

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA

**USO DEL TSP (TEAM SOFTWARE PROCESS) EN EL
DESARROLLO DE SOFTWARE**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**GEOVANNA PATRICIA BUSTOS RECALDE
CRISTHIAN PAÚL GUALLASAMIN CODENA**

DIRECTOR: ING. CARLOS MONTENEGRO ARMAS

Quito, Octubre 2007

DECLARACIÓN

Nosotros, Geovanna Patricia Bustos Recalde y Cristhian Paúl Guallasamín Codena, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Geovanna Bustos

Cristhian Guallasamín

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Geovanna Patricia Bustos Recalde y Cristhian Paúl Guallasamín Codena, bajo mi supervisión.

Ing. Carlos Montenegro Armas

DIRECTOR DE PROYECTO

AGRADECIMIENTO

Agradecemos a Dios, por haber iluminado nuestras mentes y guiado nuestros pasos por el camino verdadero, a todas las personas que se han mantenido junto a nosotros y de manera muy especial a nuestros padres, quienes nos apoyaron y brindaron su aliento para la realización de nuestros más grandes anhelos, con el fin de superarnos y ser cada día mejores.

A nuestros amigos con los cuales aprendimos el verdadero significado de la amistad y compañerismo durante nuestra vida universitaria, especialmente al grupo conformado por los VF++.

Agradecemos al Ingeniero Carlos Montenegro, que con su ayuda y guía supo esparcir en nosotros el conocimiento necesario para la creación de este proyecto.

Agradecemos al amor, comprensión, apoyo y consideración que nos tenemos, fuerzas motivadoras para el término de las metas que juntos nos hemos propuesto.

*Geovanna Bustos
Cristhian Guallasamin*

DEDICATORIA

El presente trabajo lo dedicamos con todo cariño a las lucecitas que iluminan nuestro diario vivir Dana y Steven nuestros amados hijos, los seres más importantes en nuestras vidas.

A nuestros padres y profesores, quienes nos han apoyado y guiado con mucho esfuerzo y abnegación a los largo de estos años de estudio, gracias a lo que hemos podido llegar a un feliz termino de nuestra vida estudiantil universitaria.

*Geovanna Bustos
Cristhian Guallasamin*

CONTENIDO

DECLARACION.....	i
CERTIFICACION.....	ii
AGRADECIMIENTO	iii
DEDICATORIA	iv
CONTENIDO.....	v
INDICE DE TABLAS	ix
INDICE DE FIGURAS.....	x
CONTENIDO DE ANEXOS DIGITAL.....	xi
CONTENIDO DE ANEXOS IMPRESOS	xvi
RESUMEN	xvii
PRESENTACIÓN	xviii
CAPITULO 1	1
1. MARCO TEORICO.....	1
1.1. PROCESO DE DESARROLLO DE SOFTWARE.....	1
1.1.1 Introducción.....	1
1.2. CALIDAD EN EL DESARROLLO DE SOFTWARE.....	1
1.2.1 Factores de calidad	2
1.2.1.1 Factores externos.....	2
1.2.1.2 Factores internos	3
1.2.2 Modelos de mejoramiento continuo de calidad	4
1.2.2.1 Modelos formales.....	4
1.2.2.2 Buenas prácticas.....	6
1.3. TSP	7
1.3.1 Fundamentos.....	7
1.3.2 Técnicas.....	8
1.3.3 Procesos TSP.....	11
1.3.3.1 Lanzamiento del Proyecto de Equipo (Launching a Team Project).....	11
1.3.3.1.1 Definición de las metas del equipo.....	12
1.3.3.1.2 Definición de metas por miembro del equipo	13
1.3.3.1.3 Definición de metas por rol	14
1.3.3.1.4 Coordinación del trabajo	15
1.3.3.2 Desarrollo de la Estrategia (Development Strategy).....	16
1.3.3.2.1 Creación del diseño conceptual.....	17
1.3.3.2.2 Creación de la estrategia.....	17
1.3.3.2.3 Producción del plan de administración de la configuración	19
1.3.3.2.4 Estrategia de re uso	21
1.3.3.3 Desarrollo del Plan (Development Plan).....	21
1.3.3.3.1 Proceso de planeación	21
1.3.3.3.2 Seguimiento del proyecto.....	24
1.3.3.3.3 Herramientas de apoyo.....	25
1.3.3.3.4 Plan de calidad	25
1.3.3.4 Definición de Requerimientos (Defining the Requirements).....	26
1.3.3.4.1 Obtención de los requerimientos	27
1.3.3.4.2 Especificación de requerimientos de software	27
1.3.3.4.3 Asignación de tareas de los requerimientos	28

1.3.3.4.4	Documentación de los requerimientos	28
1.3.3.4.5	Creación del plan de prueba del sistema	28
1.3.3.4.6	Inspección de los requerimientos y el plan de prueba del sistema	29
1.3.3.4.7	Modificación de los requerimientos	29
1.3.3.4.8	Revisión del SRS.....	29
1.3.3.4.9	Requerimientos básicos.....	29
1.3.3.5	Diseño con Equipos (Designing with Teams).....	30
1.3.3.5.1	Diseño en equipos	30
1.3.3.5.2	Diseño de alto nivel.....	31
1.3.3.5.3	Estándares de diseño	33
1.3.3.5.4	Diseño para re usar	35
1.3.3.5.5	Diseño para usabilidad y pruebas	36
1.3.3.5.6	Revisiones e inspecciones del diseño.....	36
1.3.3.6	Implementación del Producto (Product Implementation)	37
1.3.3.6.1	Diseño del criterio del producto terminado	37
1.3.3.6.2	Implementación.....	38
1.3.3.6.3	Estándares de implementación	39
1.3.3.6.4	Estrategia de implementación.....	40
1.3.3.6.5	Revisiones e inspecciones de la implementación	41
1.3.3.7	Pruebas de Integración y del Sistema (Integration and System Testing)	42
1.3.3.7.1	Estrategia de pruebas.....	42
1.3.3.7.2	Construcción y estrategia de integración.....	43
1.3.3.7.3	Desarrollo de pruebas.....	43
1.3.3.7.4	Construcción.....	43
1.3.3.7.5	Integración.....	44
1.3.3.7.6	Pruebas del sistema	44
1.3.3.7.7	Pruebas de regresión.....	44
1.3.3.7.8	Estrategia de pruebas del sistema	44
1.3.3.7.9	Plan de pruebas.....	45
1.3.3.7.10	Registro y medición de datos de prueba	45
1.3.3.7.11	Documentación	46
1.3.3.8	Postmortem (Postmortem)	46
1.3.3.8.1	Revisión de los datos del proceso.....	47
1.3.3.8.2	Revisión de la calidad.....	47
1.3.3.8.3	Evaluaciones del rol	48
1.3.3.8.4	Preparación del reporte del ciclo	48
1.3.3.8.5	El reporte del ciclo	48
1.3.3.8.6	Reporte de los roles	49
1.3.3.8.7	Reporte de los miembros del equipo	49
1.3.3.8.8	Producción del reporte.....	49
1.4.	RUP.....	50
1.4.1	PROCESO UNIFICADO DIRIGIDO POR CASOS DE USO	51
1.4.2	Proceso unificado centrado en la arquitectura	51
1.4.3	Proceso unificado iterativo e incremental.....	51
1.4.4	Desarrollo basado en componentes.....	52
1.4.5	UML	52
1.4.5.1	Diagramas de UML.....	52
1.4.6	Flujos de trabajo del Proceso Unificado	54
1.4.6.1	Procesos de trabajo.....	54
1.4.6.1.1	Modelado del negocio	54
1.4.6.1.2	Requisitos.....	55
1.4.6.1.3	Análisis.....	56
1.4.6.1.4	Diseño	57
1.4.6.1.5	Implementación.....	59
1.4.6.1.6	Pruebas	59
1.4.6.1.7	Despliegue o distribución	60
1.4.6.2	Soporte de procesos de trabajo.....	61
1.4.6.2.1	Gestión de cambios y configuración	61
1.4.6.2.2	Gestión de proyectos	61

1.4.6.2.3	Entorno o ambiente	62
1.4.7	Fases del ciclo de desarrollo	62
1.4.7.1	Inicio	62
1.4.7.2	Elaboración	63
1.4.7.3	Construcción	63
1.4.7.4	Transición	64
1.5.	Combinación de TSP y RUP	64
CAPITULO II.....		76
2. CASO DE ESTUDIO.....		76
2.1.	DESCRIPCIÓN DEL CASO DE ESTUDIO	76
2.1.1.	Objetivos.....	76
2.2.	DESARROLLO DEL SISTEMA USANDO TSP Y RUP	77
2.2.1.	Gestión de proyectos	77
2.2.1.1.	Conformar el equipo y asignar roles a cada uno de los integrantes.....	77
2.2.1.2.	Definir metas.....	77
2.2.1.2.1.	Definir las Metas del Equipo	77
2.2.1.2.2.	Definir las Metas por Miembro del Equipo	78
2.2.1.2.3.	Definir las Metas por Rol	78
2.2.1.3.	Coordinar el Trabajo	78
2.2.1.4.	Crear el diseño conceptual del proyecto.....	79
2.2.1.5.	Establecer riesgos potenciales.....	80
2.2.1.6.	Crear la estrategia.....	83
2.2.1.7.	Crear un plan de administración de la configuración	85
2.2.1.8.	Crear el plan del proyecto	86
2.2.2.	Modelado del Negocio.....	87
2.2.2.1.	Identificar los casos de uso del negocio y sus actores.....	87
2.2.2.1.1.	Modelo de casos de uso de negocio.....	87
2.2.2.2.	Desarrollar un modelo de objetos de un negocio	90
2.2.2.2.1.	Modelo de Dominio.....	90
2.2.3.	Requerimientos.....	90
2.2.3.1.	Obtención de los requerimientos.....	90
2.2.3.2.	Especificación de los requerimientos de software SRS	91
2.2.3.2.1.	Introducción.....	91
2.2.3.2.2.	Requisitos funcionales	94
2.2.3.2.3.	Requisitos no funcionales	94
2.2.3.2.4.	Restricciones.....	94
2.2.3.2.5.	Interfaces	95
2.2.3.3.	Crear el plan de prueba del sistema.....	95
2.2.3.4.	Revisar e inspeccionar los requerimientos	96
2.2.3.5.	Modificar los requerimientos	96
2.2.4.	Análisis	96
2.2.4.1.	Análisis de la arquitectura	96
2.2.4.2.	Análisis de los casos de uso	97
2.2.4.3.	Análisis de las clases.....	100
2.2.4.4.	Análisis de los paquetes	101
2.2.5.	Diseño.....	101
2.2.5.1.	Crear un diseño de alto nivel.....	101
2.2.5.2.	Crear un diseño detallado.....	103
2.2.5.2.1.	Diseño de la arquitectura	103
2.2.5.2.2.	Diseño de los casos de uso.....	103
2.2.5.2.3.	Diseño de clases.....	105
2.2.5.2.4.	Diseño de subsistemas	105
2.2.5.2.5.	Diseño de pantallas	105
2.2.5.3.	Crear un plan de pruebas de integración	106
2.2.5.4.	Revisar e inspeccionar el diseño	108
2.2.5.5.	Modificar el diseño	108

2.2.6.	Implementación	108
2.2.6.1.	Plan para la implementación	108
2.2.6.1.1.	Plan de pruebas de unidad	109
2.2.6.2.	Implementación de la arquitectura	110
2.2.6.3.	Implementación de los subsistemas	113
2.2.6.4.	Implementación de las clases	114
2.2.6.5.	Revisar e inspeccionar la implementación	116
2.2.6.6.	Modificar la implementación	116
2.2.6.7.	Pruebas de unidad	117
2.2.6.8.	Revisar la calidad de los componentes.....	117
2.2.6.9.	Integrar las componentes en un sistema ejecutable.....	117
2.2.7.	Pruebas	118
2.2.7.1.	Planificar las pruebas	118
2.2.7.2.	Diseñar las pruebas	118
2.2.7.2.1.	Pruebas de integración.....	118
2.2.7.2.2.	Pruebas del sistema.....	119
2.2.7.3.	Realizar las pruebas.....	120
2.2.7.3.1.	Realizar pruebas de integración.....	120
2.2.7.3.2.	Realizar pruebas del sistema.....	120
2.2.7.4.	Evaluar las pruebas.....	121
2.2.7.5.	Registrar las pruebas	122
2.2.8.	Postmortem.....	122
2.2.8.1.	Revisión de los datos.....	122
2.2.8.1.1.	Revisión de la calidad.....	123
2.2.8.2.	Evaluaciones del desempeño de los roles	126
2.2.8.3.	Preparar reportes	127
2.2.8.3.1.	Preparar el reporte del ciclo.....	127
2.2.8.3.2.	Preparar el reporte de los roles	128
2.2.8.3.3.	Preparar el reporte por miembro del equipo	128
2.2.8.4.	Producción del reporte	129
2.3.	EVALUACIÓN	130
1.5.1	Evaluación del proceso.....	130
1.5.2	Evaluación del producto	132
CAPITULO 3		134
3. CONCLUSIONES Y RECOMENDACIONES		134
3.1.	CONCLUSIONES	134
3.2.	RECOMENDACIONES.....	137
BIBLIOGRAFIA		138
GLOSARIO		140

INDICE DE TABLAS

Tabla 1.1: Estándares de defectos PSP	33
Tabla 1.2: Flujos de la combinación de TSP y RUP.	75
Tabla 2.1: Roles del equipo.....	77
Tabla 2.2: Metas del equipo	77
Tabla 2.3: Metas del miembro 1 del equipo.....	78
Tabla 2.4: Metas del miembro 2 del equipo.....	78
Tabla 2.5: Riesgos Potenciales	82
Tabla 2.6 Riesgos y beneficios de las estrategias alternativas	85
Tabla 2.7: SRS Caso de Uso: Generar Información de Clientes	92
Tabla 2.8: SRS Caso de Uso: Sincronizar Información entre la PC y la Pocket PC.....	92
Tabla 2.9: SRS Caso de Uso: Gestionar Pedidos	92
Tabla 2.10: SRS Caso de Uso: Gestionar Motivos de no Venta	92
Tabla 2.11: SRS Caso de Uso: Recibir Información del Vendedor	93
Tabla 2.12: SRS Caso de Uso: Monitorear Gestión del Vendedor	93
Tabla 2.13: SRS Prototipo de IU: Pantalla principal.....	93
Tabla 2.14: Responsabilidades de las clases de análisis.....	100
Tabla 2.15: Análisis de paquetes	101
Tabla 2.16: SDS Subsistema, módulos y funciones	103
Tabla 2.17: Diseño de pantallas: Pantalla principal.....	106
Tabla 2.18: Implementación de Clases: Ficheros.....	114
Tabla 2.19: Implementación de Clases: Métodos de los Ficheros	115
Tabla 2.20: Implementación de Clases: Métodos del Código Fuente	116
Tabla 2.21: Revisión de la calidad: Metas del equipo	124
Tabla 2.22: Revisión de la calidad: Metas del miembro 1 del equipo.....	124
Tabla 2.23: Revisión de la calidad: Metas del miembro 2 del equipo.....	125
Tabla 2.24: Revisión de la calidad del sistema: Factores externos	125
Tabla 2.25: Revisión de la calidad del sistema: Factores internos	125
Tabla 2.26: Evaluación del desempeño de los roles 1	126
Tabla 2.27: Evaluación del desempeño de los roles 2	126

INDICE DE FIGURAS

FIGURA 1.1: Proceso de lanzamiento en TSP	11
FIGURA 1.2: Proceso de planificacion (HUMPHREY, Watts. Introduction to the Team Software Process)	23
FIGURA 1.3: Procesos del monitoreo del proyecto	26
FIGURA 1.4: Pocesos unificados de desarrollo. (http://www.willydev.net/descargas/articulos/general/ingsoftware.aspx)	54
FIGURA 2.1: Diagrama de casos de uso de negocio	88
FIGURA 2.2: Diagrama de casos de uso del sistema	91
FIGURA 2.3: Paquetes de análisis	96
FIGURA 2.5: Diagrama de colaboración del caso de uso generar información de clientes	98
FIGURA 2.6: Modelo de análisis	99
FIGURA 2.7: Modelo de despliegue de la arquitectura.....	103
FIGURA 2.9: Diagrama de secuencia del caso de uso generar información de clientes...	104
FIGURA 2.10: Diseño de subsistemas	105
FIGURA 2.11: Base de datos de la PC	111
FIGURA 2.12 Base de datos de la pocket PC	112

CONTENIDO DE ANEXOS DIGITAL

ANEXOS 1.....	;Error! Marcador no definido.	
PROYECT NOTEBOOK.....	;Error! Marcador no definido.	
1.1 GESTION DE PROYECTOS.....	;Error! Marcador no definido.	
1.1.1 Forma INFO: Geovanna Bustos	;Error! Marcador no definido.	
1.1.2 Forma INFO: Cristhian Guallasamin.....	;Error! Marcador no definido.	
1.1.3 Form TASK: Geovanna Bustos	;Error! Marcador no definido.	
1.1.4 Form TASK: Cristhian Guallasamin	;Error! Marcador no definido.	
1.1.5 Form SCHEDULE: Geovanna Bustos.....	;Error! Marcador no definido.	
1.1.6 Form SCHEDULE: Cristhian Guallasamin	;Error! Marcador no definido.	
1.1.7 Form WEEK: Geovanna Bustos	;Error! Marcador no definido.	
1.1.8 Form WEEK: Cristhian Guallasamin	;Error! Marcador no definido.	
1.1.9 Form WEEK del Equipo.....	;Error! Marcador no definido.	
1.1.10 Form ITL	;Error! Marcador no definido.	
1.1.11 Form STRAT	;Error! Marcador no definido.	
1.1.12 Form SUMS.....	;Error! Marcador no definido.	
1.1.13 Form TASK: Equipo	;Error! Marcador no definido.	
1.1.14 Form SCHEDULE del Equipo	;Error! Marcador no definido.	
1.1.15 Form SUMP.....	;Error! Marcador no definido.	
1.1.16 Form SUMQ.....	;Error! Marcador no definido.	
1.1.17 Form TASK: Geovanna Bustos	;Error! Marcador no definido.	
1.1.18 Form TASK: Cristhian Guallasamin	;Error! Marcador no definido.	
1.1.19 Form SUMP.....	;Error! Marcador no definido.	
1.1.20 Form SUMQ.....	;Error! Marcador no definido.	
1.1.21 Form LOGT: Geovanna Bustos.....	;Error! Marcador no definido.	
1.1.22 Form LOGT: Cristhian Guallasamin	;Error! Marcador no definido.	
1.1.23 Form TASK: Geovanna Bustos	;Error! Marcador no definido.	
1.1.24 Form TASK: Cristhian Guallasamin	;Error! Marcador no definido.	
1.1.25 Form TASK del Equipo.....	;Error! Marcador no definido.	
1.1.26 Form SCHEDULE: Geovanna Bustos.....	;Error! Marcador no definido.	
1.1.27 Form SCHEDULE: Cristhian Guallasamin	;Error! Marcador no definido.	
1.1.28 Form SCHEDULE del Equipo	;Error! Marcador no definido.	
1.1.29 Form WEEK: Geovanna Bustos	;Error! Marcador no definido.	
1.1.30 Form WEEK: Cristhian Guallasamin	;Error! Marcador no definido.	
1.1.31 Form WEEK del Equipo.....	;Error! Marcador no definido.	
1.1.32 Form ITL	;Error! Marcador no definido.	
1.2 MODELADO DEL NEGOCIO Y REQUERIMIENTOS	;Error! Marcador no definido.	
1.2.1 TSPi Inspection Report: Form INS.....	;Error! Marcador no definido.	
1.2.2 Form LOGT: Geovanna Bustos	;Error! Marcador no definido.	
1.2.3 Form LOGT: Cristhian Guallasamin	;Error! Marcador no definido.	
1.2.4 Form TASK: Geovanna Bustos	;Error! Marcador no definido.	
1.2.5 Form TASK: Cristhian Guallasamin	;Error! Marcador no definido.	
1.2.6 Form TASK del Equipo.....	;Error! Marcador no definido.	
1.2.7 Form SCHEDULE: Geovanna Bustos.....	;Error! Marcador no definido.	
1.2.8 Form SCHEDULE: Cristhian Guallasamin	;Error! Marcador no definido.	
1.2.9 Form SCHEDULE del Equipo	;Error! Marcador no definido.	
1.2.10 Form WEEK: Geovanna Bustos	;Error! Marcador no definido.	
1.2.11 Form WEEK: Cristhian Guallasamin	;Error! Marcador no definido.	
1.2.12 Form WEEK del Equipo.....	;Error! Marcador no definido.	
1.2.13 Form ITL	;Error! Marcador no definido.	
1.2.14 Form LOGD: Geovanna Bustos	;Error! Marcador no definido.	
1.2.15 Form LOGD: Cristhian Guallasamin.....	;Error! Marcador no definido.	
1.2.16 Form SUMP.....	;Error! Marcador no definido.	
1.2.17 Form SUMQ.....	;Error! Marcador no definido.	

1.3 ANALISIS	;	Error! Marcador no definido.
1.3.1 Form LOGT: Geovanna Bustos	;	Error! Marcador no definido.
1.3.2 Form LOGT: Cristhian Guallasamin	;	Error! Marcador no definido.
1.3.3 Form TASK: Geovanna Bustos	;	Error! Marcador no definido.
1.3.4 Form TASK: Cristhian Guallasamin	;	Error! Marcador no definido.
1.3.5 Form TASK del Equipo.....	;	Error! Marcador no definido.
1.3.6 Form SCHEDULE: Geovanna Bustos.....	;	Error! Marcador no definido.
1.3.7 Form SCHEDULE: Cristhian Guallasamin.....	;	Error! Marcador no definido.
1.3.8 Form SCHEDULE del Equipo	;	Error! Marcador no definido.
1.3.9 Form WEEK: Geovanna Bustos	;	Error! Marcador no definido.
1.3.10 Form WEEK: Cristhian Guallasamin	;	Error! Marcador no definido.
1.3.11 Form WEEK del Equipo.....	;	Error! Marcador no definido.
1.3.12 Form ITL	;	Error! Marcador no definido.
1.3.13 Form SUMP.....	;	Error! Marcador no definido.
1.3.14 Form SUMQ.....	;	Error! Marcador no definido.
1.4 DISEÑO.....	;	Error! Marcador no definido.
1.4.1 TSPi Inspection Report: Form INS.....	;	Error! Marcador no definido.
1.4.2 Form LOGT: Geovanna Bustos.....	;	Error! Marcador no definido.
1.4.3 Form LOGT: Cristhian Guallasamin	;	Error! Marcador no definido.
1.4.4 Form TASK: Geovanna Bustos	;	Error! Marcador no definido.
1.4.5 Form TASK: Cristhian Guallasamin	;	Error! Marcador no definido.
1.4.6 Form TASK del Equipo.....	;	Error! Marcador no definido.
1.4.7 Form SCHEDULE: Geovanna Bustos.....	;	Error! Marcador no definido.
1.4.8 Form SCHEDULE: Cristhian Guallasamin.....	;	Error! Marcador no definido.
1.4.9 Form SCHEDULE del Equipo	;	Error! Marcador no definido.
1.4.10 Form WEEK: Geovanna Bustos	;	Error! Marcador no definido.
1.4.11 Form WEEK: Cristhian Guallasamin	;	Error! Marcador no definido.
1.4.12 Form WEEK del Equipo.....	;	Error! Marcador no definido.
1.4.13 Form ITL	;	Error! Marcador no definido.
1.4.14 Form LOGD: Geovanna Bustos	;	Error! Marcador no definido.
1.4.15 Form LOGD: Cristhian Guallasamin.....	;	Error! Marcador no definido.
1.4.16 Form SUMP.....	;	Error! Marcador no definido.
1.4.17 Form SUMQ.....	;	Error! Marcador no definido.
1.5 IMPLEMENTACIÓN.....	;	Error! Marcador no definido.
1.5.1 Form TASK: Geovanna Bustos	;	Error! Marcador no definido.
1.5.2 Form TASK: Cristhian Guallasamin	;	Error! Marcador no definido.
1.5.3 Form SCHEDULE: Geovanna Bustos.....	;	Error! Marcador no definido.
1.5.4 Form SCHEDULE: Cristhian Guallasamin	;	Error! Marcador no definido.
1.5.5 Form LOGT: Geovanna Bustos.....	;	Error! Marcador no definido.
1.5.6 Form LOGT: Cristhian Guallasamin	;	Error! Marcador no definido.
1.5.7 Form TASK: Geovanna Bustos	;	Error! Marcador no definido.
1.5.8 Form TASK: Cristhian Guallasamin	;	Error! Marcador no definido.
1.5.9 Form TASK del Equipo.....	;	Error! Marcador no definido.
1.5.10 Form SCHEDULE: Geovanna Bustos.....	;	Error! Marcador no definido.
1.5.11 Form SCHEDULE: Cristhian Guallasamin.....	;	Error! Marcador no definido.
1.5.12 Form SCHEDULE del Equipo	;	Error! Marcador no definido.
1.5.13 Form WEEK: Geovanna Bustos	;	Error! Marcador no definido.
1.5.14 Form WEEK: Cristhian Guallasamin	;	Error! Marcador no definido.
1.5.15 Form WEEK del Equipo.....	;	Error! Marcador no definido.
1.5.16 Form WEEK: Geovanna Bustos	;	Error! Marcador no definido.
1.5.17 Form WEEK: Cristhian Guallasamin	;	Error! Marcador no definido.
1.5.18 Form WEEK del Equipo.....	;	Error! Marcador no definido.
1.5.19 Form ITL	;	Error! Marcador no definido.
1.5.20 Form SUMP.....	;	Error! Marcador no definido.
1.5.21 Form SUMQ.....	;	Error! Marcador no definido.
1.5.22 Form LOGT: Geovanna Bustos.....	;	Error! Marcador no definido.
1.5.23 Form LOGT: Cristhian Guallasamin	;	Error! Marcador no definido.
1.5.24 Form TASK: Geovanna Bustos	;	Error! Marcador no definido.

1.5.25	Form TASK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.26	Form TASK del Equipo.....	¡Error! Marcador no definido.
1.5.27	Form SCHEDULE: Geovanna Bustos.....	¡Error! Marcador no definido.
1.5.28	Form SCHEDULE: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.29	Form SCHEDULE del Equipo	¡Error! Marcador no definido.
1.5.30	Form WEEK: Geovanna Bustos	¡Error! Marcador no definido.
1.5.31	Form WEEK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.32	Form WEEK del Equipo.....	¡Error! Marcador no definido.
1.5.33	Form ITL	¡Error! Marcador no definido.
1.5.34	Form SUMP.....	¡Error! Marcador no definido.
1.5.35	Form LOGT: Geovanna Bustos	¡Error! Marcador no definido.
1.5.36	Form LOGT: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.37	Form TASK: Geovanna Bustos	¡Error! Marcador no definido.
1.5.38	Form TASK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.39	Form TASK del Equipo.....	¡Error! Marcador no definido.
1.5.40	Form SCHEDULE: Geovanna Bustos.....	¡Error! Marcador no definido.
1.5.41	Form SCHEDULE: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.42	Form SCHEDULE del Equipo	¡Error! Marcador no definido.
1.5.43	Form LOGD: Geovanna Bustos	¡Error! Marcador no definido.
1.5.44	Form LOGD: Cristhian Guallasamin.....	¡Error! Marcador no definido.
1.5.45	Form SUMP.....	¡Error! Marcador no definido.
1.5.46	Form SUMQ.....	¡Error! Marcador no definido.
1.5.47	Form SUMS.....	¡Error! Marcador no definido.
1.5.48	TSPi Inspection Report: Form INS.....	¡Error! Marcador no definido.
1.5.49	Form LOGT: Geovanna Bustos	¡Error! Marcador no definido.
1.5.50	Form LOGT: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.51	Form TASK: Geovanna Bustos	¡Error! Marcador no definido.
1.5.52	Form TASK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.53	Form TASK del Equipo.....	¡Error! Marcador no definido.
1.5.54	Form SCHEDULE: Geovanna Bustos.....	¡Error! Marcador no definido.
1.5.55	Form SCHEDULE: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.56	Form SCHEDULE del Equipo	¡Error! Marcador no definido.
1.5.57	Form WEEK: Geovanna Bustos	¡Error! Marcador no definido.
1.5.58	Form WEEK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.59	Form WEEK del Equipo.....	¡Error! Marcador no definido.
1.5.60	Form ITL	¡Error! Marcador no definido.
1.5.61	Form LOGD: Geovanna Bustos	¡Error! Marcador no definido.
1.5.62	Form LOGD: Cristhian Guallasamin.....	¡Error! Marcador no definido.
1.5.63	Form SUMP.....	¡Error! Marcador no definido.
1.5.64	Form SUMQ.....	¡Error! Marcador no definido.
1.5.65	Form SUMS.....	¡Error! Marcador no definido.
1.5.66	Form LOGT: Geovanna Bustos.....	¡Error! Marcador no definido.
1.5.67	Form LOGT: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.68	Form TASK: Geovanna Bustos	¡Error! Marcador no definido.
1.5.69	Form TASK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.70	Form TASK del Equipo.....	¡Error! Marcador no definido.
1.5.71	Form SCHEDULE: Geovanna Bustos.....	¡Error! Marcador no definido.
1.5.72	Form SCHEDULE: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.73	Form SCHEDULE del Equipo	¡Error! Marcador no definido.
1.5.74	Form WEEK: Geovanna Bustos	¡Error! Marcador no definido.
1.5.75	Form WEEK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.76	Form WEEK del Equipo.....	¡Error! Marcador no definido.
1.5.77	Form ITL	¡Error! Marcador no definido.
1.5.78	Form SUMP.....	¡Error! Marcador no definido.
1.5.79	Form SUMQ.....	¡Error! Marcador no definido.
1.5.80	Form SUMS.....	¡Error! Marcador no definido.
1.5.81	Form LOGD: Geovanna Bustos	¡Error! Marcador no definido.
1.5.82	Form LOGD: Cristhian Guallasamin.....	160
1.5.83	Form LOGT: Geovanna Bustos.....	164

1.5.84	Form LOGT: Cristhian Guallasamin	164
1.5.85	Form LOGTEST: Geovanna Bustos	¡Error! Marcador no definido.
1.5.86	Form LOGTEST: Cristhian Guallasamin	164
1.5.87	Form TASK: Geovanna Bustos	¡Error! Marcador no definido.
1.5.88	Form TASK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.89	Form TASK del Equipo.....	¡Error! Marcador no definido.
1.5.90	Form SCHEDULE: Geovanna Bustos.....	¡Error! Marcador no definido.
1.5.91	Form SCHEDULE: Cristhian Guallasamin.....	¡Error! Marcador no definido.
1.5.92	Form SCHEDULE del Equipo	¡Error! Marcador no definido.
1.5.93	Form WEEK: Geovanna Bustos	¡Error! Marcador no definido.
1.5.94	Form WEEK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.95	Form WEEK del Equipo.....	¡Error! Marcador no definido.
1.5.96	Form ITL	¡Error! Marcador no definido.
1.5.97	Form SUMP.....	¡Error! Marcador no definido.
1.5.98	Form SUMQ.....	¡Error! Marcador no definido.
1.5.99	Form SUMS.....	¡Error! Marcador no definido.
1.5.100	Form LOGT: Geovanna Bustos.....	¡Error! Marcador no definido.
1.5.101	Form LOGT: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.102	Form TASK: Geovanna Bustos	¡Error! Marcador no definido.
1.5.103	Form TASK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.104	Form TASK del Equipo.....	¡Error! Marcador no definido.
1.5.105	Form SCHEDULE: Geovanna Bustos.....	¡Error! Marcador no definido.
1.5.106	Form SCHEDULE: Cristhian Guallasamin.....	¡Error! Marcador no definido.
1.5.107	Form SCHEDULE del Equipo	¡Error! Marcador no definido.
1.5.108	Form WEEK: Geovanna Bustos	¡Error! Marcador no definido.
1.5.109	Form WEEK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.5.110	Form WEEK del Equipo.....	¡Error! Marcador no definido.
1.5.111	Form ITL	¡Error! Marcador no definido.
1.5.112	Form SUMP.....	¡Error! Marcador no definido.
1.5.113	Form SUMQ.....	¡Error! Marcador no definido.
1.5.114	Form SUMS.....	¡Error! Marcador no definido.
1.6	PRUEBAS DE INTEGRACION Y DEL SISTEMA	¡Error! Marcador no definido.
1.6.1	Form LOGTEST: Geovanna Bustos	¡Error! Marcador no definido.
1.6.2	Form LOGTEST: Cristhian Guallasamin	¡Error! Marcador no definido.
1.6.3	Form LOGT: Geovanna Bustos	¡Error! Marcador no definido.
1.6.4	Form LOGT: Cristhian Guallasamin	¡Error! Marcador no definido.
1.6.5	Form TASK: Geovanna Bustos	¡Error! Marcador no definido.
1.6.6	Form TASK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.6.7	Form TASK del Equipo.....	¡Error! Marcador no definido.
1.6.8	Form SCHEDULE: Geovanna Bustos.....	¡Error! Marcador no definido.
1.6.9	Form SCHEDULE: Cristhian Guallasamin.....	¡Error! Marcador no definido.
1.6.10	Form SCHEDULE del Equipo	¡Error! Marcador no definido.
1.6.11	Form WEEK: Geovanna Bustos	¡Error! Marcador no definido.
1.6.12	Form WEEK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.6.13	Form WEEK del Equipo.....	¡Error! Marcador no definido.
1.6.14	Form ITL	¡Error! Marcador no definido.
1.6.15	Form SUMP.....	¡Error! Marcador no definido.
1.6.16	Form SUMQ.....	¡Error! Marcador no definido.
1.7	POSTMORTEM.....	¡Error! Marcador no definido.
1.7.1	Process Improvement Proposal: PIPS.....	¡Error! Marcador no definido.
1.7.2	Team and Peer Evaluation PEER: Geovanna Bustos	¡Error! Marcador no definido.
1.7.3	Team and Peer Evaluation PEER: Cristhian Guallasamin.....	¡Error! Marcador no definido.
1.7.4	Form LOGT: Geovanna Bustos.....	¡Error! Marcador no definido.
1.7.5	Form LOGT: Cristhian Guallasamin	¡Error! Marcador no definido.
1.7.6	Form TASK: Geovanna Bustos	¡Error! Marcador no definido.
1.7.7	Form TASK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.7.8	Form TASK del Equipo.....	¡Error! Marcador no definido.
1.7.9	Form SCHEDULE: Geovanna Bustos.....	¡Error! Marcador no definido.

1.7.10	Form SCHEDULE: Cristhian Guallasamin	¡Error! Marcador no definido.
1.7.11	Form SCHEDULE del Equipo	¡Error! Marcador no definido.
1.7.12	Form WEEK: Geovanna Bustos	¡Error! Marcador no definido.
1.7.13	Form WEEK: Cristhian Guallasamin	¡Error! Marcador no definido.
1.7.14	Form WEEK del Equipo.....	¡Error! Marcador no definido.
1.7.15	Form ITL	¡Error! Marcador no definido.
1.7.16	Form SUMP.....	¡Error! Marcador no definido.
1.7.17	Form SUMQ.....	¡Error! Marcador no definido.
ANEXOS 2.....		¡Error! Marcador no definido.
2.1	GESTION DE PROYECTOS.....	¡Error! Marcador no definido.
2.1.1	Definir las Metas por Rol	¡Error! Marcador no definido.
2.2	MODELADO DEL NEGOCIO.....	¡Error! Marcador no definido.
2.2.1	Modelo de Dominio.....	¡Error! Marcador no definido.
2.3	REQUERIMIENTOS.....	¡Error! Marcador no definido.
2.3.1	Prototipo de IU	¡Error! Marcador no definido.
2.4	ANALISIS	¡Error! Marcador no definido.
2.4.1	Análisis arquitectónico	¡Error! Marcador no definido.
2.4.2	Clases de análisis de los casos de uso.....	¡Error! Marcador no definido.
2.4.3	Diagramas de Colaboración de los casos de uso	¡Error! Marcador no definido.
2.4.4	Análisis de clases.....	¡Error! Marcador no definido.
2.5	DISEÑO.....	¡Error! Marcador no definido.
2.5.1	Glosario de nombres	¡Error! Marcador no definido.
2.5.2	Estándares de diseño.....	¡Error! Marcador no definido.
2.5.3	Diagrama de clases de diseño.....	¡Error! Marcador no definido.
2.5.4	Diagramas de interacción o secuencia de los casos de uso....	¡Error! Marcador no definido.
2.5.5	Diagrama de clases	¡Error! Marcador no definido.
2.5.6	Diseño de pantallas	¡Error! Marcador no definido.
2.5.7	Software design specifications (SDS).....	¡Error! Marcador no definido.
2.6	IMPLEMENTACIÓN.....	¡Error! Marcador no definido.
2.6.1	Estándares de Implementacion	¡Error! Marcador no definido.
2.6.2	Estándares de Implementacion	¡Error! Marcador no definido.
2.6.3	Pantallas del proyecto en Microsoft Visual Basic .Net para la PC	¡Error! Marcador no definido.
2.6.4	Pantallas del proyecto en Microsoft Visual Basic .Net para la Pocket PC	¡Error! Marcador no definido.
2.6.5	Calidad de los Componentes	¡Error! Marcador no definido.
2.6.6	Manual de usuario	¡Error! Marcador no definido.
2.7	PRUEBAS DE INTEGRACION Y DEL SISTEMA	¡Error! Marcador no definido.
2.7.1	Diseño de las Pruebas de Integración	¡Error! Marcador no definido.
2.7.2	Pruebas de integración	¡Error! Marcador no definido.

CONTENIDO DE ANEXOS IMPRESOS

ANEXOS 1		¡Error! Marcador no definido.
1.1	Forma INFO Individual	¡Error! Marcador no definido.
1.2	Form TASK Individual	¡Error! Marcador no definido.
1.3	Form SCHEDULE Individual	¡Error! Marcador no definido.
1.4	Form WEEK Individual	¡Error! Marcador no definido.
1.5	Form WEEK del Equipo	¡Error! Marcador no definido.
1.6	Form ITL	¡Error! Marcador no definido.
1.7	Form STRAT	¡Error! Marcador no definido.
1.8	Form SUMS	¡Error! Marcador no definido.
1.9	Form TASK DEL Equipo	¡Error! Marcador no definido.
1.10	Form SCHEDULE del Equipo	¡Error! Marcador no definido.
1.11	Form SUMP	¡Error! Marcador no definido.
1.12	Form SUMQ	¡Error! Marcador no definido.
1.13	Form INS (Requerimientos)	¡Error! Marcador no definido.
1.14	Form INS (Diseño)	¡Error! Marcador no definido.
1.15	Form INS (Implementación)	¡Error! Marcador no definido.
1.16	Form LOGD Individual	¡Error! Marcador no definido.
1.17	Form LOGT Individual	¡Error! Marcador no definido.
1.18	Form LOGTEST Individual	¡Error! Marcador no definido.
1.19	Process Improvement Proposal: PIP	¡Error! Marcador no definido.
1.20	Team and Peer Evaluation PEER Individual	¡Error! Marcador no definido.
ANEXOS 2		¡Error! Marcador no definido.
2.1	Responsabilidades de los roles del equipo	¡Error! Marcador no definido.
2.2	Standard Qual	¡Error! Marcador no definido.
2.3	Metas por Rol	¡Error! Marcador no definido.
2.4	Software Design Specifications (SDS)	¡Error! Marcador no definido.
2.5	Calidad de los componentes	¡Error! Marcador no definido.
2.6	Encuestas	¡Error! Marcador no definido.

RESUMEN

Los sistemas de software cada vez incrementan su tamaño de manera que los grupos de trabajo que se establecen para su desarrollo deben mantener una organización para realizar su trabajo eficientemente. TSP contiene un conjunto de procesos (Lanzamiento del Proyecto de Equipo, Desarrollo de la Estrategia, Desarrollo del Plan, Definición de Requerimientos, Diseño con Equipos, Implementación del Producto, Pruebas de Integración y del Sistema y Postmortem) que permiten el desarrollo de productos de software de manera eficiente, estos procesos permiten administrar, supervisar y reportar el trabajo en equipo, el cual puede estar conformado entre 2 a 20 integrantes. Mientras que RUP establece una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo, lo que permite la producción de software de calidad dentro de plazos y presupuestos predecibles.

Al combinar los procesos de TSP y los flujos de RUP (Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Distribución, Gestión de Configuración y Cambios, Gestión de Proyectos y Ambiente) se obtienen como resultado los flujos: Gestión de Proyectos, Modelado del Negocio, Requerimientos, Análisis, Diseño, Implementación, Pruebas e Integración, Postmortem, y Distribución.

Al desarrollar productos de software utilizando los flujos de la combinación de los procesos de TSP y los flujos de RUP se obtendrá como resultado un producto con calidad de manera eficiente. Para aplicar la combinación del TSP y RUP desarrollamos el caso de estudio: Desarrollo de Software para la Administración y Monitoreo de Rutas a seguir por los vendedores de Puerta a Puerta, que nos permitió obtener un producto de buena calidad con el uso de TSP en el desarrollo de software.

PRESENTACIÓN

En el presente trabajo se crea la combinación de TSP (Team Software Process) y la metodología estándar de Desarrollo de Software RUP con el propósito de aplicar los flujos de esta combinación en el desarrollo del caso de estudio: Desarrollo de Software para Administración y Monitoreo de Rutas a Seguir por los Vendedores de Puerta a Puerta.

En el capítulo 1 se presenta el proceso desarrollo de software y la factores de calidad que permiten establecer el grado de calidad de un software. Los modelos de mejoramiento continuo de calidad, los modelos formales: CMM, SPICE, ISO y las buenas prácticas PSP y TSP. Se detallan los fundamentos, técnicas y los procesos de TSP, las características, flujos de trabajo y las fases del ciclo de desarrollo de RUP.

También se definen los flujos que resultan de combinar los procesos de TSP y los flujos de trabajo del proceso unificado de desarrollo RUP: Gestión de Proyectos, Modelado del Negocio, Requerimientos, Análisis, Diseño, Implementación, Pruebas e Integración, Postmortem, y Distribución. Cada flujo contiene actividades que se han de considerar en el desarrollo de un proyecto con TSP y RUP.

En el capítulo 2 se presenta el desarrollo del caso de estudio Software para Administración y Monitoreo de Rutas a seguir por los Vendedores de Puerta a Puerta usando las fases establecidas en la combinación de TSP y RUP. Se hace una evaluación de la combinación TSP y RUP en base al desarrollo del caso de estudio.

En el capítulo 3 se presentan las conclusiones y recomendaciones que se obtuvieron y establecieron concluido el capítulo 2.

CAPITULO 1

1. MARCO TEORICO

1.1. PROCESO DE DESARROLLO DE SOFTWARE

1.1.1 INTRODUCCIÓN

El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo."¹

El proceso de desarrollo de software requiere de un conjunto de conceptos, modelos y de un lenguaje propio. Los modelos del ciclo de vida en los proyectos han evolucionado constantemente con el fin de mejorar el marco de trabajo y ajustarle a niveles de riesgo cada vez más altos.

1.2. CALIDAD EN EL DESARROLLO DE SOFTWARE

Calidad es "Un proceso de mejoramiento continuo, en donde todas las áreas de la empresa participan activamente en el desarrollo de productos y servicios, que satisfagan las necesidades del cliente, logrando con ello mayor productividad".²

Calidad es satisfacer plenamente las necesidades y expectativas del cliente, lograr productos y servicios con cero defectos. Es diseñar, producir y entregar un producto de satisfacción total de acuerdo con las normas establecidas y en el menor tiempo.

¹ ZAVALA R, Ingeniería de Software los Fundamentos,
<http://www.willydev.net/descargas/Articulos/General/ingsoftware.aspx>.

² Morea, Lucas. Calidad Total,
<http://www.monografias.com/trabajos14/calidadtotal/calidadtotal.shtml>.

1.2.1 FACTORES DE CALIDAD

1.2.1.1 Factores externos

Los factores externos son aquellos que solamente pueden ser medidos en base a como el objeto se relaciona con su ambiente.

Correctitud: Este factor indica el grado con el cual el sistema cumple con las especificaciones y los objetivos planteados por el usuario; es decir, si el sistema produce las salidas esperadas. Este factor de calidad siempre debe existir en todo proyecto.

Confiabilidad: Este factor indica el grado de confianza en el correcto funcionamiento del sistema. Por ejemplo si dentro de un programa un 10% de sus resultados no son correctos entonces se puede decir que el programa no es fiable.

Robustez: Este factor indica la habilidad que posee el sistema para funcionar en condiciones anormales, es decir, en condiciones que no fueron estimadas en las especificaciones.

Compatibilidad: Este factor indica el grado de facilidad que posee el sistema para combinar un elemento del mismo con otro.

Eficiencia: Este factor indica el grado de habilidad que posee el sistema para cumplir una función utilizando la menor demanda de recursos de Hardware como sea posible.

Facilidad de Uso: Este factor indica el grado de simplicidad para que el sistema sea fácil de usar y entender para los usuarios finales.

Funcionalidad: Este factor indica la capacidad que tiene el sistema de satisfacer las necesidades del usuario en condiciones especificadas, es decir, si el software

realiza funciones con propiedades específicas que en conjunto satisfacen las necesidades originales.

Integridad: Este factor indica el grado en el cual el sistema garantiza que los datos se mantengan íntegros.

1.2.1.2 Factores internos

Los Factores Internos son aquellos que se los puede separar del ambiente y estos pueden ser medidos con respecto a si mismos. Se puede analizar los factores internos de forma aislada.

Flexibilidad: Este factor indica el costo que se necesita para realizar modificaciones en el sistema cuando se necesitan hacer cambios en las especificaciones.

Portabilidad: Este factor indica el grado de facilidad para transportar un sistema de un medio ambiente a otro. Los medios ambientes incluyen: organizacional, hardware y software.

Facilidad de mantenimiento: Este factor indica el grado de facilidad para entender, corregir, adaptar y mejorar el sistema. Las modificaciones específicas se pueden realizar sobre el ambiente, los requerimientos y/o capacitaciones funcionales del sistema. La clave es tener documentación del software.

Facilidad de pruebas: Este factor indica el costo que se requiere para verificar que el sistema satisface los requerimientos solicitados por el usuario final.

Reusabilidad: Este factor indica el grado de facilidad para utilizar productos de software completos o partes de él en nuevas aplicaciones.

Puntualidad: Este factor indica el grado de habilidad del sistema para ser integrado y de esta manera disminuir el tiempo de entrega esperado por los usuarios finales.

Interoperabilidad: Este factor indica el grado de habilidad del sistema para interactuar con otro software.

1.2.2 MODELOS DE MEJORAMIENTO CONTINUO DE CALIDAD

1.2.2.1 Modelos formales

CMM

“CMM describe los principios y prácticas relacionadas al proceso de madurez de software e intenta ayudar a las organizaciones de software a mejorar la madurez de sus procesos de software en términos de un camino evolutivo, desde procesos caóticos hasta procesos maduros y disciplinados”.³

CMM fue creado a partir del cuestionario inicial de la Software Engineering Institute (SEI) y del Software Process Assessment (SPA). CMM se caracteriza por estar basado en experiencias anteriores sobre mejoras de procesos, gran parte de su efectividad se debe a la documentación que se encuentra al alcance de cualquier persona de forma gratuita.

CMM está compuesto por niveles de madurez, para alcanzar estos niveles se deben cumplir con las metas o Key Process Areas (KPA). Para llevar a cabo las metas se deben considerar las características comunes y las palabras clave definidas en CMM. Los niveles de madurez son: Nivel 1: Inicial, Nivel 2: Repetible, Nivel 3: Definido, Nivel 4: Administrado, Nivel 5: Optimizado.

SPICE

Software Process Improvement and Capability Determination (SPICE) es un proyecto para el mejoramiento de proyectos de software.

³ MORENO, José. Aplicación de un Sistema Experto para el desarrollo de Sistema Evaluador del modelo Capability Maturity Model(CMM) niveles dos y tres.
http://140.148.3.250/u_dl_a/servlet/mx.udlap.ict.tales.html.Block?Thesis=1565&Type=T

Los niveles de SPICE son: Nivel 0: No realizado, Nivel 1: Realizado informalmente, Nivel 2: Planificado y controlado, Nivel 3: Nivel bien definido, Nivel 4: Controlado cuantitativamente, Nivel 5: Mejoramiento continuo.

SPICE y CMM se diferencian en que SPICE está orientado al proceso mientras que CMM está orientado a la organización.

ISO⁴

International Organization for Standardization (ISO) es una organización a nivel mundial cuya misión es promover el desarrollo de la estandarización y actividades relacionadas a la técnica con el propósito de facilitar el intercambio internacional de bienes y servicios. Se enfoca en desarrollar la cooperación en las áreas de actividades intelectuales, científicas, tecnológicas y económicas. Existen distintas series ISO dependiendo del enfoque pueden ser usadas para el manejo de calidad interna (ISO 9004) y externa (ISO 9001, ISO 9002, ISO 9003).

ISO 9001 comprende el diseño, desarrollo, producción, instalación y servicio. ISO 9001 tiene como propósito asegurar a los clientes que los productos y servicios que adquiere del proveedor son de calidad.

ISO 9000-3 facilita la aplicación de la norma ISO 9001 en organizaciones dedicadas a desarrollar, suministrar y mantener software, posee extensiones que se refieren a temas como revisiones o requerimientos de compra.

ISO/IEC 15504 es una norma compuesta de nueve partes que son divididas en niveles (Nivel 0: Incompleto, Nivel 1: Realizado, Nivel 2: Administrado, Nivel 3: Establecido, Nivel 4: Predecible, Nivel 5: Optimo) enfocados a los procesos.

⁴ MORENO, José. Aplicación de un Sistema Experto para el desarrollo de Sistema Evaluador del modelo Capability Maturity Model(CMM) niveles dos y tres.
http://140.148.3.250/u_dl_a/servlet/mx.udlap.ict.tales.html.Block?Thesis=1565&Type=T

ISO es una métrica que tiene defectos en campos en los que no tiene control como las prácticas y estilos de la administración en el proceso, el producto final y la interacción en la entrega con el cliente.

1.2.2.2 Buenas prácticas

PSP

“El proceso de PSP consiste de un conjunto de métodos, formatos y reglas que muestran a los ingenieros como planear y administrar su trabajo.”⁵

PSP puede ser usado con cualquier lenguaje de programación o método de diseño, en los requerimientos de trabajo, al correr las pruebas, definir los procesos y reparar defectos. PSP fue diseñado por la SEI en la búsqueda de una aplicación de CMM. El enfoque de PSP ayuda al trabajo en empresas o equipos de proyectos pequeños. Promueve el trabajo de calidad en cada persona del proyecto para producir un software de calidad.

PSP está formado por siete procesos de desarrollo de software divididos en 4 niveles: Nivel 0: Bases del proceso personal (0.0 y 0.1), Nivel 1: Administración del proceso personal (1.0 y 1.1), Nivel 2: Administración personal de calidad (2.0 y 2.1), Nivel 3: Proceso personal Cíclico (3.0).

PSP tiene como base el modelo de cascada para el ciclo de vida del proyecto. En el modelo PSP se incluyen estándares para las prácticas del proceso. Para poner en práctica PSP no se necesita de herramientas especiales, solo que se requiere de formas y hojas de trabajo.

TSP

TSP fue creado en 1996 para solucionar los problemas de coordinación y comunicación entre los desarrolladores y los administrativos que se presentaron al implementar PSP en equipos organizacionales.

⁵ MORENO, José. Aplicación de un Sistema Experto para el desarrollo de Sistema Evaluador del modelo Capability Maturity Model(CMM) niveles dos y tres.
http://140.148.3.250/u_dl_a/servlet/mx.udlap.ict.tales.html.Block?Thesis=1565&Type=T

TSP al igual que PSP es un proceso que consiste de un conjunto de métodos, formatos y reglas que ayudan al desarrollo en equipos de un proyecto de gran calidad y madurez. TSP junto con PSP y CMM son las herramientas ideales para ayudar a las organizaciones en la formación de sistemas de software de alta calidad.

PSP y CMM proveen de una lista de habilidades necesarias para llevar a cabo un proyecto de software efectivo y TSP es una guía para la realización del proyecto.

PSP y TSP son procesos que no son rápidos de implantar, PSP puede tomar hasta 6 meses en un grupo de 30 personas en una empresa, mientras que TSP hasta un año y necesita de capacitación y emparejamiento tecnológico muy pesado.⁶

1.3. TSP

1.3.1 FUNDAMENTOS

Para ser competitivos y estar a la par con la industria del software se tiene que contar con equipos humanos de alto desempeño. Estos equipos son estructuras al interior de una empresa que con los recursos disponibles (técnicos, administrativos, ecosistema), pueden entregar productos de software de alta calidad, dentro de estándares mundiales. Para hacer esto posible se deben tener las herramientas tecnológicas y un modelo administrativo de alto nivel que forme parte del proceso de producción.

Watts S. Humphrey desarrolló PSP (Personal Software Process) y TSP (Team Software Process) como el siguiente paso de CMM (Capability Maturity Model) para formar organizaciones con un alto desempeño. PSP y TSP fueron diseñados para proveer lineamientos a los procedimientos y estrategias para el uso de métodos de desarrollo que sirvan al programador y al equipo de trabajo.

⁶ Hernández, Donato; Gutierrez, Heriberto; Canedo, Gerardo. Creación de Equipos de Alto Desempeño, Usando "Team Software Process" (TSP) y Personal Software Process" (PSP).. <http://www.sei.cmu.edu/tsp/>

TSP (Team Software Process) es un conjunto de procesos estructurados que indican qué hacer en cada fase del desarrollo del proyecto y muestra cómo conectar cada fase para construir un producto completo.⁷

TSP proporciona una definición de los procesos del equipo de producción, un marco de trabajo del equipo y un ambiente favorable de administración. TSP sirve para el desarrollo y mantenimiento de equipos de 2 a 20 ingenieros.

TSP está basado en un modelo “secuencial incremental” que divide al proceso de software en un conjunto de ciclos de desarrollo, en donde cada ciclo incluye la producción de software que cumple algunas características de los requerimientos de software.

TSP guía a los equipos en el lanzamiento y la ejecución de los proyectos, lo que permite:

- El rastreo preciso del desarrollo del equipo.
- Reducir los costos de todo el ciclo de vida.
- Duplicar la productividad.
- Consistencia con los horarios en los proyectos.
- Recorte de los costos de mantenimiento.
- Predecir el trabajo.
- Reducir la producción de ingeniería.

1.3.2 TÉCNICAS

Las técnicas de TSP abarcan los objetivos, principios, el enfoque, la lógica de TSP y la conformación de equipos efectivos de trabajo.

⁷ MONTES DE OCA, César. Team Software Process (TSP) Integración de Equipos de Desarrollo de Alto Rendimiento, Carnegie Mellon University, 2004,
http://www.emagister.com/public/pdf/comunidad_emagister/68706010050357686756485568684550-TSP_Conferencia.pdf

Objetivos de TSP

- Integrar equipos independientes de alto rendimiento que planeen y registren su trabajo, establezcan metas y sean dueños de sus procesos y planes.
- Mostrar a los gerentes como monitorear y motivar a sus equipos de trabajo y como ayudarlos a alcanzar su máxima productividad.
- Acelerar la mejora continua de procesos.
- Proveer de una guía para el mejoramiento en organizaciones maduras.
- Ayudar a la organización a establecer una práctica madura y disciplinada de la ingeniería para producir software seguro y fiable.

Principios de TSP

- Los ingenieros conocen todo sobre el trabajo que van a realizar y puede hacer los mejores planes.
- Cuando los ingenieros planean su propio trabajo adquieren un compromiso con el plan.
- El proyecto precisa un seguimiento que requiere planes detallados y datos exactos.
- Sólo las personas que hacen el trabajo pueden coleccionar los datos precisos y exactos.
- Para minimizar el tiempo del ciclo, los ingenieros deben equilibrar su trabajo.
- Para aumentar al máximo la productividad, el enfoque primordial es la calidad.

Enfoque TSP

- Planificar el trabajo antes de adquirir un compromiso o antes de comenzar con este.
- Usar un proceso definido.
- Medir y seguir el tiempo de desarrollo, tamaño y defectos.
- Planificar, medir y seguir la calidad del producto.
- Poner énfasis en la calidad desde el comienzo del trabajo.
- Analizar cada tarea y utilizar los resultados para mejorar el proceso.

Lógica de TSP

Los proyectos de software fallan generalmente por problemas de trabajo en equipo y no por problemas técnicos.⁸ Un problema de las personas es la inhabilidad de los equipos de software para manejar presiones, especialmente las que se encuentran en un calendario de desarrollo agresivo. Por lo general las personas responden a las presiones tomando atajos, usando métodos pobres, herramientas o técnicas.

Las presiones en los proyectos de software surgen de la necesidad de cumplir con un calendario apretado, que puede ser fijado por el administrador, el instructor o los compañeros de trabajo. Los equipos necesitan conocer como trabajar eficientemente y producir productos de calidad, especialmente si se consideran calendarios agresivos.

Equipos efectivos

Un equipo consiste de por lo menos dos personas, quienes trabajan bajo metas y objetivos comunes, donde a cada persona se le asigna roles y funciones para el desarrollo y donde el cumplimiento de la misión requiere de alguna forma de dependencia entre los miembros del grupo.

El equipo es un grupo de personas fuertemente unido que en conjunto es mayor que sus partes. La producción de un equipo es mayor que la de una sola persona que trabaja independientemente. La satisfacción es mayor cuando el trabajo es mejor en equipo que independientemente.

Para crear equipos efectivos solo se requiere de tipos de tareas correctas y condiciones de trabajo. El trabajo a realizarse debe ser importante y el ambiente debe soportar el trabajo en equipo. Los equipos deben cumplir con ciertas condiciones básicas: cohesión, metas, informes y un ambiente común de trabajo.

Para ser eficaces, los equipos deben empezar por:

⁸ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999.

- Definir sus metas.
- Establecer los roles en el equipo.
- Definir una estrategia de desarrollo.
- Definir el proceso.
- Producir un plan de desarrollo global.
- Detallar los planes para cada ingeniero.
- Hacer un análisis de riesgo.
- Acordar mecanismos de comunicación y de información.

1.3.3 PROCESOS TSP⁹

1.3.3.1 Lanzamiento del Proyecto de Equipo (Launching a Team Project)

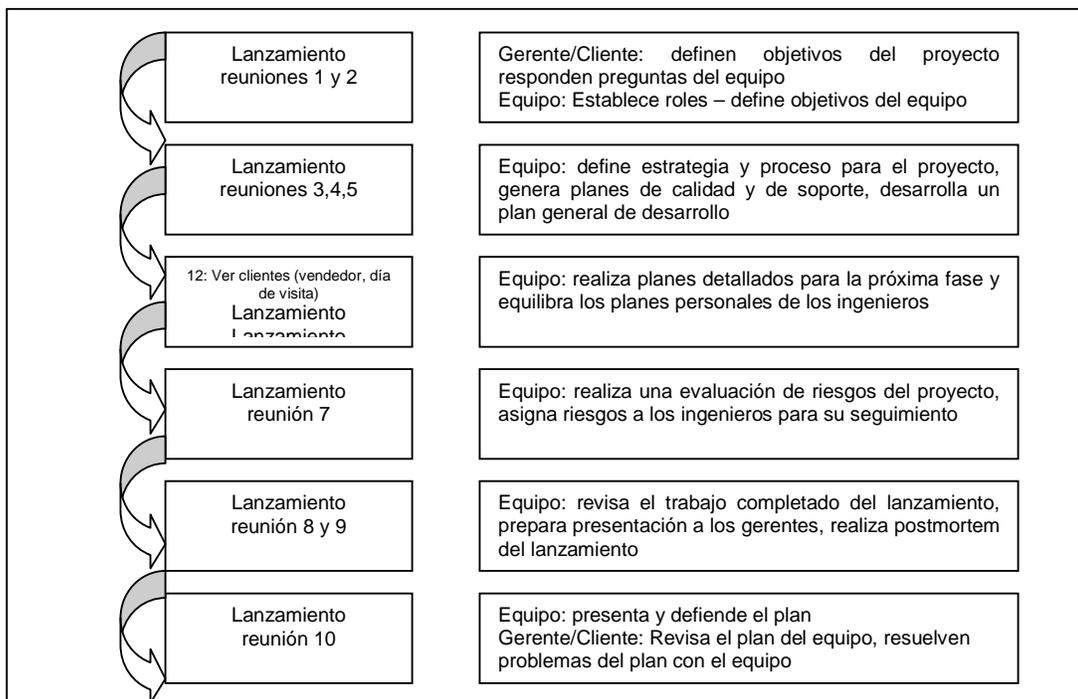


FIGURA 1.1: PROCESO DE LANZAMIENTO EN TSP

El Lanzamiento del Proyecto de Equipo (Launching a Team Project) es el primer paso de TSP que toma 3 o más días, es parte del proyecto y está dirigido por un

⁹ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999.

mentor entrenado en TSP. Primero se requiere de una capacitación de TSP. En el lanzamiento los ingenieros realizan las siguientes tareas:

- Seleccionan los roles del personal.
- Definen sus propios procesos.
- Producen planes individuales y en equipo.
- Equilibran estos planes.
- Evalúan y asignan los riesgos del proyecto.

La figura 1.1 representa las tareas del proceso de lanzamiento en cada ciclo de desarrollo.

- Líder del Equipo
- Administrador del Desarrollo
- Administrador de la Planificación
- Administrador de la Calidad/Procesos
- Administrador del Soporte

Los roles definen las responsabilidades para gestionar el entorno de trabajo de los ingenieros.

La información necesaria para que el instructor asigne los roles al equipo se toma de los datos de las formas de información INFO (Information, ver anexo 1.1) completadas por cada uno de los miembros del equipo.

1.3.3.1.1 Definición de las metas del equipo

Para la formación del equipo es necesario establecer las metas al inicio del proyecto. Los equipos deben revisar y entender las metas para usarlas como guía de su trabajo.

Si las metas son demasiado fáciles existirá poca motivación o esfuerzo. Si las metas se visualizan no realizables puede que exista poca motivación o no exista

motivación. Las metas deben ser agresivas pero realizables para que exista motivación y el equipo completo se esfuerce para conseguir las metas.

Para que el trabajo en equipo sea efectivo se requiere definir metas que guíen el trabajo. Para definir las metas se debe considerar qué sería un resultado relevante para los ojos del cliente.

Las metas definidas deben ser mensurables. Se deben definir las dimensiones de las metas y para ello se mide el desarrollo personal y el desarrollo de las metas.

Antes de cada ciclo se debe examinar el desempeño personal y las metas para mejorar. Con los datos del primer ciclo se definen nuevas metas para los siguientes ciclos.

1.3.3.1.2 Definición de metas por miembro del equipo

Todos los miembros del equipo necesitan especificar las metas mensurables. TSP tiene estándares para las metas de los miembros del equipo y los roles. La primera meta del equipo es trabajar todos en cooperación. El éxito del equipo depende de la contribución de los esfuerzos personales, el soporte a otros miembros del equipo y el trabajo en cooperación para resolver los problemas y los desacuerdos.

Cada miembro debe reportar cada fin de semana su progreso en contraste con el plan en la forma WEEK (Weekly Status Report, ver anexo 1.4).

Al conocer las metas, los miembros del equipo deben establecer planes personales para todo su trabajo. Deben llenar las formas TASK (Task Planning Template, ver anexo 1.2) y SCHEDULE (Schedule Planning template, ver anexo 1.3) para cada ciclo del desarrollo y monitorear el progreso ganado en contraste con el plan todas las semanas.

Los miembros deben escribir los resultados de la inspección de todo el equipo en la forma INS (Inspection Report, ver anexo 1.13), producir un plan de prueba

comprendido y reportar los resultados de la prueba en la forma LOGTEST (Test Log, ver anexo 1.18).

1.3.3.1.3 Definición de metas por rol

Para trabajar en cooperación todo el equipo se debe establecer las metas por rol. Estas metas evitarán que se presenten conflictos por los roles establecidos.

Líder del proyecto

- Construir y mantener un equipo efectivo.
- Motivar a todos los miembros del equipo a trabajar agresivamente sobre el proyecto.
- Resolver todos los problemas entre los miembros del equipo producidos por el líder del proyecto.
- Tener al instructor totalmente informado sobre el progreso del equipo.
- Ser el facilitador de los encuentros del equipo.

Administrador del desarrollo

- Producir un producto superior.
- Usar todas las destrezas y habilidades de los miembros del equipo.

Administrador de la planificación

- Producir un completo, preciso y correcto plan para el equipo y todos los miembros del equipo.
- Reportar de forma exacta el estado del equipo todas las semanas.

Administrador de la calidad/Procesos

- Todos los miembros del equipo usan de forma precisa y apropiada los datos de los procesos TSP.
- El equipo sigue fielmente el TSP y produce un producto de calidad.
- Todas las inspecciones del equipo son moderadas apropiadamente y reportadas.
- Todas las reuniones del equipo son reportadas con precisión y los reportes son puestos en la agenda del proyecto.

Administrador del soporte

- El equipo tiene todas las herramientas apropiadas y métodos de soporte de su trabajo.
- Ningún cambio desautorizado se hace a los productos básicos.
- Todos los riesgos y problemas del equipo son recordados en el sistema de monitoreo de problemas y reportado cada semana.
- El equipo se encuentra para rehusar las metas durante el ciclo de desarrollo.

*1.3.3.1.4 Coordinación del trabajo***Reunión del equipo**

La reunión del equipo es el mecanismo de comunicación principal entre los miembros del equipo. Proporciona el marco para coordinar el trabajo del equipo y el material usado por el equipo. Los líderes informan el progreso en la reunión. Los equipos sostienen su primera reunión semanal durante la fase del lanzamiento. En esta reunión, el primer orden de negocio es decidir donde y cuándo celebrar sus reuniones semanales regulares.

Para las reuniones se sigue el script WEEK. Antes de las siguientes reuniones cada miembro del equipo debe modificar las formas TASK y SCHEDULE y WEEK.

Los datos requeridos

Antes de cada reunión semanal, cada miembro del equipo proporciona una copia personal de la forma WEEK al administrador de la planificación. Del reporte de estos informes, el administrador de la planificación prepara el informe WEEK del equipo y lo distribuye al equipo en su reunión semanal. Durante la fase de lanzamiento los miembros del equipo discuten cómo completar estos formularios y cuando se le proporcionarán al gerente de la planificación.

Inicio del proyecto

Previo a este punto los procesos de la fase de lanzamiento deben ser completados. Para dar inicio se sigue el script STRAT (Development Strategy)¹⁰. Se requiere conocer el Project Notebook, contiene todos los registros del proyecto y el TSP Support Tool son las herramientas de soporte para asistir en la planificación, ingreso de datos y seguimiento del trabajo.

Project Notebook

El propósito del Project Notebook es proveer un camino estándar para guardar toda la información clave del proyecto. El líder del equipo es el responsable de establecer y mantener el Project Notebook, mientras que todos los miembros del equipo son responsables de proveer los materiales necesarios.

TSP Support Tool

El TSP tiene herramientas de soporte para asistir en la planificación, ingreso de datos y seguimiento del trabajo. Con las herramientas se ahorra tiempo y se asegura que los reportes son precisos y completos.

Durante la fase de lanzamiento del proyecto el director guía al equipo siguiendo los pasos descritos en el script LAU1 (Cycle 1 Team Launch) y script LAUn (Cycle n Team Launch)¹¹ dependiendo del ciclo en el que se encuentre.

1.3.3.2 Desarrollo de la Estrategia (Development Strategy)

La estrategia tiene como objetivo minimizar los riesgos para no exceder el tiempo de desarrollo disponible para el proyecto. TSP tiene una estrategia básica que consiste en desarrollar el producto en procesos cíclicos. El número de ciclos dependerá del calendario propuesto por el inspector, pero el contenido de estos ciclos depende de los integrantes del equipo que desarrollarán el producto. Al final de cada ciclo se tiene una versión del producto, al final del primer ciclo se tiene una versión básica y al finalizar los ciclos se tiene el producto final.

¹⁰ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 377.

¹¹ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 368 y 369.

1.3.3.2.1 Creación del diseño conceptual

Es el punto inicial para la creación de un proyecto. Para crear un diseño conceptual se requiere responder las siguientes preguntas:

- Basándose en lo que se conoce, ¿Cómo se puede construir el producto?
- ¿Cuáles son los principales componentes que se necesitan para construir el producto?
- ¿Qué funciones deben proveer estos componentes?
- ¿Cuán grandes deben ser los componentes?

Los componentes que son postulados van a constituir las piezas a ensamblar para el producto final. El diseño conceptual requiere de bastante tiempo lo que es un problema para los ingenieros. Una vez creado el diseño conceptual se puede estimar el tamaño del producto y el tiempo requerido para el desarrollo del mismo.

El administrador del desarrollo guía al equipo en la producción del diseño conceptual. En el primer ciclo el diseño conceptual debe contener todas las funciones que se pretenden desarrollar en los otros ciclos de desarrollo.

1.3.3.2.2 Creación de la estrategia

Para establecer una estrategia se deben considerar los siguientes pasos:

- Definir los criterios de la estrategia.
- Determinar las posibles estrategias alternativas.
- Identificar los riesgos y beneficios de cada estrategia alternativa.
- Hacer una evaluación comparativa de cada alternativa.
- Seleccionar la estrategia de desarrollo.
- Documentar la estrategia seleccionada.

Criterios para establecer una estrategia

Es importante establecer criterios para evaluar las estrategias. El principal objetivo de una estrategia es minimizar los riesgos que se puedan presentar al utilizar un calendario de desarrollo. Se sugieren los siguientes criterios:

- Durante el ciclo 1 el producto provea de una funcionalidad mínima del subconjunto del producto.
- En el ciclo 1 del producto se tiene una base que puede ser fácilmente mejorada.
- Todos los ciclos del producto son de gran calidad y pueden ser fácilmente probados.
- El diseño del producto tiene una estructura modular que permite a los miembros del equipo trabajar independientemente.

Selección de la estrategia de desarrollo

Con los criterios establecidos se procede a proponer una estrategia de desarrollo. El administrador del desarrollo es el encargado de producir el diseño conceptual durante el primer ciclo del desarrollo y de proponer una estrategia inicial.

Se debe considerar que algunos elementos de un producto probablemente sean más largos que lo planificado; así, si se incrementan funciones en el primer ciclo el tiempo planificado se incrementará.

Producción de la estimación preliminar del tamaño

Seleccionada la estrategia, se deben estimar cuantas líneas de Código (LOC) se requieren probablemente en cada función en el diseño conceptual. Esta estimación permite identificar que funciones pueden ser implementadas en el tiempo establecido. Se debe considerar los elementos del programa para establecer las nuevas LOC o cambios de LOC que probablemente se requieran.

Producción de la estimación preliminar del tiempo

Con la estimación del tamaño del programa en LOC, se estima el tiempo probable que se requerirá para desarrollar cada función.

Evaluar los riesgos

El líder del equipo guía en la identificación y evaluación de los principales riesgos del proyecto. Los riesgos pueden ser: H (altos), M (medianos) o L (bajos).

Documentar la estrategia

Al final del desarrollo de la estrategia se llena la forma STRAT (Strategy Form, ver anexo 1.7), para documentar la estrategia.

Modificar el desarrollo de la estrategia

Las modificaciones que se realicen se hacen usando el script STRATn (Cycle n Development Strategy)¹². Se ingresan los componentes ya desarrollados y se definen las funciones añadidas al siguiente ciclo de desarrollo, se señalan cuales de estas funciones van a incorporar módulos ya existentes y se definen nuevos módulos con sus respectivas funciones. Se estiman los tiempos de estas adiciones, en lo posible que no incremente mucho el tiempo disponible para el desarrollo.

1.3.3.2.3 Producción del plan de administración de la configuración

La administración de la configuración de procesos tiene varias funciones claves, entre las cuales se tienen:

- Copias de cada versión y de cada elemento del producto.
- Un registro de todos los cambios realizados durante todo el proyecto
- ¿Quién realizó el cambio?
- ¿Cuándo se realizó el cambio?
- ¿Cuál fue el cambio?
- ¿Por qué se realizó el cambio?

Este plan facilita a los miembros del equipo la realización de cambios del programa. Todos los miembros del equipo deben seguir este plan.

Para la administración de la configuración se crean los siguientes elementos de la Administración de la configuración del Software (Software Configuration Management, SCM):

¹² WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 378.

Identificación de la configuración

Los ítems a ser controlados son:

- Una lista aprobada de los ítems a ser controlados.
- Los requerimientos
- El diseño del producto.
- El código fuente del producto.
- Los materiales de prueba y los resultados de las pruebas.
- Estándares de diseño del producto: estándares de nombramiento de los módulos y del sistema, estándares de interfase, estándares de mensajes, pantallas y librerías re usables en el programa.

Procedimiento de control de la configuración

Para asegurar que no existan conflictos en los cambios no se permite al mismo tiempo a varias personas realizar cambios. Los cambios en el producto deben ser aprobados por el propietario del producto y por la tabla de control de la configuración (Configuration Control Board) antes de que pueda ser realizado.

Tabla de control de la configuración (CCB)

Asegura que todos los productos base y los cambios en estos son propiamente justificados y convenientes para la calidad. La tabla de control de la configuración (Configuration Control Board) debe ser realizada por el administrador del soporte y el administrador de desarrollo.

Respuesta a los cambios de la configuración (CCR)

Para realizar un cambio en el producto se requiere:

- Que el equipo conozca y esté de acuerdo con los criterios del CCB.
- Crear la forma de las respuestas a los cambios de la configuración (Configuration Change Request Form).
- Que el propietario del producto esté de acuerdo con los cambios.
- Que el administrador de la calidad/procesos certifique que los cambios serán desarrollados siguiendo los procesos acordados por el equipo y que los datos que se muestren del desarrollo serán de acuerdo a los criterios de calidad.

- Los cambios son consistentes con las necesidades del cliente o usuario, con el diseño del producto, y la estrategia de desarrollo.
- Los recursos adecuados están disponibles para realizar el cambio.

1.3.3.2.4 Estrategia de re uso

La estrategia de re uso se adopta con el fin de reducir problemas en el código. La estrategia puede reducir el esfuerzo y el tiempo requerido para el desarrollo, cuando se desarrolla código que cumple con múltiples necesidades o se añaden funciones con poca o ninguna modificación.

Durante la fase de desarrollo de la estrategia se siguen los pasos descritos en el STRAT1 (Cycle 1 Development Strategy) y STRATn (Cycle n Development Strategy)¹³.

1.3.3.3 Desarrollo del Plan (Development Plan)

El plan proporciona un contexto para hacer el trabajo y cumplir a tiempo con los compromisos adquiridos. Un plan puede ser simple o complejo, esto depende de la complejidad del trabajo que se va a realizar. Cuando se tiene un plan detallado el trabajo es más eficiente.

Un plan debe ser equilibrado; es decir, todos los ingenieros completan sus tareas planeadas en el orden apropiado y al mismo tiempo. Un plan detallado ayuda al monitoreo y administración del trabajo de los miembros del equipo.

1.3.3.3.1 Proceso de planeación

En la Figura 1.2 se presenta el proceso de planeación. Los pasos a seguirse son los siguientes:

Pasos previos

- Realizar el diseño conceptual durante la fase de desarrollo de una estrategia.

¹³ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 378.

- En la misma fase anterior se estima el tamaño de varias de las partes del diseño conceptual y se decide las partes del desarrollo que van en cada fase. Llenar la forma STRAT.

Producir la forma SUMS

En la fase de planificación se llena la forma SUMS (Size Summary Form, ver anexo 1.8) con los nombres y tamaños estimados de los componentes ha desarrollarse en el ciclo 1. Se identifican los procesos que se deben desarrollar en el ciclo 1 con sus respectivas estimaciones de tamaño y se llena la forma SUMS.

Producir el plan del equipo

Se listan las tareas que se requieren en la construcción del producto identificadas en el paso anterior y se estima su tamaño, entonces se llena la forma TASK. Se estima el tiempo que el equipo requiere para el proyecto cada semana y se llena la forma SCHEDULE. Con las herramientas de TSP se producen las formas TASK y SCHEDULE del equipo.

Producir un plan de calidad

Se crea el plan de calidad, se llena la forma SUMQ (Quality Summary Form, ver anexo 1.12) con las herramientas de TSP. Se determinan los defectos que son introducidos en esta fase. Se usan los criterios de calidad descritos en el Standard Qual. Después el objetivo es eliminar los defectos en cada fase. Finalmente se crea un plan de calidad que concuerde con los criterios.

Crear planes personales

Hacer los planes individuales de los ingenieros. Estimar las horas para todas las tareas y las horas semanales disponibles de cada ingeniero, para asignar las tareas a cada ingeniero por semana. Los ingenieros producen las formas TASK y SCHEDULE personales ajustándose a la forma SCHEDULE del equipo con los tiempos semanales.

Definir los planes de los miembros

Se genera las formas TASK y SCHEDULE para cada miembro del equipo.

Balance del trabajo del equipo

Con las formas anteriores se balancea el volumen de trabajo del equipo. Y se generan nuevas formas con los cambios en las tareas que deben completar cada ingeniero. Cada miembro del equipo revisa sus tareas y tiempos con el fin de balancear las tareas y cualquier cambio de los tiempos se registra en las formas TASK, SUMP (Plan Summary Form, ver anexo 1.11) y SUMQ.

Creación del plan final

Con los planes individuales se crea el plan final del equipo. Se sacan copias de las formas TASK, SCHEDULE, SUMP y SUMQ para entregar a cada ingeniero. Se consolida un plan del equipo y se registran las formas (SUMP, SUMQ, TASK y SCHEDULE) para el equipo y para cada ingeniero.

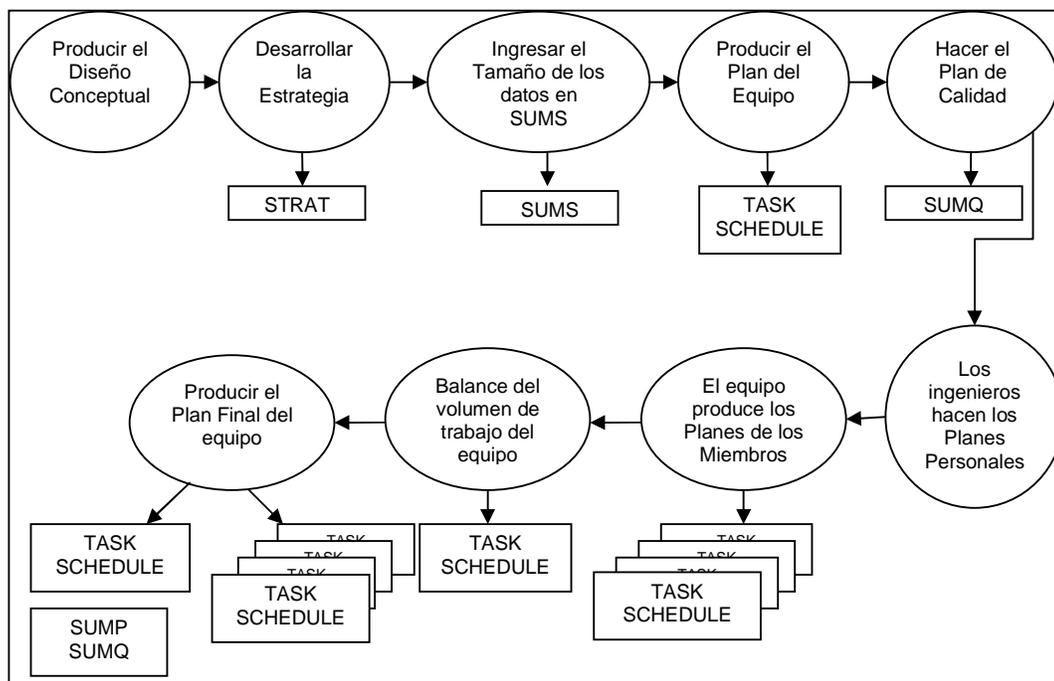


FIGURA 1.2: PROCESO DE PLANIFICACION (HUMPHREY, WATTS. INTRODUCTION TO THE TEAM SOFTWARE PROCESS)

Para tareas no programadas TSP provee la forma SUMTASK (Task Planning Form)¹⁴.

¹⁴ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 436.

1.3.3.3.2 Seguimiento del proyecto

El objetivo es determinar el estado y la calidad del trabajo. A continuación se describen los pasos que se siguen en el monitoreo del proyecto, también presentados en la Figura 1.3:

- Registrar el tiempo en la forma LOGT (Time Recording Log, ver anexo 1.17), con estos datos se modifican las formas TASK y SCHEDULE.
- Ingresar las tareas de la semana cuando estén completas en la forma TASK.
- Generar los datos de las formas TASK y SCHEDULE con las modificaciones de los tiempos en las tareas, las horas de la semana y el estado EV del trabajo. Con los datos anteriores se modifica la forma WEEK y una copia se le da al administrador de la planificación.
- Ingresar los defectos en la forma LOGD (Defect Recording Log, ver anexo 1.16) y la forma SUMP en las fases donde los defectos fueron introducidos y removidos. Se retiene el tipo de defecto y los datos para un posterior análisis.
- Ingresar del tamaño por componente. Una vez desarrollados los elementos del producto se establece su tamaño en la forma SUMS y se completan las formas SUMP y SUMQ.
- Actualizar la forma SUMP con los tiempos, tamaños y datos de los defectos de los ingenieros, para cada reunión, incluyendo datos totales de todos los ingenieros.
- Generar la forma SUMQ con los datos anteriores para cada reunión.
- Producir el resumen del estado del equipo consolidado. Todas las semanas los ingenieros entregan un reporte personal al administrador de la planificación quien consolida esta información y actualiza las formas TASK y SCHEDULE y completa la forma WEEK del equipo. Además se actualiza las formas SUMP y SUMQ con el plan y los datos actuales.

Se completan las formas: SUMT (Time Summary Form)¹⁵, SUMDI (Injected Summary Form)¹⁶ y SUMDR (Defects Removed Summary Form)¹⁷ para colaborar con la consolidación de la forma SUMP.

1.3.3.3.3 Herramientas de apoyo

Son los log, formas y templates necesarios para planificar y monitorear los proyectos. Para la planificación estas herramientas generan automáticamente todas las formas del plan necesarias.

1.3.3.3.4 Plan de calidad

El plan de calidad SUMQ presenta el estado de la calidad para todas las reuniones en el sistema. El plan posee las siguientes secciones:

- Resumen de las proporciones
- Proporción de los defectos libres (PDF)
- Defectos por página
- Defectos por KLOC
- Proporción de los defectos
- Proporción de tiempos de desarrollo
- A/FR
- Proporción de análisis
- Proporción de inspección
- Proporción de defectos introducidos
- Proporción de defectos removidos
- Rendimiento de las fases
- Rendimiento de los procesos

Durante la fase de planificación se siguen los pasos descritos en el script PLAN1 (Cycle 1 Development Plan) y PLANn (Cycle n Development Plan)¹⁸

¹⁵ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 434.

¹⁶ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 421.

¹⁷ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 423.

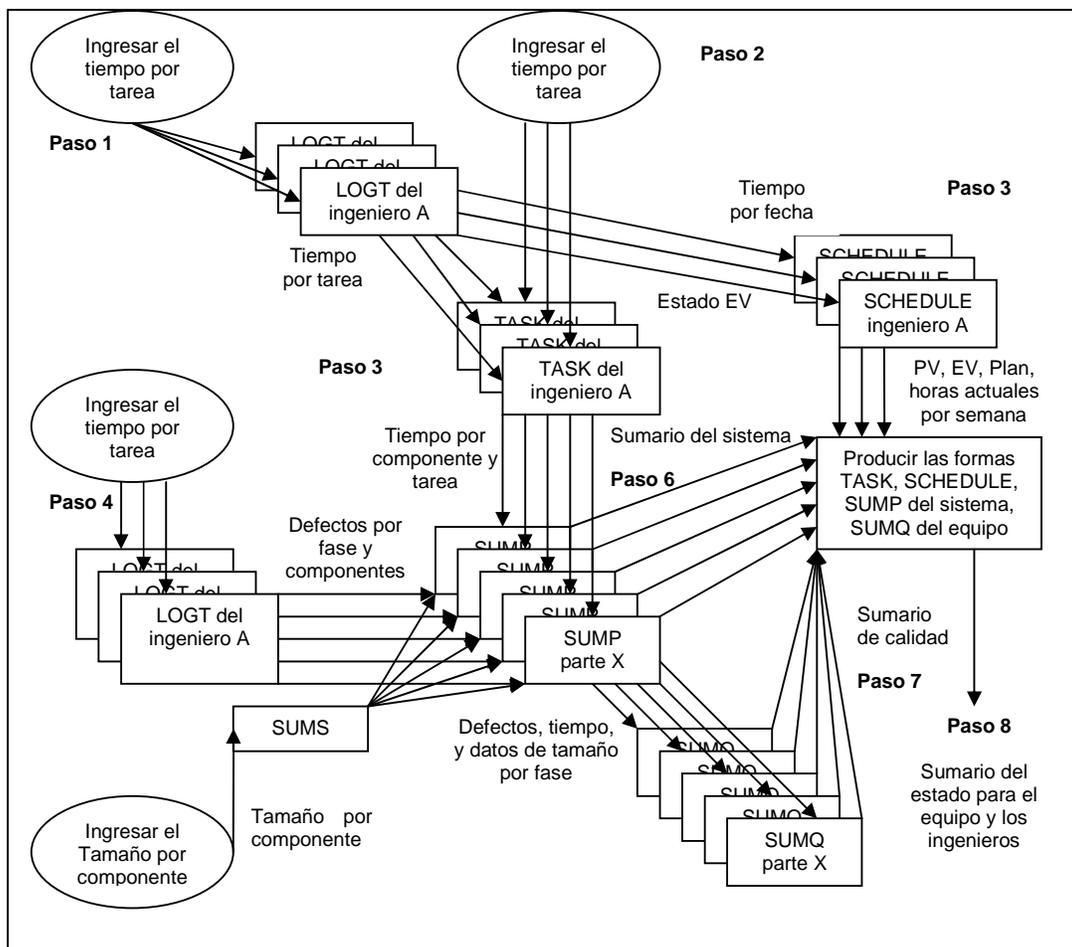


FIGURA 1.3: PROCESOS DEL MONITOREO DEL PROYECTO

1.3.3.4 Definición de Requerimientos (Defining the Requirements)

En la fase de requerimientos el equipo crea la especificación de requerimientos de software (Software Requirements Specification, SRS). El SRS debe presentar una descripción clara y no ambigua de lo que va a hacer el producto, debe presentar los criterios para evaluar que el producto al final hace lo que debe hacer. El SRS es un informe para el cliente que necesita el producto que se va a construir.

El desarrollo de los requerimientos es importante ya que permite revisar las necesidades del cliente y formular preguntas sobre como es el producto, para discutir las respuestas en el grupo de trabajo y definir qué está comprendido y

¹⁸ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 370 y 371.

qué información adicional se requiere para definir un plan de trabajo. Es importante que todos los miembros del equipo participen de la definición de los requerimientos.

Los requerimientos en su mayoría sufren cambios que deben realizarse así detengan el desarrollo del producto. Los cambios se hacen hasta que los usuarios estén de acuerdo con los requerimientos que presenta el SRS, aunque estos cambios signifiquen tiempo y dinero.

1.3.3.4.1 Obtención de los requerimientos

Por lo general es el punto inicial del proceso de requerimientos. Se presenta cuando se interroga a los clientes, usuarios y los accionistas importantes para descubrir cuales son sus necesidades reales. Los puntos principales de la obtención de requerimientos son:

- Evaluar la fiabilidad del sistema.
- Entender los problemas de la organización.
- Identificar los accionistas del sistema.
- Registrar las fuentes de requerimientos.
- Definir el ambiente de operación del sistema.
- Evaluar los problemas del negocio.
- Definir las restricciones del dominio.
- Registrar las razones de los requisitos.
- Un prototipo pobre de los requerimientos entendidos.
- Definir el uso de las situaciones.
- Definir los procesos operacionales.

1.3.3.4.2 Especificación de requerimientos de software

La especificación de requerimientos de software SRS describe un plan para desarrollar un producto y lo que se quiere que ejecute el producto. Los principales tópicos del SRS a considerar son:

- Requerimientos funcionales

- Requerimientos de las interfaces externas
- Limitaciones del diseño
- Atributos
- Otros requerimientos

El principal objetivo es hacer un documento de los acuerdos del equipo sobre las funciones y las interfaces externas. Se establecen los requerimientos identificados en la fase estratégica y se los mantiene por medio del SRS en el diseño.

Producir el SRS normalmente no debe tomar mucho tiempo del desarrollo porque tiene pocas páginas; pero cuando éste toma mucho trabajo se designa a algunos miembros del equipo para que se dediquen a producir el SRS.

1.3.3.4.3 Asignación de tareas de los requerimientos

El administrador del desarrollo debe identificar el trabajo que se requiere para producir el SRS. Divide el trabajo en secciones que se asignan a los miembros del equipo por parte del líder del equipo.

1.3.3.4.4 Documentación de los requerimientos

En la documentación el SRS produce un informe que manifiesta lo que se piensa construir. Para asegurarse de que todos están de acuerdo con el informe se puede considerar definir un caso de uso por cada función a ser usada, que posteriormente servirán para las pruebas del sistema. Una vez terminadas las secciones del SRS se le entrega al administrador del desarrollo quien guarda éste en el borrador SRS y envía una copia a cada ingeniero para que revise.

1.3.3.4.5 Creación del plan de prueba del sistema

El propósito es que las pruebas del sistema estén dirigidas a asegurar que el sistema haga lo que los usuarios en acuerdo supusieron que haría. Los casos de uso proveen de gran cantidad de material para las pruebas que requiere el sistema. Las pruebas deben ser realizadas bajo varias condiciones de error para considerar la usabilidad y los problemas de recuperación.

1.3.3.4.6 Inspección de los requerimientos y el plan de prueba del sistema

En la inspección del SRS se identifican preguntas o desacuerdos y juntos con el equipo se realiza una reunión. El líder de la inspección es el administrador de la calidad/procesos quien sigue el script de inspección INS (Inspection)¹⁹, para registrar los defectos encontrados y el tiempo necesario se usa la forma INS (Inspection Report Instruction, ver anexo 1.13). El propósito es corregir los problemas identificados después de la inspección y antes de empezar con el diseño.

1.3.3.4.7 Modificación de los requerimientos

Se debe modificar el SRS y el plan de pruebas una vez corregidos los problemas que se encontraron en la inspección. El administrador del desarrollo recibe las correcciones y combina éstas en el documento final del SRS, y entrega copias a los ingenieros.

1.3.3.4.8 Revisión del SRS

Los usuarios deben revisar el SRS y acordar que éste describa lo que ellos quieren. En caso de que existan cambios, se deben negociar para estar de acuerdo.

1.3.3.4.9 Requerimientos básicos

El administrador del soporte recibe una copia oficial del SRS que puede ser modificada por el equipo con el uso del CCR. Para este punto el SRS documenta todas las funciones que se planean implementar en este ciclo. Después de esto si el usuario desea cambiar los requerimientos, se debe negociar el costo de los cambios.

Durante la fase de Requerimientos de software se siguen los pasos de los script REQ1 (Cycle 1 Requirements Development) y REQn (Cycle n Requirements Development)²⁰ dependiendo del ciclo.

¹⁹ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 367.

²⁰ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 374 y 375.

1.3.3.5 Diseño con Equipos (Designing with Teams)

El diseño es el proceso creativo de decidir cómo construir un producto. Se tiene que crear una especificación completa y precisa de cómo se va a construir el producto. El diseño define las partes principales del producto, describe como interactúan estas partes y especifica como deben integrarse para obtener el resultado final.

Concluida la fase de requerimientos se definen los procesos de todo el software que corresponde a varios niveles de diseño. El diseño de alto nivel se diferencia del diseño detallado y de la implementación en el alcance y el detalle. El diseño de alto nivel produce una especificación que todos los ingenieros pueden usar en el diseño de cada una de las partes.

Si el diseño de alto nivel es completo y preciso, los ingenieros pueden crear rápidamente el diseño detallado de los componentes. Para esto se requiere conocer las especificaciones funcionales de cada componente, sus interfaces y su comportamiento.

Para producir el producto final los ingenieros necesitan implementar un diseño detallado que defina la estructura lógica, todo el ciclo de inicio y las condiciones para continuar los siguientes ciclos, el estado de la estructura detallada y el estado de las transiciones para cada programa.

El primer diseño es el diseño conceptual creado durante la fase de estrategia, que es la base para desarrollar la estrategia y producir el plan.

1.3.3.5.1 Diseño en equipos

Uso de todo el equipo

Dividir el trabajo es difícil cuando los sistemas de software son grandes y la estructura del mismo no está fija, una forma de resolver este problema es que todo el equipo trabaje en el diseño de la estructura global, o que se identifiquen las tareas del equipo que se pueden hacer mientras uno o dos ingenieros definen la estructura y especifican los componentes del sistema.

Estudio del diseño

Se debe empezar con el estudio de algunas ideas preliminares sobre los probables componentes del producto y sus funciones. Estas ideas son por lo general definidas durante el diseño del equipo, para que los diseñadores del sistema produzcan las especificaciones de los componentes externos y otros ingenieros piensen en alternativas de diseño de los componentes.

Uso de los talentos de todo el equipo

En el diseño del equipo se deben considerar las ideas de todos los miembros. El principal beneficio del equipo es su gran potencial, destrezas y conocimiento. Todos deben aportar con su conocimiento y experiencia para hacer un equipo efectivo.

1.3.3.5.2 Diseño de alto nivel

En el ciclo 1 se crean los conceptos generales del diseño para todos los ciclos, se documenta el diseño para las funciones de los ciclos. En los ciclos posteriores se revisa el diseño general, y si es necesario se documentan las funciones para el siguiente ciclo. Para crear el diseño de alto nivel se siguen los siguientes pasos:

- Decidir la estructura general del producto.
- Nombrar los componentes del producto.
- Asignar las funciones a los componentes.
- Crear las especificaciones externas de los componentes.
- Asignar funciones de caso de uso a los componentes.
- Identificar las tareas de diseño a ser completadas.

Estructura general del producto

La estructura general del diseño se define en el ciclo 1. Primero se define el diseño conceptual de alto nivel, en el que se identifican los componentes, los nombres de las funciones generales y como se relacionan entre si. Para crear el diseño se pueden usar diagramas de flujo, casos de uso o alguna otra metodología. El equipo debe crear la estructura de alto nivel, los nombres de los componentes y la asignación de las funciones a estos componentes.

Para la asignación de las funciones a varios componentes se usa una tabla de identificación durante la definición de la estrategia. La forma STRAT lista cada componente con sus funciones.

Especificación del diseño

El final del diseño de alto nivel es crear varios documentos de diseño que especifican la lógica y estructura del programa del sistema principal. Al definir la lógica se asegura que las funciones, componentes e interfaces estén completas.

Una vez terminada la especificación del diseño del software (Software Design Specifications, SDS), se realiza una revisión y se resuelven los problemas que se hayan encontrado. Después se le entrega al administrador del desarrollo quien incorpora este documento a las copias del equipo.

Plan de pruebas de integración

El plan de pruebas de integración debe crearse mientras se producen las especificaciones del diseño, debido a que las pruebas de integración chequean y verifican todas las interfaces entre los componentes del sistema.

Inspección del diseño

Para la inspección del diseño se usa el script de inspección INS y se documenta el resultado de la inspección en la forma INS. Se deben ingresar los defectos encontrados durante la inspección del SDS y del plan de pruebas de integración en la forma LOGD.

Actualización del diseño

Después de la inspección se resuelven los problemas encontrados en el SDS y/o en el plan de pruebas de integración y se corrige estos documentos para proporcionar éstos al administrador del desarrollo, quien crea y distribuye los documentos finales.

El administrador de soporte pone como base una copia oficial del SDS, que sólo puede ser modificada usando el CCR.

1.3.3.5.3 Estándares de diseño

El administrador de la calidad/procesos es el líder del equipo en la producción de los estándares del diseño y del glosario de nombres. A continuación se describen algunos estándares de diseño:

Convención de nombramiento: Se especifica la estructura de nombramiento y se establece un glosario de nombres. Se define la jerarquía en el programa del tipo de nombres de acuerdo al sistema, producto, componente, módulo u objeto. Se definen las convenciones a usar en el programa, archivo, variable, los nombres de parámetros, los procedimientos a establecer, controles y el cambio de nombres.

Formatos de interfase: Define el formato y el contenido de las interfases de los componentes. Incluye la definición de que parámetros son variables, error u otra condición.

Mensajes del sistema y de error: Establecer formatos estándar y procedimientos para el sistema y mensajes de error. De manera que los mensajes sean re usables y fáciles para desarrollar.

Estándares de defectos: Se pueden usar los estándares de PSP que se presentan en la tabla 1.1:

Num. Tipo	Nombre del Tipo	Descripción
10	Documentación	Comentarios, mensajes.
20	Sintaxis	Ortografía, puntuación, tipos, formatos de la instrucción.
30	Figura, paquete	Cambio de dirección, la biblioteca, el control de la versión.
40	Asignación	Declaración, nombres duplicados, alcance, límites.
50	Interfase	Llamadas a procedimientos y referencias, I/O, formatos de usuario.
60	Verificación	Mensajes de error, chequeos inadecuados.
70	Datos	Estructura, contenido.
80	Función	Lógica, indicadores, bucles, recursión, computación, defectos de función.
90	Sistema	Configuración, cronómetro, memoria.
100	Ambiente	Diseño, compilación, pruebas, soporte de problemas del sistema.

Tabla 1.1: Estándares de defectos PSP²¹

²¹ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 126.

Contador de LOC: Por lo general se define en la fase de implementación, pero debe definirse antes de empezar con el diseño.

Estándares de representación del diseño: Se define el diseño del producto que será usado como estándar para evitar problemas en la implementación y las pruebas.

Estándares de representación del diseño

La representación del diseño es importante ya que éste contiene todas las especificaciones necesarias para el equipo. Para crear el diseño del equipo se usan las cuatro plantillas de diseño de PSP: la plantilla del escenario a operar, la plantilla de especificación funcional, la plantilla de especificación del estado y la plantilla de especificación lógica.

Casos de uso o escenarios operacionales PSP

Un escenario describe el programa externamente, el comportamiento dinámico visible para describir la secuencia de acciones de entrada y el resultado que se obtiene como respuesta del sistema a estas acciones. Al crear los escenarios se presentan preguntas del diseño y problemas de usabilidad. Los escenarios pueden ser usados para plantear condiciones de prueba.

Los escenarios se crean durante el diseño de alto nivel, y permiten especificar la secuencia de pruebas de cada función clave del sistema.

Análisis de la máquina de estados

Este análisis permite identificar problemas lógicos difíciles de detectar. El análisis de estados permite entender de verdad como trabaja el programa.

Producir un diseño preciso

Para producir un diseño preciso se requiere de tiempo, pero permite ahorrar tiempo en la implementación y en las pruebas. Cuando el diseño es superficial con frecuencia se tienen problemas lógicos básicos que no son percibidos por un solo ingeniero y que para ser resueltos requieren volver a la implementación.

1.3.3.5.4 Diseño para re usar

Con la definición de estándares para re usar por el equipo se aprovecha el tiempo del que disponen los ingenieros y se incrementa su productividad. El problema en el diseño para re usar es la definición de interfases estándares y convenciones a llamadas, establecer estándares de documentación, producir productos de alta calidad y proveer soporte a la aplicación.

Re usar interfases estándares

La base de éstas interfases es que deben ser autosuficientes y aisladas. Los criterios que sugiere TSP son: que el diseño tenga poco acoplamiento y alta cohesión. Los principales estándares de re uso son las interfases de retorno a una llamada.

Al desarrollar partes re usables para un proyecto simple se debe considerar la ventaja de una arquitectura framework común y un conjunto único de estándares, que permite a los ingenieros compartir partes e incrementar sus habilidades para identificar partes comunes y rutinas.

Re usar estándares de documentación

Usar estándares de documentación ahorra tiempo y esfuerzo a los usuarios. Se requiere una lista de partes re usables con una especificación completa de cada comportamiento de la parte externa. Una buena idea es comentar las fuentes del programa para cada parte re usable.

Re usar partes con calidad

La calidad del diseño y del código se visualiza cuando el producto es claro, con limpieza, con comentarios frecuentes y precisos, y materiales de diseño completos. Para re usar un programa existente se deben considerar los siguientes aspectos:

- ¿El programa tiene un correcto funcionamiento?
- ¿Es conveniente la interfaz del programa para la nueva aplicación?
- ¿Es el desempeño del programa apropiado para la nueva aplicación?

- ¿Están todos los materiales disponibles: código fuente, casos de prueba, datos de prueba e instrucciones de aplicación?
- ¿El programa ha sido producido con estándares apropiados, como el nivel de lenguaje, estándares de codificación, estándares de archivo y nombramiento, mensajes y estándares de ayuda?
- ¿El programa es de alta calidad comprobable?

Si una de estas preguntas es no, adoptar el programa para re usar tomará una cantidad considerable de tiempo y esfuerzo.

Soporte de la aplicación

Para TSP, el administrador del soporte actúa como el partidario del re uso y es el responsable del soporte de las partes en re uso y del mantenimiento de los registros sobre estas partes.

1.3.3.5.5 Diseño para usabilidad y pruebas

Para producir productos usables se produce escenarios para cada función clave del usuario. Cuando no se esta seguro del trabajo que debe realizar una función se puede revisar estas funciones con una aplicación experta o hacer un prototipo que demuestre el trabajo de la función.

Un diseño completo y un conjunto de escenarios amplio pueden acelerar la planificación de pruebas y mejorar la efectividad de la pruebas.

1.3.3.5.6 Revisiones e inspecciones del diseño

Las revisiones e inspecciones del diseño pueden ser muy efectivas para mejorar la calidad del diseño. Para una efectiva inspección del diseño se necesita una buena documentación del diseño. Para una revisión del diseño personal o una inspección del diseño se debe hacer un análisis completo del diseño.

Durante la fase de diseño se consideran los script DES1 (Cycle 1 Design) y DESn (Cycle n Design)²².

1.3.3.6 Implementación del Producto (Product Implementation)

1.3.3.6.1 Diseño del criterio del producto terminado

Para proceder con la implementación del producto se debe tener el diseño de alto nivel completo.

Niveles de diseño

Cuando el sistema es largo el diseño de alto nivel debe considerarse para cada subsistema o componente, se deben terminar las especificaciones externas para los componentes de los subsistemas, el diseño detallado y la lógica de alto nivel de cada uno. Cuando se finalice este trabajo se tiene una especificación externa para el nivel más bajo del sistema, entonces se empieza con la implementación. Cuando el sistema es grande, los niveles sucesivos del sistema pueden ser: Sistema, Subsistema, Producto, Componente y Módulo. Los módulos son elementos básicos del sistema que se implementan directamente y su tamaño es pequeño. Los módulos pueden contener objetos de bajo nivel que son de tamaño pequeño, que están desarrollados, son re usables o son librerías disponibles.

Cuando son sistemas grandes algunas veces se implementa elementos del producto antes de finalizar con el diseño de otros productos, siempre que no se este obligado a terminar con el diseño para empezar con la implementación.

Implementación paralela

Cuando es grande el número de miembros del equipo, se puede empezar con la implementación de algún componente si ya se tienen las especificaciones externas del mismo.

²² WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 361 y 362.

1.3.3.6.2 Implementación

Planificación de la implementación

La planificación de la implementación empieza con la revisión del trabajo realizado y la asignación del trabajo a los miembros del equipo considerando las habilidades y destrezas que poseen. Luego cada uno planifica su trabajo de implementación con una estimación del tiempo que requiere para cumplir con éste. Con estas consideraciones se actualiza las formas TASK y SCHEDULE.

Revisión del diseño

Con el diseño detallado para el desarrollo de los módulos del programa los ingenieros empiezan con un programa y hacen pruebas de unidad antes de seguir con el siguiente programa. Luego se hace una revisión personal del diseño de cada módulo u objeto para chequear los ciclos y la lógica señalada en tablas o el análisis de la máquina de estados.

Desarrollo de pruebas

En el desarrollo de pruebas por lo general se encuentran más problemas de diseño que en la inspección de diseño o en las pruebas de unidad. El desarrollo de las pruebas debe seguir el plan de pruebas.

Inspección del diseño detallado

Los datos de la inspección deben ingresarse en las formas INS y los defectos en la forma LOGD.

Codificación y revisión del código

Después de la inspección del diseño se codifica el módulo y se revisa el código para encontrar defectos.

Inspección del código

Con esto se chequean los defectos del código y se establece si se tiene problemas fuertes o si los datos son malos. También se determina la calidad del programa considerando los siguientes criterios de calidad:

- El tiempo ocupado en el diseño debe ser mayor que el tiempo de codificación.
- El tiempo ocupado en la revisión del diseño debe ser mayor que el 50% del tiempo de diseño.
- El tiempo ocupado en la revisión de código debe ser mayor que el 50% del tiempo de codificación y preferiblemente más grande que el 75%.
- Se encontró por lo menos el doble de defectos en la revisión del código que en la compilación.
- Se encontró más de tres defectos por hora de revisión.
- Se revisó que la proporción es menor que 200 LOC por hora.

Pruebas de unidad

Después de la inspección del código se corren las pruebas de unidad. Se usan los materiales de prueba ya desarrollados y se sigue el plan de prueba creado en el diseño detallado.

Revisión de la calidad de los componentes

El administrador de la calidad/procesos es el encargado de determinar la calidad de los componentes dependiendo de los criterios de calidad que el equipo estableció.

Descarga de componentes

Completo el componente se pasa a la revisión de la calidad del mismo, y se ingresa al sistema base.

1.3.3.6.3 Estándares de implementación

Definir estándares permite ahorrar tiempo al equipo. La definición de estándares le corresponde crear al equipo, estos estándares se suman a los estándares definidos durante la fase de diseño. Se pueden considerar los siguientes estándares:

Revisión de los estándares

Se revisan estándares ya propuestos para ser usados. Se deben producir buenos estándares en poco tiempo. Se analizan los estándares definidos en la fase de

diseño para garantizar que son aptos y serán usados de manera apropiada. Se analiza el glosario de nombres, se chequean los nombres de los componentes y de los subelementos, se analizan las variables compartidas, parámetros, y archivos de nombres para garantizar la consistencia, se chequean las interfaces estándares y los mensajes para asegurar que todas las interfaces externas e internas y los mensajes se han definido.

Estándares de codificación

El equipo debe acordar en codificar con los mismos estándares, para asegurar consistencia y facilitar las inspecciones. Además facilita compartir código entre los miembros del equipo y disminuir tiempo en la implementación

Estándares de tamaño

TSP sugiere un contador de páginas para medir el tamaño de los documentos que se tienen, como: el SRS, el SDS, el documento del diseño detallado, pantallas y reportes, bases de datos, mensajes, scripts de pruebas y materiales de prueba.

Estándares de defectos

La razón para categorizar los defectos es ayudar el análisis y la implementación de los procesos a desarrollar.

Prevención de defectos

Para prevenir los defectos se deben entender las causas de los mismos. Las causas de los defectos se pueden categorizar en cuatro grupos:

- Educación: aprender más sobre el lenguaje, ambiente o aplicación.
- Comunicación: resolver los procesos.
- Transcripción: usar mejores herramientas.
- Descuido: pulir los procesos y usar mejores métodos.

1.3.3.6.4 Estrategia de implementación

La estrategia de implementación debe ser consistente con la estrategia de diseño, para evitar problemas en la implementación y en las pruebas.

Análisis

Se analiza que las funciones presenten los resultados esperados para asegurar que el programa cumple las expectativas. Primero se analiza que las partes del sistema que no tienen partes subordinadas cumplan con las especificaciones externas para continuar al siguiente nivel.

Al seguir una estrategia de lo particular a lo general, la implementación debe empezar por los objetos del nivel más bajo para continuar con los del siguiente nivel. Esto permite identificar temprano los problemas para resolverlos antes de continuar con la implementación. Esta estrategia facilita el re uso.

Re uso

Al seguir prácticas o estándares de implementación los programas son fáciles de entender y re usar.

Pruebas

Antes de la implementación se debe revisar que la estrategia de desarrollo asegure que la implementación del programa este de acuerdo con los planes de prueba.

*1.3.3.6.5 Revisiones e inspecciones de la implementación***Defectos randómicos**

Los errores en la transcripción que resultan de errores de digitalización son considerados defectos en la implementación.

Impacto en las pruebas

Los defectos randómicos aún cuando no son muy importantes influyen en el tiempo de producción del sistema.

Dificultad de pruebas exhaustivas

Los defectos randómicos pueden hacer que el programa tenga comportamientos ilógicos. Con las revisiones e inspecciones se detectan estos errores para hacer más eficientes las pruebas.

Inspecciones de diseño para programas seguros

En programas seguros es difícil encontrar defectos de diseño. Para producir programas con calidad se debe producir de forma completa; es decir: completar, revisar, inspeccionar y resolver los documentos de diseño antes de empezar a codificar, caso contrario se debe reinspeccionar o cambiar el diseño.

Durante la fase de Implementación se considera los scripts IMP1 (Cycle 1 Implementation) y IMPn (Cycle n Implementation)²³.

1.3.3.7 Pruebas de Integración y del Sistema (Integration and System Testing)

Durante las pruebas se deben resolver los defectos encontrados, pero la mayoría de los defectos deben resolverse antes de las pruebas. Un producto con una calidad pobre toma mucho tiempo en las pruebas. La calidad de un producto se determina cuando se desarrolla.

1.3.3.7.1 Estrategia de pruebas

Durante las pruebas se identifica la calidad de un producto. Las actividades de las pruebas son:

- Usar las partes desarrolladas y las pruebas de unidad para construir el sistema.
- Pruebas de integración del sistema para verificar que se construyó apropiadamente, que todas las partes están presentes, y que funcionan juntas.
- Pruebas del sistema para validar que el producto hace lo que se requiere que haga.
- Identificar los módulos o componentes con pobre calidad para que el administrador de la calidad/procesos valore y depure.
- Identificar componentes con pobre calidad que siguen molestando después de ser depuradas y retornan al administrador de la calidad/procesos para que sean nuevamente corregidas o remplazadas.

²³ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 365 y 366.

1.3.3.7.2 Construcción y estrategia de integración

El proceso de construcción tiene como objetivo asegurar que todas las partes necesarias están presentes para ensamblar el sistema.

La estrategia de integración define el método para las pruebas de integración, estas pruebas se refieren a chequear que todos los componentes estén presentes y que todas las llamadas y otras interacciones trabajan bien. La estrategia de integración depende principalmente de la estrategia de desarrollo, pero también depende del trabajo que se ha realizado hasta el momento.

1.3.3.7.3 Desarrollo de pruebas

Durante el desarrollo de las pruebas se pueden presentar los siguientes aspectos:

- Pruebas de integridad de la construcción.
- Procedimientos de integración para pruebas de todas las interfases bajo condiciones normales o de error.
- Los materiales de las pruebas de integración, para sistemas pequeños las pruebas de interfaces son conducidas manualmente.
- Los materiales de pruebas del sistema, estos deben probar las funciones del sistema bajo condiciones normales y de presión.
- Un informe claro de los resultados esperados de todas las pruebas.

1.3.3.7.4 Construcción

Para la construcción de las pruebas se tienen los siguientes pasos:

- Chequear todos los componentes necesarios para asegurar que están disponibles y encontrar su dependencia con los requerimientos.
- Revisar los ítems suministrados para la construcción e identificar alguna parte que falte.
- Chequear el propósito de la construcción para establecer consistencia e integridad.
- Construcción del producto.

1.3.3.7.5 Integración

Las tareas de la integración son las siguientes:

- Chequear la integridad del producto construido.
- Ejecutar las pruebas de integración.
- Registrar los datos de las pruebas en el LOGTEST.
- Si se encuentran defectos, el dueño del producto registra los defectos.

En el caso de encontrar defectos se deben resolver e inspeccionar.

1.3.3.7.6 Pruebas del sistema

Para las pruebas del sistema se sigue el plan de pruebas del sistema. Durante la integración de los componentes es preferible ir resolviendo los defectos en paralelo para no detenerse.

1.3.3.7.7 Pruebas de regresión

Ejecutar las pruebas en los ciclos de desarrollo posteriores para asegurar que el producto continúa funcionando. Se deben completar las formas LOGD, LOGT y LOGTEST.

1.3.3.7.8 Estrategia de pruebas del sistema

Durante las pruebas del sistema se deben responder las siguientes preguntas:

- ¿El sistema hace apropiadamente las funciones que se supone que realizaría?
- ¿El sistema integrado cumple con sus metas de calidad?
- ¿El sistema llega a operar apropiadamente bajo condiciones normales?
- ¿El sistema llega a operar apropiadamente bajo condiciones adversas?

Al contestar las preguntas se deben chequear un número de áreas claves. Es decir, se debe chequear si ¿se puede instalar el sistema?, ¿Inicia de manera apropiada?, ¿Se desarrollaron todas las funciones que están en los requerimientos?, ¿Se recupera a fallos de hardware o de poder?, ¿Su desempeño es adecuado?. Para el desempeño se deben evaluar los tiempos de respuesta, el

rendimiento, y la capacidad. Se debe conocer si el sistema es usable, si los usuarios encuentran conveniente el sistema, o si tienen dificultad para ejecutar las tareas y si responde normalmente a cuestiones de operación.

1.3.3.7.9 Plan de pruebas

Empieza con la creación de un diseño de pruebas conceptual, estimación de tiempos de los materiales de prueba que se desarrollan, se requiere de un plan de pruebas para la construcción, pruebas de integración y actividades de pruebas del sistema.

Al final de la planificación de pruebas se debe tener los siguientes aspectos:

- Una lista de todos los pasos de prueba a ser ejecutados.
- Los materiales de soporte requeridos para cada prueba.
- Los resultados que las pruebas producirán.
- Una estimación del tiempo para ejecutar un defecto libre, los defectos a ser encontrados y un tiempo total de cada prueba.
- Una estimación del trabajo requerido para desarrollar cada ítem del plan de pruebas.
- Una lista de todos los materiales de soporte de pruebas requeridos y en que prueba son requeridos.
- Una lista de los objetivos para todas las pruebas.
- Una lista de cuan largos se espera que sean los materiales
- Una lista de cuanto tiempo es probable que tome su desarrollo.
- Una lista de quien desarrollará cada ítem del soporte de pruebas.
- Una lista de cuando serán completadas las tareas a desarrollarse.

1.3.3.7.10 Registro y medición de datos de prueba

Los datos de las pruebas se deben registrar en LOGD y LOGT, además se deben responder las siguientes preguntas:

- ¿Cuánto tiempo tomará ejecutar esta prueba?
- ¿Cuántos defectos son encontrados?

- ¿Se requiere de un manual de intervención o puede ser archivado con otras pruebas?
- ¿Se chequeo bien esta prueba?

Al responder las preguntas se debe registrar los datos sobre el tiempo de ejecución de la prueba, el número de defectos encontrados y las condiciones de la prueba.

1.3.3.7.11 Documentación

Un sistema generalmente necesita de un extenso número de documentos que cubren la instalación, mantenimiento, pruebas y las necesidades de marketing. Los miembros del equipo que se asignan a las pruebas o a la documentación, dependen de la calidad y el contenido funcional del producto. Durante el primer ciclo es mejor asignar mayor número de miembros a las pruebas, durante los ciclos posteriores ir incrementando el número a la documentación.

Características de un documento:

- Organización
- Terminología
- Contenido
- Precisión
- Legibilidad
- Comprensibilidad

Durante la fase de Prueba se consideran los scripts TEST1 (Cycle 1 Integration and System Test) y TESTn (Cycle n Integration and System Test)²⁴.

1.3.3.8 Postmortem (Postmortem)

Conocer como fue desarrollado el software es una habilidad para los ingenieros, quienes alcanzan éxito en una competencia de software al utilizar una disciplina en el uso de las habilidades personales. La subsistencia de esta tecnología y para

²⁴ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 379 y 380.

satisfacer las necesidades más desafiantes del futuro, se debe ver cada proyecto como una oportunidad de mejorar.

El postmortem provee de una estructura de aprendizaje y mejoramiento, que permiten identificar oportunidades de mejoramiento para los siguientes ciclos de desarrollo o para otros proyectos.

La clave del mejoramiento exitoso es enfocarse en los cambios pequeños, ya que muchos cambios pequeños son vistos como un gran cambio significativo. Y aprender a mejorar continuamente en el desempeño personal.

1.3.3.8.1 Revisión de los datos del proceso

El administrador de la calidad/procesos guía al equipo en la revisión de los datos del ciclo y en completar las formas SUMP y SUMQ del producto y sus componentes. El objetivo de la revisión es:

- Examinar los datos que el equipo y los miembros del equipo hicieron.
- Identificar donde trabajaron los procesos y donde no lo hicieron.
- Comparar el desempeño del equipo con sus metas y planes.
- Identificar áreas problema y necesidades de mejora.
- Delegar el mejoramiento de procesos y preparar PIPs.

1.3.3.8.2 Revisión de la calidad

Es la comparación del desempeño del equipo y de cada ingeniero con el plan de calidad. Se analizan los datos de los defectos del equipo y se evalúa el grado con el que el equipo produce un producto de calidad. Si el producto presenta uno o dos defectos en las pruebas del sistema, se evalúa donde los procesos tienen fallas pequeñas y que se puede hacer para mejorar éstas en un futuro.

Se prepara el PIPs con las sugerencias de mejoramiento. Se evalúa el desempeño del equipo y de los miembros otra vez con las metas y se establece el inicio del ciclo de desarrollo, identificadas las áreas que tienen prioridad de mejorar.

1.3.3.8.3 Evaluaciones del rol

Se evalúan los roles del equipo enfocándose en los objetivos cumplidos. Se considera las siguientes cuestiones:

- ¿Qué trabajó?
- ¿Dónde estuvieron sus problemas?
- ¿Dónde está el ambiente de mejora?
- ¿Qué metas de mejora tienen sentido para el siguiente ciclo de desarrollo del proyecto?

Cada miembro del equipo prepara la forma PEER (Team and Peer Evaluation, ver anexo 1.20). Aquí se tiene una visión del desempeño del equipo y de los roles, para sugerir donde los roles y las tareas pueden ser mejoradas en un futuro.

1.3.3.8.4 Preparación del reporte del ciclo

El reporte del ciclo describe lo que se produce, los procesos que se usaron y los roles que se desempeñaron por cada miembro del equipo. Describe qué se trabajó y qué no se trabajó, cómo mejorar para la siguiente vez, el desempeño del equipo con respecto a sus responsabilidades y el desempeño personal como desarrollador y como el rol que se tuvo. En el informe del reporte se debe enfatizar sobre aprender lecciones y como mejorar en el futuro. También se hace una comparación del desempeño con ciclos de desarrollo anteriores y se destacan algunas tendencias.

1.3.3.8.5 El reporte del ciclo

El reporte del ciclo se debería enfocar en los siguientes puntos:

- Tabla de contenidos
- Sumario
- Reporte de Roles (Dirigencia, Desarrollo, Planificación, Proceso, Calidad y Soporte).
- Reporte de los ingenieros

De los dos primeros puntos se encarga el líder del equipo.

1.3.3.8.6 Reporte de los roles

Se escriben dos secciones: cómo evaluar el desempeño de un rol y cómo se piensa del desempeño del equipo con respecto a las perspectivas de los roles. Se discute que debe operar un rol asignado, que trabajó, que no trabajo, y cómo el rol podría operar de mejor manera la siguiente vez. El objetivo es hacer una guía para que la próxima persona se desempeñe mejor en el rol.

1.3.3.8.7 Reporte de los miembros del equipo

Los ingenieros deben reportar su desempeño personal sobre el desarrollo de tareas. Se debe considerar el desempeño personal planificado y la calidad del trabajo realizado, enfocándose en mejorar el trabajo la siguiente vez. También se incluye una o dos áreas para mejorar en el siguiente ciclo del proyecto.

1.3.3.8.8 Producción del reporte

El líder del equipo guía en la definición de los contenidos del reporte, asignando tareas a los ingenieros. Una vez completadas las tareas el líder une y distribuye el reporte del proyecto al equipo. El líder del equipo y el guía del administrador de la calidad/procesos revisan y corrigen el reporte que posteriormente el líder hará llegar al director de proyecto y a todos los miembros del equipo.

Durante la fase de Postmortem se consideran los scripts PM1 (Cycle 1 Postmortem) y PMn (Cycle n Postmortem)²⁵. Concluida esta fase se chequean todos los puntos anteriores.

²⁵ WATTS S. Humphrey, Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999. Pág. 372 y 373.

1.4. RUP

El Proceso Unificado de Desarrollo (RUP) es un método de desarrollo que se obtuvo del estudio y experiencia de Ivan Jacobson, Grady Booch y James Rumbaugh. RUP combinado con “buenas prácticas” es un ciclo iterativo es un método de desarrollo dirigido por el riesgo, dentro de un marco consistente y bien documentado.

El proceso unificado de desarrollo divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito. Estas fases son: inicio, elaboración, construcción y transición.

RUP se compone de flujos de trabajo (procesos de trabajo y procesos de soporte) que involucran un conjunto de actividades necesarias para transformar los requisitos del usuario en un producto. Los procesos de trabajo son: modelamiento del negocio, requerimientos, análisis y diseño, implementación, pruebas y distribución. Los procesos de soporte a los procesos de trabajo son: gestión de configuración y cambios, gestión de proyectos y ambiente. Estos permiten administrar el equipo de desarrollo, los artefactos que produce y el entorno en el que se desenvuelve.

RUP utiliza el Lenguaje Unificado de Modelado (UML) para visualizar, especificar, construir y documentar los artefactos que se requieren en los flujos de trabajo.

RUP pretende implementar las mejores prácticas en ingeniería de software: un desarrollo iterativo del software, la administración de requerimientos, el uso de arquitecturas basadas en componentes, el modelamiento visual del software. La verificación de la calidad del software y el control de cambios.

RUP es un proceso de desarrollo dirigido por casos de uso, centrado en la arquitectura, Iterativo e Incremental.

1.4.1 PROCESO UNIFICADO DIRIGIDO POR CASOS DE USO

En un proceso unificado basado en el modelo de casos de uso se crean modelos de diseño e implementación que llevan a cabo los casos de uso. En las pruebas se garantiza que los componentes del modelo de implementación implementan correctamente los casos de uso.

El proceso de desarrollo es dirigido por casos de uso cuando avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan y sirven de base para construir los casos de prueba. Los casos de uso se desarrollan junto con la arquitectura del sistema, ambos maduran según avanza el ciclo de desarrollo.

1.4.2 PROCESO UNIFICADO CENTRADO EN LA ARQUITECTURA

La arquitectura de un sistema de software se describe mediante diferentes vistas del sistema. La arquitectura es una vista del diseño completo con las características más importantes resaltadas.

La arquitectura es la forma de un producto y los casos de uso son la función del producto, por lo tanto hay interacción entre la arquitectura y los casos de uso. Los casos de uso deben encajar en la arquitectura y esta debe permitir el desarrollo de todos los casos de uso requeridos.

1.4.3 PROCESO UNIFICADO ITERATIVO E INCREMENTAL

Para desarrollar un producto de software se requiere de un gran esfuerzo, por lo que es práctico dividir el trabajo en partes pequeñas o mini proyectos denominadas iteraciones que deben seleccionarse y ejecutarse de una forma planificada.

En cada iteración se identifican y especifican los casos de uso relevantes, se crea un diseño utilizando la arquitectura seleccionada como guía, se implementa el diseño mediante componentes que deben satisfacer los casos de uso, y se tratan los riesgos críticos.

Cada iteración o mini proyecto esta compuesta por sus respectivos flujos de trabajo (requisito, análisis, diseño, implementación, prueba).

Las claves del Proceso Unificado para el desarrollo software son: que el sistema esté dirigido por casos de usos, se centre en una arquitectura y tenga un desarrollo iterativo e incremental. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones y los casos de uso definen los objetivos y dirigen el trabajo de cada iteración.

1.4.4 DESARROLLO BASADO EN COMPONENTES

La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o que se desarrollen y maduran sus componentes.

1.4.5 UML

Unified Modeling Language (UML) es un lenguaje de modelado que permite la representación gráfica de un sistema. Permite visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. Cubre tanto aspectos conceptuales (procesos de negocio, funciones del sistema) como cosas concretas (clases, esquemas de bases de datos, componentes reutilizables).

1.4.5.1 Diagramas de UML

Un diagrama es la representación gráfica de una colección de elementos de modelado. Dentro del UML existen 9 diagramas estándar que se dividen en vistas estáticas y vistas dinámicas.

Vistas Estáticas

- *Diagramas de casos de uso*: Describen la funcionalidad del sistema vista por los usuarios. Tiene como objetivo especificar el contexto del sistema, capturar los requerimientos, validar la arquitectura del sistema, dirigir la implementación y generar los casos de prueba.

- *Diagramas de clases:* Muestran el conjunto de clases e interfaces con sus relaciones, el diseño del sistema orientado a objetos. Se construye y refina a través del desarrollo del sistema. Tiene como objetivo nombrar y modelar conceptos en el sistema, especificar colaboraciones y esquemas lógicos de bases de datos.
- *Diagramas de objetos:* Se realizan para captura instancias y enlaces. Se construye durante el análisis y diseño. El objetivo de este diagrama es ilustrar estructuras de datos/objetos y especificar una instantánea del sistema.
- *Diagramas de componentes:* Captura la estructura física de la implementación, se construye como parte de la especificación arquitectónica. El objetivo de este diagrama es organizar el código fuente, construir un release ejecutable, especificar una base de datos física.
- *Diagramas de despliegue:* Sirven para describir la arquitectura física del sistema durante la ejecución, en términos de: procesadores, dispositivos y componentes de software y la topología del sistema, la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. El objetivo de este diagrama es especificar la distribución de componentes, identificar cuellos de botellas respecto al desempeño.

Vistas Dinámicas

- *Diagrama de secuencia:* Permiten mostrar la interacción entre las instancias del modelo en lo que se refiere al comportamiento dinámico orientado a la secuencia en el tiempo. La creación de los diagramas de secuencia depende de la formulación de los casos de uso. El objetivo de este diagrama es modelar el flujo de control, ilustrar escenarios típicos.
- *Diagramas de Colaboración:* Permiten explicar gráficamente las interacciones entre las instancias del modelo en lo que se refiere al comportamiento dinámico orientado a mensajes. El objetivo de este diagrama es modelar el flujo de control, ilustrar la coordinación de control y estructura de objetos.
- *Diagramas de estados:* Permiten explicar gráficamente las interacciones entre las instancias del modelo en lo que se refiere al comportamiento dinámico orientado a eventos. El objetivo de este diagrama es modelar el ciclo de vida

de un objeto y modelar objetos reactivos: interfaces de usuario, dispositivos, etc.

- *Diagramas de actividad:* Permiten explicar gráficamente las interacciones entre las instancias del modelo en lo que se refiere al comportamiento dinámico orientado a la actividad. El objetivo de este diagrama es modelar el flujo del Trabajo del Negocio y Modelar operaciones.

1.4.6 FLUJOS DE TRABAJO DEL PROCESO UNIFICADO

El proceso unificado de desarrollo divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito.

En la Figura 1.4 se presenta la estructura del Proceso Unificado de Desarrollo.

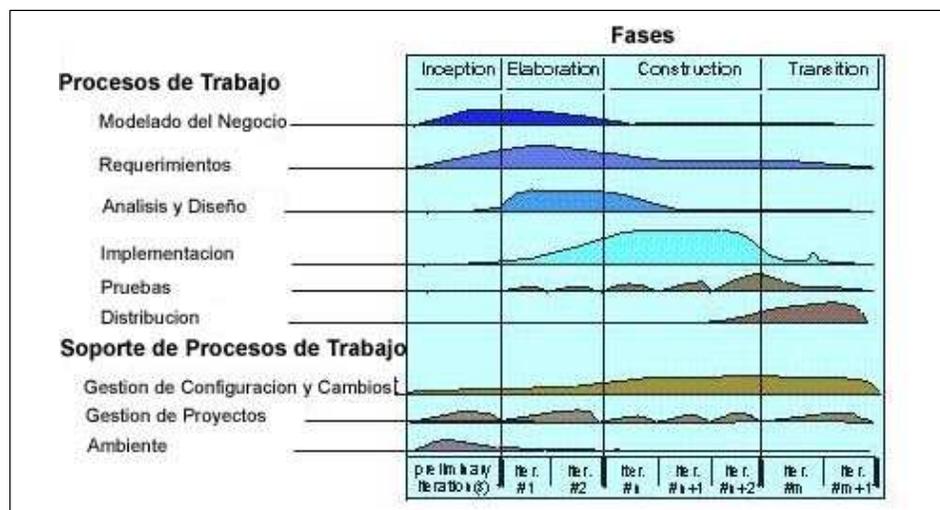


FIGURA 1.4: PROCESO UNIFICADO DE DESARROLLO.

(<http://www.willydev.net/descargas/articulos/general/inqsoftware.aspx>)

1.4.6.1 Procesos de trabajo

1.4.6.1.1 Modelado del negocio

Las empresas organizan sus actividades por medio de un conjunto de procesos de negocio, la finalidad del modelado del negocio es describir cada proceso del negocio del cliente, especificando sus datos, actividades, agentes y reglas del negocio.

Artefactos

- Glosario del negocio.
- Reglas del negocio
- Modelo de casos de uso del negocio.
- Modelo de Dominio

Trabajadores

- Analista del proceso del negocio.
- Diseñador del negocio.

Flujos de trabajo

- Identificar los casos del negocio y sus actores.
- Desarrollar un modelo de objetos de un negocio con: trabajadores, entidades de negocio y unidades de trabajo.

1.4.6.1.2 Requisitos

Este proceso de trabajo tiene como objetivo establecer y mantener un acuerdo con los clientes en lo que el sistema debe hacer, para proporcionar a los desarrolladores el conocimiento necesario de los requisitos del sistema. Sirve de base para planificar los contenidos técnicos de las iteraciones posteriores y para estimar el coste y tiempo necesarios para desarrollar el sistema. Permite definir la interfaz de usuario del sistema enfocándose en las necesidades y aspiraciones de los usuarios.

“Los requisitos son capacidades y condiciones con las cuales debe ser conforme el sistema y mas ampliamente el proyecto.”²⁶

Artefactos

- Modelo de casos de uso.
- Actor
- Caso de uso

²⁶ SORIANO, Fernando. Proceso Unificado Parte I. Universidad de Fasta. <http://pub.ufasta.edu.ar/fim28/files2005/FIM28%20UP%20I.pdf>.

- Descripción de la arquitectura.
- Glosario
- Prototipo de la interfaz del usuario.

Trabajadores

- Analista del sistema.
- Especificador de casos de uso.
- Diseñador de interfaces de usuario.
- Arquitecto.

Flujos de trabajo

- Enumerar requisitos candidatos.
- Entender el contexto del sistema.
- Capturar requisitos funcionales.
 - Identificar actores y casos de uso
 - Priorizar casos de uso
 - Detallar casos de uso
 - Prototipo de IU
 - Estructurar el modelo de casos de uso
- Capturar requisitos no funcionales
 - Hacer una lista de requisitos candidatos.

1.4.6.1.3 Análisis

El análisis es la descripción de cómo se implementará el sistema. En el análisis se debe, ejecutar las tareas y funciones descritas en los casos de uso. Satisfacer todos los requerimientos y ser flexible a cambios.

Artefactos

- Modelo de análisis
- Clase de análisis
- Realización de casos de uso-análisis.
- Paquete de análisis.
- Descripción de la arquitectura.

Trabajadores

- Arquitecto
- Ingeniero de casos de uso.
- Ingeniero de componentes.

Flujos de trabajo

- Análisis de la arquitectura
 - Identificar paquetes de análisis.
 - Identificar clases de entidad obvias.
 - Identificar requisitos especiales.
 - Identificar estilo arquitectónico.
- Análisis de los casos de uso
 - Identificar las clases de análisis.
 - Distribuir el comportamiento del caso de uso entre las clases de análisis.
 - Capturar/asignar requisitos no funcionales a clases de análisis.
- Análisis de las clases
 - Identificar las responsabilidades de las clases de análisis.
 - Identificar atributos y relaciones de las clases de análisis.
 - Capturar requisitos especiales.
- Análisis de los paquetes
 - Paquetes débilmente acoplados
 - Clases cohesionadas
 - Clases de interacción

1.4.6.1.4 Diseño

El diseño se centra en diseñar y validar la arquitectura, considerando los requisitos no funcionales y las restricciones existentes. El objetivo es descomponer el modelo de análisis en subsistemas que puedan desarrollarse en paralelo y definir la interfaz de cada subsistema. Se desarrolla una arquitectura robusta del sistema, y se adapta el diseño para que se corresponda con el ambiente de implementación, teniendo muy en cuenta el rendimiento.

Artefactos

- Modelo de diseño
- Clase de diseño
- Realización de casos de uso-diseño.
- Subsistema del diseño.
- Interfaz
- Descripción de la arquitectura. (vista del modelo de diseño).
- Modelo de despliegue.
- Descripción de la arquitectura. (vista del modelo de despliegue).

Trabajadores

- Arquitecto
- Ingeniero de casos de uso.
- Ingeniero de componentes.

Flujos de trabajo

- Diseño de la arquitectura.
- Diseño de los Casos de Uso
 - Identificar las clases de diseño y/o subsistemas necesarios para la realización del caso de uso.
 - Describir interacciones entre objetos de diseño
- Diseño de las clases
 - Identificar las responsabilidades de las clases de diseño.
 - Identificar: Operaciones, Atributos, Relaciones en las que participa, Estados, Dependencia de cualquier mecanismo genérico, Requisitos relevantes a su implementación, Métodos que soportan sus operaciones.
- Diseño de subsistemas.
 - Describir las dependencias entre los subsistemas.
 - Determinar qué clases de unos subsistemas interactúan con qué otras clases de otros subsistemas.

1.4.6.1.5 Implementación

Durante este proceso de trabajo se define la organización del código en términos de subsistemas y capas. Se convierten los elementos del diseño en elementos de implementación (fichero fuentes, binarios, ejecutables, y otros). Se realizarán las pruebas de unidad a los componentes desarrollados y se integrarán los resultados producidos en un solo sistema.

Artefactos

- Modelo de implementación.
- Componente
- Subsistema de la implementación.
- Interfaz
- Descripción de la arquitectura (vista del modelo de implementación).
- Plan de integración de construcciones.

Trabajadores

- Arquitecto
- Ingeniero de componentes.
- Ingeniero de sistemas.

Flujos de trabajo

- Implementación de la arquitectura.
- Integrar el sistema.
- Implementar un subsistema.
- Implementar una clase.
- Realizar pruebas de unidad.

1.4.6.1.6 Pruebas

El objetivo de las pruebas es encontrar y documentar defectos en la calidad del software y corregirlos antes de la instalación. Validar las suposiciones hechas en el diseño y en los requisitos mediante demostraciones concretas, se verifica la interacción entre los objetos y los componentes, se valida que el producto de

software funciona como se diseñó. También se verifica que los requisitos fueron implementados apropiadamente.

Artefactos

- Modelo de pruebas.
- Caso de prueba.
- Procedimiento de prueba.
- Componente de prueba.
- Plan de prueba
- Defecto
- Evaluación de prueba.

Trabajadores

- Ingeniero de pruebas.
- Ingeniero de componentes.
- Ingeniero de pruebas de integración.
- Ingeniero de pruebas del sistema.

Flujos de trabajo

- Planificar las prueba.
- Diseñar la prueba.
- Implementar la prueba.
- Realizar pruebas de integración.
- Realizar pruebas del sistema.
- Evaluar prueba.

1.4.6.1.7 Despliegue o distribución

Después de producir un producto se distribuye a los usuarios finales. La distribución incluye las siguientes actividades:

- Producir un “release”.
- Empaquetar el software.

- Distribuir el software.
- Instalar el software.
- Apoyar a los usuarios.
- Realizar pruebas beta.
- Migración de datos.
- Aceptación formal.

1.4.6.2 Soporte de procesos de trabajo

1.4.6.2.1 Gestión de cambios y configuración

La administración de configuración y cambios es la forma de controlar los artefactos producidos por las personas que trabajan en el proyecto. El objetivo es mantener la integridad de los artefactos y los releases en un proyecto a través de su ciclo de vida. Se identifican los elementos de configuración, se restringen y se controlan los cambios a esos elementos, y se definen y gestionan las configuraciones de esos elementos.

Los problemas más habituales son:

- Actualizaciones simultáneas
- Múltiples versiones

RUP proporciona guías para:

- Desarrollar en paralelo.
- Automatizar la construcción.
- Administrar defectos.

1.4.6.2.2 Gestión de proyectos

La gestión de proyectos de software es el arte de balancear objetivos contrarios, gestionar los riesgos, y sobreponerse a las restricciones para crear con éxito un producto que satisfaga las necesidades tanto de los clientes como de los usuarios finales. El objetivo es proporcionar un modo de gestionar proyectos intensivos de software, brindar instrucciones prácticas para la planificación, asignación de personal, ejecución y monitorización de proyectos, y proporcionar un modo de gestionar el riesgo.

Para la administración de proyectos RUP incluye:

- Un framework para el manejo de proyectos de software.
- Guías para planificación, provisión de personal, ejecución y monitoreo de planes.
- Un framework para manejar riesgos.

1.4.6.2.3 Entorno o ambiente

RUP guía en la configuración de un ambiente de proceso apropiado a cada proyecto. La gestión del entorno se enfoca en las actividades necesarias para configurar el proceso para un proyecto específico, describe las actividades necesarias para desarrollar las directrices que regirán el desarrollo del proyecto, estas actividades deben proporcionarle a la organización el entorno de desarrollo de software apropiado, que contendrá las herramientas de desarrollo y del proceso, plantillas, documentos, convenciones a seguir, y cualquier otro elemento necesario para llevar adelante con éxito el desarrollo del proyecto.

1.4.7 FASES DEL CICLO DE DESARROLLO²⁷

1.4.7.1 Inicio

Durante la fase de inicio se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el producto. En esta fase se establece la oportunidad y alcance el proyecto. Se identifican todas las entidades externas con las que se trata (actores) y se define la interacción a un alto nivel de abstracción: En esta fase se cumple con las siguientes tareas:

- Identificar todos los casos de uso.
- Describir algunos en detalle.

La oportunidad del negocio incluye:

- Criterios de éxito.
- Identificación de riesgos.

²⁷ GUERRERO, Luis. Rational Unified Process,
<http://www.dcc.uchile.cl/~luquerre/cc61j/clase1.html>

- Estimación de recursos necesarios.
- Producción del plan de las fases y de las iteraciones.
- Se propone una visión muy general de la arquitectura de software,

Del modelo de casos de uso y la lista de riesgos, se determina que casos de uso implementar primero para atacar los riesgos más críticos. Se realiza el proceso de planificación general y un plan de trabajo detallado para la siguiente fase. Se planea el plan de pruebas para ejecutarse desde la primera iteración de la fase de elaboración y que durante el ciclo de vida del proyecto se va a refinar.

1.4.7.2 Elaboración

Durante la fase de elaboración se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema. En esta fase se cumple con las siguientes tareas:

- Analizar el dominio del problema.
- Establecer una arquitectura base sólida.
- Desarrollar un plan de proyecto.
- Eliminar los elementos de mayor riesgo para el desarrollo exitoso del proyecto.

1.4.7.3 Construcción

En la fase de construcción se crea el producto. En esta fase se cumple con las siguientes tareas:

- Se desarrollan todas las componentes restantes y se incorporan al producto.
- Todo es probado en profundidad.
- El énfasis está en la producción eficiente y no ya en la creación intelectual.
- Puede hacerse construcción en paralelo, pero esto exige una planificación detallada y una arquitectura muy estable.

1.4.7.4 Transición

El objetivo de la fase de transición es traspasar el software desarrollado a los usuarios, una vez instalado si surgen nuevos elementos implicarán nuevos desarrollos. Esta fase incluye:

- Pruebas Beta para validar el producto con las expectativas del cliente.
- Ejecución paralela con sistemas antiguos.
- Conversión de datos.
- Entrenamiento de usuarios.
- Distribuir el producto.

Los objetivos de esta fase son:

- Obtener autosuficiencia de parte de los usuarios.
- Concordancia en los logros del producto de parte de las personas involucradas.
- Lograr el consenso cuanto antes para liberar el producto al mercado.

1.5. Combinación de TSP y RUP

Al combinar los procesos de TSP (Lanzamiento del Proyecto de Equipo, Desarrollo de la Estrategia, Desarrollo del Plan, Definición de Requerimientos, Diseño con Equipos, Implementación del Producto, Pruebas de integración en el sistema y Postmortem) y los flujos de trabajo del proceso unificado de desarrollo (Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Distribución, Gestión de Configuración y Cambios, Gestión de Proyectos y Ambiente) se obtienen los siguientes flujos a considerar en el desarrollo de un proyecto: Gestión de Proyectos, Modelado del Negocio, Requerimientos, Análisis, Diseño, Implementación, Pruebas, Postmortem y Distribución.

Cada flujo del ciclo de vida del sistema considera un grupo de actividades a seguir durante el desarrollo del proyecto. Cada actividad está ligada a cumplir con ciertas características y requiere la intervención de las personas con un rol específico.

Los roles definidos para la combinación son: Líder del Equipo, Administrador de la Planificación, Administrador de la Calidad de Procesos, Administrador del Desarrollo, Administrador del Soporte, además se considera un Jefe del Proyecto a quien se le reporta el progreso del desarrollo del sistema.

Gestión de Proyectos

El primer flujo Gestión de Proyectos, abarca las actividades:

- Conformar el equipo y asignar roles a cada uno de los integrantes.
- Definir metas
- Coordinar el trabajo.
- Crear el diseño conceptual del proyecto.
- Establecer riesgos potenciales.
- Crear la estrategia.
- Crear un plan de administración de la configuración.
- Crear el plan del proyecto.

Estas actividades se definieron al integrar las actividades de los procesos Lanzamiento del Proyecto de Equipo, Desarrollo de la Estrategia y Desarrollo del Plan de TSP; y los flujos de trabajo Gestión de Proyectos, Gestión de Cambios y Configuración y Entorno del proceso unificado de desarrollo. Se consideraron estos procesos y flujos porque éstos sirven de soporte al desarrollo del sistema, el objetivo de estas actividades es conformar un buen equipo de trabajo, fijar unas metas claras y realizables, establecer los riesgos potenciales para prevenir las consecuencias de los mismos, y crear un plan a seguir durante el desarrollo que permita obtener como resultado un producto que satisfaga las necesidades del usuario, en el tiempo establecido.

Modelado del Negocio

El flujo Modelado del Negocio, considera las siguientes actividades:

- Identificar los casos de uso del negocio y sus actores.
- Desarrollar un modelo de objetos de un negocio.

Estas actividades son propias del flujo Modelado del Negocio de RUP, TSP no considera estas actividades dentro de sus procesos.

Requerimientos

El flujo Requerimientos, considera las siguientes actividades:

- Obtención de los requerimientos.
- Especificación de los requerimientos de software SRS.
- Crear el plan de prueba del sistema.
- Revisar e inspeccionar los requerimientos.
- Modificar los requerimientos

A las actividades consideradas en el flujo de Requerimientos de RUP se adiciona la creación del plan de pruebas del sistema, cuyo objetivo es asegurar que el sistema al final hará lo que los usuarios supusieron que haría; es decir, las pruebas estarán dirigidas a asegurar el cumplimiento de los requerimientos del SRS. Además se adicionan las actividades: Revisar e inspeccionar los requerimientos y Modificar los requerimientos, cuyo objetivo es corregir los problemas identificados durante la revisión e inspección antes de empezar con el Análisis.

Análisis

El flujo Análisis, considera las siguientes actividades:

- Análisis de la arquitectura
- Análisis de los casos de uso
- Análisis de las clases
- Análisis de los paquetes

Estas actividades son propias del flujo Análisis de RUP, TSP no considera estas actividades dentro de sus procesos.

Diseño

El flujo Diseño, considera las siguientes actividades:

- Diseño de alto nivel

- Diseño detallado
- Crear un plan de pruebas de integración.
- Revisar e inspeccionar el diseño.
- Modificar el diseño.

El diseño de alto nivel es una actividad propia de TSP, en la que se define la estructura general del producto, los componentes y las funciones de los componentes. La actividad diseño detallado abarca las actividades del Diseño de RUP (Diseño de la arquitectura, Diseño de los casos de uso, Diseño de las clases y Diseño de subsistemas). Se adiciona la creación del plan de pruebas de integración que chequean y verifican todas las interfaces entre los componentes del sistema. Además se adicionan las actividades: Revisar e inspeccionar el Diseño y Modificar el diseño, cuyo objetivo es corregir los problemas identificados durante la revisión e inspección antes de empezar con la Implementación.

Implementación

El flujo Implementación, considera las siguientes actividades:

- Crear un plan para la implementación.
- Implementación de la arquitectura.
- Implementación de los subsistemas.
- Implementación de las clases y objetos.
- Revisar e inspeccionar la implementación.
- Modificar la implementación.
- Realizar pruebas de unidad.
- Revisar la calidad de los componentes.
- Integrar las componentes en un sistema ejecutable.

TSP considera la creación de un plan para la implementación antes de empezar la codificación, éste comprende los estándares de implementación y las pruebas de unidad del código. La implementación de la arquitectura y de los subsistemas son actividades propias de RUP. Tanto TSP como RUP definen la implementación de las clases y objetos. Terminada la codificación se realiza la revisión, inspección y

modificación del código para corregir los defectos encontrados durante la implementación. Posteriormente se realizan las pruebas de unidad, la revisión de la calidad de los componentes e integración de las componentes en un sistema ejecutable.

Pruebas

El flujo Pruebas, considera las siguientes actividades:

- Planificar las pruebas.
- Diseñar las pruebas.
- Realizar las pruebas.
- Evaluar las pruebas.
- Registrar las pruebas

TSP y RUP, ambos consideran planificar, diseñar, realizar, evaluar las pruebas. Adicional a éstas TSP define registrar las pruebas, actividad en la que se registran los datos de tiempo de ejecución de las pruebas, número de defectos encontrados y las condiciones en las que se efectuaron la pruebas.

Postmortem

El flujo Postmortem, considera las actividades propias de TSP, que son:

- Revisión de los datos.
- Evaluaciones del desempeño de los roles.
- Preparar reportes.
- Producción del reporte.

El objetivo de las actividades de postmortem es proveer una estructura de aprendizaje y mejoramiento para los siguientes ciclos del desarrollo del producto actual o para nuevos proyectos.

Distribución

El flujo Distribución, considera las actividades propias de RUP, que son:

- Producir un release.
- Empaquetar el software.

- Distribuir el software.
- Instalar el software.
- Apoyar a los usuarios.
- Realizar pruebas beta.
- Migración de datos.
- Aceptación formal.

Una vez finalizado el desarrollo se procede a la distribución del producto a los usuarios finales.

Los flujos con las actividades, características y roles resultado de la combinación de TSP y TUP se encuentran detallados en la tabla 1.2.

FLUJO	ORIGEN	ACTIVIDAD	CARACTERÍSTICAS	ROLES
Gestión de Proyectos	TSP (Lanzamiento del Proyecto de Equipo), RUP (Gestión de Proyectos)	Conformar el equipo y asignar roles a cada uno de los integrantes.	Se deben asignar los roles de acuerdo al perfil de los miembros del equipo. Se llenan las formas INFO (ver anexo 1.1) de cada miembro.	Jefe del Proyecto* Todos (guía: Líder del Equipo)
	TSP (Lanzamiento del Proyecto de Equipo), RUP (Gestión de Proyectos)	Definir metas	Se definen las metas del equipo, las metas por miembro del equipo y las metas por el rol.	Jefe del Proyecto* Todos (guía: Líder del Equipo)
	TSP (Lanzamiento del Proyecto de Equipo), RUP (Gestión de Proyectos)	Coordinar el trabajo.	Los miembros del equipo acuerdan cuando y donde serán las reuniones de trabajo. Cada miembro llena las formas TASK, SCHEDULE y WEEK (ver anexos 1.2, 1.3, 1.4). El administrador del equipo prepara la forma WEEK del equipo (ver anexo 1.5).	Todos (guía: Líder del Equipo)
	TSP (Desarrollo de la Estrategia)	Crear el diseño conceptual del proyecto.	El diseño conceptual del proyecto contiene: los usuarios del sistema, la plataforma y las herramientas que se utilizan para el desarrollo del proyecto, los componentes principales del producto, las funciones de estos componentes.	Todos (guía: Administrador del Desarrollo)
	TSP (Desarrollo de la Estrategia), RUP (Gestión de Proyectos)	Establecer riesgos potenciales.	Se define: el riesgo, la probabilidad de que ocurra, el impacto, la exposición, la prioridad y la medida de prevención. Estos se llenan en la forma ITL (ver anexo 1.6).	Todos (guía: Administrador del Soporte)
	TSP (Desarrollo de la Estrategia), RUP (Gestión de Proyectos)	Crear la estrategia	Se crea la estrategia del proyecto y se llena la forma STRAT (ver anexo 1.7).	Todos (guía: Administrador del Desarrollo)
	TSP (Desarrollo de la Estrategia), RUP (Gestión de Cambios y Configuración)	Crear un plan de administración de la configuración	Se especifica el procedimiento que se seguirá para la realización de cambios del programa.	Administrador del Soporte

	TSP (Desarrollo del Plan), RUP (Entorno o Ambiente)	Crear el plan del proyecto	<p>Un plan debe ser equilibrado. Un plan detallado ayuda al monitoreo y administración del trabajo de los miembros del equipo.</p> <p>Se detallan los nombres y tamaños de las componentes en la forma SUMS (ver anexo 1.8).</p> <p>Se registran las tareas a realizarse en la forma TASK del equipo (ver anexo 1.9).</p> <p>Se registran las horas semanales en la forma SCHEDULE del equipo (ver anexo 1.10).</p> <p>Se llena la forma SUMP y SUMQ con los datos del plan (ver anexos 1.11 y 1.12).</p> <p>Se balancea el trabajo y se registran las tareas asignadas a cada miembro en la forma TASK de cada uno.</p> <p>Se registran las horas semanales asignadas a cada miembro en la forma SCHEDULE de cada miembro.</p> <p>Se consolidan los planes si fueron cambiados los tiempos y se crean las formas TASK, SCHEDULE, SUMP y SUMQ.</p>	Todos (guías: Administrador de la Planificación y el Administrador de la Calidad de Procesos)
Modelado del Negocio	RUP (Modelado del Negocio)	Identificar los casos de uso del negocio y sus actores.	Se describe cada proceso del negocio del cliente, especificando sus datos, actividades, agentes y reglas de negocio.	Líder del Equipo
	RUP (Modelado del Negocio)	Desarrollar un modelo de objetos de un negocio.	Se crea un modelo de objetos del negocio.	Líder del Equipo
Requerimientos	TSP (Definición de los requerimientos), RUP (Requerimientos)	Obtención de los requerimientos.	Se establecen los requisitos candidatos del sistema	Líder del Equipo Jefe del Proyecto*
	TSP (Definición de los requerimientos), RUP (Requerimientos)	Especificación de los requerimientos de software SRS.	Se especifican los requisitos funcionales y no funcionales. Se documentan los requerimientos en el SRS.	Todos (guía: Administrador del Desarrollo)
	TSP (Definición de los requerimientos)	Crear el plan de prueba del sistema.	Se definen las pruebas del sistema. Las pruebas deben ser realizadas bajo varias condiciones de error para considerar la usabilidad y los problemas de recuperación.	Todos (guía: Administrador del Desarrollo)

	TSP (Definición de los requerimientos)	Revisar e inspeccionar los requerimientos.	Se identifican los problemas de los requerimientos, para corregirlos antes del Análisis. Se llena la forma INS (ver anexo 1.13) con los defectos de la inspección	Todos (guía: Administrador de la Calidad de Procesos)
	TSP (Definición de los requerimientos)	Modificar los requerimientos	Se modifican el SRS y el plan de prueba del sistema. Corregidos los problemas identificados en la revisión e inspección de los requerimientos.	Todos (guía: Administrador de la Desarrollo)
Análisis	RUP (Análisis)	Análisis de la arquitectura	Se establecen los paquetes, clases y un estilo arquitectónico.	Todos (guía: Administrador del Desarrollo)
	RUP (Análisis)	Análisis de los casos de uso	Se identifican las clases de análisis, se crean los diagramas de colaboración y un modelo de análisis	Todos (guía: Administrador del Desarrollo)
	RUP (Análisis)	Análisis de las clases	Se identifican las responsabilidades de las clases de análisis	Todos (guía: Administrador del Desarrollo)
	RUP (Análisis)	Análisis de los paquetes	Se identifican los paquetes del sistema.	Todos (guía: Administrador del Desarrollo)
Diseño	TSP (Diseño), RUP (Diseño)	Diseño de alto nivel	Se define la estructura general del producto. Se crea el SDS.	Todos (guía: Administrador del Desarrollo)
	TSP (Implementación), RUP (Diseño)	Diseño detallado	Se define la estructura de los subsistemas, módulos, componentes.	Todos (guía: Administrador del Desarrollo)
	TSP (Diseño), RUP (Diseño)	Crear un plan de pruebas de integración.	Se crean las pruebas de integración de los módulos.	Todos (guía: Administrador del Desarrollo)
	TSP (Diseño)	Revisar e inspeccionar el diseño.	Se identifican los problemas del diseño para corregirlos antes de la implementación. Se llena la forma INS (ver anexo 1.14) con los defectos de la inspección	Todos (guía: Administrador de la Calidad de Procesos)
	TSP (Diseño)	Modificar el diseño.	Se actualiza el diseño corregidos los errores encontrados en la revisión e inspección del diseño.	Todos (guía: Administrador del Desarrollo)

Implementación	TSP (Implementación)	Crear un plan para la implementación.	Se definen los estándares de implementación y se crea el plan de pruebas de unidad.	Todos (guía: Administrador del Desarrollo)
	RUP (Implementación)	Implementación de la arquitectura.	Se establecen los componentes del sistema, se convierten los elementos del diseño en elementos de implementación (fichero fuentes, binarios, ejecutables, y otros).	Todos
	RUP (Implementación)	Implementación de los subsistemas.	Se establecen los subsistemas, se implementan las pantallas del diseño de pantallas.	Todos
	TSP (Implementación), RUP (Implementación)	Implementación de las clases y objetos	Se crean los ficheros con el código fuente que implementa una clase del diseño.	Todos
	TSP (Implementación)	Revisar e inspeccionar la implementación.	Se identifican los problemas de la implementación. Se llena la forma INS (ver anexo 1.15) con los defectos de la inspección.	Todos (guía: Administrador de la Calidad de Procesos)
	TSP (Implementación)	Modificar la implementación.	Se actualiza la implementación con las correcciones de los problemas encontrados en la revisión e inspección.	Todos
	TSP (Implementación), RUP (Implementación)	Realizar pruebas de unidad	Se realizan las pruebas de unidad que fueron definidas en el plan de la implementación. Se llenan las formas: formas LOGD, LOGT y LOGTEST (ver anexo 1.16, 1.17 y 1.18)	Todos
	TSP (Implementación)	Revisar la calidad de los componentes.	Se determina la calidad de los componentes en base a los criterios de calidad.	Administrador de la Calidad de Procesos
	TSP (Implementación)	Integrar las componentes en un sistema ejecutable	Se integran los componentes de la implementación en un ejecutable.	Todos (guía: Administrador del Soporte)
Pruebas	TSP (Pruebas de integración y del sistema), RUP (Pruebas)	Planificar las pruebas.	Se describe una estrategia de prueba. Se estima los requisitos para las pruebas.	Todos (guía: Administrador del Desarrollo)
	TSP (Pruebas de integración y del sistema), RUP (Pruebas)	Diseñar las pruebas.	Se identifican y describen los casos de prueba para cada construcción y los procedimientos de prueba.	Todos (guía: Administrador del Desarrollo)
	TSP (Pruebas de integración y del sistema), RUP (Pruebas)	Realizar las pruebas.	Se realizan las pruebas de integración y las pruebas del sistema.	Todos (guía: Administrador del Desarrollo)

	TSP (Pruebas de integración y del sistema), RUP (Pruebas)	Evaluar las pruebas.	Se evalúan los resultados obtenidos de las pruebas con el plan de pruebas.	Todos (guía: Administrador del Desarrollo)
	TSP (Pruebas de integración y del sistema)	Registrar las pruebas	Se registra: el tiempo de la prueba, los defectos encontrados, las condiciones de la prueba. Se llena la forma: LOGTEST.	Todos (guía: Administrador del Desarrollo)
Postmortem	TSP (Postmortem)	Revisión de los datos.	Se examinan los datos que el equipo y los miembros del equipo hicieron. Se identifica donde trabajaron los procesos y donde no lo hicieron. Se compara el desempeño del equipo con sus metas y planes. Se identifican áreas problema y necesidades de mejora. Se analizan los datos de los defectos del equipo y se evalúa el grado con el que el equipo produce un producto de calidad. Se llena la forma PIP (ver anexo 1.19)	Todos (guía: Administrador de la Calidad de Procesos)
	TSP (Postmortem)	Evaluaciones del desempeño de los roles	La evaluación se enfoca en los objetivos cumplidos por el rol. Se llenan las formas PEER (ver anexo 1.20)	Todos (guía: Líder del Equipo)
	TSP (Postmortem)	Preparar reportes	Se preparan los reportes: reporte del ciclo, reporte de los roles, reporte por miembro del equipo.	Todos (guía: Líder del Equipo)
	TSP (Postmortem)	Producción del reporte.	Una vez corregido el reporte se entrega al Jefe del proyecto y a todos los miembros del equipo.	Todos (guía: Líder del Equipo)
Distribución	RUP (Distribución)	Producir un release.	El propósito de este flujo es asegurar la disponibilidad del producto de software para los usuarios finales.	Administrador de la Calidad de Procesos.
	RUP (Distribución)	Empaquetar el software.		Líder del Equipo
	RUP (Distribución)	Distribuir el software.		Jefe del Proyecto*
	RUP (Distribución)	Instalar el software.		Líder del Equipo Administrador del Desarrollo
	RUP (Distribución)	Apoyar a los usuarios.		Líder del Equipo Administrador del Soporte

	RUP (Distribución)	Realizar pruebas beta.		Líder del Equipo Administrador del Desarrollo
	RUP (Distribución)	Migración de datos.		Líder del Equipo Administrador del Desarrollo
	RUP (Distribución)	Aceptación formal.		Jefe del Proyecto* Líder del Equipo

Tabla 1.2: Flujos de la combinación de TSP y RUP.

CAPITULO II

2. CASO DE ESTUDIO

2.1. DESCRIPCIÓN DEL CASO DE ESTUDIO

El caso de estudio es el Desarrollo de Software para Administración y Monitoreo de Rutas a seguir por los Vendedores de Puerta a Puerta.

El software que se va a desarrollar está dirigido a satisfacer la necesidad de controlar las rutas que los vendedores realizan durante la toma de pedidos, en los domicilios o en los negocios de los clientes.

El sistema permitirá realizar la administración de las rutas seguidas por los vendedores y facilitará el seguimiento de las actividades del vendedor, de las rutas y de los clientes. Para el desarrollo del sistema se seguirán las fases establecidas anteriormente en la combinación de TSP y RUP.

2.1.1. OBJETIVOS

El caso de estudio tiene como objetivos:

- Seguir los flujos de la combinación de TSP y RUP en el desarrollo del caso de estudio.
- Desarrollar un sistema que permita administrar rutas (establecer los clientes a visitarse por los vendedores de puerta a puerta y definir las rutas seguidas por los vendedores) y monitorear las rutas del vendedor.

2.2. DESARROLLO DEL SISTEMA USANDO TSP Y RUP

Para el desarrollo del sistema que planteamos se van a seguir los flujos planteados en el **punto 1.5**.

2.2.1. GESTION DE PROYECTOS

2.2.1.1. Conformar el equipo y asignar roles a cada uno de los integrantes

El equipo de trabajo está conformado por: Geovanna Bustos y Cristhian Guallasamín. estudiantes de la Carrera de Sistemas Informáticos y de Computación, previos a obtener el título de Ingenieros.

Para designar los roles a los integrantes del equipo cada miembro debe llenar la forma INFO en la primera reunión, (ver CD, anexo 1.1.1 y 1.1.2).

Los roles a desempeñar por los integrantes del equipo de acuerdo con las responsabilidades definidas para cada rol que se encuentran en el *anexo 2.1* se presentan en la Tabla 2.1:

INTEGRANTE	ROL
Geovanna Bustos	<ul style="list-style-type: none"> • Líder del equipo • Administrador de la Planificación • Administrador de la Calidad/Procesos
Cristhian Guallasamín	<ul style="list-style-type: none"> • Administrador del Desarrollo • Administrador del Soporte

Tabla 2.1: Roles del equipo

2.2.1.2. Definir metas

2.2.1.2.1. Definir las Metas del Equipo

Las metas que el equipo estableció para el proyecto son las siguientes:

Meta 1	Desarrollar un sistema usando TSP y RUP.
Medida 1.1	El sistema se desarrollo siguiendo los flujos de la combinación de TSP y RUP.
Meta 2	Producir un sistema de acuerdo con el plan.
Medida 2.1	Los tiempos del desarrollo del sistema están de acuerdo con el plan.
Meta 3	Producir un sistema de calidad.
Medida 3.1	Porcentaje de defectos encontrados en las pruebas: 30%.
Medida 3.2	Requerimientos del sistema cumplidos al final del sistema: 100%.

Tabla 2.2: Metas del equipo

2.2.1.2.2. Definir las Metas por Miembro del Equipo

Las metas que cada miembro del equipo estableció para el proyecto se presentan a continuación:

Miembro 1: Geovanna Bustos

Meta 1	Desarrollar un sistema usando TSP y RUP.
Medida 1.1	Todas las formas TSP del miembro del equipo están completas y oportunas.
Medida 1.2	El trabajo realizado cumple con las tareas especificadas en la integración de TSP y RUP.
Meta 2	Participar activamente en el equipo, en el desarrollo del sistema. Cumplir con las horas establecidas de trabajo.
Medida 2.1	Cumplir el calendario establecido para tareas asignadas como miembro del equipo.
Medida 2.2	Colaborar con el cumplimiento de las tareas del otro miembro del equipo.
Medida 2.3	
Meta 3	El producto desarrollado cumpla con normas de calidad.
Medida 3.1	El producto sigue estándares de calidad.
Medida 3.2	Mínimo número de defectos encontrados en las pruebas.

Tabla 2.3: Metas del miembro 1 del equipo

Miembro 2: Cristhian Guallasamín

Meta 1	Desarrollar un sistema que involucre la integración de TSP y RUP.
Medida 1.1	Todas las formas TSP del miembro del equipo están completas y oportunas.
Medida 1.2	El trabajo realizado cumple con las tareas especificadas en la integración de TSP y RUP.
Meta 2	Desarrollar el sistema siguiendo estándares y un plan.
Medida 2.1	El sistema es fácil de entender.
Medida 2.2	El sistema cumple con las especificaciones del plan.
Meta 3	Producir un sistema de calidad
Medida 3.1	El producto sigue estándares de calidad.
Medida 3.2	Porcentaje de defectos encontrados en las pruebas de integración: 30%
Medida 3.3	Cumplir con todos los requerimientos del sistema.

Tabla 2.4: Metas del miembro 2 del equipo

2.2.1.2.3. Definir las Metas por Rol

Las metas se definen por los roles que se establecieron, (*ver anexo 2.3*).

2.2.1.3. Coordinar el Trabajo

- Se establece el horario de las reuniones:
Día: sábado.
Hora: 2 horas.
- Antes de cada reunión cada miembro debe llenar o modificar las formas TASK SCHEDULE y WEEK (*ver CD, anexos 1.1.5 al 1.1.8*). Para monitorear el progreso ganado en contraste con el plan durante la semana.

- Antes de cada reunión cada miembro proporciona una copia de sus formas al administrador de la planificación.
- El administrador de la planificación prepara el informe WEEK del equipo (*ver CD, anexo 1.1.9*) y lo distribuye al equipo en su reunión semanal.

2.2.1.4. Crear el diseño conceptual del proyecto.

El diseño conceptual del proyecto contiene los usuarios del sistema, la plataforma y las herramientas que se utilizan para el desarrollo del proyecto, los componentes principales del producto, las funciones de estos componentes y el tamaño de los mismos.

Usuarios

Para el caso de estudio los usuarios son: el administrador y los vendedores.

El caso de estudio no está dirigido para una empresa específica, está dirigido a satisfacer la necesidad que tienen las empresas, de controlar las rutas de los vendedores dedicados a la toma de pedidos en el domicilio o negocio del cliente.

Plataforma y Herramientas

Para construir el producto se dispuso de la plataforma Windows XP y las herramientas listadas a continuación para un correcto funcionamiento del sistema:

- Microsoft Visual Studio .Net 2003.
- Microsoft Sql Server 2000.
- MapXtreme 2004, necesario para la visualización de mapas digitales.
- Service Pack 2 para Windows XP, requisito para el MapXtreme y Visual Studio .Net 2003.
- Service Pack 4 para Sql Server 2000, necesario para el intercambio de información entre la base de datos de la Pocket PC y la PC (Sincronización).
- Microsoft Sql Server CE 2.0, necesario para el funcionamiento del publicador.
- Active Sync 4.2, software de enlace de conexión entre la PC y la Pocket PC.

Componentes Principales

Los principales componentes del producto a construir son:

- Administración de Rutas y
- Monitoreo de Rutas.

Funciones de los Componentes

Las funciones de los componentes son:

Componente: Administración de Rutas

Funciones:

- Generar información de clientes para que el vendedor realice las visitas.
- Sincronizar información entre la PC y la Pocket PC.
- Gestionar pedidos del vendedor.
- Gestionar motivos de pedido del vendedor.
- Recibir información de los vendedores.

Componente: Monitoreo de Rutas:

Funciones:

- Reporte de ruta seguida por el vendedor.
- Reporte de clientes visitados por el vendedor que hayan realizado pedido.
- Reporte de clientes visitados por el vendedor con motivo de no pedido.
- Reporte de clientes no visitados por el vendedor.

2.2.1.5. Establecer riesgos potenciales.

Los riesgos potenciales que podrían presentarse durante el desarrollo del proyecto se encuentran en la Tabla 2.5. La probabilidad de que ocurra el riesgo se establece de la siguiente manera: Alta=3, Media=2, Baja=1.

La prioridad depende de la probabilidad y el nivel de impacto que tiene el riesgo. La prioridad = probabilidad * impacto. Esta puede ser: Rutinario=1- 3, Significativo=4 -6, Crítico=7-9.

Se ingresan los riesgos en la forma ITL (*Ver CD, anexos 1.1.10*).

DESCRIPCIÓN	PROBABI- LIDAD	IMPACTO		EXPOSICIÓN	PRIORIDAD	MEDIDA DE PREVENCIÓN
		Nivel	Descripción			
Grupo de trabajo inadecuado	baja	alta	El desarrollo del proyecto se alarga.	1*3=3	rutinaria	En el grupo de trabajo debe existir la contribución de los esfuerzos personales, el soporte a otros miembros y el trabajo en cooperación para resolver los problemas y desacuerdos.
Metas agresivas e inadecuadas.	media	media	Al no existir motivación en el proyecto el desarrollo se alarga o no se completa.	2*2=4	significativo	Las metas deben ser agresivas pero realizables para que exista motivación y el equipo completo se esfuerce para conseguir las metas.
Falta de coordinación del equipo.	media	alta	Incumplimiento de plazos establecidos para las tareas.	2*3=6	significativo	Comunicación constante entre los miembros del equipo.
Diseño inadecuado.	baja	media	Demora en el desarrollo del proyecto	1*2=2	rutinaria	Diseñar el producto de manera que cumpla con los requerimientos.
Proyecto muy difuso	media	alta	El proyecto se torna muy grande	2*3=6	significativo	Especificar bien los requerimientos del proyecto.
La estrategia es inadecuada	baja	media	Se excede en el tiempo de desarrollo disponible para el proyecto.	1*2=2	rutinaria	Establecer una buena estrategia a seguir
Se abandona el plan del proyecto.	baja	alta	Retrazo en el desarrollo del sistema.	1*3=3	rutinaria	El plan será simple o complejo dependiendo de la complejidad del trabajo a realizar.
Planificación muy optimista.	media	media	Incumplimiento con los plazos establecidos para las tareas.	2*2=4	significativo	Un plan debe ser equilibrado, todos los ingenieros completan sus tareas planeadas en el orden apropiado y al mismo tiempo.
Falta de seguimiento del proyecto	alta	media	Recorte de la calidad del proyecto.	3*2=6	significativo	Realizar un seguimiento del proyecto.
Hacer un desarrollo orientado en la investigación.	media	baja	Demora en la implementación	2*1=2	rutinaria	Para el desarrollo se debe seguir las fases establecidas en la combinación de TSP y RUP.
Requerimientos muy simples	media	media	El sistema no cumpla con las necesidades del cliente. Existe insatisfacción del cliente.	2*2=4	significativo	Establecer bien los requerimientos de manera que cumplan con las necesidades del cliente.

Requerimientos muy meticulosos	media	media	El sistema cumple con tareas que no le sirven al cliente. Existe insatisfacción del cliente.	2*2=4	significativo	Establecer bien los requerimientos de manera que cumplan con las necesidades del cliente.
Herramientas no admisibles.	Baja	alta	El proyecto no sea terminado.	1*3=3	rutinaria	Tener acceso a las herramientas que se van a utilizar antes del desarrollo.
La curva de aprendizaje de la herramienta es muy larga.	media	media	El desarrollo del proyecto se alarga.	2*2=4	significativo	Buscar apoyo de un especialista o conocedor de la herramienta.
Recorte de la calidad.	media	media	El sistema no cumple con estándares, falta de satisfacción del cliente.	2*2=4	significativo	Seguir estándares y normas de calidad en el desarrollo del proyecto.
Calidad no aceptable del producto.	media	media	Rechazo del sistema por el cliente.	2*2=4	significativo	Seguir estándares y normas de calidad en el desarrollo del proyecto.
No cumple con los requerimientos del producto.	baja	alta	El producto desarrollado no sirve. Se debe volver a desarrollar.	1*3=3	rutinaria	Revisar al final de cada fase que se estén cumpliendo con los objetivos planteados.
Diseño muy sencillo o muy complejo.	media	alta	Dificultad al implementar.	2*3=6	significativo	Diseñar el producto de manera que cumpla con los requerimientos.
Utilización de metodologías desconocidas.	media	Media	Demora en el desarrollo	2*2=4	significativo	Conocer la metodología a utilizarse antes de iniciar el proyecto.

Tabla 2.5: Riesgos Potenciales

2.2.1.6. Crear la estrategia

Criterios de la estrategia

- En el ciclo 1 se tiene un producto que cumple con todos los requerimientos.
- El producto desarrollado al final del primer ciclo es de gran calidad y puede ser fácilmente probado.
- El diseño del producto tiene una estructura modular que permite a los miembros del equipo trabajar independientemente.
- Para el desarrollo del proyecto se usan los flujos de la combinación de TSP y RUP.
- El producto que se va a desarrollar contendrá dos subsistemas: Administración de Rutas y Monitoreo de Rutas,
- El subsistema Administración de Rutas contendrá las funciones: Generar información de clientes para que el vendedor realice las visitas, Sincronizar información entre la PC y la Pocket PC, Gestionar pedidos del vendedor, Gestionar motivos de no pedido del vendedor, Recibir información de los vendedores.
- El subsistema Monitoreo de Rutas contendrá las funciones: Reportar la ruta seguida por el vendedor, Reportar los clientes visitados por el vendedor que hayan realizado pedido, Reportar los clientes visitados por el vendedor con Motivo de no gestión, Reportar los clientes no visitados por el vendedor.

Posibles estrategias alternativas

Para determinar las posibles estrategias del proyecto se establecen las oportunidades, amenazas, fortalezas y debilidades que se tienen para el desarrollo del proyecto.

Oportunidades

- El sistema es pequeño.
- El caso de uso a desarrollarse podría servir de base para proyectos posteriores.

Amenazas

- Dificultad en la adquisición de los elementos visuales: mapas.
- El tiempo sea corto para el desarrollo del proyecto.

Fortalezas

- Experiencia en el desarrollo de sistemas.
- Se conoce la metodología RUP.
- Se tienen disponibles las herramientas de desarrollo.
- Se tiene un guía, el director del proyecto.

Debilidades

- Falta de experiencia en el uso de los procesos TSP.
- No se dispone de un dispositivo móvil propio.

Considerando los aspectos anteriores se establecen las siguientes estrategias:

Estrategia 1: Desarrollar un sistema que cumpla con los requerimientos básicos en el menor tiempo y de calidad.

Estrategia 2: Desarrollar un sistema que cumpla con los requerimientos, en un tiempo amplio y de alta calidad.

Estrategia 3: Desarrollar un sistema que cumpla con los requerimientos, en el tiempo establecido en el plan y de calidad.

Los riesgos y beneficios de las estrategias alternativas se detallan en la tabla 2.6

EST.	RIESGOS	BENEFICIOS
1	<ul style="list-style-type: none"> • Minimizar los requerimientos del sistema. • El sistema no sea de calidad. • No cumplir con los requerimientos. • Insatisfacción del usuario 	<ul style="list-style-type: none"> • Terminar el sistema en un corto tiempo. • Tener un prototipo del sistema que cumple con los requerimientos básicos.
2	<ul style="list-style-type: none"> • Incumplimiento de calendarios que provoque retraso en el término del proyecto. 	<ul style="list-style-type: none"> • El sistema sea de calidad. • El sistema cumpla con los requerimientos.
3	<ul style="list-style-type: none"> • No cubrir con los requerimientos, ya que el desarrollo del sistema está determinado por los requerimientos y los tiempos establecidos para el cumplimiento de los mismos. 	<ul style="list-style-type: none"> • El sistema cumpla con los requerimientos. • El sistema sea fácil de probar. • El sistema sea de alta calidad. • El sistema cumpla con los tiempos establecidos, sea puntual.

Tabla 2.6 Riesgos y beneficios de las estrategias alternativas

La estrategia seleccionada es: Desarrollar un sistema que cumpla con los requerimientos, en el tiempo establecido en el plan y de calidad.

Para documentar la estrategia se llena la forma STRAT (*Ver CD, anexos 1.1.11*).

2.2.1.7. Crear un plan de administración de la configuración

Identificación de la configuración

Los ítems a ser controlados son:

- Una lista aprobada de los ítems a ser controlados.
- Los requerimientos
- El diseño del producto.
- El código fuente del producto.
- Los materiales de prueba y los resultados de las pruebas.
- Estándares de diseño del producto, como: estándares de nombramiento de los módulos y del sistema, estándares de interfase, estándares de mensajes y pantallas, y las librerías re usables en el programa.

Un procedimiento de control de la configuración

Crear la tabla de control de la configuración (CCB) con la guía del administrador del soporte y el administrador de desarrollo.

Para realizar un cambio en estos ítems se siguen los siguientes pasos:

- Se establece el estado en el que se encuentra el ítem.
- Se establece el cambio que se quiere realizar sobre el ítem.
- Se establecen los efectos que se producen con el cambio del ítem.
- Se establece quién se va a encargar de los cambios.
- Se solicita la autorización del líder del equipo, quien requiere la autorización del director del proyecto.
- Si el cambio es autorizado, proceder a realizar el cambio.
- Registrar el cambio en las respectivas formas.
- Actualizar el Project Notebook, con el cambio realizado.

Se considera que para realizar un cambio en el proyecto o componente, este queda bloqueado, de tal manera que el resto de miembros del equipo no pueden realizar nada del proyecto o componente que esta sufriendo el cambio.

2.2.1.8. Crear el plan del proyecto

Una vez realizado el diseño conceptual y las estimaciones de tamaño de los componentes del desarrollo, se procede a la creación del plan, se llena la forma SUMS (*Ver CD, anexos 1.1.12*) con los nombres y tamaños estimados de los componentes ha desarrollarse en el ciclo 1.

Se listan las tareas que se requieren en la construcción del producto, entonces se llena la forma TASK y SCHEDULE del equipo (*Ver CD, anexos 1.1.13 Y 1.1.14*).

Se llena las formas SUMP y SUMQ (*Ver CD, anexos 1.1.15-1.1.16*) se determinan los defectos que son introducidos en esta fase para este ciclo, de manera que el plan de calidad concuerde con los criterios de calidad descritos en el Standard Qual (*Ver Anexo 2.2*).

Se crean los planes de los miembros del equipo, de acuerdo con los datos de las formas TASK y SCHEDULE del equipo (*Ver CD, anexos 1.1.17 y 1.1.18*), se actualiza la forma SUMP y SUMQ (*Ver CD, anexos 1.1.19-1.1.20*); todas estas formas constituyen el Plan final del equipo para el proyecto que se esta desarrollando. Se entregan copias del plan a los miembros del equipo y se guardan las formas en el Project Notebook.

Se realiza el seguimiento del proyecto cada fin de semana siguiendo los pasos descritos en el **punto 1.3.4.3**, (*Ver CD, anexos 1.1.21-1.1.32*)

2.2.2. MODELADO DEL NEGOCIO

2.2.2.1. Identificar los casos de uso del negocio y sus actores.

Para el caso de estudio se especifican los siguientes puntos de manera que sirvan de base para el desarrollo del sistema.

Glosario del negocio

En el glosario del negocio se definen los términos importantes usados en el modelo del negocio.

Producto = Bien o servicio que la empresa brinda a los clientes.

Administrador = Es la persona encargada de administrar la información de los vendedores, clientes, pedidos, productos, ventas y rutas.

Vendedor = Persona encargada de realizar la visita y toma de pedido a los Clientes.

Cliente = Persona que visita el vendedor.

Días de visita = Son los días en los que el vendedor realiza la visita a los clientes.

Pedido = Es la lista de productos solicitados por el cliente.

Motivo de no gestión = Es la causa por la cual el cliente no realiza un pedido.

Ruta = Es la secuencia de direcciones de los clientes visitados por el vendedor.

Reglas de negocio

- Los días de visita del vendedor a los clientes son fijados por el administrador.
- El vendedor toma los pedidos de los clientes que visita.
- El vendedor registra el motivo de no gestión cuando el cliente no realiza un pedido.
- El administrador es el encargado del control de la gestión de los vendedores.

2.2.2.1.1. *Modelo de casos de uso de negocio.*

Actores

Los usuarios que participan en el negocio son:

Administrador: Es la persona encargada de administrar la información de los vendedores, clientes, productos y rutas. Monitorea gestión del vendedor.

Vendedor: Es la persona que realiza la gestión con el cliente.

Cliente: Persona que visita el vendedor.

Diagrama de Casos de Uso de Negocio

El diagrama de casos de uso se presenta en la Figura 2.1.

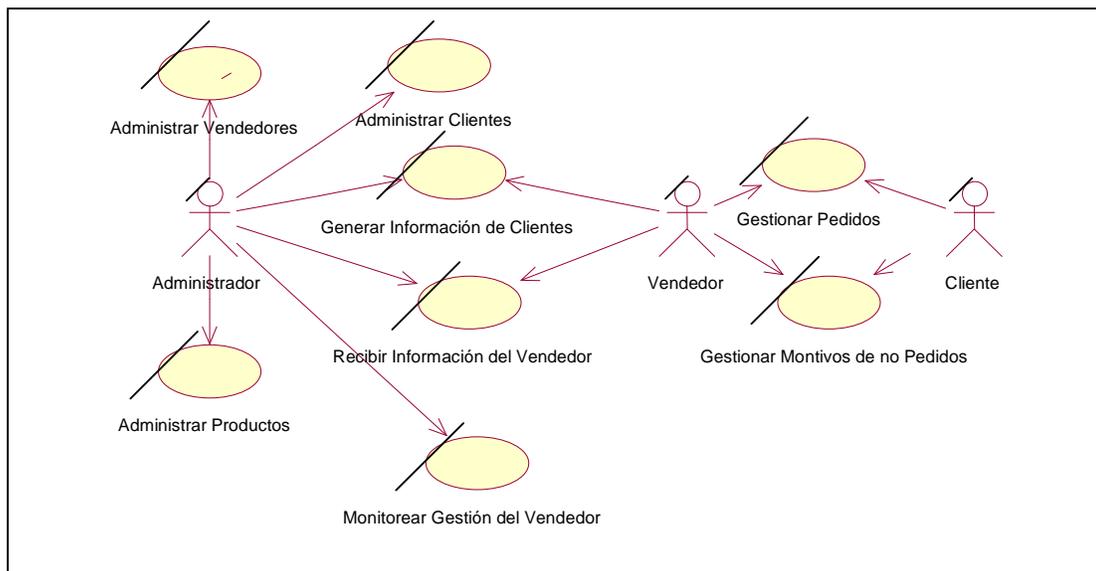


FIGURA 2.1: DIAGRAMA DE CASOS DE USO DE NEGOCIO

Descripción de Casos de Uso

Caso de uso: Administrar Clientes

Objetivo: Mantener un registro actualizado de los datos de los clientes.

Secuencia:

- El Administrador ingresa un nuevo cliente, indica: los apellidos, los nombres, el nombre del negocio, el RUC, la dirección, el teléfono, la ciudad, la provincia, el país, el vendedor responsable, la zona, la ruta, el día de visita, una observación, el estado, la latitud y la longitud.
- El sistema registra los datos del cliente.

Caso de uso: Administrar Vendedores

Objetivo: Mantener un registro actualizado de los datos de los vendedores.

Secuencia:

- a. El Administrador ingresa un nuevo vendedor, indica los apellidos, los nombres, la dirección, el teléfono, el país, el código identificador.
- b. El sistema registra los datos del vendedor.

Caso de uso: Administrar Productos

Objetivo: Mantener un registro actualizado de los productos.

Secuencia:

- a. El Administrador ingresa un nuevo producto, indica la descripción, el código, el precio unitario y el IVA.
- b. El sistema registra los datos del producto.

Caso de uso: Generar Información de Clientes

Objetivo: Obtener una lista con los clientes que el vendedor visitará.

Precondición: Tener ingresados los datos de los clientes a ser visitados, del vendedor y de los productos.

Secuencia:

- a. El administrador obtiene una lista de los clientes del vendedor.
- b. El administrador le proporciona la lista de los clientes al vendedor.

Caso de uso: Recibir Información del Vendedor

Objetivo: Recibir los pedidos de los clientes que el vendedor visitó.

Precondición: El vendedor realice la Gestión.

Secuencia:

- a. El vendedor visita a los clientes y realiza la gestión.
- b. El vendedor entrega los pedidos al administrador.
- c. El administrador valida y archiva la información de los pedidos.

Caso de uso: Monitorear Gestión del Vendedor

Objetivo: Dar seguimiento a la Gestión que realiza el vendedor.

Precondición: El vendedor realice la Gestión y entregue los pedidos al administrador.

Secuencia:

- a. El vendedor le proporciona los pedidos al administrador.
- b. El administrador monitorea los pedidos realizados por el vendedor.

Caso de uso: Gestionar Pedidos

Objetivo: Registrar los pedidos

Secuencia:

- a. El administrador entrega una lista de clientes al vendedor.
- b. El vendedor visita a los clientes.
- c. El vendedor registra el pedido.

Caso de uso: Gestionar Motivos de no Pedidos

Objetivo: Registrar los motivos por los que el cliente no realizó un pedido.

Secuencia:

- d. El administrador entrega una lista de clientes al vendedor.
- e. El vendedor visita a los clientes.
- f. El vendedor registra el motivo de no pedido.

2.2.2.2. Desarrollar un modelo de objetos de un negocio.

2.2.2.2.1. Modelo de Dominio

El modelo de dominio se presenta en el (*Ver CD, anexo 2.2.1*).

2.2.3. REQUERIMIENTOS

2.2.3.1. Obtención de los requerimientos

Requisitos candidatos

- El administrador del sistema podrá asignar los clientes a las rutas.
- El administrador del sistema podrá asignar las rutas a los vendedores.

- El vendedor podrá registrar los pedidos y los motivos por los que el cliente no realiza un pedido.
- El administrador del sistema podrá recibir la información de las rutas seguidas por los vendedores.
- El administrador del sistema podrá obtener los siguientes reportes: la ruta seguida por el vendedor, los clientes visitados por el vendedor con pedido y con motivo de no pedido y los clientes no visitados por el vendedor.

2.2.3.2. Especificación de los requerimientos de software SRS

2.2.3.2.1. Introducción

El sistema que se va a desarrollar permitirá administrar las rutas seguidas por los vendedores (establecer los clientes a visitarse por los vendedores de puerta a puerta y definir las rutas seguidas por los vendedores) y monitorear las rutas de los vendedores.

Actores

Para el presente caso de estudio los actores del sistema son: el administrador y los vendedores.

Modelo de casos de uso del Sistema

El modelo de casos de uso que se considera se presenta en la Figura 2.2.

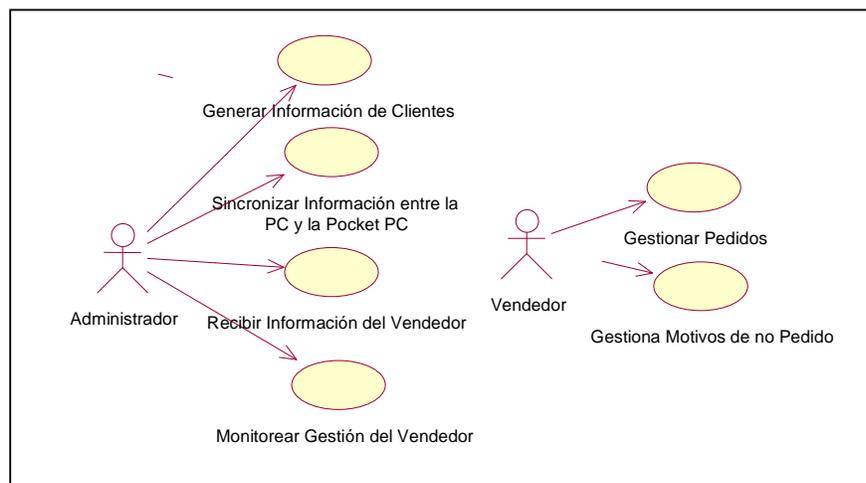


FIGURA 2.2: DIAGRAMA DE CASOS DE USO DEL SISTEMA

Casos de Uso

Se consideran los siguientes casos de usos:

Caso de uso:	Generar Información de Clientes
Actor:	Administrador
Objetivo:	Genera la información de los clientes que el vendedor visitará.
Precondición:	Tener ingresados los datos de: los clientes, de los días de visita y de los vendedores en la base de datos de la PC.
Secuencia:	<ol style="list-style-type: none"> a. El administrador seleccionará el vendedor, el día de visita y la opción ver clientes. b. El sistema le devuelve un mapa con los clientes cuyo día de visita y vendedor sean los seleccionados. c. El administrador seleccionará del mapa los clientes que el vendedor va a visitar. d. El administrador generará la información de los clientes seleccionados. e. El sistema registra la información que se generó.

Tabla 2.7: SRS Caso de Uso: Generar Información de Clientes

Caso de uso:	Sincronizar Información entre la PC y la Pocket PC
Actor:	Administrador
Objetivo:	Sincronizar la información entre la PC y la Pocket PC.
Precondición:	Base de datos de la PC. Base de datos de la Pocket PC. Publicador.
Secuencia:	<ol style="list-style-type: none"> a. El administrador sincroniza la información de la PC y la Pocket PC. b. El sistema actualiza la información de la PC y de la Pocket PC.

Tabla 2.8: SRS Caso de Uso: Sincronizar Información entre la PC y la Pocket PC

Caso de uso:	Gestionar Pedidos
Actor:	Vendedor
Objetivo:	Registrar los pedidos.
Precondición:	Sincronizar la información entre la PC y la Pocket PC.
Secuencia:	<ol style="list-style-type: none"> a. El vendedor selecciona su nombre de usuario e ingresa su contraseña. b. El sistema valida la información ingresada y presenta el menú de opciones. c. El vendedor selecciona el cliente que visita. d. El vendedor registra el pedido.

Tabla 2.9: SRS Caso de Uso: Gestionar Pedidos

Caso de uso:	Gestionar Motivos de no Pedidos
Actor:	Vendedor
Objetivo:	Registrar los motivos por los que el cliente no hizo pedido.
Precondición:	Sincronizar la información entre la PC y la Pocket PC.
Secuencia:	<ol style="list-style-type: none"> a. El vendedor selecciona su nombre de usuario e ingresa su contraseña. b. El sistema valida la información ingresada y presenta el menú de opciones. c. El vendedor selecciona el cliente que visita. d. El vendedor registra el motivo de no pedido del cliente.

Tabla 2.10: SRS Caso de Uso: Gestionar Motivos de no Venta

Caso de uso:	Recibir Información del Vendedor
Actor:	Administrador
Objetivo:	Recibir la información de la gestión realizada por el vendedor.
Precondición:	Sincronizar la información entre la PC y la Pocket PC.
Secuencia:	a. El administrador recibe la información de la ruta seguida por el vendedor. b. El sistema registra esta información.

Tabla 2.11: SRS Caso de Uso: Recibir Información del Vendedor

Caso de uso:	Monitorear Gestión del Vendedor
Actor:	Administrador
Objetivo:	Monitorear la gestión del vendedor.
Precondición:	Recibir la información del vendedor.
Secuencia:	a. El Administrador selecciona la fecha, el vendedor y el reporte que desea obtener (Reporte de la ruta seguida, de los clientes visitados que hayan realizado pedido, de los clientes visitados con motivo de no venta, de los clientes no visitados por el vendedor). b. El sistema le presenta un mapa o una lista con la información que corresponde al reporte seleccionado.

Tabla 2.12: SRS Caso de Uso: Monitorear Gestión del Vendedor

Prototipo de IU

Pantalla: Principal

El prototipo de la pantalla principal muestra un formulario con los siguientes elementos:

- Campo de texto etiquetado "Vendedor:".
- Menú desplegable.
- Campo etiquetado "Día de visita" que contiene una lista de días con casillas de selección:
 - Lunes
 - Martes
 - Miércoles
 - Jueves
 - Viernes
- Cuatro botones de acción: "Ver Clientes", "Recibir Información", "Reportes" y "Cerrar".

Botón	Función
Ver Clientes	Presentar un mapa con los clientes que son del vendedor y cuyo día de visita es el seleccionado. Activar la pantalla Mapa.
Recibir Información	Recibir la información de la gestión de los vendedores.
Reportes	Activar la pantalla de Reportes.
Cerrar	Cerrar la aplicación.

Tabla 2.13: SRS Prototipo de IU: Pantalla principal

Las pantallas restantes se encuentran en los anexos, (Ver CD, anexo 2.3.1).

2.2.3.2.2. *Requisitos funcionales*

- Desplegar un mapa con los clientes de un vendedor para un día de visita.
- Generar información de los clientes.
- Sincronizar información entre la PC y la Pocket PC.
- Desplegar los clientes del vendedor en la Pocket PC
- Registrar la Gestión de pedidos del vendedor en la Pocket PC
- Registrar la Gestión de motivos de no pedido de los clientes en la Pocket PC.
- Recibir información de los vendedores.
- Monitorear la Gestión de los vendedores, obtener los siguientes reportes: ruta seguida por el vendedor, clientes visitados por el vendedor que hayan realizado pedido, clientes visitados por el vendedor con motivo de no pedido y clientes no visitados por el vendedor.

2.2.3.2.3. *Requisitos no funcionales*

- El lenguaje de implementación es .Net.
- El sistema funcionará con el uso de una PC y de una Pocket PC.
- El sistema debe ser fácil de usar.
- El sistema requerirá de entrenamiento en el uso de la Pocket PC.
- El tiempo de respuesta de sistema estará en un rango de 2-5 seg.
- El sistema va a estar disponible tiempo completo.
- El sistema funcionará normalmente sin presentar fallas, siempre que se siga el procedimiento correctamente. Primero se genere información, luego se sincronice la PC y la Pocket, después se realice la gestión por parte del vendedor, una vez terminada la gestión se sincronice la Pocket PC con la PC y se reciba la información.
- El sistema utilizará estándares para el nombrado de variables, funciones y parámetros.

2.2.3.2.4. *Restricciones*

- El sistema no funcionará en red.
- El sistema no permitirá el registro de nuevos clientes, productos o vendedores.
- El sistema no realizará la impresión de los pedidos.

- El sistema no funcionará bajo otras condiciones de software que no sean las especificadas: plataforma Windows 98 o versiones posteriores y las herramientas: Microsoft Visual Studio .Net 2003, Microsoft Sql Server 2000, MapXtreme 2004, Service Pack 2 para Windows 2000 o Service Pack 1 para Windows XP.

2.2.3.2.5. Interfaces

Las interfaces de la PC necesitan un monitor de 14" o superior, que soporte mínimo 256 colores en una resolución de 800 x 600 pixeles. Además para manipularlas serán necesarios un teclado estándar y mouse.

Las interfaces de la Pocket PC se adaptan a las características del dispositivo disponible.

2.2.3.3. Crear el plan de prueba del sistema

El plan de pruebas considera el cumplimiento de los siguientes puntos:

- El tiempo de respuesta del sistema sea corto.
- El sistema sea fácil de usar.
- El sistema permita generar la información para el vendedor.
- El sistema permita sincronizar información entre la PC y la Pocket PC.
- El sistema permita gestionar los pedidos del vendedor.
- El sistema permita gestionar los motivos de no pedido del vendedor
- El sistema permita sincronizar información entre la Pocket PC y PC.
- El sistema permita recibir la información de la gestión del vendedor.
- El sistema permita obtener un reporte de la ruta seguida por el vendedor.
- El sistema permita obtener un reporte de los clientes visitados por el vendedor.
- El sistema permita obtener un reporte de los clientes no visitados por el vendedor con motivo de no gestión.
- El sistema permita obtener un reporte de los clientes no visitados por el vendedor.

2.2.3.4. Revisar e inspeccionar los requerimientos

Se revisa los documentos de los requerimientos y el plan de prueba, se crea la forma INS, (Ver CD, anexos 1.2.1).

2.2.3.5. Modificar los requerimientos

Después de realizar la inspección se encontró un defecto mínimo en los requerimientos de manera que se corrige el defecto que no afecta la funcionalidad del sistema. Se realiza un seguimiento del proyecto, (Ver CD, anexos 1.2.1-1.2.17).

2.2.4. ANÁLISIS

2.2.4.1. Análisis de la arquitectura

Paquetes de análisis

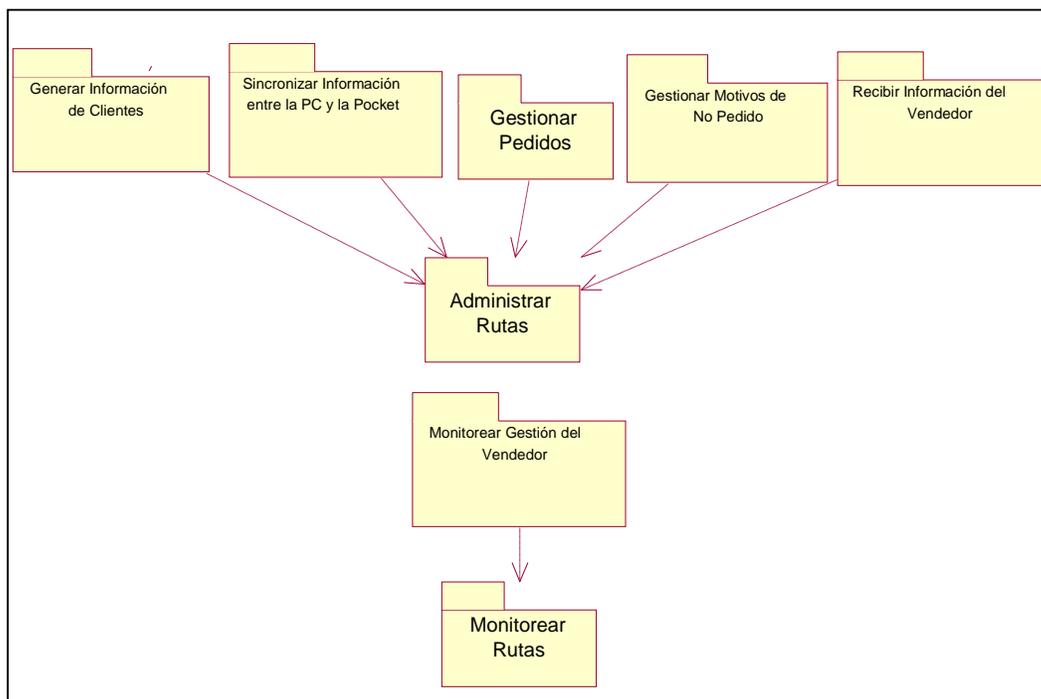


FIGURA 2.3: PAQUETES DE ANÁLISIS

Clases de entidad

VENDEDOR: Entre los atributos tendrá: apellido, nombre, dirección, teléfono, país.

CLIENTE: Entre los atributos tendrá: apellido, nombre, negocio, RUC, dirección, teléfono, ciudad, provincia, país, zona, ruta, día de visita, observación, estado, latitud y longitud.

DÍA DE VISITA: Entre los atributos tendrá: código del día y descripción.

PRODUCTO: Entre los atributos tendrá: código del producto, descripción, precio unitario, IVA.

PEDIDO: Entre los atributos tendrá: número de pedido, fecha, cantidad, código del producto, precio unitario, precio total, código del vendedor.

MOTIVO DE NO VENTA: Entre los atributos tendrá: código del motivo, descripción del motivo.

Requisitos especiales

La información generada debe ser la necesaria para sincronizar la PC con la Pocket PC, por razones de espacio físico de la Pocket PC y de tiempo de respuesta. La información generada se almacenará en tablas temporales como:

POVISITADOR_INPUT: Tendrá atributos de VENDEDOR.

POCLIENTE_INPUT: Tendrá atributos de CLIENTE.

POPRODUCTO_INPUT: Tendrá atributos de PRODUCTO.

POMOTIVO_INPUT: Tendrá atributos de MOTIVO DE NO VENTA.

POPEDIDO_OUTPUT: Tendrá atributos de PEDIDO.

PODETALLEPEDIDO_OUTPUT: Tendrá atributos de PEDIDO.

Estilo arquitectónico

El estilo arquitectónico le presentamos en los anexos, (*Ver CD, anexo 2.4.1*).

2.2.4.2. Análisis de los casos de uso

Diagramas de Colaboración

Caso de Uso: Generar Información de Clientes

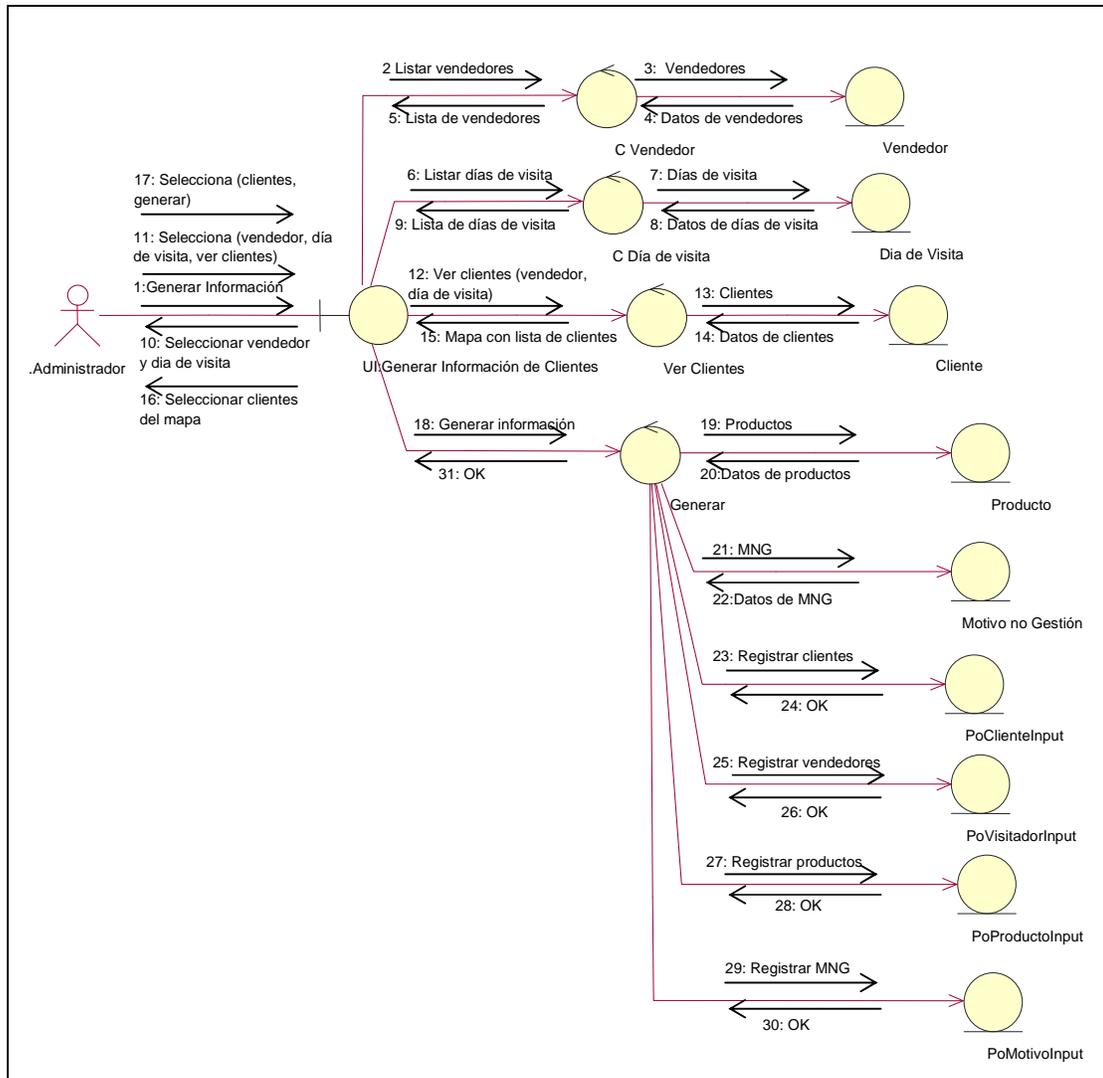


FIGURA 2.5: DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GENERAR INFORMACIÓN DE CLIENTES

Los diagramas de colaboración restantes se encuentran en los anexos, (Ver CD, anexo 2.4.2).

Modelo de análisis:

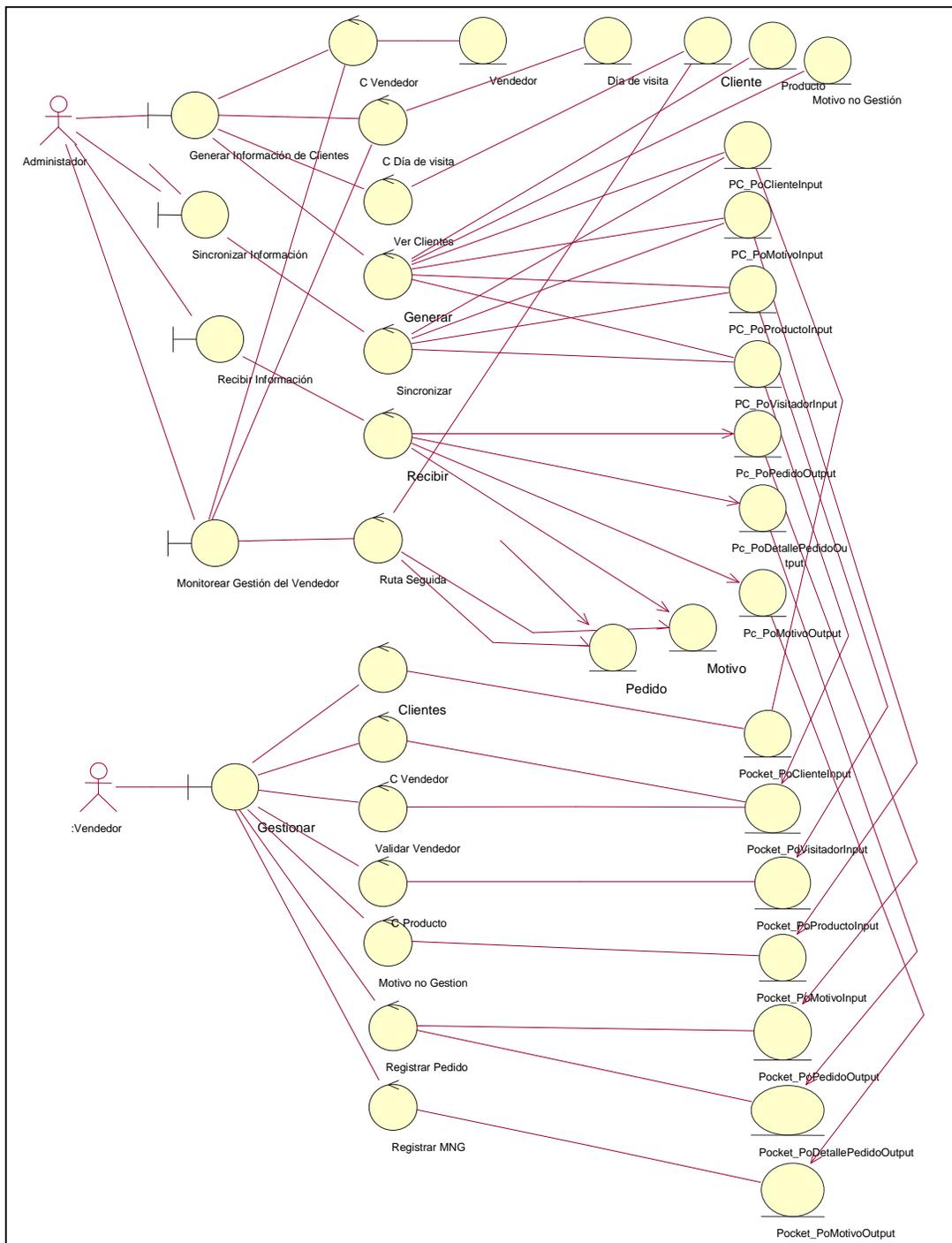


FIGURA 2.6: MODELO DE ANÁLISIS

2.2.4.3. Análisis de las clases

Identificar las responsabilidades de las clases de análisis.

Clase	Atributos	Responsabilidades
Generar Información de Clientes (PC)	"Definir IU"	Visualizar "Generar Información" Presenta lista de vendedores Presenta lista de días de visita Visualizar "Seleccione vendedor y día de visita" Leer (vendedor y día de visita, ver clientes) Presenta mapa con lista de clientes Leer (clientes, generar)
C Vendedor (PC)	(código, nombre, dirección, teléfono, login, password)	Listar vendedores Presenta datos de vendedores
Vendedor (PC)	Código, nombre, RUC, teléfono.	Obtener vendedores
C Día de visita (PC)	(días de visita)	Listar días de visita Presenta datos de días de visita
Días de visita (PC)	Código, día	Obtener días de visita
Ver Clientes (PC)		Ver Clientes (vendedor, día de visita) Lista de clientes
Cliente (PC)	Código, nombre, apellido, negocio, RUC, dirección, teléfono, provincia, ciudad, ruta, barrio, latitud, longitud.	Obtener clientes
Generar (PC)	(producto, MNG)	Generar información Lista de productos Lista de MNG
Producto (PC)	Código, descripción, unidad medida, precio, IVA	Listar productos
Motivo no Gestión (PC)	Código, descripción	Listar motivos de no gestión
PoCliente_Input (PC)	Código, nombre, RUC, razón social, dirección, teléfono, día visita, sector.	Registrar los clientes Listar clientes
PoVisitador_Input (PC)	Código, nombre, login, password.	Registrar los vendedores Listar vendedores
PoProducto_Input (PC)	Código, descripción, precio, descuento, lote, unidad de medida, stock.	Registrar los productos Listar productos
PoMotivo_Input (PC)	Código, nombre	Registrar los motivos de no gestión Listar motivos de no gestión

Tabla 2.14: Responsabilidades de las clases de análisis

El resto del análisis de clases se encuentra en los anexos, (Ver CD, anexo 2.4.3).

2.2.4.4. Análisis de los paquetes

Paquete	Caso de Uso	Actor	Hardware
Administrar Rutas	Generar Información de Clientes	Administrador	PC
	Sincronizar Información entre la PC y la Pocket PC	Administrador	PC, Pocket PC
	Gestionar Pedidos	Vendedor	Pocket PC
	Gestionar Motivos de no Gestión	Vendedor	Pocket PC
	Recibir Información del Vendedor	Administrador	PC
Monitorear Rutas	Monitorear Gestión del Vendedor	Administrador	PC

Tabla 2.15: Análisis de paquetes

Se realiza una reunión del equipo de trabajo y se hace un seguimiento del proyecto, (Ver CD, anexos 1.3.1-1.3.14).

2.2.5. DISEÑO

Para desarrollar el software planteado se procede a elaborar el diseño de alto nivel (SDS), que especifica la forma como está conformado el sistema y su funcionalidad, se realiza un diseño detallado del sistema y se crea un plan de pruebas de integración que permitirán establecer un buen diseño que facilite la implementación del sistema.

2.2.5.1. Crear un diseño de alto nivel

Software design specifications (SDS)

El producto que se va a crear “Software para Administración y Monitoreo de Rutas a seguir por los Vendedores de Puerta a Puerta” está compuesto por dos subsistemas: Administrar Rutas y Monitorear Rutas. El primer subsistema se forma de dos módulos: Generar Información de Clientes, Sincronizar Información entre la PC y la Pocket PC, Gestionar Pedidos, Gestionar Motivos de no Gestión y Recibir Información del Vendedor. El segundo se forma del módulo Monitorear Gestión del Vendedor. La estructura del sistema de acuerdo con la funcionalidad se presenta en la tabla 2.16.

El módulo Generar Información de Clientes le permitirá al administrador seleccionar el vendedor y el día de visita para obtener un mapa en el que estén representadas las direcciones de los clientes, en el cual se pueda seleccionar los clientes que el vendedor va a visitar y que se registran en la PC.

El módulo Sincronizar Información entre la PC y la Pocket PC le permitirá al administrador sincronizar los datos de la PC y la Pocket PC.

El módulo Gestionar pedidos le permitirá al vendedor registrar los pedidos de los clientes en la Pocket PC.

El módulo Gestionar motivos de no pedidos le permitirá al vendedor registrar los motivos por los que el cliente no realiza un pedido, en la Pocket PC.

El módulo Recibir Información del vendedor le permitirá al administrador registrar la información de la gestión del vendedor en la PC.

El módulo Monitorear Gestión del Vendedor le permitirá al administrador seleccionar la fecha, el vendedor y una de las siguientes opciones: un mapa con la ruta seguida por el vendedor, una lista de los clientes visitados por el vendedor que hayan realizado pedido, una lista de los clientes visitados por el vendedor con motivo de no gestión y una lista de los clientes no visitados por el vendedor.

Subsistema	Módulo	Funciones
Administrar Rutas	Generar Información de Clientes	Presentar lista de vendedores Presentar lista de días de visita Ver un mapa con los clientes. Permitir la selección de los clientes del mapa. Generar (los datos de los clientes, productos, MNG y vendedores)
	Sincronizar Información entre la PC y la Pocket PC	Sincronizar la información de la PC y la Pocket PC.
	Gestionar Pedidos	Presentar lista de vendedores Validar vendedor (vendedor, password) Ver clientes. Presentar lista de productos. Ver datos del producto. Permitir el Ingreso de datos del detalle del pedido

		Registrar un pedido.
	Gestionar Motivos de no Pedidos	Presentar lista de vendedores Validar vendedor (vendedor, password) Ver clientes. Ver MNGs Registrar MNG.
	Recibir Información del Vendedor	Recibir la información de la gestión del vendedor.
Monitorear Rutas	Monitorear Gestión del Vendedor	Presentar lista de vendedores Presentar un mapa de la ruta seguida por el vendedor. Presentar una lista de clientes con pedido. Presentar una lista de clientes con motivo de no gestión. Presentar una lista de clientes no visitados.

Tabla 2.16: SDS Subsistema, módulos y funciones

El administrador de la calidad/procesos guía el esfuerzo para producir el glosario de nombres y los estándares de diseño que se usan en el diseño (*Ver CD, anexo 2.5.1*).

2.2.5.2. Crear un diseño detallado

2.2.5.2.1. Diseño de la arquitectura

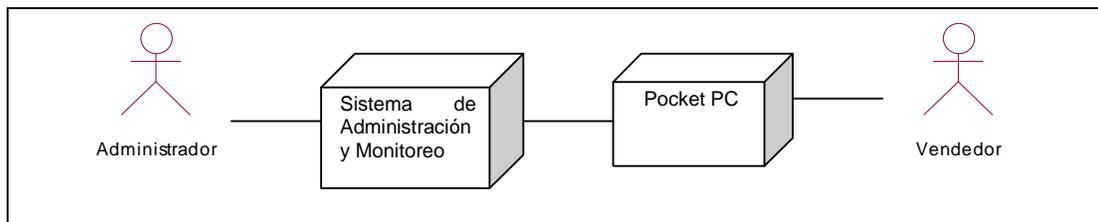


FIGURA 2.7: MODELO DE DESPLIEGUE DE LA ARQUITECTURA

2.2.5.2.2. Diseño de los casos de uso

Diagramas de secuencia

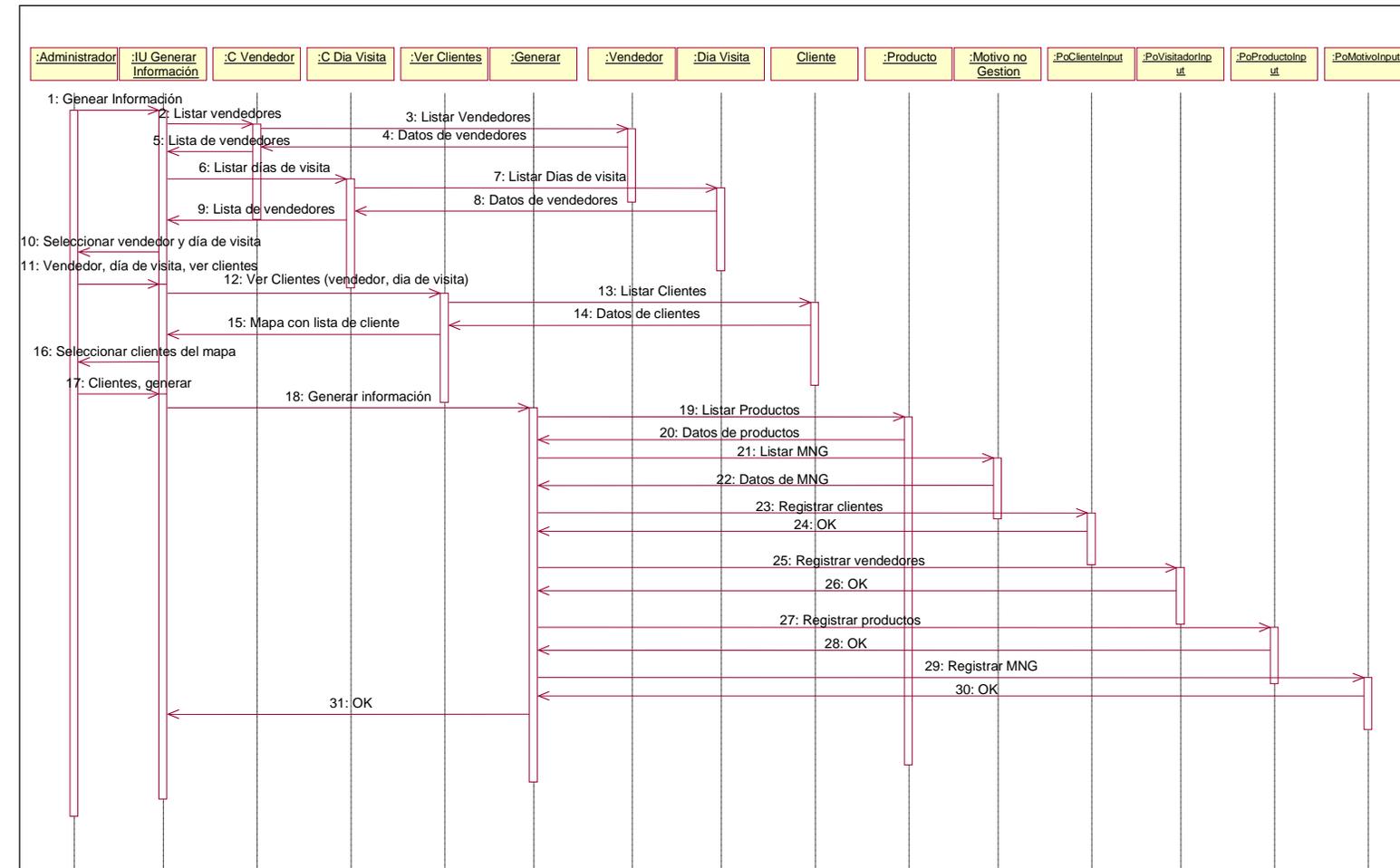


FIGURA 2.9: DIAGRAMA DE SECUENCIA DEL CASO DE USO GENERAR INFORMACIÓN DE CLIENTES

Los diagramas de secuencia de los casos de uso restantes se encuentran en los anexos, (Ver CD, anexo 2.5.3).

2.2.5.2.3. Diseño de clases

El Modelo de Clases de Diseño se encuentra en los anexos, (Ver CD, anexo 2.5.4).

2.2.5.2.4. Diseño de subsistemas

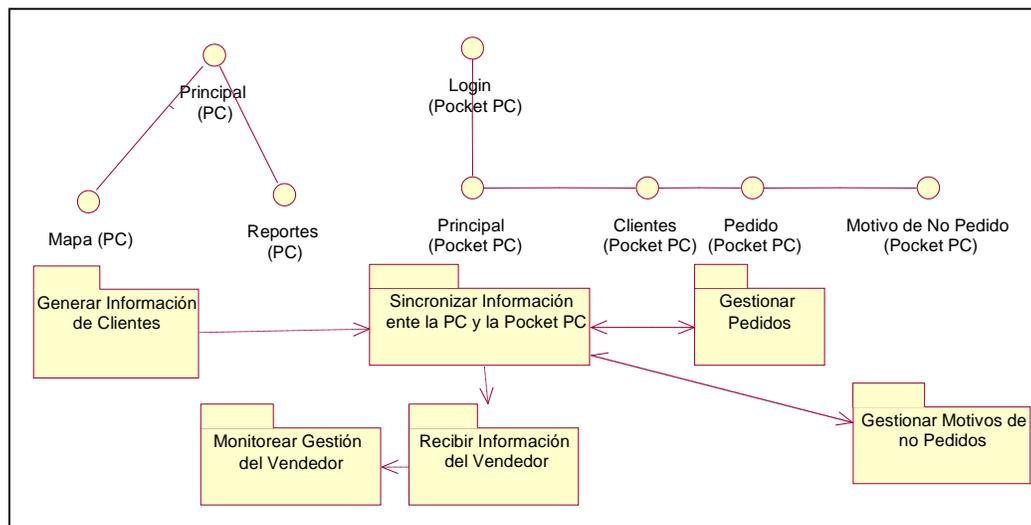


FIGURA 2.10: DISEÑO DE SUBSISTEMAS

2.2.5.2.5. Diseño de pantallas

Pantalla: Principal

Vendedor:

▼

Día de visita

Lunes
 Martes
 Miércoles
 Jueves
 Viernes

Ver Clientes

Recibir Información

Reportes

Cerrar

Botón	Activo	Al Click
Ver Clientes	Cuando se selecciona el vendedor y el día de visita	Presenta un mapa y una lista con los clientes que son del vendedor y cuyo día de visita es el seleccionado. Activa la pantalla Mapa
Recibir Información	Al cargarse la pantalla.	Recibe la información de la gestión de los vendedores.
Reportes	Al cargarse la pantalla.	Activa la pantalla de Reportes.
Cerrar	Al cargarse la pantalla.	Cierra la aplicación.

Tabla 2.17: Diseño de pantallas: Pantalla principal

El diseño de las pantallas restantes se encuentra en el CD de anexos, (Ver CD, anexo 2.5.6).

2.2.5.3. Crear un plan de pruebas de integración

El plan de pruebas considera que se deben cumplir los siguientes puntos:

En el módulo Generar Información de Clientes se cumpla con los siguientes ítems:

- Se presente una lista de los vendedores.
- Se presente los días de visita del vendedor.
- Se permita seleccionar el vendedor y el día de visita.
- Se presente un mapa con los clientes del vendedor y día de visita seleccionados.
- Se permita seleccionar los clientes para la ruta que el vendedor deberá gestionar.
- Se registren los datos de los vendedores, clientes, productos, MNG en la base de datos necesarios para la Gestión del vendedor.

En el módulo Sincronizar Información entre la PC y la Pocket PC se cumpla con los siguientes ítems:

- Cuando se va a cargar los datos para la Gestión del vendedor se actualicen los datos de la Pocket PC con los datos de la PC.
- Cuando se va a cargar los datos de la Gestión del vendedor se actualicen los datos de la PC con los datos de la Pocket PC.

En el módulo Gestionar Pedidos se cumpla con los siguientes ítems:

- Se presente una lista de los clientes.
- Se permita seleccionar un cliente de la lista para ingresar un pedido.
- Se presente una lista de los productos.
- Se permita seleccionar un producto para ingresar al detalle del pedido
- Se permita ingresar el detalle de pedido.
- Se permita eliminar el detalle de pedido.
- Se registren los pedidos.

En el módulo Gestionar Motivos de no Pedidos se cumpla con los siguientes ítems:

- Se presente una lista de los clientes.
- Se permita seleccionar un cliente de la lista para ingresar un motivo de no pedido.
- Se presente una lista de los motivos de no gestión (MNG)
- Se permita seleccionar un MNG,
- Se permita ingresar una observación del MNG.
- Se registren los MNG`s

En el módulo Recibir Información del Vendedor se registren los datos de la gestión del vendedor en la PC.

En el módulo Monitorear Gestión del Vendedor se cumpla con los siguientes ítems:

- Se presente una lista de los vendedores.
- Se permita seleccionar un vendedor
- Se permita seleccionar una Fecha.
- Se presente las opciones: Ruta seguida, Clientes visitados con pedido, Clientes visitados con MNG, Clientes no visitados.
- Se permita seleccionar una opción anterior.

- Se presente un mapa con la ruta seguida por el vendedor en la fecha seleccionada.
- Se presente una lista de los clientes visitados por el vendedor que hayan realizado un pedido.
- Se presente una lista de los clientes visitados por el vendedor que tienen un motivo de pedido.
- Se presente una lista de los clientes no visitados por el vendedor y que se establecieron en la ruta.

2.2.5.4. Revisar e inspeccionar el diseño

El administrador de la calidad/procesos guía la inspección del diseño y del plan de pruebas de integración, se llena el documento de inspección en la forma INS ver *(Ver CD, anexos 1.4.1)*.

2.2.5.5. Modificar el diseño

Con los defectos encontrados en la inspección se corrigen los documentos SDS y plan de pruebas de integración, *(Ver CD, anexo 2.5.7)*.

Se realiza una reunión del equipo de trabajo, y un seguimiento de la calidad ver desde *(Ver CD, anexos 1.4.2-1.4.17)*.

2.2.6. IMPLEMENTACIÓN

2.2.6.1. Plan para la implementación

Para realizar la implementación se mantienen los estándares de implementación *(Ver CD, anexo 2.6.1)*, y siguen los siguientes pasos:

- Se implementa la arquitectura.
- Se implementan los subsistemas.
- Se implementan las clases.
- Se revisa e inspecciona el código.
- Se ejecutan pruebas de unidad.

- Se unifica el sistema.

Se asigna a los miembros del equipo las tareas para la implementación, cada miembro actualiza las formas TASK y SCHEDULE (Ver CD, anexos 1.5.1-1.5.4), de acuerdo con el trabajo que deben desempeñar, y que se definió en las formas SUMP Y SUMQ, (Ver CD, anexos 1.4.16-1.4.17).

2.2.6.1.1. Plan de pruebas de unidad

Se sigue el script UT para desarrollar los casos de prueba, los procedimientos y los datos de prueba. Considerando el alcance del sistema las pruebas son las siguientes:

Pruebas de Lógica

- Verificar que exista conexión a la BDD para proceder a hacer una consulta, o modificación sobre la misma.
- Verificar que no existan funciones con el mismo nombre y argumentos que retornen distintas salidas.
- Verificar que los bucles funcionen correctamente y no exista un lazo infinito.
- Verificar que las funciones retornan los valores deseados.
- Verificar los cast a los tipos de datos.

Pruebas de Interfaces

- Verificar que la secuencia de interfaces en la aplicación sea correcta.
- Verificar que exista la facilidad de salir de una interfase en cualquier momento sin que el sistema colapse.
- Verificar que exista un estándar de presentación de las interfaces que facilite su uso.
- Verificar las interfaces con mensajes en caso de un error.
- Verificar que exista interfaces con mensajes de consolidación de un proceso correcto.

Pruebas de Errores

- Verificar el control del ingreso de datos incorrectos.
- Verificar el control de tipos de datos numéricos y fechas.
- Verificar la secuencia lógica de los procesos.

Pruebas de Variables

- Verificar los valores de las variables.
- Verificar los valores de los parámetros de las funciones.
- Verificar la capacidad máxima de las variables.
- Verificar los valores mínimos de las variables.
- Verificar el control sobre operaciones que involucren el 0, ninguna entrada, tipos de datos incorrectos.

Pruebas de Dispositivos

- Verificar que la Pocket PC este cargada la batería.
- Verificar que la Pocket PC este en óptimas condiciones.

2.2.6.2. Implementación de la arquitectura

Primero se instala y configura el software en la PC y en la Pocket PC, y a continuación se crean los componentes.

Componentes

Proyecto:

El proyecto "Ventas" en Visual Basic de la PC y de la Pocket PC.

Librerías:

MapInfo, System y Microsoft.

Tablas:

Las tablas de la base de datos de la PC se encuentran en la Figura 2.11, y de la Pocket PC en la Figura 2.12.

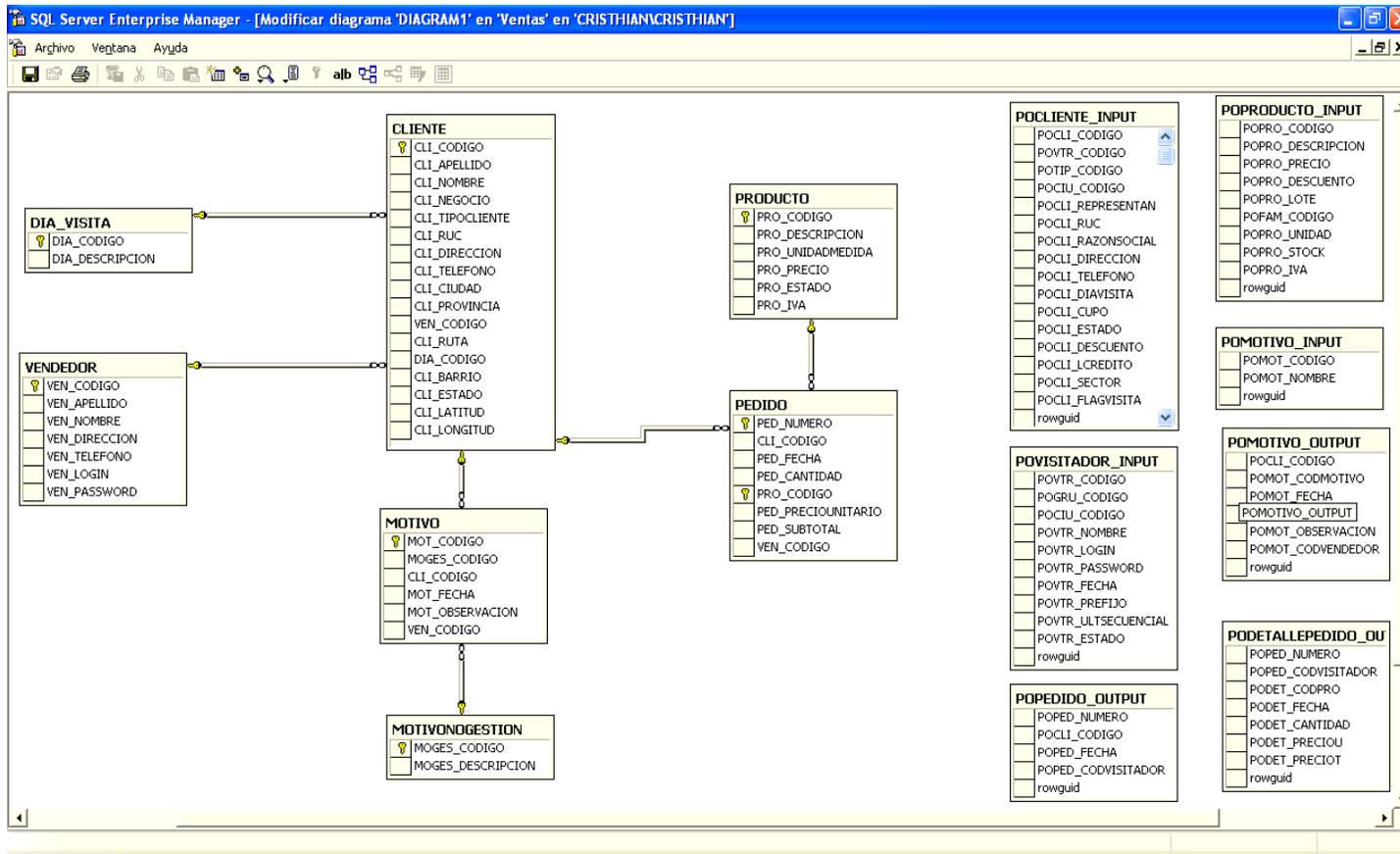


FIGURA 2.11: BASE DE DATOS DE LA PC

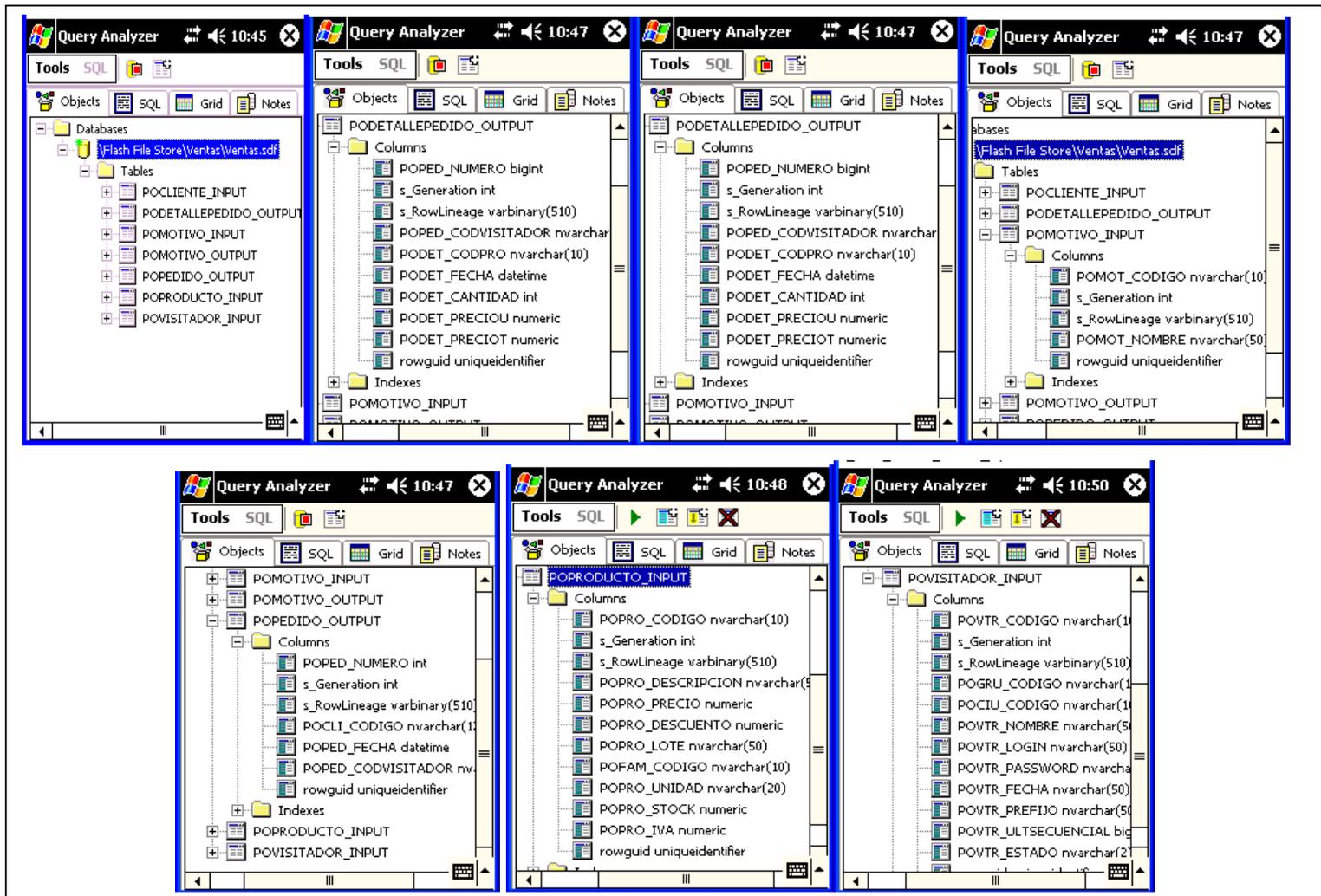


FIGURA 2.12 BASE DE DATOS DE LA POCKET PC

Publicador:

Se crea un publicador “Ventas” que permita la comunicación entre la PC y la Pocket PC.

Documentos:

Archivos EXCEL que contienen los datos de los productos, clientes, motivos de no gestión, vendedores, que sirven para ingresar información en la base de datos.

Se realiza una reunión del equipo de trabajo, y un seguimiento de la calidad ver desde (*Ver CD, anexos 1.5.5-1.5.21*).

2.2.6.3. Implementación de los subsistemas

Para el sistema se crea un proyecto “Ventas” en Visual Basic.

Subsistema: Administración de Rutas

En el módulo Generar Información de Clientes se necesita proporcionar la interfaz Mapa, entonces el componente Ver_Clientes de la interfaz Principal de la PC (frmPrincipal.vb) requiere implementar la interfaz Mapa de la PC (frmMapa.vb).

En el módulo Sincronizar Información entre la PC y la Pocket PC se implementa un componente Sincronizar en la interfaz Principal de la Pocket PC (frmPrincipal.vb).

En el módulo Recibir Información del vendedor se implementa un componente Recibir_Información en la interfaz Principal de la PC (frmPrincipal.vb).

En el módulo Gestionar Pedidos se necesita proporcionar la interfaz Principal, Clientes, Pedido, Motivo de No Gestión, la componente Validar_vendedor implementa la interfaz Principal (frmPrincipal.vb), la componente Clientes implementa la interfaz Clientes (frmClientes.vb), la componente Registrar_Pedido implementa la interfaz Pedido (frmPedido.vb).

En el módulo Gestionar Motivos de no Pedidos se necesita proporcionar la interfaz Principal, Clientes, Pedido, Motivo de No Gestión, la componente Validar_vendedor implementa la interfaz Principal (frmPrincipal.vb), la componente Clientes implementa la interfaz Clientes (frmClientes.vb), la componente Registrar_MNG implementa la interfaz Motivo de No Gestión (frmMotivosNoGestion.vb).

Subsistema: Monitorear Rutas

En el módulo Monitorear Gestión del Vendedor se necesita proporcionar la interfaz Reportes, entonces las componentes: Ruta seguida, Clientes visitados con pedido, Clientes visitados con MNG y Clientes no visitados implementan la interfaz Reportes (frmReportes.vb).

Se realiza una reunión del equipo de trabajo, y un seguimiento de la calidad ver desde (Ver CD, anexos 1.5.22-1.5.34).

2.2.6.4. Implementación de las clases

Ficheros

	Fichero	Clases de Diseño
PC	frmPrincipal.vb	Generar Información de Clientes, Recibir Información del Vendedor
	frmMapa.vb	Generar Información de Clientes
	frmReportes.vb	Monitorear Gestión del Vendedor
Pocket PC	frmLogin.vb	Gestionar Pedidos, Gestionar Motivos de no Pedidos
	frmPrincipal.vb	Gestionar Pedidos, Gestionar Motivos de no Pedidos, Sincronizar Información entre la PC y la Pocket PC:
	frmClientes.vb	Gestionar Pedidos, Gestionar Motivos de no Pedidos
	frmPedido.vb	Gestionar Pedidos
	frmMNG.vb	Gestionar Motivos de no Pedidos

Tabla 2.18: Implementación de Clases: Ficheros

Métodos de los Ficheros

PC		
Fichero	Métodos	Descripción
frmPrincipal.vb	frmPrincipal_Load()	Inicializa la pantalla.
	Cargar_diavisita()	Carga los días de visita de la BDD.
	Cargar_Combo_Vendedor()	Carga el código y nombre de los

		vendedores de la BDD.
	Cargar_Tipo_Cliente()	Carga los días de visita de la BDD.
	Recibir_Informacion_Pedidos()	Recibe la información de los pedidos de la PDA.
	Dibujar_Punto()	Dibuja los clientes en el mapa.
frmMapa.vb	frmMapa_Load()	Inicializa la pantalla.
	Cargar_Clientes()	Obtiene los datos de los clientes de la BDD y registra en una grilla
	Dibujar_Ruta()	Dibuja los clientes en el mapa
	Tools_FeatureSelected()	Obtiene los puntos seleccionados en el mapa.
	Estilo_Columna()	Define el ancho de las columnas de la grilla
	Contar_Filas()	Cuenta el número de clientes seleccionados.
	Generar_Informacion()	Guarda la información de los clientes seleccionados en las tablas temporales.
	Obtener_Guardar_Datos_Generacion()	Almacena toda la información de los clientes seleccionados.

Tabla 2.19: Implementación de Clases: Métodos de los Ficheros

El diseño de las pantallas se encuentra en el CD de anexos, (Ver CD, anexo 2.6.2).

Código fuente

DISEÑO		IMPLEMENTACIÓN	
Clase de Diseño		Objeto	Métodos
Generar Información de Clientes	C Vendedor	cmbVendedores	
	C Día de visita	cKIDias	
	Ver clientes	btnVerClientes	btnVerClientes_Click()
	Generar	btnGenerar	btnGenerar_Click()
Sincronizar Información entre la PC y la Pocket PC	Sincronizar	picSincronizacion	picSincronizacion_Click()
Gestionar Pedidos	C Vendedor	cmbVendedores	
	Validar vendedor	btnAceptar	btnAceptar_Click()
	Clientes	picVisitas	picVisitas_Click()
		dgRutero	dgRutero_Click()
	C Producto	cmbProducto	cmbProducto.SelectedIndexChanged()
	Registrar Pedido	btnAgregar	btnAgregar_Click()
		btnEliminar	btnEliminar_Click()
		mnuAceptar	mnuAceptar_Click()
		txtCantidad	txtCantidad_KeyPress()
	Motivo no Gestión	mnuCancelar	mnuCancelar_Click()
Registrar MNG	mnuAceptar	mnuAceptar_Click()	
	checkMotivoNG	Verificar_MNG()	
	txtOtros	txtOtros_TextChanged()	
Gestionar Motivos de no Pedidos	C Vendedor	cmbVendedores	
	Validar vendedor	btnAceptar	btnAceptar_Click()

	Clientes	picVisitas	picVisitas_Click()
		dgRutero	dgRutero_Click()
	Motivo no Gestión	mnuCancelar	mnuCancelar_Click()
	Registrar MNG	mnuAceptar	mnuAceptar_Click()
		checkMotivoNG	Verificar_MNG()
	txtOtros	txtOtros_TextChanged()	
Recibir Información del Vendedor	Recibir	btnRecibir	btnRecibir_Click()
Monitorear Gestión del Vendedor	Reportes	btnReportes	btnReportes_Click()
	C Vendedor	cmbVendedores	
	Ruta Seguida	btnRuta	BtnRuta_Click()
	Clientes visitados con pedido	btnClientesPedido	BtnClientesPedido_Click()
	Clientes visitados con MNG	btnClientesMNG	BtnClientesMNG_Click()
	Clientes no visitados	btnClientesnoVisitados	BtnClientesnoVisitados_Click()

Tabla 2.20: Implementación de Clases: Métodos del Código Fuente

Se realiza una reunión del equipo de trabajo, y un seguimiento de la calidad ver desde (Ver CD, anexos 1.5.35-1.5.47).

2.2.6.5. Revisar e inspeccionar la implementación

El administrador de la calidad/procesos guía la inspección de la implementación, se llena el documento de inspección en la forma INS, (Ver CD, anexos 1.5.48). Se realiza una reunión del equipo de trabajo, y un seguimiento de la calidad, (Ver CD, anexos 1.5.49 – 1.5.65).

2.2.6.6. Modificar la implementación

Realizada la revisión e inspección se corrigen los defectos encontrados en el código. Los cambios menores que no involucran cambios en la funcionalidad se realizan sin registrarlos porque no tienen impacto en el desempeño del sistema, pero se toma tiempo en realizarlos.

Se realiza una reunión del equipo de trabajo, y un seguimiento de la implementación, (Ver CD, anexos 1.5.66 – 1.5.80).

2.2.6.7. Pruebas de unidad

Se desarrollan las pruebas de la implementación considerando el plan de pruebas de unidad creado en el plan de la implementación. Se registran las formas LOGD, LOGT y LOGTEST, (*Ver CD, anexos 1.5.81 – 1.5.86*).

2.2.6.8. Revisar la calidad de los componentes

Con los datos obtenidos y los estándares de calidad (Standard Qual) se revisa la calidad de los componentes (*Ver Anexos 1.5*).

De acuerdo con los valores obtenidos se concluye que tanto el diseño de alto nivel como el diseño detallado fueron bien elaborados, ya que no se encontraron defectos graves que afecten la lógica del diseño.

El plan de calidad creado fue muy optimista, de acuerdo a los valores obtenidos se encontraron defectos en el código pero que no afectaban la funcionalidad del sistema, ya que no eran graves. Se crearon funciones que durante la codificación se requerían pero que después de realizar las pruebas de unidad se ratificó que no eran necesarias. También se pudo ver que para que el código sea re usable se requería de comentarios que fueron añadidos durante la revisión e inspección del código.

El código implementa todos los componentes que especificaron en el diseño. Y cada componente cumple con la funcionalidad descrita en el SDS.

Una vez concluida las pruebas de unidad el código es estable, entendible, seguro, ya que todos los defectos encontrados fueron resueltos.

Se realiza una reunión del equipo de trabajo, y un seguimiento de la calidad, (*Ver CD, anexos 1.5.87 – 1.5.99*).

2.2.6.9. Integrar las componentes en un sistema ejecutable.

Con los componentes funcionando correctamente se procede a integrarles en un solo sistema ejecutable y a crear un manual de usuario (*Ver CD, anexo 2.6.5*).

Se realiza una reunión del equipo de trabajo, y un seguimiento de la calidad, (Ver CD, anexos 1.5.100 – 1.5.114).

2.2.7. PRUEBAS

2.2.7.1. Planificar las pruebas

Estrategia de Prueba

Las pruebas se realizarán considerando los valores reales que deberían obtenerse del sistema y comparándoles con los que se obtienen del sistema. Para obtener los valores reales se procederá a realizar consultas a la base de datos y los cálculos aritméticos necesarios en el EXCEL.

Criterio de éxito: Los valores reales deben ser los mismos que los valores que arroja el sistema.

Requisitos para realizar las pruebas

Para realizar las pruebas se requiere disponer de:

- PC y Pocket PC con las herramientas descritas en el diseño del proyecto.
- La aplicación ejecutándose correctamente tanto en la PC como en la Pocket PC.

2.2.7.2. Diseñar las pruebas

2.2.7.2.1. Pruebas de integración

El plan de pruebas de integración considera que se deben cumplir los siguientes puntos:

Caso de prueba 1: Ver clientes Julio León día lunes.

Entrada:

- Existe un vendedor llamado Julio León que realiza visitas a los clientes el día lunes.

- Existen clientes del vendedor Julio León para el día de visita lunes.

Resultado:

- Un mapa en el que se visualicen los 28 clientes que le pertenecen al vendedor Julio León para el día de visita lunes.

Procedimiento de la prueba:

- Seleccionar el vendedor Julio León y el día de visita lunes y la opción Ver clientes de la pantalla principal. Se abre la pantalla Mapa.
- Se verifica que los clientes se presenten en el mapa de la pantalla Mapa y que el número de clientes sea 28.

Los casos de prueba restantes se encuentran en los anexos (*Ver CD, anexo 2.7.1*).

2.2.7.2.2. Pruebas del sistema

El plan de pruebas del sistema considera que se deben cumplir los siguientes puntos:

Caso de prueba 1: Administrar Rutas.

Entrada:

- Tener almacenada la información del vendedor, clientes, productos, motivos de no gestión, días de visita en la base de datos.

Resultado:

- Creadas las Rutas para la gestión del vendedor.
- Definidas las Rutas después de la gestión del vendedor.

Procedimiento de la prueba:

- Generar información para el vendedor
- Sincronizar información entre la PC y la Pocket PC.
- Realizar la Gestión del vendedor (Gestionar Pedidos, Gestionar motivos de no pedidos MNG).
- Sincronizar información entre la Pocket PC y la PC.
- Recibir la información

Caso de prueba 2: Monitorear Rutas.*Entrada:*

- Haber realizado la Administración de Rutas

Resultado:

- Reporte de la ruta seguida por el vendedor.
- Reporte de los clientes visitados por el vendedor.
- Reporte de los clientes no visitados por el vendedor con motivo de no gestión.
- Reporte de los clientes no visitados por el vendedor.

Procedimiento de la prueba:

- Administrar la Ruta
- Monitorear la Ruta

2.2.7.3. Realizar las pruebas.*2.2.7.3.1. Realizar pruebas de integración***Prueba 1:** Ver clientes Julio León día lunes.*Resultado de la prueba:*

- Un mapa en el que se visualizan 28 clientes que le pertenecen al vendedor Julio León para el día de visita lunes.

Resultado esperado:

- En la base de datos debe haber 28 clientes del vendedor Julio León para el día lunes.

Conclusión:

- Tanto el resultado de la prueba como el resultado esperado coinciden.

Las pruebas restantes se encuentran en los anexos (*Ver CD, anexo 2.7.2*).

*2.2.7.3.2. Realizar pruebas del sistema***Caso de prueba 1:** Administrar Rutas.*Resultado de la prueba:*

- Creadas las Rutas para la gestión del vendedor.

- Definidas las Rutas después de la gestión del vendedor.

Resultado esperado:

- Registrados los clientes del vendedor para el día de visita seleccionado en la PC y en la Pocket PC.
- Registrados los clientes que el vendedor visitó para el día de visita seleccionado en la PC y en la Pocket PC.

Conclusión:

Tanto el resultado de la prueba como el resultado esperado coinciden.

Caso de prueba 2: Monitorear Rutas.

Resultado de la prueba:

- Reporte de la ruta seguida por el vendedor.
- Reporte de los clientes visitados por el vendedor.
- Reporte de los clientes no visitados por el vendedor con motivo de no gestión.
- Reporte de los clientes no visitados por el vendedor.

Resultado esperado:

- Se presente un reporte de la ruta seguida por el vendedor.
- Se presente un reporte de los clientes visitados por el vendedor.
- Se presente un reporte de los clientes visitados por el vendedor con motivo de no gestión.
- Se presente un reporte de los clientes no visitados por el vendedor.

Conclusión:

Tanto el resultado de la prueba como el resultado esperado coinciden.

El criterio de éxito que se plantea en el plan de las pruebas se cumplió, es decir: los valores reales son los mismos que los valores que arroja el sistema.

2.2.7.4. Evaluar las pruebas.

Las pruebas que se realizaron al sistema no arrojaron defectos, se cumplió con el criterio de éxito que se planteó en el plan de las pruebas, ya que los resultados de las pruebas fueron iguales a los valores esperados o considerados reales.

Entonces se concluye que el sistema responde como se esperaba, cumple con los requerimientos funcionales y con los requerimientos no funcionales descritos en el SRS.

El sistema solo se acepta el ingreso de la cantidad del producto para el detalle del pedido, de manera el control que se hace es el ingreso de solo números en este campo. El resto de controles del sistema controlan la falta de requisitos para proceder a una función. Por lo que el sistema esta controlado.

Debido al alcance del sistema no se encontraron defectos durante la realización de pruebas de integración y del sistema.

2.2.7.5. Registrar las pruebas

Se registran las pruebas en el LOGTEST (*Ver CD, anexo 1.6.1, 1.6.2*).

Se documentan las pruebas. Se realiza una reunión del equipo de trabajo, y un seguimiento de la calidad ver desde (*Ver CD, anexos 1.6.3– 1.6.16*).

2.2.8. POSTMORTEM

2.2.8.1. Revisión de los datos

El administrador de la calidad/procesos guía al equipo en la realización de las siguientes actividades:

- Examinar los datos que el equipo y los miembros del equipo hicieron.

Se examinan los datos que el equipo y los miembros del equipo hicieron durante las fases de desarrollo del producto.

- Identificar donde trabajaron los procesos y donde no lo hicieron.

El producto se elaboró en un solo ciclo de manera que se trabajo en completar todas las fases definidas en la integración de TSP y RUP.

- Comparar el desempeño del equipo con sus metas y planes.

De acuerdo a los resultados obtenidos se cumple con la mayoría de las metas definidas de acuerdo con el plan propuesto.

- Identificar áreas problema y necesidades de mejora.

Por ser este el primer proyecto que el equipo desarrolla usando TSP, se requirió de tiempo para adaptarse al uso. Con la experiencia adquirida para próximos proyectos el tiempo que se necesitará será menor.

- Delegar el mejoramiento de procesos.

Al sistema desarrollado se le puede incorporar nuevas funcionalidades. El objetivo del sistema era aplicar la teoría de la combinación de TSP y RUP, por lo tanto se definieron funciones básicas que pueden ser mejoradas en próximos proyectos.

2.2.8.1.1. Revisión de la calidad

El administrador de la calidad/procesos guía al equipo en el cumplimiento de los siguientes objetivos:

- Se compara el desempeño del equipo y de cada ingeniero con el plan de calidad.

El equipo se enfatizó en desarrollar un producto final de buena calidad con cero defectos y estandarizado, en el menor tiempo posible. De acuerdo con el sistema y la funcionalidad del mismo se acoplaron los tiempos para obtener como resultado un sistema que responde satisfactoriamente.

- Se analizan los datos de los defectos del equipo y se evalúa el grado con el que el equipo produce un producto de calidad.

Los defectos que se presentaron durante el desarrollo en su mayoría fueron leves. Dada la experiencia en el desarrollo de este tipo de sistemas. El equipo se enfocó en crear bien el SRS, SDS y el diseño detallado para no encontrarse con defectos graves en la implementación. También se enfocó en corregir los defectos encontrados en la misma fase, no postergar la resolución para fases posteriores. El objetivo era tener un sistema con cero defectos graves para las pruebas de integración y del sistema.

- Se evalúa donde los procesos tienen fallas pequeñas y que se puede hacer para mejorar estas en un futuro.

El equipo tenía un tiempo reducido diario para el desarrollo del proyecto que hizo que el tiempo de duración en semanas sea grande. Se recomienda que para futuros proyectos se disponga de mayor tiempo diario para la dedicación a un proyecto.

- Se prepara el PIPs con las sugerencias de mejoramiento. Se evalúa el desempeño del equipo y de los miembros con las metas.

PIPs se encuentra en los anexos (*ver anexo 1.6.1*). De acuerdo con las metas definidas por el equipo se cumplieron en su totalidad.

Meta	Equipo	
1	Desarrollar un sistema usando TSP y RUP.	Cumplida
2	Producir un sistema de acuerdo con el plan	Cumplida
3	Producir un sistema de calidad.	Cumplida

Tabla 2.21: Revisión de la calidad: Metas del equipo

De acuerdo con las metas definidas cada miembro del equipo cumplió en su totalidad.

Miembro 1

Meta		
1	Desarrollar un sistema usando TSP y RUP.	Cumplida
2	Participar activamente en el equipo, en el desarrollo del sistema.	Cumplida
3	El producto desarrollado cumpla con normas de calidad.	Cumplida

Tabla 2.22: Revisión de la calidad: Metas del miembro 1 del equipo

Miembro 2

Meta		
1	Desarrollar un sistema que involucre la integración de TSP y RUP.	Cumplida
2	Desarrollar el sistema siguiendo estándares y un plan.	Cumplida
3	Producir un sistema de calidad	Cumplida

Tabla 2.23: Revisión de la calidad: Metas del miembro 2 del equipo

Evaluación del sistema según los factores de calidad:

Factores externos	
Correctitud	El sistema produce las salidas esperadas
Confiabilidad	El 100% de los resultados del sistema son correctos.
Robustez	El sistema considera el control de condiciones de error.
Compatibilidad	El sistema puede ser compatible fácilmente con otro que realice el registro de datos en la base de datos de la PC. También puede ser compatible con un sistema dedicado a la impresión de las facturas ya que el sistema permite crear los pedidos.
Eficiencia	El sistema necesita de la PC y la Pocket PC.
Facilidad de Uso	El sistema es fácil de usar.
Funcionalidad	El sistema cumple con las funciones especificadas en los requerimientos.
Integridad	El sistema mantiene los datos que manipula íntegros
Usabilidad	El sistema permitirá al administrador de rutas mantenerse bien informado del trabajo que realizan los vendedores, ya que provee de datos reales de la gestión del vendedor. El sistema esta enfocado a reducir el tiempo dedicado al registro de pedidos manuales.

Tabla 2.24: Revisión de la calidad del sistema: Factores externos

Factores internos	
Flexibilidad	El sistema es flexible a cambios de código ya que este esta estandarizado, comentado y entendible.
Portabilidad	El sistema funciona bajo condiciones específicas: requiere de un Administrador y Vendedores, de una PC y de Pocket PC's con el software definido.
Facilidad de mantenimiento	Se tiene documentación del software lo que facilita entender, corregir, adaptar o mejorar el sistema.
Facilidad de pruebas	Para las pruebas del sistema se requiere de la PC y la Pocket PC con el software necesario.
Reusabilidad	El código es entendible y práctico se compone de funciones que pueden ser re usadas posteriormente en nuevas aplicaciones. Al sistema se le puede incorporar nuevas funcionalidades esta adaptado para ello.
Puntualidad	Para instalar el sistema se requiere disponer de una PC con el software. Cada componente del sistema podría haber sido puesta ejecución gradualmente.
Interoperabilidad	El sistema no se llevo a la práctica por tal motivo este factor de calidad no se midió, pero se espera que pueda interactuar con otros sistemas como: un sistema de registro de datos (porque el sistema requiere los datos de vendedores, productos, clientes, motivos de no gestión registrados en la base de datos), un sistema dedicado a la impresión de las facturas (porque el sistema permite crear los pedidos), un sistema de monitoreo de ventas (porque el sistema provee de los datos de la gestión de los vendedores).

Tabla 2.25: Revisión de la calidad del sistema: Factores internos

2.2.8.2. Evaluaciones del desempeño de los roles

Se evalúan los roles del equipo enfocándose en los objetivos cumplidos. Para la evaluación de los roles se prepara la forma PEER (*ver anexos 1.6.2- 1.6.3*).

Evolución del rol: Líder del Equipo, Administrador de Planificación y de la Calidad/Procesos.

Meta	Descripción	Estado
1	Construir y mantener un equipo efectivo.	Cumplida
2	Motivar a todos los miembros del equipo a trabajar agresivamente en el proyecto	Cumplida
3	Resolver problemas entre los miembros del equipo.	Cumplida
4	Mantener al instructor totalmente informado sobre el progreso del equipo.	Cumplida
5	Desempeñar efectivamente como facilitador de las reuniones del equipo.	Cumplida
6	Producir un completo, preciso y exacto plan para el equipo y todos los miembros del equipo.	Cumplida
7	Reportar de forma exacta el estado del equipo todas las semanas.	Cumplida
8	Todos los miembros del equipo exactamente reportan y usan apropiadamente los datos TSP.	Cumplida
9	El equipo sigue fielmente TSP y produce un producto de calidad.	Cumplida
10	Todas las inspecciones del equipo son moderadas adecuadamente y reportadas.	Cumplida
11	Todas las reuniones del equipo son reportadas con precisión y los reportes son puestos en el Project Notebook.	Cumplida

Tabla 2.26: Evaluación del desempeño de los roles 1

Evolución del rol: Administrador del Desarrollo y de Soporte

Meta	Descripción	Estado
1	Producir un producto de alta calidad.	Cumplida
2	Utilizar totalmente las destrezas y habilidades de los miembros del equipo	Cumplida
3	El equipo tiene todas las herramientas apropiadas y métodos de soporte de su trabajo.	Cumplida
4	No hay cambios sin autorización sobre el producto base.	Cumplida
5	Todos los riesgos del equipo y los problemas están registrados en el registro de seguimiento de problemas (ITL) y reportados en cada semana.	Cumplida
6	El equipo tiene metas re usables para el ciclo de desarrollo.	Cumplida

Tabla 2.27: Evaluación del desempeño de los roles 2

Los roles fueron correctamente desempeñados por los miembros del equipo de acuerdo con las metas cumplidas. Se sugiere que para una mejor apreciación del desempeño de roles se involucre un número mayor de personas en el desarrollo

de un proyecto. Esto permitirá disminuir el tiempo total de desarrollo al establecer mayores tiempos diarios de trabajo.

2.2.8.3. Preparar reportes

2.2.8.3.1. Preparar el reporte del ciclo

El producto a desarrollar es un Software para Administración y Monitoreo de Rutas a seguir por los Vendedores de Puerta a Puerta, para el desarrollo se siguieron los flujos definidos en la combinación de TSP y RUP: Gestión de Proyectos, Modelado del Negocio, Requerimientos, Análisis, Diseño, Implementación, Pruebas, Postmortem, y Distribución.

Por ser un caso de estudio se siguieron los flujos hasta Postmortem. El equipo de trabajo esta conformado por 2 miembros, esto hizo que el tiempo de desarrollo se alargue. Para una próxima vez se sugiere conformar un equipo con más miembros lo que permitirá apreciar de mejor manera la utilidad de TSP en el desarrollo de un proyecto.

El equipo designo los roles de la siguiente manera: miembro 1 (los roles: Líder del Equipo, Administrador de Planificación y de la Calidad/Procesos), miembro 2 (los roles: Administrador del Desarrollo y de Soporte). El equipo cumplió con las metas propuestas para el desarrollo, tanto las metas personales como las que se deben cumplir con el rol que se desempeña.

Del desarrollo se pudo destacar que al tomar en cuenta una planificación y un control de las actividades se motiva al miembro del equipo ha trabajar para poder reportar al equipo el resultado de su trabajo. También al definir los requerimientos, realizar un análisis y un diseño de lo que se quiere desarrollar, la implementación resulta menos difícil porque se tiene claro que se necesita del sistema.

Cuando existen responsabilidades ajenas al proyecto que se interponen en el desarrollo del mismo hacen que el desarrollo se retrase. Se sugiere que para el desarrollo de un proyecto se le destine tiempo prioritario que no se vea afectado por otras responsabilidades que cada miembro tenga.

Durante el desarrollo se puede apreciar que algunas actividades requieren mayor tiempo de lo planificado. Es recomendable ajustar los tiempos semanales de los miembros para cumplir con las actividades y no posponer éstas ya que el tiempo de desarrollo se alargaría.

Por ser un caso de estudio el desarrollo se enfocó en un solo ciclo de desarrollo puesto que el objetivo era identificar la funcionalidad del TSP en el desarrollo.

2.2.8.3.2. Preparar el reporte de los roles

El equipo designó los roles: Líder del Equipo, Administrador de Planificación y de la Calidad/Procesos al miembro 1, y los roles: Administrador del Desarrollo y de Soporte al miembro 2. Los miembros del equipo cumplieron con las metas propuestas por el rol asignado, por lo que se califica como un buen desempeño de los roles establecidos.

Los roles y las responsabilidades están descritas en los anexos (*ver anexo 2.1*), en donde se puede identificar las responsabilidades que un miembro del equipo según el rol que tenga debe desempeñar. Por estar el equipo conformado por dos miembros cada persona tenía bastantes responsabilidades. Se recomienda que para un próximo proyecto el número mínimo de personas sea dependiendo del número de roles a desempeñar. Esto permitirá obtener mejor resultado de cada rol porque cada persona tiene menos responsabilidades juntas que cumplir.

2.2.8.3.3. Preparar el reporte por miembro del equipo

Miembro 1

Al final del proyecto se cumplió con las metas personales que se fijaron al inicio, durante el desarrollo existieron ciertas dificultades extras que hicieron que se alargue el tiempo semanal, pero que permitió el termino del mismo con éxito ya que la funcionalidad del sistema es buena.

Los requerimientos del sistema estaban claros lo que permitió que los flujos posteriores sean fáciles de realizar, que los defectos graves sean pocos. Esto garantizó que el producto obtenido es de buena calidad.

Para un próximo proyecto se debe tomar en cuenta el tiempo que se debe dedicar al mismo. Si se requiere que el tiempo de desarrollo sea corto deberá existir mayor tiempo diario de dedicación al mismo.

Miembro 2

Se cumplió con las metas personales establecidas inicialmente. Se cumplió con el plan sobrellevando algunas dificultades de carácter personal que impedían el desarrollo del proyecto.

El sistema funciona según los requisitos establecidos, es estable, tiene control de errores, estándares, es simple lo que garantiza una buena calidad del sistema.

Para otro ciclo de desarrollo se debería considerar bien los tiempos, funcionalidades extras requerirán cumplir con los estándares del primer ciclo esto ayudará a hacer más fáciles los flujos de trabajo.

2.2.8.4. Producción del reporte

El reporte es la documentación del Proyecto de Titulación conformado por el capítulo 2.

Se realiza una reunión del equipo de trabajo, y un seguimiento de la calidad ver desde (*Ver CD, anexos: 1.7.4 – 1.7.17*).

Completado el flujo Postmortem se tiene completo el Project Notebook con las formas de los flujos de la combinación de TSP y RUP (ver CD anexo 1).

2.3. EVALUACIÓN

2.3.1. EVALUACIÓN DEL PROCESO

Para definir el uso de TSP en el desarrollo de software se realizó la combinación de los procesos TSP y los flujos de la metodología estándar de desarrollo RUP. Se consideraron las actividades que describen tanto TSP como RUP y se tomó como base la estructura de RUP ya que esto facilitaría el desarrollo del software. Es así que de la combinación de TSP y RUP se definieron los flujos: Gestión de Proyectos, Modelado del Negocio, Requerimientos, Análisis, Diseño, Implementación, Pruebas, Postmortem y Distribución.

En la combinación, el primer flujo se denominó Gestión de Proyectos porque abarca los flujos de soporte de los procesos de trabajo de RUP (Gestión de cambios y configuración y Gestión de proyectos), y los procesos de TSP (Lanzamiento del Proyecto de Equipo, Desarrollo de la estrategia y Desarrollo del plan).

Gestión de Proyectos abarca las actividades que definen la organización del proyecto, las metas, los riesgos, la estrategia, el procedimiento para cambios y el plan para el desarrollo del sistema. Al cumplir con estas actividades se tienen claras las responsabilidades de cada miembro y del equipo, las metas que se deben alcanzar, las dificultades que se pueden presentar, la mejor estrategia que se debe cumplir, el procedimiento en caso de algún cambio para tener control sobre el sistema y el plan que se debe seguir para alcanzar las metas y desarrollar el sistema.

TSP no define actividades para el Modelado del negocio y el Análisis, por lo tanto en estos flujos se siguen las actividades definidas por RUP y se realiza un reporte de tareas y tiempos al finalizar éstos con el propósito de seguir una secuencia lógica de los reportes.

En el flujo Requerimientos las actividades de RUP se compactaron en el SRS. Se añadió la creación del plan de pruebas del sistema definido en TSP ya que éstas

están dirigidas a asegurar que el sistema cumpla con las funciones especificadas en el SRS.

En el flujo Diseño, las actividades de RUP se compactaron en el diseño detallado definido en el proceso implementación de TSP. Se añadieron las actividades: Diseño de alto nivel que describe el sistema de manera general y el Plan de pruebas de integración de los componentes del sistema que posteriormente sirve para realizar las pruebas de integración en el flujo Pruebas.

El flujo Implementación se enfoca directamente en la codificación del sistema. Se crea un plan que permite definir los pasos a seguir para la implementación, se realizan las pruebas de unidad que permiten resolver los defectos que los componentes del sistema tienen antes de integrarse en un sistema ejecutable. Esto garantiza que después de la integración los defectos existentes no sean grandes.

Las actividades del flujo Pruebas se definen tanto en las pruebas de RUP como en las pruebas de TSP de manera que resulte fácil la integración teórica y el desarrollo de este flujo durante la creación del sistema.

El flujo Postmortem se añadió después del flujo Pruebas de RUP para completar con el propósito de TSP que es proveer una estructura de aprendizaje y mejoramiento que permite identificar oportunidades de mejoramiento para los siguientes ciclos de desarrollo o para otros proyectos.

El flujo final, Distribución, se define como un flujo de soporte de los procesos de trabajo, cuyo propósito es asegurar la disponibilidad del producto de software para los usuarios finales. No existe un enfoque de TSP dedicado a este propósito, debido a que esta buena práctica no está dirigida al usuario final sino al equipo de desarrollo.

2.3.2. EVALUACIÓN DEL PRODUCTO

Para evaluar la calidad del producto se realizó una encuesta a un grupo de 5 personas (ver anexo).

Requisitos del encuestado:

- Haber usado un computador anteriormente.
- Tener claro el proceso que realizan los vendedores durante la toma de pedidos, previa una explicación.
- Conocimiento básico del uso de un dispositivo móvil.

Se realizó un análisis de las respuestas a las encuestas lo que permitió establecer las siguientes conclusiones sobre la calidad del producto creado.

- El grado de satisfacción del software es alto debido a que el sistema cumple con los requisitos establecidos. Además ofrece la novedad del uso de un dispositivo móvil y de presentación grafica de los clientes en un mapa.
- En las recomendaciones para mejorar la aplicación tenemos aspectos irrelevantes con respecto a la funcionalidad del sistema.
- La presentación visual de los clientes en el mapa son de las cosas que más llamaron la atención a los encuestados, así como la aplicación en el dispositivo móvil.
- Este tipo de aplicaciones ahorrarían tiempo y dinero en los procesos de visita los cuales muchas de las empresas las realizan manualmente, además que permite obtener un seguimiento minucioso de la gestión realizada por cada uno de los vendedores.

A continuación se presenta el nivel de satisfacción de los factores de calidad:

Factor	Nivel	
Facilidad de uso	alto	Todos los encuestados no tuvieron problemas al usar el sistema
Correctitud	alto	Los resultados presentados al usuario fueron correctos.
Confiabilidad	alto	Los valores calculados durante el proceso de toma de pedido fueron correctos. El sistema funcionaba correctamente.
Robustez	alto	El sistema posee controles para el ingreso de información
Compatibilidad	alto	La comunicación entre la PC y la Pocket PC es fluida.
Eficiencia	alto	Para la gestión en la Pocket PC se genera información específica de un vendedor, lo que disminuye el tiempo de respuesta, el tamaño de memoria a utilizarse.
Funcionalidad	alto	Para administrar y monitorear las rutas se usa la PC y la Pocket PC indistintamente, pero interactúan para completar el ciclo del proceso.
Integridad	alto	La información procesada en la PC y la Pocket PC permanecen íntegros durante todo el proceso.

CAPITULO 3

3. CONCLUSIONES Y RECOMENDACIONES

3.1. CONCLUSIONES

- El proceso unificado de desarrollo RUP es una metodología estándar de desarrollo y TSP es una buena práctica que sirve para el mejoramiento continuo de la calidad. RUP y TSP dividen el proceso de desarrollo en ciclos, teniendo un producto final al término de cada ciclo. TSP sirve para el desarrollo y mantenimiento de equipos de 2 a 20 ingenieros.
- TSP al igual que PSP es un proceso que se compone de un conjunto de métodos, formatos y reglas que ayudan al desarrollo en equipos de un proyecto de gran calidad y madurez. TSP junto con PSP y CMM son las herramientas ideales para ayudar a las organizaciones en el desarrollo de sistemas de software de alta calidad.
- Para que el desarrollo de un proyecto tenga éxito es primordial que el equipo de trabajo cumpla con ciertas condiciones básicas: cohesión, metas, informes y un ambiente común de trabajo.
- RUP divide cada ciclo en fases que finalizan con un hito y cada fase está formada por flujos de trabajo, mientras que TSP divide cada ciclo en procesos que finalizan con la producción de software que cumple con algunas características de los requerimientos de software.
- Para definir el uso de TSP en el desarrollo de software se realizó la combinación de los procesos TSP y los flujos de la metodología estándar de desarrollo RUP, ya que el perfil al que se orientaban era similar en ambos casos.

- De la combinación de TSP y RUP se obtuvieron como resultado los siguientes flujos: Gestión de Proyectos, Modelado del Negocio, Requerimientos, Análisis, Diseño, Implementación, Pruebas, Postmortem y Distribución.
- Se definió como primer flujo Gestión de Proyectos porque abarca los flujos de soporte de los procesos de trabajo de RUP (Gestión de cambios y configuración y Gestión de proyectos) y los procesos de TSP (Lanzamiento del Proyecto de Equipo, Desarrollo de la Estrategia y Desarrollo del Plan), que son previos al desarrollo del sistema.
- Cuando se cumplen las actividades del flujo Gestión de Proyectos se tienen claros los siguientes puntos: las responsabilidades de cada miembro y del equipo, las metas que se deben alcanzar, las dificultades que se pueden presentar, la mejor estrategia que se debe cumplir, el procedimiento en caso de algún cambio para tener control sobre el sistema y el plan que se debe seguir para alcanzar las metas y desarrollar el sistema.
- TSP está enfocado a la gestión de proyectos y al desarrollo del software y no abarca actividades para los flujos: Modelado del negocio, Análisis y Distribución del software que si contiene RUP.
- Para integrar TSP y RUP se consideró como base los flujos que posee RUP a los que se añaden las actividades de los procesos de TSP. Es así, que el diseño detallado de la implementación de TSP se definió en el flujo de Diseño. Porque el objetivo del diseño detallado es justamente el objetivo del diseño de RUP.
- Después de las pruebas se integró el flujo Postmortem para proveer de una estructura de aprendizaje e identificar las oportunidades de mejoramiento para los siguientes ciclos de desarrollo o para otros proyectos.
- Para el desarrollo del caso de estudio el equipo se conformó de dos personas, el numero reducido de personas hizo que la creación de los productos (SRS,

SDS, modelos, diagramas, plan de pruebas) sea sencilla, pero el registro de tiempos, tareas, calendarios, defectos, planes, inspecciones (formas TASK, SCHEDULE, LOGT, LOGD, SUMS, ITL) sea complejo en un inicio debido a la inexperiencia del equipo.

- Al finalizar el desarrollo del sistema se puede apreciar que la calidad del producto es muy buena porque no existen defectos, se cumplen con los requerimientos y el tiempo de desarrollo cumple con el plan definido en un inicio.

3.2. RECOMENDACIONES

- Se sugiere conformar un equipo de trabajo con un número mayor de miembros lo cual permitirá apreciar de mejor manera la utilidad de TSP en el desarrollo de un proyecto, puesto que se delegan menos responsabilidades por persona lo que mejorará el desempeño de los roles de las personas.
- Si se requiere desarrollar un sistema en corto tiempo, se recomienda que al definir los tiempos personales se dedique mayor tiempo diario a la creación del proyecto y se asegure que el tiempo perdido por situaciones ajenas se recupere lo más pronto posible, para no modificar los planes semanales siguientes y garantizar el término del proyecto en la fecha prevista.
- Se recomienda el uso de la combinación de TSP y RUP para el desarrollo de un proyecto porque le permite al equipo de trabajo mantener un control sobre el desempeño de cada miembro a través de los reportes y del cumplimiento de tareas. El equipo podrá apreciar quienes cumplen con las expectativas y pueden continuar o quienes no satisfacen las necesidades y deben ser reemplazados, así como también permite identificar cuales tareas requieren apoyo de personal y cuales no.
- En un proyecto que considere seguir los flujos de TSP y RUP se sugiere que para obtener una buena experiencia en el desarrollo se revise la teoría detallada en este proyecto para tener una base que le ayude y facilite el trabajo.
- Para medir la calidad de un proyecto se sugiere seguir las consideraciones de TSP, ya que propone obtener datos reales y exactos que permiten medir la calidad, para no hacer simples suposiciones y tener fundamentos que respalden la calidad.

BIBLIOGRAFÍA

- [1] WATTS S, Humphrey. Introduction to the Team Software Process, Ed. Addison Wesley Longman, Canada, 1999.
- [2] CARNEGIE MELLON SOFTWARE ENGINEERING INSTITUTE. The Team Software Process (TSP) and the Personal Software Process (PSP). <http://www.sei.cmu.edu/tsp/>, Pittsburgh, Acceso Ultimo: 2 de Diciembre del 2005.
- [3] NOOPUR, Davis; MULLANEY, Julia. The Team Software ProcessSM (TSPSM) in Practice: A Summary of Recent Results. http://www.quarksoft.net/webQS/html/dinamicos/TSP_Results_2003.pdf, Carnegie Mellon Software Engineering Institute, Pittsburgh, 2003, Acceso Ultimo 2 de diciembre del 2005.
- [4] MONTES DE OCA, César. Team Software Process (TSP) Integración de Equipos de Desarrollo de Alto Rendimiento. http://www.emagister.com/public/pdf/comunidad_emagister/68706010050357686756485568684550-TSP_Conferencia.pdf, Pittsburgh, Acceso Ultimo: 2 de Diciembre del 2005.
- [5] WATTS, Humphrey. The Team Software Process. <http://www.sstc-online.org/cd-rom/1999/slides/ATeamSof.pdf>, Pittsburgh, Acceso Ultimo: 2 diciembre del 2005.
- [6] HERNÁNDEZ, Donato; GUTIERREZ, Heriberto; CANEDO, Gerardo. Creación de Equipos de Alto Desempeño usando “Team Software Process” (TSP) y “Personal Software Process” (PSP). <http://iiisci.org/Journal/riSCI/pdfs/P878644.pdf>, México, Acceso Ultimo: 2 de diciembre del 2005.
- [7] WEBB, David; WATTS, Humphrey. Using the TSP on the TaskView Project. <http://www.stsc.hill.af.mil/crosstalk/1999/02/webb.asp>, Acceso Ultimo: 2 de diciembre del 2005.
- [8] OVER, James. Team Software Process for Secure Systems Development. <http://www.sei.cmu.edu/tsp/tsp-secure-presentation/sld001.htm>, Acceso Ultimo 2 de diciembre del 2005.
- [9] ZAVALA R, Ingeniería de Software, Los Fundamentos, <http://www.willydev.net/descargas/Articulos/General/ingsoftware.aspx>, Acceso Ultimo: 2 de Diciembre del 2005.

- [10] Morea, Lucas. Calidad Total
<http://www.monografias.com/trabajos14/calidadtotal/calidadtotal.shtml>, Acceso Ultimo: 2 de Diciembre del 2005.
- [11] MORENO, José. Aplicación de un Sistema Experto para el desarrollo de Sistema Evaluador del modelo Capability Maturity Model (CMM) niveles dos y tres,
http://140.148.3.250/u_dl_a/servlet/mx.udlap.ict.tales.html.Block?Thesis=1565&Type=T, Puebla, Acceso Ultimo: 25 de Enero del 2006.
- [12] SORIANO, Fernando. Proceso Unificado Parte I. Universidad de Fasta.
<http://pub.ufasta.edu.ar/fim28/files2005/FIM28%20UP%20I.pdf>. 2 de diciembre del 2005.
- [13] JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. El Proceso Unificado de Desarrollo de Software. Primera edición, Addison Wesley Longman, 2000.
- [14] HORMIGA, Jairo. Proceso Unificado para Desarrollo de Software (RUP).
<http://atenea.ucauca.edu.co/~gramirez/archivos/AnotacionesRUP.pdf>. Acceso Ultimo: 6 de Marzo del 2006.
- [15] GUERRERO, Luis. Rational Unified Process,
<http://www.dcc.uchile.cl/~luguerre/cc61j/clase1.html>, Chile, Acceso Ultimo: 6 de Marzo del 2006.
- [16] RUIZ, Erika. Aplicación de un Modelo Sistémico para la Evaluación de la Calidad de Software. Quito, 2004.
- [17] URBINA, Silvia; VILLAGRAN, Mónica. Diseño e Implementación de una interfaz dinámica en un ambiente Web para dispositivos móviles y PCS. Quito, 2004.

GLOSARIO

Actividad: Unidad tangible de trabajo realizada por un trabajador o trabajadores en un flujo de trabajo.

Análisis (flujo de TSP y RUP): Flujo fundamental que tiene como objetivo principal analizar los requisitos descritos en la captura de requisitos, para comprender y describir éstos de manera que sea fácil de mantener y estructurar el sistema.

Artefacto: Término general para cualquier producto de trabajo (código, gráficos Web, esquema de base de datos, documentos de texto, diagramas, modelos, etc).

Calidad: Es satisfacer plenamente las necesidades y expectativas del cliente, lograr productos y servicios con cero defectos; es diseñar, producir y entregar un producto de satisfacción total, de acuerdo con las normas establecidas y en el menor tiempo.

Casos de uso: Es un fragmento de la funcionalidad que proporciona al usuario final un resultado. Los casos de uso representan los requisitos funcionales y guían el proceso de desarrollo.

Caso de prueba: Especificación de un caso para probar el sistema, que incluye qué probar, las entradas y los resultados bajo que condiciones.

Configuration control board (CCB): Asegura que todos los productos base y los cambios en estos son propiamente justificados y convenientes para la calidad.

Defecto: Es un elemento de los requerimientos, diseño o implementación que si no es cambiado puede causar irregularidades en el diseño, la implementación, las pruebas, el uso o el mantenimiento.

Diseño (flujo de TSP y RUP): Flujo de trabajo fundamental que tiene como objetivo principal formular modelos basados en los requisitos no funcionales y en las restricciones existentes, que permite estar preparados para la implementación y las pruebas.

Diseño de alto nivel (SDS): Es el diseño del producto de forma general. Describe la estructura general del producto, los componentes, las funciones de los componentes y las especificaciones externas de los componentes.

Diseño detallado: Es el diseño a un nivel específico. Describe el diseño de la arquitectura, de los casos de uso, de las clases, de los subsistemas y de las pantallas.

Distribución (flujo de TSP y RUP): Flujo fundamental que tiene como objetivo principal distribuir el producto a los usuarios finales. Este incluye la producción de un release o comunicado, distribuir el software, instalar el software, apoyar a los usuarios, realizar pruebas beta, realizar migración de datos y terminar con la aceptación formal del usuario.

Equipo de trabajo: Es un grupo de por lo menos dos personas quienes trabajan bajo metas y objetivos comunes, donde a cada persona se le asigna roles y funciones para el desarrollo y donde el cumplimiento de la misión requiere de dependencia entre los miembros del grupo.

Especificación de requerimientos de software (SRS): Describe un plan para desarrollar un producto y lo que se quiere que ejecute el producto. Los principales tópicos del SRS a son: requerimientos funcionales, requerimientos de las interfaces externas, limitaciones del diseño, atributos y otros requerimientos.

Estrategia: Tiene como objetivo minimizar los riesgos para no exceder el tiempo de desarrollo disponible para el proyecto. La estrategia de la combinación de TSP y RUP consiste en desarrollar el producto en procesos cíclicos.

Flujos de Trabajo: Conjunto de actividades ordenadas.

Generar Información: Módulo del sistema cuya función es especificar los clientes para la gestión de un vendedor.

Gestión de proyectos (flujo de TSP y RUP): Flujo fundamental que tiene como objetivo definir la organización del proyecto, las metas, los riesgos, la estrategia, el procedimiento para cambios y el plan para el desarrollo del sistema.

Gestión del vendedor: Módulo del sistema cuya función es permitir el registro de pedidos y MNG del vendedor.

Implementación (flujo de TSP y RUP): Flujo fundamental que tiene como objetivo codificar el software en componentes: código fuente, ficheros, ejecutables, bases de datos, etc.

MNG (Motivo de no Gestión): En la gestión del vendedor se denomina MNG a las razones por las cuales no se registra un pedido para un cliente.

Modelo: Captura la vista de un sistema del mundo real. Es la abstracción de un sistema para un determinado propósito. Describe los aspectos del sistema que son relevantes a un apropiado nivel de detalle.

Modelo de casos de uso: Es la funcionalidad completa del sistema. Esta constituido por todos los casos de uso.

Modelado del negocio (flujo de TSP y RUP): Flujo fundamental cuya finalidad es describir cada proceso del negocio del cliente, especificando sus datos, actividades, agentes y reglas del negocio.

Monitorear Rutas: Módulo del sistema cuya función es presentar reportes que permiten al administrador apreciar el trabajo realizado por el vendedor.

Plan de administración de la configuración: Especifica el procedimiento que se debe seguir para la realización de cambios del software.

Plan del proyecto: Proporciona un contexto para hacer el trabajo y cumplir a tiempo con los compromisos adquiridos.

Plan para la implementación: Proporciona un contexto para la implementación, define los estándares de implementación y el plan de pruebas de unidad.

Pocket PC: Es un dispositivo móvil de uso personal.

Postmortem (flujo de TSP y RUP): Flujo fundamental cuya finalidad es proveer de una estructura de aprendizaje que permita identificar oportunidades de mejoramiento para los siguientes ciclos de desarrollo o para otros proyectos.

Pruebas (flujo de TSP y RUP): Flujo fundamental cuya finalidad es comprobar el resultado de la implementación mediante pruebas de integración y del sistema.

Pruebas Beta: Son las pruebas que se realizan bajo condiciones reales al software cuando es distribuido y permiten apreciar realmente las falencias que tiene el software.

Pruebas de unidad: Son las pruebas que se realizan al software durante la implementación para descartar errores en el código.

Pruebas del sistema: El propósito de las pruebas del sistema es asegurar que el sistema hace lo que los usuarios supusieron que haría. Se denominan pruebas de caja negra.

Pruebas de integración: Se denominan pruebas de caja blanca. Tienen como finalidad verificar que los componentes interaccionan entre sí de la forma apropiada después de haber sido integrados en una construcción

Recibir información: Módulo del sistema cuya función es actualizar los datos de la gestión del vendedor en la base de datos de la PC.

Requerimientos (flujo de TSP y RUP): Flujo fundamental en el que se describen los requisitos del software de forma clara y no ambigua.

Riesgo: Variable de un proyecto que pone en peligro o impide el desarrollo del mismo, puede generar retrasos, desviación de costos o cancelación del proyecto.

Rol: Es un cargo que contiene definidas responsabilidades para la persona a la que se le asigna el rol.

Ruta: Es una secuencia de puntos, en donde los puntos son los clientes que el vendedor visitó en una fecha y la secuencia esta definida por la hora de la visita.

Sincronizar Información: Módulo del sistema cuya función es actualizar la información almacenada en las bases de datos de la PC y de la Pocket PC.

Trabajador: Representa los comportamientos, descripciones y responsabilidades del mismo. RUP propone un modelo para la organización de personal en el que se definen los cargos y sus tareas correspondientes.

ANEXOS

1.1. FORMA INFO INDIVIDUAL

Name	Geovanna Bustos	Instructor	Ing. Carlos Montenegro				
Date	11-04-2006	Number of College Credits	11				
Major	Desarrollo sist.	Expected Graduation Date	10-12-2006				
Briefly describe your relevant experience and interests:							
Desarrollo de sistemas pequeños en el transcurso de la carrera.							
Desarrollo de un sistema mediano de contabilidad.							
Desarrollar mi proyecto de titulación en el menor tiempo posible.							
Briefly describe your work on other team projects:							
He desempeñado el rol de líder de algunos proyectos.							
Dedico mi esfuerzo y tiempo necesario al cumplimiento del proyecto en el que este involucrada.							
Briefly describe any leadership or management positions you have held (at work or in clubs/organizations):							
Líder de proyectos pequeños en el transcurso de la carrera.							
Líder del desarrollo del sistema de contabilidad.							
State your team preferences, if any:							
Apoyar como guía del desarrollo y desarrollador.							
List your class schedule and other times when you have scheduled activities such as work, ROTC, clubs, sports teams, etc.							
Time	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.
8:00– 9:00	Clases	Clases	Clases	Clases	Clases	Familia	Familia
9:15–10:15	Clases	Clases	Clases	Clases	Clases	Familia	Familia
10:30–11:30							
11:45–12:45							
13:00–14:00							
14:15–15:15	Familia	Familia	Familia	Familia	Familia	Familia	Familia
15:30–16:30	Familia	Familia	Familia	Familia	Familia	Familia	Familia
16:45–17:45							
18:00-19:00							
19:15-20:30	Familia	Familia	Familia	Familia	Familia		
20:45-21:45	Familia	Familia	Familia	Familia	Familia	Familia	Familia
22:00-23:00							
Rank from 1 (least) to 5 (most) your preferences for serving in the following team roles:							
Team Leader	1	2	3	4	5		
Development Manager	1	2	3	4	5		
Planning Manager	1	2	3	4	5		
Quality/Process Manager	1	2	3	4	5		
Support Manager	1	2	3	4	5		

1.2. FORM TASK INDIVIDUAL

Name Geovanna Bustos Date 15-04-2006
 Team DPT Instructor Ing. Carlos Montenegro
 Part/Level Gestión de Proyectos Cycle 1

Task			Plan Hours					Plan Size/Value					Actual		
Phase	Part	Task Name	# Engineers	Rol miembro 1	Rol miembro 2	Total Team Hours	Cumulative Hours	Size Units	Size	Week No.	Planned Value	Cumulative PV	Hours	Cumulative Hours	Week No.
GP	S	Conformar el equipo y asignar roles a cada uno de los integrantes.	1	1			1			1	0,205	0,205	1	1	1
GP	S	Definir las metas	1	2			3	Pág.	2	1	0,411	0,616	2	3	1
GP	S	Coordinar el trabajo.	1	1			4	Pág.	4	1	0,205	0,821	2	5	1
GP	S	Crear el diseño conceptual del proyecto.	1	1			5	Pág.	2	1	0,205	1,027	2	7	1
GP	S	Establecer riesgos potenciales.	1	2			7	Pág.	1	2	0,411	1,437			
GP	S	Crear la estrategia	1	2			9	Pág.	2	2	0,411	1,848			
GP	S	Configurar el software	1	1			10	Pág.	1	2	0,205	2,053			
GP	S	Crear del plan del proyecto	0	0			10				0,000	2,053			
MN	S	Identificar los casos de negocio y sus actores.	1	2			12	Pág.	3	3	0,411	2,464			
MN	S	Desarrollar un modelo de objetos de un negocio.	1	2			14	Pág.	1	3	0,411	2,875			
Rq	S	Obtención de los requerimientos.	1	2			16	Pág.	1	3	0,411	3,285			
Rq	S	Especificación de los requerimientos de software SRS.	1	4			20	Pág.	10	3	0,821	4,107			
Rq	S	Crear el plan de prueba del sistema	1	1			21	Pág.	1	3	0,205	4,312			
Rq	S	Revisar e inspeccionar los requerimientos	1	1			22	Pág.	2	3	0,205	4,517			
Rq	S	Modificar los requerimientos	1	1			23	Pág.	3	3	0,205	4,723			
An	S	Análisis de la arquitectura	1	1			24	Pág.	2	4	0,205	4,928			
An	S	Análisis de los casos de uso	1	6			30	Pág.	15	4	1,232	6,160			
An	S	Análisis de las clases	1	2			32	Pág.	3	4	0,411	6,571			
An	S	Análisis de los paquetes	1	1			33	Pág.	1	4	0,205	6,776			
Di	S	Diseño de alto nivel.	1	2			35	Pág.	3	5	0,411	7,187			
Di	P	Diseño detallado	1	6			41	Pág.	15	5	1,232	8,419			
Di	S	Crear un plan de pruebas de integración.	1	1			42	Pág.	2	5	0,205	8,624			

Di	S	Revisar e inspeccionar el diseño.	1	1		43	Pág	2	5	0,205	8,830			
Di	P	Modificar el diseño.	1	1		44	Pág	20	5	0,205	9,035			
Im	S	Crear un plan para la implementación	1	1		45	Pág	3	6	0,205	9,240			
Im	P	Implementación de la arquitectura.	1	26		71	LOC	200	6-7	5,339	14,579			
Im	P	Implementación de los subsistemas.	1	20		91	LOC	200	8	4,107	18,686			
Im	P	Implementación de las clases y objetos.	1	300		391	LOC	8.000	9-28	61,602	80,287			
Im	S	Revisar e inspeccionar la implementación	1	30		421	Pág	2	29	6,160	86,448			
Im	P	Modificar la implementación	1	20		441	LOC	8.400	30	4,107	90,554			
Im	P	Realizar pruebas de unidad.	1	10		451	Pág	2	31	2,053	92,608			
Im	S	Revisar la calidad de los componentes.	1	2		453	Pág	2	31	0,411	93,018			
Im	S	Integrar los componentes en un sistema ejecutable.	1	8		461	LOC	8.400	32	1,643	94,661			
PI	S	Planificar las pruebas.	1	1		462	Pág	1	33	0,205	94,867			
PI	S	Diseñar las pruebas.	1	2		464	Pág	1	33	0,411	95,277			
PI	S	Realizar las pruebas.	1	4		468			33	0,821	96,099			
PI	S	Evaluar las pruebas.	1	2		470			33	0,411	96,509			
PI	S	Registrar las pruebas.	1	2		472	Pág	4	33	0,411	96,920			
Pos	S	Revisión de los datos.	1	4		476			34	0,821	97,741			
Pos	S	Evaluaciones del desempeño de los roles.	1	3		479			34	0,616	98,357			
Pos	S	Preparar los reportes.	1	6		485			34	1,232	99,589			
Pos	S	Producción del reporte.	1	2		487	Pág	5	34	0,411	100,00			
		TOTAL		487						100,00	100,00			

1.3. FORM SCHEDULE INDIVIDUAL

Name	Geovanna Bustos	Date	15-04-2006
Team	DPT	Instructor	Ing. Montenegro
Part/Level	Gestión de Proyectos	Cycle	1

Week No.	Date	Plan			Actual			
		Direct Hours	Cumulative Hours	Cumulative Planned Value	Team Hours	Cumulative Hours	Week Earned Value	Cumulative Earned Value
1	15-04-06	5	5	1,027	7	7	1,437	1,437
2	22-04-06	5	10	2,053				
3	29-04-06	13	23	4,723				
4	06-05-06	10	33	6,776				
5	13-05-06	11	44	9,035				
6	20-05-06	14	58	11,910				
7	27-05-06	13	71	14,579				
8	03-06-06	20	91	18,686				
9-28	21-10-06	300	391	80,287				
29	28-10-06	30	421	86,448				
30	04-11-06	20	441	90,554				
31	11-11-06	12	453	93,018				
32	18-11-06	8	461	94,661				
33	25-11-06	11	472	96,920				
34	02-12-06	15	487	100,00				

1.4. FORM WEEK INDIVIDUAL

Name	Geovanna Bustos	Team	DPT	Instructor	Ing. Montenegro
Date	15-04-2006	Cycle No.	1	Week No.	1
Weekly Data				Planned	Actual
Project hours for this week				5	7
Project hours this cycle to date				5	7
Earned value for this week				1,027	1,437
Earned value this cycle to date				1,027	1,437
Total hours for the tasks completed this phase to date				5	7
Team Member Weekly Data				Hours Planned	Hours Actual
Team Leader					
Planning Manager					
Quality/Process Manager				5	7
Development Manager					
Support Manager					
Totals					

Development Tasks Completed	Hours Planned	Hours Actual	Earned Value	Planned Week
Conformar el equipo y asignar roles a cada uno de los integrantes.	1	1	0,205	1
Definir las metas	2	2	0,411	1
Coordinar el trabajo.	1	2	0,411	1
Crear el diseño conceptual del proyecto.	1	2	0,411	1
Issue/Risk Tracking				
Issue/Risk Name	Status			
Grupo de trabajo inadecuado	Baja			
Metas agresivas e inadecuadas	Baja			
Falta de coordinación del equipo.	Baja			
Other Significant Items				

1.5. FORM WEEK DEL EQUIPO

Name	Equipo	Team	DPT	Instructor	Ing. Montenegro
Date	15-04-2006	Cycle No.	1	Week No.	1
Weekly Data				Planned	Actual
Project hours for this week				9	13
Project hours this cycle to date				9	13
Earned value for this week				0,929	1,342
Earned value this cycle to date				0,929	1,342
Total hours for the tasks completed this phase to date				9	13
Team Member Weekly Data		Hours Planned	Hours Actual	Planned Value	Earned Value
Team Leader					
Planning Manager					
Quality/Process Manager		5	7	1,027	1,437
Development Manager					
Support Manager		4	6	0,830	1,245
Totals		9	13	0,929	1,342
Development Tasks Completed		Hours Planned	Hours Actual	Earned Value	Planned Week
Conformar el equipo y asignar roles a cada uno de los integrantes.		1	1	0,103	1
Definir las metas		4	4	0,413	1
Coordinar el trabajo.		2	4	0,413	1
Crear el diseño conceptual del proyecto.		2	4	0,413	1

2	Monitorear Rutas					
2.1	Reporte de ruta seguida por el vendedor.	X			10	
2.2	Reporte de clientes visitados por el vendedor con pedido.	X			10	
2.3	Reporte de clientes visitados por el vendedor con motivo de no gestión	X			10	
2.4	Reporte de clientes no visitados por el vendedor.	X			10	
Total					692	

1.8. FORM SUMS

Plan X Assembly _____ Actual _____

Name: Administrador de la Planificación G. B. **Date:** 20-04-2006
Team: DPT **Instructor:** Ing. Montenegro
Part/Level: Gestión de Proyectos **Cycle:** 1

Product or Part Names and/or Numbers	Size Measure	Base	Deleted	Modified	Added	Reused	and		Total New Reuse
							New Changed	Total	
SRS	Pág. Texto	10							10
Diseño de Alto Nivel HLD	Pág. HLD ¹	3							3
Administrar Rutas	Pág. DLD ²	10							10
	LOC	7500							7500
Monitorear Rutas	Líneas DLD	5							5
	LOC	900							900
Totals	Pág. Texto	10							10
	Pág. HLD	3							3

¹ HLD Diseño de alto nivel

² DLD Diseño de nivel detallado

1.9. FORM TASK DEL EQUIPO

Name	Equipo	Date	20-04-2006
Team	DPT	Instructor	Ing. Carlos Montenegro
Part/Level	Gestión de Proyectos	Cycle	1

Task			Plan Hours					Plan Size/Value					Actual		
Phase	Part	Task Name	# Engineers	Rol miembro 1	Rol miembro 2	Total Team Hours	Cumulative Hours	Size Units	Size	Week No.	Planned Value	Cumulative PV	Hours	Cumulative Hours	Week No.
GP	S	Conformar el equipo y asignar roles a cada uno de los integrantes.	1	1	0	1	1			1	0,103	0,103	1	1	1
GP	S	Definir las metas	2	2	2	4	5	Pág.	2	1	0,413	0,516	4	5	1
GP	S	Coordinar el trabajo.	2	1	1	2	7	Pág.	4	1	0,206	0,722	4	9	1
GP	S	Crear el diseño conceptual del proyecto.	2	1	1	2	9	Pág.	2	1	0,206	0,929	4	13	1
GP	S	Establecer riesgos potenciales.	2	2	2	4	13	Pág.	1	2	0,413	1,342			
GP	S	Crear la estrategia	2	2	2	4	17	Pág.	2	2	0,413	1,754			
GP	S	Configurar el software	2	1	1	2	19	Pág.	1	2	0,206	1,961			
GP	S	Crear del plan	1	0	4	4	23	Pág.	1	2	0,413	2,374			
MN	S	Identificar los casos de negocio y sus actores.	1	2	0	2	25	Pág.	3	3	0,206	2,580			
MN	S	Desarrollar un modelo de objetos de un negocio.	1	2	0	2	27	Pág.	1	3	0,206	2,786			
Rq	S	Obtención de los requerimientos.	1	2	0	2	29	Pág.	1	3	0,206	2,993			
Rq	S	Especificación de los requerimientos de software SRS.	2	4	4	8	37	Pág.	10	3	0,826	3,818			
Rq	S	Crear el plan de prueba del sistema	2	1	1	2	39	Pág.	1	3	0,206	4,025			
Rq	S	Revisar e inspeccionar los requerimientos	2	1	1	2	41	Pág.	2	3	0,206	4,231			
Rq	S	Modificar los requerimientos	2	1	1	2	43	Pág.	3	3	0,206	4,438			
An	S	Análisis de la arquitectura	2	1	1	2	45	Pág.	2	4	0,206	4,644			
An	S	Análisis de los casos de uso	2	6	6	12	57	Pág.	15	4	1,238	5,882			
An	S	Análisis de las clases	2	2	2	4	61	Pág.	3	4	0,413	6,295			
An	S	Análisis de los paquetes	2	1	1	2	63	Pág.	1	4	0,206	6,502			
Di	S	Diseño de alto nivel.	2	2	2	4	67	Pág.	3	5	0,413	6,914			
Di	P	Diseño detallado	2	6	6	12	79	Pág.	15	5	1,238	8,153			

Di	S	Crear un plan de pruebas de integración.	2	1	1	2	81	Pág.	2	5	0,206	8,359				
Di	S	Revisar e inspeccionar el diseño.	2	1	1	2	83	Pág	2	5	0,206	8,566				
Di	P	Modificar el diseño.	2	1	1	2	85	Pág	20	5	0,206	8,772				
Im	S	Crear un plan para la implementación	2	1	1	2	87	Pág	3	6	0,206	8,978				
Im	P	Implementación de la arquitectura.	2	26	26	52	139	LOC	200	6-7	5,366	14,345				
Im	P	Implementación de los subsistemas.	2	20	20	40	179	LOC	200	8	4,128	18,473				
Im	P	Implementación de las clases y objetos.	2	300	300	600	779	LOC	8.000	9-28	61,920	80,392				
Im	S	Revisar e inspeccionar la implementación	2	30	30	60	839	Pág	2	29	6,192	86,584				
Im	P	Modificar la implementación	2	20	20	40	879	LOC	8.400	30	4,128	90,712				
Im	P	Realizar pruebas de unidad.	2	10	10	20	899	Pág	2	31	2,064	92,776				
Im	S	Revisar la calidad de los componentes.	1	2	0	2	901	Pág	2	31	0,206	92,982				
Im	S	Integrar las componentes en un sistema ejecutable.	2	8	8	16	917	LOC	8.400	32	1,651	94,634				
Pl	S	Planificar las pruebas.	2	1	1	2	919	Pág	1	33	0,206	94,840				
Pl	S	Diseñar las pruebas.	2	2	2	4	923	Pág	1	33	0,413	95,253				
Pl	S	Realizar las pruebas.	2	4	4	8	931			33	0,826	96,078				
Pl	S	Evaluar las pruebas.	2	2	2	4	935			33	0,413	96,491				
Pl	S	Registrar las pruebas.	2	2	2	4	939	Pág	4	33	0,413	96,904				
Pos	S	Revisión de los datos.	2	4	4	8	947			34	0,826	97,730				
Pos	S	Evaluaciones del desempeño de los roles	2	3	3	6	953			34	0,619	98,349				
Pos	S	Preparar los reportes	2	6	6	12	965			34	1,238	99,587				
Pos	S	Producción del reporte.	2	2	2	4	969	Pág	5	34	0,413	100,00				
		TOTAL		487	482	969	969					100,00	100,00			

1.10. FORM SCHEDULE DEL EQUIPO

Name	Equipo	Date	20-04-2006
Team	DPT	Instructor	Ing. Montenegro
Part/Level	Gestión de Proyectos	Cycle	1

Week No.	Date	Plan			Actual			
		Direct Hours	Cumulative Hours	Cumulative Planned Value	Team Hours	Cumulative Hours	Week Earned Value	Cumulative Earned Value
1	20-04-06	9	9	0,929	13	13	1,342	1,342
2	22-04-06	14	23	2,374				
3	29-04-06	20	43	4,438				
4	06-05-06	20	63	6,502				
5	13-05-06	22	85	8,772				
6	20-05-06	28	113	11,661				
7	27-05-06	26	139	14,345				
8	03-06-06	40	179	18,473				
9-28	21-10-06	600	779	80,392				
29	28-10-06	60	839	86,584				
30	04-11-06	40	879	90,712				
31	11-11-06	22	901	92,982				
32	18-11-06	16	917	94,634				
33	25-11-06	22	939	96,904				
34	02-12-06	30	969	100,00				

1.11. FORM SUMP

Name	Equipo	Date	20-04-06
Team	DPT	Instructor	Ing. Montenegro
Part/Level	Gestión de Proyectos	Cycle	1

Product Size	Plan	Actual
Requirements pages (SRS)	10	
Other text pages	20	
High-level design pages (SDS)	3	
Detailed design pages	15	
Base LOC (B) (measured)		
Deleted LOC (D)		
	(Estimated)	(Counted)
Modified LOC (M)		
	(Estimated)	(Counted)
Added LOC (A)		
	(N-M) 8400	(T-B+D-R)
Reused LOC (R)		
	(Estimated)	(Counted)
Total New and Changed LOC (N)	8400	
	(Estimated)	(A+M)
Total LOC (T)	8400	
	(N+B-M-D+R)	(Measured)

Total New Reuse LOC
 Estimated Object LOC (E)
 Upper Prediction Interval (70%)
 Lower Prediction Interval (70%)

Time in Phase (hours)

	Plan	Actual	Actual %
Gestión de Proyectos	23	13	100%
Modelo del Negocio	4		
Requerimientos	10		
Plan de prueba del sistema	2		
Revisar e inspeccionar los requerimientos	2		
Modificar los requerimientos	2		
Análisis	20		
Diseño de alto nivel	4		
Diseño detallado	12		
Plan de pruebas de integración	2		
Revisar e inspeccionar el diseño de alto nivel	1		
Revisar e inspeccionar el diseño detallado	1		
Modificar el diseño	2		
Plan para la implementación	2		
Implementación	692		
Revisar e inspeccionar la implementación	60		
Modificar la implementación	40		
Realizar pruebas de unidad	20		
Revisar la calidad de los componentes	2		
Integrar las componentes en un sistema ejecutable	16		
Planificar las pruebas	2		
Diseñar las pruebas	4		
Realizar las pruebas	8		
Evaluar las pruebas	4		
Registrar las pruebas	4		
Postmortem	30		
Distribución	0		
Total	969	13	100%

Total Time UPI (70%)

Total Time LPI (70%)

Defects Injected

Gestión de Proyectos			
Modelo del Negocio			
Requerimientos	2		
Plan de prueba del sistema			
Revisar e inspeccionar los requerimientos			
Modificar los requerimientos			
Análisis			
Diseño de alto nivel	3		
Diseño detallado	5		
Plan de pruebas de unidad			
Revisar e inspeccionar el diseño de alto nivel			
Revisar e inspeccionar el diseño detallado			
Modificar el diseño			
Plan para la implementación			
Implementación	25		

Revisar e inspeccionar la implementación			
Modificar la implementación			
Realizar pruebas de unidad			
Revisar la calidad de los componentes			
Integrar las componentes en un sistema ejecutable			
Planificar las pruebas			
Diseñar las pruebas			
Realizar las pruebas			
Evaluar las pruebas			
Registrar las pruebas			
Postmortem			
Distribución			
Total Development	35		
Defects Removed			
Gestión de Proyectos			
Modelo del Negocio			
Requerimientos			
Plan de prueba del sistema			
Revisar e inspeccionar los requerimientos	2		
Modificar los requerimientos			
Análisis			
Diseño de alto nivel			
Diseño detallado			
Plan de pruebas de unidad			
Revisar e inspeccionar el diseño de alto nivel	4		
Revisar e inspeccionar el diseño detallado	4		
Modificar el diseño			
Plan para la implementación			
Implementación	20		
Revisar e inspeccionar la implementación	2		
Modificar la implementación			
Realizar pruebas de unidad	1		
Revisar la calidad de los componentes			
Integrar las componentes en un sistema ejecutable	1		
Planificar las pruebas			
Diseñar las pruebas			
Realizar las pruebas	1		
Evaluar las pruebas			
Registrar las pruebas			
Postmortem			
Distribución			
Total Development	35		

1.12. FORM SUMQ

Name	Equipo	Date	20-04-06
Team	DPT	Instructor	Ing. Montenegro
Part/Level	Gestión de Proyectos	Cycle	1

Summary Rates	Plan	Actual
LOC/hour	8400/969=8.67	
% Reuse (% of total LOC)	0	
% New Reuse (% of N&C LOC)	0	
Percent Defect-Free (PDF)		
In compile	$((8-2)/8)= 75$	
In unit test	$(8-1)/8= 88$	
In build and integration	$(8-0.5)=94$	
In system test	$(8-0.1)=99$	
Defect/page		
Requirements inspection	$2/10=0.2$	
HLD inspection	$3/3=1$	
DLD inspection	$2/15=0.13$	
Defects/KLOC		
DLD review	$3/8.4=0.36$	
DLD inspection	$2/8.4=0.24$	
Code review	$1/8.4=0.12$	
Compile	$20/8.4=2.38$	
Code inspection	$1/8.4=0.12$	
Unit test	$1/8.4=0.12$	
Build and integration	$1/8.4=0.12$	
System test	$1/8.4=0.12$	
Total development		
Defect Ratios		
Code review/Compile	$0.12/2.38= 0.05$	
DLD review/Unit test	$0.36/0.12= 3$	
Development time ratios (%)		
Requirements inspection/Requirements	$2/10=20.00\%$	
HLD inspection/HLD	$1/4=25.00\%$	
DLD/code	$12/692=1.73\%$	
DLD review/DLD	$1/12=8.33\%$	
Code review/code	$60/692=8.67\%$	
A/FR		
Review rates		
DLD page/hour	$15/0.5=30$	
Code LOC/hour	$8400/30=280$	
Inspection rates		
Requirement pages/hour	$10/2=5$	
HLD pages/hour	$3/0.5=6$	
DLD page /hour	$15/0.5=30$	
Code LOC/hour	$8400/30=280$	
Defect-injection Rates (Defects/Hr.)		
Requirements	$2/10=0.20$	
HLD	$\frac{3}{4}=0.75$	

DLD	$5/12=0.42$	
Code	$25/692=0.04$	
Compile	0	
Unit test	0	
Build and integration	0	
System test	0	

Defect-removal Rates (Defects/Hr.)

Requirements inspection	$2/2=1$	
HLD inspection	$3/1=3$	
DLD review	$3/0.5=6$	
DLD inspection	$2/0.5=4$	
Code review	$1/30=0.03$	
Compile	$20/10=2$	
Code inspection	$1/30=0.03$	
Unit test	$1/20=0.05$	
Build and integration	$1/16=0.06$	
System test	$1/8=0.13$	

Phase Yields

Requirements inspection	$2/2=100$	
HLD inspection	$3/(5-2)=100$	
DLD review	$3/(8-5)=100$	
Test development		
DLD inspection	$2/(10-8)=100$	
Code review	$1/(35-10)=4$	
Compile	$20/(35-11)=83.33$	
Code inspection	$1/(35-31)=25.00$	
Unit test	$1/(35-32)=33.33$	
Build and integration	$1/(35-34)=100$	
System test		

Process Yields

% before compile	$10/10=100$	
% before unit test	$32/35=91.43$	
% before build and integration	$33/35=94.29$	
% before system test	$34/35=97.14$	
% before system delivery	$35/35=100.00$	

1.13. FORM INS (REQUERIMIENTOS)

Name	Administrador Calidad/ Procesos G.B.	Date	28-04-2006
Team	DPT	Instructor	Ing: Montenegro
Part/Level	Requerimientos	Cycle	1
Moderator	C. G.	Owner	Politécnica Nacional

Engineer Data

Name	Defects		Preparation Data			Est.
	Major	Minor	Size	Time	Rate	Yield
Geovanna Bustos		1def	12 pág	0.5 h	24pág/h	100
Totals:						

Defect Data

No.	Defect Description	Defects		Engineers (finding major defects)					
		Maj	Min					A	B
1.10	G.B. Ortografía		X						
Totals			1						
Unique Defects									

Inspection Summary	-	Product Size:	_____	-	Size Measure:	_____	-
Total defects for A:	_____	-	Total defects for B:	_____	-	C (# common):	_____
Total Defects (AB/C):	_____	-	Number Found (A+B-C):	_____	-	Number Left:	_____
Meeting Time:	_____	-	Total Inspection Hours:	_____	-	Overall Rate:	_____

1.14. FORM INS (DISEÑO)

Name	Administrador Calidad/ Procesos G.B.	Date	11-05-2006
Team	DPT	Instructor	Ing: Montenegro
Part/Level	Diseño	Cycle	1
Moderator	C. G.	Owner	Politécnica Nacional

Engineer Data

Name	Defects		Preparation Data			Est.
	Major	Minor	Size	Time	Rate	Yield
Geovanna Bustos		2def	2 pág	0.5 h	4pág/h	100
Totals:						

Defect Data

No.	Defect Description	Defects		Engineers (finding major defects)					
		Maj	Min					A	B
3 ¹ .4	G.B. Redacción		X						

¹ 1 Diseño de Alto nivel, 2 Diseño Detallado, 3 Plan de Pruebas de Integración

3.6	G.B. Redacción		X						
Totals			2						
Unique Defects									

Inspection Summary - Product Size: _____ - Size Measure: _____ -
 Total defects for A: - Total defects for B: - C (# common): -
 Total Defects (AB/C): - Number Found (A+B-C): - Number Left: -
 Meeting Time: - Total Inspection Hours: - Overall Rate: -

1.15. FORM INS (IMPLEMENTACIÓN)

Name	<u>Geovanna Bustos</u>	Date	<u>17-04-2007</u>
Team	<u>DPT</u>	Instructor	<u>Ing: Montenegro</u>
Part/Level	<u>Implementación</u>	Cycle	<u>1</u>
Moderator	<u>Cristhian Guallasamin</u>	Owner	<u>Politécnica Nacional</u>

Engineer Data

Name	Defects		Preparation Data			Est.
	Major	Minor	Size	Time	Rate	Yield
A Geovanna Bustos		8	2315	6	385.83	47.06
B Cristhian Guallasamin	9		2315	6	385.83	52.94
Totals:	9	8	4630	12	771.66	

Defect Data

No.	Defect Description	Defects		Engineers (finding major defects)					
		Maj	Min					A	B
1 ¹ .1 ² .1	C.G.Eliminación de Datos (625)	X						X	X
1.2.2	C.G.Eliminación de Datos (677)	X						X	X
1.2.3	C.G.Eliminación de Datos (1184)	X						X	X
1.6.4	C.G.Eliminación de Datos (177)	X						X	X
1.6.5	C.G.Eliminación de Datos (416)	X						X	X
1.7.6	G.B. Comentario (794)		X						
1.7.7	G.B. Comentario (810)		X						
1.7.8	G.B. Comentario (824)		X						
1.7.9	G.B. Comentario (836)		X						
1.7.10	C.G.Eliminación de Datos (917)	X						X	X
1.7.11	G.B. Comentario (917)		X						
1.7.12	C.G.Eliminación de Datos (1074)	X						X	X
1.7.13	C.G.Eliminación de Datos (1085)	X						X	X
1.7.14	C.G.Eliminación de Datos (1098)	X						X	X
1.7.15	G.B. Comentario (1234)		X						

¹ MODULO: 1: Administrar Rutas, 2: Monitorear Rutas

² FORMULARIO: PC 1: frmPrincipal, 2: frmMapa, 3: frmReportes; POCKET PC: 4: frmLogin, 5: frmPrincipal, 6: frmClientes, 7: frmPedido, 8: frmMotivosNoGestion

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
19-03-07	12	20	Imp.	Imp.	15min	X
Description:	Formato de la instrucción.					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
17-04-07	13	70	Imp.	Imp.	5min	X
Description:	Eliminación de Datos innecesarios					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
17-04-07	14	70	Imp.	Imp.	5min	X
Description:	Eliminación de Datos innecesarios					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
17-04-07	15	70	Imp.	Imp.	5min	X
Description:	Eliminación de Datos innecesarios					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
17-04-07	16	70	Imp.	Imp.	5min	X
Description:	Eliminación de Datos innecesarios					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
17-04-07	17	70	Imp.	Imp.	5min	X
Description:	Eliminación de Datos innecesarios					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
17-04-07	18	70	Imp.	Imp.	5min	X
Description:	Eliminación de Datos innecesarios					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
17-04-07	19	70	Imp.	Imp.	5min	X
Description:	Eliminación de Datos innecesarios					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
17-04-07	20	70	Imp.	Imp.	5min	X
Description:	Eliminación de Datos innecesarios					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
17-04-07	21	70	Imp.	Imp.	5min	X
Description:	Eliminación de Datos innecesarios					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
30-04-07	22	70	Imp.	Pruebas de Unidad	5min	X
Description:	Eliminación de Datos innecesarios, en el formulario frmPedido, eliminación del objeto lblunidad					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
30-04-07	23	70	Imp.	Pruebas de Unidad	5min	X
Description:	Eliminación de Datos innecesarios, en el formulario frmPedido, eliminación del objeto lblunidadr					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
30-04-07	24	70	Imp.	Pruebas de Unidad	5min	X
Description:	Eliminación de Datos innecesarios, en el formulario frmPedido, eliminación del objeto lblstock					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
30-04-07	25	70	Imp.	Pruebas de Unidad	5min	X
Description:	Eliminación de Datos innecesarios, en el formulario frmPedido, eliminación del objeto lblstockr					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
30-04-07	26	70	Imp.	Pruebas de Unidad	5min	X

Description:	Eliminación de Datos innecesarios, en el formulario frmPedido, eliminación del objeto lblote					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
30-04-07	27	70	Imp.	Pruebas de Unidad	5min	X
Description:	Eliminación de Datos innecesarios, en el formulario frmPedido, eliminación del objeto lbloter					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
30-04-07	28	70	Imp.	Pruebas de Unidad	5min	X
Description:	Eliminación de Datos innecesarios, en el formulario frmPedido, eliminación del objeto lbldescuento					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
30-04-07	29	70	Imp.	Pruebas de Unidad	5min	X
Description:	Eliminación de Datos innecesarios, en el formulario frmPedido, eliminación del objeto lbldescuentor					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
01-05-07	30	20	Imp.	Pruebas de Unidad	5min	X
Description:	Cambio de Sintaxis. Cambio del formato de presentación del detalle de pedido, en el formulario frmPedido, la función Cargar_Pedido, Format(siPrecioUnitarioP, "##,###0.#0"))					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
01-05-07	31	20	Imp.	Pruebas de Unidad	5min	X
Description:	Cambio de Sintaxis. Cambio del formato de presentación del detalle de pedido, en el formulario frmPedido, la función Cargar_Pedido, Format(siTotalOrdenP, "##,###0.#0"))					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
01-05-07	32	70	Imp.	Pruebas de Unidad	1min	X
Description:	Eliminación de Datos innecesarios, en el formulario frmPedido, eliminación de la función Obtener_Descuento()					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
01-05-07	33	20	Imp.	Pruebas de Unidad	5min	X
Description:	Cambio de Sintaxis. Cambio del formato de presentación del detalle de pedido, en el formulario frmPedido, la variable dPrecioUnitario, Format(dPrecioUnitario, "##,###0.#0"))					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
01-05-07	34	20	Imp.	Pruebas de Unidad	5min	X
Description:	Cambio de Sintaxis. Cambio del formato de presentación del detalle de pedido, en el formulario frmPedido, la variable dPrecioUnitario Format(dPrecioUnitario, "##,###0.#0"))					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
01-05-07	35	20	Imp.	Pruebas de Unidad	5min	X
Description:	Cambio de Sintaxis. Cambio del formato de presentación del detalle de pedido, en el formulario frmPedido, la variable dTotalItem Format(dTotalItem, "##,###0.#0"))					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
01-05-07	36	70	Imp.	Pruebas de Unidad	1min	X
Description:	Eliminación de Datos innecesarios, en el formulario frmPedido, en la función Eliminar_Producto()					
Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
01-05-07	37	70	Imp.	Pruebas de Unidad	1min	X

Description: Eliminación de Datos innecesarios, en el formulario frmPedido, en la función Editar_Cantidad_Producto()						
01-05-07	38	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario frmPedido, eliminación de la función lblUnidadr_SelectedIndexChanged()						
01-05-07	39	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario frmPrincipal, eliminación de la función picInfo_Click()						
01-05-07	40	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario frmPrincipal, eliminación de la función lblInfo_Click()						
01-05-07	41	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario frmPrincipal, eliminación de la función InformaciónMI_Click()						
02-05-07	42	70	Imp.	Pruebas de Unidad	20min	X
Description: Eliminación de Datos innecesarios, en el formulario modGlobales, eliminación de Variables nnecesarias: sPais, sCiudad, sFechaInicio, sCodGrupo, sNomGrupo, strVisCodigo, strPocketPc, Info, Cliente, iVar, myDtExi, myDtTarifa, myDataRow, myDataRow1, myDataRow2, iBnd, jBnd, sNumeroFactura, TSP1, dIvalmp, dTotalmp, dValor, dValorConlva, dValorSinlva, dValorConlvaGlobal.						
02-05-07	43	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario modGlobales, eliminación de La función HabilitarTaskVar()						
02-05-07	44	70	Imp.	Pruebas de Unidad	1min	X
Description: Eliminación de Datos innecesarios, en el formulario modGlobales, en la función Main()						
02-05-07	45	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario modGlobales, eliminación de la función FindWindow()						
02-05-07	46	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario modGlobales, eliminación de la función EnableWindow()						
02-05-07	47	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario frmPrincipal, eliminación de la función Sincronización_Click()						
02-05-07	48	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario frmPrincipal, eliminación de la función Salir_Click()						
02-05-07	49	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario frmPrincipal, eliminación de la función Login_Click()						

02-05-07	50	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario frmPrincipal, eliminación de la función Visitas_Click()						
02-05-07	51	70	Imp.	Pruebas de Unidad	2min	X
Description: Eliminación de Datos innecesarios, en el formulario frmPrincipal, eliminación de la función Salir_ParentChanged()						

1.17. FORM LOGT INDIVIDUAL

Name	C. G.				Date	05-05-2007	
Team	DPT				Instructor	Ing. Montenegro	
Part/Level	Implementación				Cycle	1	
Date	StaZrt	Stop	Interruption Time	Delta Time	Phase/Task	Component	Comments
30-04-07	21:00	23:00	00:00	02:00	Pruebas de Unidad		
01-05-07	21:00	24:00	00:00	03:00			
02-05-07	21:00	24:00	00:00	03:00			
03-05-07	21:00	24:00	00:00	03:00			
04-05-07	21:00	24:00	00:00	03:00			

1.18. FORM LOGTEST INDIVIDUAL

Name	C. G.				Date	05-05-2007		
Team	DPT				Instructor	Ing. Montenegro		
Part/Level	Implementación				Cycle	1		
Date	Phase	Product	Start	Stop	Interruption Time	Delta Time	Problems	Comments
30-04	Imp	frmPedido	21:00	21:05	00:00	00:05		
30-04	Imp	frmPedido	21:10	21:15	00:00	00:05		
30-04	Imp	frmPedido	21:20	21:25	00:00	00:05		
30-04	Imp	frmPedido	21:30	21:35	00:00	00:05		
30-04	Imp	frmPedido	21:40	21:45	00:00	00:05		
30-04	Imp	frmPedido	21:50	21:55	00:00	00:05		
30-04	Imp	frmPedido	22:00	22:05	00:00	00:05		
30-04	Imp	frmPedido	22:10	22:15	00:00	00:05		
01-05	Imp	frmPedido	21:00	21:05	00:00	00:05		
01-05	Imp	frmPedido	21:10	21:15	00:00	00:05		
01-05	Imp	frmPedido	21:20	21:21	00:00	00:01		
01-05	Imp	frmPedido	21:25	21:30	00:00	00:05		
01-05	Imp	frmPedido	21:35	21:40	00:00	00:05		
01-05	Imp	frmPedido	21:45	21:50	00:00	00:05		
01-05	Imp	frmPedido	22:00	22:01	00:00	00:01		
01-05	Imp	frmPedido	22:05	22:06	00:00	00:01		
01-05	Imp	frmPedido	22:10	22:02	00:00	00:02		
01-05	Imp	frmPrincipal	22:15	22:17	00:00	00:02		
01-05	Imp	frmPrincipal	22:20	22:22	00:00	00:02		

01-05	Imp	frmPrincipal	22:25	22:27	00:00	00:02		
02-05	Imp	modGlobales	21:00	21:20	00:00	00:20		
02-05	Imp	modGlobales	21:30	21:32	00:00	00:02		
02-05	Imp	modGlobales	21:35	21:36	00:00	00:01		
02-05	Imp	modGlobales	21:40	21:42	00:00	00:02		
02-05	Imp	modGlobales	21:45	21:47	00:00	00:02		
02-05	Imp	frmPrincipal	21:50	21:52	00:00	00:02		
02-05	Imp	frmPrincipal	21:55	21:57	00:00	00:02		
02-05	Imp	frmPrincipal	22:00	22:02	00:00	00:02		
02-05	Imp	frmPrincipal	22:10	22:12	00:00	00:02		
02-05	Imp	frmPrincipal	22:15	22:17	00:00	00:02		

1.19. PROCESS IMPROVEMENT PROPOSAL: PIP

Name	<u>Administrador de la calidad/procesos</u>	Date	<u>22-05-2007</u>
Team	<u>DPT</u>	Instructor	<u>Ing. Montenegro</u>
Part/Level	<u>Postmortem</u>	Cycle	<u>1</u>
Process	<u>Postmortem</u>	Phase	<u>Postmortem</u>
PIP Number	<u>1</u>	Priority	<u>Normal</u>

Problem Description

Briefly describe the problem encountered and its impact.

El equipo tenía un tiempo reducido diario para el desarrollo del proyecto que hizo que el tiempo de duración en semanas sea grande.

Proposal Description

Describe suggested changes as completely as possible, including affected forms, scripts, and so on

Se recomienda disponer de más tiempo diario para el desarrollo de proyectos en un futuro.

Submit completed PIP to the quality/process manager and keep a copy.

Do not write below this line

PIP Control #	_____	Organization	_____
Received	_____	Acknowledged	_____
Updated	_____	Closed	_____
Changes	_____		

1.20. TEAM AND PEER EVALUATION PEER INDIVIDUAL

Name	<u>G.B.</u>	Team	<u>DPT</u>	Instructor	<u>Ing. Montenegro</u>
Date	<u>23-05-2007</u>	Cycle No.	<u>1</u>	Week No.	<u>34</u>

For each role, evaluate the work required and the relative difficulty in % during this cycle.		
Role	Work Required	Role Difficulty

Team Leader Planning Manager Quality/Process Manager	50%	50%
Development Manager Support Manager	50%	50%
Total Contribution (100%)	100%	100%

Rate the overall team against each criterion. Circle one number from 1 (inadequate) to 5 (superior).					
Team spirit	1	2	3	4	5
Overall effectiveness	1	2	3	4	5
Rewarding experience	1	2	3	4	5
Team productivity	1	2	3	4	5
Process quality	1	2	3	4	5
Product quality	1	2	3	4	5

Rate role for overall contribution. Circle one number from 1 (inadequate) to 5 (superior).					
Team Leader Planning Manager Quality/Process Manager	1	2	3	4	5
Development Manager Support Manager	1	2	3	4	5

Rate each role for helpfulness and support. Circle one number from 1 (inadequate) to 5 (superior).					
Team Leader Planning Manager Quality/Process Manager	1	2	3	4	5
Development Manager Support Manager	1	2	3	4	5

Rate each role for how well it was performed. Circle one number from 1 (inadequate) to 5 (superior).					
Team Leader Planning Manager Quality/Process Manager	1	2	3	4	5
Development Manager Support Manager	1	2	3	4	5

2 ANEXOS

2.1. RESPONSABILIDADES DE LOS ROLES DEL EQUIPO

Responsabilidad	Líder del Equipo	Adm. del Desarrollo	Adm. de la Planificación	Adm. de la Calidad/Procesos	Adm. del Soporte
Construir y mantener un equipo efectivo	X				
Resolver problemas entre los miembros del equipo	X				
Monitorear y reportar el progreso del equipo	X				
Representar como facilitador en las reuniones	X				
Interfase con el Jefe del Proyecto.	X				
Mantener el project notebook	X				
Ayudar a asignar las tareas de equipo	X				
Guiar todo el trabajo de desarrollo		X			
Guiar la planificación del equipo y el monitoreo de procesos			X		
Guiar la planificación y el monitoreo de la calidad				X	
Proporcionar soporte en los procesos al equipo				X	
Actuar como moderador de la inspección				X	
Mantener normas del equipo y un glosario				X	
Manejar los reportes de las reuniones				X	
Alertar al equipo de los problemas de calidad				X	
Obtener las herramientas y el soporte necesarios					X
Manejar la administración de configuración					X
Guiar los cambios del comité de control					X
Actuar como auditor del equipo					X
Manejar los problemas y el monitoreo de riesgos					X
Mantener el glosario del sistema					X
Modelado del negocio	X				
Análisis	X	X	X	X	X
Desarrollar el producto	X	X	X	X	X
Hacer planes personales	X	X	X	X	X
Monitorear el trabajo personal	X	X	X	X	X
Producir productos con calidad	X	X	X	X	X
Seguir las prácticas personales disciplinadas	X	X	X	X	X
Producir un release.				X	
Empaquetar el software.	X				
Instalar el software.	X	X			
Apoyar a los usuarios.	X				X
Realizar pruebas beta.	X				X
Migración de datos.	X	X			
Aceptación formal.	X				

2.2. STANDARD QUAL

Medida	Meta	Comentario
Porcentaje de defectos libres (PDF)		
Compilear	> 10%	
Pruebas de unidad	> 50%	
Pruebas de integración	> 70%	
Pruebas del sistema	> 90%	
Defectos/KLOC		
Total de defectos inyectados	75-150	Si no esta entrenado en PSP usar 100-200
Compilear	< 10	Todos los defectos señalados por el compilador
Pruebas de unidad	< 5	Solo defectos graves
Construcción e integración	< 0.5	Solo defectos graves
Pruebas del sistema	< 0.2	Solo defectos graves
Proporciones de Defectos		
Defectos revisados DLD/defectos en las pruebas de unidad	> 2.0	Solo defectos graves
Defectos en la revisión de código/ defectos al compilear	> 2.0	Solo defectos graves
Desarrollo de proporciones de tiempo		
Inspección de requisitos/tiempo de requisitos	> 0.25	Incluido el tiempo para obtener
Inspección HLD/ tiempo HLD	> 0.5	Solo trabajo de diseño, no estudios
DLD/tiempo de codificación	> 1.00	
Revisión DLD/ tiempo DLD	> 0.5	
Revisión de código/tiempo de codificar	> 0.5	
Proporciones de revisión e inspección		
Páginas de requerimientos/hora	< 2	Páginas de texto a un solo espacio
Páginas HLD/hora	< 5	Estructura de la lógica del diseño
Líneas de texto DLD/hora	< 100	Líneas de pseudo código igualan aproximadamente a 3 LOC cada una.
LOC de código/hora	< 200	LOC lógico
Proporciones de defectos inyectados		
Defectos de los requerimientos/hora	0.25	Solo defectos graves
Defectos HLD/hora	0.25	Solo defectos graves
Defectos DLD/hora	2.0	Solo defectos de diseño
Defectos de código/hora	4.0	Solo defectos graves
Defectos de compilación/hora	0.3	Todos los defectos señalados por el compilador
Defectos de las pruebas de unidad/hora	0.2	Solo defectos graves
Proporciones de defectos removidos		
Defectos en la inspección de requerimientos/hora	0.5	Solo defectos graves
Defectos de la inspección HLD/hora	0.5	Solo defectos graves
Defectos de la revisión DLD /hora	2.0	Solo defectos de diseño
Defectos de la inspección DLD/hora	0.5	Solo defectos de diseño
Defectos de la revisión de código/hora	6.0	Solo defectos graves
Defectos de la inspección de código/hora	1.0	Solo defectos graves
Rendimiento de las fases		
Inspección de requerimientos	~ 70%	No contabilizar comentarios de editorial
Revisión e inspección del diseño	~ 70%	Usar análisis de estado y tablas de rastreo

Revisión e inspección del código	~ 70%	Usar listas de control personales
Compilación	~ 50%	90+% de defectos de sintaxis
Pruebas de unidad de 5 a menos defectos/KLOC	~ 90%	Para defectos grandes/KLOC: 50-75%
Construcción, integración, pruebas del sistema at<1.0 defectos/KLOC	~ 80%	Para defectos grandes/KLOC: 30-65%
Rendimiento de los procesos		
Antes de la compilación	> 75%	Asumir los que parecen métodos de diseño
Antes de las pruebas de unidad	> 85%	Asumir los chequeos de las listas en la revisión.
Antes de las pruebas de construcción e integración	>97.5%	Para pequeños productos, 1 defecto máximo
Antes de las pruebas del sistema	> 99%	Para pequeños productos, 1 defecto máximo

2.3. METAS POR ROL

Líder del Equipo

Meta 1	Construir y mantener un equipo efectivo.
Medida 1.1	El desempeño del proyecto contra de los costos, calendarios, metas y calidad.
Medida 1.2	Las evaluaciones PEER de la efectividad del equipo global.
Medida 1.3	Las evaluaciones PEER de la contribución del líder del equipo.
Medida 1.4	Los valores PEER de los miembros del equipo sobre el líder del equipo.
Meta 2	Motivar a todos los miembros del equipo a trabajar agresivamente en el proyecto
Medida 2.1	Todos los miembros del equipo trabajan las horas que se comprometieron.
Medida 2.2	Todos los miembros del equipo encuentran un valor merecido de su compromiso.
Medida 2.3	Todos los miembros del equipo siguen los procesos TSP, registran todos los datos y llenan todas las formas requeridas.
Meta 3	Resolver problemas entre los miembros del equipo.
Medida 3.1	Los valores PEER de los miembros del equipo de la utilidad y apoyo del rol de líder del equipo.
Meta 4	Mantener al instructor totalmente informado sobre el progreso del equipo.
Medida 4.1	Precisos y completos los reportes del estado semanales.
Medida 4.2	Conocimiento oportuno del instructor del estado del proyecto.
Meta 5	Desempeñar efectivamente como facilitador de las reuniones del equipo.
Medida 5.1	Las evaluaciones PEER del equipo sobre el proyecto son un premio a la experiencia.

Administrador de la Planificación

Meta 1	Producir un completo, preciso y exacto plan para el equipo y todos los miembros del equipo.
Medida 1.1	El plan del equipo cubre todas las tareas del ciclo de desarrollo.
Medida 1.2	El plan es totalmente documentado en TASK y SCHEDULE.

Medida 1.3	Las horas de tarea promedio es menor que 5 y no hay tareas Individuales del equipo que sean más de 10 horas.
Medida 1.4	Las horas semanales y las horas totales del plan exactamente representan el resultado actual del ciclo.
Meta 2	Reportar de forma exacta el estado del equipo todas las semanas.
Medida 2.1	Provee un reporte de estado del equipo completo y exacto semanalmente.
Medida 2.2	Los miembros del equipo actualizan sus formas TASK, SCHEDULE y WEEK y proveen esta al mismo tiempo al administrador de la planificación.
Medida 2.3	Si uno o mas miembros del equipo no reportan a tiempo los datos, se busca ayuda al líder del equipo y al instructor.

Administrador de la Calidad/Procesos

Meta 1	Todos los miembros del equipo exactamente reportan y usan apropiadamente los datos TSP.
Medida 1.1	La magnitud con que el equipo registra fielmente y usa todos los datos TSP requeridos.
Meta 2	El equipo sigue fielmente TSP y produce un producto de calidad.
Medida 2.1	El equipo siguió el TSP.
Medida 2.2	La calidad del equipo se desempeña conforme a la calidad del plan.
Medida 2.3	El grado con el que se mantienen el líder del equipo y el instructor informados de los problemas de calidad.
Medida 2.4	El grado con el que se consiguieron las metas sin el antagonismo del equipo o de algún miembro del equipo.
Meta 3	Todas las inspecciones del equipo son moderadas adecuadamente y reportadas.
Medida 3.1	Todas las inspecciones fueron conducidas de acuerdo con el script INS y los estándares de calidad del equipo.
Medida 3.2	Las formas INS están completadas por todas las inspecciones del equipo y todos los reportes de defectos grandes se reportan en las formas LOGD de sus propietarios.
Meta 4	Todos las reuniones del equipo son reportadas con precisión y los reportes son puestos en el Project nootebook.
Medida 4.1	El porcentaje de las reuniones de los equipos con los reportes se almacenan en el Project nootebook.

Administrador del Desarrollo

Meta 1	Producir un producto de alta calidad.
Medida 1.1	El equipo produce un producto útil y totalmente documentado, que satisface los requerimientos básicos y las necesidades manifestadas.
Medida 1.2	Los requerimientos son identificados de las necesidades manifestadas en el SRS, a él SDS y al final de la implementación.
Medida 1.3	El diseño del producto es totalmente documentado y se hallan estándares de diseño del equipo.
Medida 1.4	La implementación representa fielmente el diseño.
Medida 1.5	El producto cumple con los criterios de calidad.
Medida 1.6	El producto cumple con los objetivos funcionales y operacionales.
Meta 2	Utilizar totalmente las destrezas y habilidades de los miembros del equipo
Medida 2.1	Las evaluaciones de la forma PEER de quien desempeño el rol de administrador del desarrollo.
Medida 2.2	Las evaluaciones de la forma PEER de la utilidad y el apoyo del administrador del

Medida 2.3	desarrollo. Las evaluaciones de la forma PEER de la calidad del producto.
------------	------------------------------------------------------------------------------

Administrador de Soporte

Meta 1	El equipo tiene todas las herramientas apropiadas y métodos de soporte de su trabajo.
Medida 1.1	El equipo tiene un sistema administrador de cambios, un sistema de seguimiento de problemas, un sistema de administrador de configuración, un ambiente de desarrollo común, y un sistema de soporte TSP.
Medida 1.2	El equipo usa efectivamente las herramientas que posee.
Meta 2	No hay cambios sin autorización sobre el producto base.
Medida 2.1	Todos los elementos del producto final tienen un control de configuración.
Medida 2.2	Todos los cambios tienen productos con un control de configuración.
Medida 2.3	Cuando fueron hechos los cambios en el código, estos están reflejados en la documentación del diseño base.
Meta 3	Todos los riesgos del equipo y los problemas están registrados en el registro de seguimiento de problemas (ITL) y reportados en cada semana.
Medida 3.1	El porcentaje de los riesgos y problemas que están registrados y rastreados están en el sistema de seguimiento de problemas.
Meta 4	El equipo tiene metas reusables para el ciclo de desarrollo.
Medida 4.1	El equipo tiene una lista de partes reusables.
Medida 4.2	El reuso y los porcentajes de nuevos reusos fueron medidos y rastreados.
Medida 4.3	El equipo almacena algunos reusos con el primer ciclo de desarrollo.
Medida 4.4	El nivel de reuso incrementa con cada ciclo.

2.4. SOFTWARE DESIGN SPECIFICATIONS (SDS)

El producto que se va a crear “Software para Administración y Monitoreo de Rutas a seguir por los Vendedores de Puerta a Puerta” esta compuesto por dos subsistemas: Administrar Rutas y Monitorear Rutas. El primer subsistema se forma de dos módulos: Generar Información, Sincronizar Información, Gestionar y Recibir Información. El segundo se forma del módulo Monitorear. La estructura del sistema de acuerdo con la funcionalidad se presenta en la tabla 2.7.

El módulo Generar Información le permitirá al administrador seleccionar el vendedor y el día de visita para obtener un mapa en él que estén representadas las direcciones de los clientes, en el cual se pueda seleccionar los clientes que el vendedor va a visitar y que se registran en la PC.

El módulo Sincronizar Información le permitirá al administrador sincronizar los datos de la PC y la Pocket PC.

El módulo Gestionar le permitirá al vendedor registrar tanto los pedidos como los motivos de no gestión de los clientes, con la ayuda de la Pocket PC.

El módulo Recibir Información le permitirá al administrador registrar la información de la gestión del vendedor en la PC.

El módulo Monitorear Gestión del Vendedor le permitirá al administrador seleccionar la fecha, el vendedor y una de las siguientes opciones: un mapa con la ruta seguida por el vendedor, una lista de los clientes visitados por el vendedor que hayan realizado pedido, una lista de los clientes visitados por el vendedor con motivo de no gestión y una lista de los clientes no visitados por el vendedor.

Subsistema	Módulo	Funciones
Administrar Rutas	Generar Información	Presentar lista de vendedores Presentar lista de días de visita Ver un mapa con los clientes. Permitir la selección de los clientes del mapa. Generar (clientes, productos, MNG y vendedores)
	Sincronizar Información	Sincronizar la información de la PC y la Pocket PC.
	Gestionar	Presentar lista de vendedores Validar vendedor (vendedor, password) Ver clientes. Presentar lista de productos. Ver datos del producto. Permitir el Ingreso de datos del detalle del pedido Registrar un pedido. Ver MNGs Registrar MNG.
	Recibir Información	Recibir la información de la gestión del vendedor.
Monitorear Rutas	Monitorear Gestión del Vendedor	Presentar lista de vendedores Presentar un mapa de la ruta seguida por el vendedor. Presentar una lista de clientes con pedido. Presentar una lista de clientes con motivo de no gestión. Presentar una lista de clientes no visitados.

El administrador de la calidad/procesos guía el esfuerzo para producir el glosario de nombres y los estándares de diseño que se usan en el diseño.

Glosario de nombres

Nombre	Descripción
Administrar Rutas	Subsistema
Monitorear Rutas	Subsistema
Generar Información	Componente del subsistema Administrar Rutas
Enviar Información	Componente del subsistema Administrar Rutas
Gestionar	Componente del subsistema Administrar Rutas
Recibir Información	Componente del subsistema Administrar Rutas
Monitorear Gestión del Vendedor	Componente del subsistema Monitorear Rutas

Estándares de diseño

Durante la revisión e inspección de los estándares de diseño no se encontraron defectos por lo tanto no se modificaron.

Diseño Detallado

Durante la revisión e inspección del diseño detallado no se encontraron defectos por lo tanto los diagramas no se modificaron.

Plan de pruebas de integración

El plan de pruebas considera que se deben cumplir los siguientes puntos:

En el módulo Generar Información se cumpla con los siguientes ítems:

- Se presente una lista de los vendedores.
- Se permita seleccionar un vendedor de la lista de vendedores.
- Se presente los días de visita del vendedor.
- Se permita seleccionar un día de visita de la lista.
- Se presente un mapa con los clientes del vendedor para el día de visita seleccionado.
- Se permita seleccionar los clientes para la ruta que el vendedor deberá gestionar.

- Se registren los datos de los vendedores, clientes, productos, MNG en la base de datos necesarios para la Gestión del vendedor.

En el módulo Sincronizar Información se cumpla con los siguientes ítems:

- Cuando se va a cargar los datos para la Gestión del vendedor se actualicen los datos de la Pocket PC con los datos de la PC.
- Cuando se va a cargar los datos de la Gestión del vendedor se actualicen los datos de la PC con los datos de la Pocket PC.

En el módulo Gestionar se cumpla con los siguientes ítems:

- Se presente una lista de los vendedores.
- Se permita seleccionar un vendedor de la lista para ingresar un pedido o un motivo de no gestión.
- Se presente una lista de los productos.
- Se permita seleccionar un producto para ingresar al detalle del pedido
- Se permita ingresar el detalle de pedido.
- Se permita eliminar el detalle de pedido.
- Se registren los pedidos.
- Se presente una lista de los motivos de no gestión (MNG)
- Se permita seleccionar un MNG,
- Se permita ingresar una observación del MNG.
- Se registren los MNG`s

En el módulo Recibir Información se registren los datos de la gestión del vendedor en la PC.

En el módulo Monitorear Gestión del Vendedor se cumpla con los siguientes ítems:

- Se presente una lista de los vendedores.
- Se permita seleccionar un vendedor.

- Se permita seleccionar una Fecha.
- Se presente las opciones: Ruta seguida, Clientes visitados con pedido, Clientes visitados con MNG, Clientes no visitados.
- Se permita seleccionar una opción anterior.
- Se presente un mapa con la ruta seguida por el vendedor en la fecha seleccionada.
- Se presente una lista de los clientes visitados por el vendedor y que hayan realizado un pedido.
- Se presente una lista de los clientes visitados por el vendedor y que no realizaron pedido por un motivo de no gestión.
- Se presente una lista de los clientes no visitados por el vendedor y que se establecieron en la ruta.

2.5. Calidad de los componentes

Summary Rates	Plan	Actual	Meta	Comentario
LOC/hour	8400/1039=8.08	5979/847=7.06		
% Reuse (% of total LOC)	0	0		
% New Reuse (% of N&C LOC)	0	0		
Percent Defect-Free (PDF)				
In compile	$((8-2)/8)=75$	$((8-2)/8)=75$	> 10%	De los 8 módulos 2 tienen DF, igual al plan
In unit test	$(8-1)/8=88$	$(8-1)/8=88$	> 50%	De los 8 módulos 1 tiene PDF
In build and integration	$(8-0.5)=94$			
In system test	$(8-0.1)=99$			
Defect/page				
Requirements inspection	2/10=0.2	1/12=0.08		Los defectos/página son mínimos
HLD inspection	3/3=1	0/2=0		El HLD está bien elaborado
DLD inspection	2/15=0.13	0/19=0		El DLD está bien elaborado
Defects/KLOC				
DLD review	3/8.4=0.36	0/5.979 =0		Cumple con el plan
DLD inspection	2/8.4=0.24	0/5.979 =0		Cumple con el plan
Code review	1/8.4=0.12	9/5.979 =1.51		Es mayor que en el plan
Compile	20/8.4=2.38	21/5.979 =3.51	< 10	No cumple con el plan, pero sí con los estándares de calidad
Code inspection	1/8.4=0.12	0/5.979 =0		Es mayor que en el plan
Unit test	1/8.4=0.12	5/5.979=0.84	< 5	No cumple con el plan, pero sí con los estándares de calidad
Build and integration	1/8.4=0.12			
System test	1/8.4=0.12			
Total development				
Defect Ratios				
Code review/Compile	0.12/2.38= 0.05	1.51/3.51=0.43	> 2.0	No hay defectos graves
DLD review/Unit test	0.36/0.12= 3	0/0.84=0	> 2.0	No hay defectos graves
Development time ratios (%)				

Requirements inspection/Requirements	2/12=16.67%	1/14=7.14%	> 0.25	No cumple con el plan, pero si con los estándares de calidad
HLD inspection/HLD	1/4=25.00%	0.25/2=12.5%	> 0.5	No cumple con el plan, pero si con los estándares de calidad
DLD/code	12/742=1.62%	10/578=1.73%	> 1.00	Cumple con el plan, y con los estándares de calidad
DLD review/DLD	1/12=8.33%	0.5/10=5%	> 0.5	No cumple con el plan, pero si con los estándares de calidad
Code review/code	60/742=8.09%	10/649=1.54%	> 0.5	No cumple con el plan, pero si con los estándares de calidad
A/FR				
Review rates				
DLD page/hour	15/1=15	19/0.5=38	C	El diseño detallado esta claro
Code LOC/hour	8400/30=280	5979/10=597.9	C	El código es claro y correcto
Inspection rates				
Requirement pages/hour	10/2=5	12/1=12	< 2	Los requerimientos están claros
HLD pages/hour	3/1=3	2/0.25=8	< 5	El diseño de alto nivel esta claro
DLD page /hour	15/1=15	19/0.5=38		El diseño detallado esta claro
Code LOC/hour	8400/30=280	5979/7=854.14	<200	El código es claro y correcto
Defect-injection Rates (Defects/Hr.)				
Requirements	2/12=0.17	1/14=0.07	0.25	Los defectos graves son pocos
HLD	¼=0.75	0	0.25	No hay defectos graves
DLD	5/12=0.42	0	2.0	No hay defectos graves
Code	25/742=0.03	35/649=0.05	4.0	Los defectos graves son pocos
Compile	0	0	0.3	No hay defectos graves
Unit test	0	0	C	No hay defectos graves
Build and integration	0			
System test	0			
Defect-removal Rates (Defects/Hr.)				
Requirements inspection	2/2=1	1/1=1	0.5	Los defectos encontrados son removidos
HLD inspection	3/1=3	0	0.5	No hay defectos graves
DLD review	3/0.5=6	0	2.0	No hay defectos graves
DLD inspection	2/0.5=4	0	0.5	No hay defectos graves

Code review	$1/30=0.03$	$9/11=0.81$	6.0	No hay defectos graves
Compile	$20/(742-732)=2$	$21/(578-228)=0.42$		Los defectos encontrados son removidos
Code inspection	$1/30=0.03$	$0/10=0$	1.0	Los defectos encontrados son removidos
Unit test	$1/20=0.05$	$5/28=0.18$		Los defectos encontrados son removidos
Build and integration	$1/16=0.06$			
System test	$1/8=0.13$			
Phase Yields				
Requirements inspection	$2/2=100$	$1/1=100\%$	~ 70%	Los defectos encontrados son removidos
HLD inspection	$3/(5-2)=100$	$1/(3-1)=50\%$	~ 70%	
DLD review	$3/(8-5)=100$	$1/(3-2)=100\%$	~ 70%	
Test development				Los defectos encontrados son removidos
DLD inspection	$2/(10-8)=100$	100%	~ 70%	
Code review	$1/(35-30)=20.00$	$9/(38-24)=64.29\%$	~ 70%	Existen defectos que se pasan por alto
Compile	$20/(35-10)=80.00$	$21/(38-3)=60.00\%$	~ 50%	Existen defectos que se pasan por alto
Code inspection	$1/(35-31)=25.00$	64.29%	~ 70%	Existen defectos que se pasan por alto
Unit test	$1/(35-32)=33.33$	$5/(38-33)=100\%$	~ 90%	Los defectos que se pasaron por alto son removidos
Build and integration	$1/(35-34)=100$		~ 80%	
System test			~ 80%	
Process Yields				
% before compile	$10/10=100$	$3/3=100$	> 75%	No se dejan defectos sin resolver
% before unit test	$32/35=91.43$	$33/33=100$	> 85%	El porcentaje de defectos sin remover es corto, ya que el sistema es pequeño.
% before build and integration	$33/35=94.29$	$38/38=100$	>97.5%	
% before system test	$34/35=97.14$		> 99%	
% before system delivery	$35/35=100.00$			

2.6. ENCUESTAS

ENCUESTA DEL SOFTWARE N- 1

1. En general ¿Cuál es su grado de satisfacción con la Aplicación?

- Completamente satisfecho
 Satisfecho
 Insatisfecho
 Completamente insatisfecho

2. ¿Qué aspectos de la Aplicación cree que podríamos mejorar?

Creo que se podrían variar los colores de las pantallas

3. ¿Qué es lo que más le ha gustado de la Aplicación?

El mapa en donde se presentan los clientes del vendedor

4. Tras utilizar la Aplicación, valore su grado de satisfacción con los siguientes aspectos:

	Completament e Satisfecho	Satisfech o	Insatisfeh o	Completament e Insatisfecho	No aplicable
Diseño atractivo		X			
Elementos gráficos	X				
Utilidad de la Aplicación	X				
Facilidad de uso / navegación	X				
Comodidad		X			
Rapidez de la información	X				

5. ¿Recomendaría el uso de la aplicación a otras personas?

- Si
 No

Por qué?

La utilidad que brinda, evita que el vendedor lleve facturas para llenarles a mano que es un trabajo engorroso. Proporciona reportes que facilitan el control de la gestión de los vendedores.

ENCUESTA DEL SOFTWARE N- 2

1. En general ¿Cuál es su grado de satisfacción con la Aplicación?

- Completamente satisfecho
 Satisfecho
 Insatisfecho
 Completamente insatisfecho

2. ¿Qué aspectos de la Aplicación cree que podríamos mejorar?

Creo que se podría cambiar la presentación de los productos que se van llenando en el pedido, mostrarles en una sola lista, no en dos.

3. ¿Qué es lo que más le ha gustado de la Aplicación?

El mapa de los clientes y la Pocket Pc para la toma de los pedidos

4. Tras utilizar la Aplicación, valore su grado de satisfacción con los siguientes aspectos:

	Completament e Satisfecho	Satisfech o	Insatisfeh o	Completament e Insatisfecho	No aplicable
Diseño atractivo	X				
Elementos gráficos	X				
Utilidad de la Aplicación	X				
Facilidad de uso / navegación	X				
Comodidad	X				
Rapidez de la información	X				

5. ¿Recomendaría el uso de la aplicación a otras personas?

- Si
 No

Por qué?

Ayuda a los vendedores para tomar los pedidos con mayor rapidez. Los datos que se obtienen permiten fijar la ruta que siguió el vendedor, para controlar el trabajo que realizan.

ENCUESTA DEL SOFTWARE N- 3

1. En general ¿Cuál es su grado de satisfacción con la Aplicación?

- Completamente satisfecho
 Satisfecho
 Insatisfecho
 Completamente insatisfecho

2. ¿Qué aspectos de la Aplicación cree que podríamos mejorar?

Si es posible añadir las ayudas del sistema.

3. ¿Qué es lo que más le ha gustado de la Aplicación?

El mapa de los clientes y la Pocket Pc para la toma de los pedidos

4. Tras utilizar la Aplicación, valore su grado de satisfacción con los siguientes aspectos:

	Completament e Satisfecho	Satisfech o	Insatisfech o	Completament e Insatisfecho	No aplicable
Diseño atractivo	X				
Elementos gráficos	X				
Utilidad de la Aplicación	X				
Facilidad de uso / navegación	X				
Comodidad	X				
Rapidez de la información	X				

5. ¿Recomendaría el uso de la aplicación a otras personas?

- Si

No

Por qué?

En una empresa ayudaría en la gestión del Administrador de los vendedores, ya que el sistema presenta reportes con los datos de los pedidos de los vendedores.

ENCUESTA DEL SOFTWARE N- 4

1. En general ¿Cuál es su grado de satisfacción con la Aplicación?

- Completamente satisfecho
 Satisfecho
 Insatisfecho
 Completamente insatisfecho

2. ¿Qué aspectos de la Aplicación cree que podríamos mejorar?

Añadir animaciones atractivas y colores a los títulos

3. ¿Qué es lo que más le ha gustado de la Aplicación?

Los mapas que se muestran y la Pocket Pc

4. Tras utilizar la Aplicación, valore su grado de satisfacción con los siguientes aspectos:

	Completament e Satisfecho	Satisfech o	Insatisfeh o	Completament e Insatisfecho	No aplicable
Diseño atractivo	X				
Elementos gráficos		X			
Utilidad de la Aplicación	X				
Facilidad de uso / navegación	X				
Comodidad	X				
Rapidez de la	X				

información					
-------------	--	--	--	--	--

5. ¿Recomendaría el uso de la aplicación a otras personas?

- Si
 No

Por qué?

Ahora la mayor parte de empresas tienen vendedores puerta a puerta, a quienes no se les puede controlar su trabajo, este sistema permite controlar con los reportes que proporciona.

ENCUESTA DEL SOFTWARE N- 5

1. En general ¿Cuál es su grado de satisfacción con la Aplicación?

- Completamente satisfecho
 Satisfecho
 Insatisfecho
 Completamente insatisfecho

2. ¿Qué aspectos de la Aplicación cree que podríamos mejorar?

Ninguno

3. ¿Qué es lo que más le ha gustado de la Aplicación?

El uso de un dispositivo móvil para que los vendedores faciliten su trabajo para tomar los pedidos de los clientes.

4. Tras utilizar la Aplicación, valore su grado de satisfacción con los siguientes aspectos:

	Completament e Satisfecho	Satisfech o	Insatisfech o	Completament e Insatisfecho	No aplicable
Diseño atractivo	X				
Elementos gráficos	X				
Utilidad de la Aplicación	X				
Facilidad de uso / navegación	X				

Comodidad	X				
Rapidez de la información	X				

5. ¿Recomendaría el uso de la aplicación a otras personas?

Si

No

Por qué?

Disminuye tiempo a los vendedores porque les evita llenar pedidos en papel. Ahorra dinero porque evita papel, impresiones o copias a la empresa y ofrece el control del trabajo realizado por los vendedores ya que ofrece reportes con la ruta seguida, los pedidos realizados y los clientes visitados y no visitados