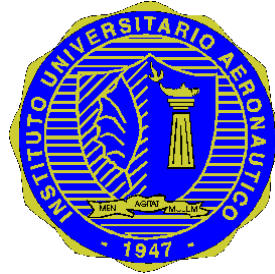




Instituto Universitario Aeronáutico FCA



Proyecto de Grado Tesis Ingeniería de Sistemas



Indicadores de calidad en el Desarrollo de Software

Versión 1.8



Historia de Revisión del Documento

Fecha	Versión	Descripción	Autor
30/03/2017	1.1	Primera versión del Documento	Pérez, Gabriela Gioino, Andrés
15/05/2017	1.2	Se envía la primer Entrega al tutor (Alejandra Boggio).	Pérez, Gabriela Gioino, Andrés
30/07/2017	1.3	Correcciones de las observaciones del tutor Sobre la respuesta del 15/05/2017 de la primera revisión.	Pérez, Gabriela Gioino, Andrés
18/08/2017	1.4	Se envía Segunda entrega al tutor (Alejandra Boggio). Se envían las correcciones de las observaciones del tutor Sobre la respuesta enviadas por el tutor el 15/05/2017 de la primera revisión.	Pérez, Gabriela Gioino, Andrés
21/10/2017	1.5	Correcciones de las observaciones del tutor Sobre la respuesta del 17/09/2017 de la segunda revisión.	Pérez, Gabriela Gioino, Andrés
03/11/2017	1.6	Se envía Tercera entrega al tutor (Alejandra Boggio). Se envían las correcciones de las observaciones del tutor Sobre la respuesta enviadas por el tutor el 17/09/2017 de la Segunda revisión.	Pérez, Gabriela Gioino, Andrés
22/06/2018	1.8	Se realiza en el análisis de los requerimientos para hacer el modelo conceptual del DW y luego el modelo lógico, en el punto 9.	Pérez, Gabriela Gioino, Andrés
...



Integrantes	Gioino, Andrés Emiliano - DNI: 29.963.477 Pérez, Gabriela Belén - DNI: 30.123.943
Tutor	Ing. Boggio, Alejandra
Mesa	Brenda Meloni



Índice de Contenidos

1	DEDICATORIA	6
1.1	Andrés	6
1.2	Gabriela.....	6
2	AGRADECIMIENTOS	6
2.1	Andrés	6
2.2	Gabriela.....	6
3	RESUMEN DEL PG	7
4	GLOSARIO DE PALABRAS	11
5	OBJETIVOS Y ALCANCE DEL PROYECTO	12
6	INTRODUCCION	13
7	MARCO TEORICO.....	14
7.1	INDICADORES DE CALIDAD EN EL DESARROLLO DE SOFTWARE	14
7.1.1	Modelos para medir la calidad del software	21
7.1.2	Modelo McCall.....	21
7.1.3	Modelo de Dromey.....	22
7.1.4	Modelo de FURPS(Functionality, Usability, Reliability, Performance, Supportability).....	23
7.1.5	Modelo CMM (Capability Maturity Model)	24
7.1.6	Modelo paradigma GQM (Goal-Question-Metric).....	25
7.2	DATA WAREHOUSE	29
7.3	PARADIGMA BILL INMON	30
7.4	PARADIGMA RALPH KIMBALL.....	32
7.5	MODELADO DE DATOS	34
7.6	MINERIA DE DATOS.....	38
8	RELEVAMIENTO Y/O DIAGNOSTICO Y CONCLUSIONES.....	43
8.1	Relevamiento.....	43
8.2	Diagnóstico.....	43
8.3	Conclusiones	44
9	PROPUESTA.....	45
9.1	APLICACIÓN GQM SOBRE GPIS	45
9.2	Modelado del Data Warehouse	46
9.2.1	Base de Datos Operacional de GPIS.	46
9.2.2	Análisis de las dimensiones.	47
9.2.3	Modelo conceptual.....	48
9.2.4	Esquema Estrella (StarSchema).	49
9.2.5	ETL con PentahoPDI (Pentaho Data Integration).....	50
10	RESULTADOS.....	51
11	CONCLUSIONES DEL PG.....	51
12	BIBLIOGRAFIA	51
13	ANEXOS.....	52



1 DEDICATORIA

1.1 Andrés

A mi hijo Franco y mi esposa.

1.2 Gabriela

A mis padres.

2 AGRADECIMIENTOS

2.1 Andrés

Agradecer a toda la familia por el apoyo de tantos años.

2.2 Gabriela

A toda mi familia y amigos por el apoyo incondicional.

A la institución y docentes por brindarnos los conocimientos y herramientas que nos permitieron comenzar a desarrollarnos como profesionales.



3 RESUMEN DEL PG

El proyecto de Grado para el título de Ingeniería de Sistemas, se basa en el software “*Gestión de Proyectos de Ingeniería de Software*” (GPIS), proyecto realizado y desarrollado para el Trabajo Final de Pregrado que se representó para el título de Analista de Sistemas.

GPIS es un sistema multiusuario, donde directores, gerentes, jefes o líderes de proyecto pueden dar de alta proyectos. Para cada proyecto, el sistema tiene las etapas del ciclo de vida de desarrollo de software: Requerimiento, Análisis, Diseño, Desarrollo, Testing e Implementación.

GPIS va guiando a cada usuario, según el rol logueado, por todo lo que debe hacer en cada etapa: estimaciones, documentaciones, procedimientos, informaciones, metodologías, entre otros.

GPIS, además, tiene control de las estimaciones e historial por si éstas van cambiando por fuerzas mayores y así tienen su justificación, y también permite ir poniendo las fechas reales que se vayan cumpliendo en cada una de las etapas del ciclo de vida.

GPIS cuenta con aprobaciones requeridas en cada etapa e informa vía e-mail dichas aprobaciones a los aprobadores correspondientes, además de informar las etapas a las que se va avanzando en el proyecto.

Resumen proyecto de grado:

- El nuevo desafío propuesto para innovar sobre este sistema de gestión de proyectos informáticos, es agregarle inteligencia e indicadores al mismo, de modo de poder capitalizar toda la información que se le ingresa día a día.

Por lo tanto, lo que se va a realizar en este proyecto de grado es la investigación y aplicación de indicadores de calidad en el desarrollo de software, aplicando técnicas de MINERIA DE DATOS, DW, BI, mediante un Software Open Source: PENTAHO BI.



Situación Problemática

La problemática que se pretende solucionar es la carencia de mediciones de objetivos y desempeño de los RRHH que utilizan el sistema, a través de indicadores numéricos concretos e inteligentes. Por ejemplo, entre otros:

- Obtener los desvíos entre las fechas estimadas y las reales, por etapa de ciclo de vida del proyecto y/o por área ejecutora y/o por actor.
- Generar reportes de calidad de desarrollo en base a cantidad de errores reportados vs. cantidad de casos de prueba, clasificando dichos errores como de análisis, construcción o históricos, por analista y desarrollador.

Esto con la idea de poder detectar oportunidades de mejora y tomar acciones sobre las causas raíces de los problemas.

También se necesitan generar reportes para todos los niveles jerárquicos de la empresa mensuales y/o anuales (u otros períodos), que permitan por ejemplo medir la productividad (cantidad de proyectos cerrados), obtener un ranking de las áreas usuarias que solicitan desarrollos, contabilizar los tipos de pedido (nueva funcionalidad, corrección de error, mejora).

Funcionalidad del Sistema

Se extraerá la información de la BD relacional productiva (se analizan las propiedades de los datos, se transforma el conjunto de datos de entrada, se selecciona y aplica técnica de minería de datos, se extrae el conocimiento, se interpretan y evalúan los datos) y se la transformará para poder visualizarla en forma sintética y precisa mediante tableros y reportes que permitan a los niveles jerárquicos tomar las acciones necesarias para poder mejorar la calidad, eficiencia y productividad de los desarrollos de software y los RRHH de los equipos de trabajo.

Midiendo por ej. :

- Calidad: cuántos desarrollos vuelven de Testing a Desarrollo y cuántos



vuelven de Producción, en este último caso, cuál fue el área que falló. Cuántos desarrollos de correcciones de error se reciben. Ranking de áreas y RRHH.

- Eficiencia y productividad: Cuántas veces se reestiman las fechas. Comparaciones de fechas estimadas vs reales. Ranking de áreas de RRHH.

Entre otros reportes que pueden obtenerse de forma mensual o anual, o con la periodicidad que se requiera y al nivel de detalle que se considere necesario (por área o por recurso puntual).

Idea a defender

La idea a defender es que se pueden explotar todos los datos operativos del día a día dándole inteligencia al negocio. Se quiere probar, entonces, con este proyecto, que obteniendo datos concretos, tangibles y medibles se pueden tomar decisiones para mejorar la calidad, eficiencia y productividad de los desarrollos de software.

Es decir, se obtendrán indicadores de calidad del desarrollo de software a partir de la aplicación de una técnica de MINERÍA DE DATOS, a través de la herramienta Open Source PENTAHO.

Aporte teórico y práctico

La justificación para la realización del proyecto es la investigación sobre los mejores indicadores de calidad en el desarrollo de software aplicando la minería de datos sobre el proyecto (GPIS) que va a ser usado en Claro AMX S.A. a través de la consultora AR Consultores S.R.L.

“La medición es el primer paso para el control y la mejora. Si no se puede medir algo, no se lo puede entender. Si no se entiende, no se puede controlar. Si no se puede controlar, no se puede mejorar” (H. James Harrington).



La idea entonces es medir lo que sucede, entenderlo y tomar las decisiones que permitan mejorar la gestión. En cierto modo, esto puede implicar un cambio cultural de la organización, ya que podrían incentivarse cambios en la comunicación y colaboración para lograr los objetivos de cumplimiento de fechas, plantearse estímulos (monetarios o no) al poder medir la productividad de los RRHH involucrados en los proyectos, así como plantear el cambio de estructuras de áreas, incrementando personal, si se detecta que los desvíos del cumplimiento de fechas de las etapas de los proyectos se presentan siempre en determinadas áreas. Por lo tanto, son muchas las aplicaciones prácticas que se pueden aplicar a partir de la MINERIA DE DATOS.

Se busca mejorar la calidad, eficiencia y productividad, logrando cambios en las habilidades blandas de gestión, pudiendo tomar decisiones a través de la inteligencia de negocio aplicada con PENTAHO sobre los datos ingresados en el sistema GPIS.



4 GLOSARIO DE PALABRAS

BI = Business Intelligence.

BD = Base de Datos.

DW = Data Warehouse.

DM = Data Mart.

GPIS = Gestión de Proyectos de Ingeniería de Software.

RRHH = Recursos Humanos.

SO = Sistema Operativo.

CVDS = Ciclo de Vida del Desarrollo de Software.

CP = Casos de Prueba.



5 OBJETIVOS Y ALCANCE DEL PROYECTO

Objeto de Estudio

Objeto de Estudio: Medidas y métricas obtenidas desde GPIS para aplicar al desarrollo de software.

Objetivo General

El objetivo general es poder tener implementada inteligencia de negocios, “extraer conocimiento a partir de bases de datos” (MINERIA DE DATOS). Para poder tener la visibilidad en tableros y reportes en todo el nivel vertical de la empresa, y así poder tomar decisiones a niveles gerenciales, como poder hacer las mediciones de objetivos de las personas. Así se podrán mejorar, la calidad, eficiencia y productividad de los desarrollos de software.

Alcance

El alcance del proyecto es aplicar al menos una técnica de la MINERIA DE DATOS, empleando PENTAHO, sobre un modelo de DW a concebirse partiendo de la base de datos relacional que se está poblando con el sistema GPIS, de modo de obtener los indicadores de calidad del desarrollo de software, que necesita la organización.



6 INTRODUCCION

Dado el objetivo de aplicar inteligencia de negocio a los datos que actualmente se poseen en la DB relacional, empleando PENTAHO, se generará un esquema de datos multidimensional, un StarSchema, y se trabajará sobre la información que se genere con el mismo, para poder obtener los indicadores de la calidad en el ciclo de vida de desarrollo de software.

Se hace a continuación un repaso sobre los conceptos principales de DW y MINERÍA DE DATOS y, además, sobre los mejores indicadores de calidad en el desarrollo de software.



7 MARCO TEORICO

7.1 INDICADORES DE CALIDAD EN EL DESARROLLO DE SOFTWARE

Establecer sistemas de medición y evaluación es una pieza básica de la gestión de calidad de software, más aún dentro de la actual tendencia a externalizar el desarrollo, que es realizado por consultoras o desarrolladores de software externos.

Pero aunque los desarrollos sean ejecutados internamente por un área de la propia entidad o externamente por una empresa que tenga certificación en calidad de software, no garantiza que su software sea de calidad.

A través de los diferentes métodos que existen en el mundo informático, se puede deducir que no existe uno en particular que garantice el 100% como resultado la calidad esperada de un producto de software, sin embargo, en ese camino, conlleva a experimentar metodologías que sumadas a criterios y experiencias particulares han sabido otorgar un grado importante de confianza a determinados procesos del ciclo de vida de un software y a los desarrolladores al momento de la evaluación final de un software.

Si bien es cierto que la calidad de un software depende mucho de la concepción del requerimiento, es mucho más importante saber lo que no se sabe al inicio. Se refiere a la elasticidad de un producto y que se adapte al cambio constante de requerimientos para que la calidad del producto no se degenere con el transcurso del tiempo.

Entre la calidad lograda al momento determinar un producto y mantener la misma calidad posterior a la implantación es el objetivo de esta investigación que trata en lo posible de especificar algunos indicadores que contribuyen de manera específica a lograr dicho objetivo. Estos se hallan al inicio del proyecto y se certifican y evalúan al final. Por lo tanto los requerimientos bien analizados definen



al final la calidad de un producto y posterior al mantenimiento del mismo para que no pierda la calidad inicial.

Naturaleza del problema

En la actualidad los mercados de mandan productos con calidad más apreciada que años anteriores. Los usuarios de hoy en día están más entrenados e informados con respecto a su negocio por lo que los desarrolladores han tomado diversas técnicas y metodologías para cumplirlas actuales exigencias.

Algunas de esas prácticas predominan para el tema de calidad, control y aseguramiento de calidad, y uno de los principales problemas con los que se encuentra la actividad de aseguramiento de la calidad en el software es la falta de apoyo por parte de la alta dirección de las organizaciones. Este apoyo es esencial para que la función de aseguramiento de calidad tenga éxito.

Los costos económicos de la función de aseguramiento de la calidad en el software han ido creciendo mientras los plazos y métodos lo contrario.

El costo se localiza en las actividades (como son revisiones periódicas y constantes de las aplicaciones) que tienen que realizar algunos desarrolladores de software, mismas que se integran a sus actividades ordinarias, pero los índices estadísticos muestran que usando alguna de estas herramientas no aseguran el 100% de calidad.

Aseguramiento de calidad vs. Control de la calidad

Es de suma importancia entender las diferencias que existen entre el control de la calidad y el aseguramiento de la calidad. El aseguramiento de la calidad aprovecha los resultados del control de calidad para evaluar y mejorar los procesos con los que se desarrolla el producto.



El control de calidad se enfoca en productos, mientras que el aseguramiento de la calidad lo hace en los procesos.

Planteamiento:

- a. Para los dos tipos de controles se deben colocar ciertas alarmas, sensores, termómetros, que en resumen se denominan “indicadores de calidad” para controlar y asegurar lo requerido, lo planeado, lo ejecutado y lo esperado.
- b. Se deberán revisar los procesos que cumple el ciclo de vida del desarrollo de software para identificar aquellos procesos críticos.
- c. Se deberán seleccionar aquellos espacios o momentos donde se puedan colocar indicadores y/o medidas que otorguen alarmas de éxito.
- d. Posterior al producto terminado, identificar los procesos lo cual ayudará a obtener calidad constante o mejorada.

A continuación se enuncian algunos conceptos de calidad:

- ✓ Definición de la norma ISO9000: “Calidad: grado en el que un conjunto de características inherentes cumple con los requisitos” o “conjunto de normas y directrices de calidad que se deben llevar a cabo en un proceso”.
- ✓ Real Academia de la Lengua española: “Propiedad o conjunto de propiedades inherentes a una cosa que permiten apreciarla como igual, mejor o peor que las restantes de su especie”.
- ✓ Philip Crosby: “Calidad es cumplimiento de requisitos”.
- ✓ Joseph Juran: “Calidad es adecuación al uso del cliente”.
- ✓ Armand V. Feigenbaum: “Satisfacción de las expectativas del cliente”.
- ✓ Genichi Taguchi: “Calidad es la menor pérdida posible para la sociedad”.
- ✓ William Edwards Deming: “Calidad es satisfacción del cliente”.
- ✓ Walter A. Shewhart: “La calidad como resultado de la interacción de dos dimensiones: dimensión subjetiva (lo que el cliente quiere) y dimensión objetiva (lo que se ofrece)”.

Nunca se debe confundir la calidad de un producto con “niveles superiores” de atributos del producto o servicio, sino con la obtención regular y permanente de los



atributos del bien ofrecido que satisfaga a los clientes para los que ha sido diseñado.

Medidas, métricas e indicadores de la calidad del software

Las métricas son un medio para entender, controlar, predecir y probar el desarrollo del software y los proyectos de mantenimiento.

Se aplican las métricas para poder evaluar la calidad de producto de ingenierías o los sistemas que se construyen.

Se aplican a todo ciclo de vida permitiendo descubrir y corregir errores potenciales.

Definiciones

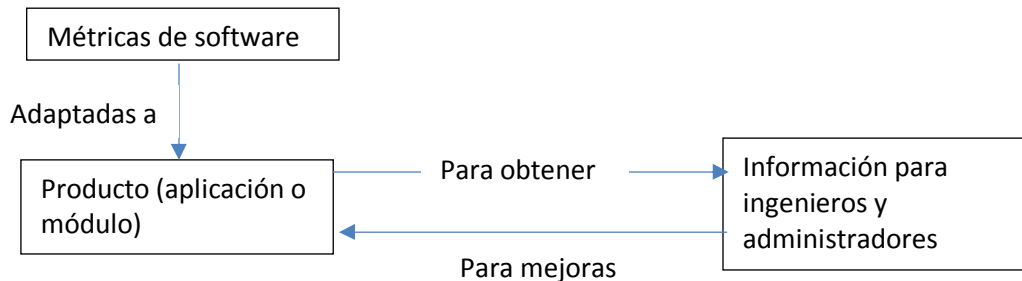
- ✓ Medida: proporciona una indicación cuantitativa de la cantidad, dimensiones o tamaño de algunos atributos de un producto.
- ✓ Medición: acto de determinar una medida.
- ✓ Métrica: es una medida del grado en que un sistema, componente o proceso posee un atributo dado.
- ✓ Indicadores: es una métrica o combinación de métricas que proporcionan una visión profunda del proceso de software.
- ✓ Error: Fallo de un producto que se descubre antes de entregar el software al usuario final.
- ✓ Defectos: un fallo que se produce una vez que se entrega el producto al usuario final.

Medir, ¿para qué?

- Controlar
- Conocer
- Comunicar



El progreso, los costos, lo que se hace bien, lo que se hace mal, dónde se necesita inversión, dónde se puede ahorrar, desempeño; etc.



Razones para medir un producto

- Para indicar la calidad del producto.
- Para evaluar la productividad de la gente que desarrolla el producto.
- Para evaluar los beneficios en términos de productividad y de calidad, derivados del uso de nuevos métodos y herramientas de la ingeniería del software.
- Para establecer una línea de base para la estimación.
- Para ayudar a justificar el uso de nuevas herramientas o de formación adicional.

Ventajas

- Determinar la calidad del producto.
- Evaluar la productividad de los desarrolladores.
- Conocimiento cuantitativo de las características del proceso del producto.
- Se podrán realizar comparaciones con otros proyectos.
- Se podrá mejorar el producto ya que las métricas sirven para detectar defectos.



Clasificación de las métricas de software

Según los criterios:

- ✓ De complejidad: métricas que definen la medición de la complejidad: volumen, tamaño, anidaciones y configuración.
- ✓ De calidad: métricas que definen la calidad del software: exactitud, estructuración, modularidad, pruebas, mantenimiento.
- ✓ De competencia: métricas que intentan valorar o medir las actividades de productividad de los programadores con respecto a su certeza, rapidez, eficiencia y competencia.
- ✓ De desempeño: métricas que miden la conducta de módulos y sistemas de un software, bajo la supervisión del SO o hardware.
- ✓ Estabilizadas: métricas de experimentación y de preferencia: estilo de código, convenciones, limitaciones, etc.

Según el contexto en que se aplican:

- ✓ Métricas de proceso:
 - Se recopilan de todos los proyectos y durante un largo período de tiempo.
 - Caracterizados por control y ejecución del proyecto y por medición de tiempos de las fases.
- ✓ Métricas del proyecto:
 - Permiten evaluar el estado del proyecto.
 - Permiten seguir la pista de los riesgos.
- ✓ Métricas del producto:
 - Se centran en las características del software y no en cómo fue producido.
 - También son productos los artefactos, documentos, modelos y componentes que conforman el software.
 - Se miden cosas como el tamaño, la calidad, la totalidad, la volatilidad y el esfuerzo.



Hablando de calidad de software

La calidad de software es el grado con el que un sistema, componente o proceso cumple los requisitos funcionales definidos y las necesidades del cliente o usuario.

Para obtener calidad se necesita:

- Gestión
- Formación
- Infraestructura
- Metodología
- Recursos humanos
- Arquitectura

Objetivos de Calidad

- Satisfacción al cliente
- Reducir costos de desarrollo
- Reducir costos de mantenimiento
- Reducir tiempo de desarrollo
- Asegurar los requisitos
- Asegurar la metodología usada por la organización



7.1.1 Modelos para medir la calidad del software

Los modelos de calidad sirven para evaluar la calidad del producto y/o del proceso.

Son modelos y no metodologías. Un modelo de calidad nos dice qué hacer pero no el cómo hacerlo.

Descomponen la calidad de software en unas series de características.

Algunos modelos:

7.1.2 Modelo McCall

Conocido como modelo de factores/criterios/métricas.

Se focaliza en el producto software.

Identifica atributos (llamados factores de calidad) claves desde el punto de vista del usuario.

Define tres vistas:

- Vista de usuario: clasifica los factores de calidad.
- Vista de la dirección: clasifica los criterios de calidad.
- Vista del desarrollador: clasifica las vistas de calidad.

Los tres factores de calidad comprendidos en la vista de usuario coinciden con tres de las características de un producto de software:

- Revisión del producto: habilidad para ser cambiado.
- Transición del producto: adaptabilidad al entorno.
- Operación del producto: características de operación.



7.1.3 Modelo de Dromey

Resalta el hecho que la calidad del producto es altamente determinada por los componentes del mismo (documentos, guías de usuarios, diseño, código).

Sugiere el uso de cuatro categorías:

<i>Factor</i>	<i>Criterio</i>
Correctitud	Funcionalidad Confiabilidad
Internas	Mantenibilidad Eficiencia Confiabilidad
Contextuales	Mantenibilidad Reusabilidad Portabilidad Confiabilidad
Descriptivas	Mantenibilidad Reusabilidad Portabilidad Usabilidad



7.1.4 Modelo de FURPS(Functionality,Usability,Reliability,Performance,Supportability)

Basado en el modelo Mccal.

Funcionalidad, usabilidad, confiabilidad, desempeño y capacidad de soporte.

Se utilizan para establecer métricas de la calidad para todas las actividades de proceso de desarrollo de un software, inclusive de un sistema de información.

Factor	Criterio
Funcionalidad	Características y capacidades del programa. Generalidad de las funciones. Seguridad del sistema.
Facilidad de Uso	Factores humanos. Factores estéticos. Consistencia de la interfaz. Documentación.
Confiabilidad	Frecuencia y severidad de las fallas. Exactitud de las salidas. Tiempo medio de fallos. Capacidad de recuperación ante fallas. Capacidad de predicción.
Rendimiento	Velocidad del procesamiento. Tiempo de respuesta. Consumo de recursos. Rendimiento efectivo total. Eficacia.
Capacidad de Soporte	Extensibilidad. Adaptabilidad. Capacidad de pruebas. Capacidad de configuración. Compatibilidad. Requisitos de instalación.



7.1.5 Modelo CMM (Capability Maturity Model)

Es motivado por los problemas que tenía el Departamento de Defensa de los Estados Unidos cuando encargaba el desarrollo de un software a otra empresa: se disparaban los presupuestos, las fechas se alargaban.

Convocaron a un concurso de participación libre en el que se proponían resolver esos problemas.

La Universidad Carnegie Mellon ganó el concurso y creó el SEI y desde este instituto se creó el modelo de calidad CMM-CMMI. Su primera versión fue publicada en el 2002.

Es un modelo que evalúa los procesos de desarrollo.

Orientado a la mejora de procesos de desarrollo de software.

Describe prácticas que conducen a mejores productores de software.

Ofrece un conjunto de prácticas importantes que deben ser implantadas por cualquier entidad que desarrolla o mantiene software.

CMM tiene como objetivo evaluar los procesos en sus distintos niveles de madurez.

Clasifica las empresas según su nivel de madurez de sus procesos de desarrollo. Existen 5 niveles:

- Nivel 1 - Inicial.
- Nivel 2- Gestionado o repetible.
- Nivel 3 - Definido.
- Nivel 4 - Cuantitativamente Gestionado.
- Nivel 5 - Optimizado.



7.1.6 Modelo paradigma GQM (Goal-Question-Metric)

Este modelo es el que se empleará en el presente proyecto.

Paradigma que permite evaluar la calidad del producto y del proceso.

Evalúa la calidad del software basado en la identificación de objetos a lograr.

Tres etapas:

1. Lista de objetivos principales en el desarrollo y mantenimiento del proyecto.
2. Para cada objetivo obtener las preguntas que deben contestarse para saber si se están cumpliendo los objetivos.
3. Decidir qué medir para poder contestar las preguntas de forma adecuada.

Algunos errores clásicos en un proyecto de software:

- ✓ Mal análisis de los requerimientos y una mala planificación.
- ✓ No tener una negociación (documento, contrato) con el cliente.
- ✓ No hacer un análisis costo - beneficio.
- ✓ Desconocer el ambiente de trabajo de los usuarios.
- ✓ Desconocer a los usuarios que trabajan con el sistema.
- ✓ Mala elección de recursos.

Recomendaciones:

- ✓ Comenzar con un sistema pequeño.
- ✓ Mostrar los datos de tal forma que aporten al negocio y a su gestión.
- ✓ No perder de vista los objetos y la estrategia del cliente.

Este método fue originalmente definido por Basili y Weiss (1984) y extendido posteriormente por Rombach (1990) como resultado de muchos años de experiencia, práctica e investigación académica.



Un programa de medición es más satisfactorio si es diseñado teniendo en cuenta las metas (objetivo perseguido).

La medición siempre debe ser realizada orientada a un objetivo.

Mejorar la calidad, reduciendo costos, riesgos y mejorando tiempos.

META – PREGUNTA–METRICA.

Cada objetivo se descompone en varias preguntas para entender los componentes del objetivo y finalmente se obtienen métricas que dan respuestas a cada una de las preguntas.

➤ NIVELES

→ Conceptual → Operativo → Cuantitativo.

- **META (Conceptual):** Se establece un objeto para cada elemento de medición, considerando el producto, proceso y los recursos, desde diferentes puntos de vista.

Las metas deben ser generales, abstractas e intangibles. Responden a la pregunta de ¿Qué se quiere alcanzar?, y la respuesta es cualitativa.

Objetos de la medición:

- ✓ Productos: Entregables y documentos que se producen durante el ciclo de vida de un sistema.
 - ✓ Procesos: Actividades relacionadas con el software y asociadas generalmente al tiempo.
 - ✓ Recursos: Elementos que los procesos utilizan para reducir sus salidas.
- **PREGUNTA (Operativo):** Con base en las metas definidas se establece un conjunto de preguntas que permiten caracterizar la evaluación.



- **METRICA (Cuantitativo):** A cada pregunta se le asocian datos que permitan dar respuesta cuantitativa a los objetivos, de manera objetiva y subjetiva.

➤ PROCESO

1. *Establecer las metas.* Una vez que se definen los objetivos del negocio, múltiples proyectos o subgrupos de la organización van a tener las bases para identificar objetivos de medida relacionados con sus roles, o alcance de influencia.

2. *Generación de preguntas.* El equipo de proyecto identifica preguntas que deben ser hechas para capturar varias perspectivas para lograr el objetivo.

Los gerentes de proyecto y los ingenieros de software proveen sus propias perspectivas del significado del objetivo en dicho entorno. Ellos hacen esto haciendo preguntas y respondiendo con sus métricas.

3. *Especificación de medidas.* Responder moviéndose de un nivel cualitativo (o nivel operacional) a un nivel cuantitativo.

Se necesitan definir métricas que provean toda la información cuantitativa para responder las preguntas del paso dos, de manera satisfactoria.

4. *Preparar recolección de datos.* Una vez que las métricas son identificadas, se determinan los datos necesarios para las métricas y cómo serán recolectados.

El plan también define y describe todas las formas, tipos de recolección de datos y herramientas automáticas que deben ser utilizadas.



Es importante entrenar a los individuos involucrados en la recolección de datos.

5. *Recolectar, validar y analizar datos para la toma de decisiones.* Consiste en el chequeo de completitud, correctitud y consistencia de los mismos.

Una vez validados, es importante almacenar los datos de medida de tal manera que puedan ser accedidos para analizarlos y generar reportes, por el inmenso volumen de los mismos se utilizan herramientas de soporte de medidas y base de datos.(Plan de análisis)

6. *Analizar los datos para el logro de los objetivos y el aprendizaje.* El último paso del proceso de GQM de Basili es observar los resultados de las medidas de modo post-mortem para evaluar los objetivos logrados y determinar las lecciones aprendidas para ser utilizadas en futuros proyectos.



7.2 DATA WAREHOUSE

“Es una colección de base de datos integradas y orientadas al usuario, diseñadas con el objeto de apoyar a la función de decisión de una organización, en el cual, cada unidad de datos es relevante en un momento dado” W.H.Inmon

Un DATA WAREHOUSE (DW) proporciona una visión global, común e integrada de los datos de la organización, independiente de cómo se vayan a utilizar posteriormente por los consumidores o usuarios. Normalmente en el almacén de datos habrá que guardar información histórica que cubra un amplio período de tiempo. Pero hay ocasiones en las que no se necesita la historia de los datos, sino sólo sus últimos valores, siendo además admisible generalmente un pequeño desfase o retraso sobre los datos operacionales. En estos casos el almacén se llama almacén operacional (ODS, Operational Data Store).

Podemos entender un DATA MART (DM) como un subconjunto de los datos del DW con el objetivo de responder a un determinado análisis, función o necesidad y con una población de usuarios específica.

¿Qué diferencia existe entonces entre un DM y un DW? Su alcance. El DM está pensado para cubrir las necesidades de un grupo de trabajo o de un determinado departamento dentro de la organización. Es el almacén natural para los datos departamentales. En cambio, el ámbito del DW es la organización en su conjunto. Es el almacén natural para los datos corporativos comunes.



7.3 PARADIGMA BILL INMON

Bill Inmon ve la necesidad de transferir la información de los diferentes OLTP (OnLine Transaction Processing), (Sistemas Transaccionales) de las organizaciones a un lugar centralizado donde los datos puedan ser utilizados para el análisis. Insiste además en que ha de tener las siguientes características:

- *Orientado a temas.* Los datos en la base de datos están organizados de manera que todos los elementos de datos relativos al mismo evento u objeto del mundo real queden unidos entre sí.
- *Integrado.* La base de datos contiene los datos de todos los sistemas operacionales de la organización, y dichos datos deben ser consistentes.
- *No volátil.* La información no se modifica ni se elimina, una vez almacenado un dato, éste se convierte en información de sólo lectura, y se mantiene para futuras consultas.
- *Variante en el tiempo.* Los cambios producidos en los datos a lo largo del tiempo quedan registrados para que los informes que se puedan generar reflejen esas variaciones.

La información ha de estar a los máximos niveles de detalle. Los DW departamentales o DM son tratados como subconjuntos de este DW corporativo, que son construidos para cubrir las necesidades individuales de análisis de cada departamento, y siempre a partir de este DW Central (del que también se pueden construir los ODS (Operational Data Stores) o similares).



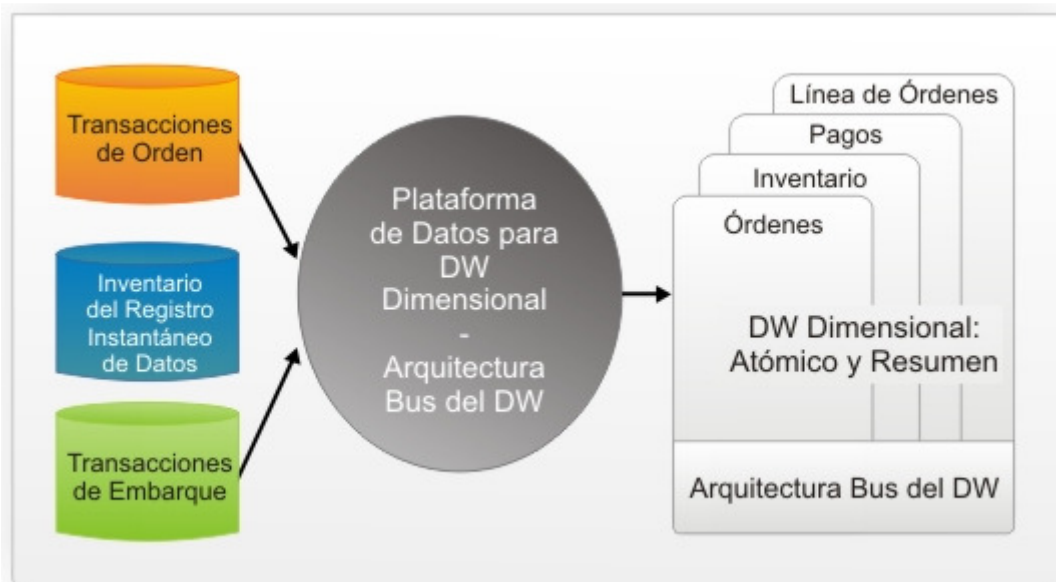
El enfoque Inmon también se referencia normalmente como **Top-down**. Los datos son extraídos de los sistemas operacionales por los procesos ETL y cargados en las áreas de stage, donde son validados y consolidados en el DW corporativo, donde además existen los llamados metadatos que documentan de una forma clara y precisa el contenido del DW. Una vez realizado este proceso, los procesos de refresco de los DM departamentales obtienen la información de él, y con las consiguientes transformaciones, organizan los datos en las estructuras particulares requeridas por cada uno de ellos, refrescando su contenido.

La metodología para la construcción de un sistema de este tipo es la habitual para construir un sistema de información, utilizando las herramientas habituales (esquema Entidad Relación, DIS (Data Item Sets), etc). Para el tratamiento de los cambios en los datos, usa la Continue and Discrete Dimension Management (inserta fechas en los datos para determinar su validez para las Continue Dimension o bien mediante el concepto de snapshot o foto para las Discrete Dimension).



7.4 PARADIGMA RALPH KIMBALL

El DW es un conglomerado de todos los DM dentro de una empresa, siendo una copia de los datos transaccionales estructurados de una forma especial para el análisis, de acuerdo al Modelo Dimensional (no normalizado), que incluye, como ya se vio, las dimensiones de análisis y sus atributos, su organización jerárquica, así como los diferentes hechos de negocio que se quieren analizar. Por un lado se tienen tablas para representar las dimensiones y por otro lado tablas para los hechos (las factstables). Los diferentes DM están conectados entre sí por la llamada bus structure, que contiene los elementos anteriormente citados a través de las dimensiones conformadas (que permiten que los usuarios puedan realizar queries conjuntos sobre los diferentes DM, pues este bus contiene los elementos en común que los comunican). Una dimensión conformada puede ser, por ejemplo, la dimensión cliente, que incluye todos los atributos o elementos de análisis referentes a los clientes y que puede ser compartida por diferentes DM (ventas, pedidos, gestión de cobros, etc).



Este enfoque también se referencia como Bottom-up, pues al final el DW Corporativo no es más que la unión de los diferentes DM, que están



estructurados de una forma común a través de la bus structure. Esta característica le hace más flexible y sencillo de implementar, pues se puede construir un DM como primer elemento del sistema de análisis, y luego ir añadiendo otros que comparten las dimensiones ya definidas o incluyen otras nuevas. En este sistema, los procesos ETL extraen la información de los sistemas operacionales y los procesan igualmente en el área stage, realizando posteriormente el llenado de cada uno de los DM de una forma individual, aunque siempre respetando la estandarización de las dimensiones (dimensiones conformadas).

La metodología para la construcción del DW incluye 4 fases: Selección del proceso de negocio, definición de la granularidad de la información, elección de las dimensiones de análisis e identificación de los hechos o métricas. Igualmente define el tratamiento de los cambios en los datos a través de las Dimensiones Lentamente Cambiantes (SCD).

En resumen, los datos en sí mismos, así como los medios para obtener esos datos, para extraerlos, transformarlos y cargarlos, las técnicas para analizarlos y generar información, así como las diferentes formas para realizar la gestión de datos son componentes esenciales de un almacén de datos.



7.5 MODELADO DE DATOS

El modelo de datos es la representación de las estructuras de almacenamiento de un sistema. El mismo debe tener la estructura más eficiente de acuerdo al problema que debe resolver. Por ello es que el DW tiene que almacenarse en una estructura que resuelva todos los requerimientos del mismo.

La estructura óptima para un DW es un hipercubo, es decir, cubos OLAP (OnLine Analytical Processing) que son bases de datos multidimensionales.

➤ COMPONENTES

Los componentes del modelo multidimensional son los siguientes:

- *Hechos*: Son atributos numéricos que representan la performance, el comportamiento o la tendencia del negocio con respecto a una o más dimensiones. Son los parámetros para establecer controles. Se almacenan en tablas de hechos. Por ej. : cantidad y monto de productos vendidos, rentabilidad, costos.
- *Dimensiones*: Son los puntos de vista a través de los cuales se desean analizar los hechos. Organizan la información de manera que se pueda preguntar qué, cuándo, dónde... Las dimensiones son almacenadas en tablas junto con los elementos y atributos de dimensión. Por ej. : producto, región, tiempo, sucursales, clientes.
- *Elementos de Dimensión*: Componentes conceptuales de una dimensión. Están organizados jerárquicamente. El usuario final puede navegar por los distintos niveles de la jerarquía de la información. Por ej. :
Dimensión Tiempo: Año, Semestre, Cuatrimestre, Trimestre, Mes, Día.
Dimensión Producto: Familia, Rubro, Subrubro, Producto
- *Atributos de Dimensión*: Son las descripciones e información relacionada a cada elemento de dimensión.



- *Miembros de una Dimensión:* Son las instancias reales de una dimensión.
- *Tablas de Hechos:* Cada tabla de hechos contiene un registro por cada combinación de claves de las dimensiones.

La granularidad de la tabla está determinada por el nivel más bajo de cada tabla de dimensión. Granularidad se refiere al nivel de detalle contenido. A mayor nivel de detalle, menor granularidad. A menor nivel de detalle, mayor granularidad.

- *Tablas de Dimensión:* Están compuestas de elementos y atributos para cada nivel de la jerarquía. El nivel más bajo de detalle que es requerido para el análisis determina el nivel más bajo de la jerarquía. Son tablas desnormalizadas donde existe redundancia de datos.

➤ ESQUEMAS

Existen diferentes esquemas o estructuras de datos para el modelo multidimensional. **Se mencionarán las características del que se empleará en este proyecto: Estrella**

Esquema estrella (StarSchema): Consiste en una Tabla de Hechos (FactTable) rodeada de tablas por cada dimensión (Dimension Tables). Este aspecto, de tabla de hechos (o central) más grande rodeada de radios o tablas más pequeñas es lo que asemeja a una estrella, dándole nombre a este tipo de construcciones.

Las Tablas de Dimensiones tendrán siempre una clave primaria simple, mientras que en la tabla de hechos, la clave principal estará compuesta por las claves principales de las tablas dimensionales.

La Tablas de Hechos generalmente consisten de valores numéricos (datos asociados específicamente con el evento), y claves foráneas que referencian a tablas de datos dimensionales que guardan información descriptiva.



En las Tablas de Dimensión hay una gran redundancia de datos debido a los niveles de la jerarquía que se representan en una sola tabla.

Las Tablas de Hechos se designan para contener detalles uniformes a bajo nivel, o sea que los hechos pueden registrar eventos a un gran nivel de atomicidad. Esto puede resultar en la acumulación de un gran número de registros en la Tabla de Hechos, a lo largo del tiempo. Tablas de Hechos se definen como una de los siguientes tres tipos:

- *Tablas de Hechos transaccionales*, registran hechos relativos a eventos específicos (por ejemplo, el evento de una venta)
- *Tablas de Hechos Snapshot*, registran hechos en un punto dado en el tiempo -con un intervalo específico para realizar la medición sobre la entidad objeto de observación- (por ej., detalles de una cuenta al final del mes)
- *Tablas Snapshot Acumulativas*, registran hechos agregados -acumulados- a un punto dado en el tiempo (por ej., ventas totales mensuales para un producto dado)

Ejemplos de datos hechos incluyen: precio de venta, cantidad vendida, fecha y hora de venta, distancia, velocidad, y medidas de peso. Ejemplos de atributos dimensionales relacionados incluyen: modelo de producto, color del producto, tamaño del producto, localización geográfica y nombre del vendedor que realizó la venta.

Las Tablas de Dimensiones generalmente tienen un bajo número de registros, en comparación a las Tablas de Hechos, pero cada registro puede tener un gran número de atributos para describir los datos del hecho. Las Dimensiones pueden definir una amplia variedad de características, algunos de los atributos más comunes definidos en las Tablas de Dimensiones incluyen:



- *Tablas de Tiempo*, describen el tiempo al más pequeño nivel de granularidad de tiempo para el cual los eventos se registran en el esquema estrella.
- *Tablas de dimensión Geográficas*, describen datos de localización, tales como país, región, provincia, estado, o ciudad.
- *Tablas de dimensión de Productos*, describen los productos
- *Tablas de dimensión de Empleados*, describen los empleados, por ejemplo, los vendedores
- *Tablas de dimensión de Rangos*, describen rangos de tiempo, valores de dólar, u otras cantidades medibles para simplificar los reportes.



7.6 MINERIA DE DATOS

La minería de datos o exploración de datos (es la etapa de análisis de "Knowledge Discovery in Databases" o KDD) es un campo de la estadística y las ciencias de la computación referido al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos. Utiliza los métodos de la inteligencia artificial, aprendizaje automático, estadística y sistemas de base de datos. El objetivo general del proceso de minería de datos consiste en extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior.

La tarea de minería de datos real es el análisis automático o semi-automático de grandes cantidades de datos para extraer patrones interesantes hasta ahora desconocidos, como los grupos de registros de datos (análisis clúster), registros poco usuales (la detección de anomalías) y dependencias (minería por reglas de asociación). Esto generalmente implica el uso de técnicas de bases de datos como los índices espaciales. Estos patrones pueden entonces ser vistos como una especie de resumen de los datos de entrada, y pueden ser utilizados en el análisis adicional o, por ejemplo, en el aprendizaje automático y análisis predictivo. Por ejemplo, el paso de minería de datos podría identificar varios grupos en los datos, que luego pueden ser utilizados para obtener resultados más precisos de predicción por un sistema de soporte de decisiones. Ni la recolección de datos, preparación de datos, ni la interpretación de los resultados y la información son parte de la etapa de minería de datos, pero que pertenecen a todo el proceso KDD como pasos adicionales.

Un proceso típico de minería de datos consta de los siguientes pasos generales:

1. *Selección del conjunto de datos*, tanto en lo que se refiere a las variables objetivo (aquellas que se quiere predecir, calcular o inferir), como a las variables independientes (las que sirven para hacer el cálculo o proceso), como posiblemente al muestreo de los registros disponibles.



2. *Análisis de las propiedades de los datos*, en especial los histogramas, diagramas de dispersión, presencia de valores atípicos y ausencia de datos (valores nulos).
3. *Transformación del conjunto de datos de entrada*, se realizará de diversas formas en función del análisis previo, con el objetivo de prepararlo para aplicar la técnica de minería de datos que mejor se adapte a los datos y al problema, a este paso también se le conoce como *preprocesamiento* de los datos.
4. *Seleccionar y aplicar la técnica de minería de datos*, se construye el modelo predictivo, de clasificación o segmentación.
5. *Extracción de conocimiento*, mediante una técnica de minería de datos, se obtiene un modelo de conocimiento, que representa patrones de comportamiento observados en los valores de las variables del problema o relaciones de asociación entre dichas variables. También pueden usarse varias técnicas a la vez para generar distintos modelos, aunque generalmente cada técnica obliga a un preprocesado diferente de los datos.
6. *Interpretación y evaluación de datos*, una vez obtenido el modelo, se debe proceder a su validación comprobando que las conclusiones que arroja son válidas y suficientemente satisfactorias. En el caso de haber obtenido varios modelos mediante el uso de distintas técnicas, se deben comparar los modelos en busca de aquel que se ajuste mejor al problema. Si ninguno de los modelos alcanza los resultados esperados, debe alterarse alguno de los pasos anteriores para generar nuevos modelos.

Si el modelo final no superara esta evaluación el proceso se podría repetir desde el principio o, si el experto lo considera oportuno, a partir de cualquiera de los pasos anteriores. Esta retroalimentación se podrá



repetir cuantas veces se considere necesario hasta obtener un modelo válido.

Una vez validado el modelo, si resulta ser aceptable (proporciona salidas adecuadas y/o con márgenes de error admisibles) éste ya está listo para su explotación.

Un proyecto de minería de datos tiene varias fases necesarias que son esencialmente:

- Comprensión: del negocio y del problema que se quiere resolver.
- Determinación, obtención y limpieza: de los datos necesarios.
- Creación de modelos matemáticos.
- Validación, comunicación: de los resultados obtenidos.
- Integración: si procede, de los resultados en un sistema transaccional o similar.

Técnicas de Minería de Datos

Las técnicas de la minería de datos provienen de la inteligencia artificial y de la estadística, dichas técnicas, no son más que algoritmos, más o menos sofisticados que se aplican sobre un conjunto de datos para obtener unos resultados.

Las técnicas más representativas son:

- Redes neuronales: Son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. Algunos ejemplos de red neuronal son: el perceptrón, el perceptrón multicapa, los mapas autoorganizados, también conocidos como redes de Kohonen.



- Regresión lineal: Es la más utilizada para formar relaciones entre datos. Rápida y eficaz pero insuficiente en espacios multidimensionales donde puedan relacionarse más de 2 variables.
- Árboles de decisión: Un árbol de decisión es un modelo de predicción utilizado en el ámbito de la inteligencia artificial y el análisis predictivo, dada una base de datos se construyen estos diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva, para la resolución de un problema. Por ej. : Algoritmo ID3, Algoritmo C4.5.
- Modelos estadísticos: Es una expresión simbólica en forma de igualdad o ecuación que se emplea en todos los diseños experimentales y en la regresión para indicar los diferentes factores que modifican la variable de respuesta.
- Agrupamiento o Clustering: Es un procedimiento de agrupación de una serie de vectores según criterios habitualmente de distancia; se tratará de disponer los vectores de entrada de forma que estén más cercanos aquellos que tengan características comunes. Ej. : Algoritmo K-means, Algoritmo K-medoids.
- Reglas de asociación: Se utilizan para descubrir hechos que ocurren en común dentro de un determinado conjunto de datos.

Según el objetivo del análisis de los datos, los algoritmos utilizados se clasifican en supervisados y no supervisados (Weiss y Indurkha, 1998):

➔ Algoritmos supervisados (o predictivos): predicen un dato (o un conjunto de ellos) desconocido a priori, a partir de otros conocidos.



→ Algoritmos no supervisados (o del descubrimiento del conocimiento): se descubren patrones y tendencias en los datos.



8 RELEVAMIENTO Y/O DIAGNOSTICO Y CONCLUSIONES

8.1 Relevamiento

En los relevamientos para poder tener el diagnóstico se detectan los principales problemas que se van a tratar:

Inconformidad de los clientes en los relevamientos:

Se detectan insatisfacciones de los clientes tanto en las entregas de los proyectos (siempre se extienden sobre las fechas pactadas) y se detectan en producción errores funcionales en el Software entregado.

8.2 Diagnóstico

Como diagnóstico se observa que no tienen indicadores básicos para poder medir controlar y tomar decisiones sobre dónde están los problemas.

Problemas:

Falta de indicadores para medir la calidad de los desarrollos de software de cada recurso semestralmente.

Falta de indicadores para medir la calidad de los desarrollos de software de cada ciclo de vida (análisis, diseño, desarrollo, testing, pruebas de usuario, implementación).

Falta de indicadores para medir productividad de cada ciclo de vida de desarrollo de software (análisis, diseño, desarrollo, testing, pruebas de usuario, implementación).

Falta de indicadores para medir las reestimaciones de cada ciclo de vida de desarrollo de software (Análisis, diseño, desarrollo, testing, pruebas de usuario, implementación).



8.3 Conclusiones

Crear los indicadores para tener datos y así poder tomar decisiones de las causas raíces de los problemas.

Datos a trabajar:

- Cantidad de errores reportados vs. Cantidad de casos de prueba
- Cantidad de errores en cada fase del ciclo de vida del desarrollo de software (análisis, diseño, desarrollo, testing, pruebas de usuario, implementación).
- Cantidad de errores detectados por etapa (en qué etapa se ha encontrado el error: análisis, diseño, desarrollo, testing, pruebas de usuario, implementación, producción).
- Cantidad de errores por usuario cometido y encontrado.
- Cantidad de reestimaciones en cada fase del ciclo de vida del desarrollo de software (análisis, diseño, desarrollo, testing, pruebas de usuario, implementación).
- Cantidad de reestimaciones por usuario.

Los errores:

-Se clasifican como de análisis, diseño, construcción o históricos.

-Pueden reportarse en la etapa de testing, pruebas de usuario o en producción.

-Se le reportan a un analista-desarrollador particular.

Cada recurso (analista-desarrollador) corresponde a un sub-módulo.

Se registran las fechas de reporte del error y de solución del mismo.

Se registran las fechas re-estimadas de cada etapa.



9 PROPUESTA

Como propuesta, se estudiaron los mejores métodos para obtener indicadores de calidad y se tuvo en cuenta la simplicidad para que se adapten al proceso y herramienta que se van a usar en la compañía. **El método que se va a utilizar es GQM.**

Luego de tener armado lo que se medirá, se realizará una desnormalización de la BD productiva de GPIS, se realizara el analisis de los requerimientos hasta llegar al modelo conceptual y luego se utilizará un esquema estrella (StarSchema) nutrido por un ETL.

Posteriormente, para descubrir los patrones de comportamiento se aplicará la técnica de minería de datos descriptiva Reglas de Asociación, mediante WEKA (Waikato Environmentfor Knowledge Analysis, es decir, entorno para análisis del conocimiento de la Universidad de Waikato), que es una plataforma de software libre escrita en java para aprendizaje automático y minería de datos.

9.1 APLICACIÓN GQM SOBRE GPIS

En esta sección va el proceso de aplicar GQM (META-PREGUNTA-METRICA) sobre GPIS. Y se va a estar enfocado sobre dos objetivos haciendo referencia a la calidad. 1- Los tiempos de entrega 2- Los errores.

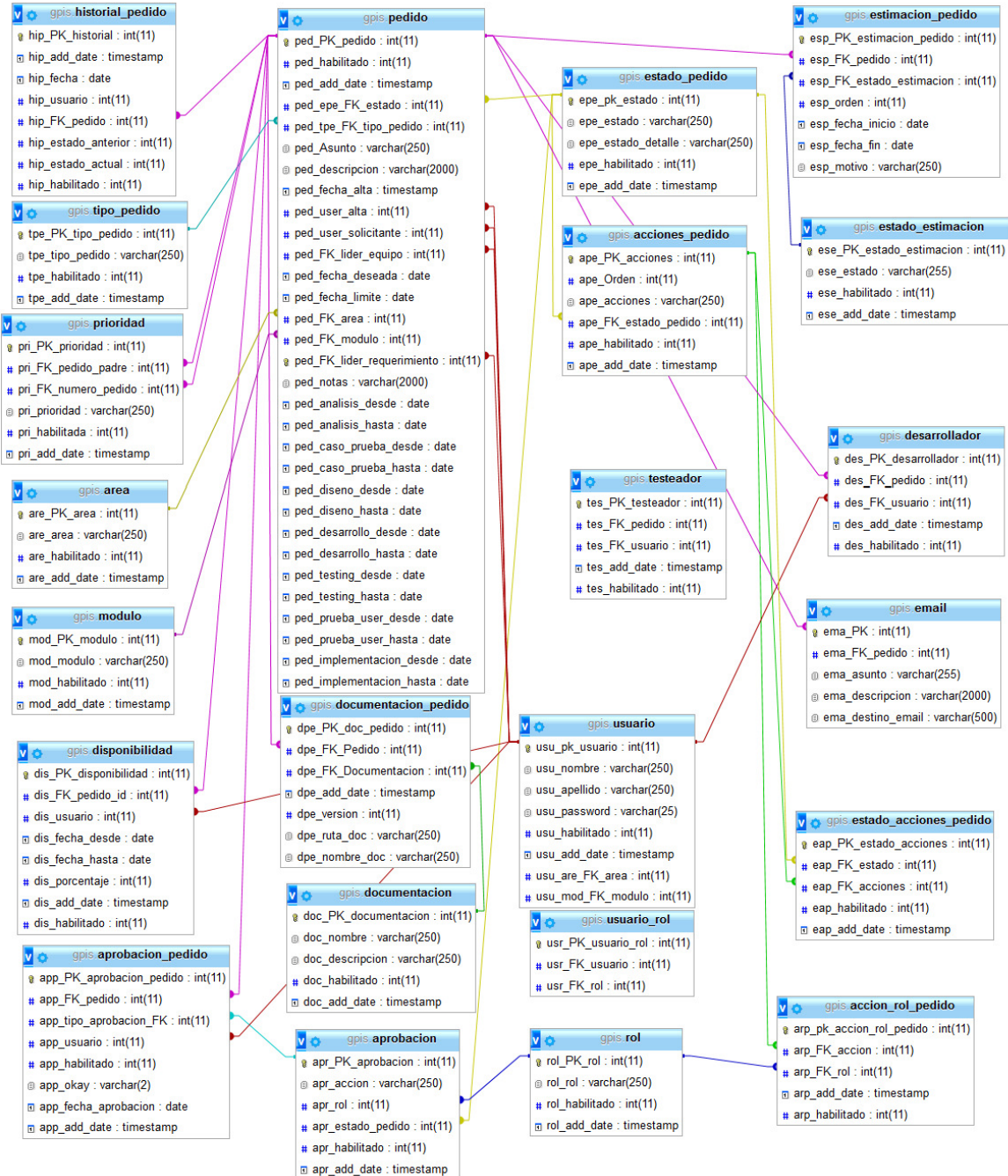
GQM	Objetivo 1	Objetivo 2
META	Cumplir con los tiempos de entrega	No implementar con errores
(Conceptual)		
PREGUNTA	Qué etapa del CVDS se re-estiman más las fechas	Qué etapa del CVDS Produce más errores
(Operativo)	Qué usuarios re-estiman más las fechas	En qué etapa del CVDS se detectan más errores
		Qué usuario tiene más errores
		Qué usuario detecta más errores
METRICA	Cuántas veces se re-estima cada proyecto	Cuántos CP tiene cada proyecto
(Cuantitativo)	Cuántas veces se re-estima cada etapa	Cuántos errores se detectan por proyecto
	Cuántas veces se reestima por usuario	Cuántos errores se detectan por etapa
		Cuántos errores se detectan en qué etapa
		Cuántos errores tiene asignado por usuario
		Cuánto es errores ha detectado cada usuario



9.2 Modelado del Data Warehouse

9.2.1 Base de Datos Operacional de GPIS.

A continuación se muestra el DER de la Base de datos relacional del cual se van a tomar los datos del ETL para poder volcar los datos en el DW.





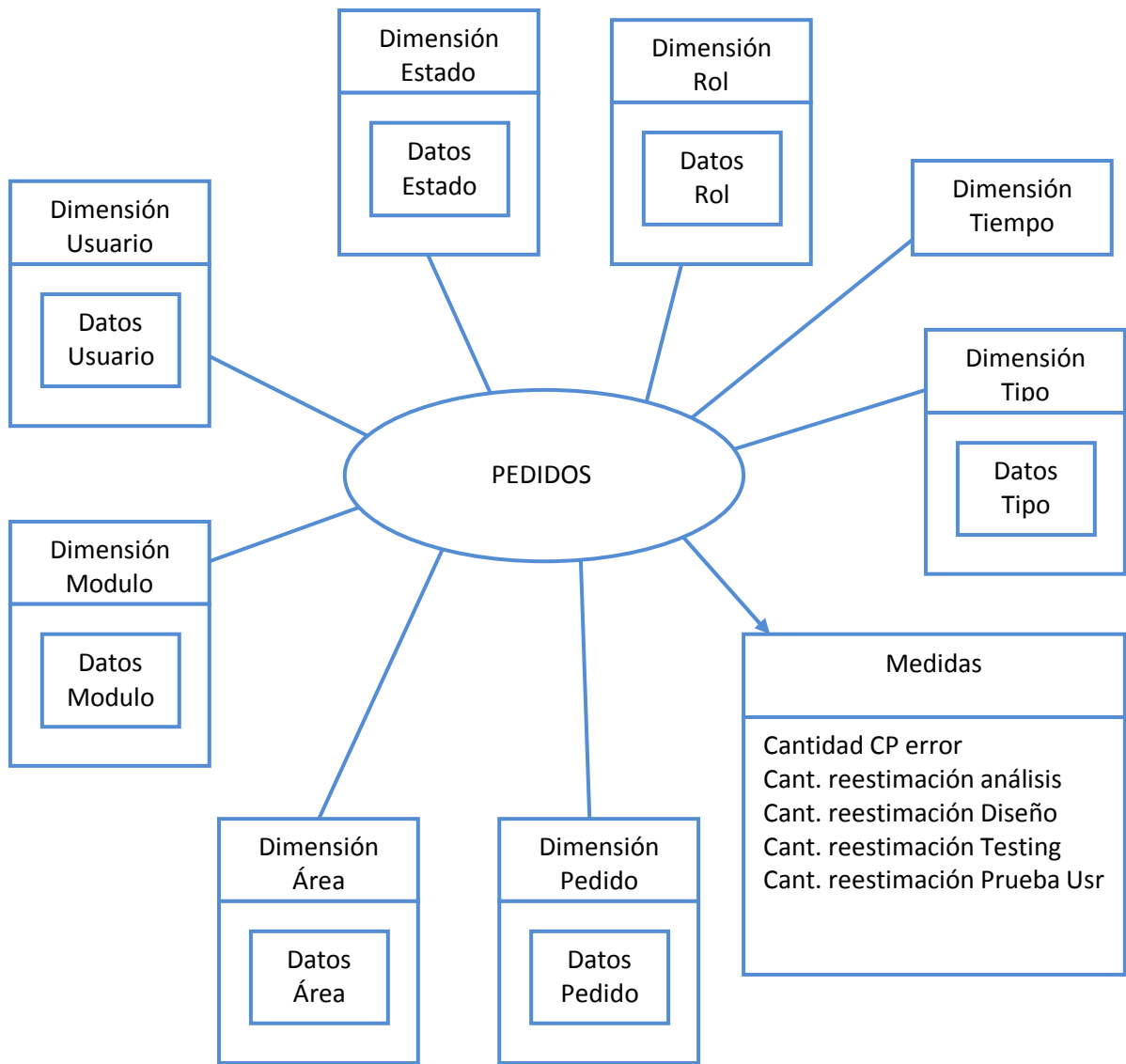
9.2.2 Análisis de las dimensiones.

→ 7 dimensiones de análisis:

- 1) Pedido: Con los detalles del requerimiento solicitado.
- 2) Estado: Con los estados en el que transcurre el requerimiento.
- 3) Modulo: Representa los módulos de IT de la empresa.
- 4) Tipo: Representa los distintos tipos de pedido (Nuevo, Mejoras, Correcciones).
- 5) Tiempo: Tiempo de permanencia de un proyecto en implementarse, en base a los semestres.
- 6) Usuario: Con los datos de los usuarios solicitantes.
- 7) Área: con los datos de las aéreas de de los usuarios solicitantes.



9.2.3 Modelo conceptual.





9.2.4 Esquema Estrella (StarSchema).

Modelado y definición de las tablas (HECHO y DIMENSIONES)

TABLA DE HECHOS

PEDIDOS	
FK	ID_pedido
FK	ID_usuario_origen
FK	ID_usuario_destino
FK	ID_estado_origen
FK	ID_estado_destino
FK	ID_rol_usr_origen
FK	ID_rol_usr_destino
FK	ID_modulo
FK	ID_area
FK	ID_tipo_pedido
FK	ID_tiempo
	Cantidad_CP_error
	Cantidad_reestimacion_analisis
	Cantidad_reestimacion_diseño
	Cantidad_reestimacion_testing
	Cantidad_reestimacion_prueba_usr

TABLAS DE DIMENSIONES

USUARIO	
PK	ID_usuario
	Nombre
	Apellido

ESTADO	
PK	ID_estado
	Estado

ROL	
PK	ID_rol
	rol

MODULO	
PK	ID_modulo
	Modulo

AREA	
PK	ID_area
	Area

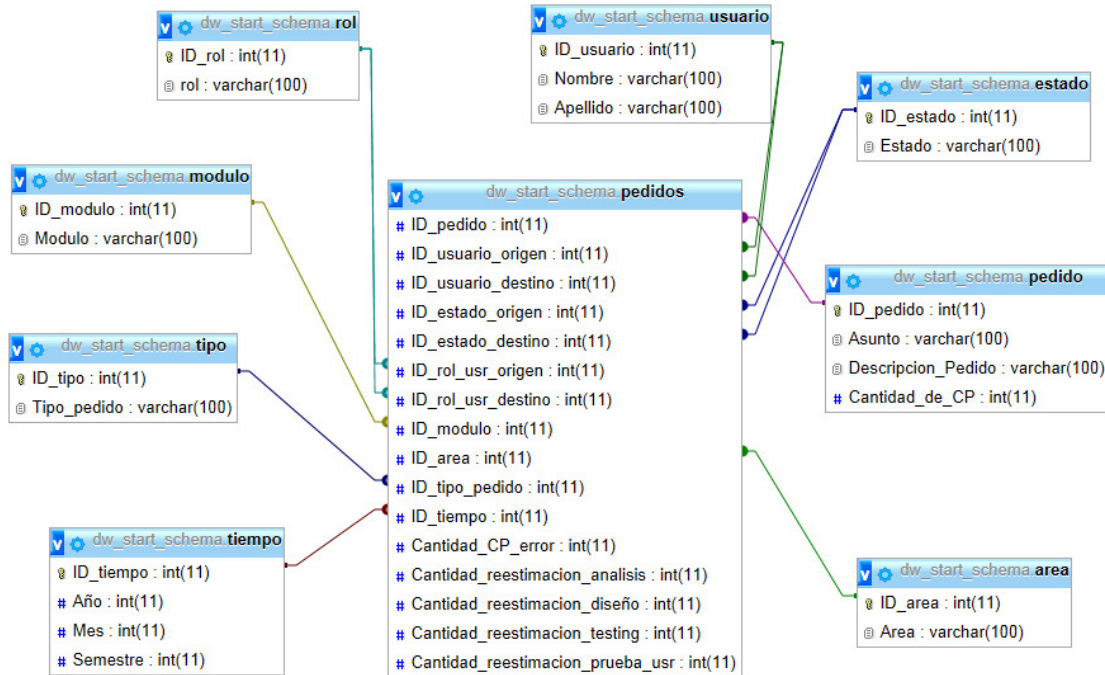
TIEMPO	
PK	ID_tiempo
	Año
	Mes
	Semestre

PEDIDO	
PK	ID_pedido
	Asunto
	Descripcion_Pedido
	Cantidad_de_CP

TIPO	
PK	ID_tipo
	Tipo_pedido



Esquema estrella (StartSchema)



9.2.5 ETL con PentahoPDI (Pentaho Data Integration).

Utilizaremos la herramienta PDI (Pentaho Data Integration) el cual es una ETL que nos permitirá extraer la información de la base de datos productiva de GPIS (BD MySQL), Transformar la información a través de un modelo dimensional y cargar los resultados de la transformación en una base de datos destino tipo Data warehouse, para que luego pueda ser consultada y analizada a través de herramientas para desarrollar reportes especializados las cuales Pentaho también posee.



PASOS PARA LA IMPLEMENTACION DE NUESTRO ETL

De manera resumida los pasos para la implementación de nuestro ETL son:

- Creación de la base de datos que contendrá el repositorio (BD MySQL).
- Creación del repositorio.
- Extracción de los datos desde la BD productiva de GPIS (BD MySQL)
- Creación de una tabla INPUT.
- Definición de las tablas de dimensiones
- Definición de la tabla de hecho
- Carga de la tabla de hecho

10 RESULTADOS

11 CONCLUSIONES DEL PG

12 BIBLIOGRAFIA

Referencias Iniciales

- Tesis presentada en diciembre del 2015, “Gestión de Proyectos de Ingeniería de Software” (GPIS)
- Minería de datos https://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos
- <https://es.wikipedia.org/wiki/Pentaho>
- <http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/viewFile/4989/4053>
- <https://prezi.com/8xgg-08nncwl/medidasmetricas-e-indicadores-de-la-calidad-del-software/>
- <https://prezi.com/uqwamihb-bup/gqm-goal-question-metric/>



- <http://asprotech.blogspot.com.ar/2010/09/goal-question-metric.html>
- <https://en.wikipedia.org/wiki/GQM>
- El Rincón del BI: <https://churriwifi.wordpress.com/2010/04/19/15-2-ampliacion-conceptos-del-modelado-dimENSIONAL/>
- Data Warehouse y Sistemas de Soporte de Decisión. Unidad III. Lic. José Luis Romanutti
- https://es.wikipedia.org/wiki/Esquema_en_estrella
- https://es.wikipedia.org/wiki/Esquema_en_copo_de_nieve
- https://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos

13 ANEXOS

N/A