

Modelo de Indicadores No Funcionales de un Sistema de Planeamiento de Recursos Empresariales (ERP). Caso de Estudio: EDITORAPERU S.A.

Jenny Susana Revilla Gonzales, Ángel Hermoza Salas

jrevilla@editorperu.com.pe, ahermoza@editorperu.com.pe

Universidad Inca Garcilaso de la Vega - Perú

Lima - Perú

Resumen: En este paper se definen indicadores de requerimientos no funcionales de un sistema ERP. Se plantea recolectar los requerimientos no funcionales que se genera dentro del propio sistema de planeamiento de recursos empresariales, de la base de datos y del sistema operativo. Para ello seguiremos el modelo planteado por Ian Somerville que define la ingeniería de requerimientos en dos: requerimientos funcionales y no funcionales. Dentro de los requerimientos no funcionales definiremos indicadores por cada requerimiento que nos ayudará a obtener los resultados que se mostrarán gráficamente.

Abstract: This paper defines non-functional requirements indicators of components in a organization. We propose to collect non-functional requirements generated within an ERP (enterprise resource planning) system, data base and operating system. This will follow the model proposed by Ian Somerville that defines requirements in engineering in functional and nonfunctional requirements. Within the non-functional requirements we define indicators for each request to help us get the results to be displayed graphically.

Palabras clave: Indicadores, Sistema para la toma de decisiones, Métricas de software, Ingeniería de Requerimientos, Requerimientos No funcionales, Sistemas ERP.

1. Introducción

Con el pasar de los años cada empresa ha ido desarrollando sistemas para satisfacer sus necesidades. Dentro de estos sistemas se encuentran los sistemas de planeamiento de recursos empresariales que brindan soluciones integrales para integrar los procesos de la empresa.

Según Ángel Hermoza [1], en la empresa Editora Perú S.A. se tiene el sistema de planeamiento de recursos empresariales Baan IV que procesa todas las operaciones y transacciones de la empresa. En varias oportunidades el sistema arroja mensajes de alertas, mensajes de error y mensajes de información requerida, inmediatamente el personal de la gerencia de informática (Administradores: BD Oracle, BAAN IV, Redes) procede a revisar manualmente el contenido de estos mensajes en los servidores para analizar lo ocurrido y dar una solución óptima. Como se muestra en la figura 1 se puede observar la situación actual de la empresa.

Para dar una solución, se plantea implementar un módulo de indicadores no funcionales en un sistema de planeamiento de recursos empresariales. Estos indicadores permitirán recolectar la información que se generan dentro del sistema de planeamiento de recursos empresariales Baan IV, de la base de datos Oracle 9i y del sistema operativo AIX 5.1. Luego se procederá a generar los archivos que permitirán identificar cada una de las transacciones realizadas por los administradores o por los usuarios y relacionarlas con los indicadores propuestos. Finalmente, se reportan los resultados y se registrara para futuros usos.

2. Trabajos Previos

Dentro de la literatura encontramos algunas definiciones:

Según SWEBOK [9]; “Los requerimientos no funcionales son los que actúan para restringir la solución. Son algunas veces conocidos como restricciones o requerimientos de calidad. Pueden ser clasificados de acuerdo con sí son requerimientos de performance, requerimientos de seguridad, requerimientos de mantenibilidad, requerimientos de confiabilidad o uno de muchos otros requerimientos de software”.

Según Somerville [4] [8] “Los requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergente de éste, como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema”.

Los tipos de requerimientos no funcionales, según Somerville:

1. **Requerimientos de Producto:** Estos requerimientos especifican el comportamiento del producto. Algunos ejemplos son los requerimientos de rendimiento en la rapidez de ejecución del sistema y cuanta memoria se requiere; los requerimientos de portabilidad y los requerimientos de usabilidad.
2. **Requerimientos Organizacionales:** Estos requerimientos se derivan de políticas y procedimientos existentes en la organización del cliente y en la del desarrollador. Algunos ejemplos son los estándares en los procesos que deben utilizarse; los requerimientos de implementación, como los lenguajes de programación o el método de diseño a utilizar, y los requerimientos de entrega que especifican cuando se entregara el producto y su documentación.

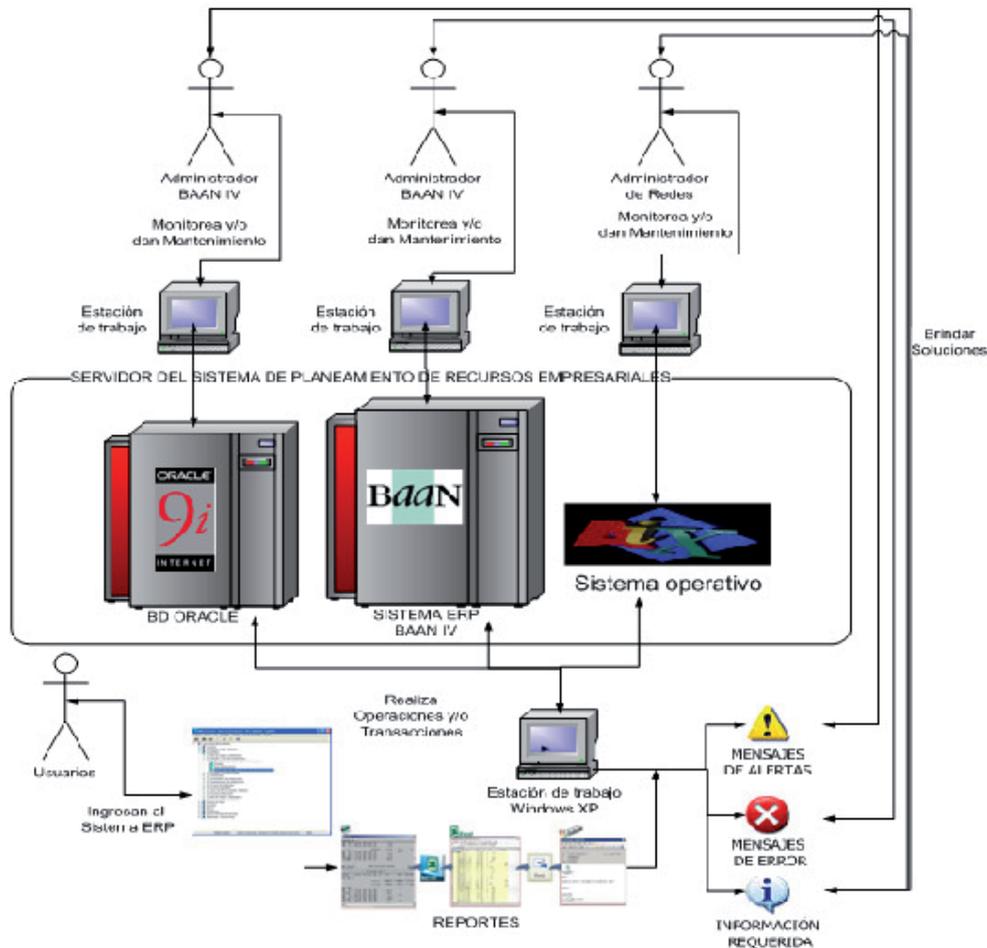


Figura 1. Representación gráfica del problema.

3. **Requerimientos Externos:** Este incluye todos los requerimientos que se derivan de los factores externos al sistema y de su proceso de desarrollo. Éstos pueden incluir requerimientos de interoperabilidad que definen la manera en que el sistema interactúa con sistemas de otras organizaciones; los requerimientos legislativos que deben seguirse para asegurar que el

sistema funcione dentro de la ley y los requerimientos éticos. Estos últimos son puestos en un sistema para asegurar que será aceptado por sus usuarios y por el público en general.

En la figura 2, se muestra la representación gráfica según el autor Somerville:

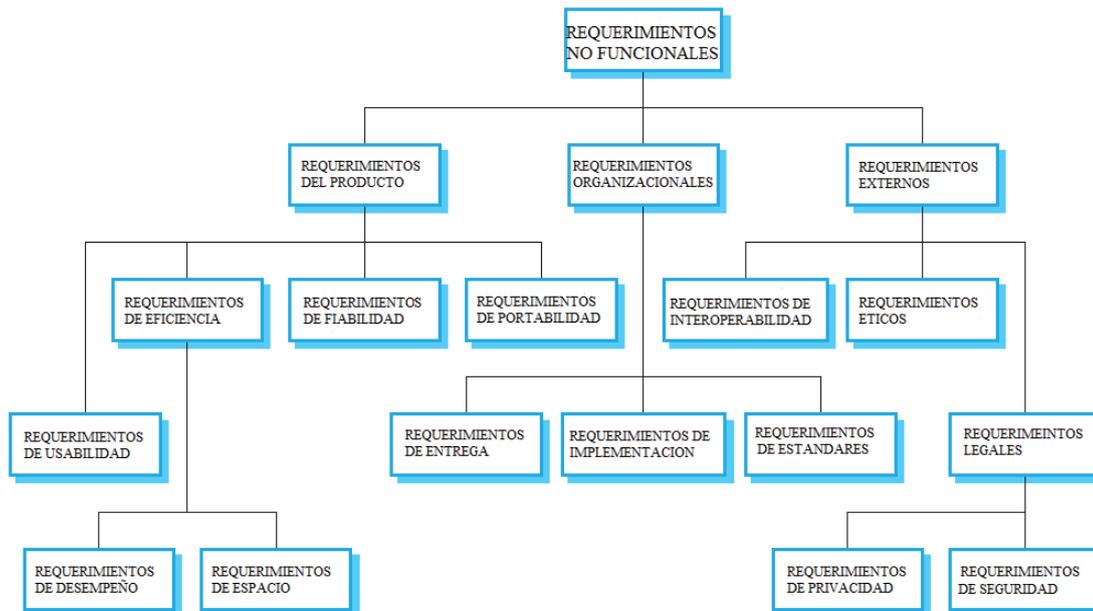


Figura 2. Diagrama de requerimientos no funcionales.

3. Otras secciones

3.1 Modelo de Indicadores:

Para la adaptación de los requerimientos no funcionales, se definirán los indicadores que se muestran en la tabla 1, según el modelo de Somerville de la figura 2.

Tabla 1. Indicadores de Requerimiento No Funcionales.

1. REQUERIMIENTOS NO FUNCIONALES			
Nivel 1	Nivel 2	Nivel 3	Métrica
1.1. REQUISA:		REQUASO REQUASD REQUASERP	%
Requerimiento de usabilidad			
1.2. REQEFI:	1.2.1. REQEFIDES:	REQEFIDESSO REQEFIDESBD REQEFIDESERP	
Requerimiento de eficiencia	Requerimiento de desempeño		
	1.2.2. REQEFIESP:	REQEFIESPSO REQEFIESPBD REQEFIESPERP	
1.3. REQFIA:		REQFIASO REQFIABD REQFIAERP	%
Requerimiento de fiabilidad			
1.4. REQPOR:		REQPORSO REQPORBD REQPORERP	
Requerimiento de portabilidad			
1.5. REQENT:		REQENTSO REQENTBD REQENTERP	%
Requerimiento de entrega			
1.6. REQIMP:		REQIMPESO REQIMPBD REQIMPERP	
Requerimiento de implementación			
1.7. REQUEST:		REQUESTSO REQUESTBD REQUESTERP	%
Requerimiento de estándares			
1.8. REQINT:		REQINTSO REQINTBD REQINTERP	
Requerimiento de interoperabilidad			
1.9. REQETI:		REQETISO REQETIBD REQETIERP	%
Requerimiento ético			
1.10. REQLEG:	1.10.1. REQLEGPRI	REQLEGPRISO REQLEGPRIED REQLEGPRIERP	
Requerimiento legal	Requerimiento de privacidad		
	1.10.2. REQLEGSEG	REQLEGSEGSO REQLEGSEGBD REQLEGSEGERP	

3.2 Fórmulas de los indicadores

1.1. **REQUISA:** Requerimiento de usabilidad (La aplicación debe ser de fácil manejo para el usuario).

Sea: $x, y \geq 0$; donde $0 \leq x \leq 100$; $0 \leq y \leq N$
 Cantidad total de usuarios = N ; que representa el 100%
 Cantidad de usuarios activos = y ; que representa $x\%$
 Usabilidad (i) = [Cantidad total de usuarios = N (100%)] – [Cantidad de usuarios activos = y ($x\%$)]
 Usabilidad (i) = $100\% - x\%$
 Componente (i) = Usabilidad(i); calculando la usabilidad por componente.

$$REQUISA = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0.$$

1.2. **REQEFI:** Requerimiento de eficiencia. (Este indicador se divide en dos: REQEFIDES y REQEFIESP).

$$REQEFI = (REQEFIDES + REQEFIESP) / 2; \text{ donde: } n > 0.$$

REQEFIDES: Requerimiento de desempeño (La aplicación debe realizarse en un plazo razonable).

Sea: $x, y \geq 0$; donde $0 \leq x \leq 100$; $0 \leq y \leq H$
 Cantidad total de horas activas = H ; que representa 100%
 Cantidad de horas no activas = y ; que representa el $x\%$
 Desempeño (i) = [Cantidad total de horas activas = H (100%)] – [Cantidad de horas no activas = y ($x\%$)]
 Desempeño (i) = $100\% - x\%$
 Componente (i) = Desempeño(i); calculando el desempeño por componente

$$REQEFIDES = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0.$$

REQEFIESP: Requerimiento de espacio (El espacio ocupado en el disco no debe ser mayor al espacio total).

Sea: $x, y, z \geq 0$; donde $0 \leq x \leq 100$; $0 \leq y \leq z$; donde $0 \leq z$
 Cantidad total de espacio = z ; que representa el 100%
 Cantidad de espacio libre = $\sum_{j=1}^m \text{Espacio}(j) = y$; que representa el $x\%$; donde $1 \leq j \leq m$
 Espacio (i) = [Cantidad total de espacio = z (100%)] – [Cantidad de espacio libre = y ($x\%$)]
 Espacio (i) = $100\% - x\%$
 Componente (i) = $\sum_{i=1}^n \text{Espacio}(i)$; calculando el espacio ocupado de cada disco por componente.

$$REQEFIESP = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0.$$

1.3. **REQFIA:** Requerimiento de fiabilidad (La aplicación debe ser capaz de recuperar los datos que tiene).

Sea: $x, y, z \geq 0$; donde $0 \leq x \leq 100$; $0 \leq y \leq N$
 Total de horas del día = N ; que representa el 100%
 Cantidad de tiempo medio de fallos = y ; que representa el $x\%$

$$\left(\sum_{k=1}^r \text{Componente}(k) \right) / n = y; \text{ que representa el } x\%$$
 Fiabilidad (i) = [Total de Horas del día = 100% - Cantidad de tiempo medio de fallos = y ($x\%$)]
 Fiabilidad (i) = $x\%$
 Componente (i) = Fiabilidad (i); calculando la fiabilidad por componente.

$$REQFIA = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0$$

1.4. **REQPOR:** Requerimiento de portabilidad (La aplicación debe ser ligera y no consumir demasiada memoria durante la ejecución del sistema).

Sea: $x, y \geq 0$; donde $0 \leq x \leq 100$; $0 \leq y \leq M$
 Cantidad total de memoria = M que representa el 100%
 Cantidad de memoria no utilizada = y ($x\%$)
 Portabilidad (i) = [Cantidad total de memoria = M (100%)] – [Cantidad de memoria no utilizada = y ($x\%$)]
 Portabilidad (i) = $100\% - x\%$

Componente (i) = Portabilidad (i); calculando la portabilidad por componente.

$$REQPOR = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0.$$

1.5. **REQENT:** Requerimiento de Entrega (Se especifican cuando se entregará el producto y su documentación).

Sea: $0 \leq x \leq 100$;

Donde: Completo= 100%, Incompleto= 75%,

Medio=50%, Deficiente=25%, Nulo=0%

Porcentaje de documentación existente = x%

Entrega (i) = Porcentaje de documentación existente = x%

Entrega (i) = x%

Componente (i) = Entrega (i); calculando la entrega por componente.

$$REQENT = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0.$$

1.6. **REQIMP:** Requerimiento de Implementación (Se debe considerar los lenguajes de programación y el método de diseño a utilizar).

Sea: $0 \leq x \leq 100$;

Donde: Completo= 100%, Incompleto= 75%,

Medio=50%, Deficiente=25%, Nulo=0%

Porcentaje de implementación = x%

Implementación (i) = Porcentaje de implementación = x%

Implementación (i) = x%

Componente (i) = Implementación (i); calculando la implementación por componente.

$$REQIMP = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0.$$

1.7. **REQUEST:** Requerimiento de Estándares (Se debe considerar estándares en los procesos que se van utilizar).

Sea: $0 \leq x \leq 100$;

Donde: Completo= 100%, Incompleto= 75%,

Medio=50%, Deficiente=25%, Nulo=0%

Porcentaje de estándares usados = x%

Estándares (i) = Porcentaje de estándares usados = x%

Estándares (i) = x%

Componente (i) = Estándares (i); calculando los estándares por componente.

$$REQUEST = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0.$$

1.8. **REQINT:** Requerimiento de Interoperatividad (Se definen la manera en que el sistema interactúa con sistemas de otras organizaciones).

Sea: $0 \leq x \leq 100$;

Donde: Muy Bueno= 100%, Bueno= 75%, Medio=50%,

Deficiente=25%, Nulo=0%

Porcentaje del nivel de compatibilidad= x%

Interoperabilidad (i) = Porcentaje del nivel de compatibilidad = x%

Interoperabilidad (i) = x%

Componente (i) = Interoperabilidad (i); calculando la interoperabilidad por componente.

$$REQINT = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0.$$

1.9. **REQETI:** Requerimiento Ético (El sistema debe funcionar dentro de los códigos éticos y deben ser aceptados por sus usuarios y por el público en general).

Sea: $0 \leq x \leq 100$;

Donde: Muy Bueno= 100%, Bueno= 75%, Medio=50%,

Deficiente=25%, Nulo = 0%

Porcentaje de aceptación del usuario = x%

Ética (i) = Porcentaje de aceptación del usuario = x%

Ética (i) = x%

Componente (i) = Ética (i); calculando los códigos éticos por componente.

$$REQETI = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0.$$

1.10. **REQLEG:** Requerimiento Legal (Este indicador se divide en dos: REQPRI y REQSEG).

$$REQLEG = (REQLEGPRI + REQLEGSEG) / 2; \text{ donde: } n > 0.$$

1.1. **REQLEGPRI:** Requerimiento Privacidad (La información estará protegida contra accesos no autorizados utilizando mecanismos de validación que puedan garantizar el cumplimiento de esto: cuenta, contraseña y nivel de acceso, de manera que cada uno pueda tener disponible solamente las opciones relacionadas con su actividad y tenga datos de acceso propios, garantizando así la confidencialidad).

Sea: $0 \leq x \leq 100$;

Donde: Muy Bueno= 100%, Bueno= 75%, Medio=50%, Deficiente=25%, Nulo = 0%

Porcentaje del nivel de privacidad = x%

Privacidad (i) = Porcentaje del nivel de privacidad = x%

Privacidad (i) = x%

Componente (i) = Privacidad (i); calculando la privacidad por componente.

$$REQLEGPRI = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0.$$

1.1. **REQLEGSEG:** Requerimiento Seguridad (Características de control de acceso al software y copias de seguridad, entre otros relacionados con la seguridad del sistema y la información. Se usarán mecanismos de encriptación de los datos que por cuestiones de seguridad no deben viajar al servidor en texto plano, como es el caso de las contraseñas. Se guardará encriptado esta información en la base de datos utilizando para ello MD5 como algoritmo de encriptación).

Sea: $0 \leq x \leq 100$;

Donde: Muy Bueno= 100%, Bueno= 75%, Medio=50%, Deficiente=25%, Nulo=0%

Porcentaje del nivel de seguridad = x%

Seguridad (i) = Porcentaje del nivel de seguridad = x%

Seguridad (i) = x%
 Componente (i) = Seguridad (i); calculando la seguridad por componente.

$$REQLEGSEG = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n; \text{ donde: } n > 0.$$

4. Experimentos y Resultados

Definición de componentes:

Para la aplicación del modelo de Ian Somerville, se procederá a seleccionar las herramientas que se van a utilizar para desarrollar la solución.

1. Componente (1): IBM AIX

Según IBM [12] [15], la herramienta AIX es un sistema operativo UNIX propiedad de IBM. AIX corre en los servidores IBM, utilizando procesadores de la familia IBM POWER de 32 y 64 bits.

2. Componente (2): ORACLE 9G

Según Oracle [17], la herramienta Base de Datos Oracle es un conjunto de datos que almacenan y recuperan información relacionada. Un servidor de base de datos es la clave para resolver los problemas de gestión de la información. En la tabla 2, se muestra las especificaciones técnicas de la herramienta.

Tabla 2. Características Técnicas de Oracle.

Oracle	32-bits	64-bits
Plataforma	Windows XP Professional Windows Seven	Windows XP Professional x 64 Edition Windows Seven x 64 Edition
Procesador	Pentium 4 o superior recomendado	Intel y procesadores AMD de 64 bits.
Memoria	Dependiendo del tamaño de las aplicaciones; mínimo 256 Mb.	Dependiendo del tamaño de las aplicaciones; mínimo 256 Mb.
Espacio en disco	Requerido 470 Mb. en total para instalación	Requerido 470 Mb. en total para instalación

3. Componente (3): ERP BAAN IV C2

Según IBM [16], el Sistema ERP Baan tiene los siguientes módulos: Manufactura, Distribución, Finanzas, Proyectos, Servicios, así como también se puede conectar con otros sistemas complementarios, tales como Cadena de Suministro (*Supply Chain*). En la tabla 3, se muestra las especificaciones técnicas de la herramienta.

Tabla 3. Características Técnicas de Baan IV.

Baan IV	32-bits	64-bits
Plataforma	Windows XP Professional Windows Seven	Windows XP Professional x 64 Edition Windows Seven x 64 Edition
Procesador	Pentium 4 o superior recomendado	Intel y procesadores AMD de 64 bits.

	superior recomendado	AMD de 64 bits.
Memoria	Dependiendo del tamaño de las aplicaciones; mínimo 512 Mb.	Dependiendo del tamaño de las aplicaciones; mínimo 2GB.
Espacio en disco	Requerido 120 Mb. en total para instalación	Requerido 120 Mb. en total para instalación

Pruebas: Para fines didácticos y debido al espacio disponible en el paper, sólo mostramos los resultados completos de los experimentos del indicador REQUISA (Indicador de Requerimiento No funcional – Usabilidad) y REQEFL.

Indicador REQUISA

Definimos los componentes para el caso de prueba: Sea n= 3:

Sea Componente (1) = REQUASO; el componente denominado Sistema Operativo donde reside el ERP.

Sea Componente (2) = REQUASABD, el componente denominado Base de Datos donde reside el ERP.

Sea Componente (3) = REQUASERP, el componente denominado aplicación ERP.

Reemplazando en la fórmula:

$$REQUISA = \left(\sum_{i=1}^n \text{Componente}(i) \right) / n$$

$$REQUISA = \left(\text{Componente}(1) + \text{Componente}(2) + \text{Componente}(3) \right) / 3$$

$$REQUISA = \left(\text{REQUASO} + \text{REQUASABD} + \text{REQUASERP} \right) / 3$$

Sea: REQUASO = 90; REQUASABD = 95 y REQUASERP = 99

Entonces:

$$REQUISA = (90+95+99)/3$$

$$REQUISA = 94.6 \%$$

Interpretación del indicador:

El indicador de usabilidad muestra el valor promedio de 94.6 %.

Interpretación de resultados del indicador REQUISA

En la Figura 3, se muestra los datos del indicador REQUISA, tomados el día lunes 25 de junio de 2012, desde las 8 am a 6 pm; mostrando los resultados por cada componente REQUASERP, REQUASABD y REQUASO. Se puede apreciar en el primer indicador existe una variación entre considerable de uso entre las 12 am y 3 pm similar al del indicador de BD.

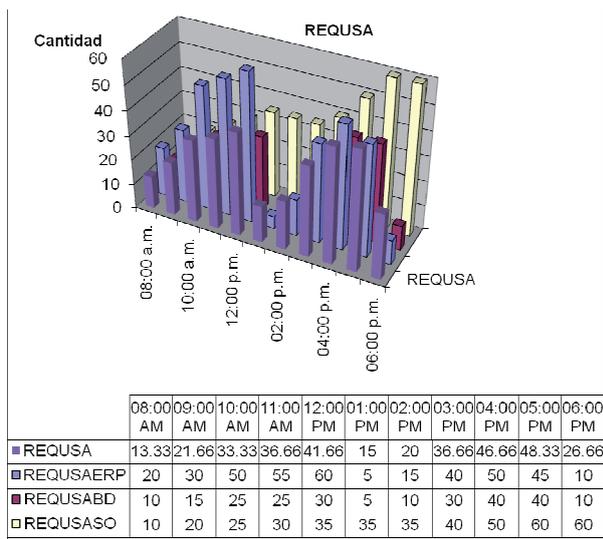


Figura 3. Estadística del Indicador REQUSA.

Indicador REQEFI

En la figura 4, se muestra los datos del indicador REQEFI, tomados el lunes 25 de junio de 2012, desde las 8 am a 6 pm; mostrando los resultados del componente REQEFIBD. Se puede apreciar que a partir de las 2pm en adelante se tiene una mayor eficiencia.

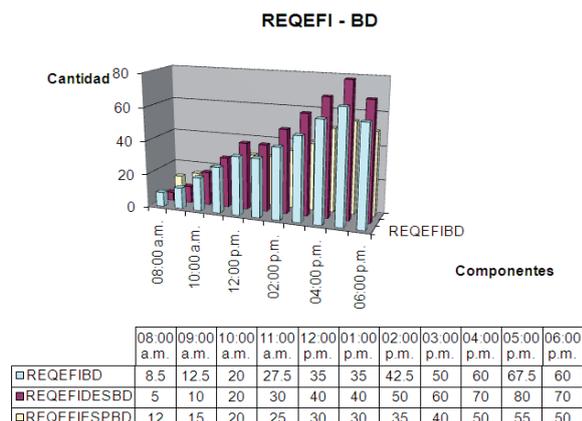


Figura 4. Estadística del Indicador REQUSA.

Discusión de los Experimentos

En la Figura 5, se muestra gráficamente la solución al problema planteado. Esta solución permitirá al personal de la Gerencia de Informática (Administradores: BD Oracle, BAAN IV, Redes) acceder a la información de la base de datos Oracle, sistema Baan IV y al sistema Operativo AIX, generar los indicadores y mostrarlos gráficamente mediante diferentes medios.

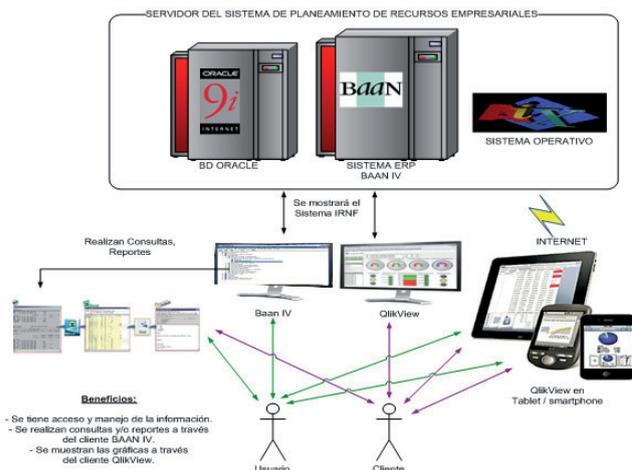


Figura 5. Solución gráfica del problema.

5. Conclusiones y trabajos futuros

- En el presente trabajo se ha definido un problema concreto dentro de una organización real.
- Se ha planteado una solución al problema, en base a la definición de requerimientos no funcionales de Ian Sommerville, proponiendo la creación de indicadores por cada uno de los requerimientos no funcionales.
- Se ha realizado las pruebas en un sistema de planeamiento de recursos empresariales.
- Se presenta dos ejemplos de indicadores como resultado de las pruebas realizadas tomando como base las fórmulas definidas.
- Se pretende, como trabajo futuro, la generalización de la presente propuesta de modo que todos los componentes puedan generar indicadores no funcionales.

Referencias bibliográficas

- [1] Angel Hermoza, Sistema de Indicadores No Funcionales de un Sistema de Planeamiento de Recursos Empresariales (ERP) utilizando Inteligencia de Negocios, Diciembre 2009.
- [2] Michael Arias Chaves, La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software, Volumen VI, Número 10, 2005, Edición Digital: 26 / 07 / 2007
- [3] Laudon K. y Laudon J., Information systems management: organization and technology, Prentice Hall, 2001. - 0132277816.
- [4] SOMMERVILLE, I.; Ingeniería de Software, Mexico 2002, Pearson Education; pags. 400.
- [5] María Cristina Carreón Suarez del Real, construcción de un catálogo de patrones de requisitos funcionales para erp, Universitat Politècnica de Catalunya, 25/06/2008.
- [6] Robertson, S., Robertson, J., Mastering the requirement Process. 1999: Addison- Wesley.
- [7] Nuseibeh, B., Easterbrook, S., Requirements Engineering: A Roadmap. International Conference

- on Software Engineering (ICSE2000), 2000: p. 35-46.
- [8] Sommerville, I., Integrated Requirements Engineering: A Tutorial. IEEE Software, 2005: p. 16- 23.
- [9] Guide to the Software Engineering Body of Knowledge, 2004 Version, IEEE Computer Society (2001-2003)
- [10] Carvallo, J., Franch, J., Quer, C., Managing Non-Technical Requirements in COTS Components Selection. 14th IEEE International Requirements Engineering Conference (RE'06). 2006.
- [11] Artículo de BusinessWeek Magazine hablando sobre la importancia de los KPIs: Giving the Boss the Big Picture: A dashboard pulls up everything the CEO needs to run the show (February 2006).
- [12] IBM Tivoli Composite Application Manager Family, January 2008
- [13] Direct Industry, El salón Virtual de la Industria
- [14] Artículo de GestioPolis, Sixtina Consulting Group, 13-03-2008.
- [15] IBM International Technical Support Organization, AIX 5L Practical Performance Tools and Tuning Guide, Abril-2005.
- [16] IBM International Technical Support Organization, Installing BAAN IV Applications on OS/390, Diciembre-1998.
- [17] Oracle 9i, Database Concepts, Marzo-2009.