

**Universidad de las Ciencias Informáticas**  
**Facultad #6**



**Título:** Propuesta de procedimiento para la adaptación de productos y soluciones que desarrolla DATEC.

Trabajo de Diploma para optar por el título de  
Ingeniero Informático

**Autor(es):** Arisuleidy Toranzo González

**Tutor(es):** José Luis Muñoz Suárez

**Co-tutor:** Maikel Zúñiga Suárez.

Ciudad de La Habana, Junio 2010.

“Año 52 de la Revolución”

## DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de Junio del año 2010.

Arisuleidy Toranzo González.

José Luis Muñoz Suárez.

---

**Autora**

---

**Tutor**

## DATOS DE CONTACTO

### **Síntesis de la Autora:**

Arisuleidy Toranzo González.

**E-mail:** [atoranzo@estudiantes.uci.cu](mailto:atoranzo@estudiantes.uci.cu)

### **Síntesis del Tutor:**

José Luis Muñoz Suárez

Ingeniero en Ciencias Informáticas

**E-mail:** [jlmunoz@uci.cu](mailto:jlmunoz@uci.cu)

### **Síntesis del Co-Tutor:**

Maikel Zúñiga Suárez.

Ingeniero en Ciencias Informáticas.

**E-mail:** [mzuniga@uci.cu](mailto:mzuniga@uci.cu)

## AGRADECIMIENTOS

- ✓ A toda mi familia, mis padres, mi hermanita Lisandra, mi abuelita Josefa y mis tíos Mariela y Modestico. A todos muchas gracias por su apoyo incondicional en todo momento, por quererme tanto, por confiar en mí, por todos sus consejos y su infinito amor y cariño hacia mí.
- ✓ A Maikel por estar conmigo siempre, por ayudarme en todo y por su apoyo. Gracias a él hoy he podido estar aquí graduándome como ingeniera. Te quiero mucho, y quiero que sepas que en mi corazón siempre va a existir un lugar para ti.
- ✓ A las muchachitas de la oficina Baby, Fifi, Odalis, Isbel, Yanelis, por ayudarme y preocuparse por mí en todo momento. A todas gracias y en especial a Baby que ha sido como una madre para mí desde que la conocí. Las voy a extrañar muchísimo cuando me vaya.
- ✓ A la profe Olga, con ella aprendí muchas cosas y en especial a ser muy responsable y preocupada por las cosas que hago. Muchas gracias por todo nunca me olvidaré de usted.
- ✓ A mi tutor José Luis, gracias por atenderme cada vez que lo necesité, sin tu ayuda no hubiese sido posible haber realizado este trabajo.
- ✓ A Imara y Aliekna por ser mis amigas de siempre. Con ustedes me divertí muchísimo, nunca las voy a olvidar.
- ✓ A todas las personas que tuvieron que ver de una forma u otra con la realización de este trabajo. Muchas gracias.

## DEDICATORIA

- ✓ Dedico este trabajo a mi mamá Claudina y a mi papá Pepe por el apoyo que me dieron en todo momento, sus preocupaciones y los sacrificios que hicieron por mí.
- ✓ En especial a mi abuelito Modesto que sé que le hubiese gustado mucho verme graduada de ingeniera.

## RESUMEN

Un procedimiento es una serie común de pasos definidos previamente que permiten realizar un trabajo de forma correcta. Los procedimientos aseguran el éxito de las empresas, ya que evitan la supresión de pasos y las actividades innecesarias. Además asegura un comportamiento uniforme de cada uno de los trabajadores, indicándoles lo que deben hacer y como lo deben hacer, aumentándose así la eficiencia, la productividad y por tanto la calidad en la prestación de los servicios.

Con vista de mejorar el control de la calidad, esta investigación tiene como objetivo elaborar un procedimiento para la adaptación de productos y soluciones que desarrolla el Centro de tecnología de Gestión de Datos (DATEC). Se obtuvo como resultado, un procedimiento que posibilitó la estandarización del proceso de adaptación de los productos y soluciones del Centro de tecnología de Gestión de Datos, alineado a las políticas de calidad del centro, y un servicio de adaptación de software organizado para todos los productos del Centro de tecnología de Gestión de Datos.

## PALABRAS CLAVE:

Adaptación de software, Mantenimiento de software, Personalización del software.

## INDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	5
1.1 Procedimiento.....	5
1.1.1 Características de los procedimientos.....	5
1.1.2 Estructura del procedimiento.....	6
1.2 Proceso de desarrollo del software.....	6
1.3 Personalización del software.....	8
1.3.1 Modelos de personalización.....	9
1.3.2 Principios clave de la personalización.....	10
1.4 Adaptación del software.....	11
1.4.1 Tipos de adaptación del software.....	11
1.5 Mantenimiento del software.....	14
1.6 Limitaciones del mantenimiento de software.....	15
1.7 Leyes del mantenimiento del software:.....	15
1.8 Tipos de mantenimiento de software.....	15
1.8.1 Mantenimiento correctivo.....	16
1.8.2 Mantenimiento adaptativo.....	16
1.8.3 Mantenimiento perfectivo.....	17
1.8.4 Mantenimiento preventivo.....	18
Conclusiones del capítulo.....	18
CAPÍTULO 2: PROPUESTA DE LA SOLUCIÓN.....	19
2.1 Plantilla del procedimiento.....	19
Introducción.....	19
Objetivo.....	19
Alcance.....	19
Definiciones y acrónimos.....	19
Referencias.....	20
Políticas para definir el procedimiento.....	20
Asignación de responsabilidades.....	21

2.2 Flujo del procedimiento.....	24
2.3 Descripción textual del procedimiento.....	24
2.3.1 Comprensión del Negocio.....	25
2.3.2 Levantamiento de Requisitos.....	26
2.3.3 Asignación de Recursos.....	27
2.3.4 Análisis del Sistema.....	27
2.3.5 Diseño del Sistema.....	28
2.3.6 Implementación.....	30
2.3.7 Pruebas de Unidad.....	31
2.3.8 Pruebas de Integración.....	32
2.3.9 Pruebas de Sistema.....	33
2.3.10 Pruebas de Aceptación.....	34
2.3.11 Realizar Capacitaciones.....	34
Conclusiones del capítulo.....	35
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	36
3.1 Método Delphi.....	36
3.2 Método de evaluación Criterio de Expertos.....	37
3.3 Selección de los expertos.....	38
3.4 Elaboración del cuestionario.....	40
3.5 Competencia de los expertos.....	40
3.6 Análisis de los Resultados del Cuestionario.....	42
Conclusiones del Capítulo.....	43
CONCLUSIONES.....	44
RECOMENDACIONES.....	45
BIBLIOGRAFÍA.....	46
ANEXOS.....	48
Anexo 1.....	48
Anexo 2.....	50
Anexo 3.....	51
Anexo 4.....	52
Anexo 5.....	53



Anexo 6 .....	54
Anexo 7 .....	55
Anexo 8 .....	58
Anexo 9 .....	60
GLOSARIO DE TÉRMINOS.....	62

## INTRODUCCIÓN

En la actualidad el éxito de las empresas radica principalmente en su imagen en el mercado. Generalmente están representadas por símbolos o marcas las cuales tienen como misión transmitir un mensaje único y distintivo, que comunique los beneficios principales del producto o servicio y que les dé una idea a las personas, de que es la mejor alternativa para obtener los bienes y servicios que dejarán satisfechas sus necesidades y expectativas.

Con la proliferación de las tecnologías de la informática y las comunicaciones (TIC) y el amplio desarrollo de la Web, se ha revolucionado el mundo de las empresas, proporcionándole ventajas significativas al permitir la promoción y venta de productos en Internet, el surgimiento de nuevas formas de trabajo como el tele trabajo, la disminución de los costos de producción, el fácil acceso a la información y las comunicaciones en la red. Su uso ha incrementado la eficiencia y la productividad en las empresas, así como la calidad en la prestación de servicios. Todas estas ventajas proporcionadas por las TIC, asociado a su carácter innovador y creativo en el acceso a nuevas formas de comunicación, han promovido la competitividad entre las empresas.

El uso eficiente de las TIC permite obtener ventajas competitivas, pero para lograr esto es preciso encontrar procedimientos acertados que posibiliten mantener tales ventajas en todo momento, así como disponer de recursos alternativos para adaptarlas a las necesidades del momento, pues las ventajas no serán siempre permanentes.

La adaptación de software es un paso fundamental para el éxito de las empresas ya que facilita la re-utilización de componentes, actividad que proporciona la optimización de los recursos de las instituciones, permite mantener los productos actualizados y contribuye a mejorar la funcionalidades de los sistemas para así mantener las ventajas competitivas en el mercado, a su vez es de gran importancia para los clientes, ya que a través de este proceso se adaptan los productos y servicios a los entornos cambiantes, lo que posibilita ofrecerles un servicio actualizado y duradero, siempre pensando en sus necesidades.

Actualmente la empresa que no se adapte a los cambios tecnológicos que vayan surgiendo, estará condenada al fracaso, hoy en día la tecnología de mayor valor es la que esté más actualizada.

El Centro de Tecnologías de Gestión de Datos (DATEC) de la Universidad de las Ciencias Informáticas (UCI), actualmente presta varios servicios, entre ellos están el despliegue de PATDSI (Paquete de ayuda en la toma de decisiones). Para acceder a este servicio el cliente lo solicita a la subdirección de negocio, esta planifica la fecha junto con el encargado de desplegar la aplicación y le entrega al cliente un documento con los requerimientos tecnológicos con los cuales debe contar. Luego cuando el cliente avise que lo tiene todo listo, el encargado de desplegar la aplicación va y la instala. Después de estar instalado el software el centro ofrece servicios de consultoría, en caso de que el cliente quiera modificar algo, se dan capacitaciones para el uso de la aplicación, se le entrega al cliente un manual de usuario de instalación y le brindan mantenimiento cada vez que lo necesiten.

Otro de los servicios es el despliegue del REDMINE. Para acceder a este servicio el cliente lo solicita en la subdirección de negocio, esta hace la planificación junto con el encargado de desplegar la aplicación y cuando se defina la fecha el encargado de desplegar la aplicación va y la instala, pero como no hay una comunicación con el cliente sobre el hardware y el software del que disponen, en ocasiones no se ha podido instalar la aplicación porque este no cuenta con la tecnología necesaria. En caso de que el cliente desee mejorar las funcionalidades del hardware o del software, no se podrá debido a varios factores que no posibilitan realizar estas acciones, ejemplo de esto es la falta de comunicación entre el centro y el cliente, lo que provoca que se desconozca lo que este desea en realidad.

El último de estos servicios es el desarrollo de los Almacenes de Datos, Para acceder a este servicio el cliente lo solicita en la subdirección de negocio. Luego los que van a desarrollar el almacén se entrevistan con el cliente y se realiza el levantamiento de los requisitos funcionales y no funcionales. Cuando está terminado el almacén se instala y se le entrega al administrador del mismo un manual de usuario donde se le capacita en cómo utilizarlo y cómo adicionar los data mart los cuales son pequeños almacenes centrados en un tema o un área de negocio específico dentro de una organización. En caso de que se quiera hacer alguna modificación al almacén, como por ejemplo agregarle una nueva funcionalidad sería preciso crear una nueva versión del mismo.

Según lo expresado anteriormente existe una gran diversidad en las formas de adaptación de los software desarrollados en el centro, esto provoca que se dificulte el proceso de control de la calidad de los mismos, ya que no existe un documento o procedimiento que estandarice este proceso. Además dificulta el proceso de gestión del conocimiento debido a que si el personal involucrado en el desarrollo o

despliegue de los productos se ausenta, provocaría que estos procesos no se puedan realizar de forma correcta, afectando la calidad de los mismos y por tanto el servicio brindado a los clientes.

Teniendo en cuenta lo expuesto anteriormente se plantea el siguiente **problema científico**: ¿Cómo estandarizar el proceso de adaptación de productos y soluciones que desarrolla el Centro de Tecnologías de Gestión de Datos?

A partir del problema científico planteado se define como **objeto de estudio**: El proceso de desarrollo del software, **enmarcado** en la adaptación de software.

Para darle cumplimiento al problema científico planteado se define como **objetivo general**: Elaborar una propuesta de procedimiento que estandarice la adaptación de productos y soluciones que desarrolla el Centro de Tecnologías de Gestión de Datos, a entornos específicos, alineado a las políticas de calidad del centro, para mejorar el control de la calidad.

Para dar cumplimiento al objetivo general planteado se definen las siguientes **tareas de investigación**:

- ✓ Estudio del estado del arte sobre el proceso de adaptación de software.
- ✓ Realización de un estudio sobre el proceso de adaptación de software en el Centro de Tecnologías de Gestión de Datos.
- ✓ Definición de la estructura y formato del procedimiento.
- ✓ Estudio y selección de los métodos de validación existentes para evaluar la propuesta.
- ✓ Realización de la fundamentación teórica de la investigación.
- ✓ Definición del alcance y el objetivo del procedimiento.
- ✓ Definición las actividades que van a conformar el procedimiento y de los roles que van a realizar estas actividades.
- ✓ Realización del diagrama de flujo del procedimiento.
- ✓ Validación de la solución propuesta.
- ✓ Elaboración del libro del procedimiento propuesto.

Para la realización de este trabajo se definen como **métodos de la investigación**:

✓ **Métodos Teóricos.**

**Método Analítico-Sintético:** Es utilizado en el análisis de la bibliografía existente sobre el tema, donde serán revisados las teorías planteadas, documentos, planteamientos, para luego extraer los elementos más importantes y adaptarlo a la situación.

✓ **Métodos Empíricos.**

**La entrevista:** Se utiliza con el propósito de conocer cómo se está desarrollando el proceso de adaptación y las características que debe tener el procedimiento a desarrollar teniendo en cuenta las necesidades del centro.

Una vez concluida la investigación se espera como **resultado**:

- ✓ Un procedimiento que guíe y controle el proceso de adaptación de las soluciones y los productos de DATEC, alineado a las políticas de calidad del centro.

El presente trabajo de diploma consta de 3 capítulos:

**Capítulo 1. Fundamentación teórica:** En el presente capítulo se definen los conceptos principales que serán usados en la investigación. Se describen las características de los procedimientos de forma general, aspectos fundamentales de la adaptación del software, el mantenimiento de software visto desde diferentes puntos de vista, sus limitaciones y los diferentes tipos de mantenimiento que existen.

**Capítulo 2. Propuesta de la solución:** En el presente capítulo se elaborará una propuesta de procedimiento de adaptación del software que contribuya a mejorar el control de la calidad del Software. Además se realizará el diagrama de Flujo del procedimiento, textual y graficado para una mejor comprensión del mismo.

**Capítulo 3. Validación de la solución propuesta:** En el presente capítulo se realizará la evaluación de la solución propuesta utilizando el método Delphi utilizando el criterio de un panel de expertos.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En la actualidad en la Universidad se confunden dos términos, el de personalización del software y el de adaptación del software. El primero se refiere al proceso de adaptar el software a las características, al gusto o a las necesidades de una persona y el segundo se refiere al proceso de modificar algo que está hecho para que se ajuste a nuevas condiciones. En el presente capítulo se definen los conceptos expresados anteriormente, se describen las características de los procedimientos de forma general, el mantenimiento de software visto desde diferentes puntos de vista, sus limitaciones y los diferentes tipos de mantenimiento que existen.

### 1.1 Procedimiento.

Un procedimiento es la acción de proceder o el método de ejecutar determinadas acciones [1]. Es una serie común de pasos definidos previamente, que permiten realizar un trabajo correctamente.

Según Melinkoff, R (1990), "Los procedimientos son la realización de una serie de labores en forma orgánica y guardando una sucesión cronológica en la manera de realizar estas labores" [2].

"Forma especificada para llevar a cabo una actividad o un proceso" [3].

#### 1.1.1 Características de los procedimientos.

- ✓ La definición de las actividades del procedimiento deben ser claras y precisas, que no den lugar a ambigüedades.
- ✓ Una vez aplicado el procedimiento el resultado debe ser el mismo donde sea aplicado.
- ✓ Para aplicar el procedimiento la capacitación sobre su uso debe ser mínimo.
- ✓ El procedimiento debe ser del conocimiento de todos los implicados en general.

Los procedimientos facilitan el éxito de las empresas en las diferentes actividades ya que ayudan al incremento de la productividad, eficiencia y la calidad en sus productos y servicios. Facilitan el proceso de hacer efectivas las normas, procesos y funciones administrativas. Facilitan las labores de auditoría y la evaluación del control interno, evitan la supresión de pasos y actividades innecesarias. Asegura un

comportamiento uniforme de cada uno de los trabajadores y aumenta su eficiencia, indicándoles lo que deben hacer y cómo deben hacerlo.

### 1.1.2 Estructura del procedimiento.

El procedimiento está estructurado por varias secciones que en su mayoría son comunes para todos los procesos. Las mismas se describen a continuación:

- ✓ **Introducción:** En esta sección se hace una explicación general sobre el contenido del procedimiento, de los fines y propósitos que se pretenden cumplir a través de él, a quien va a estar dirigido, el área de aplicación del procedimiento y la importancia de su revisión y actualización.
- ✓ **Objetivo:** Explicar el propósito que se pretende cumplir con el procedimiento.
- ✓ **Alcance:** Se describe el ámbito de aplicación del procedimiento.
- ✓ **Definiciones y acrónimos:** Aquí se expone una lista de abreviaturas con su significado, para una mejor comprensión de estas terminologías.
- ✓ **Referencias:** Hace referencia a los documentos consultados para la elaboración del procedimiento.
- ✓ **Política para definir el procedimiento:** Aquí se debe definir los límites generales dentro de los cuales ha de realizarse el procedimiento.
- ✓ **Asignación de responsabilidades:** Aquí se debe indicar quien es el responsable de la elaboración, emisión, control, revisión y actualización del procedimiento.
- ✓ **Descripción gráfica del procedimiento:** Aquí se elabora el diagrama de flujo, que es una representación gráfica que muestra la secuencia en que se realiza la actividad necesaria para realizar un trabajo determinado.
- ✓ **Descripción textual del procedimiento:** En esta sección se describe textualmente las actividades del diagrama de flujo.

### 1.2 Proceso de desarrollo del software.

Un proceso del software es un conjunto de actividades y resultados asociados que producen un producto de software. Estas actividades son llevadas a cabo por los ingenieros de software. Existen cuatro actividades fundamentales de procesos que son comunes para todos los procesos de software [4]. Estas actividades son:

1. **Especificación de software:** Donde los clientes e ingenieros definen el software a producir y las

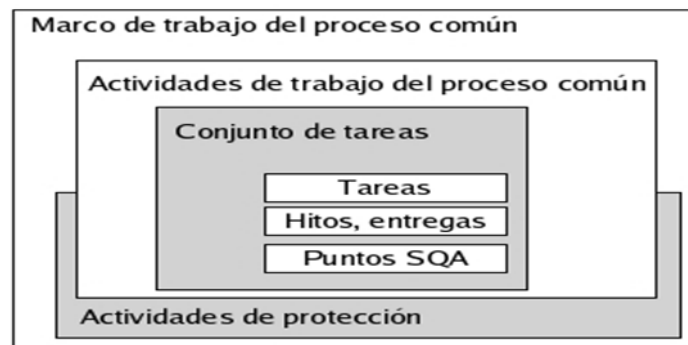
restricciones sobre su operación.

2. **Desarrollo del software:** Donde el software se diseña y se programa.
3. **Validación del software:** El software se valida para asegurar que es lo que el cliente requiere.
4. **Evolución del software:** El software se modifica para adaptarlo a los cambios requeridos por el cliente y el mercado [4].

Además de estas actividades fundamentales, Pressman menciona un conjunto de “actividades protectoras”, que se aplican a lo largo de todo el proceso del software. Ellas se señalan a continuación [5]:

1. Seguimiento y control de proyecto de software.
2. Revisiones técnicas formales.
3. Garantía de calidad del software.
4. Gestión de configuración del software.
5. Preparación y producción de documentos.
6. Gestión de reutilización.
7. Mediciones.
8. Gestión de riesgos.

Pressman caracteriza un proceso de desarrollo de software como se muestra en la siguiente figura.



**Figura1. Proceso de desarrollo del software.**



Los elementos involucrados se describen a continuación:

1. **Un marco de trabajo del proceso común**, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia del tamaño o complejidad.
2. **Un conjunto de tareas**, cada uno es una colección de tareas de ingeniería del software, hitos de proyectos, entregas y productos de trabajo del software, y puntos de garantía de calidad, que permiten que las actividades del marco de trabajo se adapten a las características del proyecto de software y los requisitos del equipo del proyecto.
3. **Las actividades de protección**, tales como garantía de calidad del software, gestión de configuración del software y medición, abarcan el modelo del proceso. Las actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso [5].

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software, que reúna los requisitos del cliente, esto significa que el software construido debe cumplir con las especificaciones establecidas, dentro del tiempo previsto, con los costos y recursos planificados o acordados.

Este proceso es intensamente intelectual, afectado por la creatividad y juicio de las personas involucradas [6].

No existe un proceso de desarrollo de software universal. Diferentes tipos de sistemas necesitan diferentes tipos de procesos de software. Se debe tener cuidado al seleccionar un proceso de software, ya que un proceso inadecuado puede reducir la calidad del producto de software y/o incrementar los costos de su desarrollo [4].

### **1.3 Personalización del software.**

Según “The Free Dictionary” personalizar es el proceso de adaptar algo a las características, al gusto o a las necesidades de una persona [7].

En el marketing la personalización es definida como “tratar de forma distinta a clientes distintos”, para elevar su nivel de vinculación y lealtad con la empresa” [8].

Personalizar es reconocer a cada comprador como una entidad independiente, asociada a un conjunto de información que la diferencia del resto de compradores y en base a la cual se orienta el modelo de negocio con un objetivo muy simple: conseguir la venta, la venta repetitiva y la fidelización.[9]

### **1.3.1 Modelos de personalización.**

Uno de los objetivos de la personalización es materializar una venta y ese objetivo puede ser alcanzado de diferentes formas, lo que da origen a que existan diferentes modelos de personalización.

**La personalización del interface:** Mediante la personalización de la interfaz, se brinda la posibilidad al usuario de que pueda personalizar los elementos que componen el interface de usuario, fundamentalmente, en cuanto a accesos a contenidos de una forma definida por el usuario.

Este modelo de personalización es especialmente agradecido por los usuarios en interfaces que presentan una sobrecarga de contenidos, como ocurre muchas veces con los portales horizontales, donde podemos encontrar más de 80 enlaces a diferentes contenidos en la página de inicio. Es como si el mando a distancia del televisor dispusiera de más de 80 botones para la selección de canales y nunca viéramos más de seis canales diferentes.

Es fácil entender que no todos los usuarios van a tener los mismos gustos ni van a estar interesados de la misma forma por todos los contenidos, con lo cual, permitir una re-definición del interface puede ser el primer paso para asegurar la victoria en la batalla por conseguir visitas repetidas de ese usuario.

**La personalización del trato:** Mediante la personalización del trato, el usuario es identificado por su nombre y no de forma anónima. Es, sin duda alguna, el modelo de personalización más simple, pero se debe reconocer que nos gusta cuando se refieren a nosotros por nuestro nombre.

**La personalización de contenidos:** Personalizar contenidos es mostrar a cada usuario los contenidos más adecuados de acuerdo a alguna preferencia, hábito o gusto que queda recogido en variables. Variables como destino preferido, perfil de inversor o género más comprado. [9]

### 1.3.2 Principios clave de la personalización.

Si se tuvieran que resumir en diez puntos los aspectos clave de una estrategia de personalización, bien podrían ser los siguientes:

- ✓ Identificar claramente los objetivos de la personalización.
- ✓ Aprender de cada movimiento del usuario.
- ✓ Contemplar todas las posibles fuentes de información.
- ✓ Obtener el máximo de información y de la forma menos intrusiva posible.
- ✓ Permitir a los usuarios que nos comuniquen que la personalización ha fallado.
- ✓ Cuidar la privacidad.
- ✓ Hacer ver al usuario que se está personalizando y que eso es un servicio para él.
- ✓ Conocer lo que no le gusta al usuario es tan importante como conocer lo que le gusta.
- ✓ Nunca dejar a un usuario esperando por una respuesta personalizada, para eso es mejor no personalizar.
- ✓ Medir el éxito de la personalización.[9]

La personalización es un factor clave para aportar valor añadido a los clientes. La personalización de determinados productos o servicios es más fácil que otra, dependiendo de su soporte. Un ejemplo claro es el de una empresa que ofrece productos y servicios de información, los cuales serían relativamente fáciles de personalizar si tales productos estuviesen en formato digital, sin embargo si la empresa ofreciera mercaderías o productos físicos, sería más complicada la adaptación al cliente [10].

El término “personalización” ha proliferado hasta acabar por aplicarse casi para cualquier acción de marketing, desde una mera carta dirigida al destinatario con su nombre y apellidos, hasta la configuración de la pantalla de un móvil o de un PC al gusto del usuario. “Personalización” es, hoy en día, una palabra que está en boca de la mayoría de expertos en marketing y ventas, pero que se traduce en auténticos resultados en contadas ocasiones [8].

En la actualidad las empresas no se preocupan por tener muchos clientes, sino por fidelizarlos, esto lo hace a través de la personalización que se traduce en satisfacción de las necesidades y expectativas de los consumidores. La personalización posibilita a las empresas la creación de productos con mayor calidad siempre pensando en satisfacer las necesidades del cliente. Aumenta su eficiencia, productividad

e ingresos de manera sorprendente ya que es una forma de lograr que el cliente se identifique con la empresa, y esto para ellos es algo muy significativo.

#### **1.4 Adaptación del software.**

Según el diccionario de la real academia española la palabra adaptar hace referencia a, hacer que un objeto o un mecanismo desempeñe funciones distintas de aquellas para las que fue construido.

Según el Oxford Advanced Learner's Dictionary 2007:

Concerniente al cambio; ser capaz de cambiar cuando sea necesario para tratar con diferentes situaciones" [11]. Por tanto, adaptativo es una propiedad que define la habilidad de cambiar o ajustarse a diferentes condiciones.

Mientras a la evolución del software le interesa modificar sistemas en tiempo de ejecución, la adaptación del software se involucra en decidir qué y cuándo las modificaciones son apropiadas. Durante la adaptación del software, los sistemas evolucionan dinámicamente para enfrentarse a nuevas metas funcionales o no funcionales. La meta evidente es crear sistemas que se adapten por sí solos, esto es, sistemas que puedan modificarse a sí mismos [12].

La adaptación de software se ocupa de proporcionar las técnicas para hacer arreglos en el software ya desarrollado, a fin de reutilizarlos en nuevos sistemas. La adaptación se relaciona con el mantenimiento, este se refiere a como los métodos de software pueden gestionar el sistema y lograr su evolución de una manera fácil y eficiente. Por lo tanto, la adaptación puede ayudar al mantenimiento al proporcionar los medios técnicos para hacer frente a algunos cambios en los requisitos. Sin embargo, los cambios y ampliaciones de funcionalidad que se abordan normalmente durante el mantenimiento en general, van más allá de la adaptación [13].

##### **1.4.1 Tipos de adaptación del software.**

Existen diferentes tipos de adaptación, los mismos se pueden clasificar atendiendo a diferentes criterios. Uno de esos criterios (el punto en el ciclo de vida del software en el que la adaptación se lleva a cabo) introduce dos categorías diferentes:

**Estática o adaptación en tiempo de diseño:** Incluye toda la adaptación hecha antes de que el sistema se esté ejecutando. Se puede referir tanto a los requerimientos de adaptación o a la adaptación ya desarrollada. Es necesaria cuando la especificación de un sistema debe extenderse para satisfacer nuevos requerimientos, o para modificar o cambiar los requisitos existentes. Ejemplo de este tipo de adaptación es introducir soporte para un nuevo protocolo de comunicación de red, adicionar nuevos atributos a un modelo de datos, etc.

**Dinámica o adaptación en tiempo de ejecución:** Este es el caso en que se ejecuta el software que debe adaptarse a fin de cambiarse la forma en que se presta el servicio. Esta es la situación típica en escenarios de la computación ubicua y móvil. Aquí, los componentes que deben adaptarse, así como las medidas para gestionar la adaptación, son desconocidos antes del momento de la adaptación [13].

Otra de las clasificaciones se basa en la forma en que se gestiona la adaptación. Atendiendo a este criterio, la adaptación podría ser:

**Adaptación manual:** Los pasos para la adaptación así como los adaptadores son especificados y desarrollados por las personas involucradas en el proceso de desarrollo del software (diseñadores, arquitectos, programadores, etc.) probablemente con la asistencia de herramientas de software.

**Adaptación automática:** Todas las medidas de adaptación y los adaptadores mismos son generados automáticamente por las herramientas de software. Estos tipos de herramientas deben ser capaces de detectar la necesidad de adaptación, así como para determinar si la adaptación necesaria es posible o no. A continuación, la herramienta procedería a determinar los pasos a realizar para gestionar la adaptación. Por último, un adaptador sería generado automáticamente [13].

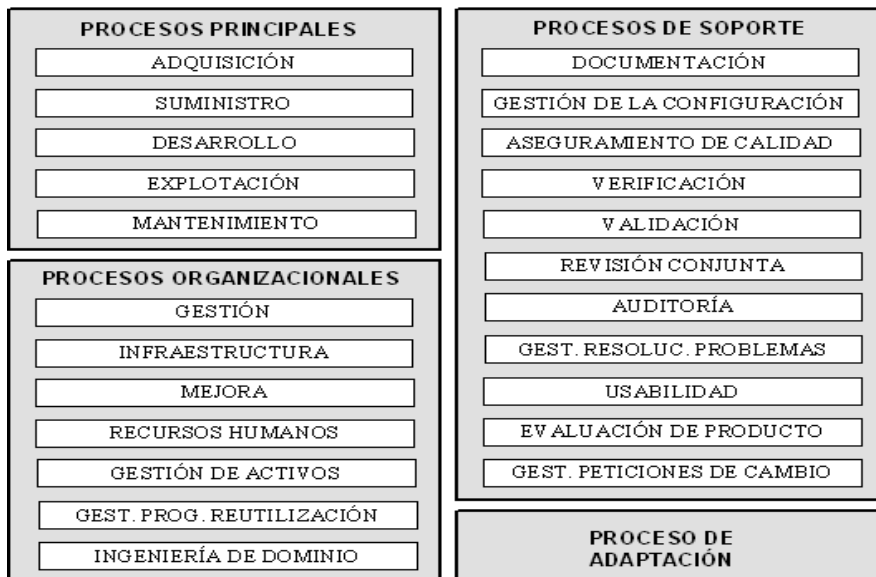
Por último, un criterio alternativo para clasificar la adaptación es introducido y se refiere al tipo de propiedades que van a ser adaptadas:

**La adaptación funcional:** Esta es la adaptación dirigida a hacer los arreglos sobre los servicios prestados por el sistema. Puede implicar adicionar nuevos servicios o modificar uno existente.

**La adaptación técnica:** Este es el caso cuando la adaptación se dirige a modificar la forma en que se prestan los servicios. Esto suele hacerse mediante la adición o eliminación de restricciones al

comportamiento de estos servicios. Ejemplos de este tipo de adaptación son la adición de restricciones en tiempo real, funciones de seguridad, el cambio de protocolos de red, etc. [13].

La norma ISO 12207 contempla dentro de sus procesos el de adaptación, el mismo es muy importante ya que posibilita adaptar el software a las necesidades de los clientes y mantenerlo funcionando correctamente y actualizado.



**Figura 2. Procesos de la norma ISO 12207**

La adaptación del software es cada vez más necesaria en estos tiempos donde el software está en constante cambio. Cada vez se necesita más la actualización y mejora de las funcionalidades de los sistemas para mantener las ventajas competitivas en el mercado. Este es un proceso complejo y con altos costes de implementación, lo que provoca que muchas empresas que no cuentan con los recursos necesarios para adaptarse a los cambios tecnológicos, se quiebren y dejen de existir. Hay que tener cuidado en el proceso de adaptación, ya que para que tenga éxito se debe tener conocimiento sobre cómo utilizar lo que ha sido cambiado y todos deben estar de acuerdo con los cambios.

Después de haber realizado un estudio sobre los diferentes tipos de adaptación que existen, la que se va a realizar en el centro es la adaptación de software funcional.

Luego de analizar los términos adaptación y personalización, se llega a la conclusión de que no se refieren a lo mismo ya que el primero se refiere a satisfacer las necesidades del usuario visto como

institución, como la empresa, y el segundo está enfocado a satisfacer las preferencias de los usuarios, es decir está enfocado al usuario como persona.

### **1.5 Mantenimiento del software.**

El proceso de cambiar el software después de su entrega se denomina a menudo mantenimiento del software [5].

La definición de “Mantenimiento del Software” del estándar IEEE 1219 es: “El mantenimiento del software es la modificación de un producto software después de la entrega para corregir fallos, para mejorar el rendimiento u otros atributos, o para adaptar el producto a un entorno modificado” [14].

El mantenimiento del software es una actividad muy importante realizada después de entregado el producto de software, es decir después que el producto está en uso, ya sea para adaptarlo a nuevos entornos o para corregirle errores, entre otros. En los diversos modelos de ciclo de vida del software, el mantenimiento se realiza al final del ciclo de vida pero no por esto es el menos importante. Existen varios autores que expresan la importancia de este a lo largo de todo el desarrollo del software, por su repercusión económica, temporal y de recursos. Por ejemplo, Pigoski (1997) resalta que hay una necesidad de comenzar a considerar el mantenimiento desde el mismo momento en que comienza el desarrollo:

El mantenimiento del software es la totalidad de las actividades necesarias para proporcionar soporte económico (rentable) al sistema software. Estas actividades se desarrollan tanto antes como después de la entrega. Las actividades previas a la entrega incluyen la planificación de las operaciones posteriores a la entrega, planificación del soporte y determinación de la logística [15].

El mantenimiento del software supone adaptar el paquete o sistema objeto del mismo a nuevas situaciones como:

- ✓ Cambio de hardware.
- ✓ Cambio de sistema operativo.

## 1.6 Limitaciones del mantenimiento de software.

- ✓ Inexistencia de métodos, técnicas y herramientas que puedan proporcionar una solución global al mantenimiento.
- ✓ La complejidad de los sistemas se incrementa paulatinamente por la realización de continuas modificaciones.
- ✓ La documentación del sistema es defectuosa o inexistente.
- ✓ Se considera el mantenimiento como una actividad poco creativa, a diferencia del desarrollo.
- ✓ Las actividades del mantenimiento se suelen realizar bajo presión de tiempo.
- ✓ Poca participación del usuario durante el desarrollo del sistema [16].

## 1.7 Leyes del mantenimiento del software:

- ✓ **Continuidad del cambio:** Un programa utilizado en un entorno del mundo real debe cambiar si no quiere dejar de ser usado. Esto se debe a que surgen nuevas funcionalidades, un nuevo hardware puede permitir mejoras en el software, se corrigen defectos, se instala en otro sistema operativo/máquina o el software necesita ser más eficiente.
- ✓ **Incremento de la complejidad:** Cuando se realizan cambios en un programa la estructura se hace más compleja si no se utilizan técnicas de ingeniería del software.
- ✓ **Evolución del programa:** Es un proceso autorregulado. Se mantienen las tendencias e invariantes de las propiedades del programa.
- ✓ **Conservación de la estabilidad organizacional:** La carga que supone el desarrollo de un programa es aproximadamente constante e independiente de los recursos dedicados a lo largo del tiempo de vida del mismo.
- ✓ **Conservación de la Familiaridad:** El incremento en el número de cambios introducidos con cada versión (release) es aproximadamente constante [17].

## 1.8 Tipos de mantenimiento de software.

La fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios debido a las



mejoras producidas por los requisitos cambiantes del cliente [6]. En la fase de mantenimiento se encuentra cuatro tipos de mantenimientos:

### **1.8.1 Mantenimiento correctivo.**

Es la modificación de un producto software después de su puesta en producción y para corregir los fallos descubiertos. El Mantenimiento correctivo tiene por objetivo localizar y eliminar los posibles defectos de los programas. [14].

**Corrección:** Incluso llevando a cabo las mejores actividades de garantía de calidad, es muy probable que el cliente descubra los defectos en el software. El mantenimiento correctivo cambia el software para corregir los defectos [5].

Se define como aquel proceso orientado a la reparación de defectos existentes en un sistema software. Este tipo de mantenimiento suele ser necesario en alguna de las siguientes ocasiones:

- ✓ Cuando el programa falla o aborta.
- ✓ Un programa produce un resultado que no es acorde con los requisitos.
- ✓ Los diseños y requisitos no están acordes con el software que los soporta.
- ✓ La documentación de usuario lleva a conclusiones erróneas al propio usuario hacia actividades que provoquen resultados incorrectos o fallos en el sistema [5].

### **1.8.2 Mantenimiento adaptativo.**

Es la modificación de un producto software, después de su puesta en producción, para mantener operativo un programa mientras se realiza un cambio en el entorno de producción. Tiene por objetivo la modificación de un programa debido a cambios en el entorno, cambios en el hardware o en el software, en el que se ejecuta. En cuanto a los cambios en el hardware cabe destacar, por ejemplo, los cambios en la plataforma de instalación, pasar de un sistema de 16 bits a uno de 32 bits; los cambios en el sistema de comunicaciones, pasar de un sistema de trabajo sin red a un sistema de red; etc. En cuanto a los cambios en el software cabe destacar, por ejemplo, cuando se desea pasar de un cliente clásico desarrollado en Visual Basic, Delphi o C/C++ a un cliente "ligero" (Thin Client) para ejecutarse sobre un navegador de Internet; etc. Este tipo de mantenimiento es el más usual debido a los rápidos cambios que se producen en la tecnología informática, que en la mayoría de las ocasiones dejan obsoletos los productos software

desarrollados, no por su inoperancia, sino por la competitividad entre las empresas, en las que cada vez influye más el software utilizado. Por ejemplo, dar acceso a los productos a través de Internet. [14].

**Adaptación:** Con el paso del tiempo, es probable que cambie el entorno original (por ejemplo: la unidad central de procesamiento (CPU), el sistema operativo, las reglas de empresa, las características externas del producto) para el que se desarrolló el software. El mantenimiento adaptativo produce modificación en el software para acomodarlo a los cambios de entorno externo [5].

Se define como el proceso para mejorar la funcionalidad del software, hardware y su documentación. Sus pasos son:

- ✓ **Definición de requisitos:** revisar y entender los requisitos de nuevos cambios.
- ✓ **Diseño del sistema:** determina dónde y cuándo implementar los cambios a nivel de sistema.
- ✓ **Diseño de datos:** los cambios en las estructuras de datos deben ser diseñados.
- ✓ **Diseño de programa:** analizar los documentos del diseño del programa para determinar dónde añadir, modificar y borrar las funciones que implementan los cambios propuestos.
- ✓ **Diseño del módulo:** analizar los documentos para determinar cómo integrar cambios en un módulo existente o bien crear uno nuevo [18].

#### **Beneficios del mantenimiento adaptativo:**

- ✓ Mejorar la productividad automatizando actividades.
- ✓ Incrementar la satisfacción del usuario.
- ✓ Incrementar las ganancias mediante la mejora de respuestas a las oportunidades de negocio [18].

#### **1.8.3 Mantenimiento perfectivo.**

Es la modificación de un producto software, después de su puesta en producción y para mejorar el rendimiento o la mantenibilidad, que es la facilidad de mantenimiento que tiene un software, es decir, la facilidad de un software para ser modificado, lo que influye directamente en los costes del mantenimiento. [14]

El objetivo principal del mantenimiento perfectivo es mejorar la calidad del software para reducir el coste e impacto de los cambios de los procesos de mantenimiento.

#### **1.8.4 Mantenimiento preventivo.**

Consiste en la modificación del producto Software sin alterar las especificaciones del mismo, para mejorar las propiedades de mantenimiento del producto y facilitar así las futuras tareas de mantenimiento. Los cambios que se llevan a cabo son en cuanto a los comentarios del código y la reestructuración de los programas para mejorar su comprensión [14]. Es el que se ejecuta para prevenir fallos antes de que éstos ocurran. Por ejemplo, un chequeo de tipos antes de compilar y ejecutar una aplicación.

Según las definiciones expuestas anteriormente se puede ver que existen varios tipos de mantenimiento según el tipo de modificación que se le desee realizar al software. Para la realización del procedimiento se va a utilizar algunas de las actividades del mantenimiento adaptativo, ya que guarda relación con la adaptación del software y es una solución que existe, que ya está probada, depurada y optimizada. Por lo tanto se va a reutilizar este tipo de mantenimiento para aplicarlo a la adaptación de software.

#### **Conclusiones del capítulo.**

El estudio realizado en este capítulo, es utilizado como fundamento teórico para la elaboración del procedimiento propuesto más adelante; brindando varios conceptos que servirán como guía durante la elaboración de la propuesta del procedimiento. Se determinó que personalización no es lo mismo que adaptación, y para la realización del procedimiento se utilizó el mantenimiento adaptativo.

## CAPÍTULO 2: PROPUESTA DE LA SOLUCIÓN.

Luego de haber analizado como es realizada la adaptación de software en el Centro de Tecnología de Gestión de Datos y con el propósito de mejorar el control de la calidad en esta actividad, surge la necesidad de confeccionar una propuesta de procedimiento para estandarizar y optimizar el control de la calidad del Software en este proceso. Los métodos utilizados, el objetivo, el alcance, y la composición del procedimiento serán descritos en el presente capítulo mediante una plantilla, la cual se muestra a continuación.

### 2.1 Plantilla del procedimiento.

#### **Introducción.**

Este documento tiene como propósito definir el proceso de adaptación de software en el Centro de Tecnología de Gestión de Datos y está dirigido a los responsables de adaptar los productos y soluciones del Centro.

#### **Objetivo.**

El objetivo de este procedimiento es proporcionar una guía para el proceso de adaptación del software en el Centro de Tecnología de Gestión de Datos.

#### **Alcance.**

Este procedimiento será aplicado en los productos y soluciones que desarrolla el Centro de Tecnología de Gestión de Datos.

#### **Definiciones y acrónimos.**

**Artefacto:** Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

**CU:** Caso de Uso.

**DATEC:** Centro de Tecnologías de Gestión de Datos.

**Prioridad auxiliar:** No son claves para la arquitectura y completan casos de usos críticos o secundarios.

**Prioridad crítico:** Más importantes para los usuarios porque cubren las principales tareas o funciones que el sistema ha de realizar.

**Prioridad opcional:** Responden a funcionalidades que pueden o no estar en la aplicación.

**Prioridad secundaria:** Sirven de apoyo a los casos de uso críticos.

## Referencias.

Código	Título
[1]	IPP-350:2008 Libro de Proceso para Definir Procesos [19].

## Políticas para definir el procedimiento.

- ✓ Cada política debe estar respaldada por materiales de apoyo como documentos, procesos, imágenes, explicaciones en caso de que no se explique por sí misma.
- ✓ Los productos empaquetados deben cumplir con todos y cada uno de los criterios de criticidad.
- ✓ Los productos entregados al departamento de calidad deben contar con una terminación decorosa donde el número de defectos o no conformidades no exceda un límite previamente establecido para cuyo caso serán abortadas las pruebas y revisiones o limitadas al tiempo planificado.
- ✓ Cada versión desplegada de un producto de DATEC deben someterse a un proceso de mejora continua, hasta hacerse estable o la muerte de la versión, siempre supervisado por soporte y postventa.
- ✓ Gestión de proyectos del centro: Establecer y mantener planes que definan las actividades del proyecto. Incluye los siguientes temas: Desarrollo del plan, Interacción con los agentes apropiados, Generar compromiso con el plan y Mantenimiento al plan.
- ✓ Los proyectos tienen una duración máxima de un año y deben alinearse, en lo posible, al concepto de año fiscal. Una iteración al final del año fiscal cerrará la ejecución de año fiscal y preparará el presupuesto del año siguiente.
- ✓ Las tareas deben ser medibles y comprobable su cumplimiento para lo que cada una tendrá un artefacto asociado donde en su conjunto pueda dar visión del estado de avance del trabajo.

### Asignación de responsabilidades.

A continuación se presenta la definición de los roles involucrados en el procedimiento:



Figura 3. Roles involucrados en el procedimiento.

**Rol:** Líder de proyecto.

**Descripción:** Es la persona que Planifica, administra y asigna los recursos, determina las prioridades, coordina las relaciones con los clientes y los usuarios y mantiene al equipo de proyecto enfocado en las tareas. Además establece las prácticas que garantizan la integridad y calidad de los artefactos del proyecto.

**Actividades:**

- ✓ Coordinar las relaciones entre el equipo de trabajo y el cliente.
- ✓ Gestionar los recursos humanos necesarios en la ejecución de las actividades por el cliente en el desarrollo del proyecto.
- ✓ Lograr el cumplimiento de las actividades planificadas por el Equipo de Desarrollo.
- ✓ Controlar y dirigir el proyecto evaluando todas las actividades que se realizan en el mismo.

**Rol:** Analista.

**Descripción:** Define los alcances del sistema e identifica a los actores y casos de uso que permiten modelar completa y consistentemente el sistema.

**Actividades:**

- ✓ Realizar modelación del negocio
- ✓ Realizar modelación del sistema.
- ✓ Capturar los requisitos del sistema, ya sean funcionales, o no funcionales.
- ✓ Realizar el diagrama de caso de uso.
- ✓ Identificar las entidades.
- ✓ Realizar el modelo de objetos.
- ✓ Realizar el glosario de términos de la empresa.
- ✓ Diseñar los casos de pruebas del sistema.

**Rol:** Arquitecto.

**Descripción:** Es la persona responsable de la arquitectura del software que incluye las decisiones técnicas claves de todo el diseño e implementación del proyecto.

**Actividades:**

- ✓ Priorizar los casos de uso. Definir la selección del conjunto de escenarios y casos de uso que serán analizados en la iteración de acuerdo a su prioridad.
- ✓ Estructurar el modelo de análisis.
- ✓ Analizar los documentos para determinar cómo integrar cambios en un modulo existente o bien crear uno nuevo.
- ✓ Analizar los documentos del diseño para determinar donde añadir, modificar y borrar las funciones que implementan los cambios propuestos.
- ✓ Estructurar el modelo de diseño.
- ✓ Estructurar el modelo de despliegue.
- ✓ Realizar la descripción de la arquitectura del sistema.
- ✓ Confeccionar el plan de integración.
- ✓ Diseñar los casos de prueba de integración.
- ✓ Mantener actualizada la arquitectura.

**Rol:** Programador.

**Descripción:** Es la persona que realiza los cambios en la aplicación.

**Actividades:**

- ✓ Escribir el código fuente de los módulos.
- ✓ Reutilizar código si es necesario.
- ✓ Realizar pruebas de unidad a la aplicación.

**Rol:** Probador.

**Descripción:** Es la persona responsable de las pruebas a la aplicación.

**Actividades:**

- ✓ Realizar las pruebas necesarias.
- ✓ Registrar el resultado de las pruebas.



## 2.2 Flujo del procedimiento.

A continuación se muestra el flujo de realización del procedimiento:

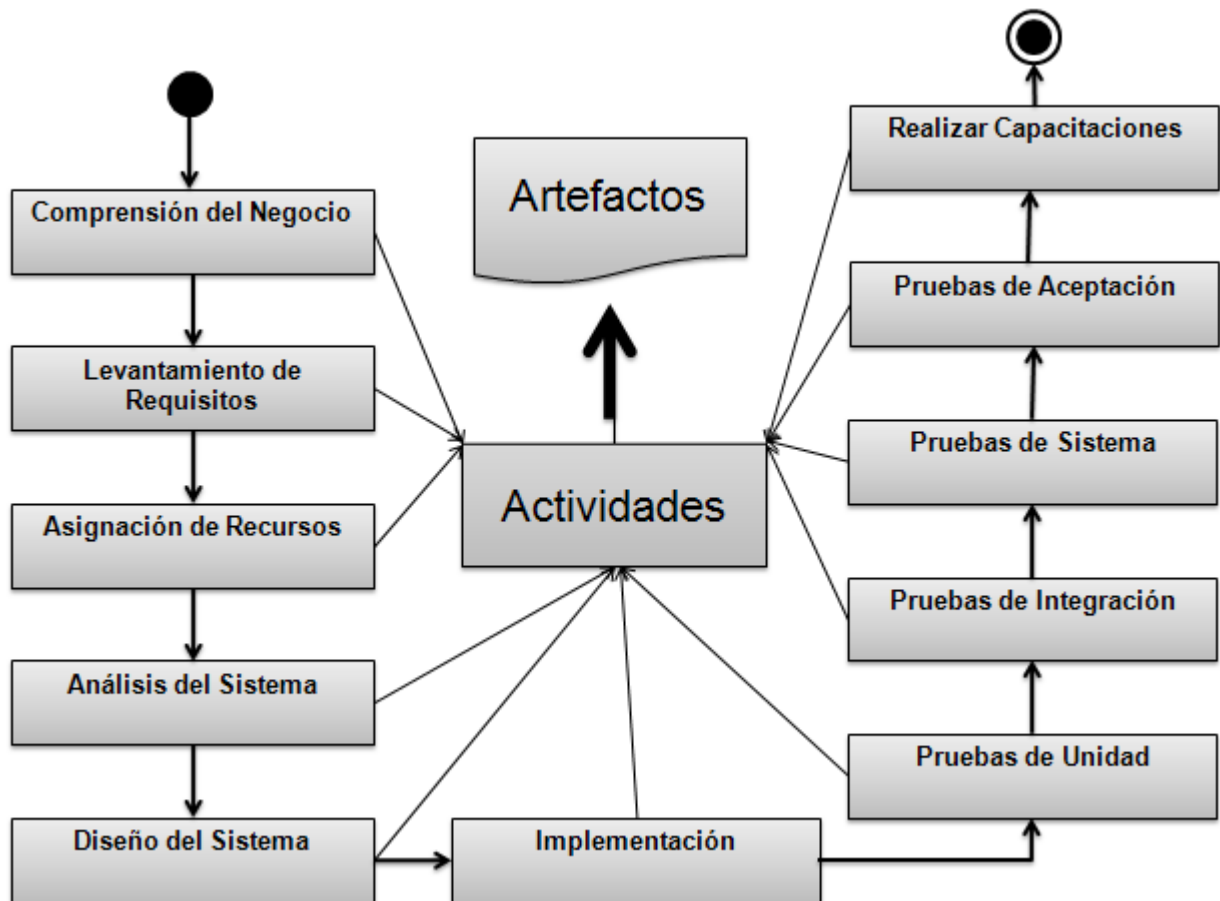


Figura 4. Flujo del procedimiento.

## 2.3 Descripción textual del procedimiento.

Este procedimiento está compuesto por varias actividades, las cuales se complementan para dar cumplimiento al objetivo general propuesto anteriormente.

Para realizar el procedimiento fue necesario el estudio detallado del mantenimiento de software, el cual ayudó a definir la lógica del procedimiento, mientras que el mantenimiento adaptativo aportó algunas ideas

sobre cuáles serían algunas de las posibles actividades a desarrollar en el proceso de adaptación del software y que serán reflejadas en el procedimiento. A continuación se describen las fases con las respectivas actividades que van a conformar dicho procedimiento:

### **2.3.1 Comprensión del Negocio.**

Esta fase está enfocada a revisar y entender los procesos de negocio de la empresa que solicita el servicio de adaptación. Se debe tener conocimiento a fondo de la funcionalidad, la visión, las metas, los objetivos, la estructura interna y los requisitos del mismo. En esta fase el analista debe entrevistarse con el cliente para obtener información sobre la organización que este representa. Una vez realizadas las entrevistas se debe hacer una comparación de lo que se hace en el centro con lo que pide el cliente. Si el pedido del cliente no está en correspondencia con los productos desarrollados por el centro, se le informa que su solicitud no puede ser realizada. En caso contrario se procede a realizar la adaptación haciendo un estudio sobre la información acopiada para comprender como funciona el negocio de la organización a la cual se le van a implementar los cambios, así como las personas que interactúan con el mismo. Para llevar a cabo esta fase se proponen las siguientes actividades:

- ✓ Recopilar información de la organización e identificar el problema a resolver.
- ✓ Hacer un estudio de la información recopilada para comprender cómo funciona el negocio de la organización a la cual se le va a implementar los cambios, así como las personas que interactúan con el mismo.
- ✓ Realizar el glosario de términos de la empresa.
- ✓ Identificar las identidades del negocio.
- ✓ Realizar el modelo de objetos del negocio.
- ✓ Realizar el modelo de casos de uso del negocio.

La planificación de todas las actividades que se vayan a realizar debe estar contemplada en el plan de trabajo.

#### **Artefactos que se generan:**

- ✓ Entrevista.
- ✓ Minuta de reunión.

- ✓ Glosario de términos.
- ✓ Entidad del negocio.
- ✓ Modelo de objetos del negocio.
- ✓ Modelo de caso de uso.
- ✓ Plan de trabajo.

**Responsable (s):** Analista.

### **2.3.2 Levantamiento de Requisitos.**

Luego de haber determinado que es posible realizar la adaptación del software, se procede a establecer cuáles son las características que debe tener el sistema para satisfacer las necesidades del cliente. Esta fase está enfocada a revisar y entender los nuevos requisitos de la solicitud del usuario. Para el desarrollo de esta fase el equipo de trabajo debe entrevistarse con el cliente para identificar el problema a resolver y revisar los nuevos requisitos. Una vez realizadas las entrevistas se procede a hacer un levantamiento de los requerimientos propuestos. Para llevar a cabo esta fase se proponen las siguientes actividades:

- ✓ Capturar los requisitos del sistema, ya sean funcionales, o no funcionales.
- ✓ Verificar y validar los requisitos del sistema.
- ✓ Realizar la modelación del sistema.
- ✓ Realizar diagrama de caso de uso del sistema.
- ✓ Priorizar los casos de uso asignándole un grado de prioridad (Crítico, Secundario, Auxiliar, Opcional), para determinar en que iteración serán desarrollados cada uno.
- ✓ Realizar revisiones con el cliente de la documentación de especificación de requisitos realizada por el analista para ir analizando lo que el usuario realmente quiere.
- ✓ Realizar revisiones a la documentación generada como artefacto, utilizando la lista de chequeo de documentos. (*Ver anexo 6*).
- ✓ En caso de existir errores al realizar la revisión, modificar para erradicar el problema detectado.
- ✓ Diseñar los casos de prueba del sistema.

**Artefactos que se generan:**

- ✓ Entrevista.
- ✓ Minuta de reunión.
- ✓ Glosario de términos.
- ✓ Especificación de requisitos.
- ✓ Diagrama de caso de uso del sistema.
- ✓ Caso de prueba.
- ✓ Plan de trabajo.

**Responsable (s):** Analista.

**2.3.3 Asignación de Recursos.**

Luego de haberse realizado el levantamiento de los requisitos del software, al cual se le va a realizar la adaptación. Se procede a realizar un análisis sobre los recursos (Máquinas, Servidores, Repositorios, Herramientas, Espacio en disco, etc.) que hacen falta para realizar el proyecto en cuestión. Después de conocerse los recursos que son necesarios para la ejecución del proyecto. Se procede a la asignación de los mismos al equipo de trabajo. Para llevar a cabo esta fase se proponen las siguientes actividades:

- ✓ Gestionar los recursos humanos y tecnológicos necesarios en la ejecución de las actividades por el cliente en el desarrollo del proyecto.
- ✓ Asignar los recursos al equipo de trabajo.

**Artefactos que se generan:**

- ✓ Inventario de recursos.

**Responsable(s):** Líder de Proyecto.

**2.3.4 Análisis del Sistema.**

Una vez identificados todos los requisitos que debe satisfacer el software que se desea adaptar, es muy importante llevar a cabo un análisis de los mismos, para una mejor comprensión, antes de tomar cualquier

decisión del diseño de dicho software y garantizar de esta manera la obtención de una arquitectura robusta, eficaz, eficiente y capaz de sobrevivir a cambios. Se debe realizar un estudio para identificar de forma detallada cada uno de los elementos afectados por el cambio mediante el análisis de la repercusión del mismo en el sistema. Este análisis tiene como objetivo determinar que parte del sistema se ve afectada y en que medida, dejando claramente definido y documentado que componentes hay que modificar. Para llevar a cabo esta fase se proponen las siguientes actividades:

- ✓ Identificar los elementos afectados por el cambio.
- ✓ Documentar los componentes modificados.
- ✓ Identificar las entidades del análisis.
- ✓ Confeccionar el diagrama de clases del análisis.
- ✓ Elaborar diagrama de Interacción para modelar los aspectos dinámicos del sistema.

**Artefactos que se generan:**

- ✓ Modelo de análisis.
- ✓ Lista de componentes modificados.
- ✓ Plan de trabajo actualizado.

**Responsable(s):** Arquitecto.

**2.3.5 Diseño del Sistema.**

En esta fase se obtiene un enfoque de cómo el sistema cumple sus objetivos. La misma tiene lugar después de haberse realizado un análisis del sistema, ya que este análisis proporciona una comprensión detallada de los requisitos. También aplica una estructura del sistema que se debe conservar cuando se le dé forma al sistema. En el diseño se modela el sistema y se encuentra la forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Además el diseño debe ser lo suficientemente explícito para que el software pueda ser implementado sin ambigüedades. En esta fase también se identifican las clases del diseño y se confecciona el diagrama de clases del diseño. Se elaboran los diagramas de secuencia del diseño para cada escenario de cada CU.

Se construye el diagrama de despliegue y se identifican los subsistemas y paquetes de diseño. Para llevar a cabo esta fase se proponen las siguientes actividades:

- ✓ Identificar los componentes reutilizables que serán utilizados en la adaptación de la aplicación.
- ✓ Analizar los documentos de la aplicación para determinar cómo integrar cambios en un módulo existente o bien crear uno nuevo.
- ✓ Analizar los documentos del diseño para determinar dónde añadir, modificar y borrar las funciones que implementan los cambios propuestos.
- ✓ Identificar los elementos del diseño (Clases, Paquetes, Subsistemas del diseño e interfaces).
- ✓ Elaborar el diagrama de clases del diseño.
- ✓ Desarrollar las realizaciones de los CU para determinar cómo se realiza un CU específico y como se ejecuta en términos de CU del diseño.
- ✓ Realizar diagramas de Colaboración.
- ✓ Realizar diagramas de Secuencia.
- ✓ Visualizar la distribución de los componentes de software en los nodos físicos para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos.
- ✓ Establecer un esquema de cómo deben ser los próximos artefactos a construir.
- ✓ Diseñar los casos de prueba de integración.

**Artefactos que se generan:**

- ✓ Modelo de diseño.
- ✓ Modelo de despliegue.
- ✓ Descripción de la arquitectura.
- ✓ Caso de prueba.
- ✓ Plan de trabajo actualizado.

**Responsable(s):** Arquitecto.

### **2.3.6 Implementación.**

Esta fase es llevada a cabo después de realizado el diseño del sistema. En la misma se implementan las clases esbozadas en el diseño. Se hace una planificación para determinar que subsistemas deben ser implementados y en qué orden deben ser integrados, formando el Plan de Integración. El programador decide en qué orden implementa los elementos del subsistema. Si encuentra errores de diseño, los notifica. Además se deben añadir comentarios al código para ayudar a su comprensión posterior, respetar el formato y estilo al hacer los cambios y asegurar que los requisitos originales no modificados no sean afectados. Todos estos aspectos deben ser tomados en cuenta para lograr una calidad superior en la creación de productos de software. Para llevar a cabo esta fase se proponen las siguientes actividades:

- ✓ Planificar para determinar que subsistemas deben ser implementados y en qué orden deben ser integrados.
- ✓ Implementar los subsistemas.
- ✓ Realizar revisiones al código escrito.
- ✓ En caso de existir errores, realizar la modificación correspondiente que contribuya a erradicar el problema detectado.
- ✓ Confeccionar el manual de usuario de la aplicación.

#### **Artefactos que se generan:**

- ✓ Código fuente.
- ✓ Ejecutable.
- ✓ Plan de integración.
- ✓ Manual de usuario.
- ✓ Plan de trabajo actualizado.

**Responsable (s):** Programador.

### 2.3.7 Pruebas de Unidad.

Esta actividad tiene lugar después que se implementan los módulos y está enfocada a realizar pruebas sobre cada módulo del software de manera independiente. El objetivo es comprobar que el módulo, entendido como una unidad funcional de un programa independiente, está correctamente codificado. En estas pruebas cada módulo será probado por separado y el encargado de realizar estas pruebas será el programador. En general, un módulo se entiende como un componente software que cumple las siguientes características:

- ✓ Debe ser un bloque básico de construcción de programas.
- ✓ Debe implementar una función independiente simple.
- ✓ Podrá ser probado al cien por ciento por separado.
- ✓ No deberá tener más de 500 líneas de código.

Para desarrollar esta fase se realizarán las pruebas de funcionalidad del módulo sin atender a su contenido y se va a revisar la estructura del código escrito. Para llevar a cabo esta fase se proponen las siguientes actividades:

- ✓ Utilizar la técnica de caja blanca o de estructura la cual tiene como objetivo diseñar casos de prueba para que se ejecuten, al menos una vez, todas las sentencias del programa, y todas las condiciones tanto en su vertiente verdadera como falsa. Para realizar esta prueba se va a utilizar la siguiente técnica:

- **La prueba del camino básico:** Esta prueba permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. (*Ver anexo 1*).

- ✓ Utilizar la técnica de caja negra o funcional la cual tiene como objetivo encontrar una serie de datos de entrada cuya probabilidad de pertenecer al conjunto de entradas que causan dicho comportamiento erróneo sea lo más alto posible. Para realizar esta prueba se va a utilizar la siguiente técnica:

- **Particiones de Equivalencia:** Esta prueba se encarga de definir casos de prueba que descubran *clases de errores*, reduciendo así el número total de casos de prueba que hay que desarrollar. (*Ver Anexo 2*).



- **Análisis de Valores Límites:** Esta técnica lleva a elegir los casos de prueba que ejerciten los valores límite. (*Ver Anexo 3*).

✓ En caso de existir errores, realizar la modificación correspondiente que contribuya a mejorar el error en cuestión.

**Artefactos que se generan:**

- ✓ Casos de prueba.
- ✓ Resultados de la prueba.

**Responsable (s):** Programador.

**2.3.8 Pruebas de Integración.**

Aún cuando los módulos de un programa funcionen bien por separado es necesario probarlos conjuntamente: un módulo puede tener un efecto adverso o inadvertido sobre otro módulo; las subfunciones, cuando se combinan, pueden no producir la función principal deseada; la imprecisión aceptada individualmente puede crecer hasta niveles inaceptables al combinar los módulos; los datos pueden perderse o malinterpretarse entre interfaces. Esta prueba está orientada a detectar fallos provocados por una incorrecta comunicación entre módulos, es el proceso de combinar y probar múltiples componentes juntos. El objetivo es tomar los componentes probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. Por lo tanto, es necesario probar el software ensamblando todos los módulos probados previamente. Esta actividad tiene lugar después que se realizan las pruebas de unidad. Para llevar a cabo esta fase se proponen las siguientes actividades:

- ✓ Utilizar la estrategia de integración incremental descendente. (*Ver Anexo 4*).
- ✓ En caso de existir errores al integrar, realizar la modificación correspondiente que contribuya a mejorar el problema detectado.

**Artefactos que se generan:**

- ✓ Plan de integración.
- ✓ Resultados de la prueba.

**Responsable (s):** Probador.

### **2.3.9 Pruebas de Sistema.**

Este tipo de pruebas tiene como propósito ejercitar profundamente el sistema para verificar que se han integrado adecuadamente todos los elementos del sistema (hardware, otro software, etc.) y que realizan las funciones adecuadas. Tiene lugar una vez que han sido realizadas las pruebas de integración. Concretamente se debe comprobar que:

- ✓ Se cumplen los requisitos funcionales establecidos.
- ✓ El funcionamiento y rendimiento de las interfaces hardware, software y de usuario.
- ✓ La adecuación de la documentación de usuario.
- ✓ Rendimiento y respuesta en condiciones límite y de sobrecarga.

Este tipo de pruebas se suelen hacer inicialmente en el entorno del programador, denominadas *Pruebas Alfa*, y seguidamente en el entorno del cliente denominadas *Pruebas Beta*. Para llevar a cabo esta fase se proponen las siguientes actividades:

- ✓ Verificar el cumplimiento de los requisitos funcionales utilizando técnicas de caja negra (*Ver Anexo 2 y Anexo 3*).
- ✓ Comprobar el correcto funcionamiento y el buen rendimiento de las interface hardware, software y de usuario utilizando la lista de chequeo de interfaz de usuario. (*Ver Anexo 7*).
- ✓ Revisar la documentación de la aplicación utilizando la lista de chequeo de documentos. (*Ver Anexo 6*).
- ✓ Para evaluar el rendimiento y la respuesta en condiciones límite y de sobrecarga realizar pruebas de stress.
- ✓ En caso de existir errores en los resultados de las pruebas, realizar la modificación correspondiente que contribuya a mejorar el problema detectado.

#### **Artefactos que se generan:**

- ✓ Resultados de la prueba.

**Responsable (s):** Probador.

### **2.3.10 Pruebas de Aceptación.**

Este tipo de prueba tiene lugar cuando el producto está listo para implantarse en el entorno del cliente.

El usuario debe ser el que realice las pruebas, ayudado por las personas del equipo de trabajo, siendo factible, que el mismo usuario sea el que elabore los casos de prueba. Estas pruebas se caracterizan por:

- ✓ Tener el usuario una participación activa, el cual debe ejecutar los casos de prueba ayudado por el equipo de desarrollo.
- ✓ Estar enfocadas a probar los requisitos del usuario. Si no se consigue demostrar que no se cumplen los requisitos o el contrato, el cliente deberá aceptar el producto.
- ✓ Corresponden a la fase final del proceso de desarrollo de software.

Para llevar a cabo esta fase se proponen las siguientes actividades:

- ✓ Realizar las pruebas en el entorno en el que se va a explotar el sistema.
- ✓ Realizar las pruebas a partir de los casos de prueba aportado por el usuario.
- ✓ Si se cumple con lo establecido en el contrato con el cliente, el mismo debe firmar para aceptar el producto.

#### **Artefactos que se generan:**

- ✓ Casos de prueba.
- ✓ Resultado de la prueba.

**Responsable:** Equipo de desarrollo.

### **2.3.11 Realizar Capacitaciones.**

Este fase tiene lugar después que le es entregado el producto al cliente, con el fin de enseñarle como trabajar con las nuevas modificaciones o funcionalidades añadidas a la aplicación existente. Para llevar a cabo esta fase se proponen las siguientes actividades:

- ✓ Entregarle al cliente un manual de usuario que exprese cómo usar la aplicación a partir de los cambios realizados.

- ✓ Planificar y llevar a cabo encuentros con el cliente, con el objetivo de enseñarle a usar la aplicación y si es necesario aclarar alguna duda que le haya surgido durante el uso de la misma.

**Artefactos que se generan:**

- ✓ Plan de Capacitación.

**Responsable (s):** Programador.

**Conclusiones del capítulo.**

En el presente capítulo se realizó la descripción de la propuesta de procedimiento del proceso de adaptación del software del Centro de Tecnología de Gestión de Datos. Con el desarrollo del procedimiento se le dio respuesta a los aspectos definidos en el objetivo general. Se espera que en los resultados de su aplicación se demuestre su eficiencia y calidad.

## CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Una vez elaborada la propuesta de procedimiento para la adaptación de productos y soluciones que desarrolla el Centro de Tecnología de Datos, se procede a validar la eficiencia y la calidad del mismo. La validación se hará empleando el método Delphi utilizando el criterio de un panel de expertos.

### 3.1 Método Delphi.

El método Delphi, cuyo nombre se inspira en el antiguo oráculo de Delphos, fue ideado originalmente a comienzos de los años 50 en el seno del Centro de Investigación estadounidense RAND Corporation por Olaf Helmer y Theodore J. Gordon, como un instrumento para realizar predicciones sobre un caso de catástrofe nuclear. Desde entonces, ha sido utilizado frecuentemente como sistema para obtener información sobre el futuro [20].

Este método extrae y maximiza las ventajas que presentan los métodos basados en grupos de expertos y minimiza sus inconvenientes. Para ello se aprovecha la sinergia del debate en el grupo y se eliminan las interacciones sociales indeseables que existen dentro de todo grupo. De esta forma se espera obtener un consenso lo más fiable posible del grupo de expertos. Este método presenta tres características fundamentales:

✓ **Anonimato:** Durante el Método Delphi, ningún experto conoce la identidad de los otros que componen el grupo de debate. Esto tiene una serie de aspectos positivos, como son:

- Impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro de los miembros o por el peso que supone oponerse a la mayoría. La única influencia posible es la de la congruencia de los argumentos.
- Permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.
- El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.

✓ **Iteración y realimentación controlada:** La iteración se consigue al presentar varias veces el mismo cuestionario. Además, como se van presentando los resultados obtenidos con los cuestionarios

anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.

✓ **Respuesta del grupo en forma estadística:** La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido. [21]

✓ **Heterogeneidad:** Pueden participar expertos de determinadas ramas de actividad sobre las mismas bases.

Este método se caracteriza por tener cuatro fases para su realización:

**Fase 1: Formulación del problema:** En esta fase las preguntas deben ser precisas, cuantificables e independientes.

**Fase 2: Selección de los expertos:** El experto será elegido por su capacidad de encarar el futuro y posea conocimientos sobre el tema consultado.

**Fase 3: Elaboración y envío de los cuestionarios:** Los cuestionarios se elaborarán de manera que faciliten, en la medida en que una investigación de estas características lo permite, la respuesta por parte de los consultados.

**Fase 4: Análisis de los resultados:** Luego de tener los resultados de los cuestionarios enviados a los expertos se procede a realizar un análisis de los resultados obtenidos de las encuestas.

### **3.2 Método de evaluación Criterio de Expertos.**

El panel de expertos puede definirse como un grupo de especialistas independientes y reputados en al menos uno de los campos concernidos por el programa que se va a evaluar, al que se reúne para que emita un juicio colectivo y consensuado sobre dicho programa. Según se les solicite, el juicio emitido puede hacer referencia a la puesta en práctica o a los efectos del conjunto o de una parte del programa [22].

Los métodos de expertos utilizan como fuente de información un grupo de personas a las que se supone un conocimiento elevado de la materia que se va a tratar. Estos métodos se emplean cuando se da alguna de las siguientes condiciones:

- ✓ No existen datos históricos con los que trabajar. Un caso típico de esta situación es la previsión de implantación de nuevas tecnologías.
- ✓ El impacto de los factores externos tiene más influencia en la evolución que el de los internos. Así, la aparición de una legislación favorable y reguladora y el apoyo por parte de algunas empresas a determinadas tecnologías pueden provocar un gran desarrollo de éstas que de otra manera hubiese sido más lento.
- ✓ Las consideraciones éticas o morales dominan sobre las económicas y tecnológicas en un proceso de evolutivo.

Los métodos de expertos tienen las siguientes ventajas:

- ✓ La información disponible está siempre más contrastada que aquella de la que dispone el participante mejor preparado, es decir, que la del experto más versado en el tema. Esta afirmación se basa en la idea de que varias cabezas son mejor que una.
- ✓ El número de factores que es considerado por un grupo es mayor que el que podría ser tenido en cuenta por una sola persona. Cada experto podrá aportar a la discusión general la idea que tiene sobre el tema debatido desde su área de conocimiento. [21]

### **3.3 Selección de los expertos.**

Para seleccionar los expertos que validarán la propuesta de procedimiento se tuvieron en cuenta los siguientes requisitos:

- ✓ Poseer grado científico de ingeniero o superior.
- ✓ Tener conocimiento y experiencia en el tema.
- ✓ Pertener a la Universidad de las Ciencias Informáticas y conocer el modelo de producción de la universidad.
- ✓ Ser buenos informadores (comunicadores).
- ✓ Poseer dotes de análisis (capacidad de reflexión).

De los especialistas seleccionados se hacía necesario conocer sus características para saber si cumplían con los requisitos para validar la propuesta de procedimiento, por lo que se les aplicó un cuestionario (*Ver anexo 8*).

El mismo recoge una serie de datos personales y de conocimientos adquiridos que brindó la información citada a continuación:

**Experto # 1:**

**Labor que realiza:** Asesor de Soporte y Postventa.

**Años de experiencia:** 2 años.      **Especialidad:** Ingeniería en Ciencias Informáticas.

**Categoría docente:** Instructor      **Categoría científica:** Actualmente cursando la Maestría de Gestión de Proyectos.

**Vinculación a proyecto:** Si.

**Roles desempeñados:** Administrador de Calidad, Asesor de Calidad, Analista.

**Años de experiencia en el desarrollo de proyectos de software:** 3 años.

**Experto # 2:**

**Labor que realiza:** Asesor Técnico en Herramientas de Gestión de Proyectos.

**Años de experiencia:** 1 año      **Especialidad:** Ingeniería en Ciencias Informáticas.

**Categoría docente:** Instructor      **Categoría científica:** ---

**Vinculación a proyecto:** No

**Roles desempeñados:** Gerente de Calidad, Probador, Subdirector de Gestión de Negocios, Asesor.

**Años de experiencia en el desarrollo de proyectos de software:** 4 años

**Experto # 3:**

**Labor que realiza:** Asesor Técnico en Herramientas de Gestión de Proyectos.

**Años de experiencia:** 1 año      **Especialidad:** Ingeniería en Ciencias Informáticas

**Categoría docente:** Instructor      **Categoría científica:** ---

**Vinculación a proyecto:** No.

**Roles desempeñados:** Analista.

**Años de experiencia en el desarrollo de proyectos de software:** 3 años.

**Experto # 4:**

**Labor que realiza:** Analista.

**Años de experiencia:** 1 año.      **Especialidad:** Ingeniería en Ciencias Informáticas.



**Categoría docente:** Instructor.

**Categoría científica:** ----

**Vinculación a proyecto:** Si.

**Roles desempeñados:** Analista, Arquitecto, Asegurador de la Calidad, Probador.

**Años de experiencia en el desarrollo de proyectos de software:** 3 años.

**Experto # 5:**

**Labor que realiza:** Especialista.

**Años de experiencia:** **Especialidad:** Ingeniería en Ciencias Informáticas.

**Categoría docente:** Instructor.

**Categoría científica:** ----

**Vinculación a proyecto:** Si.

**Roles desempeñados:** Programador, Analista, Arquitecto.

**Años de experiencia en el desarrollo de proyectos de software:** 5 años.

### 3.4 Elaboración del cuestionario.

El cuestionario (*Ver anexo 9*) elaborado consta de 10 preguntas. El mismo tiene como propósito comprobar mediante el criterio de expertos la validez de la propuesta de procedimiento.

### 3.5 Competencia de los expertos.

Una vez obtenido los resultados del cuestionario enviado a los expertos seleccionados se realizará un análisis para medir el coeficiente de conocimiento de los expertos. La competencia de los expertos se determina por el coeficiente K, el cual se calcula de acuerdo con la opinión de los expertos sobre su nivel de conocimiento sobre el problema que se está resolviendo y con las fuentes que le permiten argumentar sus criterios.

En este cuestionario se le pide al experto que marque con una X, en una escala creciente del 1 al 10, el valor que se corresponde con el grado de conocimiento o información que tienen sobre el tema de investigación.

Expertos	1	2	3	4	5
Kc	0.6	0.8	0.4	0.6	0.5

A partir de estos resultados se calcula el coeficiente de conocimiento o información, a través de la siguiente fórmula:

$$Kc = n (0,1)$$

Donde:

Kc: Coeficiente de conocimiento o información.

n: Rango seleccionado por el experto.

Luego se le hace una segunda pregunta en el cuestionario al experto que permite valorar un grupo de aspectos que influyen sobre el nivel de argumentación o fundamentación del tema de investigación. El coeficiente de argumentación (ka) se obtiene como resultado de la suma de los puntos alcanzados, a partir de la siguiente tabla patrón:

<b>Fuentes de Argumentación o fundamentación</b>	<b>Alto</b>	<b>Medio</b>	<b>Bajo</b>
Análisis teóricos realizados por usted	0.3	0.2	0.1
Su experiencia obtenida	0.5	0.4	0.2
Trabajos de autores nacionales	0.05	0.05	0.05
Trabajos de autores extranjeros	0.05	0.05	0.05
Su conocimiento del estado del problema en el extranjero	0.05	0.05	0.05
Su intuición	0.05	0.05	0.05

Esta tabla patrón se le presenta en la encuesta al experto sin cifras para que marque con una (x), el grado de influencia de las fuentes, de acuerdo a los niveles ALTO, MEDIO y BAJO.

Los aspectos que influyen sobre el nivel de argumentación o fundamentación del tema de investigación permiten calcular el coeficiente de argumentación de cada experto a partir de la siguiente fórmula:

$$Ka = a n_i (n_1 + n_2 + n_3 + n_4 + n_5 + n_6)$$

Donde:

Ka: Coeficiente de argumentación.

a n<sub>i</sub>: valor correspondiente a la fuente de argumentación i (1 hasta 6).

Los resultados son mostrados en la siguiente tabla:

Expertos	1	2	3	4	5
Ka	0.8	0.9	0.7	0.8	0.8

Una vez obtenido los valores del coeficiente de conocimiento (Kc) y el Coeficiente de argumentación (Ka) se procede a obtener el valor del coeficiente de competencia (K) que finalmente es el coeficiente que determina en realidad que experto se toma en consideración para trabajar en esta investigación. Este coeficiente (K) se determina de la siguiente forma:

$$K = 0,5 (Kc + Ka)$$

Posteriormente obtenido los resultados se valoran de la siguiente manera:

0,8 < K < 1,0 Coeficiente de competencia **Alto**.

0,5 < K < 0,8 Coeficiente de competencia **Medio**.

K < 0,5 Coeficiente de competencia **Bajo**.

Finalmente el coeficiente de competencia para cada experto es:

Expertos	1	2	3	4	5
K	0.7	0.9	0.6	0.7	0.8
Coeficiente	Medio	Alto	Medio	Medio	Alto

### 3.6 Análisis de los Resultados del Cuestionario.

Después de haber realizado un análisis de los resultados del cuestionario enviado a los expertos con el fin de validar la propuesta de procedimiento se concluye que:

El 100% de los expertos le confieren una alta relevancia a la investigación realizada, ya que es una temática en creciente desarrollo que es de interés de todos, de aquí que se concluye que se le da un alto valor a la adaptación de productos y soluciones para mejorar el control de la calidad del software.

El 100% de los expertos considera necesaria la implantación del procedimiento para estandarizar el proceso de adaptación de software en el Centro de tecnología de Gestión de Datos.

De igual forma el 100% considera que con la implantación de este procedimiento se mejorará la calidad de los productos y servicios brindados por el Centro de Tecnología de Gestión de Datos.

El 100% considera que están correctamente identificados los roles en el procedimiento aunque uno de los expertos considera que además de los roles propuestos sería bueno adicionar el rol de responsable de calidad.

El 100% de los expertos considera que el procedimiento tiene un orden lógico y las actividades están estructuradas correctamente.

En la pregunta donde se les pide responder si es posible cumplir con las actividades propuestas en el procedimiento definido de adaptación de software el 100% considera que si, ya que la propuesta es sencilla y fácil de llevar a cabo.

El 40% de los expertos consideran que se puede aplicar el procedimiento sin dificultades, que con las actividades descritas en la propuesta es suficiente para su buen desarrollo, mientras que el 60% de los expertos cree que se pueden presentar dificultades durante la aplicación del procedimiento propuesto ya que en muchos casos el personal encargado de realizar los proyectos no tiene la debida disposición a reutilizar componentes o partes de otros sistemas, puede existir rechazo al cambio por parte de las personas involucradas en el desarrollo de los proyectos y otra de las causas puede ser que por la desorganización de las empresas o centros se imposibilite llevar a cabo las actividades con el orden que se propone.

### **Conclusiones del Capítulo.**

En este capítulo se realizó una investigación sobre el Método Delphi. Con el objetivo de validar a través de un comité de experto la propuesta de procedimiento definido en el capítulo anterior. La evaluación del procedimiento propuesto fue realizada por 5 expertos previamente seleccionados según el conocimiento que tenían sobre el tema de investigación y su disposición en participar como experto. Con la valoración realizada por estos expertos se puede concluir que con la aplicación del procedimiento para la adaptación de software quedará estandarizado este proceso en el Centro de Tecnología de Gestión de Datos, dándosele cumplimiento así al objetivo general de la investigación. Por lo que se mejorará el control de la calidad en este proceso.

## CONCLUSIONES

Como resultado de la investigación y el desarrollo del presente trabajo de diploma se llegó a las siguientes conclusiones:

- ✓ Se elaboró una propuesta de procedimiento para la adaptación de los productos y soluciones que desarrolla el Centro de Tecnología de Gestión de Datos, que contribuirá a mejorar el proceso de control de la calidad.
- ✓ El procedimiento propuesto fue evaluado y aceptado usando el Método Delphi con el criterio de un panel de expertos, concluyéndose que:
  - Es muy relevante la investigación realizada.
  - Es necesaria la implantación del procedimiento para estandarizar el proceso de adaptación de software en el Centro de tecnología de Gestión de Datos.
  - Con la implantación de este procedimiento se mejorará la calidad de los productos y servicios brindados por el Centro de Tecnología de Gestión de Datos.

## RECOMENDACIONES

Para este trabajo se recomienda lo siguiente:

- ✓ Aplicar de forma práctica la propuesta de procedimiento en los productos del Centro de Tecnología de Gestión de Datos y analizar los resultados.
  
- ✓ Se continúe el estudio y la profundización de todos los aspectos tratados en el desarrollo de esta investigación, fundamentalmente la adaptación de software y la calidad del software.

## BIBLIOGRAFÍA

1. Real Academia Española. [Online] <http://buscon.rae.es/drael/>.
2. **Melinkoff, Ramón V.** *Los procesos Administrativos*. [ed.] Facultad de ciencias económicas y Sociales, Universidad Central de Venezuela Instituto de Investigaciones. 1968. p. 112. Colección Esquemas.
3. Catering.com.co. [Online] <http://www.catering.com.co/Bancomedios/archivos/normaiso9000.pdf>.
4. **Sommerville, Ian.** *Ingeniería de Software*. 7ma. Madrid : Pearson Addison Wesley, 2005. 84-7829-074-5.
5. **Pressman, Roger.** *Ingeniería de Software. Un Enfoque Práctico*. Quinta edición. Madrid : Concepción Fernández, 2005. p. 640. 84-481-32149.
6. **Valencia, Departamento de sistemas informaticos y computación. Universidad politécnica de.** Proceso de desarrollo del software. [Online] <http://www.dsic.upv.es/asignaturas/facultad/lsi/doc/IntroduccionProcesoSW.doc>.
7. The Free Dictionary. *The Free Dictionary*. [Online] 2010. <http://es.thefreedictionary.com/personalizar>.
8. **Viñals, Juan José.** Daemon Quest. [Online] 03 12, 2005. [http://www.daemonquest.com/es/research\\_and\\_insight/2005/03/12/marketing\\_one\\_to\\_one\\_estrategias\\_de\\_marketing\\_individualizado](http://www.daemonquest.com/es/research_and_insight/2005/03/12/marketing_one_to_one_estrategias_de_marketing_individualizado).
9. IDG.es. [Online] 2010. <http://www.idg.es/iworld/impart.asp?id=120333>.
10. **Vértices, Publicaciones.** *Marketing Digital*. España : Vértice. 978-84-92578-36-8.
11. OUP ESPANA. *OXFORD UNIVERSITY PRESS*. [Online] 2009. <http://www.oup.com/elt/catalogue/teachersites/oald7/?cc=es>.
12. acm. Association for Computing Machinery. [Online] 2010. [Cited: marzo 14, 2010.] <http://www.acm.org/crossroads/espanol/xrds12-4/arqcentric.html>.
13. **Canal, Carlos, Murillo, Juan Manuel and Poizat, Pascal.** Laboratoire D Recherche En Informatique. [Online] 2010. [www.lri.fr/~poizat/documents/publications/CMP06b.pdf](http://www.lri.fr/~poizat/documents/publications/CMP06b.pdf).

14. Grupo Alarcos. [Online] <http://alarcos.inf-cr.uclm.es/per/fruiz/cur/mso/comple/IEEE1219.pdf>.
15. **Miguel Angel Sicilia**. Connexions. *Definiciones del mantenimiento de Software*. [Online] <http://cnx.org/content/m/7404/latest>.
16. Grupo Alarcos. [Online] <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/tema16.pdf>.
17. EudoForge. Innovation for Education. [Online] 2005.  
[https://eduforge.org/docman/view.php/230/3278/E4.4-1-Modelo\\_del\\_usuario.pdf](https://eduforge.org/docman/view.php/230/3278/E4.4-1-Modelo_del_usuario.pdf) .
18. Arquitecturas de Software y Familia de productos. [Online] <http:// triana.escet.urjc.es/aspf/ASPF-Tema1.pdf>.
19. *IPP-3500:2008 Libro de Proceso para Definir Procesos*. La Habana : s.n., 2008. Versión 0.4.
20. Universidad Nacional de Colombia. [Online] 09 2009. [Cited: 05 02, 2010.]  
[http://www.unalmed.edu.co/~poboyca/documentos/documentos1/documentos-Juan2520Diego/Plnaifi\\_Cuencas\\_Pregrado/Sept\\_29/Metodo\\_delphi.pdf](http://www.unalmed.edu.co/~poboyca/documentos/documentos1/documentos-Juan2520Diego/Plnaifi_Cuencas_Pregrado/Sept_29/Metodo_delphi.pdf).
21. Grupo de tecnologías de la información y las comunicaciones. [Online] 2005. [Cited: 05 02, 2010.]  
<http://www.gtlic.ssr.upm.es/encuestas/delphi.htm>.
22. European Commision. [Online] 2010. [Cited: 05 02, 2010.]  
[http://ec.europa.eu/europeaid/evaluation/methodology/examples/too\\_pan\\_res\\_es.pdf](http://ec.europa.eu/europeaid/evaluation/methodology/examples/too_pan_res_es.pdf).



## ANEXOS

### Anexo 1.

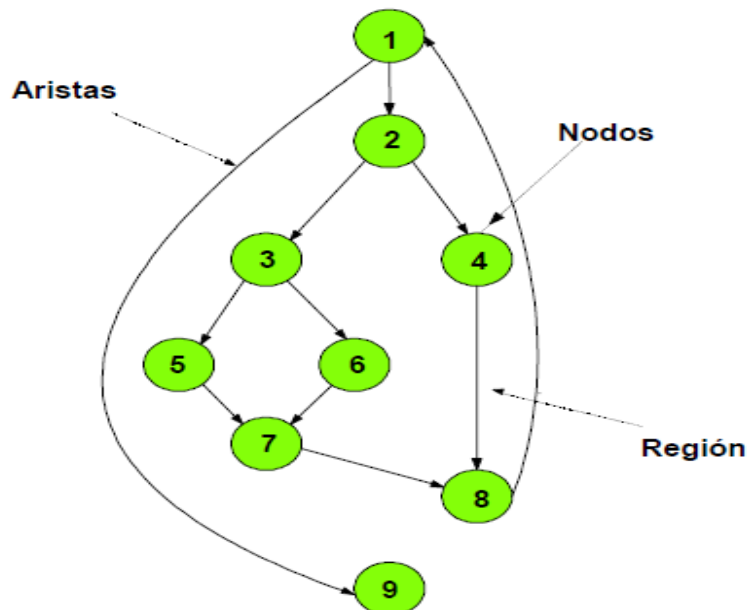
#### Prueba del Camino Básico.

Para realizar esta prueba se siguen los siguientes pasos:

##### 1. Representar el programa en un grafo de flujo.

El grafo de flujo se utiliza para representar flujo de control lógico de un programa. Para ello se utilizan los tres elementos siguientes:

- ✓ *Nodos*: representan cero, una o varias sentencias en secuencia. Cada nodo comprende como máximo una sentencia de decisión (bifurcación).
- ✓ *Aristas*: líneas que unen dos nodos.
- ✓ *Regiones*: áreas delimitadas por aristas y nodos. Cuando se contabilizan las regiones de un programa debe incluirse el área externa como una región más.
- ✓ *Nodos predicado*: cuando en una condición aparecen uno o más operadores lógicos (AND, OR, XOR,...) se crea un nodo distinto por cada una de las condiciones simples. Cada nodo generado de esta forma se denomina nodo predicado.



## 2. Calcular la complejidad ciclomática.

Existen varias formas de calcular la complejidad ciclomática de un programa a partir de un grafo de flujo:

1. El número de regiones del grafo coincide con la complejidad ciclomática,  $V(G)$ .

2. La complejidad ciclomática,  $V(G)$ , de un grafo de flujo  $G$  se define como

$$V(G) = \text{Aristas} - \text{Nodos} + 2$$

3. La complejidad ciclomática,  $V(G)$ , de un grafo de flujo  $G$  se define como

$$V(G) = \text{Nodos Predicado} + 1$$

## 3. Determinar el conjunto básico de caminos independientes.

El conjunto de caminos independientes de un grafo no es único. No obstante, a continuación, se

Muestran algunas heurísticas para identificar dichos caminos:

(a) Elegir un *camino principal* que represente una función válida que no sea un tratamiento de error. Debe intentar elegirse el camino que atraviese el máximo número de decisiones en el grafo.

(b) Identificar el segundo camino mediante la localización de la *primera decisión* en el camino de la línea básica alternando su resultado mientras se mantiene el máximo número de decisiones originales del camino inicial.

(c) Identificar un tercer camino, colocando la primera decisión en su valor original a la vez que se altera la *segunda decisión* del camino básico, mientras se intenta mantener el resto de decisiones originales.

(d) Continuar el proceso hasta haber conseguido tratar todas las decisiones, intentando mantener como en su origen el resto de ellas.

## 4. Derivar los casos de prueba que fuerzan la ejecución de cada camino.

El último paso es construir los casos de prueba que fuerzan la ejecución de cada camino. Una forma de representar el conjunto de casos de prueba.

Número del Camino	Caso de Prueba	Resultado Esperado

## Anexo 2.

### Particiones de Equivalencia.

#### 1. Identificar las clases de equivalencia.

Una clase de equivalencia representa un conjunto de estados válidos y no válidos para las condiciones de entrada de un programa. Se definen dos tipos de clases de equivalencia, las *clases de equivalencia válidas*, que representan entradas válidas al programa, y las *no válidas*, que representan valores erróneos.

En función de cuál sea la condición de entrada se pueden seguir las siguientes pautas identificar las clases de equivalencia correspondientes:

- ✓ Si una condición de entrada especifica un *rango de valores*, identificar una clase de equivalencia válida y dos clases no válidas. Por ejemplo, si un contador puede ir de 1 a 999, la clase válida sería “ $1 \leq \text{contador} \leq 999$ ”. Mientras que las clases no válidas serían “ $\text{contador} < 1$ ” y “ $\text{contador} > 999$ ”.
- ✓ Si una condición de entrada especifica un *valor o número de valores*, identificar una clase válida y dos clases no válidas. Por ejemplo, si tenemos que puede haber desde uno hasta seis propietarios en la vida de un coche. Habrá una clase válida y dos no válidas: “no hay propietarios” y “más de seis propietarios”.

#### 2. Identificar los casos de prueba.

El objetivo es minimizar el número de casos de prueba, así cada caso de prueba debe considerar tantas condiciones de entrada como sea posible. Para crear los casos de prueba a partir de las clases de equivalencia se han de seguir los siguientes pasos:

1. Asignar a cada clase de equivalencia un número único.
2. Hasta que todas las clases de equivalencia hayan sido cubiertas por los casos de prueba, se tratará de escribir un caso que cubra tantas clases válidas no incorporadas como sea posible.
3. Hasta que todas las clases de equivalencia no válidas hayan sido cubiertas por casos de prueba, escribir un caso para cubrir una única clase no válida no cubierta.

Condición de Entrada	Tipo	Clase Equivalencia Válida	Clase Equivalencia No Válida

### **Anexo 3.**

#### **Análisis de Valores Límites.**

El análisis de valores límite complementa la técnica de partición de equivalencia de manera que:

- ✓ En lugar de seleccionar cualquier caso de prueba de las clases válidas e inválidas, se eligen los casos de prueba en los extremos.
- ✓ En lugar de centrarse sólo en el dominio de entrada, los casos de prueba se diseñan también considerando el dominio de salida.

**Las pautas para desarrollar casos de prueba con esta técnica son:**

- ✓ Si una condición de entrada especifica un rango de valores, se diseñarán casos de prueba para los dos límites del rango, y otros dos casos para situaciones justo por debajo y por encima de los extremos.
- ✓ Si una condición de entrada especifica un número de valores, se diseñan dos casos de prueba para los valores mínimo y máximo, además de otros dos casos de prueba para valores justo por encima del máximo y justo por debajo del mínimo.
- ✓ Aplicar las reglas anteriores a los datos de salida.
- ✓ Si la entrada o salida de un programa es un conjunto ordenado, habrá que prestar atención a los elementos primero y último del conjunto.

## **Anexo 4**

### **Integración Descendente (Top-Down):**

Se integran los módulos moviéndose hacia abajo por la jerarquía de control. Comenzando por el módulo principal, los módulos subordinados se van incorporando a la estructura bien, en forma *primero en profundidad*, que integra todos los módulos de un camino de control principal de la estructura, o *primero en anchura*, que incorpora todos los módulos directamente subordinados a cada nivel, moviéndose por la estructura de forma horizontal.

#### **Este proceso se realiza en una serie de cinco pasos:**

1. Se usa el módulo de control principal como controlador de la prueba, disponiendo de resguardos para todos los módulos directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración elegido se van sustituyendo los resguardos subordinados uno a uno por los módulos reales.
3. Se llevan a cabo pruebas cada vez que se integra un nuevo módulo.
4. Tras terminar cada conjunto de pruebas, se reemplaza otro resguardo con el módulo real.
5. Se hace la prueba de regresión para asegurarse de que no se han introducido errores nuevos.

## **Anexo 5.**

### **Prueba de Regresión.**

La regresión consiste en la repetición selectiva de pruebas para detectar fallos introducidos durante la modificación de un sistema o componente de un sistema. Se efectuarán para comprobar que los cambios no han originado efectos adversos no intencionados o que se siguen cumpliendo los requisitos especificados.

#### **En las pruebas de regresión hay que:**

- ✓ Probar íntegramente los módulos que se han cambiado.
- ✓ Decidir las pruebas a efectuar para los módulos que no han cambiado y que han sido afectados por los cambios producidos.

Este tipo de pruebas ha de realizarse, tanto durante el desarrollo cuando se produzcan cambios en el software, como durante el mantenimiento.

## Anexo 6

### Lista de Chequeo de Documentos.

Semántica del documento					
Peso	Indicadores a Evaluar	Eval.	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	¿Ha identificado errores ortográficos?				
Crítico	¿Se entiende claramente lo que se ha especificado en el documento?				
	¿El número de páginas que aparece en el índice coincide con el contenido que se refleja realmente en dicha página?				
	¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento?				

#### LEYENDA:

**Eval.:** Evaluación que se le otorga al aspecto a evaluar: [0, 1, 3, 4,5,].

**NP:** No Procede. Aspecto que no tiene razón para medirse en el Producto, según su clasificación u objetivo.

**Comentarios:** Es obligatorio en las respuestas distintas de 5 puntos.

## Anexo 7

### Lista de Chequeo: Evaluación de la interfaz de usuario.

Nivel	Evaluación	Eval.	NP	Comentarios
!!	¿La interfaz corresponde con las características, metas y nivel de experiencia de los usuarios de esta aplicación?			
!!	¿Está la información libre de errores gramaticales, deletreado, ortográficos y de los errores tipográficos? (estas clases de errores no solamente indican una carencia del control de calidad, pero pueden producir realmente inexactitudes en la información.)			
!	¿Los iconos representan las ventanas dentro de la ventana del padre?			
!	¿Existe el menú de "archivo"?			
!	¿Existe el menú de la "ayuda"?			
!	¿Existe cualquier otro menú que es requerido por el usuario?			
!	¿Los comandos y las opciones apropiadas están en cada menú?			
!!	¿Cada una de las opciones da las respuestas apropiadas?			
!!	¿Se tiene ausencia de errores críticos al probar todas las opciones?			
!	¿Los botones en todas las barras de la herramienta se corresponden con comandos de un menú?			
!	¿Cada comando de menú tiene una secuencia hot-key la cual lo invocará cuando sea apropiado?			
!	En cajas de diálogo "tabuladas", ¿los nombres de la lengüeta o pestañas no son abreviaturas?			



!	En cajas de diálogo "tabuladas", ¿las lengüetas o pestañas se pueden alcanzar vía combinaciones de hot-key apropiadas?			
!	En las cajas dialogadas, ¿no existen hot-key duplicadas?			
!	¿Las lengüetas están colocadas horizontalmente? (evitar de colocar lengüetas verticalmente, son difíciles de leer).			
!	¿El uso apropiado de la tecla escape es correcto? (la cuál debe cancelar cualquier cambio que se haya realizado).			
!	¿Las funciones del botón de cancelación es igual que la tecla escape?			
!	Un botón de comando debe habilitarse cuando la acción se corresponda con su uso y no en otras ocasiones, ¿el botón está habilitado o no lo está cuando le corresponde?			
!	¿Los nombres de los botones de comandos no son abreviaturas?			
!	¿Los nombres de los botones de comandos no son etiquetas técnicas, pero si nombres que tienen un significado para los usuarios del sistema?			
!	¿Los botones de comando son de tamaño y forma similares?			
!	¿Cada botón de comando se puede alcanzar vía combinación de hot-key? (excepto los botones de ACEPTAR y de CANCELAR).			
!	¿Los botones de comandos en la misma caja de Windows/dialog no tienen hot-key duplicadas?			
!	¿Cada caja de Windows/dialog tiene el valor prefijado claramente marcado?			
!	¿El focus está fijado a un objeto?			

!	¿Los nombres de los botones de opción (botón de radio) no son abreviaturas?			
!	¿Los nombres de los botones de opción no sean etiquetas técnicas?			
!	¿Los nombres de la caja de opción no son abreviaturas?			
!	¿Los nombres de la caja de opción no son etiquetas técnicas y son significativas a los usuarios?			
!	¿La ventana del padre tiene una barra de estado?			
!	¿Todos los mensajes del sistema (excepto los errores) para el usuario están presentados vía la barra de estado?			
!	¿La consistencia de acciones del ratón a través de las ventanas es la correcta?			
!	¿Se utiliza un estándar de codificación?			
!	¿El código está debidamente comentado?			

## Anexo 8

### Cuestionario para validar la propuesta.

Usted ha sido seleccionado como experto para evaluar los resultados teóricos de esta investigación. Por lo cual se le pide llenar este cuestionario donde los datos obtenidos contribuirán al resultado exitoso de esta tesis. Sus respuestas serán tratadas de forma anónima.

1. ¿Qué relevancia le confiere a la investigación realizada?

Alta \_\_\_\_ Media \_\_\_\_ Baja \_\_\_\_

2. ¿Considera necesaria la implantación del procedimiento definido para estandarizar el proceso de adaptación de software en el Centro de Tecnología de Datos?

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

3. ¿Considera usted que se mejorará la calidad de los productos y servicios brindados por el centro (DATEC) si se implanta este procedimiento de adaptación de software?

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

4. ¿Considera usted que están correctamente identificados los roles del procedimiento?

Si \_\_\_\_ No \_\_\_\_ ¿Cuál o cuáles considera que se debe incluir? \_\_\_\_

5. ¿Considera usted que el procedimiento tiene un orden lógico y una correcta estructura de las actividades?

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

6. ¿Considera usted que es posible cumplir con las actividades propuestas en el procedimiento definido de adaptación de software?

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

6.1 En caso de que su respuesta sea **No todas** mencione cuales no.

---

7. ¿Cree que la propuesta es tan compleja como para que no se use?

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

8. ¿Qué grado de completitud le confiere usted a esta propuesta de procedimiento?

Alto \_\_\_\_ Medio \_\_\_\_ Bajo \_\_\_\_

9. ¿Cree que se pueden presentar dificultades durante la aplicación del procedimiento definido?

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

9.1 En caso de que su respuesta sea **Si**, mencione cuales.

---

10. Exponga su opinión sobre la propuesta de forma general.

---

---

---

---

## Anexo 9

### Cuestionario para determinar el nivel de competencia de los expertos.

La presente tesis “Procedimiento para la adaptación de productos y soluciones que desarrolla el Centro de Tecnología de Datos” se desea someter a la valoración de un grupo de expertos. Para ello se necesita conocer el grado de dominio que usted posee sobre la adaptación de software, creación de procedimientos y calidad de software; con ese fin se hace necesario que responda las preguntas mostradas a continuación. Se le informa que sus respuestas serán tratadas de forma anónima.

Nombre y apellidos: \_\_\_\_\_

Labor que realiza: \_\_\_\_\_

Años de experiencia: \_\_\_\_\_ Especialidad: \_\_\_\_\_

Categoría docente: \_\_\_\_\_ Categoría científica: \_\_\_\_\_

Vinculación a proyecto: \_\_\_\_\_

Roles desempeñados: \_\_\_\_\_

Años de experiencia en el desarrollo de proyectos de software \_\_\_\_\_

1.- Marque con una cruz (X) el grado de conocimiento o información que usted tiene sobre la temática que se investiga:

1	2	3	4	5	6	7	8	9	10

2.- Marque con una cruz (X) las fuentes que le han servido para argumentar el conocimiento que tiene usted de la temática que se investiga.

<b>Fuentes de Argumentación o fundamentación</b>	<b>Alto</b>	<b>Medio</b>	<b>Bajo</b>
Análisis teóricos realizados por usted			
Su experiencia obtenida			
Trabajos de autores nacionales			
Trabajos de autores extranjeros			
Su conocimiento del estado del problema en el extranjero			
Su intuición			

## GLOSARIO DE TÉRMINOS.

**Almacén de datos:** Es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.

**Adaptación:** Es la acción y efecto de adaptar o adaptarse, hacer que un objeto o un mecanismo cumpla con distintas funciones a aquellas para las que fue construido.

**Calidad:** La calidad es el conjunto de características de un elemento, producto o servicio, que le confieren la aptitud de satisfacer una necesidad implícita y explícita.

**Clase:** Agrupación de objetos con características comunes, tienen un conjunto de atributos para definir los datos que serán guardados por el objeto y un conjunto de métodos que precisan las acciones que este puede realizar.

**Componente:** Parte modular de un sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos.

**CPU:** La unidad central de procesamiento (significado en español). Es el componente en un ordenador, que interpreta las instrucciones y procesa los datos contenidos en los programas de la computadora.

**Data marts:** son pequeños almacenes de datos centrados en un tema o un área de negocio específico dentro de una organización.

**DATEC:** Centro de Tecnologías de Gestión de Datos.

**Diagrama de flujo:** Representación gráfica de la sucesión en que se realizan las operaciones de un procedimiento.

**Diagrama de interacción:** Muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

**Estándar:** Modelo o norma común que se sigue para realizar un proceso.

**IEEE:** Instituto de Ingenieros Electricistas y Electrónicos (Significado en español).

**Inteligencia de Negocio:** Es el conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

**Mantenimiento:** El mantenimiento es la segunda rama de la conservación y se refiere a los trabajos que son necesarios hacer con objeto de proporcionar un servicio de calidad estipulada.

**Modelo de Análisis:** Contiene clases del análisis y sus objetos organizados en paquetes que colaboran.

**Modelo de caso de uso:** Describe los procesos de negocio de una empresa en términos de casos de uso y actores del negocio, que se corresponden con los procesos del negocio y los clientes, respectivamente

**Modelo de despliegue:** Captura los elementos de configuración del procesamiento y las conexiones entre esos elementos. Visualiza la distribución de los componentes de software en los nodos físicos.

**Modelo de diseño:** Es un modelo de objetos que describe la realización de los casos de uso.

**Modelo de negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

**Modelo de Objetos:** Es un modelo de objetos que describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio.

**Nodo:** Elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos.

**Paquete de diseño:** Es una colección de clases, relaciones, realizaciones de casos de uso, diagramas y otros paquetes que estén de alguna forma relacionados.

**PATDSI:** Paquete de ayuda a la toma de decisiones.

**Personalización:** Adaptación de un producto, servicio o contenido a una persona o usuario, en función de sus características, preferencias personales o información previa que proporciona.



**Política:** Es el conjunto de directrices y estándares definidos para un producto o proceso.

**Procedimiento:** Serie común de pasos definidos que permiten realizar un trabajo correctamente. Es la descripción detallada de una actividad predeterminada, que se desarrolla de manera cíclica y donde las condiciones no tienen variaciones que afecten el objetivo principal de la actividad.

**Proceso:** Es un conjunto de actividades o eventos que se realizan o suceden con un fin determinado.

**Redmine:** Es una aplicación web que permite la gestión de proyectos.

**Servicio:** Es el resultado de llevar a cabo necesariamente al menos una actividad en la interfaz entre el proveedor y el cliente, y generalmente es intangible.

**Software:** Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

**Subsistema de diseño:** Son una forma de organizar los artefactos del modelo de diseño en piezas más manejables.

**Tecnología de la Informática y las Comunicaciones (TIC):** Es el empleo de computadoras y aplicaciones informáticas para transformar, almacenar, gestionar, proteger, difundir y localizar los datos necesarios para cualquier actividad humana.