

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

“Aplicación de PSP (Personal Software Process)
para el desarrollo de un sistema administrador de
códigos de barra a partir de la evaluación de Procesos
de Reingeniería”

INFORME DE PROYECTO DE GRADUACIÓN

Previa a la obtención del Título de:

**INGENIERO EN COMPUTACIÓN
ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN**

**INGENIERO EN COMPUTACIÓN
ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS**

Presentado por:
**ALBERTO RONALD CHÁVEZ SOLEDISPA
ENYL JIMMY DURÁN HOYOS**

**GUAYAQUIL – ECUADOR
Año 2010**

AGRADECIMIENTO

Agradecemos:

A Dios Todopoderoso por la culminación de este proyecto, a nuestros queridos padres y en general a todos quienes hicieron posible llevar a cabo este proyecto.

DEDICATORIA

A Dios,

A mis padres, hermanos

De manera especial a mi tío William Soledispa

Por su sacrificio y apoyo incondicional.

Alberto Ronald Chávez Soledispa

A Dios,

A mis padres, mis hermanos

Por brindarme todo su apoyo

Enyl Jimmy Durán Hoyos

TRIBUNAL DE SUSTENTACIÓN

Ing. Jorge Aragundi
SUBDECANO DE LA FIEC

Ing. Verónica Macías
DIRECTOR DE PROYECTO
DE GRADUACIÓN

Ing. Katherine Chiluza
MIEMBRO PRINCIPAL

Ing. Juan Moreno
MIEMBRO PRINCIPAL

DECLARATORIA EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestas en esta tesis, nos corresponden exclusivamente; y el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

Alberto Ronald Chávez Soledispa

Enyl Jimmy Durán Hoyos

ÍNDICE DE FIGURAS

Figura 2.1: Etapas del Proceso de Negocio.....	3
Figura 2.2: Enfoque del BPM.....	5
Figura 2.3: Dimensiones del BPM	8
Figura 2.4: Sistema del BPM	9
Figura 2.5: Etapas del Ciclo del BPM	11
Figura 2.6: Indicadores de Desempeño	16
Figura 3.5.1: LOC (Líneas de Código) por Mes	24
Figura 4.1.1: Proceso actual de Molemotor S.A.....	27
Figura 4.2.1: Proceso mejorado con códigos de barra	28
Figura 4.2.2: Esquema general de la solución	29
Figura 4.2.3: Caso de Uso para el SACB	31
Figura 5.1.1: Proceso de registro de marcaciones.....	33
Figura 5.4.1: Esquema propuesto	39
Figura 5.4.2: Registro de marcación.....	39
Figura 5.4.3: Esquema de marcaciones propuesto.....	40
Figura 5.4.4: Registro de actividades	40
Figura 5.6.1: Tiempo total para una OT sin usar SACB	49
Figura 5.6.2: Tiempo total para una OT usando SACB.....	50

Figura 5.6.3: Comparación de tiempos totales51

Figura 5.6.4: Tiempo total en minutos51

ÍNDICE DE TABLAS

Tabla I: Seguimiento de Tiempos	23
Tabla II: Pesos de puntos a evaluar	37
Tabla III: Métricas para concluir reingeniería	37
Tabla IV: Pesos obtenidos.....	38

ABREVIATURAS

PSP	:	<i>Personal Software Process</i>
BMP	:	<i>Business Process Management</i>
LOC	:	<i>Líneas de Código</i>
KLOC	:	<i>Miles de Líneas de Código</i>
SACD	:	<i>Sistema Administrador de Códigos de Barra</i>
MDT	:	<i>Módulo de Diarios de Trabajo</i>
MRM	:	<i>Módulo de Reporte de Marcaciones</i>
MMC	:	<i>Módulo de Marcaciones Cliente</i>
WSCD	:	<i>WebService de Codigos de Barra</i>
WSMC	:	<i>WebService de Marcaciones Cliente</i>
WSRH	:	<i>WebService de Recursos Humanos</i>
WSDT	:	<i>WebService de Diarios de Trabajo</i>
CRM	:	Customer relationship management
ERP	:	<i>Enterprise resource planning</i>
SLA	:	<i>Service level agreements</i>
KPI	:	<i>Key Perfomance Indicators</i>

ÍNDICE GENERAL

AGRADECIMIENTO.....	
DEDICATORIA	
TRIBUNAL DE GRADUACION	
DECLARACIÓN EXPRESA	
ÍNDICE GENERAL	
ÍNDICE DE FIGURAS	
ABREVIATURAS	
INTRODUCCIÓN	
CAPÍTULO 1	1
1. ESPECIFICACIONES GENERALES	1
1.1. Alcances.....	1
1.2. Objetivos	2

CAPÍTULO 2	3
2. ADMINISTRACIÓN DE LOS PROCESOS DEL NEGOCIO (BPM)	3
2.1 Definición de procesos y BPM	4
2.1.1. Definición de proceso	4
2.1.2. Definición de BPM	5
2.1 Tipos de procesos	11
2.2 Etapas del BPM	12
2.3 Indicadores clave de desempeño (KPI)	13
CAPÍTULO 3	17
3. PERSONAL SOFTWARE PROCESS (PSP)	17
3.1 Definición de PSP	17
3.2 Modelo del PSP	18
3.3 Administración del tiempo	20
3.4 Seguimiento del tiempo	21
3.5 Análisis basado en LOC	23

CAPÍTULO 4	25
4. IMPLEMENTACIÓN DEL SISTEMA ADMINISTRADOR DE CODIGOS DE BARRA (SACB)	25
4.1 Análisis del requerimiento	26
4.2 Diseño de la arquitectura	27
CAPÍTULO 5	32
5. IMPLEMENTACION DEL CASO: SISTEMA DE REPORTES DE DIARIOS DE TRABAJO	32
5.1 Análisis del proceso	33
5.3 Reingeniería del sistema.....	35
5.4 Evaluación de los resultados obtenidos de la reingeniería	38
5.5 Metodología a utilizar en la implementación del sistema.....	41
5.6 Mejoras obtenidas en base al sistema anterior	48

INTRODUCCIÓN

Como bien conocemos, en el proceso de desarrollo de software se dan inconvenientes que hacen que su tiempo de implementación aumente considerablemente de acuerdo a lo previsto. Este tipo de retrasos no solamente se presenta a nivel estudiantil sino en la industria en general ya que estos inconvenientes se ven reflejados no solo en la demora de la entrega del sistema sino también en la calidad del mismo; afectando con esto la imagen de la empresa y de la persona encargada del proyecto.

Dada esta situación queremos representar mediante un análisis en el proceso de desarrollo, cuáles son las causas que hacen que nuestros sistemas o proyectos no cumplan con su tiempo de implementación. Para esto hemos utilizado la metodología PSP para hacer una evaluación de la eficiencia de cada persona en el proyecto.

Nuestro proyecto nació de evaluar la necesidad de la empresa Molemotor S.A. en mejorar sus procesos de validación y búsqueda para lo cual se realizó en conjunto con el personal de sistemas un análisis del proceso de negocio que tenían implementado para que

en base a esta información se realice una reingeniería que mejore sus procesos.

CAPÍTULO 1

1. ESPECIFICACIONES GENERALES

1.1. Alcances

Luego de reuniones periódicas sostenidas con el personal de sistemas de la empresa Molemotor S.A. y contando con la aprobación del Gerente General hemos determinado como alcance de nuestro proyecto lo siguiente:

- ✓ Mejorar el proceso de lectura y búsqueda en los sistemas actuales de la empresa Molemotor S. A. a través del desarrollo del Sistema Administrador de Códigos de Barra –SACB- que le permita obtener con mayor eficiencia la información que necesitan.
- ✓ Cuantificar la mejora de los procesos luego de la implementación del SACB utilizando como ejemplo un sistema piloto basado en su Sistema de registro de Diarios de Trabajo.
- ✓ Presentar resultados que muestren las mejoras de implementación obtenidas al hacer uso de la metodología para

- ✓ desarrollo personal -PSP-.
- ✓ Dar a conocer la eficiencia en los procesos de obtención de información de la empresa Molemotor S. A.

1.2. Objetivos

En línea con cada uno de los alcances descritos anteriormente se tiene como objetivos principales los siguientes puntos:

- ✓ Conocer nuestras habilidades y deficiencias en el análisis, diseño, implementación y pruebas de un sistema usando la metodología PSP.
- ✓ Obtener métricas que nos ayuden a evaluar nuestro rendimiento y eficiencia al desarrollar sistemas.
- ✓ Obtener una retroalimentación personal de la forma en que usamos el tiempo en cada una de nuestras actividades ya sea en el ámbito laboral como en lo estudiantil.
- ✓ Reestructurar el proceso de toma de información y búsqueda en los sistemas como parte de una estrategia orientada a la mejora de los procesos generales del negocio –BPM-.

CAPÍTULO 2

2. ADMINISTRACIÓN DE LOS PROCESOS DEL NEGOCIO (BPM)

Dentro de una organización es importante definir los procesos de negocio los cuales son el fundamento principal de su funcionamiento y es a su vez son directrices para su administración y mejora continua.

La gestión de estos procesos nos ayuda a mejorar la eficiencia a través de modelos, automatización, integración y optimización de forma continua de los procesos definidos.



Figura 2.1: Etapas del Proceso de Negocio

2.1 Definición de procesos y BPM

2.1.1. Definición de proceso

Un proceso se define como un conjunto de tareas, actividades o acciones inter-relacionadas entre sí, las cuales reciben varias entradas y dan como resultados salidas con un valor agregado.

Hay tres elementos importantes en un proceso:

- Valor agregado: Aquellos que transforman los datos e insumos para crear información y productos o servicios para el cliente.
- Traspaso (flujo): Aquellas en las que se entrega de manera interdepartamental o externa la información y productos.
- Control: Aquellas que permiten que las actividades de traspaso se lleven a cabo de acuerdo a especificaciones previas de calidad, tiempo y costo establecido.

Algunos ejemplos de procesos pueden ser los de producción de bienes, entrega de productos o servicios, el de gestión de las relaciones con los clientes -habitualmente gestionada por un sistema CRM-, el de desarrollo de nuevos productos o servicios, etc.

Estos procesos deben estar correctamente gestionados empleando distintas herramientas de gestión de procesos -en definitiva gestión de

la organización- como puede ser un sistema de planificación de recursos empresariales (ERP), un sistema de Workflow entre otros.

2.1.2. Definición de BPM

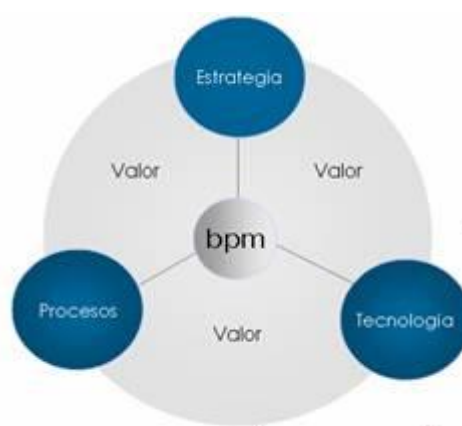


Figura 2.2: Enfoque del BPM

Smith Howard por su parte, define BPM como una nueva aproximación para abordar y gestionar procesos de innovación en las compañías que construye el mejoramiento, a partir del estado actual de un proceso en un momento determinado y que plantea una diferencia radical frente a la reingeniería; la cual construye el mejoramiento desde la redefinición total del proceso. En esta óptica BPM se convierte en una respuesta al caos operativo que presentan las

compañías en la actualidad.

Beneficios de BPM

La aplicación de BPM tiene una serie de beneficios para las empresas.

Uno de esos beneficios es el enfoque de “Las 3 C” (comunicación, colaboración y coordinación) el cual es un modo de trabajo entre empresas internas o externas. Para esto es necesario recurrir a tecnologías que orquesten los procesos, organización, sistemas, clientes, colaboradores y otros entes externos.

Los casos en los cuales se ha utilizado el concepto, han reportado beneficios que van desde la mejora en las capacidades de dirección, pasando por la reducción de obstáculos al momento de reaccionar ante cambios del mercado, hasta adquirir mayor capacidad de análisis sobre el desempeño de la empresa. Los siguientes son otros beneficios identificados:

- ✓ Visibilidad de los procesos de las empresas.
- ✓ Mayor flexibilidad y agilidad para adaptación al cambio.
- ✓ Posibilidad de integrar la información del negocio dispersa en diferentes sistemas.
- ✓ Dirigir los esfuerzos de la empresa de una manera planeada y alineada con los objetivos estratégicos.

- ✓ Adquirir la habilidad para diseñar, simular y monitorear procesos de manera automática y sin la participación de usuarios técnicos.
- ✓ Adquirir una ruta de mejoramiento y eficiencia continua al convertir actividades ineficientes en menores costos a través de uso de tecnología enfocada en procesos.
- ✓ Reducir costos futuros de integración y mantenimiento al adquirir tecnología ya preparada para abordar el cambio.

Lograr estos beneficios es el resultado de la aplicación metódica de prácticas de gestión, de la implantación y adopción de formas de operar automatizadas y estratégicamente seleccionadas.

En la práctica la implantación de esta disciplina de mejoramiento requiere por parte de la empresa, una dosis de pensamiento en procesos de negocio y la utilización de tecnologías de Información centradas en procesos.



Figura 2.3: Dimensiones del BPM

Tecnología de BPM

La tecnología que posibilita la implantación y adopción de BPM constituye una categoría nueva de sistemas de información denominada Business Process Management System (BPMS).

Inicialmente y de manera general un BPMS puede ser definido como un conjunto de utilidades de software para definir, implementar y mejorar procesos de negocio que cumplen con un grupo de características técnicas necesarias para aplicar el concepto de BPM.

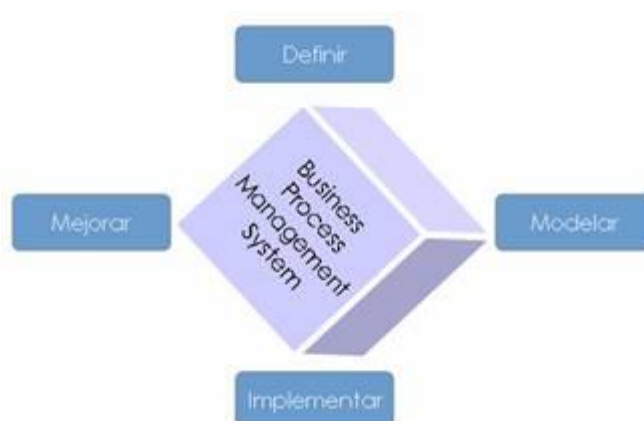


Figura 2.4: Sistema del BPM

BPMS apoya en todo el ciclo de vida de los procesos de negocio. Sus módulos principales que componen la plataforma BPMS, y que apoyan las etapas del ciclo son:

- **Modelador Gráfico de Procesos:** *-Business Modeler-* Este permite modelar los procesos de negocio, simular su ejecución, definir métricas para el monitoreo, y exportar a BPEL (lenguaje estándar de procesos). Tiene un diseñador gráfico de procesos, que permite fácilmente crear los modelos.
- **Ambiente Integración y Desarrollo:** *-Integration Developer-* Es la herramienta que permite implementar los procesos, y servicios. Esta herramienta permite integrar las pantallas -para interacción de un participante-, y los servicios -interacción con

sistemas legados-.

- **Servidor de Procesos de Negocio:** *-Process Server-* Es el motor que permite ejecutar los procesos de negocio, aquí se ejecutan las Aplicaciones Compuestas - flujos BPM-, los Workflows tradicionales, y la Orquestación de Servicios - procesos compuestos solo por servicios-. Este servidor también es el encargado de generar los datos de las métricas, y de monitoreo. Permite intervenir los procesos en tiempo real: balancear carga, cambiar flujo de negocio, y realizar acciones correctivas -según reglas de negocio-.
- **Monitor de Actividades de Negocio:** *-BAM, Business Activity Monitor-* Esta es una aplicación de administración que permite gestionar los procesos y servicios, gráficamente se pueden ver indicadores de performance, y SLA *-Service Level Agreements,* niveles de servicio a cumplir-. Se puede además definir alertas y triggers de acuerdo a eventos de negocio que sucedan en el proceso. También puede proveer datos reales a los modelos *- Business Modeler-* para ajustar las simulaciones -y lograr mejoramiento continuo-.

La siguiente imagen muestra como se ve la herramienta BPMS.

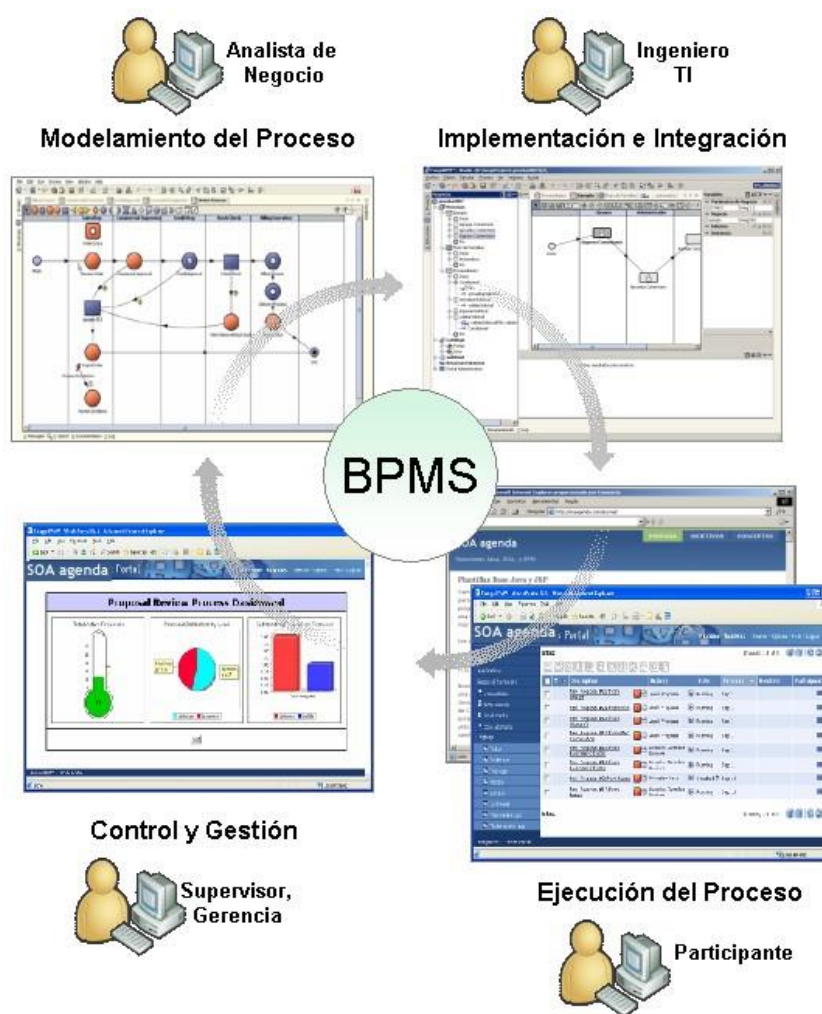


Figura 2.5: Etapas del Ciclo del BPM

2.1 Tipos de procesos

Los procesos pueden ser clasificados en función de varios criterios.

Pero la clasificación de los procesos más conocida es distinguir entre:

estratégicos, claves o de apoyo.

Procesos clave: Son también denominados operativos y son propios de la actividad de la empresa, por ejemplo, el proceso de aprovisionamiento, el proceso de producción, el proceso de prestación del servicio, el proceso de comercialización, etc.

Procesos estratégicos: Son aquellos procesos mediante los cuales la empresa desarrolla sus estrategias y define los objetivos. Por ejemplo, el proceso de planificación presupuestaria, proceso de diseño de producto y/o servicio, etc.

Procesos de apoyo: Son los que proporcionan los medios (recursos) y el apoyo necesario para que los procesos clave se puedan llevar a cabo, tales como proceso de formación, proceso informático, proceso de logística, etc.

2.2 Etapas del BPM

Las etapas del BPM muestran en resumen los diferentes enfoques que en conjunto permiten obtener una visión clara del negocio y su gestión a nivel de procesos. Estas etapas son:

Modelamiento de los Procesos de Negocio: En esta etapa se crea o modela un proceso de negocio, también es aquí donde se definen

mejoras, o cambios a los procesos para optimizarlos. En esta etapa el principal involucrado es el “Analista de Negocios”.

Implementación: En esta etapa se integran los componentes necesarios para implementar el proceso. El principal involucrado en esta etapa es el “Ingeniero de TI” para el caso de que los procesos se implementen como soluciones tecnológicas.

Ejecución de Procesos: Esta es la etapa en donde se explota el proceso desarrollado previamente, en esta etapa los principales involucrados son los “Participantes” del proceso. Además aquí es cuando se recolecta la información para control, y seguimiento.

Control y Gestión: Esta es la etapa donde se le da seguimiento a los procesos, y donde se analiza la información de su ejecución, por ejemplo: indicadores de desempeño, cuellos de botella, caminos críticos, carga de trabajo, etc., su principal características es que la información se analiza en tiempo real. En esta etapa los principales involucrados son los “Supervisores, y la Gerencia”.

2.3 Indicadores clave de desempeño (KPI)

Luego de haber definido claramente los procesos de negocio a los cuales se van a realizar un análisis es necesario tener una medición de los resultados obtenidos luego de cada ciclo de mejoramiento de

los mismos.

Como respuesta a esta necesidad de cuantificar los resultados, toman gran importancia la definición de indicadores de desempeño -*Key Performance Indicators KPI*- los cuales nos permiten tener una idea cuantificada del nivel de mejora a la que nuestros procesos se encuentran sujetos. Nos permiten conocer el “cómo” de nuestro proceso para llegar a los objetivos propuestos.

Regularmente estos indicadores están ligados a una inteligencia de negocio para lograr asistir o ayudar al estado actual de las empresas con el fin de valorar las actividades complicadas de medir tales como servicio, satisfacción del empleado y principalmente clientes.

Estos KPI's son utilizados para calcular:

- Tiempo que se utiliza para mejora de un servicio
- Nivel de satisfacción del cliente
- Tiempos relacionados con un SLA.
- Impacto de la calidad de los recursos para realizar un servicio.

Dentro de una organización es importante que al menos se definan sus propios KPI's, para esto es necesario considerar:

- Tener predefinido sus procesos de negocio

- Tener una visión clara de sus objetivos y rendimientos de sus procesos.
- Tener una medida cuantitativa/cualitativa de los resultados para posteriormente realizar un contraste de objetivos.
- Investigar variaciones y ajustar procesos o recursos para metas a corto plazo.

Cuando se definen estos indicadores se suele aplicar el acrónimo

SMART, ya que estos KPI's tienen que ser:

- Especificos (S)pecific)
- Medibles (M)esurables)
- Alcanzables (A)chievable)
- Realista (R)ealistic)
- a Tiempo (T)imely)

Su representación gráfica puede variar entre, barras de desplazamiento, semáforos, velocímetros, etc.

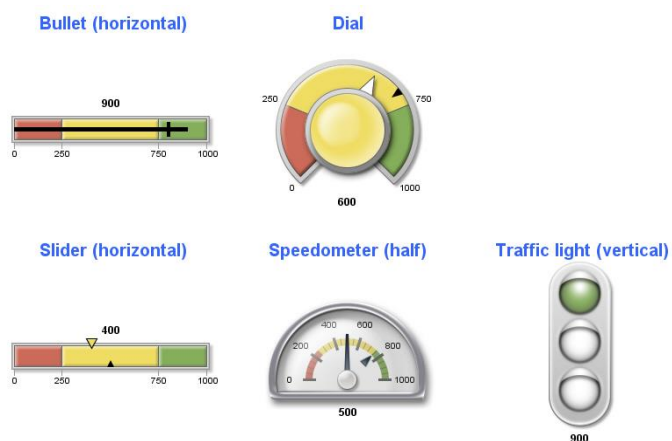


Figura 2.6: Indicadores de Desempeño

Esto puede ser resumido en lo siguiente:

1. Los datos de los que dependen los KPI tienen que ser consistentes y correctos.
2. Estos datos tienen que estar disponibles a tiempo.

CAPÍTULO 3

3. PERSONAL SOFTWARE PROCESS (PSP)

El Personal Software Process (PSP) brinda a los ingenieros la posibilidad de mejorar su desarrollo de software estableciendo un marco de disciplina personal. El proceso PSP consiste en un conjunto de métodos, formas que muestran como planificar, medir y administrar su propio trabajo.

El principal propósito es identificar y mejorar la eficiencia de nuestras actividades.

3.1 Definición de PSP

Es un conjunto de prácticas disciplinadas para la gestión del tiempo y mejora de la productividad personal de los programadores o ingenieros de software, en tareas de desarrollo y mantenimiento de sistemas. Está alineado y diseñado para emplearse en organizaciones con modelos de procesos CMMI o ISO 15504.

Se la puede considerar como una guía de trabajo personal para

ingenieros de software en organizaciones con un nivel de madurez avanzada o de capacidad de procesos que implica la medición cualitativa y mejora de los procesos.

De manera general, el principal problema es la gran cantidad de datos que hay que tomar. Estos datos nos ayudan en la elaboración de tablas.

3.2 Modelo del PSP

El PSP se divide en etapas graduales de crecimiento llamadas PSP0, PSP1, PSP2 y PSP3.

PSP0: Acepta las prácticas de desarrollo actuales del ingeniero pero requiere:

- Mantener un registro del tiempo dedicado a trabajar en un proyecto
- Registrar los defectos encontrados
- Registrar los tipos de defectos

PSP0.1: Se requiere establecer:

- Una manera estándar para definir una “línea de código” definido como LOC.
- Un marco de trabajo dentro del cual el individuo puede observar maneras de mejorar su proceso de desarrollo.

PSP1: Proceso de planificación personal. Está diseñado para ayudar al ingeniero a entender la relación entre el tamaño de los programas y el tiempo que toma desarrollarlos. Su propósito es proporcionar un “marco de trabajo ordenado” dentro del cual el individuo pueda realizar estimaciones. Hacer compromisos, evaluar el estado y registrar los resultados.

PSP1 agrega a PSP0 las siguientes aptitudes:

- Aptitud para estimar el tamaño.
- Marco de trabajo para informar los resultados de las pruebas.

PSP1.1: Agrega la habilidad para realizar:

- Tareas de programación del plan.
- Tareas de programación de tiempos.

PSP2: Proceso de administración de la calidad personal. El PSP2 está diseñado para ayudar a los ingenieros a “manejar de manera realista y objetiva” los defectos de programación. La idea es estimar tantos defectos como sea posible antes de someter el programa a una inspección formal.

- Revisión personal del diseño.
- Revisión personal del código.

PSP2.1: Agrega un marco de trabajo y lista de verificación para asegurar que se completen los diseños.

PSP3: Proceso Personal Cíclico. Está diseñado para escalar el PSP para manejar las unidades de código grandes (en miles de líneas) dividiendo un programa grande en pequeños incrementos. PSP3 agrega:

- La aplicación de PSP a cada incremento para producir una alta base de calidad para los incrementos sucesivos.
- El uso de pruebas de “regresión” para asegurar que las pruebas diseñadas para los incrementos anteriores todavía son buenas en los nuevos incrementos.

3.3 Administración del tiempo

Nos muestra cómo administrar el tiempo en función de nuestras actividades y a su vez conocer cuán importante es realizar un seguimiento del mismo. Para llegar a este propósito es importante lograr identificar nuestras actividades y el tiempo involucrado en la ejecución de cada una de ellas.

Es importante mencionar que durante el proceso existirán actividades que de cierta forma nos parecerán irrelevantes, más sin embargo el conocerlas nos hará hábiles en evitar ciertos atrasos en nuestras

tareas.

Para practicar la administración del tiempo es necesario comprender en primer lugar como utilizamos nuestro tiempo. Para este fin vamos a realizar los siguientes pasos:

1. Categorizar la mayoría de nuestras actividades
2. Registrar el tiempo que se pasa en tales actividades
3. Registrar este tiempo en una forma estándar
4. Guardar estos registros en un medio fácil de utilizar

3.4 Seguimiento del tiempo

Para esta actividad nos resultará útil tener una unidad de tiempo representativa a nuestro trabajo.

Regularmente asociamos el tiempo en base a horas de trabajo (horas/hombre). Esta apreciación si bien es cierto es muy utilizada pero no representa de forma real el tiempo que ocupamos en una tarea. La razón primordial es que no siempre se cumple con una hora exacta de trabajo, sino más bien se cumplen en unidades menores. Es decir, minutos de trabajo continuo.

En el continuo registro del tiempo vamos a encontrar una cantidad de interrupciones que posteriormente nos ayudaran a cuantificar la cantidad

de trabajo aplicado y los resultados obtenidos.

Los formularios utilizados para el registro de las actividades constan de los siguientes parámetros:

<i>Fecha</i>	:	La fecha en que se inicia la actividad.
<i>Inicio</i>	:	El tiempo en que se inicia la actividad.
<i>Fin</i>	:	El tiempo en que detiene la actividad.
<i>Interrupción</i>	:	La suma del tiempo perdido durante la actividad.
<i>Tiempo delta</i>	:	El tiempo tomado para la actividad. Este se basa en la resta del tiempo final menos el inicial y menos el tiempo de interrupción.
<i>Actividad</i>	:	Un nombre descriptivo de la actividad.
<i>Comentario</i>	:	Alguna nota que complementa la actividad o de más detalle de lo que se está realizando.
<i>C</i> <i>(Completado)</i>	:	Es un verificador que indica si dicha tarea ha sido completada.
<i>U (Unidad)</i>	:	El número de unidades en una actividad cuando fue completada.

Como ejemplo de un formulario para el seguimiento del tiempo tenemos el gráfico mostrado a continuación:

Tabla I: Seguimiento de Tiempos

Fecha	Inicio	Fin	Interrupción	Tiempo Delta	Actividad	Comentarios	C	U
2-Jan	18:00:00	20:00:00	0:40:00	1:20:00	Investigación	PSP, fuentes internet, cena	x	
4-Jan	10:00:00	12:00:00	0:15:00	1:45:00	Investigación	PSP, fuentes internet, refrigerio	x	
6-Jan	18:30:00	20:00:00	0:30:00	1:00:00	Lectura	Capítulo 1 texto guía, cena		
8-Jan	18:00:00	20:30:00	0:30:00	2:00:00	Lectura	Capítulo 1 texto guía, cena	x	1
13-Jan	20:00:00	21:30:00	0:10:00	1:20:00	Lectura	Capítulo 2 texto guía, teléfono		
15-Jan	19:00:00	21:30:00	0:30:00	2:00:00	Lectura	Capítulo 2 texto guía, cena	x	1
18-Jan	9:00:00	12:00:00	0:40:00	2:20:00	Desarrollo de ejemplos	Análisis código etiquetas, desayuno		
19-Jan	9:00:00	11:00:00	0:30:00	1:30:00	Desarrollo de ejemplos	Análisis código etiquetas, desayuno	x	1
21-Jan	20:00:00	22:00:00	0:40:00	1:20:00	Lectura	Capítulo 3 texto guía, cena		
22-Jan	21:00:00	22:00:00	0:15:00	0:45:00	Lectura	Capítulo 3 texto guía, teléfono	x	1
24-Jan	14:00:00	15:30:00	0:10:00	1:20:00	Investigación	Servicios web, fuentes internet, soporte a usuario	x	
29-Jan	16:00:00	19:00:00	0:50:00	2:10:00	Prog.	Prueba impresión código de barras, soporte a usuario	x	1
31-Jan	17:00:00	18:00:00	0:15:00	0:45:00	Investigación	JSP-GlassFish, revisión de correo personal		

Del ejemplo podemos decir que, del periodo comprendido entre el 2 de enero al 31 de enero se han realizado 5 tareas consideradas importantes dentro de la etapa de reconocimiento del problema y posibles soluciones.

Además se puede decir que en el registro del 31 de enero a partir de las 17:00 hasta las 18:00 se tomó 15 minutos en la revisión de correo electrónico personal haciendo que el tiempo total consumido en esta actividad haya sido solo de 45 minutos. Con esta información se puede obtener una base sobre la cual se establecen mejoras acordes al compromiso personal con la actividad asignada –Ver anexo-.

3.5 Análisis basado en LOC

Este análisis tiene como finalidad cuantificar el trabajo realizado en un tiempo determinado y conocer el esfuerzo realizado por una persona en términos de líneas de trabajo –LOC-.

Basados en este análisis es útil mencionar que se puede estimar tendencias que ayudarán a corregir eventos que causaron un impacto el desarrollo del sistema y a su vez evaluar el rendimiento y eficiencia de cada uno de los módulos del sistema.

En la figura mostrada a continuación se muestra el número de líneas de códigos tomados de la etapa preliminar del desarrollo del sistema administrador de códigos de barra.

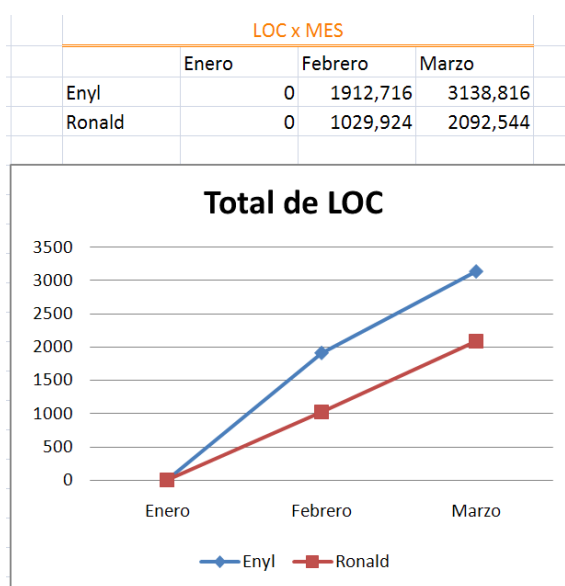


Figura 3.5.1: LOC (Líneas de Código) por Mes

CAPÍTULO 4

4. IMPLEMENTACIÓN DEL SISTEMA ADMINISTRADOR DE CODIGOS DE BARRA (SACB)

La idea de implementar el Sistema Administrador de Códigos de Barra nace por la necesidad de mejorar el proceso de búsqueda de información de los sistemas que utiliza la empresa Molemotor S.A., la cual en su mayoría consistía en realizar ingresos manuales de características de productos. Estas características en muchas ocasiones generaban búsquedas con resultados diferentes a los esperados. Adicionalmente ingresaban a un proceso de validación de la información presentada.

De esta manera, se implementó un servicio web del cual todas las aplicaciones que necesiten identificar algún activo pueden realizarlo con solo importar sus datos. Con estos datos se generan códigos de barra representativos, los cuales serán asignados a su correspondiente activo. Con esta solución cada aplicación al momento de realizar una consulta por el activo realizará una llamada al servicio web preguntando por el equivalente al código de barra obtenido. El servicio en este momento

devolverá el código identificador de esta activo relacionado a la aplicación que lo necesita.

4.1 Análisis del requerimiento

Como parte del análisis se realizó un levantamiento del proceso actual de búsqueda de información en los sistemas tomando como ejemplo sistemas de control de inventario y el de registro de entradas de personal a las instalaciones.

Estos sistemas mantienen una base de datos centralizada. Cuando se realizan búsquedas de un producto o de un empleado, la persona que hace uso del sistema ingresa manualmente características que permitan realizar una búsqueda acertada. Estas búsquedas regularmente se realizan con el propósito de realizar alguna modificación en sus bases de datos.

El tiempo tomado para realizar cada una de estas modificaciones es considerable dado que este tipo de consultas se realizan periódicamente. Adicionalmente, en algunas de estas consultas que involucran cambios se presentan con errores al ingreso de información debido a que no se contaba con la información necesaria dentro de la base de datos lo que daba como resultado tener en registros duplicados y por ende hacer otras consultas para

corregir lo ingresado anteriormente.

El diagrama presentado a continuación muestra los procesos de búsqueda para el sistema de control de inventario y para el registro de entrada del personal:

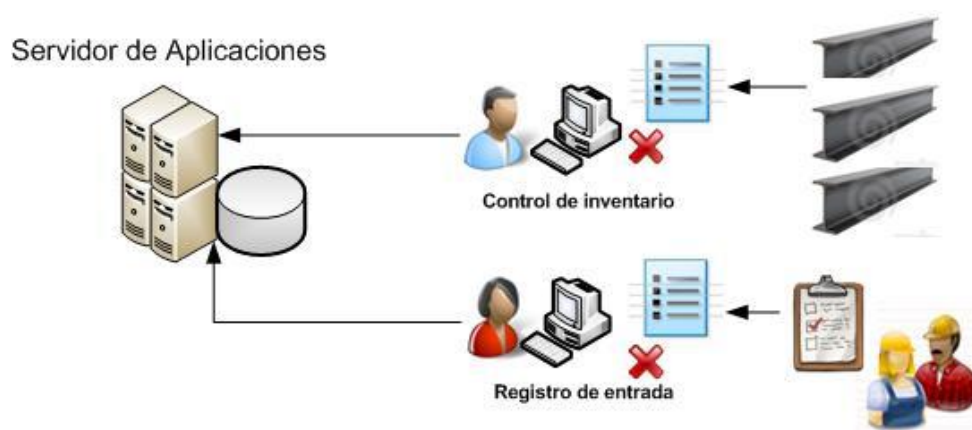


Figura 4.1.1: Proceso actual de Molemotor S.A.

4.2 Diseño de la arquitectura

Nuestra solución se basa en un reconocimiento físico de la entidad (materia prima, productos terminados, personas, etc.) a través de códigos de barra que faciliten su identificación y garanticen su control.

De esta manera minimizamos el tiempo de corrección de información y validación periódica de la misma.

En el gráfico a continuación se muestra el proceso mejorado de

búsqueda tanto para el control de inventario como para registro de entrada de personal los cuales integran la validación de códigos de barra:

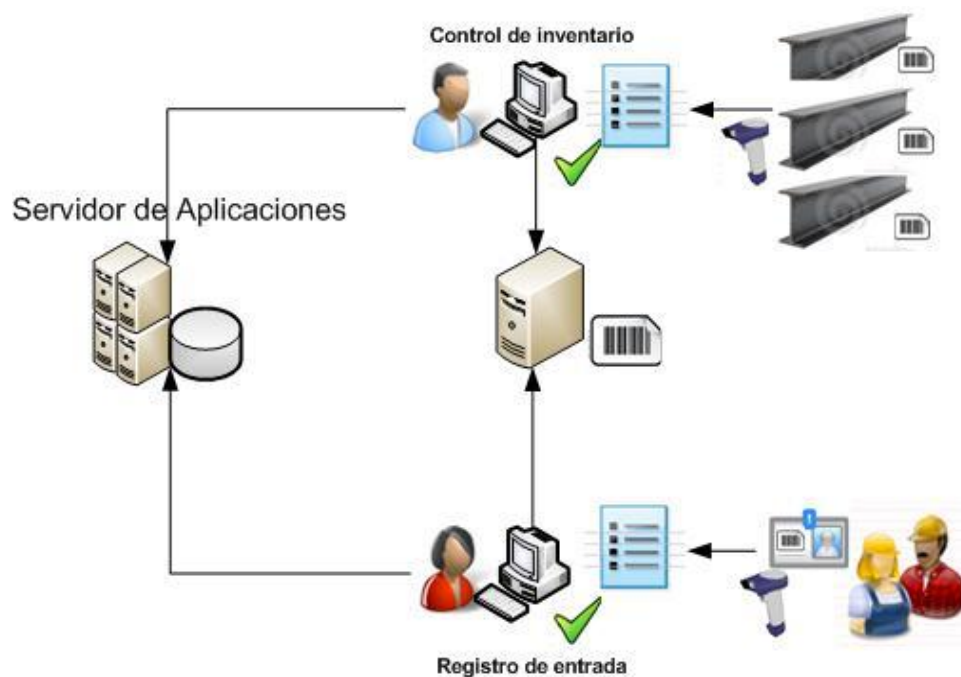


Figura 4.2.1: Proceso mejorado con códigos de barra

La arquitectura del sistema es cliente-servidor. El servicio de códigos de barra es llamado por cada una de las aplicaciones al momento de realizar una consulta y este le devuelve la información asociada al código ingresado.

Esquema general

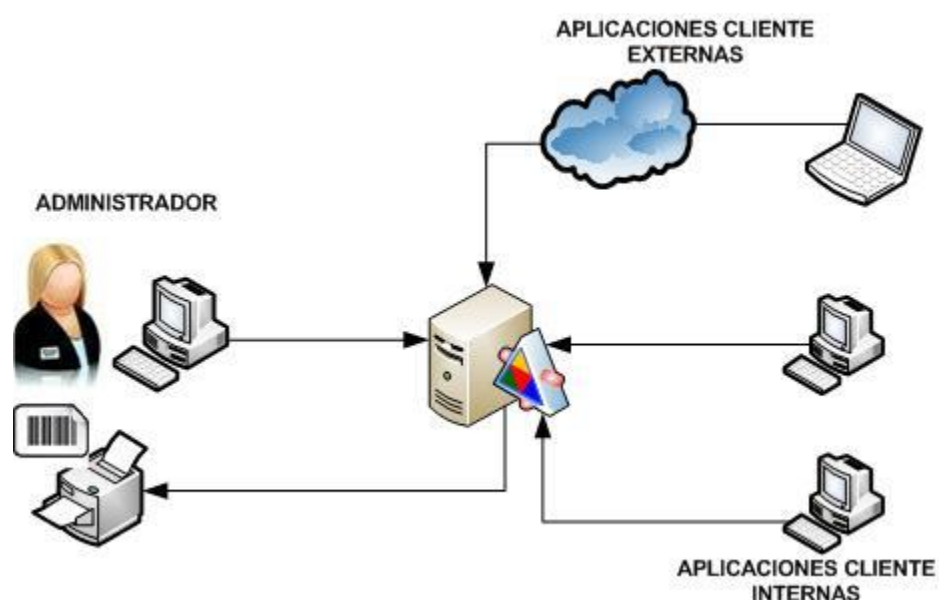


Figura 4.2.2: Esquema general de la solución

Actores del sistema

En el diagrama anterior se muestran los siguientes actores:

- Administrador del sistema: Persona encargada del mantenimiento del servicio generador de códigos de barra. Esta persona será la encargada de registrar las aplicaciones que utilizarán el servicio de búsqueda a través de códigos de barra y a su vez de realizar la impresión de los códigos para su posterior despliegue en etiquetas.

- Aplicaciones: Estas aplicaciones utilizan el servicio de códigos de barra para la validación de los códigos leídos con el fin de obtener la información del identificador correspondiente que le permita realizar búsquedas y con ellos dar el mantenimiento necesario.

Casos de uso

Los casos de uso identificados para el desarrollo del sistema son los siguientes:

- Administrar usuarios.- El administrador realiza las acciones de eliminación, modificación y creación de usuarios que pueden acceder al sistema.
- Administrar aplicaciones.- Incluye la eliminación, modificación y creación de aplicaciones que van a hacer uso del servicio de consulta a través de códigos de barra. Estas acciones son realizadas por el administrador.
- Imprimir códigos.- Esta acción es realizada por el administrador cuando al tener ingresado los datos de la aplicación procede a imprimir los códigos generados.
- Administrar códigos.- Permite crear, eliminar, modificar códigos de barra por lote o de manera individual para una aplicación. Esta acción la realiza el administrador.

- Consultar códigos.- Esta acción la realizan cada una de las aplicaciones registradas previamente por el administrador. Esta acción consiste en consultar el identificador asociado al código de barras al SACB.

A continuación se muestra el diagrama de casos de uso con sus respectivos actores:

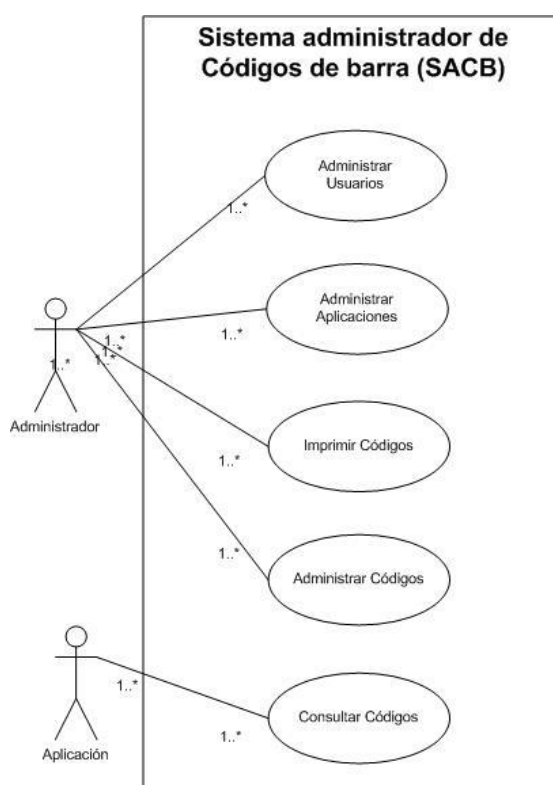


Figura 4.2.3: Caso de Uso para el SACB

CAPÍTULO 5

5. IMPLEMENTACION DEL CASO: SISTEMA DE REPORTES DE DIARIOS DE TRABAJO

El Sistema de Reportes de Diarios de Trabajo nace como respuesta a la necesidad de evaluar la eficiencia que tendrían los sistemas de la empresa Molemotor S.A. al implementar una arquitectura de búsqueda de información basada en códigos de barra. Por este motivo el enfoque de desarrollo de fue asumido tanto por la gerencia de la empresa como por nosotros como un sistema piloto para la evaluación de resultados.

Este sistema es un resumen de un conjunto de subsistemas entre los que se incluyen un módulo de reportes de marcaciones y un módulo para el registro de marcaciones.

Para el desarrollo de este sistema se realizó una evaluación de reingeniería del sistema actual de Diarios de Trabajo con el fin de constatar cuan integrable es este sistema al nuevo esquema de búsqueda y si la lógica implementada puede ser reutilizable.

5.1 Análisis del proceso

Dado que nuestro SACB es parte de una reestructuración de los procesos del negocio de la empresa Molemotor S. A. era necesario que cada proceso o sistema a integrar se remita a una revisión y evaluación del proceso mantenido para sí en conjunto con la gerencia obtener mejoras significativas con respecto a la eficiencia.

De lo mencionado anteriormente se muestra el proceso actual que involucra realizar ingreso de diarios de trabajo. Este proceso se ilustrará con el siguiente gráfico:

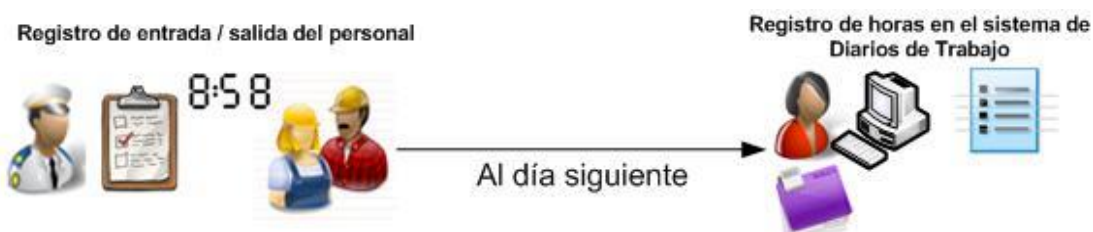


Figura 5.1.1: Proceso de registro de marcaciones

Descripción del proceso actual

1. Cada día el supervisor de cada obra anota en una bitácora el registro de la entrada y salida de cada una de las personas que están asignadas a una actividad en específico. En este tipo de registro también se incluye el tiempo de almuerzo.

2. Al finalizar el día el supervisor termina de llenar esta bitácora asignando a cada empleado la actividad que realizó y a que obra estaba asociada dicha actividad.
3. A las 12:00 pm del día siguiente esta bitácora es llevada al auditor para que realice el ingreso de la misma en el sistema de diarios de trabajo para que se contabilice el trabajo de cada empleado en horas. De esta manera se paga por los servicios de cada empleado de manera quincenal o mensual.
4. En varias ocasiones este pago es revisado debido a que los empleados presentan inconformidad. Por lo que el auditor realiza consultas de registros anteriores para la validación de las horas trabajadas.

5.2 Identificación del problema

De acuerdo a la revisión del proceso realizado el problema general del sistema es la eficiencia tanto en pagos a empleados como en reporte de horas de trabajo. Tal problema se muestra mejor con los siguientes enunciados:

- ✓ El reporte de horas de ingreso y salidas no es generado de manera automatizada lo cual no permite una exactitud en el tiempo empleado por cada trabajador.

- ✓ El reporte al ser un documento físico es ingresado al día siguiente por el auditor al sistema de diarios de trabajo. En este ingreso también incluye la revisión del pago por hora a cada empleado de acuerdo a su cargo. Esta revisión es realizada con el sistema de recursos humanos.
- ✓ El número de inconformidades presentadas por los empleados al momento de realizar los pagos quincenales. Estas atenciones restan tiempo en las tareas diarias de la persona de auditoria.

Adicionalmente existe una falta de control en los horarios que posee cada empleado, debido a que el módulo de recursos humanos no implementa esta funcionalidad.

5.3 Reingeniería del sistema

En concepto, la reingeniería de sistemas tiene por finalidad reestructurar o transformar viejos sistemas en aplicaciones más fáciles de mantener, con entornos más agradables e integradas en nuevas plataformas ya sean de hardware o software.

La reingeniería casi siempre implica cambiar la forma de un programa y mejorar su documentación. En este caso, la funcionalidad del programa no es cambiada; sólo su forma es

modificada. En otros casos, la reingeniería va más allá de la forma e incluye rediseñar para cambiar la funcionalidad del programa para buscar mejores requerimientos de usuario. En el sistema de diarios de trabajo la funcionalidad tendría que cambiar dado que debe ajustarse a un proceso de búsqueda utilizando el servicio de códigos de barra.

La siguiente lista muestra los puntos que serán evaluados en el sistema actual de diarios de trabajo:

- ✓ Frecuentes fallas de producción (fiabilidad cuestionable)
- ✓ Problemas de rendimiento
- ✓ Problemas de integración del sistema
- ✓ Código de baja calidad
- ✓ Dificultad al cambio
- ✓ Dificultad para probar
- ✓ Incremento de problemas del sistema



A cada uno de estos puntos se les asignó un peso. Cuyo valor representa el grado de importancia encontrado por la empresa dentro de su proceso de reestructuración. La tabla de pesos es mostrada a continuación:

Tabla II: Pesos de puntos a evaluar

<u>Punto a evaluar</u>	<u>Descripción</u>	<u>Peso</u>
Rendimiento	Mide cuan eficiente es el uso del sistema.	15
Nivel de integración del sistema	Muestra el nivel de complejidad del sistema al integrarse con otros programas.	25
Código de alta calidad	Mide el nivel de entendimiento de la lógica utilizada al leer el código fuente.	15
Facilidad al cambio de lógica de negocio	Muestra cuan sencillo es para el sistema adaptarse a cambios en procesos de negocio.	25
Portabilidad	Mide el nivel de aceptación por parte de los usuarios	20

Total : 100

Tabla III: Métricas para concluir reingeniería

Mayor a 70	Se reutiliza el código.	
Menor o igual 70	Se crea un nuevo sistema.	

5.4 Evaluación de los resultados obtenidos de la reingeniería

Luego de la revisión realizada del sistema actual de diarios de trabajo en base a los puntos mostrados para la evaluación de reingeniería de sistema se obtuvo lo siguiente:

Tabla IV: Pesos obtenidos

<u>Punto a evaluar</u>	<u>Peso</u>	<u>Calificación</u>
Rendimiento	15	14 / 15
Nivel de integración del sistema	25	10 / 25
Código de alta calidad	15	8 / 15
Facilidad al cambio de lógica de negocio	25	10 / 25
Portabilidad	20	10 / 20

TOTAL : 52 / 100

Dados los resultados concluimos que la mejor opción de reingeniería para el sistema de diarios de trabajo actual es que se realice el desarrollo de manera completa.

Esquema del proceso propuesto

Luego del análisis del proceso y la identificación del problema se diseñó un proceso más eficiente, el cual se muestra a continuación:

1. El supervisor inicia programa cliente instalado en un computador portátil y este al iniciar realiza una llamada al servidor central para solicitar la hora.

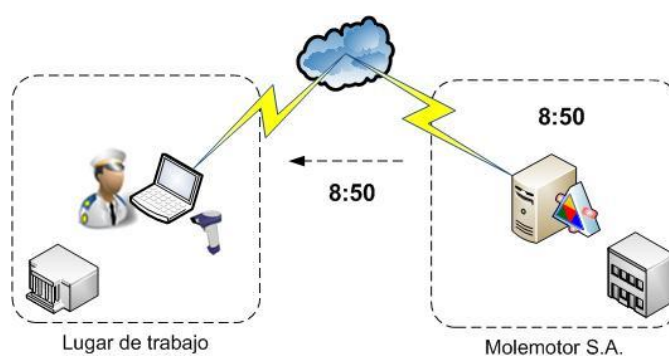


Figura 5.4.1: Esquema propuesto

2. Luego de haber iniciado registra la entrada o salida del personal.



Figura 5.4.2: Registro de marcación

3. Al finalizar cada registro lo envía al servidor central para que se almacene en el registro marcaciones de diarios de trabajo automáticamente.

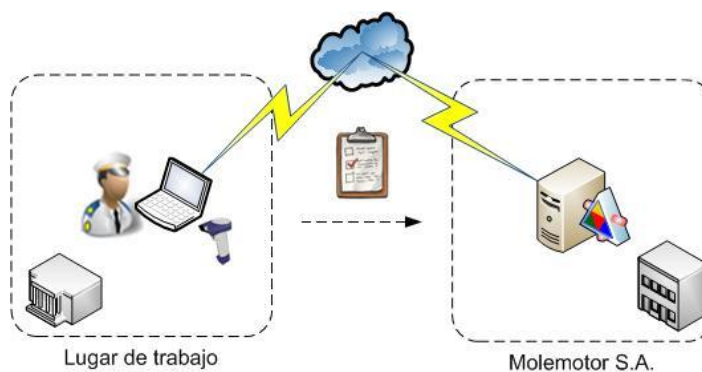


Figura 5.4.3: Esquema de marcaciones propuesto

4. El auditor lee los registros ingresados, verifica alguna anomalía y almacena con estos horarios el diario de trabajo para esta actividad.



Figura 5.4.4: Registro de actividades

5.5 Metodología a utilizar en la implementación del sistema

En la implementación se tiene como objetivo satisfacer los requisitos de la manera en que se especifican en el diseño de la solución.

Previa a esta fase, se especificó los requisitos, realizó el diseño detallado, el estándar de programación y la arquitectura de software. Cada uno de los integrantes ejecutamos el proceso de implementación de acuerdo a los siguientes pasos:

- ✓ Verificar el estándar de programación, y diseño detallado de la unidad que se va a implementar.
- ✓ Estimar el tamaño y tiempo con base a otras implementaciones de unidades.
- ✓ Registrar los defectos especificando los errores como “errores de codificación”
- ✓ Verificar si lo implementado cumple con los estándares definidos.

Ambiente y herramientas a utilizar

En la implementación del sistema se tomó como sistema operativo base Windows Server 2003 Enterprise Edition de 32 bits y para almacenamiento de datos se utilizó SQL Server 2008.

Como herramientas de desarrollo se hizo uso de una plataforma de software libre, en las que se incluye:

- Servidor de aplicaciones Glassfish v 2.1
- Lenguaje de programación JAVA
- IDE para desarrollo NetBeans 6.5

Revisión, Inspección y Reutilización

Estos tres procesos en la fase de implementación son muy importantes ya que nos ayudan a controlar los errores y estándares para desarrollar un producto de calidad.

En nuestro proyecto nos enfocamos a las revisiones y reutilización del código fuente.

Se realizaron revisiones para llevar el control del uso de estándares de programación y de diseño, se siguió el siguiente proceso:

1. Definir unidades a revisar.
2. Realizar revisiones del estándar de programación y de diseño.
3. Registrar las observaciones encontradas.

En nuestro proyecto se realizó reutilización, esto nos ayudó a evitar la redundancia de código, y el tiempo de programación. La ventaja

que se obtuvo con la reutilización de código fue producir programas más pequeños y realizar correcciones más rápidas.

Planificación de la Implementación.

La planificación de la implementación nos llevó a un mejor control de tiempos de desarrollo para cada uno de los componentes. Se planificó y diseñó las aplicaciones de manera que permitan la reutilización de código.

Es muy recomendable considerar la reutilización de código existente confiable antes de escribir el propio. Se planificó de acuerdo a las necesidades del usuario y la manera como se integran los sistemas paulatinamente, es por esta razón que se implementó cada módulo según las siguientes prioridades:

1. General: Este incluye acceso a datos, lógica de comunicación y modelo de clases.
2. AdminCodBarras: Incluye la lógica de programación para administración de aplicaciones y códigos de barra.
3. AdminCodBarras Service: Publica los servicios que serán utilizados por las aplicaciones.

4. AdminCodBarras Client: Hace uso de los servicios publicados e integra la interfaz que interactúa con el usuario administrador.

Estándares de Programación.

El uso de estándares mejora la disciplina, la legibilidad y la portabilidad de un programa, es por ello que se trabajó en añadir al plan de desarrollo el estándar de programación que está orientado a tener un esquema de formatos y de sintaxis para el desarrollo de los componentes del proyecto.

La utilización de un estándar de programación nos ayuda a mantener la continuidad del trabajo, facilita nuestra inserción en equipos de trabajo y reduce el tiempo de aprendizaje.

Manejo y presentación de Errores.

Para llevar un control de estos errores, seguimos los siguientes pasos:

1. Realizar revisiones del código
2. Registrar los errores

3. Ingresar parámetros que puedan generar error

Se clasificó tipos de errores de la siguiente manera:

Validación: Alguna validación que el usuario ha hecho notar con el uso del sistema.

Actualización de programas: Cuando los programas no han sido actualizado correctamente, es decir hay inconsistencias en los programas de back-end con los de front-end.

Programación: Los más frecuentes y ocurridos por errores del programador.

Datos: Hay inconsistencia de datos en diferentes opciones del sistema.

Observaciones: Consideraciones adicionales del usuario.

Integración: Errores presentados en la integración de los diversos módulos.

Configuración de los equipos: Errores debidos a que los equipos de los usuarios no tiene la configuración regional correcta o la instalación de un software necesario, para el correcto funcionamiento

del sistema. Ejemplo: librerías adicionales, configuración regional, etc.

La presentación de errores en nuestro sistema se lo realiza mediante mensajes clasificados en las siguientes categorías:

Informativas.- Para informar al usuario del error o campos que debe considerar para realizar una determinada acción.

Advertencia.- También considerados para campos necesarios, y para errores controlados en la base de datos, de inconsistencias en el formato de los datos de ingreso.

Erróneas.- También considerados errores controlados en la base de datos, errores fatales del sistema.

Calidad de la Implementación.

Esta sección estudia las métricas para lograr una implementación de calidad.

Las siguientes preguntas nos pueden ayudar a la revisión de alguna manera el código, específicamente las clases, y fueron aplicadas por nuestro equipo de trabajo:

- ¿Es apropiado su nombre?

- ¿Su título describe su propósito?
- ¿Se aplicaron los estándares de documentación?

La métrica usada por el equipo de trabajo es el “número de líneas”.

Número de líneas: Las “Líneas de Código” constituyen una medida útil se establece una medida estándar para contar. Por ejemplo: Cómo contar las declaraciones que ocupan varias líneas.

- ✓ Cómo contar los comentarios.
- ✓ Cómo contar las líneas que consisten en while, for, do, etc.

Esta métrica mide el tamaño de las unidades. Generalmente, entre más grande sea el tamaño del código de un componente del programa, más complejo y susceptible a errores será el componente.

El tamaño es una medida empleada fundamentalmente por tres razones: es fácil de obtener una vez que el programa ha sido completado, es uno de los factores más importantes en los métodos de desarrollo, y la productividad se expresa tradicionalmente con el tamaño del código.

La medida de tamaño más usada es la cantidad de líneas de código que se representa y se mide en LOC (Lines Of Code, líneas de código). Para programas grandes es más adecuado el uso de KLOC

(miles de líneas de código). Para muchos autores, las líneas de código medidas no deben incluir comentarios o líneas en blanco, dado que su presencia o ausencia no afectará al funcionamiento del programa. Además, incluir comentarios o líneas en blanco no supone el mismo nivel de dificultad que desarrollar una línea de código.

Se implementó la métrica “número de líneas” considerando las líneas de comentarios y líneas en blanco, el resultado obtenido se lo puede observar en los resultados de la utilización de la metodología PSP en el que se muestra el rendimiento obtenido en KLOC.

5.6 Mejoras obtenidas en base al sistema anterior

Las mejoras fueron evaluadas en base al análisis del tiempo que involucraba ingresar un reporte de diarios de trabajo en el sistema actual. Este tiempo involucra lo siguiente:

1. Tiempo de registro de entrada y salida del personal por parte del supervisor.
2. Tiempo de recepción de esta bitácora a la oficina de la auditora en Molemotor S.A.
3. Tiempo de ingreso de esta bitácora al sistema de diarios de trabajo por parte del auditor.

El tiempo total es el resultado de la suma de los tiempos mencionados anteriormente. A su vez, nos brinda una métrica para evaluar la mejora de realizar una reingeniería de procesos al hacer uso de nuestro sistema administrador de códigos de barra (SACB).

En los gráficos mostrados a continuación se resume la evaluación realizada antes y después de la implementación del SACB y de la reingeniería de procesos aplicada.

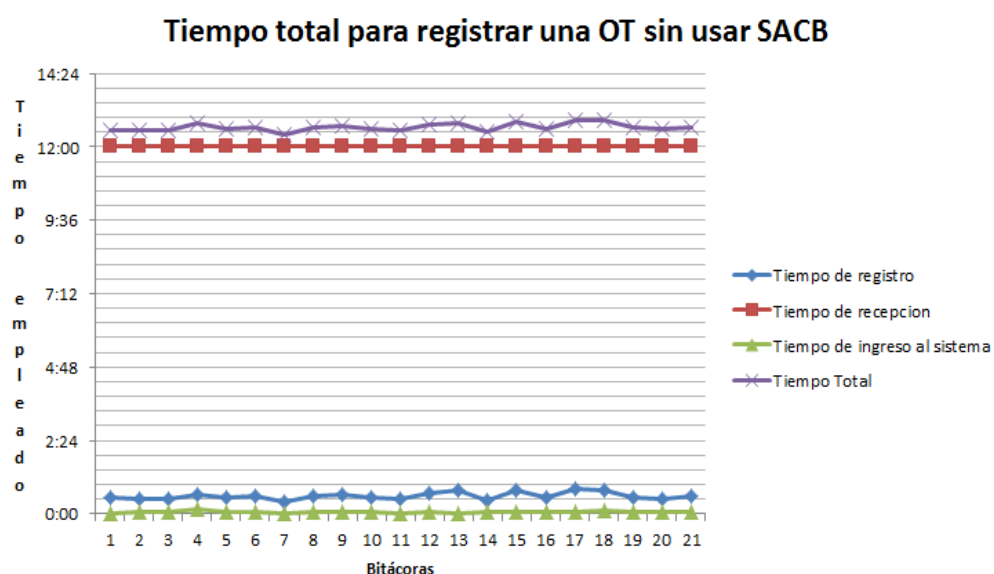


Figura 5.6.1: Tiempo total para una OT sin usar SACB

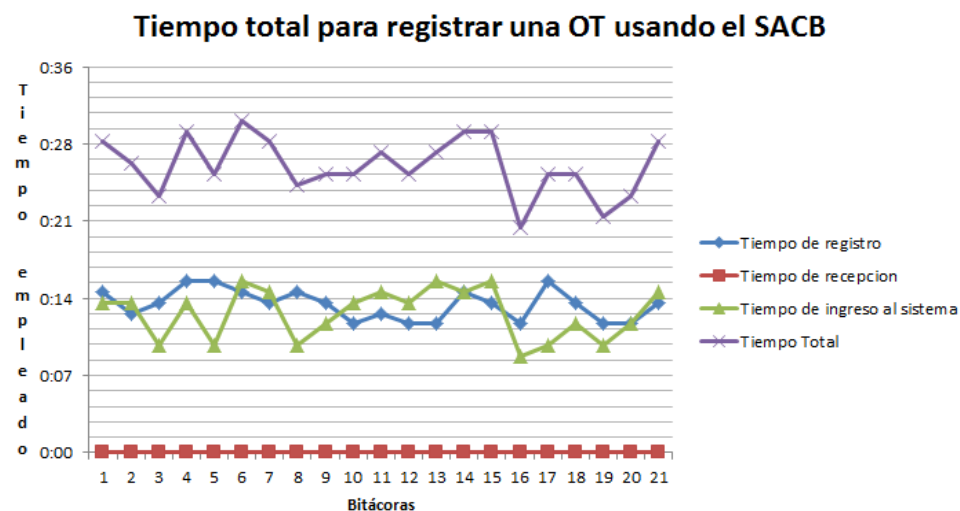


Figura 5.6.2: Tiempo total para una OT usando SACB

De manera gráfica se puede apreciar que:

1. El tiempo total ha disminuido significativamente.
2. El tiempo de recepción disminuye completamente debido al envío en línea de las marcaciones.
3. Los tiempos de registro e ingreso al sistema son variables debido a que se encuentran en una etapa de adaptación al nuevo sistema y proceso consecuentemente.

Finalmente al hacer una comparativa entre los tiempos totales obtenidos tenemos lo siguiente:

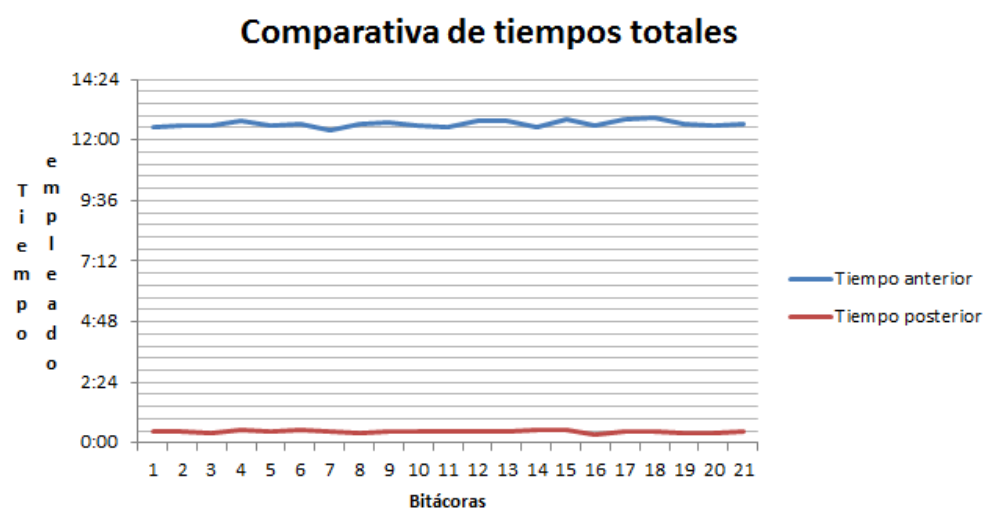


Figura 5.6.3: Comparación de tiempos totales



Figura 5.6.4: Tiempo total en minutos

Previo a la conclusión es necesario conocer los resultados del análisis de evaluación personal utilizando la metodología PSP. A continuación se muestra un resumen del desempeño de los integrantes y el tiempo utilizado.

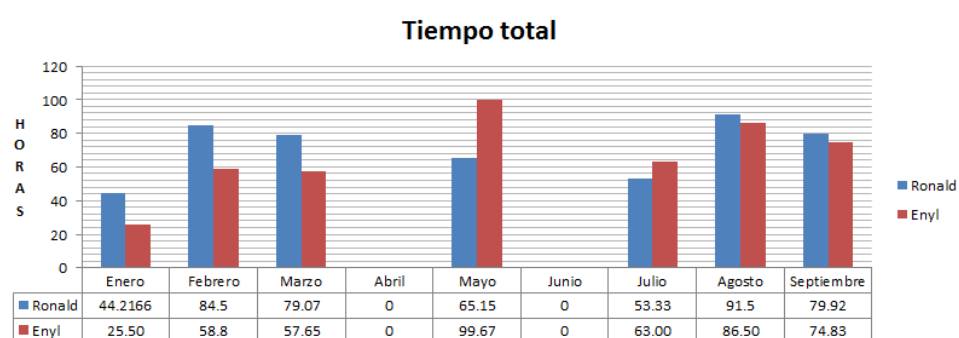


Figura 5.6.5: Tiempo total

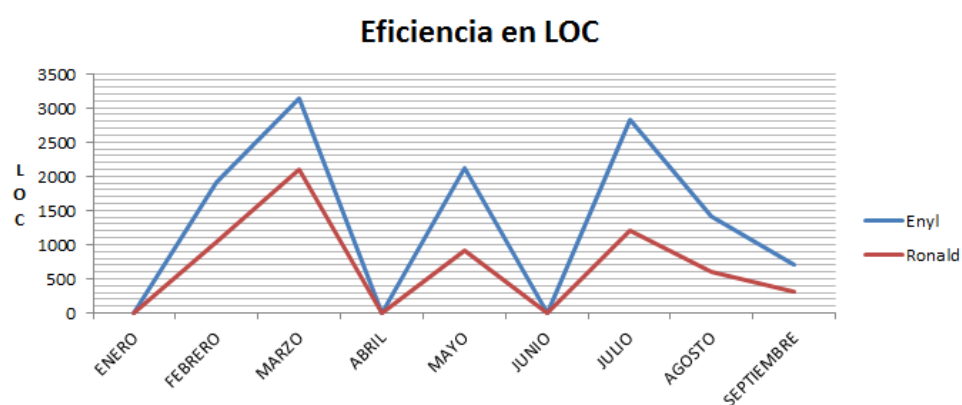


Figura 5.6.6: Eficiencia en LOC

Además se muestra el uso que se ha dado al tiempo dando a conocer las interrupciones y el tiempo útil para el desarrollo de cada actividad. Los resultados son los siguientes:

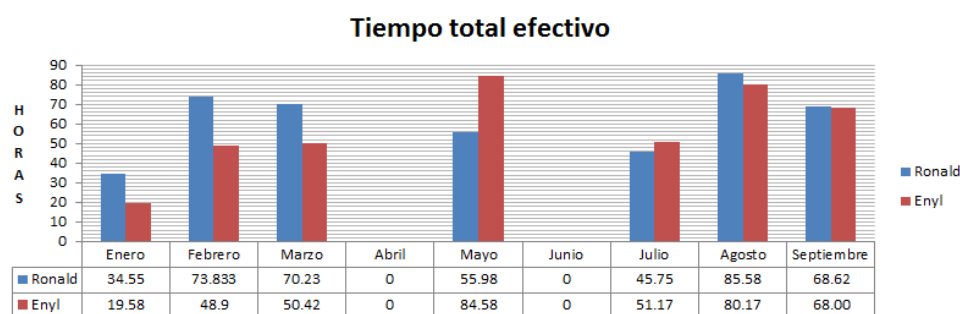


Figura 5.6.7: Tiempo total efectivo

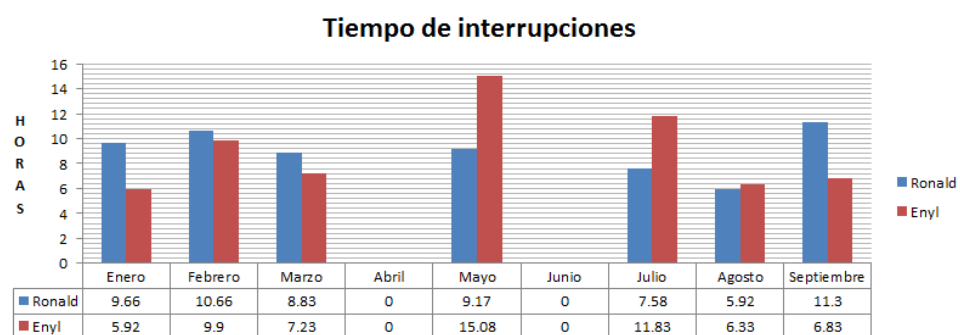


Figura 5.6.8: Tiempo de interrupciones

Conclusiones y recomendaciones

De acuerdo a los objetivos planteados al inicio del proyecto y en base a los resultados obtenidos podemos concluir que:

1. La metodología PSP es parte de un proceso cíclico que ayuda al mejoramiento personal de las actividades. Para el caso en particular ayudó al proceso de desarrollo, implementación y pruebas.
2. Se minimizó el tiempo de interrupciones teniendo un tiempo dedicado para descanso en el que se incluía actividades adicionales.
3. Los sistemas pilotos de marcaciones y diarios de trabajo mejoraron significativamente usando el SACB por lo que se evidencia la eficiencia esperada del servicio de búsquedas usando códigos de barra.
4. El proceso de reingeniería del sistema actual de Diarios de Trabajo da como resultado que es mejor realizar el desarrollo de un nuevo sistema, debido a que actualmente no es factible integrarlo a una nueva infraestructura, tampoco permite realizar integración con otras aplicaciones y finalmente el código es poco legible para comprender el proceso.

Como recomendaciones podemos dar las siguientes:

1. Poner en práctica la metodología PSP en actividades de las cuales necesitamos obtener un mejor desempeño.
2. Integrar a la metodología PSP con TSP con el fin de conocer la eficiencia del trabajo en equipo.
3. Trabajar en conjunto a una norma que regule los procesos de elaboración de proyectos como por ejemplo ISO 27001 para tener un lineamiento que permita conocer de manera global los procesos y como mitigar riesgos en nuevas implementaciones.
4. A fin de tener un respaldo formal es necesario involucrar al personal gerencial para que nos brinde su apoyo en la elaboración de un proyecto.

Referencias bibliográficas

1. BRAUDE, ERIC. Ingeniería de Software, Una perspectiva Orientada a Objetos, Boston University, Alfa Omega, Grupo Editor 2003.
2. HUMPHREY, WATTS S. Introduction to the Personal Software Process, septiembre de 1994
3. SOMMERVILLE, IAN. Ingeniería de Software. Séptima Edición, Pearson Education, Madrid 2005.
4. HUMPHREY, WATTS S. Pag. 9 Introduction to the Personal Software Process, septiembre de 1994
5. HUMPHREY, WATTS S. Pag. 19 Introduction to the Personal Software Process, septiembre de 1994
6. HUMPHREY, WATTS S. Pag. 57 Introduction to the Personal Software Process, septiembre de 1994
7. ING. QUEVEDO JUAN, REINGENIERIA PARA SISTEMAS DE EVALUACIÓN TEÓRICA,
<http://www.revistaciencias.com/publicaciones/EEAAklAEIFxpOUEqJu.Php>, 2 de enero de 2008
8. WIKIPEDIA, BUSINESS PROCESS MANAGEMENT, WIKIPEDIA,
http://en.wikipedia.org/wiki/Business_process_management, 22 de noviembre de 2010

9. WIKIPEDIA, PERSONAL SOFTWARE PROCESS, WIKIPEDIA,
http://es.wikipedia.org/wiki/Personal_Software_Process, 1 de
noviembre de 2010.
10. FLORES, NICOLÁS, RECONSTRUCCIÓN DE LA ARQUITECTURA:
UNA ACTIVIDAD DE LA REINGENIERÍA DE SOFTWARE,
[http://www.monografias.com/trabajos17/reingenieria-](http://www.monografias.com/trabajos17/reingenieria-software/reingenieria-software.shtml#reing)
[software/reingenieria-software.shtml#reing](http://www.monografias.com/trabajos17/reingenieria-software/reingenieria-software.shtml#reing), 2004
11. WIKIPEDIA, REINGENIERÍA DE PROCESOS, WIKIPEDIA,
[http://es.wikipedia.org/wiki/Reingenier%C3%ADa_de_Procesos#Defini](http://es.wikipedia.org/wiki/Reingenier%C3%ADa_de_Procesos#Definici%C3%B3n_de_proceso)
[ci.C3.B3n_de_proceso](http://es.wikipedia.org/wiki/Reingenier%C3%ADa_de_Procesos#Definici%C3%B3n_de_proceso), 11 de septiembre de 2010.