

**Universidad de las Ciencias Informáticas**

**Facultad 10**



*Propuesta de un sistema que permita gestionar  
Indicadores para SXP*

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

**Autoras:** Yumislady Arias Chamorro  
Daliana Montada Salas

**Tutoras:** Ing. Gladys Marsi Peñalver Romero  
Ing. Greidys Jorda Luege

**Ciudad de La Habana, del 2010**

**“Año 52 de la Revolución”**



*“Cuando puedas medir lo que estás diciendo y expresarlo en números, sabrás algo acerca de eso; pero cuando no puedes medirlo, cuando no puedes expresarlo en números, tus conocimientos serán escasos y no satisfactorios”*

*Lord Kelvin*

## **Declaración de autoría**

Declaramos ser autoras de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

---

Dalíana Montada Salas

Autora

---

Yumislady Arias Chamorro

Autora

---

Ing. Gladys Marsi Peñalver Romero

Tutora

---

Ing. Greidys Jorda Luege

Tutora

## Agradecimientos

---

*Agradecemos a nuestro comandante en Jefe Fidel Castro Ruz y a la Revolución Cubana por darnos la oportunidad de formar parte de este proyecto. A nuestra modesta tutora Gladys y a Greydis por guiarnos, ayudarnos, confiar en nosotras, estar siempre presente en cada momento que las necesitamos, sin ellas no hubiese sido posible y a todas las personas que de alguna forma han colaborado en nuestra formación como profesionales.*

### *Daliana*

*A mis padres por su apoyo, por estar siempre a mi lado, por enseñarme que la vida es de sacrificio y por haber confiado siempre en mí, los amo con todas mis fuerzas.*

*A mi abuela Matilde, por malcriarme, por ser mi mami linda, por estar ahí siempre que la he necesitado, la que amo con todo mi corazón y que se lo orgullosa que esta de mi.*

*A la tía que más quiero, Monalita, por quererme como a su niña y complacer todos mis caprichos.*

*A mi abuela Magali por ser la mejor abuela del mundo, por malcriarme, a mis abuelos y abuela Cuca que aunque están en el cielo fueron tan especiales conmigo y les hubiera gustado verme hecha una ingeniera.*

*A mi hermano Juan Carlitos y a mis hermanas Daylen y Milagrato, el mejor regalo que me ha dado la vida, parte de mis alegrías y preocupaciones, los quiero mucho.*

*A mi tío Víctor, Paquitín, a mi tío abuelo José, a Aníbal y a Pepín que me apoyan y me quieren.*

*A mis tías Maritza y Araceli por estar siempre presente, a Arianne por todo el apoyo que me ha brindado en todos estos años, a mi primo Pancho por cuidarme y estar siempre ahí cuando lo he necesitado.*

*A mi familia en general, por poder contar con ellos en todo momento, por el cariño que me brindan y por ser como son...*

*A mis taticas: Dayi por aconsejarme cuando no tengo ni idea de que hacer, por ser incondicional, por quererme como soy y a Yeni por darme ánimo y ayudarme a ver las cosas positivas cuando estoy tocando fondo. Ambas desde*

## Agradecimientos

---

*la primaria se convirtieron en más que amigas, por estar siempre conmigo en las buenas y en las malas, por ser mis mayores confidentes y siempre darme motivos para alegrar el día o al menos hacen el intento. También a sus padres.*

*A Karel y a Oigre, por ser mis camaroncitos duros y ayudarme cuando lo he necesitado, y por sobre todo ser tan buenos amigos.*

*A mis compañeros del grupo 506 por haber compartido durante estos 5 años, en especial a mi tato Magdiel por ser el niño más bueno del mundo y ayudarme siempre, a Yuyi por todos sus consejos y a Yazmin porque a pesar de la distancia estar siempre ahí, por haber sido mi mayor apoyo aquí en la UCI con nuestros antibajo pero le debo mucho, a todos los voy a extrañar y siempre estarán en mi.*

*A mis compañeros del grupo 507 a Mirelis, Ale, Nelio, Katia, Daylenis, Yurenia, sobre todo al Chino y a Charlie mis niños que les tengo un gran cariño, más que nada por hacerme reír y estar siempre a mi lado, y a Heydi que a pesar de su carácter es una buena amiga por darme fuerza en el momento que lo*

*necesitaba y demostrarme que cuando se quiere se puede, la amistad que ustedes me han brindado quedará para siempre.*

*A todas las personas que he conocido y han convivido conmigo durante estos 5 años en esta inmensa escuela, Yane, Dunia, Ailen, Yoel en especial a Yunet por brindarme un pedacito de su corazón y ser una gran persona, a Niurkita por todo su apoyo cuando estuvimos en Venezuela, por ofrecerme su amistad, por no dejarme desamparada, por las comiditas ricas, eres especial y a Keiver por ser tan tierno y por tener el poder de con solo un abrazo levantarme el ánimo.*

*A mi compañera de tesis, Yumi por haberme tenido que tolerar todo este tiempo, por todo el esfuerzo que hicimos junta para realizar nuestro sueño, eres una persona maravillosa, no cambies nunca.*

*Y a todas esas personas que siempre han estado conmigo y me han aceptado tal y como soy, que me han brindado apoyo incondicional y forman parte de mis triunfos en la vida.*

*Muchas Gracias*

## Agradecimientos

---

### *Yumi*

*A mi mamá y mi papá por todo el amor que siempre me han brindado, por darme todos los gustos que quise, por apoyarme incondicionalmente en todo momento y estar ahí cada vez que los necesitaba.*

*A mi hermano porque siempre he pensado en él para seguir adelante y a mi cuñada Laritza por el apoyo que me han brindado siempre los dos aunque estén lejos.*

*A mi sobrinita Claudi que a pesar de que nos fajamos, la quiero mucho, mucho y espero que te sea de ejemplo.*

*A mi Tati Julio por quererme, ayudarme y soportarme este tiempo que llevamos juntos.*

*A mi segunda madre y padre, Dalía y Jorge, por al amor y el apoyo que me han dado estos 5 años.*

*A mi prima del alma Miri que me quiere tanto, por preocuparse por mí y apoyarme siempre.*

*A mis abuelas Mima y Celina por el amor que me han dado toda mi vida.*

*A mi abuelo papi Luis que no me pudo ver graduada, pero sé que estaría muy orgulloso de mí.*

*A mi tío Tito que siempre me dio fuerzas para seguir adelante.*

*A mi tía Tina, que siempre me alegró con sus locuras, gracias por todo tu cariño.*

*A mi otra tía, Mirtha por consentirme y ser mi tía querida.*

*A toda mi familia por apoyarme y confiar en mí, y sé que se encuentran muy orgullosos de la persona en la que me he convertido.*

*A mis amigas que siempre estuvieron presente en las buenas y malas, por aguantarme estos 5 años: Dunia que aunque tuvimos nuestras diferencias sigue siendo mi hermanita, Yane por apoyarme todo este tiempo y estar ahí siempre, Ailyn que no llevamos los 5 años juntas pero desde que nos unimos me a apoyado y ayudado. A todas las quiero mucho, niñas...*

*A mi compañera de tesis Daliana, por no dejarse vencer y estar siempre dispuesta a trabajar y aguantarme los momentos de locura y estrés.*

*A Orlando que llevamos todo este tiempo de universidad y me ha ayudado tanto, mayormente a la hora de cargar los bultos.*

## Agradecimientos

---

*Al otro compañero del cuarto, Ernesto que jodé cantidad todo el tiempo, pero hace unos buenos espaguetis, y me aguantó un buen tiempo.*

*A Arlee, que me ayudó, me aguantó y siempre estuvo ahí para cuando la necesité.*

*A todo mis compañeros de aula, por el tiempo que pasamos juntos, el grupo 10101, que aunque duramos poco tiempo juntos estuvimos unidos, y al 10505 que me ha apoyado y ayudado la mayor parte de mi carrera.*

*A todos aquellos que de una forma u otra han ayudado al desarrollo de este trabajo y creyeron en mí.*

*A todos mil gracias*

### *Daliana*

*A mis papito Juan Carlos y a mi mamita Milagros que son mi razón de ser.*

*A mi abuela Matilde que es mi madre querida, el faro que ilumina mi vida.*

*A mi tía Carmen Monola por estar siempre presente.*

*A todos les dedico este gran sueño; por el amor, comprensión y dedicación que me han dado. Por no faltar nunca cuando los he necesitado, por complacer todos mis caprichos y tener tanta paciencia. Por escucharme y aconsejarme.*

*Los quiero mucho...*

### *Yumi*

*A mi mamá y mi papá por toda la confianza que siempre han tenido en mí, por ser los faros de mi vida y mi razón de vivir.*

*A mi hermano y mi cuñada por el amor y los buenos consejos que me han dado siempre.*

*A mi sobrina Claudia y al abuelo de mi corazón, papi Luis.*

En la actualidad el desarrollo de software ha evolucionado, trayendo consigo la aplicación de una gestión de proyecto detallada en todo el ciclo de vida, sumando buenas prácticas para mejorar el funcionamiento y ganar en calidad. En el 2009 se realizó en la Facultad 10, un estudio sobre las métricas, con el objetivo de confeccionar una propuesta adaptable de las mismas a la metodología ágil SXP. El objetivo fundamental del presente trabajo es a partir de la validación de la propuesta efectuada, realizar el diseño de un sistema para la gestión de indicadores de esta metodología. Se dieron respuestas a interrogantes tales como: ¿qué es un indicador?, ¿qué beneficios reporta su aplicación en todas las etapas de desarrollo de software?, ¿qué características deben tener para poder ser aplicados?, ¿en qué nivel se encuentran los propuestos para SXP? A partir del análisis de los resultados obtenidos en la aplicación de los indicadores de SXP, en el grupo UNICORNIOS y del estudio de los sistemas para la gestión de indicadores existentes, se procede a la propuesta del diseño de una aplicación web, para su gestión en SXP.

**Palabras Claves:**

Gestión de Proyecto, Calidad, SXP, Sistemas de Gestión, Indicadores.

# Tabla de Contenidos

---

Introducción .....	1
Capítulo 1: Marco Conceptual para los Indicadores y Herramientas .....	5
1.1.    Reseña del Grupo UNICORNIOS .....	5
1.2.    Panorama de la Metodología Ágil SXP.....	5
1.3.    Calidad del Software .....	6
1.4.    Indicadores de Software .....	10
1.4.1.    Tipos de Indicadores .....	12
1.4.2.    Niveles de los Indicadores .....	13
1.4.3.    Métricas Propuestas para SXP .....	15
1.5.    Herramientas para la Modelación del Diseño de la Propuesta del Sistema .....	25
1.5.1.    Visual Paradigm.....	25
1.5.2.    Lenguaje de Modelado .....	26
1.5.3.    Pencil.....	26
Capítulo 2: Validación.....	28
2.1.    Técnicas de Validación .....	28
2.2.    Proyectos donde se Aplican los Indicadores .....	31
2.2.1.    Proyecto SistClon .....	32
2.2.2.    Proyecto NovaDesk .....	32
2.2.3.    Proyecto YAREY .....	33
2.3.    Validación de los Indicadores de SXP .....	33
2.3.1.    Validación en el Proyecto SistClon .....	33
2.3.2.    Validación en el Proyecto NovaDesk.....	55
2.3.3.    Validación en el Proyecto Yarey .....	68
2.4.    Resultados Obtenidos en la Validación .....	74
Capítulo 3: Propuesta del Diseño del Sistema.....	76

## Tabla de Contenidos

---

3.1.	Sistemas Existentes para la Gestión de Indicadores.....	76
3.2.	Situación Actual .....	79
3.3.	Propuesta del Sistema .....	79
3.4.	Análisis del Negocio .....	81
3.5.	Lista de Reserva del Producto .....	83
3.6.	Historias de Usuarios.....	86
3.7.	Diagrama de Clases de Diseño con Estereotipos Web.....	116
3.8.	Beneficios del Sistema SG – Indicators.....	124
	Conclusiones .....	125
	Recomendaciones .....	126
	Referencias Bibliográficas .....	127
	Bibliografía .....	129
	Anexos.....	132
	Glosario de Términos .....	145

## Introducción

En la industria de software reducir los costos de producción y minimizar en cada fase del proceso de desarrollo el número de defectos de los artefactos; contribuyen a obtener un producto de calidad, en el menor tiempo posible, ubicarlo en el mercado y adquirir ganancias.

La utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software permiten uniformar la filosofía de trabajo, en aras de lograr mayor confiabilidad, mantenibilidad y facilidad del producto, a la vez que elevan la productividad en toda la labor de su desarrollo.

Una manera eficaz para evaluar parámetros como funcionalidad, complejidad y eficiencia durante el proceso de producción de software es la utilización de indicadores, medidas efectuadas sobre algunos aspectos del sistema en desarrollo que permiten obtener conclusiones sobre ellos. Su fin es ayudar a tomar las decisiones necesarias para favorecer la calidad en todo el ciclo de vida del proceso de desarrollo de software.

La Universidad de las Ciencias Informáticas (UCI) juega un papel importante en el desarrollo de la industria de software cubano, donde estudiantes, profesores y especialistas se dedican a la realización de software. La UCI no está aislada del problema de las mediciones, hoy la mayoría de los desarrolladores de software, dígase líderes de proyectos, programadores, diseñadores, no miden ni administran la calidad de los productos.

Desde el año 2008 en el grupo UNICORNIOS de la Facultad 10 y en otros proyectos de la universidad se utiliza como guía para su proceso de desarrollo de software SXP, híbrido cubano de metodologías ágiles. Dicha metodología fue sumando buenas prácticas para un mejor funcionamiento y ganar en calidad, por lo que surge en el año 2009 un trabajo de diploma con el título: *“Propuesta de técnicas de estimación y*

## Introducción

---

*métricas para la metodología ágil SXP*” de los ingenieros Ailema Sisinta Ayra y Adolis D. Rodríguez Reyes [1]. En el trabajo se realiza una profunda investigación sobre las métricas en las metodologías ágiles con el objetivo de realizar la propuesta de estas para SXP, las cuales no fueron validadas, por tanto se desconoce su total funcionamiento por parte del equipo de trabajo.

Actualmente en el grupo UNICORNIOS a pesar de que se dedica a la producción de software no se cuenta con la cultura adecuada de los indicadores por no haber sido aplicados. Tampoco se cuenta con una técnica automatizada que permita realizar la gestión de los indicadores para un menor número de documentos y un trabajo con mayores niveles de eficacia, efectividad, y eficiencia.

Atendiendo a los elementos antes planteados, para darle solución a la **situación problemática** descrita anteriormente, se plantea como **problema científico** a resolver ¿Cómo mejorar la calidad del software de los proyectos productivos que utilizan la metodología ágil SXP? Partiendo de la **idea** de que con una exitosa validación de las métricas propuestas, se realizará el diseño de un sistema para gestionarlas.

Siendo el **objeto de estudio** que abarca la investigación el perfeccionamiento del proceso de la calidad de software y el **campo de acción** la utilización de los indicadores en los proyectos que utilizan SXP. Para la introducción y posterior utilización de dichos indicadores en el proceso productivo se trazó el siguiente **objetivo general**: realizar el diseño de un sistema para la gestión de indicadores a partir de la validación de la propuesta de métricas para la Metodología Ágil SXP.

Como **objetivos específicos** para cumplir con el objetivo general propuesto se definieron los siguientes:

- Validar la propuesta de indicadores para SXP.
- Identificar la existencia de sistemas de gestión de indicadores a nivel internacional y nacional.
- Proponer el diseño de un sistema que permita la gestión de indicadores a los proyectos productivos que utilizan SXP.

Se pretende tener como **posible resultado** una validación exitosa de los indicadores y la realización del diseño básico del sistema que permita su gestión, para lograrlo se ha de cumplir con los objetivos trazados anteriormente por lo que se hace necesario desarrollar las siguientes **tareas de investigación**:

- Selección de proyectos productivos del grupo UNICORNIOS que se tomarán como muestra para validar los indicadores propuestos para SXP.
- Aplicación de los indicadores propuestos en los proyectos escogidos para determinar el éxito de los mismos.
- Definición cuantitativa de los resultados obtenidos al aplicar los indicadores en los proyectos seleccionados.
- Realización de un análisis de la existencia de sistemas para la gestión de indicadores a nivel internacional y nacional.
- Realización de la propuesta del diseño para el sistema de gestión de indicadores para SXP.

Para la realización de la investigación se utilizó como **método teórico** para cumplir con las tareas mencionadas, el **histórico-lógico**, que permitió realizar un análisis de la trayectoria, evolución y desarrollo de la aplicación de los indicadores de software, además realizar una síntesis del uso de los mismos extrayendo los elementos más importantes relacionados con el campo de acción. De los **métodos empíricos** se utilizó el método de **medición** para determinar la factibilidad de la aplicación de los indicadores en los proyectos y como técnica de recolección de información la **entrevista**, para profundizar a través de las opiniones de especialistas de calidad, acerca de todo lo referente a la utilización de los indicadores en la UCI.

La investigación está sustentada en 3 capítulos, estructura que se describe a continuación:

**Capítulo 1:** Se presenta una breve reseña del surgimiento del grupo UNICORNIOS y de la metodología SXP. Se hace un análisis de los diferentes conceptos y definiciones que se tendrán en cuenta durante

todo el trabajo sobre calidad de software e indicadores. También se hace referencia a los diferentes estándares para medir la calidad en los proyectos según los indicadores que se utilizan, específicamente los de SXP.

**Capítulo 2:** En este capítulo se selecciona el método de validación para los indicadores de SXP. Se describe el área de aplicación de los indicadores propuestos, así como las características que presentan los proyectos que se desarrollan en el grupo UNICORNIOS. Finalmente se procede a realizar un análisis de los resultados obtenidos.

**Capítulo 3:** En este capítulo se realiza un análisis de la existencia de sistemas para la gestión de indicadores en Cuba y el mundo, y se procede a realizar la propuesta de diseño del sistema para SXP.

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

## Capítulo 1: Marco Conceptual para los Indicadores y Herramientas

En el proceso de desarrollo de software, para poder asegurar que sus productos resultantes son efectivos o poder compararlos, es necesario asignar indicadores mediante el cual se pueda llevar a cabo dicha comparación y tomar las mejores decisiones. El objetivo fundamental de este capítulo consiste en esclarecer los aspectos fundamentales sobre los indicadores, específicamente de los propuestos para SXP.

### 1.1. Reseña del Grupo UNICORNIOS

UNICORNIOS surge en septiembre del 2005 como grupo de Soporte Técnico, inicialmente tenía como propósito crear un grupo de soporte para la imagen GNU/Linux de la docencia en la Facultad 10. Más tarde se convierte en el grupo de proyectos que es hoy, un grupo de investigación y desarrollo que forma parte del polo de Software Libre (SWL) de la UCI; con el objetivo de la migración y soporte de aplicaciones y servicios a dicho polo. Entre los diferentes productos que se desarrollaron en este proyecto se encuentra SXP, metodología para guiar el proceso de desarrollo de un software.

### 1.2. Panorama de la Metodología Ágil SXP

La metodología ágil SXP surge en el 2008 para ser aplicada en aquellos proyectos productivos con pequeños equipos de trabajo, rápidos cambios en los requisitos o requisitos imprecisos. Está formada por las metodologías SCRUM, para la gestión de proyectos de forma eficiente, y XP<sup>1</sup>, para la ingeniería de software, donde el cliente es parte del equipo de trabajo hasta llegar al éxito del producto. Consta de cuatro fases principales planificación-definición, desarrollo, entrega y por último mantenimiento; donde en cada una de estas, se realizan actividades que generan artefactos que permiten documentar todo el proceso de desarrollo. SXP ayuda a que trabajen todos juntos en el equipo, en la misma dirección y con

---

<sup>1</sup> Xtreme Programming

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

un objetivo claro, también ayuda al líder del proyecto a tener un mejor control sobre el progreso del trabajo.

Dicha metodología será usada como guía para realizar la modelación del diseño del sistema que será propuesto para la gestión de indicadores de SXP.

## 1.3. Calidad del Software

La calidad de software es un concepto que ha ido renovándose con los años y existe una gran variedad de formas de concebirla, tales como:

“Una característica o atributo de algo. American Heritage Dictionary.

La totalidad de rasgos y características de un producto, proceso o servicio que sostiene la habilidad de satisfacer estados o necesidades implícitas. Definición estándar de calidad en ISO<sup>2</sup>-8402.

Concordar explícitamente al estado funcional y a los requerimientos del funcionamiento, explícitamente a los estándares de documentación de desarrollo, e implícitamente características que son expectativas de todos los desarrolladores profesionales de software”. [2]

Según la Real Academia de la Lengua Española, la calidad es: 1. Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor. 2. Condición o requisito que se pone en un contrato.

En ámbitos empresariales es definida como:

[1] Grado en el que un conjunto de características inherentes cumple con los requisitos<sup>3</sup>.

[2] El conjunto de actividades encaminadas a descubrir y satisfacer las necesidades de un colectivo o de una sociedad en general. Satisfacción del cliente y conformidad con sus requisitos y necesidades<sup>4</sup>. [3]

Por lo antes expuesto se define que la calidad en sí, es la aptitud de un producto o servicio para satisfacer las necesidades del usuario, y en la informática, la calidad de software no es más que el grado en que un software cumple con los requisitos especificados.

---

<sup>2</sup> International Organization for Standardization

<sup>3</sup> Sistemas de gestión de la calidad. Fundamentos y vocabulario. Madrid, 2000.

<sup>4</sup> Gestión y Control de calidad. Cuadernos de la Universidad Nacional de Educación a Distancia. Madrid. 1994.

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

El principal objetivo de los ingenieros en el desarrollo de software, es obtener un producto de calidad para una mayor satisfacción del cliente. Se dice que un producto tiene calidad si cumple con los requisitos del cliente en cuanto a la funcionalidad, la ejecución, la confiabilidad, la disponibilidad y ayuda, a un costo menor o igual al que el cliente está dispuesto a pagar. Para garantizar la obtención de un producto con estas características, y eliminar así problemas como la entrega retrasada del producto, las organizaciones productoras de software se trazan modelos o estándares.

## Normas Organización Internacional para Estandarización (ISO)

Fue la primera organización encargada de crear un estándar para la creación de software, sus estándares conocidos como Internacionales ISO, son utilizados para tener una mayor calidad en el desarrollo de los productos que realizan las empresas. Entre estos estándares está la familia ISO 9000 para la implementación de sistemas de gestión y aseguramiento de la calidad que engloba varios estándares, como el ISO 9001, que se enfoca al diseño, producción e instalación; el ISO 9003, dirigido a inspecciones y ensayos finales, y otros. [4]

## Software Process Improvement and Capability Determination (SPICE)

Es el estándar ISO/IEC<sup>5</sup> 15504 se centra en la evaluación de los procesos de desarrollo de software para evaluar la capacidad de los procesos, contribuir a la mejora continua y ser la base para el comercio internacional de software. La arquitectura de este modelo se basa en dos dimensiones, la dimensión de procesos, donde agrupa a los proceso de acuerdo al tipo de actividad en tres grupos con categorías: procesos primarios, procesos de soporte y procesos organizacionales; y la dimensión de capacidad que define 6 niveles para determinar la capacidad de los procesos. Los niveles proporcionan una manera racional de progresar en la mejora de la capacidad de cualquier proceso. [5]

### Niveles de SPICE:

Nivel 0: Incompleto

---

<sup>5</sup> International Electrotechnical Commission

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

Nivel 1: Realizado

Nivel 2: Gestionado

Nivel 3: Establecido

Nivel 4: Predecible

Nivel 5: En optimización

## Modelo de Capacidad y Madurez Integrado (CMMI)

Modelo de mejora de procesos de desarrollo que se clasifica en niveles de madurez que sirven para conocer la madurez de los procesos que se realizan para producir software. Los componentes del modelo CMMI son áreas de procesos, metas genéricas, metas específicas, prácticas genéricas, prácticas específicas y sub prácticas.

Utiliza dos formas de representación como guía para efectuar las actividades de mejora de los procesos; la representación continua que se centra en la mejora de un proceso relacionado estrechamente a un área de proceso en que una organización desea mejorar, está formada por seis niveles de capacidad por donde transitan los procesos y cada nivel es construido sobre el nivel anterior; y la representación escalonada o por etapas que ofrece un método estructurado y sistemático de mejoramiento de procesos, que implica mejorar por etapas o niveles y consta de cinco niveles de madurez. (Ver Anexo 1)

El modelo está estructurado en 22 áreas de procesos; un área de proceso es un conjunto de prácticas relacionadas que cuando son implementadas colectivamente, satisfacen un conjunto de objetivos considerados importantes para mejorar esa área de proceso, están agrupadas en cuatro categorías: Administración de Procesos, Administración de Proyectos, Ingeniería y Soporte. Una de estas áreas claves de proceso es la de medición y análisis, la cual es la encargada de proporcionar las mediciones para evaluar los procesos que se realizan dentro de un proyecto, y conocer el grado de calidad con que se ha construido el software.

Actualmente la universidad realiza un programa de mejora para optar por el nivel 2 de CMMI específicamente en el área de procesos de medición y análisis, que corresponde a la categoría de

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

Soporte. Esta área establece métricas con el objetivo de entregar resultados objetivos que sirvan como base para tomar decisiones informadas y correctivas.

El área de proceso medición y análisis está compuesta por dos objetivos específicos y cada uno se desglosa a su vez en cuatro prácticas específicas. Los objetivos son [6]:

## 1. Alinear las actividades de medición y análisis.

- Establecer los objetivos de la medición para establecer y mantener los objetivos que se derivan de los requerimientos.
- Especificar las métricas a usar en el proyecto basándose en los objetivos de la medición.
- Especificar los procedimientos de recolección y almacenamiento de los datos, así como la herramienta a utilizar.
- Especificar los procedimientos del análisis de los datos de la medición y se describe como serán analizados y reportados estos datos. Se seleccionan además los métodos y las herramientas apropiadas para el análisis de los datos y se especifican y priorizan los análisis que será realizados y los reportes que serán preparados.

## 2. Proporcionar los resultados de la medición.

- Recolectar los datos de la medición mediante el procedimiento especificado obteniendo así los datos de las métricas básicas y derivadas.
- Analizar e interpretar los datos de la medición y si es necesario se dirigen análisis adicionales.
- Administrar y almacenar los datos y los resultados de la medición, además permite establecer permisos a determinados usuarios sobre los datos almacenados.
- Comunicar los resultados de las actividades de medición y análisis de forma oportuna y clara, para ayudar a que se tomen decisiones y acciones correctivas.

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

## 1.4. Indicadores de Software

En la mayoría de los desafíos técnicos, los indicadores ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. Con la utilización de los indicadores se puede determinar si el producto en desarrollo está mejorando y si posee o no la calidad requerida. La necesidad de la medición es algo evidente, es lo que permite cuantificar y por consiguiente gestionar de forma más efectiva.

¿Por qué se hace necesario medir?: Para caracterizar, evaluar, predecir y mejorar.

Caracterizar

- Comprender el proceso, producto y entorno actual.
- Proporcionar una línea base para futuras evaluaciones.

Evaluar

- Determinar el estado, de forma que los proyectos y procesos puedan ser evaluados.
- Evaluar el alcance de los objetivos de calidad.
- Evaluar el impacto de la tecnología sobre los procesos y productos.

Predecir

- Comprender la relación entre procesos y productos.
- Establecer objetivos alcanzables de calidad, coste y planificación.

Mejorar

- Identificar causas raíz y oportunidades para mejorar.
- Seguir los cambios en las realizaciones y comparar con la línea base.
- Comunicar razones para mejorar.

Para llegar al concepto de indicadores se debe aclarar los términos medida, medición y métrica.

**Medida:** indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto. [7]

**Medición:** acto de determinar una medida. [8]

**Métrica:** medida cuantitativa del grado en que un sistema o proceso posee un atributo dado. [9]

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

**Indicadores:** métrica o combinación de métricas que proporcionan una visión profunda del proceso de software, son herramientas útiles para la planeación y la gestión en general. [10]

Después de diferenciar anteriormente cada concepto, se define que los indicadores de software proporcionan una medida de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Permiten tener una visión del proceso, del proyecto o del software en sí, y realizar ajustes en busca de mejoras.

En una entrevista realizada a la Máster Irina Napal, jefa del departamento de Calidad UCI planteó que el término de métricas no está en uso, actualmente éstas son conocidas como indicadores pero de cierta forma siguen el mismo principio de lograr mayores niveles de calidad durante todo el ciclo de vida del desarrollo del software; por tanto en la presente investigación éstas se tratarán como indicadores.

Para controlar la calidad del software es necesario, ante todo, definir cuáles son los parámetros, indicadores o criterios de medición a utilizar. Los indicadores no son más que instrumentos muy importantes y valiosos para evaluar, dar seguimiento y orientar al equipo de trabajo, cómo se puede alcanzar mejores resultados en el proceso de desarrollo de software. Permiten determinar los niveles de productividad del proyecto. Proporcionan una visión profunda que permite al gestor de proyectos o a los ingenieros de software reajustar el producto, el proyecto o el proceso. Se detectan 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 asociado con éste.

Los indicadores de gestión de un proyecto ayudan a medir objetivamente que se cumplieron los objetivos parciales y finales del proyecto. Son utilizados continuamente a lo largo del ciclo de vida de todo el proceso de desarrollo de software, para evaluar el desempeño y los resultados cuantificables.

Cuanto más exacto es un indicador, más difícil es su aplicación; por tanto deben poseer cuatro características principales para que sean válidos al utilizarlos:

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

1. Cuantificables: deben estar basadas en hechos.
2. Independientes: los recursos no deben poder ser alterados por los miembros que las apliquen o utilicen.
3. Explicables: deben contar con la información documentada acerca de las mismas y su uso.
4. Precisos: deben contar con un nivel de tolerancia permitido cuando se mide.

Para que los indicadores aporten valor a los proyectos, deben ser consistentes, comparables, simples, calculables y no deben ser específicos para ningún lenguaje de programación.

## 1.4.1. Tipos de Indicadores

Existen innumerables indicadores con propósitos diferentes que reflejan o describen la conducta del software, estos pueden medir entre otros aspectos la competencia, calidad, desempeño y la complejidad del software contribuyendo a establecer de una manera sistemática y objetiva una visión interna del trabajo mejorando así la calidad del producto.

Los indicadores están orientados a tres entidades fundamentales: producto, proceso y proyecto.

- **Indicadores del producto:**

Los productos están compuestos por artefactos, los cuales pueden ser documentos, modelos, módulos, o componentes, por tanto, los indicadores del producto deben hacerse sobre la base de medir cada uno de los artefactos, describen características como el tamaño, complejidad, rasgos del diseño, rendimiento y nivel de calidad. [11]

- **Indicadores del proceso:**

Los procesos de software se pueden definir como una secuencia de pasos requeridos para desarrollar o dar mantenimiento al mismo. Los indicadores del proceso son las que cuantifican el comportamiento de los procesos, los cuales son generalmente objetivos, absolutos, explícitos y dinámicos. [12]

- **Indicadores del proyecto:**

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

Los indicadores del proyecto son aquellas que describen las características del proyecto y la ejecución de éste. Algunos ejemplos pudieran ser: el número de programadores de un software, el costo, planificación y productividad del equipo.

A veces existen proyectos que no necesitan todos los indicadores, esto puede ser variable en dependencia de las características de los proyectos y de los cambios que sufran a lo largo de su ciclo de vida. A continuación se muestran ejemplos de indicadores objetivos [13]:

**Indicador de Funcionalidad Completa:** mide avance en el proyecto cuando la funcionalidad es completa, es decir, está desarrollada, probada y estabilizada.

**Indicador de Nivel de Calidad:** detalla desde el punto de vista de calidad todo el ciclo de vida por el que va pasando el producto.

**Indicador Evolución de Prueba:** medir los defectos, cuántos aparecen y se cierran por día.

**Indicador de Cobertura de Prueba:** muestra cuánto habría que probar, cuánto se pudo probar y cuánto funciona bien. Esta medición se realiza a partir de los estados de los casos de prueba.

## 1.4.2. Niveles de los Indicadores

Se desglosan en niveles de acuerdo al nivel de madurez del proceso de desarrollo:

### Nivel 1: Proceso Inicial

En este nivel se forma la base de comparación con la forma en que las mejoras se realicen y se incrementa la madurez, se encuentran:

- El tamaño del producto
- El esfuerzo del personal

### Nivel 2: Proceso Repetible

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

En este segundo nivel se mide la cantidad de esfuerzo necesaria para desarrollar un sistema, la duración del proyecto, el tamaño y la volatilidad de los requerimientos, el costo global del proyecto, por lo que se calculan:

- Tamaño del software, o sea las líneas de código fuente no comentadas
- Los puntos de función
- Desviación de costo
- Porcentaje gastado del presupuesto general
- Para determinar el esfuerzo del trabajo de personal, se utilizan en este nivel:
- Esfuerzo real medido en unidades persona/mes

### **Nivel 3: Proceso Definido**

En este nivel de madurez, se recomienda evaluar la complejidad de los requerimientos, el diseño, el código y los planes de prueba, y evaluar la calidad de los requerimientos del diseño del código y de las pruebas. Los indicadores a utilizar son:

- Calidad de la especificación
- Diseño arquitectónico: métrica de Henry y Kafura
- Prueba
- Código fuente

### **Nivel 4: Proceso Administrativo**

Los indicadores utilizados en este nivel son usados para encontrar y estabilizar el proceso, así la productividad y la calidad coinciden con las expectativas.

- Diseño de interfaz
- Índice de madurez del software

### **Nivel 5: Optimización del proceso**

Los indicadores de este nivel son aquellos utilizados para mejorar el proceso.

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

Como parte del desarrollo de la investigación se constató que los indicadores propuestos para la metodología SXP se encuentran hasta el cuarto nivel de madurez.

## 1.4.3. Métricas Propuestas para SXP

Métricas propuestas del modelo CMMI:

- **Métricas de productividad ( $E$ )** usadas para medir el esfuerzo real de un proyecto, es decir se determina si es correcto el número de trabajadores asignados y el tiempo.

$E$  = esfuerzo

$P$  = número de personas

$T$  = cantidad de tiempo

Se calcula  $E = P * T$

Utilizada en el monitoreo de estado.

- **Estimación de plazo ( $EP$ )** para definir con precisión la duración del proyecto.

$EP$  = precisión en la estimación de plazo

$DR$  = duración real

$DE$  = duración estimada

Se calcula  $EP = DR/DE$

Utilizada en el monitoreo de estado.

- **Desviación del tiempo ( $DT$ )** usada para determinar la desviación existente del tiempo real con respecto al planificado. Con su resultado se decide si se necesita replanificar.

$DT$  = desviación del tiempo total de desarrollo respecto al tiempo total planificado

## Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

$TT$  = tiempo total de desarrollo

$TD$  = tiempo usado de desarrollo del proyecto hasta el momento

$TP$  = tiempo planificado para las actividades por realizar en el proyecto

$TPI$  = tiempo total planificado inicialmente

Se calcula  $TT = TD + TP$

$$DT = TT - TPI$$

Si  $DT = 0$  entonces el proyecto está cumpliendo todo en su plazo de tiempo planificado.

Si  $DT > 0$  entonces es necesario replanificar las actividades del proyecto porque hay un retraso en el tiempo.

Si  $DT < 0$  significa que el tiempo en que se cumplieron las actividades fue menor al planificado, entonces se dice que hay una holgura de tiempo y esta puede ser usada más adelante en caso de que exista algún retraso.

Utilizada en el monitoreo de estado.

- **Riesgos  $R$  usados** para darle un nivel de prioridad al riesgo y determinar cuánto problema puede traer.

$R$  = exposición al riesgo

$I$  = impacto sobre el proyecto en una escala de 1 a 5

$P$  = probabilidad de que ocurra

Se calcula  $R = I * P$

El riesgo y la prioridad van a ser directamente proporcional, si uno aumenta el otro también. Según la prioridad que presente cada riesgo será el orden de la mitigación de este. Se utiliza en el

## Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

monitoreo de riesgo.

- **Cumplimiento de compromisos (CC)** usada para determinar la posibilidad de que el proyecto cumpla con los compromisos requeridos, tiene 2 opciones.

En el caso de que el cumplimiento con el compromiso dependa de actividades a realizar se determinará de la siguiente forma:

$CAi$  = cantidad de actividades necesarias incumplidas hasta el momento de la revisión

$CAn$  = cantidad de actividades necesarias para cumplir con el compromiso hasta la revisión

$CAr$  = cantidad de actividades necesarias realizadas hasta el momento de la revisión

Se calcula  **$CAi = CAn - Car$**

Si  **$CAi = 0$**  entonces todas las actividades necesarias para que se cumpla el compromiso fueron realizadas y este tiene una gran probabilidad de cumplirse.

Si  **$CAi > 0$**  hubo un incumplimiento en las actividades y es posible que no se cumpla con el compromiso.

Si el cumplimiento del compromiso depende de la asignación de recursos se determinará de la siguiente forma:

$X$  = recursos no disponibles de los necesarios para cumplir el compromiso

$XN$  = recursos necesarios para cumplir el compromiso

$XD$  = recursos disponibles de los necesarios para cumplir el compromiso

Si  **$RN = 0$**  entonces es posible que se cumpla el compromiso ya que existen todos los recursos que requiere.

Si  **$RN > 0$**  es posible que el compromiso no se cumpla porque no se cuenta con todos los recursos necesarios.

## Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

Se calcula  $X = XN - XD$

Se utiliza en el monitoreo de compromisos.

- **Desviación de Costo (CD)** usada para determinar la desviación existente del costo con lo que se ha planificado realmente.

$IC$  = índice de actuación de costo

$PP$  = presupuesto planificado

$CR$  = costo real

$CTP$  = costo total planificado

Se calcula  $IC = PP + CR$

$CD = IC - CTP$

Si  $CD = 0$  significa que se está cumpliendo con el presupuesto planificado.

Si  $CD > 0$  se necesita replanificar porque hubo una desviación de costo al planificado y mientras mayor sea este, mayor será la desviación.

Si  $CD < 0$  entonces se cuenta con una reserva monetaria que puede ser usada en otros momentos.

Se utiliza en el monitoreo de costo.

- **Porcentaje gastado del presupuesto general (PG)** usada para obtener una estimación exacta de lo que se ha gastado en el proyecto, con ella se pueden comparar los resultados de cada fase para detectar desviaciones significativas.

$HF$ : costo hasta la fecha

$PPT$ : presupuesto total

Se calcula  $PG = (HF * 100) / PPT$

## Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

Se utiliza en el monitoreo de costo.

Se proponen otras métricas expuestas por Pressman para la medición de diferentes aspectos importantes, no solo de la etapa de planificación sino en el desarrollo del proceso. Estas son las siguientes:

- **Métricas basadas en la función (PF)** para predecir el tamaño de un sistema que se va a obtener de un modelo de análisis.

*Cuenta-total*: suma de todas las entradas PF obtenidas

$F_i$  ( $i = 1$  a  $14$ ): valores de ajuste de complejidad.

Se calcula  $PF = \text{cuenta} - \text{total} * (0,65 + 0,1 * \text{Sum } [F_i])$

- **Calidad de la especificación (CE)** utilizada para valorar la calidad del modelo de análisis y la especificación de requisitos.

$N_r$  = requisitos en una especificación

$N_f$  = requisitos funcionales

$N_{nf}$  = requisitos no funcionales

Se calcula  $N_r = N_f + N_{nf}$  Especificidad (ausencia de ambigüedad) de los requisitos ( $ER$ )

$N_{ui}$  = número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas.

Se calcula  $ER = N_{ui} / N_r$

Mientras más cerca de 1 esté el valor de  $ER$  menor será la ambigüedad. Complejión de los requisitos funcionales ( $CRF$ )

$U_n$  = número de requisitos únicos de función

$N_i$  = número de entradas definidas o implicadas por la especificación

$N_s$  = número de estados especificados

## Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

Se calcula  $CRF = U n / (N i * N s)$

Esta relación mide el porcentaje de funciones necesarias que se han especificado para un sistema.

Grado de validación de los requisitos (GVR)

$Nc$  = número de requisitos validados como correctos

$Nnv$  = número de requisitos que no se han validado

Se calcula  $GVR = Nc / (Nc + Nnv)$

Esta relación trata los requisitos no funcionales.

- **Diseño arquitectónico: métrica de Henry y Kafura (MHK)**

$Longitud (i)$  = número de sentencias en lenguaje de programación en el módulo  $i$

$Fin (i)$  = concentración del módulo  $i$

$Fout (i)$  expansión del módulo  $i$

Se calcula  $MHK = longitud (i) * [Fin (i) + Fout (i)]^2$

Un aumento en la MHK conduce a una mayor probabilidad de que aumente el esfuerzo de integración y pruebas del módulo.

- **Diseño de interfaz: Conveniencia de la representación (CR)** para la visión interna de la calidad y facilidad de empleo de la interfaz:

$K$  = transición específica de una entidad de representación a la siguiente al realizar una tarea específica

$CRO$  = costo de la representación óptima

$CRP$  = costo de la representación propuesta. Para asignar costos a cada secuencia de acciones:

## Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

$C$  = costos

$FT$  = frecuencia de transición

$CT$  = costo de transición

$$C = \text{Sum} [F T (k) * CT (k)]$$

Esta relación se da con todas las transiciones de una tarea en particular o conjunto de tareas requeridas para conseguir alguna función de la aplicación.

$$\text{Se calcula } CR = 100 * [(CRO) / (CRP)]$$

Donde:  $CR = 100$  para una representación óptima. Para calcular la representación óptima de una interfaz de usuario:

$N$  = posiciones

$K$  = entidades de representación

$NPD$  = número posible de distribuciones

$$\text{Se calcula } NPD = [N / (K * (N - K))] * K$$

A medida que crece el número de posiciones de representación, el número de distribuciones posibles se hace muy grande.

- **Pruebas** para descubrir errores en el módulo antes de integrarlo en un sistema.

$V$  = volumen de un programa

$NP$  = nivel del programa

$e$  = esfuerzo de la ciencia del software

$$NP = 1 / [(n1/2) * (N 2/n2)]$$

Dando lugar a:  $e = V / NP$

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

Se utiliza en la etapa de pruebas.

- **Código fuente** utilizadas para a partir del número de operadores y operando presentes en el código, valorar la calidad del programa.

Para determinar la longitud global del programa.

$n1$  = número de operadores diferentes que aparecen en el programa

$n2$  = número de operando diferentes que aparecen en el programa

$N1$  = número total de veces que aparece el operador

$N2$  = número total de veces que aparece el operando

$$N = n1 \log_2 n1 + n2 \log_2 n2$$

Para determinar el volumen de programa.

$$V = N \log_2 (n1 + n2)$$

$V$  variará con el lenguaje de programación y representa el volumen de información en bits necesarios para especificar un programa

Para determinar el volumen mínimo para un algoritmo.

$$L = 2/n1 * n2/N 2$$

$L$  debe ser siempre menor de 1.

- **Índice de madurez del software (IMS)** para obtener una indicación de la estabilidad de un producto.

$Mt$  = número de módulos en la versión actual

$Fc$  = número de módulos en la versión actual que se han cambiado

$Fa$  = número de módulos en la versión actual que se han añadido

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

$Fd$  = número de módulos en la versión anterior que se han borrado en la versión actual

$$IMS = [M_t - (F_a + F_c + F_d)] / M_t$$

A medida que el  $IMS$  se aproxima a 1 el producto se empieza a estabilizar.

Las siguientes métricas fueron propuestas para medir la satisfacción del cliente:

- **Tasa de Estabilidad (TE)**, proporciona un indicador de como de bien una aplicación cumplió las expectativas del cliente o usuario.

Se calcula  $TE = 1 - (NC / CCU)$  donde:

$NC$ : número de cambios solicitados durante el primer trimestre, es decir durante los primeros 90 días después de la implementación, o sea, cambios efectuados en el software por parte de los desarrolladores debido a que no pueden por problemas técnicos o lógicos acoplarse en un 100 % a los requisitos que exige el cliente.

$CCU$ : cantidad de Casos de Uso del tamaño de la aplicación, es decir los casos de uso del sistema obtenidos de los requisitos funcionales. El rango de la tasa de estabilidad está entre  $[-\infty; 1]$ , mientras más cerca esté el valor de 1 más estable será la aplicación, debido a que el número de cambios solicitados ha de ser mucho menor que la cantidad de casos de uso con que cuenta la aplicación.

- **Tasa de Defectos (TD)** da un gran aporte en datos cuantitativos a los distintos miembros de los equipos de desarrollo.

Se calcula  $TD = ND / CCU$

$ND$ : número de defectos, es el total de incidencias en las que la aplicación no cumplió las especificaciones o expectativas del cliente, estos pudieron haberse ocasionados por algún descuido de los miembros del equipo o bien por un mal entendimiento de los requisitos.

$CCU$ : cantidad de Casos de Uso, del tamaño de la aplicación, es decir, los casos de uso del

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

sistema obtenidos de los requisitos funcionales.

El intervalo está entre  $[0; \infty]$ , mientras más cercano esté de cero menor es la tasa de defectos, ya que en este caso el número de defectos va a ser menor que el de casos de uso.

- **Cumplimiento de la Planificación de Entrega (CPE)** para considerar los compromisos de entrega del producto final o por etapas cuantificando la desviación en los plazos de entrega en tiempo.

Se calcula  $CPE = (TR * 100) / TE$

*TR*: tiempo real, es el que se dedicó realmente a la entrega del producto. Se expresa en minutos.

*TE*: tiempo estimado o planificado, el que se pactó con el cliente desde su inicio. Se expresa en minutos.

El resultado de esta métrica indica que si es mayor que 100 %: hay retrasos en el plazo de entrega y de ser contrario el resultado, si es menor o igual que 100, entonces se están consiguiendo mejor los resultados que lo planificado. Este cálculo debe realizarse mensual o trimestralmente.

Las métricas de estabilidad y fiabilidad fueron formuladas con la condición de que su rango estuviera entre  $[-\infty; 1]$  y se encontraría el software más estable o fiable al acercarse a uno, debido a que para que algún producto cumpla con estas, no se puede poner en el rango de

$[0; \infty]$ , que mientras más tienda a cero mejor, o sea, que cuando esté en el valor uno, las dos métricas darán idea de que existen estas características para el software.

Estas fueron escogidas principalmente porque evidencian un enfoque ágil, fácilmente adaptables a SXP.

Los indicadores deben ser un instrumento que ayude a mejorar el proceso, producto o proyecto de software, no tiene mucho sentido aplicar indicadores que lejos de ayudar a los desarrolladores constituyan un problema; bien por ser demasiado complejos, porque no se entiendan correctamente los objetivos que persiguen o porque arrojen resultados imprecisos que no puedan ser interpretados por los ingenieros de

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

software.

Para realizar estimaciones debemos establecer una línea base de indicadores que permita a un proyecto sintonizar su proceso de ingeniería del software para eliminar las causas de los defectos que tienen el mayor impacto en el desarrollo del software, es fundamental que una línea base contenga datos recopilados de los proyectos, lo que requiere una investigación histórica de los mismos, la línea base no es más que la recopilación de medidas, métricas e indicadores que guíen el proyecto o el proceso. (Ver Anexo 2)

Es importante entonces que un indicador pueda obtenerse fácilmente, que se entienda por qué y para qué se utiliza, que los cálculos no produzcan resultados ambiguos o en los que existan extrañas combinaciones de unidades, y que la interpretación de valores obtenidos esté acorde a las nociones intuitivas del ingeniero de software.

## 1.5. Herramientas para la Modelación del Diseño de la Propuesta del Sistema

La correcta selección de las herramientas a utilizar en el diseño de la propuesta del sistema, es de gran importancia. Permite garantizar la calidad del mismo y reducir la posibilidad de implementar un software que no cumpla con las expectativas del cliente. A continuación se presentan las seleccionadas en la investigación.

### 1.5.1. Visual Paradigm

Como herramienta CASE<sup>6</sup> a utilizar para el modelado del sistema se utiliza el Visual Paradigm, que permite dibujar todo tipo de diagramas UML<sup>7</sup>. Facilita la organización para el diseño visual de diagramas, ayuda al equipo de desarrollo de software a documentar y cubrir el proceso de modelado, construcción y desarrollo de software, acelerando la contribución individual y a nivel de proyecto de cada equipo de

---

<sup>6</sup> Computer Aided Software Engineering o Ingeniería de Software Asistida por Computación.

<sup>7</sup> Unified Modeling Language

# Capítulo 1: Marco Conceptual de los Indicadores y Herramientas

---

desarrollo, y además soporta varios lenguajes como: Java, C++, PHP<sup>8</sup>. Presenta un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación y está disponible en múltiples plataformas, facilidad que pocas herramientas CASE brindan. [14] Además es la usada en la facultad como alternativa a la utilización de SWL.

## 1.5.2. Lenguaje de Modelado

UML, (Unified Modeling Language) es el lenguaje de modelado de sistemas de software. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Se adoptó UML para la realización de este trabajo, por ser el más utilizado para modelar los artefactos creados durante el proceso de desarrollo de software y es el utilizado en la herramienta CASE Visual Paradigm seleccionada anteriormente.

## 1.5.3. Pencil

Pencil es un proyecto libre, de código abierto y multiplataforma para la creación de diagramas y prototipos GUI<sup>9</sup>, es un plugin para Firefox. Ofrece hasta el momento:

- Conjunto de objetos más habitual: entradas de texto, botones, tablas.
- Multi-página, creación simultánea de varios documentos.
- Edición en pantalla de los elementos de texto. [15]

---

<sup>8</sup> Hypertext Preprocessor

<sup>9</sup> Interfaz Gráfica de Usuario.

## **Capítulo 1: Marco Conceptual de los Indicadores y Herramientas**

---

Con la investigación realizada se llega a la conclusión que la creciente evolución de la industria de software va sumando la necesidad de producir software de calidad, y para lograrlo se tienen en cuenta numerosos factores entre los que se encuentran los indicadores de software, una herramienta indiscutible para ayudar a mantener el control de los procesos y productos durante el desarrollo del software. La calidad es un factor determinante para lograr el éxito del proyecto y los indicadores de software contribuyen al control, seguimiento y mejora de la calidad del proceso de desarrollo de software.

### Capítulo 2: Validación

La validación es el mecanismo usado por el equipo de trabajo para planear, obtener datos, registrar datos, e interpretar datos. Está definida también como la evidencia documentada que provee un alto grado de seguridad de que un proceso específico o un equipo producirán en forma homogénea y reproducible un producto que cumplirá con especificaciones predeterminadas y sus atributos de calidad.

#### 2.1. Técnicas de Validación

Existen en el mundo diversos métodos de validación que se basan en procedimientos y estándares diferentes, estudiados por varios expertos. A continuación se explican brevemente algunos de los que se estudiaron para determinar las que se utilizará en el presente trabajo.

##### Método Delphi

El método Delphi se utiliza cuando no se tienen datos o se dispone de muy pocos acerca del sistema que se está considerando. En este método, se selecciona un grupo de expertos los cuales deben llegar a un consenso en las respuestas que den acerca de una serie de preguntas que se les plantean. En un entorno de simulación los expertos pueden ser los administradores y usuarios del sistema y las cuestiones son acerca del comportamiento del sistema bajo ciertas condiciones de operación. Este método excluye las discusiones cara a cara entre los miembros del grupo. Una crítica de este método es que consume mucho tiempo. [16]

##### Método Multicriterios

Los métodos de Evaluación y Decisión Multicriterios comprenden la selección entre un conjunto de alternativas factibles, la optimización con varias funciones objetivos simultáneamente, un agente decisor y procedimientos de evaluación racionales y consistentes, requiere gran cantidad de tiempo para su realización. En la **Decisión Multicriterios** un elemento clasificador es el número de alternativas a tener en

cuenta en la decisión, que puede ser finito o infinito. Dependiendo de esta situación existen diferentes métodos. Cuando las alternativas tienen un número infinito de valores posibles del problema se llama **Decisión Multiobjetivo**. Por el contrario cuando el número de alternativas es finito se denominan **Decisión Multicriterios Discreta**. Estos problemas son más comunes en la realidad y se utilizan para realizar una evaluación y decisión respecto a problemas que por naturaleza o diseño, admiten un número finito de alternativas de solución. [17]

### **Método Test de Turing**

Alan Turing sugirió este método como un test de inteligencia artificial. En este test, a un experto, o grupo de expertos, se le presentan resúmenes o informes de resultados de ejecución del sistema y del modelo, a los que se les ha dado el mismo formato. Estos informes se reparten aleatoriamente a los ingenieros y administradores del sistema, para ver si son capaces de discernir cuáles son los reales del sistema y cuáles la imitación resultado de la simulación. Si los expertos no son capaces de distinguir entre ambos, se puede concluir que no hay evidencias para considerar inadecuado al modelo. Si descubren diferencias las respuestas sobre lo que encuentran inconsistente se puede utilizar para realizar mejoras en el modelo. [15] Se puede considerar que este método es el inverso al método Delphi. En este método se consulta a los expertos para ver si son capaces de identificar las respuestas del sistema, mientras que en el de Delphi se pregunta a los expertos para que predigan las respuestas del sistema. Requiere mayor tiempo para su ejecución y se debe contar con varios expertos que posean un amplio conocimiento del contenido de la propuesta y a la vez dominar términos estadísticos para realizar comparaciones.

### **Método de Validación Empírica**

La validación Empírica se utiliza para obtener información objetiva sobre la utilidad de los indicadores; estos pueden ser correctos desde un punto de vista formal, pero no tener relevancia práctica para un problema determinado. El estudio empírico resulta necesario para comprobar y entender las implicaciones de las medidas de nuestros productos.

## Capítulo 2: Validación

---

Las investigaciones empíricas se dividen en tres clases principales:

- **Experimentos** se utilizan cuando tenemos el control de la situación y queremos controlar su comportamiento directo, preciso y sistemáticamente. Son apropiados para validar indicadores, de manera que se pueda asegurar que realmente miden lo que se espera de ellos.
- **Encuestas** son investigaciones en las que los sujetos del estudio son una muestra representativa de la población a la que pertenecen.
- **Investigaciones simples** son aquellas en las que no hay aleatoriedad de variables perturbadoras ni representatividad de los sujetos que componen la muestra de estudio. [18]

El proceso experimental de la validación Empírica cuenta con la técnica de validación **Interna y Externa**. La validación **Interna** es un ejercicio teórico que asegura que los indicadores tienen una caracterización numérica apropiada de lo que pretenden medir. Esto es por supuesto un pre-requisito para demostrar la utilidad de dichos indicadores.

La validación **Externa** se lleva a cabo para demostrar con evidencia real que un indicador es útil en el sentido de que esté asociado con alguna característica externa del software tal como la usabilidad, la mantenibilidad.

El proceso que se realiza para el empleo de estos métodos de validaciones consta de tres pasos:

- Primeramente el análisis de la veracidad de los indicadores a emplear.
- La recolección de los datos reales de la muestra en cuestión.
- El cálculo de los indicadores con los valores obtenidos del proceso de recolección y la realización de la valoración correspondiente a los resultados obtenidos.

De esta manera se obtendrá una idea real de la situación que presente la muestra en cuestión en cuanto a los parámetros que se pretenden medir en cada uno de los indicadores a emplear para el desarrollo del proceso de validación.

## Capítulo 2: Validación

---

Se decidió emplear los métodos de validación **Interna y Externa** ya que estos permiten demostrar la utilidad de la propuesta realizada con datos reales obtenidos de una fuente perteneciente a los proyectos hacia donde se encuentra enfocada la propuesta. Este método permite que a partir del valor matemático que se obtiene del indicador se lleguen a conclusiones y soluciones en el proyecto mediante comparaciones que determinen los factores de riesgos. El empleo de este método no requiere gran cantidad de tiempo y además se adecúa a los objetivos propuestos.

Estos métodos serán aplicados en una muestra de 3 proyectos de una población de 17 que integran el grupo UNICORNIOS, que se encuentran dentro de los 5 tipos de proyectos del polo SWL, considerados los más apropiados debido al avance en el desarrollo y por poseer una amplia organización de todo el trabajo realizado.

### 2.2. Proyectos donde se Aplican los Indicadores

De acuerdo a un estudio realizado, los proyectos productivos basados en Software Libre, según sus características se clasifican en 5 grupos:

- **Proyectos de Desarrollo de aplicaciones:** Son aquellos proyectos en los que se desarrollan fundamentalmente aplicaciones de escritorio, así como programas y sistemas computacionales.
- **Proyectos de Desarrollo Web:** Son aquellos proyectos que se dedican totalmente a la fabricación de sitios Web o portales.
- **Proyectos de servicios:** Son aquellos proyectos, que solamente se dedican a prestar servicios a clientes.
- **Proyectos de personalización:** Son aquellos proyectos, que se encargan de personalizar versiones de sistemas operativos libres, para las áreas o instituciones que lo necesiten. La nueva versión del sistema queda adecuada a las necesidades y requerimientos del cliente.

## Capítulo 2: Validación

---

- **Proyectos de investigación:** Son aquellos proyectos, que su producto final no es precisamente un software, sino un plan, realizado luego de una investigación bastante profunda sobre el tema en el que estén trabajando, y que orienta las acciones para alcanzar un objetivo final. [19]

Se seleccionaron entre los proyectos que se desarrollan en el grupo UNICORNIOS, tres para aplicar los indicadores propuestos para la Metodología Ágil SXP. Los proyectos escogidos cuentan con características diferentes, SistClon se clasifica dentro de los proyectos de Desarrollo de Aplicaciones y Desarrollo Web, NovaDesk se desarrolla como proyecto de Desarrollo Web, Servicios e Investigación y Yarey como proyecto de Servicios y Desarrollo Web.

### 2.2.1. Proyecto SistClon

El proyecto SistClon clasifica dentro de los proyectos de desarrollo de aplicaciones y web del grupo UNICORNIOS, éste desarrolla un software para la clonación y administración remota de imágenes de Sistemas Operativos (Linux/Windows). Surge a raíz de la necesidad de reducir tiempo y personal en el mantenimiento del gran número de computadoras en los laboratorios pertenecientes a la Facultad 10, cuenta con 9 integrantes dividido en roles: Gerente, Cliente, Programador, Analista, Diseñador de Interfaz, Probador y Arquitecto. Algunas de las herramientas que usa el proyecto en su desarrollo son: el Code::blocks y el Anjuta, entornos ambos de desarrollo integrado (IDE en inglés) para programas desarrollados en lenguaje C++; los lenguajes Bash, Perl, C/C++ y UML; y como Sistema de Gestión de Base de Datos, el Postgresql.

### 2.2.2. Proyecto NovaDesk

NovaDesk se desarrolla en el grupo UNICORNIOS como un proyecto de desarrollo web, servicios e investigación, se centra en la creación de una aplicación que permita gestionar el soporte técnico e incidencias que se puedan presentar en el sistema operativo Nova. Este proyecto se crea por la carencia de un sistema que permita la gestión de los incidentes que se presentan en este sistema operativo y les dé soporte a las posibles interrogantes de sus usuarios. Está compuesto por 8 roles que son divididos por

los integrantes del grupo de trabajo que lo conforma (Gerente, Cliente, Programador, Analista, Servicios, Diseñador, Probador y Arquitecto). Entre las herramientas utilizadas están: el servidor Apache para los servidores Web; como herramientas CASE, el Visual Paradigm; para gestores de base de datos, MySQL<sup>10</sup>; el PhpMyAdmin como sistemas para administrar Sistema de Gestión de Base de Datos; los lenguajes de programación Web: HTML<sup>11</sup>, JavaScript, CSS<sup>12</sup>, PHP; y como herramientas para el desarrollo web, el Zend Studio.

### 2.2.3. Proyecto YAREY

El proyecto Yarey se desarrolla en el grupo UNICORNIOS como un proyecto de desarrollo de aplicación y servicios, surge por la necesidad de brindar un servicio de render distribuido con SWL. Primeramente este servicio era para brindarlo al proyecto Meñique, producción audiovisual del ICAIC<sup>13</sup>, pero actualmente se espera proveer servicios a otras entidades, tanto nacionales como internacionales. Su grupo de trabajo lo conforman 3 miembros que se dividen los roles: Gerente, Cliente, Programador, Analista, Diseñador, Probador y Arquitecto. Para el desarrollo del render distribuido que se quiere lograr utilizan como herramientas de IDE, el Scite; y los lenguajes: C++, Python y Bash.

## 2.3. Validación de los Indicadores de SXP

A continuación se describe el proceso llevado a cabo para la validación de los indicadores. Este proceso se realizó con una muestra real de datos de los proyectos, permitiendo la obtención de resultados que ayudarán a la mejora de la calidad de los proyectos que utilizan SXP.

### 2.3.1. Validación en el Proyecto SistClon

SistClon es un producto que se encuentra terminado.

---

<sup>10</sup> Structured Query Language

<sup>11</sup> HyperText Mark-up Language

<sup>12</sup> Cascading Style Sheets

<sup>13</sup> Instituto Cubano del Arte e Industria Cinematográficos

Validación de los indicadores de la etapa de Definición - Planificación.

- **Productividad**

**Fórmula:**

$$E = P * T$$

**Datos:**

La cantidad de tiempo se halla sumando los puntos reales de las historias de usuarios de cada iteración, y además se le agrega el tiempo de capacitación al equipo que es de 8 semanas debido a las diferentes habilidades de los miembros del equipo de trabajo. El tiempo resultante está en semanas por lo que se lleva a mes.

Preparación de los miembros del proyecto = 8 semanas

Iteración1 (HU\_6 implementadas) = 7 semanas

Iteración2 (HU\_10 implementadas) = 10 semanas

Iteración3 (HU\_3 implementadas) = 5 semanas

Iteración4 (HU\_4 implementadas)= 4 semanas

Iteración5 (HU\_5 implementadas) = 5 semanas

Iteración6 (HU\_10 implementadas) = 13 semanas

Iteración7 (HU\_2 implementadas) = 2 semanas

Pruebas de Aceptación = 12 semanas

En total son 66 semanas que para llevarlo a mes se divide entre 4.

$$T = 16,5 \text{ mes}$$

**Sustitución de los datos:**

$$E = 9 * 16,5 = 148,5$$

El esfuerzo real del proyecto es de 148,5 persona / mes.

Con este valor se obtiene la información del avance de la productividad del proyecto, al compararlo con el de etapas anteriores.

- **Duración del proyecto**

**Fórmula:**

$$EP = DR / DE$$

**Datos:**

$$DR = 66 \text{ semanas}$$

La Duración Estimada del proyecto es de:

Planificación = 4 semanas

Iteración1 (HU\_6 implementadas) = 8 semanas

Iteración2 (HU\_10 implementadas) = 10 semanas

Iteración3 (HU\_3 implementadas) = 6 semanas

Iteración4 (HU\_4 implementadas) = 5 semanas

Iteración5 (HU\_5 implementadas) = 7 semanas

Iteración6 (HU\_10 implementadas) = 30 semanas

Iteración7 (HU\_2 implementadas) = 4 semanas

Pruebas de Aceptación = 8 semanas

Para un total de 82 semanas

### **Sustitución de los datos:**

$$EP = 66 / 82$$

$$EP = 0,81 \text{ semanas}$$

La estimación de plazo es de 0,81 semanas.

Este resultado le permite al líder del proyecto tener un conocimiento de la estimación real que se está realizando para facilitar la toma de decisiones en cuanto a este parámetro para mejorar o mantener la calidad.

- **Desviación existente del tiempo real con respecto al planificado.**

### **Fórmula:**

$$TT = TD + TP$$

$$DT = TT - TPI$$

### **Datos:**

$$TT = 66 \text{ semanas}$$

TP = 0, puesto que el proyecto culminó y no le quedan tareas por realizar.

$$TPI = 82 \text{ semanas}$$

### **Sustitución de los datos:**

$$TT = 66 + 0$$

$$TT = 66 \text{ semanas}$$

$$DT = 66 - 82$$

$$DT = - 16$$

Como la desviación total es menor que 0 se concluye que el tiempo en que se realizaron las

actividades fue menor al planificado, por tanto, se dice que hay una holgura de tiempo que puede ser usada más adelante en otras actividades o si existiera algún retraso.

- **Riesgo**

**Fórmula:**

$$R = I * P$$

**Datos:**

En la lista de riesgo del proyecto SistClon contamos con 10 riesgos. Para priorizar los riesgos se hallan la exposición al riesgo(R) de cada uno.

I = impacto sobre el proyecto en una escala de 1 a 5 donde:

**Tabla 2.1. Escala de impacto de los riesgos**

Impacto	Escala
Retrasos	5
Tiempo	4
Desarrollo	3
Producto	2
Equipo de trabajo	1

- Los **retrasos** son aquellos problemas que ocasionan un desagrado por parte del equipo, obteniendo así un retraso de las actividades, de los trabajos para la terminación del producto y en la entrega del mismo.
- El **tiempo** son aquellos obstáculos que hacen que la duración del proyecto sea mayor que el tiempo planificado.
- El **desarrollo** son las dificultades que originan que avance del proyecto no sea el más óptimo.

## Capítulo 2: Validación

- El **producto** son los inconvenientes que causan un producto sin la mayor calidad.
- El **equipo de trabajo** está identificado por el atraso en el trabajo del personal y las dudas en la calidad del mismo.

P = probabilidad de que ocurra donde:

**Tabla 2.2. Escala de probabilidad**

Probabilidad	Escala
0,9	Alta
0,5	Media
0,1	Baja

**Tabla 2.3. Sustitución de datos**

No	Riesgos	I	P	R
1	No existen computadoras para todo el personal	7	0,9	6,3
2	Programar un sistema de archivos universal	3	0,9	2,7
3	Ausencia de un programador, analista, documentador o un diseñador	5	0,5	2,5
4	Rotura de una máquina donde se encuentra información importante	5	0,9	4,5
5	Enfermedad de uno de los miembros del equipo	5	0,9	4,5
6	No existencia de un especialista capacitado para responsabilizarse con algunas tareas de prioridad para el proyecto	2	0,9	1,8
7	Afectación del fluido eléctrico en el laboratorio de producción	5	0,1	0,5
8	Inconformidad en la arquitectura y diseño del producto	1	0,1	0,1
9	No existen sillas suficientes para el personal	1	0,5	0,5

## Capítulo 2: Validación

---

10	Cambio del Sistema operativo en el que el personal acostumbraba a trabajar	3	0,9	2,7
----	--	---	-----	-----

Con el indicador de riesgo obtendremos un orden de mitigación de los riesgos, de esta forma el proyecto podrá ir eliminando los riesgos de mayor dificultad para el desarrollo del producto.

Orden de Mitigación de los riesgos:

Riesgo 1

Riesgo 4 y Riesgo 5

Riesgo 2 y Riesgo 10

Riesgo 3

Riesgo 6

Riesgo 7 y Riesgo 9

Riesgo 8

- **Cumplimiento de Compromiso**

Para calcular el cumplimiento de compromiso existen dos variantes, la variante 1 es que depende de las actividades a realizar y la variante 2 es que depende de la asignación de recursos. En este caso se utiliza la variante 1.

**Fórmula:**

$$CA_i = C_{An} - C_{Ae}$$

**Datos:**

$C_{An}$  serán la cantidad de historias de usuarios que se necesitan para realizar el producto.

$C_{Ae}$  serán la cantidad de historias de usuarios, que son las actividades realizadas hasta la

entrega.

$$CAn = 40$$

$$CAe = 40$$

**Sustitución de los datos:**

$$CAi = 40 - 40$$

$$CAi = 0$$

Como la cantidad de actividades incumplidas (CAi) es 0 el compromiso se cumple debido a que todas las actividades necesarias para su cumplimiento fueron realizadas.

Es un proyecto que realiza un producto cubano y no es financiado debido a que se desarrolla en los proyectos productivos de la UCI por lo que no se validan los siguientes indicadores:

- **Desviación del Costo**

**Fórmula:**

$$IC = PP + CR$$

$$CD = IC - CTP$$

- **Estimación Exacta del gasto.**

**Fórmula:**

$$PG = (HF * 100) / PPT$$

- **Calidad de Especificación**

**Fórmula:**

$$Nr = Nf + Nnf$$

**Sustitución de datos:**

## Capítulo 2: Validación

$$Nr = 40 + 6$$

$$Nr = 46$$

Para determinar la ausencia de ambigüedad se calcula la Especificación de Requisitos (ER)

### **Fórmula:**

$$ER = Nvi / Nr$$

### **Sustitución de datos:**

$$ER = 40 / 46$$

$$ER = 0.87$$

**Tabla 2.4. Escala de evaluación de los resultados de la Calidad de Especificación de Requisitos**

	Valores	Evaluación
<b>Calidad de Especificación de Requisitos</b>  <b><math>0 \leq ER \leq 1</math></b> <b>Valor Ideal 0.75</b>	<b><math>0 \leq ER \leq 0.5</math></b>	<b>Mal</b>
	<b><math>0.5 \leq ER \leq 0.75</math></b>	<b>Regular</b>
	<b><math>0.75 \leq ER \leq 1</math></b>	<b>Bien</b>

Debido a que la especificación de los requisitos es mayor que 0.75 se determina que la ambigüedad en ellos es poca, por tanto la calidad de especificación de requisitos está evaluada de Bien.

### **Fórmula:**

$$CRF = Un / (Ni * Ns)$$

### **Sustitución de los datos:**

$$CRF = 40 / (40 * 40)$$

$$CRF = 0,025$$

La compleción de los requisitos funcionales es de 0,025.

Para medir el porcentaje de funciones necesarias que se han especificado para un sistema se determina el Grado de Validación de los Requisitos (GVR).

**Fórmula:**

$$GVR = Nc / (Nc + Nnu)$$

**Sustitución de los datos:**

$$GVR = 36 / (36 + 10)$$

$$GVR = 0,78 \text{ convertido a por ciento es } 78 \%$$

Se obtiene un grado de validación de los requisitos de 78%.

Es necesario destacar la importancia del empleo de este indicador en el momento del levantamiento de requisitos, es primordial el conocimiento de los errores que se cometen en esta fase.

Validación de los indicadores para la etapa de desarrollo del producto.

- **Basada en la función**

**Fórmula:**

$$PF = \text{cuenta total} * [0,65 + 0,1 * \Sigma [Fi]]$$

**Datos:**

Cuenta\_Total = a la suma de todos los parámetros de medición obtenidos.

Parámetros de medición

## Capítulo 2: Validación

---

- **Número de entrada de usuarios** es cada entrada del usuario que proporciona al software diferentes datos orientados a la aplicación.
- **Número de salida de usuarios** es cada salida que proporciona al usuario información orientada a la aplicación. Se refiere a informes, pantallas con mensajes de error. Los elementos de datos individuales dentro de un informe se cuentan por separados.
- **Número de consultas o peticiones al usuario** una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva. Se cuenta cada petición por separado.
- **Número de archivos** es cada archivo maestro lógico, o sea una agrupación lógica de datos que puede ser una parte en una gran base de datos o un archivo independiente.
- **Número de interfaces externas** son todas las interfaces legibles por la máquina.

Fi ( $i = 1$  a  $i = 14$ ) = valores de ajuste de complejidad en una escala de 0 a 5.

### Preguntas para los valores de ajustes

1. ¿Requiere el sistema copias de seguridad y recuperación fiables?
2. ¿Se requiere comunicación de datos?
3. ¿Existen funciones de procesamiento distribuido?
4. ¿Es crítico el rendimiento del sistema?
5. ¿Será ejecutado el sistema en un entorno operativo y frecuentemente utilizado?
6. ¿Requiere el sistema entrada de datos interactivo?
7. ¿Requiere la entrada de datos interactivo que las transiciones de entrada se lleven a cabo sobre múltiples o variadas operaciones?
8. ¿Se actualizan los archivos maestros en forma interactiva?

## Capítulo 2: Validación

9. ¿Son complejas las entradas, las salidas, los archivos o peticiones?
10. ¿Es complejo el procesamiento interno?
11. ¿Se ha diseñado el código para ser reutilizable?
12. ¿Están incluidos en el diseño la conversión y la instalación?
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

Tabla 2.5. Escala de Ajustes

0	1	2	3	4
Sin influencia	Incidental	Moderado	Medio	Significativo

- **Sin influencia (0):** El sistema no contempla este atributo.
- **Incidental (1):** La influencia de este atributo es muy poco significativa.
- **Moderado (2):** El sistema contempla este atributo y su influencia, aunque pequeña, ha de ser considerada.
- **Medio (3):** La importancia de este atributo debe ser tomada en cuenta, aunque no es fundamental.
- **Significativo (4):** Este atributo tiene una gran importancia para el Sistema.
- **Esencial (5):** Este atributo es esencial para el Sistema y ha de ser tenido en cuenta a la hora del diseño.

### Sustitución de datos:

$$\Sigma F_i = 4 + 5 + 4 + 4 + 5 + 3 + 5 + 4 + 4 + 5 + 4 + 3 + 4 + 5 = 47$$

SistClon cuenta con 3 entradas de usuarios: Cambiar IP, Configurar Servidor de Imágenes y

## Capítulo 2: Validación

Configurar Acceso a Base de Datos. Contiene 1 salidas de usuarios: listado de las computadoras conectadas a la red. Cuenta con 2 consultas: Mensaje de error para el servidor de imágenes y mensaje de error para la base de datos. Posee un archivo (Base de Datos) y 6 interfaces externas: principal, para cambiar IP, configurar servidor de imágenes, configurar acceso a base de datos, conectar y desconectar computadoras.

El factor de ponderación puede ser simple, medio o complejo, según determine el analista.

**Tabla 2.6. Factor de ponderación de SistClon**

Parámetros de Medición	Cuenta	Simple	Media	Compleja	Total
Número de entradas del usuario	3	3	4	6	9
Número de salidas del usuario	1	4	5	7	4
Número de consultas del usuario	2	3	4	6	6
Número de archivos	1	7	10	15	7
Número de interfaces externas	6	5	7	10	30
Cuenta Total					56

$$PF = 56 * [0,65 + 0,1 * 47]$$

$$PF = 299$$

Basándose en el valor previsto de PF obtenido del modelo de análisis, el equipo del proyecto puede estimar que el tamaño global de implementación de SistClon se supone que sea de 299 puntos de función aproximadamente.

- **Henry y Kafura**

### **Fórmula:**

$$MHK = longitud (i) * [Fint (i) + Fout (i)]^2$$

Fint (i): concentración del módulo i, es decir el 50 % de las funcionalidades críticas.

## Capítulo 2: Validación

Fout(i): expansión del módulo i, o sea las funcionalidades críticas del módulo.

Longitud(i): número de sentencias del lenguaje de programación en el módulo i.

La aplicación no posee definida las funcionalidades por módulos para determinar la estructura arquitectónica y la eficiencia de los mismos, por lo que se determina para el software completo.

**Tabla 2.7. Sustitución de los datos:**

Módulos	longitud (i)	Fint (i)	Fout (i)	MHK
Software	37	13	26	56277

Con el cálculo de este indicador se determina que el software es bastante complejo y por tanto requiere un mayor esfuerzo de integración y pruebas.

- **Diseño de la Interfaz**

**Fórmula:**

$$CR = 100 * [CRO / CRP]$$

$$C = \text{sum} [ FT (k) * CT(k) ]$$

$$NPD = [N! / K! * (N - K) !] * K!$$

Según las características de los indicadores estos deben ser simple, medibles, calculables, precisos y deben contar con toda la información documentada. No tiene sentido aplicar un indicador que lejos de ayudar al equipo de trabajo, constituya un problema por ser demasiado complejo y porque no se entienda correctamente los objetivos que persigue, debido a que arroja resultados imprecisos que no puedan ser interpretados por los miembros del proyecto, como es el caso; por lo tanto, se propone lo siguiente :

El diseño de la interfaz está relacionado con el uso del espacio y la manera en que la información está dispuesta en la pantalla. Su propósito es hacer que las interacciones entre el usuario y el

## Capítulo 2: Validación

---

software sean más fáciles mediante la presentación de la información de una manera adecuada. La disposición de la pantalla es extremadamente importante en términos de motivar al usuario y estructurar la información. Los siguientes aspectos están incluidos: organización del texto en la interfaz, los gráficos y el uso del color. [20]

### **Fórmula:**

$$DI = \left[ \left( \sum_{j=1}^m \sum_{i=1}^n e_{ij} * p \right) * 100 \right] / n * m$$

### **Datos:**

DI: Valor total del Diseño de Interfaz.

j: Número de columnas.

i: Números de filas.

m: Cantidad de niveles.

n: Cantidad de preguntas.

eij: Elemento de la fila i y columna j.

p: Valor del nivel.

**Escala:**  $60 \leq DI \leq 100$ .

**Tabla 2.8. Escala de niveles**

1	2	3	4	5
Muy Bajo	Básico	Normal	Alto	Excelente

**Nivel 1:** Valor mínimo. Se le asigna este valor a la métrica cuando el producto no presenta el parámetro que se evalúa o presenta un grado de satisfacción muy bajo.

**Nivel 2:** Se le asigna este valor a la métrica cuando el producto presenta el parámetro a evaluar pero con un grado de satisfacción bajo.

## Capítulo 2: Validación

**Nivel 3:** Se le asigna este valor a la métrica cuando el producto presenta el parámetro a evaluar con un grado de satisfacción medio.

**Nivel 4:** Se le asigna este valor a la métrica cuando el producto presenta el parámetro a evaluar con un grado de satisfacción alto.

**Nivel 5:** Valor máximo. Se le asigna este valor a la métrica cuando el producto cumple con el parámetro establecido y presenta un grado de satisfacción muy alto.

Para cada nivel las preguntas toman valores entre 0 y 1:

**Valor 0:** Cuando no presenta calidad o la categoría no es satisfecha.

**Valor 1:** Cuando presenta calidad o la categoría es satisfecha.

**Tabla 2.9. Valores del diseño de interfaz de SistClon**

Nº.	Diseño de la interfaz.	1	2	3	4	5
1	El texto está en correspondencia con el contenido.	0	0	0	0	1
2	El texto es claro y de fácil comprensión.	0	0	1	0	0
3	Cuando contiene palabras rebuscadas le ayuda al usuario a entenderlas.	0	1	0	0	0
4	El texto permite leer de manera lógica y sistemática.	0	0	0	1	0
5	Es coherente.	0	0	0	1	0
6	Los caracteres son claros y atractivos.	0	0	0	1	0
7	El color de la letra es adecuado para la lectura.	0	0	0	0	1
8	El tamaño es el adecuado.	0	0	0	1	0
9	Presenta palabras calientes con sus significados.	1	0	0	0	0
10	La estructura del contenido está de forma correcta.	0	0	0	1	0
11	El fondo de la pantalla es el correcto.	0	0	1	0	0
12	Los colores utilizados van acorde con el tema.	0	0	1	0	0
13	Presenta gráficos que ayudan a entender un tema en particular.	0	0	0	1	0
14	Las ilustraciones e imágenes ayudan al usuario a entender de lo que trata el tema.	0	0	0	1	0

## Capítulo 2: Validación

15	Las ilustraciones e imágenes están acorde con el tema.	0	0	0	0	1
16	Los gráficos ayudan a entender el contenido expuesto.	0	0	1	0	0
17	Presenta colores claros que ayudan al usuario a no desviarse del contenido.	0	0	0	0	1
18	La combinación de colores es variada, agradable.	0	0	1	0	0
19	El contraste entre: los colores de fondo, la letra, gráficos e ilustraciones es adecuado.	0	0	0	1	0
20	Hay suficiente contraste entre los colores del texto o ilustraciones y el color del fondo.	0	0	0	1	0

### **Sustitución de los datos:**

$$DI = 740 / 100$$

$$DI = 74$$

El sistema SistClon cuenta con una buena calidad y facilidad de empleo de la interfaz de usuario, se encuentra en la escala  $60 \leq DI \leq 100$ .

- **Calidad del programa**

### **Fórmula:**

$$N = n1 \log_2 n1 + n2 \log_2 n2$$

Para determinar el Volumen del programa

$$V = N \log_2 (n1 + n2)$$

Para determinar el volumen mínimo para un algoritmo

$$L = 2 / n1 * n2 / N2$$

**Tabla 2.10. Datos de los operadores**

## Capítulo 2: Validación

n1	Operador	N1
1	Fin de sentencia	135
2	,	76
3	=	74
4	if()	25
5	printf	8
6	+	8
7	for	4
8	while	3
9	<=	3
10	Fin de programa	3
11		2
12	>=	1
Total		317

Tabla 2.11. Datos de los operandos

n2	Operando	N2
1	NULL	9
2	array	8
3	Error e	7
4	argv	5
5	argc	4
6	opcion	4
7	true	3
8	false	3
9	error1	2
10	error2	2

Total		47
-------	--	----

### **Sustitución de los datos:**

$$N = 12 \log_2 (12) + 10 \log_2 (10)$$

$$N = 77$$

Volumen del programa

$$V = 77 \log_2 (12 + 10)$$

$$V = 343 \text{ bit}$$

Volumen mínimo

$$L = 2 / 12 * 10 / 47$$

$$L = 0,036$$

- **Errores en la etapa de prueba**

### **Fórmula:**

$$e = V / NP$$

$$NP = 1 / [(n_1 / 2) * (n_2 / 2)]$$

### **Sustitución de los datos:**

$$NP = 1 / [(12/2) * (10 / 2)]$$

$$NP = 1/30$$

$$NP = 0,033$$

$$V = 343$$

Por tanto:

$$e = 343 / 0,033$$

$$e = 10394$$

El esfuerzo de la ciencia del software es de 10394.

Validación de los indicadores de la etapa de entrega del producto

- **Tasa de Estabilidad**

**Fórmula:**

$$TE = 1 - (Nc / CHU)$$

**Datos:**

$Nc = 0$  puesto que en el primer trimestre el proyecto no contaba con un cliente que adaptara la aplicación a los requisitos que exigía mediante cambios en esta etapa.

$$CHU = 40$$

**Sustitución de los datos:**

$$TE = 1 - (0 / 40)$$

$$TE = 1$$

**Tabla 2.12. Escala de estabilidad**

	Valores	Evaluación
<b>Tasa de Estabilidad</b> $0 \leq TE \leq 1$ Valor Ideal 0.75	$0 \leq TE \leq 0.5$	Mal
	$0.5 \leq TE \leq 0.75$	Regular

## Capítulo 2: Validación

---

	$0.75 \leq TE \leq 1$	Bien
--	-----------------------	------

Como la tasa de estabilidad es 1 la aplicación es totalmente estable, pues, no se realizaron cambios solicitados por el cliente, siendo así la cantidad de historias de usuarios con que cuenta la aplicación mayor que los cambios efectuados.

- **Tasa de defectos (TD)**

**Fórmula:**

$$TD = ND / CHU$$

ND: Número de defectos o total de incidencias en las que la aplicación no cumplió las especificaciones o expectativas del cliente.

CHU: cantidad de historias de usuarios del tamaño de la aplicación, es decir, las historias de usuarios del sistema obtenidos de los requisitos funcionales.

**Sustitución de los datos:**

$$TD = 0 / 40$$

$$TD = 0$$

Debido a que TD es 0, se determinó que hay pocos defectos en el sistema, ya que el número de defectos es menor que el de las HU.

- **Cumplimiento de la Planificación de Entrega.**

**Fórmula:**

$$CPE = (TR * 100) / TE$$

**Sustitución de los datos:**

$$CPE = (525600 * 100) / 525600$$

$$CPE = 100$$

El resultado de esta métrica indica que se está coincidiendo con un mejor resultado que lo planificado para la entrega del producto.

Validación de los indicadores de la etapa de mantenimiento.

- **Estabilidad del producto**

**Fórmula:**

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

Donde:

Mt: Número de módulos en la versión actual

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

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

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

Sustitución de los datos:

$$IMS = [4 - (1 + 0 + 0)] / 4$$

$$IMS = 3 / 4$$

$$IMS = 0,75$$

**Tabla 2.13. Escala de índice de madurez del software**

Índice de madurez del software $0 \leq IMS \leq 1$	Valores	Evaluación
	$0 \leq IMS \leq 0.5$	Mal
	$0.5 \leq IMS \leq 0.75$	Regular

Valor Ideal 0.75	$0.75 \leq IMS \leq 1$	Bien
------------------	------------------------	------

Se concluye que el producto hasta el momento se encuentra en el intervalo de  $0.75 \leq IMS \leq 1$ , o sea bastante estable y mientras más se aproxime a 1 mayor estabilidad alcanzará.

### 2.3.2. Validación en el Proyecto NovaDesk

El proyecto NovaDesk cuenta con una versión del producto ya terminada y se encuentra actualmente en la elaboración de la segunda. La validación se realiza con la documentación de la versión 0.1, debido a que en la 2.0 el proyecto está cambiando de tecnología y por tanto el expediente del mismo.

Validación de los indicadores de la etapa de Definición - Planificación.

- **Productividad**

**Fórmula:**

$$E = P * T$$

**Datos:**

P = 6 personas

Preparación de los miembros del proyecto = 8 semanas

Iteración1 (HU\_1 implementadas) = 1 semanas

Iteración2 (HU\_1 implementadas) = 1 semanas

Iteración3 (HU\_1 implementadas) = 1 semanas

Iteración4 (HU\_1 implementadas)= 1 semanas

Iteración5 (HU\_1 implementadas) = 3 semanas

Iteración6 (HU\_1 implementadas) = 2 semanas

Iteración7 (HU\_1 implementadas) = 2 semanas

Iteración8 (HU\_2 implementadas) = 7 semanas

Iteración9 (HU\_1 implementadas) = 2 semanas

Pruebas de Aceptación = 1 semana

La duración real del proyecto es de 29 semanas, que llevadas a meses es 7 meses.

### **Sustitución de los datos:**

$$E = 6 * 7$$

$$E = 42 \text{ persona/mes}$$

La productividad del proyecto es 42 persona / mes.

- **Duración del proyecto**

### **Fórmula:**

$$EP = DR / DE$$

### **Datos:**

$$DR = 7 \text{ meses.}$$

La Duración Estimada del proyecto es de:

Preparación de los miembros del proyecto = 8 semanas

Iteración1 (HU\_1 implementadas) = 1 semanas

Iteración2 (HU\_1 implementadas) = 2 semanas

Iteración3 (HU\_1 implementadas) = 1 semanas

Iteración4 (HU\_1 implementadas)= 1 semanas

Iteración5 (HU\_1 implementadas) = 3 semanas

Iteración6 (HU\_1 implementadas) = 2 semanas

Iteración7 (HU\_1 implementadas) = 3 semanas

Iteración8 (HU\_2 implementadas) = 7 semanas

Iteración9 (HU\_1 implementadas) = 2 semanas

Pruebas de Aceptación = 1 semana

La duración estimada es de 31 semanas, es decir 8 meses.

### **Sustitución de los datos:**

$$EP = 7 / 8$$

$$EP = 0.87$$

La estimación de plazo es de 0.87.

- **Desviación existente del tiempo real con respecto al planificado.**

### **Fórmula:**

$$DT = TT - TPI$$

$$TT = TD + TP$$

### **Sustitución de los datos:**

$$TT = 7 + 0$$

$$TT = 7$$

$$DT = 7 - 8$$

$$DT = - 1$$

## Capítulo 2: Validación

La desviación existente del tiempo real con respecto al planificado es menor que 0, por tanto se dice que el tiempo en que se cumplieron las actividades es menor al planificado, dedicando el tiempo restante a otras tareas o de reserva para casos de retrasos.

- **Riesgo**

En la lista de riesgo del proyecto NovaDesk contamos con 12 riesgos. Para realizar el orden de mitigación de estos, se halla la exposición al riesgo (R) de cada uno.

**Fórmula:**

$$R = I * P$$

**Tabla 2.14. Sustitución de datos**

No	Riesgos	I	P	R
1	Falta de máquinas	7	0,9	6,3
2	Actividades políticas que surgen a última hora(marchas, tribunas)	5	0,9	4,5
3	Problemas familiares	5	0,5	2,5
4	Falta de preparación o conocimiento de las herramientas de trabajo.	5	0,5	2,5
5	Riesgo de estimación en la elaboración del software.	5	0,5	2,5
6	Ausencia de personal	5	0,5	2,5
7	Rotura de la PC	5	0,9	4,5
8	Fluido eléctrico	5	0,1	0,5
9	Inconformidad de la Arquitectura	1	0,1	0,1
10	El cliente no tiene estandarizado su proceso	3	0,9	2,7
11	Exámenes de Nivel	7	0,5	3,5
12	Incumplimiento de las tareas del proyecto.	5	0,9	4,5

Entonces el Orden de Mitigación de los riesgos del proyecto es:

Riesgo 1

Riesgo 2, Riesgo 7 y Riesgo 12

Riesgo 11

Riesgo 10

Riesgo 3, Riesgo 4, Riesgo 5 y Riesgo 6

Riesgo 8

Riesgo 9

- **Cumplimiento de Compromiso**

El compromiso a cumplir es la revisión del producto utilizando la variante 1.

**Fórmula:**

$$CA_i = C_{an} - C_{ae}$$

**Sustitución de datos:**

$$CA_i = 17 - 17$$

$$CA_i = 0$$

Como la cantidad de actividades incumplidas ( $CA_i$ ) 0, se dice que no hubo un incumplimiento en las actividades y se cumple con el compromiso.

- **Desviación del Costo**

El proyecto no es financiado debido a que se desarrolla en la producción de la UCI, por tanto no se aplican este indicador ni el siguiente.

- **Estimación Exacta del gasto.**

- **Calidad de Especificación**

**Fórmula:**

$$N_r = N_f + N_{nf}$$

**Sustitución de datos:**

$$N_r = 17 + 2$$

$$N_r = 19$$

Para determinar la ausencia de ambigüedad se calcula la Especificación de Requisitos (ER).

**Fórmula:**

$$ER = N_{vi} / N_r$$

**Sustitución de datos:**

$$ER = 10 / 19$$

$$ER = 0,53$$

Debido a que la especificación de los requisitos es de 0,53, se halla en el intervalo de  $0.5 \leq ER \leq 0.75$ , se concluye que la probabilidad de ambigüedad en ellos es regular.

Para determinar la compleción de los requisitos funcionales

$$CRF = U_n / (N_i * N_s)$$

$$CRF = 17 / (17 * 17)$$

$$CRF = 0,058$$

La compleción de los requisitos funcionales es de 0,058.

Para medir el porcentaje de funciones necesarias que se han especificado para un sistema se determina el Grado de Validación de los Requisitos (GVR).

$$GVR = N_c / (N_c + N_{nu})$$

$$GVR = 17 / (17 + 2)$$

## Capítulo 2: Validación

GVR = 0,89 convertido a por ciento es 89 %

Se obtiene un grado de validación de los requisitos de un 89%.

Validación de los indicadores para la etapa de desarrollo del producto.

- **Métrica basada en la función**

**Fórmula:**

$$PF = \text{cuenta total} * [0,65 + 0,1 * \Sigma [Fi]]$$

**Sustitución de datos:**

$$\Sigma Fi = 5 + 3 + 2 + 4 + 4 + 5 + 3 + 5 + 1 + 4 + 5 + 2 + 4 + 5$$

$$Fi = 52$$

NovaDesk cuenta con 7 entradas de usuarios: Autenticación, Crear Reportes, Editar Reportes, Buscar Reportes, Preguntas, Editar perfil del Usuario y Sala de Chat. Tiene 6 salidas de usuarios: Listas de Reportes, Reportes Cerrados, Respuestas, Exportar a .pdf, Exportar a .csv y Sala de Chat. Posee un archivo que es la base de datos y 7 interfaces externas: Autenticación, Crear Reportes, Editar Reportes, Reportes Cerrados, Buscar Reportes, Editar Perfil y Sala de Chat.

**Tabla 2.15. Factor de Ponderación de NovaDesk**

Parámetros de Medición	Cuenta	Simple	Media	Compleja	Total
Número de entradas del usuario	7	3	4	6	21
Número de salidas del usuario	6	4	5	7	24
Número de consultas del usuario	0	3	4	6	0
Número de archivos	1	7	10	15	7
Número de interfaces externas	7	5	7	10	35
Cuenta Total					87

## Capítulo 2: Validación

$$PF = 87 * [0,65 + 0,1 * 52]$$

$$PF = 1640$$

Basándose en el Punto de Función obtenido se puede decir que el tamaño del sistema es de aproximadamente 1640 puntos de función.

- **Henry y Kafura**

**Fórmula:**

$$MHK = longitud (i) * [Fint (i) + Fout (i)]^2$$

Esta aplicación no tiene definidas las funcionalidades por módulos para determinar la estructura arquitectónica y la eficiencia de los mismos, por lo que se determina del software completo.

**Tabla 2.16. Sustitución de los datos**

Módulos	longitud (i)	Fint (i)	Fout (i)	MHK
Software	447	6	12	144828

Se concluye que es un software bastante complejo y por tanto requiere de un gran esfuerzo de integración y pruebas.

- **Diseño de la Interfaz**

**Fórmula:**

$$5 \ 20$$

$$DI = [(\sum_{j=1}^n (\sum_{i=1}^m e_{ij} * p)) * 100] / n * m$$

$$j=1 \ i=1$$

$$\text{Rango: } 60 \leq DI \leq 100$$

**Tabla 2.17. Valores del diseño de interfaz de NovaDesk**

## Capítulo 2: Validación

Nº.	Diseño de la interfaz.	1	2	3	4	5
1	El texto está en correspondencia con el contenido.	0	0	0	0	1
2	El texto es claro y de fácil comprensión.	0	0	0	0	1
3	Cuando contiene palabras rebuscadas, le ayuda al usuario a entenderlas.	0	0	0	0	1
4	El texto permite leer de manera lógica y sistemática.	0	0	0	1	0
5	Es coherente.	0	0	0	1	0
6	Los caracteres son claros y atractivos.	0	0	0	1	0
7	El color de la letra es adecuado para la lectura.	0	0	0	0	1
8	El tamaño es el adecuado.	0	0	0	1	0
9	Presenta palabras calientes con sus significados.	0	1	0	0	0
10	La estructura del contenido está de forma correcta.	0	0	0	1	0
11	El fondo de la pantalla es el correcto.	0	0	0	1	0
12	Los colores utilizados van acorde con el tema.	0	0	0	1	0
13	Presenta gráficos que ayudan a entender un tema en particular.	0	0	0	0	0
14	Las ilustraciones e imágenes ayudan al usuario a entender de lo que trata el tema.	0	0	0	0	0
15	Las ilustraciones e imágenes están acorde con el tema.	0	0	1	0	0
16	Los gráficos ayudan a entender el contenido expuesto.	0	0	0	0	0
17	Presenta colores claros que ayudan al usuario a no desviarse del contenido.	0	0	0	0	1
18	La combinación de colores es variada, agradable.	0	0	0	1	0
19	El contraste entre: los colores de fondo, la letra, gráficos e ilustraciones es adecuado.	0	0	0	0	1
20	Hay suficiente contraste entre los colores del texto o ilustraciones y el color del fondo.	0	0	1	0	0

### Sustitución de los datos:

$$DI = 70 * 100 / 20 * 5$$

$$DI = 70$$

El sistema cuenta con buena calidad y facilidad de empleo de la interfaz de usuario puesto que está comprendido en el rango de  $60 \leq DI \leq 100$ .

- **Calidad del programa**

**Fórmula:**

$$N = n1 \log_2 n1 + n2 \log_2 n2$$

Para determinar el Volumen del programa

$$V = N \log_2 (n1 + n2)$$

Para determinar el volumen mínimo para un algoritmo

$$L = 2 / n1 * n2 / N2$$

**Tabla 2.18. Datos de los operadores**

n1	Operador	N1
1	Fin de sentencia	5000
2	=	4086
3	+	243
4	if()	213
5	,	150
6	>=	125
7	<=	115
8	for	98
9	&&	63
10	while	50
11		40

## Capítulo 2: Validación

12	Echo	30
13	Fin de programa	5
Total		10218

Tabla 2.19. Datos de los operandos

n2	Operando	N2
1	array	224
2	options	65
3	width	56
4	X0	40
5	X1	39
6	Y0	32
7	Y1	23
8	mes	20
9	true	17
10	false	15
11	fecha	12
12	dia	7
Total		550

### Sustitución de los datos:

$$N = 13 \log_2 (13) + 12 \log_2 (12)$$

$$N = 92$$

Volumen del programa

$$V = 92 \log_2 (13 + 12)$$

$$V = 430 \text{ bit}$$

Volumen mínimo

$$L = 2 / 13 * 12 / 550$$

$$L = 0,0032$$

- **Errores en la etapa de prueba**

**Fórmula:**

$$NP = 1 / [(n1 / 2) * (n2 / 2)]$$

$$e = V / NP$$

**Sustitución de los datos:**

$$NP = 1 / [(13 / 2) * (12 / 2)]$$

$$NP = 1/39$$

$$NP = 0,026$$

$$V = 430$$

Entonces:

$$e = 430 / 0,026$$

$$e = 16538$$

El esfuerzo de la ciencia del software es de 16538.

Validación de los indicadores de la etapa de entrega del producto.

- **Tasa de Estabilidad**

**Fórmula:**

$$TE = 1 - (Nc / CHU)$$

**Sustitución de los datos:**

$$TE = 1 - (0 / 10)$$

$$TE = 1$$

Como la tasa de estabilidad es 1 la aplicación es totalmente estable, pues no se realizaron cambios solicitados por el cliente en el primer trimestre, siendo así la cantidad de historias de usuarios con que cuenta la aplicación mayor que los cambios efectuados.

- **Tasa de defectos (TD)**

**Fórmula:**

$$TD = ND / CHU$$

**Sustitución de los datos:**

$$TD = 0 / 10$$

$$TD = 0$$

Debido a que TD es 0, se determinó que hay pocos defectos en el sistema, ya que el número de defectos es menor que el de las HU.

- **Cumplimiento de la Planificación de Entrega.**

**Fórmula:**

$$CPE = (TR * 100) / TE$$

**Sustitución de los datos:**

$$CPE = (6 * 100) / 6$$

$$CPE = 100 \%$$

## Capítulo 2: Validación

---

El cumplimiento de la planificación de la entrega es de un 100 %, por tanto se están cumpliendo con los resultados mejor que lo planificado y se cuenta con un software bastante estable y fiable.

Validación de los indicadores de la etapa de mantenimiento.

- **Estabilidad del producto**

**Fórmula:**

$$\text{IMS} = [\text{Mt} - (\text{Fa} + \text{Fc} + \text{Fd})] / \text{Mt}$$

**Sustitución de los datos:**

$$\text{IMS} = [7 - (0 + 2 + 0)] / 7$$

$$\text{IMS} = 5 / 7$$

$$\text{IMS} = 0.71$$

Con esta métrica se determina que como el Índice de Madurez es de 0.71, la estabilidad del producto es regular.

### 2.3.3. Validación en el Proyecto Yarey

Este proyecto se encuentra en la fase de planificación dentro de su desarrollo, por tanto serán validados los indicadores que cumplan las condiciones hasta el momento.

Validación de los indicadores en la etapa de Definición - Planificación.

- **Productividad**

**Fórmula:**

$$E = P * T$$

### **Datos:**

Preparación de los miembros del proyecto = 4 semanas

Iteración1 (HU\_4 implementadas) = 5 semanas

El total de semanas es de 9 semanas, o sea 2 meses hasta el momento.

### **Sustitución de los datos:**

$$E = 3 * 2$$

$$E = 6 \text{ persona / mes}$$

El esfuerzo real del proyecto es de 6 persona / mes.

- **Duración del proyecto**

### **Fórmula:**

$$EP = DR / DE$$

### **Datos:**

DR = 2 meses por ahora.

La Duración Estimada del proyecto es de:

Preparación de los miembros del proyecto = 4 semanas

Iteración1 (HU\_4 implementadas) = 5 semanas

$$DE = 2 \text{ meses}$$

$$DR = 2 \text{ meses}$$

### **Sustitución de los datos:**

$$EP = 2 / 2$$

$$EP = 1$$

La estimación de plazo de este proyecto es de 1 mes.

- **Desviación existente del tiempo real con respecto al planificado.**

**Fórmula:**

$$TT = TD + TP$$

$$DT = TT - TPI$$

**Sustitución de los datos:**

$$TT = 2 + 2$$

$$TT = 4$$

$$DT = 4 - 6$$

$$DT = -2$$

El proyecto tiene una desviación de tiempo menor que 0, entonces se puede afirmar que las actividades del mismo se están cumpliendo mejor que lo planificado hasta el momento.

- **Riesgo**

**Fórmula:**

$$R = I * P$$

**Tabla 2.20. Sustitución de los datos**

No	Riesgos	I	P	R
1	Ausencia de la persona encargada de enviar el trabajo a renderizar.	5	0,5	2,5
2	Salida de la universidad o del proyecto del encargado de enviar a realizar la renderización.	5	0,5	2,5

## Capítulo 2: Validación

3	Rotura de una máquina donde se encuentra información importante.	5	0,9	4,5
4	Computadoras que no son capaces de conectarse automáticamente para la realización del renderizado.	5	0,9	4,5
5	Realización del apagado de algunas PC por parte de algún personal.	5	0,9	4,5
6	El cliente abandona el interés por el producto.	9	0,9	8.1

Entonces el Orden de Mitigación de los riesgos del proyecto es:

Riesgo 6

Riesgo 3, Riesgo 4 y Riesgo 5

Riesgo 1 y Riesgo 2

- **Cumplimiento de Compromiso**

**Fórmula:**

$$CA_i = C_{an} - C_{ae}$$

**Sustitución de los datos:**

$$CA_i = 6 - 0$$

$$CA_i = 6$$

Como la cantidad de actividades incumplidas ( $CA_i$ ) es mayor que 0, se tiene un incumplimiento en las actividades y no se está cumpliendo con el compromiso.

Es un producto cubano que se realiza en los proyectos productivos de la UCI, por lo que no se le asignó ningún presupuesto, debido a esto no se evalúan los siguientes indicadores:

- **Desviación del Costo**

- Estimación Exacta del gasto.
- Calidad de Especificación

**Fórmula:**

$$N_r = N_f + N_{nf}$$

**Sustitución de los datos:**

$$N_r = 6 + 5$$

$$N_r = 11$$

Para determinar la ausencia de ambigüedad se calcula la Especificación de Requisitos (ER)

**Fórmula:**

$$ER = N_{vi} / N_r$$

**Sustitución de los datos:**

$$ER = 4 / 11$$

$$ER = 0.36$$

Debido a que la especificación de los requisitos es de 0.36, se concluye que tiene una evaluación de mal, por tanto existe ambigüedad en ellos.

Para determinar la compleción de los requisitos funcionales

**Fórmula:**

$$CRF = U_n / (N_i * N_s)$$

**Sustitución de los datos:**

$$CRF = 11 / (11 * 11)$$

CRF = 0,09

La compleción de los requisitos funcionales es de 0,09.

Para medir el porcentaje de funciones necesarias que se han especificado para un sistema se determina el Grado de Validación de los Requisitos (GVR).

**Fórmula:**

$$\text{GVR} = \text{Nc} / (\text{Nc} + \text{Nnu})$$

**Sustitución de los datos:**

$$\text{GVR} = 0 / (0 + 11)$$

$$\text{GVR} = 0$$

Se obtiene un grado de validación de los requisitos de un 0 % puesto que no se ha validado ninguna de las funcionalidades del sistema.

Por el estado actual del proyecto, no se puede determinar los indicadores enunciadas a continuación:

- **Basada en la función**
- **Henry y Kafura**
- **Diseño de la Interfaz**
- **Calidad del programa**
- **Errores en la etapa de prueba**
- **Estabilidad del producto**
- **Tasa de Estabilidad**
- **Tasa de defectos**
- **Cumplimiento de la Planificación de Entrega**

### 2.4. Resultados Obtenidos en la Validación

Luego de la realización de la validación de los indicadores propuestos en los proyectos se concluyó como **resultados positivos** los siguientes:

- La efectividad de la aplicación de cada uno de los indicadores.
- Medición de la calidad en cada etapa del ciclo de vida del proyecto.
- Mejora de las estadísticas del proceso de desarrollo de software.

Además se obtuvieron resultados un tanto **preocupantes** como:

- Presencia de ambigüedad en los requisitos del proyecto Yarey.
- Empleo de mayor tiempo para la obtención de la aplicación, ocasionado por tener que dedicar tiempo al arreglo de errores en fases anteriores, lo que trae consigo la entrega tardía del producto final.
- La estabilidad del producto NovaDesk no es la más óptima.

Estos son algunos de los factores que impiden que en los proyectos analizados no se produzca software con la mejor calidad posible.

La excelencia del software puede medirse después de elaborado el producto, pero esto puede resultar muy costoso si se detectan problemas, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software, y para lograr esto es necesario que se utilicen los indicadores durante todo el desarrollo del producto. En SXP esto es mucho más fácil de ver debido a que se desarrolla en las revisiones cada 15 días se van realizando entregas de pequeños funcionales con la calidad requerida donde se va comprobando la calidad y eficiencia de lo que se ha estado desarrollando.

Finalmente se concluye que los indicadores propuestos para SXP favorecerán la calidad de los productos, y brindarán una medida de cómo está progresando el desarrollo del software. Para lograr esto a continuación se define donde se obtiene mejor resultado al aplicar los indicadores con la metodología SXP:

En la etapa de Definición – Planificación se utiliza:

- Productividad
- Estimación de plazo
- Desviación de tiempo
- Riesgo
- Cumplimiento de compromisos
- Desviación de costo
- Porcentaje gastado del presupuesto general
- Calidad de la especificación.

En la etapa de Desarrollo:

- Basada en función
- Henry y Kafura para el diseño arquitectónico
- Conveniencia de representación del diseño de la interfaz
- Código fuente
- Prueba

En la etapa de entrega del producto:

- Tasa de estabilidad.
- Tasa de defectos.
- Cumplimiento de la planificación de entrega.

En la etapa de mantenimiento:

- Índice de Madurez del Software.

## Capítulo 3: Propuesta del Diseño del Sistema

---

### Capítulo 3: Propuesta del Diseño del Sistema

El uso de los sistemas para la gestión de indicadores en la producción de software, permite el trabajo del equipo de desarrollo con mayor organización y en menor tiempo. Además que puedan consultar, actualizar y obtener información detallada para su mejor aplicación; de manera tal, que la evaluación de la calidad del producto alcance niveles elevados de efectividad.

#### 3.1. Sistemas Existentes para la Gestión de Indicadores

Existen varios sistemas para la gestión de indicadores, pero ninguno se adapta a las características necesarias para la metodología SXP, por esta razón se propone una herramienta web que permita realizar la medición para los proyectos que utilizan esta metodología. A continuación se describen algunos de los sistemas existentes:

##### SourceMonitor

SourceMonitor es un programa gratuito que le permite ver el interior del código fuente del software para determinar la cantidad de código que tiene e identificar la complejidad relativa de los módulos. Por ejemplo permite identificar el código que es más probable que contenga defectos y, por tanto merece una revisión formal.

SourceMonitor establece lo siguiente:

- Recoge los indicadores de una manera rápida.
- Contiene medidas para el código fuente escrito en C ++, C, C #, Java, Delphi, Visual Basic 6 o HTML.
- Guarda indicadores en los puestos de control para la comparación en los proyectos de desarrollo de software.
- Muestra e imprime indicadores en tablas y gráficos.

## Capítulo 3: Propuesta del Diseño del Sistema

---

- Funciona dentro de un estándar de interfaz gráfica de usuario de Windows o en el interior sus scripts usando el comando de archivos XML<sup>14</sup>.

Las exportaciones de indicadores para XML, los archiva para su posterior tratamiento con otras herramientas.

La descarga es un instalador totalmente funcional contenido en un archivo único. [21]

Es un programa que corre sobre Windows y se centra en indicadores que evalúan el código fuente, por lo tanto no es válido para los indicadores de SXP.

### PSMInsight

Programa que trabaja con modelos de procesos tipo CMMI o 1550415, implementado en SWL. Está pensado para proyectos grandes, con ciclos de desarrollo predictivos que desarrollan sistemas con niveles de criticidad elevados. Es muy configurable y quitando muchos de los indicadores, categorías y plantillas que incorpora por defecto, puede resultar útil para proyectos pequeños con modelos de gestión ágiles.

Su instalación es molesta y si no se encuentra el motor de Borland Database Engine<sup>16</sup> instalado en la máquina lo añada, y toma prestado el directorio de Windows para dejar todo lo que le necesite para su funcionamiento. Lo ha desarrollado la fundación americana para la medición en el desarrollo de software.

[22]

Este sistema no es factible para los indicadores de SXP, debido a que es una aplicación de escritorio, que solo puede ser ejecutada por una persona a la vez. Aunque gestiona y organiza el proceso de medición de CMMI, no brinda un espacio de uso colectivo que ayude a mantener a todos los equipos de desarrollo bien informados en cuanto al uso de los indicadores.

### Sprintometer

---

<sup>14</sup> Extensible Markup Language

<sup>15</sup> SPICE. Estándar ISO/IEC 15504.

<sup>16</sup> Es un motor de conexión a bases de datos que permite manejar bases de datos de escritorio.

## Capítulo 3: Propuesta del Diseño del Sistema

---

Sprintómetro es un programa gratuito, contenido en un único fichero ejecutable, que no necesita instalación, para gestión, medición y seguimiento de programas con modelos ágiles tipo XP o Scrum. Registra las historias de usuario o funcionalidades, las tareas, estimaciones y su asignación a las personas del equipo. Se puede llevar el seguimiento diario, exportar los datos a excel, exportar/importar datos del proyecto al formato XML con una interfaz moderna y otras funcionalidades. Los datos de cada proyecto se pueden guardar en un fichero local, o en un servidor de base de datos.

Fue creado en un principio para que lo usaran los programadores de proyectos ágiles, pero ahora es un producto para cualquier usuario. Desarrollado por un equipo de programadores rusos, para emplearlo en sus proyectos.

El programa planifica y realiza sprints (iteraciones) con composición de equipo variable. El sprint sirve para ofrecer una versión parcial del producto.

Esta herramienta, independientemente de ser software gratis, solo se encuentra hasta el momento, disponible para su ejecución sobre sistemas operativos propietarios. [23]

Este sistema fue propuesto en la investigación realizada por los Ingenieros Ailema Sisinta y Adolis Rodríguez, es un sistema para la gestión de proyecto de manera eficiente y no para la gestión de indicadores.

### Sistema de Gestión de Métricas de la Dirección de Calidad de Software

A nivel nacional se cuenta con un sistema para la gestión de métricas realizado en la UCI, en el trabajo de Diploma de los ingenieros Dunieski Sarmiento Méndez. y Romel Rodríguez Torres en el año 2009. Esta herramienta se encarga de insertar, modificar y eliminar métricas, ya sean derivadas o bases. Garantiza la gestión de los tipos de proyectos a los cuales van a estar dirigidas las métricas, la clasificación de las mismas y las metas a las cuales van a tributar una vez que sean insertadas en el sistema de medición, además de gestionar todo el contenido que será visto por el usuario en la página principal de la aplicación informativa. Esta aplicación admite la gestión de los usuarios que sean necesarios para administrarla,

## Capítulo 3: Propuesta del Diseño del Sistema

---

permite mostrar las informaciones de la portada, consultar las métricas existentes en la base de datos a la comunidad, así como filtrarlas, de acuerdo a su tipo, clasificación y las metas que satisfacen. [24]

El sistema es independiente de plataforma, podrá ser usado bajo los sistemas operativos Windows y Linux; gestiona cualquier tipo de métricas, pero, no se encuentra implantado aún, razón por la cual no se puede utilizar para SXP.

### 3.2. Situación Actual

Específicamente para la gestión, planificación y seguimiento del grupo UNICORNIOS, se utiliza el Dotproject y el Gforge, dichas herramientas siguen una línea común más bien encaminada al tiempo de desarrollo planificado de los proyectos. En este grupo, no se han aplicado indicadores, ni conocen o dominan los beneficios de esta terminología, lo cual ha hecho que los productos que se han desarrollado no sean entregados con la calidad suficiente y requerida.

Actualmente en UNICORNIOS, no se aplica un proceso de medición automático que arroje resultados y permita analizarlos, y perfeccionar los procesos en cada software que realizan. Por tal razón se les dificulta hacerle llegar a cada equipo de desarrollo las últimas definiciones respecto a las mediciones, así como las particularidades que puede tener el proceso de medición.

### 3.3. Propuesta del Sistema

Para dar solución a la situación planteada, se propone elaborar una herramienta Web llamada SG-Indicators, a la cual tendrán acceso para consultar la información disponible, los miembros de los proyectos que utilizan SXP. El sistema tiene como objetivo primordial el cálculo de los indicadores, les permitirá a los usuarios la administración de los proyectos, o sea, adicionar datos que brindan información básica de estos, modificarlos, conocer el listado de los mismos y eliminarlos en caso que se desee. Los resultados podrán ser exportados a formato hoja de cálculo o pdf, y se mostrarán en un gráfico para que

## Capítulo 3: Propuesta del Diseño del Sistema

---

se conozca el nivel de calidad que va alcanzando cada proyecto. Además contará con un administrador que se encargará de darles los permisos necesarios a los usuarios registrados según su rol.

Para el desarrollo de la herramienta se propone como lenguaje PHP, que es gratuito y multiplataforma y le permite al cliente interactuar con la aplicación de forma rápida, segura y eficiente. Equipado con un conjunto de funciones de seguridad, que previenen la inserción de órdenes dentro de una solicitud de datos.

Como framework a utilizar se usará Symfony, que proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja; además, de automatizar las tareas más comunes. Es un framework desarrollado íntegramente en PHP 5, diseñado para optimizar el desarrollo de las aplicaciones web. Cuenta con una amplia documentación desde sus inicios en el año 2005, razón que lo ha hecho sobresalir por encima de muchos frameworks usados en la actualidad, y utiliza Modelo Vista Controlador (MVC) que separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Por tanto para la arquitectura del sistema se propone el MVC, el cual estructura el sistema de una forma lógica para que responda a los requisitos funcionales para los cuales fue creado en tres capas que se comunican como se muestra en la figura 1, dependiendo de su naturaleza.

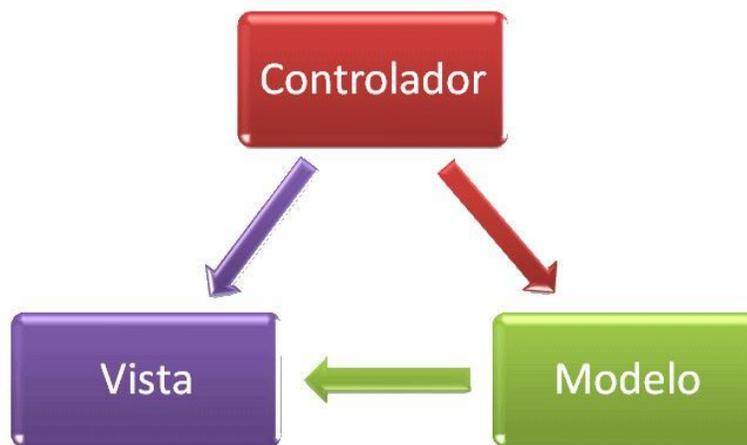


Figura 1. Arquitectura Modelo Vista Controlador

## Capítulo 3: Propuesta del Diseño del Sistema

---

Como gestor de base de datos se usará PostgreSQL 8.0 o superior, dado que ofrece muchas ventajas respecto a otros sistemas de bases de datos, el mismo es multiplataforma, extensible y está diseñado para ambientes de alto volumen. Además de su estabilidad y confiabilidad, cuenta con herramientas gráficas de diseño y administración de bases de datos.

El servidor web escogido es Apache 2.0 o superior, pues, ofrece una perfecta combinación entre estabilidad y sencillez, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma y se distribuye en casi todas las implementaciones de Linux.

### 3.4. Análisis del Negocio

Durante la etapa de análisis de negocio del ciclo de vida del sistema, el cliente caracteriza la estructura y funcionamiento que desea obtener en el producto. Estas tareas quedan plasmadas en el modelo de proceso del negocio, el cual define un conjunto de procesos relacionados lógicamente llevados a cabo para lograr un resultado de negocio definido. Este modelo es una forma de abordar la comunicación entre los clientes y los técnicos, representando el flujo de actividades que se realizan para lograr la meta establecida y puede ser realizado mediante la técnica BPMN<sup>17</sup>; paradigma para la administración de procesos de negocio, que persigue el modelado de las actividades de esta etapa para lograr una mejor administración, automatización y optimización.

El proceso de gestión de indicadores es inicializado cuando el equipo de trabajo extrae los datos de cada proyecto del expediente. Después fija la fase en que se encuentra el mismo, y en dependencia de esto determina los indicadores de SXP a aplicar. Si se encuentra en la fase de Definición - Planificación establece los indicadores a calcular en esta (Productividad, Duración del proyecto, Desviación del tiempo, Riesgo, Cumplimiento de compromiso, Desviación del costo, Desviación exacta del gasto y Especificación

---

<sup>17</sup> Business Process Management Notation

## Capítulo 3: Propuesta del Diseño del Sistema

---

de requisitos), de encontrarse en la fase de Desarrollo le agrega los indicadores de la misma (Puntos de función, Henry y Kafura, Diseño de la interfaz, Código fuente y Prueba) a los anteriores, si la fase en que está es Entrega adiciona los correspondientes a esta (Tasa de estabilidad, Tasa de defectos y Cumplimiento de entrega) a los de las fases anteriores, y si está en la fase de Mantenimiento agrega el indicador de la misma (Índice de madurez del software). Luego calcula los indicadores determinados y el resultado es registrado, para finalmente medir la calidad del proyecto en cada fase.

Glosario de las actividades del Modelo de Procesos del Negocio:

**Recopilar datos de los proyectos de SWL:** en esta actividad se extraen los datos del expediente del proyecto, que se encuentra en el gestor de los mismos.

**Buscar la fase en que se encuentra cada proyecto:** determinar por el avance del expediente la fase del proyecto.

**Indicadores para planificación:** recoge los indicadores a aplicar en esta fase.

**Adicionar indicadores para desarrollo:** Agrega los indicadores para la fase desarrollo del proyecto a los anteriores.

**Agregar los indicadores para entrega:** Adiciona los indicadores de la presente fase a los de las anteriores.

**Adicionar indicador para mantenimiento:** Se agrega el indicador de esta fase a los anteriores.

**Calcular los indicadores:** según la etapa del proyecto se calculan los indicadores apropiados aplicar.

**Registrar resultados obtenidos:** se guardan los resultados.

**Medir la calidad:** según los resultados obtenidos se determina la calidad del proyecto.

Para mayor comprensión a continuación se presenta un diagrama de proceso que explica el funcionamiento.

## Capítulo 3: Propuesta del Diseño del Sistema

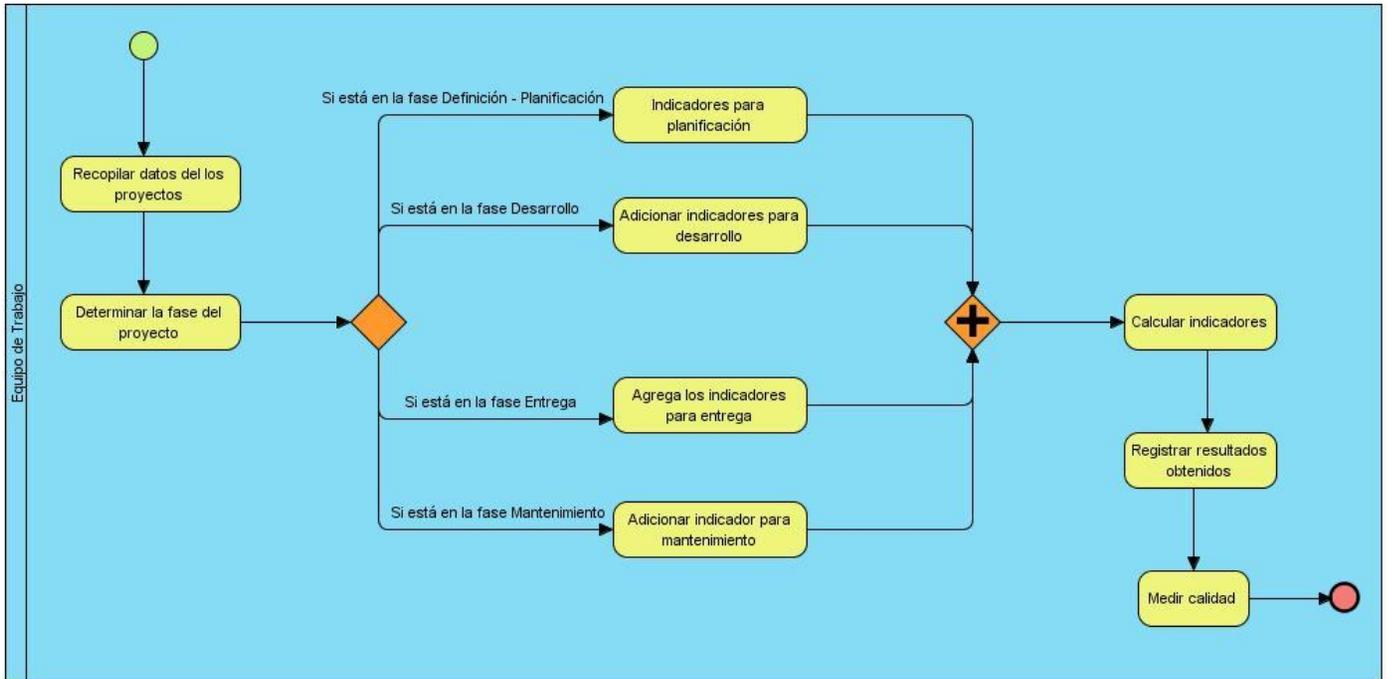


Figura 2. Modelo de Procesos del Negocio

### 3.5. Lista de Reserva del Producto

La lista de reserva del producto (LRP) es una lista priorizada donde se especifican los requisitos funcionales y no funcionales para tener en cuenta en el futuro proyecto a desarrollar, o sea cuenta con los requisitos que debe llevar la aplicación establecidos por el cliente, y se define en la etapa de planificación del proyecto.

Tabla 3.1. LRP

Prioridad	Ítem *	Descripción
RF		

## Capítulo 3: Propuesta del Diseño del Sistema

<i>Muy Alta</i>		
	1	Calcular indicador productividad
	2	Calcular indicador duración del proyecto
	3	Calcular indicador desviación del tiempo
	4	Calcular indicador riesgos
	5	Calcular indicador cumplimiento de compromisos
	6	Calcular indicador desviación del costo
	7	Calcular indicador estimación exacta del gasto
	8	Calcular indicador especificación de los requisitos
	9	Calcular indicador puntos de función
	10	Calcular indicador Henry y Kafura
	11	Calcular indicador diseño de interfaz
	12	Calcular indicador código fuente
	13	Calcular indicador prueba
	14	Calcular indicador tasa de estabilidad
	15	Calcular indicador tasa de defectos

## Capítulo 3: Propuesta del Diseño del Sistema

	16	Calcular indicador cumplimiento de entrega
	17	Calcular indicador índice de madurez del software
<b>Alta</b>		
	18	Medir la calidad de los proyectos en cada fase
	19	Introducir datos del proyecto
	20	Modificar datos del proyecto
	21	Listar proyectos
	22	Eliminar proyectos
	23	Exportar Archivo
	24	Acceder al sistema
<b>Media</b>		
	25	Crear una cuenta de usuario
<b>Baja</b>		
<b>RNF</b>		
	1	Lograr que el sistema sea extensible, permitiendo agregar nuevas funcionalidades en un futuro.
	2	Documentar el sistema de forma tal que en caso de mantenimiento el tiempo

## Capítulo 3: Propuesta del Diseño del Sistema

---

		que se requiera sea el mínimo.
	3	Diseñar interfaz amigable para el usuario, navegable y fácil de usar.
	4	Lograr que el sistema sea rápido y navegable.
	5	Necesita requerimientos mínimos de hardware (512 MB de memoria RAM y 1.36 GHz de velocidad de procesamiento).
	6	Se utilizará un servidor con sistema operativo Linux.
	7	En las computadoras de los clientes se requiere del navegador Mozilla Firefox.
	8	La información que brinda el sistema puede ser consultada por cualquier persona, que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.
	9	El sistema solo puede ser administrado por la persona que cuenta con los permisos necesarios para hacerlo.

### 3.6. Historias de Usuarios

HU 1: Calcular Indicadores

HU 2: Administrar proyectos

HU 3: Acceder al sistema

HU 4: Mostrar datos de la calidad de los proyectos

HU 5: Crear cuenta de usuario para acceder al sistema

## Capítulo 3: Propuesta del Diseño del Sistema

HU 6: Exportar Archivos

Tabla 3.2. HU1\_Calcular Indicadores

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre Historia de Usuario:</b> Calcular Indicadores
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b>	<b>Iteración Asignada:</b>
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b>
<p><b>Descripción:</b> Realiza los cálculos de los indicadores, según la fase en que se encuentra el proyecto el usuario selecciona el indicador a calcular habilitado y se le muestra un formulario con los datos a llenar de acuerdo al mismo.</p>	
<p><b>Observaciones:</b> 1. Si el proyecto está en la fase de Definición – Planificación se seleccionan los siguientes indicadores a calcular y se llena el formulario correspondiente: (Ver Interface 1)</p> <p><i>Productividad:</i> se introduce en el formulario el número de personas que integran el proyecto y el tiempo del mismo o duración real. El resultado se expresa en persona/mes y se guarda en la base de datos. (Ver Interface 1.1)</p> <p><i>Duración del proyecto:</i> se llena el formulario con la duración real del proyecto y la duración. El resultado se expresa en semanas y se guarda en la base de datos. (Ver Interface 1.2)</p>	

## Capítulo 3: Propuesta del Diseño del Sistema

---

Desviación del tiempo: primeramente se calcula el tiempo total de desarrollo sumando el tiempo usado de desarrollo del proyecto que es la duración real y el tiempo planificado para las actividades por realizar en el proyecto, con el resultado obtenido más el tiempo total planificado inicialmente que se introduce en el formulario se obtiene la desviación del tiempo, resultado final del indicador. Si la desviación del tiempo es igual que 0, el proyecto cumplió su plazo de tiempo planificado y se guarda en la base de datos; si es mayor que 0 entonces es necesario replanificar las actividades porque hay un atraso en el tiempo, se le informa al equipo de trabajo del atraso existente para tomar las medidas pertinentes y se guarda en la base de datos. (Ver Interface 1.3)

Riesgos: Se llena en el formulario los campos de la probabilidad de que ocurra el riesgo y el impacto. (Ver Interface 1.4)

Cumplimiento de compromisos: para determinar la posibilidad de que un proyecto cumpla con los compromisos requeridos, se selecciona una de las 2 opciones (Ver Interface 1.5): a) si el compromiso depende de las actividades a realizar: se muestra un formulario para poner la cantidad de HU que se necesitan para realizar el producto y la cantidad de HU realizadas hasta el momento para calcular la cantidad de HU necesarias incumplidas hasta el momento de la revisión (Ver Interface 1.5a), b) Si el compromiso depende de la asignación de recursos se introduce los siguientes datos: recursos necesarios para cumplir el compromiso y los recursos disponibles de los necesarios para cumplir el compromiso, con esto se calcula los recursos no disponibles de los necesarios para cumplir el compromiso (Ver Interface 1.5b). Si los resultados obtenidos en a) o b) es igual que 0 hay una gran probabilidad de que se cumpla con el compromiso y se guarda en la base de datos; si es mayor que 0 puede ser que no se cumpla con el compromiso, porque hubo incumplimiento en la realización de las HU o no se contaba con los recursos necesarios, por lo que se alerta al equipo de trabajo sobre el problema a resolver y se guarda en la base de datos.

Desviación de costo: se llena en el formulario los campos de el presupuesto planificado y el costo real para calcular el índice de actuación de costo, con el valor obtenido y el costo total planificado introducido en el formulario se calcula la desviación de costo; si el resultado final es igual a 0 significa que se está cumpliendo con el presupuesto planificado, se informa y se guarda en la base de datos; si es mayor que 0 se necesita

## Capítulo 3: Propuesta del Diseño del Sistema

---

replanificar porque hubo una desviación de costo respecto al planificado, se informa el problema y se guarda en la base de datos; y si es menor que 0, se cuenta con una reserva monetaria que puede ser usada en otro momento, se informa la cantidad y se guarda en la base de datos. (Ver Interface 1.6)

Estimación exacta del gasto: se introduce el costo hasta la fecha y el presupuesto total. (Ver Interface 1.7)

Especificación de requisitos: se introduce el número de requisitos funcionales y no funcionales, para calcular los requisitos en una especificación, con el resultado y el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas introducidos en el sistema se determina la calidad de especificación de los requisitos mediante la escala de evaluación de los resultados, se informa el resultado y se guarda en la base de datos. Luego se introduce el número de requisitos únicos de función, el número de entradas definidas o implicadas por la especificación y el número de estados planificados, para medir la compleción de los requisitos funcionales. Para obtener el porcentaje de funciones necesarias que se han especificado para un sistema se introduce el número de requisitos validados como correctos y el número de requisitos que no se han validado. (Ver Interface 1.8)

2. Para la etapa de Desarrollo del proyecto se habilitan para calcular los siguientes indicadores: (Ver Interface 1)

Puntos de función: el usuario selecciona según su criterio el factor de ponderación (simple, media o compleja) (Ver Interface 2.1), se direcciona a otra interfaz que muestra la tabla con la columna del factor de ponderación seleccionado, la columna de parámetros de medición (número de entradas, de salidas y de consultas del usuario, número de archivos y de interfaces externas), la columna cuenta para introducir la cantidad de cada uno de los parámetros, la multiplicación de la columna cuenta y factor de ponderación se muestra en la columna total, y la sumatorias de esta columna calcula la cuenta total, que en unión de las respuestas a las 14 preguntas que se muestran también, según los valores de ajustes determinan el tamaño global de la implementación expresados en puntos de función, luego se vuelve a la página anterior para mostrar el resultado que se guarda en la base de datos. (Ver Interface 2.1a)

Henry y Kafura: se introduce en el formulario el número de funcionalidades críticas por módulo y el 50 % de

## Capítulo 3: Propuesta del Diseño del Sistema

---

estas, luego en el botón examinar se sube el código del módulo para contar el número de sentencias, y se procede al cálculo del indicador. (Ver Interface 2.2)

Diseño de interfaz: se muestra un formulario con datos del indicador, se acepta (Ver Interface 2.3) y se muestra otra interfaz con la tabla de parámetros de la interfaz (Ver Interface 2.3a), a los cuales se les da valor 0 o 1, en una escala de 1 a 5, se suman en forma de matriz, el resultado se evalúa en la escala  $60 \leq DI \leq 100$ , se vuelve a la página anterior, se muestra el resultado; si no está en la escala se le informa al usuario la baja calidad de la interfaz y se guarda en la base de datos.

Código fuente: para calcular este indicador el usuario debe subir el código, para contar el número de operadores y operandos diferentes que aparecen respectivamente, así como determinar la cantidad de veces que aparece cada uno, el resultado muestra la longitud global del programa, también se determina el volumen del programa y el volumen mínimo para un algoritmo, lo obtenido se guarda en la base de datos y se le muestra al usuario. (Ver Interface 2.4)

Prueba: para el cálculo de este indicador el usuario sube el código del módulo para contar la cantidad de operandos y operadores diferentes que aparecen en el programa, así como el número total de veces que aparece el operando, el volumen es el resultado de una de las ecuaciones del indicador anterior guardado en la base de datos. El resultado final se muestra y se guarda en la base de datos. (Ver Interface 2.5)

3. En la etapa de Entrega del producto se calculan los indicadores para la satisfacción del cliente habilitados: (Ver Interface 1)

Tasa de estabilidad: se introduce el número de cambios solicitados durante el primer trimestre por el cliente, o sea los cambios efectuados en el software por parte de los desarrolladores debido a que no pudieron acoplarse en un 100 % a los requisitos que exige el cliente y la cantidad de HU. El resultado se mide en una escala de 0 a 1 tomando la evaluación de Bien, Regular o Mal para informarlo al usuario, y se guarda en la base de datos. (Ver Interface 3.1)

## Capítulo 3: Propuesta del Diseño del Sistema

---

Tasa de defectos: se llena en el formulario los campos de el total de incidencias en las que la aplicación no cumplió las especificaciones o expectativas del cliente y la cantidad de HU. El resultado se guarda en la base de datos y se le muestra al usuario. (Ver Interface 3.2)

Cumplimiento de entrega: se llenan los campos con el tiempo real dedicado a la entrega del producto expresado en minutos y el tiempo estimado o planificado que se pactó con el cliente desde su inicio expresado en minutos. Si el resultado es mayor que el 100 % hay retraso en el plazo de entrega y si es menor o igual que 100 entonces se están consiguiendo mejor los resultados de lo planificado, se le muestra al usuario y se guarda en la base de datos. (Ver Interface 3.3)

4. En la etapa de Mantenimiento del producto se calcula: (Ver Interface 1)

Índice de madurez del software: se introduce en el formulario el número de módulos en la versión actual, el número de módulos que se han cambiado y los que se han añadido, además del Número de módulos en la versión anterior que se han borrado en la versión actual. El resultado se mide en una escala de 0 a 1 evaluando la estabilidad del proyecto en Bien, Regular o Mal, el mismo se guarda en la base de datos y se muestra al usuario. (Ver Interface 2.6)

## Capítulo 3: Propuesta del Diseño del Sistema

### Prototipo de interface:

#### Interface 1



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores

Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

Puntos de Función

Tasa de Estabilidad

Índice de Madurez del Software

Duración del Proyecto

Henry y Kafura

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

Código Fuente

Cumplimiento de Compromiso

Prueba

Desviación del Costo

Estimación Exacta del Gasto

Especificación de Requisitos

## Capítulo 3: Propuesta del Diseño del Sistema

### Interface 1.1



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

#### Productividad

Puntos de Función

Tasa de Estabilidad

Índice de Madurez del Software

Duración del Proyecto

Henry y Kafura

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

Código Fuente

Cumplimiento de Compromiso

Prueba

Desviación del Costo

Estimación Exacta del Gasto

Especificación de Requisitos

[Exportar a pdf](#)

[Exportar a hoja de cálculo](#)

Nombre: **Productividad**

Propósito: Medir el esfuerzo real del proyecto.

Fórmula:  **$E = P * T$**

**Datos** Número de personas (P):

Cantidad de tiempo (T):

Medida: Persona / mes

Interpretación: El esfuerzo real del proyecto es de ...

Aceptar

Cancelar

### Interface 1.2

## Capítulo 3: Propuesta del Diseño del Sistema

**SXP** Metodología Ágil de Desarrollo

**SG-INDICATORS**  
Sistema para la Gestión de Indicadores

Administrar Datos del Proyecto    Calcular Indicadores    Medir Calidad    Salir

Productividad	Puntos de Función	Tasa de Estabilidad	Índice de Madurez del Software
<b>Duración del Proyecto</b>	Henry y Kafura	Tasa de Defectos	
Desviación Existente del Tiempo Real	Diseño de Interfaz	Cumplimiento de Entrega	
Riesgo	Código Fuente		
Cumplimiento de Compromiso	Prueba		
Desviación del Costo			
Estimación Exacta del Gasto			
Especificación de Requisitos			

[Exportar a pdf](#)    [Exportar a hoja de cálculo](#)

Nombre: **Duración del Proyecto**

Propósito: Definir con precisión la duración del proyecto

Fórmula: **EP = DR / DE**

**Datos**    Duración Real (DR):

                 Duración Estimada (DE):

Medida:    Semanas

Interpretación: La estimación de plazo es de ...

**Interface 1.3**

## Capítulo 3: Propuesta del Diseño del Sistema



<a href="#">Administrar Datos del Proyecto</a>	<a href="#">Calcular Indicadores</a>	<a href="#">Medir Calidad</a>	<a href="#">Salir</a>
--	--------------------------------------	-------------------------------	-----------------------

Productividad	Puntos de Función	Tasa de Estabilidad	Índice de Madurez del Software
Duración del Proyecto	Henry y Kafura	Tasa de Defectos	
<b>Desviación Existente del Tiempo Real</b>	Diseño de Interfaz	Cumplimiento de Entrega	
Riesgo	Código Fuente		
Cumplimiento de Compromiso	Prueba		
Desviación del Costo			
Estimación Exacta del Gasto			
Especificación de Requisitos			

<a href="#">Exportar a pdf</a>	<a href="#">Exportar a hoja de cálculo</a>
--------------------------------	--

Nombre: **Desviación del Tiempo**

Propósito: Decidir si se necesita replanificar.

Fórmulas: **TT = TD + TP**    **DT = TT - TPI**

**Datos**

Tiempo usado de desarrollo (TD):

Tiempo planificado para las actividades (TP):

Tiempo total planificado inicial (TPI):

Interpretación: Depende del resultado...

**Interface 1.4**

## Capítulo 3: Propuesta del Diseño del Sistema



Administrar Datos del Proyecto	Calcular Indicadores	Medir Calidad	Salir
Productividad	Puntos de Función	Tasa de Estabilidad	Índice de Madurez del Software
Duración del Proyecto	Henry y Kafura	Tasa de Defectos	
Desviación Existente del Tiempo Real	Diseño de Interfaz	Cumplimiento de Entrega	
<b>Riesgo</b>	Código Fuente		
Cumplimiento de Compromiso	Prueba		
Desviación del Costo			
Estimación Exacta del Gasto			
Especificación de Requisitos			
<a href="#">Exportar a pdf</a>	<a href="#">Exportar a hoja de cálculo</a>		

Nombre: **Riesgo**

Propósito: Darle un nivel de prioridad a los riesgos.

Fórmula:  **$R = I * P$**

**Datos**

Impacto sobre el proyecto(I):

Probabilidad de que ocurra (P):

Interpretación: Depende del resultado...

**Interface 1.5**

## Capítulo 3: Propuesta del Diseño del Sistema

The screenshot displays the SG-INDICATORS software interface. At the top left is the SXP logo with the text 'Metodología Ágil de Desarrollo'. To its right is the title 'SG-INDICATORS Sistema para la Gestión de Indicadores'. Below this is a navigation bar with four tabs: 'Administrar Datos del Proyecto', 'Calcular Indicadores', 'Medir Calidad', and 'Salir'. The main area contains a grid of indicators:

Productividad	Puntos de Función	Tasa de Estabilidad	Índice de Madurez del Software
Duración del Proyecto	Henry y Kafura	Tasa de Defectos	
Desviación Existente del Tiempo Real	Diseño de Interfaz	Cumplimiento de Entrega	
Riesgo	Código Fuente		
<b>Cumplimiento de Compromiso</b>	Prueba		
Desviación del Costo			
Estimación Exacta del Gasto			
Especificación de Requisitos			

A modal dialog box is open over the 'Cumplimiento de Compromiso' indicator, containing the following information:

Nombre: **Cumplimiento de Compromiso**

Propósito: Determinar que el proyecto cumpla con los compromisos requeridos.

Opciones:  Depende de las actividades  
 Depende de la asignación de recursos

Buttons: Aceptar, Cancelar

Interface 1.5a)

## Capítulo 3: Propuesta del Diseño del Sistema



Administrar Datos del Proyecto	Calcular Indicadores	Medir Calidad	Salir
Productividad	Puntos de Función	Tasa de Estabilidad	Índice de Madurez del Software
Duración del Proyecto	Henry y Kafura	Tasa de Defectos	
Desviación Existente del Tiempo Real	Diseño de Interfaz	Cumplimiento de Entrega	
Riesgo	Código Fuente		
<b>Cumplimiento de Compromiso</b>	Prueba		
Desviación del Costo			
Estimación Exacta del Gasto			
Especificación de Requisitos			
<a href="#">Exportar a pdf</a>	<a href="#">Exportar a hoja de cálculo</a>		

Nombre: **Cumplimiento de Compromiso**

Propósito: Determinar que el proyecto cumpla con los compromisos requeridos.

Fórmula:  **$CA_i = CA_n - CA_r$**

**Datos**

Número de actividades necesarias para cumplir hasta la revisión (CA<sub>n</sub>):

Número de actividades necesarias realizadas hasta la revisión (CA<sub>r</sub>):

Interpretación: Depende del resultado...

**Interface 1.5b)**

## Capítulo 3: Propuesta del Diseño del Sistema



Administrar Datos del Proyecto	Calcular Indicadores	Medir Calidad	Salir
Productividad	Puntos de Función	Tasa de Estabilidad	Índice de Madurez del Software
Duración del Proyecto	Henry y Kafura	Tasa de Defectos	
Desviación Existente del Tiempo Real	Diseño de Interfaz	Cumplimiento de Entrega	
Riesgo	Código Fuente		
<b>Cumplimiento de Compromiso</b>	Prueba		
Desviación del Costo			
Estimación Exacta del Gasto			
Especificación de Requisitos			
<u>Exportar a pdf</u>	<u>Exportar a hoja de cálculo</u>		

Nombre: **Cumplimiento de Compromiso**

Propósito: Determinar que el proyecto cumpla con los compromisos requeridos.

Fórmula:  $X = XN - XD$

**Datos**

Recursos necesarios para cumplir el compromiso (XN):

Recursos disponibles de los necesarios para cumplir el compromiso (XD):

Interpretación: Depende del resultado...

Aceptar Cancelar

## Capítulo 3: Propuesta del Diseño del Sistema

### Interface 1.6



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

Puntos de Función

Tasa de Estabilidad

Índice de Madurez del Software

Duración del Proyecto

Henry y Kafura

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

Código Fuente

Cumplimiento de Compromiso

Prueba

#### Desviación del Costo

Estimación Exacta del Gasto

Especificación de Requisitos

[Exportar a pdf](#)

[Exportar a hoja de cálculo](#)

Nombre: **Desviación de Costo**

Propósito: Determinar la desviación existente del costo con lo que se ha planificado realmente.

Fórmulas: **IC = PP + CR**    **CD = IC - CTP**

#### Datos

Presupuesto planificado (PP):

Costo real (CR):

Costo total planificado (CTP):

Medida: Pesos

Interpretación: Depende del resultado...

Aceptar

Cancelar

## Capítulo 3: Propuesta del Diseño del Sistema

### Interface 1.7



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

Puntos de Función

Tasa de Estabilidad

Índice de Madurez del Software

Duración del Proyecto

Henry y Kafura

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

Código Fuente

Cumplimiento de Compromiso

Prueba

Desviación del Costo

**Estimación Exacta del Gasto**

Especificación de Requisitos

Exportar a pdf

Exportar a hoja de cálculo

Nombre: **Estimación Exacta del Gasto**

Propósito: Comparar los resultados de cada fase para detectar desviaciones significativas.

Fórmula:  **$PG = (HF * 100) / PPT$**

**Datos**  
Costo hasta la fecha (HF):   
Presupuesto total (PPT):

Medida: Pesos

Interpretación: Depende del resultado...

Aceptar

Cancelar

# Capítulo 3: Propuesta del Diseño del Sistema

## Interface 1.8



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

Puntos de Función

Tasa de Estabilidad

Índice de Madurez del Software

Duración del Proyecto

Henry y Kafura

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

Código Fuente

Cumplimiento de Compromiso

Prueba

Desviación del Costo

Estimación Exacta del Gasto

**Especificación de Requisitos**

Exportar a pdf

Exportar a hoja de cálculo

Nombre: **Especificación de Requisitos**

Propósito: Valorar la calidad de la especificación de los requisitos.

Fórmulas: **Nr = Nf + Nnf**                      **ER = Nui / Nr**

**CRF = Un / (Ni \* Ns)**                      **GYR = Nc / (Nc + Nnv)**

**Datos**

Requisitos funcionales (Nf):

Requisitos no funcionales (Nnf):

Número de requisitos con interpretaciones idénticas (Nui):

Número de requisitos únicos de función (Un):

Número de entradas definidas (Nd):

Número de estados especificados (Ns):

Número de requisitos validados como correctos (Nc):

Número de requisitos que no se han validado (Nnv):

Interpretación: Depende del resultado...

Aceptar

Cancelar

## Capítulo 3: Propuesta del Diseño del Sistema

### Interface 2.1



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores

Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

**Puntos de Función**

Tasa de Estabilidad

Índice de Madurez del Software

Duración del Proyecto

Henry y Kafura

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

Código Fuente

Cumplimiento de Compromiso

Prueba

Desviación del Costo

Estimación Exacta del Gasto

Especificación de Requisitos

Exportar a pdf

Exportar a hoja de cálculo

Nombre: **Puntos de Función**

Propósito: Predecir el tamaño de un sistema que se va a obtener del modelo de análisis.

Fórmulas:  **$PF = Cuenta\_Total * (0.65 + 0.1 * Sum [Fi])$**

Factor de ponderación:

Medidas: Puntos de Función

Interpretación: El tamaño global de la implementación se supone que sea de ... aproximadamente.

Aceptar

Cancelar

### Interface 2.1a)

# Capítulo 3: Propuesta del Diseño del Sistema



**SXP**  
Metodología Ágil de Desarrollo

**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto
Calcular Indicadores
Medir Calidad
Salir

Parámetros de Medición	Cuenta	Simple	Media	Compleja	Total
Número de entradas del usuario	<input type="checkbox"/>	3	4	6	
Número de salidas del usuario	<input type="checkbox"/>	4	5	7	
Número de consultas del usuario	<input type="checkbox"/>	3	4	6	
Número de archivos	<input type="checkbox"/>	7	10	15	
Número de interfaces externas	<input type="checkbox"/>	5	7	10	
<b>Cuenta_Total</b>					

Darle valor entre 0 y 5 a cada pregunta

1. ¿Requiere el sistema copias de seguridad y recuperación fiables?
2. ¿Se requiere comunicación de datos?
3. ¿Existen funciones de procesamiento distribuido?
4. ¿Es crítico el rendimiento del sistema?
5. ¿Será ejecutado el sistema en un entorno operativo y frecuentemente utilizado?
6. ¿Requiere el sistema entrada de datos interactivo?
7. ¿Requiere la entrada de datos interactivo que las transiciones de entrada se lleven a cabo sobre múltiples o variadas operaciones?
8. ¿Se actualizan los archivos maestros en forma interactiva?
9. ¿Son complejas las entradas, las salidas, los archivos o peticiones?
10. ¿Es complejo el procesamiento interno?
11. ¿Se ha diseñado el código para ser reutilizable?
12. ¿Están incluidos en el diseño la conversión y la instalación?
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

**Sum Fi:** Total

Calcular

Cancelar

## Capítulo 3: Propuesta del Diseño del Sistema

### Interface 2.2



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

Puntos de Función

Tasa de Estabilidad

Índice de Madurez del Software

Duración del Proyecto

**Henry y Kafura**

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

Código Fuente

Cumplimiento de Compromiso

Prueba

Desviación del Costo

Estimación Exacta del Gasto

Especificación de Requisitos

Exportar a pdf

Exportar a hoja de cálculo

Nombre: **Henry y Kafura**

Propósito: Determinar el esfuerzo de integración y pruebas del módulo.

Fórmula:  **$MHK = longitud(i) * [Fint(i) + Fout(i)]^2$**

**Datos**

Concentración del módulo i (Fint(i)):

Expansión del módulo i (Fout(i)):

longitud (i):

Interpretación: Depende del resultado...

### Interface 2.3

## Capítulo 3: Propuesta del Diseño del Sistema



Administrar Datos del Proyecto	Calcular Indicadores	Medir Calidad	Salir
Productividad	Puntos de Función	Tasa de Estabilidad	Índice de Madurez del Software
Duración del Proyecto	Henry y Kafura	Tasa de Defectos	
Desviación Existente del Tiempo Real	<b>Diseño de Interfaz</b>	Cumplimiento de Entrega	
Riesgo	Código Fuente		
Cumplimiento de Compromiso	Prueba		
Desviación del Costo			
Estimación Exacta del Gasto			
Especificación de Requisitos			
<a href="#">Exportar a pdf</a>	<a href="#">Exportar a hoja de cálculo</a>		

Nombre: **Diseño de Interfaz**

Propósito: Determinar la calidad y facilidad de empleo de la interfaz.

Fórmula:  $DI = \frac{[(\sum_{j=1}^m \sum_{i=1}^n e_{ij} * p)] * 100}{n * m}$

Interpretación: Depende del resultado...

**Interface 2.3a)**

## Capítulo 3: Propuesta del Diseño del Sistema



**SXP**  
Metodología Ágil de Desarrollo

**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto
Calcular Indicadores
Medir Calidad
Salir

Darle valor 0 y 1 a cada pregunta

No.	Diseño de la interfaz.	1	2	3	4	5
1	El texto está en correspondencia con el contenido.	<input type="checkbox"/>				
2	El texto es claro y de fácil comprensión.	<input type="checkbox"/>				
3	Cuando contiene palabras rebuscadas le ayuda al usuario a entenderlas.	<input type="checkbox"/>				
4	El texto permite leer de manera lógica y sistemática.	<input type="checkbox"/>				
5	Es coherente.	<input type="checkbox"/>				
6	Los caracteres son claros y atractivos.	<input type="checkbox"/>				
7	El color de la letra es adecuado para la lectura.	<input type="checkbox"/>				
8	El tamaño es el adecuado.	<input type="checkbox"/>				
9	Presenta palabras calientes con sus significados.	<input type="checkbox"/>				
10	La estructura del contenido está de forma correcta.	<input type="checkbox"/>				
11	El fondo de la pantalla es el correcto.	<input type="checkbox"/>				
12	Los colores utilizados van acorde con el tema.	<input type="checkbox"/>				
13	Presenta gráficos que ayudan a entender un tema en particular.	<input type="checkbox"/>				
14	Las ilustraciones e imágenes ayudan al usuario a entender de lo que trata el tema.	<input type="checkbox"/>				
15	Las ilustraciones e imágenes están acorde con el tema.	<input type="checkbox"/>				
16	Los gráficos ayudan a entender el contenido expuesto.	<input type="checkbox"/>				
17	Presenta colores claros que ayudan al usuario a no desviarse del contenido.	<input type="checkbox"/>				
18	La combinación de colores es variada, agradable.	<input type="checkbox"/>				
19	El contraste entre: los colores de fondo, la letra, gráficos e ilustraciones es adecuado.	<input type="checkbox"/>				
20	Hay suficiente contraste entre los colores del texto o ilustraciones y el color del fondo.	<input type="checkbox"/>				

## Capítulo 3: Propuesta del Diseño del Sistema

### Interface 2.4



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores

Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

Puntos de Función

Tasa de Estabilidad

Índice de Madurez del Software

Duración del Proyecto

Henry y Kafura

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

**Código Fuente**

Cumplimiento de Compromiso

Prueba

Desviación del Costo

Estimación Exacta del Gasto

Especificación de Requisitos

Exportar a pdf

Exportar a hoja de cálculo

Nombre: **Código Fuente**

Propósito: Estimar la calidad del programa.

Fórmulas:  $N = n1 \log_2 n1 + n2 \log_2 n2$      $V = N \log_2 (n1 + n2)$   
 $L = 2 / n1 + n2 / N2$

**Datos** Subir el código:

Medida: Bits

Interpretación: Depende del resultado...

## Capítulo 3: Propuesta del Diseño del Sistema

### Interface 2.5



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

Puntos de Función

Tasa de Estabilidad

Índice de Madurez del Software

Duración del Proyecto

Henry y Kafura

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

Código Fuente

Cumplimiento de Compromiso

**Prueba**

Desviación del Costo

Estimación Exacta del Gasto

Especificación de Requisitos

Exportar a pdf

Exportar a hoja de cálculo

Nombre: **Prueba**

Propósito: Descubrir errores en el módulo antes de integrarlo a un sistema.

Fórmulas:  $NP = 1 / [(n1 / 2) * (N2 / n2)]$   
 $e = V / NP$

**Datos** Subir el código:

Interpretación: Depende del resultado...

## Capítulo 3: Propuesta del Diseño del Sistema

### Interface 3.1



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

Puntos de Función

**Tasa de Estabilidad**

Índice de Madurez del Software

Duración del Proyecto

Henry y Kafura

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

Código Fuente

Cumplimiento de Compromiso

Prueba

Desviación del Costo

Estimación Exacta del Gasto

Especificación de Requisitos

Exportar a pdf

Exportar a hoja de cálculo

Nombre: **Tasa de Estabilidad**

Propósito: Determinar si la aplicación cumplió las expectativas del cliente.

Fórmula: **TE = 1 - (NC / CHU)**

**Datos** Número de cambios solicitados (NC):

Cantidad de HU del tamaño de la aplicación (CHU):

Interpretación: Depende del resultado...

Aceptar

Cancelar

## Capítulo 3: Propuesta del Diseño del Sistema

### Interface 3.2



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

Puntos de Función

Tasa de Estabilidad

Índice de Madurez del Software

Duración del Proyecto

Henry y Kafura

**Tasa de Defectos**

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

Código Fuente

Cumplimiento de Compromiso

Prueba

Desviación del Costo

Estimación Exacta del Gasto

Especificación de Requisitos

Exportar a pdf

Exportar a hoja de cálculo

Nombre: **Tasa de Defectos**

Propósito: Permite conocer el número de defectos a los miembros del equipo de desarrollo.

Fórmula: **TD = ND / CHU**

**Datos**

Número de defectos (ND):

Cantidad de HU (CHU):

Interpretación: Depende del resultado...

Aceptar

Cancelar

## Capítulo 3: Propuesta del Diseño del Sistema

### Interface 3.3



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

Puntos de Función

Tasa de Estabilidad

Índice de Madurez del Software

Duración del Proyecto

Henry y Kafura

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

**Cumplimiento de Entrega**

Riesgo

Código Fuente

Cumplimiento de Compromiso

Prueba

Desviación del Costo

Estimación Exacta del Gasto

Especificación de Requisitos

Exportar a pdf

Exportar a hoja de cálculo

Nombre: **Cumplimiento de Entrega**

Propósito: Considerar los compromisos de entrega del producto.

Fórmula:  **$CPE = (TR + 100) / TE$**

**Datos** Tiempo real dedicado a la entrega (PP):

Tiempo estimado para la entrega (TE):

Medida: Minutos

Interpretación: Depende del resultado...

Aceptar

Cancelar

## Capítulo 3: Propuesta del Diseño del Sistema

### Interface 4.1



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



Administrar Datos del Proyecto

Calcular Indicadores

Medir Calidad

Salir

Productividad

Puntos de Función

Tasa de Estabilidad

**Índice de Madurez del Software**

Duración del Proyecto

Henry y Kafura

Tasa de Defectos

Desviación Existente del Tiempo Real

Diseño de Interfaz

Cumplimiento de Entrega

Riesgo

Código Fuente

Cumplimiento de Compromiso

Prueba

Desviación del Costo

Estimación Exacta del Gasto

Especificación de Requisitos

Exportar a pdf

Exportar a hoja de cálculo

Nombre: **Índice de Madurez del Software**

Propósito: Determinar la estabilidad del producto.

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

**Datos**

Número de módulos de la versión actual (Mt):

Número de módulos de la versión actual cambiados (Fc):

Número de módulos de la versión actual añadidos (Fa):

Número de módulos de la versión anterior borrados en la versión actual (Fd):

Interpretación: Depende del resultado...

Aceptar

Cancelar

## Capítulo 3: Propuesta del Diseño del Sistema

Tabla 3.3. HU4\_Mostrar datos de la calidad

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre Historia de Usuario:</b> Mostrar datos de la calidad del proyecto
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b>	<b>Iteración Asignada:</b>
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Alta	<b>Puntos Reales:</b>
<p><b>Descripción:</b> Le muestra al usuario mediante un gráfico el nivel de calidad que va alcanzando el proyecto por fase.</p>	
<p><b>Observaciones:</b> Se muestra un gráfico en la pestaña medir calidad que contiene en una escala de 1 a 100 la calidad de cada fase del proyecto, estos datos se obtienen de la base de datos, de la tabla resultados promediados. Se propone para crear el gráfico usar las librerías <b>HighCharts</b>, que permite la graficación en la web escrita completamente en JavaScript y no requiere complementos adicionales instalados en los navegadores para realizar su funcionamiento, muestra los datos auxiliándose de diversos tipos de gráficos con excelente calidad visual, entre ellos los gráficos de pastel, líneas, barras, columnas, y presenta licencia de software libre para uso no comercial; o la librería <b>eZComponents</b> de Symfony, cuenta con un componente denominado <u>Graph</u> que permite crear diferentes tipos de gráficos profesionales.</p>	
<b>Prototipo de interface:</b>	

## Capítulo 3: Propuesta del Diseño del Sistema

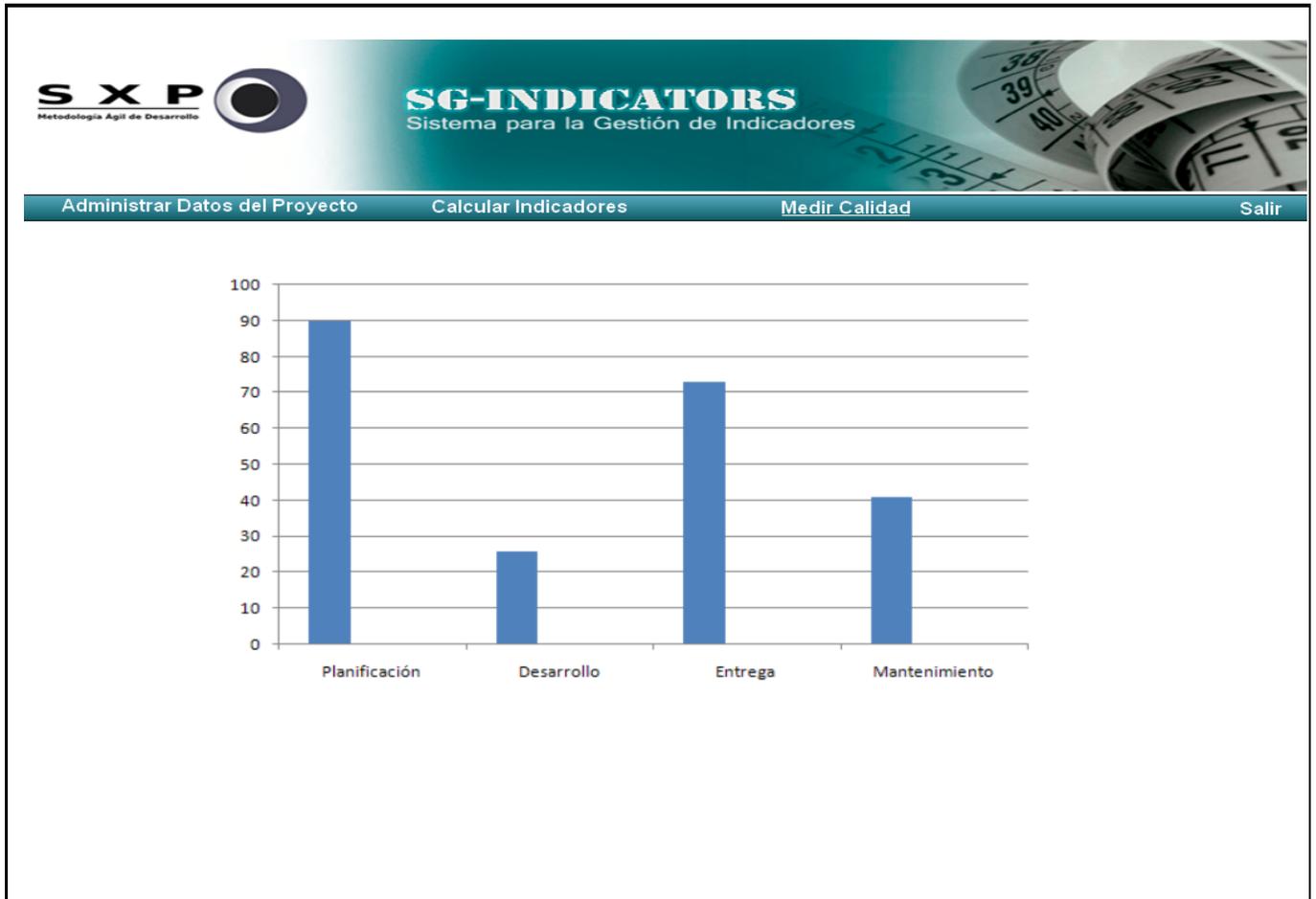


Tabla 3.4. HU6\_Exportar Archivo

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Exportar Archivos
Modificación de Historia de Usuario Número:	

## Capítulo 3: Propuesta del Diseño del Sistema

<b>Usuario:</b>	<b>Iteración Asignada:</b>
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b>
<b>Descripción:</b> Habilita la opción al usuario de exportar los resultados en formato hoja de cálculo o pdf.	
<b>Observaciones:</b> Exporta los resultados que se tienen guardados en la base de datos en la tabla resultados a formato pdf u hoja de cálculo. Se propone para exportarlo el uso de las librerías <b>Cpdf</b> o <b>fpdf</b> para exportar formato pdf y <b>phpExcel</b> para formato hoja de cálculo; o <b>laiguExtGridPlugin</b> , plugin para Symfony que permite que todos los datos mostrados se puedan exportar a formato pdf o Excel.	
<b>Prototipo de interface:</b>	

Para consultar el resto de las historias de usuarios (ver Anexo 3).

### 3.7. Diagrama de Clases de Diseño con Estereotipos Web

Para el desarrollo del sistema se propuso el patrón MVC, por tal razón los diagramas quedan divididos en 3 paquetes y 2 subsistemas que son del framework que se utilizará, Symfony. El paquete controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el paquete vista y modelo, cuenta con un controlador frontal que se encarga de atender todas las peticiones y en dependencia de la URL<sup>18</sup> que tengan manda a ejecutar la acción correspondiente. El paquete vista contiene las clases de diseño con estereotipos web, que permite representar cada unas de las clases que conforman el sistema con los estereotipos “Server Page”, “Client Page” y “Form” empleados para el código

---

<sup>18</sup> Uniform Resource Locator

## Capítulo 3: Propuesta del Diseño del Sistema

---

servidor, código cliente y formularios respectivamente. Transforma el modelo en una página web que permite al usuario interactuar con ella; además cuenta con el archivo layout.php que almacena el código HTML que es común en todas las páginas de la aplicación para no tener que repetirlo. El paquete modelo representa la información con que trabaja la aplicación, se relaciona con el subsistema Propel, encargado de transformar del lenguaje orientado a objeto al modelo relacional. El subsistema componentes Symfony es un enrutador que permite la relación entre la vista y el controlador para poder realizar las peticiones del usuario. A continuación se muestran los diagramas de las funcionalidades críticas del sistema:

# Capítulo 3: Propuesta del Diseño del Sistema



Figura 3. Diagrama Clases de Diseño con Estereotipos Web de la HU\_Calculador Indicadores

## Capítulo 3: Propuesta del Diseño del Sistema

---

### Descripción de las clases en cada paquete y sus funcionalidades

#### Paquete Controlador

**Clase Calcular Indicadores:** encargada de realizar los cálculos de los indicadores.

- Calcular Indicador Productividad
- Calcular Indicador Duración Proyecto
- Calcular Indicador Desviación Existente
- Calcular Indicador Riesgo
- Calcular Indicador Cumplimiento de Compromiso
- Calcular Indicador Desviación Costo
- Calcular Indicador Estimación Exacta del Gasto
- Calcular Indicador Especificación de Requisitos
- Calcular Indicador Puntos de Función
- Calcular Indicador Henry y Kafura
- Calcular Indicador Diseño de Interfaz
- Calcular Indicador Código Fuente
- Calcular Indicador Prueba
- Calcular Indicador Índice de Madurez del Software
- Calcular Indicador Tasa de Estabilidad
- Calcular Indicador Cumplimiento de Entrega

**Clase Controlador Frontal:** encargada de realizar las tareas comunes a todos los controladores, ejemplo, manejo de la seguridad, de las peticiones de los usuarios, carga de la configuración de la aplicación y delega la responsabilidad de responder a las peticiones al módulo específico que tiene la acción enviada en la petición.

#### Paquete Vista

**Clase Layout:** encargada de almacenar el código HTML que es común en todas las páginas de la aplicación para no tener que repetirlo.

## Capítulo 3: Propuesta del Diseño del Sistema

---

**Clases Server Page:** encargada de construir las clases client page para cada indicador.

**Clases Client Page:** son las interfaces que se le muestran al usuario y contienen los formularios de indicador.

**Formularios:** es donde el usuario introduce los datos del indicador a procesar en los cálculos.

**Paquete Modelo:** contiene todas las tablas en las que se guardan los datos de cada indicador en la base de datos, por tanto hay una tabla para cada uno.

## Capítulo 3: Propuesta del Diseño del Sistema

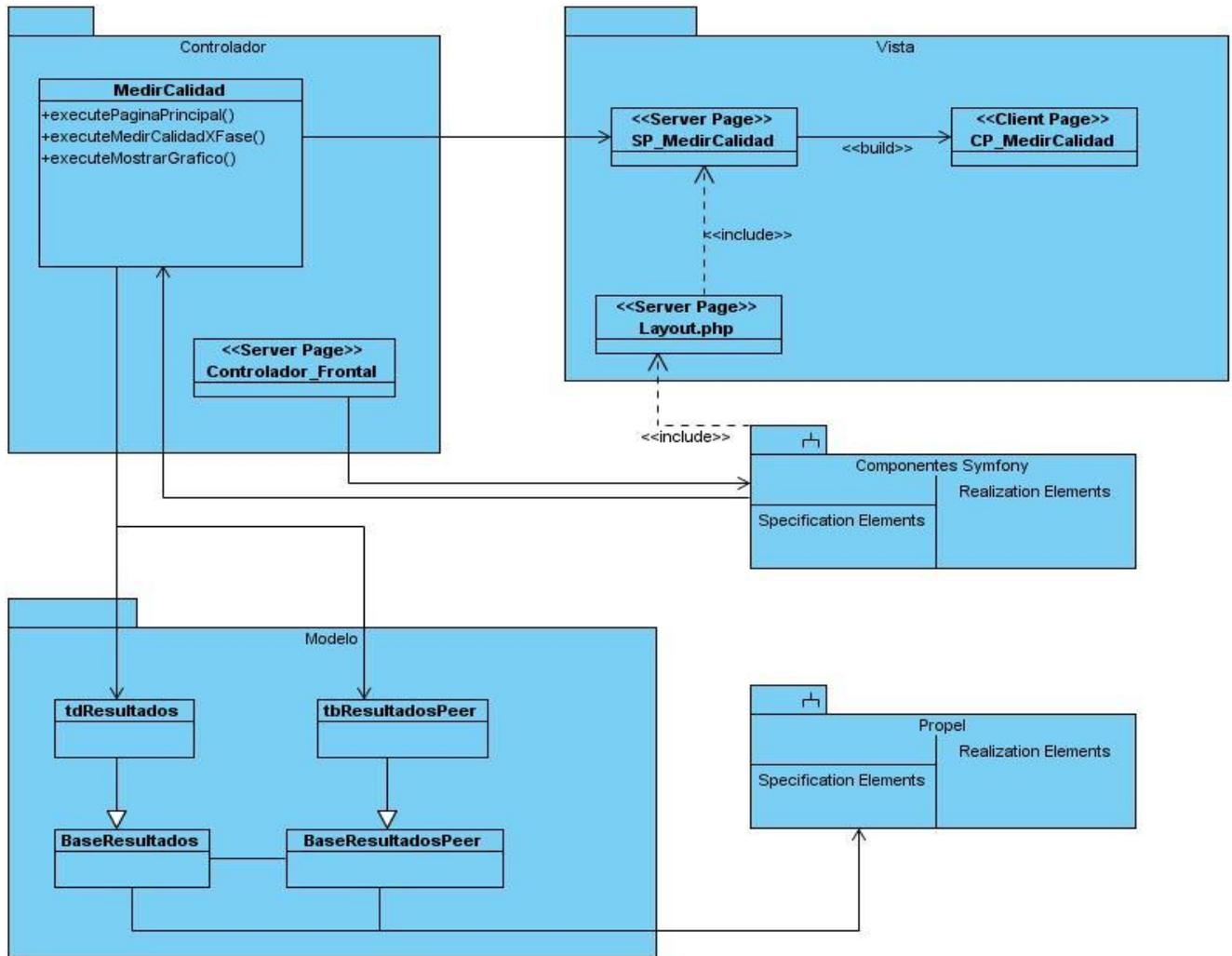


Figura 4. Diagrama Clases de Diseño con Estereotipos Web de la HU\_Medir Calidad

Descripción de las clases en cada paquete y sus funcionalidades

### Paquete Controlador

**Clase Medir Calidad:** encargada de realizar la medición de la calidad por fase para ser mostradas en un gráfico.

## Capítulo 3: Propuesta del Diseño del Sistema

---

- Medir calidad por fase
- Mostrar gráfico

**Clase Controlador Frontal:** encargada de realizar las tareas comunes a todos los controladores, ejemplo, manejo de la seguridad, de las peticiones de los usuarios, carga de la configuración de la aplicación y delega la responsabilidad de responder a las peticiones al módulo específico que tiene la acción enviada en la petición.

### **Paquete Vista**

**Clase Layout:** encargada de almacenar el código HTML que es común en todas las páginas de la aplicación para no tener que repetirlo.

**Clases Server Page:** encargada de construir la clase client page que muestra el gráfico de la calidad.

**Clases Client Page:** encargada de mostrarle al usuario el gráfico de la calidad por cada fase.

**Paquete Modelo:** contiene la tabla que guarda los resultados de cada indicador en la base de datos.

## Capítulo 3: Propuesta del Diseño del Sistema

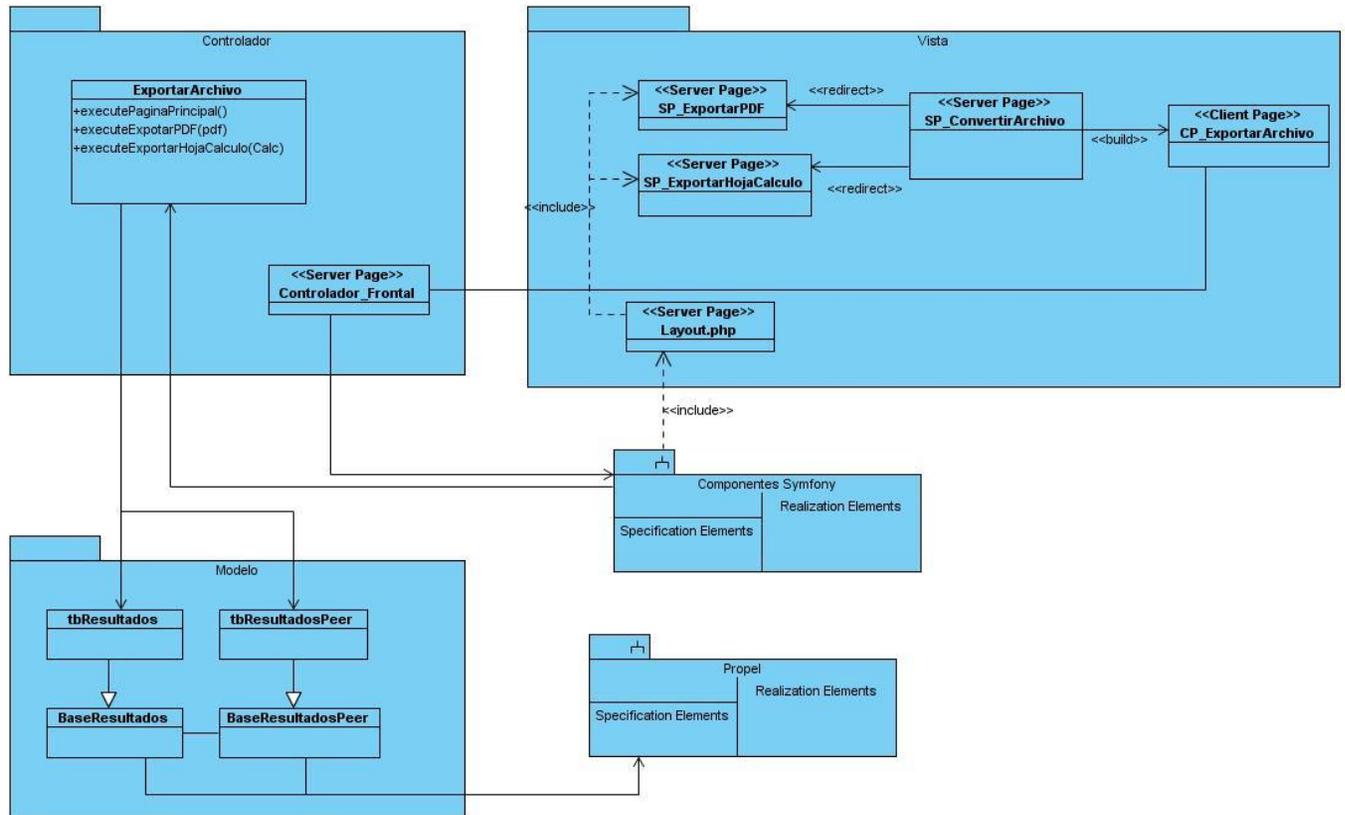


Figura 5. Diagrama Clases de Diseño con Estereotipos Web de la HU\_Exportar Archivos

### Descripción de las clases en cada paquete y sus funcionalidades

#### Paquete Controlador

**Clase Exportar Archivos:** encargada de exportar los resultados de cada indicador a formato pdp u hoja de cálculo para futuras revisiones.

- Exportar a pdf
- Exportar a hoja de cálculo

**Clase Controlador Frontal:** encargada de realizar las tareas comunes a todos los controladores, ejemplo, manejo de la seguridad, de las peticiones de los usuarios, carga de la configuración de la aplicación y

## Capítulo 3: Propuesta del Diseño del Sistema

---

delega la responsabilidad de responder a las peticiones al módulo específico que tiene la acción enviada en la petición.

### **Paquete Vista**

**Clase Layout:** encargada de almacenar el código HTML que es común en todas las páginas de la aplicación para no tener que repetirlo.

**Clases Server Page:** encargada de construir la clase client page que brinda la opción de exportar archivo.

**Clases Client Page:** encargada de mostrarle al usuario la opción de exportar archivo a formato pdf o formato hoja de cálculo.

**Paquete Modelo:** contiene la tabla que guarda los resultados de cada indicador en la base de datos para exportarlos.

Para consultar el resto de los diagramas de clases de diseño con estereotipos web (ver Anexo 4).

### **3.8. Beneficios del Sistema SG – Indicators**

Con el uso de SG-Indicators en los proyectos productivos que utilizan la metodología ágil SXP se mejorará la calidad en cada fase de desarrollo y se obtendrá mayor satisfacción del cliente debido a que se podrá:

- Medir el rendimiento de los softwares que se realizan en los proyectos.
- Comprender qué afecta la calidad del producto, así como determinar la utilidad y funcionalidad del mismo.
- Contar con una documentación menos exhaustiva, ahorrando tiempo de trabajo e incrementando la productividad del equipo.
- Mantener actualizado todo los miembros del grupo de trabajo al consultar la información según los roles que presentan.
- Alcanzar mejor planificación, estimación y organización en el trabajo del proyecto, al permitir generar reportes que ayudarán a predecir y planear el trabajo, posibilitando que posteriormente sean comparados con resultados más recientes para tomar decisiones factibles para el proceso.

### Conclusiones

Con la realización del presente trabajo se dio cumplimiento a cada uno de los objetivos trazados, destacando de manera general las siguientes conclusiones:

- El marco conceptual de los indicadores, demostró que son una herramienta indiscutible para ayudar a mantener el control y la calidad de los artefactos que se generan durante el desarrollo del software.
- El empleo de los indicadores arrojó beneficios para los proyectos seleccionados. Los resultados obtenidos sirvieron de guía para conocer el estado de la calidad en que se encuentran estos productos.
- El estudio de las herramientas existentes para la gestión de indicadores, permitió la realización de la propuesta de un sistema para gestionar los de la metodología SXP.

### Recomendaciones

Para dar continuidad a esta investigación, se proponen las siguientes recomendaciones:

- Realizar la implementación del sistema SG-Indicators propuesto para SXP.
- Añadir como nueva funcionalidad al sistema de gestión de indicadores, conectarse con el gestor de proyecto que se esté utilizando en ese momento, para agilizar el proceso de recopilación de datos.
- Profundizar sobre las necesidades de la utilización de los indicadores.
- Continuar el estudio del tema para actualizar los indicadores de SXP, y así obtener mejoras en futuras versiones del sistema.

## Referencias Bibliográficas

---

### Referencias Bibliográficas

- [1] **Rodriguez, A y Sisinta, A.** *Propuesta de técnicas de estimación y métricas para la Metodología SXP.* Ciudad de La Habana : s.n., 2009.
- [2] **Lovelle, Juan M.** Calidad del Software. [En línea] 21 de Octubre de 2009. [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF)
- [3] **Minguet, Jesus M y Hernández, Juan F.** *La calidad del Software y su medida.* Madrid : CENTRO DE ESTUDIOS RAMON ARECES, S.A., 2003.
- [4] Ídem [1]
- [5] **De Amescua, A, Lloréns, J y García, A.** SPICE: UN MARCO PARA LA EVALUACIÓN DE PROCESOS SOFTWARE. [En línea] <http://www.ie.inf.uc3m.es/grupo/Investigacion/LineasInvestigacion/Articulos/spice.doc>
- [6] **Villagra, S.** Una Introducción a CMMI. [En línea] 2006. [www.sergiovillagra.com/Contenidos/Recursos/WP03%20Una%20Introduccion%20a%20CMMI.pdf](http://www.sergiovillagra.com/Contenidos/Recursos/WP03%20Una%20Introduccion%20a%20CMMI.pdf)
- [7] **Pressman, R.** *Un Enfoque Práctico.* s.l. : 5ta Edición, 2002.
- [8] Ídem [7]
- [9] Ídem [7]
- [10] Ídem [7]
- [11] **Kan, Stephen H.** *Metrics and Models in Software Quality Engineering.* s.l. : 2da Edición, 2006.
- [12] **Humphrey, Watts S.** *A Discipline for Software Engineering.* s.l. : 1ra Edición, 1995.
- [13] **Pussacq, Juan P.** *Indicadores Objetivos para control de proyectos de Desarrollo de Software.* Buenos Aires : s.n., 2004.
- [14] *Visual Paradigm.* [En línea] Headquarters, 2010. <http://www.visual-paradigm.com/product/vpum/>

## Referencias Bibliográficas

---

- [15] **Cabello, P.** Prototipos rápidos con Pencil. [En línea] 7 de marzo de 2010. <http://mozillalinksespanol.wordpress.com/2008/07/07/prototipos-rapidos-con-pencil/>
- [16] **Serrano, Jose M.** Verificación y validación. *Universidad de Jaen*. [En línea] 2009. <http://www.di.ujaen.es/asignaturas/computacionestadistica/pdfs/tema6.pdf>
- [17] **Curbelo, Mario A.** Los métodos de evaluación y decisión multicriterio. *Monografías*. [En línea] 2006. <http://www.monografias.com/trabajos40/decision-multicriterio/decision-multicriterio.shtml>
- [18] **Serrano, M.** *Método de Definición de Métricas*. Universidad de Castilla-La Mancha, s.l. : 2006.
- [19] **Penalver, Gladys M.** *MA-GMPR-UR2 Metodología ágil para proyectos de software libre*. Ciudad de La Habana : s.n., 2008.
- [20] **Cordovi, N.** *Propuesta de un Estándar y Métricas para el control del Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela*. Ciudad de la Habana : UCI, 2008.
- [21] **Jim.** SourceMonitor Version 2.5. *Campwood Software*. [En línea] 2010. <http://www.campwoodsw.com/sourcemonitor.html>.
- [22] **Palacios, J.** Gratis para el área de Medición y Análisis (CMM - CMMI) . *Navegapolis.net*. [En línea] 29 de Abril de 2007. <http://www.navegapolis.net/content/view/609/86/>.
- [23] **Sprintómetro gratuito.** *Navegapolis.net*. [En línea] 15 de Noviembre de 2008. <http://www.navegapolis.net/content/view/836/87/>.
- [24] **Sarmiento, D y Rodríguez, R.** *Sistema para la gestión de métricas*. Ciudad de la Habana : UCI, 2009.

### Bibliografía

- Aguilar A., y Galanga G. Métricas del proceso y proyecto - Procesos de Ingeniería de software. 2008. [cited 23 Marzo 2010]. [http://www.slideshare.net/galo\\_priva/mtricas-del-proceso-y-proyecto-procesos-de-ingeniera-de-software](http://www.slideshare.net/galo_priva/mtricas-del-proceso-y-proyecto-procesos-de-ingeniera-de-software)
- Apa, C., Robaina, R., De León, S., y Vallespir D. Conceptos de Ingeniería de Software Empírica. 2010. [www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR1002.pdf](http://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR1002.pdf)
- Cordovi N. "Propuesta de un Estándar y Métricas para el control del Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela". 2008.
- Curbelo, Mario A. Los métodos de evaluación y decisión multicriterio. *Monografías*. [En línea] 2006. <http://www.monografias.com/trabajos40/decision-multicriterio/decision-multicriterio.shtml>
- De Amascua A., Lloréns J., y García A. SPICE: Un Marco para la Evaluación de Procesos de Software. 2006. [cited 19 Marzo 2010]. [www.ie.inf.uc3m.es/grupo/Investigacion/LineasInvestigacion/Articulos/spice.doc](http://www.ie.inf.uc3m.es/grupo/Investigacion/LineasInvestigacion/Articulos/spice.doc)
- Dos M. El Test de Turing. 2007. [cited 19 Marzo 2010]. <http://axxon.com.ar/rev/170/c-170divulgacion.htm>
- Dreger J. Function Point Analysis. 1989 [cited 23 Marzo 2010]. <http://www.sc.ehu.es/jiwdocoj/mmis/fpa.htm>
- Fuentes C. M, y Teuber H. P. Modelo de calidad CMMI. [cited 23 Marzo 2010]. <http://www.monografias.com/trabajos57/modelo-calidad-cmmi/modelo-calidad-cmmi.shtml#xresum>.
- Fundamentos de ingeniería de software. [cited 23 Marzo 2010]. <http://148.202.148.5/cursos/cc321/fundamentos/temario.htm>
- GForge-Facultad10: Bienvenid@s. [cited 23 Marzo 2010]. <http://svn.gforge.f10.uci.cu/svn/unicornios/>
- Gonzalo M. Métricas Internas de la Calidad del Producto de Software. Marzo 2006. [cited 23 Marzo 2010]. [http://www.mena.com.mx/gonzalo/maestria/calidad/presenta/iso\\_9126-3/](http://www.mena.com.mx/gonzalo/maestria/calidad/presenta/iso_9126-3/)

## Bibliografía

---

- Humphrey, Watts S. A Discipline for Software Engineering. 1ra 1995 [cited 30 Marzo 2010].
- Infante C. G. Propuesta de tareas para elevar la calidad del proceso de desarrollo en el proyecto AKADEMOS. 2007. <http://biblioteca.uci.cu/sbd/biuci/index.html>
- Kan, Stephen H. Metrics and Models in Software Quality Engineering. 2da 2006 [cited 30 Marzo 2010].
- Lopez E. Estimación Basada en Puntos de Función y Soluciones Híbridas. [cited 23 Marzo 2010]. <http://www.monografias.com/trabajos55/estimacion-por-puntos-de-funcion/estimacion-por-puntos-de-funcion2.shtml?monosearch>
- Lovelle Juan M. Calidad de Software. Octubre 1999. [cited 19 Marzo 2010]. [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF).
- Minguet, Jesus M y Hernández, Juan F. La calidad del Software y su medida. Madrid : CENTRO DE ESTUDIOS RAMON ARECES, S.A., 2003.
- Morales N. Procedimiento para realizar la medición y análisis en un proyecto. 2008. [cited 23 Marzo 2010]. <http://www.monografias.com/trabajos64/procedimiento-medicion-analisis-proyecto/procedimiento-medicion-analisis-proyecto2.shtml?monosearch>
- Park R. E , Goethert W. B, y Florac W. A. Goal-Driven Software Measurement - A Guidebook. 1996.
- Pererira, B; Ayaach, F; Quintero, H; Granadillo, I y Bustamante, J. Métricas de Calidad de Software. <http://www ldc.usb.ve/~abianc/materias/ci4712/metricas.pdf>
- Pressman R. Ingeniería de Software. Un enfoque práctico. 4ta 1997.
- Pressman R. Ingeniería de Software. Un Enfoque Práctico. 5ta [México]: Área de Informática y Computación, 2002
- Pussacq, Juan P. Indicadores Objetivos para control de proyectos de Desarrollo de Software. Buenos Aires : s.n., 2004.
- Reynoso I, Batista E, y Martínez H. Métricas. [cited 23 Marzo 2010]. <http://www.it.aut.uah.es/juanra/docencia/GestiondeProyectos/traspas/Metricas.pdf>

## Bibliografía

---

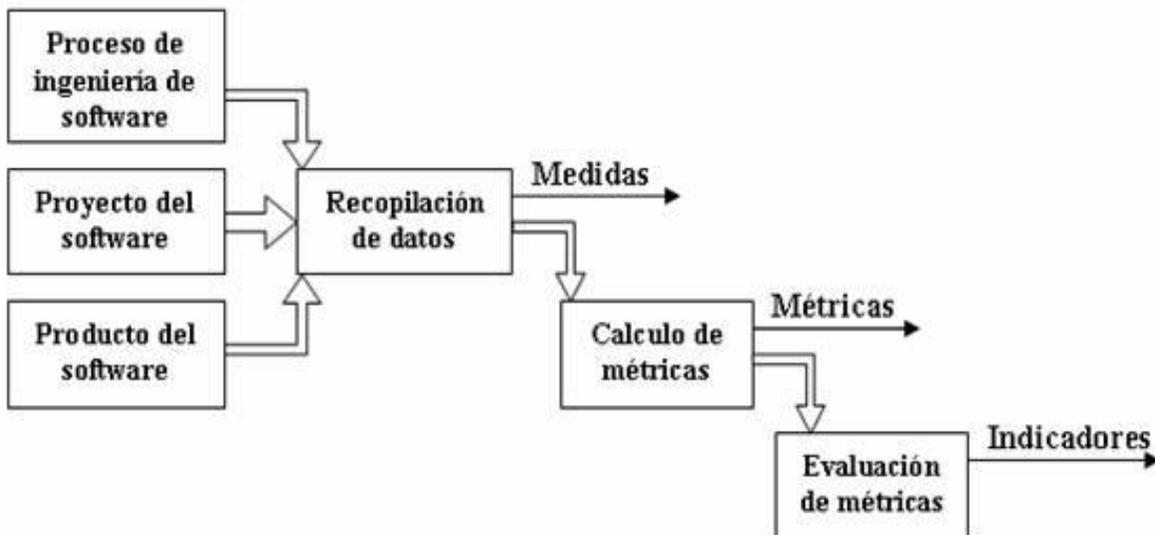
- Sarmiento D, y Rodríguez R. Sistema para la gestión de métricas. 2009.
- Serrano, Jose M. Verificación y validación. *Universidad de Jaen*. [En línea] 2009. <http://wwwdi.ujaen.es/asignaturas/computacionestadistica/pdfs/tema6.pdf>
- Sicilia M. Métricas relacionadas con el proceso. [cited 23 Marzo 2010]. <http://cnx.org/content/m17467/latest/>
- SlideShare Inc. Métricas Técnicas Del Software. [cited 23 Marzo 2010]. <http://www.slideshare.net/juic/metricas-tecnicas-del-software-presentation>
- Soto L. Métricas Basadas En Función. [cited 23 Marzo 2010]. <http://www.mitecnologico.com/Main/MetricasBasadasEnFuncion>.
- Vega C, Rivera Laura S, y García A. Mejores Prácticas para el Establecimiento y Aseguramiento de Calidad de Software. Electrónica Febrero 2008 <http://www.eumed.net>
- Villagra S. Una Introducción a CMMI. 2006. [cited 19 Marzo 2010]. [www.sergiovillagra.com/Contenidos/Recursos/WP03%20Una%20Introduccion%20a%20CMMI.pdf](http://www.sergiovillagra.com/Contenidos/Recursos/WP03%20Una%20Introduccion%20a%20CMMI.pdf).
- Yáñez, C. Verificación y Validación de Modelos para la Estimación del Esfuerzo de Desarrollo de Software. 2008.
- Yera L, y García Carmen R . “Métricas para Evaluar la Calidad en los Proyectos del Polo de Telecomunicaciones.”. Junio 2009.

**Anexos**

**Anexo 1. Niveles de representación continua y escalonada.**

	<i>Representación Continua</i>	<i>Representación Escalonada</i>
	Nivel de Capacidad	Nivel de Madurez
Nivel 0	Incompleto	-
Nivel 1	Realizado	Inicial
Nivel 2	Manejado	Manejado
Nivel 3	Definido	Definido
Nivel 4	Manejado cuantitativamente	Manejado cuantitativamente
Nivel 5	Optimizando	Optimizando

**Anexo 2. Proceso de recopilación de métricas del software**



## Anexo 3. Historias de Usuarios

### Anexo 3.1. HU\_Administrar Proyecto

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre Historia de Usuario:</b> Administrar Proyecto
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b>	<b>Iteración Asignada:</b>
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b>
<p><b>Descripción:</b> Esta sección garantiza toda la administración de los proyectos, permite que se agreguen y se modifiquen los datos básicos de los mismos en el sistema, así como, poder listarlos y eliminar los existentes.</p>	
<p><b>Observaciones:</b> Para agregar un proyecto es necesario que el usuario llene los siguientes datos: Nombre del proyecto, Descripción del mismo (opcional), Clasificación (seleccionar en los checkbox de acuerdo a los 5 tipos de proyectos del Polo SWL dentro de cuales se desarrolla el que se está registrando), Fase (se escoge en el combobox una de las siguientes fases: Definición - Planificación, Desarrollo, Entrega o Mantenimiento) de acuerdo a la fase seleccionada se le habilitan los indicadores a calcular para la misma, Fecha de inicio y Fin del proyecto, y Establecer una contraseña para mayor seguridad. (Ver Interface 1)</p> <p>En caso de que ocurra algún error o cambio en los datos introducidos, el usuario tiene la opción de modificarlos, si el proyecto avanza a la siguiente fase, se cambia la misma y entonces se habilitan los nuevos indicadores a calcular. (Ver Interface 2)</p>	

En la opción listar proyecto, se puede eliminar el que se desea, en caso de continuar trabajando sobre algún proyecto, se selecciona el mismo y se pasa a la opción de calcular indicadores. (Ver Interface 3)

## Prototipo de interface:

### Interface 1: Introducir datos del proyecto

**SXP** Metodología Ágil de Desarrollo
 **SG-INDICATORS**  
 Sistema para la Gestión de Indicadores

Administrar Datos del Proyecto	Calcular Indicadores	Medir Calidad	Salir
Introducir Datos del Proyecto	Modificar Datos del Proyecto	Listar Proyectos	

Nombre:

Descripción:

Clasificación:

Fase:

Fecha Inicio:

Fecha Fin:

Password:

Interface 2: Modificar datos del proyecto

**SXP** Metodología Ágil de Desarrollo

**SG-INDICATORS**  
Sistema para la Gestión de Indicadores

Administrar Datos del Proyecto    Calcular Indicadores    Medir Calidad    Salir

Introducir Datos del Proyecto    **Modificar Datos del Proyecto**    Listar Proyectos

Nombre:

Descripción:

Clasificación:  ▼

Fase:  ▼

Fecha Inicio:

Fecha Fin:

Password:

Interface 3: Listar o eliminar proyectos



**SG-INDICATORS**  
Sistema para la Gestión de Indicadores



<a href="#">Administrar Datos del Proyecto</a>	<a href="#">Calcular Indicadores</a>	<a href="#">Medir Calidad</a>	<a href="#">Salir</a>
<a href="#">Introducir Datos del Proyecto</a>	<a href="#">Modificar Datos del Proyecto</a>	<a href="#">Listar Proyectos</a>	

Proyectos

- SistClon
- NovaDesk
- Yarey



Pulse Aquí para Eliminar

Anexo 3.2. HU\_Autenticarse

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre Historia de Usuario:</b> Acceder al sistema
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b>	<b>Iteración Asignada:</b>
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b>
<b>Descripción:</b> Permite que le usuario se registre en el sistema.	
<b>Observaciones:</b> Se debe introducir usuario y contraseña para acceder al sistema (Ver Interface 1), y puede establecer una nueva si se le olvidó la contraseña (Ver Interface 2).	
<b>Prototipo de interface:</b>	
Interface 1	

**SXP**  
Metodología Apt de Desarrollo

**SGP-INDICATORS**  
Sistema para la Gestión de Indicadores

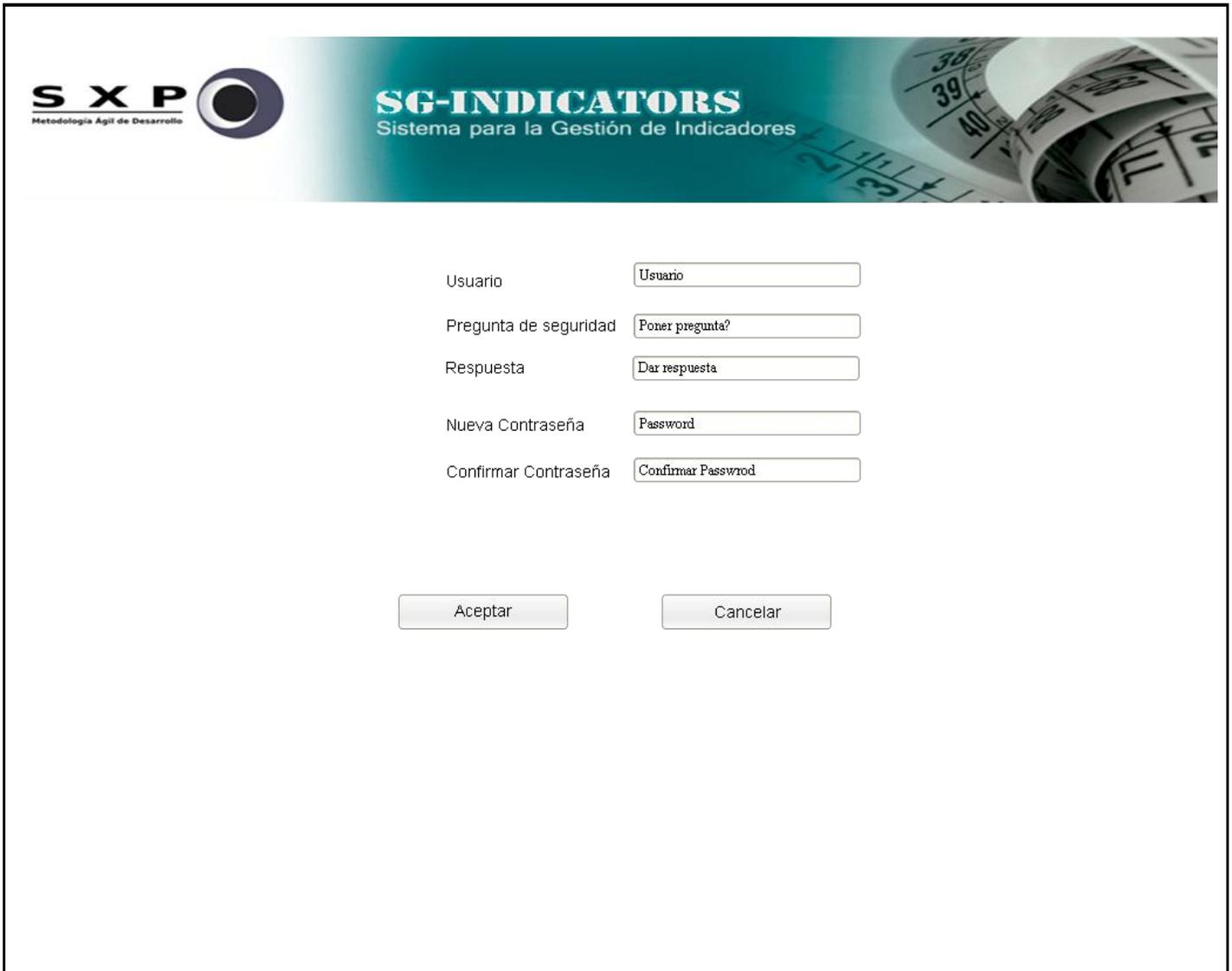
Nombre de Usuario:

Contraseña:

Enviar

[Registrarse](#) | [¿Olvidó su Contraseña?](#)

Interface 2



The image shows a login form for the SG-INDICATORS system. At the top left is the SXP logo with the text 'Metodología Ágil de Desarrollo'. To the right is the title 'SG-INDICATORS' and subtitle 'Sistema para la Gestión de Indicadores'. The form contains five input fields: 'Usuario', 'Poner pregunta?' (for security question), 'Dar respuesta' (for security answer), 'Password', and 'Confirmar Password'. At the bottom are 'Aceptar' and 'Cancelar' buttons. The background features a teal gradient and a close-up of a measuring tape.

**S X P**  
Metodología Ágil de Desarrollo

**SG-INDICATORS**  
Sistema para la Gestión de Indicadores

Usuario

Pregunta de seguridad

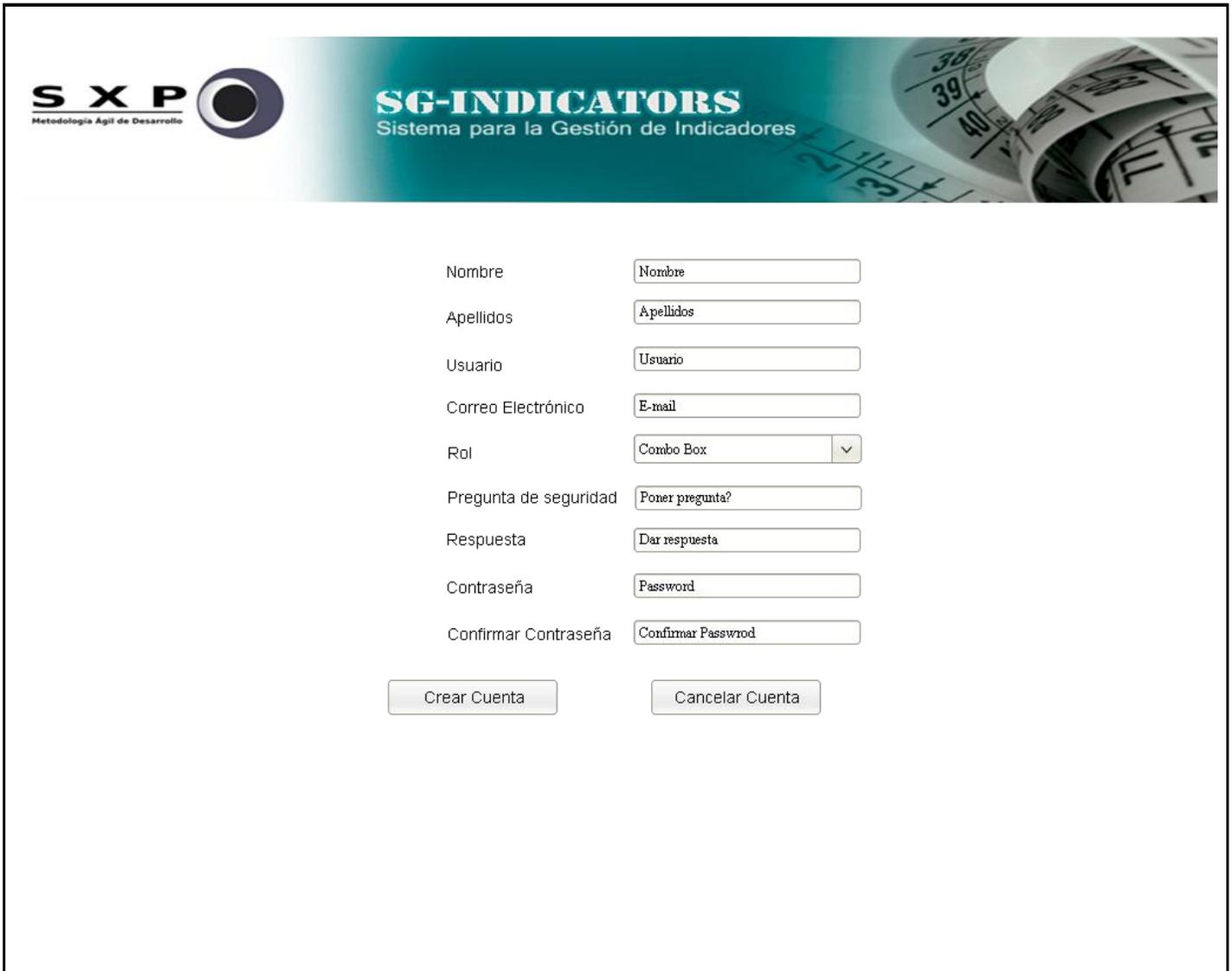
Respuesta

Nueva Contraseña

Confirmar Contraseña

Anexo 3.3. HU\_Crear cuenta de usuario

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre Historia de Usuario:</b> Crear cuenta de usuario
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b>	<b>Iteración Asignada:</b>
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Media	<b>Puntos Reales:</b>
<b>Descripción:</b> Le permite al usuario no registrado crearse una cuenta para poder acceder al sistema.	
<b>Observaciones:</b> Se debe introducir Nombre y Apellidos, Usuario para autenticarse, Correo electrónico, Rol dentro del proyecto que se selecciona dentro del checkbox (Líder del proyecto, Analista, Programador y Consultor para visitar el sistema, no tiene permisos solo puede consultar la información), una pregunta para la seguridad del perfil, su respuesta, Contraseña y confirmar la misma.	
<b>Prototipo de interface:</b>	



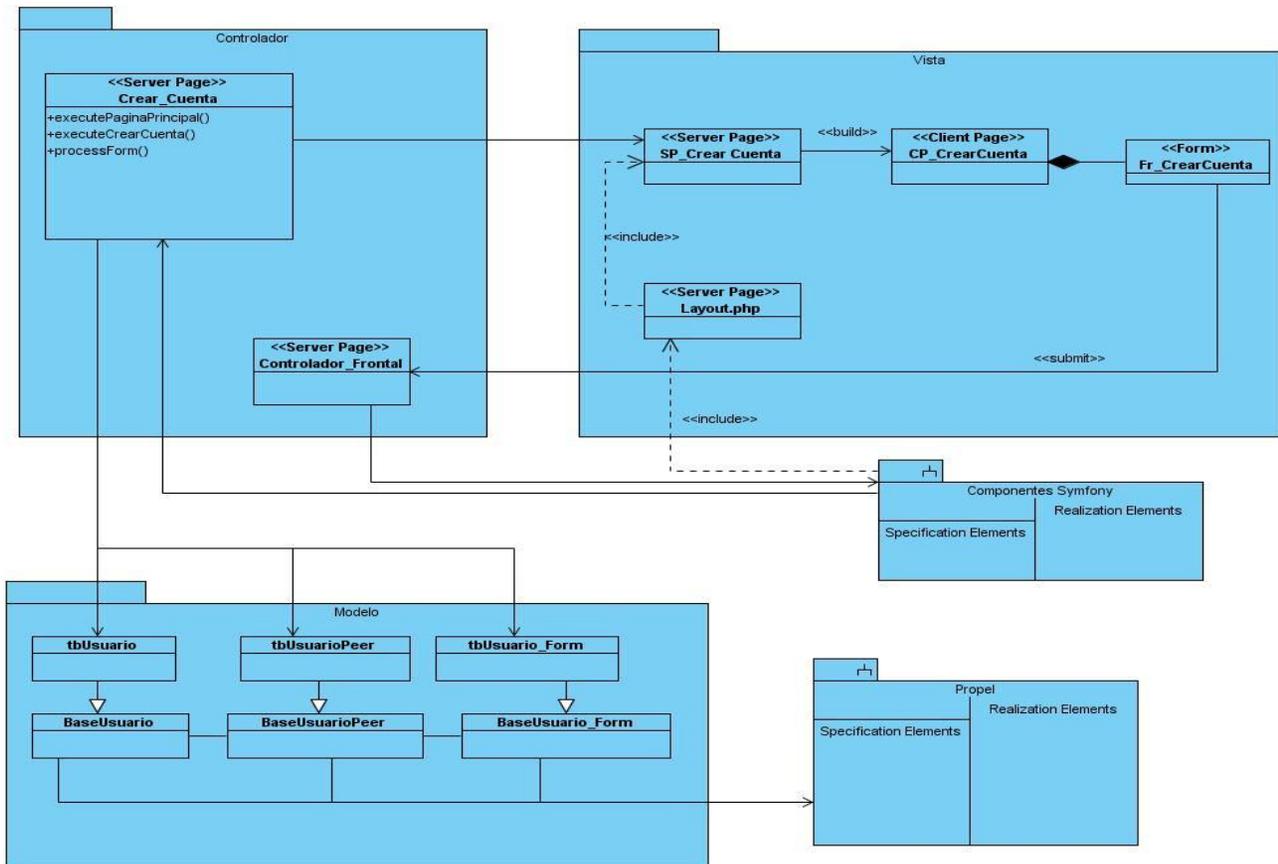
The image shows a registration form for the SG-INDICATORS system. At the top left is the SXP logo with the text 'Metodología Ágil de Desarrollo'. To its right is the title 'SG-INDICATORS' and subtitle 'Sistema para la Gestión de Indicadores'. The background features a teal gradient and a close-up of a measuring tape. The form fields are as follows:

Nombre	<input type="text" value="Nombre"/>
Apellidos	<input type="text" value="Apellidos"/>
Usuario	<input type="text" value="Usuario"/>
Correo Electrónico	<input type="text" value="E-mail"/>
Rol	<input type="text" value="Combo Box"/> ▼
Pregunta de seguridad	<input type="text" value="Poner pregunta?"/>
Respuesta	<input type="text" value="Dar respuesta"/>
Contraseña	<input type="text" value="Password"/>
Confirmar Contraseña	<input type="text" value="Confirmar Passwrod"/>

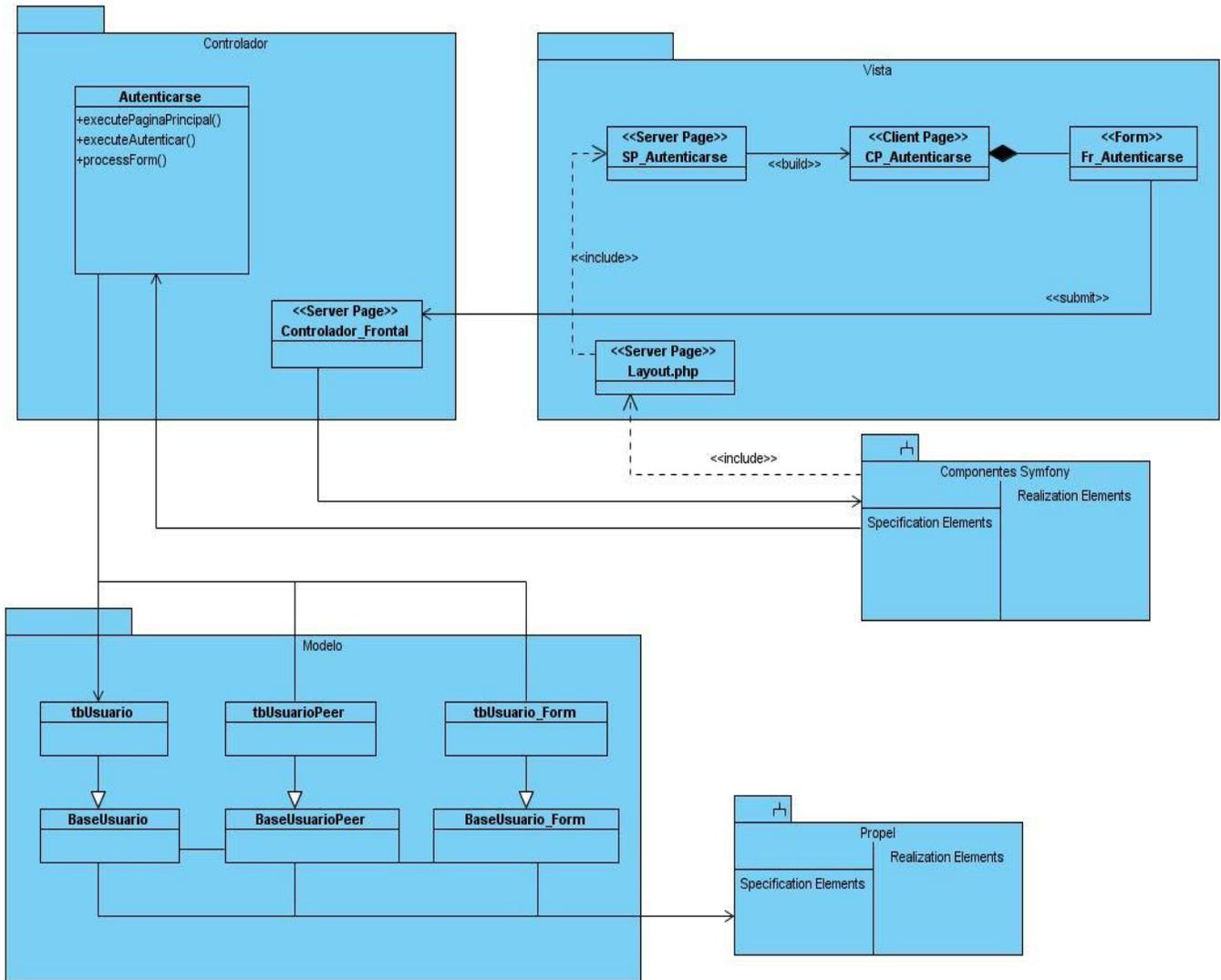
At the bottom, there are two buttons: 'Crear Cuenta' and 'Cancelar Cuenta'.

Anexo 4. Diagramas de Clases de Diseño con estereotipos Web

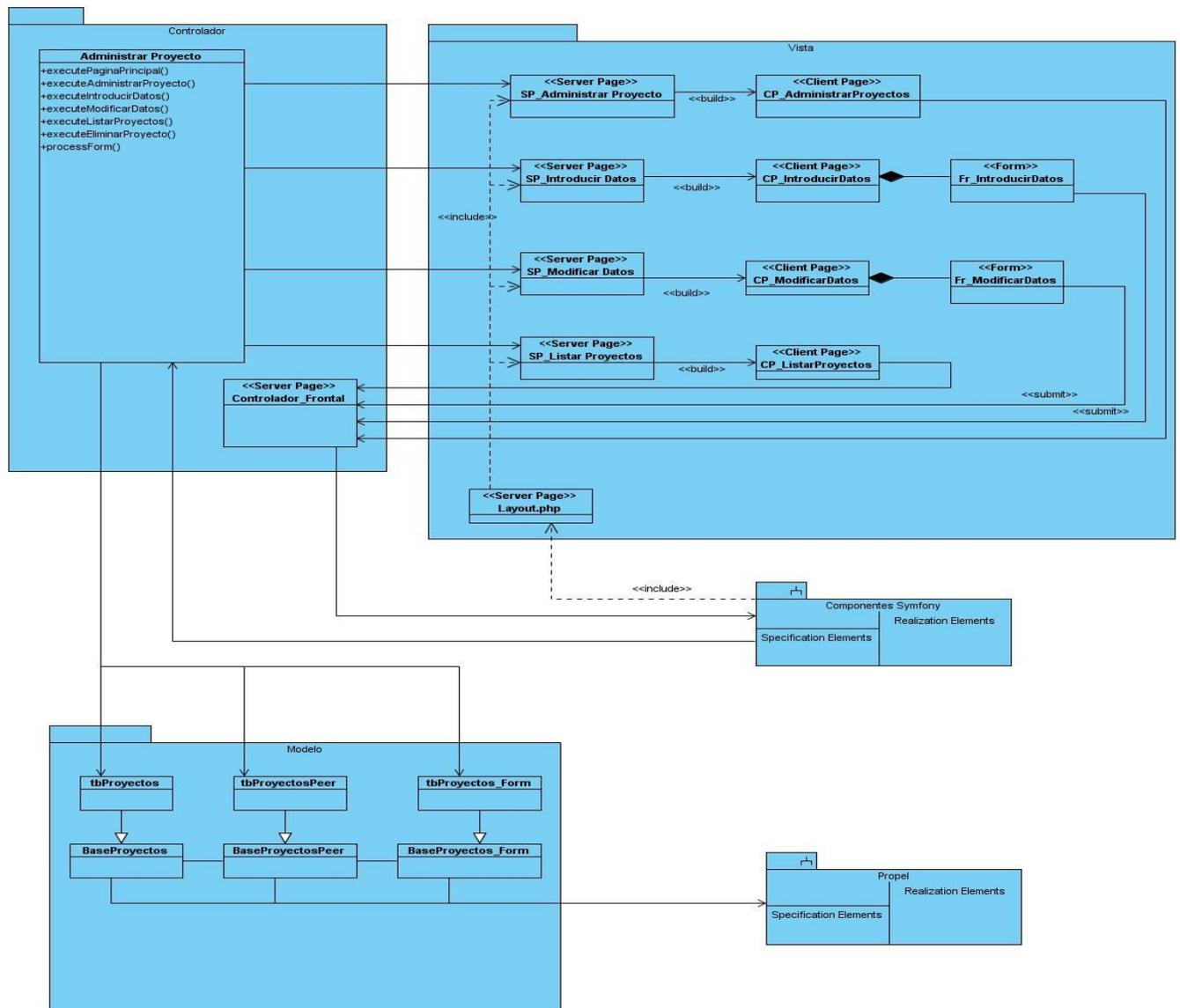
Anexo 4.1. Diagrama Clases de Diseño con Estereotipos Web de la HU\_Crear Cuenta



Anexo 4.2. Diagrama Clases de Diseño con Estereotipos Web de la HU\_Acceder al Sistema



## Anexo 4.3. Diagrama Clases de Diseño con Estereotipos Web de la HU\_Administrar Proyecto



### Glosario de Términos

**Ciclo de vida:** Proceso por el cual los analistas de sistemas, los ingenieros de software, los programadores y los usuarios finales elaboran sistemas de información y aplicaciones informáticas, definiendo el inicio y fin del proyecto.

**Validación:** es el acto documentado de probar que cualquier procedimiento, proceso, actividad o sistema conduce realmente al resultado esperado.

**Metodología Ágil:** principios y técnicas pragmáticas que pueden la entrega del proyecto menos complicada y más satisfactoria tanto para los clientes como para los equipos de entrega.

**Eficiencia:** Capacidad de disponer de alguien o de algo para conseguir un efecto determinado.

**Eficacia:** Capacidad de lograr el efecto que se desea o se espera.

**Efectividad:** Capacidad de lograr el efecto que se desea o se espera.

**Guía:** Aquello que dirige o encamina el desarrollo del software.

**Metodologías Tradicionales o Pesadas:** son aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado.

**Proceso de Software:** es aquel en 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.

**Requisitos:** son las capacidades, condiciones o cualidades que el sistema debe cumplir y tener.

**Artefactos:** es un producto tangible resultante del proceso de desarrollo de software.

**Estándares:** son los procesos que parten de los criterios de auto evaluación de un proceso y sirven para medir la excelencia del producto.

**Herramientas:** son el medio de apoyo necesario para realizar las prácticas en la Ingeniería de Software.

**Métodos:** son las formas en que se realizan las tareas de Ingeniería de Software.

**Línea base:** es el punto de referencia para examinar la evolución de lo planificado contra lo realmente ejecutado en el desarrollo del producto.

## Glosario de Términos

---

**Especificación:** explicación detallada de las características o cualidades del producto.

**Requisitos no funcionales:** verifican cómo un sistema debería de ser. Son a menudo llamados las cualidades de un sistema.

**Módulo:** es un software que agrupa un conjunto de subprogramas y estructuras de datos.

**Interna:** Puede ser aplicada a un producto de software no ejecutable (como una especificación o código fuente) durante el diseño y la codificación.

**Externa:** Usa medidas de un producto de software, derivadas del comportamiento del mismo, a través de la prueba, operación y observación del software.

**Punto de Función:** se define como una función comercial de usuario final.