



Universidad de las Ciencias Informáticas

“Facultad 2”

**Título:** Propuesta de modelo para probar la confiabilidad de los productos desarrollados con PostgreSQL en el Centro de Tecnología y Gestión de Datos (DATEC).

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

**Autores:** Yulenia Manso Pampillo.

Yuleidis Rodríguez Cuadra.

**Tutores:** Ing. Ariagnis Yero Guevara.

Ing. Dariena Ramírez Luján.

Cuidad de la Habana, Junio, 2010. Cuba

## Agradecimientos

---

*Le agradezco a mi hermano, a mis padres que tanto se ha esforzado para que mi estancia aquí en la Universidad fuese la mejor. Por todos los sacrificios que han hecho por mí, por darme fuerza y aliento cuando más lo necesitaba. Por confiar en mí en todo momento y no perder la fe. Por siempre recordarme que yo si podía cuando yo pensaba que todo estaba perdido. Por darme la vida, por educarme tan bien como lo han hecho y por quererme tanto. LOS QUIERO MUCHO.*

*Le agradezco a Liudmeylis que era la persona que corría conmigo cada vez que necesitaba alguna información. Por apoyarme en los momentos más difíciles que pasé. Por aguantar mis llantos y levantarme los ánimos siempre, por ser la amiga extraordinaria que ha sido en tan poco tiempo. GRACIAS DE CORAZÓN.*

*Le agradezco a Heydi que fue la persona que nos ayudó y nos sacó adelante cuando pensábamos que todo estaba perdido y no podíamos lograrlo.*

*Le agradezco a Dailien que me ayudó muchísimo con el documento.*

*Le agradezco a Yoandy que muchas veces nos prestó su máquina cuando no teníamos ninguna para poder trabajar y gracias a él pudimos adelantar muchísimas cosas. GRACIAS.*

**Yuleidis**

## Agradecimientos

---

*Le agradezco a mis padres por apoyarme, comprenderme y confiar en mí en todo momento, por darme fuerzas cuando más lo necesitaba y por siempre estar conmigo aún estando lejos, a mi hermano y mi abuelo que me ayudaron mucho con la realización de este trabajo de diploma y que sin ellos no hubiera podido alcanzar mis metas.*

*Le agradezco a mi novio Yoandy que me apoyó y creyó en mí cuando yo misma no creía.*

*Le agradezco a mi compañera de tesis Yuleidis que estuvo a mi lado en todo momento ayudándome en la realización de este trabajo de diploma y supo comprenderme y explicarme con paciencia todo en lo que tuve dudas.*

*Le agradezco a todas mis amistades que tanto dentro de la Universidad como fuera supieron darme ánimos y fuerzas para poder seguir adelante.*

*Le agradezco a Heydis por ayudarnos cuando más lo necesitábamos y cuando nadie más pudo hacerlo.*

*Les agradezco a todas las personas que de un modo u otro contribuyó en mi aprendizaje en la Universidad, permitiendo que pudiera lograr mis metas.*

**Yulenia**

# Resumen

---

El presente trabajo de diploma, tiene como objetivo proponer un modelo que pruebe la confiabilidad de los productos desarrollados con PostgreSQL en el Centro de Tecnología y Gestión de Datos (DATEC).

En este documento quedaron plasmados los resultados del estudio realizado concerniente a las pruebas de software y a algunos de los modelos de calidad más reconocidos en el mundo como son la NC ISO/IEC 9126-1, McCall y Boehm. Se definieron los conceptos de calidad de software, modelos de calidad, la confiabilidad, modelos de prueba y prueba, que son fundamentales para un mejor conocimiento del tema.

El Modelo V fue el modelo de prueba escogido para que fuera la base del Modelo de Confiabilidad V debido a que este modelo permite que se realicen verificaciones y validaciones en toda la etapa de prueba del software, mediante un grupo de pruebas que son realizadas siguiendo la estructura interna del mismo.

Con la propuesta del Modelo de Confiabilidad V se logró obtener un medio para probar la confiabilidad de los productos de software desarrollados con PostgreSQL en DATEC, para de esta forma mejorar la calidad de los mismos.

**Palabras claves:** Calidad de Software, Modelos de Calidad, Modelo de Confiabilidad V, Confiabilidad, Prueba, PostgreSQL.

# ÍNDICE

---

INTRODUCCIÓN.....	1
CAPITULO I. FUNDAMENTACIÓN TEÓRICA.....	6
1.1    Introducción.....	6
1.2    Calidad de Software.....	6
1.2.1    Calidad de Software en Cuba.....	7
1.3    Estándares y Normas.....	8
1.3.1    NC-ISO/IEC 9126.....	8
1.4    Modelos de Calidad.....	9
1.4.1    Modelo de Calidad (NC-ISO/IEC 9126-1).....	10
1.4.2    Modelo de Calidad Boehm.....	11
1.4.3    Modelo de Calidad McCall.....	12
1.5    Modelos de Prueba.....	14
1.5.1    Modelo Cascada.....	14
1.5.2    Modelo V.....	16
1.6    Confiabilidad.....	18
1.7    Gestores de Base de Datos.....	19
1.8    Pruebas.....	21
1.8.1    Niveles de Pruebas.....	21
1.9    Herramientas de Prueba.....	32
1.9.1    JMeter.....	32
1.9.2    DTM Data Generator.....	33
1.9.3    JUnit.....	33
1.10    Conclusiones.....	33
CAPÍTULO II: PROPUESTA DE UN MODELO DE CONFIABILIDAD PARA DATEC.....	34
2.1    Introducción.....	34
2.2    Características del centro DATEC.....	<b>¡Error! Marcador no definido.</b>
2.3    Análisis de las entrevistas realizadas en DATEC.....	34
2.3.1    Resultados de las entrevistas.....	38

2.4	Resultados Científicos .....	38
2.5	Propuesta del Modelo de Prueba para probar la Confiabilidad de los productos desarrollados con Gestores de Base de Datos en DATEC.....	39
2.5.1	Análisis de la propuesta del modelo de prueba para probar la confiabilidad de los productos desarrollados con tecnologías de Base de Datos en DATEC.....	40
2.5.2	Cálculo de la Confiabilidad .....	46
2.5.3	Herramientas de prueba propuestas.....	50
2.6	Conclusiones .....	51
CAPÍTULO III: VALIDACIÓN DE LA PROPUESTA.....		53
3.1	Introducción .....	53
3.2	Evaluación .....	53
3.2.1	Desarrollo de la validación de la propuesta.....	53
3.2.2	Resultado de las encuestas realizadas a los expertos. ....	59
3.3	Aplicación de la propuesta del Modelo de Confiabilidad V en DATEC. ....	60
3.4	Conclusiones .....	66
CONCLUSIONES .....		67
RECOMENDACIONES .....		68
BIBLIOGRAFÍA.....		69
REFERENCIAS BIBLIOGRÁFICAS.....		72
ANEXOS.....		<b>¡Error! Marcador no definido.</b>
GLOSARIO DE TÉRMINOS .....		74

## INTRODUCCIÓN

---

Las empresas que se dedican a la fabricación de productos de software necesitan garantizar su calidad. Esta *“es el grado en el que el producto de software incorpora un conjunto deseable, y definido por la industria, de características para aumentar su desarrollo a lo largo de todo su tiempo y uso”* (ISO, 2008). Existen varias formas para garantizar la calidad, una de ellas es mediante la utilización de varios estándares y normas a partir de los cuales se puedan regir las empresas para de esta forma llevar un control estricto de la calidad.

La Organización Internacional para Estandarización, por sus siglas en inglés International Organization for Standardization (ISO), se ha dedicado a crear normas y estándares los cuales constituyen *“acuerdos documentados y contienen especificaciones técnicas u otros criterios precisos como reglas, lineamientos o definiciones de características para ser usados constantemente”* (ISO, 2008). Dentro de las normas existentes se puede mencionar la ISO/IEC 9126 cuyo propósito va a ser garantizar la calidad del software. En Cuba se hizo una adaptación de la misma y se denominó NC-ISO/IEC 9126 la cual se tendrá en cuenta en el desarrollo de la investigación.

Otra forma de asegurar la calidad de los productos de software es mediante la utilización de los modelos de calidad, los cuales son *“un conjunto de buenas prácticas a seguir durante todo el ciclo de vida del software, enfocados en los procesos de gestión y desarrollo de proyectos”* (Quiñones, 2003 - 2009). Además de los modelos de calidad se utilizan también los modelos de prueba, utilizados para realizarles un conjunto de pruebas al software con el objetivo de encontrar y corregir la mayor cantidad de errores.

La Universidad de las Ciencias Informáticas (UCI) es todavía muy joven en el desarrollo de software, la calidad de estos es tratada con mucha rigurosidad ya que es uno de los aspectos fundamentales a tener en cuenta durante la producción del software. Actualmente en la Universidad no existe ningún modelo a través del cual se pueda probar la confiabilidad de los productos de software, un ejemplo de ello se evidencia en el Centro de Tecnología y Gestión de Datos (DATEC), el cual se enfoca en la investigación y desarrollo de productos dedicados al almacenamiento de datos. Este centro desarrolla sus productos utilizando la tecnología PostgreSQL como Gestor de Base de Datos.

Para poder tener conocimiento de los problemas existentes en dicho centro se realizó una serie de entrevistas, mediante la cuales se obtuvo las dificultades existentes relacionadas con la calidad de los productos de software, específicamente con la confiabilidad y las pruebas que les realizan, como es el caso que las pruebas las realizan en la etapa de liberación de los productos y que no utilizan de forma consiente prácticamente ningún criterio de confiabilidad para medir si los productos de software que desarrollan son confiables. Además de otros problemas que fueron encontrados mediante estas entrevistas, los cuales se analizarán detalladamente más adelante.

No contar con un producto confiable puede traer consigo que el software tenga una alta probabilidad de fallos, desgaste, envejecimiento, que los resultados obtenidos no sean precisos ni correctos, que el producto no pueda retornar a su estado estable en un tiempo determinado a causa de una falla, pérdida de información y tiempo. Todo esto por ende conlleva a la insatisfacción por parte de los clientes. La confiabilidad, es la *“capacidad del producto de software para mantener un nivel de ejecución específico cuando se usa bajo las condiciones especificadas”* (NC-ISO/IEC 9126-1, 2005).

Teniendo en cuenta que tanto en la UCI como en DATEC no existe ningún método a través del cual se pueda probar la confiabilidad de los productos de software y teniendo un conocimiento de las dificultades que existen en este último con respecto a la confiabilidad y los problemas que pueden ocasionar no tener un producto de software confiable, surge la necesidad de diseñar un modelo que pruebe la confiabilidad de los productos teniendo en cuenta la tecnología utilizada en dicho centro.

Se propone fundamentar esta investigación en el siguiente **Problema Científico**:

¿Cómo probar la confiabilidad de los productos desarrollados con PostgreSQL en DATEC?

Identificando como **Objeto de Estudio** del presente trabajo las pruebas de la confiabilidad de los productos desarrollados en DATEC.

El **Campo de Acción** de este trabajo se centró en las pruebas de confiabilidad de los productos desarrollados en DATEC con PostgreSQL.

El **Objetivo General** que se plantea en esta investigación:



Definir un modelo de prueba de la confiabilidad de los productos diseñados con PostgreSQL en DATEC.

Definiéndose así los siguientes **Objetivos Específicos:**

- Caracterizar la situación existente con respecto al proceso de prueba de la confiabilidad de los productos diseñados con PostgreSQL en DATEC.
- Proponer elementos integrantes del modelo de prueba de confiabilidad de los productos diseñados con PostgreSQL en DATEC.
- Aplicar el modelo diseñado en los productos existentes desarrollados por DATEC con PostgreSQL.
- Evaluar la efectividad del modelo diseñado.

Para dar cumplimiento a los objetivos específicos anteriormente planteados se proponen las siguientes

**Tareas de Investigación:**

- Análisis de modelos de calidad tales como ISO/IEC 9126-1:2001, NC ISO/IEC 9126-1:2005, McCall, Boehm.
- Descripción del modelo diseñado acorde a las condiciones del entorno analizado.
- Selección de una muestra de productos desarrollados en DATEC con PostgreSQL para aplicar el modelo diseñado.
- Evaluación del modelo de pruebas de confiabilidad diseñado.

Esta investigación presenta la siguiente **Idea a Defender:**

- La propuesta del modelo de confiabilidad para probar la calidad de los productos desarrollados con PostgreSQL en DATEC garantizará la obtención de productos con mayor calidad.

Como **Posibles Resultados:**

- Investigación acerca de modelos de confiabilidad para probar los productos de software desarrollados en DATEC con PostgreSQL.

- Modelo para efectuar pruebas de confiabilidad a los productos de software desarrollados con PostgreSQL.

Para dar cumplimiento a las tareas planteadas se utilizan los siguientes **Métodos teóricos**:

- **Analítico – Sintético:**

Haciendo un análisis minucioso de las tendencias, documentos y teorías relacionados con la calidad del software. Al tener la cantidad de información necesaria y suficiente, se pudo resumir y sintetizar la misma en busca de los objetivos perseguidos con la investigación. Además de que a partir de ella se pudieron extraer los elementos más importantes concernientes con el presente trabajo.

- **Inductivo – Deductivo:**

La utilización de este método permitió que a partir de conocimientos generales se pudieran derivar casos particulares, luego de haber realizado un razonamiento lógico a partir de la información recopilada. Teniendo en cuenta esto es que se podrá sustentar la idea a defender que se va a tener la investigación.

- **Modelación:**

Este método se refleja mediante la utilización de un modelo como vía de solución de la investigación. La propuesta planteada fue el “**Modelo de Confiabilidad V**” que a partir de este se podrá comprobar la confiabilidad de los productos desarrollados en DATEC.

Como **Métodos empíricos** se utilizan:

- **Observación:**

Este método permitió tener una mejor visión de la situación planteada anteriormente, además de cómo es tratada la calidad en DATEC. También es necesario tener conocimiento de las principales tecnologías de Base de Datos utilizadas en el centro. De esta forma, se conoce que el Gestor de Base de Datos utilizado es PostgreSQL y la confiabilidad no es tratada a fondo cuando desarrollan sus productos.

- **Entrevista:**

Se realizaron entrevistas a diferentes personas que laboran en DATEC con el objetivo de conocer como es tratada la calidad de sus productos mediante la utilización de las pruebas de software. Además de conocer si a la hora de desarrollar sus productos tienen en cuenta la confiabilidad de los mismos.

El presente documento consta de cuatro partes fundamentales: resumen, introducción, desarrollo y conclusiones. El desarrollo está estructurado en 3 capítulos: Fundamentación Teórica, Propuesta de un modelo de confiabilidad para DATEC y Validación de la Propuesta. Estos capítulos están organizados de la siguiente manera:

### **Capítulo I: FUNDAMENTACIÓN TEÓRICA.**

En el capítulo 1 se desarrollan los temas de la investigación referente a los modelos de confiabilidad, gestión de la calidad y gestión de proyectos. Se abordarán aspectos relacionados con la calidad del software en Cuba, en la UCI y específicamente en DATEC que es el principal objetivo. Se expondrán definiciones relacionadas con el Gestor de Base de Datos PostgreSQL; así como los tipos de pruebas que existen y son utilizados para mejorar la calidad de un producto de software.

### **Capítulo II: PROPUESTA DE UN MODELO DE CONFIABILIDAD PARA DATEC.**

En el capítulo 2 se obtiene y explica la propuesta del modelo de prueba para probar la confiabilidad de los productos de software desarrollados con PostgreSQL en DATEC.

### **Capítulo III: VALIDACIÓN DE LA PROPUESTA**

En el capítulo 3 se presentará la validación y aplicación de la propuesta del modelo de prueba y se mostrarán los resultados obtenidos.

## CAPITULO I. FUNDAMENTACIÓN TEÓRICA

---

### 1.1 Introducción

En el mundo de la informática la calidad del software es un aspecto fundamental, puesto que indica que el producto de software obtenido cumple con todos los requisitos impuestos por el cliente. Durante su ciclo de vida, el software pasa por un estricto control de la calidad el cual se lleva a cabo mediante la gestión de la calidad, garantizando de esta forma que el producto de software tenga la eficiencia requerida. La gestión de la calidad es un proceso que se realiza a medida que se va construyendo el software teniendo en cuenta los estándares y normas existentes.

En este capítulo se realizará un estudio de algunos de los modelos de calidad; tales como McCall, Boehm y la NC-ISO/IEC 9126-1 y los modelos de prueba como son el Modelo V y el Modelo Cascada. Además, se darán a conocer conceptos fundamentales para una mejor comprensión de este trabajo investigativo, como son calidad de software, modelos de calidad, confiabilidad, modelos de pruebas y los distintos tipos de pruebas, Base de Datos y PostgreSQL, siendo este el gestor que se utiliza en DATEC.

### 1.2 Calidad de Software

La calidad es *“la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”* (Pressman, 2002). Hablar de calidad de software trae consigo la necesidad de tener parámetros a través de los cuales se puedan establecer los niveles mínimos que un producto de este tipo debe alcanzar para que se considere de excelente calidad.

La calidad es *“el grado en el que el producto de software incorpora un conjunto deseable, y definido por la industria, de características de manera de aumentar su performance a lo largo de todo su tiempo de uso”* (ISO 9001). Esta definición de calidad dada por la ISO 9001 es bastante aceptada puesto que los productos de software deben presentar un conjunto de particularidades que hagan posible que el mismo tenga la calidad y estabilidad requerida por los usuarios. La siguiente definición dada por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) confirma las dos definiciones anteriormente planteadas y dice

que la calidad es *“el grado con el que un sistema, componente o proceso cumplen los requerimientos especificados y las necesidades o expectativas del cliente o usuario”* (IEEE).

Después de haber visto las definiciones anteriores se puede decir que las tres son muy acertadas, pero la que se escoge como guía para la presente investigación es la propuesta por la IEEE, debido a que es de fácil comprensión permitiendo que se pueda entender la necesidad de obtener productos de software con calidad. Cuando se habla de calidad de software es preciso destacar su importancia en todo el mundo y sus aplicaciones. En la actualidad las compañías que se dedican a la producción de software reconocen que la calidad mejora de una manera significativa los productos haciéndolos más confiables y reduce de una forma considerable los costos.

La industria de desarrollo de software no es la excepción, por lo que en los últimos años han elaborado grandes trabajos para utilizar los conceptos de calidad en el ámbito del desarrollo de software y han implantado algunas normas que les permitan controlar la calidad de una forma más eficiente, es el caso de la familia de Normas ISO en especial la ISO/IEC 9126 a la cual se le hizo una adaptación en nuestro país llamada NC-ISO/IEC 9126 que será tratada posteriormente.

### **1.2.1 Calidad de Software en Cuba**

Cuba no está ajena a la importancia que tiene la calidad de software cuando se va a crear un producto, es por ello que se hace un gran énfasis por lograr la calidad de todos los productos de software que se desarrollan a lo largo de todo el país. Se considera que el producto no va a estar completamente terminado hasta que no se asegure que su calidad es óptima, para ello existen varios centros que además de desarrollar software informáticos, hacen que los productos de software cuenten con la calidad requerida por los clientes. Como es el caso de la Empresa Cubana Nacional de Software más conocida por Desoft S.A, el Ministerio de la Informática y las Comunicaciones de Cuba (MIC) que es el organismo rector de estas disciplinas en Cuba y la UCI que es una Universidad que se dedica al desarrollo de software.

La UCI es una Universidad de excelencia, que se dedica al desarrollo de numerosos productos de software de diversas ramas como salud, telecomunicaciones, entre otras. Todos estos software pasan por un estricto control de la calidad, para ello existe un centro en la Universidad que se especializa en la

realización de pruebas de calidad a todos los productos que se desarrollan en ella. Tal es el caso de CALISOFT, el cual está dividido por diferentes departamentos como son el laboratorio de certificación, grupo de auditoría y revisiones, aseguramiento de la calidad y métricas. Cada uno de estos departamentos va a realizar la parte que les corresponde dentro de CALISOFT obteniendo al final un producto con una calidad óptima.

### **1.3 Estándares y Normas**

Los estándares y las normas surgieron debido a la necesidad de asegurar la calidad de los productos. La ISO fue la principal creadora de estos, para que las entidades que se dedicaban a la producción de software obtuvieran productos de alta calidad. Un estándar según la ISO *“son acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados consistentemente como reglas, guías o definiciones de características para asegurar que los materiales, productos, procesos y servicios cumplan con su propósito”* (ISO, 2008). Es el valor de cumplimiento aceptable según el criterio de calidad especificado. Define el rango en el que se admite un nivel de calidad alcanzado en un proceso establecido.

Las normas *“son un modelo, un patrón, ejemplo o criterio a seguir. Una norma es una fórmula que tiene valor de regla y tiene por finalidad definir las características que deben poseer un objeto y los productos que han de tener una compatibilidad para ser usados a nivel internacional”* (Las normas ISO, 1998). La ISO cuando creó estas normas tenía como propósito agrupar un conjunto de características para guiar, ordenar y reducir las pérdidas económicas que trae consigo un producto con mala calidad.

Actualmente su uso se hace más amplio y existe un gran interés en seguir las normas existentes ya que son de gran utilidad porque dan respuesta al reto de las nuevas tecnologías. Otra de las normas estudiadas es la NC-ISO/IEC 9126 que incluye cuatro partes para evaluar la calidad de un producto de software y características que se pueden medir con métricas.

#### **1.3.1 NC-ISO/IEC 9126**

La utilización de las métricas dentro de la evaluación de la calidad de los productos de software es importante. Las métricas son *“una serie de medidas o pasos que ayudan a definir con mayor exactitud el*

*desarrollo y calidad de un producto*” (Pressman, 2002). Estas permiten precisar con detalle cada una de las características del producto evaluado. La ISO 9000 fue publicada el año 1987, trata de estandarizar los sistemas de calidad de las diferentes empresas y sectores. Esta proporciona una guía para la gestión de la calidad y normaliza las verificaciones que se han de realizar sobre el sistema de calidad. Otra de las normas que presenta es la NC-ISO/IEC 9126 que es una de las más utilizadas cuando se quiere evaluar la calidad de un producto.

La ISO, bajo la norma ISO/IEC 9126, implantó un estándar internacional para evaluar y estimar la calidad de productos de software el cual fue publicado en 1992, con el nombre de “Software Product Evaluation”, donde se implantan las características de calidad para productos de software. La NC-ISO/IEC 9126 expresa que todos los componentes de la calidad del software pueden ser descritos teniendo en cuenta las seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales especifican a través de un conjunto de sub-características que permiten profundizar en la evaluación de la calidad de productos de software. En Cuba se le hace una adaptación y surge la NC ISO/IEC 9126 bajo el título de “*Ingeniería de Software – Calidad del Producto*”, La misma consta de cuatro partes para evaluar la calidad:

Parte 1: Modelo de Calidad. 9126-1.

Parte 2: Métricas Externas. 9126-2.

Parte 3: Métricas Internas. 9126-3.

Parte 4: Métricas de Calidad de Uso. 9126-4.

## **1.4 Modelos de Calidad**

Un modelo de calidad es “*un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos. Un modelo de calidad te dice QUE hacer, no COMO hacerlo*” (Quiñones, 2003 - 2009). Se debe tener en cuenta que un modelo de calidad depende de las metodologías que uses y de tus objetivos de negocio. En el mundo existen varios modelos de calidad tales

como los que plantean la NC ISO/IEC 9126-1, el modelo Boehm, McCall que son importantes tenerlos en cuenta para la realización de la investigación.

### 1.4.1 Modelo de Calidad (NC-ISO/IEC 9126-1)

La ISO 9126 - 1 define un modelo de calidad basado en dos partes diferentes:

**Calidad interna y externa:** La calidad interna, entendida como “la totalidad de las características del producto de software, desde un punto de vista interno”, y la calidad externa se define como “la totalidad de las características de producto de software desde un punto de vista externo” influye en la calidad del proceso, al mismo tiempo que la calidad de uso influye sobre las anteriores. La calidad interna, externa y de uso están relacionadas, una se apoya en la otra como capas sucesivas. La calidad del proceso influye en la calidad del producto que a su vez es relevante en la calidad de uso.

Calidad de software (Interna y Externa)					
Funcionalidad	Confiabilidad	Usabilidad	Eficiencia	Mantenibilidad	Portabilidad
Adecuación	Madurez	Fácil comprensión	Comportamiento frente al tiempo	Facilidad de análisis	Adaptabilidad
Exactitud	Tolerancia de fallos	Fácil aprendizaje	Uso de recursos	Capacidad de cambios	Facilidad de instalación
Interoperabilidad	Recuperabilidad	Operatividad	Adherencia a normas	Estabilidad	Coexistencia
Seguridad	Conformidad con la confiabilidad	Software atractivo		Facilidad de pruebas	Facilidad de reemplazo
Adherencia a normas		Adherencia a normas		Adherencia a normas	Adherencia a normas

**Fig. 1: Calidad Interna y Externa**

Esta propone además un modelo categorizando la calidad de los atributos software en seis características (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad), las cuales son subdivididas en sub-características.

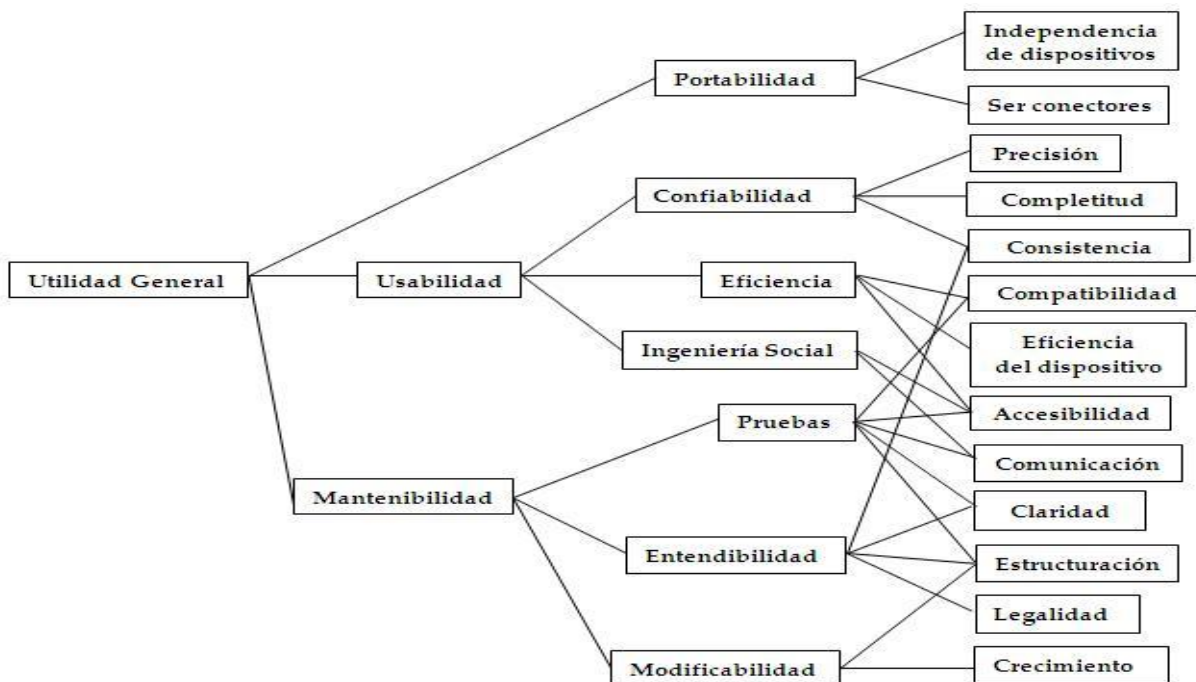
Las sub-características de la confiabilidad que establece la NC-ISO/IEC 9126 son: La Madurez, la Tolerancia a fallas, la Recuperabilidad y la Conformidad con la Confiabilidad. (Ver Fig. 1).



La NC-ISO/IEC 9126 trata la confiabilidad teniendo en cuenta estos factores mediante los cuales se podrá evaluar, probar o medir la confiabilidad de los productos de software desarrollados por cualquier organización o empresa. El modelo planteado por esta norma no es el único que existe, ni el único que es usado para evaluar la calidad de los productos de software. Otro de estos modelos es el Modelo Boehm el cual se explicará brevemente a continuación.

### 1.4.2 Modelo de Calidad Boehm

Otro modelo que es importante resaltar es el de Boehm. Este modelo se destaca por ser uno de los mejor definidos. *“El modelo es de naturaleza jerárquica y los criterios de calidad se presentan en tres grandes divisiones. La primera división es hecha acorde a los servicios que el sistema ofrece (Portabilidad). La segunda se hace de acuerdo a la operación del producto (Usabilidad) y la tercera división se hace de acuerdo a la Mantenibilidad del producto de software”* (Dávila Nicanor, et al., 2008).



**Fig. 2: Modelo Boehm**

Boehm divide la confiabilidad en varios criterios tales como: completitud, consistencia y precisión (Ver Fig. 2).

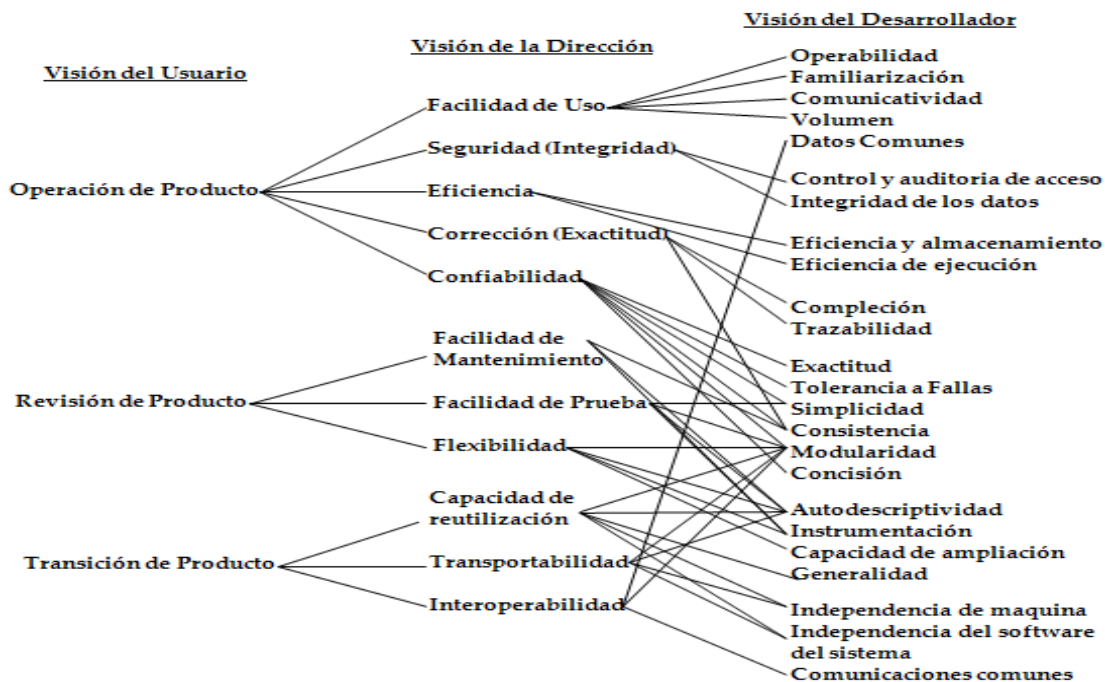
A partir de estos criterios se trata la confiabilidad para buscar la máxima calidad en los productos de software. Otro de los modelos que se estudiará es el de Modelo de Calidad McCall.

### 1.4.3 Modelo de Calidad McCall

La calidad de un producto de software ha sido esencia de trabajo de varios grupos de investigación, de los cuales uno de los más destacados es el modelo de McCall. Fue el primero en ser presentado en 1977. Se centra en el producto final, identificando atributos significativos desde el punto de vista del usuario, los cuales se denominan factores de calidad. Es uno de los más antiguos y extendidos y ha sido tomado como ejemplo y referencia para otros modelos e iniciativas de medida de la calidad del software. Establece tres áreas principales que intervienen en la calidad del software:

- **Calidad en la operación del producto:** Requiere que el software pueda ser entendido fácilmente, que opere eficientemente y que los resultados obtenidos sean los requeridos inicialmente por el usuario.
- **Revisión de la calidad del producto de software:** Tiene como objetivo realizar revisiones durante el proceso de desarrollo para detectar los errores que afecten a la operación del producto.
- **Calidad en el proceso:** Requiere de la definición de estándares y procedimientos que sirven como base para el desarrollo del software.

Los factores de calidad son muy abstractos para medirlos directamente, así que por cada uno de ellos se introduce atributos de bajo nivel denominados criterios de calidad. McCall propone tres perspectivas para agrupar los factores de calidad tales como **revisión del producto, que es la** habilidad que tiene este para ser cambiado, la **transición del producto** que es la adaptabilidad al nuevo ambiente y por último la **operación del producto** que son las características de manipulación del mismo.



**Fig. 3: Modelo de Calidad de McCall.**

El Modelo de McCall define la confiabilidad en varios criterios, estos son: la precisión, la consistencia, la tolerancia a fallos, la modularidad, la simplicidad y la exactitud. (Ver Fig. 3).

Después de haber estudiado todos los modelos de calidad anteriores y haber observado los factores que los componen, se hace énfasis en el factor confiabilidad, siendo este el que se va a tener en cuenta en la presente investigación. Cada uno de estos modelos de calidad define la confiabilidad mediante diferentes criterios los cuales fueron mencionados anteriormente estos son:

- **Madurez:** Permite medir la frecuencia de falla por errores en el software.
- **Tolerancia a fallas:** Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.
- **Recuperabilidad:** Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como el tiempo y el esfuerzo necesario para lograrlo.

- **Conformidad con la Confiabilidad:** La capacidad del producto de software para adaptarse a estándares, convenciones y regulaciones referidas a la confiabilidad.
- **Completitud:** El grado con que se ha logrado la implementación total de una función. Con él se verifica que los requisitos cumplan con las necesidades del cliente.
- **Exactitud:** La precisión de los cálculos, control y emisión de resultados.
- **Precisión:** Atributos del software que posibilitan la implementación de funciones de la forma más comprensible posible.
- **Simplicidad:** Grado de facilidad con que se puede entender un programa. Atributos del software que posibilitan la implementación de funciones de la forma más comprensible posible.
- **Modularidad:** Independencia funcional de los componentes del programa. Atributos del software que proporcionan una estructura de módulos altamente independientes.
- **Concisión:** Lo más compacto que es el programa en términos de líneas de código.
- **Consistencia:** El empleo de un diseño uniforme y de técnicas de documentación a lo largo del proyecto de desarrollo del software.

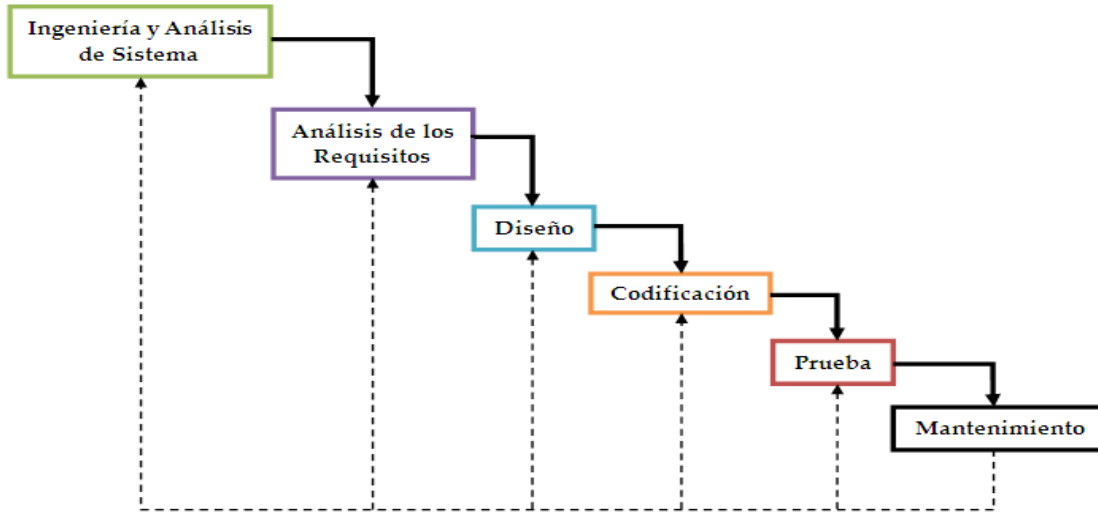
## 1.5 Modelos de Prueba

Un modelo de prueba es un conjunto de pruebas que se realizan ya sea de forma automática o manual, durante todo el ciclo de vida del software o en un espacio determinado. Proponen pruebas en las que cada institución debe hacer énfasis, cumpliendo así con el propósito para el que fue creado. Pueden involucrar pruebas detalladas, así como más generales ofreciendo una visión del sistema en consideración. Integran diferentes prácticas que permiten medir los avances de la calidad.

Existen varios modelos de pruebas de los cuales sólo vamos a hacer énfasis en la presente investigación en dos de ellos, estos son el Modelo en Cascada y el Modelo V, los cuales son explicados a continuación.

### 1.5.1 Modelo Cascada

El modelo de ciclo de vida en cascada (waterfall model) comenzó a diseñarse en 1966 y se terminó de definir alrededor de 1970 por Winston Royce. Se define como una secuencia de fases en la que al final de cada una de ellas se reúne la documentación necesaria para garantizar que cumple las especificaciones y los requisitos antes de pasar a la fase siguiente.



**Fig. 4: Modelo Cascada**

**Desventajas:**

- Los proyectos reales raramente siguen el flujo secuencial que propone el modelo, siempre hay iteraciones y se crean problemas en la aplicación del paradigma.
- Normalmente, es difícil para el cliente establecer explícitamente al principio todos los requisitos. El ciclo de vida clásico lo requiere y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos productos.
- El cliente debe tener paciencia. Hasta llegar a las etapas finales del proyecto, no estará disponible una versión operativa del programa. Un error importante no detectado hasta que el programa esté funcionando puede ser desastroso.

**Ventaja:**

- La ventaja de este método se halla en su simpleza ya que sigue los pasos instintivos necesarios a la hora de desarrollar el software.

Al ser un modelo muy sencillo es ideal para que de él surjan otros modelos; es por esto que a partir de este surge el Modelo V.

## 1.5.2 Modelo V

*“Este modelo considera las pruebas como un proceso que corre en paralelo con el análisis y el desarrollo, en lugar de constituir una fase aislada al final del proyecto”* (García Palomo, et al., 2007). El Modelo V que también se conoce como V-model o modelo de cuatro niveles, es una evolución del Modelo en Cascada y tiene muchas mejoras en comparación con este.

En el Modelo V se diseñan las pruebas en cada fase, este además, como lo dice su nombre tiene forma de V y su primera mitad del lado izquierdo está formado por cuatro fases, la Fase de Requerimientos, Fase de Diseño del Sistema, Fase de Arquitectura y por último la de Implementación. La otra mitad, compuesta por cuatro niveles de pruebas: Unidad, Integración, Sistema y Aceptación, tiene como finalidad hacer pruebas correspondientes a cada una de las fases asociadas a la mitad anterior, demostrando la relación que existe entre las fases del ciclo de vida del desarrollo del software y las pruebas que se le realizan.

Para cada una de las fases, se encuentra un nivel de prueba correspondiente o paralela de verificación o validación, respetando así el principio de que para cada fase del desarrollo debe existir un resultado verificable. Esta estructura también indica que la proximidad entre una fase del desarrollo del software y su fase de verificación correspondiente va decreciendo a medida que aumenta el nivel dentro de la V. La longitud de esta separación intenta ser proporcional a la distancia en el tiempo entre una fase y su similar de verificación (Ver fig. 5).

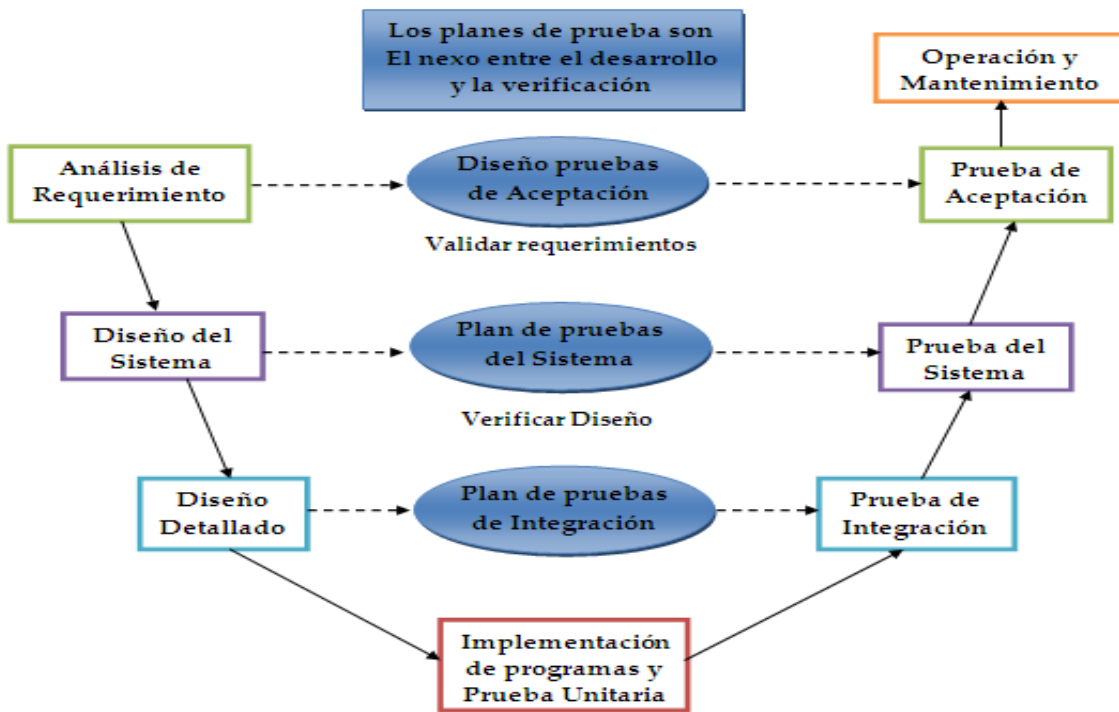


Fig. 5: Modelo V

### Ventajas

- En este modelo V se aprecia una ventaja fundamental con respecto al Modelo Cascada, y se refiere a que este modelo incluye chequeos de cada una de las fases del modelo de cascada.
- El modelo V despliega un método bien estructurado en el cual cada fase se puede poner en práctica por la documentación detallada de la fase anterior. Las acciones de prueba como la de la Fase Diseñar; su verificación comienza al principio del proyecto bien antes de la Fase de Implementación y por lo tanto ahorran una cantidad enorme del tiempo del proyecto.

### Desventajas:

- El modelo no contempla la posibilidad de retornar a etapas inmediatamente anteriores, cosa que en la realidad puede ocurrir.

- Se toma toda la complejidad del problema de una vez y no en iteraciones o ciclos de desarrollo, lo que aumenta el riesgo.

El Modelo V se tendrá en cuenta en capítulos posteriores para cumplir con el objetivo de la investigación porque a partir de este es que se desarrollará la propuesta final del presente trabajo.

Una vez que fueron estudiados estos modelos se decide utilizar el Modelo V para la realización de la presente investigación debido a que el mismo en si es una mejora del Modelo Cascada. El Modelo V permite que se puedan hacer validaciones y verificaciones en conjunto a los productos de software que se desarrollen. Este brinda la posibilidad de diseñar las pruebas en cada fase que serán realizadas posteriormente en la etapa de prueba, mientras que con el Modelo Cascada hay que esperar a que el software sea terminado para diseñar las pruebas y aplicarlas también.

Con la utilización de los modelos de calidad que fueron explicados anteriormente se puede saber si el producto final presenta la calidad esperada por el cliente. Actualmente uno de los términos más importantes cuando se habla de calidad es la confiabilidad, que centra su atención en la probabilidad de fallos en los productos de software.

## **1.6 Confiabilidad**

En la actualidad la confiabilidad es uno de los principales aspectos que se tienen en cuenta cuando se construye un producto de software. A través de esta se pueden predecir las posibilidades de fallo de cualquier producto, de esta forma, se conoce con antelación los posibles errores y deficiencia que pueden presentar los productos. Esta contiene sub-características que permiten profundizar en la evaluación de la calidad de los productos de software.

La confiabilidad *“es la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido”* (Figuroa, 2006). Es una medida del grado de operabilidad y capacidad de un sistema para prestar el servicio requerido en cualquier momento de su tiempo de misión, suponiendo su disponibilidad en el instante inicial. Agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido.



Teniendo en cuenta los modelos de calidad estudiados anteriormente; escogidos porque en estos se tiene en cuenta la confiabilidad, que es uno de las principales vías a seguir para poder lograr productos con alta calidad y sumamente confiables. La confiabilidad se ha convertido en una necesidad o requisito fundamental para los productos de software. La insuficiencia de estas medidas se refleja en la frecuencia de fallos de un sistema software, ya sea este de almacenamiento de datos o un software de gestión. En el presente trabajo las pruebas de confiabilidad estarán enmarcadas en tecnologías de Base de Datos, de forma tal que se pueda confiar justificadamente en sus servicios.

## **1.7 Gestores de Base de Datos**

Las Bases de Datos surgieron debido a la necesidad de almacenar y administrar la información que se iba obteniendo a lo largo de los años de forma tal que un programa de computadora pueda obtener la información cuando lo necesite. *“Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesiten”* (Masadelante, 1999 - 2010). Las Bases de Datos pueden considerarse como una recopilación de datos que varían con el tiempo y que están relacionados entre sí y almacenados en la computadora.

Constituyen una importante herramienta para almacenar y recuperar la información, para utilizar esta herramienta es necesario ofrecer una interfaz al usuario que elimine la complicación de su uso. Sin esta interfaz la Base de Datos puede seguir utilizándose pero el usuario ha de conocer el lenguaje de la misma para hacer uso de ésta.

### **Gestor de Base de Datos que se utiliza en DATEC**

En este centro el gestor de Base de Datos que se utiliza es PostgreSQL, aunque se tienen planes y contratos con algunos clientes, para utilizar también como gestor Oracle para el desarrollo de productos de software, pero hasta el momento sólo se utiliza PostgreSQL. Este es muy utilizado debido a la gran variedad de ventajas que presenta explicadas más adelante. En la UCI este es uno de los gestores más utilizados debido a que su licencia es gratis y es de código abierto. Las ventajas que este presenta son:

- **Instalación Ilimitada**

Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software. Esta ventaja es una de las principales razones por la que este gestor es utilizado en el centro, ya que permite ser modificado o usado sin restricciones que impidan su cambio.

- **Soporte**

Además de las ofertas de soporte, se tiene una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios y contribuir. El uso de esta comunidad de profesionales permite que PostgreSQL esté en un constante cambio y perfeccionamiento, haciendo que se tenga una variada información del mismo.

- **Ahorros considerables en costos de operación**

PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.

- **Estabilidad y Confiabilidad Legendarias**

Es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. Ni una sola vez. Simplemente funciona. Esto hace que los productos que se desarrollan en con este gestor tengan una alta probabilidad de no presentar ningún problema con él durante su desarrollo.

- **Extensible**

El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esta ventaja de PostgreSQL es lo que le permite que haya tantos fanáticos de este y que se puedan hacer cambios según se necesite.

- **Multiplataforma**

PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y ahora en versión nativa para Windows.

- **Diseñado para ambientes de alto volumen**

PostgreSQL está diseñado para almacenar un gran volumen de información y esta es una de las principales ventajas por la cual las empresas prefieren utilizarlo en lugar de otro gestor de Base de Datos.

- **Herramientas gráficas de diseño y administración de BD**

Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Data Architect).

## **1.8 Pruebas**

Las pruebas son un elemento muy importante para garantizar la calidad del software. Su principal objetivo va a ser obtener la mayor cantidad de errores posible. Las pruebas deben ocurrir durante todo el ciclo de vida del proyecto y se van a enfocar sobre la lógica interna del software y las funciones externas, una prueba tiene éxito si se descubre un nuevo error, es por eso que un buen caso de prueba es aquel que tiene alta probabilidad de mostrar un error no descubierto hasta entonces.

Cuando se realizan las pruebas no se puede asegurar la ausencia de defectos; sólo se puede demostrar que existen errores en el software. Las pruebas deben empezar por lo pequeño y avanzar hacia lo grande y planificarse mucho antes de que empiecen. Las pruebas se van a dividir por niveles y dentro de cada nivel de prueba van a existir otras que van a perfeccionar las pruebas de ese nivel, haciendo que estas sean más completas.

### **1.8.1 Niveles de Pruebas**

Al dividir las pruebas por niveles se logra tener una mejor organización de las pruebas que se hacen en cada nivel teniendo un orden a seguir. Las pruebas que se tendrán en cuenta son las Pruebas de Unidad, Integración, Sistema y por último las de Aceptación.

## Pruebas de Unidad

En las **Pruebas de Unidad** se prueban los caminos de control importantes con el fin de descubrir errores dentro del límite de este. Está orientado a **Pruebas de Caja Blanca** y se pueden realizar pruebas en varios Módulos paralelamente. Su propósito es encontrar errores en la lógica, datos o algoritmos en componentes o subsistemas individuales. También se pueden realizar **Pruebas de Caja Negra** a los casos de uso.

En la prueba de unidad se utilizan las pruebas de caja blanca y las pruebas de caja negra, que son tipos de pruebas que nos van a ayudar a comprobar si un producto de software tiene algún error y los que menor cantidad de esfuerzo y de tiempo van a requerir al hacer las pruebas.

- **Las pruebas de Caja Blanca** permiten examinar la estructura interna del programa, aseguran que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han probado adecuadamente. Al usar este tipo de prueba se intenta garantizar que se obtengan todos los casos de prueba y que todos los caminos de ejecución del programa quedan probados.

***Algunos de los métodos empleados en las pruebas de caja blanca son los siguientes:*** (Pérez, 2010)

1. **Pruebas del camino básico:** Es una técnica que 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. Los casos de prueba obtenidos del conjunto básico garantizarán que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.
2. **Pruebas de condición:** Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa, para que todas las condiciones del programa se evalúen a cierto/falso.
3. **Pruebas de bucles:** Es una técnica que se centra exclusivamente en la validez de las construcciones de bucles (bucles simples, anidados, concatenados y no estructurados).

***Algunos de los métodos empleados en las pruebas de caja negra son los siguientes:***

1. **Prueba de los valores límite:** Los errores tienden a darse más en los límites del campo de entrada que en el centro. Por ello, se ha desarrollado el análisis de valores límite (AVL) como técnica de prueba. El análisis de valores límite lleva a una elección de casos de prueba que ejerciten los valores límite.
2. **El análisis de valores límite:** Es una técnica de diseño de casos de prueba que completa a la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la elección de casos de prueba en los extremos de la clase. En lugar de centrarse solamente en las condiciones de entrada, el AVL obtiene casos de prueba también para el campo de salida.
3. **Prueba de la partición equivalente:** Presenta la partición equivalente como un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.
4. **Prueba de la tabla ortogonal:** Aplicaciones donde el número de parámetros de entrada es pequeño y los valores de cada uno está claramente delimitado. Cuando estos números son muy pequeños, es posible considerar cada permutación de entrada y comprobar exhaustivamente el proceso del dominio de entrada. En cualquier caso, cuando el número de valores de entrada crece y el número de valores diferentes para cada elemento de dato se incrementa, la prueba exhaustiva se hace impracticable.
5. **La prueba de la tabla ortogonal:** Puede aplicarse a problemas en que el dominio de entrada es relativamente pequeño pero demasiado grande para posibilitar pruebas exhaustivas. El método de prueba de la tabla ortogonal es particularmente útil al encontrar errores asociados con fallos localizados, una categoría de error asociada con defectos de la lógica dentro de un componente software.

6. **Adivinando el error:** Dado un programa particular, se conjetura, por la intuición y la experiencia, ciertos tipos probables de errores y entonces se escriben casos de prueba para exponer esos errores. Es difícil dar un procedimiento para esta técnica puesto que es en gran parte un proceso intuitivo.

***Dentro de estas pruebas los tipos de errores que se encuentran son:***

1. Funciones incorrectas o ausentes.
2. Errores de interfaz.
3. Errores de estructuras de datos o en accesos a bases de datos externas.
4. Errores de rendimiento.
5. Errores de inicialización o de terminación.

### **Pruebas de Integración (Mancha, 2005)**

**Las Pruebas de integración** cogen los módulos probados en la prueba de unidad y construyen una estructura de programa que esté de acuerdo con lo que dicta el diseño. En la prueba de integración el foco de atención es el diseño y la construcción de la arquitectura del software y las técnicas que más prevalecen son las de diseño de casos de prueba de caja negra, aunque se pueden llevar a cabo unas pocas pruebas de caja blanca. Existen dos formas de integración:

1. **Integración no incremental:** Se combinan todos los módulos por anticipado y se prueba todo el programa en conjunto.
2. **Integración incremental:** El programa se construye y se prueba en pequeños segmentos.

### **Pruebas del Sistema**

**Prueba del sistema** verifica que cada elemento encaja de forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total. La prueba del sistema está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Algunas de estas pruebas son:

## **1. Prueba de integridad de los datos y la base de datos**

### ***Objetivo de la prueba***

Asegurar que los métodos y procesos de acceso a la base de datos funcionan correctamente y sin corromper datos.

### ***Técnica***

- Invoque cada método o proceso de acceso a la base de datos con datos válidos y no válidos.
- Inspeccione la base de datos para asegurarse de que se han guardado los datos correctos, que todos los eventos de la base de datos ocurrieron correctamente, o repase los datos devueltos para asegurar que se recuperaron datos correctos por la vía correcta.

## **2. Prueba de Funcionalidad**

La prueba de funcionalidad se enfoca en requerimientos para verificar que se corresponden directamente a casos de usos o funciones y reglas del negocio. Los objetivos de estas pruebas son verificar la aceptación de los datos, el proceso, la recuperación y la implementación correcta de las reglas del negocio. Este tipo de prueba se basa en técnicas de caja negra, que consisten en verificar la aplicación y sus procesos interactuando con la aplicación por medio de la interface de usuario y analizar los resultados obtenidos.

### ***Objetivo de la prueba***

Asegurar la funcionalidad apropiada del objeto de prueba, incluyendo la navegación, entrada de datos, proceso y recuperación.

### ***Técnica***

Ejecute cada caso de uso, flujo de caso de uso, o función usando datos válidos y no válidos, para verificar lo siguiente:

- Se obtienen los resultados esperados cuando se usan datos válidos.

- Cuando se usan datos no válidos se despliegan los mensajes de error o advertencia apropiados.
- Se aplica apropiadamente cada regla del negocio.

### **3. Prueba de Performance**

En esta prueba se miden y evalúan los tiempos de respuesta, los tiempos de transacción y otros requerimientos sensitivos al tiempo. El objetivo de la prueba es verificar que se logren los requerimientos de performance. La prueba de performance es implementada y ejecutada para poner a punto los destinos de pruebas de performance como función de condiciones de trabajo o configuraciones de hardware.

#### ***Objetivo de la prueba***

Verificar la performance de determinadas transacciones o funciones de negocio bajo ciertas condiciones:

- Condiciones de trabajo normales conocidas.
- Peores casos de condiciones de trabajo conocidas.

#### ***Técnica***

- Usar procedimientos de prueba desarrollados para verificar funciones o ciclos de negocio.
- Modificar archivos de datos para aumentar el número de transacciones o los procedimientos de prueba para aumentar el número de iteraciones de ocurrencia de transacciones.
- Las pruebas se deben ejecutar en una máquina (mejor caso de prueba un solo usuario, una sola transacción) y se debe repetir con múltiples usuarios (virtuales o reales).

### **4. Prueba de Carga**

La prueba de carga somete los objetos a verificar a diferentes cargas de trabajo para medir y evaluar los comportamientos de performance y la habilidad de los objetos de continuar funcionando



apropiadamente bajo diferentes cargas de trabajo. El objetivo es determinar y asegurar que el sistema funciona apropiadamente en circunstancias de máxima carga de trabajo esperada. Además, evaluar las características de performance, como tiempos de respuesta, tiempos de transacciones y otros elementos sensitivos al tiempo.

### ***Objetivo de la prueba***

Verificar el comportamiento de performance de determinados componentes del software bajo condiciones de trabajo diferentes.

### ***Técnica***

Usar pruebas desarrolladas para funciones o ciclos de negocios y modificar archivos de datos para aumentar el número de transacciones o las pruebas para aumentar la cantidad de ocurrencia de transacciones.

## **5. Prueba de Esfuerzo (stress, competencia por recursos, bajos recursos)**

La prueba de esfuerzo es un tipo de prueba de performance implementada y ejecutada para encontrar errores cuando hay pocos recursos o cuando hay competencia por recursos. Poca memoria o poco espacio de disco pueden revelar fallas en el software que no aparecen bajo condiciones normales de cantidad de recursos. Otras fallas pueden resultar al competir por recursos compartidos como bloqueos de bases de datos o ancho de banda de red. La prueba de esfuerzo también puede usarse para identificar el trabajo máximo que el software puede manejar.

### ***Objetivo de la prueba***

El objetivo de la prueba de esfuerzo es identificar y documentar las condiciones bajo las cuales el sistema falla y no continua funcionando apropiadamente. Verificar que el software funciona apropiadamente y sin error bajo condiciones de esfuerzo, como son:

- Poca memoria o sin disponibilidad de memoria en el servidor.
- Cantidad máxima de clientes conectados.

- Múltiples usuarios realizando la misma operación sobre los mismos datos.
- Peor caso de volumen de operaciones.

### ***Técnica***

- Usar las pruebas desarrolladas para Performance y Prueba de Carga.
- Para probar recursos limitados, las pruebas se deben ejecutar en una sola máquina, y se debe reducir o limitar la memoria en el servidor.
- Para las pruebas de esfuerzo restantes, deber usarse múltiples clientes, cualquiera que ejecute las mismas pruebas o pruebas complementarias para producir el peor caso de volumen de operaciones.

## **6. Prueba de Volumen**

La Prueba de Volumen somete el software a grandes cantidades de datos para determinar si se alcanzan límites que causen la falla del software. La Prueba de Volumen identifica la carga máxima continua que puede manejar el software a prueba en un período dado.

### ***Objetivo de la prueba***

Verificar que el software funciona correctamente con volúmenes de datos grandes:

- Máximo (real o físicamente posible) número de clientes conectados, o simulados, todos realizando la misma operación (peor caso de operación) por un período de tiempo extenso.
- Máximo tamaño de base de datos y múltiples consultas ejecutadas simultáneamente.

### ***Técnica***

Usar pruebas desarrolladas para Prueba de Performance y Prueba de Carga.

## **7. Prueba de Fallas y Recuperación**

Las Pruebas de Fallas y Recuperación aseguran que el software puede recuperarse de fallas de hardware, software o mal funcionamiento de la red sin pérdida de datos o de integridad de los datos.

La Prueba de Recuperación es un proceso en el cual la aplicación o sistema se expone a condiciones extremas, o condiciones simuladas, para causar falla, como fallas en dispositivos de Entrada/Salida o punteros a la base de datos inválidos. Los procedimientos de recuperación se invocan y la aplicación o sistema es monitoreado e inspeccionado para verificar que se recupera apropiadamente la aplicación o sistema y se logre la recuperación de datos.

### ***Objetivo de la prueba***

Verificar que los procesos de recuperación (manual o automáticos) recuperen apropiadamente la base de datos, aplicaciones y sistema a un estado conocido y deseado. En la prueba se incluyen los siguientes tipos de condiciones:

- Interrupción de energía al cliente.
- Interrupción de energía al servidor.
- Interrupción de comunicaciones mediante los servidores de la red.
- Interrupción de comunicación o pérdida de energía de los discos del servidor o con los controladores.
- Ciclos incompletos (procesos de filtro de datos interrumpidos, procesos de sincronización de datos interrumpidos).
- Punteros a la base de datos o claves inválidos.
- Elementos de datos en la base de datos inválidos o corruptos.

### ***Técnica***

Se deben usar las pruebas creadas para probar Funcionalidad y Ciclos de negocio para crear una serie de operaciones. Una vez logrado el punto de comienzo deseado, se deben realizar o simular las siguientes acciones, individualmente:

- Interrumpir la energía del cliente: apagar el PC.
- Interrumpir la energía del servidor: simular o iniciar el proceso de apagado del servidor.
- Interrupción por medio de los servidores de red: simular o iniciar la pérdida de comunicación con la red (desconectar físicamente la comunicación o apagar el servidor de red o router.
- Interrumpir la comunicación o quitar la energía de los discos del servidor o sus controladores: simular o eliminar físicamente a la comunicación con uno o más controladores de disco o los discos.
- Una vez que se lograron o simularon estas condiciones, se deben invocar los procedimientos de recuperación.
- Las pruebas de ciclos incompletos utilizan la misma técnica excepto que los procesos de bases de datos deben ser abortados a sí mismos o terminados prematuramente.
- Las últimas dos pruebas requieren que se logre un estado conocido de la base de datos. Se deben corromper manualmente campos de la base de datos, punteros y claves trabajando directamente sobre la base de datos (utilizando herramientas para la base de datos). Se deben ejecutar las pruebas de Funcionalidad y Ciclo de negocio y verificar que los ciclos se completen.

## **8. Prueba de Seguridad y Control de Acceso**

Se enfoca en dos áreas de seguridad:

- Seguridad en el ámbito de aplicación, incluyendo el acceso a los datos y a las funciones de negocios.
- Seguridad en el ámbito de sistema, incluyendo conexión, o acceso remoto al sistema.

La seguridad en el ámbito de aplicación asegura que, basado en la seguridad deseada los actores están restringidos a funciones o casos de uso específicos o limitados en los datos que están disponibles para ellos.

La seguridad en el ámbito de sistema asegura que, solo los usuarios con derecho a acceder al sistema son capaces de acceder a las aplicaciones y solo a través de los puntos de ingresos apropiados.

### ***Objetivo de la prueba***

- **Seguridad en el ámbito de aplicación:** Verificar que un actor pueda acceder solo a las funciones o datos para los cuales su tipo de usuario tiene permiso.
- **Seguridad en el ámbito de sistema:** Verificar que solo los actores con acceso al sistema y a las aplicaciones, puedan acceder a ellos.

## **9. Pruebas de Usabilidad**

Se determina la calidad de la experiencia de un usuario en la forma en la que éste interactúa con el sistema, se considera la facilidad de uso y el grado de satisfacción del usuario.

### **Pruebas de Aceptación**

Son las que hace el cliente, se determina que el sistema cumple con lo deseado y se obtiene la conformidad del cliente.

### ***Objetivo de la prueba***

- Si un sistema satisface sus criterios de la aceptación o no.
- Para permitir al cliente determinarse si aceptar el sistema o no.
- Para verificar el sistema está acorde con las necesidades originales.

### ***Procedimientos para conducir la prueba de aceptación:***

- El primer paso es definir los criterios de la aceptación, ejemplo:

- Requisitos de la funcionalidad.
- Requisitos de funcionamiento.
- Requisitos de calidad del interfaz.
- Requisitos de calidad del software.

Para realizar las pruebas es de gran ayuda el uso de algunas herramientas de pruebas. Muchas son diferentes de la otra, para variadas funciones y diversos lenguajes de programación. Aunque hay muchas herramientas que surgen de otras ya existentes, según las necesidades que vayan surgiendo para su uso, cada una con un objetivo específico, facilitar la realización de las pruebas.

## **1.9 Herramientas de Prueba**

En este epígrafe se dan a conocer varias herramientas de pruebas, con la expectativa de que sean utilizadas en el centro DATEC una vez que sean cumplidos los objetivos de la presente investigación. Estas se proponen para que sirvan de ayuda para la aplicación de la propuesta, del centro depende su utilización o no.

### **1.9.1 JMeter**

Apache JMeter “es una aplicación 100% puro Java de escritorio diseñado para cargar el comportamiento de pruebas funcionales y medir los resultados. Originalmente se diseñó para probar aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba” (Licea Castellanos, 2009). Apache JMeter puede ser utilizado para probar el rendimiento tanto en los recursos estáticos y dinámicos (archivos, Servlets, scripts de Perl, Java Objects, bases de datos y consultas, servidores FTP y mucho más).

Puede ser utilizado para simular una carga pesada en un servidor, la red o un objeto para poner a prueba su resistencia o para analizar el rendimiento global por diferentes tipos de carga. Puede usarlo para hacer un análisis gráfico de rendimiento o para probar su servidor/script/ comportamiento del objeto con carga pesada concurrentes.

## 1.9.2 DTM Data Generator

Es una potente y totalmente personalizable herramienta, que genera los datos, tablas, vistas, procedimientos para la base de datos con fines de prueba: las pruebas de rendimiento, pruebas de control de calidad, la carga de ensayos o pruebas de usabilidad. La gran ventaja de este programa es que es capaz de crear una amplia variedad de tablas de prueba y soporta plantillas definidas por el usuario. Estos datos pueden ser fácilmente insertados dentro de una base de datos.

## 1.9.3 JUnit

JUnit *“se trata de un Framework Open Source para la automatización de las pruebas (tanto unitarias, como de integración) en los proyectos de Software. El framework provee al usuario de herramientas, clases y métodos que le facilitan la tarea de realizar pruebas en su sistema y así asegurar su consistencia y funcionalidad”* (Licea Castellanos, 2009). Esta es una herramienta que se utiliza fundamentalmente para la aplicación de pruebas en java, pero su facilidad de uso y eficiencia para realizar las pruebas, hacen que muchas organizaciones hayan adaptado JUnit a diferentes tipos de lenguajes.

## 1.10 Conclusiones

En este capítulo se han visto conceptos fundamentales relacionados con los temas que serán abordados a lo largo de toda la investigación. Con la ayuda de este capítulo se podrá avanzar en la investigación teniendo un conocimiento previo de los conceptos más fundamentales profundizando cada uno de ellos, conceptos de vital importancia para esta investigación como es el de calidad de software, calidad de software en Cuba, gestión de calidad y gestión de proyecto, estándares y normas, NC-ISO/IEC 9126, modelo de calidad, confiabilidad, bases de datos, Gestores de Bases de Datos, pruebas y modelos de prueba, que se utilizan para probar los productos; así como asegurar la calidad de los mismos.

A partir de esto se define que se van a utilizar los criterios de confiabilidad de los modelos de calidad mencionados anteriormente y como guía para la obtención de la propuesta, el Modelo V, para dar paso a la realización del próximo capítulo y de esta forma cumplir con el objetivo principal de la presente investigación.

## CAPÍTULO II: PROPUESTA DE UN MODELO DE CONFIABILIDAD PARA DATEC

---

### **2.1 Introducción**

En este capítulo se presenta la propuesta de modelo de prueba para medir la confiabilidad de los productos desarrollados en DATEC. La misma se conformó a partir de los resultados obtenidos de las entrevistas realizadas a varias personas que laboran en el centro, a través de las cuales se pudieron recoger las necesidades de prueba que presenta el mismo.

También, se caracteriza de forma breve al centro y se hace un análisis de las entrevistas realizadas. Se explica además porque se utiliza un modelo como vía de solución de la investigación y se proponen además herramientas que servirán de apoyo en la aplicación del modelo. La propuesta se realiza teniendo como guía el Modelo V; modelo de prueba utilizado en DATEC para comprobar la calidad de sus productos.

### **2.2 Análisis de las entrevistas realizadas en DATEC**

La entrevista *“es una conversación planificada para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado y puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación”* (Entorno Virtual de Aprendizaje, 2010).

Para obtener la información necesaria sobre las características del centro DATEC, se realizaron una serie de entrevistas (Ver anexo #1) al personal del centro, con el objetivo de conocer la situación existente y los principales problemas que atentan contra la calidad de sus productos. Mediante ella se desea profundizar en cómo y cuándo se hacen las pruebas en este centro; así como cuáles son las pruebas que son aplicadas, las tecnologías que se utilizan y otros elementos relacionados con la confiabilidad de los productos de software que ahí se desarrollan.

Para el mejor entendimiento de la entrevista se mostraron diferentes conceptos aclaratorios sobre algunos términos de la confiabilidad; así como su definición y la recopilación de datos teniendo en cuenta varios aspectos que tributan a dicho concepto. Esto facilitó que los trabajadores a los que se les realizaron la



entrevista comprendieran mejor las preguntas. Toda esta información que se pudo recopilar es muy importante para dar paso a la aplicación del modelo.

***Las personas entrevistadas en el centro son:***

- Roberto Sarmiento: Línea de Soluciones Integrales, Arquitecto de datos.
- Ernesto Herrera: Línea de soluciones Integrales y Herramientas, Arquitecto de Datos.
- Daymel Bonne Solis: Arquitecto de Datos.
- Anisley Delfino Rodríguez: Línea de Almacenes de Datos, jefa del grupo de Inteligencia de Negocio.
- Asnioby Hernández López: Jefe de la línea de Almacenes de Datos.
- Yudisney Vázquez Ortiz: Jefa de la línea de PostgreSQL.

A continuación se presentan los resultados obtenidos en la entrevista realizada a trabajadores del centro DATEC.

En el centro se realizaron preguntas concernientes con la confiabilidad de los productos de software dividiéndolo en diferentes criterios de confiabilidad, los cuales fueron explicados en el capítulo anterior:

En cuanto al criterio Precisión se realizaron 4 preguntas: El total de los entrevistados plantearon que no poseen pruebas que les indique si un producto es preciso o no y por ende no tienen cómo saber si los resultados obtenidos mediante las pruebas son parecidos o mejores que los de iteraciones anteriores. La mayoría de los entrevistados dijeron que no se registran las respuestas que va dando el software, mientras que la otra parte dijo que sólo se registran las respuestas que se obtienen del software cuando este va ser liberado.

En cuanto a la Exactitud se realizaron 6 preguntas: Todos los entrevistados plantearon que en el centro si se realizan pruebas, pero los resultados obtenidos no son analizados, para saber si son los esperados. Sólo se registran estos resultados si son negativos y son registrados en los Casos de Pruebas y en los Documentos de No Conformidades.

Para el criterio Tolerancia a fallas y Madurez se realizaron un total de 12 preguntas: Obteniendo que todos los entrevistados afirmaron que presentan varios errores en los proyectos y que los más comunes son: de ortografía, inconsistencia con los tipos de datos, errores de ubicación de requisitos y de formato. Estos errores son registrados en el Documento de no conformidad, pero no son analizados. Para saber si ocurrió algún error utilizan las pruebas, los casos de pruebas en la implementación y la comunicación con los clientes. Además de que en caso de que ocurra alguna falla se arregla y se analiza para encontrar donde ocurrió el defecto.

Para el criterio Recuperabilidad se realizaron 4 preguntas: Todos los entrevistados afirmaron que cuando ocurre una falla todos los datos están en una Base de Datos y se guardan en un Backup, para que en caso que ocurra una falla se puedan recuperar los mismos montando de nuevo el servidor y se comprueba que toda la información fue recuperada comparando los datos que se encuentran en la Base de Datos con los reportes y viendo si son los mismos.

El criterio Simplicidad fue analizado mediante la realización de 6 preguntas: Se obtuvo que la mayoría de los entrevistados plantean que en el centro utilizan estándares de codificación que permiten que el código sea fácil de entender, mientras que el resto dice que ellos mismos se organizan sin guiarse por ningún estándar para codificar. Todos los entrevistados afirman que no le ven nada malo a la reutilización de código, que de hecho las funciones las implementan para que puedan ser reutilizadas y lo más sencillas posible; que las mismas son registradas en los servidores centrales de Base de Datos, permitiendo que sólo tengan acceso a ellas los que tengan permiso y los que estén implicados con el desarrollo.

Para el criterio Consistencia y Completitud que son criterios muy parecidos porque se centran en la implementación se realizaron 5 preguntas: Mediante estas preguntas se observó que en el centro revisan que las funciones que son requeridas sean implementadas, pero lo hacen en la fase de liberación del producto y registran si cumplieron con la implementación o no en el Subversión. Además, de que las líneas de código de sus programas no están tan compactadas.

Para el criterio Conformidad con la Confiabilidad se realizaron 2 preguntas: Identificándose con ellas que si se aplican a sus proyectos regulaciones o leyes además de la utilización de los lineamientos de calidad planteados por la UCI. Una de ella es la utilización de software libre como Linux para la realización de sus

productos, adaptándolos después según las necesidades que tenga el cliente y tratando de usar estas leyes en todo momento.

Para el criterio de Modularidad se realizaron 8 preguntas: La mayoría de los entrevistados plantearon que en el centro se desarrollan los productos por módulos, mientras que el resto dijeron que se hacía mediante módulos y componentes. Todos los entrevistados plantearon que en el centro si se realizan pruebas para probar los módulos, como es el caso de las pruebas de unidad. Además, se pudo saber que en el centro la única forma que tienen para comprobar si un módulo es más complejo que otro es mediante la apreciación.

En las entrevistas también se realizaron algunas preguntas relacionadas con las pruebas de software que son realizadas en el centro siendo esta información muy necesaria, para la realización el presente trabajo investigativo.

Se realizaron 8 preguntas relacionadas con las pruebas realizadas en el centro y todos los entrevistados respondieron que si se realizan pruebas, la gran mayoría dijeron que estas son realizadas en la fase final del desarrollo del producto o en su fase de liberación mientras que una parte muy pequeña de los entrevistados dijo que se hacían a partir de la fase de implementación, sacando como conclusión que las pruebas se le realizan a los productos de software cuando van a ser liberados. Todos los entrevistados respondieron que sólo son registrados los resultados que son negativos y que lo hacen en el Documento de No Conformidades. Además de que todos los entrevistados confirmaron la utilización del Modelo V como guía para realizar el proceso de pruebas. Aunque este modelo es el que se utiliza en el centro para las pruebas la gran mayoría de los entrevistados dijeron que las pruebas que más se realizan en el centro con las Pruebas del Sistema, mediante las Pruebas Funcionales y las de Aceptación, mientras que otra parte más pequeña de los de los entrevistados tenían conocimiento de la realización de las pruebas, pero no del tipo de prueba que se realiza en el centro.

Además de las preguntas realizadas en las entrevistas, para tener una información relacionada con el uso de los criterios de confiabilidad y de las pruebas, se hace necesario también tener un cierto conocimiento de las tecnologías que se utilizan en el centro.

Se realizaron un total de 5 preguntas relacionadas con las tecnologías: Definiéndose con ellas que en el centro el único Gestor de Base de Datos que utilizan es PostgreSQL, aunque también se tienen algunos planes futuros en los que se utilizará Oracle como gestor para la realización de algunos de sus proyectos, pero hasta el momento en el centro se utiliza PostgreSQL, esto se debe a que es de código abierto y su licencia es gratis la cual es una de las principales razones por la que está definido su uso en la UCI y en el centro DATEC, además de otras ventajas que este brinda las cuales fueron explicadas en el capítulo anterior. También se obtuvo mediante las entrevistas realizadas información de que no se ha presentado en el centro ninguna falla relacionada con este gestor hasta el momento.

### **2.2.1 Resultados de las entrevistas**

Una vez realizado un análisis de las respuestas obtenidas mediante las entrevistas realizadas en el centro, relacionadas con los criterios de confiabilidad, pruebas y tecnologías, se consiguió determinar que los criterios de confiabilidad en los que se debe hacer más énfasis para la realización de la propuesta son: Precisión, Exactitud, Tolerancia a fallas, Madurez, Simplicidad, Completitud, Recuperabilidad, Consistencia y Modularidad, puesto que son los criterios que más se ve afectado su uso en el centro. Haciendo que se vea la necesidad de que se cree un método científico que sea capaz de probar el nivel de confiabilidad que van a tener los productos que se desarrollan en el centro. Además, se observa que las pruebas se realizan en la fase de liberación de los productos y que los resultados negativos de estas pruebas van a ser registrados en el Documento de No Conformidad. Que en DATEC se utiliza el Modelo V para hacerle pruebas a los productos de software que allí son desarrollados. A partir de esta información y conociendo que se tiene una cierta experiencia en la utilización del mismo es que se llega a conclusión de que la propuesta estará basada en dicho modelo. Sin dejar de tener en cuenta que la propuesta será realizada para productos que son desarrollados con PostgreSQL.

Con la realización de estas entrevistas se pudo obtener información de la situación existente en DATEC, observando mediante las mismas las necesidades y los problemas existentes en el centro. Utilizando esta información obtenida para dar paso a la realización de la propuesta de un modelo para probar la confiabilidad de los productos desarrollados con PostgreSQL en DATEC.

### **2.3 Resultados Científicos**

*"Un resultado científico es el producto de una actividad en la cual se han utilizado procedimientos científicos, que permiten ofrecer solución a algo, se plasma en recomendaciones, descripciones, publicaciones, que contienen conocimientos científicos o una producción concreta material, o su combinación y resuelven determinada necesidad económica y social" (López, 2007).*

Teniendo en cuenta la definición anterior un resultado científico es el producto de una investigación que permiten dar solución a un determinado problema ya sea teórico o práctico. Los resultados científicos se pueden clasificar de diferentes maneras, teniendo en cuenta el aspecto de la realidad que abordan: la teoría o la práctica.

Los resultados teóricos son aquellos que *"permiten enriquecer, modificar o perfeccionar la teoría científica, aportando conocimientos sobre el objeto"* (López, 2007). Los resultados prácticos son aquellos que *"tienen un carácter instrumental para transformar el funcionamiento del objeto en la realidad haciéndolo más eficiente, más productivo y más viable, entre ellos se encuentran: programas, estrategias, tecnologías, metodologías de trabajo, medios de enseñanza y modelos"* (López, 2007).

De esta forma, se decide utilizar un modelo para darle solución a la problemática planteada anteriormente, ya que con este se puede perfeccionar el trabajo que se realiza en el centro DATEC, para de esta forma poder cumplir con el objetivo previsto.

Con la realización de las entrevistas anteriormente analizadas se pudo obtener información que evidencian la utilización de un modelo y como para ello se necesita conocer información referente el objeto en cuestión se decide usar un modelo como resultado científico de la presente investigación. Además de que en el centro se utiliza el Modelo V para hacerle pruebas a los productos de software que allí son desarrollados, por lo que se tiene una cierta experiencia en la utilización del mismo. Modelo que tiene como objetivo principal probar la confiabilidad de los productos desarrollados con PostgreSQL en DATEC.

## **2.4 Propuesta del Modelo de Prueba para probar la Confiabilidad de los productos desarrollados con Gestores de Base de Datos en DATEC**

El modelo se conformó teniendo como guía el Modelo V. Permitiendo que cuando se vaya a utilizar este en el centro ya se tenga una base anterior. De modo que no sea un modelo completamente nuevo para el personal del proyecto evitando la pérdida de tiempo en aprendizaje para la utilización del mismo, además este modelo permite el uso de la verificación y la validación, ventaja que se utilizará también en la propuesta. La misma va a recibir el nombre de **“Modelo de Confiabilidad V”** cuyo objetivo y propósito fundamental va a ser probar la confiabilidad de los productos que son desarrollados con PostgreSQL en DATEC.

Para la realización de la propuesta se tuvo en cuenta las entrevistas realizadas en el centro mediante las cuales se obtiene un conocimiento previo de la situación existente en el mismo, permitiendo que con esta información obtenida se pudiera dar paso a la realización de la propuesta.

#### **2.4.1 Análisis de la propuesta del modelo de prueba para probar la confiabilidad de los productos desarrollados con tecnologías de Base de Datos en DATEC**

Para la realización de la propuesta se tuvo presente cada una de las ventajas que presenta el Modelo V. Haciendo énfasis en estas para aplicarlas y que de esta forma no se dejaran pasar por alto con la adaptación hecha al mismo. Se profundizó además en las desventajas que presenta dicho modelo, con el objetivo de poder cambiar o mejorar las insuficiencias que este posee.

Una vez analizadas todas las imperfecciones descubiertas en el centro con las entrevistas realizadas, se da a conocer la propuesta del **“Modelo de Confiabilidad V”** (Ver figura 6). Este modelo va a ser utilizado para probar el nivel de confiabilidad que van a tener los productos de software que se desarrollan con PostgreSQL en DATEC.

Este centro ha hecho un gran esfuerzo para obtener productos de software que sean lo más confiables como sea posible, para ayudarlos a lograr su objetivo es necesario que se cuente en el centro con un modelo de confiabilidad que pruebe y diga el nivel de confiabilidad que presentan los productos que en él se desarrollan y para ayudarlos con su propósito es que se da paso a la realización de la propuesta.

La propuesta consta de varias fases las cuales se van a retroalimentar unas con otras, permitiendo que cada vez que ocurra algún error o cambio y exista la necesidad de retornar a la fase anterior a hacer algún cambio, se le realicen nuevamente las pruebas al producto software.

Las pruebas se escogieron para que fueran realizadas en cada fase según las necesidades de cada una que se obtuvieron en las entrevistas realizadas al centro. Permitiendo que se vayan obteniendo los errores que se presenten en el producto de software durante su paso de una fase a otra y que el mismo se realice pensando siempre en el cliente. Al hacer estas pruebas en cada una de las fases, se asegura que desde el comienzo del desarrollo del producto de software se tenga conocimiento de algunos de los errores que se van presentando y que los productos sean moldeados según las necesidades que tenga el cliente.

***Estas pruebas por fases van a ser:***

**Prueba de Requerimientos** (Fase de Requisitos) esta prueba se realiza con el principal objetivo de que los requerimientos sean los que el cliente realmente quiere. Es de vital importancia para el desarrollo del producto porque una vez que sean los correctos, se puede avanzar a la siguiente fase con la convicción de que se desarrollará un producto de software que cumple con lo que verdaderamente el cliente desea y necesita. Esta prueba se va a realizar con la ayuda de una lista de chequeo, “se entiende por lista de chequeo (cheks list) a un listado de preguntas, en forma de cuestionario que sirve para verificar el grado de cumplimiento de determinadas reglas establecidas a priori con un fin determinado” (Bichachi, 2007). La lista de chequeo utilizada para la realización de estas pruebas son las que utiliza Calisoft escogiendo de ellas las preguntas que más se identificaban con el presente trabajo investigativo (Ver Anexo 4).

**La Prueba de Usabilidad** (Fase de Diseño y Fase de Arquitectura) con ella se va a lograr que el diseño de la interfaz del usuario sea fácil de entender y de usar. Permitiendo que el producto sea adecuado para que lo puedan utilizar los clientes que no tengan experiencia o sea que nunca hayan trabajado con el producto de software.

**La Prueba de Unidad** (Fases de Diseño de Módulos y Fase de Implementación) es una prueba muy importante para estas fases, porque cuando se empieza a implementar es necesario ir revisando la lógica y el código interno del producto de software a medida que se vaya implementando y de esta forma ir obteniendo los errores parte por parte. Para la realización de las Pruebas de Unidad se utilizan varias

técnicas o pruebas como son las **Pruebas de Cajas Blanca** que van a ser las que van a revisar el código interno del producto de software, dentro de esta prueba se escogió aplicar el método o las **Pruebas de Camino Básico**. Otra de las técnicas o pruebas que se utilizan para la realización de las Pruebas de Unidad son las **Pruebas de Caja Negra** las cuales van a probar los casos de uso, dentro de esta prueba se va a aplicar las **Pruebas de Partición Equivalente**.

Para poder probar si un producto de software es confiable además de las pruebas por fases, se realizan otras que van a obtener algunos de los errores que se pudieron haber quedado o no fueron encontrados en las que se realizaron en cada fase. Estas pruebas son de vital importancia para que los productos de software que se desarrollan en DATEC presenten un alto nivel de confiabilidad. Estas pruebas van a complementar a las pruebas mencionadas anteriormente y son:

**Las Pruebas de Integración** se realizan después de las Pruebas de Unidad, tienen como principal objetivo probar de una forma integrada el producto de software. Están centradas en la construcción del diseño y la arquitectura del software.

Después de haber realizado las Pruebas de Integración se da paso a las **Pruebas del Sistema**, donde se probará el funcionamiento del mismo, mediante una serie de pruebas que van a comprobar el correcto rendimiento del sistema antes de pasar a las Pruebas de Aceptación. Las pruebas que complementan las del sistema son: **la Prueba de Integridad de los datos y la Base de Datos, Prueba de Performance, Prueba de Carga, Pruebas de Estrés, Pruebas de Volumen, Prueba de Fallas y Recuperación y la Prueba de Seguridad**.

Estas pruebas se le realizan fundamentalmente a los productos de software que tengan que ver con el almacenamiento de datos y el uso de Base de Datos, como es el caso de los productos para los cuales fue diseñado este modelo de confiabilidad que son desarrollados con PostgreSQL. Mediante ellas se puede observar cómo se desenvuelve la aplicación y cómo se comporta su funcionamiento. También se puede saber la cantidad de información que logra soportar el producto de software. Además, de ver cómo responde la aplicación ante fallas externas y su capacidad de recuperación, de esta manera, comprobar cuan ágil y rápido puede ser este para recuperarse.



Todas estas pruebas están enfocadas al Gestor de Base de Datos que se utiliza en el centro y para su utilización es necesario tener en cuenta algunos requisitos.

### **1. Prueba de integridad de los datos y la base de datos.**

- La prueba requiere un entorno de administración de DBMS (Data Base Management System) en español Sistemas de Gestión de Bases de Datos o controladores para ingresar o modificar información directamente en la base de datos.
- Los procesos deben ser invocados manualmente.
- Se deben usar bases de datos pequeñas para aumentar la facilidad de inspección de los datos para verificar que no sucedan eventos no aceptables.

### **2. Prueba de Performance.**

Las pruebas de performance deben incluir un trabajo de fondo en el servidor. Esto se puede realizar de distintas formas:

- Enviar transacciones directamente al servidor, generalmente en la forma de consultas (SQL).
- Crear usuarios virtuales para simular muchos clientes, generalmente varios cientos. Se pueden usar herramientas de Emulación de Terminar Remota para lograr este objetivo. Esta técnica también se usa para cargar la red con “tráfico”.
- Usar muchos clientes físicos, cada uno corriendo procedimientos de prueba.
- La prueba de performance se debe realizar en una máquina dedicada para permitir control total y medición exacta.
- Las bases de datos usadas para las pruebas de performance deben tener un tamaño similar a las reales.

### **3. Prueba de Carga.**

- La prueba de carga debe realizarse en una máquina dedicada para tener control total y exactitud de mediciones.
- Las bases de datos usadas para la prueba deben tener un tamaño similar a las reales.

#### **4. Prueba de Estrés.**

- Las pruebas de esfuerzo de red pueden requerir herramientas de red para cargar la red con mensajes o paquetes.
- La cantidad de disco del servidor usada por el sistema debe ser reducida temporalmente para restringir el espacio disponible para crecimiento de la base de datos.
- Sincronizar el acceso simultáneo de varios clientes accediendo a los mismos datos.

#### **5. Prueba de Volumen.**

- Se deben usar múltiples clientes, ejecutando las mismas pruebas o pruebas complementarias para producir el peor caso de volumen de operaciones o mezcla en un período de tiempo extenso.
- Se debe crear el tamaño máximo de base de datos (real, escalado o con datos representativos) y múltiples clientes ejecutando consultas simultáneamente por un período de tiempo extenso.

#### **6. Prueba de Fallas y Recuperación.**

- Los procedimientos para desconectar cables (simulando falta de energía o pérdida de comunicación) no son deseables o factibles. Se pueden requerir métodos alternativos, como software de diagnóstico. Se requieren los grupos de recursos de Sistemas, Bases de datos y Red.
- Estas pruebas deben ejecutarse fuera del horario de trabajo normal o en una máquina aislada.

#### **7. Prueba de Seguridad.**

- Seguridad en el ámbito de aplicación: Identificar y hacer una lista de cada tipo de usuario y las funciones y datos sobre las que cada tipo tiene permiso.

- Crear pruebas para cada tipo de usuario y verificar cada permiso creando operaciones específicas para cada tipo de usuario.
- Modificar el tipo de usuario y volver a ejecutar las pruebas para los mismos usuarios. En cada caso, verificar que las funciones o datos adicionales están correctamente disponibles o son denegados.
- Acceso en el ámbito de sistema: El acceso al sistema debe ser discutido con el administrador del sistema o la red. Esta prueba no puede requerirse como tal, es una función del administrador del sistema o de la red.

En el modelo se aplica otra prueba que en conjunto con las explicadas anteriormente definen la confiabilidad de los productos de software desarrollados en el centro. Esta no está tan relacionada con las tecnologías, pero de igual forma cumple una labor esencial probando las funcionalidades del sistema para ver de qué forma responde este ante algún cambio o entrada de datos no esperados esta prueba es: la **Prueba de Funcionalidad**.

Las **Pruebas de Aceptación** se realizan después de haberle hecho al software todas las pruebas que hayan sido necesarias, para la obtención de un producto más confiable y por ende con menos probabilidad de que el cliente presente problemas con el mismo. Estas pruebas se hacen con el cliente ya que él es el que tiene que aceptar el producto de software terminado.

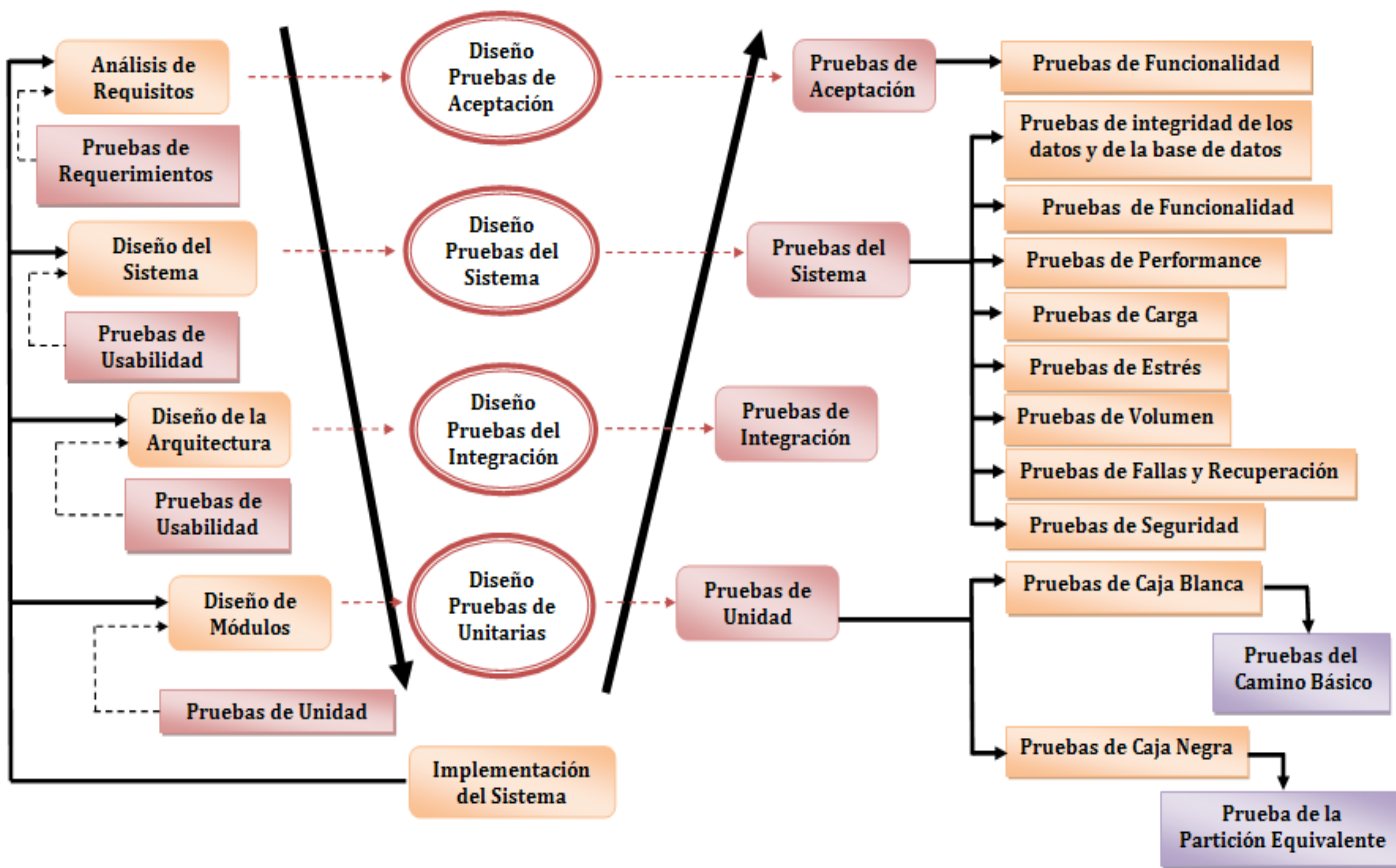


Fig. # 6: Modelo de Confiabilidad V

## 2.4.2 Cálculo de la Confiabilidad

Para definir mediante el Modelo de Confiabilidad V si un producto de software es confiable se le atribuyen a los criterios de confiabilidad planteados en el capítulo anterior, todas las pruebas que son utilizadas para probar en el modelo. Estas pruebas por criterio son:

### Pruebas por Criterios

- **Completitud**

Mediante este criterio se puede saber si los requisitos cumplen con las necesidades del cliente. Para poder comprobar que en el producto de software se tenga en cuenta la confiabilidad se hace

necesario el cumplimiento del mismo. Para ello se propone la Prueba de Requerimientos que con la realización de esta se podrá saber si los requisitos cumplen con las expectativas del cliente.

- **Consistencia**

Para poder comprobar la existencia de este criterio en el software se debe realizar la **Prueba de Requerimientos** ya que con esta se podrá saber si los requisitos están bien elaborados, son entendibles y no sean absurdos.

- **Madurez**

Con este criterio se puede medir la capacidad del software de evitar una imperfección provocada por una falla. Este se podrá comprobar mediante la utilización de la **Prueba de Volumen**. Con ella se podrá probar la resistencia del producto de software ante los fallos que se puedan presentar. La otra es la **Prueba de integridad de los datos y la base de datos** que con la realización de esta prueba se puede verificar que aunque el software posea una gran resistencia a fallos, si llegara a ocurrir alguno inesperado, este tenga la capacidad de evitar la pérdida de los datos.

- **Tolerancia a fallas**

Este criterio se podrá comprobar con la realización de las siguientes pruebas, **Pruebas de Volumen**, que con la realización de esta se puede verificar de que el error no sobrepase el nivel en el cual ocurrió el mismo después de introducir una gran cantidad de volumen de datos no soportados por el software. Otra es la **Prueba de Integridad de los datos y la Base de Datos** con la ejecución de esta prueba se puede comprobar que aunque el software posea una gran resistencia a fallos, si llegara a ocurrir alguno inesperado, este sea capaz de mantenerse entrando datos a la Base de Datos sin que ocurran pérdidas de los mismos. La última prueba es **Prueba de Fallas y Recuperación** que a través de esta se puede verificar de que cuando el sistema sea sometido a condiciones extremas para causar una falla o que cuando ocurra una falla ya sea de software, hardware o de mal funcionamiento de la red, este sea capaz de mantener su nivel de ejecución sin problemas y sin pérdidas de datos ni de la integridad de los mismos.

- **Recuperabilidad**

Para la comprobación de este criterio se hace necesario el uso de la **Prueba de Estrés**. Con la realización de esta prueba se verifica que cuando el sistema se encuentre bajo una extrema sobrecarga, este sea capaz de recuperar todos los datos que sean dañados a causa de esto y que el tiempo utilizado para hacerlo sea el menor posible. Otra forma de comprobar este criterio es mediante la **Prueba de Fallas y Recuperación** que con la realización de esta prueba se puede verificar que cuando el sistema sea sometido a condiciones extremas para causar una falla, este sea capaz de recuperar todos los datos que sean dañados a causa de esto en el menor tiempo posible. También se pueden utilizar la **Prueba de Integridad de los datos y la Base de Datos**. Con la realización de esta prueba se verifica que cuando el sistema recupere todos los datos los mismos no sean corrompidos. Otra de las pruebas que se utilizan para la comprobación de este criterio es la **Pruebas de Performance** que con la realización de esta prueba se puede observar la rapidez con que responde el sistema cuando se realiza una transacción o llamadas y ver como se recupera este ante algún fallo inesperado ya sea externo o interno. Además, se puede utilizar la **Pruebas de Carga** que con la realización de esta prueba se puede verificar que cuando el sistema sea sometido a diferentes cargas de trabajo para comprobar su rendimiento y ocurra algún evento inesperado, ver cómo se comporta el mismo a la hora de recuperar los datos que se dañaron a causa del error ocurrido y tiempo que demora en hacerlo.

- **Precisión y Exactitud**

Para la comprobación de este criterio se necesita realizar la **Prueba de Funcionalidad**, Con la realización de esta prueba se garantiza que el software sea capaz de devolver los resultados esperados, después de que se le introduzcan una serie de datos incorrectos e inválidos o cuando estos sean correctos.

- **Simplicidad**

Este criterio se puede comprobar mediante la utilización de la **Prueba de Usabilidad** ya que con la realización de esta prueba se puede verificar que el sistema sea lo más amigable y entendible para el usuario con o sin experiencia. Además, se utiliza la **Prueba de Seguridad** para comprobar que este criterio sea fácil de usar, pero a la vez seguro.

- **Modularidad**

Este criterio se puede comprobar mediante la realización de la **Prueba de integración** con la realización de esta prueba se puede garantizar que todos los módulos que componen el sistema sean probados e integrados como un todo. Es decir, que con esta se puede saber cómo funciona el sistema una vez que se hayan completado la realización del sistema en general. Otra de las pruebas que se utiliza para comprobar este criterio es la **Prueba de Unidad** que con la realización de esta prueba se puede verificar que todos los módulos que componen el sistema sean probados de forma individual. Para de esta forma poder saber cómo se comporta cada uno de ellos.

Con la aplicación de estas pruebas se podrá ver cómo influyen o afectan las mismas sobre los criterios de confiabilidad de los cuales forman parte. Esto se podrá saber mediante las no conformidades, que son los errores que se obtienen con la realización de las pruebas, los cuales van a ser registrados en el Documento de No Conformidad creado y utilizado por el centro Calisoft para registrar los resultados obtenidos en las pruebas (Ver Anexo 5) y mediante un peso que se le asigna a las pruebas según la importancia que tiene esta dentro de los criterios de confiabilidad. Este peso será asignado a cada una de forma variable o sea que va a ser dado según la importancia que tenga la prueba para el que este aplicando el modelo.

Con la asignación de este peso y con la obtención de las no conformidades que se van a obtener al aplicar las pruebas, se podrá obtener el nivel de confiabilidad que va a tener cada criterio, este nivel de confiabilidad se calcula con el peso asignado a cada prueba sacando un promedio por criterio, el resultado que se obtiene será asignando en una escala del 1 al 3 divididas de forma equitativa la cual determinará el nivel de confiabilidad que tendrá cada criterio que se calcula utilizando la siguiente fórmula:  $NCC = \frac{\sum(PC/P)}{P}$ .

Una vez que se tiene el nivel de confiabilidad de cada criterio se le asignaran a los mismos un peso del 1 al 3 el cual va a medir la importancia que tienen estos, para lograr la confiabilidad de los productos software desarrollados en el centro, este peso se va a multiplicar por el valor obtenido anteriormente de cada criterio de confiabilidad y se saca el promedio. Este valor resultante se va insertar en una escala del 1 al 3 también divididas equitativamente. Obteniéndose finalmente el nivel de confiabilidad que va a tener

el producto de software el cual podría tener un nivel de confiabilidad alto, medio o bajo (Ver Tabla # 13). Este se va a calcular a través de la siguiente fórmula:  $NCT = \sum (PromC * PC)/C$

P: Pruebas

C: Criterios

PromC: Promedio de los criterios

PC: Peso del criterio

NCC: Nivel de confiabilidad de cada criterio

NCT: Nivel de Confiabilidad Total

### 2.4.3 Herramientas de prueba propuestas

Cuando se va a construir un producto este tiene todo un ciclo para su desarrollo, pasa por varias etapas donde se construyen diferentes partes del producto. Una de estas etapas por las que atraviesa el software es la etapa de validación, que en esta el producto se somete a una serie de pruebas que evidencian la calidad del mismo.

Estas se pueden hacer de forma manual o automática, siendo esta última una de las mejores formas para poder comprobar la efectividad de cualquier producto de software. Para poder aplicar el **Modelo de Confiabilidad V** en DATEC y realizar las pruebas que lo componen se proponen varias herramientas de apoyo mediante las cuales se puede llevar a cabo esta etapa. Estas herramientas no son de uso obligatorio en el centro, simplemente las planteamos en caso de que no se utilice ninguna de ellas, en el caso de que utilicen otras las propuestas no se imponen, sólo se recomiendan. Las mismas ya fueron estudiadas a profundidad en otra investigación a la cual se hace referencia.

Estas herramientas son: **JMeter** para las pruebas Carga, Estrés, además se puede usar la herramienta **DTM Data Generator** que se utiliza también para esas pruebas y además para las de Performance y Usabilidad; y por último está la herramienta **JUnit** para pruebas de Unidad e Integridad. Estas herramientas se proponen para que sean utilizadas al aplicar el modelo en el centro para facilitar uso de



las pruebas. Para que las pruebas sean satisfactorias se recomienda la utilización de estas herramientas, pero se pueden utilizar otras que sean del conocimiento del centro.

<b>Pruebas</b>	<b>Criterios</b>	<b>Herramientas</b>
Pruebas de Requerimientos	Consistencia Compleitud	Manual
Pruebas de Usabilidad	Simplicidad	DTM Data Generator
Pruebas de Unidad	Modularidad	JUnit
Pruebas de Integración	Modularidad Exactitud	JUnit
Prueba de integridad de los datos y la base de datos	Madurez Recuperabilidad Conformidad Tolerancia ante fallos	JMeter
Prueba de Seguridad	Simplicidad	Manual
Pruebas de Funcionalidad	Precisión	JMeter
Prueba de Performance	Recuperabilidad	DTM Data Generator
Prueba de Carga	Recuperabilidad	JMeter
Prueba de Estrés	Recuperabilidad	JMeter
Prueba de Volumen	Madurez Tolerancia ante fallos	DTM Data Generator
Prueba de Fallas y Recuperación	Tolerancia ante fallos Recuperabilidad	Manual

**Tabla # 1: Criterios y herramientas por pruebas.**

## **2.5 Conclusiones**

En el presente capítulo se da a conocer la propuesta del modelo de prueba para probar la confiabilidad de los productos desarrollados en DATEC con PostgreSQL. Explicándose detalladamente cada uno de los pasos del modelo propuesto, como son las pruebas y los criterios de confiabilidad a los que tributan las

mismas, para con ellos conocer cómo se calcula el nivel de confiabilidad que van a tener los productos de software que se desarrollan en el centro.

Además, para la realización de la propuesta se llevaron a cabo una serie de entrevistas que permitieron conocer la situación existente en el centro y se hizo también una breve reseña de la estructura del centro DATEC, esencialmente como está organizado y sus principales líneas de trabajo, para que de esta forma se deduzca mejor donde se dirigió principalmente la investigación.

También en este capítulo se proponen herramientas para que sean usadas en la aplicación del modelo y faciliten la realización de las pruebas, mediante ellas se le da un mayor grado de validez científico al Modelo de Confiabilidad V. De este modo la investigación da cumplimiento a su principal objetivo, obtener la propuesta de un modelo de confiabilidad con el cual se pruebe el nivel de confiabilidad que presentan los productos desarrollados con PostgreSQL en DATEC.

## CAPÍTULO III: VALIDACIÓN DE LA PROPUESTA

---

### 3.1 Introducción

En el presente capítulo se valida la propuesta de este trabajo investigativo, por un grupo de expertos escogidos en varios centros de la UCI. Obteniendo los criterios de cada uno de ellos acerca de varios parámetros relacionados con el Modelo de Confiabilidad V. Con estas valoraciones de los expertos se pasó a aplicar la propuesta a un producto desarrollado en DATEC, siendo este el centro sede donde se va a aplicar la investigación. Este modelo también se puede aplicar en cualquier otro centro o institución que desarrollen productos de software con PostgreSQL. Además, se documenta la validación y aplicación de la propuesta presentada en el capítulo anterior, comprobando la utilidad y la eficiencia del modelo elaborado.

### 3.2 Evaluación

El Modelo de Prueba se validará utilizando como guía el libro de Rolando Alfredo Hernández León. Este método combina los métodos multicriterios con los cuantitativos, mediante el criterio de varios expertos. Utilizando procedimientos estadísticos este método determina de manera muy rápida un índice de prioridad que facilita la toma de decisiones sobre la posibilidad de éxito, determinando así la aceptación de la propuesta.

#### 3.2.1 Desarrollo de la validación de la propuesta

Para realizar la validación de la propuesta se llevaron a cabo un conjunto de pasos los cuales se muestran a continuación:

1. Se definen los criterios de evaluación que fueron utilizados, agrupados en los grupos o categorías siguientes:

##### **Grupo No 1: Criterios de mérito científico.**

1. Valor científico del modelo.
2. Calidad de la investigación.

3. Contribución científica.
4. Responsabilidad científica del investigador.

**Grupo No 2: Criterios de implantación.**

5. Necesidad de empleo del modelo.
6. Obtención de productos finales con calidad.
7. Posibilidades de aplicación.
8. Satisfacción de las necesidades de la producción.
9. Facilidades de comprensión.

**Grupo No 3: Criterios de generalización.**

10. Facilidad de uso.
11. Capacidad del modelo para la realización de cambios que impliquen mejoras.

**Grupo No 4. Criterios de impacto.**

12. Repercusión del modelo.
13. Contribución al proceso de desarrollo de software.
14. Contribución a la confiabilidad del producto de software.
15. Organización en el proceso de desarrollo de software.

2. Se establece el peso referente de cada grupo, asignándole el porcentaje que representa cada grupo de criterios del total teniendo en cuenta los intereses necesarios para la evaluación.

Grupo No.1 \_\_\_\_\_ 20

Grupo No.2 \_\_\_\_\_ 35

Grupo No.3 \_\_\_\_\_ 15

Grupo No.4 \_\_\_\_\_ 30

3. Se hace una selección de los expertos según su experiencia en la materia y de ser posible sus años realizando el papel de experto.

Los expertos proporcionan un peso para cada criterio en una escala del 1 al 10, sin olvidar que la suma del valor dado a cada criterio de un grupo no exceda del peso relativo asignado a este. Siendo C el número de criterios, E el número de expertos y  $E_p$  es el promedio de cada criterio. Ver **Tabla 2**.

Grupos	C / E	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>	E <sub>p</sub>
20	C <sub>1</sub>	8	4	8	7	5	6	5	6,1
	C <sub>2</sub>	5	6	3	5	5	5	6	5,000
	C <sub>3</sub>	4	3	5	3	4	4	5	4,000
	C <sub>4</sub>	3	7	4	5	6	5	4	4,857
35	C <sub>5</sub>	4	13	15	13	6	5	10	9,429
	C <sub>6</sub>	13	8	7	5	9	10	4	8
	C <sub>7</sub>	5	4	3	7	6	5	7	5,286
	C <sub>8</sub>	5	5	4	5	6	5	4	4,857
	C <sub>9</sub>	10	5	6	5	8	10	10	7,714
15	C <sub>10</sub>	5	8	5	5	7	8	10	6,857
	C <sub>11</sub>	8	7	10	10	8	7	5	7,857
30	C <sub>12</sub>	5	5	8	10	6	4	4	6
	C <sub>13</sub>	5	10	7	10	7	6	7	7,429
	C <sub>14</sub>	15	10	8	5	7	10	14	9,857
	C <sub>15</sub>	5	5	7	5	10	10	5	6,714
<b>Total</b>		<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100,0</b>

**Tabla # 2: Resultado del trabajo de los expertos.**

- Se verifica la consistencia del trabajo realizado por los expertos, utilizándose el coeficiente de concordancia Kendall y el estadígrafo Chi cuadrado ( $\chi^2$ ).

Para la realización de esto se sigue el siguiente procedimiento:

- Para cada criterio se determina  $\sum E$  que representa la sumatoria del peso dado por cada experto

- Se calcula el peso medio de cada criterio ( $M\Sigma E$ ) y se determina la desviación de la media ( $\Delta C$ ), que posteriormente se eleva al cuadrado para obtener la dispersión (S) por la expresión.

$$S = \sum (\Sigma E - M \Sigma E / C)^2$$

- Conociendo la dispersión se puede calcular el coeficiente de concordancia de Kendall (W)

$$W = S / E^2 (C^3 - C) / 12$$

- El coeficiente de concordancia de Kendall permite calcular el Chi cuadrado real

$$X^2_{(real)} = E (C-1) W$$

- El Chi cuadrado calculado se compara con el obtenido de las tablas estadísticas.

Si se cumple que:  $X^2_{real} < X^2_{(\alpha, c-1)}$ . Se puede decir que existe concordancia en el trabajo de los expertos. Ver Tabla # 3.

E/C	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>	Ep	ΣE	ΣE/C	ΔC	ΔC <sup>2</sup>
C <sub>1</sub>	8	4	8	7	5	6	5	6,1	43	2,866	-3,67	13,444
C <sub>2</sub>	5	6	3	5	5	5	6	5,000	35	2,333	-11,667	136,111
C <sub>3</sub>	4	3	5	3	4	4	5	4,000	28	1,866	-18,667	348,444
C <sub>4</sub>	3	7	4	5	6	5	4	4,857	34	2,266	-12,667	160,444
C <sub>5</sub>	4	13	15	13	6	5	10	9,429	66	4,4	19,333	373,778
C <sub>6</sub>	13	8	7	5	9	10	4	8	56	3,733	9,333	87,111
C <sub>7</sub>	5	4	3	7	6	5	7	5,286	37	2,466	-9,667	93,444
C <sub>8</sub>	5	5	4	5	6	5	4	4,857	34	2,266	-12,667	160,444
C <sub>9</sub>	10	5	6	5	8	10	10	7,714	54	3,6	7,333	53,778
C <sub>10</sub>	5	8	5	5	7	8	10	6,857	48	3,2	1,333	1,778
C <sub>11</sub>	8	7	10	10	8	7	5	7,857	55	3,666	8,333	69,444

<b>C<sub>12</sub></b>	5	5	8	10	6	4	4	6	42	2,8	-4,667	21,778
<b>C<sub>13</sub></b>	5	10	7	10	7	6	7	7,429	52	3,466	5,333	28,444
<b>C<sub>14</sub></b>	15	10	8	5	7	10	14	9,857	69	4,6	22,333	498,778
<b>C<sub>15</sub></b>	5	5	7	5	10	10	5	6,714	47	3,133	0,333	0,111
<b>DC</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100,0</b>	<b>70</b>	<b>46,67</b>		<b>2047,3</b>
<b>M ΣE</b>	46,67											
<b>S</b>	2047,3											
<b>W</b>	0,149											
<b>X<sup>2</sup>real</b>	14,624											
<b>X<sup>2</sup> (α, c-1)</b>	36,12											

**Tabla # 3: Resultado del cálculo de la concordancia y la dispersión.**

Según los resultados de los cálculos que se muestran en la Tabla # 3, se aprecia que  $X^2$  real <  $X^2$  tabla por lo que se puede decir que existe concordancia en el criterio de los expertos. Para alfa con un valor de  $\alpha = 0,001$ .

5. Se identifica el peso relativo de cada criterio (P) y se calcula el Índice de Aceptación (IA) de la propuesta.

Para ello se sigue el procedimiento siguiente:

- Conociendo el número de experto que realizan la evaluación (E) y la sumatoria de las puntuaciones de cada criterio  $\Sigma C$  se puede calcular el peso de cada criterio (P).
- Conociendo el peso de cada criterio (P) y la calificación dada por los evaluadores (c) en una escala de 1-5 se puede calcular el valor de  $P \times c$ .

Para recoger la calificación dada por los expertos a cada uno de los criterios se realizó una encuesta la cual se muestra en el Anexo # 2 y Anexo # 3 del trabajo. Ver Tabla # 4.

Criterios	Calificación (c)					P	P x c
	1	2	3	4	5		
C1			x			0,061	0.184
C2				x		0,050	0.200
C3			x			0,040	0.120
C4				x		0,049	0.1943
C5					x	0,094	0.471
C6					x	0,080	0.400
C7			x			0,053	0.159
C8				x		0,049	0.194
C9					x	0,077	0.386
C10				x		0,069	0.274
C11				x		0,079	0.314
C12			x			0,060	0.180
C13					x	0,074	0.371
C14				x		0,099	0.394
C15					x	0,067	0.336
<b>Total</b>							<b>4,1786</b>
IA							

**Tabla # 4: Calificación de los criterios.**

6. Con el valor anterior se calcula el Índice de Aceptación del proyecto.

$$IA = P \times C / 5 = 4,1786/5$$

$$IA = 0,836$$

7. Se ubica el IA calculado anteriormente en rangos ya predefinidos, en dependencia de donde se ubique, será la probabilidad de éxito que tenga la propuesta.

El Índice de Aceptación calculado es **0,836**.



### Rangos predefinidos de Índice de Aceptación.

IA > 0,7 Existe alta probabilidad de éxito.

0,7 > IA > 0,5 Existe probabilidad media de éxito.

0,5 > IA > 0,3 Probabilidad de éxito baja.

0,3 > IA Fracaso seguro.

### 3.2.2 Resultado de las encuestas realizadas a los expertos.

Para la validación de la propuesta se seleccionaron 7 expertos teniendo en cuenta sus años vinculados a la UCI, además de la experiencia que posean debe ser en la Gestión de Proyectos y Calidad de Software. Al elegir a los expertos con estas características se asegura que los resultados que se obtengan tengan una buena calidad, permitiendo así, que las opiniones dadas sean confiables y válidas para el objetivo propuesto.

La información de los expertos seleccionados se observan a continuación:

No Expertos	Nombre y Apellidos	Puesto actual	Proyecto
1	Heydi Menéndez Ávalos	Profesor	Departamento de Pruebas de Calisoft
2	Liane Figueroa Hernández	Especialista del Departamento Central de Pruebas de Software.	Laboratorio Industrial de Pruebas de Software (LIPS)
3	Daimi Bretones Lorenzo	Asesor de Calidad	Centro de Tecnología y Gestión de Datos (DATEC)
4	Dorisbel Muro Fumero	Profesor	Centro de Desarrollo de Arquitectura Empresariales (CDAE)

5	Yisel Niño Benítez	Administrador de calidad	Centro para la Informatización de Gestión de Entidades (CIGE)
6	Yania Cedeño Oliva	Jefe de Laboratorio de Pruebas UCID	Unidad de Compatibilización Integración y Desarrollo de Software para la Defensa (UCID)
7	Heney Díaz Pérez	Profesor del Departamento de Pruebas de Software	Cuerpo de Investigación Científica Penales y de Criminalística (CICPC)

**Tabla # 5: Relación de los expertos**

Según las encuestas realizadas a personas con conocimientos de calidad y pruebas (expertos), se puede resumir que el Modelo de Confiabilidad V puede tener una influencia positiva en la calidad de los productos de software desarrollados en DATEC. Siendo fácil de entender y de usar. La mayoría de los expertos proponen la aplicación del modelo a diferentes productos de software del centro y de la UCI. Plantean que con la aplicación del modelo se van a obtener muchas mejoras de los productos de software en cuanto a la calidad y la confiabilidad final del producto.

### **3.3 Aplicación de la propuesta del Modelo de Confiabilidad V en DATEC.**

En esta etapa del capítulo la propuesta ya fue validada por el método de Validación por Experto y de ellos se obtuvo un alto índice de aceptación. Al tener una opinión de estos sobre la aplicación del Modelo de Confiabilidad V, se pasa a la aplicación de este en DATEC.

El modelo fue aplicado a un producto terminado en diciembre del año 2009, llamado Generador de Reportes que fue el producto que pudo gestionar el centro para probar el modelo. Dicho producto aunque fue totalmente liberado no se tiene la documentación completa del mismo por problemas internos del centro, lo que dificultó un poco la aplicación del modelo, ocasionando que sólo se pudiera aplicar una parte de este. Es por ello que la propuesta únicamente se aplicó en la etapa de Implementación y de Pruebas del Sistema, siendo esta la única documentación que se tenía del producto en el centro.

Para la aplicación de este modelo se trabajó con los documentos de No Conformidades, que son los errores obtenidos de las pruebas que se le realizan al producto de software, registrados como no conformidades en este documento y con los Casos de Pruebas.

Como la aplicación del modelo se realizó en un producto ya terminado y probado, para poder aplicar el mismo se revisaron y estudiaron a fondo los Documentos de No Conformidades. Teniendo en cuenta dichos documentos y con los Casos de Pruebas se observaron cuales de ellas se aplicaron al producto por el centro como es el caso de las pruebas de Unidad, Integración y las del Sistema, las cuales también son aplicadas en el modelo propuesto.

Teniendo en cuenta los estudios realizados a los casos de pruebas y a las no conformidades obtenidas en la última iteración de pruebas realizada, se observa que muchos de los errores encontrados se pueden obtener mediante algunas de las pruebas que se proponen en el Modelo de Confiabilidad V, estas son: Las Pruebas de Funcionalidad, Pruebas de Integridad de los datos y de la base de datos, Pruebas de Performance, Cargas, Estrés y por último la Prueba de Fallas y Recuperación (Ver Tabla # 7).

Este modelo está compuesto también por otras pruebas que aunque no se pudieron aplicar para probar en el producto Generador de Reportes; son también importantes y al realizarlas todas en conjunto se puede obtener un cálculo más preciso del nivel de confiabilidad que van a tener los productos de software, a los que se les aplique el Modelo de Confiabilidad V, ya que estas pruebas tributan a criterios de confiabilidad.

Aunque no se tengan todas las pruebas que se propone en el modelo se puede calcular la confiabilidad de los productos desarrollados en el centro con PostgreSQL de igual forma, aunque los resultados no van a ser tan precisos, como lo serían con la aplicación de todas las pruebas que propone el modelo, pero si se puede obtener un cierto nivel de confiabilidad.

Para indicar cómo se comporta la confiabilidad del producto Generador de Reportes se le asignarán pesos a las pruebas según la importancia que tengan las mismas, este peso es variable, ya que va a ser asignado por el que aplique el modelo según la importancia que tengan las pruebas, para esa persona en el momento en que las esté realizando. Los pesos serán asignados del 1-3.

Estos pesos fueron asignados, para el producto Generador de Reportes, según la importancia que le dieron a las pruebas algunas de las personas que nos ayudaron a la aplicación del modelo en el centro (Ver Tabla # 6).

<b>Pruebas</b>	<b>Pesos</b>
Pruebas de Requerimientos	3
Pruebas de Usabilidad	1
Pruebas de Unidad	3
Pruebas de Integración	3
Prueba de integridad de los datos y la base de datos	2
Prueba de Seguridad	2
Pruebas de Funcionalidad	2
Prueba de Performance	2
Prueba de Carga	2
Prueba de Estrés	2
Prueba de Volumen	2
Prueba de Fallas y Recuperación	2

**Tabla # 6: Peso asignado a las pruebas según su importancia.**

Una vez que se tiene la importancia que va a tener la aplicación de cada una de estas pruebas, para el centro, se hace un análisis para determinar cuáles de ellas fueron las que encontraron algún error, esto se supo mediante el estudio realizado de las no conformidades y de los casos de pruebas, obteniéndose las pruebas que fueron mencionadas anteriormente (Ver Tabla # 10).

Teniendo en cuenta el peso que van a tener estas pruebas y cuáles son las que se ven afectadas, se observaran los criterios de confiabilidad a los que tributan las mismas, los cuales se van a ver afectados de un modo u otro (Ver Tabla # 7).

En la siguiente tabla aparecen las pruebas que se ponen de manifiesto en el Documento de No Conformidad y en los Casos de Pruebas y se asocia a cada una de ellas el porcentaje de no conformidades

encontrado, sus respectivos criterios de confiabilidad (Ver Tabla # 8) y los pesos que fueron asignados a ellas en la tabla anterior.

<b>Pruebas</b>	<b>Porciento de errores por prueba</b>	<b>Criterios</b>	<b>Pesos</b>
Funcionales	26%	P, E	2
Unidad	15%	Mo.	3
Integración	13%	Mo.	3
Integridad	11%	Ma, R, TF	2
Fallas y Recuperación	10%	R, TF	2
Carga	10%	R	2
Estrés	8%	R	2
Performance	7%	R	2

**Tabla # 7: Tabla de pruebas asociadas a los criterios.**

<b>Siglas</b>	<b>Significados</b>
P	Precisión
Ma.	Madurez
Mo.	Modularidad
R	Recuperabilidad
E	Exactitud
TF	Tolerancia a fallas

**Tabla # 8: Significado de las siglas de los criterios.**

Una vez que se tienen los criterios que se ven afectados y el peso de cada prueba por criterio, se pasa a hallar el nivel de confiabilidad de cada uno. Para poder realizar esto, se suman los pesos por criterios y se dividen por la cantidad de pruebas, obteniendo el promedio de cada uno de los criterios de confiabilidad (Ver Tabla # 9).

<b>Pruebas</b>	<b>P</b>	<b>Ma</b>	<b>Mo</b>	<b>R</b>	<b>E</b>	<b>TF</b>
Funcionales	2				2	

Unidad			3			
Integración						
Integridad		2	3	2		2
Fallas y Recuperación				2		2
Carga				2		
Estrés				2		
Performance				2		
<b>Promedio</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>2</b>	<b>1.5</b>	<b>2</b>

**Tabla # 9: Cálculo de la confiabilidad por criterios.**

Una vez que se obtienen estos resultados, pasan por la tabla de rangos los cuales fueron divididos lo más equitativamente posible dentro de un rango del 1 al 3 y se le asignan los niveles según corresponda a cada valor (Ver Tabla # 10).

<b>Rango</b>	<b>Nivel de Confiabilidad</b>
2.5-3	Bajo
1.5-2.4	Medio
1-1.4	Alto

**Tabla # 10: Rango del Nivel de Confiabilidad por criterios.**

Obteniendo que el criterio precisión, madurez, recuperabilidad, tolerancia a falla y exactitud presentan un nivel de confiabilidad medio, mientras que el de modularidad presenta un nivel de confiabilidad bajo.

Para calcular el Nivel de Confiabilidad general de este producto, se le asigna un peso a cada criterio. Este peso se determina dependiendo de la importancia que tenga este, la cual se determina teniendo en cuenta los resultados obtenidos en las entrevistas realizadas en el capítulo anterior, donde se pueden observar las necesidades que existe en el centro de que sean aplicados estos criterios. Dependiendo de estas es que se les atribuye un nivel de importancia a los criterios de confiabilidad asignando un valor del 1-3, siendo el 3 el valor más alto de importancia (Ver Tabla # 11).

<b>Criterios de Confiabilidad</b>	<b>Pesos</b>
-----------------------------------	--------------

Precisión	1
Madurez	3
Modularidad	3
Recuperabilidad	2
Exactitud	1
Tolerancia a fallas	2
Consistencia	2
Simplicidad	1
Compleitud	2

**Tabla # 11: Peso por Criterios.**

En la siguiente tabla se utilizan los resultados obtenidos en la Tabla # 9 y se le asigna a cada uno de los criterios un peso, los cuales fueron definidos anteriormente en la Tabla # 11. El resultado de cada criterio se multiplica por el peso asignado y el valor que se obtiene se suma y se divide por la cantidad de criterios total. Obteniéndose un valor total de **4.4** (Ver Tabla # 12).

<b>Criterios</b>	<b>Valor</b>	<b>Peso</b>	<b>Multiplicación</b>
P	2	1	2
Ma.	2	3	6
Mo.	3	3	9
R	2	2	4
E	1.5	1	1.5
TF	2	2	4
<b>Promedio</b>			<b>32/6</b>
<b>Resultado</b>			<b>4.4</b>

**Tabla # 12: Confiabilidad total del producto.**

Para saber a qué nivel de confiabilidad pertenece este valor, se calculó el rango mediante el mismo procedimiento explicado en la Tabla # 12, utilizando el mismo valor de confiabilidad por criterio calculado

en la Tabla # 9, pero asignándole el mayor peso posible a cada criterio, que sería 3, este resultado da el rango total que se puede llegar a alcanzar, el cual va a ser dividido lo más equitativamente posible.

Rango	Nivel de Confiabilidad Total
4.5-6	Bajo
3.5-4.4	Medio
1-3.4	Alto

**Tabla # 13: Rango para la confiabilidad total del producto.**

Teniendo un conocimiento del rango y del valor obtenido en la Tabla # 12, podemos decir que el producto Generador de Reportes con la aplicación de un 66,6% de los criterios de confiabilidad del total de los que se utilizan en el modelo, los cuales tributan las pruebas que se realizan en el Modelo de Confiabilidad V, presenta un nivel de confiabilidad **medio**. Este nivel puede ser más preciso con la aplicación de la propuesta completa a un producto desde su fase inicial.

Con la aplicación del Modelo de Confiabilidad V podemos saber el grado de confiabilidad que presentan los productos desarrollados con PostgreSQL en DATEC, permitiendo que al tener un conocimiento de esto el centro pueda mejorar sus productos, aumentando la posibilidad de obtener un producto con un grado de confiabilidad más elevado.

### 3.4 Conclusiones

En este capítulo se realizó el método de Evaluación de Expertos y se obtuvo un alto índice de aceptación por parte de ellos con respecto al Modelo de Confiabilidad V. Además de que se aplicó una parte del modelo en el producto Generador de Reportes desarrollado en DATEC y se calculó el nivel de confiabilidad que presenta este modelo con la aplicación de las pruebas, las cuales tributan a los criterios de confiabilidad que fueron utilizados para poder realizar estos cálculos, tanto por criterios de confiabilidad como el nivel general de confiabilidad del modelo, obteniendo que el producto Generador de Reporte tiene un nivel de confiabilidad Medio.

Con la aplicación de este modelo de confiabilidad se puede tener un conocimiento del nivel de confiabilidad que presentan los productos de software, fundamentalmente los desarrollados con



PostgreSQL en DATEC, permitiendo que con este conocimiento el centro pueda saber si un producto de software es confiable o no y según estos resultados pueda realizarle cambios a los productos para que sean más confiables.

## CONCLUSIONES

---

- En este trabajo investigativo se caracterizó la situación existente de las pruebas para la confiabilidad, de los productos diseñados con PostgreSQL en DATEC. Llegando a la conclusión que es necesaria la aplicación de la propuesta en el centro, para obtener el nivel de confiabilidad de los productos de software mediante la utilización de los criterios de confiabilidad y de las pruebas que propone el modelo.
- Se propusieron los elementos que integran el modelo de prueba de confiabilidad de los productos diseñados con PostgreSQL en DATEC y se explicó el uso de cada uno de ellos.
- Se evaluó la efectividad del modelo diseñado mediante el método de Validación de Expertos.
- Se aplicó el **Modelo de confiabilidad V** en el producto Generador de Reportes, desarrollado con PostgreSQL en DATEC.

## RECOMENDACIONES

---

Se recomienda que el modelo sea aplicado en un futuro a un producto desde el comienzo de su desarrollo, para que se obtengan resultados positivos. En el mismo en cuanto a un factor que es muy importante para obtener una buena calidad en un producto, la confiabilidad.

Se recomienda además que este modelo se haga extensivo hacia todas las áreas de la Universidad donde se desarrollen software con PostgreSQL, por la importancia que tiene la aplicación del mismo.

## BIBLIOGRAFÍA

---

**Axentia y Sergio Villagra. 2006.** *Una Introducción a CMMI*. s.l. : White Paper, 2006.

**Bench-Colombia.** <http://bdigital.eafit.edu.co>. <http://bdigital.eafit.edu.co>. [En línea]

**Bichachi, Dra. Diana Susana. 2007.** El uso de las Listas de Chequeo ( Chesk-List) como herramienta para controlar la calidad de la ley. *Consejo Provincial de las Mujeres*. [En línea] 03 de 07 de 2007. [Citado el: 10 de 05 de 2010.] [http://www.consejomujeres.gba.gov.ar/doc/check\\_lista.pdf](http://www.consejomujeres.gba.gov.ar/doc/check_lista.pdf).

**Boucchechter, Carolina Zibert van Gricken e Israel. 2005.** *Pruebas de Confiabilidad*. 2005.

**Castillo, Dr. C. Manuel Capote. 2006.** monografias.com. *¿Qué resultados científicos se pueden obtener en una investigación educacional?* [En línea] 2006. [Citado el: 10 de 04 de 2010.] <http://www.monografias.com/trabajos64/resultados-cientificos-investigacion-educacional/resultados-cientificos-investigacion-educacional.shtml>.

**Coral Calero, Ismael Caballero, M<sup>a</sup> Ángeles Moraga, Manuel Serrano. 2008/2209.** *Introducción a la calidad. Modelos de calidad. ISO 9126. Ejemplos de modelos de calidad: PQM, WQM*. 2008/2209.

**Corporativo, Pearson Addison Wesley.SA. 2005.** *Ingeniería del software. Séptima edición*. España : Pearson Addison Wesley, 2005. ISNB 10, ISBN 13.

**Dávila Nicanor, Leticia y Mejía Alvarez, Pedro. 2008.** *Evaluación de la Calidad de Software en Sistemas de Información en Internet*. México : s.n., 2008. DF. 07300.

**Definición ABC. 2010.** *Definición de Entrevista*. 2010.

**Dña. Amaya Atencia Yépez, D. Luis Redondo López. 2007.** *ANÁLISIS DE FIABILIDAD DE SISTEMAS APLICANDO TÉCNICAS DE CRECIMIENTO DE FIABILIDAD DEL SOFTWARE*. s.l. : RPM-AEMES, 2007. ISSN: 1698-2029.

**Entorno Virtual de Aprendizaje. 2010.** Entorno Virtual de Aprendizaje. *El diseño metodológico de la investigación científica*. [En línea] 22 de 06 de 2010. [Citado el: 1 de 05 de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=2756>.

—. EVA. EVA. [En línea] [Citado el: 5 de 12 de 2010.] <http://eva.uci.cu>.

**Escorial, Jorge Salamanca.** *CALIDAD DEL SOFTWARE*.

**Figueroa, María Antonieta Abud. 2006.** *Calidad en la Industria del Software. La Norma ISO-9126*. 2006.

**Fillottrani, Pablo R. 2007** . *Calidad en el Desarrollo de Software*. Universidad Nacional del Sur : s.n., 2007 .

**García Palomo, Miguel Ángel y Mamdouh, Elcuera. 2007.** *Modelo para la Capacitación de los Especialistas en Pruebas de*. Madrid : Paseo de la Castellana,, 2007. ISSN 1988-3455.

**Gonzalez Jardon, Carlos. 2006.** *La gestión del proceso de software. Primera parte.* 2006.

**Goñi, J.R. Zubizarreta, J. Iturrioz.UPV/EHU. 2006/07.** *Evaluación / Pruebas del Software*. País Vasco : s.n., 2006/07.

**Gracia, Joaquin. 2005.** CMM - CMMI. *Calidad* . [En línea] 14 de Agosto de 2005. [Citado el: 10 de 02 de 2010.] <http://www.ingenierossoftware.com/calidad/cmm-cmmi.php>.

**IEEE.**

**ISO. 2008.** International Standards for Business, Government and Society. *ISO/IEC 9126-2:2003 Software engineering -- Product quality -- Part 2: External metrics*. [En línea] 17 de 12 de 2008. [Citado el: 29 de 10 de 2009.] [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=22750](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22750)..

**L. A. Olsina<sup>1</sup>, M. F. Bertoa, G. J. Lafuente, M. A. Martín, M. Katrib, A. Vallecillo.** *Un Marco Conceptual para la Definición y Explotación de Métricas de Calidad*. Cuba, España, Argentina : s.n.

*Las normas ISO.* **Sans, M<sup>a</sup> Carme. 1998.** 129, España : s.n., 1998, Revista Bibliográfica de Geografía y Ciencias Sociales, Vol. Biblio 3W. [ISSN 1138-9796].

**León, Rolando Alfredo Hernández. 2009.** *Una Introducción a la Gestión de Proyectos*. Ciudad de la Habana : s.n., 2009.

**Licea Castellanos, Andrés Luis, Charro Pérez, Karel Alejandro. 2009.** *Propuesta de Herramientas para Pruebas de Software Automáticas*. Habana : s.n., 2009.

**López, Rosalina González. 2007.** *APROXIMACIÓN METODOLÓGICA PARA LA METAEVALUACIÓN DE LA CALIDAD PROGRAMÁTICA*. [En línea] 3 0 de 01-12 de 2007. [Citado el: 03 de 05 de 2010.] <http://www.saber.ula.ve/bitstream/123456789/173032/articulo14.pdf>. 1 5 4 - 1 6 1.

**López, Rubén Gómez. 2009.** Informática Profesional. *Informática Profesional*. [En línea] 20 de 1 de 2009. [Citado el: 2 de 5 de 2010.] <http://www.adictosaltrabajo.com>.

**M & T Consulting. 2009.** [En línea] 2009. [Citado el: 18 de 02 de 2010.] [http://www.myt.com.pe/principal/Eva\\_CMMi.php](http://www.myt.com.pe/principal/Eva_CMMi.php).

**Mancha, Beatriz Pérez Ia. 2005.** Facultad de Ingeniería. *Facultad de Ingeniería*. [En línea] 2005. [Citado el: 16 de 06 de 2010.] <http://www.fing.edu.uy/~bperez/develoPro/web/plantillas/>.

**Masadelante. 1999 - 2010.** masadelante.com/servicio y recursos para tener exito en internet. *¿Qué es una Base de datos? - Definición de Base de datos.* [En línea] España, 1999 - 2010. [Citado el: 6 de 12 de 2009.] <http://www.masadelante.com/faqs/base-de-datos>.

**Ministerio de Educación. 2007.** *Orientaciones para el Monitoreo y Evaluación de los Planes Operativos de las Direcciones Regionales de Educación y Unidades de Gestión Educativa Local, dependientes de los Gobiernos Regionales.* Lima, Perú : s.n., 2007.

**Morilla, José Joaquín Ruiz.** *ISO 9126 vs. SQuaRE.*

**NC-ISO/IEC 9126-1. 2005.** *INGENIERÍA DE SOFTWARE—CALIDAD DEL PRODUCTO—PARTE 1: MODELO DE LA CALIDAD.* Cuba : s.n., 2005. ICS: 35.080.

**Organización Internacional de Normalización (ISO).**

**Palazzolo, Lic. Cecilia. 2005.** *Calidad de Software, Herramientas de Software.* 2005.

**Pérez, Ivan Jose Sanes. 2010.** *Diseño de casos de prueba. Diseño de casos de prueba.* [En línea] 2010. [Citado el: 16 de 06 de 2010.] <http://www.angelfire.com/empire2/ivansanes/bywbox.htm>.

**Prado, Elena Raja. 2007.** *Casi todas las pruebas del software.* s.l. : Quality Assurance & Regulatory Affairs, NTE S.A, 2007. ISSN 1988-3455.

**Pressman. 1992.** 1992.

**Pressman, Roger S. 2002.** *Ingeniería de software. Quinta Edición.* España : McGraw Hill, 2002.

**Quiñones, Ernesto. 2003 - 2009.** *Modelos de Calidad de Software y Software Libre.* Perú : s.n., 2003 - 2009.

**RODRÍGUEZ YUNTA, Luis. 2001.** *Bases de datos documentales: estructura y uso.* Madrid : CINDOC-CSIC, 2001.

**SCALONE, LIC. FERNANDA. 2006.** *ESTUDIO COMPARATIVO DE LOS MODELOS Y.* Buenos Aires : s.n., 2006.

**The Apache Software Foundation. 1999-2007.** Apache Jakarta Project. *Apache Jakarta Project.* [En línea] 1999-2007. [Citado el: 9 de 4 de 2010.] <http://jakarta.apache.org/>.

## REFERENCIAS BIBLIOGRÁFICAS

---

**(Entorno Virtual de Aprendizaje, 2010) Entorno Virtual de Aprendizaje. 2010.** Entorno Virtual de Aprendizaje. *El diseño metodológico de la investigación científica*. [En línea] 22 de 06 de 2010. [Citado el: 1 de 05 de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=2756>.— EVA. EVA. [En línea] [Citado el: 5 de 12 de 2010.] <http://eva.uci.cu>.

**(Figuroa, 2006) Figuroa, María Antonieta Abud. 2006.** *Calidad en la Industria del Software. La Norma ISO-9126*. 2006.

**(ISO, 2008) ISO. 2008.** International Standards for Business, Government and Society. *ISO/IEC 9126-2:2003 Software engineering -- Product quality -- Part 2: External metrics*. [En línea] 17 de 12 de 2008. [Citado el: 29 de 10 de 2009.] [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=22750](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22750).

**(Las normas ISO, 1998) Las normas ISO. Sans, M<sup>a</sup> Carme. 1998.** 129, España : s.n., 1998, Revista Bibliográfica de Geografía y Ciencias Sociales, Vol. Biblio 3W. [ISSN 1138-9796].

**(Dávila Nicanor, et al., 2008) Leticia Dávila Nicanor, Pedro Mejía Alvarez. 2008.** *Evaluación de la Calidad de Software en Sistemas de Información en Internet*. México : s.n., 2008. DF. 07300.

**(Licea Castellanos, 2009) Licea Castellanos, Andrés Luis, Charro Pérez, Karel Alejandro. 2009.** *Propuesta de Herramientas para Pruebas de Software Automáticas*. Habana : s.n., 2009.

**(López, 2007) López, Rosalina González. 2007.** *APROXIMACIÓN METODOLÓGICA PARA LA META-EVALUACIÓN DE LA CALIDAD PROGRAMÁTICA*. [En línea] 30 de 01-12 de 2007. [Citado el: 03 de 05 de 2010.] <http://www.saber.ula.ve/bitstream/123456789/173032/articulo14.pdf>. 154 - 161.

**(Mancha, 2005) Mancha, Beatriz Pérez Ia. 2005.** Facultad de Ingeniería. *Facultad de Ingeniería*. [En línea] 2005. [Citado el: 16 de 06 de 2010.] <http://www.fing.edu.uy/~bperez/develoPro/web/plantillas/>.

**(Masadelante, 1999 - 2010) Masadelante. 1999 - 2010.** [masadelante.com/servicio](http://masadelante.com/servicio) y recursos para tener éxito en internet. *¿Qué es una Base de datos? - Definición de Base de datos*. [En línea] España, 1999 - 2010. [Citado el: 6 de 12 de 2009.] <http://www.masadelante.com/faqs/base-de-datos>.

**(NC-ISO/IEC 9126-1, 2005) NC-ISO/IEC 9126-1. 2005.** *INGENIERÍA DE SOFTWARE—CALIDAD DEL PRODUCTO—PARTE 1: MODELO DE LA CALIDAD*. Cuba : s.n., 2005. ICS: 35.080.

**(Pérez, 2010) Pérez, Ivan Jose Sanes. 2010.** Diseño de casos de prueba. *Diseño de casos de prueba*. [En línea] 2010. [Citado el: 16 de 06 de 2010.] <http://www.angelfire.com/empire2/ivansanes/bywbox.htm>.

**(Pressman, 2002)** Pressman, Roger S. 2002. *Ingeniería de software. Quinta Edición.* España : McGraw Hill, 2002.

**(Quiñones, 2003 - 2009)** Quiñones, Ernesto. 2003 - 2009. *Modelos de Calidad de Software y Software Libre.* Perú : s.n., 2003 - 2009.

## GLOSARIO DE TÉRMINOS

---

**ETL:** Extracción, Transformación y Carga

**Estándares:** Son acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados consistentemente como reglas, guías o definiciones de características para asegurar que los materiales, productos, procesos y servicios cumplan con su propósito

**Modelos de Calidad:** Son un conjunto de buenas prácticas durante todo el ciclo de vida del software, enfocados en los procesos de gestión y desarrollo de proyectos.

**Modelos de Pruebas:** Son un conjunto de pruebas que se realizan ya sea de forma automática o manual. Proponen pruebas en las cuales la industria debe hacer énfasis para de esta forma cumplir con el propósito para el cual fue creado. Pueden incluir pruebas generales o detalladas ofreciendo así una visión del sistema en consideración. Integran diferentes prácticas que permiten medir los avances de la calidad.

**Base de Datos:** Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesiten.

**Gestor de Base de Datos:** Son los que se encargan de administrar la base de datos.

**Pruebas:** Son un elemento importante para garantizar la calidad de los productos. Su principal objetivo es encontrar la mayor cantidad de errores en el software. No aseguran la ausencia de defectos en el software, sólo demuestran la presencia de errores en el mismo.

**Software:** Es el conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica.

**SGDB:** Sistema Gestor de Base de Datos.

**BD:** Base de Datos