

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**“Implementación de un Sistema integrado de Control  
de Nómina, Órdenes de Compra y Facturación  
utilizando PSP (Personal Software Process) y TSP  
(Team Software Process)”**

**TESIS DE GRADO**

Previa a la obtención del Título de:

**INGENIERO EN COMPUTACIÓN, ESPECIALIZACIÓN**

**SISTEMAS TECNOLÓGICOS**

Presentado por:

Gina Angelina Ruíz López

Arturo Javier Aguilera Mocha

Guayaquil - Ecuador

Año 2006

## **AGRADECIMIENTO**

Agradecemos:

Primeramente a Dios por la terminación de este trabajo, y a todos quienes hicieron posible llevar a cabo este proyecto.

## DEDICATORIA

A Dios,

A mi hermano,

Y con mucho cariño,

A mis padres,

Por su sacrificio y apoyo incondicional.

GINA ANGELINA RUIZ LOPEZ

A Dios,

A mis hermanos,

Pero con mucho cariño,

A mis padres,

Por brindarme todo su apoyo

ARTURO JAVIER AGUILERA MOCHA

# TRIBUNAL DE GRADUACIÓN

---

Ing. Holger Cevallos  
SUBDECANO DE LA FIEC

---

Ing. Mónica Villavicencio  
DIRECTOR DE TESIS

---

Ing. Verónica Macías  
MIEMBRO PRINCIPAL

---

Ing. Freddy Pincay  
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”

Gina Angelina Ruíz López

Arturo Javier Aguilera Mocha

# ÍNDICE GENERAL

|   |      |
|---|------|
| AGRADECIMIENTO .....  | II   |
| DEDICATORIA .....   | III  |
| TRIBUNAL DE GRADUACIÓN .....  | IV   |
| DECLARACIÓN EXPRESA .....   | V    |
| ÍNDICE GENERAL .....  | VI   |
| ÍNDICE DE FIGURAS .....   | VII  |
| ABREVIATURAS .....  | VIII |
| INTRODUCCIÓN...   | 15   |
| CAPÍTULO I .....  | 16   |
| 1. INTRODUCCIÓN AL TSP (TEAM SOFTWARE PROCESS) Y PSP<br>(PERSONAL SOFTWARE PROCESS) ..... | 16   |
| 1.1. Definición de PSP .....  | 16   |
| 1.2. Principios de PSP. ....  | 17   |
| 1.3. Modelo del PSP .....   | 19   |
| 1.4. Plantillas de PSP .....  | 22   |
| 1.5. Herramientas del PSP .....   | 26   |
| 1.6. Definición de TSP .....  | 26   |
| 1.7. Principios del TSP .....   | 27   |
| 1.8. Modelo del TSP .....   | 30   |
| 1.9. Plantillas del TSP .....   | 32   |

|   |    |
|---|----|
| CAPÍTULO 2.....                                       | 43 |
| 2. ADMINISTRACION DEL PROYECTO.....                   | 43 |
| 2.1. Visión del Proyecto.....                         | 43 |
| 2.1.1. Propósito.....                                 | 44 |
| 2.1.2. Alcance.....                                   | 44 |
| 2.2. Los Apostadores.....                             | 45 |
| 2.1.3. Definición.....                                | 45 |
| 2.1.4. Descripción.....                               | 45 |
| 2.3. Estructura Organizacional.....                   | 49 |
| 2.4. Etapas del Proyecto.....                         | 51 |
| 2.5. Metodología de Desarrollo.....                   | 53 |
| 2.6. Herramientas de desarrollo y soporte.....        | 56 |
| 2.7. Riesgos y planes de contingencias.....           | 57 |
| CAPITULO 3.....                                       | 63 |
| 3. FASE DE PLANEACIÓN.....                            | 63 |
| 3.1. Identificación de Entregables.....               | 63 |
| 3.2. Diagrama WBS.....                                | 66 |
| 3.3. Diagrama Gantt.....                              | 66 |
| 3.4. Iteraciones del Diagrama de Gantt en base a..... | 68 |
| las etapas del proyecto.....                          | 68 |

|  |    |
|--|----|
| CAPÍTULO 4.....                                      | 73 |
| 4. INGENIERIA DE REQUISITOS .....                    | 73 |
| 4.1. Preparación de Entrevista.....                  | 73 |
| 4.2. Requisitos C (cliente).....                     | 75 |
| 4.2.1. Propósito .....                               | 76 |
| 4.2.2. Alcance.....                                  | 76 |
| 4.2.3. Descripción Global .....                      | 77 |
| 4.2.4. Funciones del Proyecto .....                  | 77 |
| 4.2.5. Características del Usuario.....              | 80 |
| 4.2.6. Restricciones .....                           | 80 |
| 4.2.7. Distribución de Requisitos. ....              | 82 |
| 4.3. Requisitos D (Desarrollador).....               | 84 |
| 4.3.1. Requisitos Específicos.....                   | 85 |
| 4.3.1.1. Casos de Uso .....                          | 85 |
| 4.3.1.2. Limitaciones y Fronteras del Proyecto ..... | 86 |
| 4.3.2. Restricciones de Diseño .....                 | 86 |
| 4.3.3. Atributos del Sistema.....                    | 87 |
| CAPÍTULO 5.....                                      | 89 |
| 5. DISEÑO DETALLADO.....                             | 89 |
| 5.1. Especificación de Clases y Métodos.....         | 89 |
| 5.2. Diagramas de secuencia por caso de uso .....    | 89 |
| 5.3. Diagramas de flujo por métodos .....            | 90 |



|  |     |
|--|-----|
| 5.4. Modelo de Trabajo para los Casos de Uso .....                           | 90  |
| CAPÍTULO 6.....  | 93  |
| 6. IMPLEMETACIÓN .....   | 93  |
| 6.1. Revisión, Inspección y Reutilización .....                              | 93  |
| 6.2. Planificación de la Implementación.....                                 | 94  |
| 6.3. Estándares de Programación.....   | 95  |
| 6.4. Manejo y presentación de errores.....                                   | 96  |
| 6.5. Calidad en la Implementación.....                                       | 99  |
| CAPÍTULO 7.....  | 101 |
| 7. PRUEBAS DE UNIDADES .....   | 101 |
| 7.1. Diseño de Pruebas .....   | 102 |
| 7.2. Casos de Prueba .....   | 102 |
| 7.3. Resultado de las Pruebas .....  | 103 |
| 7.3.1. Registro de Pruebas .....   | 103 |
| 7.3.2. Informe de incidentes de Pruebas .....                                | 105 |
| 7.3.3. Resumen de Pruebas .....  | 105 |
| CAPÍTULO 8.....  | 106 |
| 8. METRICAS.....   | 106 |
| 8.1. Métricas: ¿Que son y para que sirven? .....                             | 106 |
| 8.2. Métricas utilizadas en el proceso de desarrollo .....                   | 107 |
| de software y su justificación .....   | 107 |
| 8.3. Valores de las Métricas obtenidas durante cada etapa del desarrollo del |     |

|   |     |
|---|-----|
| software .....  | 109 |
| CAPÍTULO 9.....   | 117 |
| 9. ROL DE LÍDER DEL EQUIPO Y ADMINISTRADOR DE CALIDAD ..... | 117 |
| 9.1. Papel que Desempeña .....                              | 117 |
| 9.2. Plantillas, Modelos, Metodologías y Estándares.....    | 118 |
| 9.3. Factores y Métricas de los Modelos.....                | 124 |
| 9.4. Experiencias y Responsabilidades .....                 | 125 |
| 9.5. Análisis General del Rol del Líder.....                | 126 |
| CAPÍTULO 10.....  | 129 |
| 10. COMENTARIOS FINALES Y CONCLUSIONES.....                 | 129 |
| 10.1. Glosario de Términos.....                             | 129 |
| 10.2. Análisis Comparativo: Antes y Después de .....        | 130 |
| TSP.....  | 130 |
| 10.3. Conclusiones y Recomendaciones.....                   | 132 |
| APENDICES .....   | 138 |
| BIBLIOGRAFÍA.....   | 139 |

# ÍNDICE DE FIGURAS

|  |     |
|--|-----|
| Figura 1.1. Evolución del PSP (3).....                 | 19  |
| Figura 1.2. Plantilla LOG T (5) .....                  | 22  |
| Figura 1.3 Plantilla LOGD (5).....                     | 24  |
| Figura 1.4 Plantilla STRAT (5) .....                   | 33  |
| Figura 1.5 Plantilla PEER (5) .....                    | 36  |
| Figura 1.6 Plantilla TASK (5).....                     | 39  |
| Figura 1.7 Plantilla CCR (5) .....                     | 42  |
| Figura 2.1 Estructura Organizacional.....              | 51  |
| Figura 2.2 Etapas del Proyecto.....                    | 51  |
| Figura 2.3 Divisiones de las etapas del Proyecto ..... | 52  |
| Figura 2.4 Desarrollo Incremental (8) .....            | 55  |
| Figura 7.1 Plantilla Plan de Pruebas.....              | 104 |
| Figura 9.21 Plantilla Inspección Documentación. ....   | 120 |
| Figura 9.2.2 Plantilla Usabilidad (28) .....           | 123 |

# ÍNDICE DE TABLAS

|  |     |
|--|-----|
| Tabla 2.1 Tipos de Riesgo .....  | 59  |
| Tabla 2.2 Clasificación de Riesgo por Tipo.....                        | 60  |
| Tabla 4.1 Distribución de Requisitos. Módulo de Órdenes de Compra..... | 83  |
| Tabla 4.2 Distribución de Requisitos. Módulo de Nómina.....            | 83  |
| Tabla 4.3 Distribución de Requisitos. Módulo de Facturación .....      | 84  |
| Tabla 8.1 Métricas por Rol.....  | 108 |
| Tabla 8.2 Errores por Página en Req C.....                             | 110 |
| Tabla 8.3 Defectos Detectados por Módulo .....                         | 111 |
| Tabla 8.5 Grado de Usabilidad por Módulo .....                         | 113 |
| Tabla 8.6 Contribución Total por Cada Rol.....                         | 114 |
| Tabla 8.7 Defectos en Pruebas de Aceptación .....                      | 116 |

# ABREVIATURAS

|       |  |
|-------|--|
| PSP:  | Personal Software Process.               |
| TSP:  | Team Software Process.                   |
| LOC:  | Líneas de Código                         |
| KLOC: | Miles de Líneas de Código.               |
| MNO:  | Módulo de Nómina.                        |
| MCIB: | Módulo de Control de Inventario y Bodega |
| MFAC: | Módulo de Facturación                    |
| MOT:  | Módulo de Ordenes de Trabajo.            |
| MOC:  | Módulo de Ordenes de Compra.             |
| MCP:  | Módulo de Costos de Producción.          |
| MPO:  | Módulo de Presupuesto por Obra.          |

# INTRODUCCIÓN

Actualmente los ingenieros de software deben sobrevivir en un mercado altamente competitivo; en donde entregar productos de calidad y a menor costo, de acuerdo con el cronograma y superando la expectativa de los clientes es cuestión de sobrevivencia. La realidad de los ingenieros de software es pasar por una presión constante de mejoras (restricciones de presupuesto, plazos, clientes más exigentes y competencia creciente), baja productividad (calificación de los profesionales, calidad de los productos), entre otros.

Según el análisis exploratorio que se llevo a cabo por el proyecto VLIR se determinó que de 77 empresas, pocas utilizan, utilizaron o piensan utilizar estándares de calidad para el desarrollo de software. Por lo tanto en esta tesis se investigó y se puso en práctica nuevas técnicas y metodologías para el desarrollo de software como lo son PSP y TSP.

En estas metodologías se proporcionan estándares, plantillas y métricas que sirven para evaluar el desempeño del grupo y la calidad del sistema, enfocadas en roles principales: Líder del equipo, Administrador de Configuración, Administrador de Planificación, Administrador de Desarrollo y

Administrador de Calidad.

Cabe mencionar que la tesis abarca desde la investigación y aprendizaje de todo el equipo de trabajo en dos metodologías de desarrollo de software nuevas, el levantamiento de requerimientos del software a desarrollar con los usuarios, desarrollo de un sistema de Nómina, Órdenes de Compra y Facturación, y la liberación del producto final al cliente.

# CAPÍTULO I

## 1. INTRODUCCIÓN AL TSP (TEAM SOFTWARE PROCESS) Y PSP (PERSONAL SOFTWARE PROCESS)

### 1.1. Definición de PSP

El PSP es un conjunto ordenado de procesos definidos que orientan a los ingenieros de software a medir, evaluar y monitorear la manera de hacer sus tareas (1). Los principales objetivos del PSP son (1):

- Mejorar las estimaciones de tamaño del software y tiempo de desarrollo.
- Mejorar el planeamiento de cronogramas.
- Proteger a los integrantes del equipo contra el exceso de compromisos;
- Crear un compromiso personal con la calidad;
- Procurar involucrar al desarrollador en la mejora continua de procesos;
- Mejorar la calidad a través de la reducción de la incidencia de errores;



Los ingenieros de software normalmente desarrollan productos a partir de sus propios métodos y técnicas, o a partir de ejemplos obtenidos de los más experimentados. Lo que el PSP ofrece es una forma de mejorar la calidad, predictibilidad y productividad del trabajo (1).

El modelo PSP está dividido en niveles implantados de manera incremental. Los niveles superiores adicionan características a los niveles ya implantados, lo que minimiza el impacto de los cambios en los hábitos del desarrollador. Este deberá tan sólo adaptar nuevas técnicas a las ya existentes y conocidas (1).

Lo más importante en el proceso de aprendizaje del PSP son los datos recogidos después de cada fase, pues en base a los resultados obtenidos en la fase actual se propone mejorar el desempeño personal para la siguiente fase (1).

## **1.2. Principios de PSP.**

El diseño de PSP se basa en los siguientes principios de planeación y de calidad (2):

- Cada ingeniero es esencialmente diferente; para ser más precisos, los ingenieros deben planear su trabajo y basar sus

planes en sus propios datos personales.

- Para mejorar constantemente su funcionamiento, los ingenieros deben utilizar personalmente procesos bien definidos y medidos.
- Para desarrollar productos de calidad, los ingenieros deben sentirse personalmente comprometidos con la calidad de sus productos.
- Cuesta menos encontrar y arreglar errores en la etapa inicial del proyecto que encontrarlos en las etapas subsecuentes.
- Es más eficiente prevenir defectos que encontrarlos y arreglarlos.
- La manera correcta de hacer las cosas es siempre la manera más rápida y más barata de hacer un trabajo.

Para hacer correctamente un trabajo de ingeniería de software, los ingenieros deben planear de la mejor manera su trabajo antes de comenzar y deben utilizar un proceso bien definido para realizar una buena planeación del trabajo (2).

Para que los ingenieros desarrolladores lleguen a entender su funcionamiento de manera personal, deben medir el tiempo que pasan en cada proceso, los defectos que inyectan y remueven de

cada proyecto y finalmente medir los diferentes tamaños de los productos que llegan a producir (2).

Para producir constantemente productos de calidad, los ingenieros deben planear, medir y rastrear constantemente la calidad del producto y deben centrarse en la calidad desde el inicio de un trabajo (2).

Finalmente, deben analizar los resultados de cada trabajo y utilizar estos resultados para mejorar sus procesos personales (2).

### 1.3. Modelo del PSP

El PSP se divide en etapas graduales de crecimiento llamadas PSP0, PSP1, PSP2 y PSP3 (3).

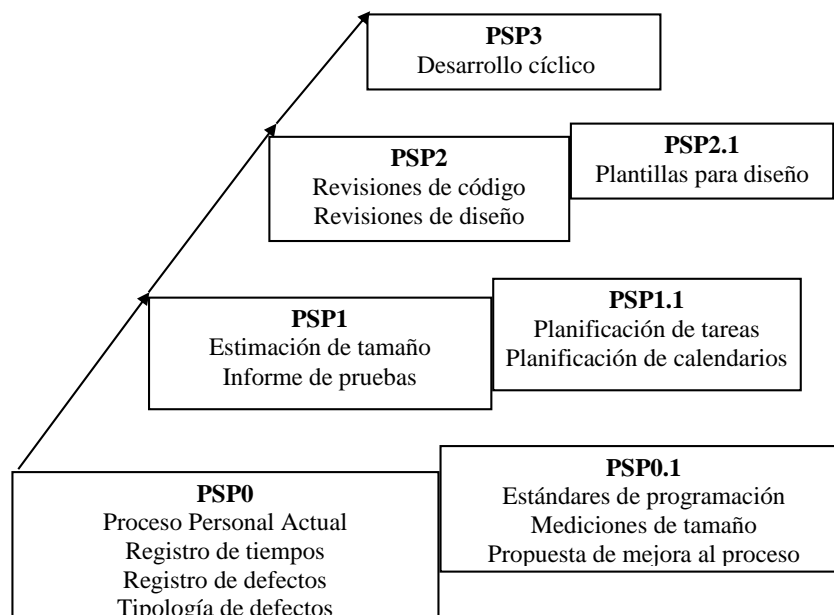


Figura 1.1. Evolución del PSP (3)

**PSP0:** Acepta las prácticas de desarrollo actuales del ingeniero pero requiere (3):

- 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 (3):

- 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 (3).

PSP1 agrega a PSP0 las siguientes aptitudes (3):

- Aptitud para estimar el tamaño.

- Marco de trabajo para informar los resultados de las pruebas.

**PSP1.1:** Agrega la habilidad para realizar (3):

- 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 (3).

- 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 (3).

**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 (3):

- Se aplica PSP a cada incremento para producir una alta base

de calidad para los incrementos sucesivos.

- Se usan pruebas de “regresión” para asegurar que las pruebas diseñadas para los incrementos anteriores todavía son buenas en los nuevos incrementos (3).

## 1.4. Plantillas de PSP

A continuación se describen los formatos utilizados en PSP0 y PSP 0.1 y que se basan generalmente en dos medidas importantes: el tiempo empleado y los defectos encontrados en cada fase (4).

El equipo de trabajo usó el formato que se presenta a continuación, él mismo que sirve para registrar el tiempo empleado por cada fase y contiene diversos campos. Conforme se avanza de nivel, se van agregando más campos (4).

| REGISTRO DE TIEMPO                           |        |       |                 |                                       |             |  |            |
|--|--------|-------|-----------------|---------------------------------------|-------------|--|------------|
| Nombre: Arturo Aguilera Mocha                |        |       |                 | Fecha: 5 de Mayo del 2005             |             |  |            |
| Equipo: TSOFT                                |        |       |                 | Instructor: Ing. Mónica Villavicencio |             |  |            |
| Tema: Desarrollo Módulo de Ordenes de Compra |        |       |                 | Incremento: Segundo                   |             |  |            |
| Fecha  | Inicio | Fin   | T. Interrupción | Δ Tiempo                              | Fase/ Tarea | Tema   | Comentario |
| 22/02/2005                                   | 14:25  | 14:45 | 0               | 20                                    | Desarrollo  | Creación de Vistas                           |            |
| 22/02/2005                                   | 15:20  | 15:45 | 0               | 25                                    | Desarrollo  | Se revisó el código en el spread de la Orden |            |
| 24/02/2005                                   | 16:00  | 16:50 | 0               | 50                                    | Desarrollo  | Se aumentó campos en la base de datos        |            |
| 24/02/2005                                   | 17:10  | 17:38 | 8               | 20                                    | Desarrollo  | Se validó la entrada de datos de la Orden    |            |
|  |        |       |                 |                                       |             |  |            |
|  |        |       |                 |                                       |             |  |            |
|  |        |       |                 |                                       |             |  |            |
|  |        |       |                 |                                       |             |  |            |

Figura 1.2. Plantilla LOG T (5)

El contenido de este formato marca el principio del proceso PSP y por lo tanto es parte del nivel inicial de PSP junto con el PSP 0.1. Los datos son utilizados para complementar el resumen del plan del proyecto (4).

Como información general lo único que se necesita es registrar el tiempo total que se emplea en el proyecto; este tiempo debe estar registrado en minutos y ser lo más preciso posible (4).

Los campos que forman parte de PSP 0 y de éste formato son (4):

- **Encabezado.-** Los datos que se deben agregar a este campo son el nombre del desarrollador, la fecha actual, el nombre del supervisor o instructor y el número que le corresponde al incremento que se está desarrollando.
- **Fecha.-** Corresponde a la fecha en la que se ingresa el dato en la plantilla.
- **Inicio.-** La hora cuando se comienza a trabajar en el proyecto.
- **Término.-** La hora cuando se deja de trabajar en el proyecto.
- **Tiempo de interrupción.-** Aquí se registran todas las interrupciones que se llevaron a cabo durante el proyecto, es decir, el tiempo que no se emplea en trabajar en el proyecto.
- **Tiempo delta.-** Es el tiempo que se obtiene de la resta del tiempo empleado en el proyecto menos el tiempo de

interrupción.

- **Fase.-** Aquí se introduce el nombre de la fase en la que se trabaja.
- **Comentarios.-** Se tiene que procurar hacer todo tipo de comentarios útiles que puedan recordar ciertas circunstancias.

Todos los tiempos llevan el formato de horas y minutos (HH:MM), en caso de que sólo sean minutos los empleados entonces se utiliza el formato (:MM) (4).

Conjuntamente con éste se debe trabajar con el formato de registro de defectos que se muestra en la Figura 1.3.

| REGISTRO DE DEFECTOS ENCONTRADOS   |        |               |            |           |                  |                 |
|--|--------|---------------|------------|-----------|------------------|-----------------|
| <b>Nombre:</b> Gina Ruiz López   |        |               |            |           | Tipos Defectos:  |                 |
| <b>Equipo:</b> TSOFT   |        |               |            |           | 10 Documentación | 60 Comprobación |
| <b>Fecha:</b> 08 de Enero del 2005   |        |               |            |           | 20 Sintaxis      | 70 Datos        |
| <b>Instructor:</b> Ing. Mónica Villavicencio   |        |               |            |           | 30 Construcción  | 80 Función      |
|  |        |               |            |           | 40 Asignación    | 90 Sistema      |
|  |        |               |            |           | 50 Interfaz      | 100 Entorno     |
| Fecha  | Número | Tipo          | Insertado  | Eliminado | T. Correg.       | Def. Correg.    |
| 08/01/2005   | 1      | Documentación | Planeación | Diseño    | 20 Min           | X               |
| Descripción: Se corrigió errores de sintáxis y faltas ortográficas en los requerimientos de Nómina |        |               |            |           |                  |                 |
| Fecha  | Número | Tipo          | Insertado  | Eliminado | T. Correg.       | Def. Correg.    |
| 08/01/2005   | 2      | 70            | Code       | Pruebas   | 10               | X               |
| Descripción: Se corrigió variables mal declaradas  |        |               |            |           |                  |                 |
| Fecha  | Número | Tipo          | Insertado  | Eliminado | T. Correg.       | Def. Correg.    |
| 08/01/2005   | 3      | 20            | Code       | Compile   | 10               | X               |
| Descripción: Se corrigió declaración de if, else, if mal declarados                                |        |               |            |           |                  |                 |

Figura 1.3 Plantilla LOGD (5)



El propósito general de llevar este registro de defectos reside en promover la mejora continua en el desarrollo de cada proyecto.

Cada fase de PSP debe de contar con un registro de defectos, ya sean revisiones, compilaciones y/o pruebas (4).

Los campos que intervienen en este formato son (4):

- **Fecha.-** Fecha en la que se detectó el defecto.
- **Número.-** Número de defecto (secuencial) encontrado en cada programa.
- **Tipo.-** El tipo de defecto va desde 10 hasta 100 en intervalos de 10 en 10.
- **Insertado.-** Nombre de la fase en la que se encontró el defecto.
- **Eliminado.-** Nombre de la fase en la que se removió el defecto.
- **Tiempo corregir.-** Es el tiempo que tomó en reparar el defecto encontrado.
- **Defecto arreglado.-** Este campo quiere decir si se encontró algún defecto extra mientras se reparaba un defecto detectado anteriormente, en caso de no haber ninguno, se coloca una X.

Cada vez que se encuentra un defecto, se debe mantener un

registro de estos errores ya que para proyectos futuros es importante evitar, repetir o caer en los mismos errores. De esta manera se evita que resulte infructuoso utilizar PSP para mejorar constantemente porque el propósito es el de apoyar a los ingenieros a estar innovando sus métodos de desarrollo (4).

## **1.5. Herramientas del PSP**

Las diferentes plantillas utilizadas para el control personal fueron desarrolladas en formato XLS. . La Herramienta fue seleccionada gracias a que brinda mayores facilidades en el cálculo de resultados, y debido a su gran facilidad de uso.

## **1.6. Definición de TSP**

El TSP es un modelo o proceso de trabajo en equipo enfocado a aminorar varios de los problemas, tanto técnicos como administrativos, que se presentan en el desarrollo de software. El TSP provee un esquema de trabajo donde cada desarrollador tiene perfectamente definido sus roles, sus actividades, y sus responsabilidades. Asimismo, el TSP incluye procedimientos para la mejora continua del proceso de desarrollo, para mejorar la calidad del software producido, para mejorar la estimación del

tiempo de desarrollo, para la disminución de defectos en el producto y para promover la integración del equipo de desarrollo. Es decir, el TSP apoya tanto al equipo de desarrollo como a los administradores del proyecto para la culminación a tiempo y dentro de presupuesto de proyectos de desarrollo de software (4).

Trabajar en equipo no es tan fácil como podría suponerse, se necesita un método para saber como trabajar unidos, para definir el trabajo que debe hacerse y como plantear una estrategia para realizarlo. Para poder mantener una buena relación de equipo es necesario tener metas comunes, estar de acuerdo en el plan de acción y tener un liderazgo apropiado. Se necesita que cada miembro del equipo entienda las virtudes y carencias de los otros miembros, que los apoye y que esté dispuesto a pedir ayuda cuando se necesite. Trabajar en equipo no es una habilidad que se adquiere al nacer, se adquiere a través de la práctica y se mejora día a día con la experiencia (4).

## **1.7. Principios del TSP**

El objetivo principal de TSP es completar con éxito a través de varios ciclos de desarrollo incremental un proyecto de software con calidad, siguiendo fielmente el proceso y manteniendo durante

cada ciclo de desarrollo un equipo eficiente y colaborativo (4).

TSP es una guía paso a paso para lograr un proyecto de software en equipo. Enseña como aplicar conocimientos de ingeniería de software y procesos en un ambiente de trabajo en equipo. Define claramente los roles que cada miembro debe desempeñar, así como sus responsabilidades. Nos muestra qué debemos hacer, cómo hacerlo y cuándo hacerlo. Permite practicar y desarrollar una buena actitud de equipo de trabajo (4).

Para lograr un aprendizaje óptimo de TSP, se debe de tener experiencia en programación y un conocimiento previo de PSP. Nuestro equipo de trabajo dedicó tres meses al aprendizaje de TSP y PSP.

TSP proporciona un balance entre El/Los proceso(s), El producto y El equipo de trabajo. Sus fases y tareas están bien definidas. Contiene todas las formas, guiones y estándares necesarios para poder registrar y seguir el proceso. Nos enseña los procedimientos para iniciar un proyecto, los pasos para poder guiarlo y nos muestra como analizar y reportar los datos obtenidos durante todo el proyecto (4).

Se escogió la metodología PSP y TSP ya que era la más aplicable para el desarrollo del proyecto, haciendo una breve comparación con la metodología XP (Extreme Programming) podemos resumir los siguientes aspectos importantes:

La metodología XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes. Para el desarrollo de nuestro proyecto no era aplicable debido a que todos los miembros del equipo trabajamos y no era posible visitar al cliente periódicamente, otro factor que influyó fue la distancia de las instalaciones físicas del cliente, por lo tanto no se podía realizar visitas a menudo.

La metodología XP propone roles tales como: Programador, Cliente, Encargado de Pruebas, Encargado de seguimiento, Entrenador, Consultor, Gestor (Vínculo en entre clientes y programadores), nuestro equipo de trabajo se compone de cinco integrantes, los cuales fueron adaptables fácilmente a los cinco roles definidos en TSP. No se contaba con el recurso humano para aplicar la metodología XP.

Además se escogió la Metodología PSP y TSP para el desarrollo

del proyecto ya que la ESPOL había adquirido material relacionado con la metodología el mismo que estaba a nuestra disposición.

## **1.8. Modelo del TSP**

Existen muchas formas para lograr un diseño de un proceso, en TSP existen siete decisiones de diseño principales, que en conjunto constituyen su estrategia. A continuación se describen brevemente (4):

1. Proveer un esquema simple que se construya basándose en PSP.- El entrenamiento previo en PSP facilita el aprendizaje de TSP, muchas formas y guiones son muy similares y la disciplina adquirida es esencial al aplicarse en TSP.
2. Desarrollar productos en varios ciclos.- En un proyecto completo de TSP se pueden completar alrededor de dos o tres ciclos de desarrollo de un proyecto. Cada uno incluye requerimientos completos, planeación, diseño, Implementación y pruebas. En el primer ciclo el objetivo es tener un producto que sirva de base para los siguientes ciclos. En cada ciclo pueden hacerse cambios de roles, un ajuste en el proceso o tomarse la decisión de aumentar la disciplina en la calidad. Después de completar dos o más ciclos ya se sabe con

precisión que es lo que está funcionando mejor dentro del equipo.

3. Establecer medidas estándar para rendimiento y calidad.-Sin medidas objetivas de un proyecto no podemos hablar de alta calidad en el trabajo. PSP proporciona las medidas que se necesitan para evaluar la calidad del trabajo y enseña como poder interpretarlas y aplicarlas. TSP enfatiza las metas y métricas que permitan visualizar los beneficios de las medidas de calidad evaluando la planeación y los datos registrados del proyecto.
4. Proporcionar medidas precisas para el equipo con los datos registrados en TSP.- Se hace transparente el desempeño personal de cada miembro del equipo. Todos los integrantes del equipo conocen como han trabajado los demás y esto ayuda a mejorar el esfuerzo del equipo en general.
5. Utilizar evaluaciones de equipo y rol.- La idea de realizar evaluaciones es estar mejor informados acerca del rendimiento de otros y del equipo en su totalidad. Si los miembros del equipo están dispuestos a hacer evaluaciones honestas, el instructor o gerente podrá estar mejor informado. TSP cuenta con evaluaciones para el equipo y los roles desempeñados, la idea es evaluar como fue el desempeño de cada rol y como

impactó en el rendimiento del equipo.

6. Enfatizar en la disciplina.- Los ingenieros de software tradicionalmente no tienen un trabajo personal disciplinado, no existen modelos de rol que seguir. TSP requiere de alta disciplina en el seguimiento de las tareas dentro del proceso.
7. Proveer una guía sobre los problemas de los equipos de trabajo.- Hasta en el mejor proyecto es común tener problemas al trabajar en equipo. Cada miembro del equipo tiene su propia personalidad y esto puede generar problemas en el equipo, sin embargo con una buena guía y apoyo se puede llegar a ser un miembro efectivo en el equipo.

## 1.9. Plantillas del TSP.

**STRAT (Strategic Form):** Esta forma es usada durante el desarrollo de la estrategia para asignar funciones del producto a los ciclos (5).

Los campos que intervienen en este formato son (5):

- **Nombre.-** Nombre de la persona que esta ingresando la plantilla.
- **Equipo.-** Nombre del equipo que esta desarrollando el proyecto.
- **Parte/Nivel.-** Fase en la que se encuentra el proyecto.



- **Fecha.-** Fecha en la que se realiza la estrategia.
- **Instructor.-** Nombre del instructor a cargo del proyecto.
- **Incremento.-** Campo que indica el incremento que se esta desarrollando
- **Referencia.-** Usar esta columna para enlistar las funciones
- **Función.-** columna que lista todas las funciones que estarán incluidas en todos los incrementos
- **Ciclo Loc.-** Columna para estimar el LOC de cada función
- **Ciclo Hora.-** Columna que indica el tiempo estimado requerido para desarrollar la función

| ESTRATEGIA   |  |                      |     |   |            |      |   |
|--------------|--|----------------------|-----|---|------------|------|---|
| Nombre:      | Grupo Oficina                                      | Fecha: 25-Julio-2005 |     |   |            |      |   |
| Equipo       | Tsoft  | Instructor: MV       |     |   |            |      |   |
| Parte/Nivel: | Estrategia   | Incremento: 3        |     |   |            |      |   |
|              |  | Ciclo Loc            |     |   | Ciclo Hora |      |   |
| Referencia   | Función  | 1                    | 2   | 3 | 1.00       | 2.00 | 3 |
| <b>1</b>     | <b>Administración de los Datos de los Clientes</b> |                      |     |   |            |      |   |
| 1.1          | Ingresar Datos del Proveedores                     | 250                  |     |   | 5.00       |      |   |
| 1.2          | Modificar Datos del Proveedores                    |                      | 65  |   |            | 2.00 |   |
| 1.3          | Eliminar Datos del Proveedor                       |                      | 65  |   |            | 2.00 |   |
| 1.4          | Consultar Datos del Proveedores                    | 55                   |     |   | 1.00       |      |   |
| <b>2</b>     | <b>Administración General Contratos</b>            |                      |     |   |            |      |   |
| 2.1          | Ingresar Datos de Órdenes de Compra                | 100                  |     |   | 4.00       |      |   |
| 2.2          | Eliminar Órdenes de Compra                         |                      | 80  |   |            | 3.00 |   |
| 2.4          | Consultar Datos de Órdenes de Compra               | 55                   |     |   | 2.00       |      |   |
| <b>3</b>     | <b>Administración General Facturas</b>             |                      |     |   |            |      |   |
| 3.1          | Ingresar Datos de Facturas                         | 550                  |     |   | 45.00      |      |   |
| 3.2          | Modificar Datos de Facturas                        |                      | 100 |   |            | 5.00 |   |

Figura 1.4 Plantilla STRAT (5)

**PEER (Team and Peer Evaluation):** Esta forma permite llevar a cabo las evaluaciones equipo (5).

Los campos que intervienen en este formato son (5):

- **Nombre.-** Nombre de la persona que esta ingresando la plantilla.
- **Equipo.-** Nombre del equipo que esta desarrollando el proyecto.
- **Fecha.-** Fecha en la que se realiza la evaluación.
- **Instructor.-** Nombre del instructor a cargo del proyecto.
- **Incremento.-** Campo que indica el incremento que se esta desarrollando
- **Número de Semana.-** Campo en el que indica el número de la semana en la que se realiza la evaluación.
- **Evaluación del trabajo requerido y la relativa dificultad en % durante el incremento.-** Campo en el que se evaluará a cada miembro del equipo, de acuerdo al trabajo y la dificultad de las tareas asignadas por incremento.
- **Tasa total del equipo contra cada criterio.-** Campo en la que se evaluará a los miembros del equipo de acuerdo a su desempeño en el trabajo, la escala de evaluación va de 1 al 5, en la cual 1 es inadecuado y 5 es superior.
- **Tasa de la contribución total de cada rol.-** Campo en la que se evaluará a los miembros del equipo de acuerdo a su contribución al equipo, la escala de evaluación va de 1 al 5, en la cual 1 es el más bajo promedio de contribución y 5 el más

alto.

- **Tasa de cada rol por utilidad y soporte.-** Campo en la que se evaluará a los miembros del equipo de acuerdo a su utilidad y soporte en el desarrollo del proyecto.
- **Tasa de cada rol por su desempeño.-** Campo en la que se evaluará a los miembros del equipo de acuerdo a su desempeño ejecutando actividades del rol al que se le fue asignado.

| <b>Evaluación de equipos y miembros</b>   |                      |   |   |   |   |
|---|----------------------|---|---|---|---|
| <b>Nombre:</b> Mauricio Echeverría  | <b>Equipo:</b> TSOFT |   | <b>Instructor:</b> Mónica Villavicencio |   |   |
| <b>Fecha:</b> 15-Marzo-2006   |                      |   |   |   |   |
| Primer Incremento Bodega y Nomina   |                      |   |   |   |   |
| Taza de la contribución total de cada rol. Encierra en un círculo donde 1 (inferior) a 5 (superior) |                      |   |   |   |   |
| Lider del Equipo  | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Desarrollo   | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Calidad  | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Planeación   | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Configuración  | 1                    | 2 | 3                                       | 4 | 5 |
| Contribución Total (100%)   | 1                    | 2 | 3                                       | 4 | 5 |
| Segundo incremento Ordenes de Trabajo y Ordenes de Compra   |                      |   |   |   |   |
| Taza de la contribución total de cada rol. Encierra en un círculo donde 1 (inferior) a 5 (superior) |                      |   |   |   |   |
| Lider del Equipo  | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Desarrollo   | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Calidad  | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Planeación   | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Configuración  | 1                    | 2 | 3                                       | 4 | 5 |
| Contribución Total (100%)   | 1                    | 2 | 3                                       | 4 | 5 |
| Tercer Incremento Integración Ordenes de Compra y Bodega  |                      |   |   |   |   |
| Taza de la contribución total de cada rol. Encierra en un círculo donde 1 (inferior) a 5 (superior) |                      |   |   |   |   |
| Lider del Equipo  | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Desarrollo   | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Calidad  | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Planeación   | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Configuración  | 1                    | 2 | 3                                       | 4 | 5 |
| Contribución Total (100%)   | 1                    | 2 | 3                                       | 4 | 5 |
| Cuarto Incremento Presupuesto por Obra y Facturación  |                      |   |   |   |   |
| Taza de la contribución total de cada rol. Encierra en un círculo donde 1 (inferior) a 5 (superior) |                      |   |   |   |   |
| Lider del Equipo  | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Desarrollo   | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Calidad  | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Planeación   | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Configuración  | 1                    | 2 | 3                                       | 4 | 5 |
| Contribución Total (100%)   | 1                    | 2 | 3                                       | 4 | 5 |
| Quinto incremento Costos de Producción  |                      |   |   |   |   |
| Taza de la contribución total de cada rol. Encierra en un círculo donde 1 (inferior) a 5 (superior) |                      |   |   |   |   |
| Lider del Equipo  | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Desarrollo   | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Calidad  | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Planeación   | 1                    | 2 | 3                                       | 4 | 5 |
| Administrador de Configuración  | 1                    | 2 | 3                                       | 4 | 5 |
| Contribución Total (100%)   | 1                    | 2 | 3                                       | 4 | 5 |

Figura 1.5 Plantilla PEER (5)

**TASK (Task Planning Template):** El propósito de esta forma es:

Estimar el tiempo desarrollado para cada tarea del proyecto (5).

Los campos que intervienen en este formato son (5):

- **Nombre.-** Nombre de la persona que esta ingresando la plantilla.
- **Equipo.-** Nombre del equipo que esta desarrollando el proyecto.
- **Parte/Nivel.-** Fase en la que se encuentra el proyecto.
- **Fecha.-** Fecha en la que se realiza la planeación de las tareas.
- **Instructor.-** Nombre del instructor a cargo del proyecto.
- **Incremento.-** Campo que indica el incremento que se esta desarrollando
- **Tareas.-** Campo que indicará las tareas a realizar
  - **Fase.-** Columna que indica la fase en la cual la tarea será desarrollada
  - **Parte-** Indica la parte a la que esta asociada la tarea.
  - **Nombre de Tarea-** Nombre de la tarea que se va a desarrollar
  - **# de Ingenieros.-** Ingenieros que participarán en las tareas
- **Planeación de Horas.-** Horas planeadas para cada tarea
  - **Líder del equipo.-** Campo en la que indicará las

horas planeadas para el líder del equipo

- **Administrador de Desarrollo.-** Campo en la que indicará las horas planeadas para el administrador de desarrollo
- **Administrador de Planeación.-** Campo en la que indicará las horas planeadas para el administrador de planeación.
- **Administrador de Calidad.-** Campo en la que indicará las horas planeadas para el administrador de calidad.
- **Administrador de Configuración.-** Campo en la que indicará las horas planeadas para el administrador de configuración.
- **Total horas equipo.-** Total de horas planificadas para todos los miembros del equipo.
- **Unidad tamaño.-** Columna que indica el tamaño de la unidad. Para un programa el tamaño de la unidad es el LOC y para los documentos el tamaño de la unidad es por página.

- **Plan Tamaño/Valor.-**
  - **Tamaño.-** Columna que indica el tamaño de la tarea.
  - **Valor Planificado.-** Porcentaje del valor planificado por cada tarea.
- **Actual.-** Campo que indica las horas actuales en la que los miembros del equipo desarrollaron las tareas.
  - **Horas.-** Total de horas al completar una tarea

| PLANEACIÓN DE TAREAS       |              |   |                     |              |                 |                 |              |                    |                   |                                  |             |       |        |
|----------------------------|--------------|---|---------------------|--------------|-----------------|-----------------|--------------|--------------------|-------------------|----------------------------------|-------------|-------|--------|
| Nombre: Denisse Echeverria |              |   |                     |              |                 |                 |              |                    |                   | Fecha: 05-agosto-2005            |             |       |        |
| Equipo: TSOFT              |              |   |                     |              |                 |                 |              |                    |                   | Instructor: Mónica Villavicencio |             |       |        |
| Parte / Nivel: Desarrollo  |              |   |                     |              |                 |                 |              |                    |                   | Incremento: Segundo              |             |       |        |
| Tareas                     |              |   | Planeación de Horas |              |                 |                 |              |                    |                   |                                  | Tamaño Plan |       | Actual |
| Fase                       | Parte        | Nombre Tarea                              | # Ingenieros        | Lider Equipo | Adm. Desarrollo | Adm. Planeación | Adm. Calidad | Adm. Configuración | Total Hora Equipo | Unidad                           | Tamaño      | Horas |        |
| Desarrollo                 | Codificación | Administración de Datos de Proveedores    | 1                   | 8            | 2               | 1               | 1            | 1                  | 15                | Loc                              | 435         | 18    |        |
| Desarrollo                 | Codificación | Administración de las Ordenes de Compra   | 1                   | 49           | 4               | 2               | 4            | 4                  | 64                | Loc                              | 800         | 73    |        |
| Desarrollo                 | Codificación | Administración de Documentos Relacionados | 1                   | 16           | 3               | 4               | 3            | 3                  | 30                | Loc                              | 310         | 35    |        |
|                            |              |   |                     |              |                 |                 |              |                    |                   |                                  |             |       |        |
|                            |              |   |                     |              |                 |                 |              |                    |                   |                                  |             |       |        |
|                            |              |   |                     |              |                 |                 |              |                    |                   |                                  |             |       |        |

Figura 1.6 Plantilla TASK (5)

**CCR (Configuration Change Request):** Esta forma se utiliza para establecer el control de configuración (5).

Los campos que intervienen en este formato son (5):

- **Nombre.-** Nombre de la persona que esta ingresando la plantilla.
- **Equipo.-** Nombre del equipo que esta desarrollando el proyecto.
- **Parte/Nivel.-** Fase en la que se encuentra el proyecto.
- **Fecha.-** Fecha en la que se realiza la petición de cambio
- **Instructor.-** Nombre del instructor a cargo del proyecto.
- **Incremento.-** Campo que indica el incremento que se esta desarrollando
- **Información del producto.-** Columna que contiene la información del producto o cambio (5):
  - **Nombre.-** Nombre del producto o numero de identificación
  - **Propietario del Producto.-** Nombre de persona que desarrollo el producto.
  - **Tamaño Producto.-** Estimado o tamaño actual de el producto o cambio
  - **Medida.-** Medida del producto por Ejemplo: LOC o páginas



- **Dirección de Respaldo.-** Columna en la que describe la dirección de respaldo del producto.
- **Información de Cambio.-** Campo en el que se detallarán las razones, beneficios, impacto y descripción del cambio que se realizará (5).
  - **Razones.-** Campo en que se detallará las razones por la que se realiza el cambio por Ejemplo: Cambio de Standard de documentación, cambio de requerimientos, etc.
  - **Beneficios.-** Campo en el que se detallará los beneficios del cambio, por Ejemplo: Mejora de la calidad del sistema, etc.
  - **Impacto.-** Campo en el que se detallará el impacto del cambio que se realizará, horas requeridas para realizar el cambio por los miembros del equipo.
- **Estado.-** Campo en el que se indicará si la petición de cambio fue aprobada, negada o de que se necesita más información acerca del cambio (5).
- **Aprobación.-** Campo en el que los involucrados darán la aprobación al cambio (5).

| REQUERIMIENTO DE CAMBIO DE CONFIGURACIÓN  |   |
|---|---|
| Nombre: Jose Luis Asencio   | Fecha: 10 - Marzo - 2005                        |
| Equipo: TSOFT   | Instructor: Mónica Villavicencio                |
| Parte/Nivel: Desarrollo   | Incremento: Uno                                 |
| <b>Información del Producto</b>   |   |
| Nombre: Módulo de Nómina  | Propietario del Producto: Arturo Aguilera       |
| Tamaño Producto/Cambio: 5830  | Medida: LOC                                     |
| Dirección de Respaldo: PC Gina Ruiz: C:\Mis Documentos\Nomina\Base  |   |
| <b>Información de Cambio</b>  |   |
| <b>Razones:</b> Cambios en Modelo Relacional del Módulo de Nómina en la Tabla "RO_Personal" por división en dos tablas "Ro_Personal" y "Ro_Laboral" |   |
|   |   |
|   |   |
| <b>Beneficios:</b> Llevar un mejor control de los Ingresos y Salidas del Personal de la Empresa   |   |
|   |   |
|   |   |
| <b>Impacto:</b> Alto, debido a que corresponde a un cambio en la estructura de la base de datos   |   |
|   |   |
|   |   |
| <b>Estado</b>   |   |
| <b>Aprobado:</b> <input checked="" type="checkbox"/>  | Información Adicional: <input type="checkbox"/> |
| Desaprobado: <input type="checkbox"/>   |   |
| Información Necesaria:  |   |
|   |   |
|   |   |
| <b>Aprobación</b>   |   |
| Propietario del Producto: Arturo Aguilera   | Fecha: 12-Enero-2005                            |
| Administrador Calidad/Proceso: Gina Ruiz  | Fecha: 12-Enero-2005                            |
| Administrador de Configuración: José Luis Asencio   | Fecha: 18-Enero-2005                            |

Figura 1.7 Plantilla CCR (5)

# CAPÍTULO 2

## 2. ADMINISTRACION DEL PROYECTO

### 2.1. Visión del Proyecto.

**Título del proyecto:** Desarrollo de un Sistema Integrado de Control de Nómina, Ordenes de Compra y Facturación.

Este sistema se implementará aplicando dos metodologías:

- 1) Personal Software Process (PSP).
- 2) Team Software Process (TSP).

PSP y TSP nos permitirán controlar individualmente y como equipo nuestro desempeño en el desarrollo del presente proyecto de tesis. Un equipo de trabajo que utiliza TSP está conformado por 5 personas, las mismas que durante todos los ciclos del proyecto asumen los siguientes roles:

Líder de Equipo

Administrador de Calidad.

Administrador de Planeación.

Administrador de Desarrollo.

Administrador de Configuración.

La asignación de roles a cada integrante del equipo se explicará

posteriormente en la sección “El sistema equipo”. Para mayor información refiérase al Apéndice T.

#### **2.1.1. Propósito.**

El propósito de esta tesis es obtener una disciplina de desarrollo de productos de calidad, recolectando y analizando métricas para poder definir estándares que nos ayuden en la mejora de los procesos y demostrar así la aplicabilidad de PSP y TSP.

#### **2.1.2. Alcance**

Obtener un producto de software que cumpla con los procedimientos establecidos por PSP y TSP, realizando un seguimiento de la información relativa a la percepción del cliente con respecto al cumplimiento de sus requisitos. El producto a desarrollar es un software denominado “Sistema Integrado de Nómina, Órdenes de Compra y Facturación”, que permitirá ingresar, modificar, consultar e imprimir datos e información relacionada a estos procesos y que apoyen a la toma de decisiones de los directivos de la empresa.

## **2.2. Los Apostadores**

### **2.1.3. Definición**

Los Apostadores son gente que será afectada por el sistema y que tienen una influencia directa o indirecta sobre los requisitos del sistema. Esto incluye usuarios finales del sistema, gerentes y otros involucrados en los procesos organizacionales influenciados por el sistema, ingenieros responsables por el desarrollo y mantenimiento del sistema, clientes de la organización que usarán el sistema para proveer algunos servicios, cuerpos externos tales como autoridades reguladoras o de certificación (6).

### **2.1.4. Descripción**

Hemos definido dos grandes grupos de sistemas, el primero es el “socio-económico” y el segundo el “del proyecto”.

En el sistema socio-económico encontramos varios subsistemas, los mismos que describimos a continuación:

**Sistema de Competidores:** Se definen así a todos las entidades, instituciones o personas que desarrollan soluciones de software que pueden interferir en el desarrollo del sistema del proyecto, objeto de esta tesis.

**Sistema Ecológico:** Se refiere al sistema del medio ambiente, el cual puede afectar indirectamente al equipo de trabajo, y, en algunos casos, retrasar la ejecución del proyecto.

**Sistemas Sociales:** En él encontramos todos los sistemas que tienen lugar en la economía del país, del cliente, o del grupo de trabajo, entre ellos podemos citar a:

- Instituto Ecuatoriano de Seguridad Social, IESS.
- Servicio de Rentas Internas, SRI.

**Sistemas de Proveedores:** Como su nombre lo indica, son los que proveen alimentos, servicios básicos, seguros, materiales, consumibles, maquinarias, vehículos, etc. Este sistema es muy importante y es tomado en cuenta en todo el desenvolvimiento del proyecto, al igual que los otros sistemas.

El sistema del proyecto se encuentra dividido en 2 subsistemas, que son:

### **El Sistema Cliente**

El sistema cliente representa a la entidad en la que se desarrolló la solución del problema por parte del equipo de desarrollo y se llama MOLEMOTOR S.A. a esta entidad la denotaremos "EL CLIENTE". Este sistema cliente se encuentra desglosado en varios niveles, que son:

Nivel Ejecutivo.- En el cual se toman las decisiones trascendentales para la empresa. En este nivel se encuentran las personas que ocupan los siguientes cargos:

Gerencia General

Gerencia Administrativa - Financiera.

Nivel Asesor.- Está conformado por profesionales que asesoran a los tomadores de decisiones de la empresa. En este nivel participan:

Asesor Legal

Asesor Informático.

### Auditoría externa

Nivel de Apoyo.- En este nivel se encuentra todo el staff administrativo de la empresa, los cuales serán los usuarios directos del sistema informático. En el nivel de apoyo intervienen:

#### Personal Administrativo:

Ingenieros de Obra

Contadora

Control Interno

Auxiliar de Servicios

Jefe de Recursos Humanos

Auxiliar Contable

Jefe informático

Secretaria

Nivel Operativo.- En este nivel se ha considerado al personal de planta que labora en la empresa.

#### Personal Operativo.

Jefe de Almacén

Supervisores

Operadores

Auxiliares de Almacén



**El Sistema del Equipo:** Está conformado por cinco personas:

Líder de equipo: Arturo Aguilera.

Administrador de Desarrollo: Mauricio Echeverría.

Administrador de Calidad: Gina Ruiz.

Administrador de Planeación: Denisse Echeverría.

Administrador de Configuración: José Luis Asencio

Mera. Ver Anexo T. Ecosistema de Proyecto

### **2.3. Estructura Organizacional**

Nuestro equipo de trabajo está compuesto por 5 personas:

Líder: Arturo Aguilera

Administrador de Calidad: Gina Ruiz

Administrador de Planeación: Denisse Echeverría

Administrador de Desarrollo: Mauricio Echeverría

Administrador de Configuración: José Asencio

A continuación se describen brevemente los roles desempeñados por cada miembro del equipo:

Líder.- Dirige el equipo y se asegura que los ingenieros (estudiantes tesistas) reporten los datos del proceso y completen su trabajo de la manera planeada.

Administrador de Desarrollo.-Conduce y guía al equipo en la definición, diseño, desarrollo y pruebas del producto.

Administrador de Calidad.- Apoya al equipo en elaborar el plan de calidad, y en mantener seguimiento en el proceso y en la calidad del producto.

Administrador de Planeación.- Apoya y guía a los integrantes del equipo en la planeación y seguimiento de su trabajo.

Administrador de Configuración.- Colabora con el equipo en la determinación, obtención y mantenimiento de las herramientas necesarias para cumplir con las necesidades administrativas y aplicar la tecnología definida.

La estructura organizacional con respecto al equipo es la siguiente.

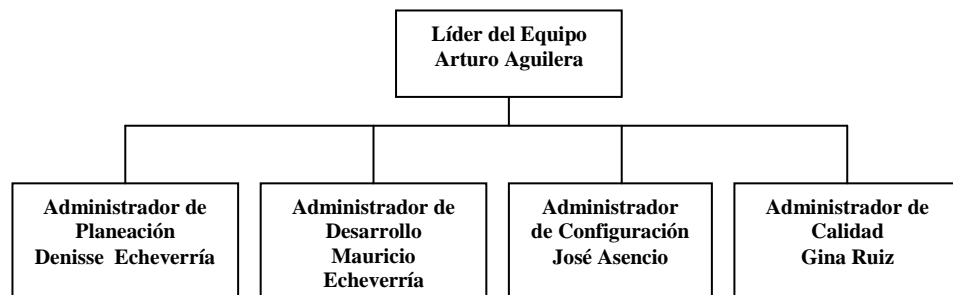


Figura 2.1 Estructura Organizacional

## 2.4. Etapas del Proyecto

El trabajo del proyecto está separado en tres partes básicas: definición, desarrollo y producción.

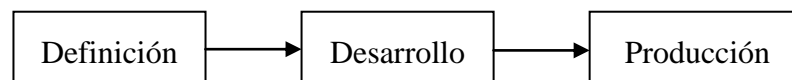


Figura 2.2 Etapas del Proyecto

**Definición.-** Durante el proceso de definición se identifica la información que va a ser procesada, se realiza el lanzamiento de la posible solución del software y se especifican las tareas del equipo para el desarrollo del producto.

Esta sección está enfocada en 4 fases muy importantes, que son: fase de introducción, fase de lanzamiento, fase de estrategia, fase de planeación.

**Desarrollo.**- Durante esta etapa se realizan los procesos generales del desarrollo del software, agrupados en las siguientes fases: requerimientos, diseño e implementación.

**Producción.**- Esta etapa se enfoca en las últimas fases del desarrollo del software, como son: pruebas.

En esta fase se entrega el producto requerido por el cliente.

En el siguiente gráfico se muestran las etapas y sus fases.

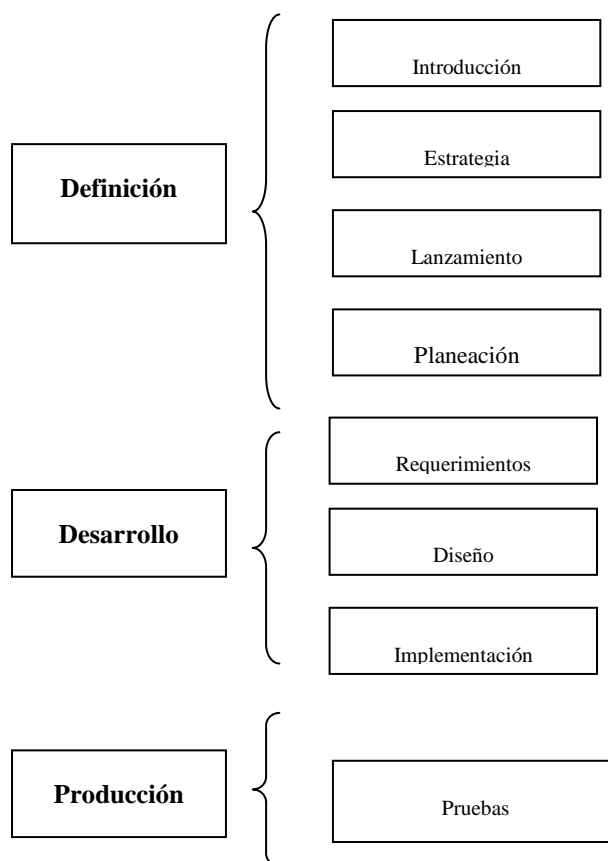


Figura 2.3 Divisiones de las etapas del Proyecto

## 2.5. Metodología de Desarrollo

El modelo de desarrollo incremental fue utilizado para desarrollar el producto de software, objeto de nuestra tesis.

En un proceso de desarrollo incremental se identifican cuales servicios son los más importantes y cuales son los menos importantes. Entonces se definen varios incrementos en donde cada uno proporciona un subconjunto de funcionalidad del sistema. La asignación de servicios a los incrementos depende de la prioridad del servicio. Los servicios de prioridad más alta son los que se entregan primero al cliente. En nuestro proyecto los incrementos fueron identificados según sus necesidades más críticas.

En el desarrollo de los incrementos identificados, nos dimos cuenta que los servicios que sufrieron mas pruebas fueron los primeros.

Cada incremento consta de cuatro fases: análisis, diseño, implementación, y pruebas.

Este modelo cumple con el efecto iterativo, ya que para obtener el producto final se deben de implementar e integrar cada uno de los

incrementos (7).

A continuación citamos las ventajas más significativas del método incremental:

- Naturaleza interactiva pero se diferencia de aquellos en que al final de cada incremento se entrega un producto completamente operacional.
- Entrega el software en partes pequeñas, pero utilizables, llamadas incrementos.
- Cada incremento se construye sobre aquel que ya ha sido entregado.
- El primer incremento es un producto esencial (afronta los requisitos básicos).
- Como resultado de la utilización/evaluación se desarrolla un plan para el incremento siguiente.
- El modelo es muy útil cuando se tiene una fecha de entrega imposible de cambiar.
- Este modelo permite administrar fácilmente el proyecto en comparación con el modelo en cascadas, además su separación en el diseño y la implementación conduce a sistemas robustos susceptibles al cambio (8).

- Con este modelo podemos tomar en cuenta el primer incremento como experiencia para poder desarrollar los siguientes.
- Podemos identificar los servicios que proveerá el sistema, a los cuales los organizamos según su importancia.

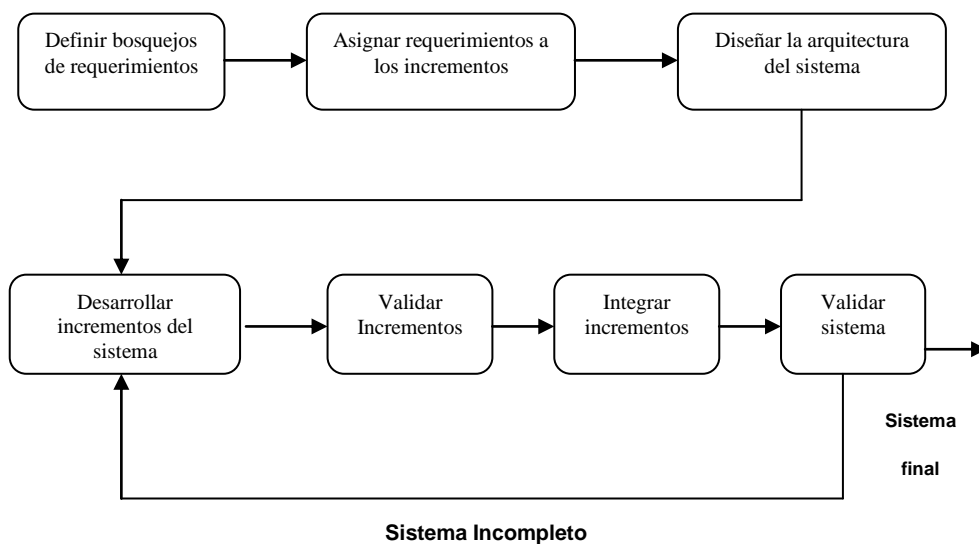


Figura 2.4 Desarrollo Incremental (8)

Inicialmente se determinaron los siguientes incrementos:

### **Incremento 1**

Módulo de Control de Inventario y Bodega - Módulo Nómina

### **Incremento 2**

Módulo de Ordenes de Trabajo – Procesos que involucran al

Módulo de Control de Inventario y Bodega con el Módulo Nómina

### **Incremento 3**

Módulo de Presupuesto por Obra – Módulo Órdenes de Compra

### **Incremento 4**

Módulo de Costos de Producción – Módulo de Facturación

## **2.6. Herramientas de desarrollo y soporte**

Las herramientas de desarrollo, soporte y comunicación que se utilizaron para este proyecto fueron las siguientes:

### **Herramientas de programación**

Visual Basic 6.0

SQL Server 2000

Crystals Reports 8.0

### **Herramientas de Control**

Visual Source Safe 6.0

Microsoft Project 2003

### **Herramientas de metodología**

Plantillas TSP y PSP

### **Utilitarios.**

Microsoft Office: Word, Excel



## Herramientas de Comunicación

Teléfono

Correo Electrónico

## 2.7. Riesgos y planes de contingencias

Un aspecto fundamental en el desarrollo de proyectos de software es el control de los riesgos.

Los riesgos surgen de la incertidumbre que rodea a las decisiones y a los resultados de los proyectos. La mayoría de individuos asocian el concepto de riesgo a la pérdida potencial de un valor, control, función, calidad o a la falta de puntualidad en el plazo de entrega de un proyecto. Otro elemento de riesgo a ser considerado es la incertidumbre en la toma de decisiones, lo cual puede conducir al no cumplimiento de las expectativas del cliente (9).

Para administrar los riesgos se deben realizar las siguientes actividades:

1. Identificar los riesgos.
2. Analizar los riesgos.
3. Elaborar planes de contingencia.
4. Controlar el estado de los riesgos.
5. Analizar los resultados y aprender de ellos.

## **Identificación de Riesgos:**

La identificación de riesgos consiste en escribir todas las inquietudes y preocupaciones que están relacionadas con el proyecto, para lo cual todos los integrantes del equipo aportan mencionando riesgos que ellos consideran estén asociados al proyecto (10).

Las fuentes utilizadas para identificar los riesgos son las siguientes:

1. La descripción de los productos y servicios.
2. El cronograma.
3. Los recursos asignados.
4. El proceso de estimación de tiempos.
5. Las restricciones.
6. Las suposiciones.

De acuerdo con la identificación, podemos categorizar los riesgos en los siguientes tipos:

| <b>Tipo de Riesgo</b> | <b>Descripción</b>  |
|-----------------------|---|
| Personal              | Referentes al equipo de desarrollo.                                     |
| Procesos              | Referente a los procesos que se realizan en el desarrollo del proyecto. |
| Producto              | Referente al producto de software que se desarrolla.                    |
| Tecnología y Equipos  | Basado en la tecnología utilizada para desarrollar el producto.         |
| Proyecto              | Referente al entorno del trabajo.                                       |

Tabla 2.1 Tipos de Riesgo

A continuación se detallan los riesgos identificados en el proyecto:

| <b>Riesgos</b>   | <b>Tipo</b>          |
|--|----------------------|
| Desentendimiento con los usuarios.   | Producto             |
| Falta de compromiso de los usuarios.   | Proyecto             |
| Abandono de un miembro del equipo.   | Proyecto             |
| Falta de capacitación en la metodología de desarrollo.                       | Personal             |
| Sobreestimación de las ventajas del empleo de nuevas herramientas o métodos. | Tecnología y Equipos |
| Cambiar de herramientas de desarrollo y soporte a mitad del proyecto.        | Tecnología y Equipos |
| Falta de control automático del código fuente.                               | Tecnología y Equipos |
| Cambios de Requerimientos.   | Producto             |
| Imprecisa estimación de Tiempos  | Producto             |

|  |                      |
|--|----------------------|
| Sobreestimación o subestimación del Tamaño del producto                            | Producto             |
| Planificación excesivamente optimista de las tareas que se realizan en el proyecto | Procesos             |
| Pérdida de tiempo en el inicio de cualquier actividad.                             | Procesos             |
| Planificar ponerse al día más adelante (Dejar todo para después)                   | Procesos             |
| Insuficiencia en el control de versiones del código fuente                         | Producto             |
| Retrasos en las especificaciones de nuevos requerimientos                          | Producto             |
| Falta de tiempo del equipo en el desarrollo de las actividades                     | Personal             |
| Retraso en las entregas del producto   | Personal             |
| Daños de materiales y equipos de trabajos  | Tecnología y Equipos |

Tabla 2.2 Clasificación de Riesgo por Tipo

## **Análisis de Riesgos:**

La severidad de cualquier riesgo se define en términos de tres características:

1. Impacto: El efecto que un riesgo tendrá si éste ocurre.
2. Posibilidad: Probabilidad de que ocurra.
3. Precisión: El nivel en que el riesgo es conocido y entendido (11).

Para mayor información, refiérase al Apéndice F. Administración de Riesgos.

## **Planes de contingencia:**

Para disminuir la ocurrencia de los riesgos se desarrolla un plan de acción contra los riesgos llamado plan de contingencia. Este incluye opciones que permitan disminuir las amenazas y aumenten las oportunidades para lograr los objetivos del proyecto (11).

Los aspectos fundamentales que identifican el plan de contingencia son los siguientes:

- Apropiado a la severidad de cada riesgo.
- Efectivo en costos.
- Oportuno para ser exitoso.
- Realista en el contexto del proyecto.
- Acordado por las partes involucradas (11).
- Para mayor información, refiérase al Apéndice F. Administración de Riesgos.

## **Control del Estado de los Riesgos:**

Para controlar los riesgos se necesita definir hitos en las fases y las etapas a cumplir en donde se visualicen el estado de los riesgos. El seguimiento se lo registra en la plantilla de control de riesgos ITL.

## **Aprendizaje de los resultados:**

Cuando se controlan los riesgos se pueden observar los cambios e implicaciones de éstos sobre el proyecto. Los resultados de la observación sirven de base para proyectos posteriores. Con el aprendizaje se puede estimar y prever la ocurrencia de un riesgo.

# CAPITULO 3

## 3. FASE DE PLANEACIÓN

### 3.1. Identificación de Entregables

Los entregables en el desarrollo del proyecto permiten mostrar de manera formal los diferentes productos que desarrolla el equipo.

Se identificaron los entregables de acuerdo a las diferentes fases del proyecto y de acuerdo a los incrementos, estos entregables son controlados e insertados en las líneas bases definidas por el equipo.

Los productos a entregar son hitos, pero estos no necesariamente son productos a entregar. Dichos hitos pueden ser registrados internamente, utilizados por los administradores para verificar avances pero no se entregan a los clientes (12).

Se identificaron dos tipos de entregables:

- Entregables enfocados al desarrollo del proyecto
- Entregables enfocados al cliente

Los entregables enfocados al desarrollo del proyecto definidos por el equipo son:

**Fase de Lanzamiento**

Estándar de documentación.

Objetivos del Equipo.

Objetivos de los miembros del equipo

Objetivos del Rol.

Objetivos del Producto.

**Fase de Estrategia**

Criterios de Estrategias.

Forma STRAT del proyecto.

Plan de Configuración.

Plan de Riesgos y Contingencias.

**Fase de Planeación**

Forma TASK.

Diagrama Gantt

**Fase de Requerimientos**

Especificación de Requerimientos del cliente del proyecto.

Forma CCR.



**Fase de Diseño**

Especificación de Requerimientos del desarrollador.

Diseño Detallado.

Forma CCR.

**Fase de Implementación**

Estándar de codificación.

Forma LOGT.

Forma LOGD.

Módulos desarrollados.

**Fase de Pruebas**

Plan de pruebas.

Pruebas unitarias.

Pruebas de regresión.

Pruebas de integración.

Pruebas de aceptación

Pruebas de usabilidad.

Forma LOGD.

Forma LOGTEST.

Forma CCR.

Forma Peer.

Los entregables enfocados al cliente son los siguientes:

- Requerimientos C del módulo de Órdenes de Compra, Facturación y Nómina.
- Pruebas unitarias, integración y de regresión del sistema.
- Manual de Usuario
- Manual Técnico.

### **3.2. Diagrama WBS**

El work breakdown consiste en segmentar el proyecto para identificar bloques de actividades y subactividades (13). Para mayor información, refiérase al Apéndice I. Work Breakdown

### **3.3. Diagrama Gantt**

Es la esquematización de las tareas a realizar por parte de los miembros del grupo, en el cual se puede hacer un seguimiento de su nivel de avance y la carga de trabajo a los integrantes del equipo.

En el diagrama Gantt podemos realizar el registro de los recursos encontrados en el sistema de proyecto, estos pueden ser recursos humanos (de trabajo) y recursos materiales (herramientas).

Se toman en cuenta para la planificación del proyecto distintas tareas que involucran a los recursos anteriormente mencionados, este proceso demanda mucho tiempo ya que de acuerdo a la planificación se desarrollará el sistema.

Para el desarrollo del diagrama Gantt se tomaron en cuenta varios factores descritos a continuación:

- Se establecieron actividades.
- Se establecieron hitos
- Se establecieron entregables.

Los que se consideran hitos para el desarrollo del proyecto son la finalización de una fase cuyos resultados internos son utilizados para verificar avances, pero no se entregan a los clientes (14).

De acuerdo al grupo de trabajo se identificaron las siguientes fases del proyecto.

- Lanzamiento.
- Estrategia.
- Planeación.
- Requerimientos
- Diseño

- Implementación
- Pruebas

### **3.4. Iteraciones del Diagrama de Gantt en base a**

#### **las etapas del proyecto**

Para identificar las iteraciones del Gantt, tomamos en consideración las etapas de proyecto, metodología de desarrollo y los riesgos.

De acuerdo a la metodología incremental se identificaron incrementos que representan productos pequeños que se van desarrollando cronológicamente.

Inicialmente para el desarrollo del proyecto se agruparon los diferentes módulos para definir cuatro incrementos que fueron los siguientes:

- Incremento 1

Módulo de Control de Inventario y Bodega - Módulo Nómina

- Incremento 2

Módulo de Órdenes de Trabajo – Procesos de Módulo de Control de Inventario y Bodega con el Módulo Nómina

- Incremento 3

Módulo de Presupuesto por Obra – Módulo Órdenes de Compra

- Incremento 4

Módulo de Costos de Producción – Módulo de Facturación

Los incrementos fueron planeados de esta manera debido a que se identificaron los módulos más importante y necesarios para la empresa en función a las prioridades, la definición de los incrementos se la realizó en conjunto con los miembros del equipo y los directivos de la empresa.

Se realizó un Gantt para el primer incremento considerando todas las fases, el tiempo estimado para realizar este incremento fue de 5 meses, tomando en cuenta que es uno de los principales módulos según las necesidades de la empresa y que sirve como línea base para la realización de los módulos posteriores.

En la ejecución del plan, el equipo identificó el riesgo en la fase de pruebas que es la falta de compromiso por parte del usuario, esto llevó a que exista descoordinación a la entrega de los incrementos y lo que ocasionó otro riesgo que es el retraso en la entrega de los productos.

Para estos riesgos identificados en el primer incremento cuyo impacto en el desarrollo del proyecto fue muy alto, el equipo decidió aplicar los planes de contingencia establecido en el plan de administración de riesgos.

Se planificó el desarrollo de los posteriores incrementos según la metodología hasta la fase de implementación. Al concluir con el desarrollo de los cuatro módulos se procedió a realizar las pruebas y entregas del sistema incrementalmente.

Por esta razón se realizaron 6 iteraciones del Diagrama Gantt que se aplicaron en este proyecto.

La primera iteración estuvo orientada hasta la fase de implementación del módulo de bodega y control de inventario con roles y nómina.

La segunda iteración se dirigió hasta la fase de implementación del módulo de órdenes de trabajo y órdenes de compra.

La tercera iteración se orientó hasta la fase de implementación de

los procesos de módulo de bodega y órdenes de compra.

La cuarta iteración se dirigió hasta la fase de implementación del módulo presupuesto por obra y facturación.

La quinta iteración se enfocó la fase de implementación del módulo de costos de producción.

Cronograma de pruebas y entrega de sistema integrados para realización de pruebas.

Cronograma de capacitación de los módulos.

Los efectos y cambios más importantes identificados luego de la implementación de los sistemas en cada uno de los incrementos se pueden señalar:

- Procesos ejecutados en menor tiempo
- Menor consumo de recursos (Papel)
- Mayor control en los procesos
- Procesos integrados e información exacta disponible Ej.: consulta de materiales disponibles, consulta de empleados próximos a finalizar su contrato, entre otros.

- Mejor integración y comunicación entre los departamentos  
Ejm: Órdenes de Pedido (Bodega) y Órdenes de Compra

Para mayor información, refiérase al Apéndice J. Diagrama de Gantt



# CAPÍTULO 4

## 4. INGENIERIA DE REQUISITOS

### 4.1. Preparación de Entrevista

Las entrevistas son la técnica de elicitación más utilizada y de hecho son prácticamente inevitables en cualquier desarrollo de software, ya que son una de las formas de comunicación más naturales entre personas. En las entrevistas se pueden identificar tres fases: Preparación, Realización y Análisis (15).

Las entrevistas no deben improvisarse, por lo que conviene realizar las siguientes tareas previas (15):

Estudiar el dominio del problema: Conocer las categorías y conceptos de la comunidad de clientes y usuarios es fundamental para poder entender las necesidades de dicha comunidad y su forma de expresarlas, y para generar en los clientes y usuarios la confianza de que el ingeniero de requerimientos entiende sus problemas (15).

Seleccionar a las personas a las que se va a entrevistar: Se debe minimizar el número de entrevistas a realizar, por lo que es

fundamental seleccionar a las personas a entrevistar. Normalmente se comienza por los directivos, que pueden ofrecer una visión global, y se continúa con los futuros usuarios, que pueden aportar información más detallada, y con el personal técnico, que aporta detalles sobre el entorno operacional de la organización. Conviene también estudiar el perfil de los entrevistados, buscando puntos en común con el entrevistador que ayuden a romper el hielo (15).

Determinar el objetivo y contenido de las entrevistas: Para minimizar el tiempo de la entrevista es fundamental fijar el objetivo que se pretende alcanzar y determinar previamente su contenido. Previamente a su realización se pueden enviar cuestionarios que los futuros entrevistados deben llenar y devolver, y un pequeño documento de introducción al proyecto de desarrollo, de forma que el entrevistado conozca los temas que se van a tratar y el entrevistador recoja información para preparar su entrevista. Es importante que los cuestionarios se preparen cuidadosamente teniendo en cuenta quien los va a responder para no incluir conceptos que se asumen conocidos y pueden no serlo (15).

Planificar las entrevistas: La fecha, hora, lugar, y duración de la

entrevista debe fijarse teniendo en cuenta siempre la agenda del entrevistado. En general, se deben buscar sitios agradables donde no se produzcan interrupciones y que resulten naturales a los entrevistados (15).

## **4.2. Requisitos C (cliente)**

Los requisitos se los puede definir como una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal (15).

El gran reto que se enfrenta es expresar con claridad lo que los clientes desean y necesitan. Las palabras solas pueden ser apropiadas, pero para muchas aplicaciones, el texto redactado necesita complementarse con figuras de varios tipos (15).

El equipo de trabajo levantó los requerimientos del cliente en forma clara y explícita luego de tener diferentes entrevistas con los usuarios y aclarar varias dudas que fueran surgiendo en el desarrollo del proyecto. Estos requerimientos fueron documentados y firmados por cada uno de los usuarios dueños de

la información de los módulos.

#### **4.2.1. Propósito**

El propósito de los requisitos C es describir lo que se supone debe hacer el programa o sistema cuando esté terminado. Sin ellos el equipo no sabe en realidad cuales son las metas que intenta lograr, no puede inspeccionar ni probar su trabajo y no puede satisfacer a sus clientes. En resumen no existe la ingeniería profesional sin los requisitos escritos (16).

El propósito de los requisitos C es definir las necesidades del cliente sobre las actividades que se realizan para llevar el control de la Nómina, Ordenes de Compra y Facturación.

#### **4.2.2. Alcance**

El alcance de los requisitos C es definir que aspectos de la aplicación se intenta cubrir en el documento de requisitos. El objetivo es guiar a los desarrolladores en la selección de un diseño adecuado para la aplicación a escala completa (16).

El alcance de los requerimientos es formar una idea sobre la lógica de negocio en la que opera la Empresa Molemotor S.A. enfocado al Control de Nómina, Órdenes de Compra y

Facturación. A partir de la recopilación de información que permita determinar los atributos, restricciones y fronteras del sistema a desarrollarse.

#### **4.2.3. Descripción Global**

La descripción global de los requerimientos es una definición suficientemente general de la funcionalidad de los sistemas de tal manera que sea poco probable que cambie en versiones futuras versiones del Proyecto.

Los sistemas de Nómina, Órdenes de Compra y Facturación automatizarán los procesos que se realizan en la empresa Molemotor S.A. para administrar la nómina de los empleados Administrativos, de Planta y los rubros quincenales que deben percibir. Crear y administrar las Órdenes de Compra. Y finalmente facturar las obras negociadas y realizadas en cada etapa de su avance, de acuerdo a las diferentes formas de pago negociadas por los clientes.

#### **4.2.4. Funciones del Proyecto**

Son el resumen de la mayoría de las funciones de la aplicación. Aquí se especifica la funcionalidad global de la

aplicación, pero pretende proporcionar las especificaciones completas (requisitos detallados). Los miembros del equipo decidimos que usar casos de uso es una manera adecuada para especificar la funcionalidad global del Sistema (16).

El proyecto consta de 3 módulos que ofrecen las facilidades de realizar consultas e imprimir reportes, actualizar constantemente los datos y asegurar la confiabilidad en el manejo de la información.

Los principales procesos que realizan estos módulos se describen a continuación:

### **Módulo de Nómina.**

Ingreso de Empleados

Actualizar Datos de los Empleados

Faltas de Trabajo

Control Accidentes de Trabajo

Descuentos a Empleados

Ingresos Adicionales

Calamidad del Empleado

Generar Rol

Administrar Beneficios de Ley

Actas de Finiquito

Cálculo de Utilidades

### **Módulo de Ordenes de Compra.**

Administrar Proveedores

Cotización Precios de Materiales

Administrar Órdenes de Compra

Administrar Documentos Relacionados

Registro Notas de Crédito

### **Módulo de Facturación.**

Administrar Clientes

Generar Contratos

Generar Facturas

Generar Facturas Detalladas

Documentos Relacionados

Anular Facturas

Administrar Notas de Crédito

Actualizar impuestos

#### **4.2.5. Características del Usuario**

Indican que tipos de personas es probable que sean los usuarios típicos del sistema, Ejemplo: Novatos, profesionales de software, contador con cinco años de experiencia, etc.

Son las personas que tienen interés en los resultados del producto como grupo y constituyen el cliente de la aplicación (16).

Para el desarrollo del proyecto se identificaron tres tipos de usuarios:

Novato.- Poco conocimiento del uso del computador.

Nivel Medio.- Conocimiento promedio en el uso del computador.

Nivel Avanzado.- Usuario experimentado y hábil en el uso del computador.

La mayoría de usuarios que utilizarán el sistema son novatos y de nivel medio.

#### **4.2.6. Restricciones**

Son todas las condiciones que pueden limitar las opciones del desarrollador. Estas opciones pueden surgir de muchas fuentes (16).



Debe determinarse cualquier restricción ambiental, presupuestaria, de tiempo, técnica y de factibilidad que limite el sistema que se va a construir (16).

El módulo de Nómina es un proceso que se alimenta de diferente información, especialmente el Rol del Pagos del personal de planta, el cual requiere un cálculo de horas en los diferentes horarios de los trabajadores, control de permisos, y cálculo de semana integral. Por lo tanto este proceso debe ser hecho en una PC que tenga instalado las herramientas cliente de SQL para correr el proceso batch que requiere para su terminación exitosa.

En el módulo de Ordenes de Compra los descuentos ingresados pueden ser únicamente del total de la Orden, el sistema no permite realizar descuentos por cada Item.

En el módulo de Facturación existe una restricción al momento de imprimir las facturas, debido a que cada cliente tiene un modelo diferente de desglose por cada factura, por lo tanto el sistema no puede parametrizar todos los casos posibles que se presenten.

#### **4.2.7. Distribución de Requisitos.**

Indica el orden en que deben implementarse los requisitos (17).

En este paso se busca identificar la importancia que tiene un requerimiento en términos de implementación. A esta característica se le conoce como prioridad y debe ser usada para establecer la secuencia en que ocurrirán las actividades de diseño y prueba de cada requisito. La prioridad de cada requerimiento dependerá de las necesidades que tenga el negocio (17).

Para la distribución de requisitos se asigna un orden basado en su prioridad y la estructura del proceso.

El equipo de desarrollo distribuyó los requisitos por módulo de la siguiente manera:

### Módulo de Ordenes de Compra

| Requisitos  | Orden de Atención |
|---|-------------------|
| Ingreso de Solicitud de Pedido.                     | Primero           |
| Aprobación de Pedido.                               | Segundo           |
| Consulta de precios.                                | Tercero           |
| Generación de Orden de Compra.                      | Cuarto            |
| Ingreso de facturas o notas de crédito              | Quinto            |
| Reportes de acuerdo a especificaciones del usuario. | Sexto             |

Tabla 4.1 Distribución de Requisitos. Módulo de Órdenes de Compra

### Módulo de Nómina

| Requisitos   | Orden de Atención |
|--|-------------------|
| Ingreso de Reporte de Horas                          | Primero           |
| Registro de Ingresos y descuentos para cada empleado | Segundo           |
| Generación de Rol de pagos                           | Tercero           |
| Reportes de acuerdo a especificaciones del usuario   | Cuarto            |

Tabla 4.2 Distribución de Requisitos. Módulo de Nómina

### Módulo de Facturación

| Requisitos  | Orden de Atención |
|---|-------------------|
| Ingreso de Proforma.                                | Primero           |
| Generar Contratos                                   | Segundo           |
| Ingresar Anticipo y monto a facturar.               | Tercero           |
| Generación de Factura.                              | Cuarto            |
| Reportes de acuerdo a especificaciones del usuario. | Quinto            |

Tabla 4.3 Distribución de Requisitos. Módulo de Facturación

### 4.3. Requisitos D (Desarrollador)

Los ingenieros de software necesitan una base para el diseño e implementación. Esta base consiste en los requisitos detallados. También se conocen como “requisito específicos”, “especificaciones funcionales”, “requisitos de desarrollador”, o “requisitos D”. Los requisitos D son una lista completa de las propiedades específicas y la funcionalidad que debe tener la aplicación, expresada con todo detalle. Cada uno de estos requisitos se enumera, etiqueta y se registra su traza durante toda

la implantación. Son consistentes con, y son un refinamiento de, los requisitos C (16).

#### **4.3.1. Requisitos Específicos**

Primero se describen los diagramas de secuencia requeridos para expresar los casos de uso de los requisitos C (cliente). Las clases requeridas para expresar estos casos de uso se utilizan para clasificar los requisitos detallados. Los diagramas de secuencia son representaciones gráficas del flujo de control y son útiles en particular para visualizar los casos de uso (16).

##### **4.3.1.1. Casos de Uso**

Los casos de uso son una técnica para la especificación de requerimientos funcionales propuesto inicialmente por [Jacobson et al., 1993] y que actualmente forma una propuesta de UML, el caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra y en la que los actores obtienen resultados observables. Los actores son personas u otros

sistemas que interactúan con el sistema cuyos requerimientos se están describiendo [Schneider y winters, 1998]. Los casos de uso presentan ciertas ventajas sobre la descripción meramente textual de los requerimientos funcionales [Firesmith, 1998], ya que facilitan la elicitación de los proyectos y son fácilmente comprensibles por los clientes y usuarios (15).

#### **4.3.1.2. Limitaciones y Fronteras del Proyecto**

En esta sección se describen las limitaciones de diseño que tienen los Módulos de Órdenes de Compra, Facturación y Nómina que se encuentran detallados en los Requisitos D (Desarrollador), **Apéndice L. Requerimientos D.**

#### **4.3.2. Restricciones de Diseño**

En esta sección se especifican las restricciones de diseño. Si no existen restricciones los diseñadores son libres de crear cualquier diseño (bueno) que satisfaga los requisitos (16).

Las restricciones de diseño para los módulos de Órdenes de Compra, Facturación y Nómina se los puede ver en los

requerimientos C de cada uno de los módulos, **Apéndice K. Requisitos C.**

### **4.3.3. Atributos del Sistema**

#### **4.3.2.1 Confiabilidad**

Significa hasta que punto se puede confiar en el funcionamiento del sistema sin errores. Por ejemplo, si un programa suma dos números, pero en un 25% de los casos el resultado que da no es correcto, es poco fiable (16). Para el atributo de confiabilidad dentro de las aplicaciones desarrolladas se efectuaron diferentes pruebas especialmente de tal manera que los resultados obtenidos sean 100% fiables.

#### **4.3.2.1 Seguridad**

Atributos del software que proporcionan control de acceso al software y los datos que maneja cada sistema (16). Para garantizar la seguridad lógica de las aplicaciones se implementó un módulo de seguridad donde se definen roles y perfiles de acceso para cada usuario dando únicamente los

accesos necesarios dependiendo del cargo y del papel que desempeñan los usuarios dentro de la empresa.

#### **4.3.2.1 Mantenimiento**

El coste de localizar y corregir defectos en un programa que aparecen durante su funcionamiento (16). Los sistemas desarrolladas se encuentran documentados con estándares de desarrollo, para que la lectura del código sea comprensible, adicionalmente se desarrollaron manuales técnicos que detallan la estructura de las tablas de tal manera que cuando existan errores el mantenimiento no sea muy complicado.



# CAPÍTULO 5

## 5. DISEÑO DETALLADO

### 5.1. Especificación de Clases y Métodos

La meta del diseño detallado es proporcionar un proyecto completo a partir del cual se pueda desarrollar un programa. Un buen proyecto para construir una casa deja al constructor tan pocas dudas como sea posible acerca de las intenciones del diseñador, y lo mismo se cumple para el diseño detallado del software (18).

Los diagramas de clases detallados deben incluir todos los nombres de atributos y operaciones, firmas, visibilidad, tipos de resultados, etc. (19).

### 5.2. Diagramas de secuencia por caso de uso

Los diagramas de secuencia son representaciones gráficas de flujo de control y son útiles para el análisis de requisitos, en particular para visualizar la ejecución de los casos de uso.

Los diagramas de secuencia requieren que se piense en términos de objetos: En un diagrama de secuencia, la vida de cada objeto

involucrado se muestra como una línea continua vertical con el nombre del objeto y su clase al principio. Cada interacción entre objetos se muestra mediante una flecha horizontal del objeto que indica el servicio al objeto que lo proporciona (20).

### **5.3. Diagramas de flujo por métodos**

Los diagramas de flujo constituyen los métodos gráficos más antiguos para describir algoritmos. Dos de los esquemas de diagramas de flujo más frecuentes son los siguientes: las decisiones (rombos) y los procesos (rectángulos).

El propósito de los diagramas de flujo es crear la inspección antes de la programación. El uso de los diagramas de flujo disminuyó a partir de las décadas de 1980 y 1990, debido en parte a que la orientación a objetos que tiende a reducir la cantidad de ramificaciones al dividir la funcionalidad en clases separadas y aprovechar el concepto de funciones virtuales (20).

### **5.4. Modelo de Trabajo para los Casos de Uso**

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un Caso de Uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema.

Un Caso de Uso es una unidad de trabajo significativo; por ejemplo crear una solicitud (20).

Cada Caso de Uso tiene una descripción que especifica la funcionalidad que se incorporará al sistema propuesto. Un Caso de Uso puede 'incluir' la funcionalidad de otro Caso de Uso o puede 'extender' otro Caso de Uso con su propio comportamiento (20).

Los casos de uso típicamente se relacionan con 'actores'. Un actor es un humano o una máquina que interactúa con el sistema para realizar un trabajo significativo (20).

Un Caso de Uso es una representación de una unidad discreta de trabajo realizada por un usuario (u otro sistema) usando el sistema en operación. Se ejecuta en su totalidad o no se ejecuta nada, devolviendo algo de valor al usuario. Algunos ejemplos de casos de uso son:

Generar Orden de Compra.

Anular Orden de Compra.

Una descripción de Caso de Uso generalmente incluirá:

- Comentarios generales y notas que describen el Caso de Uso;
- Requisitos: cosas que el Caso de Uso debe permitir hacer al usuario, tales como <capacidad de actualizar orden>, <capacidad de modificar orden>, etc. (19).
- Restricciones: las reglas sobre qué se puede hacer y qué no se puede. Incluyen precondiciones que tienen que ser verdaderas antes de que se ejecute el Caso de Uso (por ejemplo <crear orden> debe preceder a <modificar orden>); también incluyen Postcondiciones que tienen que ser verdaderas una vez que el Caso de Uso se ejecutó (19).
- Actores: Un actor es un usuario del sistema. Esto incluye usuarios humanos y otros sistemas computacionales. Un actor usa un Caso de Uso para ejecutar una porción de trabajo de valor para el negocio. El conjunto de casos de uso al que un actor tiene acceso define su rol en el sistema y el alcance de su acción (19).

# CAPÍTULO 6

## 6. IMPLMETACIÓN

### 6.1. Revisión, Inspección y Reutilización

La meta real de la ingeniería de software es crear el código correcto (es decir justo el apropiado para los requisitos), pero los compiladores sólo pueden verificar la sintaxis y generar un código objeto. Lo correcto es responsabilidad humana. Por lo tanto es esencial que los ingenieros de desarrollo estén convencidos de la exactitud del código antes de someterlo a compilación. Aunque en principio es posible compilar primero y verificar que sea exacto o correcto después, esto es tan poco efectivo como corregir la sintaxis de una carta antes de estar seguro que expresa el pensamiento adecuado más aun los programadores tienden a omitir la verificación de un programa una vez que compila sin errores (de sintaxis) (21).

Para implementar adecuadamente el código, nuestro equipo de trabajó realizó lo siguiente:

- Planear la estructura y el diseño (Diseño Detallado).
- Autoinspeccionar el diseño y/o estructura.
- Teclear el código.
- Autoinspeccionar el código: Asegurar que el código realice el trabajo requerido.
- Compilar el Código.
- Probar el código (Pruebas Unitarias).

En nuestro proyecto reutilizamos código, esto nos ayudó a disminuir redundancia de código y el tiempo de programación:

Rehusar código permite producir programas más pequeños y realizar correcciones rápidamente. La consulta personalizada que implementamos es un claro ejemplo de reutilización puesto que fue utilizada en los diferentes módulos.

Los resultados de las inspecciones y el código reutilizado se muestran en la sección 8.

## **6.2. Planificación de la Implementación**

El equipo de trabajo 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. Una búsqueda de código confiable en Internet casi siempre es una inversión útil.

La implementación de los sistemas se la planificó de acuerdo a las necesidades del usuario y a la secuencia lógica de integración. A continuación se muestra el orden en el cual fueron implementados los módulos:

- Módulo de Ordenes de Compra.
- Módulo de Facturación.
- Módulo Nómina.

La planificación de la implementación nos permitió controlar los tiempos de desarrollo para cada uno de los componentes.

### **6.3. Estándares de Programación**

El uso de estándares mejora la disciplina, la legibilidad, y la portabilidad de un programa, es por ello que nuestro equipo de trabajo desarrolló un estándar de programación con nombres para las variables, clases, entre otros. Los ingenieros tienden a ser emocionales en cuanto a sus convenciones favoritas y muchas veces el consenso es imposible. De todos modos las

convenciones son necesarias. Debe asignarse un tiempo limitado para decidir las convenciones y otro tiempo para cumplirlas (21).

El estándar de programación empleado fue desarrollado por el Administrador de Desarrollo, posteriormente fue aprobado por el líder del equipo y finalmente éste fue difundido a todos los miembros del equipo. El estándar de desarrollo está disponible en el **Apéndice H. Estándar de Programación.**

#### **6.4. Manejo y presentación de errores.**

Los desarrolladores se enfrentan de manera constante a la manipulación de datos potencialmente ilegales. Un ejemplo de datos ilegales es un número de cuenta que no corresponde a una cuenta del banco real. Aunque se intente hacer la manipulación más sencilla posible, el mundo real no es sencillo. Una gran parte de la programación se dirige al manejo de errores. Para llevar un control de los errores seguimos los siguientes pasos:

- Realizar revisiones de código.
- Registrar los errores en la plantilla Log D.
- Ingresar parámetros que puedan generar error.

Los errores reportados por el usuario se han clasificado de la



siguiente forma:

- **Validación:** Alguna validación que el usuario ha hecho notar con el uso del sistema.
- **Actualización de programas:** Cuando los programas no han sido actualizados correctamente, es decir hay inconsistencia en los programas de back end con los del front end.
- **Programación:** Los mas frecuentes errores del programador.
- **Datos:** Cuando hay inconsistencia de datos en las diferentes opciones del sistema.
- **Observaciones:** Consideraciones adicionales del usuario.
- **Integración:** Errores presentados en la integración de los diversos módulos.
- **Configuración de los equipos:** Errores debido a que los equipos de los usuarios no tienen la configuración regional correcta o la instalación del algún software necesario para el correcto funcionamiento del sistema (Ejemplo: Office).

La presentación de errores se la realizó mediante mensajes en la

pantalla. Estos errores fueron clasificados en las siguientes categorías:

- **Informativos.-** Para informar al usuario del error o campos que debe considerar para realizar una determinada acción.
- **Advertencia.-** Aparecen para advertir el ingreso de campos necesarios y prevenir las inconsistencias en el formato de los datos de ingreso.
- **Erróneas.-** cuando existen errores controlados en la base de datos y errores fatales del sistema.

Las unidades en las cuales se encontraron la mayor cantidad de errores fueron las siguientes:

- Rol de pagos.
- Ingreso de empleados (Datos Laborales).
- Generar factura.
- Consulta retenciones.
- Generar Orden de Compra.
- Imprimir Orden de Compra.

## 6.5. Calidad en la Implementación

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

Las siguientes preguntas nos pueden ayudar a la revisión del código, específicamente las clases, y fueron formuladas por nuestro equipo de trabajo al momento de escribir código (21):

Es apropiado su nombre?

Su título describe su propósito?

Se aplicaron los estándares de documentación?

Una de las métricas usada por el equipo de trabajo es Cuenta Líneas, es decir contar las líneas de código, que tienen los sistemas desarrollados.

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 (22).

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 (21).

El equipo de trabajo implementó la métrica cuenta líneas considerando las líneas de comentarios y líneas en blanco, el resultado obtenido se lo puede observar en el capítulo de métricas en la medida utilizada por el Administrador de Desarrollo.

# CAPÍTULO 7

## 7. PRUEBAS DE UNIDADES

Las pruebas de unidades consisten en probar las partes estructurales de una aplicación en desarrollo (23).

Estas pruebas nos ayudan a detectar y reparar defectos en transcurso del desarrollo del software. Además nos permiten obtener un producto confiable y con una tasa de errores mínimo (24).

El propósito de probar es encontrar el mayor número de defectos con el más alto nivel de severidad posible (25).

El objetivo de las pruebas no es demostrar que una aplicación es satisfactoria, sino determinar con firmeza en qué parte no lo es (25).

Las pruebas ejecutadas por el equipo estuvieron de acuerdo al plan de pruebas, el cual especifica las unidades a probar, las fechas en las que se realizarían las pruebas, los responsables y los resultados esperados.

Elaboramos un plan para pruebas unitarias, de regresión y aceptación.

## 7.1. Diseño de Pruebas

Las pruebas que se realizaron en el proyecto son las siguientes:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas de regresión.
- Pruebas de aceptación.

La técnica de pruebas empleada para efectuar las pruebas fue la de “caja negra”. La técnica de caja negra o funcional consiste en realizar pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar.

## 7.2. Casos de Prueba

Cuando un ingeniero desarrolla un código, se forma una visión de lo que debe de hacer ese código y, al mismo tiempo, desarrolla circunstancias típicas en las que debe de ejecutarse el código. De manera consciente o no, éstas constituyen los casos de prueba del desarrollador. Así, cuando un individuo prueba su propio código, tiende a ocultar justo eso que debe descubrir (23).

Para describir los casos de pruebas partimos de los casos de usos y sus posibles escenarios, con ellos se establecieron los datos y procedimos a ejecutar la prueba registrando los resultados obtenidos.

### **7.3. Resultado de las Pruebas**

Los datos obtenidos del resultados de las pruebas se usan para evaluar el estado de la aplicación y pronosticar la calidad que tendrá el producto. Además, los datos se convierten en parte del registro histórico de la organización (26).

Más de la mitad del tiempo dedicado a los proyectos se emplea para probar el software. La recompensa de encontrar un defecto pronto en el proceso es al menos un ahorro de diez veces comparado con detectarlo en la etapa de integración o, peor aún, después de la entrega (25).

#### **7.3.1. Registro de Pruebas**

En el registro de pruebas los miembros del equipo que participaron en esta fase registraron el caso de uso, fecha de realización, el nombre del escenario, las condiciones que necesita la unidad en prueba, los datos de entrada y

descripción en pasos de lo que se debe realizar.

Una vez ejecutadas las pruebas los miembros del equipo identificaron y registraron el resultado obtenido con sus observaciones.

A continuación el esquema utilizado por los miembros del equipo en el registro de pruebas:

**CASO DE USO:** < *NOMBRE DE CASO DE USO* >

**RESPONSABLES:** <*NOMBRE DEL PERSONAL QUE EJECUTO LAS PRUEBAS*>

**FECHA DE REALIZACION:** <>

**IDENTIFICACION Y NOMBRE DEL ESCENARIO:** <>

|                                       |  |
|---------------------------------------|--|
| <b>IDENTIFICACION DE CASO DE USO:</b> |  |
| <b>PRECONDICIONES:</b>                |  |
| <b>DATOS DE ENTRADA:</b>              |  |
| <b>DESCRIPCION EN PASOS:</b>          |  |
| <b>RESULTADO ESPERADO:</b>            |  |
| <b>RESULTADO OBTENIDO:</b>            |  |
| <b>OBSERVACIONES:</b>                 |  |

Figura 7.1 Plantilla Plan de Pruebas



### **7.3.2. Informe de incidentes de Pruebas**

En el informe de incidentes de pruebas mostramos la comparación de los resultados obtenidos en las pruebas, es decir: Pruebas exitosas y no exitosas.

Los informes de incidentes de pruebas están enfocados a las pruebas unitarias, de integración, de regresión y de aceptación.

### **7.3.3. Resumen de Pruebas**

El resumen consiste en la recopilación de los datos obtenidos de las pruebas y la verificación del estado en el que se encuentra el producto.

Este resumen de pruebas es entregado a los usuarios para que tengan los registros de las unidades que están utilizando.

# CAPÍTULO 8

## 8. METRICAS

### 8.1. Métricas: ¿Que son y para que sirven?

Las métricas son escalas de unidades sobre las cuales puede medirse un atributo cuantificable. Cuando se habla de software nos referimos a la disciplina de recoger y analizar datos basándonos en mediciones reales de software, así como a las escalas de medición (27).

Una métrica de software es cualquier tipo de medida relacionada con un sistema, proceso o documentación (27).

Para realizar el trabajo de medición se realizó el siguiente proceso:

- Selección de medidas a realizar
- Selección de los componentes a medir
- Medir los componentes
- Identificar las mediciones críticas
- Análisis los valores de las métricas

Las métricas a recabar dependen de los objetivos del proyecto en particular. Los desarrolladores de Software tienen a la vez objetivos comunes como, respetar el presupuesto y respetar los plazos, minimizar las tasas de defectos antes y después de la entrega del producto e intentar mejorar la calidad y la productividad (27).

## 8.2. Métricas utilizadas en el proceso de desarrollo de software y su justificación

Las métricas utilizadas son clasificadas por cada administrador según el rol que desempeñan, a continuación se muestra la tabla con las métricas utilizadas.

| Responsable                           | Métrica  |
|---------------------------------------|--|
| Arturo Aguilera Líder de Equipo       | <ul style="list-style-type: none"> <li>• Tasa de la contribución total por cada Rol</li> <li>• Densidad de defectos detectados en las pruebas de aceptación.</li> </ul>                                      |
| Gina Ruiz<br>Administrador de Calidad | <ul style="list-style-type: none"> <li>• Tasa de errores por página en los Requerimientos C</li> <li>• Tasa de inyección y detección de defectos por Módulo.</li> <li>• Satisfacción del usuario.</li> </ul> |

|   |   |
|---|---|
| <p>Mauricio Echeverría<br/>Administrador de<br/>Desarrollo</p>  | <ul style="list-style-type: none"> <li>• Soporte a todos los miembros del equipo</li> <li>• Componentes reutilizados en los diferentes documentos</li> </ul>  |
| <p>Denisse Echeverría<br/>Administrador de<br/>Planeación</p>   | <ul style="list-style-type: none"> <li>• Horas de equipo trabajadas por incremento.</li> <li>• Horas planificadas por Rol Vs Horas trabajadas.</li> </ul>   |
| <p>José Luis Asencio<br/>Administrador de<br/>Configuración</p> | <ul style="list-style-type: none"> <li>• Número de cambios de requerimientos por Módulo.</li> <li>• Número de versiones de elementos de configuración.</li> <li>• Eficiencia en realizar los cambios</li> </ul> |

Tabla 8.1 Métricas por Rol

Las razones que justifican las métricas escogidas son las siguientes:

- Evaluar la productividad del equipo de desarrollo
- Evaluar la calidad del producto.
- Establecer una línea base de estimación.

Las métricas son utilizadas para el control de los proyectos, no son ni estándares, ni universales. Cada proyecto debe seleccionar sus propias métricas (27).

### **8.3. Valores de las Métricas obtenidas durante cada etapa del desarrollo del software**

#### **Etapa de Definición**

En esta etapa no se midió ninguna métrica

#### **Etapa de Desarrollo**

##### **1. Tasa de errores por página en los documentos :**

En la etapa de desarrollo se efectuó la revisión e inspección de requerimientos C (Cliente) para todos los módulos.

La revisión de los defectos en los documentos se los clasificó de la siguiente manera:

Sintaxis, Ortografía, Versiones, Fechas, Nombres de los Documentos, Formato. Debido a que estos errores fueron los más frecuentes en todos los documentos.

La plantilla utilizada para la inspección de la documentación se muestra en la Figura 10.1.

Luego de efectuar las inspecciones en los documentos se muestra los resultados obtenidos en la documentación de los Requerimientos C (cliente) de todos los módulos:

| Descripción del Defecto   | MCIB      | MCP       | MFAC      | MN        | MOC       | MOT       | MPO       | Total      |
|---------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| Fechas                    | 3         | 1         | 1         | 2         | 2         | 3         | 2         | 14         |
| Formato                   | 5         | 1         | 2         | 3         | 3         | 2         | 1         | 17         |
| Nombres de los Documentos | 6         | 1         | 0         | 2         | 1         | 3         | 2         | 15         |
| Ortografía                | 22        | 4         | 17        | 22        | 15        | 8         | 4         | 92         |
| Sintaxis                  | 18        | 1         | 4         | 18        | 8         | 7         | 5         | 61         |
| Versiones                 | 5         | 2         | 0         | 5         | 2         | 3         | 1         | 18         |
| <b>Total</b>              | <b>59</b> | <b>10</b> | <b>24</b> | <b>52</b> | <b>31</b> | <b>26</b> | <b>15</b> | <b>217</b> |
| Número de Páginas         | 27        | 8         | 16        | 22        | 15        | 13        | 17        | 118        |
| Tasa errores por página   | 2.19      | 1.25      | 1.50      | 2.36      | 2.07      | 2.00      | 0.88      | 1.84       |

Tabla 8.2 Errores por Página en Req C.

En promedio el equipo de trabajo obtuvo 1,84 errores por página en los requerimientos del cliente. Como se puede observar en la tabla existe mayor número de defectos en los módulos de MN y MCIB, esto se debe principalmente a que son los documentos que mayor número de requerimientos y páginas contienen, adicionalmente fueron los dos módulos desarrollados en el primer incremento. Se puede notar la disminución de la tasa de errores en la documentación posterior a los Módulos de MN y MCIB.

## 2. Tasa de inyección y detección de defectos por módulo.

Para evaluar la tasa de inyección y detección de defectos en cada módulo, todos los miembros del equipo usaron la plantilla LOGD que se muestra en la Figura 1.3.

Como se puede observar en la siguiente tabla el mayor número de errores, se encontró en el código de las aplicaciones.

| Módulo       | code       | compile  | design    | planning  | test      | Total      |
|--------------|------------|----------|-----------|-----------|-----------|------------|
| MCIB         | 74         | 0        | 8         | 28        | 21        | 131        |
| MN           | 65         | 3        | 15        | 23        | 18        | 124        |
| MOT          | 15         | 0        | 1         | 10        | 5         | 31         |
| MOC          | 23         | 0        | 5         | 9         | 7         | 44         |
| MPO          | 18         | 1        | 8         | 13        | 4         | 44         |
| MFAC         | 18         | 0        | 4         | 4         | 3         | 29         |
| MCP          | 3          | 0        | 2         | 4         | 2         | 11         |
| <b>Total</b> | <b>216</b> | <b>4</b> | <b>43</b> | <b>91</b> | <b>60</b> | <b>414</b> |

Tabla 8.3 Defectos Detectados por Módulo

Esto se debió en primera instancia a la falta de experiencia en la herramienta de desarrollo utilizada por la mayoría de integrantes del equipo de trabajo, además al inicio del proyecto fue difícil educarnos en el nuevo método de desarrollo debido a que no se efectuaban en ningún momento revisiones del código como lo indica la metodología por lo tanto el mayor número de errores fue encontrado en el código.

Adicionalmente como se puede observar el mayor número de defectos corresponden a los Módulos de Nómina y de Bodega, como se mencionó anteriormente son los dos módulos que tienen el mayor número de líneas de código y fueron los primeros que se empezaron a desarrollar. Se puede observar que a medida que se

desarrollaban los diferentes módulos el número de errores fue disminuyendo.

## **Etapas de Producción**

### **1. Satisfacción del usuario, Pruebas de Usabilidad.**

Las pruebas de usabilidad son una medida concreta y objetiva de la usabilidad de una herramienta o sistema tomada a partir de usuarios verdaderos con tareas reales. Este término es frecuentemente usado para referirnos a las técnicas y métodos utilizados para evaluar un producto o sistema (27).

El equipo de trabajo luego de haber efectuado las pruebas de aceptación, ejecutó las pruebas de usabilidad las mismas que fueron realizadas por todos los usuarios de todos los módulos con el objetivo evaluar la usabilidad en nuestros sistemas, para ejecutar las pruebas de usabilidad se implementó la plantilla que se observa en la Figura 10.2.

Las pruebas de usabilidad se efectuaron a los usuarios luego de haber puesto en producción los sistemas, como se puede observar en la Figura 10.2 las preguntas fueron calificadas con



el valor de cero (0) para el mínimo y nueve (9) para el mayor puntaje obtenido. Para evaluar los resultados se calculó el promedio obtenido en cada pregunta efectuada, los resultados obtenidos luego de consolidar la información recolectada es la siguiente:

| MODULO                | Uso del sistema | Interfaz del sistema | Información que maneja el sistema | Aprendizaje en el uso del sistema | Capacidades del sistema | Promedio Total |
|-----------------------|-----------------|----------------------|-----------------------------------|-----------------------------------|-------------------------|----------------|
| MOT                   | 6.50            | 7.75                 | 7.08                              | 6.83                              | 6.63                    | 6.96           |
| MCP                   | 7.43            | 9.00                 | 8.50                              | 9.00                              | 8.00                    | 8.39           |
| MCIB                  | 6.50            | 7.38                 | 7.92                              | 7.83                              | 7.13                    | 7.35           |
| MPO                   | 6.96            | 8.19                 | 8.21                              | 8.42                              | 7.56                    | 7.87           |
| MNO                   | 8.14            | 9.00                 | 8.83                              | 9.00                              | 9.00                    | 8.80           |
| MOC                   | 7.57            | 8.00                 | 7.17                              | 5.67                              | 7.00                    | 7.08           |
| MFAC                  | 7.43            | 8.25                 | 8.67                              | 8.00                              | 8.25                    | 8.12           |
| <b>Promedio Total</b> | <b>7.22</b>     | <b>8.22</b>          | <b>8.05</b>                       | <b>7.82</b>                       | <b>7.65</b>             | <b>7.79</b>    |

Tabla 8.4 Grado de Usabilidad por Módulo

Como se puede observar el promedio total de uso del sistema de todas las aplicaciones en general es de 7.79 sobre 9, consideramos el resultado obtenido satisfactorio ya que los sistemas desarrollados son el primer trabajo efectuado por nuestro equipo de trabajo.

## 2. Tasa de la contribución total de cada rol-peer:

Para medir el promedio de contribución total de cada rol en el desarrollo del proyecto fue necesario evaluar el desempeño de cada administrador en cada incremento, para efectuar la medición se usó la forma PEER, la plantilla se la puede

observar en la Figura 1.5. En donde uno (1) representa el menor valor y cinco (5) el máximo.

A continuación se detallan los resultados obtenidos del desempeño por cada rol luego de la consolidación de las calificaciones de todos los miembros del equipo.

| ROLES                          | 1er incremento<br>MCIB y MN | 2do incremento<br>OT y OC | 3er incremento<br>integración<br>MOC con MCIB | 4to incremento<br>MPO y MFAC | 5to incremento<br>MCP | Total |
|--------------------------------|-----------------------------|---------------------------|---|------------------------------|-----------------------|-------|
| Administrador de Calidad       | 4                           | 4.4                       | 4.2   | 4                            | 4.2                   | 4.16  |
| Administrador de Configuración | 4                           | 4                         | 4.2   | 4.2                          | 4.2                   | 4.12  |
| Administrador de Desarrollo    | 5                           | 4.2                       | 4.8   | 4.8                          | 4.6                   | 4.68  |
| Administrador de Planeación    | 4                           | 4                         | 4.2   | 4.2                          | 4.4                   | 4.16  |
| Contribución Total (100%)      | 4.2                         | 4.2                       | 4.4   | 4.2                          | 4.4                   | 4.28  |
| Lider del Equipo               | 4.2                         | 4.4                       | 5   | 4.6                          | 3.8                   | 4.4   |
| Total                          | 4.23                        | 4.20                      | 4.47  | 4.33                         | 4.27                  | 4.30  |

Tabla 8.5 Contribución Total por Cada Rol

En el promedio de contribución de cada rol podemos observar que todos han cumplido con el objetivo propuesto por los miembros del equipo cuya contribución por rol debía ser mayor a 3 como se indica el **Apéndice A. Objetivos del Equipo**.

Como se puede observar, el Administrador de desarrollo fue quien obtuvo el mayor promedio de contribución durante la ejecución del proyecto. Esto se debe a su experiencia y conocimiento en el uso de las herramientas de desarrollo empleadas.

### **3. Densidad de defectos detectados en las pruebas de aceptación.**

Luego de realizar las pruebas de aceptación con los usuarios de todos los módulos implementados, se encontraron algunas observaciones y errores principalmente cuando se efectuaron consultas, las observaciones se refirieron principalmente a la visualización de los datos en pantalla y a ciertas validaciones que no constituyen errores de importancia al momento de ejecutar un proceso en el sistema, los errores de consulta encontrados se debieron principalmente a inconsistencias en los datos debido a que los sistemas no estaban usándose al 100%, cuando se pusieron las primeras versiones en producción.

Para efectuar las pruebas de aceptación se utilizó la Figura 7.1, los datos tomados en cuenta para el análisis fueron el número de observaciones encontradas en el plan de pruebas ejecutado. La siguiente tabla muestra la densidad de observaciones encontradas en cada uno de los módulos:

| Modulo       | Total     |
|--------------|-----------|
| MCIB         | 7         |
| MCP          | 0         |
| MFAC         | 5         |
| MNO          | 2         |
| MOC          | 4         |
| MOT          | 2         |
| MPO          | 6         |
| <b>Total</b> | <b>26</b> |

Tabla 8.6 Defectos en Pruebas de Aceptación

Como se puede observar en la tabla anterior los módulos que mayor observaciones tuvieron fueron: MCIB, MPO y MFAC, esto se debió a la complejidad y tamaño que presentan cada uno de estos módulos, en contraste se puede notar que el módulo de MCP no surgieron observaciones por parte de lo usuarios, lo que demuestra el mejoramiento en la implementación del último módulo realizado por el equipo de trabajo.

# CAPÍTULO 9

## 9. ROL DE LÍDER DEL EQUIPO Y ADMINISTRADOR DE CALIDAD

El líder del Equipo dirige el equipo y asegura que los ingenieros reporten los datos del proceso y completen su trabajo de manera planeado a tiempo.

El administrador de calidad apoya al equipo en la definición de las necesidades del proceso, y da seguimiento en el proceso y calidad del software.

### 9.1. Papel que Desempeña

El papel del Líder del Equipo es uno de los más importantes dentro de TSP puesto que debe llevar a cabo las siguientes tareas:

- Construir y mantener un equipo eficaz.
- Motivar a los integrantes del equipo a comprometerse en el trabajo del proyecto.
- Resolver todos los conflictos entre los integrantes del equipo.

- Desarrollar eficazmente como mediador en las reuniones del equipo.

El papel que desempeña el Administrador de calidad es vital para asegurar que el las aplicaciones desarrolladas cumplen con los niveles mínimos de calidad llevando a cabo las siguientes tareas:

- Asegurar que todos los miembros del equipo lleven a cabo correctamente la metodología TSP.
- Realizar inspecciones a los documentos entregados y al código fuente de tal manera que se asegure que cumplen con los estándares de documentación y desarrollo.
- Alertar al líder del equipo de los problemas de calidad.
- Mantener los estándares de codificación desarrollado por el administrador de desarrollo.

## **9.2. Plantillas, Modelos, Metodologías y Estándares**

La plantilla usada por el líder del equipo es la siguiente:

PEER: Esta forma permite al líder llevar a cabo las evaluaciones de desempeño a todos los miembros del equipo. Ver Modelo de Plantilla en la Figura 1.5. Esta plantilla permitió conocer las

falencias en el desempeño de los miembros del equipo.

El modelo para desempeñar el Rol del líder dentro del equipo fue el siguiente:

- Dar seguimiento de todas las tareas efectuadas por los integrantes del equipo.
- Verificar el estado del proyecto en cuanto a riesgos y otros asuntos.
- Mantener la comunicación con el Director del proyecto y dar a conocer las inquietudes y dudas del grupo en cuanto al desarrollo del proyecto.
- Desempeñar adicionalmente el papel de ingeniero de desarrollo para la efectiva implementación de los módulos de Órdenes de Compra, Facturación y Nómina.

La plantilla usada por el Administrador de Calidad para la verificación de documentos fue la siguiente:

|   |                                |                     |
|---|--------------------------------|---------------------|
| <b>Nombre Documento:</b>                |                                | <b>Fecha:</b>       |
| <b>Equipo:</b>                          |                                | <b>Instructor:</b>  |
| <b>Parte/Nivel:</b>                     |                                | <b>Módulo:</b>      |
| <b>Moderador:</b>                       |                                | <b>Propietario:</b> |
| <b>Datos de Defectos:</b>               |                                |                     |
| <b>No.</b>                              | <b>Descripción del Defecto</b> | <b># de Veces</b>   |
| 1                                       | Sintaxis                       |                     |
| 2                                       | Ortografía.                    |                     |
| 3                                       | Versiones                      |                     |
| 4                                       | Fechas                         |                     |
| 5                                       | Nombres de los Documentos.     |                     |
| 6                                       | Formato                        |                     |
| <b>Total Errores</b>                    |                                |                     |
| <b>Número de Páginas del Documento:</b> |                                |                     |
| <b>Tasa de errores por página</b>       |                                |                     |

Figura 9.21 Plantilla Inspección Documentación.

Esta plantilla fue diseñada con el objetivo de registrar los errores presentados en el desarrollo de la documentación.

Adicionalmente se efectuó el seguimiento a los defectos detectados utilizando la plantilla LOGD. Ver el diseño de plantilla en la Figura 1.3.

Se efectuó el análisis de las pruebas de usabilidad realizadas a los usuarios en cada uno de los módulos, para efectuar las pruebas de usabilidad a todos los usuarios se ejecutó el siguiente cuestionario de preguntas.







|  |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |        |
|--|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--------|
| <b>Recordar nombres y uso de comandos es</b> | Difícil  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Fácil  |
| <b>Mensajes de ayuda en la pantalla</b>      | Inútiles |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Útiles |

**Capacidades del Sistema**

| <b>Opiniones</b>                                   |           | <b>0</b> | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> |  |                 |
|--|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|--|-----------------|
| <b>Velocidad del sistema</b>                       | Muy lento |          |          |          |          |          |          |          |          |          |          |  | Bastante rápido |
| <b>Confiabilidad del sistema</b>                   | No fiable |          |          |          |          |          |          |          |          |          |          |  | Confiable       |
| <b>El sistema tiende a ser</b>                     | Aburrido  |          |          |          |          |          |          |          |          |          |          |  | Atractivo       |
| <b>Diseñado para todos los niveles de usuarios</b> | Nunca     |          |          |          |          |          |          |          |          |          |          |  | Siempre.        |

**En las preguntas del 1 al 5, exprese su opinión acerca del Sistema. Marque de peor (0) a mejor (9) según su criterio.**

| <b>Opiniones</b> |   | <b>0</b> | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> |
|------------------|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>1</b>         | Aprendizaje en el uso del sistema         |          |          |          |          |          |          |          |          |          |          |
| <b>2</b>         | Eficiencia                                |          |          |          |          |          |          |          |          |          |          |
| <b>3</b>         | Se requiere memorizar acciones anteriores |          |          |          |          |          |          |          |          |          |          |
| <b>4</b>         | Exactitud de los mensajes de error        |          |          |          |          |          |          |          |          |          |          |
| <b>5</b>         | Satisfacción del Usuario.                 |          |          |          |          |          |          |          |          |          |          |

Figura 9.2.2 Plantilla Usabilidad (28)

### 9.3. Factores y Métricas de los Modelos

El factor de medición del líder del equipo que fue tomado en cuenta para evaluar el trabajo efectuado por el equipo de trabajo en todas las etapas del desarrollo del proyecto fue principalmente:

Medir la contribución de cada administrador en la terminación exitosa del producto final y en términos generales la efectividad completa del equipo de trabajo, para lo cual se utilizó la Platilla PEER Figura 1.5.

Evaluar la Densidad de defectos y observaciones encontradas en las pruebas de aceptación, para asegurar que no existan defectos y errores en las aplicaciones implementadas.

Para asegurar la calidad del desarrollo de software por todos los miembros del equipo, se tomaron en cuenta los factores que pueden causar debilidades en la calidad como el desarrollo de la documentación, los defectos en el desarrollo del software y la usabilidad de los sistemas. El modelo a seguir para desarrollar el rol de administrador de calidad fue el siguiente:

- Revisión e Inspección de los Requerimientos C.
- Seguimiento a los defectos inyectados en el desarrollo de

los sistemas.

- Medir los niveles de usabilidad de los sistemas.

## 9.4. Experiencias y Responsabilidades

La experiencia obtenida desempeñando el rol de líder del equipo fue principalmente el aprender a controlar a un grupo de personas con diferentes personalidades y habilidades, y actuar como mediador en las discusiones presentadas, ser líder del equipo requiere de mucha paciencia y sobre todo tener una mente amplia para saber escuchar a todos los miembros del equipo.

La responsabilidad que se debe afrontar como líder de un equipo es la siguiente:

- Motivar a los integrantes a desarrollar sus tareas.
- Identificar las tareas a realizar y quienes serán los responsables.
- Ayudar al equipo en la asignación de tareas y la resolución de otros asuntos.
- Actuar como mediador y estar pendiente del tiempo en las reuniones.

Una de las experiencias más importantes adquiridas como administrador de calidad es el hecho de poder determinar si un producto desarrollado tiene un nivel mínimo de calidad de acuerdo a los estándares definidos por el equipo de trabajo.

Al desempeñar el rol de administrador de calidad se adquiere habilidades, destrezas y experiencia en la inspección de código y de documentación de tal manera que los defectos encontrados sean reducidos en el desarrollo de nuevas versiones tanto de software como de documentación.

La responsabilidad del Administrador de Calidad es muy importante dentro del desarrollo de un producto ya que se debe de alguna manera tratar que el software desarrollado esté libre de errores y cumpla con los niveles de calidad mínimos de acuerdo a los estándares desarrollados por los miembros del equipo.

## **9.5. Análisis General del Rol del Líder**

El rol del líder del equipo es uno de los más importantes ya que es vital para mantener el orden y de alguna manera motivar a todos los miembros a desarrollar sus tareas.

Al inicio del desarrollo del proyecto surgieron muchos inconvenientes debido a la falta de comunicación con el resto del equipo, debido a que todos los integrantes del equipo empezamos a conocernos sobre la marcha del proyecto por lo tanto el nivel de comunicación al inicio fue muy bajo. En el transcurso del proyecto la comunicación mejoró notablemente por lo que los problemas e inconvenientes presentados fueron más fáciles de resolver, las reuniones de trabajo fueron más frecuentes por lo que el nivel de comunicación con todos los administradores mejoró.

Las tareas del líder no son fáciles de llevar a cabo debido a que se debe tomar en cuenta la opinión de todos los miembros del equipo y se debe ser imparcial en las decisiones a tomar.

El Rol del Líder del equipo en el desarrollo del proyecto es un rol que se debe tratar de alguna manera segregado de las demás funciones debido a que el tiempo que debe dedicarle el líder al proyecto es mayor, ya que debe mantener la comunicación con todos los miembros del equipo y adicionalmente se debe cumplir las funciones de ingeniero de desarrollo.

El Rol de Administrador de calidad cumple con un papel muy importante dentro del desarrollo de un proyecto, al inicio de proyecto se definieron los estándares de programación y de documentación que lograron agilizar y mantener el mismo estándar para los productos a entregar.

En el primer incremento surgieron problemas al realizar las revisiones de documentación y de código fuente, esto fue principalmente a la falta de experiencia para efectuar revisiones e inspecciones, la falta de experiencia de los miembros del equipo en seguir una nueva metodología y seguimiento de estándares.

Al desempeñar el Rol de Administrador de calidad al igual que el Rol del Líder del equipo existen tareas que deben estar segregadas por ejemplo: las inspecciones realizadas al código fuente y a la documentación generada por el administrador de calidad deben necesariamente ser inspeccionadas por otra persona ya que es difícil realizar una inspección a los productos que la misma persona ha desarrollado.



# CAPÍTULO 10

## 10. COMENTARIOS FINALES Y CONCLUSIONES

### 10.1. Glosario de Términos

TSP (Team Software Process): Proceso de Desarrollo de Software en Equipo.

PSP (Personal Software Process): Proceso de desarrollo Personal de Software

Workbreakdown: Conocida como estructura de trabajo detallada o estructura de descomposición o desglose del trabajo. Es la base de toda la planeación, definiendo los entregables del proyecto. Los entregables pueden ser etapas o procesos del proyecto (plan del proyecto, documentación de diseño, etc.) o partes del producto final (pantallas, documentación, ventanas, etc.) (29).

Apostadores: Individuos u organizaciones que están involucrados activamente en el proyecto o cuyos intereses se pueden ver afectados positiva o negativamente por el resultado de la ejecución del proyecto o la terminación del mismo.

Hitos: Son puntos identificables en el proyecto que denotan el requerimiento de un reporte o la terminación de una actividad importante (29).

## **10.2. Análisis Comparativo: Antes y Después de TSP**

Antes de adquirir conocimiento acerca de la metodología TSP ningún miembro del equipo de trabajo tenía como procedimiento utilizar estándares tanto de documentación como desarrollo, no se efectuaban planificaciones por lo tanto no se distribuían las tareas de forma efectiva cuando se trabajaba en grupo, rara vez se documentaba el software y algo muy importante el análisis y diseño de las aplicaciones desarrolladas era muy pobre. No se definían fechas de entrega ni se respetaba un cronograma de trabajo, no se tenía como costumbre efectuar pruebas cuando se efectuaban desarrollos, ni se realizaban inspecciones en la documentación, no se controlaba los cambios efectuados en el sistema, en fin no se conocía en realidad lo que era trabajar en equipo.

Después de educarnos sobre la metodología TSP se encontró una

mejor organización en el desarrollo de la documentación, se obtuvo una disciplina en el desarrollo del trabajo, los procesos son mucho más controlados.

Después se aprendió a estimar y planificar los tiempos invertidos en las actividades por cada administrador.

Desarrollar el proyecto de manera más efectiva, efectuando un buen análisis y diseño de los sistemas.

En el desarrollo del proyecto se presentaron diferentes problemas debido a que el grupo de trabajo no tenía antecedentes de proyectos desarrollados con este tipo de metodología. No se podían estimar los tiempos ya que no se contaba con la experiencia para poder estimar fechas de entrega, la utilización de métricas en el proceso de desarrollo de software ha permitido tomar acciones correctivas a tiempo, mejorando sucesivamente los procesos definidos.

Luego del proceso de aprendizaje, adaptación y puesta en práctica el uso de la metodología se puede planificar el trabajo y estimar los tiempos para llevar a cabo el proyecto. Como se llevará un

registro de los errores y tiempos de desarrollo se puede descubrir las habilidades, debilidades y fortalezas al momento de programar.

### **10.3. Conclusiones y Recomendaciones.**

#### **Conclusiones:**

1. El TSP funciona de una mejor manera, siempre y cuando los miembros del equipo tengan un mismo lugar de trabajo. Esto se debe a que se puede llevar un mejor control de las tareas de cada uno de los administradores, se tiene un mejor control de los cambios en los programas y de las versiones, entre otras cosas. La comunicación entre los miembros del equipo es un elemento fundamental en la metodología TSP, lo cual se complica cuando los administradores tienen diferentes lugares de trabajo.
2. La metodología TSP está enfocada a administradores del proyecto, para que exista un mejor seguimiento del trabajo efectuado a los desarrolladores, estas funciones deben ser segregadas. Por ejemplo: Administrador de Calidad, no debería efectuar las pruebas e inspecciones a su propio código desarrollado, para que los resultados obtenidos sean

imparciales.

3. La metodología TSP sin duda ayuda a llevar un mejor control de las tareas, tiempos y recursos, pero consideramos que es aplicable para empresas que tengan mayores recursos humanos y logísticos.
4. El registro en las plantillas tanto de PSP como TSP que se utilizaron nos ayudaron a ser más ordenados con el registro en el proceso de desarrollo de software.
5. La fase de especificación de requerimientos del cliente consideramos que es una de las más importantes debido a que la mala interpretación de lo que requiere el cliente o la mala definición de lo que realmente necesita o espera del sistema por parte del usuario, puede acarrear errores en el diseño y en la implementación muy costosos.
6. El modelo incremental es el que más se ajusta a la metodología PSP y TSP ya que se puede definir pequeños incrementos y ver las mejoras a medida que se va desarrollando los mismos.

7. El equipo de trabajo es más productivo cuando trabaja en conjunto con los demás miembros que cuando se trabaja individualmente. Por lo que es indispensable que exista una buena comunicación entre todos los integrantes.
  
8. Los errores que se muestran en el Capítulo 8 fueron corregidos a su debido tiempo. Para mitigar el impacto de los errores encontrados en los sistemas antes de haber sido puestos en producción, y entregados de manera formal al cliente, se pactó un período de garantía con el cliente de seis meses donde se resolverá cualquier error de procesamiento presentado en cualquiera de los sistemas implementados.

**Recomendaciones:**

1. Conocer y aprender correctamente la metodología TSP y PSP antes de aplicarla.
  
2. Para ayudar a que los futuros ingenieros de la ESPOL se eduquen en un mejor proceso de desarrollo de software se puede aplicar el PSP y TSP en las materias de Software I y Software II, donde los estudiantes de

software II se formarán en grupos de administradores y software I aplicaran PSP, de tal manera que las funciones de los administradores y desarrolladores estén segregadas.

3. Educar a los alumnos a la reutilización de software, crear una base de conocimientos para que ayuden a los nuevos ingenieros de software a reutilizar código de tal manera que en los nuevos proyectos, se implemente únicamente las funciones o requerimientos que se necesiten. La base de conocimiento puede almacenar información como: Programas, objetos, clases, funciones, documentación, entre otros. Desarrollados por alumnos o profesionales politécnicos.
4. El tiempo en el levantamiento de requerimientos, análisis y diseño en el desarrollo de un proyecto no se debe escatimar, así como también la formalización de requerimientos por parte de los usuarios antes de empezar el desarrollo del software.
5. Para implementar y aplicar la metodología TSP y PSP

los miembros del equipo deben disponer de tiempo, ya que el éxito de las metodologías antes mencionadas está en documentar todo lo que lo que los ingenieros de software realicen ya sea como administradores o como desarrolladores.

6. Luego de nuestra experiencia en la aplicación de las metodologías TSP y PSP podemos aconsejar o proponer que a los futuros estudiantes de los primeros niveles se debería introducir y educar en las diferentes metodologías de desarrollo de software.
7. Para las personas que se inclinen por adoptar PSP y TSP como metodología de desarrollo de software les recomendamos trabajar la mayor parte del proyecto en equipo en un solo lugar de trabajo, ya que uno de los mayores inconvenientes que tuvo nuestro equipo al inicio del proyecto fue la falta de comunicación entre sus miembros y el trabajo se lo realizaba en forma individual. Los problemas se los fueron resolviendo a medida que el grupo se fue integrando y se tomó como decisión trabajar la mayor parte en un solo lugar de trabajo.



8. Para la recolección de métricas los equipos de trabajo a más de adoptar las plantillas que la metodología PSP Y TSP recomienda, pueden consultar diferentes plantillas ya sea en libros, Internet, etc. O bien pueden diseñarlas como es el caso de la plantilla Figura 7.1 la cual fue diseñada por el equipo de trabajo para registrar los resultados en las diferentes pruebas efectuadas y la plantilla que se observa en la figura 7.2 que fue diseñada con el propósito de documentar las inspecciones en la Documentación con el fin de mejorar la calidad.

# APÉNDICES

# BIBLIOGRAFÍA

1. TEODORIVICK, JOSÉ CARLOS, Centro Federal de Educación Tecnológica de Paraná, Disponible en Internet, [http://pessoal.cefetpr.br/dergint/dergint/daad/artigos/dow\\_2001/Dergint\\_Teod\\_Altec2001\\_Artigo\\_espF.PDF](http://pessoal.cefetpr.br/dergint/dergint/daad/artigos/dow_2001/Dergint_Teod_Altec2001_Artigo_espF.PDF), ingresado Septiembre 2005.
2. HUMPHREY, WATTS S. Universidad de las Américas, Puebla. Tesis Profesional, Disponible en Internet [http://www.pue.udlap.mx/~tesis/lis/pelaez\\_r\\_jj/capitulo2.pdf](http://www.pue.udlap.mx/~tesis/lis/pelaez_r_jj/capitulo2.pdf), ingresado septiembre 2005.
3. BRAUDE, ERICK. Ingeniería de Software, Una perspectiva Orientada a Objetos, Boston University, Alfa Omega, Grupo Editor 2003.
4. MONTESDEOCA CÉSAR, Universidad de las Américas, Puebla. Tesis Profesional, Disponible en Internet [http://www.pue.udlap.mx/~tesis/lis/pelaez\\_r\\_jj/capitulo3.pdf](http://www.pue.udlap.mx/~tesis/lis/pelaez_r_jj/capitulo3.pdf), ingresado septiembre 2005.
5. HUMPHREY, WATTS S. Introduction to the Team Software Process
6. FACULTAD DE CIENCIAS EXACTAS TANDIL ARGENTINA, Ingeniería de Requisitos, Disponible en Internet <http://www.exa.unicen.edu.ar/catedras/ingrequi/Clase%20Introduccion.doc>, ingresado Julio 2004
7. SOMMERVILLE, IAN. Ingeniería de Software. Séptima Edición, Pearson Education
8. SOMMERVILLE, IAN. Pág. 51, Ingeniería de Software, Sexta Edición, Pearson Education, México, 2002

9. MICROSOFT TECHNET, Disciplina de administración de riesgos v.1.1, 2002 Microsoft Corporation. Disponible en Internet [<http://www.microsoft.com/latam/technet/articulos/200304/art02/>]
10. BRAUDE, ERICK. Pág. 86. Ingeniería de Software, Una perspectiva Orientada a Objetos, Boston University., Alfa Omega, Grupo Editor 2003.
11. BERMÚDEZ, ANDRÉS G. PMP. "Gerencia de Riesgos del Proyecto", I Jornada de Gerencia de Proyectos de T.I 2004
12. SOMMERVILLE, IAN. Ingeniería de Software. Séptima Edición, Pearson Education
13. SOMMERVILLE, IAN. Ingeniería de Software. Sexta Edición Capítulo 4, Pearson Education
14. SOMMERVILLE, IAN. Ingeniería de Software. Sexta Edición Capítulo 4, Pearson Education
15. CHOQUE ASPIAZU, GUILLERMO. Profesor de la Universidad Mayor de San Andrés. La Paz Bolivia. Artículo de Divulgación elaborado por la materia Ingeniería de Software Septiembre 2003. Disponible en Internet.  
[http://www.umsanet.edu.bo/docentes/gchoque/Art\\_IngRequerim.pdf](http://www.umsanet.edu.bo/docentes/gchoque/Art_IngRequerim.pdf), ingresado septiembre 2005.
16. BRAUDE, ERICK Ingeniería de Software, Una perspectiva Orientada a Objetos, Boston University.
17. <http://www.monografias.com/trabajos6/resof/resof.shtml#inge>, ingresado septiembre 2005
18. BRAUDE, ERICK Ingeniería de Software, Una perspectiva Orientada a Objetos, Boston University.

19. Information Engineering, Grupo de investigación perteneciente al Departamento de Informática de la Universidad Carlos III de Madrid. Disponible en Internet.  
<http://www.ie.inf.uc3m.es/grupo/docencia/reglada/psii/unidad101112-DOC.pdf>, ingresado Septiembre 2005.
  
20. GEOFFREY, SPARKS. Sparx Systems, Australia. Traducción: Fernando Pinciroli (Solus S.A., Argentina) y Aleksandar Orlic (Craftware Consultores Ltda., Chile). Disponible en Internet.  
[http://www.sparxsystems.cl/WhitePapers/EI\\_Modelo\\_de\\_Casos\\_de\\_Uso.pdf](http://www.sparxsystems.cl/WhitePapers/EI_Modelo_de_Casos_de_Uso.pdf), Ingresado Septiembre 2005
  
21. BRAUDE, ERICK. Ingeniería de Software, Una perspectiva Orientada a Objetos, Boston University.
  
22. Universidad politécnica de Madrid. Proyectos informáticos. Disponible en Internet.  
<http://www.oei.eui.upm.es/Asignaturas/PInformaticos/ficheros/temario/PROYINF-2.pdf>, ingresado Septiembre 2005
  
23. BRAUDE, ERICK. Pág. 395, Ingeniería de Software orientada a objetos, Alfa Omega Grupo Editor 2003
  
24. BRAUDE, ERICK Pág. 393, Ingeniería de Software orientada a objetos, Alfa Omega Grupo Editor 2003
  
25. BRAUDE, ERICK Pág. 394, Ingeniería de Software orientada a objetos, Alfa Omega Grupo Editor 2003
  
26. BRAUDE, ERICK Pág. 407, Ingeniería de Software orientada a objetos, Alfa Omega Grupo Editor 2003
  
27. Departamento de Control de Calidad y Auditoria Informática. Control de Calidad del Software. Disponible en Internet,  
<http://gamenon.uniandes.edu.co:80/sistemas/metricasdelsoftware.pdf>,

28. Control de Calidad en los sistemas. Universidad de las Américas, Puebla. Tesis Profesional, Disponible en Internet  
[http://www.pue.udlap.mx/~tesis/lis/pelaez\\_r\\_jj/capitulo2.pdf](http://www.pue.udlap.mx/~tesis/lis/pelaez_r_jj/capitulo2.pdf), ingresado septiembre 2005.
29. II Jornadas de Gerencia de Proyectos de TI. ACIS. Disponible en Internet:  
<http://www.acis.org.co/memorias/JornadasGerencia/IIJNGP/Alguna%20terminologia%20comun%20en%20el%20ambito%20de%20IGP.pdf>, ingresado, Octubre 2005