realizarán pruebas unitarias para verificar todos los flujos de control y de sistema para comprobar que esté aceptable el producto y que cumple con las funcionalidades principales. Todas las pruebas se realizarán de forma manual a medida que se vayan implementando los distintos módulos.

A medida que se vaya terminando la implementación de cada módulo los diseñadores de casos de pruebas se prepararán para diseñar los casos de pruebas necesarios y alguna lista de chequeo en caso necesario, con el objetivo de probar cada uno de estos por separado para comprobar sus funcionalidades. En un principio los diseñadores de casos de pruebas revisan la documentación necesaria para la realización de los casos de pruebas entregados por los desarrolladores. Con esta información en sus manos definen el objetivo de las pruebas, los métodos a utilizar para el diseño y se elaboran las listas de chequeo de comprobación. Al concretar todo lo anterior el equipo de diseñadores se encuentra listo para comenzar a diseñar las pruebas registrándolas en el documento de Diseño de Casos de Pruebas definido por la Dirección de Calidad del Software de la UCI.

Una vez diseñado los casos de pruebas correspondientes se comenzarán a ejecutar las pruebas cuando se encuentren preparados los recursos tanto humanos como materiales y el producto que será objeto de las pruebas. Si alguno de estos elementos no está listo y ha llegado la fecha planificada para la realización de las pruebas se plasma en el documento de no conformidades, explicando las causas y se ponen de acuerdo con los desarrolladores sobre cuándo el producto estará listo, planificando una nueva fecha. Si el producto está listo para realizarle la prueba el administrador de calidad es el responsable de asignar las tareas a los probadores y se comienza a ejecutar.

Capítulo 3. Propuesta y Descripción de la Solución

La jornada de prueba consta de 3 partes, la sesión de la mañana trabajando en el horario de 8:30a.m. a 12:00 m, la sesión tarde de 2:00 p.m. a 6:00 p.m. y por la noche de 8:00 p.m. a 12:00 p.m. Con esta planificación aseguramos cumplir las pruebas en el tiempo requerido. Las pruebas se comenzarán a aplicar a partir del 12 de mayo del 2009.

Conjuntamente a la ejecución de las pruebas se irán comparando los resultados que se obtienen con los previstos en el diseño de las pruebas, si se revelan diferencias estas se registran en el documento de No Conformidades, las cuales deben ser entregadas a los desarrolladores lo más rápido posible para que las analicen y corrijan, y a su vez se realiza el registro de respuesta a los no conformidades detectadas. Al concluir cada iteración se deben valorar los resultados alcanzados con las pruebas, analizar la tendencia de las no conformidades detectadas y la rapidez en darle solución, con la finalidad de adquirir experiencia para otras iteraciones de las pruebas.

Para la culminación de las pruebas se analizará como criterio de parada: el cumplimiento con el cronograma de trabajo. Luego de la valoración y el análisis correspondiente que se realiza al final de cada iteración, se debe decidir si se ejecutarán nuevamente las pruebas en la siguiente, esto se hace considerando si es posible efectuarlas sin incurrir en el cumplimiento de los plazos de entrega del módulo del proyecto que se está probando.

Un diagrama de actividades que describe detalladamente este proceso de pruebas está reflejado en los anexos ([Ver Anexo 12](#)).

3.4.9 Herramientas, Técnicas y Metodologías utilizadas en las actividades del Plan.

Las herramientas, técnicas y metodologías que apoyan el plan de aseguramiento de la calidad se nombran a continuación:

- Para la realización de las revisiones a los diferentes productos de trabajo se utilizarán varias técnicas de revisiones descritas anteriormente como son las RTF, Reuniones Informales, Presentación formal.
- Aplicación de los LC de la Universidad.
- Listas de chequeo para el desarrollo de las revisiones.
- Se utilizará la metodología Rational Unified Process (RUP) como apoyo para el establecimiento de la estrategia de pruebas, el plan de pruebas, el diseño de los casos de pruebas.

- Plantillas del expediente de proyecto establecido por la DCS para registrar la documentación del proyecto.
- Libro oficial de CMMI.
- Regulaciones y normas establecidas por la dirección de informatización, dirección de diseño y la dirección técnica de la IP.
- Documento Plan de Desarrollo de Software.

3.4.10 Gestión de Configuración

La Gestión de Configuración del Software es una disciplina encargada del control de la evolución de los productos de software.

Gestión de Configuración es *“el proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.”*[20]

El propósito de la Gestión de Configuración del Software es establecer y mantener la integridad de los productos de software a través del ciclo de vida del proceso de software.

El objetivo del SQA en esta área es asegurar que se realizan las actividades de gestión de configuración establecidas en el Plan de GCS y que se realizan tal como están establecidas en el proceso. Incluye identificar los métodos para la gestión de la configuración. Estos métodos se utilizarán para mantener, almacenar, asegurar y documentar las versiones controladas identificadas en las fases de desarrollo, también se utilizarán para proteger el almacenamiento adecuado de los programas, documentación, etc., así como también la prevención de acceso sin autorización, daño, etc., lo cual será definido en conjunto con el responsable de gestión de la configuración del software. En el desarrollo de software los cambios, debidos principalmente a modificaciones de requisitos y fallos, son inevitables.

- Se debe llevar un control y registro de los cambios con el fin de reducir errores.
- Evitar los problemas que puede acarrear una incorrecta sincronización en dichos cambios, al afectar a otros elementos del sistema o a las tareas realizadas por otros miembros del equipo de proyecto.

Capítulo 3. Propuesta y Descripción de la Solución

- La gestión de configuración se debe realizar durante todas las actividades asociadas al desarrollo del sistema, y continúa registrando los cambios hasta que éste deja de utilizarse.
- Se debe facilitar el mantenimiento del sistema, aportando información precisa para valorar el impacto de los cambios solicitados y reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo.
- Deberá permitir el control del sistema como producto global a lo largo de su desarrollo.
- Se deberán obtener informes sobre el estado de desarrollo en que se encuentra y reducir el número de errores de adaptación del sistema, lo que se traduce en un aumento de calidad del producto, de la satisfacción del cliente y, en consecuencia, de mejora de la organización.
- Se deberá mantener la integridad de los productos que se obtienen a lo largo del desarrollo de los sistemas de información, garantizando que no se realizan cambios incontrolados y que todos los participantes en el desarrollo del sistema disponen de la versión adecuada de los productos que manejan.
- Los elementos de configuración software como: ejecutables y código fuente, modelos de datos, modelos de procesos, especificaciones de requisitos y pruebas, deberán ser controlados.

La gestión de Configuración del Software será llevada a cabo por un equipo de trabajo conformado por: Líder de Proyecto, Administrador de Calidad y Jefe de Gestión de la Configuración; se hará el uso de herramientas de control como el Subversión y Trac como herramienta de planificación.

3.4.11 Registros de Calidad

Los registros de calidad son documentos que guardan información específica y relacionada a un procedimiento o instrucción de trabajo. Estos comprueban que el proyecto cumple sus procedimientos y normas establecidas.

El equipo de calidad tiene establecido como política a seguir que toda la documentación de calidad que se genere sea conservada durante todo el ciclo de vida del proyecto e incluso después de concluido, pues constituye una evidencia del trabajo realizado y una fuente de consulta muy valiosa que se puede utilizar en otros proyectos productivos de la Facultad o Universidad.

Los tipos de registros que se guardarán serán:

Capítulo 3. Propuesta y Descripción de la Solución

Tabla 9 Registros de Calidad

Registro	Ubicación	Tiempo de Almacenamiento
Actas de reuniones efectuadas	Carpetas de Calidad	Durante todo el proyecto
Listas de Chequeo aplicadas	Repositorio del proyecto	Durante todo el proyecto
Casos de Prueba	Repositorio del proyecto	Durante todo el proyecto
Informes de No Conformidades	Repositorio del proyecto	Durante todo el proyecto
Informes de respuesta a No Conformidades	Repositorio del proyecto	Durante todo el proyecto
Correos de trabajo y comunicación de intercambio de trabajo	Carpetas de Calidad	Durante todo el proyecto
Informes de revisiones	Repositorio del proyecto	Durante todo el proyecto
Informes de auditorías	Repositorio del proyecto	Durante todo el proyecto
Informes de estado de la configuración	Repositorio del proyecto	Durante todo el proyecto
Cronogramas	Repositorio del proyecto	Durante todo el proyecto
Resúmenes de Resultados	Repositorio del proyecto	Durante todo el proyecto

3.4.12 Entrenamiento

Los equipos de analistas y arquitectos deben poseer conocimientos en el lenguaje de modelado ApEM_L, el cual será el usado para modelar los guiones de los contenidos educativos del proyecto. A su vez, este último equipo, junto a los de montadores e implementadores, deben conocer el uso del lenguaje HTML

Capítulo 3. Propuesta y Descripción de la Solución

para realizar las páginas web, así como CSS y Javascript, ya que algunas funcionalidades pueden implementarse haciendo uso de las potencialidades que brindan los mismos. Por su parte el equipo de calidad también necesita prepararse en vistas a una mejor realización de sus actividades dentro del proyecto. A su vez todo el equipo de desarrollo debe ser capaz de gestionar archivos en el repositorio del proyecto, acción de suma importancia para mantener actualizada y segura toda la información relacionada con el producto.

Debido a la falta de conocimientos de múltiples integrantes del proyecto “Contenidos Educativos Digitales” en estas técnicas necesarias para desarrollar el producto, el entrenamiento es una actividad de suma importancia dentro de la estrategia de aseguramiento de la calidad planteada para lograr un producto con calidad y cumplir cabalmente con el cronograma de trabajo. Basado en estas dificultades detectadas se realiza una propuesta sobre los cursos optativos a vencer por los diferentes equipos de trabajo, y que en gran medida ayudarán a revertir la situación real.

Tabla 10 Propuesta de Cursos Optativos

Equipo	Cursos Optativos Propuestos
Calidad	<ul style="list-style-type: none">• Introducción a la calidad• Aseguramiento de la Calidad• Tortoise• Evaluación y Prueba de software• CMMI
Analistas	<ul style="list-style-type: none">• ApEM_L• Tortoise
Arquitectos	<ul style="list-style-type: none">• ApEM_L• Tortoise• HTML• CSS• Tortoise
Implementadores	<ul style="list-style-type: none">• HTML• CSS

Capítulo 3. Propuesta y Descripción de la Solución

	<ul style="list-style-type: none">• Tortoise
Montadores	<ul style="list-style-type: none">• HTML• CSS• Tortoise

Nota: Esta propuesta es fundamentada a partir de las responsabilidades que juega el rol dentro del proyecto.

3.5 Actividades para el chequeo del Plan de Aseguramiento de la Calidad

Con el objetivo de lograr que el Plan de Aseguramiento de la Calidad sea bien implantado en el proyecto a desarrollar, se han diseñado un conjunto de actividades para medir en qué grado se cumple cada una de las mismas y confirmar que el producto final sea entregado con elevados niveles de calidad.

Estas actividades fundamentalmente deben chequear la administración, documentación, la adherencia a los estándares, la revisión, el testeado del producto así como las auditorías que se le apliquen al mismo.

Tabla 11 Actividades para el chequeo de la administración

Actividad	
Criterios de Entrada	Documento con la especificación de la estructura gerencial del equipo de SQA
Revisión	Examinar estructura gerencial de la organización encargada del SQA Identificar tareas de cada integrante de la gerencia Definir responsabilidades a cada integrante de la gerencia
Criterios de Salida	Estructura de la administración del departamento SQA revisada

Capítulo 3. Propuesta y Descripción de la Solución

--	--

Tabla 12 Actividades para el chequeo de la documentación

Actividad	
Criterios de Entrada	Documentación
Revisión	Revisión y análisis del plan de documentación - Buscar discrepancias - Discutir discrepancias con el líder del proyecto
Criterios de Salida	Documentación revisada

Tabla 13 Actividades para el chequeo de la adherencia a los estándares

Actividad	
Criterios de Entrada	Documentación, código, comentarios, casos de prueba, métricas
Documentación	Monitorear adherencias de los documentos a los estándares
Codificación	Monitorear adherencias de la codificación a los estándares

Capítulo 3. Propuesta y Descripción de la Solución

Comentarios	Monitorear adherencias de los comentarios a los estándares
Prueba	<p>Monitorear adherencias de las pruebas a los estándares</p> <p>Monitorear adherencia de las pruebas a las prácticas definidas</p>
Métricas	Revisar la métrica definida
Conformidad	Monitorear la conformidad que existe en el sistema
Criterios de Salida	<p>Proceso de Documentación revisado</p> <p>Proceso de Codificación revisado</p> <p>Proceso de Comentarios revisado</p> <p>Proceso de Pruebas revisado</p> <p>Métricas definidas revisadas</p> <p>Conformidad revisada</p>

Tabla 14 Actividades para el chequeo de las revisiones y las auditorías

Actividad	
Criterios de Entrada	Código, requerimientos de las revisiones

Capítulo 3. Propuesta y Descripción de la Solución

Revisión	Revisar el propósito de cada revisión Participar en revisiones de código Examinar argumentos de revisión y auditoría Verificar que el mecanismo de revisión sea acorde al tipo de proyecto Identificar los requerimientos mínimos para las revisiones
Criterios de Salida	Proceso de Revisión y Auditoría revisadas

3.6 Conclusiones del capítulo

- Se propuso una serie de actividades de chequeo y revisión para el Plan de Desarrollo de Software, el Plan de Gestión de la Configuración de Software del proyecto para que estos se desarrollen de la manera correcta y beneficien el proceso de producción, ya que todas las actividades se rigen por estos planes.
- Se desarrolló un Plan de Aseguramiento de la Calidad para el proyecto que dirigirá de la manera correcta a los desarrolladores a crear un producto final con adecuados niveles de calidad, aplicando las métricas, estándares, metodologías y herramientas apropiadas.
- Para una correcta aplicación de esta estrategia el equipo de desarrollo de software y el equipo de SQA deben estar en constante intercambio.
- La preparación del equipo de SQA posibilitará una mejora en las actividades de aseguramiento de la calidad, y un mejor proceso de producción de software.

Capítulo

4

Capítulo 4. Validación de la Propuesta

4.1 Introducción

En el presente capítulo se analizan los resultados obtenidos luego de la aplicación de la propuesta de solución en el proyecto “Contenidos Educativos Digitales”. Dicha propuesta ha sido aplicada solamente hasta Elaboración, fase donde se encuentra actualmente desarrollándose el proyecto.

4.2 Resultados de la Aplicación de la Propuesta de Solución

4.2.1 Entrenamiento del Equipo de Desarrollo

Hasta el momento se ha impartido el curso de Tortoise, el cual permitió principalmente conocer el manejo de este programa, con el propósito de gestionar archivos en el repositorio. Entre las ventajas que brindó este curso se pueden enumerar las siguientes:

- 1) Se logró mantener actualizada la documentación del proyecto.
- 2) Se logró un control eficaz del acceso al repositorio.
- 3) Se instruyó a todo el equipo de desarrollo de CED en el uso de esta herramienta.

Se impartió además el curso de ApEM-L a los equipos de analistas y de arquitectos con el propósito de que tuvieran un mayor conocimiento en el uso del lenguaje, actividad muy importante pues se consiguió

un mayor entendimiento a la hora de modelar los guiones enviados desde SIS-Copextel, reduciendo así los errores en los documentos resultantes de esta actividad.

Los equipos de arquitectos, montadores e implementadores recibieron cursos de HTML y CSS, adquiriendo de esta forma conocimientos que en este momento están aplicando en la implementación de los módulos del producto que se encuentran en desarrollo. Gracias a las clases recibidas, actualmente pueden enfrentarse a la tarea de implementar los módulos del producto mejor capacitados. Dentro de las propuestas de cursos para estos tres equipos se recomienda que se incluya Javascript, pues los integrantes expresan que con el uso de este lenguaje pueden implementarse fácilmente muchas funcionalidades del producto y que no están capacitados para su uso.

Por su parte el equipo de calidad no ha tenido oportunidad de pasar los cursos necesarios para su correcta formación, por lo que ha presentado algunos problemas para enfrentarse a las actividades propias de este equipo. Solo un integrante recibió el curso de Aseguramiento de la Calidad, por lo que se encuentra asesorando a los demás estudiantes de este grupo, asimismo otros alumnos han pasado cursos que están fuera de los propuestos en la estrategia, pero que de alguna forma contribuyen a su instrucción como responsables de la calidad.

Con el fin de verificar si con la propuesta de entrenamiento se ha resuelto el problema concerniente a la falta de conocimientos que presentaba el equipo de desarrollo, y validar hasta qué punto se ha logrado mejorar en este sentido, se realizaron entrevistas a diferentes integrantes del proyecto que desempeñan distintos roles dentro del mismo.

El objetivo de esta entrevista era principalmente medir el nivel de aceptación que tuvieron los cursos dentro del equipo, así como obtener una valoración sobre el impacto que tuvo la preparación adquirida en vistas a desarrollar el producto.

Después de realizadas 10 entrevistas se pudo constatar lo siguiente:

- El 100% de los entrevistados afirman que estos cursos fueron de suma importancia para su preparación a la hora de enfrentarse a las actividades de sus roles dentro del proyecto, que los mismos constituyeron la base que les permitirá desarrollar las tareas que requiere un proyecto para salir adelante.
- El 100% plantea que contribuyeron a su formación como futuros ingenieros en ciencias informáticas.

- Algunos interrogados dicen que a pesar de presentar anteriormente conocimientos preliminares en el uso de algunas de estas técnicas y lenguajes, los cursos sirvieron para profundizar y afianzar sus conocimientos.
- Otros plantean que a pesar de haber tenido gran importancia, algunos cursos se quedaron por debajo de sus expectativas, y solo les brindaron conocimientos básicos sobre la materia que se trataba.

Los autores de este trabajo valoran que en dependencia del interés presentado por el estudiante y las ambiciones de prepararse autodidactamente, será el nivel de preparación que adquiera el mismo. Que no deben limitarse a los temas tratados dentro de los planes del curso, sino que pueden hacer un uso adecuado de las tecnologías existentes en la UCI para prepararse mejor y aumentar así sus ideas a la hora de enfrentarse al proceso de desarrollo. En el caso particular del equipo de calidad que no tuvo oportunidad de optar por estos cursos propuestos, como alternativa posible se pueden realizar investigaciones sobre cómo los equipos de calidad de otros proyectos desempeñan sus labores dentro del mismo y aplicarlos en “Contenidos Educativos Digitales”.

4.2.2 Definición del Equipo Asegurador de la Calidad

La aplicación de la estrategia comienza con la reestructuración del equipo Asegurador de la Calidad, dado que la antigua organización no satisfacía de una manera óptima las necesidades reales del equipo de desarrollo, se hizo necesario redefinir los roles de todo este grupo. Luego de haber definido qué integrantes desempeñaría cada rol, se analiza lo positivo que trajo consigo esta nueva estructura:

- 1) Con esta nueva organización se experimentaron mejoras en la organización de las actividades del equipo de calidad.
- 2) El estudiante se siente más identificado con sus funciones dentro del equipo, se especializa en sus responsabilidades específicas, ganando experiencia y seguridad en su rol.
- 3) Se ahorró tiempo en la realización de las tareas, ya que no solo se cumplió con el cronograma de trabajo sino que en algunos casos se adelantaron en los plazos de entrega.

4.2.3 Revisiones

Se realizaron las revisiones propuestas como parte del plan de aseguramiento de la calidad en la fase de Inicio, como resultado de las mismas se pudo corroborar que existen problemas en algunas de las planillas que conforman el expediente de proyecto, entre estas el Plan de Gestión de Configuración de Software, en el cual se encontró que no se había definido ningún procedimiento de gestión de configuración de software para señalar los cambios, así como para registrarlos y divulgarlos. Como resultado de la revisión de esta planilla se emitió un registro con todas las no conformidades encontradas, y a partir de que se les dio solución se comenzó a realizar un mejor trabajo en esta área. Se sometieron a revisiones otros documentos y en todos los casos se pudo comprobar que en sentido general no se habían llenado correctamente.

En la fase de Elaboración se procedió a revisar el artefacto “Documentación Ingenieril del CED”, en la primera iteración se encontraron múltiples problemas, los analistas no se habían guiado correctamente por las pautas establecidas para la elaboración de este documento, omitiendo algunas normas que el mismo planteaba. Luego de esta primera revisión y dada la gran cantidad de incongruencias detectadas se realiza una reunión con el jefe del equipo de analistas para notificar los problemas más comunes en que incurrieron los analistas. Inmediatamente se realizó una reunión donde participaron el equipo de calidad y el de analistas con el fin de analizar las razones por las cuales se encontraron repetidamente algunos problemas. A partir de este momento durante la segunda iteración de la fase se revisó nuevamente la Documentación Ingenieril del CED y se confirmó que habían disminuido en gran medida las dificultades. En este artefacto en particular tuvieron un impacto positivo las revisiones periódicas, ya que a partir del resultado de las mismas se pudo refinar el modelado de los guiones en vistas a una mejor aceptación por parte de SIS-Copextel.

4.3 Conclusiones del capítulo

- La validación permitió corregir y perfeccionar el sistema de acciones propuesto, pues todos los problemas detectados sirvieron para refinar la elaboración final de la estrategia, la cual quedó satisfactoriamente validada.

- La aplicación de la propuesta de solución demostró que la estrategia es una herramienta de gran utilidad para la realización del proceso de aseguramiento de la calidad del proyecto “Contenidos Educativos Digitales”.

Conclusiones

- Se estudiaron los diferentes enfoques de aseguramiento de la calidad según algunos de los modelos, normas y estándares para el desarrollo del software.
- Se definió una estrategia de aseguramiento de la calidad para el proyecto “Contenidos Educativos Digitales” sustentada por el Plan de aseguramiento de la calidad como artefacto fundamental, además de la definición del equipo de SQA, y los procesos de revisiones y pruebas en el proyecto.
- Los roles propuestos dentro del equipo de SQA permitirán un mayor control de las actividades propuestas.
- Se definió un conjunto de actividades de revisión para hacer más rigurosa la aplicación correcta del plan de aseguramiento de la calidad.
- A medida que se fue aplicando la estrategia en el proyecto se registraron los resultados obtenidos, esto permitió validar la eficiencia e importancia de la solución propuesta, que sin duda contribuirá a elevar la eficacia del proceso de desarrollo de software empleado en el proyecto “Contenidos Educativos Digitales”.

Recomendaciones

- Aplicar la propuesta realizada para las futuras fases del proyecto “Contenidos Educativos Digitales”.
- Terminar de aplicar en el proyecto las actividades de prueba planificadas que asegurarán la calidad en la terminación del producto que se desarrolla.
- Capacitar a todo el personal de aseguramiento de la calidad del proyecto, en los diferentes aspectos presentes en la estrategia para que puedan dar cumplimiento de manera efectiva a las actividades orientadas en la propuesta.
- Analizar los planes de estudio de los cursos optativos propuestos como parte del entrenamiento del equipo de desarrollo, en vistas a posibles modificaciones de los mismos con el objetivo de enfocarlos a las necesidades reales del proyecto “Contenidos Educativos Digitales”.

Referencias Bibliográficas

- [1] *Calidad en Ingeniería del Software*. 2002.[1]
- [2] IDEM 1
- [3] *R.A.E Diccionario de la Lengua Española*. 2002.
- [4] **Torregosa Sánchez, R.** [En línea] [Citado el: 10 de enero de 2009.]
http://chguv.san.gva.es/Descargas/Gerencia/PlaniCalidad/Kiosco/Calidad_generalidades.pdf .
- [5] *ISO. Administración y Aseguramiento de la calidad*. 1994.
- [6] *NC-ISO 9000 "Sistema de Gestión de la Calidad. Fundamentos y Vocabulario"*. 2005.
- [7] **Jurán, J M.** GestioPolis. [En línea] [Citado el: 5 de febrero de 2009.]
<http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/aucalid.htm>..
- [8] **Deming, E.** GestioPolis. *Autores de la Gestión de la Calidad*. [En línea] 1989. [Citado el: 15 de enero de 2009.]
- [9] **Crosby, P.** GestioPolis. [En línea] 1989. [Citado el: 5 de febrero de 2009.]
<http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/aucalid.htm>..
- [10] **Scalone, Fernanda.** *Estudio Comparativo de los Modelos y Estándares de Calidad del Software* . Buenos Aires : s.n., 2006.
- [11] **Pressman, R S.** *Ingeniería del Software, Un enfoque práctico*. La Habana : s.n.
- [12] Capability Maturity Model Integration. *Software Engineering, I*. 2002.
- [13] **Moreno García, M N, García Peñalvo, F J y Polo Martín, M J.** *Medición de la calidad del software y el ámbito de la especificación de Requerimientos*. 2000.
- [14] *Factores que determinan la calidad del software* . (s.f.). Recuperado el 13 de diciembre de 2008, de <http://dmi.uib.es/~bbuades/calidad/sld021.htm>
- [15] **Guerrero, L A.** Taller UML. [En línea] [Citado el: 25 de noviembre de 2008.]
- [16] **Febles, Ailyn.** *Un modelo de Referencia para la Gestión de Configuración en la PYME de Software. Tesis Doctoral*. Ciudad de La Habana : s.n., 2003.
- [17] **Software, R.** *Rational Rose Enterprise Edition*. . 2003.

- [18] **IBM Rational, S.C.** *Rational Unified Process. Help.* . 2003.
- [19] **Goodman, P.** [En línea] [Citado el: 10 de enero de 2009.] <http://sharespot.flazx.net/download/1931332266.zip.htm>. ISBN: 1- 931332-26-6..
- [20] **Calero Muñoz, C.** *Calidad de Sistemas de Software.* 2006.
- [21] IDEM 11
- [22] IDEM 11
- [23] Calidad de software asegurada. [En línea] [Citado el: 15 de enero de 2009.] http://download.boulder.ibm.com/ibmdl/pub/software/dw/ssa/Artigo-ualidade_de_software_assegurada.pdf.
- [24] IDEM 6
- [25] IDEM 10
- [26] *La Industria de Software y Servicios Informáticos.* (s.f.). Recuperado el 30 de mayo de 2009, de <http://www.econlink.com.ar/cluster-software/industria>
- [27] *El software argentino busca su nuevo despliegue.* (s.f.). Obtenido de <http://www.wharton.universia.net/index.cfm?fa=viewArticle&ID=749>
- [28] *La revista del empresario cubano.* (s.f.). Recuperado el 30 de mayo de 2009, de http://www.betsime.disaic.cu/secciones/tec_julsep_08.htm
- [29] *Desoft Exelencia en software. Villa Clara .* (s.f.). Recuperado el 30 de mayo de 2009, de http://www.vcl.desoft.cu/index.php?option=com_frontpage&Itemid=1
- [30] *Informática 2009-06-04.* (s.f.). Recuperado el 30 de mayo de 2009, de <http://www.informaticahabana.com/files/relatoria/Calidad%20jueves%2012%20ma%C3%B1ana.pdf>
- [31] *Calisoft. Centro para la excelencia en el desarrollo de productos tecnológicos.* (s.f.). Recuperado el 30 de mayo de 2009, de http://calidadsoft.prod.uci.cu/index.php?option=com_content&view=article&id=34&Itemid=27
- [32] **Manso, E.** *Calidad del Software .* 2005.
- [33] Villagra, S. (s.f.). *Una introducción a CMMI.* Obtenido de <http://www.sergiovillagra.com/Contenidos/Recursos/WP03%20Una%20Introduccion%20a%20CMMI.pdf>

Bibliografía

[A D, A. \(2002\). *Gestión , Control y Garantía de la Calidad de Software.*](#)

[Abud Figueroa, M. A. \(s.f.\). *MECSE: Conjunto de Métricas para Evaluar Software Educativo.*](#)

[Abud Figueroa, M. A. \(s.f.\). *MECSE: Conjunto de Métricas para Evaluar Software Educativo.* Recuperado el 27 de marzo de 2009, de <http://www.revistaupiicsa.20m.com/Emilia/RevSepDic05/Antonieta.pdf>](#)

[Alarcón, A. S. \(2004\). *Modelos de calidad.*](#)

[Bañares, J. P. *Introducción a ISO/IEC 15504.*](#)

[Calero, C. *Calidad en la IS: el papel de la Universidad.*](#)

[Cueva Louvelle, J. M. \(1999\). Recuperado el 4 de febrero de 2009, de \[http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF\]\(http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF\)](#)

[Cukier, J. J. \(2006\). Obtenido de \[http://www.bloggers.com.ar/system/noticia_detalle.php?id_prod=626\]\(http://www.bloggers.com.ar/system/noticia_detalle.php?id_prod=626\)](#)

[Dirección de calidad del software. \(s.f.\). Recuperado el 8 de febrero de 2009, de \[http://calidadsoft.prod.uci.cu/index.php?option=com_content&view=article&id=34&Itemid=27\]\(http://calidadsoft.prod.uci.cu/index.php?option=com_content&view=article&id=34&Itemid=27\)](#)

[Escalona, M., & Koch, N. \(2002\). Recuperado el 23 de enero de 2009, de <http://lsiweb.lsi.us.es/docs/informes/LSI-2002-4.pdf..>](#)

[Estrada, A. F., & Cárdenas, A. S. *calidad de Software y la empresa, enseñanza de un tema imprescindible para el Ingeniero Informático.*](#)

[Ferré Grau, X. \(2005\). *Marco de Integración de la Usabilidad en el Proceso de Desarrollo de Software.*](#)

García, J. (14 de febrero de 2005). Recuperado el 23 de enero de 2009, de <http://www.ingenierosoftware.com/calidad/cmm-cmmi.php>

Informática. (2000). Obtenido de <http://sistemas.dgsca.unam.mx/publica/pdf/Control%20de%20Calidad.PDF>.

ISO 9000 : Sistema de Gestion de la Calidad. (2001).

Mendoza, L. E. *Sistemas de Información III*.

NC ISO/IEC 9126-1:"Modelo de Calidad". (2005).

Torregosa Sánchez, R. (s.f.). Recuperado el 10 de enero de 2009, de http://chguv.san.gva.es/Descargas/Gerencia/PlaniCalidad/Kiosco/Calidad_generalidades.pdf.

Villena, A. M., & Martín, A. J. *Modelos de calidad de software*.

Glosario de Términos

- **SQA**: Aseguramiento de la Calidad del Software (Software Quality Assurance). El aseguramiento de calidad del software es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) requiere para satisfacer los requerimientos dados de calidad por parte del cliente.
- **RUP**: Proceso Unificado Racional (Rational Unified Process). Es un proceso de desarrollo de software. Es una metodología que se utiliza para realizar la documentación de productos de software, tiene como objetivo asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles.
- **CMMI**: Modelo Integrado de Capacidad y Madurez (Capability Maturity Model Integration) es un modelo para la mejora de procesos que proporciona a las organizaciones los elementos esenciales para procesos eficaces.
- **ISO**: Organización Internacional para la Normalización (International Organization for Standardization) Es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica.
- **ApEM-L**: Lenguaje para la Modelación de Aplicaciones Educativas, notación para representar los elementos estructurales, lógicos, funcionales, pedagógicos y de patrones de ingeniería al modelar los software educativos en la UCI
- **Métricas del software**: La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo, así el desarrollador junto al empleo de estas técnicas mejorará el proceso y sus productos
- **RTF**: Revisiones Técnicas Formales. Tipo de revisión que se realiza al software para encontrar errores durante el proceso, de forma que no se conviertan en defectos después de la entrega del software.

- **Tick-IT:** Modelo creado por el Departamento de Comercio e Industria del Reino Unido, para desarrollar un sistema de certificación aceptable en el mercado, estimular a los desarrolladores de software a implementar sistemas de calidad, dando la dirección y guías necesarias para tal efecto.
- **ISO-SPICE:** Software Process Improvement and Capability Determination. Modelo de evaluación de procesos de software para la mejora continua, la evaluación de la capacidad como base para el comercio internacional de software.
- **TICs:** Tecnologías de la Información y la Comunicación. Conjunto de servicios, redes, software y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario.
- **MIC:** Ministerio de la Informática y las Comunicaciones. Organismo encargado de regular, dirigir, supervisar y controlar la política del Estado y del Gobierno en cuanto a las actividades de tecnologías informáticas, telecomunicaciones, redes de infocomunicaciones, entre otras.
- **Desoft:** Empresa Nacional de Software en Cuba, entidad legal con personalidad jurídica que de ella surge la división Desoft Villa Clara, dedicada a prestar servicios de desarrollo, producción y comercialización mayorista de software y aplicaciones informáticas de todo tipo, prestar servicios de ejecución y comercialización mayorista de soluciones informáticas integrales, representa, sirve de intermediaria y distribuye en el país productos y soluciones informáticas nacionales y extranjeras y ofrece sus servicios asociados, brindando servicios de instalación mantenimiento garantía y postventa , servicios de capacitación y formación en tecnologías de la información y automatización, para la gestión, la administración y el control, fundamentalmente en forma de aplicaciones dirigidas a la informatización de cualquier tema o rama de la administración, el comercio o la industria, contemplando todas las formas de automatización que potencialmente puedan realizarse.
- **Calisoft:** Centro de calidad del software en Cuba. Entidad que asesora al cliente en la ejecución de pruebas de aceptación, con el objetivo de diseñar sistemas informáticos de alto nivel, sus principales objetivos son: entregar productos de gran calidad con estricta puntualidad esta entidad está integrada por especialistas de distintas universidades del país y empresas del MIC.
- **CED:** Contenidos Educativos Digitales, proyecto que se encuentra actualmente en desarrollo en la Facultad 8 de la Universidad de las Ciencias Informáticas

- **CENIT**: Centro Nacional de Innovación Tecnológica. Es una fundación adscrita al Ministerio del Poder Popular para Ciencia y Tecnología (Mppct) en Venezuela que tiene como objeto propiciar la investigación, el desarrollo y la innovación en el área de las tecnologías de comunicación e información, de acuerdo con las necesidades del modelo socio productivo del país.
- **SIS–Copextel**: Empresa cubana perteneciente al Ministerio de la Informática y las Comunicaciones.
- **Artefactos**: Producto tangible resultante del proceso de desarrollo de software.
- **Lista de Chequeo**: técnica que se utiliza para realizar las inspecciones al software.
- **LC**: Lineamientos de Calidad, políticas de calidad para la producción de software en la Universidad de la Ciencias Informáticas.
- **IP**: Infraestructura Productiva de la UCI.
- **GCS**: Gestión de Configuración de Software. Proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.
- **Tortoise**: Curso impartido en el proyecto con el propósito de gestionar archivos en el repositorio.
- **HTML**: Lenguaje de Marcas de Hipertexto (**HyperText Markup Language**), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.
- **CSS**: Hojas de Estilo en Cascada (*Cascading Style Sheets*, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML)

Anexos

Anexo 1. Factores. Satisfacción del Cliente

Factor	Ejemplo
Producto o servicio	Diseño de calidad de las materias primas, calidad del producto o servicio, homogeneidad, fiabilidad.
Ventas y Posventas	Publicidad, garantías, devoluciones, quejas, servicios, plazos, precios, etc....
Cultura	Valores que la organización proyecta conscientemente o inconscientemente...

Anexo 2. Roles y responsabilidades

Rol	Responsabilidad
Líder de Proyecto	<ul style="list-style-type: none"> ➤ Gestionar y asignar recursos humanos. ➤ Planificar y asignar las tareas de la forma más razonable. ➤ Motivar y organizar el equipo de trabajo para lograr el éxito. ➤ Informar sobre el estado actual del proyecto. ➤ Mantener el plan del proyecto. ➤ Establecer los horarios de trabajo del equipo de desarrollo.
Analista Principal	<ul style="list-style-type: none"> ➤ Responsable de que sean documentados los guiones llevándolo a un estado entendible por los arquitectos y diseñadores. ➤ Define los sistemas, subsistemas de cada contenido.

Analista	<ul style="list-style-type: none"> ➤ Documentar los guiones siguiendo las pautas establecidas.
Arquitectos	<ul style="list-style-type: none"> ➤ Responsable de crear la plantilla.
Líder del Equipo de Desarrollo	<ul style="list-style-type: none"> ➤ Responsable de la implementación y pruebas. ➤ Guía al equipo de desarrollo en una estrategia de desarrollo. ➤ Guía al equipo en el más alto nivel de diseño.
Implementador	<ul style="list-style-type: none"> ➤ Programa todo lo escrito en la etapa de diseño.
Asegurador de Calidad	<ul style="list-style-type: none"> ➤ Planificar el proceso de Aseguramiento de la Calidad en el proyecto (Plan de Calidad). ➤ Realizar el Plan de Prueba, de Revisión y Auditoría de cada iteración. ➤ Coordinar el proceso de recopilación, análisis y reporte de las estadísticas de calidad (Plan de mediciones). ➤ Dar seguimiento a estos planes. ➤ Guiar las revisiones técnicas formales. ➤ Guiar las pruebas que se realicen. ➤ Manejar todo lo relacionado con los riesgos de calidad (Lista de Riesgos). ➤ Realizar el Resumen de Evaluación ➤ Identificar técnicas apropiadas, herramientas e instrucciones para la implementación de las pruebas. ➤ Diseñar los casos de pruebas. ➤ Definir listas de chequeo para las pruebas.
Montador	<ul style="list-style-type: none"> ➤ Responsable de insertar los contenidos en las plantillas.

Anexo 3. Políticas y Pautas de documentación

Proyecto: Producción y Gestión de Contenidos Educativos Digitales

1. Introducción

Con el objetivo de que el proyecto “Producción y Gestión de Contenidos Educativos Digitales (PGCED) de la facultad 8 conste de un buena calidad en la documentación de sus productos, el grupo de analista de dicho proyecto ha elaborado la presente guía de estilo de documentación.

Esta guía contiene un grupo de normas básicas, así como políticas y pautas a utilizar cuando se documenta utilizando el lenguaje de modelación orientado a objetos para aplicaciones educativas multimedia (ApEM_L). Dichas normas deben ser adoptadas por todos los productos que se desarrollen en el proyecto PGCED.

En caso de existir alguna sugerencia, favor escribir a yherrera@uci.cu

2. Normas, políticas y pautas de documentación utilizando el lenguaje de modelación ApEM_L.

2.1 Identificadores.

-
- Escoja identificadores de clases significativos y de ser posible breves. En cualquier caso prefiera la claridad a la brevedad sin exceso de caracteres. Por ejemplo, es preferible el identificador ***CV_Presentación*** que el de ***CV_Prest.***
 - Los identificadores de las clases tipo ***vistas de presentación*** siempre comienzan con la abreviatura “CV” en mayúscula, luego el guión bajo y por último el nombre de la vista de presentación en minúscula excepto para la letra inicial.

Ejemplo:

CV_Cuestionario

- Los identificadores de las clases tipo ***medias*** siempre comienzan con la abreviatura “CMEM” en mayúscula, luego el guión bajo y por último el nombre de la de la media en minúscula excepto para la letra inicial.

Ejemplo:

CMEM_Texto

CMEM_TextoInstructivo

- Los identificadores de las clases tipo ***interacción y estáticos*** siempre comienzan con la abreviatura “CInt” o “CEst” en mayúscula la primera y segunda letra del identificador, luego el guión bajo y por último el nombre de la de la clase interacción o estática en minúscula excepto para la letra inicial.

Ejemplo:

CEst_Presentación

CInt_Interacción.

- Los identificadores de las clases tipo ***botón*** siempre comienzan con la abreviatura “CB” en mayúscula, luego el guión bajo y por último el nombre del botón

Ejemplo:

CB_PáginaInicial

- En caso de usar más de una palabra para el nombre de una clase no usar separadores, ni guiones bajos, es decir, escriba:

CMEM_ImagenFondo

En lugar de:

CMEM_Imagen Fondo

CMEM_Imagen_Fondo

- Para los atributos utilice identificadores en minúsculas excepto la inicial que será en mayúscula. Si el identificador consta de varias palabras se colocan separadas del guión bajo. Cada una de ellas se escribe en minúsculas con la inicial mayúscula.

Por ejemplo:

Id_imagen

- En el caso de los métodos, el identificador debe ser preferentemente un verbo y debe usarse en minúsculas con la inicial en mayúscula. Si el identificador contiene varias palabras se colocan separadas con el guión bajo. Ejemplo válido sería:

Get_texto()

- Para el identificador de los paquetes se recomienda usar minúsculas con la inicial en mayúscula. Si el identificador contiene varias palabras se colocan separadas con un espacio. Ejemplo valido sería: *Obra Poética*.

- Los identificadores de los estereotipos de las relaciones siempre van en minúsculas. Ejemplo: <<utiliza>>, <<usa>>, <<levanta>>

2.2 Tipos de clases de los modelos.

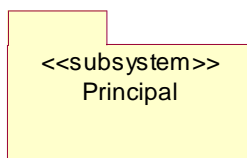
Clases de tipo <i>Vistas de Presentación (DEP y DEN)</i>	Clases de tipo <i>Media (DEP y DEN)</i>
--	---

Estereotipo:	<<Vista>>	Estereotipo:	<<Mod_Ent_Med>>
Identificador:	CV_	Identificador:	CMEM_
Notación:		Notación:	
Clases de tipo <i>Interacción (DEP)</i>		Clases de tipo <i>Estáticos (DEP)</i>	
Estereotipo:	<<Interacción>>	Estereotipo:	<<Estáticos>>
Identificador:	CInt_	Identificador:	CEst_
Notación:		Notación:	
Clases de tipo <i>botón de Imagen (DEP)</i>		Clases de tipo <i>botón de texto (DEP)</i>	
Estereotipo:	En dependencia del leguaje de alto nivel (HLL) que se utilice por ejemplo: <ul style="list-style-type: none"> • <<HTML>> • <<Flash>> • <<AcciónScript>> 	Estereotipo:	En dependencia del leguaje de alto nivel (HLL) que se utilice por ejemplo: <ul style="list-style-type: none"> • <<HTML>> • <<Flash>> • <<AcciónScript>>
Identificador:	BtnImagen_	Identificador:	BtnTexto_
Notación:		Notación:	
Clases de tipo <i>Botón (DEN)</i>		Clases de tipo <i>Consulta (DEN)</i>	
Estereotipo:	<<Botón>>	Estereotipo:	<<Consulta>>
Identificador:	CB_	Identificador:	CCon_

Notación:	<pre> <<Botón>> CB_GuiadeUsuario (from Clases Botón) </pre>	Notación:	<pre> <<Consulta>> CCon_Temas </pre>
Clases de tipo <i>Menú</i> (DEN)			
Estereotipo:	<<Menú>>		
Identificador:	CMen_		
Notación:	<pre> <<Menú>> CMen_Presentacion Temas Ejercicios </pre>		

2.3 Normas del Diagrama Vista de Gestión del Modelo.

- Los subsistemas deben estar representados en paquetes; utilizar el estereotipo <<subsystem>> quedando de la siguiente forma.



- Las relaciones entre los subsistemas son relaciones de dependencia y se utiliza el estereotipo decorativo: -----> .

2.4 Normas del Diagrama de Estructura de Presentación (DEP).

- Las clases de tipo **Interacción y Estáticos** siempre pertenecen al **diagrama de estructura de presentación** y su nombre debe ser igual al nombre de la vista de presentación a la que pertenece:

Ejemplo:

Si la vista se nombra CV_Presentación, las clases estáticas y de interacción son respectivamente:

CEst_Presentación

CInt_Interacción.

- Los tipos de multiplicidad son de uno a uno, de uno a una cantidad específica de elementos y de uno a muchos. Su identificador puede ser:

Uno a uno: 1 ----- 1

Uno a cantidad específica: 1-----1..3

Uno a muchos: 1----- 1..n

- *Solo se representa un DEP por vista de presentación identificada. En caso de que la vista de presentación contenga otra vista de presentación se debe representar en el diagrama utilizando la relación de dependencia -----> y el estereotipo <<utiliza>> o <<usa>>*
- El DEP de las vistas de presentación contenidas pueden aparecer representado de dos formas:
 1. En el DEP de la vista contenedora. (Ver fig 1)
 2. Agruparlo en un paquete y crear la relación de dependencia. El paquete contendría el DEP de la vista contenida. (Ver fig 2)

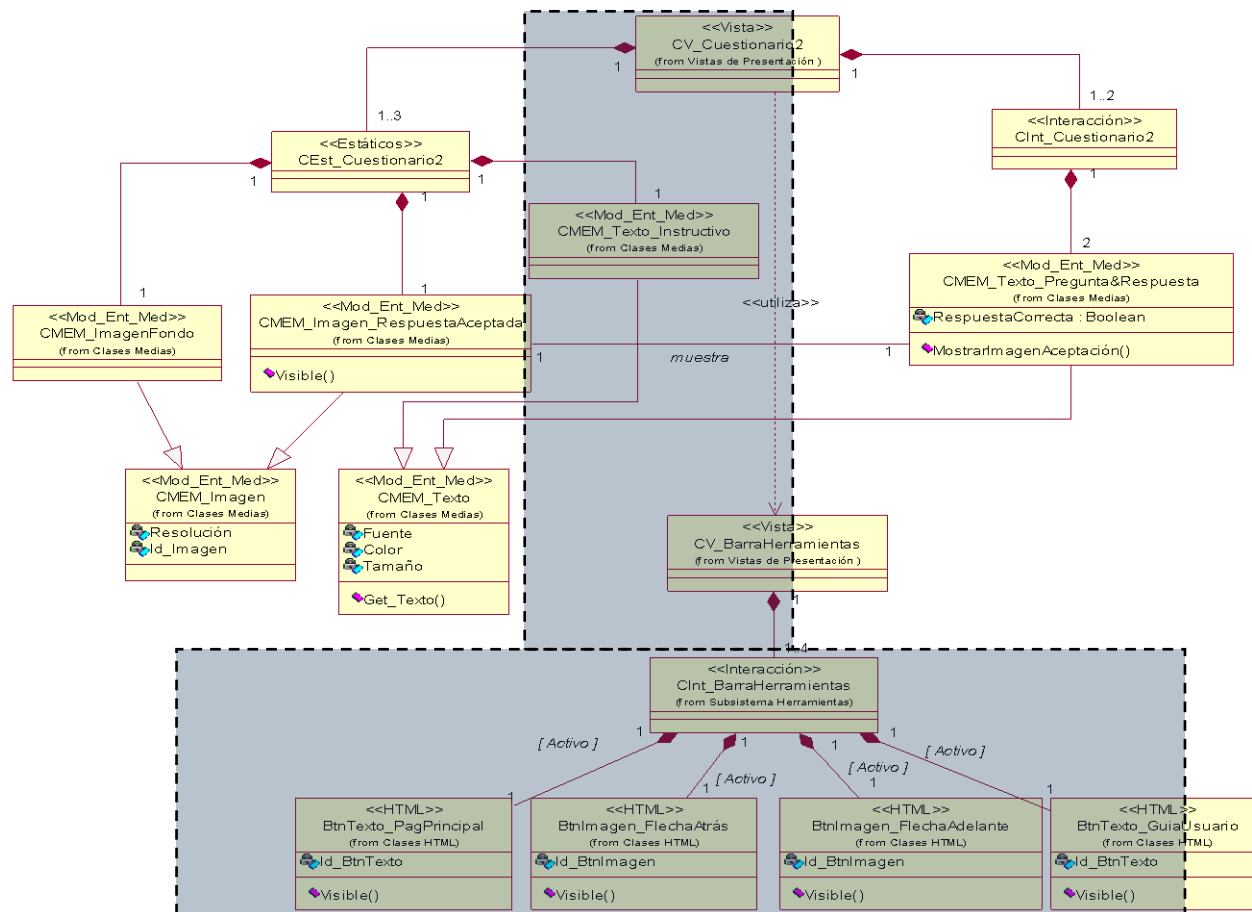


Fig 1. Tipo de representación válida

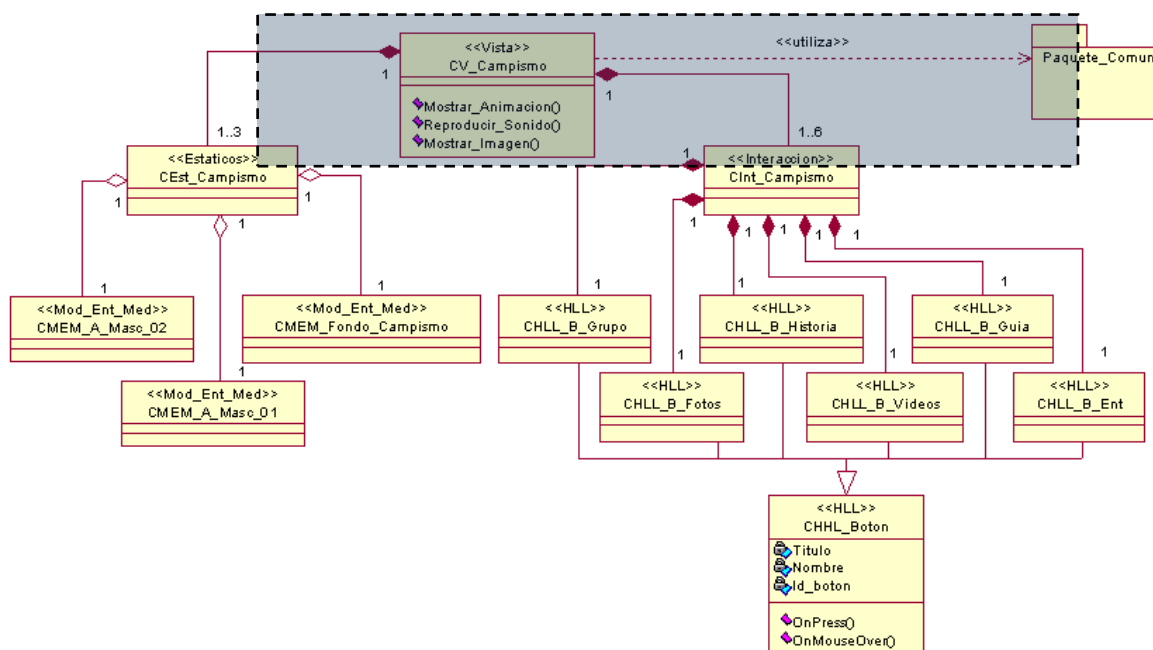
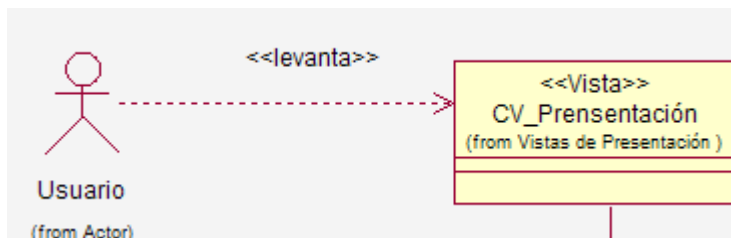



Fig 2. Tipo de representación válida.

2.5 Normas del Diagrama de Estructura de Navegación (DEN).

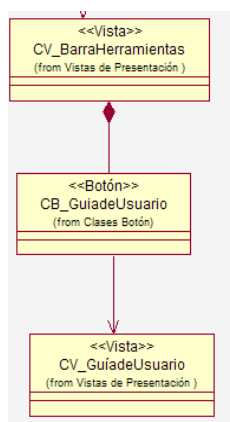
- Puede existir tantos digramas de estructura de navegación como transacciones navegables existan en un subistema de la aplicación.
- Debe estar reflejado en el DEN el artor que da inicio a la navegación.
- Las relaciones que se utilizan para representar la inicialización de la navegación son relaciones de dependencia: -----> y siempre van en dirección a la clase vista de presentación que se interactua.
- El estereotipo que se utiliza en la relación que da inicio a la nevación es: <<levanta>> o <<interactua>> y deben estar reflejado en el diagrama.

Ejemplo:



- Solo se representa en el DEN los elementos de la vista de presentación que denotan navegación en la misma.
- Para denotar la navegación de un elemento hacia otro elemento del DEN se utiliza el la relacion de asociación  en dirección del elemento que denota la navegación hacia el elemento destino.

Ejemplo:



- Las clases de tipo **Botón**, **Consulta**, **Menú**, **CMEM_Imagen**, **CMEM_Texto** son clases que pertenecen al **diagrama de estructura de navegación** siempre y cuando denoten navegación en el sistema.
- Si el DEN contiene estereotipos restrictivos que denoten navegación en el sistema, estos deben estar descritos entre llaves ({...}) y debe reflejar condición expresada en lenguaje natural o leguaje de programación.

Ejemplo:

{Tiempo_Finalización_Animación = 0}
 {CB_Siguiente.dirección (id)=CV_Presetación}

2.6 Normas de la descripción textual del artefacto vista de presentación (DTVP).

- La DTVP sigue la siguiente estructura:

Descripción Textual de la vista <<Nombre de la vista>>		
Actores de la vista de presentación	<se listan los actores que intervienen en la vista y se señala entre paréntesis el actor que inicia la vista>	
Propósito	<se menciona de forma breve el propósito de la vista para dejar claridad de la función de la misma y del beneficio que tendrá para el actor del sistema que lo inicia>	
Objetivos instructivos	<Se mencionan los objetivos que desde el punto de vista pedagógico se lograra en el actor del sistema con la realización de la vista >	
Resumen	<se redacta de forma breve el resumen de la vista, dejando claridad de la actividad que da inicio a la misma así como de lo más importante del flujo y de las actividades que concluyen en la misma>	
Vistas asociadas	<se listan las vistas relacionadas con esta>	
Referencias	<se establecen los requerimientos funcionales y no funcionales a los cuales satisface la vista>	
Precondiciones	<se numeran las precondiciones que deben de cumplirse para que la vista se ejecute>	
Poscondiciones	<se numeran las poscondiciones que existirán luego que la vista culmine su ejecución>	
Curso Normal de los eventos		
Acción del actor	Respuesta del sistema	Elementos de la Vista
<se numeran las acciones del actor que generan respuestas por el sistema, en orden de ocurrencia>	<se numeran en correspondencia con las acciones del actor, las acciones que dan respuesta a las acciones del actor, para de esta forma completar transacciones del flujo de sucesos en la vista>	<Se describen los elementos de interacción que intervienen en el flujo de suceso de la vista.>
Curso alterno de los eventos		

Acción	Curso Alterno		
<se refleja la acción del sistema que tiene una acción alternativa>	<se expresa que ocurre en términos de acciones del sistema como respuesta al flujo alterno que tomaría el sistema ante la situación determinada>		
Prioridad	<se expresa la prioridad de implementación del caso de uso, siendo: crítica, secundaria u opcional>		
Mejoras	<se numeran las mejoras que implementa la vista de presentación sobre el curso normal de los eventos descritos>		
Tipo de Media	Nombre	Descripción	Estado
Imagen		<se listan las medias o su descripción de estas de no existir que den claridad de las características o condiciones que tienen que cumplir las mismas para ser utilizadas>	<se refleja el estado de la media, sea: - existente, - en localización, - en construcción>
Video o animación			
Sonido			
Texto			
Reglas Instructivas	<se reflejan los elementos que no pueden ser expresados en el cuerpo de la vista (reglas pedagógicas o instructivas) a tomar en consideración para la implementación de la vista>		
Representación de la vista	<se muestra una delimitación dentro de la interfaz de usuario, que represente la vista que se está describiendo>		

- a) Normas básicas de la sección Nombre de la vista de presentación (VP)
- El nombre de la VP debe estar en infinitivo y debe reflejar de manera clara el objetivo de la VP.
 - El nombre de la VP debe ser único.
- b) Normas básicas de la sección Actores de la vista de presentación.
- La vista de presentación debe estar relacionada con al menos un actor.
 - Siempre debe especificarse que actor inicia la vista de presentación. Se pone al lado y entre paréntesis.

Ejemplo: Usuario(inicia)

c) Normas de la sección del Curso normal de los eventos.

- Debe comenzar diciendo “La vista de presentación se inicia cuando el actor...”.
- Debe terminar diciendo “...y se culmina la vista de presentación”.
- Cada acción debe comenzar con el nombre del que lo realiza.
- En el flujo interno de la VP no se describe el funcionamiento interno del sistema, sus componentes o diseño, sino que describa el sistema en base a las responsabilidades que tiene.

d) Normas de la sección Estado.

- Solo existen tres tipos de estados de las medias a utilizar: Existente, en construcción, en localización.

Formas de presentar la información.

- Debe estar escrito en presente. No debe decir que el sistema mostrará una lista de datos.
- Debe describir de manera comprensible y detallada las acciones del actor frente al sistema. Lo más parecido a un manual de ayuda posible.
- Deben estar correctamente reflejadas en el flujo básico las llamadas a las vistas de presentación necesarias.
- Debe dejarse claro la culminación de la vista de presentación que se está describiendo.

2.7 Normas de organización de la documentación.

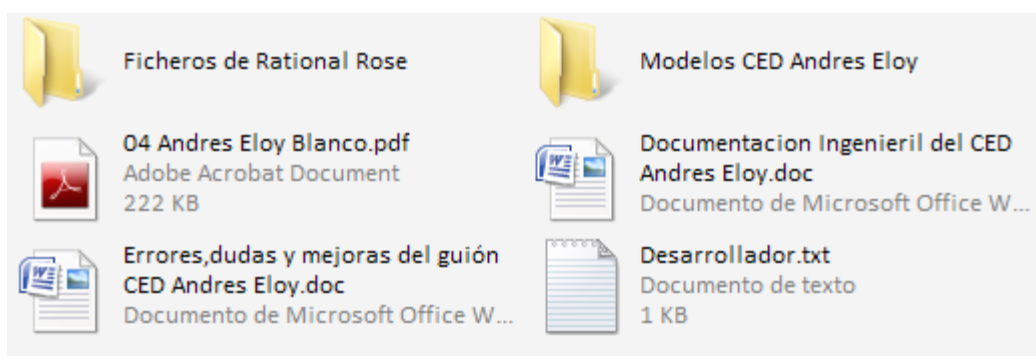
- La documentación estará compuesta por 1 directorio principal y se nombrará con la siguiente estructura: <palabra reservada “CED”> <número del CED> <nombre del CED>

Ejemplo:

CED 4 Descubriendo a Ramos Sucre.

- EL directorio principal está compuesta por 6 componentes:

1. El informe oficial y rector de la documentación ingenieril del CED.
2. Un directorio con el fichero generado por la herramienta CASE utilizada.
3. Un directorio con los modelos del CED generados por la herramienta CASE a través de una WEB.
4. El guión del CED.
5. Un documento con los errores, dudas y mejoras del guión.
6. Un fichero txt con el nombre del proyecto, en nombre del desarrollador y la fecha con la última versión de desarrollo de la documentación del CED.



- La estructura del informe oficial y rector de la documentación del CED en su versión 1.1 estará constituida por: una *Portada*, un *Indice*, el *Capitulo 1* destinado a la identificación de los tipos de pantallas del guión y el *Capitulo 2* destinado a la descripción y modelación del sistema utilizando ApEM_L.

Anexo 4 Tabla de Métricas propuestas para la calidad del software educativo.

Aspecto	Factor	Atributo
Pedagógico	Facilidad de uso· Facilidad de aprendizaje	<ul style="list-style-type: none"> • Diversidad en las actividades propuestas • Actividades adecuadas para reforzar el aprendizaje • Actividades motivadoras para el alumno • Diversidad en las formas de evaluación • Se toma en cuenta el nivel inicial de conocimientos del alumno

		<ul style="list-style-type: none"> • Existe relación con lo que el profesor enseña • Puntaje apropiado en la evaluación
	Funcionalidad	<ul style="list-style-type: none"> • Seguridad del alcance de los objetivos educativos • Manejo automático del historial académico
Interfaz H-C	Facilidad de uso	<ul style="list-style-type: none"> • Son atractivos los elementos del software • Conformidad con el diseño del menú • Novedad en la interfaz
	Funcionalidad	<ul style="list-style-type: none"> • Imágenes claras y comprensibles • Música apropiada al tema • Videos adecuados al tema • Colores adecuados • Manejo adecuado de las características del texto • El puntero del mouse maneja diferentes formas de acuerdo a las acciones a ejecutar • Acceso directo a temas • Acceso al tema anterior • Organización adecuada de las pantallas
De contenido	Facilidad de uso	<ul style="list-style-type: none"> • El contenido es comprensible • La presentación tiene una secuencia lógica
	Confiabilidad	<ul style="list-style-type: none"> • El contenido es confiable • El contenido se apega al programa de estudios vigente
Transportabilidad	Facilidad de uso	<ul style="list-style-type: none"> • Las funciones de cada módulo del software se comprenden fácilmente • Facilidad de aprendizaje del software • Hay ayuda disponible desde cualquier punto del software
	Funcionalidad	<ul style="list-style-type: none"> • La ejecución del software se realiza de manera fácil • Se cuenta con un manual sencillo y completo

		<ul style="list-style-type: none"> Se cuenta con un manual que indica la forma de instalación del software
	Eficiencia	<ul style="list-style-type: none"> La respuesta del software a cualquier acción del usuario es adecuada El tiempo de respuesta del software es adecuado
	Confiabilidad	<ul style="list-style-type: none"> Facilidad de recuperación cuando ocurren errores Permite grabar la sesión en forma automática
		<ul style="list-style-type: none"> Facilidad de instalación

Anexo 5 Lista de Chequeo para el Plan de Gestión de Configuración de Software

Evaluación	Eval.	NP	Comentario
¿Se ha hecho el cambio especificado en la UCI?			
¿Se han incorporado modificaciones adicionales?			
¿Se ha llevado a cabo una revisión técnica formal para evaluar la corrección técnica?			
¿Se han seguido adecuadamente los estándares de ingeniería de software?			
¿Se han "recalcado" los cambios en los ECS?			
¿Se han especificado la fecha del			

cambio y el autor?			
¿Reflejan los cambios los atributos del objeto de configuración?			
¿Se han seguido procedimientos del GCS para señalar el cambio, registrarlo y divulgarlo?			
¿Se han actualizado adecuadamente todos los ECS relacionados?			

Anexo 6 Registro de No Conformidades

1. Aspectos generales

[Descripción de Aspectos Generales a tener en cuenta a la hora de analizar el resultado de las pruebas, incidencias en el momento de su desarrollo y otros aspectos relevantes.]

1.1 Elementos Probados.

[Descripción general o lista de los Elementos Probados, y otros aspectos importantes a tener en cuenta a la hora analizar las No Conformidades Detectadas.]

1.2 Elementos no Probados y causas.

[Descripción general o lista de los Elementos no Probados, la causa de que no se hayan podido realizar las Pruebas y cualquier otro elemento importante que aporte la información necesaria para que sean analizadas estas causas y resueltas para la siguiente iteración.]

2. Tabla de No Conformidades Detectadas

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Importancia	Recomendación
<Nombre del Elemento>	< 1>	<Descripción de la No Conformidad>	<Descripción del Aspecto correspondiente>	<Etapas de detección del error>	<X>	<X>

3. Recomendaciones

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección
<Nombre del Elemento>	< 1>	<Descripción de la No Conformidad>	<Descripción del Aspecto correspondiente>	<Etapas de la detección del error>

Anexo 7 Lista de Chequeo para el Plan de Desarrollo de Software

Criterio de evaluación	Evaluación	S	N	N.P.	Observaciones
¿Se identifica el Polo Productivo al que pertenece el proyecto y se define el tipo de proyecto y la metodología de desarrollo?					
¿Se establece el alcance del proyecto?					
¿Se encuentra identificada la lista de riesgos?					
¿Se establece la estructura organizacional, los roles y las responsabilidades de los miembros del proyecto?					
¿Existe una correspondencia del rol con las tareas asignadas?					
¿Hay vinculación con los procesos docentes e investigativos?					
¿Se registra el capital humano, los recursos materiales y los resultados del proyecto?					

Anexo 8 Lista de Chequeo de los Documentos Ingenieriles

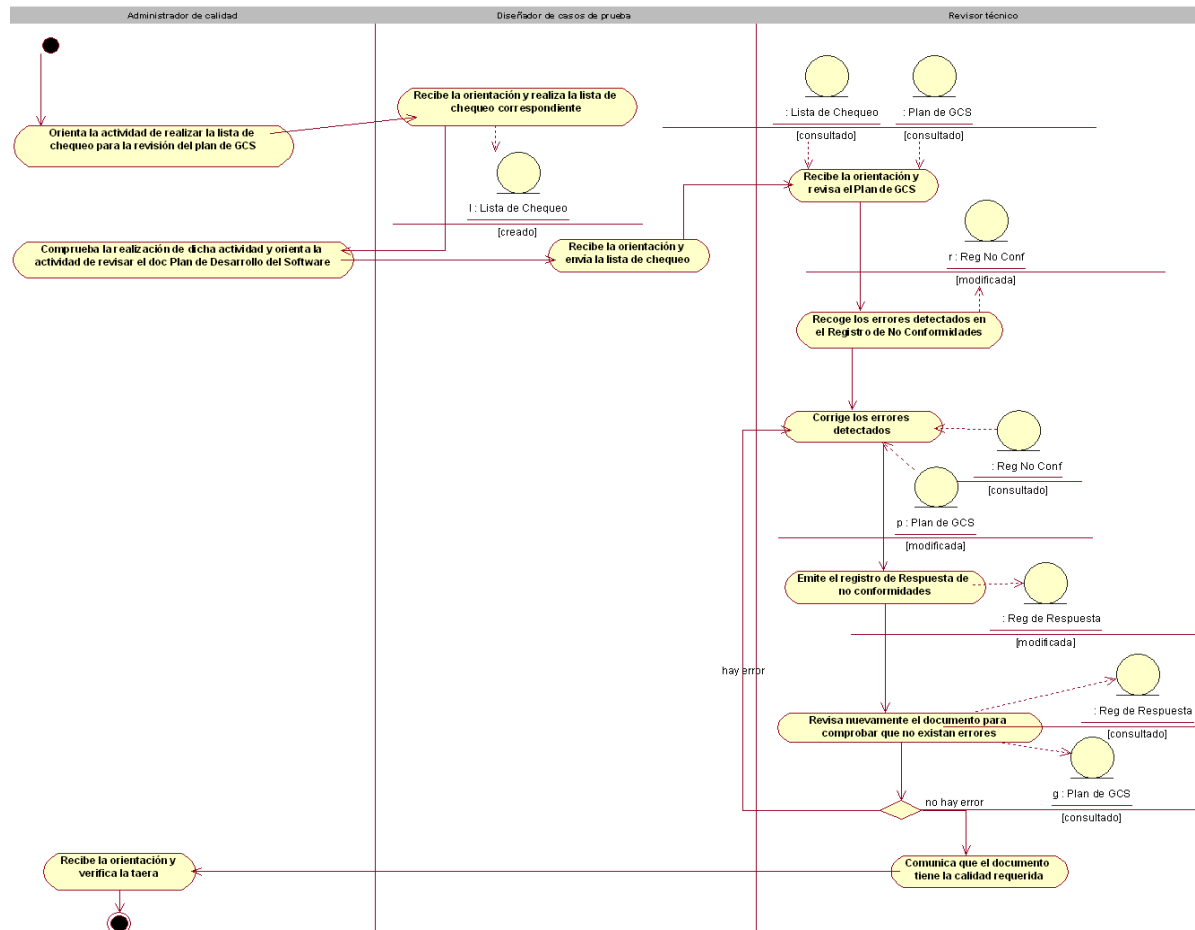
Criterio de evaluación	Evaluación	N.P.	Observaciones
Presentación			
¿El documento a revisar presenta un título de fácil comprensión y escrito correctamente?			

¿Está acorde el título de la presentación con la temática del contenido?			
¿Los identificadores de las clases son significativos, claros y breves?			
¿Los atributos están escritos con letra minúscula excepto la letra inicial?			
¿El nombre de la vista se corresponde con el contenido que presenta?			
¿La vista presenta descripción textual?			
¿Se corresponde la imagen con la temática y el sonido?			
¿Los subsistemas del diagrama Vista de Gestión de Modelos están representados en paquetes?			
¿El documento presenta los diagramas de Vista de Presentación para cada subsistema representados correctamente?			
¿La información esta escrita en presente?			
Módulo de Contenidos			
¿Existe flexibilidad en la selección de contenidos?			
¿Tiene un lenguaje acorde con el usuario?			
¿Tiene un lenguaje acorde con el medio?			
¿Se corresponde el contenido con el público al que va dirigido?			
¿Tiene fiabilidad conceptual?			

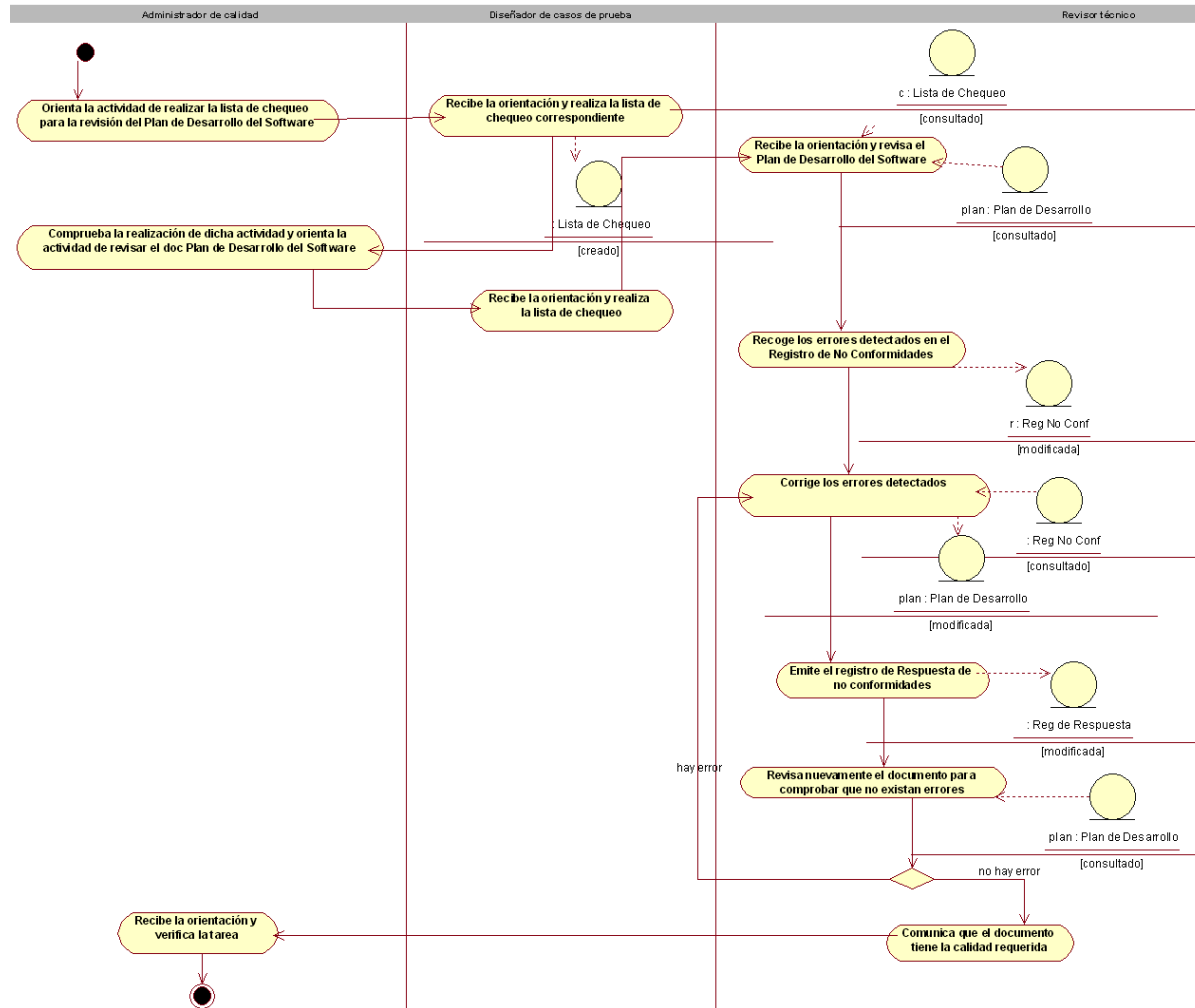
Imágenes.			
¿Existe fiabilidad funcional?			
¿Se corresponde las imágenes con los objetivos propuestos?			
¿Tienen calidad las imágenes?			
¿Existe un nivel instructivo en la imagen?			
Sonidos			
¿Presentan los sonidos fiabilidad funcional?			
¿Se corresponde los sonidos con los objetivos propuestos?			
¿Tiene calidad el sonido?			
¿El sonido presenta un nivel instructivo?			
Videos			

¿Presentan los videos fiabilidad funcional?			
¿Se corresponden los videos con los objetivos propuestos?			
¿Existe calidad en los videos?			
¿Presentan los videos un nivel instructivo?			

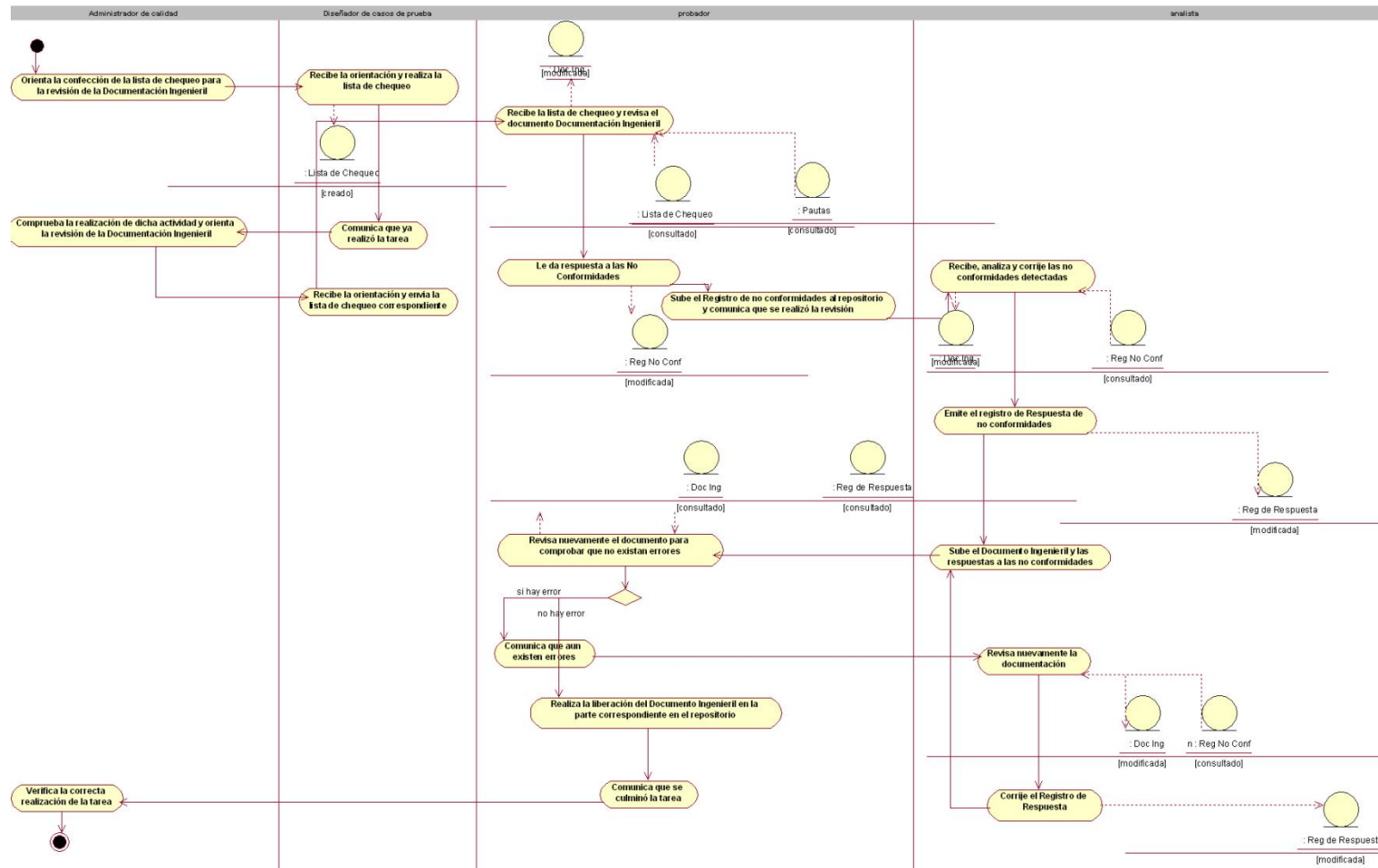
Anexo 9 Diagrama de Actividades de la Revisión del Plan de Gestión de Configuración de Software



Anexo 10 Diagrama de Actividades de la Revisión del Plan de Desarrollo de Software



Anexo 11 Diagrama de Actividades de la Rev. de la Documentación Ingenieril del CED



Anexo 12 Diagrama de Actividades del Proceso de Pruebas

