

Universidad de las Ciencias Informáticas

Facultad 4



Título: “Gestión de indicadores influyentes en la estimación de tiempo y esfuerzo en los proyectos productivos de la Universidad de las Ciencias Informáticas.”

Trabajo de Diploma para optar por el título de
Ingeniero Informático.

Autores: Yadira Machado Peña.
Mairelis Quintero Rios.

Tutores: Violena Hernández Aguilar.
Isledys Hernández Bermúdez.

Ciudad de La Habana, Junio del 2008.



"Si el presente es lucha el futuro es nuestro."

Ernesto "Che" Guevara.

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 19 días del mes de junio del año 2008.

Yadira Machado Peña

Mairelis Quintero Ríos

Ing. Violena Hernández Aguilar

Ing. Isledys Hernández Bermúdez

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado “Gestión de indicadores influyentes en la estimación de tiempo y esfuerzo de los proyectos productivos de la Universidad de las Ciencias Informáticas” define indicadores necesarios para apoyarse en la tarea de estimar el tiempo y esfuerzo de desarrollo de los proyectos productivos, por tal motivo la Dirección de Informatización considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

Totalmente X

Parcialmente en un %

La Universidad de las Ciencias Informáticas (UCI) cuenta con los recursos necesarios para lograr automatizar el sistema propuesto ya que como toda empresa productora de software, poseer un registro histórico de todos los proyectos es un objetivo primario, herramienta importante para el proceso inicial de los proyectos, pues hace que la producción de nuevos productos sea más rápida, eficiente y con mayor calidad.

Actualmente la UCI no cuenta con un sistema que registre datos históricos de los proyectos y considerando que en la universidad se desarrollan disímiles proyectos de gestión, multimedia, portales, realidad virtual, e investigación contando ya con productos suficientes para desarrollar la base histórica, ya que cada proyecto acumula experiencias propias en sus estimaciones y planificaciones que sirven de apoyo a futuros proyectos que posean características similares para utilizar los indicadores como referencia.

Y para que así conste, se firma la presente en el mes de junio del año 2008.

Ing. Manuel Alejandro Gil Martín

Representante de la entidad

Jefe de Grupo de Plataforma

Dirección de Informatización

Cargo

DATOS DE CONTACTO

Autora: Mairelis Quintero Ríos.

Correo electrónico: mquintero@estudiantes.uci.cu, mairelis86@gmail.com

Autora: Yadira Machado Peña.

Correo electrónico: ymachadop@estudiantes.uci.cu, ymachadop@gmail.com

Tutora: Ing. Violena Hernández Aguilar

Teléfono: (07) 2024588 (Ciudad Habana, Playa)

Correo electrónico: violena@uci.cu

- Ingeniera Informática del Instituto Superior Politécnico José Antonio Echeverría (2005).
- Profesora de la Universidad de las Ciencias Informáticas, en la Disciplina de Ingeniería y Gestión de Software desde el año 2005.
- Cuenta con 3 años de trabajo en la Educación Superior.
- Cursa la Maestría en Gestión de Proyectos Informáticos.
- Se desempeña laboralmente como Especialista General de la Dirección de Calidad de la infraestructura Productiva de la UCI.

Tutora: Isledys Hernández Bermúdez.

Teléfono: (07) 8302665 (Ciudad Habana, Vedado)

Correo electrónico: ihernandezb@uci.cu

- Ingeniera en Ciencias Informáticas en la primera graduación de la UCI (2007).
- Profesora de la Universidad de las Ciencias Informáticas, en la Disciplina de Técnicas y Programación desde el año 2007.
- Cursa el diplomado de Calidad de Software.
- Se desempeña laboralmente como Especialista del Grupo de Apoyo a la Toma de Decisiones de la Dirección General de Producción.

AGRADECIMIENTOS GENERALES

A nuestro **Fidel** por brindarnos la posibilidad de formarnos como ingenieros informáticos.

A nuestra **Tutora Violena** por guiarnos en todo momento, por su paciencia y dedicación de velar porque nuestro Trabajo de Diploma se desarrolle lo mejor posible.

A nuestra **Tutora Isledys** por apoyarnos y darnos la calma necesaria para concluir la investigación con éxito.

A **Teodoro** por toda su ayuda brindada.

Muchas Gracias

AGRADECIMIENTOS

A la **familia tan linda que tengo** por apoyarme en todo lo que he decidido en vida.

A mi **novio** por ser tan paciente, por ayudarme y apoyarme en todo lo que estuvo a su alcance. A **su familia** por todo su cariño.

A mi **compañera de tesis** por su apoyo, por estar tan unidas en las buenas y las malas.

A **Yurima y Mechi** por apoyarme cuando más lo necesité y que de una forma u otra siempre han estado presentes en mi corazón.

A **Annelis, Alionuska y Yadira** que me han ayudado desde que las conozco, por ser las mejores amigas del mundo, nunca las voy a olvidar.

A **Roig, Alejandro, Adonis, Daymi, Isledys, Silvia** por ser muy buenos y por todo sus cariños....

A **mis amigos** que hoy no están en la universidad, saben que están presentes en todo momento.

A **todos mis amigos y amigas**, que les tengo mucho cariño. A Caridad, Giselle, Diormis, Yadira Calimano, Maydelin, William, Matos y su familia, Dariel, Yunior, Cheveto. A todos por estar presentes en momentos importantes de mi carrera.

Mairelis

A **mis padres** por darme todo el apoyo y confianza durante toda mi vida y el transcurso por la universidad.

A mi **Tía Norma** por el amor que me ha brindado y todos sus consejos.

A **mi abuelita** Cacha por todo el cariño y preocupación en todos los momentos.

A todos mis primos y tíos que siempre han estado pendientes de mí.

A mi compañera de tesis por su amistad, optimismo y constante empeño en la investigación.

A Annelis, Dayannis, Tahirí, Giselle, Mairelis por ocupar un espacio especial en mi corazón.

A todos mis amigos de siempre en la universidad por aguantarme todos estos años como Cary y Alionuska. También a Jane, Yudisleydis, Yuricel, Nilien, Delmys, Maylen y a todos los que de una forma u otra han estado presente durante estos 5 años.

Yadira

DEDICATORIA

*A mi papá por darme fuerza a pesar de no estar físicamente, por ser la luz de
mi vida, toda mi carrera se la debo a él.*

*A mi mamá por ser la mejor madre del mundo, porque fue madre y padre a la
vez y me enseñó a ser como soy y a luchar por lo que quiero sin miedo a nada.*

*A mi hermana que fue mi inspiración por darle un buen ejemplo a seguir en
su vida.*

*A mi padrastro por ser fuerte e intransigente para enseñarme que la vida es
de sacrificios.*

A mi amor por darme la oportunidad de ver la felicidad que tenía al frente.

Mairelis

A mi Papá y a mi Mamá

A mi Tía Norma

Yadira

RESUMEN

La Gestión de Proyectos es una tarea vital en el desarrollo del software, es la disciplina que organiza y administra los recursos logrando culminar el proyecto dentro del alcance y tiempo definido. La planificación de proyectos dentro de las actividades de gestión es esencial pues define las actividades a realizar y los recursos necesarios estimándose el esfuerzo y el tiempo necesario para el desarrollo.

Varios proyectos de la Universidad de las Ciencias Informáticas (UCI) han presentado problemas en su desarrollo debido a su incorrecta estimación y planificación de esfuerzo y tiempo, reflejándose en cronogramas incoherentes con la realidad e incumplimiento con plazos y fechas de entrega, una de las causas de estos problemas es la falta de experiencia acumulada en un registro histórico de proyectos anteriores mediante el cual los jefes de proyectos puedan apoyar sus estimaciones y planificaciones.

La UCI debe de poseer un registro histórico conteniendo resultados de los proyectos desarrollados, que proporcione indicadores para ayudar a estimar el tiempo y esfuerzo de desarrollo de los proyectos productivos.

Esta investigación persigue como objetivo proponer un conjunto de indicadores influyentes en las estimaciones de tiempo y esfuerzo en los proyectos y desarrollar el análisis de una propuesta de sistema que permita recoger datos históricos de los proyectos productivos de la UCI basada en los indicadores propuestos.

PALABRAS CLAVES

Gestión de Proyectos, Planificación, Estimación, Tiempo, Esfuerzo, Indicadores, Registro Histórico.

TABLA DE CONTENIDOS

AGRADECIMIENTOS GENERALES IV

AGRADECIMIENTOS..... V

DEDICATORIA VI

RESUMEN..... VIII

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA5

1.1 Reseña de la Gestión de Proyectos..... 6

 1.1.1 ¿Qué es la Gestión de Proyectos?7

 1.1.2 La Guía PMBOK para la Gestión de Proyectos.....8

 1.1.3 La Norma ISO para la Calidad en la Gestión de Proyectos.9

1.2 Planificación de un Proyecto de Software..... 10

 1.2.1 Objetivos de la Planificación de un Proyecto de Desarrollo de Software. 11

 1.2.2 Actividades asociadas a la planificación. 11

1.3 Estimación de un Proyecto de Software..... 12

 1.3.1 Principios básicos para estimar eficientemente..... 13

 1.3.2 Esfuerzo de desarrollo del proyecto. 15

 1.3.3 Tamaño del producto. 15

 1.3.4 Costo del Software. 16

 1.3.5 Riesgos en la estimación y planificación. 16

 1.3.6 Principales Métodos de estimación. 17

 1.3.7 Herramientas para Estimación y Planificación. 20

1.4 Estudio de Indicadores..... 21

 1.4.1 Mediciones y Métricas para estimar los proyectos de software..... 23

 1.4.2 Indicadores en la Gestión de Proyectos. 25

 1.4.3 Diferencia entre medidas, estadísticas e indicadores. 27

1.5 Datos Históricos en la Gestión de Proyectos. 28

 1.5.1 Fundamentos de las Herramientas Utilizadas. 29

1.6 Conclusiones parciales..... 30

CAPÍTULO 2: PROPUESTA DE INDICADORES	31
2.1 Panorámica de la situación de la producción en la UCI	31
2.2 Propuesta de Indicadores	33
2.3 Fundamentos de los Indicadores Propuestos	35
2.3.1 Indicador Tipo de Proyecto	35
2.3.2 Indicador Entorno de Proyecto	35
2.3.3 Indicadores de Proceso	36
2.3.3.1 Indicador Gestión de la Calidad.....	37
2.3.3.2 Indicador Pruebas de Software.....	38
2.3.3.3 Indicador Despliegue de la Aplicación.....	39
2.3.4 Indicadores de Estimación	41
2.3.4.1 Indicador Tiempo de desarrollo.....	42
2.3.4.2 Indicador Esfuerzo de desarrollo.....	43
2.3.4.3 Indicador Tamaño del Producto.....	44
2.3.4.4 Indicador Influencia de los Riesgos en la estimación.....	45
2.3.4.5 Indicador Cantidad de Versiones.....	46
2.3.5 Indicadores de Recursos	47
2.3.5.1 Indicador Metodología de Desarrollo de Software.....	47
2.3.5.2 Indicador Estilo Arquitectónico.....	49
2.3.5.3 Indicador Patrón de Arquitectura.....	50
2.3.5.4 Indicador Lenguaje de Programación.....	51
2.3.5.5 Indicador Componentes reutilizables utilizados.....	52
2.3.5.6 Indicador Herramienta de Modelado Visual.....	54
2.3.5.7 Indicador Herramienta de Programación.....	54
2.3.5.8 Indicador Herramienta de Sistema Gestor de Base de Datos.....	54
2.3.5.9 Indicador Herramienta para la Planificación.....	55
2.3.5.10 Indicador Herramienta para la Gestión de Configuración.....	55
2.3.5.11 Indicador Herramientas utilizadas en la estimación.....	55
2.3.5.12 Indicador Cantidad de PC.....	57
2.3.5.13 Indicador Cantidad de Integrantes del equipo de desarrollo.....	57
2.3.5.14 Indicador Nivel de conocimiento de los integrantes del equipo.....	57
2.4 Propuesta de Análisis Estadístico de los resultados	58
2.4.1 Ejemplo de cómo utilizar el Principio de Pareto.....	61
2.5 El Método Delphi en la validación de la propuesta de indicadores	67
2.5.1 Desarrollo práctico y explotación de resultados.....	72

CAPÍTULO 3: PROPUESTA DEL SISTEMA	79
3.1 Necesidad que justifica el desarrollo del sistema	79
3.2 Modelo de Dominio	79
3.2.1 Justificación del uso del modelo de dominio.	82
3.3.1 Especificación de Requisitos de Software.....	83
3.3.1.1 Requisitos Funcionales	83
3.3.1.2 Requisitos No Funcionales	86
3.4 Modelo de Casos de uso del sistema	87
3.4.1 Definición de los actores del sistema.....	88
3.4.2 Descripción Textual de los Casos de uso.....	89
3.5 Modelo de Análisis de la solución propuesta	94
3.5.1 Paquetes de Análisis.....	94
3.5.2 Diagrama de clases de Análisis.	95
3.6 Prototipos de Interfaz de usuario	98
3.7 Conclusiones Parciales	103
CONCLUSIONES	104
RECOMENDACIONES	105
REFERENCIAS BIBLIOGRAFICAS	106
BIBLIOGRAFÍA	107
ANEXOS	108
Anexo 1 Listado de Expertos que colaboraron.	108
Anexo 2 Cálculo del coeficiente de conocimiento de los expertos.	109
Anexo 3 Resultado de los puntos de corte para los indicadores.....	111
Anexo 4 Descripciones Textuales de los Casos de uso.....	114
GLOSARIO DE TÉRMINOS	151

INTRODUCCIÓN

Actualmente a pesar de los avances alcanzados en la tecnología, el desarrollo de software continúa siendo un gran desafío para quienes se dedican a esta actividad. Uno de los principales obstáculos es lograr una buena estimación y planificación del proyecto de software. La gestión de un proyecto es una tarea vital para obtener éxito en el desarrollo, en ella se definen dos actividades fundamentales: planificación y estimación.

El objetivo principal de la estimación de proyectos de software es proporcionar un marco de trabajo que permita hacer estimaciones razonables de recursos, costos y duración. La estimación de plazos y esfuerzo en los proyectos es una tarea crítica, que puede conducir al fracaso del proyecto debido a la extensión de una actividad o a una errónea estimación del esfuerzo necesario para ejecutarla.

La planificación permite obtener una visión del esfuerzo y el tiempo necesario para concluir una actividad dentro del proyecto, organiza un calendario de cumplimiento de actividades permitiendo conocer el estado de avance del proyecto en todo momento.

Los resultados de las estimaciones y planificaciones de los proyectos de software están dados por las características y resultados propios del proceso de desarrollo, estos resultados sirven de indicadores y pueden recogerse en un registro histórico para apoyar estimaciones y planificaciones que se acerquen más a las reales.

Los indicadores son necesarios para mejorar el desarrollo de proyectos de software contando con la experiencia y los resultados de proyectos similares mejorando así la calidad de los productos.

La Universidad de las Ciencias Informáticas (UCI) es un centro de producción de software y una universidad innovadora de excelencia científica, académica y productiva que forma profesionales integrales comprometidos con la patria, incrementando la competitividad internacional de la industria cubana del software, para lograr cumplir con su propósito es de vital importancia la estimación y planificación correcta de los proyectos tanto de exportación como los nacionales y cumplir con las expectativas del cliente entregando un producto a tiempo que satisfaga todas sus necesidades. La UCI se encuentra desarrollando disímiles proyectos de gestión, multimedia, portales y realidad virtual. Cada proyecto acumula experiencias propias en sus estimaciones y planificaciones que sirven de apoyo a futuros proyectos que posean características similares para utilizar esa experiencia como referencia.

En la UCI el proceso de estimación y planificación del software posee un desarrollo incipiente, donde los diferentes proyectos trabajan métodos y técnicas de forma descentralizada y los resultados no son óptimos. Debido a estos problemas, muchos proyectos no se han ajustado al

tiempo de entrega y existe mucha desorganización en la planificación de las actividades, por otra parte los datos que arrojan finalmente los proyectos no se recogen lo cual no permite una mayor precisión en futuras estimaciones.

La UCI no registra datos históricos de proyectos ya finalizados, que permitan a los jefes de proyectos realizar un análisis estadístico y obtener una estimación más realista y certera adaptada a sus proyectos. La UCI utiliza para estimar principalmente los proyectos de exportación de la empresa comercializadora ALBET. SA, un método creado por un grupo de especialistas de la Infraestructura Productiva (IP) con cierto grado de experiencia en la gestión y dirección de proyectos, el cual se elaboró sobre una base empírica e intuitiva y no tiene en cuenta riesgos que influyen a la hora de estimar el esfuerzo y el tiempo en los proyectos de software, esta técnica no arroja resultados realistas en los proyectos que lo han utilizado, debido a que la UCI lleva pocos años de creada y no cuenta con un registro de las experiencias acumuladas de otros proyectos similares donde dicho método pueda apoyar sus resultados.

Las imprecisiones en la estimación del tamaño del software y del tiempo, cuando es llevada a cabo en fases tempranas del desarrollo, sin contar con registro histórico basado en indicadores que permitan guiarse para estimar mejor el producto, ocasionan un impacto desfavorable en la buena gestión de los proyectos de software y a su vez dañan el prestigio de la UCI en el mercado internacional. La mayoría de los proyectos usualmente toman más tiempo, cuestan mucho más o no proveen lo que el usuario desea o necesita.

Todo proyecto de desarrollo de software consume tiempo y esfuerzo en cada una de las etapas que lo conforman y se necesita establecer cuales son los factores que influyen directa e indirectamente en ellas. Los jefes de proyectos necesitan conocer cual es el nivel de productividad que tiene su grupo de trabajo, medir y comparar sus proyectos con los demás que se desarrollen a la par o que ya han cerrado, pero desafortunadamente no cuentan con indicadores que recojan experiencia, evitando errores ya cometidos y logrando entregar productos de mayor calidad en el tiempo estimado.

La toma de decisiones en la administración de proyectos de software no está sustentada por un registro histórico de indicadores que permitan mejores estimaciones generando así información de mejor calidad y fiabilidad. Es necesario disponer de indicadores influyentes sobre las desviaciones del proyecto según su planificación y que proporcione estimaciones más ajustadas.

Debido a la situación expuesta, se identificó el siguiente **problema**: No se cuenta con un registro histórico de indicadores influyentes en la estimación de tiempo y esfuerzo en los proyectos productivos que permitan estimaciones más certeras.

Partiendo del problema planteado, el **objeto de estudio** se enfocará hacia el proceso de desarrollo de software. El **campo de acción** será el proceso de estimación y planificación de los proyectos productivos de la Universidad de las Ciencias Informáticas.

Para resolver el problema que se planteó con anterioridad, se trazó como **objetivo general**: Proponer un registro histórico de indicadores influyentes en la estimación de tiempo y esfuerzo de los proyectos productivos de la Universidad de las Ciencias Informáticas.

Para ello se trazaron los siguientes **Objetivos Específicos** de la investigación:

- Proponer indicadores influyentes en la estimación de tiempo y esfuerzo de los proyectos productivos de la UCI definiendo un primer análisis estadístico.
- Desarrollar un análisis de la propuesta de sistema para recoger datos históricos de los proyectos productivos de la UCI basada en los indicadores propuestos.

Para lograr los objetivos trazados, se desarrollaron las siguientes **Tareas de la Investigación**:

- Investigar los principales problemas del proceso de desarrollo de software en el mundo y en la UCI, analizando el proceso de estimación y planificación de recursos en la gestión de proyectos.
- Realizar entrevistas al personal involucrado en la actividad de estimación y planificación en la UCI.
- Estudiar las tendencias de la utilización de indicadores en la producción de software.
- Proponer indicadores influyentes en las estimaciones de tiempo y esfuerzo.
- Validar la propuesta de los indicadores mediante un grupo de expertos en el tema.
- Proponer un análisis estadístico para obtener resultados de los indicadores.
- Desarrollar una propuesta de sistema como registro histórico de indicadores basado en los propuestos.

Métodos Científicos utilizados en la investigación.

Métodos teóricos

Para la realización de la presente investigación se utilizaron los métodos teóricos para estudiar las características del objeto de investigación, entre ellos se utilizaron los métodos de análisis e histórico, los cuales ayudaron a estudiar la trayectoria real de los fenómenos y acontecimientos en su historia. El método lógico permitió investigar las leyes generales del funcionamiento y desarrollo de los fenómenos.

Métodos empíricos

La investigación se apoyó en los métodos empíricos para extraer informaciones de los fenómenos analizados, entre ellos el método de encuesta y de entrevista. Además se utilizó el método de medición, matemático o estadístico para realizar un procedimiento de obtención de información numérica acerca de una propiedad o cualidad.

Estructura del Informe.

El presente informe del trabajo de diploma queda estructurado en 3 capítulos, en el Capítulo 1 se plasman los fundamentos teóricos de la investigación, en el Capítulo 2 se proponen una serie de indicadores influyentes en la estimación y planificación de los proyectos de software de la UCI y se realiza un análisis de su factibilidad mediante un grupo de expertos en el tema, así como una propuesta inicial de un posible estudio estadístico sobre los indicadores. El Capítulo 3 se dedica al análisis de una propuesta de sistema como registro histórico que permita gestionar los indicadores en los proyectos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El avance de la tecnología que existe en la actualidad ha aumentado el desarrollo de proyectos de software, así como la complejidad y el tamaño de estos sistemas. Las aplicaciones de software cada día están más presentes en todas las actividades, las cuales van desde un sistema financiero de un banco hasta el sistema de control de líneas aéreas, software de ayuda a la construcción, sistemas de apoyo al trabajo cooperativo y software educativos.

El incremento en la complejidad de los productos de software se debe a que los usuarios son cada vez más exigentes, en términos de interacción, seguridad y retroalimentación, aumentan los pedidos de sistemas compuestos de múltiples subsistemas donde intervengan bases de datos, interfaces gráficas, comunicación entre componentes, recuperación ante fallas e interacción con dispositivos multimedia.

Si bien este crecimiento es un contexto prometedor para el desarrollo y fortalecimiento de una industria de software, los problemas relacionados con el desarrollo y mantenimiento de estos sistemas son muchos y más grandes. Desafortunadamente, no es difícil encontrar usuarios y clientes inconformes con el software que utilizan debido a que este no satisface sus necesidades, es ineficiente, no es amigable ni seguro.

El desarrollo de software es una de las principales tareas del Gobierno Cubano, por lo que se han desarrollado estrategias con el fin de elevar la producción y calidad del software. Una solución a estos problemas se enfoca hacia la producción de software organizada y estructurada. Por otra parte las empresas dependen de un mercado cada vez más exigente en tiempo, costo y calidad.

Una innovadora solución es la vinculación Universidad-Empresa, esta es una alianza estratégica de intercambio donde la universidad obtiene la facilidad de aplicar sus investigaciones y de vincular sus estudiantes y profesores al mundo empresarial funcionando como una entidad y la empresa recibe el conocimiento y la innovación constante que generan las universidades. Existen diferentes variantes como son los contratos entre universidades y los parques científicos y tecnológicos los cuales son espacios donde se materializan estas relaciones.

La Universidad de las Ciencias Informáticas (UCI) es la primera universidad surgida al calor de la Batalla de Ideas, la misma pretende ser la vanguardia del desarrollo de las empresas de software en Cuba, regir y propiciar un avance tecnológico de la industria del software en Cuba y convertirse en un renglón fundamental de la economía cubana para insertarse en el mercado internacional.

En la UCI existe un alto número de proyectos informáticos que ofrecen soluciones de software. Una de las primeras actividades a desarrollar es la planificación y estimación del producto como actividades esenciales de la gestión de proyectos.

1.1 Reseña de la Gestión de Proyectos.

La gestión de proyectos es una disciplina reciente que comenzó a forjarse en los años 70. La necesidad de su utilización surgió en el ámbito militar. En los años 50, el desarrollo de grandes proyectos militares requería la coordinación del trabajo conjunto de equipos y disciplinas diferentes en la construcción de sistemas únicos. Bernard Schriever, arquitecto de desarrollo de misiles balísticos es considerado el padre de la gestión de proyectos, el cual consiguió reducir considerablemente los tiempos de ejecución de varios proyectos donde fue posible relacionar los nuevos proyectos con otros, viéndose en su desarrollo la necesidad de descomponer los proyectos en partes de menor dimensión para realizar planificaciones lo que constituyó una guía para el desarrollo de nuevos sistemas, ya que brinda la visión de cómo relacionar proyectos realizados con proyectos en ejecución corrigiendo errores cometidos por diferentes motivos y lograr con esto una mayor probabilidad de éxito en el producto final.

El desarrollo de sistemas complejos, que requerían el trabajo conjunto y sincronizado de varias disciplinas o ingenierías, hizo evidente en los años 60 la necesidad de desarrollar métodos de organización y de trabajo para evitar los problemas que se repetían con frecuencia en los proyectos tales como:[1]

- Desbordamiento de agendas.
- Desbordamiento de costes.
- Calidad o utilidad del resultado obtenido.
- Requerimientos incorrectos, incompletos o ambos.
- Muchas especificaciones de requerimientos son inestables y sujetas a cambios.
- La planificación de costos y plazos no es actualizada y se basa en las necesidades del mercado y no en las necesidades de los requerimientos del sistema.
- Es difícil estimar el tamaño y complejidad del proyecto para realizar una estimación de costos y plazos realista.
- Los costos y plazos no son estimados cuando los requerimientos del sistema o el ambiente de desarrollo cambian.
- No se manejan factores de riesgo.
- La mayoría de las organizaciones no recolectan datos de proyectos pasados.

Todos los problemas mencionados influyen en el fracaso de un proyecto por lo que muchos son abandonados antes del tiempo debido. Cancelar un proyecto implica perder una enorme cantidad de dinero y a su vez el prestigio de cada uno de los desarrolladores.

La **Figura 1.1** Muestra un estudio en 2005 con una muestra de 8000 proyectos.

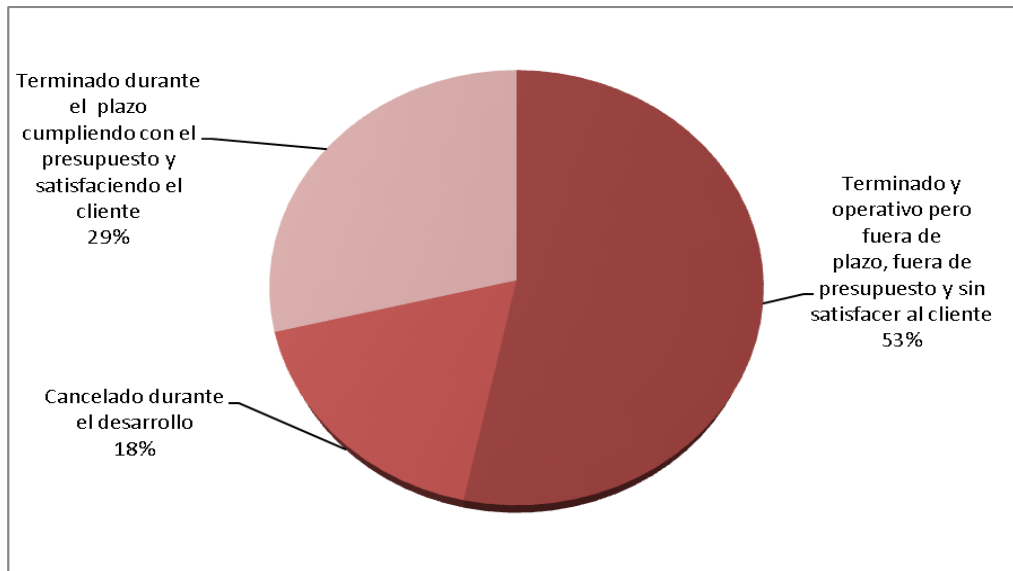


Figura 1.1. Estudio del grupo Standish en 2005.

Para dar respuesta a esta necesidad, a partir de los años 60 se ha desarrollado un conjunto de conocimientos y las prácticas necesarias para gestionar una metodología que controle el desarrollo de software, con las mejores garantías de calidad de los resultados. Ese cuerpo de conocimientos se ha desarrollado y configurado como una nueva disciplina que garantiza el éxito de los proyectos: la gestión de proyectos.

1.1.1 ¿Qué es la Gestión de Proyectos?

La Gestión de Proyectos es el proceso por el cual se planifica, dirige y controla el desarrollo de un sistema con un costo mínimo y dentro de un período de tiempo específico. A continuación se muestran definiciones de diferentes autores sobre la gestión de proyectos.

“Conjunto de actividades, planificadas, ejecutadas y supervisadas que, con recursos finitos, tiene por objeto crear un servicio o producto único” [2]

“Combinación de todos los recursos necesarios, reunidos en una organización temporal, para la transformación de una idea en realidad” [3]

“El proceso de Gestión de Proyectos describe los métodos y prácticas que deben tenerse en consideración desde que se inicia un proyecto hasta su finalización. La aplicación de estas prácticas permitirá llevar una buena gestión del proyecto y mantener un mayor control, permitiendo a los líderes y a su equipo realizar proyectos de manera eficaz y eficiente (en alcance, tiempo, coste), así como asegurar la calidad y transparencia a lo largo de toda la vida del proyecto.”[4]

La Gestión de Proyectos es la planificación, organización, seguimiento y control de todos los aspectos de un proyecto, así como la motivación de todos aquellos implicados en el mismo, para alcanzar los objetivos del proyecto de una forma segura, cumpliendo las especificaciones definidas

de plazo, coste y rendimiento. Además incluye el conjunto de tareas de liderazgo, organización y dirección técnica del proyecto, necesarias para su correcto desarrollo.

La Gestión de Proyectos es el flujo de ingeniería primordial para el desarrollo de un proyecto ya que dirige, mantiene y controla todo el proceso del software. Como bien se plantea en los conceptos anteriores la Gestión de Proyectos tiene como finalidad principal la planificación, el seguimiento y control de las actividades de los recursos humanos y materiales que intervienen en el desarrollo de un proyecto de software.

1.1.2 La Guía PMBOK para la Gestión de Proyectos.

La guía de los Fundamentos de la Dirección de Proyectos PMBOK (Project Management Body Of Knowledge) es el estándar de gestión de proyectos desarrollado por el Instituto de Administración de Proyectos, PMI (Project Management Institute). El PMI es una organización que atiende a las necesidades relacionada con la gestión de proyectos de los profesionales de cualquier disciplina.

PMBOK describe las mejores prácticas que son aceptadas en la mayoría de los proyectos

Las áreas de conocimientos que describen el PMBOOK son:[5]

- **Gestión de la Integración del Proyecto:** Describe los procesos y actividades que forman parte de los diversos elementos de la dirección de proyectos, que se identifican, definen, combinan, unen y coordinan dentro de los Grupos de Procesos de Dirección de Proyectos.
- **Gestión del Alcance del Proyecto:** Describe los procesos necesarios para asegurarse de que el proyecto incluya todo el trabajo requerido para completarlo satisfactoriamente.
- **Gestión del Tiempo del Proyecto:** Describe los procesos relativos a la puntualidad en la conclusión del proyecto.
- **Gestión de los Costes del Proyecto:** Describe los procesos involucrados en la planificación, estimación, presupuesto y control de costes de forma que el proyecto se complete dentro del presupuesto aprobado.
- **Gestión de la Calidad del Proyecto:** Describe los procesos necesarios para asegurarse de que el proyecto cumpla con los objetivos por los cuales ha sido emprendido.
- **Gestión de los Recursos Humanos del Proyecto:** Describe los procesos que organizan y dirigen el equipo del proyecto.
- **Gestión de las Comunicaciones del Proyecto:** Describe los procesos relacionados con la generación, recogida, distribución, almacenamiento y destino final de la información del proyecto en tiempo y forma.
- **Gestión de los Riesgos del Proyecto:** Describe los procesos relacionados con el desarrollo de la gestión de riesgos de un proyecto.

- **Gestión de las Adquisiciones del Proyecto:** Describe los procesos para comprar o adquirir productos, servicios o resultados, así como para contratar procesos de dirección.

El PMBOK sugiere variadas técnicas incluyendo las de estimación, así como técnicas para la gestión de riesgos. La calidad queda garantizada con el uso de variadas técnicas para la planificación y control de calidad.

1.1.3 La Norma ISO para la Calidad en la Gestión de Proyectos.

La calidad en la gestión de proyecto implica por una parte la calidad del desarrollo de software y por otra la calidad del producto. Ambos son imprescindibles y requieren un tratamiento sistemático. Debe asegurarse la satisfacción del cliente dentro de los márgenes que proporcionan las reglas y objetivos de la empresa y del propio equipo de proyecto. Es por eso que surge la norma 10006 para brindar una serie de pasos logrando con calidad la gestión de proyectos.

La familia de normas ISO 9000 son normas de calidad establecidas por la Organización Internacional para la Estandarización (ISO) que se pueden aplicar en cualquier tipo de organización. Se componen de estándares y guías relacionados con sistemas de gestión y de herramientas específicas como los métodos de auditoría (el proceso de verificar que los sistemas de gestión cumplen con el estándar).

La norma ISO 10006 se refiere a las directrices para la calidad en la gestión de proyectos. Fue preparada por el comité técnico de Gestión de calidad y aseguramiento de calidad, la norma ISO 10006 lleva como nombre "Directrices para la gestión de la calidad en los proyectos"

Esta norma tiene como objetivo servir de guía en aspectos relativos a elementos, conceptos y prácticas de sistemas de calidad que pueden implementarse en la gestión de proyectos o que pueden mejorar la calidad de la gestión de proyectos.

Se basa en los procesos claves para gestionar proyectos, algunos de estos procesos se mencionan a continuación:[5]

Proceso estratégico: Sirve para planificar el establecimiento, la implementación y el mantenimiento de un sistema que llevaría a cabo la organización del proyecto.

Proceso relacionado con los recursos: Se realizará la planificación y control de recursos realizando las revisiones adecuadas para asegurarse de que se dispone de recursos suficientes para cumplir los objetivos del proyecto. Para esto se debería identificar y analizar las desviaciones respecto al plan de recursos.

Proceso relativo al personal: Se definirá la estructura organizativa con su asignación de recursos y responsabilidades, brindándole la oportunidad de capacitarse del tema y adquirir experiencia.

Proceso relacionado con el alcance: Se debe desarrollar en el sentido de definir las líneas maestras del producto final, identificando las actividades y pasos a realizar pero con un control que garantice la consecución de los objetivos.

Proceso relativo al tiempo: En estos procesos se debe de conseguir una planificación de dependencia de actividades con la estimación y calendario adecuado. Para asegurar el control adecuado de las actividades y los procesos de los proyectos, deberían establecerse los tiempos de revisión del programa y la frecuencia de recopilación de datos.

Proceso relacionado con el riesgo: Se identificará haciendo una evaluación del sistema de la probabilidad de ocurrencia, así como del impacto en el proyecto para poder desarrollar e implementar planes de respuestas. También es importante tener una base histórica que te brinde una noción del riesgo.

Proceso relativo a la mejora: Tanto la organización originaria como la organización encargada del proyecto deberían aprender de los proyectos. Deben medir todos los datos derivados a los proyectos de su organización.

1.2 Planificación de un Proyecto de Software.

La planificación involucra la especificación de objetivos y metas para un proyecto y las estrategias, políticas, planes y procedimientos para alcanzarlos. La planificación se dirige a las metas del proyecto, riesgos potenciales y problemas que puedan interferir con el cumplimiento de estas metas.

En un proyecto de software la planificación firme depende de la planificación detallada de sus avances, mitigando problemas que pueden surgir y preparando con anterioridad soluciones tentativas. El líder del proyecto conjuntamente con el planificador son los responsables de la planificación desde el levantamiento de requisitos hasta la entrega del sistema terminado, realizan varias iteraciones de planificación como un proceso de descubrimiento de factores necesarios, dando lugar a la realización de estimaciones más exactas a medida que avance el proyecto.

Existen dos grandes etapas en las que la planificación cobra el máximo protagonismo. La primera es necesaria para estudiar y establecer la viabilidad de un proyecto, ya sea interno o externo a la organización. Se deberán realizar los correspondientes estudios técnicos, de mercado, financieros, de rentabilidad, así como una estimación de los recursos necesarios y los costes generados. Todo ello constituye el elemento fundamental en el que se apoya el cliente para decidir sobre la realización o no del proyecto.

La segunda etapa importante de planificación tiene lugar una vez que se ha decidido ejecutar el proyecto. Se realiza una planificación detallada punto por punto. Uno de los errores más importantes y graves en la gestión de proyectos es querer iniciarlo sin haber prestado la atención

debida a una serie de tareas previas de preparación, organización y planificación que son imprescindibles para garantizar la calidad de la gestión y el éxito posterior.

Es importante remarcar la importancia de la planificación del proyecto en función de los objetivos definidos, no es razonable planificar un proyecto y pensar que esa planificación es ya definitiva e inmutable. En la mayoría de los casos, la realidad no coincide exactamente con lo previsto, por lo que es necesario hacer ajustes periódicos. La planificación es una herramienta para la gestión y la toma de decisiones, no para imaginar en un primer momento una evolución que posteriormente el tiempo se encargará de demostrar que estaba equivocada.

1.2.1 Objetivos de la Planificación de un Proyecto de Desarrollo de Software.

La planificación del proyecto de software tiene como objetivo proporcionar un marco de trabajo que permita al planificador hacer estimaciones razonables de recursos, costos y planificación temporal. Las planificaciones se ejecutan dentro de un marco de rango de tiempo al comienzo de un proyecto de software, y deberían actualizarse regularmente a medida que progresa el proyecto. Además las planificaciones deberían definir los escenarios del mejor y peor caso, de modo que los resultados del proyecto puedan delimitarse.

Para que la planificación sea un proceso de descubrimiento de la información llegando a una estimación razonable es necesario haber realizado un buen estudio del entorno del proyecto, de los recursos disponibles, recursos humanos, los recursos que se puedan reutilizar y los diferentes aspectos a tener en cuenta para una buena planificación.

1.2.2 Actividades asociadas a la planificación.

Ámbito del Software.

El ámbito del software es la primera actividad llevada a cabo durante la planificación del proyecto de Software.[6]

Al inicio del desarrollo del software se deben evaluar la función y el rendimiento que le asignaron para establecer un ámbito del proyecto actual, y comprensible para directivos y técnicos.

El ámbito del software describe la función, el rendimiento, las restricciones, las interfaces y la fiabilidad evaluándose las funciones del ámbito y en algunos casos se refinan para dar más detalles antes del comienzo de la estimación. Las restricciones de rendimiento acoplan los requisitos de tiempo de respuesta y funcionamiento, identificando los límites del software originados por el hardware externo, por la memoria disponible y por otros sistemas existentes.

Recursos

La planificación del desarrollo de Software tiene como segunda tarea la estimación de los recursos requeridos para estimar el esfuerzo de desarrollo de Software, donde las Herramientas (hardware y Software), son la base proporcional de soporte al esfuerzo de desarrollo, en segundo nivel se encuentran los componentes reutilizables como un recurso importante para acortar el tiempo de desarrollo y por último se encuentra el recurso primario: las personas (el recurso humano).

Recursos Humanos.

Solo se puede asegurar cuántos equipos de trabajo se necesita en el desarrollo de software luego de estimar el esfuerzo de desarrollo (cantidad de equipos por módulos) y seleccionar su posición en el equipo de trabajo y la especialidad que desempeñara cada persona.

Recursos o componentes de software reutilizables.

Se gana muchísimo tiempo utilizando lo que ya está implementado y revisado, convertido ya en un release (versión inicial o actualizada de un software) es por ello que cualquier estudio sobre recursos de software estaría incompleto sin estudiar la reutilización.

Recursos de Entorno.

El entorno es donde se adapta el proyecto incorporando Hardware y Software. Un planificador de proyectos debe determinar la ventana temporal requerida para el Hardware y el Software verificando que estos recursos estén disponibles y proporcionando una plataforma con las herramientas (Software) requeridas para producir los productos que son el resultado de la buena práctica de la Ingeniería del Software. Cada elemento de hardware debe ser especificado por el planificador del proyecto de Software.

1.3 Estimación de un Proyecto de Software.

La estimación juega un papel fundamental a la hora de evaluar el éxito o fracaso de un proyecto. Esta afirmación surge a raíz de los resultados de las relaciones entre el tiempo, esfuerzo y calidad planificadas desde el surgimiento de los proyectos de software.

La estimación no es más que “la predicción del personal, del esfuerzo, de los costos y del tiempo que se requerirán para realizar todas las actividades y construir todos los productos asociados con el proyecto”[6]; “su objetivo es conocer en etapas tempranas y de manera aproximada, el costo, la duración y los recursos necesarios para el desarrollo de proyectos de software”. [7]

La estimación no es más que una aproximación de la duración de un proyecto y del tamaño del software. Las estimaciones están asociadas con el esfuerzo y el tiempo de las actividades identificadas del proyecto. Se trata entonces de evaluar el número de períodos de trabajo

estimados necesarios para completar cada actividad. Los jefes de proyectos deben tener las respuestas a las siguientes preguntas: ¿Cuánto esfuerzo (personal necesario) se requiere para completar una actividad? ¿Cuánto tiempo se necesita para completar una actividad? ¿Cuál es el costo total de una actividad?

La estimación es importante para determinar la viabilidad del proyecto ya que no tendría sentido iniciarlo sin disponer de tiempo y presupuesto para completarlo o si este no alcanza la funcionalidad y calidad requeridas. Para que la estimación se ajuste a estos objetivos debe ser suficientemente temprana y concisa. El software representa hoy, la mayor parte del costo de los sistemas, por lo tanto una estimación acertada y temprana de su costo es esencial.

La estimación es un proceso continuo y no termina hasta el final del desarrollo y comienza utilizando pocas variables en un nivel alto de abstracción. A esta primera etapa se le denomina macroestimación y permite obtener valores cercanos de costo, tiempo y esfuerzo para estudiar la viabilidad del proyecto. Luego del comienzo del proyecto cuando se han recogido los requisitos que cumplirá el producto y obtenidos estos valores se hace una estimación un poco más precisa y se pueden efectuar las primeras comparaciones detectando desvíos en el plan y realizando los primeros ajustes correspondientes. A medida que el proyecto se desarrolle, cuando ya se tiene la línea base de la arquitectura y aumenta la información del mismo, los valores de los parámetros descriptivos de las etapas al inicio se convierten en otros más detallados (como por ejemplo cantidad de módulos y cantidad de personas a capacitar), en ese momento se hace una nueva estimación más real. El aporte de un mayor caudal de información, proveniente del diseño, programación e implementación disminuye el margen de error, hasta hacerse mínimo en la fase de aceptación del software.

En líneas generales, estimar consiste en determinar el valor de una variable desconocida a partir de otras conocidas, o de una pequeña cantidad de valores conocidos de esa misma variable. Es “el proceso que proporciona un valor, a un conjunto de variables para la realización de un trabajo, dentro de un rango aceptable de tolerancia”. [6]

1.3.1 Principios básicos para estimar eficientemente.

En el momento en que se solicita a los desarrolladores que estimen tiempo y esfuerzo, estos tienen muy poco dominio del proyecto lo que puede provocar que las estimaciones pesimistas sean mayores que las optimistas. Esta incertidumbre explica la llamada la Crisis del Software, caracterizada por sobrepasar sistemáticamente los presupuestos y tiempos planificados. Así muchos proyectos están destinados al fracaso desde su nacimiento debido a estimaciones inadecuadas durante su concepción donde generalmente se realizan trabajos apresurados.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

La mayoría de los modelos del proceso de desarrollo en Ingeniería de Software parten de la idea poco realista de que existe una especificación de requisitos y que esta cambia. Esta es una debilidad vital ya que el cambio de los requisitos durante el ciclo de desarrollo es inevitable. Incluso los requisitos como artefactos está siendo crecientemente cuestionado, cada vez más es considerado que no constituye una base adecuada para estimar tiempo y esfuerzo.

La estimación del software es un tema abordado de forma general por todos los autores dedicados a la ingeniería del software .En este sentido, suelen coincidir que a la hora de realizar una estimación, hay que tener en consideración los siguientes principios básicos:[8]

1. Aplicar la cantidad de recursos correcta para crear y refinar las estimaciones. Para determinar el nivel de detalle hay que considerar:

- La magnitud del proyecto.
- Riesgo de las estimaciones inexactas.
- Incertidumbre del proyecto.

2. La estimación de recursos requerida para un escenario dado no puede cambiarse arbitrariamente.

3. Las características que afectan a la precisión requerida de la estimación pueden ser:

- Riesgo inherente al proyecto.
- Fiabilidad de la información usada.
- Efectividad del proceso de estimación.
- Estimar con frecuencia: A medida que evoluciona el proyecto se dispone de más información que confirmará o refutará las estimaciones originales. Esa información servirá de base para realizar estimaciones más exactas del resto del proyecto.

Para que una estimación se acerque más a la realidad, el proyecto necesita conocer la cantidad de recursos para aproximar la duración del mismo. Para aproximar el tiempo de realización de un software se necesita conocer el tamaño del proyecto y los diferentes factores que pueden hacer que fracase. Para cambiar estas aproximaciones tienen que existir pruebas que afirmen que la estimación no es la adecuada y tener un equipo que se dedique a estimar continuamente.

Las principales causas que hacen necesarias las estimaciones son:[8]

- Investigación de la factibilidad del desarrollo o compra de un sistema nuevo.
- Investigación del impacto del cambio en la funcionalidad de un sistema existente.
- Determinación del personal de soporte necesario para el desarrollo del proyecto.
- Cuantificar el coste económico del sistema nuevo.

La estimación forma parte de la inferencia estadística. Los principales parámetros de estimación de software son: el esfuerzo para realizar el mismo, el coste y el tiempo que se tardará en desarrollarse el software.

1.3.2 Esfuerzo de desarrollo del proyecto.

Esfuerzo es probablemente una de las más populares métricas de software.[9]

La cuestión de que todos los líderes de proyectos quieren responder la pregunta de ¿Cuánto esfuerzo se va a tomar? es una de las principales causas posibles de la medición de esfuerzo para así calcular con facilidad el tiempo que demorará el proyecto. El esfuerzo tiene en cuenta el tamaño del software y los recursos humanos implicados.

Las principales causas de no calcular con exactitud el esfuerzo de un proyecto son:

- Confusión con el calendario previsto y los objetivos a estimar.
- El desarrollo de los equipos suele ser apresurado debido a las necesidades de las empresas y no estiman con exactitud la verdadera velocidad con que los desarrolladores puedan trabajar. Cuando las universidades o empresas firman un contrato pasan la mayor parte del tiempo formando el equipo de desarrollo sin conocer realmente el nivel de conocimiento que tendrá el equipo.
- La falta de un buen proceso de estimación.

1.3.3 Tamaño del producto.

Definir el tamaño del software representa el primer reto a cumplir por el líder del proyecto, siendo una producción cuantificable del proyecto de software. El tamaño del software influye en las estimaciones.

El manejador de costo principal para un proyecto de desarrollo de software es sin duda el tamaño del producto. La medida del tamaño debe ser tal que esté en relación directa con el esfuerzo de desarrollo, por lo que las mediciones de tamaño tratan de considerar todos los aspectos que influyen en el costo, como tecnología, tipos de recursos y complejidad. En torno a esta problemática, se han propuesto diversas soluciones para disminuir la incertidumbre en la estimación del tamaño. Entre otras, se encuentra el uso de registros históricos de medidas del tamaño de la aplicación y esfuerzo requerido.

La precisión de una estimación del proyecto de software se predice basándose en:[8]

- El grado en que el planificador ha estimado el tamaño de software.
- La habilidad para traducir la estimación del tamaño en esfuerzo humano, tiempo y costo.
- El grado en que el plan del proyecto refleja las habilidades del equipo de software.

- La estabilidad de los requisitos del software y el entorno que soporta el esfuerzo de la ingeniería del software.

1.3.4 Costo del Software.

Las técnicas de estimación del costo de software tienen sus propias fortalezas y debilidades. Se deben utilizar varias técnicas de estimación de costos y comparar sus resultados. Si estos predicen costos radicalmente diferentes, indica que no se tiene suficiente información para generarlos. Se debe buscar más información y repetir el proceso hasta que la estimación converja.

Un gran error en la estimación del costo puede ser lo que marque la diferencia entre beneficios y pérdidas, la estimación del costo y del esfuerzo del software nunca será una ciencia exacta, son demasiadas las variables: humanas, técnicas, de entorno, políticas, que pueden afectar el costo final del software y el esfuerzo aplicado para desarrollarlo.

Para realizar estimaciones seguras de costos y esfuerzos se tienen varias opciones posibles:[10]

- Dejar la estimación para más adelante (obviamente se puede realizar una estimación fiable después de haber terminado el proyecto).
- Basar las estimaciones en proyectos similares ya terminados.
- Utilizar técnicas de descomposición relativamente sencillas para generar las estimaciones de costos y esfuerzo del proyecto.
- Desarrollar un modelo empírico para el cálculo de costos y esfuerzos del Software.

La primera opción aunque parece atractiva no es práctica, la segunda puede funcionar bien si el proyecto actual es bastante similar a los esfuerzos pasados y si otras influencias del proyecto son similares. Las opciones restantes son métodos viables para la estimación del proyecto de software. Desde el punto de vista ideal, se deben aplicar conjuntamente las técnicas indicadas usando cada una de ellas como comprobación de las otras.

Antes de hacer una estimación, el planificador del proyecto debe comprender el ámbito del software a construir y generar una estimación de su tamaño.

1.3.5 Riesgos en la estimación y planificación.

Una de las personalidades más importantes que tiene la Ingeniería de Software, Roger. S. Pressman, cita la siguiente definición acerca del Riesgo, la cual fue planteada por Robert Charrette en su libro Análisis y Gestión de Riesgo.

“En primer lugar, el riesgo afecta a los futuros acontecimientos. El hoy y el ayer están más allá de lo que nos pueda preocupar, pues ya estamos cosechando lo que sembramos previamente con nuestras acciones del pasado. La pregunta es, podemos por tanto, cambiando nuestras acciones actuales, crear una oportunidad para una situación diferente y con suerte, mejor para nosotros en el

futuro. Esto significa, en segundo lugar, que el riesgo implica cambio, que puede venir dado por cambios de opinión, de acciones, de lugares... En tercer lugar, el riesgo implica elección y la incertidumbre que entraña la elección. Por tanto, el riesgo, como la muerte, es una de las pocas cosas inevitables de la vida”.

La estimación conlleva un riesgo inherente y es este riesgo el que lleva a la incertidumbre [8]. Entre las diversas clasificaciones dadas de riesgos se encuentran:

Los riesgos de proyecto: Son aquellos que amenazan el plan del proyecto. Si estos llegan hacerse realidad pueden provocar un retraso en la planificación temporal y los costos pueden aumentar considerablemente. Además de identificarse otros problemas que están enfocados principalmente en el personal, recursos, los clientes, requisitos y el impacto que pueda causar este en el proyecto.

Los riesgos técnicos: Son los que tienden amenazar la calidad del producto y la planificación temporal. Si este tipo de riesgo se llega a convertir en realidad, la implementación y otras disciplinas como el diseño, prueba y mantenimiento pueden llegar a ser difíciles o imposibles.

Los riesgos de negocio: Ponen en peligro la viabilidad del software a construir. A menudo amenazan el éxito del proyecto o el producto.

Existen una gran cantidad de herramientas de software disponibles relacionadas con la identificación de riesgos, las mismas se enfocan sólo en una categoría de riesgos o están orientadas a compañías que poseen una amplia base de datos organizacional que les permite generar información de categorías propias de riesgos.

1.3.6 Principales Métodos de estimación.

Se han desarrollado varias técnicas de estimación para el desarrollo de software, aunque cada una tiene sus puntos fuertes y débiles. Algunas de estas técnicas son:

Modelo COCOMO.

Es un modelo de estimación de costes de software que incluye tres modelos, donde cada uno ofrece un nivel de detalle y aproximación cada vez mayor a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado. Pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas y está orientado a la magnitud del producto final, midiendo el tamaño del proyecto en líneas de código principalmente, constituyendo uno de los modelos más documentados en la actualidad y fácil de utilizar. Es un modelo constructivo de costos con la siguiente jerarquía:[10]

- **Modelo I.** El Modelo COCOMO básico calcula el esfuerzo y el costo del desarrollo de Software en función del tamaño del programa, expresado en las líneas estimadas.

- **Modelo II.** El Modelo COCOMO intermedio calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de conductores de costos que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.
- **Modelo III.** El modelo COCOMO avanzado incorpora todas las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costos en cada caso (análisis y diseño) del proceso de ingeniería de Software.

Puntos de Función.

Es un método para medir el tamaño del software, el cual pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y explotación del software, y también ser útil en cualquiera de las fases de vida del software, desde el diseño inicial hasta la explotación y mantenimiento. Está basado en la descomposición funcional, es un método de cálculo de tamaño, estimación y medida del software que asegura independencia de la tecnología.

El análisis de los Puntos de Función se desarrolla considerando cinco parámetros básicos externos del Sistema:[8]

- Entrada (EI, External Input).
- Salida (EO, External Output).
- Consultas (EQ, External Query).
- Ficheros Lógicos Internos (ILF, Internal Logic File).
- Ficheros Lógicos Externos (EIF, External Interface File).

Puntos de Casos de Uso.

El método de Puntos de Casos de Uso es un método de estimación y cálculo de tamaño del software basado en cuentas realizadas sobre los casos de uso para un sistema de software.

El método exige la existencia de un modelo de casos de uso, por lo que debe utilizarse cuando exista algún entendimiento del dominio del problema o cuando se estén realizando las tareas de arquitectura y del tamaño del sistema. Por lo general, estas condiciones están dadas al término de las actividades de Análisis.

Puntos de Función ajustados y Coeficientes de Conversión.

Este método es difícil de aplicar si no se cuenta con una base histórica de proyectos que provea los coeficientes de conversión y la mayoría de los proyectos y empresas no cuentan con valores estadísticos. Es un método para medir el tamaño del software, pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y

explotación del software y también ser útil en cualquiera de las fases de vida del software, desde el diseño inicial hasta la explotación y mantenimiento. Éste método es uno de los más usados en la actualidad para la estimación del esfuerzo cuando se tiene información histórica a la cual recurrir.

COCOMO II y Puntos de Función sin ajustar como entrada.

Resulta muy útil para estimar un proyecto en forma global, cuando se tiene un conjunto de Casos de Uso bastante amplio (del orden de 50) y con escaso nivel de detalle. La técnica mencionada no está adaptada para proyectos menores a 2000 líneas de código, por lo cual no es aplicable a proyectos muy pequeños.

Modelos Basados en la Experiencia.

Se basan en la comparación de los resultados reales de diferentes proyectos, siempre entregados al cliente y en los que se ha aplicado la experiencia de los expertos como fuente de conocimiento esencial. Entre estos métodos se encuentran:

Opinión de expertos.

Se consultan varios expertos en las técnicas de desarrollo de software propuestas y en el dominio de aplicación. Cada uno de ellos estima el proyecto, estas estimaciones se comparan y discuten, luego el proceso de estimación se itera hasta que se acuerda una estimación.

La desventaja de este método es el alto costo en tiempo y recursos humanos necesarios para su implantación, así como la subordinación al nivel de experiencia y conocimientos en el entorno que puedan aportar los expertos. Las ventajas que posee indican que las estimaciones parciales son neutralizadas y se presenta una estimación global. Por otro lado las estimaciones suministradas por este grupo de expertos difícilmente pueden ser obviadas gracias a la trascendencia que la organización otorga a este proceso, al proporcionar costosos recursos a esta tarea.[11]

Estimación por analogía.

Esta técnica es aplicable cuando otros proyectos en el mismo dominio de aplicación se han completado. Se estima un nuevo proyecto por analogía con proyectos completados, dicha técnica es basada en información histórica de duraciones reales de actividades anteriores similares como bases de datos, conocimientos y experiencia del equipo de proyecto.

En la analogía pueden variar los siguientes factores como el tamaño, complejidad, usuarios y proceso de desarrollo de software utilizado. Utiliza medidas de los atributos del modelo empírico a fin de caracterizar el caso actual, para el que se realiza la estimación. Las medidas conocidas para el caso actual son usadas para buscar un conjunto de datos que identifiquen casos análogos. La predicción se hace intercalando desde uno o varios casos análogos al caso actual. Las ventajas

que proporciona este método es un menor costo en tiempo y recursos que el método del juicio del experto. Vale destacar que las estimaciones de proyectos anteriores no siempre se ajustan a nuevos proyectos, ya que muchos de los factores de estas estimaciones no siempre se mantienen, pero es una buena opción apoyar las estimaciones en proyectos similares.

1.3.7 Herramientas para Estimación y Planificación.

Las herramientas automáticas de estimación permiten estimar costos y esfuerzos, así como llevar a cabo análisis, con importantes variables del proyecto, tales como la fecha de entrega o la selección del personal. Aunque existen muchas herramientas automáticas de estimación, todas exhiben las mismas características generales y todas requieren de una o más clases de datos.

A partir de estos datos, el modelo implementado por la herramienta proporciona estimaciones del esfuerzo requerido para llevar a cabo el proyecto, los costos, la carga de personal, la duración y en algunos casos la planificación temporal de desarrollo y riesgos asociados.

En resumen el Jefe del Proyecto debe de estimar tres cosas antes de que comience el proyecto: cuánto durará, cuánto esfuerzo requerirá y cuántas personas estarán implicadas. Además debe predecir los recursos de hardware y software que requerirá y el riesgo implicado.

Para obtener estimaciones exactas para un proyecto, generalmente se utilizan al menos dos de los principales métodos de estimación referidos anteriormente. Mediante la comparación y la conciliación de las estimaciones obtenidas con las diferentes técnicas, el planificador puede obtener una estimación más exacta. La estimación del proyecto de software nunca será una ciencia exacta, pero la combinación de buenos datos históricos y técnicas puede mejorar la precisión de la estimación. Ejemplo de herramientas automáticas para estimar y/o planificar son las siguientes:

USC Cocomo II: Programa gratuito que implementa el modelo COCOMO II de estimación de tamaño y esfuerzo para proyectos software. Es una herramienta poco amigable y consta con una interfaz totalmente sencilla y práctica. Fue creada en el año 1999 y tiene una ayuda personalizada, que explica paso a paso como trabajar en él. [12]

ESTIMAC v1.0: Permite automatizar los cálculos necesarios para realizar la estimación de un proyecto informático basándose en el Análisis de puntos de casos de uso. Fue realizado en la Universidad de las Ciencias Informáticas por el ingeniero Henry Raúl González Brito en el curso 2005-2006 como apoyo a la actividad docente de la asignatura Ingeniería de Software. Esta herramienta pone en práctica la técnica de estimación Análisis de puntos de casos de uso de manera eficiente.

Para completar con éxito un proyecto de software, se necesita tener un control riguroso sobre el tiempo, las personas o los imprevistos que puedan surgir, como cambios en el software. Para

ayudar en la planificación y gestión de proyectos, Microsoft proporciona dos herramientas básicas: Microsoft Project y Microsoft Solutions Framework.

Microsoft Project (MSP): Es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

dotproject: Creado en el año 2000, construida con aplicaciones de código abierto basándose en la Web permitiendo tener un sistema multiusuario, es una herramienta de gestión de proyecto, creada por la comunidad de Software libre internacional, teniendo un sistema integrado de gestión. La composición del dotproject está dividida en dos módulos: Usuario y Administración.

- Permite crear y manipular nuevas empresas, nuevos proyectos, y nuevas tareas.
- Visualiza tareas y eventos.
- Permite gestionar ficheros de los proyectos.
- Muestra información de los usuarios del sistema.
- Permite crear foros y asociarlos a los proyectos.

El modulo de administración permite gestionar información acerca de los usuarios y permite configurar todos los parámetros del sistema.[13]

Trac: Es una herramienta de uso libre con una interfaz Web simple que integra herramientas para la comunicación, gestión, seguimiento de proyectos y gestión de la configuración. Cuenta con una interfaz de gestión de versiones y proporciona el seguimiento de hitos, eventos y evolución del desarrollo.

Gantt Project: Es una herramienta para el planeamiento de proyectos muy similar a MS Project. Está desarrollada en Java por lo que corre en cualquier sistema operativo, es bastante sencilla de usar y tiene como características importantes que puede importar archivos de MS Project y exportar a HTML, PDF y MS Project.

1.4 Estudio de Indicadores.

Los indicadores son “Herramientas para clarificar y definir, de forma más precisa, objetivos e impactos (...) son medidas verificables de cambio o resultado (...) diseñadas para contar con un estándar contra el cual evaluar, estimar o demostrar el progreso (...) con respecto a metas establecidas, facilitan el reparto de insumos, produciendo (...) productos y alcanzando objetivos”. [14]

Los indicadores son elementos informativos del control de cómo funciona una actividad, pues hacen referencia a parámetros estables que sirven de magnitud de comprobación del funcionamiento de ésta. La utilidad y fiabilidad del control de gestión de software se vincula

necesariamente a la utilidad y fiabilidad de los indicadores. Una vez elegidos los indicadores, se imponen sobre ellos técnicas de seguimiento.

Los indicadores controlan y evalúan mediante una cuantificación objetiva cualquier fenómeno de interés en una actividad determinada, además posibilitan el análisis del comportamiento y la comparación de las distintas actividades como medida para facilitar la toma de decisiones por parte de los Jefes de Proyectos.

Los indicadores son mediciones de los logros y el cumplimiento de la misión y objetivos de un determinado proceso de gestión que permiten administrar realmente un proceso, estos ayudan a establecer y administrar acciones concretas para hacer realidad las tareas y trabajos programados y planificados. Sirven como herramienta para el mejoramiento continuo de la calidad en la toma de decisiones, lo cual se traduce en una mejor calidad del producto o del servicio resultado del proceso.

Ventajas de la utilización de indicadores en el proceso de desarrollo de software:[15]

- Impulsar la eficiencia, eficacia y productividad de las actividades de cada uno de los negocios.
- Disponer de una herramienta de información sobre la gestión del negocio, para determinar qué también se están logrando los objetivos propuestos.
- Identificar oportunidades de mejoramiento en actividades que por su comportamiento requieren reforzar o reorientar esfuerzos.
- Contar con información que permita priorizar actividades basadas en la necesidad de cumplimiento de objetivos a corto, mediano y largo plazo.
- Disponer de información que permita contar con parámetros para establecer prioridades de acuerdo con los factores críticos de éxito y las necesidades y expectativas de los clientes.
- Evaluar y visualizar periódicamente el comportamiento de las actividades claves de la organización y la gestión general de la empresa con respecto al cumplimiento de su misión y objetivos.
- Reorientar políticas y estrategias con respecto a la gestión del proyecto.

El uso de indicadores permite de forma general evaluar en forma periódica el avance en el cumplimiento de la misión y de los objetivos en las diversas actividades de la empresa analizando los resultados del proyecto y estableciendo las variables que son factores críticos de éxito en cada uno de los negocios, actividades y operaciones de la empresa. Los indicadores posibilitan analizar las tendencias de estos en las diversas actividades claves y en los atributos que son factores críticos de éxito promoviendo el desarrollo de una cultura basada en la gerencia de datos y hechos, en busca del mejoramiento de la calidad.

1.4.1 Mediciones y Métricas para estimar los proyectos de software.

Las respuestas a los principales indicadores se consiguen a medida que avanza el desarrollo de software, para lo cual es necesario realizar mediciones sobre el producto de software que se está construyendo. En este sentido se puede definir a las medidas del software o métricas del software como: “La aplicación continua de técnicas basadas en las mediciones de los procesos de desarrollo del software y sus productos, para producir una información de gestión significativa y a tiempo” [6] que puede ser usada, entre otras, para mejorar la calidad de los procesos y productos mencionados.

Las métricas no son más que medidas para el desarrollo del software. Usualmente en los procesos de desarrollo las métricas o medidas tienen gran importancia ya que permiten un mejor entendimiento tanto de los procesos como de los productos. El proceso se mide con el objetivo de mejorarlo, mientras que el producto se mide para aumentar su calidad.

La necesidad de medir es algo evidente donde se arrojan resultados que permitan la gestión del proyecto. No obstante, la realidad es diferente y genera controversias acerca de la técnica a utilizar, la exactitud de los datos, las comparaciones entre personas, productos o procesos. Las métricas o medidas son fundamentales para la calidad de un software ya que son factores primordiales para el éxito del proyecto.

Métricas sobre el producto

Las métricas sobre el producto están orientadas a estimar las características del mismo antes de su desarrollo. Estas estimaciones se basan en el conocimiento que los desarrolladores adquieren a partir de datos obtenidos de proyectos anteriores.

- **Tamaño estimado:** La forma más obvia y la que se ha utilizado históricamente para estimar el tamaño es contar el número de líneas de código. Con ciertas normas para determinar qué es lo que se cuenta (líneas de comentario, código incluido) y siempre referido a un lenguaje concreto, el valor permite comparar con otros casos y estimar el esfuerzo necesario en futuros desarrollos.
- **Complejidad estimada:** Con el fin de superar el problema de las estimaciones del tamaño, se ha prestado recientemente atención a medidas de complejidad no basadas en estimaciones de números de líneas.
- **Robustez:** Por robustez de un programa se entiende la ausencia de fallos en su ejecución con diferentes datos de entrada durante intervalos de tiempo predeterminados. La robustez de un programa está ligada a la aparición de problemas durante su ejecución. Generalmente, el número de fallos encontrados durante la fase de prueba y posteriormente durante el

mantenimiento del sistema, constituye una medida de la calidad del producto de software e indirectamente, de la calidad del proceso de desarrollo.

Métricas sobre el proceso

Las métricas mencionadas anteriormente estaban orientadas a conocer la complejidad del producto para estimar los recursos necesarios para su realización. Según se vayan acumulando los datos y se analicen estadísticamente, las estimaciones serán cada vez más acertadas sirviendo para planificar mejor futuros desarrollos. Existen otros tipos de datos que se pueden tomar durante el desarrollo de un producto de software y que no están ligados al producto sino a los procesos implicados. El análisis de cómo estos procesos se realizan a partir de medidas tomadas en el desarrollo es la base para su posterior mejora.

Algunos de los elementos a medir son:

- Distribución del esfuerzo en cada una de las fases, con el objetivo de estimar los recursos necesarios. Esta medida es complementaria a las de tamaño que permitían conocer los recursos globales necesarios; de lo que se trata aquí es de obtener medidas reales y vincularlas a futuros proyectos.
- Productividad medida en número de líneas de código documentadas que es capaz de producir una persona en una unidad de tiempo. Los valores típicos de productividad por persona están entre 30 y 50 líneas de código por día de trabajo.

Métricas del Proyecto.

El proyecto debe ser caracterizado en términos de tamaño, tipo, complejidad y formalidad porque condicionan expectativas sobre las distintas tendencias a seguir a la hora de las mediciones. Algunas de las características que describen son: el número de programadores de un software, el costo, la planificación, y la productividad del equipo.

Las principales métricas a tener en cuenta son:[16]

- Modularidad: Promedio de daños debido a cambios perfectivos o correctivos en la implementación.
- Adaptabilidad: Promedio de esfuerzo debido a cambios perfectivos o correctivos en la implementación.
- Madurez: Tiempo de prueba activo / número de cambios correctivos.
- Mantenimiento: Mantenimiento productivo / desarrollo productivo.
- Progreso del proyecto: Debe reportarse basándose en el plan del proyecto desde la perspectiva del valor.

Las mediciones permiten caracterizar y comparar evaluaciones futuras, evaluando y controlando el avance del proyecto. Además ayudan a predecir, planificar y estimar en base a datos históricos para mejorar la calidad del producto.

Las métricas de proyecto y los indicadores derivados de ellas los utilizan los jefes de proyectos y su equipo para adaptar el flujo de trabajo del proyecto y las actividades técnicas. La primera aplicación de métricas de proyectos en la mayoría de los proyectos de software ocurre durante la estimación. Las métricas recopiladas de proyectos anteriores se utilizan como una base desde la que se realizan las estimaciones del esfuerzo y del tiempo para el actual trabajo de software. A medida que avanza un proyecto, las medidas del esfuerzo y del tiempo consumido se comparan con las estimaciones originales. El jefe de proyecto utiliza estos datos para supervisar y controlar el avance.

Métricas de Recursos.

Las métricas de los recursos medirán los elementos que incluyen a personas (experiencia, habilidades, coste y funcionamiento), los métodos y las herramientas (en términos de efecto sobre productividad, calidad y coste), tiempo, esfuerzo, y presupuesto (recursos consumidos y recursos restantes).

Métricas de Productividad.

La productividad se define como la razón entre entradas/salidas. Para el caso de la ingeniería de software, se define como la cantidad de esfuerzo requerido para lograr un cierto grado de productividad.

1.4.2 Indicadores en la Gestión de Proyectos.

“Un ingeniero de software recopila medidas y desarrolla métricas para obtener indicadores“.[8] Un indicador es una métrica o combinación de métricas que proporciona una visión profunda del proceso, del proyecto o del producto. También se puede definir como un instrumento de medición de las variables asociadas a las metas.

Los indicadores de gestión por su parte, se entienden como la expresión cuantitativa del comportamiento o el desempeño de toda una organización o una de sus partes, cuya magnitud al ser comparada con algún nivel de referencia, puede estar señalando una desviación sobre la cual se tomarán acciones correctivas o preventivas según el caso. El valor del indicador es el resultado de la medición del indicador y constituye un valor de comparación, referido a su meta asociada.

En el desarrollo de los indicadores se deben identificar necesidades propias del área involucrada, según la naturaleza de los datos y la necesidad del indicador. Esto es fundamental para el

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

mejoramiento de la calidad, debido a que son medios económicos y rápidos de identificación de problemas.

El principal objetivo de los indicadores es evaluar el desempeño del área mediante parámetros establecidos en relación con las metas, así mismo observar la tendencia en un lapso de tiempo durante un proceso de evaluación.

Cada medidor o indicador debe satisfacer los siguientes criterios:[17]

- **Medible:** El medidor o indicador debe ser medible. Esto significa que la característica descrita debe ser cuantificable en términos ya sea del grado o frecuencia de la cantidad.
- **Entendible:** El medidor o indicador debe ser reconocido fácilmente por todos aquellos que lo usan.
- **Controlable:** El indicador debe ser controlable dentro de la estructura de la organización.

Los indicadores se clasifican en indicadores de proceso, proyecto, eficacia, eficiencia, estratégicos, tácticos y operacionales. A continuación se hace referencia a cada una de estas clasificaciones.

Los indicadores Estratégicos miden la relación que tienen los stakeholders con el desarrollo de software. Se clasifican en: Indicadores Cognoscitivos que miden el grado de conocimiento que los stakeholders tienen frente a los mensajes generados por la organización. Expresan qué se conoce de la organización. Los indicadores Estratégicos Afectivos miden las percepciones, emociones, sentimientos y reputación que una organización genera entre sus stakeholders. Expresan qué piensan de la organización. Los Indicadores Estratégicos Conductuales: Miden las conductas probables que pueden asumirse por parte de los stakeholders hacia la organización. Dicen qué hacen o podrían hacer con relación a la organización

Los indicadores de Proceso permiten al equipo de desarrollo del software tener una visión profunda de la eficacia de un proceso ya existente (por ejemplo: el paradigma, las tareas de ingeniería del software, productos de trabajo e hitos). También permiten que los gestores evalúen lo que funciona y lo que no. Tienen mucha relación con las métricas del proceso, se recopilan de todos los proyectos y durante un largo período de tiempo. Su intento es proporcionar indicadores que proporcionen mejoras de los procesos de software a largo plazo.

Los indicadores de Proyecto permiten al líder evaluar el estado del proyecto en curso siguiendo la pista de los riesgos potenciales y detectando las áreas de problemas antes de que se conviertan en críticas. Además permiten también ajustar el flujo y las tareas del trabajo evaluando la habilidad del equipo de proyecto en controlar la calidad de los productos de software. Por su parte las métricas del proyecto que se recopilan de proyectos anteriores, permiten estimar esfuerzo, costo y tiempo para el actual proyecto, sirven para supervisar y controlar el avance del proyecto.

Los indicadores de Eficacia miden el logro de los resultados propuestos, indicando si se hicieron las tareas que se habían planificado, es decir los aspectos correctos del proceso. Los indicadores de eficacia se enfocan en el qué se debe hacer, por tal motivo, en el establecimiento de un indicador de eficacia es fundamental conocer y definir operacionalmente los requerimientos del cliente para comparar lo que entrega el proceso contra lo que él espera. De lo contrario, se puede estar logrando una gran eficiencia en aspectos no relevantes para el cliente.

Los indicadores de Eficiencia miden el nivel de ejecución del proceso, se concentran en el cómo se hicieron las cosas y miden el rendimiento de los recursos utilizados por un proceso estando muy relacionados con la productividad.

Por su parte los indicadores Tácticos miden la calidad de las diferentes vías de comunicación que tienen los stakeholders con la gerencia del proyecto. Dentro de los indicadores tácticos encontramos los indicadores tácticos controlados los cuales miden la calidad de los escenarios de comunicación utilizados por la organización para comunicarse con sus stakeholders. Además los indicadores tácticos no controlados miden la calidad del contenido divulgado a través de medios masivos.[18]

Se han desarrollado herramientas para la recolección manual de los indicadores de productividad y calidad de empresas de Software. Entre ellas se encuentran:

PSP (Personal Software Process): Es usado por los ingenieros de software para recolectar y analizar datos sobre su trabajo. Los estudios publicados generalmente usan datos recolectados usando PSP para esbozar las conclusiones cuantitativas sobre el impacto en el comportamiento de los programadores y la calidad del producto.

TSP (Team Software Process): Es un proceso que sirve como herramienta para equipos de 2 a 20 ingenieros de software entrenados en el uso de PSP.

PSP y TSP se desarrollaron en el SEI, instituto creado en 1986 bajo el patrocinio del departamento de defensa de los Estados Unidos, con el fin de mejorar las prácticas de ingeniería de software.

1.4.3 Diferencia entre medidas, estadísticas e indicadores.

Continuamente se confunde a los indicadores con series de datos disponibles, medidas y estadísticas; por ello, es pertinente aclarar sus diferencias.

Un dato hace referencia tanto a los antecedentes necesarios para el conocimiento de un determinado objetivo como a cada una de las cantidades que constituyen la base de un problema matemático y estadístico, esto constituye un insumo para un proceso de construcción de conocimiento y de estadísticas.

Una medida remite a imponer parámetros de cantidad, peso y volumen permitiendo evaluar en términos cuantitativos la importancia de un objeto o fenómeno comparándolo con otro de la misma

especie, pero que difiere de tamaño. La precisión y valoración son las principales cualidades que una medida permite efectuar entre los diversos objetos, sin embargo, no todas las medidas indican algo, en el sentido utilitario y semántico del término, por lo cual, si bien todos los indicadores son medidas, no ocurre igual de manera inversa.

De forma tradicional, la estadística ha sido la ciencia que se ha dedicado a la unificación de todos los hechos que se pueden valorar de forma numérica para hacer comparaciones entre las cifras y obtener conclusiones aplicando la teoría de las probabilidades. Incluye al conjunto de datos que son la materia prima para la construcción de estadísticas, así como el proceso metodológico para su tratamiento.

Como ciencia pura, la estadística debe preocuparse por definir, implantar y monitorear las metodologías y requisitos técnicos mínimos que las estadísticas deben cumplir para considerarse válidas, consistentes, confiables y representativas del fenómeno que se está midiendo. Ella define los estándares técnicos y operativos fundamentales para la obtención de datos necesarios con validez y utilidad. Las estadísticas sirven para diferentes propósitos, entre los que se encuentran el conteo y registro. Sólo una pequeña proporción de las estadísticas pueden ser consideradas como indicadores; para ello se necesita, además de cumplir con todos los requisitos técnicos de una estadística, que la misma responda a ciertos requerimientos de información representativos.

1.5 Datos Históricos en la Gestión de Proyectos.

Debido a la problemática de la estimación del software, una solución para disminuir la incertidumbre en la estimación se encuentra el uso de registros históricos de medidas del tamaño de la aplicación, esfuerzo y tiempo requerido y el uso de distintos sistemas que aporten inteligencia al análisis de esta información.

Una de las etapas en la gestión del proyecto es la estimación del esfuerzo y plazo de cada una de las actividades de las que constará, por lo tanto, es necesario disponer de buena información sobre la duración y el esfuerzo necesarios para realizar cada tarea.

Disponer de esa información facilitará la gestión del proyecto tanto a nivel de costes como de plazos, facilitando la distribución de recursos y disminuyendo los riesgos o zonas críticas.

La información histórica, procedente del cierre de proyectos anteriores, proporcionará la base de conocimiento sobre la que se podrán aplicar las técnicas para extrapolar este conocimiento a futuros proyectos.[19]

El control del coste y su precisión a lo largo de la vida del proyecto y la recopilación de los datos característicos de cada proyecto al finalizar éste, producen un efecto de corrección continua que permite fijar modelos específicos que facilitarán la planificación de los nuevos proyectos en todos los aspectos: costes, plazos y recursos.

Una buena opción para estimar es la utilización de registros históricos de la propia organización. Es decir, establecer una base de datos histórica de proyectos que registre tamaño, coste y duración, estos datos históricos ofrecerán la mejor fuente para la estimación de proyectos futuros.

Si un proyecto es bastante similar en tamaño y funcionalidades, es muy probable que un nuevo proyecto requiera aproximadamente la misma cantidad de esfuerzo y que su tiempo de desarrollo sea similar al proyecto anterior. Si el proyecto es totalmente distinto entonces puede que la experiencia obtenida en el registro histórico no sea suficiente.

Estimar significa, de alguna forma, predecir el futuro. Una de las formas de predecir el futuro es tomar en cuenta lo que sucedió en el pasado. En ese sentido, las técnicas de estimación se basan de una forma u otra en datos históricos y experiencias previas sobre elementos similares al que se debe estimar. Las comparaciones están basadas en datos históricos usando técnicas de analogía, también usando el consejo de expertos. La comparación de proyectos con otros similares es una actividad crítica para lograr una estimación exitosa, por lo que los datos históricos ayudan a predecir y a planear.

1.5.1 Fundamentos de las Herramientas Utilizadas.

UML (Lenguaje Unificado de Modelación)

El Lenguaje Unificado de Modelado es un lenguaje de sistemas de software muy conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Ofrece un estándar para describir un modelo del sistema, 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.

RUP (Proceso Unificado de Desarrollo de Software)

En el desarrollo de la propuesta del sistema para registrar los datos históricos se decidió utilizar como metodología RUP, por sus características y las facilidades que aporta a todo el proceso de desarrollo. Dicha metodología se utiliza conjuntamente el UML para expresar gráficamente todos los esquemas de un sistema de software, sin dejar de mencionar que este lenguaje es bastante independiente del proceso de desarrollo que se sigue.

RUP es un proceso de desarrollo de software que unido con UML constituye una metodología muy utilizada para el análisis, diseño e implementación de sistemas. RUP brinda un proceso integrado de forma iterativa e incremental a partir de la identificación e implementación de los casos de uso centrándose en la arquitectura. Es un proceso que de manera ordenada define las tareas y sus responsables.

Visual Paradigm

Como herramienta para modelar visualmente se utilizó Visual Paradigm pues a pesar de que Rational Rose Enterprise Suite es muy recomendado y altamente profesional, existen algunos puntos en los que Visual Paradigm lo supera. Rational Rose no genera documentos a diferencia del Visual Paradigm, Rational no es multiplataforma como Visual Paradigm sino que está disponible para varios sistemas operativos como son: Windows, Linux y Unix.

Es una herramienta amigable para el usuario, puede ser usada en varios idiomas y cada componente utilizado en el diagrama que se esté creando, sugiere nuevos posibles componentes a utilizar, por lo que ya no es necesario localizarlos en la barra y así se crea fácilmente cualquier tipo de diagrama. Posee un amplio número de estereotipos que proporciona la creación de diagramas de fácil entendimiento, además de que estos diagramas los organiza automáticamente.

1.6 Conclusiones parciales.

Después de analizar los conceptos más importantes que formarán parte de este trabajo se ha podido apreciar la variedad de métodos que existen para estimar el tiempo, esfuerzo y tamaño de los proyectos de software. Entre los diferentes métodos se aprecia la importancia de los datos históricos para las estimaciones mediante analogía para tomar decisiones en estimaciones futuras. Se estudió profundamente la gestión de proyectos basándose principalmente en la guía de los Fundamentos de la Dirección de Proyectos (PMBOK) y en la Norma ISO 10006 dirigida a la calidad en la Gestión de Proyectos. Además se estudiaron los diferentes indicadores y métricas que permitirán recoger los diferentes datos históricos durante toda la etapa de desarrollo de software explicándose las diferentes herramientas para lograr una buena planificación y estimación.

CAPÍTULO 2: PROPUESTA DE INDICADORES.

En este capítulo se ilustra una panorámica de la situación actual que presenta la UCI en la producción y los principales problemas en la gestión de proyectos de software definiendo como solución una propuesta de indicadores influyentes para estimar el tiempo y el esfuerzo en los proyectos productivos de software, sometiendo esta propuesta al criterio de un panel de expertos en el tema. Además se plantea una propuesta inicial del análisis estadístico de los datos. Los indicadores propuestos servirán de referencia en la toma de decisiones en la producción de la UCI.

2.1 Panorámica de la situación de la producción en la UCI.

Dentro de las características distintivas de la UCI se encuentra la producción de software, actualmente este renglón tan importante puede considerarse incipiente ya que solo cuenta con 6 años de creada y quizás menos tiempo de experiencia acumulada en la producción. En la universidad, se desarrollan proyectos tanto nacionales como internacionales, en los proyectos nacionales generalmente no se estima el tiempo ni el esfuerzo a realizar aunque en el Expediente de Proyecto se recogen una serie de datos importantes a la hora de estimar. En los proyectos internacionales se estableció una estrategia que aporta rapidez en la creación de los contratos de los proyectos, la cual consiste en realizar un mini proyecto técnico llamado Ficha Técnica que recoge los aspectos más importante a tener en cuenta para el costo del software entre los que se encuentra contar con la visión de la organización y las necesidades de informatización, con lo cual se calcula el tiempo y el esfuerzo de los proyectos y por último utilizando estos dos elementos se calcula el costo. Después de la firma del contrato se acaba de realizar el proyecto técnico, en el mismo se define la distribución del presupuesto dado para la realización del proyecto.

Una ventaja importante que tiene este proceso es la rapidez con que se logra realizar dicho contrato, característica esta que permite retener al cliente, pero los resultados hasta el momento no han sido los más idóneos, porque en la ficha técnica no se define la complejidad de sus procesos y por ende el costo no es muy cercano a la realidad y no se tienen en cuenta los riesgos que atentan contra la etapa de desarrollo, lo que conlleva que el tiempo estimado no sea el mismo que el tiempo real. También en la ficha técnica no se definen los recursos, ni el nivel de conocimiento que puede tener el equipo de desarrollo, aspectos importantes para definir el esfuerzo y el tiempo de desarrollo; no definir los recursos inicialmente puede causar que el tiempo y el esfuerzo estimado no coincidan con el tiempo y esfuerzo real.

CAPÍTULO II: PROPUESTA DE INDICADORES

En este caso existen muchos proyectos cuyo tiempo estimado y real no coinciden, ejemplo de esta situación fueron los proyectos de exportación con Venezuela SIGEP, CICPC y CTAISC, la comparación de los valores en la fase de inicio de cada proyecto se muestra en la **Tabla 2.1** a continuación:

Fase de Inicio	Proyecto SIGEP	Proyecto CICPC	Proyecto CTAISC
Etapas Estimadas(Semanas)			
Levantamiento de Requisitos.	2	2	4
Prototipo	2	1	3
Total	4	3	7
Etapas Reales(Semanas)			
Modelado del Negocio	5	6	7
Levantamiento de Requisitos.	12	5	9
Total	17	11	16

Tabla 2.1 Tiempo estimado y real de proyectos de la UCI.

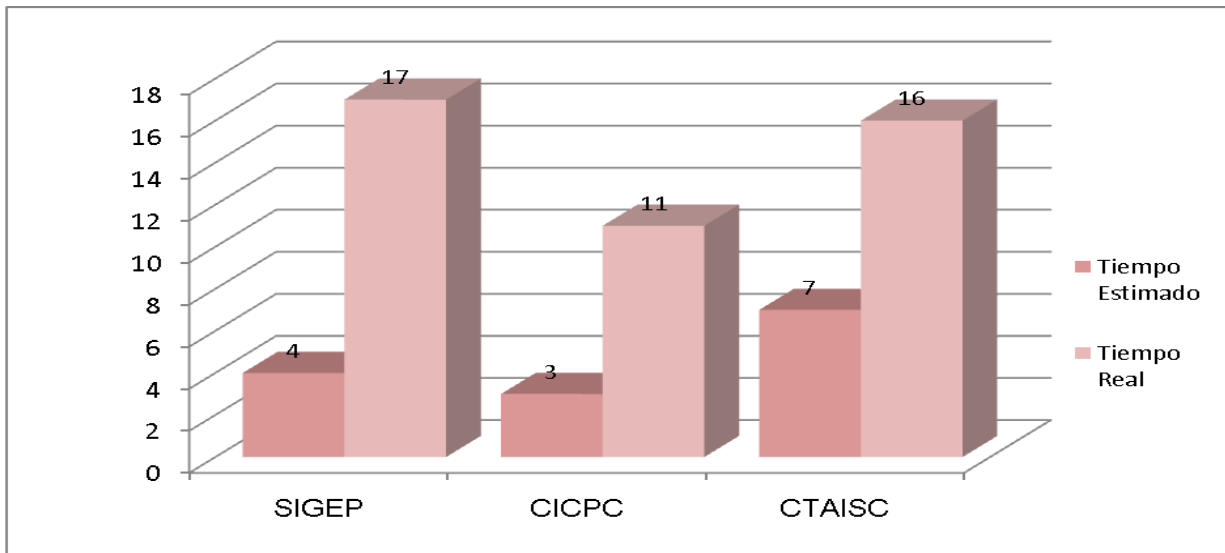


Figura 2.1 Comparación del tiempo de desarrollo en semanas de proyectos de la UCI.

En los proyectos de la figura anterior durante la fase de inicio se estimó el tiempo del levantamiento de requisitos y prototipo, sin embargo quedó demostrado que las actividades anteriores no eran suficientes y que debían ser completadas con la descripción del negocio, tarea que les permitió ganar en entendimiento y claridad. En estos proyectos no se tuvo en cuenta el impacto de los riesgos como por ejemplo el nivel de experiencia que tenían los desarrolladores, aspecto importante

CAPÍTULO II: PROPUESTA DE INDICADORES

para el desarrollo del proyecto. Otro aspecto que influyó fue la inadecuada estimación de los procesos a automatizar causando un atraso de más del doble de lo estimado.

La Norma ISO 10006 según se estudió en el Capítulo 1 describe la mejora continua del desarrollo donde define la importancia de plasmar las experiencias de los proyectos para que nuevos proyectos de desarrollo no cometan los errores que cometieron esos proyectos en el pasado. Como ya se definió anteriormente, la solución para que se realicen con más éxito las estimaciones del tiempo y esfuerzo en el inicio de la etapa de desarrollo de software es desarrollar un sistema que recoja la experiencia de todos los proyectos realizados en la universidad, esto logra una mejor decisión para definir los recursos necesarios en el proyecto, definir con mejor precisión la cantidad de procesos a automatizar y definir el impacto que tendrán los riesgos que influyen en el desarrollo de software. Para recoger los datos se propone definir indicadores y posteriormente cuando la base histórica contenga suficientes datos llegar a conclusiones estadísticas sobre el comportamiento general de los indicadores en la producción de la UCI. Contar con indicadores influyentes en las estimaciones y planificaciones en la toma de decisiones de los proyectos será un gran apoyo en la gestión de proyectos.

2.2 Propuesta de Indicadores.

Los indicadores que a continuación se proponen están basados en la experiencia de varios jefes de proyectos de la UCI y en la principal bibliografía consultada como el libro Fundamentos de la Dirección de Proyectos (PMBOK) y en la Norma ISO 10006 dirigida a la calidad en la Gestión de Proyectos referenciada en el capítulo anterior, así como en las actividades asociadas a la planificación vistas en el mismo capítulo. Los indicadores propuestos son los siguientes:

No	Indicadores de Proyecto
1	Tipo de Proyecto
2	Entorno del Proyecto.

Tabla 2.2 Indicadores de Proyecto Propuestos

No	Indicadores de Proceso
1	Gestión de la Calidad.
2	Pruebas de Software
3	Despliegue de la aplicación

Tabla 2.3 Indicadores de Proceso Propuestos

CAPÍTULO II: PROPUESTA DE INDICADORES

No	Indicadores de Recursos
Recursos de Entorno	
1	Metodología de desarrollo de software
2	Lenguaje de Programación
3	Estilo Arquitectónico
4	Patrón de Arquitectura
5	Herramienta de Modelado Visual
6	Herramienta de Programación
7	Herramienta Sistema Gestor Base de Datos
8	Herramientas utilizadas en la Planificación
9	Herramientas utilizadas en la Gestión de Configuración
10	Herramientas utilizadas en la Estimación
11	Cantidad de PC.
Recursos Reutilizables	
1	Componentes Reutilizables.
Recursos Humanos	
1	Cantidad de integrantes del equipo de desarrollo.
2	Nivel de conocimiento de los integrantes del equipo

Tabla 2.4 Indicadores de Recursos Propuestos

No	Indicadores de Estimación
1	Cantidad de versiones.
2	Esfuerzo de desarrollo
3	Tiempo de desarrollo
4	Tamaño del producto
5	Cantidad de procesos de complejidad de alta
6	Cantidad de procesos de complejidad de media
7	Cantidad de procesos de complejidad de baja
8	Influencia de los Riesgos en la estimación

Tabla 2.5 Indicadores de Estimación Propuestos.

2.3 Fundamentos de los Indicadores Propuestos.

2.3.1 Indicador Tipo de Proyecto.

El Tipo de Proyecto es un indicador cualitativo que especifica el tipo del producto que se está desarrollando. En la UCI se desarrollan 5 tipos de proyectos: Gestión, Multimedia, Portales, Realidad Virtual e Investigación, donde cada uno engloba todos los recursos, actividades y artefactos específicos para ese tipo de proyecto y se organizan para lograr un producto de software. La aplicación de la métrica de proyecto en la mayoría de los proyectos de software ocurre durante la estimación, ya que mide la complejidad del producto según sus características. Los datos recopilados de proyectos anteriores se utilizan como una base para la realización de las estimaciones del esfuerzo y del tiempo para el actual trabajo del software. A medida que avanza un proyecto, las medidas del esfuerzo y del tiempo consumido se comparan con las estimaciones originales y la planificación de proyectos según su clasificación. Este indicador permite recoger la cantidad de proyectos de cada tipo que se están desarrollando en la UCI.

2.3.2 Indicador Entorno de Proyecto.

El Entorno del Proyecto es el indicador que mide el impacto del marco donde los desarrolladores han trabajado, así sea en la planificación, en el desarrollo de software, o en el despliegue. El PMBOOK describe que todos los proyectos se planifican e implementan en un contexto social, político, económico y ambiental donde tienen impactos positivos y negativos y que puede influir notablemente en el tiempo y esfuerzo del proyecto.

Cuando hay una afectación cultural y social, el equipo tiene que comprender cómo afecta a los integrantes del proyecto. Esto puede requerir una comprensión de los aspectos económicos, demográficos, educativos, éticos, étnicos, religiosos, y de otras características de las personas a quienes afecta el proyecto o que puedan tener un interés en éste. El líder del proyecto también debe examinar la cultura de la organización y determinar si se reconoce que la dirección de proyectos desempeña un rol válido con responsabilidad y autoridad para gestionar el proyecto. En el entorno internacional y político es posible que algunos miembros del equipo tengan que estar familiarizados con las leyes y costumbres internacionales, nacionales, regionales y locales aplicables, así como con el clima político que podría afectar al proyecto. Otros factores internacionales a tener en cuenta son las diferencias de husos horarios, los días festivos nacionales y regionales, los requisitos de viaje para reuniones cara a cara y la logística de teleconferencias. Si en el proyecto se afecta su ámbito físico, algunos miembros del equipo deberán estar familiarizados con la ecología local y la geografía física que podrían afectar al proyecto o ser afectadas por el proyecto.

CAPÍTULO II: PROPUESTA DE INDICADORES

Para medir el impacto que tuvo el entorno en los diferentes proyectos se definió la siguiente escala con su correspondiente criterio de decisión.

Valor	Escala	Criterios
1	Muy Satisfactorio	<ul style="list-style-type: none"> • Cuando la entrega de un producto no fue atrasada por ningún factor social, político económico ó ambiental y que la entrega haya sido aceptada por el cliente. • Si hubo alguna afectación como los días feriados del país y el líder trazó una estrategia para enfrentar la situación como. • Si se implantó un plan de riesgos dirigido a factores de entorno y su mitigación fue exitosa.
2	Satisfactorio	<ul style="list-style-type: none"> • Cuando la entrega de un producto no fue atrasada por ningún factor social, político, económico ó ambiental y que la entrega haya sido aceptada por el cliente. • Si hubo alguna afectación para la cual el líder haya trazado una estrategia para enfrentar la situación. • Si con la estrategia planificada dirigida al entorno de proyecto provocó un mayor esfuerzo del equipo de desarrollo sin afectar la fecha de entrega de un producto.
3	Poco Satisfactorio	<ul style="list-style-type: none"> • Cuando la entrega de un producto fue atrasada por factores sociales, políticos, económicos y la entrega no fue aceptada por el cliente. • Si hubo alguna afectación que el líder ó un pequeño grupo no haya trazado una estrategia para enfrentar la situación, que haya provocado el aumento del esfuerzo estimado.
4	No satisfactorio	<ul style="list-style-type: none"> • Cuando la entrega de un producto fue atrasada por un factor social, político, económico ó ambiental que se pudo haber previsto.

Tabla 2.6 Escala de impacto del indicador Entorno de Proyecto.

2.3.3 Indicadores de Proceso.

En la propuesta de indicadores de proceso se recogen todas las características del proceso de desarrollo del software que permitan al líder de proyecto y al equipo de desarrollo tener una visión

CAPÍTULO II: PROPUESTA DE INDICADORES

profunda de la eficacia y calidad de un proceso. Su intento es proporcionar indicadores que permitan mejoras de los procesos de software, incluyendo el despliegue de la aplicación debido a que es una etapa a la cual generalmente no se le confiere la importancia adecuada. Estos indicadores servirán de referencia y experiencia de procesos ya concluidos en proyectos productivos, procesos vitales como la gestión de la calidad del software y la ejecución de las pruebas, que pueden influir en el tiempo y esfuerzo de los proyectos si se realizan incorrectamente.

2.3.3.1 Indicador Gestión de la Calidad.

El Indicador de Gestión de la Calidad proporcionará el impacto que tuvieron las actividades de calidad en los proyectos, incluyen todas las actividades de la organización ejecutante que determinan las políticas, los objetivos y las responsabilidades relativos a la calidad de modo que el proyecto satisfaga las necesidades por las cuales se emprendió. Ejemplo de las actividades de las que se desea recoger y medir el impacto es la planificación de la calidad, si se realizó del plan de aseguramiento de la calidad y realizar un control de la calidad, también es posible conocer si se utilizaron herramientas para el control de la calidad, lo que permitirá a los líderes de proyectos o asesores de calidad contar con los resultados y el comportamiento de este indicador de calidad en la UCI.

Para medir el impacto que tuvo la gestión de la calidad en los proyectos se definió la siguiente escala con su correspondiente criterio de decisión.

Valor	Escala	Criterios
1	Muy Satisfactorio	<ul style="list-style-type: none">• La gestión de la calidad en el proyecto no influyó en el tiempo y esfuerzo de desarrollo.• Se desarrolló y ejecutó el plan de aseguramiento de la calidad satisfactoriamente.• Se identificaron normas de calidad relevantes para el proyecto y se implantaron con éxito.• Se supervisaron los resultados específicos del proyecto, para determinar si cumplen con las normas de calidad identificadas.• Se planificó y estimó el coste de la calidad para que no influyera de forma negativa en la planificación de calendarios.• Se desarrolló la planificación de calidad de forma

CAPÍTULO II: PROPUESTA DE INDICADORES

Valor	Escala	Criterios
		paralela a los demás procesos de planificación. <ul style="list-style-type: none">• El uso de herramientas para el control de la calidad fue muy satisfactorio.
2	Satisfactorio	<ul style="list-style-type: none">• Se cumplió con los requisitos del cliente pero haciendo trabajar en exceso al equipo de desarrollo.• Cumplieron con los objetivos del cronograma del proyecto ejecutando apresuradamente las inspecciones de calidad.
3	Poco Satisfactorio	<ul style="list-style-type: none">• El cliente estuvo parcialmente de acuerdo con la calidad del producto.• El plan de aseguramiento de la calidad definido no abarcaba todos los aspectos para cumplir con toda la calidad requerida.
4	No satisfactorio	<ul style="list-style-type: none">• El producto no cumplió con las expectativas del cliente.• No se cumplió con el tiempo estimado de desarrollo debido a que las tareas de aseguramiento de la calidad no estaban planificadas.• No se desarrolló un plan de aseguramiento de la calidad.

Tabla 2.7 Escala de impacto del indicador gestión de la calidad.

2.3.3.2 Indicador Pruebas de Software.

Las pruebas de software es un proceso utilizado para identificar posibles fallos de implementación, calidad, usabilidad, estabilidad y seguridad de un sistema.

El único instrumento adecuado para determinar el status de la calidad de un producto de software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante técnicas de prueba. El proceso de pruebas, sus objetivos y los métodos y técnicas usados se describen en el plan de prueba.

Este indicador permitirá conocer como se desarrollaron las pruebas en los proyectos de la UCI, se medirá de acuerdo a la realización de las distintas actividades de este proceso y que tipo de pruebas se llevaron a cabo con su impacto correspondiente en el proyecto, si fue satisfactorio o no realizar las pruebas que definieron.

CAPÍTULO II: PROPUESTA DE INDICADORES

Para medir el impacto que tuvo el indicador de pruebas de software en los proyectos se definió la siguiente escala con su correspondiente criterio de decisión.

Valor	Escala	Criterios
1	Muy Satisfactorio	<ul style="list-style-type: none">• Se estableció una estrategia de pruebas que abarcara todas las posibles pruebas a desarrollar.• La ejecución del plan de pruebas trazado cumplió con todos los objetivos planteados sin afectar el tiempo y esfuerzo estimado para el desarrollo del software.• Se realizaron pruebas de caja negra y caja blanca con resultados muy satisfactorios.• Las técnicas utilizadas probaron todos los caminos posibles en la busca de defectos.• Se desarrollaron las pruebas en todos sus niveles(desarrollador, independiente, integración, sistema, unidad y aceptación)
2	Satisfactorio	<ul style="list-style-type: none">• Se cumplió con el plan de prueba trazado pero haciendo trabajar en exceso al equipo de pruebas.• Se cumplió con el tiempo de desarrollo pero las tareas de prueba se hicieron apresuradamente.
3	Poco Satisfactorio	<ul style="list-style-type: none">• No se desarrollaron las pruebas de caja negra y caja blanca en su totalidad.• No se desarrollaron las pruebas en la mayoría de sus niveles(desarrollador, independiente, integración, sistema, unidad y aceptación)
4	No satisfactorio	<ul style="list-style-type: none">• No se estableció una estrategia de pruebas.• El plan de pruebas estuvo incompleto por lo que influyó en el tiempo y esfuerzo estimado para el desarrollo del software.

Tabla 2.8 Escala de impacto del indicador pruebas de software.

2.3.3.3 Indicador Despliegue de la Aplicación.

Por cada release del producto de software se genera un despliegue, este garantiza que el producto esté disponible para los stakeholders. Para que el despliegue se efectúe con calidad se requiere de un plan de despliegue donde se documente cómo y cuándo se debe implantar el

CAPÍTULO II: PROPUESTA DE INDICADORES

producto para el cliente. También incluye la instalación del software, la migración al nuevo software, así como la ayuda y formación de nuevos usuarios y la estrategia para mitigar los riesgos pertinentes a esta etapa. También son imprescindibles las pruebas en el despliegue porque son el resumen de que el producto satisface al cliente. Desarrollando este aspecto y cumpliendo con lo que establece el plan de despliegue se podrá garantizar que el tiempo y esfuerzo en esta etapa de desarrollo sea lo más eficiente posible para dar paso al soporte, este es el que garantiza darle mantenimiento al software, pero no todos los proyectos cumplen con lo establecido por el plan, por eso es que en muchos casos hay atrasos y problemas con el software ó el cliente, una de las maneras que existen para mitigar este tipo de problema es recoger el impacto que tuvo el despliegue en los diferentes proyectos, para que así sirva de base a la planificación de futuros proyectos.

Para medir el impacto que tuvo el indicador de pruebas de software en los proyectos se definió la siguiente escala con su correspondiente criterio de decisión.

Valor	Escala	Criterios
1	Muy Satisfactorio	<ul style="list-style-type: none">• Se estableció un Plan de despliegue cumpliendo con el tiempo y esfuerzo estimado.• Se estableció una documentación de usuario donde el cliente pudo entender el funcionamiento del producto.• En caso que firmen un contrato para dar soporte, que haya tenido éxito cumpliendo con todos los requisitos del usuario.• Se estableció la lista de materiales requeridas para el despliegue y se garantizó que todos los componentes del producto estén disponibles y se tengan en cuenta.
2	Satisfactorio	<ul style="list-style-type: none">• Se estableció un Plan de despliegue y se cumplió con el tiempo estimado pero aumentó el esfuerzo en esa etapa para cumplir con el cronograma.• Se estableció una documentación de usuario la cual no estaba en términos claros para el cliente.• En caso que firmen un contrato para dar soporte, que haya tenido éxito cumpliendo con todos los requisitos del usuario.

CAPÍTULO II: PROPUESTA DE INDICADORES

Valor	Escala	Criterios
		<ul style="list-style-type: none"> Se estableció la lista de materiales requeridas para el despliegue y se garantizó que todos los componentes del producto estén disponibles y se tengan en cuenta.
3	Poco Satisfactorio	<ul style="list-style-type: none"> Se estableció un Plan de despliegue pero tuvo impacto negativo porque no cumplió con el tiempo de entrega. Se estableció una documentación de usuario la cual no estaba en términos claros. En caso que firmen un contrato para dar soporte, tuvo éxito pero requirió de más esfuerzo del equipo de desarrollo. Se estableció la lista de materiales requeridas para el despliegue pero no se garantizó que un 80 % de todos los componentes del producto estén disponibles y se tengan en cuenta.
4	No satisfactorio	<ul style="list-style-type: none"> Se estableció un Plan de despliegue pero tuvo impacto negativo porque no cumplió con el tiempo de entrega y aumentó el esfuerzo estimado. Se estableció una documentación de usuario la cual no estaba en términos claros para el cliente. En caso que firmen un contrato para dar soporte, este tuvo éxito pero requirió de más esfuerzo del equipo de desarrollo. Se estableció la lista de materiales pero no se realizó ninguna búsqueda.

Tabla 2.9 Escala de impacto del despliegue de la aplicación.

2.3.4 Indicadores de Estimación

Los indicadores de estimación recogen todos los valores de los proyectos en la gestión del tiempo, esfuerzo y tamaño. La métrica del proyecto explica que estos deben caracterizarse en términos de tamaño, tipo, complejidad y formalidad porque condicionan expectativas sobre las diferentes tendencias a seguir a la hora de medir el producto. El tamaño tradicionalmente se expresaba por la cantidad de líneas de código existentes en el software, luego por la cantidad de casos de uso. La métrica del tamaño argumenta que se debe de considerar el costo, los recursos y la complejidad,

aspectos influyentes en el tamaño y que es un aspecto determinante para calcular el esfuerzo. El esfuerzo es la cantidad de desarrolladores que se necesita para realizar el software en un tiempo determinado, existen muchas métricas que calculan el esfuerzo dependiendo del tamaño del software y los recursos implicados. Dado el esfuerzo y el tamaño se calcula el tiempo necesario para su desarrollo y partir de ahí se empieza planificar ese tiempo en todas las etapas de desarrollo.

2.3.4.1 Indicador Tiempo de desarrollo.

Estimar el tiempo es primordial para llegar a un acuerdo sobre el costo del proyecto. “La Gestión de los Costes del Proyecto incluye los procesos involucrados en la planificación, estimación, preparación de presupuesto y control de costes de forma que el proyecto se pueda completar dentro del presupuesto aprobado“.[4]

Esta tarea no todos los ingenieros de software la realizan con efectividad. El proceso de estimar el tiempo utiliza información sobre el alcance del proyecto, los tipos de recursos necesarios, las cantidades de recursos estimadas y los calendarios de recursos con su disponibilidad. El alcance es la definición del proyecto, los objetivos que deben cumplirse, aborda y documenta las características y los límites del proyecto, y sus productos y servicios relacionados, así como los métodos de aceptación. Las restricciones del alcance se tienen en cuenta al estimar las duraciones de las actividades del cronograma.

Muchos determinan el tiempo estimado a través de la técnica de expertos sin tener en cuenta los detalles que pueden marcar una diferencia sensible entre el nuevo proyecto y sus experiencias anteriores. Actividades como definir y medir los recursos necesarios (personas, metodología y herramientas) para realizar el proyecto y calcular el tamaño aproximado que tendrá el software, son fundamentales para obtener los elementos necesarios con los cuales calcular el tiempo de duración. Además es importante tener en cuenta el impacto de los riesgos porque se evita que el tiempo y esfuerzo estimado no sean erróneos. El tiempo de desarrollo del proyecto se define en dos etapas, uno es el tiempo estimado que es el tiempo según los cálculos debe durar y el otro es el tiempo real consumido en la realización del software.

El tiempo estimado se calcula antes de iniciar el proyecto, los líderes de proyectos se basan en diferentes técnicas de estimación, por ejemplo: técnica por analogía, por Puntos de Casos de Uso, COCOMO, explicadas en el Capítulo 1. Siempre se aconseja calcular el tiempo con más de dos técnicas para comprobar si los resultados han sido correctos. En la UCI generalmente realizan el contrato tres desarrolladores, el ingeniero industrial, el líder del proyecto y el económico, con lo que no es suficiente para calcular el tamaño del producto ya que se requiere de más especialistas

CAPÍTULO II: PROPUESTA DE INDICADORES

dentro del equipo inicial de concepción del proyecto contando con mayor experiencia en la planificación.

El tiempo real es el tiempo que duró el desarrollo del software realmente, en el que se desarrolla todo lo definido en el Proyecto Técnico. Generalmente el tiempo real y el tiempo estimado no coinciden, una de las causas es que no tienen en cuenta los riesgos que pueden afectar el desarrollo del proyecto y que no se cuenta con informaciones históricas de proyectos similares.

Definir solo el tiempo total no cumple todo el objetivo porque el proceso de desarrollo de software se divide en varias etapas de desarrollo con diferentes características por lo tanto todas no ocupan la misma cantidad de tiempo, por eso también es importante recoger el tiempo estimado y real de las etapas de desarrollo. Para adaptar el indicador de tiempo a todas las metodologías permitiendo recoger el tiempo de cada etapa independientemente de cual se utilice en el proyecto, se propone dividir el tiempo de acuerdo a las fases definidas por Pressman en su libro *Un Enfoque Práctico*, Quinta Edición. Las cuales son: Fase de Definición, Fase de desarrollo, y Fase de Mantenimiento.

Fase de definición: Se centra sobre el Qué, es decir que durante la definición los desarrolladores intentan identificar los procesos a automatizar, la planificación de la actividades, que interfaces van a ser establecidas, que restricciones existen, y que criterios de validación necesitan para definir un sistema correcto, de alguna manera tendrán lugar tres tareas principales: ingeniería de sistemas, planificación del proyecto del software y análisis de requisitos.

Fase de desarrollo: Se centra en el Cómo, es decir que se define como ha de diseñarse la estructura de datos, como ha de implementarse la función dentro de una arquitectura de software, los procedimientos, como han de caracterizarse las interfaces, como han de traducirse el diseño en un lenguaje de programación y como han de realizarse las pruebas.

Fase de mantenimiento: El mantenimiento es una de las actividades más comunes en la Ingeniería de Software siendo el proceso de mejora y optimización del software desplegado. Así como la corrección de sus errores. La fase de mantenimiento de software involucra cambios al software en orden de corregir defectos y dependencias encontradas durante su uso tanto como la adición de una nueva funcionalidad para mejorar.

El tiempo dividido en cada una de las etapas planteadas será de gran utilidad para registrarse como material de consulta y referencia de los proyectos de la UCI.

2.3.4.2 Indicador Esfuerzo de desarrollo.

La correcta estimación del esfuerzo que requiere realizar un proyecto de software es un aspecto crucial para el desarrollo de software. Una estimación errónea provoca un costo incierto, desventajoso para la empresa que desarrolla, además se pierde prestigio ante el mercado mundial

CAPÍTULO II: PROPUESTA DE INDICADORES

perdiendo clientes por aplicaciones no deseadas. De ahí la importancia de que las estimaciones se realicen de la forma más eficaz posible.

Para estimar el esfuerzo del personal que trabajará en un proyecto se requiere de grandes requisitos. El personal o el equipo de desarrollo debe tener conocimiento y experiencias en el ámbito en el que van a desarrollar, al igual que en los diferentes recursos de hardwares y software que requiere el sistema.

Existen muchas maneras de calcular el esfuerzo de un proyecto pero ninguna con total éxito. Una es la métrica de tamaño tradicional para estimar el esfuerzo de desarrollo y productividad ha sido LOC (Líneas de código) o SLOC (Líneas de código fuente) .Los principales problemas de utilizar líneas de código como métrica para estimación del esfuerzo es la falta de una definición universal de línea de código, su dependencia con el lenguaje de desarrollo y la dificultad de estimar en fases tempranas del desarrollo la cantidad de líneas que tendrá una aplicación.

El análisis por puntos de función es un método para cuantificar el tamaño y la complejidad de un sistema software en términos de las funciones del usuario que este desarrolla o desarrollará, esta técnica es independiente del lenguaje, herramienta o metodología utilizada en la implementación; por ejemplo, no tienen que considerar lenguajes de programación, sistemas de administración de bases de datos, hardware, o cualquier otra tecnología de procesamiento de datos, es una de sus desventajas porque no mide las características de los diferentes recursos de hardware y de software que puede influir en un mayor esfuerzo y tiempo de desarrollo. Además por mucho que se describa el sistema en las posteriores etapas ocurren cambios que provocarán una nueva estimación del esfuerzo del personal involucrado.

Existen otras de igual problemas para obtener un esfuerzo factible. Para medir el esfuerzo del equipo de desarrollo siempre se requiere de más de un método de estimación. El indicador de esfuerzo propuesto será la base para estimar junto a las técnicas de estimación, el esfuerzo de su equipo de desarrollo ya que recogerá el esfuerzo de los proyectos basado en las diferentes experiencias de proyectos anteriores.

2.3.4.3 Indicador Tamaño del Producto.

Uno de los aspectos que dificultan la labor de los jefes de proyectos en la planificación es la difícil tarea de realizar una estimación de costos y plazos realistas.

El manejador de costo principal para un proyecto de desarrollo de software es sin duda el tamaño del producto. La medida del tamaño debe ser tal que esté en relación directa con el esfuerzo de desarrollo, por lo que las métricas de tamaño tratan de considerar todos los aspectos que influyen en el costo, como tecnología, tipo de recursos y complejidad.

CAPÍTULO II: PROPUESTA DE INDICADORES

Hay diferentes formas de estimar el tamaño, por líneas de código, por puntos de función, o por la complejidad de los procesos, generalmente la UCI estima el tamaño por la complejidad del proceso. Cuando se realiza la ficha técnica del proyecto se define el tamaño del producto pero generalmente con un alto porcentaje de defectos porque estiman una cantidad de procesos de complejidad alta, media y baja y a medida de que avanza el proyecto generalmente se demuestra que esa no era la estimación real por lo que aumenta el esfuerzo y el tiempo en el desarrollo de software.

Debido a esta problemática, se han propuesto diversas soluciones para disminuir la incertidumbre en la estimación del tamaño. Entre otras está contar con una base de datos histórica que permita consultar la cantidad de procesos de este nuevo proyecto con el real y estimado de otros proyectos similares que tengan su información en los datos históricos.

Es necesario conocer también la cantidad de casos de uso del proyecto para tener una referencia más detallada del tamaño en proyectos anteriores. La complejidad que se propone recoger está basada en el método de estimación por Punto de Casos de Uso el cual define a los CU de complejidad alta como los que tengan más de 8 transacciones, los de complejidad media contienen de 4 a 7 transacciones y los de complejidad simple contienen de 1 a 3 transacciones. La cantidad de transacciones se determina a partir de la descripción textual del CU. Entre más detallada esté la descripción textual, más transacciones se pueden encontrar y la estimación será más exacta. Las transacciones están representadas por uno o más pasos del flujo de eventos principal del caso de uso, pudiendo existir más de una transacción dentro del mismo Caso de Uso.

2.3.4.4 Indicador Influencia de los Riesgos en la estimación.

Los riesgos del proyecto amenazan al plan del proyecto, si los riesgos del proyecto se hacen realidad es probable que la planificación temporal del proyecto se retrase y que los costos aumenten. Los riesgos del proyecto identifican los problemas potenciales de presupuesto, planificación temporal, personal, recursos, clientes y requisitos y su impacto en el proyecto de software. El riesgo acompaña a todo cambio, lo más frecuente es que el equipo de software no haga nada, hasta que algo vaya mal, luego este tiene que apresurarse para corregir el problema.

La gestión de riesgos es la práctica compuesta de métodos y herramientas que posibilita la gestión de los riesgos en un proyecto y que provee un entorno disciplinado para la toma de decisiones en base a determinar constantemente que puede ir mal, identificando cuales son los riesgos más importantes en los cuales enfocarse e implementar estrategias para gestionarlos. La influencia de los riesgos en la estimación de cada proyecto es necesario recogerla para llegar a conclusiones del impacto e influencia de los riesgos en la producción de la UCI.

Para medir el impacto que tuvieron los riesgos en los diferentes proyectos se definió la siguiente escala con su correspondiente criterio de decisión.

CAPÍTULO II: PROPUESTA DE INDICADORES

Valor	Escala	Criterios
1	Muy Satisfactorio	<ul style="list-style-type: none"> • El riesgo planificado tuvo impacto satisfactorio en el desarrollo del proyecto. • La estrategia de mitigación para el riesgo funcionó perfectamente.
2	Satisfactorio	<ul style="list-style-type: none"> • El riesgo impactó el proyecto pero se logró mitigar según la estrategia trazada y no tuvo influencia en el tiempo ni el esfuerzo necesario para el desarrollo. • Los errores provocaron impactos mínimos en el esfuerzo y/o planificación temporal. • El proyecto tuvo un aumento de tiempo insignificante. • El proyecto tuvo una degradación de la calidad apenas perceptible.
3	Poco Satisfactorio	<ul style="list-style-type: none"> • El riesgo impactó el proyecto y no se pudieron mitigar completamente sus consecuencias influyendo en el tiempo y el esfuerzo necesario para el desarrollo. • El riesgo influyó en posibles retrasos en el plazo de entrega del producto.
4	No satisfactorio	<ul style="list-style-type: none"> • El riesgo no se logró mitigar en el proyecto, atrasó el tiempo y el esfuerzo necesario para el desarrollo. • El proyecto obtuvo resultados no satisfactorios debido al impacto de los riesgos aumentando los costos y retrasos de la planificación temporal.

Tabla 2.10 Escala de impacto del indicador Influencia de los riesgos.

2.3.4.5 Indicador Cantidad de Versiones.

Para facilitar la gestión, los líderes de proyectos o la organización pueden dividir los proyectos en pequeñas entregas para un mejor desarrollo, con los enlaces correspondientes a las operaciones de la organización. El conjunto de estas entregas contiene un ciclo de vida del proyecto.

La cantidad de versiones a realizar en un proyecto es proporcional a la magnitud del proyecto y el tipo de proyecto. Cada versión debe ser controlada y se debe abordar una parte de la funcionalidad total, trabajando en todos los flujos de trabajo relevantes, refinando la arquitectura y ejecutando las actividades de aseguramiento de la calidad. Cada versión se analiza cuando termina. Se puede

CAPÍTULO II: PROPUESTA DE INDICADORES

determinar si han aparecido nuevos requerimientos o han cambiado los existentes. Este es un indicador cuantitativo que permite recoger la cantidad de versiones entregadas de distintos proyectos de la UCI tomándose de referencia para proyectos similares

2.3.5 Indicadores de Recursos.

Los indicadores de recursos recogen los recursos necesarios en un proyecto, ya sean humanos, herramientas de software o hardware o recursos de reutilización. Una de las tareas fundamentales que tienen los líderes de proyectos es la estimación de recursos para estimar el esfuerzo y el tiempo del desarrollo de software, por lo tanto recoger los datos de todos los recursos utilizados en proyectos durante el desarrollo de software es muy importante. La métrica de recursos explica que estos miden los elementos que incluyen a personas (experiencia, habilidades, coste y funcionamiento), los métodos y las herramientas (en términos de efecto sobre productividad y calidad, coste), tiempo, esfuerzo, y presupuesto (recursos consumidos, recursos restantes) y se miden a través del impacto que tuvieron esas herramientas en el desarrollo de software

"El proceso de estimar las duraciones de las actividades del cronograma utiliza información sobre el alcance del trabajo de la actividad del cronograma, los tipos de recursos necesarios, las cantidades de recursos estimadas y los calendarios de recursos con su disponibilidad." [4]

Por lo tanto tener datos históricos de los tipos de recursos utilizados en los proyectos es importante porque define a la hora de estimar el tiempo de desarrollo conocer los resultados de los proyectos que utilizaron determinados recursos. El PMBOK recomienda que antes de definir soluciones, se debe consultar una base histórica para que el líder del proyecto pueda tomar alguna decisión en cuanto a los recursos. Si estuviera disponible, se revisa la información histórica relacionada con los tipos de recursos que fueron necesarios para un trabajo similar en proyectos anteriores.

Recoger el impacto que tuvieron los diferentes recursos ayudará a seleccionar con más precisión los recursos de los futuros proyectos de la UCI.

2.3.5.1 Indicador Metodología de Desarrollo de Software.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Durante el proceso de desarrollo de software es necesario llevar un estricto control de lo que se hace y más importante aún definir los pasos, procedimientos y técnicas a seguir para la concepción de un producto.

Todo desarrollo de software posee riesgos y si no se utiliza una metodología con herramientas para controlar las diferentes actividades que permita controlar a los desarrolladores, que guíe al proyecto en el cumplimiento de los objetivos y el contrato establecido, como resultado se obtendrán a

CAPÍTULO II: PROPUESTA DE INDICADORES

clientes y desarrolladores insatisfechos, por lo que las metodologías aparecen por la necesidad de organizar y guiar al proceso de construcción del software.

Es recomendable elegir la metodología adecuada para tener éxito en el proyecto porque si se elige una metodología sencilla para desarrollar un proyecto que tenga procesos complejos con el objetivo de concluir con más rapidez el software, el resultado que se obtendrá es que el software sea deficiente implicando más tiempo de desarrollo. Si se elige una adecuada metodología el proyecto obtendrá buenos resultados. El líder de proyecto deberá escoger una metodología que se ajuste a los requerimientos de los servicios demandados por el cliente.

Para medir el impacto tuvo la metodología de desarrollo en los diferentes proyectos se definió la siguiente escala con su correspondiente criterio de decisión.

Valor	Escala	Criterios
1	Muy Satisfactorio	<ul style="list-style-type: none">• La metodología de desarrollo seleccionada fue la más adecuada para su proyecto, dando como resultado:• El cliente quedó satisfecho con su producto cumpliendo con la calidad requerida.• Se cumplió con el cronograma de trabajo.• Se seleccionó la cantidad de integrantes adecuada para el desarrollo de software.• Se cumplió con el esfuerzo estimado.• Se documentaron todos los artefactos que exige la metodología utilizada en el tiempo establecido.
2	Satisfactorio	<ul style="list-style-type: none">• El cliente quedó satisfecho con su producto.• Se cumplió con el cronograma de trabajo
3	Poco Satisfactorio	<ul style="list-style-type: none">• El cliente no quedó totalmente satisfecho con su producto.• El producto no cumplía con todos los requisitos solicitados por el cliente.
4	No satisfactorio	<ul style="list-style-type: none">• El cliente no quedó satisfecho con su producto incumpliendo con la calidad del producto.• No se cumplió con el cronograma de trabajo debido a problemas surgidos a causa de la metodología.• La metodología no cumple con los requisitos de adquisición de documentación requerida por el cliente.

CAPÍTULO II: PROPUESTA DE INDICADORES

Valor	Escala	Criterios
		<ul style="list-style-type: none">• Los artefactos de los desarrolladores no fueron suficientes para guiar el proceso de desarrollo.• La metodología no permitía un desarrollo equilibrado por parte de los integrantes del equipo.

Tabla 2.11 Escala de impacto del indicador Metodología de Desarrollo de Software.

2.3.5.2 Indicador Estilo Arquitectónico.

"La arquitectura de software es la organización fundamental de un sistema distribuido en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución." [20]

Una arquitectura seleccionada de acuerdo a las características de la aplicación a desarrollar escogida como mejor variante entre todas las existentes, disminuye el tiempo de desarrollo del software ya que establece los fundamentos para que analistas, diseñadores, programadores, y otros roles, trabajen en una línea común que permita alcanzar los objetivos cubriendo todas las necesidades del sistema. Además ayuda a la planificación de recursos y la asignación de tareas, debido a que el trabajo de desarrollo puede ser particionado a través de los subsistemas y los esfuerzos de desarrollo individual pueden proceder en paralelo. Esto demuestra que influye en el tiempo de desarrollo establecido, una arquitectura compleja y difícil de operar sobre ella atrasa el desarrollo del proyecto.

Al diseñar una arquitectura de software debemos crear y representar componentes que interactúen entre ellos y tengan asignadas tareas específicas, además de organizarlos de forma tal que se logren los requerimientos establecidos. Podemos partir con soluciones ya probadas, con la intención de no comenzar sin una base de propuestas y utilizar modelos que han funcionado en proyectos similares. Estas soluciones probadas se conocen como estilos arquitectónicos, se definen como una familia de sistemas de software en términos de un patrón de organización estructural, que define un vocabulario de componentes y tipos de conectores y un conjunto de restricciones de cómo pueden ser combinadas. Los estilos expresan componentes y las relaciones entre éstos, con las restricciones de su aplicación y la composición asociada, así como también las reglas para su construcción. Así mismo, se considera como un tipo particular de estructura fundamental para un sistema de software, conjuntamente con un método asociado que especifica cómo construirlo. Estos estilos utilizan una serie de patrones arquitectónicos que son los que especifican como desarrollar el estilo.

2.3.5.3 Indicador Patrón de Arquitectura.

Los patrones expresan esquemas de organización estructural fundamentales para los sistemas de software, estos se ocupan de cuestiones que están más cerca del diseño, la práctica, la implementación, el proceso y el refinamiento de código. Los patrones arquitectónicos, se han materializado con referencia a lenguajes y paradigmas también específicos de desarrollo. Debido a la importancia que requiere la utilización de un patrón de arquitectura adecuado se propone definirlo como indicador para que señalen a los líderes de proyectos y arquitectos cual es la mejor solución que se adapte a su proyecto, consultando información referida al éxito o fracaso del uso de un determinado patrón de arquitectura en un proyecto.

Como se ha demostrado es de gran utilidad recoger el valor de los indicadores de patrón y estilo de arquitectura pues proporcionan una medida de la utilización y resultados de estos y su impacto en los proyectos de la UCI. Para medir el impacto que tuvo el patrón de arquitectura y los estilos utilizados en los diferentes proyectos se definió la siguiente escala con su correspondiente criterio de decisión.

Valor	Escala	Criterios
1	Muy Satisfactorio	<ul style="list-style-type: none"> • El patrón de arquitectura contribuyó a dar flexibilidad y extensibilidad a la aplicación. • Permitió la posibilidad de adaptación al cambio de los requerimientos. • Facilitó la comunicación entre todos los integrantes del equipo de desarrollo siguiendo una línea base común. • Definió correctamente el esquema organizativo estructural de sistema. • Permitió la reutilización y la evolución del sistema con facilidad. • La arquitectura estuvo basada y le daba cumplimiento a todos los requisitos no funcionales del sistema. • El usuario confió en el buen funcionamiento del sistema con la arquitectura desarrollada. • El sistema cumplió con una posible interoperabilidad, pudiendo interactuar con otras aplicaciones externas.
2	Satisfactorio	<ul style="list-style-type: none"> • Definió el esquema organizativo estructural de sistema. • Facilitó la comunicación entre todos los integrantes del

CAPÍTULO II: PROPUESTA DE INDICADORES

Valor	Escala	Criterios
		equipo de desarrollo siguiendo una línea base común.
3	Poco Satisfactorio	<ul style="list-style-type: none"> • No logró definir totalmente el esquema organizativo estructural de sistema. • El patrón de arquitectura no contribuyó a dar flexibilidad y extensibilidad suficiente a la aplicación. • Algunos requerimientos no se podían adaptar a la arquitectura.
4	No satisfactorio	<ul style="list-style-type: none"> • El patrón de arquitectura no contribuyó a dar flexibilidad y extensibilidad a la aplicación. • No facilitó la comunicación entre los integrantes del equipo de desarrollo impidiendo seguir una línea base común. • No permitió la posibilidad de adaptación al cambio de los requerimientos.

Tabla 2.12 Escala de impacto del indicador patrón y estilos de arquitectura.

2.3.5.4 Indicador Lenguaje de Programación.

Los lenguajes de programación son claves para la creación de programas y la construcción del software en el proceso de desarrollo del proyecto. El lenguaje de programación utilizado en los proyectos es un aspecto crucial para el éxito y avance del mismo. Los programadores pueden tener un conjunto común de instrucciones que puedan ser comprendidas para realizar la construcción del programa al mismo tiempo, es decir trabajando de forma paralela. Es necesario recoger el valor de este indicador en cada proyecto productivo pues proporciona una medida de la utilización y resultado de los lenguajes de programación y su impacto en los proyectos de la UCI.

Para medir el impacto que tuvo el lenguaje de programación en los diferentes proyectos se definió la siguiente escala con su correspondiente criterio de decisión.

Valor	Escala	Criterios
1	Muy Satisfactorio	<ul style="list-style-type: none"> • El desarrollo del proyecto no fue atrasado en ningún momento por problemas con el lenguaje de programación. • El esfuerzo estimado para la etapa donde se puso en práctica el lenguaje de programación fue suficiente durante el desarrollo. • El lenguaje resultó intuitivo y fácil de aprender con un

CAPÍTULO II: PROPUESTA DE INDICADORES

Valor	Escala	Criterios
		<p>código legible y seguro.</p> <ul style="list-style-type: none"> • El lenguaje resultó compatible con las tecnologías utilizadas por el equipo de desarrollo para la programación. • El lenguaje permitió la comunicación del equipo de desarrollo. • El lenguaje permitió un buen mantenimiento y prueba de los algoritmos. • El lenguaje permitió una optimización temprana de código.
2	Satisfactorio	<ul style="list-style-type: none"> • El desarrollo del proyecto no fue atrasado en ningún momento por problemas con el lenguaje de programación. • El lenguaje resultó intuitivo y fácil de aprender con un código legible y seguro.
3	Poco Satisfactorio	<ul style="list-style-type: none"> • El desarrollo del proyecto fue atrasado por problemas surgidos con el lenguaje de programación debido a la falta de experiencia de los integrantes del equipo con el lenguaje. • El lenguaje no permitió totalmente la comunicación del equipo de desarrollo. • El lenguaje resultó muy difícil de aprender por parte de los desarrolladores.
4	No satisfactorio	<ul style="list-style-type: none"> • El desarrollo del proyecto fue atrasado por problemas surgidos con el lenguaje de programación debido a la falta de experiencia de los integrantes del equipo con el lenguaje. • El lenguaje no permitió un buen mantenimiento y prueba de los algoritmos. • El lenguaje resultó muy difícil de aprender por parte de los desarrolladores.

Tabla 2.13 Escala de impacto del indicador lenguaje de programación.

CAPÍTULO II: PROPUESTA DE INDICADORES

2.3.5.5 Indicador Componentes reutilizables utilizados.

La reutilización de software es el uso de cualquier tipo de artefacto o parte del mismo, creado con anterioridad en un proyecto.

La reutilización de software es una práctica común en las organizaciones dedicadas al desarrollo de aplicaciones informáticas. Su propósito es la reutilización parcial de las aplicaciones ya existentes en el desarrollo de nuevos sistemas. La reutilización de software es un método efectivo para la reducción de tiempo y esfuerzo en los proyectos además de utilizar la experiencia adquirida en el pasado por las organizaciones. La reutilización del software no cubre sólo la utilización de códigos, abarca diferentes posibilidades de uso de metodología, ciclos de vida, planes del proyecto, especificaciones de requisitos, diseños, arquitectura de software, planes de validación, juegos de prueba y documentación. Será de gran ayuda tener registrado que componentes reutilizables se utilizó en cada proyecto para conocer la influencia de esta reutilización en el tiempo y el esfuerzo del equipo de desarrollo.

Para medir el impacto que tuvo el uso de componentes reutilizables en los diferentes proyectos se definió la siguiente escala con su correspondiente criterio.

Valor	Escala	Criterios
1	Muy Satisfactorio	<ul style="list-style-type: none">• La reutilización proporcionó un desarrollo más rápido, mayor calidad del software y ahorro de esfuerzo y tiempo.• Disminuyó considerablemente la carga de trabajo de los integrantes del equipo.• Proporcionó mayor eficiencia y fiabilidad al utilizar componentes que ya han sido probados.• Redujo el riesgo del fracaso sin la utilización de nuevos procedimientos.• La reutilización permitió concentrarse en aspectos realmente únicos del proyecto.• Permitted una integración de los diversos módulos más fáciles.
2	Satisfactorio	<ul style="list-style-type: none">• La reutilización proporcionó un desarrollo más rápido.• Redujo el riesgo del fracaso sin la utilización de nuevos procedimientos.

CAPÍTULO II: PROPUESTA DE INDICADORES

Valor	Escala	Criterios
3	Poco Satisfactorio	<ul style="list-style-type: none">• No redujo completamente el riesgo.• La complejidad de los algoritmos y procedimientos aumentó.
4	No satisfactorio	<ul style="list-style-type: none">• No proporcionó eficiencia porque los componentes reutilizados no se acoplaban correctamente donde se querían utilizar.• Impidió una integración de los diversos módulos del proyecto.

Tabla 2.14 Escala de impacto del indicador componentes reutilizables utilizados.

2.3.5.6 Indicador Herramienta de Modelado Visual.

Las herramientas de modelado visual permiten la comunicación con el usuario de una manera enfocada, sin distraer el equipo de desarrollo con asuntos y características ajenas al sistema. Además si los desarrolladores no entendieron lo que deseaba el cliente o el cliente desea otro requisito a parte de los que pidió anteriormente se pueden hacer cambios en el modelo o hacer uno nuevo de ser necesario. Este indicador permitirá conocer las herramientas más utilizadas y de mejores resultados en los proyectos de la UCI.

2.3.5.7 Indicador Herramienta de Programación.

La plataforma de programación no es más que la herramienta en que van a escribir el código del software el equipo de desarrollo o los implementadores. Esta herramienta incide en el desarrollo de software por sus diferentes características, reflejando ser un indicador cualitativo pues a través de este se puede definir el tiempo que puede durar el proyecto utilizando una plataforma u otra en dependencia de la complejidad del lenguaje y del sistema. Todas las herramientas poseen sus ventajas y desventajas, algunas son más fáciles de utilizar que otras y el esfuerzo para trabajar en ellas varía de una herramienta a otra.

2.3.5.8 Indicador Herramienta de Sistema Gestor de Base de Datos.

Un Sistema Gestor de Bases de Datos (SGBD) es el programa que permite almacenar un conjunto de datos relacionados entre sí y un grupo de programas para tener acceso a esos datos, permitiendo concurrencia y recuperación. La persistencia de un SGBD hace referencia a la conservación de los datos después de la finalización del proceso que los creó y la concurrencia se refiere a la capacidad del sistema para gestionar a múltiples usuarios interactuando al mismo tiempo. Las herramientas utilizadas para administrar la base de datos en los proyectos es

importante registrarlas pues definen el éxito que tuvo la herramienta en cada proyecto, la complejidad de la herramienta que puede influir en el esfuerzo del equipo de desarrollo y en general conocer si realmente fue satisfactorio su uso.

2.3.5.9 Indicador Herramienta para la Planificación.

La herramienta utilizada para la planificación de un proyecto es esencial para obtener éxito, por lo que se debe seleccionar la herramienta adecuada según las características del proyecto. Estas herramientas apoyan el proceso de estimación de una manera más eficaz ya que controla todo el proceso de desarrollo, el cumplimiento de las actividades y tareas, además el líder de proyecto puede conocer el avance del mismo. Mediante este indicador es posible conocer el tiempo de duración de los proyectos que la utilizaron y los beneficios que trajo utilizar esta herramienta, si fue eficiente su uso o no, ya que el éxito no solo depende del buen empleo de estas herramientas pero si es posible conocer si apoyó realmente la planificación de los proyectos.

2.3.5.10 Indicador Herramienta para la Gestión de Configuración.

La Gestión de Configuración del Software es una actividad de autoprotección que se aplica durante el proceso del software y que constituye "...una parte esencial de una buena gestión de proyecto y una práctica formal de la Industria del Software". [8]

El éxito en el desarrollo de un proyecto depende en gran parte de la utilización de las herramientas de configuración ya que brindan un soporte al proyecto para controlar los cambios y las versiones. La selección de estas herramientas se hace basándose en las características propias del proyecto y de la metodología de desarrollo de software empleada, pero sería de gran apoyo tener registrado los resultados del uso de cada herramienta en los proyectos de la UCI, para que los líderes de proyectos cuando seleccionen un recurso para utilizarlo en su proyecto tengan un aspecto más por el cual basarse y decidir la más apropiada según los resultados en las experiencias anteriores de los proyectos.

2.3.5.11 Indicador Herramientas utilizadas en la estimación.

Unos de los objetivos de la planificación es obtener estimaciones del esfuerzo humano requerido y de la duración del proyecto y del costo. En la mayoría de los casos las estimaciones se hacen valiéndose de la experiencia pasada como única guía. Aunque en algunos casos puede que la experiencia no sea suficiente por eso siempre es recomendable utilizar diferentes métodos de estimación que calculen el tiempo y esfuerzo. Es una forma de resolución de problemas cuando en la mayoría de los casos, el problema a resolver es demasiado complejo para considerarlo como una sola parte. Por esta razón, se descompone el problema, caracterizándolo como un conjunto de pequeños problemas. Este indicador permite medir el impacto que tuvieron las herramientas

CAPÍTULO II: PROPUESTA DE INDICADORES

utilizadas para estimar ya sean automatizadas o los propios métodos de estimación brindándoles a los líderes de proyectos experiencias anteriores.

Todas las herramientas utilizadas en los proyectos son indicadores para medir el impacto de las mismas y permiten conocer si utilizando las herramientas definidas al inicio del proyecto les resultó satisfactorio o no. La siguiente es la escala para medir el impacto de las diferentes herramientas en el proyecto:

Valor	Escala	Criterios
1	Muy Satisfactorio	<ul style="list-style-type: none"> • La persona encargada de utilizar la herramienta o el equipo encargado no tuvieron problemas con la herramienta utilizada por lo tanto se cumplió con el esfuerzo estimado. • La utilización de la herramienta ayudó a que se cumpliera con el tiempo estimado.
2	Satisfactorio	<ul style="list-style-type: none"> • El equipo de desarrollo nunca había trabajado con la herramienta y el plan de gestión de proyecto estableció la capacitación incluyendo un esfuerzo adicional por el equipo y el resultado fue satisfactorio. • La utilización de la herramienta ayudó a que se cumpliera con el tiempo estimado.
3	Poco Satisfactorio	<ul style="list-style-type: none"> • La persona encargada de utilizarla o el equipo tuvieron problemas con la herramienta utilizada provocando un mayor esfuerzo de lo estimado porque no estuvo planificado la capacitación del equipo de desarrollo. • La utilización de la herramienta ayudó a que se cumpliera con el tiempo estimado.
4	No satisfactorio	<ul style="list-style-type: none"> • La utilización de la herramienta no ayudó a que se cumpliera con el tiempo estimado. • No utilizaron algunas herramientas de las antes explicadas.

Tabla 2.15 Escala de impacto del indicador herramientas utilizadas.

2.3.5.12 Indicador Cantidad de PC.

Las computadoras son recursos imprescindibles para el desarrollo de sistemas informáticos. El avance del proyecto productivo depende en gran medida de la utilización y condiciones de los recursos de hardware. Los recursos materiales con que cuenta un proyecto productivo, entre ellos las máquinas computadoras son el soporte de hardware esencial para el desarrollo de sistemas informáticos. El indicador cantidad de PC es un indicador cuantitativo que te permite conocer la cantidad de máquinas utilizadas por el proyecto.

2.3.5.13 Indicador Cantidad de Integrantes del equipo de desarrollo.

Las personas como recursos humanos dentro del desarrollo de proyectos de software son un eslabón principal en su desarrollo. Cada proyecto necesita una determinada cantidad de desarrolladores vinculados, si hay involucrados menos que los necesarios muchos de los integrantes tendrán excesiva carga de trabajo, si por el contrario hay muchas más personas entonces se reflejará desorganización y no se podrá controlar el trabajo de cada integrante de la mejor manera. Este indicador cuantitativo permite recoger la cantidad de integrantes de un equipo de desarrollo para que líderes de proyectos con similares cifras conozcan el esfuerzo de desarrollo y el tiempo empleado en la culminación de un proyecto utilizando una determinada cantidad de integrantes.

2.3.5.14 Indicador Nivel de conocimiento de los integrantes del equipo.

El nivel de conocimiento de cada integrante del equipo de desarrollo distribuido por roles es importante conocerlo para reflejar la calidad de la fuerza de trabajo con que contaba el equipo que desarrolló un determinado proyecto. También es necesario conocer los roles que se asignaron y la cantidad de estos, ya que los roles y las responsabilidades definen los cargos, las habilidades y las competencias que requiere el proyecto. Es responsabilidad del líder de proyecto mejorar continuamente las habilidades de los miembros del equipo a fin de aumentar su capacidad de completar las actividades del proyecto. El tipo y la cantidad de miembros del equipo del proyecto a menudo pueden cambiar a medida que avanza el proyecto.

Los niveles de competencia reales de los miembros del equipo pueden hacer que las duraciones de las actividades y el cronograma varíen. Es importante que los líderes de proyecto conozcan realmente el rendimiento de su equipo. El PMBOK recomienda hacer una evaluación del rendimiento del equipo periódicamente y afirma que la planificación de los Recursos Humanos determina los roles del proyecto, los cuales pueden designarse para personas o grupos. Esas personas o grupos pueden estar dentro o fuera de la organización que lleva a cabo el proyecto, lo importante es conocer su experiencia en el tema que se desarrolla.

CAPÍTULO II: PROPUESTA DE INDICADORES

Para medir el impacto que ha tenido el nivel de conocimiento de los integrantes de un equipo de desarrollo en los diferentes proyectos se definió la siguiente escala con su correspondiente criterio.

Valor	Escala	Criterios
1	Muy Satisfactorio	<ul style="list-style-type: none">• El nivel de conocimiento de los desarrolladores de un determinado rol era muy alto.• Se desarrollaron cursos de capacitación a los integrantes del equipo con atención diferenciada por roles.
2	Satisfactorio	<ul style="list-style-type: none">• El nivel de conocimiento de los desarrolladores de un determinado rol era alto.
3	Poco Satisfactorio	<ul style="list-style-type: none">• El nivel de conocimiento de los desarrolladores de un determinado rol era medio.
4	No satisfactorio	<ul style="list-style-type: none">• El nivel de conocimiento de los desarrolladores de un determinado rol era bajo.• No se planificó capacitación para el equipo de desarrollo.

Tabla 2.16 Escala de impacto del indicador nivel de conocimiento de los integrantes del equipo.

2.4 Propuesta de Análisis Estadístico de los resultados.

La realización de los estudios de los indicadores implica finalmente emitir unos resultados cuantificables de dicho estudio. La claridad de dicha presentación es de vital importancia para la comprensión de los resultados y la interpretación de los mismos. La mejor manera de comprenderlos es haciendo un análisis estadístico. El análisis estadístico es una rama de la matemática que se refiere a la recolección, estudio e interpretación de los datos obtenidos en un estudio, para representar los primeros resultados de un análisis estadístico de un modo adecuado, se aconseja que se haga de manera descriptiva. La estadística descriptiva se dedica a los métodos de recolección, descripción, visualización y resumen de datos originados a partir de los fenómenos en estudios, tales datos son variables a observar, como variables cualitativas que se definen expresándose en distintas cualidades, características o modalidades, las variables cualitativas se clasifican en ordinales y nominales, las ordinales toman distintos valores ordenados siguiendo una escala y las nominales son los aspectos observados que no pueden ser sometidos a un criterio de orden. Además se clasifican en variables cuantitativas que se expresan mediante cantidades numéricas.

CAPÍTULO II: PROPUESTA DE INDICADORES

En esta investigación se propondrá un primer análisis estadístico basado en los resultados de los indicadores a través de representación gráfica, brindando un resultado para la mejora continúa de los proyectos a través del Principio de Pareto.

El principio de Pareto afirma que en todo grupo de elementos o factores que contribuyen a un mismo efecto, unos pocos son responsables de la mayor parte de dicho efecto. Pareto plantea que un el 20% producirá el 80% de los efectos, mientras que el 80% restante sólo cuenta para el 20% de los efectos.

El análisis de Pareto es una comparación cuantitativa y ordenada de elementos o factores según su contribución a un determinado efecto, sus objetivos es clasificar en dos categorías: Factores Críticos (los elementos más críticos) y los "Muchos Triviales" (los elementos poco importantes en ella).

Para realizar el análisis estadístico es necesario recoger datos para obtener los resultados, lo que se pretende es obtener los problemas más frecuentes de los proyectos de la UCI, que influyen en la estimación, para así contribuir a una mejora. La Fundación Iberoamericana de la Gestión de Calidad hace referencia de este método, reflejando su gran importancia para la mejora del desarrollo de software.

Las características principales son: [21]

- Priorización: Identifica los elementos que más peso o importancia tienen dentro de un grupo.
- Unificación de Criterios: Enfoca y dirige el esfuerzo de los componentes del grupo de trabajo hacia un objetivo prioritario común.
- Carácter objetivo: Su utilización fuerza al grupo de trabajo a tomar decisiones basadas en datos y hechos objetivos y no en ideas subjetivas.

Para representar el principio de Pareto se hace el diagrama que no es más que la representación gráfica de la Tabla de Pareto correspondiente.

Las características de los diagramas de Pareto son:[21]

- Simplicidad: Tanto la Tabla como el Diagrama de Pareto no requieren ni cálculos complejos ni técnicas sofisticadas de representación gráfica.
- Impacto visual: El Diagrama de Pareto comunica de forma clara, evidente y de un "vistazo", el resultado del análisis de comparación y priorización.

Para la realización del diagrama de Pareto se efectúan los siguientes pasos:

Paso 1: Preparación de los datos.

Como en todas las herramientas de análisis de datos, el primer paso consiste en recoger los datos correctos o asegurarse de que los existentes lo son.

CAPÍTULO II: PROPUESTA DE INDICADORES

En nuestro caso la población sería la suma de todos los proyectos de un mismo tipo.

Y como muestra serían los factores poco satisfactorios y no satisfactorios.

Paso 2: Cálculo de las contribuciones parciales y totales. Ordenación de los elementos o factores incluidos en el análisis

- Para cada elemento contribuyente sobre el efecto, anotar su magnitud.
- La magnitud sería la sumatoria de los factores poco probables y lo no probables (MC).
- Ordenar dichos elementos de mayor a menor, según la magnitud de su contribución.
- Calcular la magnitud acumulada (MET).

Paso 3: Calcular el porcentaje y el porcentaje acumulado, para cada el elemento de la lista ordenada. El porcentaje de la contribución de cada elemento se calcula:

$$\% = (\text{magnitud de la contribución (MC)} / \text{magnitud del efecto total (MET)}) \times 100$$

El porcentaje acumulado para cada elemento de la lista ordenada se calcula:

- Por suma de contribuciones de cada uno de los elementos anteriores en la tabla, más el elemento en cuestión como magnitud de la contribución, y aplicando la fórmula anterior.
- Por la suma de porcentajes de contribución de cada uno de los elementos anteriores más el porcentaje del elemento en cuestión. En este caso habrá que tener en cuenta el que estos porcentajes, en general, han sido redondeados.

Una vez completado este paso tenemos construida la Tabla de Pareto.

Paso 4: Trazar y rotular los ejes del Diagrama.

- El eje vertical izquierdo representa la magnitud del efecto estudiado.
- El eje horizontal contiene los distintos elementos o factores que contribuyen al efecto.

Paso 5: Dibujar un Gráfico de Barras que represente el efecto de cada uno de los elementos contribuyentes.

La altura de cada barra es igual a la contribución de cada elemento tanto medida en magnitud por medio del eje vertical izquierdo, como en porcentaje por medio del eje vertical derecho.

Paso 6: Trazar un Gráfico Lineal cuyos puntos representan el porcentaje acumulado de la Tabla de Pareto.

Marcar los puntos del gráfico en la intersección de la prolongación del límite derecho de cada barra con la magnitud del porcentaje acumulado correspondiente al elemento representado en dicha barra.

Paso 7: Señalar los elementos "Factores Críticos" y los "Muchos Triviales"

- Trazar una línea vertical que separa el Diagrama en dos partes y sirve para visualizar la frontera entre los "Factores Críticos" y los "Muchos Triviales", basándose en el cambio de

inclinación entre los segmentos lineales correspondientes a cada elemento hasta llegar al valor más aproximado a 80 %.

- Rotular las dos secciones del Diagrama.
- Rotular el porcentaje acumulado del efecto correspondiente al último elemento incluido en la sección " Factores Críticos".

Como resultado de este análisis se obtendrán los indicadores que afectan a la estimación durante el desarrollo de software, así se podrán enfocar y dirigir las acciones a esos indicadores.

Si el líder de proyecto desea conocer qué valores de esos indicadores afectan directamente a las estimaciones desarrolla el mismo procedimiento con la diferencia que la población sería el conjunto de valores de esos indicadores que se utilizan en el tipo de proyecto determinado y la muestra sería los factores poco probables y no probables.

Utilización en las fases del proceso de solución de problemas:

Este análisis es aplicable en todos los casos en que se deban establecer prioridades para no dispersar el esfuerzo y optimizar el resultado de dicha inversión.

En particular:

- Para asignar prioridades a los problemas durante la definición y selección de proyectos.
- Para identificar las causas claves de un problema.
- Para comprobar los resultados de un grupo de trabajo una vez implantada la solución propuesta por el mismo.

A este fin se compara el Diagrama de Pareto de la situación inicial con la actual y se comprueba que la contribución de los elementos inicialmente más importantes haya disminuido notablemente.

2.4.1 Ejemplo de cómo utilizar el Principio de Pareto.

Datos iniciales:

La población sería: 20 Proyectos de Gestión.

Problema: Afectación en la estimación de tiempo y esfuerzo.

Causa: Los Factores poco probables y no probables.

Paso 1: Preparación de los datos

CAPÍTULO II: PROPUESTA DE INDICADORES

No	Indicadores
1	Influencia de los Riesgos en la estimación
2	Nivel de conocimiento de los integrantes del equipo
3	Gestión de Calidad
4	Metodología de desarrollo de software
5	Patrón de Arquitectura.
6	Estilo Arquitectónico
7	Herramientas utilizadas en la Planificación.
8	Herramientas utilizadas en la Gestión de Configuración.
9	Despliegue de la aplicación
10	Pruebas de Software
11	Lenguaje de Programación
12	Entorno de Proyecto
13	Sistema Gestor Base de Datos
14	Técnica usada en estimación
15	Cantidad de PC.
16	Herramienta de Programación
17	Herramienta de modelado visual

Tabla 2.17 Factores de Errores.

Paso 2: Cálculo de las contribuciones parciales y totales. Ordenación de los elementos o factores incluidos en el análisis.

- **FE:** Factor de Error.
- **CFPP:** Cantidad de Factores pocos probables.
- **CFNP:** Cantidad de Factores no probables.

No	FE	CFPP	CFNP	Total
1	Influencia de los Riesgos en la estimación	8	10	18
2	Nivel de conocimiento de los integrantes del equipo	5	10	15
3	Calidad del	6	4	10

CAPÍTULO II: PROPUESTA DE INDICADORES

No	FE	CFPP	CFNP	Total
	Proyecto			
4	Metodología de desarrollo de software	5	3	8
5	Patrón de Arquitectura.	2	5	7
6	Estilo Arquitectónico	2	4	6
7	Herramientas utilizadas en la Planificación.	2	3	5
8	Herramientas utilizadas en la Gestión de Configuración.	1	3	4
9	Despliegue de la aplicación	0	3	3
10	Prueba de Software	0	2	2
11	Lenguaje de Programación	0	2	2
12	Entorno de Proyecto	0	0	0
13	Sistema Gestor Base de Datos	0	0	0
14	Técnica usada en estimación	0	0	0
15	Herramienta de Programación	0	0	0
16	Herramienta de modelado visual	0	0	0
	Total	31	49	80

Tabla 2.18 Frecuencia acumulada de los Factores de errores.

Paso 3: Calcular el porcentaje y el porcentaje acumulado, para cada elemento de la lista ordenada.

CAPÍTULO II: PROPUESTA DE INDICADORES

No	Indicadores	Total Factor de error	Total de errores acumulados	% del total	% del total Acumulado.
1	Influencia de los Riesgos en la estimación	18	18	22.5	22.5
2	Nivel de conocimiento de los integrantes del equipo	15	33	18.75	41.25
3	Gestión de Calidad	10	43	12.5	53.75
4	Metodología de desarrollo de software	8	51	10	63.75
5	Patrón de Arquitectura.	7	58	8.75	72.5
6	Estilo Arquitectónico	6	64	7.5	80
7	Herramientas utilizadas en la Planificación.	5	69	6.25	86.25
8	Herramientas utilizadas en la Gestión de Configuración.	4	73	5	91.25
9	Despliegue de la aplicación	3	76	3.75	95
10	Prueba de Software	2	78	2.5	97.5
11	Lenguaje de Programación	2	80	2.5	100
12	Entorno de Proyecto	0	80	0	100
13	Sistema Gestor Base de Datos	0	80	0	100
14	Técnica usada en	0	80	0	100

CAPÍTULO II: PROPUESTA DE INDICADORES

No	Indicadores	Total Factor de error	Total de errores acumulados	% del total	% del total Acumulado.
	estimación				
15	Herramienta de Programación	0	80	0	100
16	Herramienta de modelado visual	0	80	0	100
	Total	80	80	100	100

Tabla 2.19 Tabla del diagrama de Pareto.

Paso 4: Trazar y rotular los ejes del Diagrama

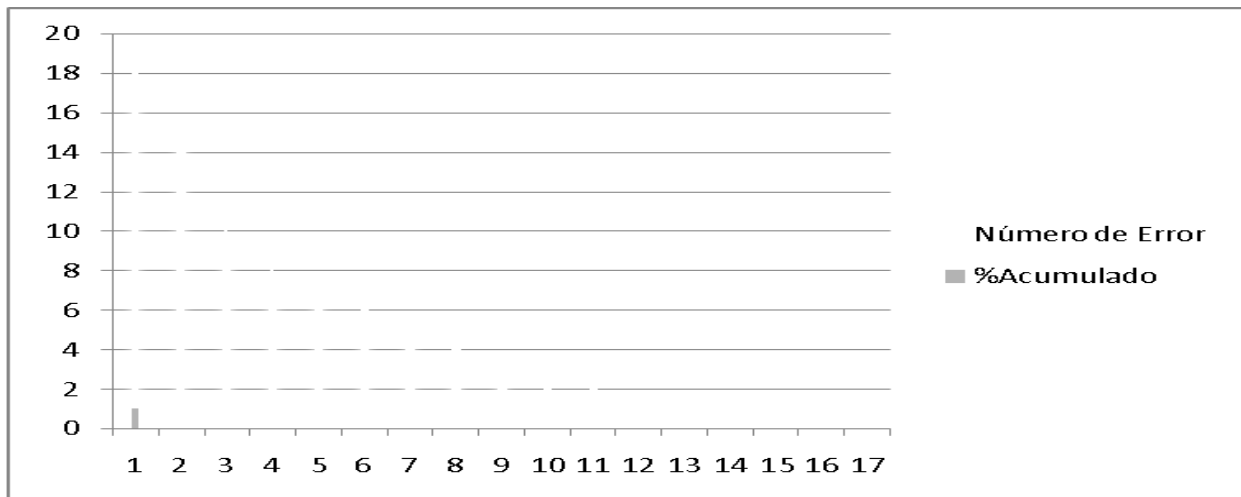


Figura 2.2 Rotular el eje de escala.

Paso 5: Dibujar un Gráfico de Barras que representa el efecto de cada uno de los elementos contribuyentes.

CAPÍTULO II: PROPUESTA DE INDICADORES

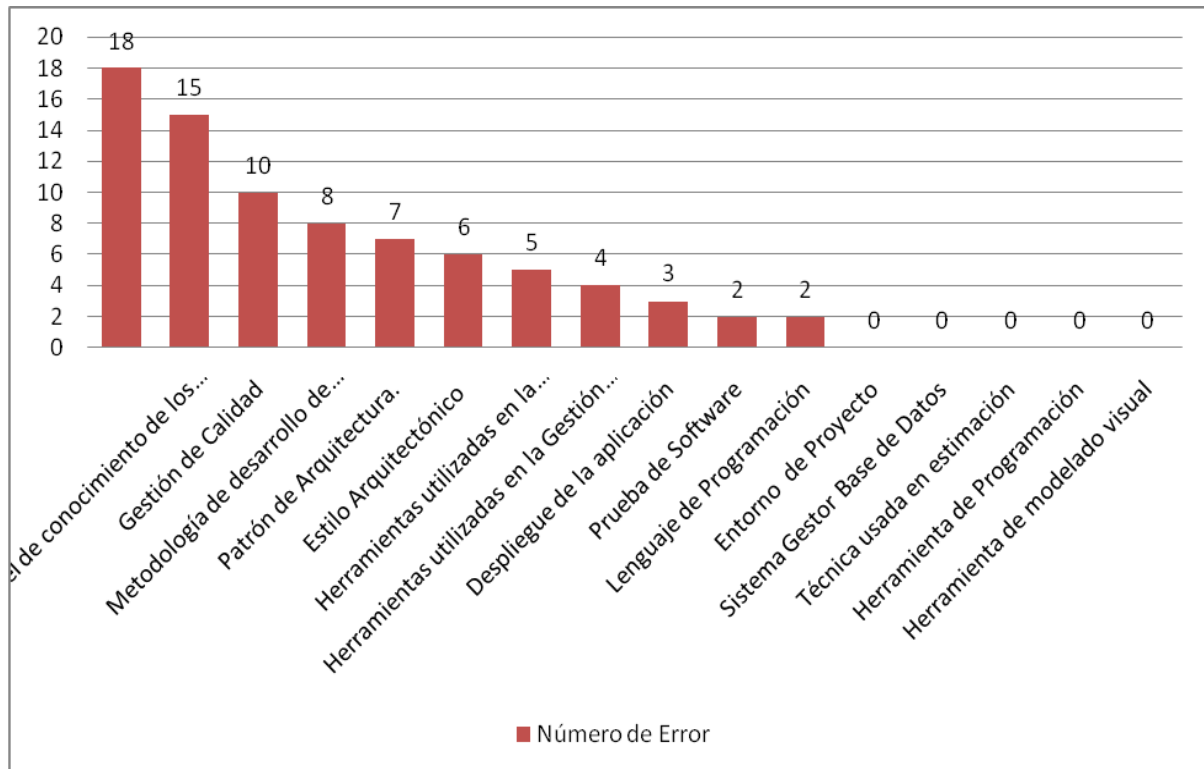


Figura 2.3 Tipos de Errores.

Paso 6: Trazar un Gráfico Lineal cuyos puntos representan el porcentaje acumulado de la Tabla de Pareto.

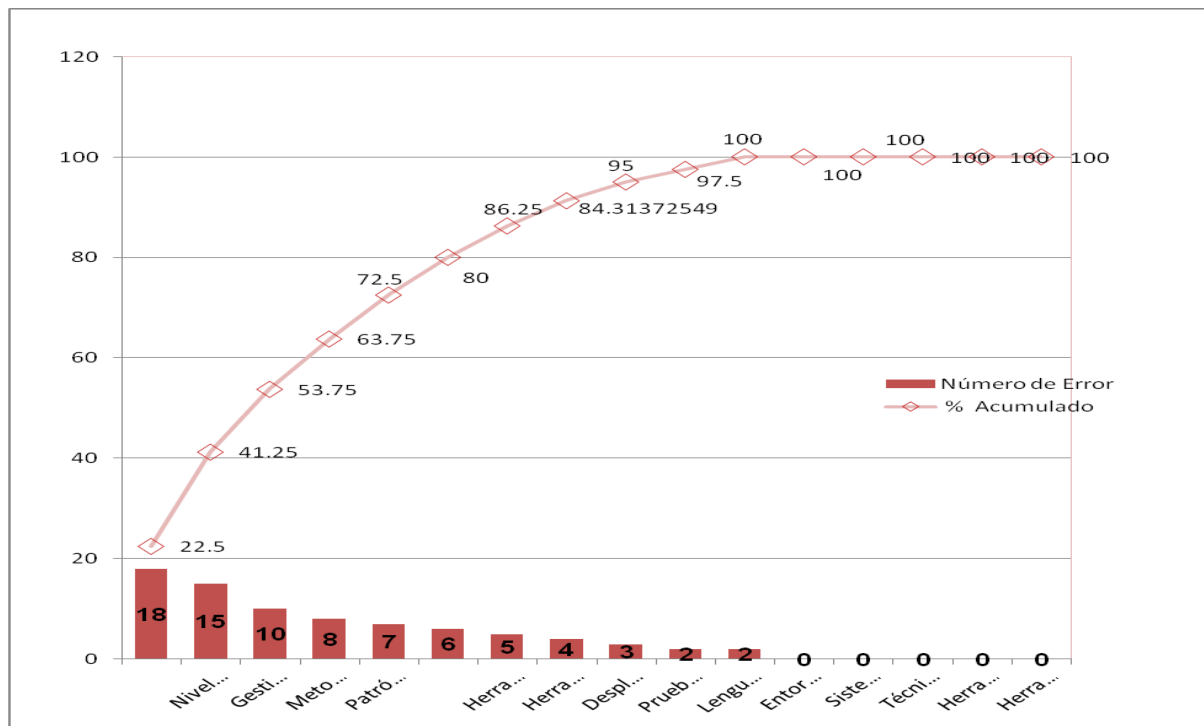


Figura 2.4 Diagrama de Pareto.

Paso 7: Señalar los elementos "Factores Críticos" y los "Muchos Triviales"

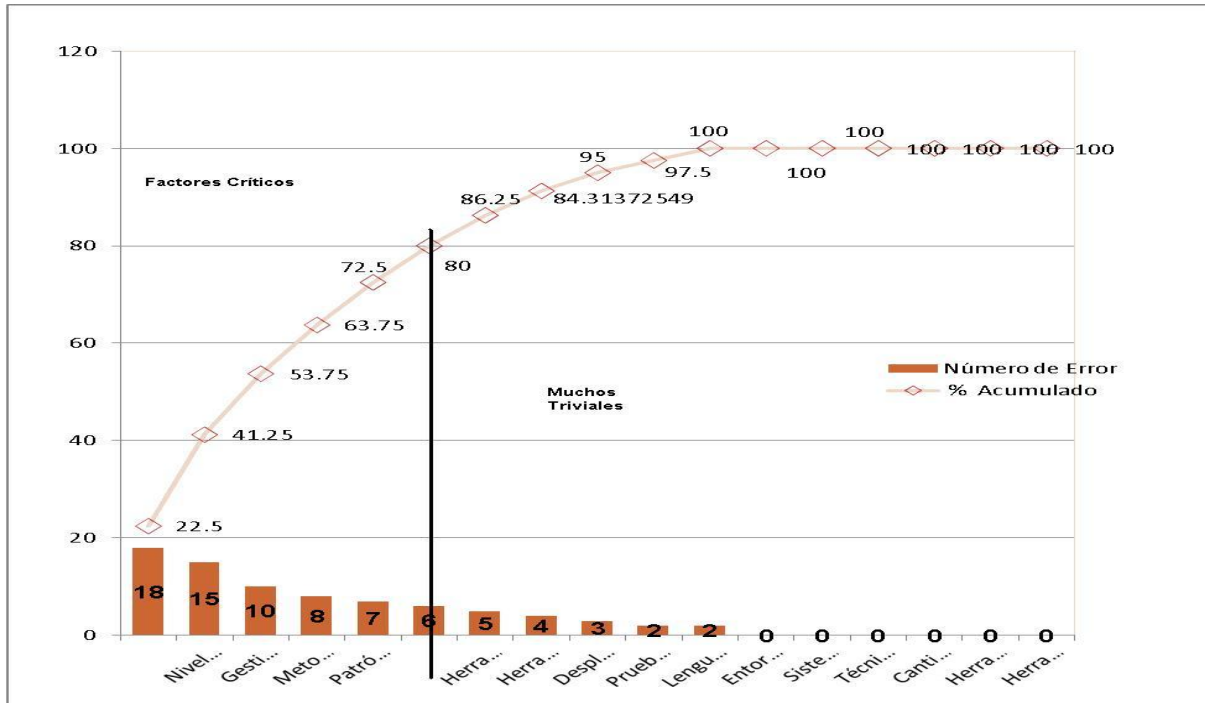


Figura 2.5 Rotular Los Factores Críticos y Muchos triviales.

Como resultado se obtiene que donde hay que incidir más para que el tiempo y esfuerzo estimado coincida con el real es en:

- Influencia de los Riesgos en la estimación
- Nivel de conocimiento de los integrantes del equipo
- Herramientas utilizadas en la Planificación.
- Patrón de Arquitectura.
- Herramientas utilizadas en la Gestión de Configuración.
- Metodología de desarrollo de software
- Lenguaje de Programación

Si el usuario interesado desea ver específicamente que valor de ese indicador causa mayor cantidad de efectos negativos en el tiempo y esfuerzo de desarrollo, realiza la misma operación con el indicador específico.

2.5 El Método Delphi en la validación de la propuesta de indicadores.

La validación de los indicadores propuestos se realizó a través de criterios de expertos mediante el uso de técnicas propuestas por el método Delphi, el cual es uno de los métodos subjetivos de pronóstico más confiables, su origen se remonta a la década de los 60, con el objetivo de

CAPÍTULO II: PROPUESTA DE INDICADORES

elaborar pronósticos referentes a posibles acontecimientos en varias ramas de la ciencia, la técnica y la política.

Delphi constituye un procedimiento para confeccionar un cuadro de la evolución de situaciones complejas, a través de la elaboración estadística de las opiniones de un grupo de expertos en el tema tratado. Permite rebasar el marco de las condiciones actuales más señaladas de un fenómeno y alcanzar una imagen integral y más amplia de su posible evolución, reflejando las valoraciones individuales de los expertos que pueden estar basadas en un análisis lógico, como en su experiencia intuitiva. Debido a lo anterior es que se ha decidido el uso del método Delphi.

Para aplicar el método se siguieron tres etapas fundamentales:

- Elección de expertos.
- Elaboración del cuestionario para la validación de la propuesta.
- Desarrollo práctico y explotación de resultados.

Selección del grupo de expertos.

“Se entiende por experto, tanto al individuo en sí como a un grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia.”[22]

La selección del posible equipo de experto se realizó bajo los siguientes criterios:

- Vinculación al desarrollo de productos informáticos.
- Experiencia en liderazgo de proyectos productivos.
- Conocimientos y habilidades en actividades de gestión de proyectos.

Los posibles candidatos se seleccionaron en la UCI, se les aplicó una encuesta de autovaloración para determinar el coeficiente de competencia, finalmente quedaron los expertos que se muestran en el **Anexo 1**. La encuesta puede verse a continuación.

Encuesta para determinar el coeficiente de competencia de los expertos

Compañero (a):

En la ejecución de la presente tesis, deseamos someter a la valoración de un grupo de expertos una propuesta de indicadores influyentes en las estimaciones que permitan recoger datos históricos de los proyectos productivos para facilitar la tarea de los líderes de proyectos contando con experiencia acumulada de proyectos anteriores. Para ello necesitamos conocer el grado de dominio que UD. posee sobre el proceso de estimación y planificación en la gestión de proyectos y con ese fin deseamos que responda lo que se le pide a continuación.

CAPÍTULO II: PROPUESTA DE INDICADORES

Nombre y apellidos: _____

Labor que realiza: _____

Años de experiencia: _____ **Especialidad:** _____

Categoría docente: _____ **Categoría científica:** _____

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

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

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

No.	Fuentes de argumentación	Grado de Influencia.		
		Alto	Medio	Bajo
1.-	Análisis realizado por Ud.			
2.-	Experiencia.			
3.-	Trabajos de autores nacionales.			
4.-	Trabajos de autores extranjeros.			
5.-	Su propio conocimiento del tema.			
6.-	Su intuición.			

Gracias por su colaboración.

Cálculo del coeficiente de competencia.

Lo que se obtiene del formulario de autovaloración es el coeficiente de competencia que se determina mediante la fórmula: $K = \frac{1}{2} (k_c + k_a)$, donde k_c es el coeficiente de conocimiento y k_a es el coeficiente de argumentación.

El coeficiente de conocimiento se obtiene de la primera pregunta del cuestionario y el coeficiente de argumentación de la segunda pregunta, el coeficiente de conocimiento se multiplica por 0.1 para que los resultados estén ajustados a probabilidades. El coeficiente de argumentación viene dado por la siguiente tabla de valores, **Tabla 2.20**, según las respuestas del experto se suman todos los valores obtenidos dando como resultado k_a .

CAPÍTULO II: PROPUESTA DE INDICADORES

No.	Fuentes de argumentación	Grado de Influencia		
		Alto	Medio	Bajo
1.-	Análisis realizado por Ud.	0.3	0.2	0.1
2.-	Experiencia.	0.5	0.4	0.2
3.-	Trabajos de autores nacionales.	0.05	0.05	0.05
4.-	Trabajos de autores extranjeros.	0.05	0.05	0.05
5.-	Su propio conocimiento del tema.	0.05	0.05	0.05
6.-	Su intuición.	0.05	0.05	0.05
7.-	Totales	1.0	0.8	0.5

Tabla 2.20 Valores del coeficiente de Argumentación.

El código para la interpretación de los coeficientes de competencia es el siguiente:

- Si $0.8 < k < 1.0$, el coeficiente de competencia es alto
- Si $0.5 < k < 0.8$, el coeficiente de competencia es medio.
- Si $k < 0.5$ el coeficiente de competencia es bajo.

El resultado del coeficiente de competencia de de todos los cuestionarios de autovaloración de los 15 expertos puede verse en el **Anexo 2**. A continuación se muestra el resultado general del coeficiente de competencia de los expertos que colaboraron, ver **Figura 2.6**.

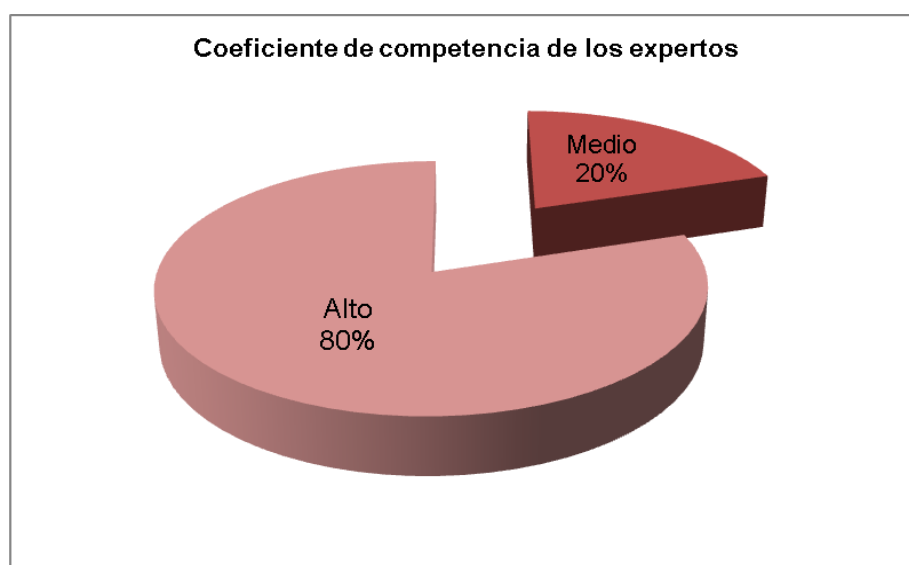


Figura 2.6 Resultado del coeficiente de competencia de los expertos.

CAPÍTULO II: PROPUESTA DE INDICADORES

Luego de seleccionar los expertos se elaboró un cuestionario que se adaptara a sus características para validar la propuesta de indicadores.

Encuesta de selección de indicadores.

Compañero (a):

La presente tesis propone un grupo de indicadores para registrar datos históricos de los proyectos productivos.

A continuación se relacionan un conjunto de indicadores, que fueron seleccionados a partir de la consulta bibliográfica, la experiencia en los proyectos productivos y la consulta a especialistas, además estuvo basada en la guía Fundamentos para la Dirección de Proyectos y en la Norma 10006 referente a la calidad en la gestión de proyectos.

Los indicadores se han agrupado en 4 clasificaciones. Valore el grado de factibilidad de cada indicador de acuerdo a la escala siguiente:

1. Muy Adecuado.
2. Bastante Adecuado.
3. Adecuado.
4. Poco Adecuado.
5. No adecuado.

No	Indicadores de Proyecto	Factibilidad
1	Tipo de Proyecto	
2	Entorno del Proyecto.	

No	Indicadores de Proceso	Factibilidad
1	Gestión de la Calidad	
2	Pruebas de Software	
3	Despliegue de la aplicación	

No	Indicadores de Recursos	Factibilidad
Recursos de Entorno		
1	Metodología de desarrollo de software	
2	Lenguaje de Programación	
3	Estilo Arquitectónico	
4	Patrón Arquitectura	
5	Herramienta de Modelado Visual	
6	Herramienta de Programación	

CAPÍTULO II: PROPUESTA DE INDICADORES

No	Indicadores de Recursos	Factibilidad
7	Herramienta sistema Gestor Base de Datos	
8	Herramientas utilizadas en la Planificación	
9	Herramientas utilizadas en la Gestión de Configuración	
10	Herramienta utilizada en estimación	
11	Cantidad de PC.	
Recursos Reutilizables		
1	Componentes Reutilizables.	
Recursos Humanos		
1	Cantidad de integrantes del equipo de desarrollo.	
2	Nivel de conocimiento de los integrantes del equipo	

No	Indicadores de Estimación	Factibilidad
1	Cantidad versiones	
2	Esfuerzo de desarrollo	
3	Tiempo de desarrollo	
4	Tamaño del producto	
5	Cantidad de procesos de complejidad de alta	
6	Cantidad de procesos de complejidad de media	
7	Cantidad de procesos de complejidad de baja	
8	Influencia de los Riesgos en la estimación	

Tabla 2.21 Indicadores Propuestos.

a) Si lo considera conveniente, proponga otros:

Expresa otros criterios o recomendaciones que pudieran servir para perfeccionar los indicadores propuestos.

Gracias por su colaboración.

2.5.1 Desarrollo práctico y explotación de resultados.

Se envió el cuestionario a los expertos seleccionados, vía e-mail o se les entregó una copia impresa, se le explicó a cada experto las finalidades del método Delphi, así como las condiciones prácticas del desarrollo de la encuesta y el plazo de respuesta.

Los resultados se recogen en la **Tabla 2.22** que se muestra a continuación.

CAPÍTULO II: PROPUESTA DE INDICADORES

Tabla de frecuencias absolutas:							
No	Elementos	C1	C2	C3	C4	C5	Total
1	Tipo de Proyecto	7	2	5	1		15
2	Entorno del proyecto	3	1	1			15
3	Gestión de la Calidad	8	5	2			15
4	Pruebas de Software	7	7	1			15
5	Despliegue de la aplicación	6	7	2			15
6	Metodología de desarrollo de software	9	4	2			15
7	Herramienta de modelado visual	3	2	7	3		15
8	Lenguaje de Programación	4	7	4			15
9	Herramienta de Programación	4	7	4			15
10	Herramienta sistema Gestor Base de Datos	5	4	5	1		15
11	Herramientas utilizadas en la Planificación	6		6	3		15
12	Herramientas utilizadas en la Gestión de Configuración	2	7	4	2		15
13	Cantidad de integrantes del equipo de desarrollo	7	4	4			15
14	Cantidad de PC	3	7	4	1		15
15	Nivel de conocimiento de los integrantes del equipo	7	4	3	1		15
16	Componentes Reutilizables	6	6	1	2		15
17	Estilo Arquitectónico	7	5	3			15
18	Patrón de Arquitectura	9	4	1	1		15
19	Herramienta utilizada en estimación	8	2	4	1		15
20	Cantidad de Versiones	3	6	5	1		15
21	Esfuerzo de desarrollo	7	5	2	1		15
22	Tiempo de desarrollo	9	3	3			15
23	Tamaño del producto	8	4	3			15
24	Cantidad de procesos de complejidad de alta	7	5	3			15
25	Cantidad de procesos de complejidad de	9	3	2	1		15

CAPÍTULO II: PROPUESTA DE INDICADORES

Tabla de frecuencias absolutas:							
No	Elementos	C1	C2	C3	C4	C5	Total
	media						
26	Cantidad de procesos de complejidad de baja	8	5	1	1		15
27	Influencia de los Riesgos en la estimación	9	3	3			15
Total de Aspectos a validar: 27							

Tabla 2.22 Frecuencias Absolutas.

Tabulados los datos, se realizan los siguientes pasos para obtener los resultados deseados:

Primer paso: Se construye una tabla de frecuencias acumuladas, **Tabla 2.23**.

Tabla de frecuencias absolutas acumuladas:						
No	Elementos	C1	C2	C3	C4	C5
1	Tipo de Proyecto	7	9	14	15	
2	Entorno del proyecto	3	4	5		
3	Gestión de la Calidad	8	13	15		
4	Pruebas de Software	7	14	15		
5	Despliegue de la aplicación	6	13	15		
6	Metodología de desarrollo de software	9	13	15		
7	Herramienta de modelado visual	3	5	12	15	
8	Lenguaje de Programación	4	11	15		
9	Herramienta de Programación	4	11	15		
10	Herramienta sistema Gestor Base de Datos	5	9	14	15	
11	Herramientas utilizadas en la Planificación	6		12	15	
12	Herramientas utilizadas en la Gestión de Configuración	2	9	13	15	
13	Cantidad de integrantes del equipo de desarrollo	7	11	15		
14	Cantidad de PC	3	10	14	15	
15	Nivel de conocimiento de los integrantes del equipo	7	11	14	15	
16	Componentes Reutilizables	6	12	13	15	
17	Estilo Arquitectónico	7	12	15		

CAPÍTULO II: PROPUESTA DE INDICADORES

Tabla de frecuencias absolutas acumuladas:						
18	Patrón de Arquitectura	9	13	14	15	
19	Herramienta utilizada en estimación	8	10	14	15	
20	Cantidad de Versiones	3	9	14	15	
21	Esfuerzo de desarrollo	7	12	14	15	
22	Tiempo de desarrollo	9	12	15		
23	Tamaño del producto	8	12	15		
24	Cantidad de procesos de complejidad de alta	7	12	15		
25	Cantidad de procesos de complejidad de media	9	12	14	15	
26	Cantidad de procesos de complejidad de baja	8	13	14	15	
27	Influencia de los Riesgos en la estimación	9	12	15		

Tabla 2.23 Frecuencias Absolutas Acumuladas.

Segundo paso: Se construye la tabla de frecuencias relativas acumulativas, **Tabla 2.24.**

Tabla de frecuencias relativas acumuladas:						
No	Elementos	C1	C2	C3	C4	C5
1	Tipo de Proyecto	0.46666667	0.6	0.9333	0.9999	
2	Entorno del proyecto	0.6	0.8	0.9999		
3	Gestión de la Calidad	0.53333333	0.8667	0.9999		
4	Pruebas de Software	0.46666667	0.9333	0.9999		
5	Despliegue de la aplicación	0.4	0.8667	0.9999		
6	Metodología de desarrollo de software	0.6	0.8667	0.9999		
7	Herramienta de modelado visual	0.2	0.3333	0.8	0.9999	
8	Lenguaje de Programación	0.26666667	0.7333	0.9999		
9	Herramienta de Programación	0.26666667	0.7333	0.9999		
10	Herramienta Sistema Gestor Base de Datos	0.33333333	0.6	0.9333	0.9999	
11	Herramientas utilizadas en la Planificación	0.4		0.8	0.9999	
12	Herramientas utilizadas en la Gestión de Configuración	0.13333333	0.6	0.8667	0.9999	
13	Cantidad de integrantes del equipo de	0.46666667	0.7333	0.9999		

CAPÍTULO II: PROPUESTA DE INDICADORES

Tabla de frecuencias relativas acumuladas:						
	desarrollo					
14	Cantidad de PC	0.2	0.6667	0.9333	0.9999	
15	Nivel de conocimiento de los integrantes del equipo	0.46666667	0.7333	0.9333	0.9999	
16	Componentes Reutilizables	0.4	0.8	0.8667	0.9999	
17	Estilo Arquitectónico	0.46666667	0.8	0.9999		
18	Patrón de Arquitectura	0.6	0.8667	0.9333	0.9999	
19	Herramienta utilizada en estimación	0.53333333	0.6667	0.9333	0.9999	
20	Cantidad de Versiones	0.2	0.6	0.9333	0.9999	
21	Esfuerzo de desarrollo	0.46666667	0.8	0.9333	0.9999	
22	Tiempo de desarrollo	0.6	0.8	0.9999		
23	Tamaño del producto	0.53333333	0.8	0.9999		
24	Cantidad de procesos de complejidad de alta	0.46666667	0.8	0.9999		
25	Cantidad de procesos de complejidad de media	0.6	0.8	0.9333	0.9999	
26	Cantidad de procesos de complejidad de baja	0.53333333	0.8667	0.9333	0.9999	
27	Influencia de los Riesgos en la estimación	0.6	0.8	0.9999		

Tabla 2.24 Frecuencias relativas acumuladas.

Esta tabla se logra dividiendo por el número total de expertos, en este caso 15 cada uno de los números de la tabla anterior.

Tercer paso: Se buscan los puntos de corte de la tabla anterior.

A la misma tabla se le adicionan tres columnas y una fila para colocar los resultados que se mencionan a continuación.

1. Suma de las columnas.
2. Suma de filas.
3. Promedio de las columnas.
4. Los promedios de las filas se obtienen de forma similar, en este caso también se divide por cuatro porque quedan 4 categorías ya que la última se eliminó.

CAPÍTULO II: PROPUESTA DE INDICADORES

5. Para hallar N, se divide la suma entre el resultado de multiplicar el número de indicadores por el número de preguntas.

6. El valor N-P da el valor promedio que otorgan los expertos para cada indicador propuesto.

El Anexo 3 resume lo dicho en los puntos anteriores:

Las sumas obtenidas en las cuatro primeras columnas dan los puntos de cortes. Los puntos de corte se utilizan para determinar la categoría o grado de adecuación de cada criterio según la opinión de los expertos consultados. Con ellos se opera del modo siguiente, según el método Delphi. **Tabla 2.25:**

Muy adecuado	Bastante adecuado	Adecuado	Poco adecuado	No adecuado
0,44	0,74	1,53	3,72	

Tabla 2.25. Valores de los puntos de corte.

Resultado Final de la Validación de los indicadores propuestos.

Luego de aplicar el método Delphi a un grupo de 15 expertos, el criterio general acerca de la propuesta de indicadores se muestra a continuación, ver **Figura 2.7**.

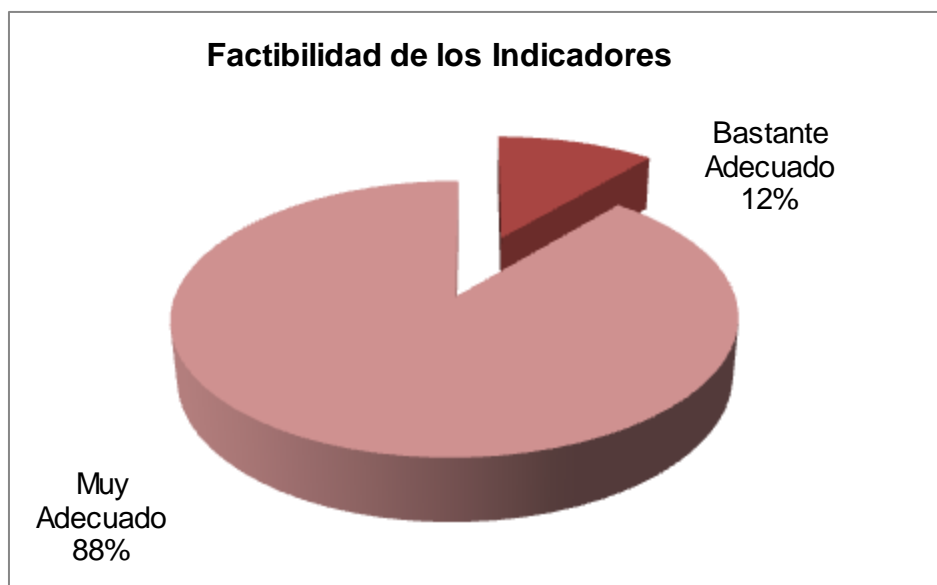


Figura 2.7 Resultado Final de la Validación de los indicadores propuestos

2.6 Conclusiones parciales

En este capítulo se describió una panorámica de la situación de la producción en la UCI, justificando la necesidad del registro histórico para la estimación de los proyectos. Además se desarrolló la propuesta de los indicadores que servirán de apoyo y referencia a los líderes de proyectos para estimar el tiempo y esfuerzo del proyecto que van a dirigir, dándole la solución a uno de los objetivos específicos de nuestro trabajo de diploma. Se definió la forma de medir estos indicadores según el impacto que tuvieron en los proyectos y su influencia real en el tiempo de desarrollo y el esfuerzo. Además se definió una propuesta inicial de análisis estadístico. Los indicadores propuestos se sometieron a la validación de un grupo de 15 expertos en gestión de proyectos, la mayoría líderes de proyectos los cuales le otorgaron excepto a 3 de los 27 propuestos la máxima calificación de Muy Adecuados.

CAPÍTULO 3: PROPUESTA DEL SISTEMA

En este capítulo se tienen en cuenta los temas referentes al desarrollo de la solución que se propone. Se modeló la aplicación siguiendo lo especificado por los flujos de trabajos que plantea RUP hasta análisis, se generarán los siguientes artefactos: modelo de dominio, especificación de los requisitos y modelo de casos de uso del sistema. En el caso del análisis, se generan artefactos que corresponden a este flujo como son: diagramas de paquetes generales y los diagramas de clases de análisis por cada caso de uso. Para culminar el desarrollo de la solución propuesta, se desarrolló un prototipo de interfaz de usuario.

3.1 Necesidad que justifica el desarrollo del sistema.

Una vez planteada la propuesta de indicadores del Capítulo 2 se hace necesario contar con un sistema automatizado que la sustente y permita recoger datos de los proyectos para conformar el registro histórico, debe de existir un sistema que permita gestionar todos los indicadores para apoyar las estimaciones y planificaciones de los proyectos productivos de la UCI, la cual no tiene hasta el momento ninguna referencia histórica que como toda empresa productora de software debe de poseer para registrar los resultados de todos los proyectos que se han desarrollado en ella. La tarea de consultar los indicadores de los proyectos se rige por niveles de acceso en dependencia del cargo que ocupe la persona que desee consultar los datos, los jefes de proyectos solo pueden acceder a la información de su proyecto, mientras que los jefes de polo pueden consultar todos los indicadores de los proyectos de su polo para conocer el comportamiento de estos. Si se desea consultar información que no se encuentre a su alcance se deberá elevar la solicitud a un nivel superior que tenga acceso a la información, entregándose posteriormente un reporte de los datos que se soliciten. Por su parte a los usuarios directivos que el administrador registre se les otorgará el permiso de acceso a la información según el cargo que posea la persona.

3.2 Modelo de Dominio.

El modelo del dominio representa los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan conceptos que existen o eventos que suceden en el entorno en el que trabaja el sistema. A continuación en la **Figura 3.1** se muestra el modelo de dominio que se desarrolló y en la **Tabla 3.1** se recoge el significado de cada concepto del modelo.

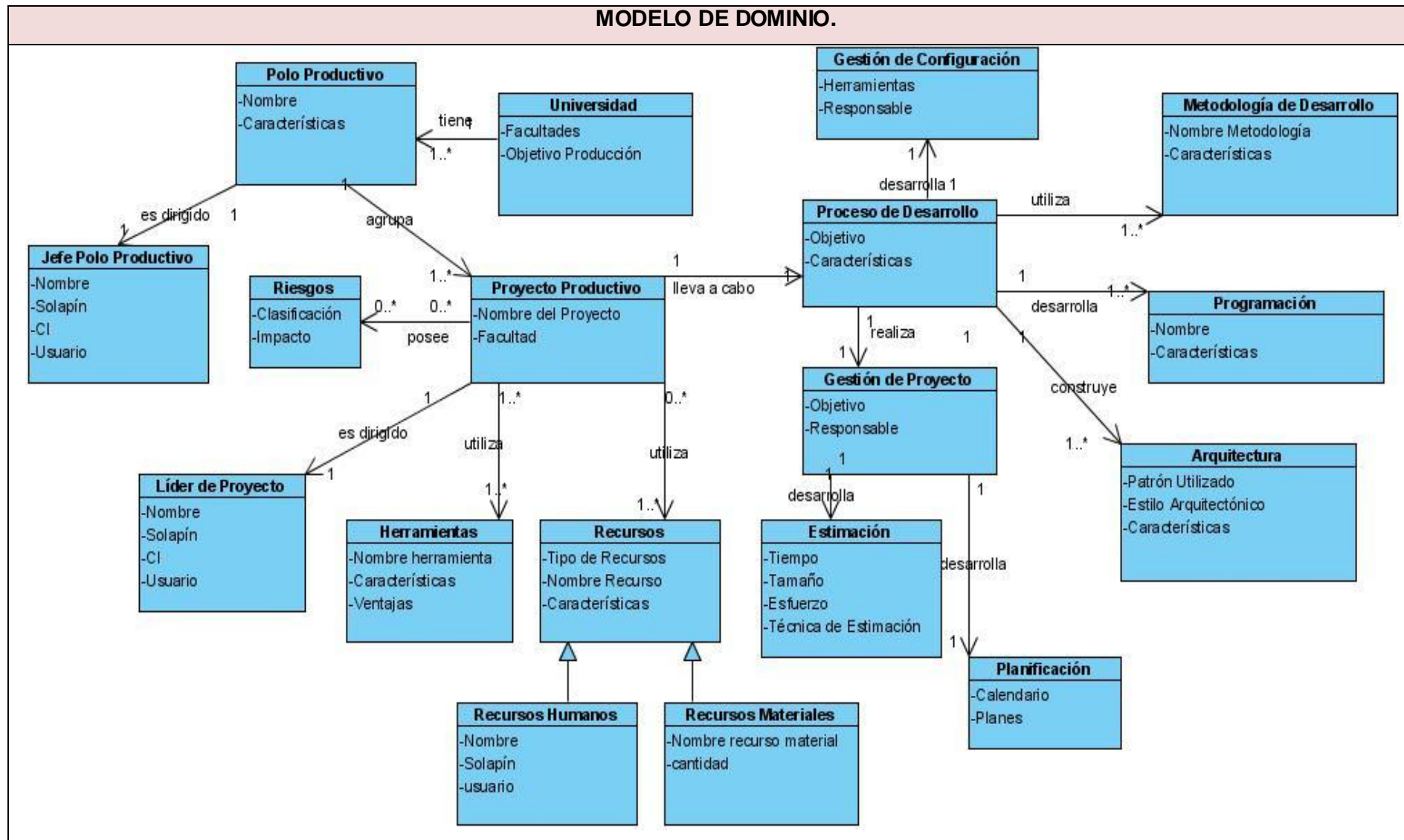


Figura 3.1 Modelo de Dominio.

CAPÍTULO III: PROPUESTA DEL SISTEMA

Conceptos del Modelo de Dominio.

Concepto	Descripción
Proyecto Productivo	Es un conjunto de actividades planificadas y supervisadas que tiene como objetivo crear un producto vinculando estudiantes, profesores y trabajadores para el cumplimiento del objetivo.
Jefe de Proyecto	Persona que dirige el Proyecto encargada de la gestión y administración de los recursos, planificaciones y estimaciones.
Polo Productivo	Estructura que incluye y supervisa varios proyectos relacionados con el tema definido en el polo.
Jefe de Polo	Persona que supervisa y controla un polo productivo y los proyectos vinculados a ese polo.
Experiencia	Experiencia acumulada por el proyecto en el transcurso del desarrollo del software.
Universidad	Universidad de las Ciencias Informáticas.
Proceso de desarrollo	Es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto.
Estimación	Actividad que se ejecuta en el proyecto para obtener un valor aproximado del tiempo, esfuerzo y tamaño que tendrá el proyecto.
Planificación	Actividad para planificar el desarrollo del software, incluye calendario de actividades y metas que se cumplen en el tiempo establecido.
Arquitectura	La arquitectura es el esqueleto o base de una aplicación, este concepto se refiere a los patrones de arquitectura o estilos utilizados.
Programación	Se refiere al concepto que reúne todas las herramientas y lenguajes utilizados en la implementación del proyecto.

CAPÍTULO III: PROPUESTA DEL SISTEMA

Concepto	Descripción
Herramientas	Todas las herramientas que utilice el proyecto en su desarrollo. Software que se utiliza para automatizar las actividades definidas en el proceso.
Metodología de Desarrollo	Es un proceso configurable para el desarrollo del proyecto que guía los esfuerzos de desarrollo del proyecto.
Gestión de Configuración	Es la disciplina que controla los cambios y las versiones en el proyecto, el concepto recoge las estrategias utilizadas y las herramientas.
Recursos	Recursos materiales y humanos involucrados en el proyecto durante el desarrollo.

Tabla 3.1 Conceptos del Modelo de Dominio.

3.2.1 Justificación del uso del modelo de dominio.

Durante el desarrollo de los proyectos siempre resulta útil la creación de modelos que organicen y presenten los detalles importantes de problemas reales que se vinculan con el sistema informático a construir. Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio. El modelo del negocio es una de las técnicas para la especificación de los requisitos más importantes del sistema, lo cual refuerza la idea de que sea el propio negocio el que determine los requisitos.

Dependiendo de la situación o escenario que se presente, hay varias alternativas de desarrollar este proceso. Si se determina que no es necesario un modelo completo del negocio se realizará un modelo del dominio.

Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema, no incluye las responsabilidades de las personas que ejecutan las actividades. El modelo del dominio se considera en RUP un subconjunto del llamado modelo de objetos del negocio

El modelo de dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema.

En la UCI no existe un registro histórico de indicadores que los líderes de proyectos utilicen en las planificaciones, estimaciones y definición de los recursos, por lo que es muy difícil modelar un negocio actual debido a que los procesos no están claramente definidos, lo más conveniente fue

realizar un modelo de dominio que representara visualmente los conceptos de interés en el área del problema que se estudió.

3.3 Requerimientos del Sistema.

Los requerimientos son condiciones o capacidades que necesita un usuario para resolver un problema o lograr un objetivo. Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos.

3.3.1 Especificación de Requisitos de Software.

3.3.1.1 Requisitos Funcionales

Los requerimientos funcionales (RF) se pueden definir como capacidades o condiciones que el sistema debe cumplir. Dentro de los requisitos funcionales del sistema se encuentran:

RF1 Gestionar usuarios de la aplicación.

- 1.1 Registrar los datos del usuario (nombre, solapín, usuario).
- 1.2 Modificar los datos del usuario (nombre, solapín, usuario).
- 1.3 Eliminar el usuario.

RF2. Autenticar los usuarios.

- 2.1 Verificar que el usuario esté registrado en el dominio de la UCI.
- 2.2 Verificar que el usuario esté registrado en la aplicación.
- 2.3 Cargar la configuración correspondiente al usuario registrado.

RF3. Gestionar el registro del proyecto.

- 3.1 Registrar los datos del proyecto.
 - Tipo de Proyecto
 - Entorno del proyecto
 - Gestión de la Calidad
 - Pruebas de Software
 - Despliegue de la aplicación
 - Metodología de desarrollo de software
 - Herramienta de modelado visual
 - Lenguaje de Programación
 - Herramienta de Programación
 - Sistema Gestor Base de Datos
 - Herramientas utilizadas en la Planificación
 - Herramientas utilizadas en la Gestión de Configuración
 - Cantidad de integrantes del equipo de desarrollo

- Cantidad de PC
- Nivel de conocimiento de los integrantes del equipo
- Componentes Reutilizables
- Estilo Arquitectónico
- Patrón de Arquitectura
- Herramientas utilizadas en la estimación
- Cantidad de versiones
- Esfuerzo de desarrollo
- Tiempo de desarrollo
- Tamaño del producto
- Cantidad de procesos de complejidad de alta
- Cantidad de procesos de complejidad de media.
- Cantidad de procesos de complejidad de baja.
- Influencia de los riesgos en la estimación.

2.2 Modificar los datos de un proyecto.

2.3 Eliminar el proyecto.

2.4 Mostrar datos del proyecto registrado.

2.5 Cancelar las operaciones cuando el actor desee.

2.6 Validar los datos registrados.

RF4. Gestionar los indicadores.

3.1 Registrar indicador.

- Nombre
- Justificación

4.1 Modificar los datos del indicador.

4.2 Eliminar el indicador.

4.3 Cancelar las operaciones cuando el actor desee.

4.4 Validar los datos registrados.

RF5. Gestionar los valores de los indicadores.

5.1 Registrar valores de un indicador.

- Nombre del valor
- Características

5.2 Modificar los valores de un indicador.

5.3 Eliminar el indicador.

5.4 Cancelar las operaciones cuando el actor desee.

5.5 Validar los datos registrados.

RF6 Gestionar polos productivos.

- 6.1 Registrar los valores de un polo productivo.
- 6.2 Modificar los valores de un polo productivo.
- 6.3 Eliminar un polo productivo.
- 6.4 Cancelar las operaciones cuando el actor desee.
- 6.5 Validar los datos registrados.

RF7. Registrar Jefes de Polos.

- 7.1 Cancelar las operaciones cuando el actor desee.
- 7.2 Validar los datos registrados.

RF8. Gestionar la Ficha del Proyecto

- 8.1 Comprobar los datos del líder de proyecto en el servidor web de la UCI.
- 8.2 Asignar el permiso de líder de proyecto.
- 8.3 Registrar el nombre del proyecto y asignárselo al líder del proyecto.
- 8.4 Cancelar las operaciones cuando el actor desee.
- 8.5 Validar los datos registrados.

RF9. Consultar información de los proyectos.

- 8.1 Buscar los proyectos por los siguientes criterios de búsqueda.
 - Tipo de proyecto
 - Entorno del proyecto
 - Gestión de la Calidad
 - Pruebas de Software
 - Despliegue de la aplicación
 - Metodología de desarrollo de software
 - Herramienta de modelado visual
 - Lenguaje de Programación
 - Herramienta de Programación
 - Sistema Gestor Base de Datos
 - Herramientas utilizadas en la Planificación
 - Herramientas utilizadas en la Gestión de Configuración
 - Cantidad de integrantes del equipo de desarrollo
 - Cantidad de PC
 - Nivel de conocimiento de los integrantes del equipo
 - Componentes Reutilizables
 - Estilo Arquitectónico
 - Patrón de Arquitectura

- Herramientas utilizadas en la estimación
- Cantidad de versiones
- Esfuerzo de desarrollo
- Tiempo de desarrollo
- Tamaño del producto
- Cantidad de procesos de complejidad de alta
- Cantidad de procesos de complejidad de media.
- Cantidad de procesos de complejidad de baja.

9.1 Cancelar las operaciones cuando el actor desee.

9.2 Validar los datos registrados.

9.3 Mostrar la información de un proyecto seleccionado.

9.4 Mostrar resultados estadísticos.

RF10 Permitir cambiar la contraseña del usuario del administrador creada por defecto.

RF11 Permitir obtener resultados estadísticos de todos los proyectos registrados.

3.3.1.2 Requisitos No Funcionales

Los requerimientos no funcionales (RNF) son propiedades o cualidades que el producto debe tener. Son las características que hacen al producto atractivo, usable, rápido y confiable. Estos requisitos son muy importantes para que los clientes y usuarios puedan valorar características no funcionales del software como: cuán agradable, segura, y confiable es la aplicación.

Apariencia o interfaz externa:

- El software debe brindar una interfaz simple, amigable y fácil de usar.
- Diseño para resoluciones de 800x600, pero preparado para verse en otras resoluciones.

Usabilidad:

- Debe contar con un mapa que indique a los usuarios la manera más fácil de navegar.

Confiabilidad:

- El sistema deberá garantizar absoluta confiabilidad de la información que brinda así como de su manejo.
- La aplicación debe estar disponible las 24 horas al día los 7 días de la semana de forma tal que se pueda acceder a sus funcionalidades en cualquier momento.

Seguridad:

- El sistema debe comunicarse usando un protocolo seguro. (Https).
- El usuario debe autenticarse antes de entrar al sistema, su autenticación será negociada con el servicio, Lightweight Directory Access Protocol (LDAP) de la UCI.
- Permitir que las funcionalidades del sistema se muestren y correspondan a los usuarios que tengan permisos para ello.

CAPÍTULO III: PROPUESTA DEL SISTEMA

- Realizar copias de la Base de Datos para evitar pérdidas de información.

Rendimiento:

- Tiempos de respuestas de no más de 5 segundos.

Hardware:

- La máquina donde se encontrará el servidor debe tener como mínimo las siguientes características de hardware: Procesador Pentium IV 1.5 GHz o superior, 512 Mb de memoria RAM o superior y 40 Gb de capacidad en disco duro.
- La máquina cliente debe tener procesador Pentium o superior con 128 Mb de RAM como mínimo y 20 Gb de memoria en el disco duro.

Software

- La máquina donde se encontrará el servidor requiere tener instalado el sistema Operativo Linux.
- Se utilizará el interprete de PHP 5.0
- La aplicación utilizará como base de datos PostgreSQL.
- Se utilizará como servidor web Apache 2.0.

Legales:

- El sistema debe estar acorde con las políticas de seguridad informática establecidas en la UCI.

Portabilidad:

- El sistema debe correr en ambientes multiplataforma.

3.4 Modelo de Casos de uso del sistema.

El modelo de casos de uso del sistema ayuda al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre cómo utilizar el sistema. Cada actor utiliza el sistema al interactuar con los casos de uso. Por tanto todos los actores y casos de uso del sistema forman el modelo de casos de uso que se muestra en la **Figura 3.2**.

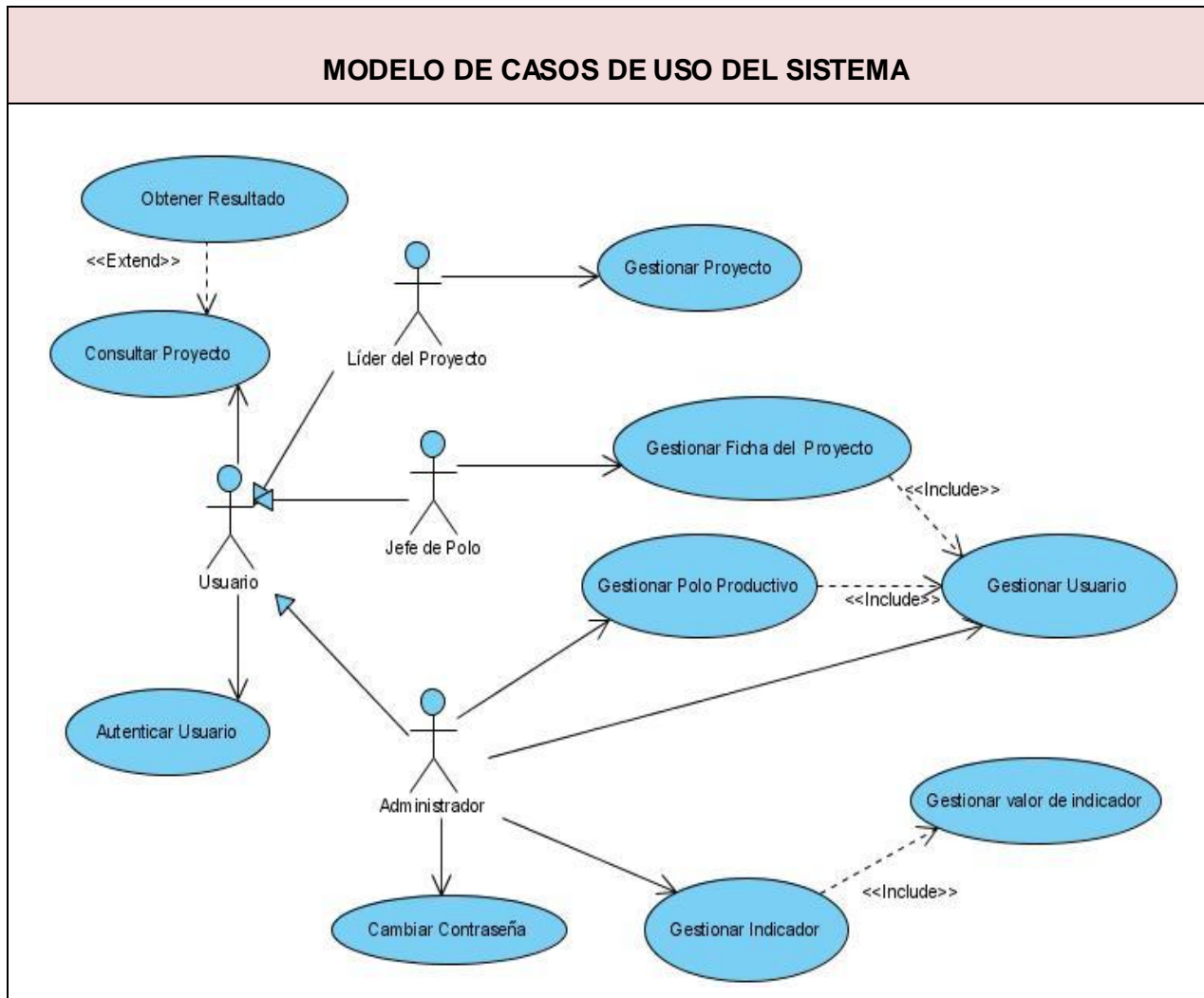


Figura 3.2 Modelo de casos de uso del sistema.

3.4.1 Definición de los actores del sistema.

Los actores del sistema son los que interactúan con la aplicación, no son parte del sistema, pueden intercambiar información con él y pueden ser un recipiente pasivo de información. También pueden representar el rol que juega una o varias personas de un equipo o un sistema automatizado. Los actores del sistema propuesto son descritos seguidamente en la **Tabla 3.2**:

Actor	Descripción
Usuario	Es el actor del sistema que interactúa con los CU comunes para todos los actores respondiendo a una jerarquía de herencia. Realiza el caso de uso Autenticar usuario común para todos los actores y consulta la

CAPÍTULO III: PROPUESTA DEL SISTEMA

Actor	Descripción
	información de los proyectos.
Jefe de Proyecto	El jefe de proyecto se encarga de gestionar la información de su proyecto, además de consultar todos los proyectos.
Jefe de Polo	El jefe de polo se encarga de definir la ficha del proyecto especificando y administrando los datos del líder de proyecto. Crea la asignación de un proyecto de su polo al jefe de proyecto, el cuál posteriormente concluirá de introducir todo el registro del proyecto.
Administrador	El administrador se encarga de gestionar polos productivos y los usuarios que soliciten ver los diferentes proyectos así como el análisis de los datos, y gestiona los indicadores con sus respectivos valores. Puede cambiar la contraseña del usuario del administrador definido por defecto en el sistema.

Tabla 3.2 Actores del sistema.

3.4.2 Descripción Textual de los Casos de uso.

Para entender la funcionalidad asociada a cada caso de uso no es suficiente con la representación gráfica del diagrama de casos de uso. La descripción textual apoya el diagrama de casos de uso describiendo los pasos dentro de la funcionalidad, puede ser elaborada de forma breve o extendida a continuación en la **Tabla 3.3** se muestra la descripción textual del caso de uso Gestionar Proyecto, las demás descripciones pueden verse en el **Anexo 4**.

CAPÍTULO III: PROPUESTA DEL SISTEMA

Nombre del CU	Gestionar Proyecto	
Actores	Jefe Proyecto (inicia)	
Propósito	Crear, modificar o eliminar un Proyecto.	
Resumen	El caso de uso se inicia cuando el actor accede a la opción que le permite gestionar los datos de un proyecto determinado. En caso de registrar los datos de su proyecto el sistema brinda la posibilidad de introducir los datos. En caso de modificar los datos de su proyecto, el sistema muestra todos los datos y brinda la posibilidad de añadir nuevos datos o modificar los ya existentes, actualizando los datos del proyecto en el sistema. En caso de eliminar su proyecto, el sistema brinda la posibilidad de eliminar el proyecto.	
Referencias	RF3	
CU Asociados		
Precondiciones	El usuario debe de estar autenticado. El sistema debe de tener cargado el nombre del proyecto y el nombre del líder de proyecto.	
Poscondiciones	Se registraron los datos del proyecto, se modificó o eliminó el proyecto.	
Flujo Básico		
Acciones del Actor	Respuesta del Sistema	
1 El caso de uso se inicia cuando el actor accede a la opción de gestionar los datos de su proyecto.		
	2 El sistema brinda la posibilidad de introducir los siguientes datos y su impacto en el proyecto: Proyecto: <ul style="list-style-type: none"> • Tipo de Proyecto • Entorno de Proyecto. Proceso	

CAPÍTULO III: PROPUESTA DEL SISTEMA

	<ul style="list-style-type: none">• Gestión de la Calidad• Pruebas de Software• Despliegue de la Aplicación <p>Recursos:</p> <ul style="list-style-type: none">• Metodología de desarrollo de software.• Herramienta de modelado visual.• Herramienta sistema Gestor Base de Datos.• Herramientas utilizadas en la Planificación.• Herramientas utilizadas en la Gestión de Configuración.• Cantidad de integrantes del equipo de desarrollo.• Cantidad de PC.• Nivel de conocimiento de los integrantes del equipo.• Componentes Reutilizables Utilizados.• Herramienta utilizada en estimación.• Influencia de los riesgos en la estimación. <p>Programación:</p> <ul style="list-style-type: none">• Lenguaje de Programación.• Herramienta de Programación. <p>Arquitectura:</p> <ul style="list-style-type: none">• Patrón Arquitectura.• Estilo Arquitectónico. <p>Estimación:</p> <ul style="list-style-type: none">• Tiempo realización del Proyecto técnico• Cantidad de versiones.
--	---

CAPÍTULO III: PROPUESTA DEL SISTEMA

	<ul style="list-style-type: none"> • Esfuerzo de desarrollo. • Tiempo de desarrollo. • Tamaño del producto. • Cantidad de procesos de complejidad de alta. • Cantidad de procesos de complejidad de media. • Cantidad de procesos de complejidad de baja <p>El sistema permite realizar las acciones siguientes:</p> <ul style="list-style-type: none"> • Registrar los datos de los indicadores del proyecto. • Modificar los datos de su proyecto. Ver Sección 1: "Modificar Proyecto". • Eliminar su proyecto. Ver Sección 2: "Eliminar Proyecto".
<p>3 El actor introduce los datos y su impacto y selecciona la opción registrar.</p>	
	<p>4 El sistema valida los datos.</p>
	<p>5 El sistema muestra un mensaje de confirmación "Los datos del proyecto se han registrado correctamente".</p>
	<p>6 El sistema muestra los datos introducidos.</p> <p>El sistema permite realizar las acciones:</p> <ul style="list-style-type: none"> • Modificar los datos de su proyecto. Ver Sección 2: "Modificar Proyecto". • Eliminar su Proyecto. Ver Sección

CAPÍTULO III: PROPUESTA DEL SISTEMA

	3:"Eliminar Proyecto".
Curso Alterno	
2a. El actor selecciona la opción de cancelar.	
	2.1a El sistema cancela la operación.
2b. El actor selecciona la opción de eliminar un valor introducido.	
	2b.1 El sistema elimina el valor introducido.
	4a. El sistema muestra el siguiente mensaje indicando que existen datos incorrectos:" Existen datos incorrectos. Por favor verifique los datos".
Sección 1 Modificar Proyecto	
1 El actor selecciona la opción de modificar su proyecto.	
	2 El sistema muestra los datos registrados del proyecto en modo editable y brinda la posibilidad de modificar los datos.
3 El actor modifica los datos que necesite.	
4 El actor selecciona la opción de actualizar los datos del proyecto.	
	5 El sistema valida los datos.
	6 El sistema actualiza los datos del proyecto con la información que fue introducida o seleccionada por el actor.
	7 El sistema muestra el mensaje de información: "Se han actualizado los cambios sobre el proyecto.
Curso Alterno	
1a. El actor selecciona la opción de cancelar.	
	1.1a El sistema cancela la operación.
	5a. El sistema muestra el siguiente mensaje

CAPÍTULO III: PROPUESTA DEL SISTEMA

	indicando que existen datos incorrectos:” Existen datos incorrectos. Por favor verifique los datos”.
Sección 2 Eliminar Proyecto	
1 El actor selecciona la opción de eliminar el proyecto.	
	2 El sistema pide confirmación si está seguro de eliminar el proyecto.
3 El actor confirma eliminar el proyecto.	
	4 El sistema elimina el proyecto.
Curso Alterno	
1a. El actor selecciona la opción de cancelar.	
	1.1a El sistema cancela la operación.
Prioridad: Crítico	

Tabla 3.3 Descripción Textual del CU Gestionar Proyecto.

3.5 Modelo de Análisis de la solución propuesta.

El modelo de análisis es usado para definir con mayor detalle los requisitos funcionales del sistema en construcción, además de ser la base del modelo de diseño. Su principal ventaja radica en que permite abstraer mejor el vocabulario de modelación del problema, además de representar una trazabilidad del modelo de diseño que posteriormente será implementado y una abstracción de la tecnología a implementar. Se describió en un lenguaje más técnico las colaboraciones entre objetos del sistema desarrollándose una visión del sistema centrada en los requisitos funcionales. Se estructuraron los requisitos de manera que se facilite su comprensión y modificación y en general su mantenimiento. La estructura está basada en clases de análisis y paquetes.

3.5.1 Paquetes de Análisis.

Los paquetes de análisis sirven de organización para dividir el modelo en otros más pequeños que colaboran entre sí. La división se hizo sobre la base de los requerimientos funcionales y el dominio del problema. Se agruparon en paquetes los casos de uso requeridos para dar soporte a un determinado proceso de negocio como se muestra en la **Figura 3.3**.

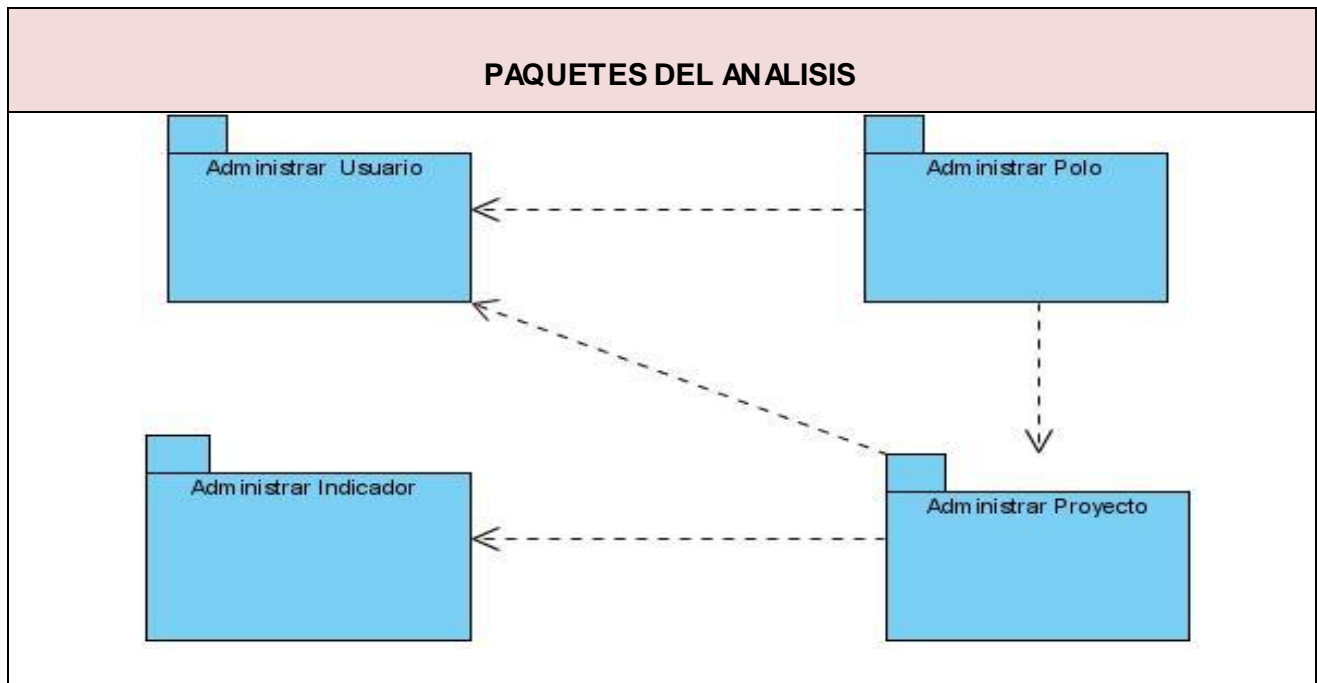


Figura 3.3 Paquetes de Análisis.

3.5.2 Diagrama de clases de Análisis.

Los diagramas de clases del análisis son artefactos en los que se representan los conceptos en el dominio del problema. Se desarrollaron con el objetivo de facilitar la entrada al diseño, por lo que es un paso inicial y una primera aproximación conceptual que en el diseño se puede especificar con más detalle. Las clases de análisis se centraron en los requisitos funcionales, se clasifican en Interfaz, Controladora y Entidad. La clase Interfaz modela la interacción del actor con el sistema, la controladora coordinan las operaciones en el caso de uso y las entidades manejan información de larga duración. A continuación se muestran los diagramas de clases de análisis de cada caso de uso del sistema contenido en los paquetes definidos anteriormente.

Diagramas de clases de Análisis del Paquete Administrar Usuario.

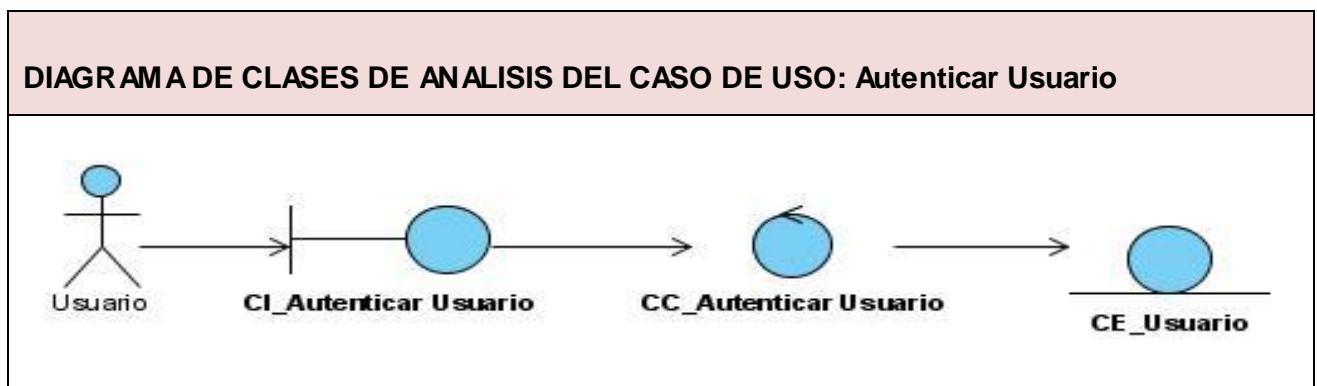


Figura 3.4 Diagrama de clases de análisis del CU Autenticar Usuario.

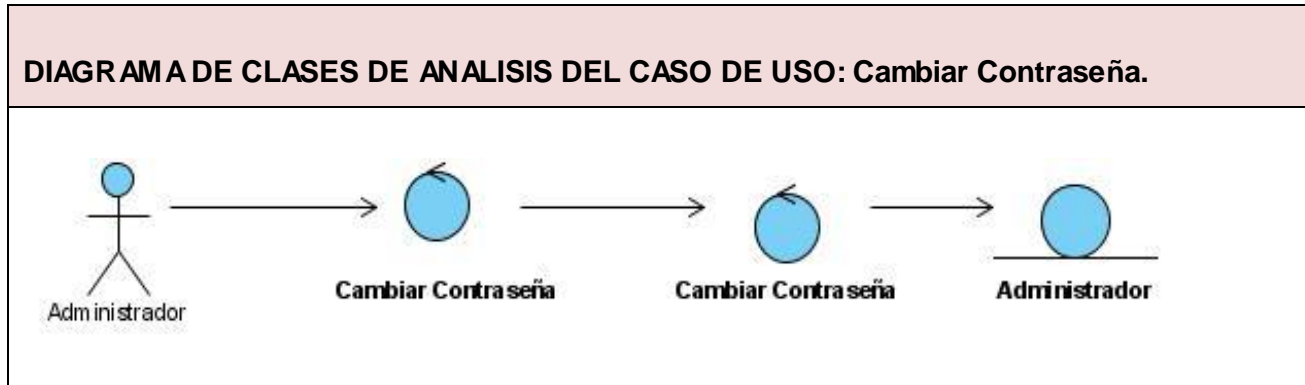


Figura 3.5 Diagrama de clases de análisis del CU Cambiar Contraseña.

Diagramas de clases de Análisis del Paquete Administrar Polo.

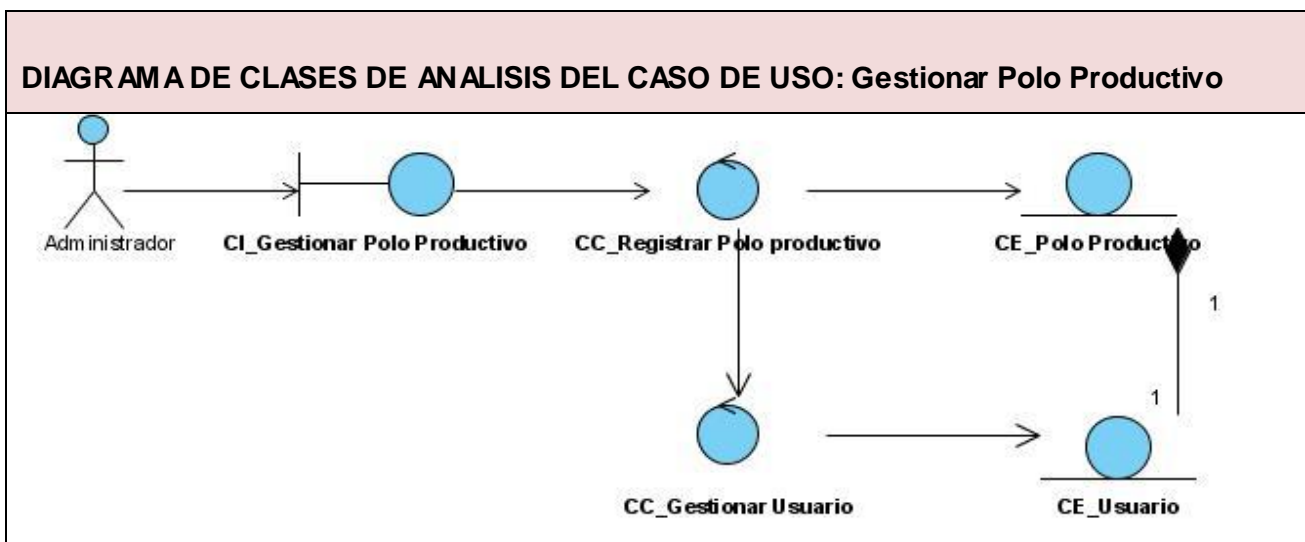


Figura 3.6 Diagrama de clases de análisis del CU Gestionar Polo Productivo.

Diagramas de clases de Análisis del Paquete Administrar Proyecto.

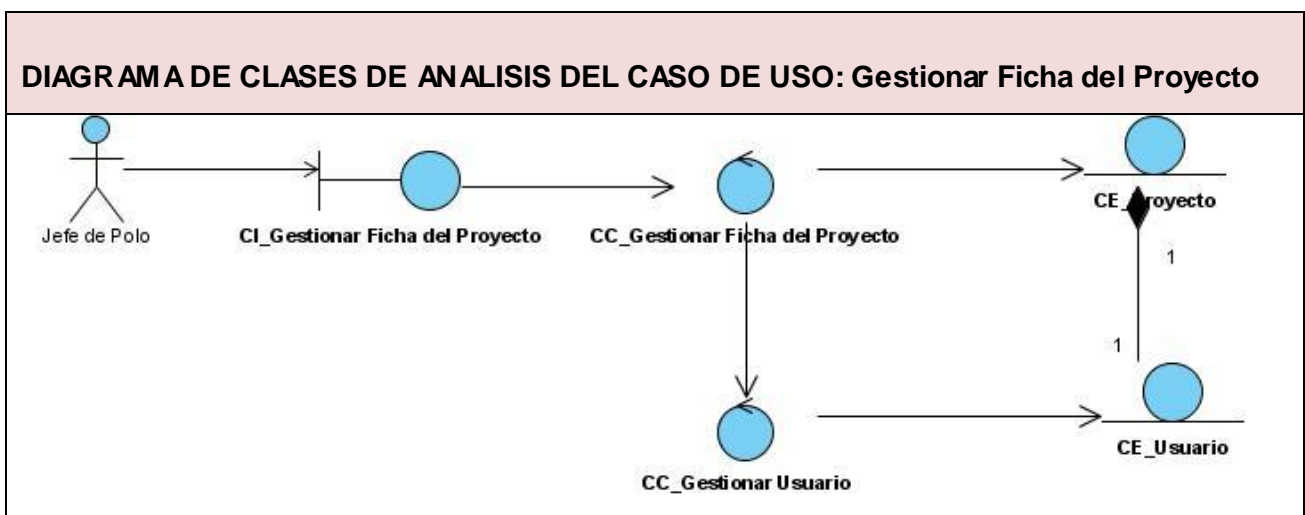


Figura 3.7 Diagrama de clases de análisis del CU Gestionar Ficha del Proyecto.

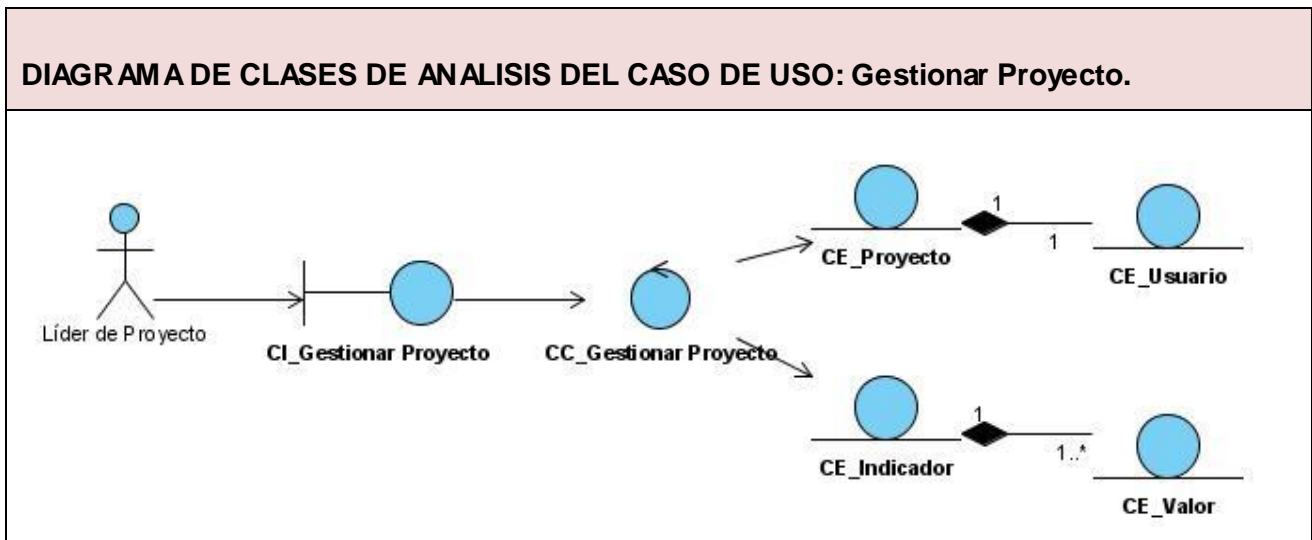


Figura 3.8 Diagrama de clases de análisis del CU Gestionar Proyecto.

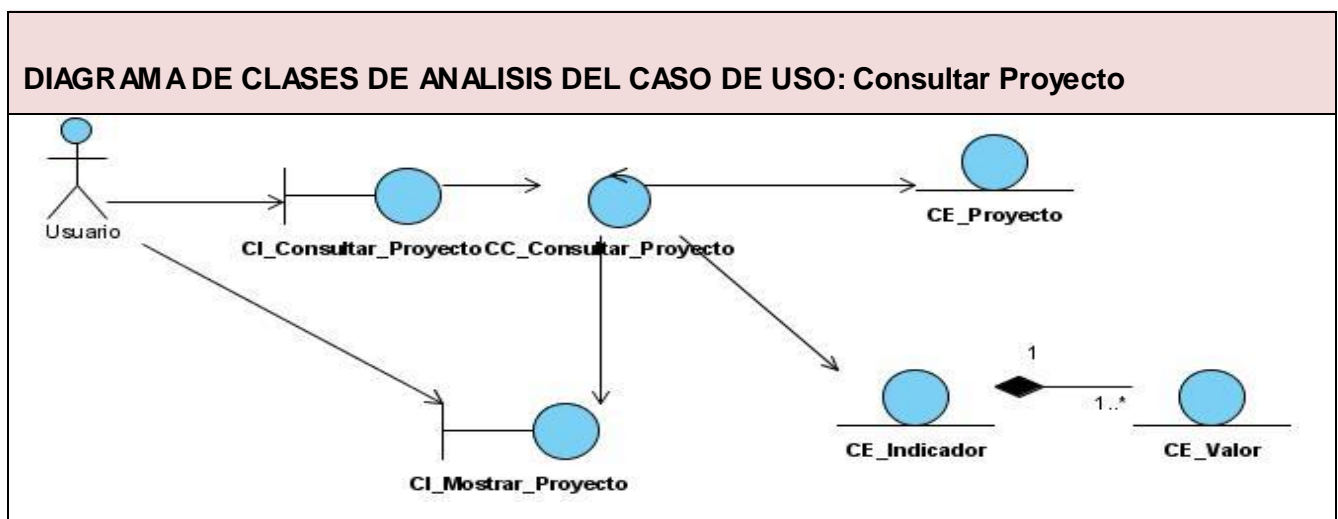


Figura 3.9 Diagrama de clases de análisis del CU Consultar Proyecto.

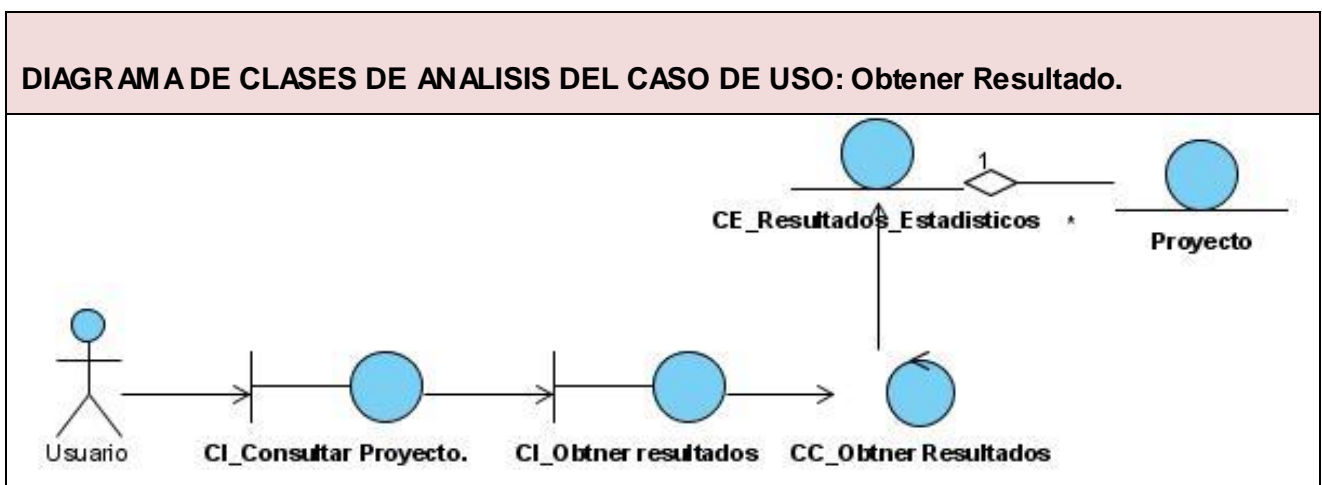


Figura 3.10 Diagrama de clases de análisis del CU Obtener Resultado.

Diagramas de clases de Análisis del Paquete Administrar Indicador.

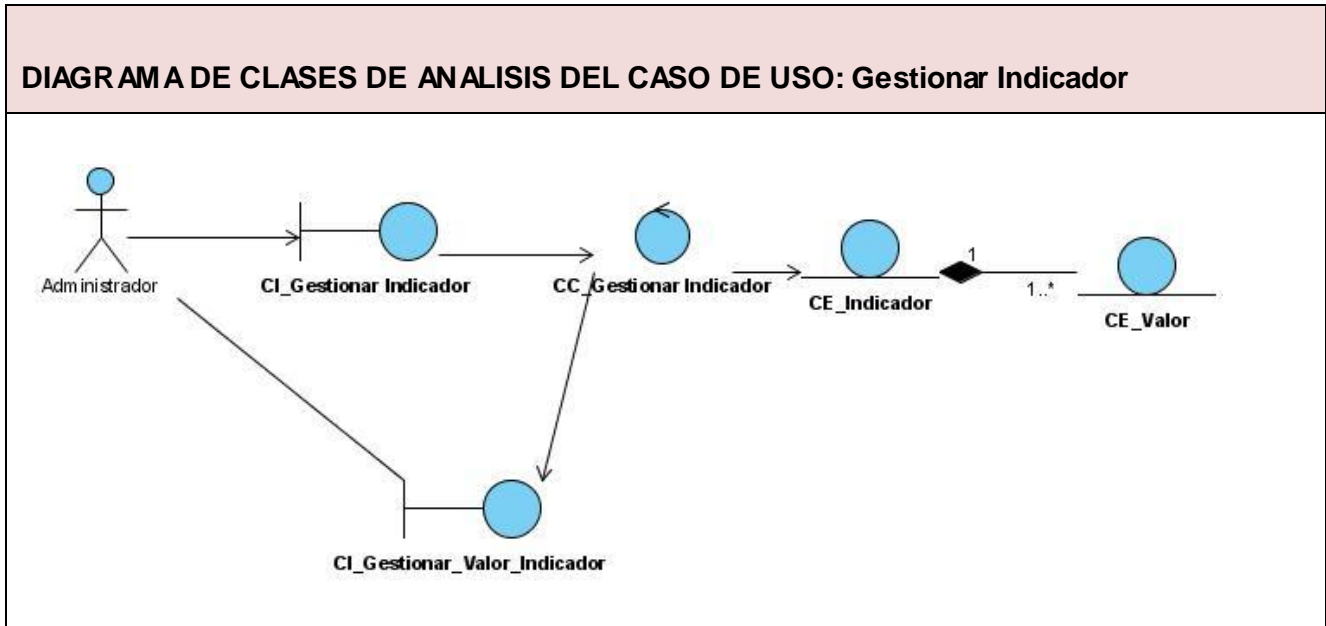


Figura 3.11 Diagrama de clases de análisis del CU Gestionar Indicador.

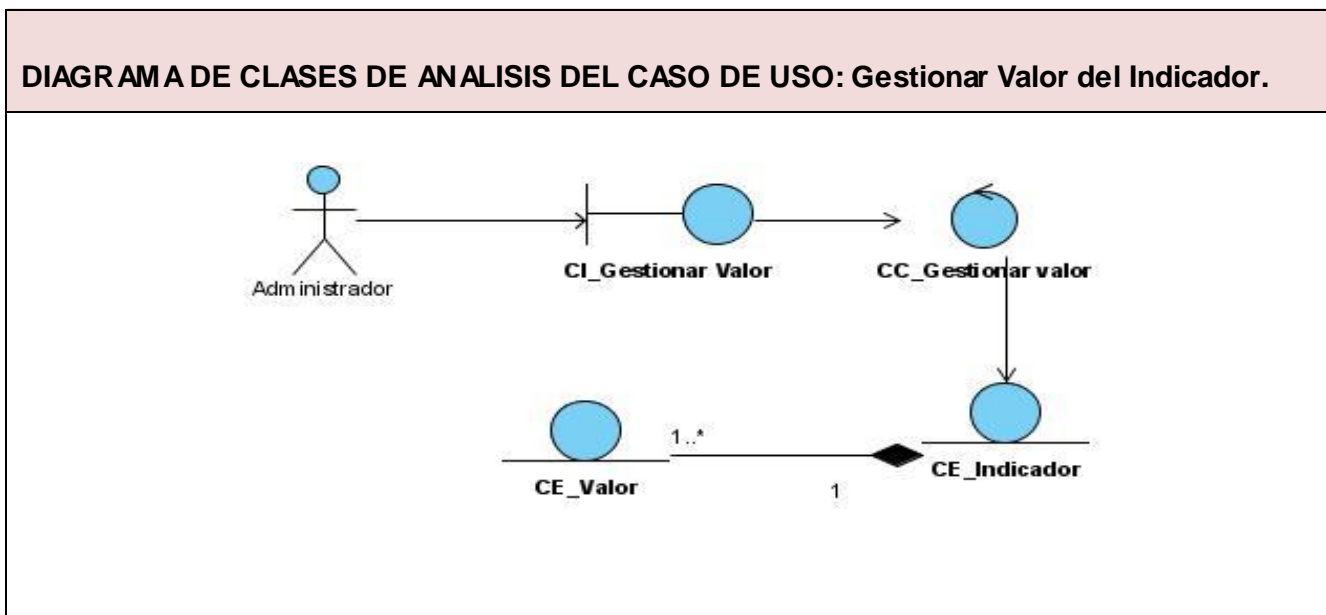


Figura 3.12 Diagrama de clases de análisis del CU Gestionar Valor del Indicador.

3.6 Prototipos de Interfaz de usuario.

Los prototipos de interfaz de usuario ayudan a comprender mejor los casos de uso especificando las interacciones entre actores y el sistema de una forma visual. Durante la captura de requisitos son un apoyo indispensable para la comunicación con el cliente. A continuación se muestra una propuesta de interfaz de usuario para el Caso de Uso Gestionar Proyecto, en el **Anexo 5** aparecen los restantes prototipos de interfaz de usuario.

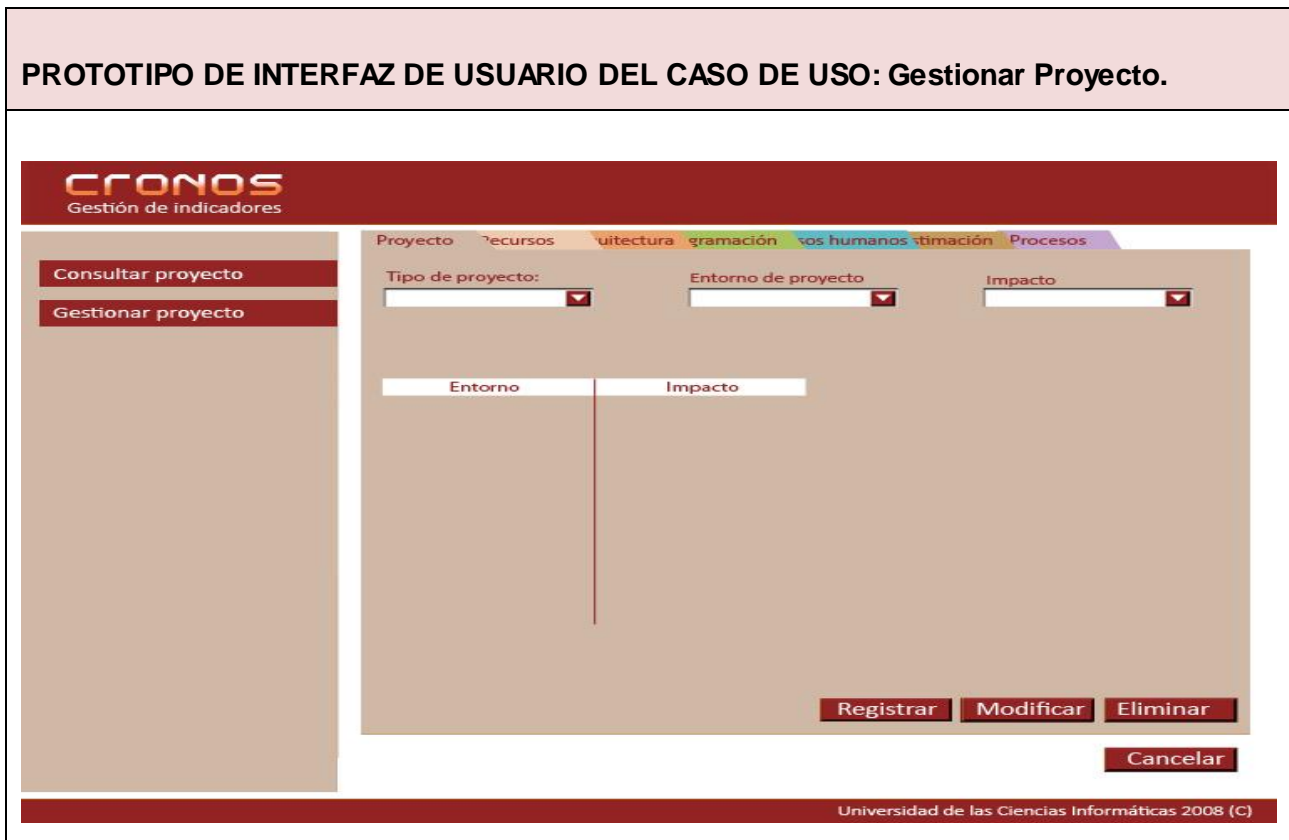


Figura 3.13 Prototipo de Interfaz de usuario del CU Gestionar Proyecto.



Figura 3.14 Prototipo de Interfaz de usuario del CU Gestionar Proyecto.



Figura 3.15 Prototipo de Interfaz de usuario del CU Gestionar Proyecto.



Figura 3.16 Prototipo de Interfaz de usuario del CU Gestionar Proyecto.

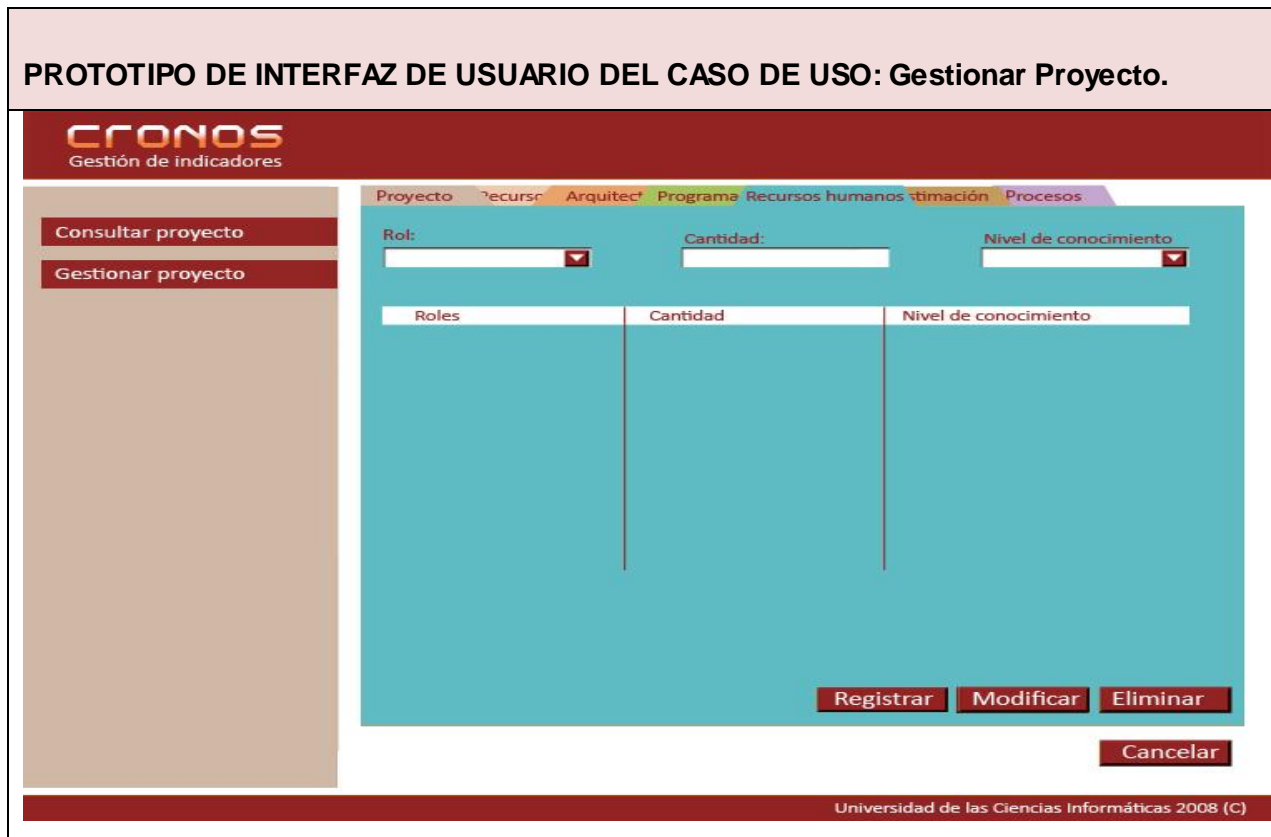


Figura 3.17 Prototipo de Interfaz de usuario del CU Gestionar Proyecto.

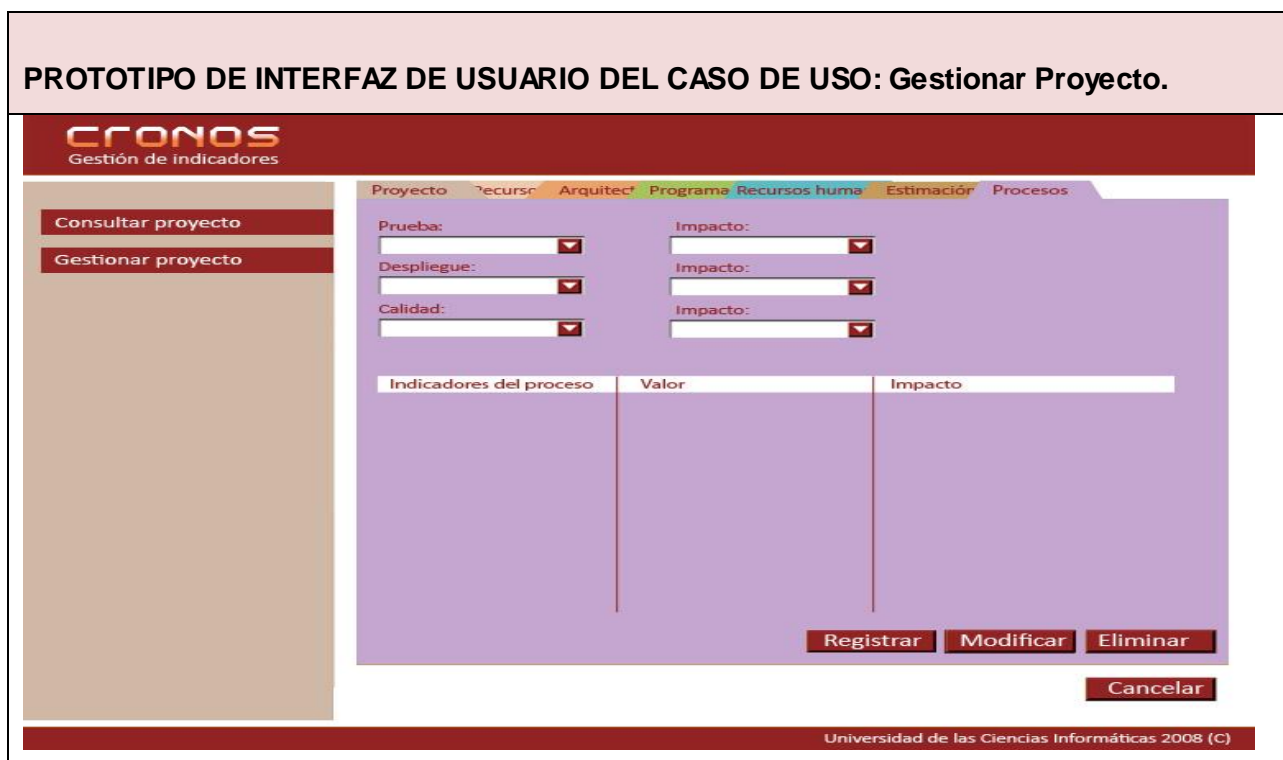


Figura 3.18 Prototipo de Interfaz de usuario del CU Gestionar Proyecto

PROTOTIPO DE INTERFAZ DE USUARIO DEL CASO DE USO: Gestionar Proyecto.

CRONOS
 Gestión de indicadores

Proyecto
Recursos
Arquitect
Programa
Recursos huma
Estimación
Procesos

Consultar proyecto

Gestionar proyecto

Cantidad de iteraciones:	<input type="text"/>		Estimado:	<input type="text"/>
Esfuerzo total de desarrollo:	<input type="text"/>		Estimado:	<input type="text"/>
Tiempo total de desarrollo:	<input type="text"/>		Real:	<input type="text"/>
	Estimado:			<input type="text"/>
Definición:	<input type="text"/>			<input type="text"/>
Desarrollo:	<input type="text"/>			<input type="text"/>
Mantenimiento:	<input type="text"/>			<input type="text"/>
Tamaño producto:	<input type="text"/>		Estimado:	<input type="text"/>
	Estimado:		Real:	<input type="text"/>
Cantidad de procesos de complejidad alta:	<input type="text"/>			<input type="text"/>
Cantidad de procesos de complejidad media:	<input type="text"/>			<input type="text"/>
Cantidad de procesos de complejidad baja:	<input type="text"/>			<input type="text"/>
Influencia de riesgo en la estimación:				
riesgo		Impacto		

Registrar
Modificar
Eliminar

Cancelar

Universidad de las Ciencias Informáticas 2008 (C)

Figura 3.19 Prototipo de Interfaz de usuario del CU Gestionar Proyecto.

3.7 Conclusiones Parciales.

En este capítulo se modeló el negocio mediante un modelo de dominio ya que no existían procesos definidos para encontrar actores y casos de uso, se definieron los requisitos funcionales y no funcionales que el sistema debe de cumplir. Se identificaron los actores y casos de uso del sistema desarrollándose las descripciones textuales de cada caso de uso. Para el análisis, el sistema se dividió en paquetes para su mejor organización. Se elaboraron los diagramas de clases del análisis por cada caso de uso descrito y se definió una propuesta de interfaz de usuario, con todas estas acciones desarrolladas se cumplió uno de los objetivos específicos planteados en la investigación.

CONCLUSIONES

Con el cumplimiento de esta investigación se realizó un estudio de los principales problemas de la producción de software, fundamentalmente en el proceso de estimación de esfuerzo y tiempo en los proyectos, enfocándose en los proyectos productivos de la UCI de forma general. Además se cumplió el objetivo general y los objetivos específicos alcanzando los siguientes resultados:

- Se desarrolló una propuesta de 27 indicadores influyentes en la estimación necesarios para registrar datos de la experiencia acumulada en los proyectos productivos de la Universidad de las Ciencias Informáticas.
- La propuesta de indicadores quedó finalmente validada por un conjunto de 15 expertos otorgándoles la categoría de muy adecuado a 24 indicadores de los 27 propuestos.
- Se propuso un primer análisis estadístico de los indicadores propuestos que permitirá conocer cuales indicadores influyen más en el tiempo y esfuerzo de desarrollo.
- Se obtuvo la modelación de un sistema para la gestión y control de los indicadores de los Proyectos de la Universidad de las Ciencias Informáticas.
- El registro histórico constituye una innovadora solución que apoyará las estimaciones de tiempo y esfuerzo en los proyectos, así como la planificación de los recursos.

RECOMENDACIONES

- Diseñar e implementar completamente la solución propuesta.
- Para el desarrollo de la aplicación se debe garantizar que el sistema sea multiplataforma y que pueda ser desarrollado con software libre. El más adecuado es el PHP por ser libre, multiplataforma, su flexibilidad de comunicación con los principales gestores de bases de datos y por su potencialidad en funcionalidades y rapidez. Entre los principales usos está la programación con información dinámica y posee un fuerte completamiento de código proporcionando ayudas desde la creación y gestión de proyectos hasta la depuración de código.
- Se recomienda como sistema gestor de base de datos PostgreSQL ya que soporta integridad referencial, la cuál es utilizada para garantizar la validez de los datos de la base de datos. Es considerado uno de los sistemas de bases de datos de código abierto más avanzados. Soporta tanto la programación orientada a objetos como la funcional.
- Utilizar Minería de Datos en el sistema cuando se cuente con suficientes proyectos para facilitarle una mejor decisión a los jefes de proyectos a la hora de seleccionar los diferentes recursos que influyen en la estimación y planificación. La Minería de Datos permite extraer información implícita y desconocida a simple vista en grandes cantidades de datos.
- Profundizar en el análisis de los datos mediante el principio de Pareto para definir los problemas que presenta la UCI en el uso de las herramientas y planificación de los recursos identificando los problemas y las causas.
- Profundizar el estudio de los indicadores que influyen en las estimaciones y planificaciones de los proyectos de la UCI en cada una de las etapas de desarrollo del software.

REFERENCIAS BIBLIOGRAFICAS

1. Palacio, J. *Origen de la gestión de proyectos*. 2006 [cited; Available from:http://www.navegapolis.net/files/s/NST-001_01.pdf].
2. Ajenjo, A.D., *Dirección y gestión de proyectos: un enfoque práctico* 2000.
3. Cos, C.M.d., *Teoría General del Proyecto, Volumen I y II*. 2000.
4. PMI, *Introducción al PMBOK* 2008.
5. Suárez, R.C., *Doctorado en dirección de proyectos. Metodología de gestión de proyecto en las administraciones públicas según la ISO 10006*. 2007.
6. Capuchino, A.M.S., *Estimación de Proyectos Software*. . 1998.
7. Públicas., M.d.A. *MÉTRICA. VERSIÓN 3 Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de Información* 2008 [cited; Available from:<http://www.csi.map.es/csi/metrica3/gespro.pdf>].
8. Pressman, R., *Ingeniería de Software, Un enfoque práctico. Cuarta edición*. 1998.
9. IEEE, *Software Measurement and Estimation A Practical Approach*. 2006.
10. Concepción, P. (2008) *Planificación de Proyectos de Software*.
11. Escorial, J., *Calidad de Software: Medidas del Proceso*. 2006.
12. *Resumen del Manual. ESTIMACIÓN DE ESFUERZO CON USC COCOMO II ENTORNO DE LA HERRAMIENTA*. 2007.
13. *PROYECTO MANUAL USUARIO DOTPROYECT*. 2008.
14. Pérez, A.R.M., *¿Qué son los indicadores?* 2005.
15. Granados, H.V.G., *Indicadores de Gestión por procesos. Herramienta básica para el mejoramiento*. . 2008.
16. RUP, *Rational Unified Process*. 2003.
17. Cabrera, *Indicadores de Gestión* 2003.
18. Prado, G.C., *Indicadores de Gestión en Comunicación Estratégica*. 2006.
19. ISO, *ISO 10006. Guía de gestión de calidad. Calidad en Gestión de Proyectos*. 2003.
20. Kaiser, *Documento de IEEE* 2000.
21. Calidad, (2007) *Diagrama de Pareto*.
22. *El método Delphi*. 2007.

BIBLIOGRAFÍA

- 1) CAO, I.J.I., *Principios para un método de estimación de proyectos basado en los escenarios principales.* . 2006.
- 2) Galorath, D.D., *Software Sizing, Estimation and Risk Management.* 2006, United States of America.
- 3) *Gestión de Proyectos.* 2006.
- 4) G., R.P., *Productividad de proyectos de desarrollo de software. Factores determinantes e indicadores.* 1998.
- 5) *Gestión de Indicadores* 2007.
- 6) ISO, *ISO 10006. Guía de gestión de calidad. Calidad en Gestión de Proyectos.* 2003.
- 7) Laird, L.M., *Software Measurement and Estimation A Practical Approach.* 2006.
- 8) Métricas, *Otoniel Pérez Giraldo.* 2002.
- 9) PMI, *Introducción al PMBOK* 2008.
- 10) RUMBAUGH, *El proceso unificado de desarrollo de software.* 2006.
- 11) Sevilla, U.d. *Gestión de Proyectos Software Estimación.* 2006.
- 12) Suárez, R.C., *Doctorado en dirección de proyectos. Metodología de gestión de proyecto en las administraciones públicas según la ISO 10006.* 2007.
- 13) Varas, M., *Una Experiencia con la Estimación del Tamaño del Software.* 1998.
- 14) Varas, M.P., *Modelo de Gestión de Proyectos Software: Estimación del Esfuerzo de Desarrollo.* 1995.

ANEXOS

Anexo 1 Listado de Expertos que colaboraron.

No.	Nombre y Apellidos	Labor que realiza	Año de experiencia	Especialidad
1	Saumel Tejada Diaz	Líder de Proyecto (MINFAR)	3	Ing. Informático
2	Madelín Haro Pérez	Líder de Proyecto (SIGEP) Asesora de departamento de Preparación Profesional(PP)	6	Ing. Informático
3	Henry Raúl González Brito.	Líder de Proyecto ERP-Cuba	3	Ing. Informático
4	Yaillet Martínez Pérez	Especialista Superior de la dirección de Producción.	5	Ing. Informático
5	Yenín Calderín Abad	Jefe de de Despliegue de Capacitación, fue Líder del Proyecto MINFAR, Máster Informática Aplicada	4	Ing. Informática
6	Manuel Alejandro Gil Martín	Jefe de Grupo de Plataforma de la Dirección de Informatización.	3	Ing. Informático
7	Armando Ortiz Cabrera	Analista Principal y Arquitecto del Proyecto SACGIR de la Facultad 9.	1	Ing. Informático
8	Abel Meneses Abad	Líder de Proyecto Unicornios.	3	Ingeniero en Telecomunicaciones y

No.	Nombre y Apellidos	Labor que realiza	Año de experiencia	Especialidad
				Electrónica.
9	Yoandro Echevarría Toranzo	Líder de Proyecto de ERP- MINFAR	3	Ing. Informático
10	Julio Diaz Vera	Líder de Proyecto (ADUANA) Jefe de Departamento de Ingeniería de Software facultad 4.	5	Ing. en Telecomunicaciones.
11	Yadira Ruiz Constanten	Jefe de Departamento de ISW, Máster en Gestión de Proyecto.	4	Lic. Ciencias de la Computación.
12	Irina Cancela	Líder de Proyecto de CICPC de la Facultad 8	3	Ing. Informático
13	Karel Tamaño Peña	Líder de Proyecto Alas RIS	1	Ing. Informático
14	Osay González	Líder de Proyecto Arquitectura Facultad 1	1	Ing. Informático
15	Alain Eduardo Rodriguez Arias	Asesor de Arquitectura y Tecnología	3	Ing. Informático

Anexo 2 Cálculo del coeficiente de conocimiento de los expertos.

No	Nombre del Experto	P1	P 2						Ka	Kc	K	Competencia
		Con	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6				
1	Saumel Tejada Diaz	6	0.2	0.2	0.05	0.05	0.05	0.05	0.6	0.6	0.6	MEDIO
2	Karel Tamayo Peña	5	0.2	0.2	0.05	0.05	0.05	0.05	0.6	0.5	0.55	MEDIO
3	Julio Diaz Vera	9	0.3	0.5	0.05	0.05	0.05	0.05	1	0.9	0.95	ALTO
4	Yaillet Martínez	6	0.3	0.5	0.05	0.05	0.05	0.05	1	0.6	0.8	ALTO

		P1	P 2									
No	Nombre del Experto	Con	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6	Ka	Kc	K	Competencia
	Pérez											
5	Madelín Haro Pérez	6	0.3	0.5	0.05	0.05	0.05	0.05	1	0.6	0.8	ALTO
6	Manuel Alejandro Gil Martín	7	0.3	0.5	0.05	0.05	0.05	0.05	1	0.7	0.85	ALTO
7	Yenin Calderín Abad	7	0.3	0.5	0.05	0.05	0.05	0.05	1	0.7	0.85	ALTO
8	Yadira Ruiz Constanten	7	0.3	0.4	0.05	0.05	0.05	0.05	0.9	0.7	0.8	ALTO
9	Yoandro Hechevarría Toranzo	8	0.3	0.5	0.05	0.05	0.05	0.05	1	0.8	0.9	ALTO
10	Irina Cancela Nieto	7	0.3	0.5	0.05	0.05	0.05	0.05	1	0.7	0.85	ALTO
11	Alain Eduardo Rodríguez Arias	5	0.3	0.4	0.05	0.05	0.05	0.05	0.9	0.5	0.7	MEDIO
12	Henry Raúl Gonzales Brito	7	0.3	0.5	0.05	0.05	0.05	0.05	1	0.7	0.85	ALTO
13	Abel Meneses Abad	7	0.2	0.5	0.05	0.05	0.05	0.05	0.9	0.7	0.8	ALTO
14	Armando Ortiz Cabrera	7	0.2	0.5	0.05	0.05	0.05	0.05	0.9	0.7	0.8	ALTO
15	Osay González	6	0.3	0.5	0.05	0.05	0.05	0.05	1	0.6	0.8	ALTO
		6.666 67	0.27	0.45	0.05	0.05	0.05	0.05	0.92	0.67	0.79	ALTO

Anexo 3 Resultado de los puntos de corte para los indicadores.

MA: Muy Adecuado, BA: Bastante Adecuado

Puntos de corte										N= 1.02
No	Aspectos	C1	C2	C3	C4	C5	Suma	P	N-P	
1	Tipo de proyecto	-0.08	0.25	1.50	3.72	5.39	1.35	-0.36	-0.08	MA
2	Entorno del proyecto	0.25	0.84	3.72		4.81	1.60	-0.61	0.25	MA
3	Gestión de la Calidad	0.08	1.11	3.72			4.91	1.64	-0.65	MA
4	Pruebas de Software	-0.08	1.50	3.72			5.14	1.71	-0.72	MA
5	Despliegue de la aplicación	-0.25	1.11	3.72			4.58	1.53	-0.54	MA
6	Metodología de desarrollo de software	0.25	1.11	3.72		5.08	1.69	-0.70	0.25	MA
7	Herramienta de modelado visual	-0.84	-0.43	0.84	3.72	3.29	0.82	0.17	-0.84	BA
8	Lenguaje de Programación	-0.62	0.62	3.72		3.72	1.24	-0.25	-0.62	MA
9	Herramienta de Programación	-0.62	0.62	3.72		3.72	1.24	-0.25	-0.62	MA
10	Herramienta sistema Gestor Base de Datos	-0.43	0.25	1.50	3.72	5.04	1.26	-0.27	-0.43	MA
11	Herramientas utilizadas en la Planificación	-0.25		0.84	3.72	4.31	1.44	-0.44	-0.25	MA
12	Herramientas utilizadas en la Gestión de Configuración	-1.11	0.25	1.11	3.72	3.97	0.99	0.00	-1.11	BA
13	Cantidad de	-0.08	0.62	3.72		4.26	1.42	-1.42	-0.08	MA

Puntos de corte										N= 1.02
No	Aspectos	C1	C2	C3	C4	C5	Suma	P	N-P	
	integrantes del equipo de desarrollo									
14	Cantidad de PC	-0.84	0.43	1.50	3.72	4.81	1.20	-0.21	-0.84	MA
15	Nivel de conocimiento de los integrantes del equipo	-0.08	0.62	1.50	3.72	5.76	1.44	-0.45	-0.08	MA
16	Componentes Reutilizables	-0.25	0.84	1.11	3.72	5.42	1.35	-0.36	-0.25	MA
17	Estilo Arquitectónico	-0.08	0.84	3.72			4.48	1.49	-2.07	MA
18	Patrón de Arquitectura	0.25	1.11	1.50	3.72	6.58	1.65	-0.65	0.25	MA
19	Herramienta utilizada en estimación	0.08	0.43	1.50	3.72	5.73	1.43	-0.44	0.08	MA
20	Cantidad de versiones	-0.84	0.25	1.50	3.72	4.63	1.16	-0.17	-0.84	BA
21	Esfuerzo de desarrollo	-0.08	0.84	1.50	3.72	5.98	1.49	-0.50	-0.08	MA
22	Tiempo de desarrollo	0.25	0.84	3.72		4.81	1.60	-0.61	0.25	MA
23	Tamaño del producto	0.08	0.84	3.72		4.64	1.55	-0.56	0.08	MA
24	Cantidad de procesos de complejidad de alta	-0.08	0.84	3.72		4.48	1.49	-0.50	-0.08	MA
25	Cantidad de procesos de complejidad de	0.25	0.84	1.50	3.72	6.32	1.58	-0.59	0.25	MA

Puntos de corte										N= 1.02	
No	Aspectos	C1	C2	C3	C4	C5	Suma	P	N-P		
	media										
26	Cantidad de procesos de complejidad de baja	0.08	1.11	1.50	3.72	6.41	1.60	-0.61	0.08	MA	
27	Influencia de los Riesgos en la estimación	0.25	0.84	3.72		4.81	1.60	-0.61	0.25	MA	

Anexo 4 Descripciones Textuales de los Casos de uso.

Nombre del CU	Gestionar Usuario.	
Actores	Jefe de Polo, Administrador	
Propósito	Permitir registrar, modificar, o eliminar un usuario.	
Resumen	El caso de uso se inicia cuando el sistema brinda la posibilidad de registrar, modificar, mostrar o eliminar un usuario. Si el actor selecciona registrar un usuario el sistema permite introducir los datos, si el actor desea modificarlos el sistema muestra los datos en forma editable, si el actor desea eliminar un usuario el sistema elimina el usuario.	
Referencias	RF1	
CU Asociados		
Precondiciones		
Poscondiciones	Se registró, mostró, eliminó ó modificó un usuario.	
Flujo Básico		
Acciones del Actor	Respuesta del Sistema	
	<p>1. El sistema brinda la posibilidad de introducir el nombre, el usuario y el solapín. Brinda la posibilidad de:</p> <ul style="list-style-type: none"> • Completar los datos. • Registrar los datos del usuario con el rol que juega en la aplicación (jefe de polo o líder de proyecto.), junto con el proyecto que se crea o con el polo que se crea. • Cancelar la operación. <p>El Sistema permite:</p> <ul style="list-style-type: none"> • Modificar Usuario(Ver sección 1) • Eliminar Usuario (Ver sección 2) 	

2. El actor introduce los datos completos del usuario y selecciona la opción de registrar	
	3. El sistema valida los datos.
	4. El sistema registra los datos.
Curso Alterno	
2. El actor selecciona la opción cancelar.	
	2.1 El sistema cancela la operación.
2a. El actor introduce el nombre y el usuario y selecciona la opción de completar los datos.	
	2a.1 El sistema valida los datos.
	2a.2 El sistema completa los datos requeridos.
	2a.1 El sistema muestra un mensaje: "Existen datos incorrectos, verifique los datos."
2b. El actor introduce el nombre y el solapín y selecciona la opción de completar los datos.	
	2b.1 El sistema valida los datos.
	2b.2 El sistema completa los datos requeridos.
	2b.1 El sistema muestra un mensaje: "Existen datos incorrectos, verifique los datos."
2c. El actor introduce el solapín y selecciona la opción de completar los datos.	
	2c.1 El sistema valida los datos.
	2c.2 El sistema completa los datos requeridos.
	2c.1 El sistema muestra un mensaje: "Existen datos incorrectos, verifique los"

	datos.”
2d. El actor introduce el usuario y selecciona la opción de completar los datos.	
	2d.1 El sistema valida los datos.
	2d.2 El sistema completa los datos requeridos.
	2d.1 El sistema muestra un mensaje: “Existen datos incorrectos, verifique los datos.”
	3. El sistema muestra un mensaje: “Existen datos incorrectos, verifique los datos.”
Sección 1 Modificar Usuario.	
Acciones de actor	Acciones del Sistema
	1. El sistema brinda la posibilidad de mostrar los datos del usuario: <ul style="list-style-type: none"> • Usuario • El nombre • solapín El sistema permite: <ul style="list-style-type: none"> • Modificar el usuario. • Cancelar la operación.
2. El actor selecciona la opción modificar.	
	3. El sistema habilita los campos del usuario. Permite actualizar los datos.
4. El actor modifica los datos y selecciona la opción actualizar.	
	5. El sistema valida los datos.
	6. El sistema actualiza los datos.
Curso Alterno	
4a. El actor modifica el nombre y el usuario, si desconoce los restantes campos selecciona la	

opción completar los datos.	
	2a.1 El sistema valida los datos.
	2a.2 El sistema completa los datos requeridos.
	2a.1 El sistema muestra un mensaje: "Existen datos incorrectos, verifique los datos."
2b. El actor modifica el nombre y el solapín, si desconoce los restantes campos selecciona la opción completar los datos.	
	2b.1 El sistema valida los datos.
	2b.2 El sistema completa los datos requeridos.
	2b.1 El sistema muestra un mensaje: "Existen datos incorrectos, verifique los datos."
2c. El actor modifica el solapín, si desconoce los restantes campos selecciona la opción completar los datos.	
	2c.1 El sistema valida los datos.
	2c.2 El sistema completa los datos requeridos.
	2c.1 El sistema muestra un mensaje: "Existen datos incorrectos, verifique los datos."
2d. El actor modifica el usuario, si desconoce los restantes campos selecciona la opción completar los datos.	
	2d.1 El sistema valida los datos.
	2d.2 El sistema completa los datos requeridos.

	2d.1 El sistema muestra un mensaje: “Existen datos incorrectos, verifique los datos.”
	5. El sistema muestra un mensaje: “Existen datos incorrectos, verifique los datos.”
Sección 2 Eliminar Usuario	
Acciones de actor	Acciones del Sistema
	1. El sistema muestra los datos del usuario <ul style="list-style-type: none"> • Usuario • El nombre • Solapín El sistema permite: <ul style="list-style-type: none"> • Eliminar el usuario • Cancelar la Operación.
2. El actor selecciona la opción de eliminar.	
	3. El sistema muestra un mensaje: ¿Está seguro que desea eliminar el usuario?
4. El actor confirma eliminar el usuario.	
	5. El sistema elimina el usuario.
Curso Alterno	
4. El actor selecciona la opción que no está seguro.	
	4.1 El sistema cancela la operación.
Prioridad: Crítico.	

Nombre del CU	Autenticar Usuario	
Actores	Usuario(Inicia)	
Propósito	Autenticar el usuario en el sistema.	
Resumen	El caso de uso se inicia cuando el usuario accede a la interfaz de ingresar en el sistema, el sistema muestra opciones para insertar el usuario y contraseña, el usuario introduce los datos solicitados, el sistema consulta internamente que existe correspondencia entre los datos introducidos y los datos almacenados y carga la configuración del sistema definida para el usuario.	
Referencias	RF1	
CU Asociados		
Precondiciones		
Poscondiciones	Usuario autenticado	
Flujo Básico		
Acciones del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el actor accede a la interfaz de ingresar en el sistema.		
	2. El sistema solicita los siguientes datos: <ul style="list-style-type: none"> • Usuario • Contraseña 	
3. El actor introduce su usuario y la contraseña.		
	4. El sistema verifica los datos del dominio.	
	5. El sistema verifica que el usuario está registrado en el sistema.	
	6. El sistema verifica el rol del usuario en el sistema.	
	7. Si el rol autenticado es el de Administrador	

	<p>el sistema carga la configuración correspondiente a este rol. Tiene permiso a acceder a:</p> <ul style="list-style-type: none"> • Gestionar Indicador. • Gestionar Polo Productivo. • Cambiar Contraseña. • Consultar Proyecto.
	<p>8. Muestra el usuario de la persona que se encuentra autenticada y la interfaz correspondiente de bienvenida al sistema.</p>
Curso Alterno	
	<p>4a. El usuario no se encuentra registrado en el sistema y se muestra un mensaje de error "El usuario no se encuentra registrado en el sistema".</p>
	<p>4b. La contraseña es introducida incorrectamente. Muestra mensaje de error "La contraseña introducida es incorrecta".</p>
	<p>4c. Existen campos vacíos. Muestra un mensaje "Existen campos en blanco".</p>
	<p>5a. El sistema muestra un sistema: "El usuario no se encuentra en el sistema."</p>
	<p>7a. Si el rol autenticado es el de Jefe de Proyecto el sistema carga la configuración correspondiente a este rol. Tiene permiso a acceder a:</p> <ul style="list-style-type: none"> • Gestionar Ficha de Proyecto. • Consultar Proyecto (Permite consultar los datos solamente de su proyecto).
	<p>7b. Si el rol autenticado es el de Jefe de Polo el sistema carga la configuración correspondiente a este rol. Tiene permiso a acceder a:</p>

	<ul style="list-style-type: none">• Gestionar Indicador.• Gestionar Ficha de Proyecto.• Consultar Proyecto (Permite consultar los datos de los proyectos de su polo).
Prioridad: Crítico	

Nombre del CU	Cambiar Contraseña	
Actores	Administrador(inicia)	
Propósito	Permitir cambiar la contraseña del administrador.	
Resumen	El caso de uso se inicia cuando el actor selecciona cambiar la contraseña, registra la contraseña anterior y la nueva con su confirmación, y selecciona la opción de cambiar contraseña.	
Referencias	RF10	
CU Asociados		
Precondiciones	El Administrador debe de estar autenticado.	
Poscondiciones	Se cambió la contraseña del administrador.	
Flujo Básico		
Acciones del Actor	Respuesta del Sistema	
1. El actor selecciona la opción de cambiar contraseña.		
	2. El sistema muestra los campos que debe de llenar (contraseña anterior, nueva contraseña, confirmación de contraseña). El sistema permite : <ul style="list-style-type: none"> • Cancelar la operación. • Cambiar la contraseña. 	
3. El actor introduce los datos y selecciona la opción de cambiar la contraseña.		
	4. El sistema verifica que coincida la contraseña anterior.	
	5. El sistema verifica que coincidan la contraseña nueva y la confirmación de esta.	

	6. El CU se termina.
Curso Alterno	
3. El actor selecciona la opción cancelar.	
	3.1 El sistema cancela la operación. Regresar al paso 2 del flujo básico.
	4. El sistema muestra un mensaje: “La contraseña anterior es incorrecta”
	5. El sistema muestra un mensaje: “No coincide la contraseña nueva y la confirmación”.
Prioridad: Secundario.	

Nombre del CU	Consultar Proyecto	
Actores	Usuario (inicia)	
Propósito	Mostrar los datos de un proyecto seleccionado.	
Resumen	El caso de uso se inicia cuando el actor accede a la opción que le permite consultar los datos de un proyecto determinado. El sistema solicita criterios de búsqueda que el actor debe de introducir, el sistema brinda un listado de proyectos que se correspondan con los criterios introducidos por el actor. El sistema permite seleccionar un proyecto para ver sus detalles. El actor selecciona un proyecto para ver sus datos y el caso de uso termina.	
Referencias	RF9	
CU Asociados		
Precondiciones	El usuario debe de estar autenticado.	
Poscondiciones	Se consultó la información de los proyectos.	
Flujo Básico		
Acciones del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el actor accede a la opción de consultar los datos de un proyecto.		
	<p>2. El sistema brinda la posibilidad de seleccionar los criterios elementales de búsqueda:</p> <ul style="list-style-type: none"> • Tipo de Proyecto. <p>Y permite:</p> <ul style="list-style-type: none"> • Realizar una búsqueda a partir de los datos introducidos. • Realizar una búsqueda avanzada. 	

	<ul style="list-style-type: none"> • Cancelar la operación. • Obtener Resultados Estadísticos.
<p>3. El actor introduce opcionalmente los criterios de búsqueda elementales.</p> <ul style="list-style-type: none"> • Tipo de Proyecto. <p>Y selecciona la opción de realizar una búsqueda a partir de los datos introducidos.</p>	
	4. El sistema valida los datos.
	5. El sistema consulta los datos y muestra una lista de posibles coincidencias mostrando los campos de tipo de proyecto, nombre del proyecto y líder de proyecto.
6. El actor selecciona un proyecto de la lista de coincidencias para ver sus datos.	
	<p>7. El sistema muestra los siguientes datos del proyecto seleccionado.</p> <ul style="list-style-type: none"> • Tipo de Proyecto. • Entorno de Proyecto. • Gestión de la Calidad. • Pruebas de Software. • Despliegue de la Aplicación. • Metodología de desarrollo de software • Herramienta de modelado visual. • Lenguaje de Programación. • Herramienta de Programación. • Herramienta sistema Gestor Base de Datos. • Herramientas utilizadas en la

	<p>Planificación.</p> <ul style="list-style-type: none"> • Herramientas utilizadas en la Gestión de Configuración. • Cantidad de integrantes del equipo de desarrollo • Cantidad de PC. • Nivel de conocimiento de los integrantes del equipo. • Componentes Reutilizables Utilizados • Estilo Arquitectónico. • Patrón Arquitectura utilizado. • Herramientas utilizadas en estimación • Cantidad Versiones. • Esfuerzo de desarrollo (horas/hombres) • Tiempo de desarrollo (meses). • Tamaño del producto (Cantidad de Caso de Uso). • Cantidad de procesos de complejidad de alta. • Cantidad de procesos de complejidad de media. • Cantidad de procesos de complejidad de baja. <p>El sistema permite:</p> <ul style="list-style-type: none"> • Realizar una nueva búsqueda.
Curso Alterno	
2a. El actor selecciona la opción de cancelar.	
	2.1a El sistema cancela la operación.
2b. El actor selecciona la opción de búsqueda avanzada.	

2c. El actor selecciona obtener resultados estadísticos. (Ver a CU extendido Obtener Resultados)	
	<p>2.1 b El sistema muestra los siguientes criterios de búsqueda.</p> <p>Proyecto:</p> <ul style="list-style-type: none"> • Tipo de Proyecto • Entorno de Proyecto. <p>Proceso</p> <ul style="list-style-type: none"> • Gestión de la Calidad • Pruebas de Software • Despliegue de la Aplicación <p>Recursos:</p> <ul style="list-style-type: none"> • Metodología de desarrollo de software. • Herramienta de modelado visual. • Herramienta sistema Gestor Base de Datos. • Herramientas utilizadas en la Planificación. • Herramientas utilizadas en la Gestión de Configuración. • Cantidad de integrantes del equipo de desarrollo. • Cantidad de PC. • Nivel de conocimiento de los integrantes del equipo. • Componentes Reutilizables Utilizados • Herramientas utilizadas en estimación

	<ul style="list-style-type: none"> • Influencia de los riesgos en la estimación. <p>Programación:</p> <ul style="list-style-type: none"> • Lenguaje de Programación. • Herramienta de Programación. <p>Arquitectura:</p> <ul style="list-style-type: none"> • Patrón Arquitectura. • Estilo Arquitectónico. <p>Estimación:</p> <ul style="list-style-type: none"> • Cantidad de Versiones. • Esfuerzo de desarrollo (horas/hombres) • Tiempo de desarrollo (meses). • Tamaño del producto (Cantidad de Caso de Uso). • Cantidad de procesos de complejidad de alta. • Cantidad de procesos de complejidad de media. • Cantidad de procesos de complejidad de baja.
	<p>3a. El sistema muestra el siguiente mensaje indicando que existen datos incorrectos:” Existen datos incorrectos. Por favor verifique los datos”.</p>
	<p>3b. El sistema muestra el mensaje de información indicando que no se introdujo ningún criterio de búsqueda “Debe especificar un criterio de búsqueda”.</p>
<p>6. El actor selecciona realizar una nueva búsqueda.</p>	

	6.1 Regresar al paso 2 del flujo básico.
Prioridad: Crítico.	

Nombre del CU	Gestionar Ficha de Proyecto	
Actores	Jefe de Polo (Inicia)	
Propósito	Permitir gestionar la ficha del proyecto, registrando un proyecto y asignándole el líder.	
Resumen	El caso de uso se inicia cuando el actor desea registrar, modificar o eliminar una Ficha del Proyecto, si selecciona registrar una nueva ficha introduce el nombre del proyecto y la facultad a que pertenece y registra el líder del proyecto, si es modificar o eliminar selecciona uno de los proyectos que ya están registrados y procede a modificar o eliminar su ficha.	
Referencias	RF8	
CU Asociados	Gestionar Usuario(Incluido)	
Precondiciones	El Jefe de Polo debe de estar autenticado.	
Poscondiciones	Se definió la ficha del proyecto con el nombre del proyecto y el líder del proyecto.	
Flujo Básico		
Acciones del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el actor selecciona la opción de gestionar la Ficha de un proyecto.		
	2. El sistema brinda la posibilidad de: <ul style="list-style-type: none"> • Registrar una nueva Ficha de Proyecto. • El Sistema permite: • Modificar Ficha de Proyecto(Ver Sección 1) • Eliminar Ficha de Proyecto(Ver Sección 2) • Cancelar la operación. 	
3. El actor selecciona registrar una Ficha de		

Proyecto.	
	4. El sistema brinda la posibilidad de introducir el nombre del proyecto, facultad del proyecto. Ir al CU incluido Gestionar Usuario.
5. El usuario introduce los datos de la ficha del proyecto.	
	6. El sistema valida los datos.
	7. El sistema crea una ficha del proyecto nuevo registrando el líder del proyecto que lo dirige.
	8. El sistema muestra un mensaje: "Los datos han sido registrados correctamente "
Curso Alterno	
2. El actor selecciona la opción cancelar.	
	2.1 El sistema cancela la operación. Regresar al paso 2 del flujo básico.
	6. El sistema muestra el siguiente mensaje indicando que existen datos incorrectos:" Existen datos incorrectos. Por favor verifique los datos".
Sección 1 Modificar Ficha del Proyecto	
1. El actor selecciona la ficha del proyecto que desea modificar.	
	2. El sistema muestra los datos de la ficha seleccionada. El Sistema permite: <ul style="list-style-type: none"> • Modificar los datos • Cancelar la operación
3. El actor selecciona la opción que le permite modificar los datos de la ficha del proyecto.	
	4. El sistema permite introducir los nuevos

	datos.
5.El actor modifica los datos	
	6. El sistema valida los nuevos datos.
	7. El sistema muestra un mensaje: “Los datos han sido actualizados correctamente ”
Curso Alterno	
1. El actor selecciona la opción de cancelar.	
	1.1 El sistema cancela la operación.
	6a. El sistema muestra el siguiente mensaje indicando que existen datos incorrectos:” Existen datos incorrectos. Por favor verifique los datos”.
	6b. El sistema muestra el mensaje de información indicando que no se introdujo ningún criterio de búsqueda “Debe especificar un criterio de búsqueda”.
Sección 2 Eliminar Ficha del Proyecto	
1. El actor selecciona la ficha del proyecto que desea eliminar.	
	2. El sistema muestra los datos de la ficha del proyecto seleccionada. El Sistema permite: <ul style="list-style-type: none"> • Modificar Datos • Cancelar Operación
3. El actor selecciona la ficha del proyecto y selecciona eliminar.	
	4. El sistema muestra los valores de la ficha del proyecto. Ir al CU incluido Gestionar Usuario sección 2.
	5. El sistema muestra un mensaje de confirmación: ¿Está seguro que desea eliminar la ficha del proyecto?

6. El actor selecciona aceptar para eliminar la ficha del proyecto.	
	7. El sistema elimina la ficha del proyecto.
Curso Alterno	
1. El actor selecciona la opción cancelar.	
	1.1 El sistema cancela la operación.
6. El actor selecciona la opción de no eliminar la ficha del proyecto.	
	6.1 El sistema cancela la operación.
Prioridad: Crítico.	

Nombre del CU	Gestionar Indicador	
Actores	Administrador (inicia)	
Propósito	Permitir registrar, modificar, o eliminar los valores de un indicador.	
Resumen	El caso de uso se inicia cuando el Administrador accede a la opción que le permite registrar, modificar o eliminar un indicador. Cuando el Administrador selecciona registrar un nuevo indicador introduce el nombre del indicador y una descripción luego introduce los valores de este indicador, cuando selecciona modificar un indicador el actor selecciona los datos y los actualiza, cuando elige eliminar un indicador busca el indicador que desea eliminar y lo elimina con todos sus valores.	
Referencias	RF4	
CU Asociados	Gestionar Valor del Indicador (CU Incluido)	
Precondiciones	El usuario debe de estar autenticado.	
Poscondiciones	Se registró, eliminó ó modificó un Indicador.	
Flujo Básico		
Acciones del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el actor selecciona la opción de gestionar un indicador.		
	2. El sistema brinda la posibilidad de realizar las acciones: <ul style="list-style-type: none"> • Registrar un Indicador. • Modificar un Indicador (Ver sección 1). • Eliminar un Indicador (Ver sección 2). 	
3. El actor selecciona la opción de registrar un nuevo indicador.		
	4. El sistema brinda la posibilidad de introducir los siguientes datos: <ul style="list-style-type: none"> • Nombre 	

	<ul style="list-style-type: none"> • Descripción <p>El sistema permite:</p> <ul style="list-style-type: none"> • Cancelar la operación.
5. El actor introduce el nombre y una breve descripción del indicador y selecciona la opción registrar indicador.	
	6. El sistema valida los datos introducidos.
	7. El sistema registra los datos y muestra el mensaje: "Los datos se han registrado correctamente."
	8. El sistema muestra los datos del indicador y permite: <ul style="list-style-type: none"> • Gestionar valores de estos indicadores.(Ir al CU incluido Gestionar Valores)
9. El actor selecciona la opción de registrar el indicador.	
	10. El sistema registra el indicador.
Curso Alterno	
5. El actor selecciona la opción cancelar.	
	5.1 El sistema cancela la operación. Regresar al paso 4 del flujo básico
	6. El sistema muestra un mensaje: "Los datos introducidos son incorrectos"
Sección 1 Modificar Indicador.	
Acciones de actor	Acciones del Sistema
1. El actor selecciona un indicador.	
	2. El sistema muestra los datos del indicador. El sistema permite: <ul style="list-style-type: none"> • Cancelar la operación. • Modificar el indicador.

3. El actor selecciona modificar el indicador.	
	4. El sistema habilita los campos.
5. El actor modifica los valores deseados.	
	6. El sistema valida los datos.
	7. El sistema actualiza los datos y muestra un mensaje: "Los datos han sido actualizados correctamente."
Curso Alterno.	
3. El actor selecciona la opción cancelar.	
	3.1 El sistema cancela la operación.
	6. El sistema muestra un mensaje: "Los datos introducidos son incorrectos"
Sección 2 Eliminar Indicador.	
Acciones de actor	Acciones del Sistema
1. El actor selecciona eliminar un indicador.	
	2. El sistema permite seleccionar el indicador.
3. El actor selecciona el indicador que desea eliminar.	
	4. El sistema muestra los datos del indicador. El sistema permite: <ul style="list-style-type: none"> • Cancelar la operación. • Eliminar el indicador.
5. El actor selecciona la opción eliminar un indicador.	
	6. El sistema muestra un mensaje para confirmar: ¿Está seguro que desea eliminar el indicador?
7. El actor selecciona eliminar el indicador.	
	8. El sistema elimina el indicador.

Curso Alterno	
1. El actor selecciona la opción cancelar.	
	1.1 El sistema cancela la operación.
7. El actor no confirma eliminar el indicador.	
	7.1 El sistema cancela la operación.
Prioridad: Crítico.	

Nombre del CU	Gestionar Valores del Indicador (CU incluido)	
Actores	Administrador(inicia), Jefe de Proyecto	
Propósito	Permitir registrar, modificar, o eliminar un valor del indicador.	
Resumen	El caso de uso se inicia cuando el sistema brinda la posibilidad de registrar, modificar, mostrar o eliminar un valor del indicador. Cuando el actor selecciona la opción de registrar un valor el sistema registra el nombre y la justificación del valor, si el actor desea modificar selecciona el valor, el sistema habilita los campos y permite modificar el valor, si el actor selecciona eliminar el valor el sistema elimina el valor seleccionado.	
Referencias	RF5	
CU Asociados		
Precondiciones		
Poscondiciones	Se registró, eliminó ó modificó un valor del indicador.	
Flujo Básico		
Acciones del Actor	Respuesta del Sistema	
1. El actor selecciona la opción de gestionar e valor de un indicador.		
	2. El sistema brinda la posibilidad de seleccionar el valor. Brinda la posibilidad de: <ul style="list-style-type: none"> • Registrar Valor del Indicador. • Modificar Valor del Indicador Ver Sección1 • Eliminar Valor del indicador Ver Sección 2 • Cancelar la operación. 	
3. El actor selecciona registrar un nuevo valor del indicador.		

	4. El sistema brinda la posibilidad de introducir el nombre y la justificación del valor del indicador.
5. El actor introduce los datos del valor de indicador.	
	6. El sistema valida los datos.
	7. El sistema registra los datos.
Curso Alterno	
2. El actor selecciona la opción cancelar.	
	2.1 El sistema cancela la operación.
Sección1 Modificar Valor del Indicador.	
Acciones de actor	Acciones del Sistema
1. El actor selecciona la opción de modificar un valor del indicador.	
	2. El sistema brinda la posibilidad de seleccionar el valor del indicador y modificarlo.
3. El actor selecciona un valor del indicador.	
	4. El sistema muestra los datos del valor seleccionado.
5. El actor selecciona la opción de modificar los datos del valor.	
	6. El sistema habilita los campos del valor. Permite actualizar los datos.
7. El actor modifica los datos y selecciona la opción actualizar.	
	8. El sistema valida los datos.
	9. El sistema actualiza los datos.
Curso Alterno	
1a El actor selecciona la opción cancelar operación.	
	1a.1 El sistema cancela la operación.

	8. El sistema muestra un mensaje: "Exister datos incorrectos, verifique los datos."
Sección 2 Eliminar Valor del Indicador.	
Acciones de actor	Acciones del Sistema
1. El actor selecciona la opción de eliminar un valor del indicador.	
	2. El sistema brinda la posibilidad de seleccionar el valor del indicador.
3. El actor selecciona el valor del indicador que desea eliminar.	
	4. El sistema muestra los datos del valor del indicador seleccionado.
5. El actor selecciona la opción de eliminar el valor del indicador.	
	6. El sistema muestra un mensaje ¿Está seguro que desea eliminar?
7. El actor selecciona la opción que si está seguro de eliminar el valor del indicador.	
	8. El sistema elimina el valor del indicador.
Curso Alterno	
1. El actor selecciona la opción cancelar.	
	1.1 El sistema cancela la operación.
7. El actor selecciona la opción de no eliminar.	
	7.1 El sistema cancela la operación.
Prioridad: Crítico.	

Nombre del CU	Gestionar Polo Productivo.	
Actores	Administrador (inicia)	
Propósito	Permitir registrar, modificar, o eliminar un polo productivo.	
Resumen	El caso de uso se inicia cuando el Administrador accede a la opción que le permite registrar, modificar o eliminar un polo productivo. Cuando el Administrador selecciona registrar un nuevo polo productivo introduce el nombre del polo y características de ese polo productivo, si selecciona modificar un polo, el sistema permite seleccionar el polo que desea modificar, el actor selecciona el polo y modifica los datos cuando el actor elige eliminar un polo productivo selecciona el que desea eliminar y el sistema lo elimina.	
Referencias	RF6	
CU Asociados	CU Gestionar Usuario (incluido).	
Precondiciones	El usuario debe de estar autenticado.	
Poscondiciones	Se registró, eliminó ó modificó un polo productivo.	
Flujo Básico		
Acciones del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el actor selecciona la opción de gestionar un polo productivo.		
	2. El sistema brinda la posibilidad de seleccionar el nombre del polo productivo. <ul style="list-style-type: none"> • Cancelar la operación. • Registrar un nuevo Polo Productivo. El Sistema permite: <ul style="list-style-type: none"> • Modificar Polo Productivo(Ver sección 1) • Eliminar Polo Productivo (Ver sección 2) 	
3. El actor selecciona registrar un nuevo Polo		

Productivo.	
	4. El sistema brinda posibilidad de introducir el nombre del Polo Productivo y facultad. Ir al CU incluido Gestionar Usuario.
5. El actor introduce los datos del Polo Productivo.	
	6. El sistema valida los datos.
	7. El sistema crea un Polo Productivo registrando el Jefe de Polo que lo dirige.
	8. El sistema muestra un mensaje: "Los datos se han registrado correctamente "
Curso Alterno	
3. El actor selecciona la opción cancelar.	
	3.1 El sistema cancela la operación. Regresar al paso 2 del flujo básico.
	6. El sistema muestra el siguiente mensaje indicando que existen datos incorrectos:" Existen datos incorrectos. Por favor verifique los datos".
Sección 1 Modificar Polo Productivo	
1. El actor selecciona el Polo Productivo que desea modificar.	
	2. El sistema muestra los datos del Polo Productivo seleccionado. El Sistema permite: <ul style="list-style-type: none"> • Modificar Datos • Cancelar Operación
3. El actor selecciona modificar los datos.	
	4. El sistema permite introducir los nuevos datos.
5. El actor modifica los datos.	

	6. El sistema valida los datos.
	7. El sistema muestra un mensaje: “El Polo Productivo se ha modificado correctamente ”
Curso Alterno	
3. El actor selecciona la opción de cancelar.	
	3.1 El sistema cancela la operación.
	6a.El sistema muestra el siguiente mensaje indicando que existen datos incorrectos:” Existen datos incorrectos. Por favor verifique los datos”.
Sección 2 Eliminar Polo Productivo.	
1. El actor selecciona la opción de eliminar un Polo Productivo.	
	2. El sistema permite seleccionar el Polo Productivo que desea eliminar.
3. El actor selecciona el Polo Productivo que desea eliminar.	
	4. El sistema muestra los datos del Polo Productivo seleccionado. El Sistema permite: <ul style="list-style-type: none"> • Eliminar el Polo. • Cancelar la Operación
5. El actor selecciona el Polo Productivo y selecciona eliminar.	
	6. El sistema muestra los valores del Polo Productivo.
	7. El sistema muestra un mensaje para confirmar ¿Está seguro que desea eliminar el Polo Productivo?
8. El actor acepta eliminar el Polo Productivo.	
	9. El sistema elimina el Polo Productivo.
Curso Alterno	

1. El actor selecciona la opción cancelar.	
	1.1 El sistema cancela la operación.
8. El actor selecciona la opción de no eliminar el polo.	
	8.1 El sistema cancela la operación.
Prioridad: Crítico.	

Nombre del CU	Obtener Resultados.	
Actores	Usuario(inicia)	
Propósito	Obtener resultados estadísticos de los indicadores propuestos.	
Resumen	El sistema brinda la posibilidad de introducir el Tipo de Proyecto, el actor selecciona el tipo de proyecto y selecciona obtener resultado, el sistema suma todos los factores poco probables y no probables, calcula la frecuencia acumulada, y el por ciento acumulado y genera el diagrama de Pareto. El actor si desea puede seleccionar que se genere el diagrama de Pareto por cada indicador crítico para consultar que valor de ese indicador presenta resultados no satisfactorios.	
Referencias		
CU Asociados		
Precondiciones	El usuario debe de estar autenticado en el sistema.	
Poscondiciones	Se obtuvieron resultados estadísticos.	
Flujo Básico		
Acciones del Actor	Respuesta del Sistema	
	1. El sistema brinda la posibilidad de introducir el tipo de proyecto y permite obtener resultados estadísticos.	
2. El actor selecciona el tipo de proyecto y selecciona obtener resultado.		
	3. El sistema calcula internamente y genera el diagrama de Pareto.	
	4. El sistema muestra el diagrama de Pareto y brinda la posibilidad de: <ul style="list-style-type: none"> • Ver resultados estadísticos críticos. • Cancelar operación 	
5. El actor selecciona obtener resultados estadísticos críticos.		

	6. El sistema genera el diagrama de Pareto por cada indicador que resulte crítico.
	7. El sistema le muestra el ó los diagramas de Pareto.
Curso Alterno	
5. El actor selecciona cancelar la operación.	
	5.1 El sistema cierra la pantalla.
Prioridad: Secundario.	

Anexo 5 Prototipos de interfaz de usuario.

PROTOTIPO DE INTERFAZ DE USUARIO DEL CASO DE USO: Autenticar Usuario

Logo: **CRONOS**
Gestión de indicadores

Usuario

Contraseña

Aceptar

PROTOTIPO DE INTERFAZ DE USUARIO DEL CASO DE USO: Cambiar Contraseña.

Logo: **CRONOS**
Gestión de indicadores

Consultar proyecto

Gestionar indicador

Gestionar polo productivo

Cambiar contraseña

Nueva contraseña

Confirmar contraseña

Aceptar **Cancelar**

Cancelar

PROTOTIPO DE INTERFAZ DE USUARIO DEL CASO DE USO: Gestionar Polo Productivo

CRONOS
Gestión de indicadores

Polos productivos

Polo productivo 01 ▼

Datos del polo

Nombre:

Facultad:

Datos jefe de polo

Nombre:

Usuario:

Solapín:

Registrar Actualizar Eliminar

Cancelar

Universidad de las Ciencias Informáticas 2008 (C)

Consultar proyecto

Gestionar indicador

Gestionar polo productivo

Cambiar contraseña

PROTOTIPO DE INTERFAZ DE USUARIO DEL CASO DE USO: Gestionar Ficha del Proyecto

CRONOS
Gestión de indicadores

Proyectos productivos

Proy productivo 01 ▼

Datos del proyecto

Nombre:

Facultad:

Datos jefe de proyecto

Nombre:

Usuario:

Solapín:

Registrar Modificar Eliminar

Cancelar

Universidad de las Ciencias Informáticas 2008 (C)

Consultar proyecto

Gestionar ficha proyecto

PROTOTIPO DE INTERFAZ DE USUARIO DEL CASO DE USO: Consultar Proyecto

The interface features a dark red header with the logo 'CRONOS' and the text 'Gestión de indicadores'. On the left, a vertical sidebar contains four menu items: 'Consultar proyecto', 'Gestionar indicador', 'Gestionar polo productivo', and 'Cambiar contraseña'. The main content area is titled 'Tipo de proyecto' and includes a dropdown menu, a 'Buscar' button, and a 'Búsqueda avanzada' button. A 'Cancelar' button is located at the bottom right of the main area. The footer contains the text 'Universidad de las Ciencias Informáticas 2008 (C)'.

PROTOTIPO DE INTERFAZ DE USUARIO DEL CASO DE USO: Consultar Proyecto

This version of the interface includes a horizontal navigation bar with tabs for 'Proyecto', 'Recursos', 'Arquitectura', 'Formación', 'Recursos humanos', 'Estimación', and 'Procesos'. The sidebar remains the same. The main search area is more complex, featuring three dropdown menus for 'Tipo de proyecto:', 'Entorno de proyecto', and 'Impacto'. Below these are two input fields labeled 'Entorno' and 'Impacto'. At the bottom right, there are four buttons: 'Registrar', 'Eliminar', 'Mostrar', and 'Cancelar'. The footer text 'Universidad de las Ciencias Informáticas 2008 (C)' is also present.

PROTOTIPO DE INTERFAZ DE USUARIO DEL CASO DE USO: Gestionar Indicador

The screenshot shows the 'Gestionar Indicador' interface. At the top left is the 'CRONOS' logo with the subtitle 'Gestión de indicadores'. Below the logo is a vertical sidebar with four buttons: 'Consultar proyecto', 'Gestionar indicador', 'Gestionar polo productivo', and 'Cambiar contraseña'. The main content area has three tabs: 'Indicadores' (selected), 'Datos del indicador', and 'Datos de los valores'. Under the 'Indicadores' tab, there is a dropdown menu with 'Nuevo' selected, and a list of 'Indicador 01', 'Indicador 02', and 'Indicador 03'. At the bottom of the main area are four buttons: 'Registrar', 'Modificar', 'Eliminar', and 'Gestionar valores'. A 'Cancelar' button is located at the bottom right. The footer contains the text 'Universidad de las Ciencias Informáticas 2008 (C)'.

PROTOTIPO DE INTERFAZ DE USUARIO DEL CASO DE USO: Gestionar Valor del Indicador.

The screenshot shows the 'Gestionar Valor del Indicador' interface. It features the same 'CRONOS' logo and sidebar as the previous screen. The main content area has three tabs: 'Indicadores', 'Datos del indicador', and 'Datos de los valores' (selected). Under the 'Datos de los valores' tab, there is a dropdown menu with 'Nuevo' selected, and a list of 'Valor 01', 'Valor 02', and 'Valor 03'. At the bottom of the main area are four buttons: 'Registrar', 'Modificar', 'Eliminar', and 'Gestionar valores'. A 'Cancelar' button is located at the bottom right. The footer contains the text 'Universidad de las Ciencias Informáticas 2008 (C)'.

GLOSARIO DE TÉRMINOS

ALBET. SA: Empresa Comercializadora de la UCI.

Analogía: Comparación o relación entre varias razones o conceptos; comparar o relacionar dos o más objetos o experiencias, apreciando y señalando características generales y particulares.

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

Calidad de software: Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con características implícitas que se espera de todo software desarrollado profesionalmente.

CASE: Computer Aided Software Engineering, siglas en inglés de Ingeniería de Software Asistida por Ordenador.

Casos de uso: Fragmentos de funcionalidad del sistema.

Expediente de Proyecto: Término utilizado para referirse a toda la documentación de un proyecto productivo.

Hitos: Meta u objetivo de un determinado proceso, es un logro de resultados importantes en el proyecto.

Infraestructura Productiva (IP): Estructura central encargada de dirigir la producción de la UCI y supervisar los proyectos productivos.

Ingeniería de Software: Es una tecnología multicapa en la que se pueden identificar: los métodos que indican cómo construir técnicamente el software, el proceso que es el fundamento de la Ingeniería de Software y las herramientas que son un soporte para el proceso y los métodos. Tiene varios modelos o paradigmas de desarrollo en los cuales se puede apoyar para la realización de software.

LDAP: Es un protocolo a nivel de aplicación, que permite el acceso a un servicio para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) al que pueden realizarse consultas. Habitualmente, almacena la información del usuario y contraseña y es utilizado para autenticarse en el dominio uci.cu.

Minería de datos: Consiste en la extracción no-trivial de información que reside de manera implícita en los datos. Dicha información era previamente desconocida y podrá resultar útil para algún proceso. La minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos.

Modelo empírico: Modelo basado en la experiencia y en la percepción.

Muestra: En estadística una muestra debe ser definida en base de la población determinada y las conclusiones que se obtengan de dicha muestra sólo podrán referirse a la población en referencia.

Paradigma: Modelo o Patrón que se utiliza como guía.

PMBOK: Guía de los Fundamentos de la Dirección de Proyectos, es un estándar de gestión de proyectos desarrollado por el Instituto de Administración de Proyectos.

Población: Conjunto de elementos que presentan una característica común.

Puntos de función sin ajustar: La sumatoria del número de puntos por función basándose en el tipo de componente y su complejidad, que se le asigna a los componentes del sistema informático en términos de transacciones.

Puntos de función: Miden la aplicación desde una perspectiva del usuario, dejando de lado los detalles de codificación. Se define como una función comercial del usuario final.

Rational Rose Enterprise Suite: Es una herramienta de desarrollo basada en modelos utilizada para el modelado visual de aplicaciones y da soporte a UML (Lenguaje de Modelado Visual).

Release: Versión de un producto.

Rol: Define el comportamiento y responsabilidad de un individuo o grupo de individuos en un sistema.

SEI: Instituto federal estadounidense de investigación y desarrollo, fundado por el Congreso de los Estados Unidos en 1984 para desarrollar modelos de evaluación y mejora en el desarrollo de software.

Software libre: Se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Stakeholders: Interesado o partes interesadas en el proyecto, incluye a clientes y desarrolladores. Todos aquellos impactados por una organización, incluyendo individuos, grupos y organizaciones.

Viabilidad: Condición que hace posible el funcionamiento del sistema, proyecto o idea al que califica, atendiendo a sus características.