

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
Facultad 9



"Indicadores de Calidad para el Proceso de Desarrollo  
de Software de Simulación"

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS

AUTORES: Yurisbel Laffita Labañino  
Irina Quintana Griñan

TUTOR: Ing. Enrique Pérez Rodríguez  
COTUTOR: Ing. Nilberto Caridad Chavez Marquez

Ciudad de la Habana, Julio 2008.  
"Año 50 de la Revolución"

*“Así como el hierro se oxida por falta de uso, así también la inactividad destruye el intelecto”*

*Leonardo da Vinci*

*Dedico mi trabajo de diploma a todas las personas que significan muchos para mi, a todas aquellas que compartieron mis alegrías y mis tristezas y que estuvieron cuando las necesitaba. A mi mamá Leonor Labañino por haberme dado la vida y enseñarme a salir adelante, por estar siempre presente en los momentos en que más la necesité, por haber sido mi guía y mi sustento toda la vida, por haberme dado esperanza cuando la había perdido toda. Mami todo mi esfuerzo va dedicado a ti.*

*A mi hermana mayor Yodenia que siempre ha sido mi guía y ejemplo obligándome a ser mejor cada día para no defraudarla. A mi papá Santiago Laffita que aunque ya no está conmigo, mientras pudo, aportó su granito de arena. A Robert Hernández por haberme hecho reír en los momentos más difíciles y por haberme dado fuerza para seguir adelante. A mis abuelitos Ramona y Atico que sin su ayuda y dedicación este sueño no se hubiese hecho realidad. A mi novio Nilberto que sin su apoyo y su comprensión no hubiese salido adelante, gracias por existir, gracias "Titi".*

*A todas aquellas personas que han logrado de una manera ser especial en mi vida.*

*Yuris*

*Dedico mi tesis a mis adorables y queridos padres, por ser mi mayor inspiración de querer ser cada día una persona mejor en todos los aspectos de la vida. A mi mamita linda, por darme la oportunidad de vivir, gracias mima por tu apoyo incondicional de toda la vida, por tu amor y tu confianza, gracias por estar siempre a mi lado, por ser mi amiga y mi guía, a mi papito José, el hombre más loco e inteligente que he conocido en mi vida, gracias por la confianza depositada en mi, por tu amor y entrega, tu eres mi orgullo, gracias a los dos, por esto y mucho más, los quiero.*

*Dedico mi tesis a mi segunda mamá, que no por ser la segunda la quiero menos que a la primera, mi abuelita Rosa, gracias mi viejita por tu crianza, por tus consejos, por tu dedicación de siempre, a ti te debo gran parte de mi vida, gracias abuelita, te adoro, a ti te dedico mi gloria.*

*Dedico mi tesis a mi hermana Yamilka, que la quiero con mi vida, por soportar mis locuras y pesadeces, gracias mi herma por existir, te quiero. Dedico mi tesis a todos mis familiares que de una forma u otra han contribuido con mi educación y mi enseñanza y me han brindado su ayuda incondicional durante todos estos años de sacrificio y nostalgia, pero no puedo dejar de mencionar a mi tía Marisol la más carismática e intelectual de la familia, gracias tía, por tus consejos, un beso para ti.*

*Dedico mi tesis a mi novio, por soportarme todos estos años, por ser paciente, por los buenos y malos momentos, gracias "totingui" por llegar a mi vida y brindarme tu amor, te amo.*

*Irina*

Realizar un trabajo de diploma no es tarea fácil, lleva tiempo y dedicación y sin ayuda de muchas personas éste no habría podido realizarse, por lo que no podemos despedirnos antes, sin darles las gracias a todas aquellas personas que de una forma u otra estuvieron involucradas en la realización de nuestro trabajo.

En primer lugar les damos las gracias a Fidel y a la Revolución por darnos la oportunidad de formar parte de este proyecto del futuro, de esta universidad de excelencia.

Gracias a nuestros queridos padres por su apoyo incondicional, su amor y dedicación todos los años de nuestras vidas, gracias por enseñarnos andar por el camino correcto, los queremos mucho.

A nuestras hermanitas por cuidarnos, por querernos mucho y por siempre estar ahí para nosotras.

Gracias a Enrique Pérez por ser nuestro tutor y defender nuestro trabajo con inteligencia y brindarnos su ayuda para la realización del mismo.

Gracias a Nilberto Chávez por ser un cotutor excepcional, por ser muy paciente, por su ayuda constante y por ayudarnos a superar cada día las pruebas más difíciles.

Gracias a nuestros amigos por estar siempre cerca, gracias por su amistad verdadera y desinteresada.

Gracias a nuestros compañeros de aula y profesores, que nos hicieron reír durante estos cinco años, gracias por hacernos la estancia en la UCI más placentera e inolvidable.

Gracia a todas aquellas que no confiaron en nosotras e intentaron impedirnos la llegada hasta aquí, gracias por darnos las fuerza suficiente para demostrarle lo contrario.

En fin al mundo entero por enseñarnos a ser buenas compañeras, amigas, por enseñarnos a ser mejores cada día.

A todos muchas gracias

Irina y Yurisbel.

**Tutor:**

Ing. Enrique Pérez Rodríguez.

Profesor Adiestrado Universidad de las Ciencias Informáticas. Asesor de Calidad de la Facultad 9. Jefe de Colectivo de la asignatura Ingeniería de Software 1. Pertenece al área de desarrollo de investigación. Actualmente es profesor de la facultad

Email: [enriquepr@uci.cu](mailto:enriquepr@uci.cu)

Teléfono de apto: 837 2763

**Cotutor:**

Ing. Nilberto Caridad Chavez Marquez.

Profesor recién graduado, profesor adiestrado del departamento de PP-ISW, miembro de la reserva del comandante. Actualmente es profesor de Práctica Profesional en la facultad

Email: [nchavez@uci.cu](mailto:nchavez@uci.cu).

Teléfono del apto: 835 8894

**Asesora:**

Lic. Angie Rosa Lozano Pérez.

Profesora recién graduada, profesora adiestrada del departamento de humanidades .Licenciada en Lenguas Extranjeras. Graduada de la Universidad de Oriente. Actualmente profesora de ingles en la facultad.

Email: [angie@uci.cu](mailto:angie@uci.cu)

Teléfono de apto: 835 8849

## **Resumen**

Hoy en día la calidad es un término que preocupa a las empresas productoras de software y que debe tenerse en cuenta en todas las etapas del desarrollo del mismo, con el objetivo de satisfacer las necesidades de los clientes. Los indicadores de calidad proporcionan información objetiva que contribuye al mejoramiento de los procesos y productos de software, lo cual favorece al logro de la calidad.

El propósito de esta investigación fue proponer un conjunto de indicadores que ayudaran al mejoramiento de la gestión de la calidad durante el proceso de desarrollo de software de simulación en la facultad nueve. Para desarrollar la propuesta fue necesario realizar una investigación acerca de la aplicación de indicadores en la actualidad, analizando el escenario mundial, nacional y de la Universidad de las Ciencias Informáticas.

El trabajo contempla el análisis de la situación actual del polo de simulación de la facultad nueve, mediante la realización de encuestas a los líderes y estudiantes del proyecto cuyos resultados justifican la necesidad de poner en práctica los indicadores.

### **Palabras claves:**

Calidad de software, Proceso de desarrollo de Software de Simulación, Indicadores de calidad.

**Índice de Contenido**

Introducción ..... 10

Capítulo I "Fundamentación Teórica" ..... 15

1.1 ¿Qué es Calidad? ..... 15

    1.1.1 Calidad de Software ..... 17

    1.1.2 Calidad del proceso de desarrollo de software ..... 18

    1.1.3 Calidad del producto software ..... 20

    1.1.4 Modelos de Gestión de la Calidad de Software ..... 24

1.2 Proceso de desarrollo de software ..... 28

    1.2.1 Proceso de desarrollo de Software de Simulación..... 31

    1.2.2 ¿Cómo se controla el proceso de desarrollo de software?..... 33

1.3 Indicadores de Calidad ..... 34

    1.3.1 Indicadores de calidad de Software ..... 35

    1.3.2 Tipos de indicadores de calidad ..... 36

    1.3.3 ¿Qué hace un indicador de calidad de software? ..... 40

    1.3.4 ¿Cómo se formulan los indicadores de calidad de Software? ..... 41

    1.3.5 Situación problemática de los indicadores de calidad para software de simulación. .... 42

1.4 Análisis de otras soluciones existentes ..... 44

    1.4.1 Modelo Sistemático de Calidad (Mosca)..... 44

1.5 Conclusiones Parciales..... 45

CAPÍTULO 2: "Estudio del flujo de trabajo del polo de simulación de la facultad nueva y resultados".. 46

2.1 Características del polo de Simulación..... 46

2.2 Estrategias del Polo de Simulación ..... 47

2.3 Proceso de desarrollo del software de simulación del polo de la facultad nueva. .... 47

2.4 Principales problemas detectados en el polo de simulación de la facultad nueva. .... 49



2.5 Resultados de la encuesta realizada a líderes y estudiantes del polo de simulación de la facultad nueve.....	50
2.6 Conclusiones Parciales.....	54
CAPÍTULO 3: “Propuestas de Indicadores de Calidad”.....	55
3.1 Propuesta de los indicadores de calidad para software de simulación.....	55
3.1.1 Indicadores para evaluar la fiabilidad:.....	55
3.1.2 Indicadores para evaluar la Eficiencia en el Proceso Productivo y del Producto Software: .....	57
3.1.3 Indicadores para evaluar la mantenibilidad del software.....	59
3.2 Conclusiones Parciales.....	60
Conclusiones.....	62
Recomendaciones.....	63
Referencias Bibliográficas.....	64
Bibliografía:.....	67
Anexos.....	71
Glosario de términos.....	77

•

**Tablas y Figuras**

Figura 1: La calidad en el ciclo de vida del software(11) ..... 21

Figura 2: Característica de la calidad externa e interna (13) ..... 23

Figura 3: Características de la calidad en uso (13) ..... 24

Figura 4: Niveles del modelo de capacidad y madurez(8) ..... 26

Figura 5: Sistema de gestión de la calidad(15) ..... 27

Figura 6: Elementos del proceso del software(4) ..... 30

Figura 7: Relación entre elementos del proceso del software(19) ..... 30

Figura 8: Ciclo de desarrollo de un software de simulación(20) ..... 32

Figura 9: Gráfica de los resultados de la encuesta realizada al polo de simulación de la facultad nueve.  
..... 52

Figura 10: Modelo de medición de PSM [27]..... 73

Tabla 1: Puntos más significativo del polo de simulación ..... 76

## Introducción

Cada día el mundo de la informática evoluciona y se desarrolla. Su influencia es apreciada en todas las esferas de la sociedad y aún con la existencia de la brecha digital, se amplía cada vez más, los paradigmas con los cuales se veía la economía durante décadas no son los mismos hoy en día. El desarrollo de software constituye un sector de gran importancia a nivel mundial, el cual se encuentra en el centro de todas las grandes transformaciones sobre todo si se considera que los grandes temas del momento como lo son la economía digital, la evolución de las empresas y la administración del conocimiento, se resuelven con la ayuda del software.

Las grandes transnacionales y empresas de software con tecnología de punta van absorbiendo a las pequeñas y medianas empresas productoras de software, logrando situarse entre las más rentables a nivel mundial. En medio de las situaciones adversas a lo que se le suma un férreo bloqueo económico y tecnológico por parte de la principal potencia en esa rama en el mundo (EE.UU), Cuba no ha cesado en su empeño de desarrollar su industria de software.

De esta manera en años recientes, han comenzado a tener un rápido y masivo auge en el país el estudio y la aplicación de las Ciencias Informáticas, como parte de una estrategia que incluye desde la diseminación de conocimientos elementales en todos los sectores de la sociedad hasta la creación de una Universidad de las Ciencias Informáticas (UCI), en aras de dar respuesta a la demanda de recursos humanos altamente calificados en el sector de las Tecnologías de la Información y las Comunicaciones (Tics), capaces de dar respuesta a necesidades concretas de investigación, producción, los servicios y en general de la vida práctica.

La UCI desde su surgimiento ha mantenido como principio, la formación docente de sus recursos humanos, desde la producción mediante un proceso de aprendizaje que se basa esencialmente en la participación activa en un proceso productivo, que conlleva a soluciones de problemas reales de alto impacto social o económico, en consecuencia la institución se dio a la tarea de crear diferentes estructuras de organización para obtener dichos objetivos.

Con una matrícula de dos mil estudiantes distribuidos inicialmente en seis facultades, ya existiendo en la actualidad diez mil estudiantes distribuidos en diez facultades, respectivamente, las cuales están estructuradas y diseñadas para llevar paralelamente a la docencia una segunda formación que permite formar informáticos capaces de desarrollar habilidades en otros campos relacionados con la especialidad, que garantice el desarrollo económico del país en beneficio de la sociedad.

Esa segunda formación ha estado sujeta a procesos de organización que han ido evolucionando con el transcurso del tiempo, el desarrollo de las tecnologías, las necesidades de mercado y por consiguiente, las necesidades de la institución, todo ha ido progresando hasta la conformación de los polos productivos como célula fundamental del trabajo de formación, investigación para poder aportar a la producción en determinada área del conocimiento, que es indudablemente una de las mejores ideas organizativas de las que se ha pensado en el tiempo de fundada la institución.

La visión a largo plazo, aún sin tener resultados en el corto y mediano tiempo se puede decir que es la característica de más valor que aporta esta idea, de los polos productivos. Uno de los polos que se va desarrollando en la institución, no tan solo por lo novedoso sino por los resultados significativos que va teniendo, es el de Simulación.

En el polo de Simulación se centran todos los software de simulación que se desarrollan en la UCI, actualmente distribuidos en las facultades cinco y nueve respectivamente y que van destinados a la automatización de procesos encaminados a diseñar modelos de sistemas reales y llevar a término experiencias con ellos con la finalidad de comprender el comportamiento de dichos sistemas o evaluar nuevas estrategias dentro de los límites impuestos por cierto criterio o conjuntos de ellos.

Esta investigación se concentrará en el polo de simulación ubicado en la facultad nueve, polo creado en este año, que como idea nueva, quedan muchas cosas por definir, documentar e implementar; dada la necesidad que existe de incursionar no solo en el mercado nacional sino también en el internacional y de ir logrando una mayor competitividad entre las empresas de desarrollo de software a nivel mundial.

Por lo antes expuesto y dada la madurez que se quiere lograr en la utilización de metodologías, procedimientos y estándares para el análisis, diseño, programación y prueba del software de simulación, es necesario un conjunto de indicadores que garantice la uniformidad de la filosofía de trabajo, con el objetivo de lograr una mayor confiabilidad, mantenibilidad, facilidad de prueba y a la vez que eleven la productividad en la UCI. Es difícil idear buenos indicadores de la calidad, estos deben proporcionar información fehaciente, objetiva y pertinente sobre asuntos de importancia; deben ser sensibles a los cambios en el desempeño; y deben ser fáciles de calcular con los datos disponibles. En estos momentos el centro no cuenta con dichos indicadores para medir la calidad del software de simulación, por lo que el problema a resolver en esta investigación lo constituye **la inexistencia de un conjunto de indicadores de calidad que regulen el proceso productivo del software de simulación en el polo de simulación de la facultad nueve de la UCI.**

Para resolver el problema antes esbozado se plantea como objetivo general **proponer un conjunto de indicadores que permitan garantizar la calidad del software en el polo de simulación de la facultad nueve de la UCI**. Los objetivos específicos que se derivan de este objetivo general son:

- Identificar indicadores de calidad empleados en el desarrollo de software.
- Determinar cuáles de los indicadores de calidad identificados se pueden emplear para la medición de la calidad en el desarrollo de software de simulación en la facultad nueve de la UCI.

Para darle cumplimiento a dicho objetivo fue preciso centrar la investigación en el objeto de estudio que no es más que el **proceso de desarrollo de software de simulación**.

El conjunto de indicadores que serán planteados como solución de este trabajo tendrá su incidencia final en **los softwares de simulación del polo de la facultad nueve en la UCI**, lo cual determina el campo de acción.

Para darle cumplimiento a los objetivos trazados se determinó que las tareas a realizar en esta investigación están dirigidas a:

- Investigar el estado del arte de los indicadores utilizados para mejorar la calidad de software en Cuba y el mundo.
- Estudiar el proceso de control de la calidad de software de simulación en la UCI.
- Estudiar los procesos en los cuales se desarrollan el software de simulación.
- Investigar sobre los indicadores de calidad empleadas en la UCI para software de simulación.
- Describir un conjunto de indicadores que sirvan de guía para el control de la calidad de software de simulación tanto en la facultad nueve como en la UCI.

El desarrollo de la investigación tendrá como hipótesis de trabajo la siguiente:

**Si se aplican indicadores de calidad que permitan evaluar la calidad del software de simulación en el polo de la facultad nueve, se logrará una mayor calidad en el proceso de desarrollo de software de simulación en dicha facultad y servirá a la vez de punto de partida para futuros proyectos productivos de características similares en la universidad.**

Donde es una hipótesis de tipo de dos variable:

**Variable independiente:** la aplicación de indicadores de calidad

**Variable dependiente:** mayor calidad en el proceso de desarrollo de software de simulación en la facultad nueve.

Para el desarrollo de la investigación se utilizan métodos teóricos y empíricos debido a que en esta fase la elaboración de un buen instrumento determina en gran medida la calidad de la información que permite a su vez tener los datos necesarios para dar respuesta al problema; dentro de los métodos teóricos que son los que permiten las relaciones que fluyen alrededor del objeto de estudio, se utilizan los de **análisis y síntesis, el histórico-lógico y el causal**. Dentro de los métodos empíricos serán utilizados: **la entrevista y la encuesta**. A continuación se especifica el por qué de la selección de los mismos.

- La encuesta y las entrevistas a líderes y estudiantes de proyectos de software de simulación así como al personal de Calidad UCI con el fin de recopilar toda la información que éstas pueden suministrar, para conocer como se desenvuelve el proceso de desarrollo del software de simulación en la UCI.
- El método de la entrevista que se utilizó fue la no estructurada, que prevé el tema pero no lleva un cuestionario rígido y puede variar de una persona a otra, es más flexible. Se aplica a especialistas en el tema, con el objetivo de obtener criterios de expertos.
- El método causal se utiliza para estudiar los factores que provocan la necesidad de un proceso de este tipo en la institución.
- El método análisis se utiliza para comprender y enunciar los indicadores con el fin de mejorar la calidad del proceso de desarrollo de los software de simulación y el de síntesis que es para plantear, describir y resumir el proceso que se desarrollará en la facultad nueve a partir de la investigación.
- El método histórico-lógico se utiliza para investigar si existe un proceso planteado en la facultad nueve y en la universidad que guíe en la actualidad el proceso de desarrollo de software de simulación.

Con el propósito de obtener resultados positivos se identificó una población homogénea constituida por 68 personas que representan un **Conjunto de líderes, profesores y estudiantes del polo de Simulación en la facultad nueve**. Para la selección de la muestra, se utilizó la técnica de muestreo no probabilístico intencional, donde el buen juicio permitió escoger a 10 integrantes de polo, representando un 15% de la población, cifra significativa para garantizar el comportamiento de la población de manera general.

El trabajo que se muestra a continuación está estructurado por tres capítulos. En el capítulo uno, "Fundamentación Teórica", se exponen los principales conceptos que contribuyen al mejor entendimiento del problema en cuestión, se describen detalladamente todos los argumentos que esclarecen el objeto de estudio. Por su parte en el capítulo dos "Estudio del flujo de trabajo del polo de simulación de la facultad nueve y resultados" se hace un estudio preliminar del polo de simulación con el fin de obtener los principales problemas existentes en el polo en cuanto a la utilización de indicadores de calidad de software; y en el capítulo tres, "Propuesta de Indicadores de Calidad de Software", se realizan diversas propuestas de un conjunto de indicadores para solucionar los problemas detectados y mejorar la calidad de los software de simulación en el polo de simulación de la facultad nueve en la UCI.

## Capítulo I “Fundamentación Teórica”

En este capítulo se abordan los elementos teóricos-conceptuales asociados al dominio del problema que se desea resolver, para que se pueda comprender de forma clara y exhaustiva el entorno que rodea el objeto de estudio en cuestión. También se describe de forma clara y detallada todos los argumentos y todo lo referente al tema de indicadores en cuanto al estado del arte en Cuba y en el Mundo, además se abordan los términos que sirven de soporte teórico a la investigación desarrollada y que estarán presentes a lo largo de este trabajo de diploma.

### 1.1 ¿Qué es Calidad?

Hasta el momento se ha mencionado el término Calidad sin una previa definición, es válido aclarar que depende en gran medida de la evaluación y criterio del público al que esté dirigido un producto o proceso. La Calidad Total es el estadio más evolucionado dentro de las sucesivas transformaciones que ha sufrido el término Calidad a lo largo del tiempo que ha avanzado dependiendo de las necesidades del contexto histórico y los objetivos a perseguir.

En un primer momento se habla de Control de Calidad, que se basa en técnicas de inspección aplicadas a la Producción, para evitar la salida de bienes defectuosos. Posteriormente nace el Aseguramiento de la Calidad, fase que persigue garantizar un nivel continuo de la calidad del producto o servicio proporcionado. Finalmente se llega a lo que hoy en día se conoce como Calidad Total, un sistema de gestión empresarial íntimamente relacionado con el concepto de Mejora Continua y que incluye las dos fases anteriores, centrada en la permanente satisfacción de las expectativas del cliente. Se tienen diversos conceptos como se ha mencionado anteriormente, que han venido variando según los años y las necesidades que se presentan en el transcurso de los mismos, para esto se presentan las definiciones de la calidad de ciertos precursores con respecto al tema:

Según **Edwards Deming** ,la calidad no es otra cosa más que "una serie de cuestionamientos hacia una mejora continua".(1)

De acuerdo a **Dr. J. Juran**, la calidad es "la adecuación para el uso, satisfaciendo las necesidades del cliente". (1)

**Kaoru Ishikawa** define a la calidad como la encargada de "desarrollar, diseñar, manufacturar y



mantener un producto de calidad que sea el más económico, útil y siempre satisfactorio para el consumidor".(1)

**Rafael Picolo**, Director General de Hewlett Packard afirma que " la calidad, no como un concepto aislado, ni que se logra de un día para otro, descansa en fuertes valores que se presentan en el medio ambiente, así como en otros que se adquieren con esfuerzos y disciplina".(1)

Según la Norma ISO 8402, se define Calidad como la totalidad de las características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades expresadas o implícitas. (2)

Por otra parte, el Diccionario de la Real Academia Española, conceptúa la Calidad como la "propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor", y es sinónimo de "buena calidad" la "superioridad o excelencia". (3)

Se considera que la palabra Calidad, resume las características, propiedades, cualidades y en general atributos propios de un producto, que determinan sobre éste la ausencia de defectos y la conformidad de todo el personal que de una forma u otra se vinculan con él, (productores, clientes, usuarios, etc.); es además la asociación de varios conceptos que se mencionan a continuación:

**Sistema de Calidad:** Conjunto de la estructura, responsabilidades, actividades, recursos y procedimientos de la organización de una empresa, que ésta establece para garantizar que lo que ofrece cumple con las especificaciones establecidas previamente por la empresa y el cliente, asegurando una calidad continua a lo largo del tiempo.

**Control de calidad:** El control está dirigido al cumplimiento de requisitos, no es más que un conjunto de actividades y técnicas operativas, utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio.

**Garantía de calidad:** Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisface los requerimientos dados sobre calidad, o sea tiene como finalidad inspirar confianza en que se cumplirá el requisito pertinente.

**Gestión de la calidad:** La gestión de la calidad es un conjunto de actividades y medios necesarios para definir e implantar un sistema de la calidad, responsabilizarse de su control, aseguramiento y mejora continua.

Por lo anterior, se puede concluir que la calidad se define como un proceso de mejoramiento continuo,

en donde todas las áreas de la empresa participan activamente en el desarrollo de productos y servicios, que satisfagan las necesidades del cliente, logrando con ello mayor productividad.

### 1.1.1 Calidad de Software

Cada empresa u organización puede definir sus propios factores de calidad. En el mundo del software por ejemplo la calidad está definida según la IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) por los siguientes factores: ausencia de defectos, satisfacción del usuario y conformidad con los requerimientos.

En este caso la Calidad de Software se define como la “concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” (4)

La calidad de software se conoce también como la ausencia de errores de funcionamiento, la adecuación a las necesidades del usuario, y el alcance de un desempeño apropiado (tiempo, volumen, espacio), además del cumplimiento de los estándares. Los objetivos que la calidad persigue son: La aceptación (utilización real por parte del usuario) y la mantenibilidad (posibilidad y facilidad de corrección, ajuste y modificación durante largo tiempo). (5)

El estándar de la IEEE para la calidad de software (6) define el término calidad del software como:

- La totalidad de rasgos y características de un producto de software que se refieren a su habilidad para satisfacer necesidades específicas.
- El grado en el cual el software posee una combinación deseada de atributos.
- El grado en el cual un usuario o cliente percibe que el software cumple con sus expectativas.
- Las características del software que determinan el grado en el cual dicho software en uso, cumple con las expectativas del cliente.

El interés por la calidad crece de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar los productos poco fiables o que realmente no dan respuesta a sus necesidades. Por lo antes mencionado se puede decir que la calidad de software es un conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia, está dada por lo que el cliente

quiere ,precisa y espera, y se plantea un adecuado balanceo de eficiencia, confiabilidad, facilidad de mantenimiento, portabilidad, facilidad de uso, seguridad e integridad.

Para poder cumplir con este balanceo entre los diferentes parámetros que deben garantizar la calidad, se debe hacer una buena medición de la misma, que como parte del mejoramiento continuo de los procesos, es la verificación sistemática del desempeño de un proceso a través del grado de cumplimiento de ciertos estándares e indicadores seleccionados.

A la hora de definir la calidad del software se pueden adoptar diferentes aproximaciones. Como primera aproximación es importante diferenciar entre la calidad del producto software y la calidad del proceso de desarrollo. No obstante, las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para la calidad del proceso de desarrollo, ya que la calidad del producto va a estar en función de la calidad del proceso de desarrollo. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto.

### **1.1.2 Calidad del proceso de desarrollo de software**

A raíz del avance vertiginoso de las tecnologías de la información y las comunicaciones (TIC) la calidad del proceso se ha vuelto más importante, y el interés de la calidad aumenta cada día más, haciendo necesario el desarrollo de la calidad del proceso. Actualmente, es importante resaltar que las organizaciones de software en general han comprendido que la clave de una entrega exitosa de un producto (en tiempo, en presupuesto, con la expectativa de calidad) radica en una efectiva gestión de su proceso software.

Pero antes de hablar de la calidad del proceso, se debe tener bien claro qué es un proceso en sí. Para este término existen varias definiciones que se darán a continuación:

- Proceso es “un conjunto de actuaciones, decisiones, actividades y tareas que se encadenan de forma secuencial y ordenada para conseguir un resultado que satisfaga plenamente los requerimientos del cliente al que va dirigido”(4)
- De esta manera se puede decir que un proceso de software es: “Un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software“(7).
- También se puede plantear que es un “Conjunto de actividades, métodos, prácticas y transformaciones que la gente usa para desarrollar y mantener software y los productos

de trabajo asociados (planes de proyecto, diseño de documentos, código, pruebas y manuales de usuario)".(8)

Teniendo en cuentas estos conceptos se resume que proceso de software no es más que un conjunto de pasos, actividades y métodos que las personas utilizan para desarrollar y mantener software y sus productos asociados, para conseguir un resultado que satisfaga plenamente con los objetivos del cliente al que va dirigido.

La calidad de proceso no es más que el control del proceso, que consiste en aplicar la misma al proceso de fabricación de un producto. Para ello se utilizan técnicas como el control de procesos estadísticos (SPC Statistical process control) aplicadas sobre muestras tomadas del producto. Al controlar el proceso, se evita que el producto corra el riesgo de salir defectuoso. Las ventajas de esta técnica suponen una menor pérdida, pues se evita que un producto defectuoso se siga creando en mal estado, creando mayores costes.

En pocas palabras, esto significa que se debe medir variables clave en el proceso para detectar cualquier variación inaceptable. De esta manera, se corrige la variación en el proceso y se evita la manufactura de productos fuera de especificación. Es importante considerar el estudio del proceso software, dada su influencia en la calidad del producto final.

Para la calidad de procesos se realiza el modelado de procesos de ciclos de vida, que la misma tiene como beneficio (9):

- Facilidad de entendimiento y comunicación, lo que requiere que un modelo de proceso contenga suficiente información para su representación.
- Soporte y Control de la Gestión del Proceso.
- Provisión para la automatización orientada al rendimiento del proceso
- Provisión para el soporte automático a la ejecución
- Soporte a la Mejora del Proceso, que necesita de la reutilización de procesos de software efectivos y bien definidos, la comparación de procesos alternativos y el soporte al proceso de desarrollo.

### 1.1.3 Calidad del producto software

El software, como producto de desarrollo, posee un conjunto de características que lo diferencian fuertemente del resto de los productos industriales. Esta diferenciación está marcada por lo intrínseco del producto, en tanto que es un producto lógico, abstracto, no físico; como producto se desarrolla, no se fabrica; la complejidad en su definición está dada por la volatilidad de los requisitos y la incorporación de nuevas funcionalidades a partir de lo desarrollado(10).

Con el desarrollo de las Tecnologías de la información y las comunicaciones (TIC), la calidad del producto se ha vuelto más importante, el interés de la calidad aumenta cada día más y los usuarios se vuelven cada día más selectivos y comienzan a rechazar productos poco confiables o que no dan respuesta a sus necesidades, por lo que la Calidad del Software (CS) es una disciplina más dentro de la Ingeniería del Software.

El principal instrumento para garantizar la calidad de las aplicaciones sigue siendo el Plan de Calidad, el cual se basa en normas o estándares genéricos y en procedimientos particulares. Los procedimientos pueden variar en cada organización, pero lo importante es que estén escritos, personalizados, adaptados a los procesos de la organización y que se sean cumplidos.

Es importante destacar que la calidad de un producto software debe ser considerada en todos sus estados de evolución (especificaciones, diseño, código, etc.). No basta con tener en cuenta la calidad del producto una vez finalizado, cuando los problemas de mala calidad ya no tienen solución o la solución es muy costosa.

Como se dijo anteriormente sin un buen proceso de desarrollo es muy difícil obtener un buen resultado del producto, por lo tanto éste va a presentar diferentes aspectos de la calidad(11):

- **Interna:** medible a partir de las características intrínsecas, como el código fuente.
- **Externa:** medible en el comportamiento del producto, como en una prueba.
- **En uso:** durante la utilización efectiva por parte del usuario.

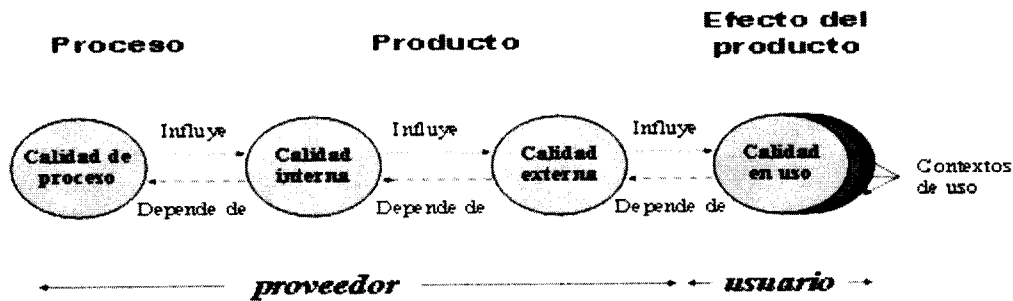


Figura 1: La calidad en el ciclo de vida del software(11)

Según Angélica de Antonio a la hora de hablar de la calidad de un producto software, los **principales problemas** a los que se enfrentan son(12):

- **La definición misma de la CS:** ¿Es realmente posible encontrar un conjunto de propiedades en un producto software que den una indicación de su calidad? Para dar respuesta a estas preguntas aparecen los Modelos de Calidad.
  - Modelos de calidad: En los modelos de calidad, la calidad se define de forma jerárquica. Resuelven la complejidad mediante la descomposición.
- **La comprobación de la CS:** ¿Cómo medir el grado de calidad de un producto software? Aquí aparece el concepto de Control de Calidad, que no es más que:
  - Control de Calidad: Actividades para evaluar la calidad de los productos desarrollados.
- **La mejora de la CS:** ¿Cómo utilizar la información disponible acerca de la calidad del producto software para mejorar su calidad a lo largo del ciclo de vida? No sólo es posible “medir” la calidad, sino también “construir” la calidad durante el proceso de desarrollo del producto. En este eje aparecen dos conceptos importantes:
  - Gestión de Calidad: Determinación y aplicación de las políticas de calidad de la empresa (objetivos y directrices generales)
  - Garantía o Aseguramiento de Calidad: Conjunto de actividades planificadas y sistemáticas necesarias para proporcionar confianza en que el producto software cumplirá los requisitos dados de calidad.

En resumen la calidad de un producto software es algo que siempre va a depender de las necesidades o requisitos que el producto deba satisfacer. Un producto de muy alta calidad para un usuario puede ser de muy baja calidad para otro.

## **Características de la Calidad Interna y Externa**

Los seis factores claves de calidad definidos por la ISO 9126(13) son:

- **Funcionalidad:** Es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuándo el software se usa bajo las condiciones especificadas(13). Se evalúa a partir de los subatributos: idoneidad, corrección, interoperabilidad, conformidad y seguridad.
- **Confiabilidad o fiabilidad:** La capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas(13). Está referido por los siguientes subatributos: madurez, tolerancia a fallos y facilidad de recuperación.
- **Usabilidad:** Capacidad del producto de software de ser comprendido, aprendido, utilizado y de ser atractivo para el usuario, cuando se utilice bajo las condiciones especificadas(13). Viene reflejado por los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.
- **Eficiencia:** Capacidad del producto de software para proporcionar una ejecución o desempeño apropiado, en relación con la cantidad de recursos utilizados, cuando se usa bajo condiciones establecidas(13). Viene reflejado por los siguientes subatributos: tiempo de uso y recursos utilizados.
- **Mantenibilidad:** Capacidad del producto de software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales. Está indicada por los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.
- **Portabilidad:** Capacidad de producto de software de ser transferido de un ambiente a otro. Está referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

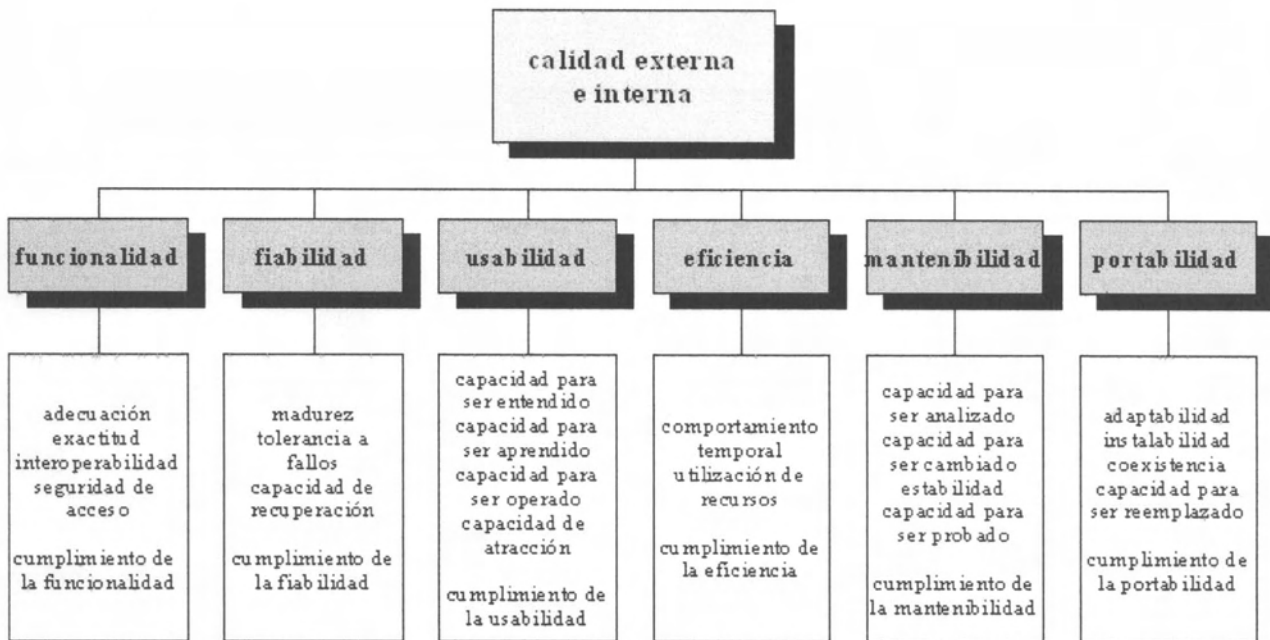


Figura 2: Característica de la calidad externa e interna (13)

#### Características de la Calidad durante el uso.

**Eficacia:** Capacidad del producto de software de permitir que los usuarios logren objetivos especificados con precisión e integridad en un contexto especificado.(13)

**Productividad:** Capacidad del producto de software de permitir que los usuarios dediquen una cantidad de recursos apropiada en relación con la eficacia alcanzada en un contexto de uso especificado. Entre los recursos se pueden incluir el tiempo para completar la tarea, los esfuerzos del usuario, los materiales o el costo de utilización en términos financieros.(13)

**Seguridad:** Capacidad del producto de software de alcanzar niveles aceptables de riesgo de daños a las personas, el negocio, el software, la propiedad o el ambiente en un contexto de uso especificado(13).

**Satisfacción:** Capacidad del producto de software de satisfacer a los usuarios en un contexto de uso especificado(13).



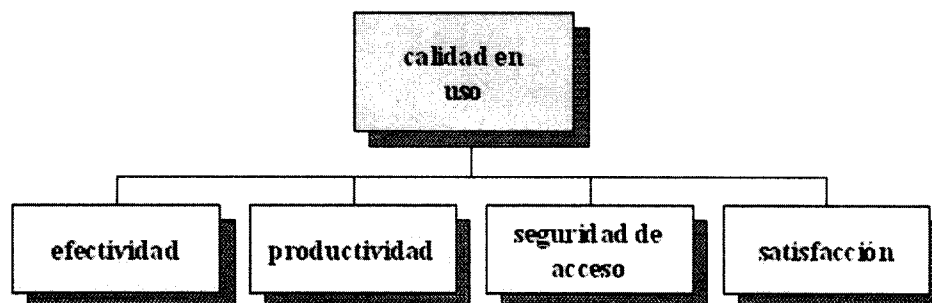


Figura 3: Características de la calidad en uso (13)

#### 1.1.4 Modelos de Gestión de la Calidad de Software

Para lograr la calidad, las empresas de software trabajan en pos de certificar sus productos teniendo en cuenta algunos de los modelos de calidad existentes.

Los modelos de calidad del software vienen a ayudar en la puesta en práctica del concepto general de calidad que se presentó en el apartado anterior, ofreciendo una definición más operacional, por su parte, se resume lo mencionado en epígrafes anteriores, la Gestión de la calidad no es más que todo lo que hace la organización para asegurar que sus productos y servicios satisfagan los requerimientos de calidad del cliente y cumplan con todas las regulaciones aplicables a dichos productos y servicios.

Cada día, más clientes se tornan compradores con conciencia de la calidad. Desean saber desde el comienzo qué empresa puede satisfacer sus necesidades. Un Sistema de Gestión de la Calidad certificado demuestra el compromiso de su empresa con la calidad y la satisfacción del cliente. Teniendo un Sistema de Gestión de Calidad se demuestra previsibilidad en las operaciones internas, así como capacidad para satisfacer los requisitos del cliente.

Por otro lado, “la gestión de la calidad es un conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades que permite una visión general que hace más fácil manejar, medir y mejorar los procesos internos.”

Los modelos de gestión de la calidad de software son un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos. Dichos modelos te dicen Qué hacer, no Cómo hacerlo, dependen de la metodología que se use y de los objetivos que se tracen en el negocio. A continuación se describen algunos modelos de gestión de la calidad:

### ➤ CMMI

CMMI (Modelo de Capacidad y Madurez Integrado) es un modelo que constituye un marco de referencia de la capacidad de desarrollo de software de diferentes empresas, siendo una base para evaluar la madurez de las mismas y una guía para llevar a cabo una estrategia de mejora continua. Este modelo surgió a partir de CMM (Modelo de Capacidad y Madurez) como resultado de la integración de varios modelos que definían la madurez en diferentes disciplinas, pero, dada la variedad constituían un problema para las empresas por lo que fue necesario agruparlos en el CMMI. (14)

La capacidad del proceso de software describe el rango de resultados esperados que se obtienen siguiendo un proceso de software, mientras que el desempeño representa los resultados reales obtenidos. La madurez del proceso de software se refiere a cuán explícitamente definido, administrado, medido, controlado y efectivo ha sido un proceso en específico.

CMMI dirige su enfoque a la mejora de procesos en una organización, los estudia y mide la capacidad para construir un software de calidad, atendiendo a una escala de cinco niveles (inicial, repetible, definido, dirigido y optimizado) (Ver anexo 1). Para que cada organización pueda enfocar la mejora de sus procesos se especifica en cada nivel de madurez un conjunto de áreas de proceso, que se describen en términos de prácticas; éstas son un conjunto de actividades que contribuyen a la implementación eficiente de un área de proceso, si se cumplen todas las prácticas y se satisfacen todas las áreas de proceso de un determinado nivel entonces la empresa podrá certificarse en ese nivel de madurez.

La versión más actual de modelo CMMI y de las más utilizada mundialmente es la versión 1.2 que fue creada por el Instituto de Ingeniería de Software de la Universidad Carnegie Mellon (SEI) en 1997 con el objetivo de la integración de los modelos de diversas disciplinas. Orientado a la mejora de procesos en diferentes niveles de madurez, más hacia proyectos específicos, recogiendo mejores prácticas. Enseñan el camino para alcanzar un nivel de madurez o nivel de capacidad en un área de proceso.

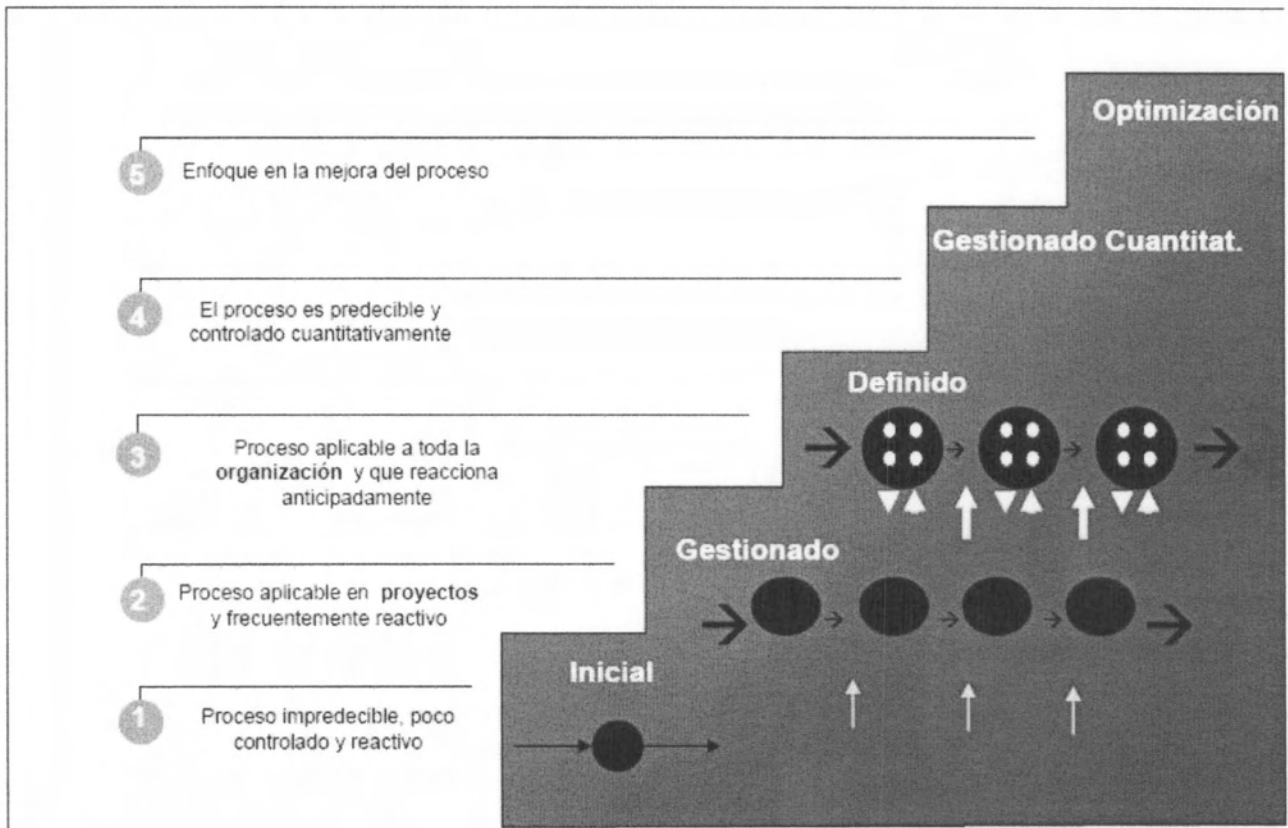


Figura 4: Niveles del modelo de capacidad y madurez(8)

➤ **Norma ISO/IEC 12207**

Diseñado por International Organization for Standardization (ISO) orientado al proceso de ciclo de vida para el software que incluye procesos y actividades que se aplican desde la definición de requisitos, pasando por la adquisición y configuración de los servicios del sistema, hasta la finalización de su uso. Este estándar tiene como objetivo principal proporcionar una estructura común para que compradores, proveedores, desarrolladores, personal de mantenimiento, operadores, gestores y técnicos involucrados en el desarrollo de software usen un lenguaje común, que dicho lenguaje se establece en forma de procesos bien definidos.

➤ **ISO 9000**

Modelo internacional de gestión de la calidad adoptado por la mayoría de las organizaciones. Se compone de estándares y guías relacionados con sistemas de gestión y de herramientas específicas como los métodos de auditoría (el proceso de verificar que los sistemas de gestión cumplen con el

estándar).

Dentro del contexto de ISO 9000 la definición estandarizada de calidad se refiere a todas las características de un producto o servicio que son requeridas por el cliente. Requieren que la organización establezca sistemas que mejoren continuamente su desempeño en esta área de gestión de la calidad. Se preocupa de la forma en que la organización hace su trabajo y no directamente en el resultado del mismo, es decir, se preocupa de los procesos y no de los productos. La certificación de ISO 9000 no es un requerimiento legal para acceder a mercados internacionales, pero puede ser beneficioso.

La certificación ISO 9000 puede servir como una forma de diferenciación "clase" de proveedores, particularmente en áreas de alta tecnología, donde la alta seguridad de los productos es crucial. En otras palabras, si dos proveedores están compitiendo por el mismo contrato, el que tenga un certificado de ISO 9000 puede tener una ventaja competitiva con algunos compradores.

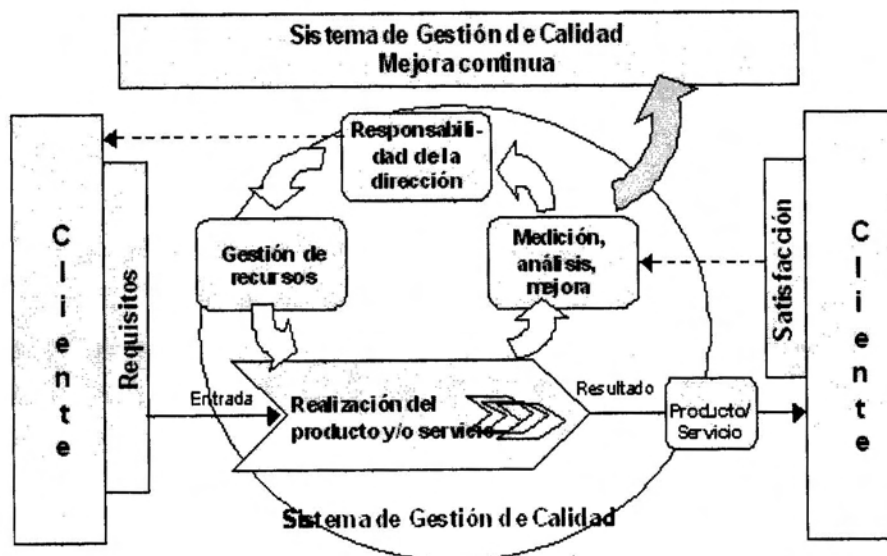


Figura 5: Sistema de gestión de la calidad(15)

➤ ISO 9001

La norma ISO 9001:2000, reconocida internacionalmente, es general. No es una norma para producto sino que puede aplicarse a cualquier sector industrial o de servicios. Ha sido creada por la Organización Internacional de Normalización (ISO), y su objetivo consiste en establecer requisitos internacionales para los Sistemas de Gestión de la Calidad.

La norma ISO 9001:2000 es la última versión de la norma, emitida el 15 de Diciembre de 2000. La nueva edición ha combinado las ediciones del 1994 de las normas ISO 9001, ISO 9002 e ISO 9003. La nueva norma pone mayor énfasis en la mejora continua y la satisfacción del cliente que las versiones de 1994. Se aplica a los procesos de la organización que influyen sobre la calidad.

### ➤ Norma ISO 9126

La norma ISO 9126 fue publicada en el año 1991. Sintetiza una serie de características que deben reunir los programas para que sean considerados de calidad, es decir, trata acerca de la calidad de los productos de software. La norma profundiza en otros detalles que hacen a las bondades de un software. Está redactada en un lenguaje sencillo y con el mismo esquema, punto por punto, que un contrato legal. Tres años más tarde de la publicación de la ISO 9126, la misma organización consideró necesario escribir una norma específica para los paquetes de software.

Este estándar establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de subcaracterísticas que permiten profundizar en la evaluación de la calidad de productos de software(13). En resumen, certifica si el software sirve para lo que dice que sirve y para evaluar esto, se conforma por 4 parte fundamentales:

- Modelo de Calidad
- Métricas Externas
- Métricas Internas
- Métricas de Calidad en Uso

### 1.2 Proceso de desarrollo de software

Anteriormente en la introducción se refleja que el objeto de estudio es el "Proceso de desarrollo del software de simulación ", donde es necesario plantear un conjunto de conceptos teóricos-conceptuales que definen el proceso de desarrollo de software de forma general.

Según I. Jacobson, G. Booch y J. Rumbaugh plantean que un proceso de desarrollo de software define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo(16).

En otras palabras un proceso de desarrollo de software no es más que un conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto.

También se define un proceso de desarrollo de software como en el que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo"(17).

A pesar de que el proceso de desarrollo de software no es único y que existe una variedad de propuestas de éste, existe un conjunto de actividades fundamentales que se encuentran presentes en todo el proceso de desarrollo(18):

- Especificación de software: Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.
- Diseño e Implementación: Se diseña y construye el software de acuerdo a la especificación.
- Validación: El software debe validarse, para asegurar que cumpla con lo que quiere el cliente.
- Evolución: El software debe evolucionar, para adaptarse a las necesidades del cliente.

Por su parte Roger S. Pressman caracteriza el proceso de desarrollo de software como un marco de trabajo de las tareas que se requieren para construir software de alta calidad. Dicho proceso se muestra en la figura 4, los elementos que se involucran se describen a continuación(4):

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

protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.

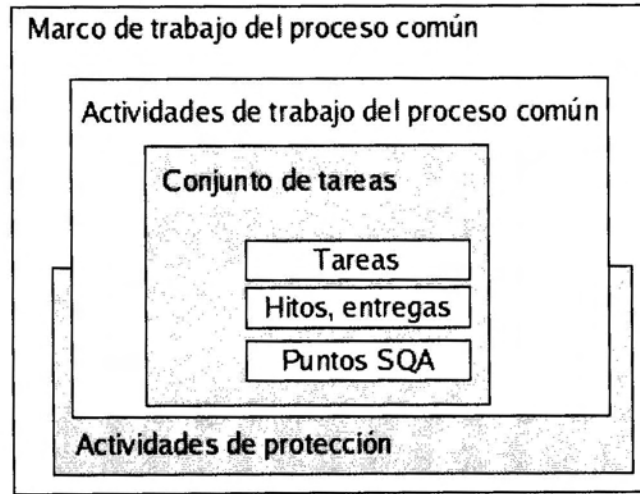


Figura 6: Elementos del proceso del software(4)

Otra perspectiva utilizada para determinar los elementos del proceso de desarrollo de software es establecer las relaciones entre elementos que permitan responder Quién debe hacer Qué, Cuándo y Cómo debe hacerlo(19).

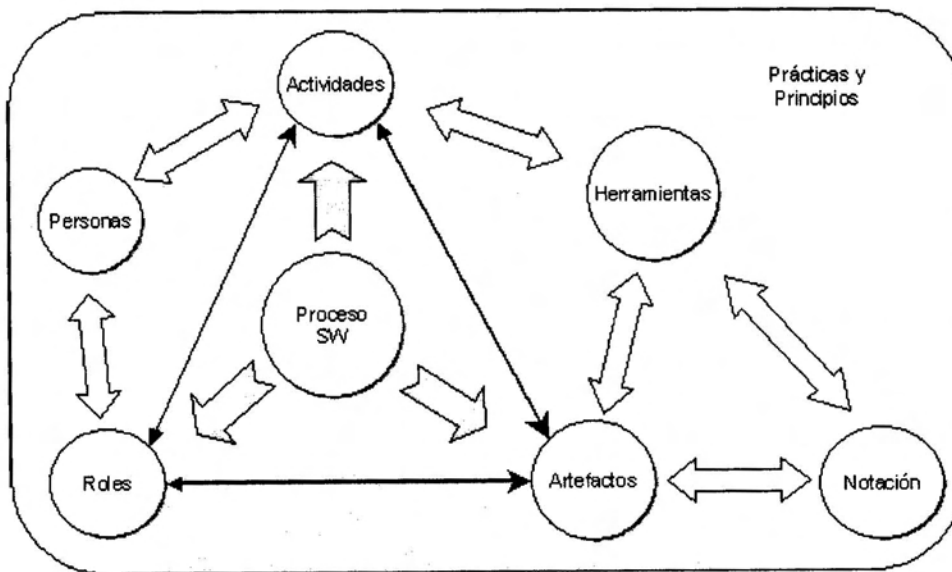


Figura 7: Relación entre elementos del proceso del software(19)

En la Figura 6 se muestran los elementos de un proceso de desarrollo de software y sus relaciones. Así las interrogantes se responden de la siguiente forma:

- Quién: Las Personas participantes en el proyecto de desarrollo desempeñando uno o más Roles específicos.
- Qué: Un Artefacto es producido por un Rol en una de sus Actividades. Los Artefactos se especifican utilizando Notaciones específicas. Las Herramientas apoyan la elaboración de Artefactos soportando ciertas Notaciones.
- Cómo y Cuándo: Las Actividades son una serie de pasos que lleva a cabo un Rol durante el proceso de desarrollo. El avance del proyecto está controlado mediante hitos que establecen un determinado estado de terminación de ciertos Artefactos.

### **1.2.1 Proceso de desarrollo de Software de Simulación**

Primero, antes de hablar de proceso de desarrollo de software de simulación, se debe hacer la pregunta ¿Qué es Simular? Pues la respuesta es muy sencilla. La simulación es un proceso virtual, un acercamiento a la realidad, aún si el modelo es dinámico, puede ser un reflejo exacto de lo que realmente sucede en un proceso, ya que no es posible representarlo matemáticamente en toda su complejidad.

La simulación es una herramienta de análisis que permite sacar conclusiones sin necesidad de trabajar directamente con el sistema real que se está simulando, ésta puede ser muy útil cuando no se dispone del sistema real o cuando resulta ser demasiado peligroso realizar experimentos sobre él(20). En fin la simulación es una técnica para analizar y estudiar sistemas complejos, permitiendo reunir información pertinente sobre el comportamiento del sistema porque ejecuta un modelo computarizado.

Según Winston (21) se puede definir la simulación como la técnica que imita el funcionamiento de un sistema del mundo real cuando evoluciona en el tiempo. La simulación no es una técnica de optimización, más bien es una técnica para estimar las medidas de desempeño del sistema modelado.

Aunque generalmente el informático utiliza la simulación como una herramienta de trabajo, se debe conocer que el proceso de desarrollo de software de simulación se centra o presenta dos objetivos principales: estimar el tiempo medio de respuesta del sistema y estimar la capacidad máxima del sistema, que no es más que el número máximo de peticiones por unidad de tiempo que puede soportar el sistema.



Este proceso presenta 3 modelos fundamentales en su ciclo de desarrollo, los cuales se describen a continuación y están representados en la figura 8:

**Modelo del sistema:** una vez fijados los objetivos se debe analizar la estructura y comportamiento del sistema e intentar plasmarlo en el papel. Se debe diseñar un modelo que abstraiga las partes más relevantes del sistema.(20)

**Modelo computacional:** Cuando el modelo ha sido suficientemente especificado (y a ser posible validado por algún experto) se pasa a la implementación del simulador.(20)

**Experimentación con el simulador:** una vez implementado y depurado el simulador debe diseñarse la batería de prueba que permiten extraer los resultados y a partir de estos las conclusiones de la simulación.(20)

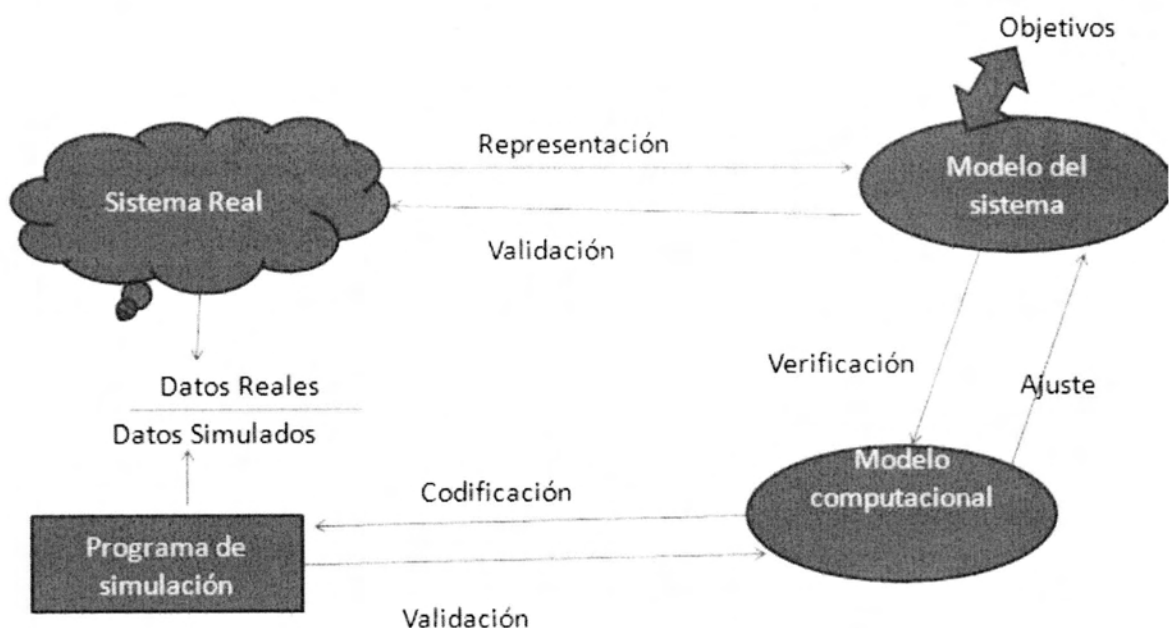


Figura 8: Ciclo de desarrollo de un software de simulación(20)

Con el paso del tiempo, el avance del proceso de desarrollo del software de simulación ha traído consigo numerosas ventajas tanto en el mundo de la informática como en el mundo real, permitiendo analizar grandes problemas complejos para los que no están disponibles resultados analíticos, permitiéndole al tomador de decisiones apreciar numerosas políticas y argumentos diferentes sin cambiar o experimentar ciertamente con el sistema real existente.

### 1.2.2 ¿Cómo se controla el proceso de desarrollo de software?

Generalmente lo que se utiliza para controlar el proceso de desarrollo, es básicamente, el uso de las metodologías de desarrollo, que sirve para subir la "calidad" del software, fundamentalmente permite producir lo esperado en el tiempo anhelado y con el costo estimado. Pues bien, para que un proyecto sea escalable y flexible a los cambios es lógico pensar que para lograr eso se necesita tomar en cuenta una de las muchas metodologías para el proceso de desarrollo de software.(22)

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Para dar una idea de qué metodología se puede utilizar y cual se adapta más a este medio, a continuación se mencionan dos de ellas que se consideran las más importantes: RUP y XP

#### **RUP (Rational Unified Process-Proceso Unificado de Desarrollo)**

Éste es uno de los procesos más generales que existe, está enfocado a cualquier tipo de proyecto así no sea de software, se basa en la documentación generada en cada uno de sus cuatro fases: 1. Inicio (puesta en marcha), 2. Elaboración (definición, análisis y diseño), 3. Construcción (implementación) y 4. Transición (fin del proyecto y puesta en producción) en las cuales se ejecutarán varias iteraciones (según el tamaño del proyecto)(23).

RUP se basa en casos de uso para describir lo que se tiene y lo que se espera del software, está muy orientado a la arquitectura del sistema a implementarse, documentándose de la mejor manera, basándose en UML (Unified Modeling Language - Lenguaje de Modelado Unificado)

Para poder usar RUP antes hay que adaptarlo a las características de la empresa, y medir de manera exacta el tiempo, costos y todos los demás recursos involucrados en el proceso.

#### **XP (Extreme Programming - Programación Extrema)**

XP, se fundamenta en el trabajo orientado directamente al objetivo, basándose para esto en las relaciones interpersonales y en la velocidad de reacción para la implementación y para los cambios que puedan surgir durante el desarrollo del proceso.(22)

Esto se logra, minimizando el riesgo de fallo del proceso manteniendo dentro del equipo a un representante "competente" del cliente, este representante es quién responderá a todas las preguntas

y dudas que surjan por parte del equipo de desarrollo durante el proceso, de forma que no se retrase la toma de decisiones.

XP se basa en historias de uso, estas historias las escribe el cliente o su representante dentro del equipo y describen los escenarios claves del funcionamiento del software, a partir de éstas se generan los releases entre el equipo y el cliente.

Una característica saltante de XP, es que el código siempre se produce en parejas, parejas que van cambiando constantemente para lograr así que todo el equipo sepa y pueda modificar según necesidades el código generado, esto logra en el equipo que los integrantes aprendan entre sí y compartan todo el código.

### **1.3 Indicadores de Calidad**

La vida cotidiana posee un gran número de aspectos, los cuales se comportan como variables, es decir, tienen características que cambian de modo tanto cualitativo como cuantitativo.

Dada la necesidad de describir, se debe operacionalizar la obtención de información sobre las variables que se observan. Lo que se usa para realizar esta operacionalización son los indicadores, que no son otra cosa que descriptores que se construyen con la información a la que se puede acceder (disponible o por recoger) y que dice algo de un aspecto (una variable) de la realidad.

Existe gran variedad de indicadores y su uso es muy común (horas y minutos como indicadores de tiempo, kilogramos y metros como indicadores de peso y longitud respectivamente, etc.), como mismo existe gran variedad de indicadores, existe varios conceptos dichos por varios autores, que se mencionan a continuación:

- La palabra indicador, proviene del latín *indicare* (señalar, avisar, estimar), alude a hechos o datos concretos que aprueban la existencia de cambios que conducen hacia resultados e impactos buscados(24).
- Son elementos del sistema de control de gestión, que proporciona información significativa sobre aspectos críticos o claves de una organización mediante la relación de dos o más datos(24).

- Es una medida cuantitativa que puede usarse como guía para controlar y valorar la calidad de las diferentes actividades. Es decir, la forma particular (normalmente numérica) en la que se mide o evalúa cada una de los criterios(24).
- Un indicador es una medida cuantitativa que valora el grado de cumplimiento de un estándar, que determina la brecha existente entre el desempeño ideal y el real, se expresa como un número, promedio, proporción o porcentaje y consta, en su mayoría, de numerador y denominador(24).

### 1.3.1 Indicadores de calidad de Software

Además existen varios tipos de indicadores, que se presentarán más adelante, pero esta investigación se concentrará en los indicadores de calidad de software:

- Un indicador de calidad, no son más que: "Herramientas para clarificar y definir, de forma más precisa, objetivos e impactos (...) son medidas verificables de cambio o resultado (...) diseñadas para contar con un estándar contra el cual evaluar, estimar o demostrar el progreso (...) con respecto a metas establecidas, facilitan el reparto de insumos, produciendo (...) productos y alcanzando objetivos"(25).
- Se puede definir un indicador de calidad de software como una medida cualitativa o cuantitativa del grado al que un componente del sistema, o un proceso posee un atributo dado. Está calculado sobre un solo componente o puede que se trate de un indicador compuesto basado en dos o más medidas. (Diccionario estadístico).
- Un indicador de calidad es un instrumento que permite a los actores de un proyecto observar o medir resultado. Constituyen señales de cambio reconocidas por diferentes actores(24).

Por lo antes expuesto se puede decir que los indicadores de calidad de software son medidas que describen la calidad relacionada a los estándares, para evaluar, estimar y demostrar el progreso de proyectos, produciendo productos con la calidad máxima y alcanzando los objetivos trazados. Siempre son elementos sectoriales que suelen responder de manera lineal a los incrementos o decrementos de los parámetros que los integran. Naturalmente, los indicadores pueden expresar porcentajes, pero siempre son resultado de cálculos aritméticos sencillos y de respuestas lineales. A partir de este nuevo concepto se guía el desarrollo de la investigación

Existen varias dificultades a la hora de elegir los indicadores, los mismos deben simplificar al máximo la información sacada de un conjunto de datos complejos (de manera que integran sólo parámetros observables fáciles para evaluar). Paralelamente, deben satisfacer a criterios de claridad y de representatividad. Se trata de una tarea difícil, que no excluye el riesgo de omitir unas informaciones importantes. Los indicadores pueden ser descritos como una pantalla que filtra la información. En consecuencia, si los indicadores no están escogidos de manera estrictamente rigurosa y apropiada, las señales de alerta y los cambios del sistema correrían peligro de pasar inadvertidos. Es por eso que la elección de "buenos" indicadores exige una buena comprensión y una cierta experiencia del sistema.

### 1.3.2 Tipos de indicadores de calidad

Los indicadores cubren una inmensa variedad de temas y áreas, de modo que existen múltiples clasificaciones. Los indicadores de calidad se construyen a partir de la experiencia, del conocimiento sobre el sector en el que se trabaja y, como es natural, respetando ciertas recomendaciones(26):

1. Deben ser siempre fáciles de capturar.
2. Deben enunciarse con objetividad y de la forma más sencilla posible.
3. Deben resultar relevantes para la toma de decisiones.
4. No deben implicar mucha dificultad en su interpretación.
5. Deben abarcar un amplio número de posibilidades. Siempre recordar que cuando se pone en práctica un indicador se observa, en la práctica, que éste mejorará pero que otros aspectos no recogidos por los indicadores podrían empeorar.
6. Normalmente debe construirse un "cuadro de mando" que permita monitorizar el indicador de forma sencilla y, siempre que se pueda, automatizada (la ayuda de la micro-informática aquí es decisiva).
7. Los términos usados en el indicador que puedan inducir a dudas, o sean susceptibles de diferentes interpretaciones, deben ser definidos para que todos los profesionales entiendan y midan lo mismo y de idéntica forma.

**Indicadores de Rendimiento:** son medidas con las que puede realizar gráficos de las deficiencias y progreso de la empresa. Para el éxito en la evaluación en una empresa, depende en gran medida de las métricas, de la identificación y prioridad de las que realmente importan en el momento adecuado.

Un uso incorrecto de ellas les puede causar daños a la empresa, al brindar un resultado truncado. Peor es en el caso de los indicadores claves de rendimiento que su uso incorrecto puede crear una ilusoria confidencialidad con respecto a la dirección de la empresa.(27)

**Indicadores de Gestión :** Es una expresión cuantitativa del comportamiento o el desempeño de toda una organización o una de sus partes, cuya magnitud al ser comparada con algún nivel de referencia, señala una desviación sobre la cual se toma acciones correctivas, proactivas o de control según el caso. Para diseñarlos se deben identificar las áreas críticas de la organización, así como las debilidades en cada uno de los procesos que involucre la funcionalidad de la empresa. Después es necesario asignar un indicador de medición para cada área respecto con la meta fijada(24).

### **Indicadores según tipo de información:**

- Indicadores Cuantitativos
- Indicadores Cualitativos
- Indicadores Mixtos

**Indicadores Cuantitativos:** son aquellos basado en metodologías cuantitativas de obtención de información. La información que suministran es fácilmente cuantificable, y la extrapolación estadística no presenta dificultades extraordinarias, sirven para determinar grados o niveles de calidad, es decir, dan como resultado un valor medible(25). Sus principales características son las siguientes:

- Expresan cantidad (cuánto/as) y frecuencia.
- Las unidades de medida son el número y el porcentaje.
- Los principales métodos e instrumentos utilizados para verificarlos son los censos, las encuestas, las entrevistas estructuradas, los modelos matemáticos y los modelos econométricos.

### **Ejemplos:**

- Porcentaje de mujeres miembros de organizaciones de productores en relación con el número de miembros hombres, por tipo de organización.
- Porcentaje de mujeres, por tipo de organización, que forman parte de las directivas.

**Indicadores Cualitativos:** sirven más para describir percepción o satisfacción, son particularmente útiles para comprender el punto de vista, los intereses y las prioridades de los actores del proyecto. Cuando el cambio esperado se refiere a variaciones en conocimientos, actitudes y prácticas (capacidades) de las personas para desarrollar sus potencialidades y superar la pobreza, la aplicación de este tipo de indicadores es relevante y necesaria durante y después de la implementación del proyecto(25). Sus principales características principales son las siguientes:

- Expresan cualidades, características o fenómenos intangibles.
- Se refieren a percepciones, prácticas, opiniones, habilidades o hechos.
- Describen, entre otras cosas, la situación y condiciones de vida de las personas; los cambios en la sensación, satisfacción y comprensión de las personas sobre algún hecho.
- Por lo general, las unidades de verificación de los indicadores cualitativos son: tipo, grado y nivel.
- Para su formulación se requiere la definición de escalas

**Ejemplos:**

- Grado de capacidad de las personas para gestionar recursos, expresado en la siguiente escala:
  - I. Hacen trámites legales relativos a su actividad;
  - II. Elaboran planes estratégicos y operativos para sus organizaciones;
  - III. Elaboran y presentan proyectos ante entidades públicas y privadas.

**Indicadores Mixtos:** Son aquellos que permiten verificar cambios cualitativos y, al mismo tiempo, determinar en cuántos individuos, organizaciones o instancias se presentan esos cambios. Por lo anterior, en los indicadores mixtos siempre se presenta una unidad de medida (número o porcentaje) y una unidad de verificación (tipo, grado o nivel). Los indicadores mixtos son particularmente importantes en la evaluación, ya que permiten verificar tanto la magnitud o profundidad de los cambios obtenidos como la amplitud con que se presentan, lo que hace posible una mayor comprensión acerca de la obtención de los resultados esperados(25).

### **Ejemplos:**

- Número (cantidad) de desarrolladores según grado (calidad) de participación en la toma de decisiones en la gestión de microempresas.
- Número (cantidad) y tipo (calidad) de microempresas dirigidas desarrolladores en dependencia del sexo.

Para redactar los indicadores mixtos correctamente, es necesario tener en cuenta, entre otros, los siguientes criterios:

*Validez:* Los indicadores deben permitir generar información válida; por ello, es preciso que representen con la mayor precisión posible el fenómeno que se quiere verificar.

*Mensurabilidad:* Deben incluir unidades o variables de medida (porcentaje, número, grado, nivel), para permitir la medición y verificación de lo que se pretende conocer.

*Relevancia:* Significativos para revelar una realidad mayor, y correspondientes con el propósito (objetivo de mayor nivel) y resultados del proyecto.

*Oportunidad:* Asegurar que proporcionen información a intervalos necesarios, de tal suerte que resulten útiles para tomar decisiones de manera oportuna.

*Neutralidad:* No deben dar lugar a interpretaciones previas.

Ejemplo de indicador no neutro: porcentaje de organizaciones exitosas. Ejemplo de indicador neutro: porcentaje de organizaciones según nivel de fortalecimiento. En el ejemplo, el éxito puede ser interpretado de diversas maneras por diferentes personas que apliquen el indicador.

*Sensibilidad al género:* Los indicadores deben permitir observar el avance hacia la disminución de las brechas de género.

**Indicadores para la Evaluación del Desempeño:** son los indicadores que describen lo que hace, son muy conocidos como inductores (Lead measures Drivers), generalmente de corto plazo. Los indicadores de desempeño son instrumentos que sirven como guía para vigilar y evaluar la calidad de importantes funciones directivas, administrativas, clínicas y de apoyo(28). A continuación se puntualizan definiciones y relaciones de los indicadores utilizados para la evaluación del desempeño de un sistema, relacionados con la calidad y la productividad: eficiencia, efectividad y eficacia.



1. **Eficiencia:** miden el nivel de ejecución del proceso, se concentran en el "cómo" se hicieron las cosas y miden el rendimiento de los recursos utilizados por un proceso. Tienen que ver con la productividad. Se define "como la virtud y facultad para lograr un efecto determinado". La eficiencia expresa la cualidad de la gestión económica característica de la formación económica dada. En el Capitalismo dicho indicador se conoce como, la cuota de ganancia, cuyo criterio es la satisfacción de los intereses económicos privado, pero no expresa los intereses económicos de la clase dominante en su conjunto, pues cada capitalista lucha por tener más y vivir mejor que otro de su misma clase, vivir en una constante competencia. Sin embargo en el socialismo se lucha por obtener cada vez un mayor crecimiento de la eficiencia económica, pues la misma responde a los intereses de todas las sociedades, pues ella es la única dueña de los medios de producción y los mayores resultados que se obtengan con el nivel mínimo de los gastos beneficiará a toda la sociedad en su conjunto, mediante la aplicación de un sistema de leyes propia del socialismo.
2. **Efectividad:** Es la relación entre los resultados logrados y los resultados propuestos, o sea, permite medir el grado de cumplimiento de los objetivos planificados. Dicho criterio se vincula con la productividad a través de impactar en el logro de mayores y mejores productos (según el objetivo); sin embargo, adolece de la noción del uso de recursos. El indicador de efectividad sirve para medir determinados parámetros de calidad que toda organización debe preestablecer y también para poder controlar los desperdicios del proceso y aumentar el valor agregado.
3. **Eficacia:** Mide el logro de los resultados propuestos. Indica si se hicieron las cosas que se debían hacer, los aspectos correctos del proceso. "Para ser eficaz se deben priorizar las tareas y realizar ordenadamente aquellas que permiten alcanzarlos mejor y rápidamente, aquel que logre realmente satisfacer al cliente o impactar en el mercado. Es un criterio muy relacionado con lo que se ha definido como calidad (adecuación al uso, satisfacción del cliente)".

### 1.3.3 ¿Qué hace un indicador de calidad de software?

El concepto de indicador es esencial para poder medir la pertinencia de los sistemas de seguimiento y evaluación y anotar los cambios operados. Los indicadores pretenden simplificar la descripción y la explicación de un sistema (proceso o situación). Efectivamente, permiten evaluar, de manera sencilla y fiable, los aspectos complejos de un sistema (proceso o situación). Constituyen así elementos esenciales (puntos de referencia) y forman parte integrante de las actividades de seguimiento y evaluación(28).

Los indicadores se refieren a datos esencialmente cuantitativos, que permite darse cuenta de cómo se encuentran las cosas en relación con algún aspecto de la realidad que interesa conocer. Pueden ser medidas, números, hechos, opiniones o percepciones que señalen condiciones o situaciones específicas.

Los indicadores se utilizan para observar, describir y evaluar los diferentes aspectos de una situación actual, formular situaciones deseadas o comparar una situación común con relación a una situación deseada. Estas cifras sencillas y estas evaluaciones descriptivas o normativas resumen la complejidad del mundo en que se vive en una información significativa y fácil para administrar. Los indicadores sirven para comunicar informaciones sobre el sistema o el proceso, El conocimiento científico y el juicio son los criterios dominantes en los cuales descansan las particularidades de un indicador.

Los indicadores deben reflejarse adecuadamente a la naturaleza, peculiaridades y nexos de los procesos que se originan, por tanto, no es suficiente con uno solo de ellos para medir la gestión de empresas o proyecto, sino que se impone la necesidad de considerar los sistemas de indicadores, es decir, un conjunto interrelacionado de ellos que abarque la mayor cantidad posible de magnitudes a medir.

El indicador debe permitir medir de alguna manera lo que describe. Esto es importante para poder comparar el resultado de la observación que se hace con el de otra observación o con información de referencia. Unas veces la medición resulta un proceso bastante directo y sencillo (por ejemplo: el tiempo que el software permanece en ejecución), pero otras no lo son (por ejemplo: medir la calidad del proceso de un software de simulación en ejecución). (28)

#### **1.3.4 ¿Cómo se formulan los indicadores de calidad de Software?**

Un indicador generalmente se compone de un solo dato simple expresado en forma de variable o de un valor extraído de un conjunto de datos (agregación de variables). Puede ser descriptivo o normativo y expresar una información cuantitativa o cualitativa. Varios indicadores son necesarios para la evaluación de un sistema.

El primer paso en la formulación es **identificar qué se quiere describir**. Para ello hay que retomar los estándares de calidad y los resultados esperados. Una vez que se sabe que se quiere medir, se debe **describir y formular el indicador**. El indicador puede ser uno que ya se utiliza regularmente en el programa o servicio de atención, aunque es mejor que sea creado por el equipo y ser específico al proceso de mejoramiento de la calidad que se está implementando(28).

Ya descrito el indicador, se procede a formular cómo será operacionalizado. Para determinar *la fuente de información* para desarrollar el indicador, dichos indicadores necesitan de datos para ser desarrollados, siendo indispensable identificar la fuente de información de donde se obtendrán los datos.

Si se ha identificado correctamente lo que se va a medir y se ha formulado adecuadamente el indicador, será más fácil identificar la fuente de información que proporcionará los datos para desarrollarlo. A continuación se dan a conocer algunas recomendaciones que se deben tener en cuenta a la hora de formular indicadores de manera general, ya que en dependencia a su tipo, se pueden formular de diferentes formas(28):

- Un solo indicador no puede captar la compleja totalidad de un hecho ni sus múltiples dimensiones. Por ello, se requiere seleccionar cuidadosamente un conjunto de indicadores que expliquen mejor el hecho o fenómeno que se quiere evaluar y comunicar.
- Los indicadores no son rígidos ni invariables, sino más bien referenciales. Estos se podrán ir ajustando o corroborando a la luz de la experiencia práctica. Los indicadores requieren ser puestos a prueba permanentemente.
- Es recomendable e importante contar con el número necesario de indicadores, con el fin de asegurar su utilización y de justificar la inversión para el levantamiento de información.
- Los indicadores deben corresponder a las necesidades de información de los actores. Una vez identificadas las demandas, será importante acordar qué actores participarán en procesos de:
  - Recolección de información;
  - Análisis;
  - Comunicación de la información;
  - Toma de decisiones (uso de la información).

#### **1.3.5 Situación problemática de los indicadores de calidad para software de simulación.**

Teniendo una breve información de lo que es indicadores de calidad de manera general, sólo resta decir, cómo se comportan dichos indicadores para el desarrollo de software de simulación tanto en el mundo como en Cuba, particularmente en la UCI. En la investigación que se ha realizado, no se ha

encontrado mucha información acerca de estos indicadores. En el ámbito internacional se producen softwares de simulación para diferentes sectores de la sociedad como por ejemplo, dentro de la simulación de procesos se encuentra el software OPTIMA! perteneciente a la empresa Micrografx, de EEUU. Para el modelado y simulación de sistemas dinámicos no lineales complejos, se encuentra el software VisSim. En el mundo de la robótica se encuentran PC ROSET para robots Kawasaki, DTSP para robots Panasonic, etc.

En Cuba existe poco conocimiento de indicadores de calidad para el desarrollo del software de simulación, ya que no es un producto muy explotado en el país. Como se mencionaba anteriormente, la simulación se está desarrollando poco a poco en los últimos tiempos en el país y como idea nueva falta mucho por definir.

En la UCI con el surgimiento y exportación de sus primeros softwares y el crecimiento en la actividad, se hizo evidente la necesidad de aplicar normativas de calidad que respalden los productos desarrollados. En estos momentos no se está realizando de manera formal en la universidad un Proceso de Mediciones de Software, por lo que se hace necesario definir un proceso de este tipo, e implementarlo como una disciplina clave dentro del ciclo de vida del desarrollo del software(29). En la universidad existen varios proyectos dedicados a la producción de software de simulación, pero hasta el momento, no existe un proceso de mediciones efectivo que garantice la calidad de su producto final, ya que la UCI no cuenta con un conjunto de indicadores para certificar la eficacia de la calidad del software. Esta medición la realizan a través de fórmulas matemáticas, que no se puede asegurar con esto, un resultado exacto y eficaz.

Según propuesta del Ing. Maypher Román en una conferencia desarrollada en el evento Informática 2007 asegura que cada facultad debe definir su Plan de Medición de PSM (Mediciones Prácticas de Software y Sistemas)(anexo 2) acorde a su perfil productivo, ya que define la medición como un proceso sistemático y flexible aplicado a la Ingeniería de Software y de Sistemas. Permite evaluar sus procesos de desarrollo basados en la experiencia de las mediciones realizadas en proyectos anteriores; además evaluar el estado de todo el proceso productivo de la Universidad, con el propósito de tomar decisiones estratégicas(29). Aún esta propuesta no se ha materializado, siendo una de las razones que motivan la definición de mediciones aplicables al proceso de desarrollo de software de simulación de la facultad nueve, para comenzar a llevar un control de calidad de sus procesos o productos.

## 1.4 Análisis de otras soluciones existentes

### 1.4.1 Modelo Sistemático de Calidad (Mosca)

MOSCA: Integra los modelos de calidad donde el primer modelo enlaza las ideas del Estándar ISO/IEC 9126 para el producto y del Modelo de Dromey. El resultado es un conjunto de atributos que deben estar presentes en un sistema de calidad y los factores que influyen a éstos, los cuales se agrupan en cuatro grupos representados por las cuatro dimensiones del enfoque de Callaos. El segundo modelo desarrollado en LISI, integra el enfoque de calidad sistémica con las características presentes en el modelo de procesos SPICE, adaptado a la realidad venezolana, conformado por una jerarquía de 5 niveles : Ciclos de Vida, Categorías, Procesos, Principios, y Bases Prácticas, que son un conjunto de directrices a ser ejecutadas por la organización para lograr alcanzar un principio. Este modelo garantiza el balance entre la eficiencia y la efectividad del proceso de desarrollo en el software de simulación fundamentalmente, a través de una propuesta equilibrada de prácticas bases. Mosca consta de 4 fases:

**Nivel 0:** Dimensiones. Eficiencia del proceso, Efectividad del proceso, Eficiencia del producto y Efectividad del producto son las cuatro dimensiones propuestas en el prototipo de modelo. Sólo un balance y una buena interrelación entre ellas garantizan la calidad Sistémica global de una organización.

**Nivel 1:** Categorías. Se contemplan 11 categorías: Producto: Funcionalidad (FUN), Fiabilidad (FIA), Usabilidad (USA), Eficiencia (EFI), Mantenibilidad (MAB) y Portabilidad (POR), Proceso: Cliente-Proveedor (CUS), Ingeniería (ENG), Soporte (SUP), Gestión (MAN) y Organizacional (ORG). Esta división no implica un desligamiento entre ellas, simplemente se realiza para identificar a que sector o submodelo pertenecen.

**Nivel 2:** Características. Cada categoría tiene asociado un conjunto de características, las cuales definen las áreas claves a satisfacer para lograr, asegurar y controlar la calidad tanto en el producto como en el proceso. Entre las características asociadas a cada categoría del producto, se proponen en el modelo MOSCA, una serie de características del proceso.

Esto se debe, a que algunas características de la calidad del proceso, impactan directamente en las categorías del producto al igual que ciertas características de la calidad del producto definen categorías del proceso.

**Nivel 3: Métricas.** Para cada característica se propone una serie de métricas utilizadas para medir la calidad sistémica. Dada la cantidad de métricas asociadas a cada una de las características que conforman MOSCA (587 en total), éstas no serán presentadas en el presente artículo.

### **1.5 Conclusiones Parciales**

- La calidad es un aspecto fundamental en el proceso de desarrollo de software, ya que es un conjunto de cualidades que caracterizan y determinan su utilidad y existencia, está dada por lo que el cliente desea.
- Resaltar que las organizaciones de software en general han comprendido que la clave de una entrega exitosa de un producto (en tiempo, en presupuesto, con la expectativa de calidad) radica en una efectiva gestión de su proceso software
- La calidad debe tener un adecuado balanceo de eficiencia, confiabilidad, facilidad de mantenimiento, portabilidad, facilidad de uso, seguridad e integridad.
- Para mostrar los progresos efectivos de un programa o proyecto, los indicadores deben relacionarse directamente con su propósito y resultados; y para que sean aceptados por todos los actores, es conveniente que se formulen en el marco de un proceso participativo.
- Los indicadores, por tanto, deben ser construidos en y por los proyectos; sólo de esta forma se garantizará una correspondencia con los resultados, estrategias y acciones esperados y con el contexto sociocultural en el cual se desenvuelven los proyectos.

**CAPÍTULO 2: “Estudio del flujo de trabajo del polo de simulación de la facultad nueve y resultados”.**

En este capítulo se hace un análisis de la situación actual del polo de simulación de la facultad nueve, donde se abordarán las principales dificultades que se presentan durante el proceso de desarrollo de software de simulación. Además se hace un análisis estadístico de los resultados obtenidos a partir de encuestas realizadas a los diferentes líderes y varios estudiantes de este polo con el objetivo de definir los principales problemas que presentan en cuanto a los indicadores de calidad de software.

**2.1 Características del polo de Simulación**

El polo de simulación enmarcado en la facultad nueve es uno de los polos que a pesar de que es una idea nueva que acaba de surgir se convirtió en un polo novedoso, integral y productivo, en el cual se desarrollan proyectos de simulación de procesos matemáticos y donde se forman sus integrantes con una fuerte preparación y compromiso productivo a través de la combinación del modelo Docencia-Producción-Investigación, guiados por los principios éticos de los revolucionarios cubanos.

A raíz de la elevada preparación que deben tener los integrantes del polo, las principales asignaturas que se imparten son: Simulación de procesos, Arquitectura y componentes de Simuladores, Sistemas numéricos sobre ecuaciones Diferenciales y Elementos de Programación Avanzada.

Sus principales objetivos para la creación de los diferentes simuladores está dado por:

- Desarrollar simuladores de procesos basados en métodos matemáticos.
- Formar estudiantes con un segundo perfil de simulación.
- Promover temas relacionados con el perfil del polo.
- Desarrollar las líneas de investigación del polo.

El polo de simulación tiene dentro de sus principales software a producir los siguientes:

1. Simulador para la industria química cubana versión 4.0.2 (STA 4.0.2).
2. Implementación de una herramienta para la creación de la base de conocimiento por indicadores de un sistema experto para el análisis de resultados.
3. Desarrollo de un Kernel para el desarrollo de nuevos simuladores.

## **2.2 Estrategias del Polo de Simulación**

El polo de simulación presenta como estrategia: disponer de una plataforma informática que permita simular en forma modular, analizar y sintetizar soluciones para los procesos de la industria química cubana, superando sus limitaciones actuales e incrementar su eficiencia.

También presenta como parte de su estrategia elaborar el marco teórico donde se defina la base teórica que sustenta la investigación y quede definida la posición del grupo de investigaciones. Además debe crear un Simulador de la Industria Química Cubana (SPQ) informativamente novedoso, el cual debe disponer de los módulos totalmente validados que permitan simular los procesos siguientes: Azúcar, Alcohol, Refinación de Petróleo, Cemento, Níquel, Cobalto y Cerveza, con la prioridad en el tiempo en que están listados.

El SPQ debe ser lo suficientemente flexible para poder añadir en un futuro proyecto, sin realizar grandes cambios en la programación, nuevos procesos, otras técnicas de optimización, integración y síntesis de procesos.

## **2.3 Proceso de desarrollo del software de simulación del polo de la facultad nueve.**

En el proceso para el desarrollo del software de Simulación del polo de la facultad nueve se definen claramente las actividades fundamentales a desarrollar, de manera que los proyectos puedan ser verificados por los asesores de calidad de la facultad con resultados satisfactorios.

Durante todo el proceso se le confiere gran importancia a las revisiones técnicas de cada una de las actividades desarrolladas por el equipo y que son pactadas con el cliente, cada uno de los pasos es revisado con el mismo para lograr un entendimiento con el equipo de desarrollo de manera tal que las inconformidades puedan detectarse a tiempo y no constituyan un mayor problema a medida que avanza el proyecto.



El proceso consta de dos etapas fundamentales y que evidentemente guardan una estrecha relación, estas son: la concepción y planificación del proyecto, y la ejecución del mismo. Es de destacar que utilizan una fusión de metodologías, combinando RUP con XP con el fin de lograr los objetivos propuestos por el equipo de desarrollo. Esta combinación facilita un mejor entendimiento de los procesos, una mayor identificación de los mismos, y el aprovechamiento de las facilidades de llevar el proceso de desarrollo por un camino iterativo e incremental, que permita verificar en cada iteración lo que se va obteniendo, mientras que la XP se mezcla con el objetivo de que este proceso sea rápido y de una mayor eficiencia al identificar los procesos críticos e ir realizándolos todos, obteniendo respuestas y resultados en cada etapa. En resumen el proceso de desarrollo se guía basándose en lo tradicional creando un híbrido al fusionar dos metodologías, tomando de ambas lo que es conveniente, y adaptándolas a las necesidades propias del proyecto.

Desde un inicio del proyecto no se definió un negocio como tal. En una primera etapa, el cliente requirió de un simulador estacionario, por lo que el proyecto engloba todas las actividades que deben realizarse antes de la firma del contrato, la primera de ellas es pactar las tareas con el cliente. Éste es un punto crucial pues es aquí donde se fijan los momentos de encuentro entre las partes, se comprende lo que desea el cliente y se explica lo que la organización es capaz de hacer. Para esto se realizan reuniones de inicio para obtener el propósito de lo que quiere el cliente.

Como resultado de la entrevista se obtendrá un convenio marco con las próximas tareas a realizar hasta que se firme el contrato. A partir de este momento es necesario definir el Staff a nivel global incluyendo los roles a desarrollar dentro del proyecto, también es necesario realizar un plan de Gestión de Configuración inicial. Se procede a la modelación del Negocio (en caso de ser necesario), en este caso como se mencionó anteriormente se utiliza una fusión de metodologías ya que no se rige estrictamente por la metodología de RUP( más utilizada en la universidad) por lo que pasa directamente al flujo de trabajo de requerimiento donde se efectúan los requisitos funcionales de alto nivel quedando conformado el documento visión que incluye, entre otras cosas, el análisis económico, las actividades que van a automatizarse y los beneficios que reportarán, también debe hacerse un análisis de los riesgos, y evaluar si se continúa o no con el proyecto. En caso de una respuesta acertada se desarrolla el Plan de proyecto a nivel global, el Plan de aseguramiento de la Calidad y el Plan de Mediciones, revisados por el cliente y finalmente si ambas partes están de acuerdo se confecciona el contrato y se firma finalmente.

En esta primera etapa se trabaja en conjunto, el cliente y el equipo de desarrollo, hasta la parte de análisis y diseño que trabajan los desarrolladores solos. En esta etapa de ejecución del proyecto se desarrolla el plan de la versión beta y su plan de pruebas, consultándolos con el cliente, si no existen problemas comienza entonces la captura de requisitos para esta versión, una vez detalladas todas las funcionalidades se procede al análisis, diseño, implementación y pruebas de unidad e integración de la versión, esto debe hacerse iterativamente hasta obtener la última versión del producto. Finalmente se procede a las pruebas de liberación del grupo de calidad y las de aceptación del cliente, finalizando así, el proceso de desarrollo con la entrega del software de simulación.

### **2.4 Principales problemas detectados en el polo de simulación de la facultad nueve.**

En el polo de simulación ubicado en la facultad nueve de la Universidad de las Ciencias Informáticas (UCI) se realizó un estudio riguroso a través de un conjunto de encuestas, entrevistas a líderes y estudiantes del polo donde se definieron como los principales sus problemas existentes los que se exponen a continuación:

1. Desmotivación laboral de los miembros del polo productivo
2. No se tiene descrito claramente el proceso de producción de software en su totalidad.
3. No se le da un enfoque a proceso a la organización en su totalidad.
4. Presentan resistencia al cambio.
5. No presentan un equipo de desarrollo completo
6. No se conocen las principales normas, modelos, estándares de producción y calidad de SW.
7. No se mide la calidad de software con la exigencia requerida para software de estos tipos.
8. No presentan ninguna variable de control
9. No presentan ni conocen ningún indicador de calidad de software que se pueda aplicar en el polo.
10. No presentan un equipo de aseguramiento de la calidad para dicho software.
11. No se le hace prueba de calidad a dicho software en su etapa de desarrollo.

12. Desaprovechamiento de la capacidad instalada en los laboratorios destinados a la producción.

13. Deficiencias tecnológicas existentes.

Todos los problemas detectados traen consigo atrasos a la hora de la entrega del producto, el incumplimiento de los cronogramas establecidos, por lo que el software(o producto) no saldrá con la calidad requerida y puede no satisfacer las necesidades del cliente.

Entre los problemas identificados uno de los que más atenta en contra del buen funcionamiento del polo es el desconocimiento de indicadores de calidad de software, solucionar este problema sería muy ventajoso pues los indicadores son un conjunto de métricas que proporcionan una visión profunda del proceso del software que ayudará a la toma de decisiones más fundamentadas. Estos indicadores permiten evaluar el estado del proyecto en curso, seguir la pista de los riesgos potenciales, detectar las áreas de problemas antes de que se conviertan en críticas, ajustan el flujo y las tareas del trabajo y permiten evaluar la habilidad del equipo del proyecto en controlar la calidad de los productos software. Por tales razones se puede decir que dándole solución al problema de los indicadores, se resuelven los problemas restantes.

### **2.5 Resultados de la encuesta realizada a líderes y estudiantes del polo de simulación de la facultad nueve.**

Teniendo en cuenta el análisis realizado se calculó el tamaño de la muestra (10) .Continuamente se pasó a seleccionar a los 10 integrantes del polo de simulación, para esto se tomaron en cuenta varios criterios o características con vista a que exista una representatividad de los mejores razonamientos con vista a una alta precisión en los resultados, dichos criterios son:

1. Integrantes con más experiencia de trabajo en el polo de simulación
2. Integrantes con básicos conocimientos de aseguramiento de la calidad del SW
3. Integrantes con básicos conocimientos de ISW
4. Integrantes con conocimientos básicos de indicadores de calidad.

Cumpliendo dichas características y según el tamaño de la muestra calculado (10) son escogidos a 4 líderes del proyecto, y 6 estudiantes pertenecientes al grupo de calidad del polo de simulación debido a que cumplen con las características antes mencionadas.

A continuación se muestran los puntos más significativos (Ver anexo 4) que demuestran la verdadera situación de los proyectos de simulación del polo en cuanto a los indicadores de calidad de software. Para este análisis se hace uso de los casos favorables a la proporción, es decir, los resultados positivos obtenidos en las encuestas realizadas y el tamaño de la muestra (10). Para este cálculo se recurrió a la siguiente ecuación:

$$\text{Proporción (P)} = \text{Casos Favorables (X)} / \text{tamaño de la muestra (n)}$$

Los estudios realizados en el polo de simulación de la facultad nueve, muestran sobre el nivel de conocimiento y aplicación de indicadores de calidad de software, la situación en que se encuentran los mismos y la necesidad de aplicar los indicadores de calidad necesarios para mejorar la calidad del proceso productivo. De manera, los resultados de las encuestas (Anexo 3) realizadas muestran los siguientes datos:

- No cuentan hasta el momento con ningún grupo de aseguramiento de calidad, y no tienen bien definidos los objetivos de calidad del proyecto.
- No tienen conocimiento de los diferentes indicadores de calidad que pueden aplicar para evaluar el software de simulación desarrollado

Los resultados se muestran en la siguiente gráfica de barra, lo que demuestra la existencia de una situación poco favorable en el polo de simulación de la facultad nueve referente al tema de calidad, debido a los datos esbozados anteriormente.

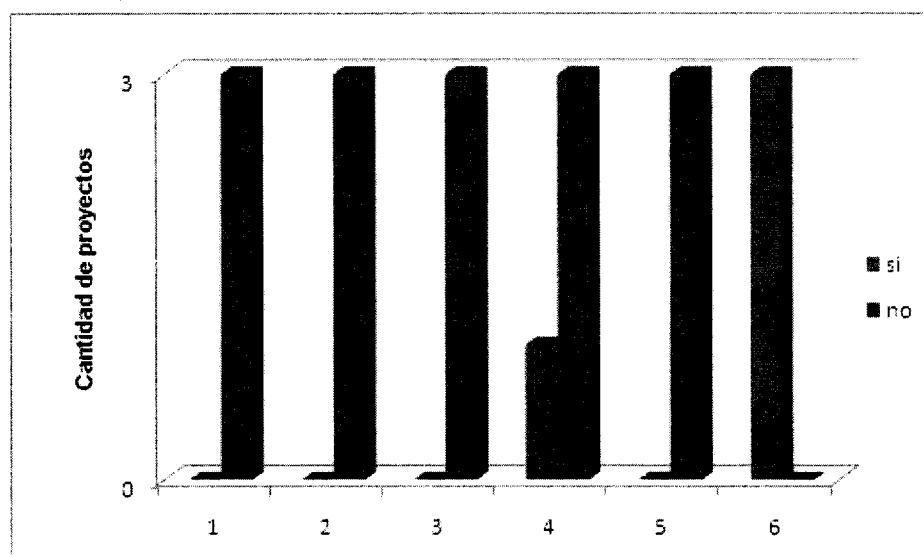


Figura 9: Gráfica de los resultados de la encuesta realizada al polo de simulación de la facultad nueve.

A continuación se muestra el porcentaje que representa cada una de las respuestas a las preguntas desarrolladas en la encuesta realizada en el polo de simulación con sus respectivas gráficas, para un mejor entendimiento:

**Pregunta 17:** ¿Conoce algunos Indicadores de calidad de software?

$$P = X/n = 0/10 = 0$$

La respuesta dada representa un 100 % negativo, es decir, que el polo no tiene conocimiento alguno de los indicadores de calidad existente en el mundo

**Pregunta 18:** ¿Aplican indicadores de calidad de software durante el proceso de desarrollo de software de simulación?

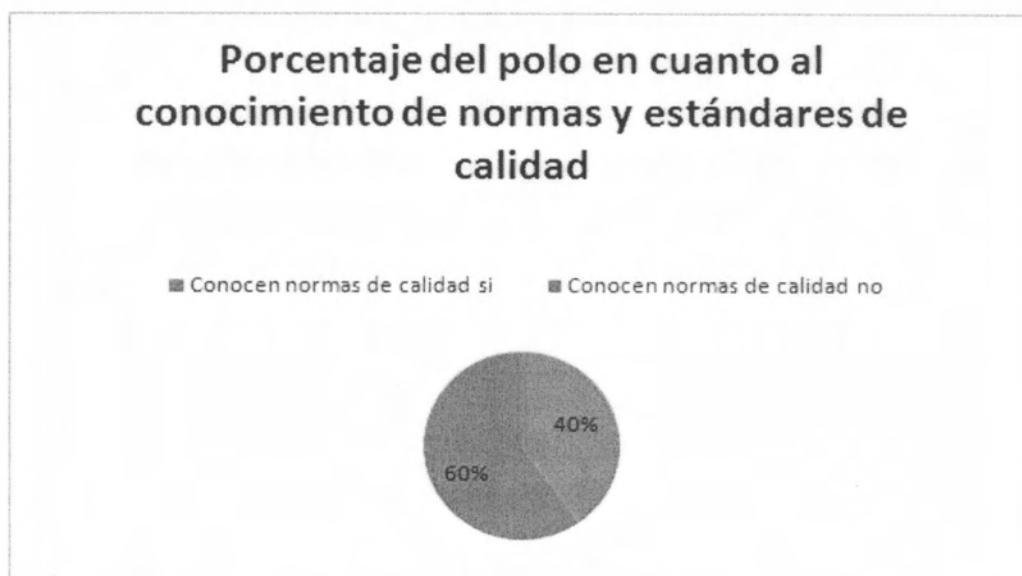
$$P = X/n = 0/10 = 0$$

Esta pregunta tiene mucha relación con la anterior, por lo tanto, al no tener conocimiento de los indicadores de calidad, no aplican ninguno indicador de calidad al software que están desarrollando, por lo que los resultados demuestran un 100% negativo.

**Pregunta 21:** ¿Conoce algún modelo, normas o estándares de calidad de software?

$$P = X/n = 4/10 = 0.4$$

El equipo de desarrollo del polo de simulación, tiene muy poco conocimiento de las normas y modelos de calidad, pues como idea nueva, necesitan obtener experiencia con respecto al tema de aseguramiento y garantía de la calidad del producto final. El análisis de la proporción, brinda como resultado que solo el 40% del equipo de desarrollo conoce de los modelos, normas o estándares de calidad, mientras que el 60% no.



**Pregunta 22:** ¿Aplican algunas normas, modelos o estándares de calidad?

$$P=X/n=0/10=0$$

Hasta el momento, no aplican ninguna norma o estándar de calidad, evalúan la calidad del proceso a través de pruebas al software, que no son suficiente para garantizar la calidad del software, por tanto la respuesta dada representa un 100% negativo.

**Pregunta 24:** ¿Conoce los indicadores de calidad de software que pueden aplicar al software en producción?

$$P=X/n=0/10=0$$

Al no tener el mínimo conocimiento de los indicadores de calidad, menos pueden conocer los indicadores que pueden aplicarle al software de simulación en particular, por lo tanto, el resultado, representa un 100% negativo.

**Pregunta 26:** ¿Presentan algún documento histórico?

$$P=X/n=10/10=1$$

El polo de simulación presenta un documento histórico, donde tiene registrado todas las actividades y tareas realizadas por el equipo durante el proceso de desarrollo del software. En la grafica se muestra un 100% positivo de la respuesta dada.

En estos momentos no se está realizando de manera formal en la Universidad un Proceso de Mediciones de Software; por lo que se hace necesario definir un proceso de este tipo e implementarlo como una disciplina clave dentro del ciclo de vida del desarrollo del software, para obtener una buena calidad del producto final que es el objetivo principal de esta industria del software, la UCI.

Según los resultados arrojados en las encuestas se puede concluir que el polo de simulación de la facultad nueve en su minoría (25%) tiene conocimiento de los modelos, normas y estándares de calidad y que a pesar de dichos conocimientos, estos no se tienen en cuenta durante el proceso de desarrollo del software de simulación. También se puede observar que durante el proceso no se aplican indicadores de calidad debido al desconocimiento de ellos en un 100% y no saber cual indicador debe aplicársele a dicho software.

### **2.6 Conclusiones Parciales**

El desarrollo del capítulo presentado ha permitido:

- Que la comparación establecida y el diagnóstico realizado dieran a conocer todas las características esenciales del polo de simulación.
- Las encuestas y entrevistas realizadas a líderes y estudiantes de los diferentes proyectos de simulación, arrojaron un significativo número de deficiencias que atentan contra el correcto desarrollo del proceso de software de simulación.
- Existen graves problemas en éste con el tema de la garantía y aseguramiento de la calidad del software.
- Los indicadores de calidad de software que serán propuestos en el capítulo 3 estarán encaminados a detectar los principales errores que van desde la omisión de algún requisito, en las etapas tempranas del desarrollo del proyecto, hasta la detección de errores en el código de algún producto de software.

### **CAPÍTULO 3: “Propuestas de Indicadores de Calidad”**

En este capítulo se propone un conjunto de indicadores de calidad, donde se darán a conocer todas las características de cada uno de ellos, cómo se calculan y se formulan.

#### **3.1 Propuesta de los indicadores de calidad para software de simulación.**

Como se menciona en el capítulo 1 existen varios indicadores de calidad, que en resumen, son datos que se marcan, medibles para conocer si se alcanzan las metas trazadas, por tanto los indicadores son metas para satisfacer a los clientes, metas que son trazadas por cada empresa o equipo de desarrollo respectivamente y que nadie ajeno a la empresa lo debe saber. A continuación se muestra la descripción detallada de un conjunto de indicadores, formulados a partir de la base del estudio preliminar en el polo de simulación de la facultad nueve, destacando las principales categorías de la calidad, presentes en software de simulación, que las mismas son: fiabilidad, eficiencia y mantenibilidad.

##### **3.1.1 Indicadores para evaluar la fiabilidad:**

1. **El tiempo medio entre fallos:** consiste en detectar una tasa de fallos, es decir, el tiempo medio que está entre el instante de tiempo en el que el sistema o un integrante del equipo de desarrollo detecta un fallo en el software y el tiempo que demoran en repararlo, este indicador se utilizará fundamentalmente en la etapa de diseño. Con esto se puede medir el impacto sobre la fiabilidad del software de que sea escaso o no.

Para evaluar este indicador se utiliza la siguiente ecuación:

$$TMEF = TMDF + TMDR$$

Donde:

TMDF: Tiempo Medio de Fallo.

TMDR: Tiempo Medio de Reparación.

##### **Análisis del procedimiento:**

El objetivo del indicador es lograr disminuir el tiempo que el equipo de desarrollo emplea para reparar el fallo detectado, mientras menor sea el tiempo empleado, más eficaz será el producto y el impacto de la fiabilidad no será escaso. Para esto se establece un rango de comparación que va a depender de la



magnitud del fallo encontrado, si es detectado un error sencillo para el tiempo sea bueno, deben demorarse como máximo 2 días, si por el contrario, el error es complejo como máximo 7 días.

2. **Proporción de la eliminación de defectos (P):** Es la capacidad para medir el rendimiento que tiene la eliminación de defectos durante cualquier etapa del proceso de desarrollo de software. En muchas ocasiones se introducen defectos, que son fácilmente identificados y eliminados. Se considera defectos evadidos a todos los defectos que se insertaron antes o durante una etapa determinada, pero que no fueron encontrados en esa etapa, sino en otra posterior, estos traen consigo una costosa inversión de tiempo y esfuerzos para su eliminación.

Para evaluar este indicador, se consulta la métrica de rendimiento de la eliminación de defectos(30) que se basa en la relación entre los defectos eliminados y todos los existentes en una etapa determinada. Este indicador se calcula con la siguiente ecuación:

$$P = (DE * 100) / (DE + DV)$$

Donde:

DE: Defectos Eliminados

DV: Defectos Evadidos

#### **Análisis del procedimiento:**

El objetivo o meta es lograr un rendimiento de 100%, o sea que todos los defectos sean encontrados y eliminados en la etapa que se esté analizando y se reduzcan los defectos evadidos. Es importante tener identificados los posibles defectos que ocurran frecuentemente en una etapa o fase determinada dentro del proceso de desarrollo, de manera que se pueda evitar o solucionar a tiempo. Suponiendo que se logre un rendimiento entre el 95% y 100% se puede considerar que el equipo de desarrollo esta en buen estado, de lo contrario no cumple con están cumpliendo con su trabajo en la eliminación de errores y aumentaría los defectos evadidos, por lo tanto estarían en mal estado.

3. **Madurez de las pruebas (MP):** es un indicador que se desarrolla en el proceso de pruebas, no solo se preocupa de la completitud de los casos de prueba según los definidos para cumplir los requisitos, sino que también se interesa por cuales han obtenido resultados satisfactorios, para ello es necesario llevar un control de los casos de prueba que arrojaron resultados

satisfactorios y el total de los casos de prueba definidos para el cumplimiento de los requisitos(30).

La madurez de las pruebas está dada por la siguiente ecuación:

$$MP = CPS / CPR$$

Donde:

CPS: Número de casos de pruebas que han obtenido un resultado satisfactorio.

CPR: Número de casos de pruebas a ejecutar requeridos para cubrir los requisitos.

#### **Análisis del procedimiento:**

Mientras mayor sea el número de casos de pruebas que han obtenido un resultado satisfactorio de los casos realmente ejecutados y que se requieren para cubrir los requisitos, mayor será la madurez alcanzada por el equipo de probadores del software, lo ideal es un valor cercano a 1. Cuando el valor se encuentre entre 0.97 y 1 será mayor la madurez alcanza, cuando se encuentre entre 0.50 y 0.96 tendrá una trivial madurez, de lo contrario si el valor es menor que 0.50 la madurez será muy mínima, ya que se han obtenidos muy poco casos de pruebas con resultados satisfactorios del total de casos de pruebas diseñados.

#### **3.1.2 Indicadores para evaluar la Eficiencia en el Proceso Productivo y del Producto**

##### **Software:**

1. **El porcentaje de la eficiencia en la eliminación de errores en un tiempo determinado (PE):** es un indicador que permite evaluar la eficiencia del equipo de desarrollo, tanto individual como general en la eliminación de lo problemas y errores detectados en el simulador.

Para poder evaluar este indicador, las autoras se apoyan de la métrica de la Eficiencia en la Eliminación de Problemas de donde se puede decir que el indicador se calcula de la siguiente forma:

$$PE = (\text{total (DD)} / \text{total (DE)}) * 100$$

Donde:

DD: número de *defectos detectados* en el proceso de desarrollo del simulador.

DE: número de *defectos eliminados* en el proceso de desarrollo del simulador.

**Análisis del procedimiento:**

Un valor de eficiencia alto es mucho mejor que un valor bajo, el cual es un resultado pobre. El objetivo o meta es una eficiencia de 100%, o sea, que se traten de eliminar los defectos o errores encontrados en el menor tiempo posible. El rango establecido es, en dependencia del número de errores detectados, se suponen que encuentran 3 fallos en un día, para que sea eficiente el equipo de desarrollo, deben demorarse como máximo 2 días, de lo contrario esto significa que el equipo de desarrollo no es muy bueno en la eliminación de errores.

2. **Número máximo de defectos eliminados por unidad de tiempo (MDET):** se propone para tener un control de la efectividad del tiempo dedicado a resolver los defectos encontrados durante cada una de las etapas del proceso de desarrollo de software, para ser corregidos en el menor tiempo posible.

Para la evaluación de este indicador, donde se recurrió a la métrica de Defectos eliminados por unidad de tiempo(30), se utilizó la siguiente ecuación:

$$MDET = n * DT / TiD$$

Donde:

TiD: Tiempo invertido en la eliminación de defectos en una etapa, dado en la unidad de tiempo n.

DT: Defectos Totales encontrados en una etapa de desarrollo.

n: Unidad de tiempo.

**Análisis del procedimiento:**

Los defectos por unidad de tiempo indican la efectividad del tiempo dedicado a las revisiones así como la eficiencia relativa de los múltiples métodos de eliminación de defectos que se estén usando. Si los defectos encontrados por unidad de tiempo han disminuido y el rendimiento no se ha incrementado, puede significar que se está dedicando mucho tiempo a revisiones improductivas. En la medida que el rendimiento se incrementa, una disminución de los defectos por unidad de tiempo es natural. Para que el tiempo sea efectivo deben eliminar de 10 a 12 errores encontrado en 2 días como máximo, de lo contrario el tiempo estaría en mal estado.

3. **Efectividad en el proceso productivo (EPP):** Este indicador centra su atención, como bien su nombre lo indica, en la efectividad del proceso de desarrollo. Que se puede calcular de la siguiente manera:

$$EPP = PD/PP$$

Donde

PD: número de productos software defectuosos

PP: número total de productos software producidos

**Análisis del procedimiento:**

Este indicador establece el rendimiento de los productos defectuosos sobre el total de la producción, esto se puede aplicar en un período de tiempo específico o una orden específica. Para esto se establece un criterio de comparación, entre los valores de 5 a 15 productos defectuosos, la producción está bien, entre 16 a 30 la producción es regular, mayor que esta cifra el rendimiento de la producción estaría muy mal.

**3.1.3 Indicadores para evaluar la mantenibilidad del software.**

1. **Indicador de Madurez de software (IMS):** Este indicador proporciona una indicación de la estabilidad del producto, basado en los cambios que ocurren en cada versión del producto, es decir, permite evaluar el rendimiento del producto software. (6)

Se determina de la siguiente forma:

$$IMS = [Mt - (Fa + Fc + Fd)] / Mt$$

Donde:

Mt: Número de módulos en la versión actual.

Fa: Número de módulos en la versión actual que se han añadido.

Fc: Número de módulos en la versión actual que se han cambiado

Fd: Número de módulos de la versión anterior que se han borrado en la versión actual.

**Análisis del procedimiento:**

A medida que el IMS se aproxima a 1,0 el producto se empieza a estabilizar. Para esto se establece que cuando el valor se encuentre entre 0.97 y 1 ya se considera que el producto está estabilizado, cuando se encuentre entre 0.50 y 0.96 está muy cerca de llegar a la estabilización, de lo contrario si el valor es menor que 0.50 la estabilización es mínima, lo que significa que le queda mucho por trabajar en las versiones de productos hasta llegar la última versión.

2. **Indicador de la disponibilidad (D):** permite determinar la probabilidad de que un programa funcione de acuerdo con los requisitos en un momento dado. Este indicador se determina con la siguiente ecuación:

$$D = \text{TMDf} / (\text{TMDf} + \text{TMDR}) \times 100$$

Donde:

TMDf: Tiempo medio de fallos

TMDR: Tiempo medio de reparación

**Análisis del procedimiento:**

La medida de disponibilidad es algo más sensible al TMDR ya que es una medida indirecta de la facilidad de mantenimiento del software. En dependencia de los requisitos establecidos, si el valor está entre 0.97 y 1 se considera que el programa puede funcionar, cuando se encuentre entre 0.50 y 0.96 está muy cerca de llegar a funcionar correctamente, de lo contrario si el valor es menor que 0.50 la probabilidad de que el simulador funciones es muy remota.

**3.2 Conclusiones Parciales**

- La definición de un conjunto de Indicadores dividido en varios grupos y su definición formal, facilitó el desarrollo de la investigación y representa una modesta contribución a la mejor asimilación de los indicadores por parte de desarrolladores y líderes del polo de simulación.
- Permitió establecer una relación clara de los indicadores con cada uno de los procesos y categorías dentro de la Gestión de la Calidad
- La aplicación de los nuevos indicadores que fueron planteados como solución del trabajo de diploma en dicho proyecto, tiene como objetivo lograr la uniformidad en la filosofía de trabajo,

garantizando con esto la calidad requerida en el proceso de desarrollo del software de simulación.

### **Conclusiones**

El desarrollo de esta investigación ha permitido llegar a las siguientes conclusiones donde se evidencia el cumplimiento de los objetivos propuestos:

- La Calidad de software es un factor determinante en el proceso de desarrollo del software de simulación.
- Los indicadores de calidad son herramientas de gran importancia que contribuyen al control y mejoramiento de la calidad durante el proceso de desarrollo del software de simulación
- Existe muy poco conocimiento tanto en Cuba como en la UCI acerca de los indicadores de calidad para evaluar los software de simulación
- Se puede concluir que tanto el objetivo general como los específicos del presente trabajo de diploma fueron cumplidos por los autores de la investigación.

### Recomendaciones

Aún cuando el trabajo desarrollado cumple con los objetivos trazados, los autores del mismo consideran que nunca se ha hecho todo lo necesario y que mucho queda por hacer en aras de mejorar la calidad del software de simulación. Por tal motivo se recomienda lo siguiente:

- Profundizar en la investigación del proceso de desarrollo del software de simulación en la universidad de las ciencias Informáticas.
- Continuar trabajando en el desarrollo de nuevos indicadores para el control de la calidad en software de simulación.
- Aplicar los indicadores propuestos al polo de simulación de la facultad nueva y extenderlo a todos los proyectos con características similares en la UCI.
- Describir y estandarizar cada uno de los procesos que se desarrollan en el Polo productivo de Simulación, para facilitar un mejor funcionamiento del mismo.
- Dar un enfoque a proceso a la organización del Polo productivo de Simulación.
- Completar el equipo de desarrollo en su totalidad que garantice el control y la aplicación de los indicadores de calidad propuestos.
- Realizar un estudio de las principales normas, modelos, estándares de producción y calidad de SW que puedan ser aplicados en el Polo dentro de los lineamientos establecidos en la UCI.



**Referencias Bibliográficas**

1. AREVALOS, C. G. *La calidad según Edwards Deming* Disponible en:  
<http://secretosenred.com/articles/6064/1/CALIDAD-SEGUN-EDWARDS-DEMING/Paacutegina1.html>.
2. ÁLVAREZ, R. S. *Aseguramiento de calidad de software* Última actualización: 2008. Disponible en:  
[http://weblogs.udp.cl/nicolas.boettcher/archivos/\(9448\)Aseguramiento\\_de\\_calidad\\_del\\_software\\_1a.ppt](http://weblogs.udp.cl/nicolas.boettcher/archivos/(9448)Aseguramiento_de_calidad_del_software_1a.ppt).
3. INFOCALIDAD. *Infocalidad* Disponible en: [http://www.infocalidad.net/gest\\_calidad\\_def/definicion.asp](http://www.infocalidad.net/gest_calidad_def/definicion.asp).
4. PRESSMAN, R. S. *Ingeniería del Software: Un enfoque práctico*. McGrawHill, 1998.
5. SOFTWARE, I. D. *Ingeniería de Software* Disponible en:  
<http://www.monografias.com/trabajos15/ingenieria-software/ingenieria-software.shtml>.
6. PRESSMAN, R. S. *Ingeniería del Software: Un enfoque práctico*. 5ta ed. McGrawHill, 2005. vol. 2,
7. FUGETTA, A. *The Future of Software Engineering 2000*. publicado el: 2007 de 2000, última actualización: 2007. 25 - 34 p.
8. SEI, A. C., BROWNSWORD LISA, BENTLEY DAVID, BONO THOMAS, MORRIS EDWIN, AND PRUITT DEBORAH *"Evolutionary Process for Integrating COTS-Based Systems (EPIC) Building, Fielding, and Supporting Commercial-off-the-Shelf (COTS) Based Solutions*. publicado el: 2008 de 2002, última actualización: 2008.
9. CURTIS, B., KELLNER, M. I. AND OVER, J. *Process Modeling. Communications of the ACM*. 1992. vol. 31,
10. JACOBSON, L., BOOCH, G., Y RUMBAUGH, J., *Unified Software Development Process*. Addison-Wesley, 1999.
11. GROSSO, I. L. A. Taller de calidad marzo 2006, n°
12. ANTONIO, A. D. Gestión, Control y Garantía de la Calidad del Software. 2008 2007, n°
13. FIGUEROA, M. A. A. Calidad en la Industria del Software. La Norma ISO-9126. 2006, n° [Consultado el: 2008].
14. VARGAS. *Gestión de Procesos de Software Introducción a CMMI y SCAMPI*. 2007,

15. MAQUINADOS STOP, S. A. Modelo del Sistema de Gestión de la Calidad ISO 9001: 2000. 2003, nº
16. PAULK, M. *Comparing ISO 9001 and the Capability Maturity Model for Software*, " Software Quality J, 1993.
17. JACOBSON, I. Applying UML in the Unified Process. Presentación. Rational Software. 1998, nº  
Disponible en: <http://www.rational.com/uml>.
18. SOMMERVILLE, I. Ingeniería de Software. 2002, nº
19. LIETER, P. Proyecto docente e investigador. 2003, nº
20. LLAVORI, R. B. Apuntes de Simulación Informáticas. 2006, nº
21. RAFAEL, B. L. Apuntes de Simulación Informática. 2006, nº
22. JACK. *Metodología RUP y XP. Procesos de desarrollo*. 2007, Disponible en:  
<http://jackopc.blogspot.com/2007/05/metodologias-rup-y-xp-procesos-de.html>.
23. SANCHEZ, M. A. M. *Metodologías del desarrollo de software*. publicado el: 2008 de 2004, última actualización: 2008. Disponible en:  
<http://www.willydev.net/InsiteCreation/v1.0/Descargas/cualmetodologia.pdf>.
24. CUELLAR DÍAZ, L. T. G., HELIODORO. *Los Indicadores de Gestión como herramienta de administración en la pequeña y mediana empresa*. publicado el: 2008 de 2008, última actualización: 2008. Disponible en: <http://www.tablero-decomando.com>.
25. PÉREZ, A. R. M. *¿Que son los indicadores?* publicado el: 2007 de 2002, última actualización: 2007.  
Disponible en:  
<http://www.inegi.gob.mx/inegi/contenidos/espanol/prensa/contenidos/articulos/economicas/indicadores.pdf>.
26. MIRA, J. J. *Criterio, Indicador y Estándar*. publicado el: 2008 de 2008, última actualización: 2008.
27. BAYLES, F. *Cómo determinar los KPI correctos para su empresa*. publicado el: 2008 de 2006, última actualización: 2008. Disponible en:  
<http://www.microsoft.com/spain/medianaempresa/businessvalue/businesskpis.msp>.
28. *Estándares e Indicadores de Calidad*. 2006,

29. DURÁN, M. R. *Mediciones Prácticas de Software y Sistemas (PSM): Una Propuesta para la producción de software en la UCI*. publicado el: 2008 de 2007, última actualización: 2008.
30. LA TORRE HERNÁNDEZ, L. C. N., MARIELA. *Propuesta de Métricas para perfeccionar la gestión de la Calidad en los Procesos de Desarrollo de software*

Investigativa, Universidad de las Ciencias Informática, 2007.

**Bibliografía:**

1. Estándares e Indicadores de Calidad. 2006,
2. ACUÑA, E. A. Calidad del proceso. 2001, [Consultado el: 2008].
3. ALFARO, W. V. simulación de proceso de 2008]. Disponible en:  
<http://www.monografias.com/trabajos6/sipro/sipro.shtml>.
4. ÁLVAREZ, R. S. Aseguramiento de calidad de software Última actualización: 2008. Disponible en:  
[http://weblogs.udp.cl/nicolas.boettcher/archivos/\(9448\)Aseguramiento\\_de\\_calidad\\_del\\_software\\_la.ppt](http://weblogs.udp.cl/nicolas.boettcher/archivos/(9448)Aseguramiento_de_calidad_del_software_la.ppt).
5. ANTONIO, A. D. Gestión, Control y Garantía de la Calidad del Software. 2008 2007, nº
6. AREVALOS, C. G. La calidad según Edwards Deming Disponible en:  
<http://secretosenred.com/articulos/6064/1/CALIDAD-SEGUN-EDWARDS-DEMING/Paacutegina1.html>.
7. BAYLES, F. Cómo determinar los KPI correctos para su empresa. Publicado el: 2008 de 2006, última actualización: 2008. Disponible en:  
<http://www.microsoft.com/spain/medianaempresa/businessvalue/businesskpis.msp>.
8. BLAYA, I. Gestión por procesos. Publicado el: 2008 de 2006, última actualización: 2008. Disponible en: [http://www.upm.es/innovacion/calidad/documentos/Gestion\\_Procesos.pp](http://www.upm.es/innovacion/calidad/documentos/Gestion_Procesos.pp)
9. CRISTANCHO, M. J. A. EVALUACIÓN DE LA CALIDAD DEL SOFTWARE EDUCATIVO BAJO EL ESTÁNDAR ISO 9126. 2007, nº
10. CUELLAR DÍAZ, L. T. G., HELIODORO. Los Indicadores de Gestión como herramienta de administración en la pequeña y mediana empresa. Publicado el: 2008 de 2008, última actualización: 2008. Disponible en: <http://www.tablero-decomando.com>
11. CURTIS, B., KELLNER, M. I. AND OVER, J. Process Modeling. Communications of the ACM. 1992. vol. 31
12. DE LA CRUZ, J. Foro "Indicadores de Calidad de Software de Simulación". 2008, nº
13. DURÁN, M. R. Mediciones Prácticas de Software y Sistemas (PSM): Una Propuesta para la producción de software en la UCI. Publicado el: 2008 de 2007, última actualización: 2008.

14. ENGINEERS (IEEE), I. O. E. A. E. calidad de software. Publicado el: 2008 de 1990, última actualización: 2008.
15. FIGUEROA, M. A. A. Calidad en la Industria del Software. La Norma ISO-9126. 2006, nº [Consultado el: 2008].
16. FUGETTA, A. The Future of Software Engineering 2000. Publicado el: 2007 de 2000, última actualización: 2007. 25 - 34 p.
17. GROSSO, I. L. A. Taller de calidad marzo 2006, nº
18. INFOCALIDAD. Infocalidad Disponible en:  
[http://www.infocalidad.net/gest\\_calidad\\_def/definicion.asp](http://www.infocalidad.net/gest_calidad_def/definicion.asp).
19. INTERPAÍSES, P. C. Medición de la Calidad: Estándares e indicadores. Publicado el: 2008 de 2008, última actualización: 2008. Disponible en:  
<http://www.qaproject.org/pubs/PDFs/LACHSR/Anexos/Anexo%202.Manual%20de%20Capacitaci%C3%B3n.Unidad%20Dos.pdf>
20. JACOBSON, I. Applying UML in the Unified Process. Presentación. Rational Software. 1998, nº Disponible en: <http://www.rational.com/uml>
21. JACOBSON, L., BOOCH, G., Y RUMBAUGH, J. (1999). Unified Software Development Process. Addison-Wesley
22. LA TORRE HERNÁNDEZ, L. C. N., MARIELA. Propuesta de métricas para perfeccionar la gestión de la calidad en los procesos de desarrollo de software. Investigativa, Universidad de las Ciencias Informática, 2007.
23. LIC. ANGELA MÉRIDA MINGARRO, L. M. H. V. Validación de un sistema de indicadores para medir el desempeño en la empresa de materiales de la construcción de Holguín. Publicado el: 2008 de 2006, última actualización: 2008. Disponible en:  
<http://www.monografias.com/trabajos15/valoracion/valoracion.shtml>.
24. LIETER, P. Proyecto docente e investigador. 2003, nº
25. LOVELLE, M. C. calidad de software de 2007]. Disponible en:  
[http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF).

26. MAQUINADOS STOP, S. A. Modelo del Sistema de Gestión de la Calidad ISO 9001: 2000. 2003, nº
27. MIRA, J. J. Criterio, Indicador y Estándar. Publicado el: 2008 de 2008, última actualización: 2008.
28. PACHECO, J. C. C., WIDBERTO. CAICEDO, CARLOS HERNÁN. . Indicadores Integrales de Gestión. 1ra ed. Bogota: 2002. ISBN 9584102060.
29. PAULK, M. Comparing ISO 9001 and the Capability Maturity Model for Software," Software Quality J, 1993.
30. PÉREZ, A. R. M. ¿Que son los indicadores? publicado el: 2007 de 2002, última actualización: 2007. Disponible en:  
<http://www.inegi.gob.mx/inegi/contenidos/espanol/prensa/contenidos/articulos/economicas/indicadores.pdf>
31. PRESSMAN, R. S. Ingeniería del Software: Un enfoque práctico. McGrawHill, 1998.
32. PRESSMAN, R. S. Ingeniería del Software: Un enfoque práctico. 5ta ed. McGrawHill, 2005. vol. 2
33. RAFAEL, B. L. Apuntes de Simulación Informática. 2006, nº
34. ROTONDO, E. V., GLORIA. Indicadores de Género. Perú: 2004, [Consultado el: 2008]. ISBN 9972-9373-1-3.
35. SEI, A. C., BROWNSWORD LISA, BENTLEY DAVID, BONO THOMAS, MORRIS EDWIN, AND PRUITT DEBORAH "Evolutionary Process for Integrating COTS-Based Systems (EPIC) Building, Fielding, and Supporting Commercial-off-the-Shelf (COTS) Based Solutions. publicado el: 2008 de 2002, última actualización: 2008.
36. SOFTWARE, I. D. Ingeniería de Software Disponible en:  
<http://www.monografias.com/trabajos15/ingenieria-software/ingenieria-software.shtml>
37. SOMMERVILLE, I. Ingeniería de Software. 2002, nº
38. UNESCO, W. Informe mundial para la revalorización de los recursos de agua. Publicado el: 2008 de 2003, última actualización: 2008. Disponible en:  
<http://www.watermonitoringalliance.net/index.php?id=782&L=2>

39. VARGAS. Gestión de Procesos de Software Introducción a CMMI y SCAMPI. 2007,
40. Software, I. d. (2005). "Ingeniería de Software." Disponible en:  
<http://www.monografias.com/trabajos15/ingenieria-software/ingenieria-software.shtml>.
41. INFORMACIÓN, C. T. Y. D. N. N. C. D. T. D. L. "Norma Cubana, ISO/IEC 9126-1:2001, Ingeniería de Software, Calidad del Producto", Oficina Nacional de Normalización, 2005.
42. HERNÁNDEZ LEÓN, R. A. C. G., SAYDA "EL Paradigma Cuantitativo de la Investigación Científica". Ciudad de la Habana: 2002. ISBN 959-16-0343-6.
43. GONZÁLEZ, R. A. H. L. S. C. El paradigma cuantitativo de la investigación científica. 2002,
44. LA TORRE HERNÁNDEZ, L. C. N., MARIELA. Propuesta de Métricas para perfeccionar la gestión de la Calidad en los Procesos de Desarrollo de software. Investigativa, Universidad de las Ciencias Informática, 2007.

## Anexos

### Anexo 1: Niveles de CMMI.

- Nivel 1. Inicial o a medida.
  - Basado en la competencia y acciones individuales de las personas.
- Nivel 2. Gestionado (gestión básica de procesos)
  - Gestión de acuerdos con los proveedores de productos y servicios.
  - Selección y supervisión de los proveedores.
  - Medición y análisis.
  - Aseguramiento de la calidad del producto y del proceso.
  - Desarrollo de los requisitos del: cliente, producto y componente del producto.
  - Diseño, desarrollo y puesta en práctica de soluciones técnicas.
  - Asegurar la integración del producto.
  - Verificación.
  - Validación Enfoque a la organización hacia la gestión de los procesos.
- Nivel 3. Definido (estandarización de procesos)
  - Correcta definición de los procesos de la organización.
  - Educación y entrenamiento para mejorar la eficacia y la eficiencia.
  - Gestión integrada de los proyectos (proceso + productos) Gestión de riesgos.
  - Análisis sistemático y puesta en práctica de las decisiones acordadas.
  - Ambiente organizativo adecuado para el desarrollo integrado del producto y del proceso.
  - Formar y mantener un equipo para el desarrollo integrado.
  - Gestión integrada de proveedores.



- Nivel 4 Gestionado de forma cuantitativa
  - Evaluación de los procesos de la organización (datos del rendimiento de los procesos).
  - Gestión cuantitativa de los proyectos.
  - Gestión cuantitativa de los proveedores
  - Innovación y despliegue a lo largo de toda la organización (mejoras incrementales y su posterior generalización).
  
- Nivel 5 Optimizado
  - Gestión de cambios tecnológicos.
  - Análisis y resolución de las causas que generan los diferentes problemas y errores.

## Anexo 2: Modelo de medición del PSM

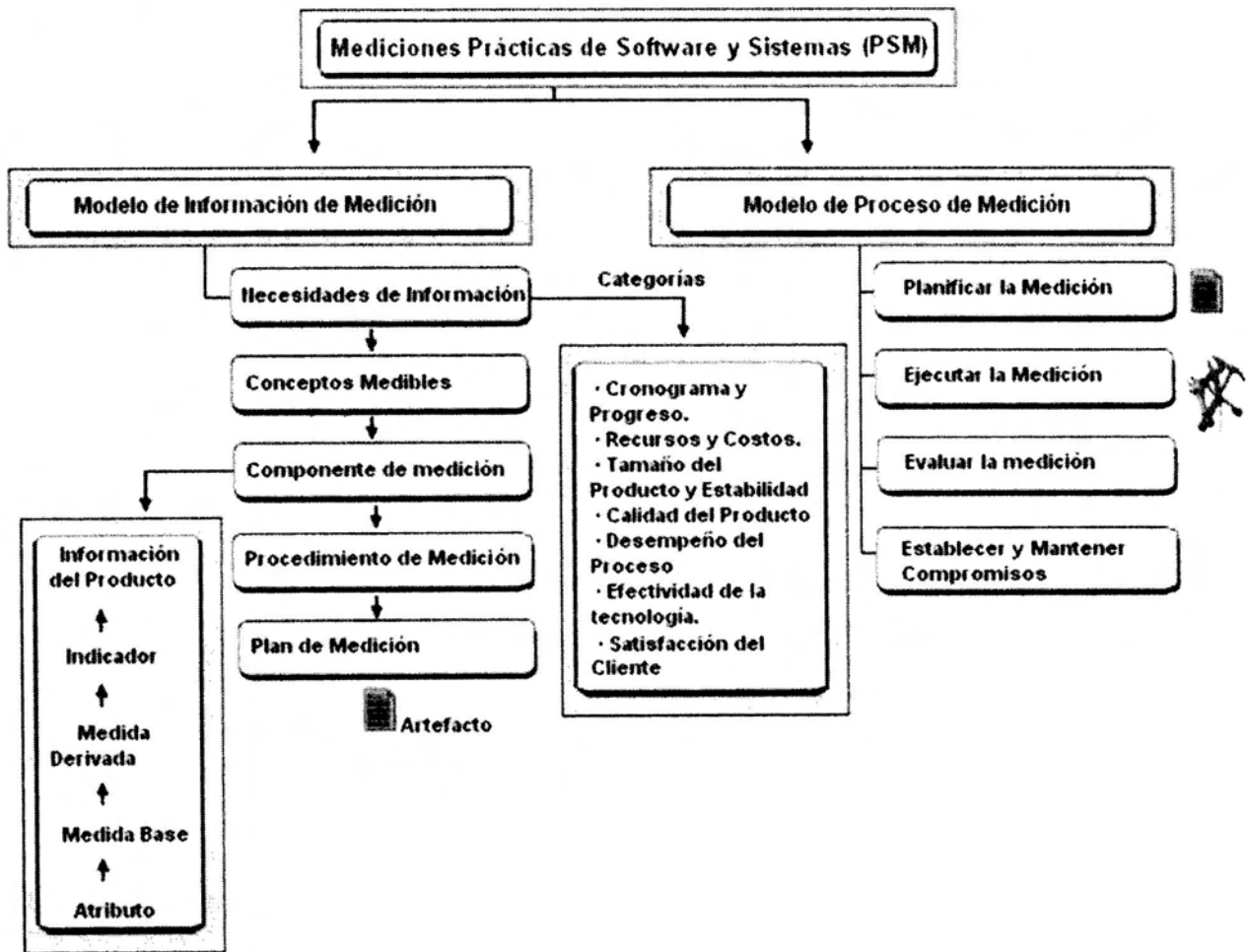


Figura 10: Modelo de medición de PSM [27]

**Anexo 3: Encuesta realizada a líderes y estudiantes del polo de simulación.**

Datos Generales

Nombre del proyecto:

1. ¿Cómo se organiza o estructura el polo de simulación de la facultad nueve?
2. ¿De qué forma se controla el avance del proyecto?  
Por el jefe \_\_\_\_ Reuniones Técnicas \_\_\_\_ Tormenta de ideas \_\_\_\_ Otras \_\_\_\_ ¿Cuál (es)?  
\_\_\_\_\_
3. ¿Cómo fueron seleccionados los estudiantes y/o profesores que pertenecen al proyecto?
4. ¿Los integrantes del proyecto tenían preparación previa, se encontraban familiarizados con el lenguaje de programación, las herramientas a utilizar para la elaboración del producto requerido?
5. ¿Cómo está organizado el equipo de desarrollo de software?
6. ¿Existe documentación alguna de lo que se va desarrollar en el proyecto?
7. ¿De qué especialidad son los profesores que se encuentran trabajando en el polo?
8. ¿Cómo se reflejan las actividades referentes a la calidad del software?
9. ¿Existe alguien en su proyecto responsable del proceso de aseguramiento de la calidad?  
Si \_\_\_\_ No \_\_\_\_
10. ¿Cuentan con un grupo independiente del aseguramiento de la calidad?  
Si \_\_\_\_ No \_\_\_\_
11. ¿Qué roles tienen definidos?
12. ¿Cómo se organiza el proceso de desarrollo de dicho software en el polo de simulación de la facultad nueve en la UCI?

13. ¿Tienen establecido la política de calidad de la organización?

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

14. ¿Tienen establecidos los objetivos de calidad del polo?

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

15. ¿Cómo se evalúa la calidad del proceso de desarrollo de software?

16. ¿Cómo se evalúa la calidad del producto de software?

17. ¿Aplican algún indicador durante el proceso de desarrollo del software de simulación con el fin de controlar la calidad de este?

18. ¿Cuáles son los principales problemas que se han presentando durante el desarrollo del producto?

19. ¿Qué modelos, estándares o normas de calidad utilizan?

20. ¿Tienen conocimiento de estos modelos, estándares o normas de calidad?

21. ¿Utilizan alguna variable de control? ¿Cuáles?

22. Los estudiantes involucrados en el polo tienen algún conocimiento acerca de las métricas e indicadores de calidad de software que se utilizan en el mundo

23. ¿Tienen conocimiento sobre los posibles indicadores que pueden aplicarse en el polo de simulación?

Si\_\_\_ No\_\_\_ Cuáles\_\_\_

24. Presentan algún registro histórico guardado

## Anexo 4: Resultado de la encuesta, los puntos más significativo del polo de simulación

Polo de Simulación						
Simuladores	¿Conoce algún indicador de calidad?	¿Aplican indicadores de calidad?	¿Conoce los que puede aplicar?	¿Conoce algunas normas, modelos de calidad?	¿Aplican algunas de dichas normas y modelos de calidad?	¿Tienen guardado algún registro histórico?
Simulador para la IQC en SW libre	No	No	No	Si	No	Si
Simulador Experto para STA	No	No	No	Si	No	Si
Simulador Termo Azúcar	No	No	No	Si	No	Si

Tabla 1: Puntos más significativo del polo de simulación

### Glosario de términos

**Artefacto:** es una pieza de información que (1) es producida, modificada o usada por el proceso, (2) define un área de responsabilidad para un rol y (3) está sujeta a control de versiones. Un artefacto puede ser un modelo, un elemento de modelo o un documento.

**Kernel:** es una arquitectura básica que permite el desarrollo de diferentes aplicaciones con características similares.

**Gestión de la configuración:** Tarea de definir y mantener las configuraciones y versiones de los artefactos. Esto incluye la definición de la línea base, el control de versiones de ésta y el control del almacenamiento de los artefactos.

**RUP (Rational Unified Process):** es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML y trabajo de muchas metodologías utilizadas por los clientes.

**XP (Extreme programming):** La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.

**Híbrido:** Proceso heterogéneo, mezclado con ambas metodologías de desarrollo

**Staff:** Conjunto de recursos que asesoran y colaboran con un componente específico dentro del proyecto, para lograr los objetivos trazados.

**SPC (Statistical process control):** El Control Estadístico de Procesos es un conjunto de herramientas estadísticas que permiten recopilar, estudiar y analizar la información de procesos repetitivos para poder tomar decisiones encaminadas a la mejora de los mismos, es aplicable tanto a procesos productivos como de servicios siempre y cuando cumplan con dos condiciones: Que sea mensurable (observable) y que sea repetitivo.

**STA:** Simulador Termo Azúcar, software desarrollado en el proyecto de simulación de la facultad nueve.

**Metodología:** Es el conjunto de técnicas y procedimientos que nos permiten conocer los elementos necesarios para definir un proyecto de software; en palabras sencillas es la base para la edificación de un proyecto de software, la etapa fundamental para lograr los objetivos buscados con dicho proyecto.

**Releases:** significa entrega, son los que permiten definir las iteraciones necesarias para cumplir con los objetivos, de manera que cada resultado de la iteración sea un programa aprobado por el cliente de quien depende la definición de las siguientes iteraciones.

**Operacionalizar:** Es la asignación de significado a las variables, especificando las actividades u operaciones que han de realizarse para medirlas. Es la interpretación de las variables en términos empíricos a través de indicadores que revelan la existencia de cualidades esenciales en rasgos observables.

**Desperdicios de la producción:** en este caso se refieren a las versiones beta que realizan antes de la última versión, es decir los releases, todas estas versiones que cuando el cliente la prueba están defectuosas todavía.