

Universidad de las Ciencias Informáticas

Facultad 5



**Título: Estrategia para el Aseguramiento de la
Calidad del Proyecto Juegos CNeuro.**

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

Autores: Eniel Corzo Rodríguez

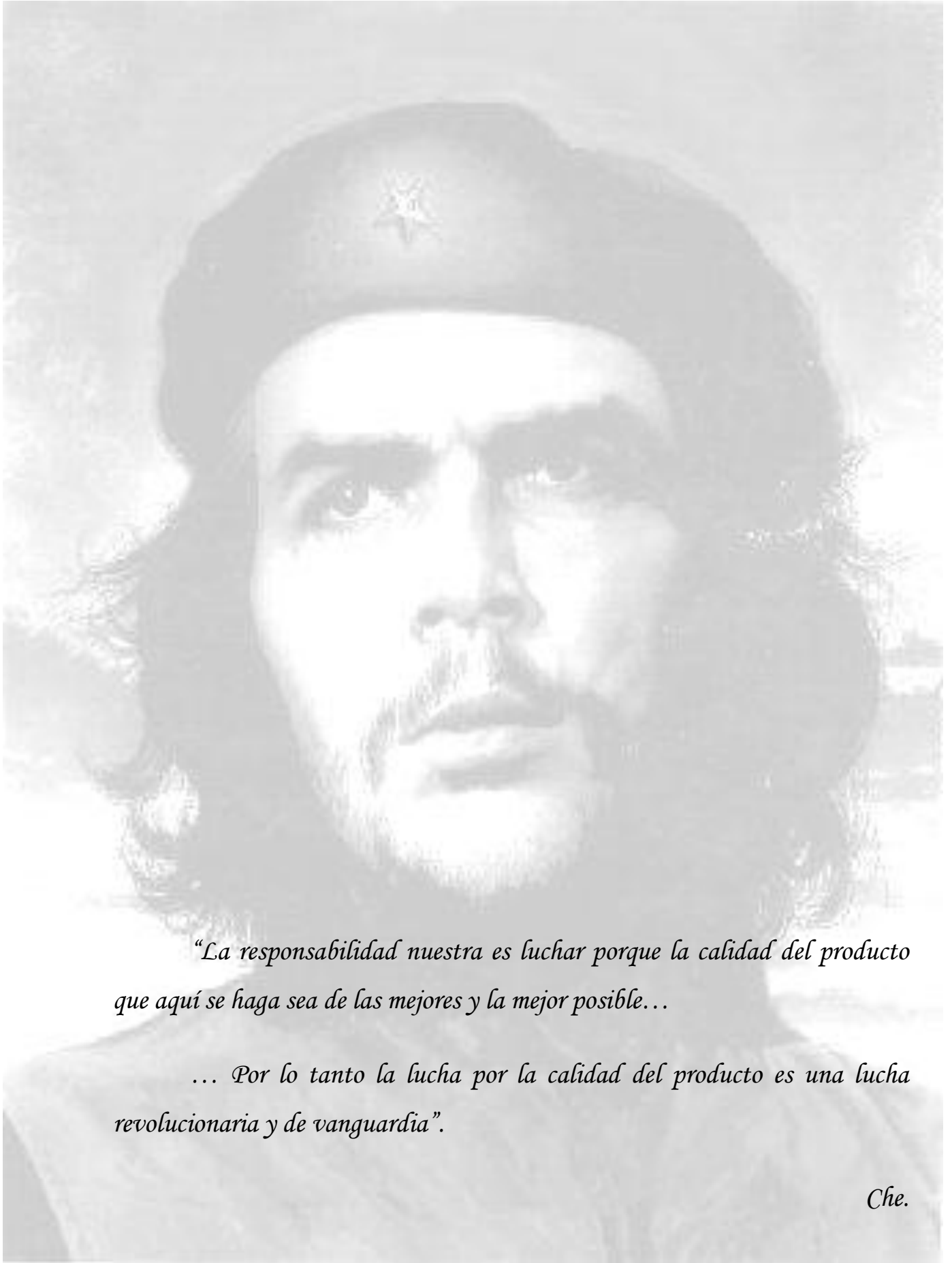
Pedro Jorge Camero Cárdenas

Tutores: Ing. Jandrich Domínguez Fortún

Ing. Gerandys Hernández Casanova

Julio 2008

“Ciudad de La Habana”



“La responsabilidad nuestra es luchar porque la calidad del producto que aquí se haga sea de las mejores y la mejor posible...”

... Por lo tanto la lucha por la calidad del producto es una lucha revolucionaria y de vanguardia”.

Che.

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

Firma del Autor.

Pedro Jorge Camero Cárdenas

Firma del Autor.

Eniel Corzo Rodríguez

Firma del Tutor.

Ing. Gerandys Hernández Casanova

Firma del Tutor.

Ing. Jandrich Domínguez Fortún

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.

En primer lugar quiero agradecer a mi familia, razón de mi existir. Tu mami con tus consejos y complicidad en cada paso. Tu papi por ser ejemplo de hombre y amigo para mí, por ofrecer tu criterio imprescindible para mis acciones diarias. A ti Yami, mi hermanita y mi segunda madre a la vez. A ustedes mis hermanos Jorge y Enrique, por ayudarme a crecer y por los cocotazos que me dieron cuando los necesité. A mis sobrinos todos, Osana, Roxana, Rosangela, Erick, Luis Enrique, Alejandro y Yariel, por ayudarme a crear esta paciencia que hoy tengo. Por soportar mis malcriadeces durante todos estos años, a todos muchas gracias. Parece que ya crecí o al menos estoy en eso

A Arielena, mi novia, mi compañera y mucho más que eso. A sus padres Ariel y Elba. A Jesús, al Jabao, Flor, Hilda, Ana, Frank, Gema, Toni y Juya, por aceptarme. Por compartir conmigo inolvidables momentos.

Un agradecimiento especial a mi compañero de tesis, Eniel, gracias por ayudarme cuando me hizo falta y gracias por confiar en mí para hacer este trabajo. Saludos especiales a sus familiares.

A mis amigos, Yuset, Yorislandy y Humberto por brindarme su mano amiga cuando los necesité. A Terio y al Insu, por ser compañeros de lucha, constante. A mis amigos Pablo, Camilo, Patricio, Manuel, El Tuti, El Yaser, El Chupi, El Boliche, El Ena, Nonó, Robertón, Rolo, Rusly, Luisito, Jose, Luis, Polo, Alfalfa, El Cojo, El Fácil, Junito, Guille, El Yordan, Jorge. A las chicas de mi grupo en estos años, Olquita, Ilirama, Leidi, Lázara, Rendy, Karina, Raiza, Marelis y todas las demás.

A personas muy especiales en estos años de universidad. A las chicas de la peluquería, María Esther, Tania, Mercedes, Chuli, Lili, Yanara, Suney, Yailín, Yoisi y las demás. A las mujeres de la casona, Magela, Dayneris, Lázara, Mileidis, Bamboleo y las demás. A la people de la 4, a las chicas, Ingris, Yaniris, Yasleinis, Rowe, Dainelys y Anita y a los socios, Yordanis, El Cangri, Boris, Carlos y Malagón; a todos muchas gracias por los buenos momentos que pasamos. A Zulema, Alice, Mary, Ania y Andrea, por atenderme. A las tías del comedor, Idania, Martica y las demás. A los que me soportaron en la residencia, Flor, Gilberto, Yudelmis, Juana y los demás. A todas estas personas gracias por ayudarme a llegar hasta aquí y aguantar mis cosas.

A mis profes: Brigida, por enseñarme el inglés que sé y por ser más que profe una madre para conversar y aconsejar, Jandrich, por ser mi profesor, tutor y amigo durante estos años, Gerandys, mi hermano, amigo, profesor y tutor, él sabe que lo quiero y lo estimo mucho, Pepe, por ser amigo y profesor y por enseñarme a dar las clases que impartí en estos años, Aniuska, por confiar en mí como Alumno Ayudante y por enseñarme lo que sé de Contabilidad y Finanzas. A Antero, por guiarme en mis primeros años de adolescente, para ser un hombrecito.

Quiero agradecer de manera especial a la Revolución, por darme la oportunidad de convertirme en un hombre de bien. A Fidel y a Raúl por ser los principales líderes de este proceso tan hermoso con que contamos hoy y del cual nació la UCI, a quien agradezco haberme abrigado, en su seno, durante estos cinco años, y por tener dentro tantas personas lindas e importantes para mí.

Quiero agradecer a tantas personas que todas no caben en una sola cuartilla y si olvido alguna me voy a sentir muy mal, por eso quiero decir que agradezco a todas y cada una de esas personas que me dedicaron un minuto de su tiempo en todos estos años de vida. Me gustaría agradecerles desde el fondo de mi corazón, pero para ustedes, mi gente, mi corazón no tiene fondo. A todos ustedes muchas gracias...

Pedro Jorge

Es difícil resumir en una hoja el agradecimiento que siento por todos los que me hicieron más felices estos largos años de universidad, en este momento se aglomeran los recuerdos en la mente y al final siempre se nos queda un nombre, gracias a todos y en especial:

A mi abuelo Enrique por sus sabios consejos.

A mi mamá por estar siempre a mi lado, por ser una madre maravillosa, por la persona que más quiero en la vida, es la luz que me ilumina y la razón de mi existir. Esta es la realización de uno de sus grandes sueños. A ella le debo la inspiración y los deseos de seguir esta carrera.

A mi padre por su apoyo incondicional, por ser mi amigo en todos los momentos, por guiarme en la vida y por despertar en mí desde pequeño esa pasión que siento por la música y el audio.

A mi tía Sandra por ser mi segunda mamá.

A mi tío Enrique por su apoyo, sus buenos consejos, por ser como un hermano y un ejemplo que intento seguir desde pequeño.

A mi novia Mariela por toda la ayuda, el amor y la comprensión que siempre ha tenido conmigo, dándome aliento en todos los momentos, por dedicarse a tiempo entero a enseñarme a programar.

A Rafael por su amistad.

A mis tutores Jandrich y Gerandys por la confianza que depositaron en mí, por todas las horas que dedicaron de su tiempo a esta investigación.

A mis amigos de siempre Lester, Martínez y Maiquel por toda la ayuda que me brindaron durante estos 5 años de carrera y por ayudarme en la programación.

A todos los de mi grupo por encontrar en ellos los mejores amigos, en especial a esos compañeros inolvidables de noches de estudio, fiestas, laboratorios, reuniones, charlas... en fin, por ser protagonistas de mis mejores recuerdos de la UCI.

En especial a Fidel y a la Revolución por hacerme partícipe de esta obra.

A todos los que me ayudaron en estos largos años de universidad, aunque no los haya mencionado aquí, porque solamente es una hoja de agradecimientos...

Eniel.

Dedicatoria.

A mis abuelitos Lilia, Virginia y Juan Miguel dondequiera que estén.

A mi abuelito Enrique.

A mis padres.

A mi hermanito Ediel para que se inspire y dentro de unos años me dedique su tesis.

A mis primas Liliancis, Lidianis y Lietis.

A mis tíos Sandra y Enrique.

Eniel.

A mis padres ante todo.

A mi hermana Yami y hermanos Enrique y Jorgito.

A mis sobrinos, todos ellos.

A mis socios del barrio y a toda la gente que me quiere.

Pedro Jorge

Resumen.

La calidad del software es un tema que debe ser tratado de manera muy específica en cada proyecto, teniendo en cuenta las características de este, su entorno de desarrollo y los requerimientos o funcionalidades planteadas por el cliente, para lograr esto es necesario conocer a fondo el proceso de desarrollo que se lleva a cabo para obtener el producto y a partir de ahí, trazar una estrategia que asegure y controle la calidad durante todo el proceso de desarrollo.

En el presente trabajo de diploma: “Estrategia de Aseguramiento de la Calidad del Proyecto Juegos CNeuro” se persigue como objetivo principal: diseñar y aplicar una estrategia de Aseguramiento de la Calidad para el proyecto Juegos CNeuro.

Para ello se realizó un análisis de la situación actual del proyecto, el cual arrojó la existencia de problemas críticos con respecto a la calidad del proceso y el producto. Con estos resultados se procedió al diseño de una estrategia que propone actividades de aseguramiento y control de la calidad para todas las etapas del proceso de desarrollo del software y logra corregir las deficiencias detectadas. Dicha estrategia se ha aplicado hasta la etapa de desarrollo actual en que se encuentra el proyecto y los resultados obtenidos se registran en esta investigación.

Palabras Claves:

- ✓ Gestión de la Calidad.
- ✓ Planificación de la Calidad.
- ✓ Aseguramiento de la Calidad.
- ✓ Control de la Calidad.

Índice.

Capítulo 1. Fundamentación Teórica.....	4
1.1 Definición de Calidad.....	4
1.1.1 Calidad a nivel de proceso de Software.....	5
1.1.2 Calidad a nivel de producto.	6
1.1.3 Factores que determinan la calidad.....	7
1.2 Gestión de la Calidad.	9
1.2.1 Planificación de la Calidad.....	10
1.2.2 Aseguramiento de la Calidad.	12
1.2.3 Control de la Calidad.....	14
1.2.4 Mejora de la Calidad.	16
1.3 Metodología Proceso Unificado de Rational.	18
1.3.1 RUP para el Aseguramiento de la Calidad.....	19
1.4 Estándares de Calidad.....	20
1.4.1 Importancia del uso de estándares de calidad.	21
1.4.2 Serie ISO 9000.....	21
1.4.3 Modelo de Madurez y Capacidad Integrado (CMMI).	23
1.5 Enfoques del Aseguramiento de la Calidad.	26
1.5.1 Aseguramiento de la Calidad según ISO 90003.	26
1.5.2 Aseguramiento de la Calidad según CMMI.....	27
1.6 Calidad del Software. Situación Actual.	28
1.6.1 Calidad del Software a nivel Internacional.	28
1.6.2 Calidad del Software en Cuba.	29
1.6.3 Calidad del Software en la UCI.....	30

Capítulo 2. Análisis de la Situación Actual.....	33
2.1 Neuro Ciencias. Antecedentes.....	33
2.2 Discalculia.	34
2.3 Juegos CNeuro. Estructura y Composición.....	35
2.3.1 Introducción.....	35
2.3.2 Roles y responsabilidades definidos.	36
2.3.3 Recursos humanos por roles.....	37
2.3.4 Soporte Tecnológico.	37
2.3.5 Módulos del producto.....	38
2.3.6 Producto generado.	39
2.3.7 Arquitectura.	39
2.3.8 Ciclo de Desarrollo.	41
2.3.9 Ambiente de trabajo.	42
2.4 Situación real de Juegos CNeuro.	42
2.5 Resultados esperados de Juegos CNeuro.....	45
Capítulo 3. Descripción de la Solución.	46
3.1 Estrategia de Aseguramiento de la Calidad para Juegos CNeuro.....	46
3.1.1 Roles y responsabilidades.	47
3.1.2 Plan de Aseguramiento de la Calidad para Juegos CNeuro.	48
Conclusiones.....	66
Recomendaciones.....	67
Bibliografía.	68
Anexos.	70
Anexo1. Factores. Satisfacción del Cliente.	70
Anexo 2. Mejores Prácticas definidas por RUP.....	70

Anexo 3. Fases y Flujos de Trabajo en RUP.	71
Anexo 4. Responsabilidades para el Rol de Diseñador de Pruebas.....	71
Anexo 5. Responsabilidades para el Rol de Revisor Técnico.	72
Anexo 6. Responsabilidades para el Rol de Probador.	72
Anexo 7. Niveles de CMMI.....	73
Anexo 8. Imágenes del Centro de Neurociencias de Cuba.	73
Anexo 9. Roles y responsabilidades de Juegos CNeuro.....	74
Anexo 10. Normas y Estándares utilizados.....	75
Anexo 11. Lista de Chequeo. Revisión Inicial.	76
Anexo 12. No Conformidades. Revisión Inicial.	77
Anexo 13. Lista de Chequeo. Revisión de los Requisitos.	78
Anexo 14. Lista de Chequeo. Diagrama de Casos de Uso.	79
Anexo 15. No Conformidades. Revisión al Diagrama de Casos de Uso.	80
Anexo 16. Lista de Chequeo. Revisión de la Arquitectura.....	81
Anexo 17. No Conformidades. Revisión de la Arquitectura.....	83
Anexo 18. Lista de Chequeo. Revisión del Diseño.....	85
Anexo 19. No Conformidades. Revisión del Diseño.....	86
Anexo 20. Lista de Chequeo. Revisión a la Gestión de la Configuración.	87
Anexo 21. No Conformidades. Revisión a la Gestión de la Configuración.	88

Anexo 22. CPR. ActivateTask().	89
Anexo 23. CPR. Gestionar Perfil.....	89
Anexo 24. CPR. Seleccionar Jugador.....	91
Anexo 25. CPR. Manipular Sonido.....	92
Anexo 26. CPR. Cargar Nivel.	93
Glosario de Términos.....	94

Introducción.

La Industria del Software en el mundo crece y se desarrolla a un ritmo vertiginoso, aunque la producción sigue siendo baja, los costos y la duración de los procesos son superiores a los estimados y la satisfacción de las expectativas de los clientes todavía no alcanza los niveles de aceptación requeridos. Esta situación se debe en la mayoría de los casos a la Calidad del Software, debido a la mala o no aplicación de las normas y estándares de calidad establecidos para desarrollar productos de esta naturaleza a nivel mundial. Estos son problemas que afectan de manera general el Proceso de Desarrollo del Software en sus diferentes fases.

En nuestro país, a pesar de contar con una economía subdesarrollada, desde hace varios años, la máxima dirección del gobierno decidió fomentar el desarrollo, consolidación y expansión de la industria del Software, debido al amplio espectro económico que posee este mercado. A raíz de esta decisión tiene lugar en el año 2002 el surgimiento de una universidad dedicada al estudio, investigación y producción de Software, la Universidad de las Ciencias Informáticas (UCI), donde desde entonces se preparan y forman profesionales altamente calificados en la tarea de producir Software de alta calidad, para, de esta manera, poder incluir al país en el mercado mundial y al mismo tiempo potenciar el proceso de desarrollo e informatización de la sociedad cubana.

En los pocos años que lleva la Universidad de las Ciencias Informáticas de creada se ha logrado formar y consolidar una estructura productiva de manera acelerada y obligada por las grandes potencialidades tecnológicas que posee la universidad, en la cual, actualmente, existen proyectos productivos capaces de desarrollar productos de Software muy buenos y que representan en muchos casos fuentes de ingresos económicos para el país.

El Aseguramiento de la Calidad del Software, es uno de los tópicos de investigación más importantes a nivel global, dentro de la Ingeniería de Software. En la Universidad de las Ciencias Informáticas, es de vital importancia el dominio, por parte de sus estudiantes y profesionales, de los temas referentes a este proceso. Esto determina la adopción de métricas, normas y estándares internacionales de Ingeniería de Software.

Este tema aún se encuentra en fase de estudio en la Universidad por parte de la Dirección de Calidad de Software, en busca de un proceso que permita que los productos de Software cumplan con los requisitos que exigen los clientes y que además tenga la calidad y funcionalidades requeridas.

En la Facultad 5 de la UCI se desarrollan productos de Realidad Virtual y Automatización. El Proyecto Juegos CNeuro forma parte de los proyectos productivos en este contexto, el mismo surge debido a la necesidad de desarrollar un software para la intervención de la Discalculia en los niños, o sea para ayudar a rehabilitarlos. El Proyecto Juegos CNeuro cuenta con personal preparado para el diseño del Software pero existen limitaciones con el Aseguramiento de la Calidad del Proceso y Producto a desarrollar.

Como consecuencia de estas limitaciones a los autores de este trabajo de diploma se les asignó la tarea de desempeñarse en el rol de Administrador de Calidad. Esta tarea es asumida como principal motivación para la realización de esta tesis de diploma. A partir de los antecedentes citados surge el siguiente **problema científico**:

¿Cómo garantizar que los procesos y productos del trabajo del proyecto Juegos CNeuro, de la Facultad 5 de la UCI, se comporten según lo planificado?

Una vez planteado este problema científico se determina que el **objeto de estudio** de este trabajo es el Proceso de Producción de Software en el Proyecto Juegos CNeuro.

Se define como **campo de acción** el Proceso de Aseguramiento de la Calidad en el Proyecto Juegos CNeuro.

Para esto se plantea el siguiente **objetivo general**:

✓ Diseñar y aplicar una estrategia de Aseguramiento de la Calidad para el proyecto Juegos CNeuro.

Y se establecen como **objetivos específicos**:

✓ Profundizar a partir de los antecedentes a nivel nacional, con énfasis en la UCI, el tema de Aseguramiento de la Calidad basado en el área de proceso Aseguramiento de la Calidad del Producto y el Proceso, del Modelo de Madurez de Capacidad Integrado (CMMi).

✓ Analizar los diferentes estándares de calidad de software existentes en el contexto nacional e internacional.

Como **idea a defender** se plantea que con el diseño y puesta en práctica de una estrategia de Aseguramiento de la Calidad en el Proyecto Juegos CNeuro se logrará mayor eficiencia en el desarrollo del proceso y el producto.

Con el diseño y ejecución de las investigaciones que forman parte de esta tesis se pretende aportar:

✓ Una estrategia en función del proceso de Aseguramiento de la Calidad como proceso y producto basado en la métrica Modelo de Madurez de Capacidad Integrado y que asegure la calidad del proceso y del producto en este proyecto productivo.

Para el desarrollo de esta investigación se emplearán algunos métodos científicos de la investigación, que guiarán el estudio de cada tema y posibilitarán arribar fácilmente a la solución esperada. Por ejemplo, para la confección de la Fundamentación Teórica se utilizará el método **Analítico-sintético** y el **Histórico-lógico**, el primero permitirá identificar de todo el monto de información que aborda el tema estudiado, los elementos particulares para esta investigación y el Método Histórico-lógico ayudará a comprender el desarrollo, la historia y la lógica del fenómeno analizado.

Por otra parte, en lo concerniente a la elaboración de la solución que será propuesta en esta tesis de diploma, se empleará el método **Inductivo-deductivo** que permitirá a partir del estudio de hechos aislados arribar a proposiciones generales e inferir casos particulares a través un razonamiento lógico.

Capítulo 1. Fundamentación Teórica.

Introducción.

El tema de la calidad del software es uno de los eslabones, dentro de todo el proceso de desarrollo del software, que en ocasiones y no para bien se descuida un poco, por parte del equipo de desarrolladores; siendo, en muchos casos, razón para inconformidades por parte de los clientes finales. Para evitar este tipo de situaciones en todos los casos desagradables e indeseables, para toda institución que desarrolle productos de software, se hace necesario el conocimiento de los pasos y procesos que permiten el aseguramiento de la calidad y al final el logro de un producto de calidad que satisfaga las necesidades de los usuarios.

En este capítulo se exponen conceptos y definiciones que fundamentan teóricamente el presente trabajo de diploma, divididos en diferentes epígrafes y sub-epígrafes. En los primeros dos epígrafes se hace una definición del término “calidad” mediante un análisis de lo que plantean diferentes especialistas así como lo que opinan los autores de este trabajo y se analizan diferentes procesos que tienen lugar dentro de la calidad del software. En el tercer y cuarto epígrafe se analiza lo que plantean la metodología del Proceso Unificado de Rational y los estándares de calidad existentes respectivamente; prestando especial atención al área de proceso Aseguramiento de la calidad dentro de Modelo de Madurez de Capacidad Integrado, cuyo análisis y puesta en práctica es razón fundamental de estudio de esta tesis.

Además en otro de los epígrafes se analizan diferentes enfoques sobre el Aseguramiento de la calidad, planteados por la Organización Internacional de Normalización (ISO) y por el Modelo de Madurez de Capacidad Integrado. Por último se hace un estudio de los antecedentes del tema en Cuba y en la UCI.

1.1 Definición de Calidad.

El significado histórico de la palabra calidad es el de aptitud o adecuación al uso. 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 producto o servicio confieren al cliente.

- ✓ Fiabilidad o capacidad que tiene el producto o servicio para cumplir las funciones especificadas, sin fallo y por un período determinado de tiempo.
- ✓ Servicio o medida en que el fabricante y distribuidor responden en caso de fallo del producto o servicio.

La Sociedad Americana para el Control de 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. (1)

La Calidad del Software es “la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente. (2)

Calidad es el conjunto de características de una entidad, que le confieren la aptitud para satisfacer las necesidades establecidas y las implícitas. (3)

Calidad es el grado en el que un conjunto de características inherentes cumple con los requisitos. (4)

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

- ✓ Adecuación (del producto) al uso. (5)
- ✓ Conformidad con requisitos y confiabilidad en el funcionamiento. (6)
- ✓ Cero defectos. (7)

Los autores de este trabajo de diploma consideran como calidad, la capacidad que adquiere dicho producto elaborado, de satisfacer las necesidades y exigencias del cliente.

1.1.1 Calidad a nivel de proceso de Software.

El Instituto de Ingeniería de Software (SEI) en su modelo CMMi define la calidad como:

- ✓ El grado en el cual un sistema, componente o proceso cumple con los requisitos especificados.
- ✓ El grado en el cual el sistema, componente o proceso cumple con las expectativas del cliente o usuario. (8)

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

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 y aseguramiento durante todas las etapas del ciclo de vida del software.

Para alcanzar la "Calidad", 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. (9)

Para realizar el control de la calidad de un sistema, no basta solamente con lograr la calidad al sistema una vez elaborado, sino que es de vital importancia que cada uno de los procesos que forman parte del desarrollo de dicho sistema, sean llevados a cabo con calidad; es decir que la calidad de un sistema dependerá de la calidad del proceso de desarrollo, puesto que si posteriormente son detectados errores o fallas en él esto quiere decir que el proceso mediante el cual se desarrollo no tuvo calidad. La Calidad del Software se diseña conjuntamente con el sistema, nunca al final. (9)

Los autores consideran que este es un tema de suma importancia puesto que un producto desarrollado correctamente y con calidad significa que en el futuro habrá menos gastos de mantenimiento, lo que proporciona satisfacción tanto para la empresa que desarrolla el producto como para el cliente. Por tanto el tema de la calidad del producto final debe convertirse en un elemento fundamental dentro del proceso de desarrollo de un software.

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

1.1.2 Calidad a nivel de 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. (10)

La calidad de un producto software debe evaluarse usando un modelo de calidad que tiene en cuenta criterios para satisfacer las necesidades de los desarrolladores, mantenedores, adquiridores y usuarios finales. (11)

Los autores de esta tesis opinan 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 lo 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 requisitos y las especificaciones funcionales. (12)

1.1.3 Factores que determinan la calidad.

Se centran en tres aspectos importantes de un producto software:

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

Características operativas.

- ✓ Corrección. ¿Hace lo que quiero?
- ✓ Fiabilidad. ¿Lo hace de forma fiable todo el tiempo?
- ✓ Eficiencia. ¿Se ejecutará en mi hardware lo mejor que pueda?
- ✓ Seguridad (Integridad). ¿Es seguro?
- ✓ Facilidad de uso. ¿Está diseñado para ser usado?

Capacidad de soportar los cambios.

- ✓ Facilidad de mantenimiento. ¿Puedo corregirlo?
- ✓ Flexibilidad. ¿Puedo cambiarlo?
- ✓ Facilidad de prueba. ¿Puedo probarlo?

Adaptabilidad a nuevos entornos.

- ✓ Portabilidad. ¿Podré usarlo en otra máquina?
- ✓ Reusabilidad. ¿Podré reutilizar alguna parte del software?
- ✓ Interoperabilidad. ¿Podré hacerlo interactuar con otro sistema? (9)

Se definen también tres factores fundamentales que influyen directamente en la satisfacción del cliente y como la calidad y la satisfacción del cliente guardan una estrecha relación, los autores de este trabajo de diploma consideran importante reflejarlos en el [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. (1)

Los autores opinan que 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 dicho proceso, ya que un descuido o un mal trabajo de algún factor (miembro del equipo) puede comprometer, de manera considerable, la calidad final del producto en cuestión. Por ello estar involucrados en el proceso, formar parte de cada 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.

1.2 Gestión de la Calidad.

La Gestión de la Calidad de Software es un conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades. Se basa en la determinación y aplicación de las políticas de calidad de la empresa. (9)

Se puede definir la Gestión de la Calidad del Software como:

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

La Gestión de la Calidad de Software es una actividad esencial en cualquier empresa de software para asegurar la calidad de sus productos, y la competitividad frente a la oferta del mercado. Es un conjunto de actividades de la función general de la Dirección que determina la calidad, los objetivos y las responsabilidades. (13)

Los procesos de Gestión de la Calidad del Proyecto incluyen todas las actividades de la organización ejecutante que determinan las políticas, los objetivos y las responsabilidades relativos a la calidad de modo que el proyecto satisfaga las necesidades por las cuales se emprendió. Implementa el sistema de gestión de calidad a través de la política, los procedimientos y los procesos de planificación de calidad, aseguramiento de calidad y control de calidad, con actividades de mejora continua de los procesos que se realizan durante todo el proyecto, según corresponda. (14)

Existen ocho principios de gestión de la calidad que constituyen la base de las normas de sistemas de gestión de la calidad de la familia ISO 9000, ellos son:

- ✓ **Organización enfocada al cliente:** Las organizaciones dependen de sus clientes y por lo tanto deberían comprender las necesidades actuales y futuras de los clientes, satisfacer los requisitos de los clientes y esforzarse en exceder las expectativas de los clientes.
- ✓ **Liderazgo:** Los líderes establecen unidad de propósito y dirección de la organización. Ellos deberían crear y mantener un ambiente interno, en el cual el personal pueda llegar a involucrarse totalmente en el logro de los objetivos de la organización.

- ✓ **Participación del personal:** El personal, con independencia del nivel de la organización en el que se encuentre, es la esencia de una organización y su total implicación posibilita que sus habilidades sean usadas para el beneficio de la organización.
- ✓ **Enfoque basado en procesos:** Los resultados deseados se alcanzan más eficientemente cuando los recursos y las actividades relacionadas se gestionan como un proceso.
- ✓ **Gestión basada en sistemas:** Identificar, entender y gestionar un sistema de procesos interrelacionados para un objetivo dado, mejora la eficacia y eficiencia de una organización.
- ✓ **Mejora continua:** La mejora continua debería ser un objetivo permanente de la organización.
- ✓ **Toma de decisiones basada en hechos:** Las decisiones eficaces se basan en el análisis de los datos y la información.
- ✓ **Relación mutuamente beneficiosa con el proveedor:** Una organización y sus proveedores son interdependientes, y una relación mutuamente beneficiosa intensifica la capacidad de ambos para crear valor. (13)

Los autores de esta tesis opinan que la gestión de la calidad es una actividad que no debe, ni puede ser descuidada, por parte de la administración de cada una de las instituciones que se dedican a la producción de productos de software. Por otra parte esto contribuye a organizar y orientar el trabajo durante el proceso de producción del software. Es un aspecto que proporciona, además, a las empresas productoras de software estar al día en cuanto a lo que el mercado exige en materia de calidad para productos de software.

1.2.1 Planificación de la Calidad.

La planificación de la calidad es la parte de la gestión de la calidad enfocada al establecimiento de los objetivos de la calidad y a la especificación de los procesos operativos necesarios y de los recursos relacionados para cumplir los objetivos de calidad. (13)

La planificación de calidad implica identificar qué normas de calidad son relevantes para el proyecto y determinar cómo satisfacerlas. Es uno de los procesos clave a la hora de llevar a cabo el Grupo de Procesos de Planificación y durante el desarrollo del plan de gestión del proyecto, y debería realizarse de forma paralela a los demás procesos de planificación del proyecto. (14)

La Planificación de la Calidad del Software es la parte de la Gestión de la Calidad encargada de realizar el proceso administrativo de desarrollar y mantener una relación entre los objetivos y recursos de la organización; y las oportunidades cambiantes del mercado. (15)

El objetivo es modelar y remodelar los negocios y productos de la empresa, de manera que se combinen para producir un desarrollo y utilidades satisfactorias. Los aspectos a considerar en la Planificación de la Calidad de Software son: Modelos/Estándares de Calidad de Software a utilizar, Costos de la Calidad de Software, Recursos humanos y materiales necesarios, etc. Los factores que determinan el Modelo o Estándar de Calidad de Software a elegir son:

1. La complejidad del proceso de diseño.
2. La madurez del diseño.
3. La complejidad del proceso de producción.
4. Las características del producto o servicio.
5. La seguridad del producto o servicio.
6. Económico. (9)

“La planificación de la calidad facilita el modo de adaptar la planificación del sistema de gestión de la calidad a un proyecto específico, producto o contrato. La planificación de la calidad puede incluir referencias genéricas y/o proyecto / producto / contrato específico de procedimientos, como apropiados. La planificación de la calidad debería ser revisada de nuevo junto con el progreso del diseño y desarrollo, y los elementos, en cada fase, deberían ser completamente definidos al comienzo de dicha fase”. (15)

El plan de calidad de un proyecto se hace con el propósito de proporcionar una adecuada seguridad de que el producto que se obtendrá y los procesos del ciclo de vida del proyecto estén conformes con los requisitos específicos y se ajusten a los planes establecidos. En este se definen los atributos de calidad más importantes del producto a ser desarrollado y define el proceso de evaluación de la calidad.

Un plan de calidad puede tener la siguiente estructura:

- ✓ **Introducción al Producto:** Una descripción del producto, su objetivo en el mercado y expectativas de calidad del producto.

- ✓ **Planes del producto:** Fechas críticas respecto de la liberación del producto y responsabilidades del producto respecto de su distribución y servicio.
- ✓ **Descripciones del proceso:** Procesos de desarrollo y servicios que serían usados en el desarrollo y en la administración.
- ✓ **Objetivos de Calidad:** Objetivos y planes de calidad del producto, los cuales incluyen la identificación de los atributos de calidad del producto.
- ✓ **Manejo del riesgo:** Principales riesgos que pueden afectar la calidad del producto. (16)

Toda esa información es presentada mediante diferentes documentos o planillas las que conforman el Plan de Calidad.

1.2.2 Aseguramiento de la Calidad.

El Aseguramiento de la calidad, es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza que el software satisfará los requisitos dados de calidad. (2)

Aseguramiento de la Calidad del Software consiste en la Revisión de los productos y su documentación relacionada, para verificar su cobertura, corrección, confiabilidad y facilidad de mantenimiento. (17)

Pressman recomienda una serie de actividades necesarias para lograr un correcto Aseguramiento de la calidad, que deben ser diseñadas antes de que comience el desarrollo del proyecto:

- ✓ **El establecimiento del plan de calidad del proyecto:** Se realiza en las primeras etapas del proyecto y es un documento que planifica y rige todas las actividades de aseguramiento de la calidad así como la forma de aplicación en el proyecto.
- ✓ **La participación en el desarrollo de la descripción del proceso de software del proyecto:** Es tarea del equipo de aseguramiento de la calidad la revisión del proceso que se lleva a cabo en el proyecto vigilando que se ajuste a las políticas y los estándares internos del software.
- ✓ **La revisión de las actividades de ingeniería del software para verificar su ajuste al proceso de software definido:** El grupo de aseguramiento de la calidad identifica, documenta y sigue la pista de las desviaciones desde el proceso y verifica que se han hecho las correcciones.
- ✓ **Auditoría de los productos de software designados para verificar el ajuste con los definidos como parte del proceso del software:** El grupo de aseguramiento de la calidad revisa los productos

seleccionados; identifica, documenta y sigue la pista de las desviaciones; verifica que se han hecho las correcciones, e informa periódicamente de los resultados de su trabajo al gestor del proyecto.

✓ **Asegurar la documentación de los productos de software:** Documentar debidamente toda actividad que se realice en el proyecto es una práctica importante que se debe seguir para obtener un producto de calidad. El equipo de aseguramiento de la calidad esta encargado de desarrollar una estrategia para la revisión de la documentación que se genera.

✓ **Registrar los desajustes en concordancia con los requisitos:** Consiste en darle seguimiento a estos errores hasta que sean resueltos. (2)

Además tener en cuenta que el Aseguramiento de la Calidad del Software está presente en:

- ✓ Métodos y herramientas de análisis, diseño, programación y prueba.
- ✓ Inspecciones técnicas formales en todos los pasos del proceso de desarrollo del software
- ✓ Estrategias de prueba multiescala.
- ✓ Control de la documentación del software y de los cambios realizados.
- ✓ Procedimientos para ajustarse a los estándares (y dejar claro cuando se está fuera de ellos).
- ✓ Mecanismos de medida (métricas).
- ✓ Registro de auditorias y realización de informes. (2)

Las Revisiones Técnicas Formales (RTF) conocidas también como Inspecciones son una de las actividades más efectivas desde el punto de vista del Aseguramiento de la Calidad y es un medio efectivo para mejorar la calidad del software. (9)

Su objetivo principal es encontrar errores durante el proceso, de forma que se conviertan en defectos después de la entrega del software. El beneficio de estas revisiones técnicas formales es el descubrimiento de errores al principio para que no se propaguen al paso siguiente del proceso de software, por lo tanto es recomendable aplicarlas desde el comienzo de la vida del Software.

Las actividades de diseño introducen entre el 50 y 65% de todos los errores durante el proceso de software. Sin embargo, se ha demostrado que las revisiones técnicas formales son efectivas en un 75% a la hora de detectar errores. (9)

Los autores de este trabajo consideran que las revisiones técnicas formales, representan un filtro para el proceso de desarrollo del software, se deben aplicar en diferentes momentos de la vida del software ya que sirven para detectar defectos, que pueden ser eliminados de manera sistemática y no antes de

entregar un producto, ya que esto se traduce en gastos muy grandes y además innecesarios para la entidad productora del software. Por tanto tienen un alto índice de precisión a la hora de comprobar un software en su ciclo de desarrollo. Mediante el uso de revisiones técnicas formales se contribuye de manera considerable al aseguramiento de la Calidad del producto final.

1.2.3 Control de la Calidad.

El control de calidad es una serie de inspecciones a lo largo del proceso del software para asegurar que cada producto cumple con los requisitos que le han sido asignados. (2)

El control de la calidad es la parte de la gestión de la calidad orientada al cumplimiento de los requisitos de la calidad. (13)

El control de calidad incluye un ciclo de realimentación (feedback) del proceso que creó el producto. Las actividades de control de la calidad pueden ser manuales, completamente automáticas o la combinación de herramientas automáticas e interacción humana. El bucle de retroalimentación es imprescindible para reducir los defectos producidos. (9)

Realizando el Control de la Calidad se supervisan los resultados específicos del proyecto, para determinar si cumplen con las normas de calidad relevantes y se identifican modos de eliminar las causas de un rendimiento insatisfactorio. (14)

El control de la calidad hoy en día es de vital importancia en una empresa para lograr un producto con los requisitos de calidad. Las actividades principales del control de la calidad lo constituyen las pruebas.

Veamos lo que opinan algunos autores:

Las pruebas permiten al desarrollador determinar si el producto generado satisface las especificaciones establecidas. Así mismo, una prueba de software permite detectar la presencia de errores que pudieran generar salidas o comportamientos inapropiados durante su ejecución. (18)

La prueba de software es un elemento crítico e imprescindible para la garantía de la calidad y de ahí la necesidad de aplicarla.

La prueba no es:

- ✓ Demostración que no hay errores.
- ✓ Demostración que el software desempeña correctamente sus funciones.
- ✓ Establecimiento de confianza que un programa hace lo que debe hacer. (9)

En Juegos CNeuro se desarrolla el proceso y el producto siguiendo la Metodología del Proceso Unificado de Rational (RUP). La misma propone una serie de pruebas durante todo el proceso de desarrollo del software. A continuación veremos el proceso, de aplicación y evaluación de las pruebas, planteado por esta metodología:

Aplicación de Pruebas.

- ✓ **Prueba de Unidad:** Tiene como propósito probar los componentes implementados como unidades individuales.
- ✓ **Pruebas de Integración:** Se utilizan para verificar que los componentes interaccionan entre sí de la forma apropiada después de haber sido integrados en una construcción.
 - **Incrementales (ascendente y descendente):** Se combina un módulo con el siguiente módulo que se debe probar o con el conjunto de módulos que ya están probados.
- ✓ **Pruebas de sistema:** Se llevan a cabo principalmente para verificar la interacción entre los actores y el sistema.

Las técnicas para la aplicación de cada una de estas pruebas son las siguientes:

- ✓ **La prueba de especificación, o "prueba de caja negra":** Verifica el comportamiento de la unidad observable externamente.
- ✓ **La prueba de estructura, o "prueba de caja blanca":** Verifica la implementación interna de la unidad.

Evaluación de las pruebas.

El propósito de evaluar las pruebas es evaluar los esfuerzos de prueba de una iteración.

El Administrador de Calidad evalúa los resultados de las pruebas, comparando los resultados obtenidos con los objetivos esbozados en el Plan de Pruebas establecido. Si una prueba no alcanza sus objetivos, los casos y procedimientos de prueba, deben ser redefinidos para lograrlos.

Resumen de las Pruebas:

Se debe tener como principal resultado de la prueba el modelo de pruebas que contiene los elementos que fueron probados y sus principales resultados. (19)

Los autores de esta tesis opinan que todo proceso de desarrollo de un software está sujeto a equivocaciones y errores por parte de sus desarrolladores, a partir de que siempre en estos procesos hay constantes cambios y esto conlleva en muchos de los casos a equivocaciones humanas. Para contrarrestar esto se hace necesario controlar lo que se hace desde que se comienza a desarrollar un producto y para ello son utilizadas las pruebas para combatir estos errores y lograr calidad en el proceso y el producto que se elabora.

A modo de resumen los autores creen que control de la calidad no es más que el proceso constante de verificación de los procesos que dan lugar a un producto final. Desde el inicio de la producción del Software se debe tener una seguridad de que los procesos están llevándose a cabo de la manera adecuada según la política de calidad que se ha trazado en el proyecto.

1.2.4 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. (9)

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. (4)

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.

Acciones correctivas

La organización debe tomar acciones para eliminar la causa de no conformidades con objeto de prevenir su repetición. Las acciones correctivas deben ser apropiadas a los efectos de las no conformidades encontradas.

Debe establecerse un procedimiento documentado para definir los requisitos para:

- ✓ Revisar las no conformidades (incluyendo las quejas de los clientes).
- ✓ Determinar las causas de las no conformidades.
- ✓ Evaluar la necesidad de adoptar acciones para asegurar que las no conformidades no vuelven a ocurrir.
- ✓ Determinar e implementar las acciones necesarias.
- ✓ Registrar los resultados de las acciones tomadas.
- ✓ Revisar las acciones correctivas tomadas.

Acciones preventivas

La organización debe determinar acciones para eliminar las causas de no conformidades potenciales para prevenir su ocurrencia. Las acciones preventivas tomadas deben ser apropiadas a los efectos de los problemas potenciales.

Debe establecerse un procedimiento documentado para definir los requisitos para:

- ✓ Determinar no conformidades potenciales y sus causas.
- ✓ Evaluar la necesidad de actuar para prevenir la ocurrencia de no conformidades.
- ✓ Determinar e implementar las acciones necesarias.
- ✓ Registrar los resultados de las acciones tomadas.
- ✓ Revisar las acciones preventivas tomadas. (20)

Los autores opinan que la mejora de la calidad la constituyen todas las acciones continuas para corregir las deficiencias encontradas durante el desarrollo del producto, para de esta manera lograr capacidad en el producto de satisfacer las exigencias y funcionalidades definidas.

Muchas de las entidades no mantienen un hábito de constancia en la mejora de sus productos y servicios, lo cual trae muchas deficiencias en cada uno de sus procesos. Lo ideal es que se planteen una buena práctica de constancia de mejora, para de esta forma poder competir con las demás empresas y sobre todo permanecer en el mercado, ya que muchas de las organizaciones no duran mucho por que no son constantes en la mejora de sus procesos. Por tal motivo deben mejorar constantemente en los procesos de planeación, producción y servicio. Para de esa manera poder reducir los costos.

1.3 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. (21)

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) (22)

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

1. Dirigido por Casos de Uso.
2. Centrado en la arquitectura.
3. 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 Riesgo, Control de la Calidad, Gestión de Proyectos y Gestión de configuración. (22)

1.3.1 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. (Ver [Anexo 2](#)). (23)

RUP provee de planillas para la documentación durante todo el proceso de desarrollo del software, se divide en 4 fases y describe varios flujos de trabajo de desarrollo y de soporte (Ver [Anexo 3](#)). 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. (24)

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: (25)

- ✓ Diseñador de Pruebas. (Ver [Anexo 4](#)).
- ✓ Revisor Técnico. (Ver [Anexo 5](#)).
- ✓ Probador. (Ver [Anexo 6](#)).

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.

Los autores consideran necesarios estos roles, planteados por la Metodología, como parte del equipo de calidad de Juegos CNeuro. Además de esto consideran necesario definir el rol de Administrador de Calidad, propuesto por el Departamento Central de Calidad de la UCI. Para esto como parte de la estrategia elaborada, en el Capítulo 3 de este trabajo de diploma, se fundamenta como quedarían los roles del equipo de calidad dentro del proyecto, para lograr asegurar la calidad durante todo el proceso de producción.

1.4 Estándares de Calidad.

Los Modelos de Calidad son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos claves y permiten medir los avances en calidad. (26)

Nota: Dígase “procesos claves”, el Proceso de Aseguramiento de la Calidad.

1.4.1 Importancia del uso de estándares de calidad.

Las organizaciones desarrolladoras de software han estudiado y definido un conjunto de modelos y estándares que han marcado el camino para la producción de software de calidad. Hoy en día, se hace necesario certificar los productos de acuerdo a alguno de los estándares existentes para poder competir en el mercado mundial. Por otra parte, el empleo de estos modelos, guía el trabajo de los desarrolladores hacia las mejores prácticas de desarrollo, evitando repetir errores anteriores y capturando el conocimiento de valor para la organización. También definen un marco de trabajo alrededor del que se implementa el proceso de Aseguramiento y Gestión de la Calidad. Facilitan continuidad del producto, pues nuevos integrantes del equipo pueden entender la organización una vez que logre comprender los estándares que utiliza la misma.

En los últimos años los estándares de calidad internacionales se han reorientado hacia el Aseguramiento y Gestión de la Calidad. Esto ha dado paso a la aparición de importantes modelos de referencia, que tienen en común la evaluación de la capacidad de los procesos en niveles de desarrollo o madurez en la industria del Software. Al compás de esto, las Normas de Calidad se han ido perfeccionado con el objetivo de estandarizar los sistemas de calidad de las diferentes empresas y sectores que desarrollan productos de Software.

Los autores de esta Investigación consideran que el uso adecuado de los Modelos y Estándares de Calidad ayuda a obtener una mejor Gestión de la Calidad, ayuda a crear un lenguaje común entre las diferentes organizaciones productoras de software a nivel mundial. Esto permite limar asperezas en el proceso de producción de software ya que al existir un estándar de comunicación se hace mucho más ágil la labor para el equipo de desarrollo.

1.4.2 Serie ISO 9000.

Con el objetivo de estandarizar los sistemas de calidad de las diferentes empresas y sectores, se publican las normas ISO 9000, que son un conjunto de normas editadas y revisadas periódicamente por la Organización Internacional de Normalización, sobre la garantía de calidad de los procesos.

Las normas de la familia ISO 9000 se han elaborado para asistir a las organizaciones, de todo tipo y tamaño, en la implementación y la operación de sistemas de la calidad eficaces. (13)

Todas estas normas juntas forman un conjunto coherente de normas de sistemas de gestión de la calidad que facilitan la mutua comprensión en el comercio nacional e internacional.

Este trabajo de diploma centrará su estudio principalmente en la Norma ISO 9000-3:2004: Sistemas de Calidad. Modelo de Aseguramiento de la Calidad para la inspección final y pruebas. A continuación veremos detalles de esta Norma de Calidad.

✓ **Generalidades**

- **Título.**

Normas de gestión de la calidad y garantía de la calidad. Parte 3: Orientaciones para la aplicación de la Norma ISO 9001 al desarrollo, suministro y mantenimiento del software.

- **Naturaleza.**

Norma internacional.

- **Campo de aplicación y alcance.**

Esta parte de la ISO 9000 contiene orientaciones que facilitan la aplicación de la Norma ISO 9001 a las organizaciones dedicadas al desarrollo, suministro y mantenimiento del software.

Se pretende con ella dar orientaciones en relación con situaciones en las que un contrato entre dos partes exija la demostración de la capacidad de determinado proveedor para desarrollar, suministrar y mantener productos de software.

Tales orientaciones describen las clases de control y los métodos sugeridos para la producción del software, que satisfagan los requisitos establecidos. Esto será posible principalmente a través de la prevención de "no-conforme" a lo largo de todas las fases del proceso, desde el desarrollo hasta el mantenimiento.

- **Detalles.**

El **objetivo** de la ISO 9000-3 es proveer las especificaciones de cómo aplicar la ISO 9001 al desarrollo del software, implementación y mantenimiento. Se incluyen algunos temas que no se encuentran en las normas ISO 9000 genéricas, tales como Administración de la Configuración o Planeación de Proyectos. Sería poco probable lograr resultados de calidad en un proyecto de desarrollo software de

tamaño mediano, sin haber tomado las provisiones necesarias para el control de configuración. Esto implica que para ciertos productos o servicios, la especificación de requerimientos contenida en las normas genéricas ISO 9000 no es suficiente para asegurar la calidad, y esto justifica la necesidad de otras normas o guías más específicas.

La norma ISO 9000-3 es requerida por todas las compañías desarrolladoras de software:

- ✓ Para poder incursionar en la competencia del mercado europeo.
- ✓ Como un medio para cubrir las expectativas de los clientes.
- ✓ Para obtener beneficios de calidad y ventajas competitivas en el mercado.
- ✓ Como parte de la estrategia del mercado.
- ✓ Estrategia para reducir los costos de producción.

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

- ✓ Mejor documentación de los sistemas.
- ✓ Cambio cultural positivo.
- ✓ Incremento en la eficiencia y productividad.
- ✓ Mayor percepción de calidad.
- ✓ Se amplía la satisfacción del cliente.
- ✓ Se reducen las auditorías de calidad de los clientes.
- ✓ Agiliza el tiempo de desarrollo de un sistema. (15)

1.4.3 Modelo de Madurez y Capacidad Integrado (CMMI).

El Modelo de Madurez y Capacidad Integrado, es un modelo que mide la madurez organizacional o la capacidad de ejecutar procesos de una entidad de software, permite implantar los procesos de software que recogen las mejores prácticas, incluye buenas prácticas reconocidas, referencias para fijar objetivos, referencias para fijar prioridades y estándares que son resultado de la experiencia de la industria.

CMMI está caracterizado por áreas de proceso para las 4 disciplinas que cubre actualmente: Ingeniería de Sistemas, Ingeniería del Software, Desarrollo Integrado del Producto y del Proceso y la Fuente proveedora. Las áreas de proceso son un conjunto de actividades agrupadas para facilitar el camino de la mejora, establecen la capacidad de proceso de la organización y se pueden agrupar en 4

categorías generales: (1) Gestión de Proyectos, (2) Gestión de Procesos, (3) Ingeniería y (4) Apoyo; y dos categorías opcionales: (1) Desarrollo Integrado del Producto y del Servicio; y (2) Gestión de Compras.

CMMI enseña el camino para alcanzar un nivel de madurez de la organización o un nivel de capacidad de un área de proceso.

CMMI propone 5 distintos modelos de madurez de las organizaciones (Ver [Anexo 7](#)).

✓ **Inicial o Nivel 1 CMMI:** Este es el nivel en donde están todas las empresas que no tienen procesos. Los presupuestos se disparan, no es posible entregar el proyecto en fechas, te tienes que quedar durante noches y fines de semana para terminar un proyecto. No hay control sobre el estado del proyecto, el desarrollo del proyecto es completamente opaco, no sabes lo que pasa en él.

✓ **Repetible o Nivel 2 CMMI:** Quiere decir que el éxito de los resultados obtenidos se pueden repetir. La principal diferencia entre este nivel y el anterior es que el proyecto es gestionado y controlado durante su desarrollo. El desarrollo no es opaco y se puede saber el estado del proyecto en todo momento.

Los procesos que hay que implantar para alcanzar este nivel son:

1. Gestión de requisitos.
2. Planificación de proyectos.
3. Seguimiento y control de proyectos.
4. Gestión de proveedores.
5. Aseguramiento de la calidad.
6. Gestión de la configuración.

✓ **Definido o Nivel 3 CMMI:** Los procesos del software para la administración y las actividades de ingeniería están documentado, estandarizado, e integrado a un proceso estándar del software para la organización. Todos los proyectos usan una versión aprobada, hecha a la medida del proceso estándar del software de la organización para desarrollar y mantener software. La mayoría de las empresas que llegan al nivel 3 paran aquí, ya que es un nivel que proporciona muchos beneficios y no ven la necesidad de ir más allá porque tienen cubiertas la mayoría de sus necesidades.

Los procesos que hay que implantar para alcanzar este nivel son:

1. Desarrollo de requisitos.
2. Solución Técnica.
3. Integración del producto.
4. Verificación.
5. Validación.
6. Desarrollo y mejora de los procesos de la organización.
7. Definición de los procesos de la organización.
8. Planificación de la formación.
9. Gestión de riesgos.
10. Análisis y resolución de toma de decisiones.

✓ **Cuantitativamente Gestionado o Nivel 4 CMMI:** Los proyectos usan objetivos medibles para alcanzar las necesidades de los clientes y la organización. Se usan métricas para gestionar la organización.

Los procesos que hay que implantar para alcanzar este nivel son:

1. Gestión cuantitativa de proyectos
2. Mejora de los procesos de la organización.

✓ **Optimizado o Nivel 5 CMM – CMMI:** Los procesos de los proyectos y de la organización están orientados a la mejora de las actividades. Mejoras incrementales e innovadoras de los procesos que mediante métricas son identificadas, evaluadas y puestas en práctica.

Los procesos que hay que implantar para alcanzar este nivel son:

1. Innovación organizacional
2. Análisis y resolución de las causas.

Normalmente las empresas que intentan alcanzar los niveles 4 y 5 lo realizan simultáneamente ya que están muy relacionados.

El Modelo de Madurez y Capacidad Integrado posee dos enfoques que tienen como finalidad atender a las diversas necesidades de las organizaciones que quieren realizar la mejora de sus procesos, estos son:

1. Continuo.

2. Escalonado.

- ✓ El enfoque continuo hace hincapié en la capacidad de ciertas áreas para realizar sus actividades de manera adecuada.
- ✓ El enfoque escalonado o por niveles hace especial énfasis en el grado de madurez de los procesos o sea la madurez de la organización aumenta al incorporar un conjunto de prácticas. (8)

1.5 Enfoques del Aseguramiento de la Calidad.

1.5.1 Aseguramiento de la Calidad según ISO 90003.

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 requisitos 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:

1. Implementación del proceso.
2. Aseguramiento del producto.
3. Aseguramiento del proceso.
4. Aseguramiento del sistema de la calidad.

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 requisitos 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. (27)

1.5.2 Aseguramiento de la Calidad según CMMI.

El Modelo Madurez y Capacidad Integrado (CMMI) posee un área de proceso para el aseguramiento de la calidad y se nombra Aseguramiento de la Calidad de los Procesos y los Productos del trabajo. El objetivo de dicho modelo en esta área de procesos es proveer al equipo y la administración de un mejor entendimiento de los procesos y los trabajos relacionados al producto.

Para obtener un producto confiable y con la calidad requerida hay que tener en cuenta los siguientes aspectos:

- ✓ 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 de que los elementos no conformes sean arreglados.

CMMI para asegurar la calidad exige 2 objetivos específicos, los cuales incluyen a su vez dos prácticas específicas, compuestas por subprácticas específicas.

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

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

Objetivo Específico: Proporcionar un entendimiento objetivo.

- ✓ Comunicar y asegurar la resolución de las no conformidades emitidas.
- ✓ Establecer los registros. (8)

Los autores consideran que el área de Aseguramiento de la Calidad del Proceso y el Producto, es un área de proceso clave, en el desarrollo de productos de Software, que a veces no se le da la suficiente importancia, pero que sin ella es imposible implantar un modelo de calidad.

1.6 Calidad del Software. Situación Actual.

1.6.1 Calidad del Software a nivel Internacional.

Las tres "I" del desarrollo del Software.

En los últimos años La India, Irlanda e Israel han tenido un desarrollo vertiginoso en la ISW. Estos países proporcionan una gran variedad de servicios a la industria, inclusive realizan la parte más sofisticada del desarrollo del software.

El éxito de la industria del software en la India ha sido el tránsito por la calidad, comenzando por productos, después proyectos, normas ISO y el modelo de Madurez y Capacidad (CMM). En Israel las claves fundamentales que lo convirtieron en un actor importante en el desarrollo de software son están asociadas con el alto desarrollo en la industria de capitales de riesgo, la alta disponibilidad de recursos capacitados y motivados, la necesidad de desarrollo de tecnología militar avanzada, las instituciones académicas con reconocimiento mundial y los programas activos por parte del gobierno, centros de investigación y desarrollo de incubadores tecnológicos. En el caso de Irlanda las claves fundamentales que contribuyeron al desarrollo de la ISW fueron la ubicación geográfica privilegiada para convertirse en proveedor de los mercados internacionales, la disminución de los costos del transporte, las tasas de interés a nivel internacional y la alta rentabilidad lograda. (22)

Situación en Latinoamérica.

Las empresas y los grandes compradores de software en Latinoamérica invierten desde hace algunos años grandes cantidades de dinero en la mejora de sus procesos de desarrollo para poder unificar los criterios de la evaluación de la calidad y aspirar a la certificación cumpliendo las normas ISO 9000 o el modelo estadounidense conocido como Capability Maturity Model (CMM). (22)

Los principales problemas a los que se enfrenta la industria de software en Latinoamérica en estos momentos pueden clasificarse en tres categorías:

- ✓ Déficit de recursos humanos en cantidad y calidad.

- ✓ Fortalecimiento de una institución como representante de los intereses de los productores de software.
- ✓ La implementación de estándares de calidad de categoría internacional.

Dentro de estas se destacan dos variables que necesariamente deben considerarse para el desarrollo de un software: la calidad y la productividad. La primera se encuentra afectada por factores como el personal a cargo del proceso, el tipo de herramientas utilizadas y las metodologías aplicadas. El segundo aspecto se refiere a la capacidad de la empresa de reducir costos previniendo las fallas antes que se produzcan.

Entre las acciones que se están realizando en estos países se encuentra la elaboración de una norma nacional que haga referencia a aspectos técnicos y administrativos promoviendo la realización de programas y certificados de calidad, así como la divulgación e implantación en la industria de estándares internacionales de calidad. (22)

En Argentina el 90% de las 229 empresas que se acogieron a la Ley de Software, disponen de una certificación de calidad internacional, hecho muy auspicioso que revela la creciente necesidad de actuar en un mercado altamente competitivo, donde los precios no son la única variante para ganar mercados. (28)

1.6.2 Calidad del Software en Cuba.

Dentro de las proyecciones del gobierno cubano se encuentra el fomento de una Industria Cubana del Software, que permita diseñar y proveer de equipos electrónicos y sistemas informáticos que beneficien a la sociedad y también con posibilidad de exportarlos para aportar a la base material de todos los programas del país. (22)

En las empresas productoras de software cubanas no se han aplicado procesos de certificación ni tampoco de auto evaluación que permitan valorar el estado de la organización en cuanto a la madurez de sus procesos. (29)

El desarrollo de una Industria Nacional de Software es una tarea de gran prioridad para el estado cubano debido a la alta perspectiva económica que posee, así como para el aseguramiento de un grupo importante de actividades del país. A pesar de ello, los resultados alcanzados no cubren las expectativas, ya que la productividad es baja, la cantidad real de recursos a consumir en tiempo

principalmente es casi impredecible y el trabajo realizado casi nunca tiene la calidad y profesionalidad requerida. Los proyectos están excesivamente tarde y los beneficios que pudieran obtenerse al utilizar los mejores métodos e instrumentos en las distintas etapas, no se detectan en este medio indisciplinado y caótico de desarrollo. (22)

Hoy en día la industria del software no ha acabado de salir de la fase artesanal debido a la desorganización y la falta de planificación. La disciplina Ingeniería del Software es relativamente reciente y sus conceptos más importantes están aun inmaduros. Se carece de un corpus de conocimiento aceptado, mayoritariamente que sirva de fundamentos, además de que se tiene una escasa presión del mercado. En una organización inmadura existen procesos de software normalmente improvisados, y en caso de que se hayan especificado no se siguen rigurosamente. Presentan una organización reactiva (resolver crisis inmediatas). Los planes y presupuestos exceden sistemáticamente, al no estar basados en estimaciones realistas.

En las organizaciones inmaduras si existen plazos rígidos, se sacrifican funcionalidad y calidad del producto para satisfacer el plan. (30)

En el país existen empresas productoras de software como Softel, Desoft, las cuales no poseen una estrategia de Aseguramiento de la Calidad definida en común, los procesos son inestables e inmaduros, debido a que son empresas con poca experiencia en la producción del Software y no se encuentran certificadas con ningún modelo de calidad. No existe un departamento central que dirija el proceso de producción de Software para lograr un correcto aseguramiento. Como necesidad de controlar la calidad se creó el centro de Calidad de Software (CaliSoft).

1.6.3 Calidad del Software en la UCI.

En la Universidad de las Ciencias Informáticas se logró formar y consolidar una estructura productiva formada por los Proyectos Productivos de las 10 Facultades. Para lograr una producción de Software con la calidad necesaria y poder insertar los productos en el mercado se crea la Dirección de Calidad quien dirige y controla a nivel general la calidad en la UCI, un Grupo de Calidad por facultad dirigidos por su correspondiente Asesor de Calidad, que controlan la calidad de los proyectos respectivos a la facultad.

Además se puede decir que se ha logrado definir, Lineamientos Mínimos de Calidad, por parte de la Dirección de Calidad de Software, donde cada facultad y proyecto productivo, debe redefinir los

mismos en dependencia de sus necesidades de producción y la naturaleza del producto que desarrollan.

También es válido mencionar que se ha logrado establecer una serie de documentos y plantillas que tributan a la organización interna de cada proyecto, donde se registran todas las actividades, artefactos y procesos realizados en un proyecto dando lugar al Expediente del Proyecto, formado por cuatro grupos principales: Gestión de Proyecto, Ingeniería, Legal y Soporte, y cada uno de estos desglosados en subgrupos, la estructura del mismo es de la forma que sigue:

1. Gestión de Proyectos.

- 1.1 Plan del proyecto.
- 1.2 Riesgos.
- 1.3 Recursos.
- 1.4 Acuerdos de trabajo.
- 1.5 Contratación.
- 1.6 Informes
- 1.7 Reuniones.

2. Ingeniería.

- 2.1 Requisitos.
- 2.2 Arquitectura y Diseño.
- 2.3 Implementación y Pruebas.
- 2.4 Despliegue e Instalación.

3. Legal.

4. Soporte.

- 4.1 Aseguramiento de la Calidad.
- 4.2 Gestión de Configuración.

Los proyectos se rigen para el Aseguramiento de la Calidad por los modelos ISO/IEC 15504 y CMMi, no existe una estrategia definida en común ya que cada proyecto define su propia estrategia de Aseguramiento de la Calidad en dependencia al tipo de producto que desarrolla y las metodologías y estándares de desarrollo que son definidos para la realización del mismo.

Capítulo 2. Análisis de la Situación Actual.

Introducción.

En este capítulo se realiza primeramente una descripción detallada de los antecedentes y el surgimiento del Centro de Neurociencias de Cuba (CNEURO), como institución, abordando de manera breve el tema de la Discalculia. Posteriormente se describe el surgimiento de Juegos CNeuro como proyecto, así como su estructura, funcionamiento interno y otros aspectos de interés relacionados con el mismo.

2.1 Neuro Ciencias. Antecedentes.

Los orígenes del Centro de Neurociencias de Cuba (CNEURO) datan de 1966 cuando el Centro Nacional de Investigaciones Científicas (CNIC) creó el Departamento de Neurofisiología con el propósito de garantizar la introducción de métodos cuantitativos y el uso de la computación en el análisis de la actividad eléctrica espontánea y evocada del cerebro. Ver en [Anexo 8](#), imágenes del centro.

En 1982, como resultado del incremento y diversificación de la labor investigativa del Departamento se creó la Dirección de Neurociencias del CNIC con el objetivo de perfeccionar las tecnologías diseñadas e introducirlas en Programas Nacionales de Salud y el 11 de mayo de 1990 por decisión del Consejo de Estado se crea el Centro de Neurociencias de Cuba como una unidad científico – productiva del Centro Nacional de Investigaciones Científicas (CNIC) a partir de la fusión de la hasta entonces Dirección de Neurociencias del CNIC con el grupo MEDICID del Instituto Central de Investigaciones Digitales (ICID), con el objetivo de crear una unidad dedicada a la investigación, la producción y la comercialización de tecnologías avanzadas para el diagnóstico y tratamiento de enfermedades del cerebro.

El 4 de marzo de 1991 en virtud de la Escritura 121 se crea la empresa NEURONIC S.A., la que funciona como dependencia del Centro de Neurociencias para la comercialización de las tecnologías desarrolladas en el centro.

Finalmente el 1ro de enero del 2005 en virtud de la Resolución No. 7 se crea como centro independiente subordinado al Consejo de Estado, el Centro de Neurociencias de Cuba con un objeto social dentro del cual caben señalar las posibilidades de:

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

- ✓ Realizar, organizar y dirigir investigaciones en el campo de las neurociencias, desde niveles moleculares y genéticos hasta niveles sistemáticos y su interacción con factores ambientales y sociales.
- ✓ Llevar a cabo la conducción de la red nacional de neurofisiología clínica, así como los programas de salud o educación relacionados con las neurociencias.
- ✓ Producir y comercializar de forma mayorista productos relacionados con las neurociencias y la biomedicina.
- ✓ Producir y comercializar de forma mayorista software relacionados con las neurociencias y la biomedicina en moneda nacional y en el extranjero.
- ✓ Brindar servicios de transferencia de tecnologías o productos derivados de sus investigaciones.
- ✓ Brindar servicios de capacitación y adiestramiento a través de cursos de postgrados, doctorados, maestrías y otras actividades académicas en especialidades relacionadas con las neurociencias.
- ✓ Diseñar, implementar, conducir y realizar ensayos clínicos relacionados con investigaciones y productos de las neurociencias. (31)

2.2 Discalculia.

El dominio de las capacidades numéricas básicas desde edades muy tempranas pudiera hacer pensar que todos los individuos disponen del equipamiento cognitivo necesario para desarrollar estas capacidades exitosamente, cuando en realidad, la dificultad para el aprendizaje de las matemáticas pueden alcanzar una prevalencia que oscila entre el 2,5 y el 6,4% en la población escolar.

Esta dificultad de aprendizaje frecuentemente denominada Discalculia del Desarrollo se define según el Manual Diagnóstico y Estadístico de los Trastornos Mentales de la Asociación Psiquiátrica Americana en su cuarta edición (DSM-IV) (1995) como “una capacidad aritmética (medida mediante pruebas normalizadas de cálculo o razonamiento matemático administradas individualmente) que se sitúa sustancialmente por debajo de la esperada en individuos de edad cronológica, coeficiente de inteligencia y escolaridad concordantes con la edad (Criterio A). El trastorno del cálculo interfiere significativamente en el rendimiento académico o las actividades de la vida cotidiana que requieren habilidades para las matemáticas (Criterio B). Si hay un déficit sensorial, las dificultades en la aptitud matemática deben exceder de las asociadas habitualmente a él (Criterio C).”

Uno de los problemas metodológicos más importantes en el estudio de la Discalculia del Desarrollo está relacionado con la definición de las pruebas con las que se diagnostica el trastorno. Estas, en su

mayoría, miden el grado de desempeño matemático según criterios estadísticos: se evalúa a los sujetos teniendo en cuenta si resuelven adecuadamente ejercicios matemáticos de acuerdo a la edad cronológica y al nivel escolar.

El diagnóstico obtenido de este modo no brinda información acerca de los procesos que están afectados y que condicionan el bajo rendimiento exhibido, ni controla el efecto de factores que producen dificultades para el aprendizaje de las Matemáticas y no Discalculia; como la falta de motivación por el aprendizaje de las matemáticas o la mala enseñanza por parte del maestro. Por último la mayoría de las pruebas no toman en consideración la velocidad de cómputo sino únicamente el resultado obtenido durante la realización de un ejercicio. Frecuentemente los individuos discalcúlicos producen respuestas correctas usando procedimientos poco eficientes de manera mecánica y sin seguridad.

En un intento por resolver estas limitaciones, recientemente se ha desarrollado un nuevo enfoque para la evaluación de la Discalculia basado no sólo en la medición del desempeño matemático sino de las capacidades numéricas básicas a través de tareas que controlan el tiempo de reacción del individuo durante su ejecución. En los últimos cinco años se ha notado un creciente interés en el estudio del procesamiento numérico y sus dificultades por parte de investigadores de diferentes laboratorios.

2.3 Juegos CNeuro. Estructura y Composición.

2.3.1 Introducción.

La Facultad 5 de la UCI, desarrolla productos de Realidad Virtual y Automatización. Entre sus proyectos se encuentra el proyecto Juegos CNeuro, el cual desarrolla un juego para la intervención de la Discalculia en escolares de 6to grado.

En diciembre del año 2007 surge como proyecto productivo de la Facultad 5, estimándose obtener un producto terminado que cuente con 6 niveles para enero del año 2009. La primera versión del producto se planifica esté disponible a finales del año 2008, la cual contará solamente con los tres primeros niveles del juego.

El mismo va orientado directamente a los niños que padecen la enfermedad de Discalculia y a los especialistas que atienden este padecimiento. El problema que dio origen a este proyecto fue el no contar en Cuba, y de manera similar en el mundo, con un método eficaz para la intervención de la Discalculia.

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

Producto de esta situación imperante, actualmente, hay muchas personas que viven con este padecimiento, fundamentalmente, los niños, son los que más lo sufren ya que les dificulta mucho su aprendizaje en la escuela, lo que puede afectarlos psicológicamente, además de impedirles avanzar en sus conocimientos.

En un futuro se piensa desarrollar un juego para cada uno de los grados de escolaridad primaria, desde 1ero al 5to.

2.3.2 Roles y responsabilidades definidos.

En Juegos CNeuro se definieron inicialmente, y se encuentran actualmente los roles y responsabilidades, según se muestra en el [Anexo 9](#):

Los autores consideran que el rol de Responsable de Calidad definido dentro del proyecto no satisface las necesidades del mismo, en cuanto al tema de Aseguramiento de la Calidad, ya que el rol antes mencionado se asocian las actividades de:

- ✓ 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.
- ✓ Guiar las auditorías que se realice.
- ✓ 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, por parte de los autores, que estas actividades son demasiadas para estar asignadas a un solo rol dentro del proyecto. Para ello se realiza, en el Capítulo 3 de esta tesis, una propuesta de cómo deben quedar definidos los roles encargados de Asegurar la Calidad dentro de Juegos CNeuro.

2.3.3 Recursos humanos por roles.

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

✓ Roles de profesores:

- Líder de Proyecto: 1 persona.
- Arquitecto: 1 persona.
- Jefe de Grupo de Desarrollo: 3 persona.
- Trabajo Educativo y Psicólogo: 1 persona.

✓ Roles de estudiantes.

- Analista: 2 personas.
- Calidad: 1 persona.
- Implementador: 11 personas.
- Gestión de Configuración: 2 personas.

A modo de resumen, Juegos CNeuro cuenta con un total de 22 integrantes en su composición, divididos en 16 estudiantes y 6 profesores. Los autores de esta tesis consideran que debe ser asignado, por parte del líder del proyecto, más personal a la parte de Calidad dentro del proyecto, para lograr poner en práctica la Estrategia de Aseguramiento de la Calidad que se expone en el Capítulo 3 de este trabajo.

2.3.4 Soporte Tecnológico.

Como soporte tecnológico, Juegos CNeuro cuenta con 10 estaciones de trabajo formadas por 10 computadoras, en perfectas condiciones técnicas y que están siendo utilizadas. Se cuenta además con una computadora que hace función de servidor donde se almacena la información contenida en el repositorio del proyecto.

2.3.5 Módulos del producto.

El producto que se genera en Juegos CNeuro consta de tres módulos, estos son: Kernel, Niveles y Algoritmo Adaptativo. A continuación se hace una breve descripción de cada uno de estos módulos:

✓ **Módulo Kernel.**

Es el núcleo de la aplicación. En él se implementan todas las funcionalidades generales del sistema ya que es el encargado de:

- Cargar el juego.
- Cargar los niveles.
- Procesar todos los datos.
- La interfaz de usuario.
- Definir todas las funcionalidades que después serán implementadas en cada uno de los niveles.

Debe ser lo más genérico posible, con el objetivo de garantizar la escalabilidad del sistema, además esto permite que este módulo pueda ser reutilizado por otros sistemas similares.

✓ **Módulo Niveles.**

En este módulo se encuentran todos los niveles del juego. En todos los casos muchas de las funcionalidades del Kernel son utilizadas por cada uno de los niveles, ya que están implementadas en él. Algunas de estas funcionalidades las redefinen y otras solamente las utilizan. En cada nivel se implementarán las características particulares de cada uno, es decir, las que son específicas del nivel, ya que esas no se pueden implementar en el Kernel.

✓ **Módulo Algoritmo Adaptativo.**

El Algoritmo Adaptativo, aunque pertenece al Kernel, se trata como un módulo independiente por su complejidad e importancia. Este módulo es quien rige todo el proceso de intervención del niño. Sus funcionalidades principales son:

- Generar el estímulo que se le mostrará al niño en cada momento.
- La forma en que se le mostrará cada estímulo.

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

- El tiempo que tendrá el niño para responder al estímulo dado.
- Calcular el espacio de conocimiento del niño.
- Indicar cuándo el niño ha llegado al máximo de su rendimiento.

Este módulo, al igual que el Kernel, permite ser reutilizado, ya que sus funcionalidades pueden ser similares a las de otros juegos.

2.3.6 Producto generado.

El producto generado por Juegos CNeuro recibe el nombre de “Venganza”, el mismo va a contribuir en gran medida, en la intervención de la Discalculia en escolares de 6to grado. Este producto presenta las siguientes funcionalidades y características:

- ✓ Seleccionar un personaje con el que el usuario se sienta identificado.
- ✓ Seleccionar el nivel que se desea jugar.
- ✓ Registro automático de todas las acciones realizadas por el usuario para el posterior análisis de las mismas por parte de los especialistas.
- ✓ Selección automática de las tareas a mostrar según el nivel de afectación del usuario.
- ✓ Generación de entornos dinámicos.

Los atributos distintivos o los elementos claves, distintivos, diferenciadores que agregan valor, de este producto son:

- ✓ Entornos 3D.
- ✓ Diseño gráfico llamativo.
- ✓ Fácil de utilizar.
- ✓ Modelado del espacio de conocimiento del usuario.

2.3.7 Arquitectura.

✓ Representación Arquitectónica.

La arquitectura del producto Venganza 1.0 a desarrollar por el proyecto Juegos CNeuro tiene un patrón de tres capas Application, Logic y Render. El producto se desarrolla sobre la herramienta *SceneToolkit* y el patrón tres capas sobre el cual se desarrolla la *SceneToolkit*, está separada en tres componentes:

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

- Engine: Parte que hace todo el cálculo para cada “ciclo de escena”.
- Render: Encargada del dibujado con la tarjeta gráfica.
- Application: Encargada de la interacción con el Sistema Operativo y eventos, etc.

Para realizar el producto se realiza una especialización de la capa Application del STK, usamos el Engine del *SceneToolkit* en la capa Logic (encargada de todo lo referente a los procesos del juego, aquí se encuentran la capa General (Kernel) y la Específica (Niveles) y utilizamos el mismo Render *SceneToolkit*.

✓ **Objetivos y Restricciones Arquitectónicas.**

El juego Venganza debe cumplir con el requerimiento de ser multiplataforma, que corra igualmente sobre Sistemas Operativos Windows y Linux, y extensible a otros sistemas operativos. Por esto se propone el diseño abordado con el uso de la *SceneToolkit* la cual cumple con este requisito, y se define como lenguaje de trabajo el C++ estándar y que se usen los estándares de datos para el control de listas etc. de la std.

✓ **Tamaño y Rendimiento.**

El sistema no debe contar con una cantidad de código demasiado grande por lo que esto no debe afectar la arquitectura del mismo. En cuanto al rendimiento se puede decir que debe ser una aplicación que corra en tiempo real y que consuma pocos recursos debido a que el hardware en que será utilizada tendrá pocas prestaciones en la mayoría de los casos.

✓ **Vista de procesos.**

En la realización de este producto se definió el proceso de cargar todo lo referente a los entornos que se van a visualizar y de los correspondientes ficheros. Además se tiene la respuesta a los eventos a los que se le va a dar respuesta en dependencia del momento en que se encuentre el juego. Todos estos procesos son secuenciales.

✓ **Vista de despliegue.**

No es necesario mostrar el despliegue ya que consta de una PC (La PC del niño que está siendo intervenido).

✓ **Vista de Datos.**

El sistema tendrá como datos persistentes el perfil de los usuarios que utilizarán el juego, el mismo incluye, además de los datos tradicionales (Nombre, edad, etc), el avance que ha tenido el niño en el juego a través del registro de las tareas que ha realizado. Estos datos se almacenarán utilizando los ficheros XML.

✓ **Calidad.**

La arquitectura del juego Venganza garantiza la portabilidad y extensibilidad de la misma hacia varios sistemas operativos y otras versiones donde se incorporarán más niveles y tareas.

2.3.8 Ciclo de Desarrollo.

El proyecto Juegos CNeuro se realiza siguiendo la metodología del Proceso Unificado de Rational (RUP), la misma consta de cuatro Fases: Inicio, Elaboración, Construcción y Transición. En Juegos CNeuro se planificó, por parte del líder del proyecto las iteraciones necesarias para obtener un producto que satisfaga las necesidades del cliente o usuarios finales, las mismas se citan a continuación:

✓ Fase Inicio

- Primera Iteración:

- Captura de requisitos.
- Modelado del Negocio.

✓ Fase Elaboración

- Primera Iteración:

- Análisis y diseño del módulo Kernel.
- Análisis y diseño del módulo Niveles.

- Segunda Iteración:

- Iteración final para obtener la buena interacción entre los módulos.

✓ Fase Construcción

- Primera Iteración:
 - Obtener prototipo funcional de los tres primeros niveles (No incluye algoritmo adaptativo, solo la parte gráfica).
- Segunda Iteración:
 - Versión del juego.
- ✓ Fase Transición
- Primera Iteración:
 - Revisión inicial del cliente para verificar cumplimiento de los requisitos.
- Segunda Iteración:
 - Responder a las No conformidades del cliente.

2.3.9 Ambiente de trabajo.

En Juegos CNeuro se definieron una serie de herramientas o tecnologías a utilizar para el desarrollo del producto, las cuales se mencionan a continuación:

- ✓ Para la programación de los módulos se utiliza como lenguaje de Programación C++.
- ✓ Para la implementación se utiliza como Entorno Integrado de Desarrollo (IDE), Visual Studio 2003 que requiere como mínimo 512 MB de RAM, recomendado 1GB.
- ✓ No se utiliza ningún Sistema Gestor de Base de Datos.
- ✓ Se trabaja sobre el Sistema Operativo Windows XP.
- ✓ Para el modelado se utiliza Rational Rose 2003, el cual demanda como mínimo 256 MB de RAM y recomendado 512 MB de RAM.
- ✓ Para el Control de las versiones, se utiliza el Subversion.
- ✓ Se utiliza como herramienta de Gestión de Proyectos el Trac.

2.4 Situación real de Juegos CNeuro.

Los autores consideran importante resaltar que en el momento de su incorporación a este proyecto, había comenzado y no se contaba con personal calificado para desempeñarse en los roles de Calidad dentro del proyecto.

Existían además incongruencias en cuanto al guión que debía seguir el juego, lo que dificultó en sobremanera establecer una estrategia definitiva para asegurar la calidad del proceso de desarrollo, ya que este proceso dependía del guión. A partir de esta situación los autores de esta investigación se dieron a la tarea de evaluar y analizar la situación existente en el proyecto, para a partir de estos resultados poder implementar el proceso de Aseguramiento de la Calidad dentro del proyecto. Para realizar el diagnóstico se tuvo en cuenta primeramente la situación en que se encontraba Juegos CNeuro.

Comienzo de Juegos CNeuro.

Para dar comienzo al proyecto se realizaron reuniones, por parte del líder del proyecto, con Neuro Ciencias, posteriormente a esto se reúnen los miembros del equipo de desarrollo con el líder del proyecto. Durante estos encuentros se puntualizaron los detalles necesarios para confeccionar los primeros artefactos que inician el proceso de desarrollo de Software:

- ✓ Riesgos del proyecto.
- ✓ Plan de Mitigación de Riesgos.
- ✓ Propuesta inicial de roles.
- ✓ Documento Visión.

Captura de requisitos.

La captura de los requisitos funcionales y no funcionales se realizó a partir del guión elaborado en cada una de sus versiones, lo que propició que se tuvieron que realizar varias iteraciones para lograr llegar a los requisitos que hasta el mes de junio del 2008 están definidos. Se realizaron además encuentros con el cliente, para conocer sus exigencias y necesidades en cuanto al producto, el resultado de estas entrevistas fue relevante para el proceso de la captura de requisitos.

Esta actividad no pudo ser revisada a tiempo, debido a la situación existente con el equipo de calidad, que es el encargado de revisar cada uno de los artefactos generados en las diferentes fases y flujos de trabajo del proyecto.

Casos de Uso del Sistema.

Los requisitos capturados por el equipo de desarrollo, constituyeron el artefacto de entrada para la confección de los Casos de Uso del sistema. Esta actividad se realizó acorde con lo planificado en el cronograma, generando como artefacto el Diagrama de Casos de Uso del Sistema Ver en el Expediente Proyecto Juegos CNeuro.

Esta actividad, igualmente a la captura de requisitos, no pudo ser revisada a tiempo debido a la misma situación.

Estas actividades fueron sometidas a revisiones posteriormente, cuando los autores de este trabajo de diploma se incorporaron a Juegos CNeuro y se llevó a cabo la confección del Plan de Aseguramiento de la Calidad, donde se tuvo en cuenta la revisión de las actividades de Ingeniería de SW, dentro de las que se incluyen las mencionadas anteriormente.

Situación de los módulos.

Los módulos del proyecto se encontraban en el Flujo de Trabajo de Análisis y diseño, en la primera iteración de la Fase de Elaboración, esta situación se debía a que no se había llegado a una versión final del guión del juego por parte del cliente y esto retrasaba las actividades planificadas para el comienzo del producto. El cliente realizó varias iteraciones par poder llegar a conformar una última versión del guión, la cual propició que se pudiera elaborar el guión técnico, correspondiente a los tres primeros niveles del juego, por parte de los desarrolladores y el cliente.

Este documento elaborado es capaz de proporcionar los datos para una primera versión del producto que estaría formada por los tres primeros niveles. Hasta el mes de junio del 2008 según muestra el cronograma de actividades de Juegos CNeuro (Ver Expediente Proyecto de Juegos CNeuro), los módulos se encuentran realizando el Flujo de trabajo de Implementación dentro de la primera iteración de la Fase de Construcción. El diseño de la interfaz gráfica y ambiente gráfico del juego se encuentra retrasado. El proceso de diseño gráfico comienza cuando se entrega el guión por parte del proyecto al diseñador de interfaz gráfica, esta actividad, se encuentra retrasada aproximadamente un mes, ya que Juegos CNeuro debía contar con todas conceptualizaciones para que estas sean entregadas a los realizadores de los entornos (encargados de realizar los modelos 3D de los entornos diseñados). A partir de que esté listo esto se podrá comenzar a realizar el montaje de los entornos, de la primera versión, por parte del equipo de desarrollo de Juegos CNeuro.

2.5 Resultados esperados de Juegos CNeuro.

Se espera que con este producto mejore la afectación que presentan estos niños, además de poder ser utilizado, como material de estudio, por los que no la padecen. Existe una demanda de intervención entre un 3% y un 6% de los escolares cubanos con Discalculia del Desarrollo. Este trastorno del aprendizaje no es susceptible de mejorar con intervención pedagógica, pues sus causas son de origen biológico. No existen vías para potenciar las capacidades numéricas en escolares en edades tempranas del desarrollo. En el mundo esta situación se comporta de manera muy similar a la de Cuba. Estos juegos tendrán entornos atractivos, con lo cual se lograría una mayor motivación de los niños. En un futuro se piensa desarrollar un juego para cada uno de los grados de escolaridad primaria, desde 1ero al 5to grado.

De manera concreta y resumida los resultados esperados del producto desarrollado en Juegos CNeuro son los siguientes:

1. Desarrollar un Kernel potente y genérico que se pueda reutilizar en juegos similares.
2. Desarrollar un algoritmo adaptativo que cuente con todas las tareas necesarias para intervenir la Discalculia.
3. Desarrollar un juego para la intervención de la Discalculia con entornos atractivos para con esto lograr una mejora en los niños de 6to grado que padecen esta enfermedad.

Los autores opinan que este es un proyecto que aunque no está completamente desarrollado, si tiene grandes expectativas, ya que da tratamiento a un tema que es medular para la sociedad como lo es el tema de esta discapacidad en los niños. Se considera además que es un Juegos que va a despertar gran interés por parte de las instituciones de enseñanza ya que el mismo va a constituir una herramienta para el tratamiento y rehabilitación de pacientes, en este caso los niños que padecen de la Discalculia, para lograr su completo restablecimiento y lograr un normal proceso de aprendizaje en esta edad tan difícil para los escolares.

Capítulo 3. Descripción de la Solución.

Introducción.

En este capítulo se describe la estrategia de Aseguramiento de la Calidad para el proyecto Juegos CNeuro. Planteando las actividades realizadas así como sus resultados. Las tareas que se continuarán aplicando durante el proceso de desarrollo del software que se produce, quedan igualmente expuestas en este capítulo.

Se especifica cómo se asegurará la calidad del proyecto Juegos CNeuro, con la intención de proporcionar seguridad de que el producto que se obtendrá y los procesos del ciclo de vida del proyecto estén conformes con los requisitos específicos y se ajusten a los planes establecidos.

3.1 Estrategia de Aseguramiento de la Calidad para Juegos CNeuro.

La estrategia de Aseguramiento de la Calidad del Proyecto Juegos CNeuro está conformada por las siguientes actividades:

- ✓ Redefinir los roles que conformarán el equipo de calidad.
- ✓ Elaborar el Plan de Aseguramiento de la Calidad.
- ✓ Aplicar el Plan de Aseguramiento de la Calidad.

- Revisiones Técnicas Formales.
 - Revisión de las actividades de Ingeniería de Software.
- Revisiones externas a Juegos CNeuro.
- Actividades de corrección a las recomendaciones de las revisiones externas.
- Pruebas de software al producto.
 - Diseñar las pruebas.
 - Ejecutar las pruebas.
 - Registrar resultados de las pruebas.
 - Evaluar resultados de las pruebas.

3.1.1 Roles y responsabilidades.

Para la aplicación de la estrategia definida los autores de esta investigación, proponen que debe existir un Equipo de Calidad definido dentro del proyecto; que debe realizar las actividades de aseguramiento, dentro de Juegos CNeuro, como se explicó en el Capítulo 2 de este trabajo de diploma el rol definido: “Responsable de Calidad” no satisface las necesidades de Aseguramiento y las actividades asociadas son demasiadas para un solo rol; para de esta forma poder detectar con facilidad los errores que pueden afectar la Calidad del Software en cada iteración. Estos roles, y sus actividades asociadas, son los siguientes:

✓ **Administrador de calidad**, responsable de:

- Asegurar la calidad en el proceso de desarrollo de software.
- Asegurar que la aplicación que se produce se ajusta a las especificaciones.
- Elaborar Plan de Pruebas.
- Proporcionar, a los diseñadores de prueba una metodología para realizar las pruebas.
- Coordinar y dirigir las pruebas de calidad internas.
- Evaluar los resultados que se obtienen en las pruebas de calidad.

✓ **Diseñador de Pruebas**, responsable de:

- Diseñar los Casos de prueba.
- Definir las Listas de Chequeo a aplicar.

✓ **Probador**, responsable de:

- Ejecutar las pruebas.
- Documentar los resultados de las pruebas.

✓ **Revisor Técnico** responsable de:

- Revisar la arquitectura, el diseño, los requisitos.
- Mantener actualizada la documentación que conforma el Expediente del Proyecto.

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

3.1.2 Plan de Aseguramiento de la Calidad para Juegos CNeuro.

Se elaboró el Plan de Aseguramiento de la Calidad del proyecto 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 de proyecto como responsable también en el proceso de Aseguramiento.

Abarca las revisiones y pruebas que se realizaron en las diferentes fases del Proceso de desarrollo de Juegos CNeuro.

Para la elaboración de este Plan se tuvieron en cuenta los documentos siguientes:

- ✓ Lineamientos Mínimos de Calidad establecidos por la Dirección de Calidad de Software de la UCI.
- ✓ Plan de Desarrollo de Software.
- ✓ Expediente del proyecto Juegos CNeuro.
- ✓ Estándares y normas de calidad.

El Plan de Aseguramiento elaborado se trazó los siguientes objetivos de calidad:

- ✓ Asegurar la calidad desde el mismo inicio del proyecto para cumplir con los requerimientos del cliente, tomando como referencia los procedimientos de la UCI, normas internacionales existentes y el área de proceso del estándar de calidad CMMI: Aseguramiento de la Calidad del Producto y el Proceso.
- ✓ Aplicar las mejores prácticas de ingeniería y gestión de software.
- ✓ Lograr que todo el personal cuente con los conocimientos y habilidades necesarias para realizar las tareas y actividades solicitadas por el cliente.

Para la puesta en práctica del Plan elaborado se realizaron las actividades que a continuación se describen. La estrategia definida se puso en práctica, a partir de las Revisiones Técnicas Formales que persiguen como principal objetivo el de: Garantizar que el Software ha sido representado de acuerdo a los estándares predefinidos y detectar la mayor cantidad de errores posibles antes de continuar con el proceso de desarrollo.

Para concretar estas revisiones se elaboraron Listas de Chequeo utilizando los Lineamientos Mínimos de Calidad establecidos por la Dirección Central de Calidad de la UCI y Estándares y Normas de calidad, todos estos resumidos en el [Anexo 10](#).

Actividades de Aseguramiento de la Calidad.

Revisión de las actividades de Ingeniería de software.

1. Revisión al Inicio del proyecto.

✓ **Objetivo:** Revisar la documentación elaborada durante la primera fase o fase de inicio del proyecto, así como su adecuación a las plantillas establecidas en el Expediente del Proyecto. Entre los documentos revisados se encuentran:

- Lista de Riesgos del Proyecto.
- Plan de Mitigación de Riesgos.
- Documento Visión.
- Propuesta de Roles del Proyecto.
- Plan de Capacitación.

✓ **Responsable:** Revisor Técnico.

✓ **Artefactos de entrada:**

- Lista de Chequeo para la Revisión Inicial.

✓ **Artefactos que genera:**

- No Conformidades.
- Minuta de Reunión.

✓ **Fase o etapa de aplicación:** Fase de Inicio del proyecto.

✓ **Descripción:**

Se elaboró y aplicó una Lista de Chequeo (Ver [Anexo 11](#)) para verificar la elaboración correcta de los artefactos generados durante esta primera fase del proceso de desarrollo. La misma fue elaborada

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

teniendo en cuenta los Lineamientos Mínimos de Calidad establecidos por la Dirección de Calidad de Software de la UCI.

Esta actividad es considerada como crítica dentro del proceso de desarrollo del producto, ya que de su realización depende que el proyecto inicie correctamente y con una estructura lo más sólida posible.

Durante esta revisión se detectaron algunas inconformidades, las cuales fueron recogidas en un documento de No Conformidades (Ver [Anexo 12](#)), pudiéndose destacar como principales errores los siguientes:

- En las reglas de Confidencialidad no se especifica el proyecto o empresa involucrada (Lista de Riesgos).
- No se han terminado de llenar las secciones siguientes del Plan de Mitigación de Riesgos:
 - Organización y Responsabilidades.
 - Presupuesto.
 - Herramientas.
 - Técnicas.
- No se ha terminado de elaborar el documento visión por parte del líder del proyecto.
- El rol definido (Responsable de Calidad) no satisface las necesidades del proyecto en cuanto al tema de Aseguramiento de la Calidad ya que se le asocian demasiadas actividades para un solo rol (Propuesta de Roles).

2. Revisión de los Requisitos del proyecto Juegos CNeuro.

✓ **Objetivo:** Revisar los requisitos elaborados.

✓ **Responsable:** Revisor Técnico.

✓ **Artefactos de entrada:**

- Lista de Chequeo para la Revisión de los Requisitos de Juegos CNeuro.

✓ **Artefactos que genera:**

- No Conformidades.
- Minuta de Reunión.

✓ **Fase o etapa de aplicación:** Fase de Inicio y Fase de Elaboración.

✓ **Descripción:**

Para la realización de esta actividad se elaboró y aplicó una Lista de chequeo (Ver [Anexo 13](#)), teniendo en cuenta para su confección la IEEE 830: "Especificaciones de los requisitos del Software".

Durante la aplicación de este artefacto, en la primera y segunda fase del proyecto, no se detectaron errores, lo que demuestra que el proceso fue realizado correctamente.

Por otra parte se considera importante resaltar que hasta el momento en que se realiza este trabajo se agregan requisitos nuevos por parte de los desarrolladores para lograr cumplir con las exigencias y expectativas del cliente.

Esta fue la etapa más importante dentro del proceso de desarrollo de software, debido a que los requisitos son la base de dicho proceso, si no se hace una buena captura de requisitos, no se cumple con las especificaciones del cliente.

3. Revisión del Diagrama de Casos de Uso del proyecto Juegos CNeuro.

✓ **Objetivo:** Revisar Diagrama de Casos de Uso del proyecto Juegos CNeuro.

✓ **Responsable:** Revisor Técnico.

✓ **Artefactos de entrada:**

- Lista de Chequeo para la Revisión del Diagrama de Casos de Uso de Juegos CNeuro.

✓ **Artefactos que genera:**

- No Conformidades.

- Minuta de Reunión.

✓ **Fase o etapa de aplicación:** Fase de Inicio y Fase de Elaboración.

✓ **Descripción:**

Se elaboró y aplicó una Lista de Chequeo para revisar el Diagrama de Casos de Uso de Juegos CNeuro, (Ver [Anexo 14](#)), teniendo en cuenta los aspectos planteados por la metodología RUP para la

realización de esta actividad y lo planteado por Pressman en el libro “Ingeniería de Software, un enfoque práctico”.

Dicho artefacto, abarca los casos de uso que dan lugar a los módulos que conforman el producto Venganza; resultado de Juegos CNeuro; por lo que es de vital importancia la realización correcta y en tiempo de esta actividad, ya que de ella depende en gran parte la estructura modular del producto.

Los errores detectados durante la revisión se reflejan en un documento de No Conformidades, (Ver [Anexo 15](#)), entre los cuales se pueden citar:

- Errores ortográficos.
- Los casos de uso no están agrupados en paquetes.

4. Revisión a la Arquitectura del proyecto Juegos CNeuro.

✓ **Objetivo:** Realizar una revisión exhaustiva a la arquitectura definida en Juegos CNeuro.

✓ **Responsable:** Revisor Técnico.

✓ **Artefactos de entrada:**

- Lista de Chequeo para la Revisión de la Arquitectura de Juegos CNeuro.

✓ **Artefactos que genera:**

- No Conformidades.
- Minuta de Reunión.

✓ **Fase o etapa de aplicación:** Fase de Elaboración.

✓ **Descripción:**

Se elaboró y aplicó una Lista de Chequeo (Ver [Anexo 16](#)), teniendo en cuenta los Lineamientos Mínimos de Calidad establecidos por la Dirección de Calidad de Software, (Ver Sección 1, inciso I, LMC) y la Norma IEEE 1471: “Recomendaciones para la Descripción Arquitectónica de los Sistemas Intensivos de Software”.

Los errores detectados durante la revisión se reflejan en un documento de No Conformidades, (Ver [Anexo 17](#)). Entre las principales deficiencias encontradas se pueden destacar los siguientes:

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

- No se identifican los usuarios, adquiridores, desarrolladores y mantenedores del sistema.
- No se define la viabilidad de construcción del sistema.
- No se definen los riesgos del desarrollo del sistema y las operaciones para los usuarios, adquiridores y desarrolladores del sistema.
- No se definen puntos de vista arquitectónicos.
- Cada vista definida no se corresponde con exactamente un punto de vista de los seleccionados anteriormente.

5. Revisión al Diseño del proyecto Juegos CNeuro.

- ✓ **Objetivo:** Detectar todos los errores existentes antes de pasar a la Implementación.
- ✓ **Responsable:** Revisor Técnico.
- ✓ **Artefactos de entrada:**

- Lista de Chequeo para la Revisión del Diseño de Juegos CNeuro.

✓ **Artefactos que genera:**

- No Conformidades.
- Minuta de Reunión.

- ✓ **Fase o etapa de aplicación:** Durante cada iteración de la Fase de Elaboración.

✓ **Descripción:**

La correcta elaboración del diseño, se verificó a partir de una Lista Chequeo, (Ver [Anexo 18](#)), elaborada a partir de los Lineamientos Mínimos de Calidad establecidos por la Dirección de Calidad de Software (Ver Sección 2, aspecto 10 y 11, LMC), la Norma IEEE 1016: “Prácticas Recomendables para las descripciones del Diseño del Software.”, y la ISO 12207: “Procesos del ciclo de vida del Software”.

Dado el objetivo de esta actividad, la misma, se debe realizarse de manera efectiva para evitar errores posteriores en la codificación del producto. Durante la revisión los errores detectados fueron recogidos en un documento de No Conformidades, (Ver [Anexo 19](#)), destacándose entre los principales que:

- No se cuenta con ninguna vista de diseño en el proyecto.
- Los artefactos del análisis y diseño no se encuentran documentados.

6. Revisión a la Gestión de la Configuración del proyecto Juegos CNeuro.

✓ **Objetivo:** Verificar que se estén realizando las actualizaciones a los pedidos de cambio por parte de los desarrolladores y que estén identificados todos los elementos de configuración del software.

✓ **Responsable:** Revisor Técnico.

✓ **Artefactos de Entrada:**

- Lista de Chequeo para la Revisión de la Gestión de la Configuración de Juegos CNeuro.

✓ **Artefactos que genera:**

- No Conformidades.
- Minuta de Reunión.

✓ **Fase o etapa de aplicación:** Antes de incluir la línea base del proyecto en el repositorio de elementos de configuración, en cada iteración realizada a cada uno de los módulos del proyecto.

✓ **Descripción:**

Para llevar a cabo esta revisión, se elaboró y aplicó una Lista de Chequeo, (Ver [Anexo 20](#)), en conformidad con los Lineamientos Mínimos de Calidad establecidos por la Dirección de Calidad de Software (Ver Sección 1, inciso D, LMC), y la Norma ISO 12207: "Procesos del ciclo de vida del Software."

Se verificó que se están controlando los cambios, y ante cada cambio se realiza la actualización correspondiente. Se comprobó la existencia y actualización de la documentación y el expediente del proyecto.

Las inconformidades encontradas durante la revisión se recogieron en un documento de No conformidades, (Ver [Anexo 21](#)), destacándose entre las principales deficiencias que:

- No se elaboró el plan de Gestión de la Configuración desde el inicio del proyecto.
- No aparece en el plan el cronograma de las actividades de Gestión de la Configuración.

- No se registra el proceso de liberación del producto una vez modificado.
- Los informes no constan de toda la información necesaria.

Revisiones Externas al Proyecto

7. Revisión externa, por parte de la Dirección de Calidad de Software de la UCI, al proyecto Juegos CNeuro.

✓ **Objetivo:** Los objetivos definidos por la Dirección de Calidad de Software de la UCI para esta revisión fueron los siguientes:

- Verificar la definición y cumplimiento del cronograma.
- Verificar la definición y cumplimiento del plan de resultados.
- Contribuir a mejorar la aplicación de los elementos claves de la ingeniería y gestión de software en los proyectos productivos.
- Evaluar la aplicación de los procedimientos y disposiciones establecidas para la producción. (Comienza con los Lineamientos Mínimos de Calidad y el Expediente del Proyecto).

✓ **Responsable:** Dirección de Calidad de Software de la UCI.

✓ **Artefactos de Entrada:**

- Guía para la revisión de la documentación de la Dirección de Calidad de Software de la UCI.

✓ **Artefactos que genera:**

- Lista de Recomendaciones.

✓ **Fase o etapa de aplicación:** Según planificación de revisiones de la Dirección de Calidad de Software de la UCI.

✓ **Descripción:**

Del 14 al 24 de marzo del 2008 se llevó a cabo una revisión a la documentación por parte de la Dirección de Calidad de Software de la UCI, en la cual se detectaron algunos desajustes en cuanto a la elaboración y actualización del Expediente del Proyecto. Entre los errores detectados, se encuentran los siguientes:

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

- Se encuentra el documento Glosario de Término, los Requisitos Funcionales y no Funcionales, pero no están en la plantilla que propone el expediente de proyecto.
- Se encuentra el documento de Cronograma de Trabajo pero las actividades se encuentran desordenadas.
- El Cronograma de Trabajo se encuentra atrasado.

Ante estos defectos se emitió una Lista de Recomendaciones, por parte de los revisores, la misma se encuentra en el Expediente del Proyecto de Juegos CNeuro.

Actividades de Corrección.

8. Dar respuesta a las Recomendaciones producto de la revisión externa a Juegos CNeuro, realizada por parte de la Dirección de Calidad de Software.

✓ **Objetivo:**

- Revisar la documentación de Juegos CNeuro, prestando especial atención a los documentos que conforman el Expediente del Proyecto.
- Comprobar el estado de las recomendaciones realizadas por la Dirección de Calidad de Software de la UCI en su revisión.

✓ **Responsable:** Revisor Técnico.

✓ **Artefactos de Entrada:**

- Para realizar estas actividades de corrección se asumió, por parte del equipo de calidad del proyecto, como artefacto de entrada, las Recomendaciones elaboradas por la Dirección de Calidad de Software de la UCI.

✓ **Artefactos que genera:**

- Minuta de Reunión.

✓ **Fase o etapa de aplicación:** Posterior a la revisión llevada a cabo por la Dirección de Calidad de Software de la UCI.

✓ **Descripción:**

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

Para corregir las deficiencias detectadas se elaboró por parte del líder del proyecto un documento de respuesta a estas No Conformidades encontradas, para monitorear la erradicación de los errores señalados.

Se llevó a cabo una revisión interna a la documentación de Juegos CNeuro por parte del equipo de calidad del proyecto dándole respuesta a las recomendaciones realizadas por la Dirección de Calidad de Software.

9. Pruebas. Control de la Calidad.

El control de la calidad es un tema que no se debe descuidar dentro de un proyecto de Software por la importancia que este reviste para la detección y eliminación de errores durante cada una de las fases del ciclo de vida del producto. Dentro del control de la calidad ocupan un lugar privilegiado las pruebas. Por esta razón, se definió un proceso de pruebas que serán aplicadas al producto durante cada una de sus fases de desarrollo. Dicho proceso consta de varias etapas, las cuales son explicadas a continuación:

1. Planificación de las Pruebas.
2. Diseño de los Casos de Pruebas.
3. Ejecución de las Pruebas.
4. Evaluación y resumen del proceso de Pruebas.

Las pruebas que serán aplicadas al producto durante su desarrollo, serán las siguientes:

1. Pruebas de Unidad.
2. Pruebas de Integración.
3. Pruebas de Sistema.

Las mismas fueron planificadas de manera anticipada a su aplicación. Para ello se confeccionaron los documentos que fundamentan y explican detalladamente, el cómo, quién y en qué momento se aplicarán cada una de las pruebas.

Planificación de las Pruebas.

- ✓ **Objetivo:** Planificar las pruebas a aplicar al producto desarrollado.
- ✓ **Responsable:** Administrador de Calidad.

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

✓ **Artefactos de entrada.**

- Plan de Desarrollo.
- Requisitos.
- Casos de Uso para los posibles casos de prueba.

✓ **Artefactos que genera:**

- Planes de Pruebas.

✓ **Fase o etapa de aplicación:** Fase de Elaboración.

✓ **Descripción:**

Para dar comienzo a este proceso se elaboró el Plan de Pruebas correspondiente a cada uno de los tipos de pruebas que serán aplicadas al producto. Dicha actividad constituye la base del proceso, porque en este documento se deben plasmar acotaciones necesarias para establecer el marco de desarrollo de las actividades de prueba.

Para la confección de los Planes de Pruebas se contemplaron, fundamentalmente, los aspectos o secciones siguientes:

- Nombre del Documento. (Ejemplo: Plan de Pruebas Unitarias).
- Nombre del proyecto. (Ejemplo: Juegos CNeuro).
- Nombre del producto: Nombre del producto que se genera. (Ejemplo: Venganza).
- Versión del producto. (Ejemplo: 1.0) Se especificó la versión del producto a la cual se le realizan las pruebas que se plantean, además de dedicar un espacio para el Control de Versiones y cambios realizados en el producto, especificando la actividad que se realiza, su descripción, fecha y el responsable o autor de la misma).
- Reglas de confidencialidad del documento. Se especificó el carácter o la clasificación de la información contenida en las páginas del documento.
- Introducción: Se especificó el alcance, las definiciones, acrónimos y abreviaturas a utilizar a lo largo del Plan, así como las referencias a los documentos que se tuvieron en cuenta para la confección del mismo.

- Organización del equipo de pruebas: Se especificó cuales son los roles que conforman el equipo de prueba y las responsabilidades de cada uno de ellos dentro del proceso.
- Arquitectura Técnica: Se describió mediante diagramas (despliegue, componentes) las partes del sistema bajo prueba.
- Especificaciones de Hardware y Software: Se especificaron los componentes lógicos y físicos necesarios para las pruebas que se plantean.
- Descripción del Plan de Pruebas: En esta sección se realizó primeramente una descripción de los requerimientos (lista de todos los requerimientos a probar dentro de las pruebas establecidas, cualquier requerimiento que no esté contemplado en dicha lista estará fuera del alcance de las pruebas), requerimientos funcionales (funciones a ser probadas), después se realizó una breve descripción del Caso de Uso (Se especificó el número del Caso de Uso y se resaltaron los aspectos necesarios para ser probado).

Todos estos aspectos mencionados se resumen y se detallan, con mayor exactitud, en los Planes de Pruebas elaborados, presentes en el Expediente del Proyecto de Juegos CNeuro.

Diseño de los Casos de Prueba.

- ✓ **Objetivo:** Elaborar los diseños de Casos de Prueba necesarios para las pruebas.
- ✓ **Responsable:** Diseñador de Casos de Prueba.
- ✓ **Artefactos de entrada.**

- Descripción de los Casos de Uso.

- ✓ **Artefactos que genera:**

- Diseños de Casos de Prueba.

- ✓ **Fase o etapa de aplicación:** Fase de Elaboración.
- ✓ **Descripción:**

Una vez elaborado el Plan de Pruebas se realizó el diseño de los Casos de Prueba, como segunda actividad dentro del proceso de pruebas, a partir de los Casos de Uso.

Para el caso específico de las Pruebas de Caja Blanca, el rol de Diseñador de Casos de Prueba fue asumido por los desarrolladores de cada uno de los módulos del producto.

Para la realización del Diseño de Casos de Prueba se tuvieron en cuenta los aspectos o secciones siguientes:

- Nombre del Documento. (Ejemplo: Diseño de Casos de Pruebas de Caja Blanca).
- Nombre del proyecto. (Ejemplo: Juegos CNeuro).
- Nombre del producto (Ejemplo: Venganza).
- Versión del producto. (Ejemplo: 1.0).
- Reglas de Confidencialidad: Se especificó el carácter o la clasificación de la información contenida en las páginas del documento.
- Descripción General. Se describió de manera general los aspectos a tener en cuenta para la realización de las pruebas.
- Funcionalidad a revisar.
 - Nombre de la funcionalidad.
 - Descripción de la funcionalidad.
 - Flujo Central.
 - Condiciones de Ejecución.
- Iteraciones. (Clases válidas e inválidas, resultados esperados, resultados de la prueba y observaciones).
- Registro de defectos y dificultades encontradas. (Elemento, Número, No conformidad, Aspecto correspondiente, etapa de detección, Importancia y recomendación).
- Anexos. Poner imágenes de los errores encontrados.

Los Diseños de Casos de Prueba fueron elaborados a partir de los Casos de Uso, a continuación se hace referencia a los mismos:

Pruebas de Unidad.

A continuación se muestra un ejemplo de los Casos de Prueba diseñados para las Pruebas de Caja Blanca, pertenecientes a las Pruebas de Unidad. Los elementos a ser probados durante estas pruebas son los siguientes:

✓ **Clase Level 1.**

- CPR 1 ActivateTask().
- CPR 2 CheckStayTask().
- CPR 3 AnimateTakeTask().

Los diseños de casos de Prueba restantes se encuentran en el Expediente del Proyecto de Juegos CNeuro. En el [Anexo 22](#) se puede observar el diseño para el CPR 1 ActivateTask().

Pruebas de Integración.

Los Casos de Uso a ser probados durante estas pruebas son los siguientes:

- CPR 1 Gestionar Perfil.
- CPR 2 Seleccionar Jugador.
- CPR 3 Cargar nivel.
- CPR 4 Mover Jugador.
- CPR 5 Gestionar tareas.

Los Diseños de Casos de Prueba para las pruebas de Integración se encuentran en el Expediente del Proyecto de Juegos CNeuro, en el [Anexo 23](#) y [Anexo 24](#) se muestran el diseño de los CPR Gestionar Perfil y CPR Seleccionar Jugador respectivamente.

Pruebas de Sistema.

Los Casos de Uso a ser probados durante estas pruebas son los siguientes:

- CPR 1 Gestionar Perfil.

- CPR 2 Seleccionar Personaje.
- CPR 3 Gestionar Tareas.
- CPR 4 Manipular sonido.
- CPR 5 Cargar nivel.
- CPR 6 Iniciar juego.
- CPR 7 Jugar Básico Nivel 2.
- CPR 8 Mover Jugador.
- CPR 9 Jugar Básico nivel 1.
- CPR 10 Jugar Bajo Presión nivel 1.Mover Jugador.

Los Diseños de Casos de Prueba para las pruebas de Sistema se encuentran en su totalidad en el Expediente del Proyecto de Juegos CNeuro, en el [Anexo 25](#) y [Anexo 26](#) se aprecian los CPR Manipular Sonido y CPR Cargar Nivel respectivamente.

Ejecución de las Pruebas.

9.1 Aplicar Pruebas de Unidad.

Aplicar estas pruebas para comprobar los componentes implementados. Se utilizará para su puesta en práctica la técnica descrita a continuación:

Pruebas de Caja Blanca.

- ✓ **Objetivo:** Comprobar la calidad del código, de cada módulo.
- ✓ **Responsable:** Probador.
- ✓ **Artefactos de entrada:**
 - Diseño de Casos de Pruebas de Caja Blanca para Juegos CNeuro.
- ✓ **Artefactos que genera:**
 - Registro de No Conformidades.

✓ **Fase o etapa de aplicación:** Fase de Elaboración. Durante la segunda iteración de esta fase, ya que para este momento se pretende lograr una buena interacción entre los módulos.

✓ **Descripción:**

Se debe aplicar el Diseño de Casos de Prueba realizado, teniendo en cuenta la norma ISO 9126: "Ingeniería de Software. Calidad del producto. Parte 1: Modelo de Calidad" y la ISO 12207: "Procesos del ciclo de vida del Software".

Los autores de esta investigación consideran importante resaltar que en el rol de Probador, para esta actividad, debe ser asumido por los desarrolladores de cada uno de los Grupos de Desarrollo, pertenecientes a los tres módulos del proyecto. Para lograr de esa manera, que las pruebas, arrojen el mayor número de errores posibles.

Las deficiencias encontradas deben ser recogidas en un documento de No Conformidades.

9.2 Aplicar Pruebas de Integración.

Aplicar estas pruebas, para verificar que los componentes interaccionan entre sí de la forma correcta. Se utilizará para su aplicación el tipo de pruebas, descrito a continuación:

Pruebas Incrementales ascendentes.

✓ **Objetivo:** Evaluar la integración de los módulos de manera progresiva.

✓ **Responsable:** Probador.

✓ **Artefactos de entrada:**

- Diseño de Casos de Prueba.

✓ **Artefacto que genera:**

- Registro de No Conformidades.

✓ **Fase o Etapa de aplicación:** Fase de Construcción.

✓ **Descripción:**

Se combinarán los módulos de manera ascendente para de esta manera detectar los errores en la interacción entre los mismos. Las deficiencias encontradas deben ser recogidas en un documento de No Conformidades.

9.3 Aplicar Pruebas de Sistema.

Aplicar estas pruebas, para probar el sistema como un todo. Para esto se utilizará como técnica la descrita a continuación:

Pruebas de caja Negra.

✓ **Objetivo:** Comprobar la funcionalidad del producto en su primera versión.

✓ **Responsable:** Probador.

✓ **Artefactos de entrada.**

- Diseño de Casos de Prueba de Caja Negra.

✓ **Artefactos que genera:**

- Registro de No Conformidades.

✓ **Fase o etapa de aplicación:** Segunda iteración de la fase de Construcción.

✓ **Descripción:**

Se debe aplicar el Diseño de Casos de Prueba realizado, para el cual se tuvo en cuenta la norma ISO 9126: "Ingeniería de Software. Calidad del producto. Parte 1: Modelo de Calidad", para probar las funcionalidades del sistema.

Las deficiencias encontradas serán recogidas en un documento de No Conformidades.

Evaluación y Resumen de las Pruebas.

✓ **Objetivo:** Evaluar el proceso de Pruebas.

✓ **Responsable:** Administrador de Calidad.

✓ **Artefactos de entrada.**

- Resultados de las Pruebas.

✓ **Artefactos que genera:**

- Minuta de Reunión.

✓ **Fase o etapa de aplicación:** Al finalizar la aplicación de cada una de las pruebas planificadas.

✓ **Descripción:**

Una vez concluidas las pruebas realizadas al producto se debe realizar una Reunión de resumen del proceso de pruebas, que debe ser convocada y planificada por el líder del proyecto y debe estar presente el equipo que realizó las pruebas.

Esta actividad se realiza al término de cada una de las pruebas aplicadas y en la misma el Administrador de Calidad evalúa los resultados documentados por el probador y compara estos resultados con los objetivos definidos en los Planes de Prueba establecidos. En caso de que los objetivos no sean cumplidos, los procedimientos y Casos de Pruebas deben ser redefinidos, para el logro de los mismos.

Es responsabilidad del Administrador de Calidad, documentar este proceso de evaluación y resumen de las pruebas, y generar al final de esta reunión un documento como constancia del fin del proceso, que debe contener aspectos como los participantes en el proceso de evaluación y resumen de las pruebas (participantes en la reunión realizada), los puntos tratados y los acuerdos tomados. Se recomienda utilizar la Plantilla Minuta de Reunión, que se encuentra en el Expediente del Proyecto de Juegos CNeuro.

Los autores de este trabajo de diploma consideran que los resultados alcanzados son satisfactorios, a pesar de que la estrategia diseñada no pudo ser aplicada desde el comienzo del proyecto. Se realizaron las actividades con el máximo de rigor posible, de acuerdo a las condiciones del producto y el proceso en el momento de la incorporación, quedando planificadas las actividades de pruebas que deben ser aplicadas al software para su completo desarrollo y terminación.

Conclusiones.

El proyecto Juegos CNeuro de la Facultad 5 de la UCI, desarrolla un producto de Juegos para niños que padecen de Discalculia. Para ello se planificó un proceso de desarrollo que se ve afectado por no contar con una estrategia que garantice la calidad del proceso y el producto que en este proyecto tienen lugar.

El objetivo de plantear estrategias de Aseguramiento de la Calidad es ayudar a los Ingenieros de Software a estandarizar los procesos de desarrollo, logrando de esta manera una mayor eficiencia en la producción de software. Durante el desarrollo de este trabajo de diploma se realizó un análisis del área de procesos Aseguramiento de la Calidad del Proceso y el Producto perteneciente al Modelo de Madurez y Capacidad Integrado, lo cual permitió diseñar y aplicar una estrategia para el Aseguramiento de la Calidad del Proyecto Juegos CNeuro, dando cumplimiento así, al objetivo planteado en esta investigación. Se definió, además, como parte de esta estrategia, el Proceso de Pruebas a aplicar al producto que se desarrolla en el proyecto para chequear que el mismo satisfaga las necesidades planteadas por el cliente.

A medida que se fue aplicando la estrategia en el proyecto se registraban los resultados obtenidos, esto permitió validar la eficiencia e importancia de la solución propuesta en esta tesis que sin duda contribuirá a elevar la eficacia del proceso de desarrollo de software empleado en el proyecto Juegos CNeuro.

Recomendaciones.

- ✓ Terminar de aplicar en Juegos CNeuro las actividades de prueba planificadas que asegurarán la calidad en la terminación del producto que se desarrolla.
- ✓ Aplicar la estrategia planteada a próximas versiones del producto.

Bibliografía.

1. **Torregrosa Sánchez, Rafael.** Calidad. Concepto y Generalidades. [En línea] [Citado el: 25 de 06 de 2008.] http://chgubv.san.gva.es/Descargas/Gerencia/PlaniCalidad/Kiosco/Calidad_generalidades.pdf.
2. **Pressman, Roger S.** *Ingeniería de Software. Un Enfoque Práctico.* [ed.] Mc Graw Hill. 2002.
3. **ISO.** *Administración y Aseguramiento de la Calidad.* 1994.
4. —. *NC-ISO 9000:"Sistema de Gestión de la Calidad. Fundamentos y Vocabulario".* 2005.
5. **Jurán, Joseph M.** Autores de la Gestión de la Calidad|GestioPolis. [En línea] 1993. [Citado el: 25 de 06 de 2008.] <http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/aucalid.htm>.
6. **Deming, Edward.** Autores de la Gestión de la Calidad|GestioPolis. [En línea] 1989. [Citado el: 25 de 06 de 2008.] <http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/aucalid.htm>.
7. **Crosby, Philips.** Autores de la Gestión de la Calidad|GestioPolis. [En línea] 1979. [Citado el: 25 de 06 de 2008.] <http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/aucalid.htm>.
8. **Software Engineering, Institute.** *Capability Maturity Model Integration v1.1.* . Marzo 2002.
9. **Scalone, Fernanda.** *Estudio Comparativo de los Modelos y Estándares de Calidad del Software.* Buenos Aires : s.n., Junio 2006. Tesis Doctoral.
10. **Moreno García, María N., García Peñalvo, Francisco J. y Polo Martín, María José.** Medición de la calidad del software en el ámbito de la especificación de Requisitos. [En línea] 2000. [Citado el: 25 de 06 de 2008.]
11. **ISO.** *ISO 9000: "Sistema de Gestión de la Calidad".* 2001.
12. —. *NC ISO/IEC 9126-1:"Modelo de Calidad".* . 2005.
13. —. *ISO 9000:"Sistema de Gestión de la Calidad".* 2000.
14. **Project Management, Institute.** *Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK)* . Tercera Edición. 2004.
15. **ISO.** *NC ISO/IEC 90003: "Guía para la Aplicación de la ISO 9000 al Software de Computación".* 2004.
16. **Humphrey.** 1989.
17. **Kan.** 1995.
18. **Napal, Irina.** *Las Pruebas de Software. Su Aplicación al Config.Case.* Ciudad de La Habana : s.n., Junio 2003.

19. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* . 2000.
20. **ISO.** *ISO/FDIS 9001:"Sistema de Gestión de la Calidad. Requisitos".* 2000.
21. **Guerrero, L.A.** Taller UML. [En línea] [Citado el: 25 de 06 de 2008.]
22. **Febles, Ailyn.** *Un modelo de Referencia para la Gestión de Configuración en la PYME de Software.* . Ciudad de La Habana. : s.n., 2003. Tesis Doctoral.
23. **Perez Moya, Osiris.** *Entregables que garantizan la calidad del producto en el proceso de producción de software educativo.* Ciudad de La Habana : s.n., Mayo, 2007.
24. **Software, R.** *Rational Rose Enterprise Edition.* . 2003.
25. **IBM Rational, S.C.** *Rational Unified Process. Help.* . 2003.
26. **García, Piattini.** *Calidad en el Desarrollo y Mantenimiento del Software.* s.l. : Madrid : RA-MA, 2003.
27. **García, A. M. y N. Aragón** Análisis de modelos de calidad internacionales con respecto a su aplicación a la industria cubana del software Revista Cubana de Ciencias, 2006.
28. Mercado Digital [En línea] [Citado el: 27 de 06 de 2008.]
http://www.mercado.com.ar/mercado/vercanal_nota.asp?id=356776
29. **Water Rodriguez, Rubier.** *Aseguramiento de la calidad en el proyecto "Sistema Integrado de Gestión Bibliotecaria para la Biblioteca Nacional José Martí".* Ciudad de La Habana : s.n., Junio 2007.
30. **Buades.** 2002
31. Centro de Neuro Ciencias. [En línea] [Citado el: 25 de 06 de 2008.]
<http://www.cneuro.co.cu/?q=es/cnch>.

Anexos.

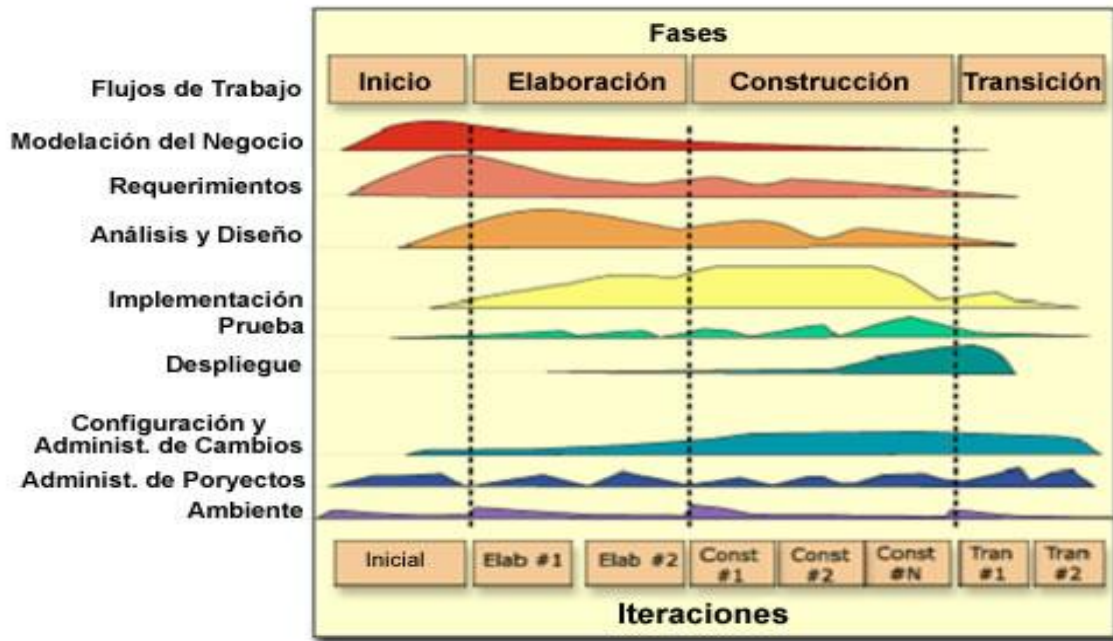
Anexo1. Factores. Satisfacción del Cliente.

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

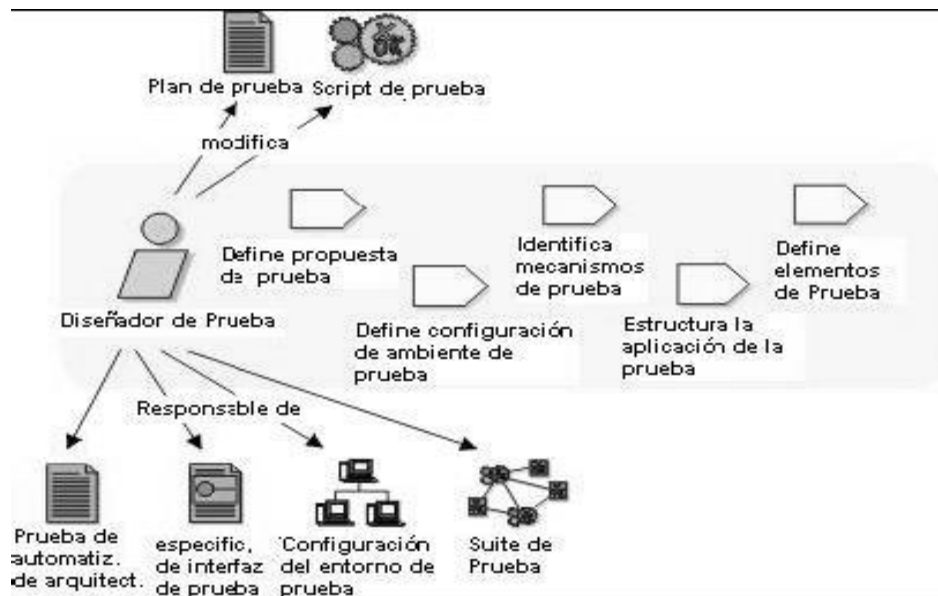
Anexo 2. Mejores Prácticas definidas por RUP.



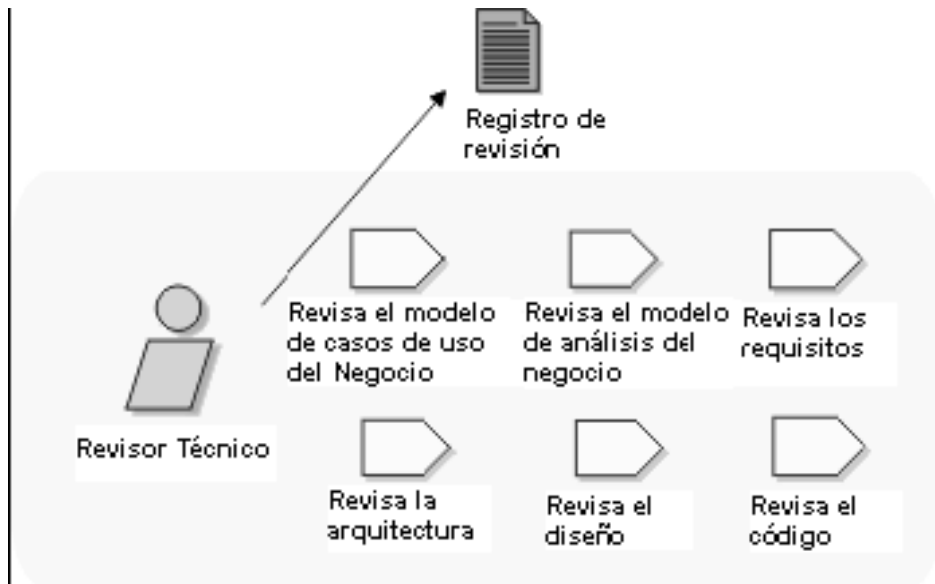
Anexo 3. Fases y Flujos de Trabajo en RUP.



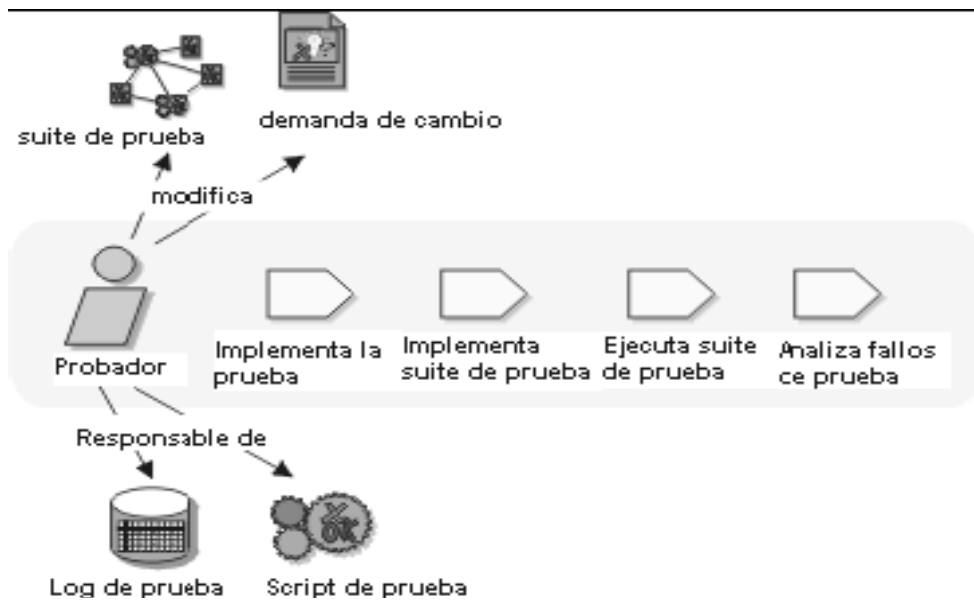
Anexo 4. Responsabilidades para el Rol de Diseñador de Pruebas.



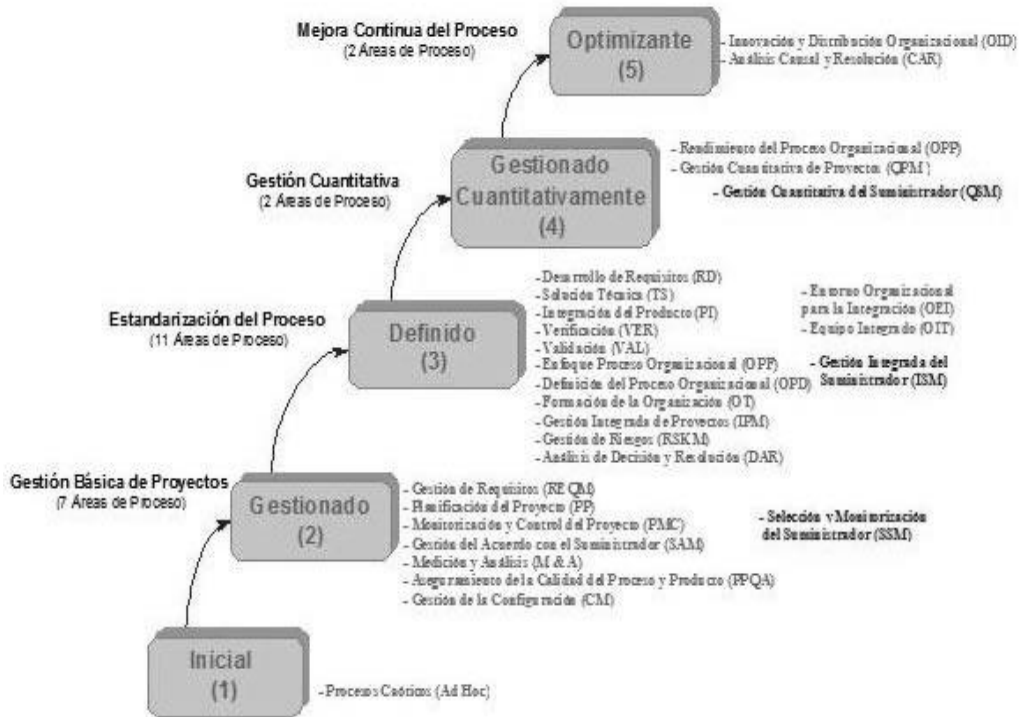
Anexo 5. Responsabilidades para el Rol de Revisor Técnico.



Anexo 6. Responsabilidades para el Rol de Probador.



Anexo 7. Niveles de CMMI.



Anexo 8. Imágenes del Centro de Neurociencias de Cuba.



Anexo 9. Roles y responsabilidades de Juegos CNeuro.

Rol.	Responsabilidades.
Líder del proyecto.	<ul style="list-style-type: none"> ✓ Dirige a los desarrolladores y les orienta las tareas a realizar. ✓ Chequea el trabajo que ya se ha concluido. ✓ Elabora el plan de capacitación y evalúa al personal en las distintas tareas. ✓ Desarrolla reuniones de información y control del proyecto. ✓ Informa a los miembros del proyecto y a instancias superiores sobre el estado actual del mismo. ✓ Participa en la selección del grupo de proyecto.
Jefe de grupo de desarrollo.	<ul style="list-style-type: none"> ✓ Dirige un grupo de trabajo encargado de realizar un módulo o cualquier otra tarea asignada a ese grupo.
Implementador.	<ul style="list-style-type: none"> ✓ Es el encargado de generar el código, a partir de lo que hagan los analistas y diseñadores.
Analista.	<ul style="list-style-type: none"> ✓ Descripción de los casos de usos. ✓ Extracción de los requisitos del sistema.
Documentador.	<ul style="list-style-type: none"> ✓ Trabaja en la actualización del expediente del proyecto.
Arquitecto.	<ul style="list-style-type: none"> ✓ Responsable de la arquitectura del software. ✓ Decisiones técnicas más importantes en cuanto a las restricciones del diseño global e implementación del proyecto.
Trabajo educativo.	<ul style="list-style-type: none"> ✓ Responsable de velar por los resultados docentes de los estudiantes del proyecto así como de su comportamiento en las tareas del mismo. ✓ Organiza actividades entre los integrantes del proyecto.
Responsable de de Gestión Configuración.	<ul style="list-style-type: none"> ✓ Planificar y chequear el proceso de gestión de configuración (Plan de Gestión de Configuración). ✓ Responsable del sistema de gestión de información, salvadas, etc.
Responsable 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.

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

	<ul style="list-style-type: none"> ✓ 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. ✓ Guiar las auditorias que se realice. ✓ 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.
Psicólogo.	<ul style="list-style-type: none"> ✓ Debe velar por la estética del producto para que el mismo se adecúe al personal a quien va dirigido.

Anexo 10. Normas y Estándares utilizados.

Estándares y Normas.
LMC de Calidad de la UCI. (Ver 5.3)
CMMI Integración del Modelo de Capacidad y Madurez.
ISO 9001:2000 Sistema de Gestión de la Calidad. Requisitos.
ISO 9126 Métricas de Software.
ISO 12207 Procesos del ciclo de vida del software.
ISO 90003:2004 Guía para la aplicación de la norma ISO 9000:2000 al software de computación.
IEEE 828: Estándar para planes del manejo de las configuraciones de software.
IEEE 829 Estándar para la documentación de las pruebas del software.
IEEE 830 Especificaciones de los requisitos del software.
IEEE 1016 Prácticas para la descripción del diseño de software.
IEEE 1471 Práctica recomendada para la descripción arquitectónica de sistemas Software-Intensivos.

Anexo 11. Lista de Chequeo. Revisión Inicial.

Artefacto: Documento Visión.					
No.	Criterio de Evaluación.	S	N	NP	Observaciones.
1.	¿Se encuentra elaborado el documento visión?	X			
2.	¿Está elaborado utilizando la plantilla propuesta en el EP?	X			
3.	¿Están completamente documentados todos los aspectos de la plantilla?		X		
4.	¿Se discutió la propuesta de visión del proyecto con todos los miembros del equipo de desarrollo?	X			
Artefacto: Lista de Riesgos.					
5.	¿Están definidos los riesgos del proyecto?	X			
6.	¿Está elaborado utilizando la plantilla propuesta en el EP?	X			
7.	¿Están completamente documentados todos los aspectos de la plantilla?		X		
Artefacto: Plan de Mitigación de Riesgos.					
8.	¿Se encuentra elaborado el Plan de Mitigación de Riesgos del proyecto?	X			
9.	¿Está elaborado utilizando la plantilla propuesta en el EP?	X			
10.	¿Están completamente documentados todos los aspectos de la plantilla?		X		
Artefacto: Propuesta de Roles.					
11.	¿Se encuentran definidos todos los roles necesarios para el proyecto?		X		
12.	¿Se definen las responsabilidades a cada uno de los roles definidos?	X			
13.	¿El documento presenta buena redacción y Ortografía?	X			

Anexo 12. No Conformidades. Revisión Inicial.

Elemento	No	No Conformidad	Etapa de detección	Importancia	Recomendación
Propuesta de Roles	1.	Se definen actividades de calidad insuficientes para asegurar el proceso y el producto dentro del proyecto.	Revisión	X	Definir otras actividades de aseguramiento, además de las propuestas.
Propuesta de Roles	2.	El rol definido (Responsable de Calidad) no satisface las necesidades del proyecto en cuanto al tema de Aseguramiento de la Calidad ya que se le asocian demasiadas actividades para un solo rol.	Revisión	X	Redefinir los roles. Propuesta: - Definir equipo de calidad.
Propuesta de Roles	3.	No se define el Rol de Diseñador dentro del documento de Roles definido.	Revisión	X	Definir rol de Diseñador, ver roles propuestos por RUP y roles propuestos por la Dirección de Calidad de SW.
Documento visión	4.	No se ha terminado de elaborar el documento visión por parte del líder del proyecto.	Revisión	X	Completar documento visión. Tener en cuenta en próxima revisión.
Documento visión	5.	Llenar tabla de control de versiones.	Revisión	X	Llenar tabla de control de versiones que se encuentra en el documento visión de Juegos CNeuro.
Documento visión	6.	No se especifican los competidores del producto.	Revisión	X	Especificar competidores del producto.

Documento visión	7.	No se han concluido los aspectos siguientes: Falta por terminar el: - Perfil de los involucrados. - Perfil de los usuarios. - Principales necesidades de los Involucrados y los usuarios. - Resumen de capacidades.		X	Terminar de especificar los datos señalados.
Plan de Mitigación de Riesgos.	8.	No se han terminado de llenar las secciones siguientes: - Organización y Responsabilidades. - Presupuesto. - Herramientas y Técnicas.	Revisión	X	Terminar de llenar las secciones señaladas.
Plan de Mitigación de Riesgos.	9.	En las reglas de Confidencialidad no se especifica el proyecto o empresa involucrada.	Revisión	X	Especificar.
Lista de Riesgos.	10.	En las reglas de Confidencialidad no se especifica el proyecto o empresa involucrada.	Revisión	X	Especificar.

Anexo 13. Lista de Chequeo. Revisión de los Requisitos.

Artefacto: Requerimientos.					
No.	Criterio de Evaluación.	S	N	NP	Observaciones.
1.	¿Está el requisito claramente definido? ¿Puede interpretarse mal?	X			
2.	¿El requisito es correcto?	X			
3.	¿El requisito es inequívoco?	X			
4.	¿El requisito es completo?	X			
5.	¿El requisito tiene su naturaleza definida?	X			

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

6.	¿El requisito incumple alguna restricción definida?	X			
7.	¿El requisito es comprobable o verificable?	X			
8.	¿El requisito es modificable?	X			
9.	¿El requisito es identificable?	X			
10.	¿Se puede seguir el requisito en el modelo del sistema que hemos desarrollado?	X			
11.	¿Se puede localizar el requisito en el conjunto de objetivos del sistema/producto?	X			
12.	¿El documento presenta buena redacción y Ortografía?	X			

Anexo 14. Lista de Chequeo. Diagrama de Casos de Uso.

Artefacto: Modelo de Casos de Uso del Sistema.					
No.	Criterio de Evaluación.	S	N	NP	Observaciones.
Compleitud.					
1.	¿Han sido considerados todos los sistemas externos con los cuáles interactuará el sistema?	X			
2.	¿Se presenta una descripción resumida de todos los casos de uso del negocio?	X			
3.	¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?	X			
4.	¿Se presenta una descripción detallada (descripción extendida esencial) de todos los casos de uso del negocio?	X			
Consistencia.					
5.	¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?	X			
6.	¿Las acciones en el flujo de eventos están correctamente asignadas corresponde?	X			
7.	¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos?	X			
8.	¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema claramente identificable?	X			

9.	¿Existen casos de uso en los que intervengan más de un actor?			X	
10.	¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos?	X			
Correctitud.					
11.	¿Existe para cada caso de uso de negocio por lo menos un usuario responsable?	X			
12.	¿Se ajusta la representación del diagrama del caso de uso de acuerdo a lo normado en la metodología?	X			
Complejidad.					
13.	¿Se ha realizado una agrupación de los casos de uso en paquetes?		X		
14.	¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?	X			
15.	¿El documento presenta buena redacción y Ortografía?		X		

Anexo 15. No Conformidades. Revisión al Diagrama de Casos de Uso.

Elemento	No	No conformidad	Etapas de detección	Importancia	Recomendación
Modelo de casos de uso	1.	Falta de ortografía en la palabra manipular (Ver Anexo 3.1).	Revisión	X	Corregir.
Modelo de casos de uso	2.	Falta de ortografía en la palabra animación (Ver Anexo 3.2).	Revisión	X	Corregir.
Modelo de casos de uso	3.	Los casos de uso no están agrupados en paquetes.	Revisión	X	Agrupar en paquetes los Casos de uso.

Anexo 16. Lista de Chequeo. Revisión de la Arquitectura.

Artefacto: Arquitectura.					
No.	Criterio de Evaluación.	S	N	NP	Observaciones.
1.	¿Se realizaron estudios para identificar patrones de arquitectura ajustables al proyecto?				
2.	¿Se aplican patrones de arquitectura ajustables al proyecto?				
3.	¿Se priorizan los Casos de Uso arquitectónicamente significativos?	X			
4.	¿El modelo de arquitectura cumple con los objetivos pactados?				
5.	¿Se realizaron estudios para identificar herramientas apropiadas para el desarrollo del SW?				
6.	¿Se aplican herramientas apropiadas para el desarrollo del SW?				
Descripción Arquitectónica (AD).					
7.	<p>¿La AD contiene la siguiente información referente a la AD como un todo?</p> <p>a) Fecha de expedición y estado. b) Organización que la expide. c) Historial de cambios. d) Sumario. e) Alcance. f) Contexto. g) Glosario. h) Referencias.</p>	X			El contenido de esta información y el nivel de detalle, es determinado por el proyecto. Aunque es recomendable usar la estructura propuesta, ya que es más explícita.
Identificación de los stakeholders y sus intereses.					
8.	¿Se identifican los usuarios del sistema?				
9.	¿Se identifican los adquiridores del sistema?		X		
10.	¿Se identifican los desarrolladores del sistema?		X		
11.	¿Se identifican los mantenedores del sistema?		X		
12.	¿Se define el propósito o misión del sistema?	X			

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

13.	¿Se establecen las propiedades del sistema, para su uso, en la realización de sus misiones?		X		
14.	¿Se define la viabilidad de construir el sistema?		X		
15.	¿Se definen los riesgos del desarrollo del sistema y las operaciones para los usuarios, adquiridores y desarrolladores del sistema?		X		
16.	¿Se define la capacidad de mantenimiento, despliegue y evolución del sistema?		X		
Puntos de Vista y Vistas de la Arquitectura.					
17.	¿Se definen puntos de vista?		X		
18.	¿Se especifica para cada punto de vista lo siguiente? a) Nombre del punto de vista. b) Los stakeholders a ser tratados según el punto de vista. c) Las preocupaciones a ser tratadas según el punto de vista. d) El lenguaje, técnicas de modelado o métodos analíticos empleados construir una vista basada en el punto de vista. e) La fuente (podría incluir al autor, la fecha, la referencia a otros documentos; según determine la organización).		X		
19.	¿Cada vista definida corresponde con exactamente un punto de vista de los seleccionados anteriormente?		X		
20.	¿Cada vista se realiza conforme a las especificaciones de su punto de vista correspondiente?		X		

21.	¿Cada vista incluye los elementos siguientes? a) Un identificador y otra información introductoria, definidos por la organización. b) Representación del sistema construido con el lenguaje, métodos, técnicas analíticas y de modelados asociadas con la vista. c) Información de la configuración definida por la organización.		X		
22.	¿El documento presenta buena redacción y Ortografía?	X			

Anexo 17. No Conformidades. Revisión de la Arquitectura.

Elemento	No	No conformidad	Etapas de detección	Importancia	Recomendación
Identificación de los stakeholders y sus intereses.	1.	No se identifican los usuarios, adquiridores, desarrolladores y mantenedores del sistema.	Flujo de trabajo de análisis y diseño.	X	Definir los interesados y relacionados de alguna forma con el sistema.
Identificación de los stakeholders y sus intereses.	2.	No se define la viabilidad de construcción del sistema.	Flujo de trabajo de análisis y diseño.	X	Fundamentar si el sistema que se está por construir es viable y por qué.
Identificación de los stakeholders y sus intereses.	3.	No se definen los riesgos del desarrollo del sistema y las operaciones para los usuarios, adquiridores y	Flujo de trabajo de análisis y diseño.	X	Especificar los riesgos de construcción o desarrollo del sistema así

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

		desarrolladores del sistema.			como las obligaciones y/u operaciones de los involucrados.
Identificación de los stakeholders y sus intereses.	4.	No se define la capacidad de mantenimiento, despliegue y evolución del sistema.	Flujo de trabajo de análisis y diseño.	X	Definir la capacidad del sistema para su mantenimiento, despliegue y evolución.
Puntos de Vista y Vistas de la Arquitectura.	5.	No se definen puntos de vista arquitectónicos.	Flujo de trabajo de análisis y diseño.	X	Definir puntos de vista arquitectónicos.
Puntos de Vista y Vistas de la Arquitectura.	6.	No se especifica en los puntos de vista la estructura: f) Nombre del punto de vista. g) Los stakeholders a ser tratados según el punto de vista. h) Las preocupaciones a ser tratadas según el punto de vista. i) El lenguaje, técnicas de modelado o métodos analíticos empleados construir una vista basada en el punto de vista.	Flujo de trabajo de análisis y diseño.	X	Especificar la estructura planteada para los puntos de vista.
Puntos de Vista y Vistas de la Arquitectura.	7.	Cada vista definida no se corresponde con exactamente un punto de vista de los seleccionados anteriormente.	Flujo de trabajo de análisis y diseño.	X	Hacer corresponder cada vista definida con exactamente un punto de vista de los definidos anteriormente.
Puntos de Vista y Vistas de la Arquitectura.	8.	Cada vista no se realiza conforme a las especificaciones de su punto de vista correspondiente	Flujo de trabajo de análisis y diseño.	X	Realizar vista conforme a las especificaciones del punto de vista correspondiente

Puntos de Vista y Vistas de la Arquitectura.	9.	Cada vista no incluye los siguientes: d) Un identificador y otra información introductoria, definidos por la organización. e) Representación del sistema construido con el lenguaje, métodos, técnicas analíticas y de modelados asociadas con la vista. f) Información de la configuración definida por la organización.	Flujo de trabajo de análisis y diseño.	X	Incluir los elementos definidos en cada vista.
----------------------------------------------	----	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------	---	------------------------------------------------

Anexo 18. Lista de Chequeo. Revisión del Diseño.

Artefacto: Diseño de Juegos CNeuro.					
No	Criterios de evaluación	S	N	NP	Observaciones
1	¿Se realizó alguna DDS en Juegos CNeuro?	X			
2	¿Se realizó el diseño de las entidades?	X			
3	¿Están especificados todos los requerimientos en las entidades de diseño?	X			
4	¿El diseño está correcto, consistente con los requisitos y muestra una adecuada trazabilidad hacia los mismos?	X			
5	¿El diseño puede ser elaborado partiendo de los requisitos?	X			
6	¿Se realizó la selección y descripción de los atributos?	X			
7	¿Se cuenta con alguna vista de diseño en el proyecto?		X		
8	Si existe alguna vista ¿que tipos de vista de diseño se utiliza?		X		

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

9	¿Las vistas de diseño cumplen con los requerimientos de alcance, uso y representación?	X		
10	¿El diseño implementa una secuencia correcta de eventos, entradas, salidas, interfaces, flujo lógico?	X		
11	¿Se cumplieron las actividades de Análisis y Diseño?	X		
12	¿Están documentados los artefactos de Análisis y diseño?	X		
13	¿Se realizó y documentó un diseño detallado de las interfaces?	X		
14	¿Está actualizada la documentación del usuario por parte del desarrollador?	X		
15	¿El documento presenta buena redacción y Ortografía?	X		

Anexo 19. No Conformidades. Revisión del Diseño.

Elemento	No	No conformidad	Etapas de detección	Importancia	Recomendación
Modelo de Diseño	1.	No se cuenta con ninguna vista de diseño en el proyecto.	Revisión	X	Elaborar las vistas de diseño con los aspectos necesarios.
Modelo de Diseño	2.	No se cumplieron todas las actividades de Análisis y Diseño.	Revisión	X	Cumplir las actividades de Análisis y Diseño restantes.
Modelo de Diseño	3.	Los artefactos del análisis y Diseño no se encuentran documentados.	Revisión	X	Documentar los artefactos indocumentados.
Modelo de Diseño	4.	No se realizó y documentó un diseño detallado de las interfaces.	Revisión	X	Realizar y documentar un diseño detallado de las interfaces.

Anexo 20. Lista de Chequeo. Revisión a la Gestión de la Configuración.

Artefacto: Gestión de la Configuración.					
No	Criterio de evaluación	S	N	NP	Observaciones
1.	¿Se elaboró un plan de Gestión de la Configuración desde la etapa inicial del proyecto?		X		
2.	¿El plan tiene descrito las actividades de gestión de la configuración?	X			
3.	¿En el plan se destaca el cronograma para ejecutar estas actividades?		X		
4.	¿Se especifican los responsables de ejecutar estas actividades?	X			
5.	¿Se hace identificación de los artículos de software y de las versiones que serán controladas por el proyecto?	X			
6.	¿Para cada artículo de software y sus versiones se han especificado las referencias de versión y la documentación que lo identifica?	X			
7.	¿Se esta llevando a cabo el control de la configuración?	X			
8.	¿Están identificadas y registradas las solicitudes de cambio?	X			
9.	¿Se ejecuta el análisis y la evaluación de la solicitud de cambios?	X			
10.	¿Se ejecuta y registra el proceso de aprobación o desaprobación de la solicitud?	X			
11.	¿Se ejecuta y registra el proceso de verificación y liberación del producto modificado?		X		
12.	¿Se realiza la contabilidad del estado de la configuración?				
13.	¿Se almacenan en un registro el estado y el historial de los artículos de software controlados?	X			

14.	¿Los informes constan de la siguiente información? - Número de cambios al proyecto. - Últimas versiones del artículo de software. - Los identificadores de la liberación. - El número de liberaciones. - Las comparaciones entre liberaciones.	X		
15.	¿Se verifica la integridad funcional del artículo de software de acuerdo con los requisitos?	X		
16.	¿Se realiza una apropiada gestión de la liberación y entrega?	X		
17.	¿Se conservan las copias maestras del código y la documentación?	X		
18.	¿El documento presenta buena redacción y Ortografía?	X		

Anexo 21. No Conformidades. Revisión a la Gestión de la Configuración.

Elemento	No	No conformidad	Etapas de detección	Importancia	Recomendación
Plan de Gestión de la Configuración	1	No se elaboró el plan de GC desde el inicio del proyecto.	Revisión	X	Realizar los documentos en tiempo.
Plan de Gestión de la Configuración	2	No aparece en el plan el cronograma de las actividades de GC.	Revisión	X	Agregar el cronograma al Plan de GC
Gestión de la liberación y entrega.	3	No se registra el proceso de liberación del producto una vez modificado.	Revisión	X	Se debe registrar en un documento del proyecto.
Contabilidad del estado de la configuración.	4	Los informes no constan de toda la información necesaria.	Revisión	X	Se debe redactar los informes con la siguiente información: - Número de cambios al proyecto. - Últimas versiones del artículo de

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

					software. - Los identificadores de la liberación. - El número de liberaciones. - Comparaciones entre liberaciones.
--	--	--	--	--	-----------------------------------------------------------------------------------------------------------------------------

Anexo 22. CPR. ActivateTask().

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
La tarea se activa.		Debe hacerse nítida, o sea que se vea correctamente.		
	La tarea no se activa	La tarea se mantiene con el estado visual e interno igual.		

Anexo 23. CPR. Gestionar Perfil.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El usuario elige la opción de crear un perfil		El sistema muestra una interfaz diseñada para la captación de los datos del nuevo perfil.		
	El usuario entra datos inválidos para llenar su perfil.	El sistema muestra un mensaje de error y da la oportunidad de volver entrar los datos al sistema.		
El usuario elige salvar los datos de su perfil, y luego cuando el		El sistema verifica si el usuario desea salvar el perfil. El sistema crea un nuevo perfil. Los datos		

sistema le hace la pregunta de confirmación él confirma dicha acción.		del perfil son salvados en ficheros XML.		
	El usuario elige salvar su perfil sin haber llenado todos los campos de información necesarios.	El sistema muestra un mensaje de error y da la posibilidad de ingresar los datos nuevamente.		
El usuario elige salvar los datos de su perfil, pero cuando el sistema le lanza la pregunta de confirmación de la acción el usuario responde que NO acepta que los datos sean salvados.		El sistema no crea el perfil, y termina la funcionalidad.		
El usuario elige modificar información de su perfil.		El sistema pide los datos primarios de su perfil. El sistema carga el perfil.		
El usuario modifica los datos de su perfil.		El sistema verifica que el usuario desea salvar los datos modificados.		
El usuario confirma que desea salvar las modificaciones.		El sistema modifica la información el perfil, y lo almacena.		
El usuario confirma que no que		El sistema no modifica la información el perfil, y		

desea salvar las modificaciones.		termina la funcionalidad.		
El usuario elige la opción de eliminar un perfil.		El sistema pide datos primario de su perfil El sistema busca el fichero del perfil del usuario y lo selecciona. El sistema envía un mensaje al usuario para confirmar si realmente desea borrar su perfil.		
El usuario confirma sus deseos de borrar su perfil.		El sistema borra el perfil seleccionado. El sistema le envía un mensaje al usuario confirmando que el perfil ha sido borrado.		
El usuario no confirma sus deseos de borrar el perfil.		El sistema no borra el perfil seleccionado y finaliza la funcionalidad.		

Anexo 24. CPR. Seleccionar Jugador.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El usuario desea seleccionar un personaje.		El sistema carga de ficheros de memoria todos los personajes del juego.		
El usuario seleccionar el personaje con el que desea jugar.		El sistema muestra todas las características del personaje y verifica si el usuario desea jugar con este personaje.		
El usuario confirma que desea jugar con este personaje.		El sistema salva en su perfil de usuario el personaje seleccionado y termina la funcionalidad.		

Autores: Eniel Corzo Rodríguez y Pedro Jorge Camero Cárdenas

El usuario no acepta jugar con este personaje.		El sistema le da la oportunidad al usuario de volver a escoger un personaje.		
------------------------------------------------	--	------------------------------------------------------------------------------	--	--

Anexo 25. CPR. Manipular Sonido.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El actor desea Inicializar los sonidos.		El sistema inicializa la biblioteca de sonido.		
Se levanta la interfaz de inicio del juego.		Se cargan los sonidos de la interfaz.		
Se ejecuta un evento de interfaz.		Se reproduce el sonido asociando a ese evento.		
Se selecciona el nivel a cargar.		Se cargan todos los sonidos asociados a ese nivel. Se reproducen los sonidos ambientales.		
Se produce un efecto especial.		Se reproducen los sonidos asociando al efecto especial mostrado.		
Se ejecuta una animación.		Se reproducen el sonido asociado a la animación.		
Se finaliza la aplicación.		Se vacía el buffer de sonidos.		
Se cambia de nivel.		Se quitan del buffer los sonidos asociados al nivel cerrado. Se cargan los sonidos asociados al nivel seleccionado.		

Anexo 26. CPR. Cargar Nivel.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El usuario indica el nivel que desea jugar.		El sistema busca el archivo correspondiente al nivel indicado. El sistema carga todas las escenas referentes al nivel seleccionado.		
El usuario indica iniciar un nivel que no existe en el juego.		El sistema envía un mensaje de error y se termina la funcionalidad.		

Glosario de Términos.

C

CPR: Casos de Prueba.

D

Descripción Arquitectónica (AD): Conjunto de elementos que se van a utilizar para documentar la arquitectura del sistema.

E

Entorno 3D: Entorno virtual en tercera dimensión.

R

Realidad Virtual: Tecnología que integra soluciones de software y hardware que le permiten al usuario interactuar con entornos virtuales que simulan los comportamientos físicos y lógicos del mundo real.

S

Stakeholders: Implicados con el sistema; dígase usuarios, beneficiados del sistema o producto final, desarrolladores y los encargados de mantener el Software posteriormente.