

SEIS SIGMA EN EL DESARROLLO DE PROYECTOS DE INFORMACIÓN
TECNOLÓGICA DE LA DIVISIÓN DE SERVICIOS DE INFORMACIÓN Y SU
APLICACIÓN EN UN MÓDULO DE LA INTERFACE WEB DEL SISTEMA DE
LA BIBLIOTECA DE LA UIS

DIANA MARCELA ZAMBRANO OLARTE
SERGIO ANDRÉS CONTRERAS NIGRINIS

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2004

SEIS SIGMA EN EL DESARROLLO DE PROYECTOS DE INFORMACIÓN
TECNOLÓGICA DE LA DIVISIÓN DE SERVICIOS DE INFORMACIÓN Y SU
APLICACIÓN EN UN MÓDULO DE LA INTERFACE WEB DEL SISTEMA DE
LA BIBLIOTECA DE LA UIS

DIANA MARCELA ZAMBRANO OLARTE
SERGIO ANDRÉS CONTRERAS NIGRINIS

Proyecto de grado presentado como requisito parcial
para optar al título de Ingeniero de Sistemas

Director
ENRIQUE TORRES LOPEZ
Jefe División de Servicios de Información

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2004

AGRADECIMIENTOS

Ahora que finaliza una etapa más de nuestras vidas, queremos agradecer a todas aquellas personas que aportaron para que culmináramos con éxito una de las metas de nuestra vida.

Queremos antes que nada agradecer a Dios, por estar siempre presente en nuestras vidas y ser nuestra luz, nuestro camino y nuestra fortaleza.

Al Ingeniero Edwin Suarez, por la confianza depositada, por todas sus enseñanzas y por su apoyo incondicional.

Al Ingeniero Enrique Torres Lopez, por permitirnos pertenecer a la División de Servicios de Información y permitirnos crecer profesionalmente y personalmente.

Al Ingeniero Gilberto Rivas Rincón, por su tiempo compartido y por sus enseñanzas en todos los aspectos de la vida, que Dios lo tenga en su gloria.

A todos los compañeros DSI, que en cualquier momento nos brindaron su apoyo y que fueron partícipes de este aprendizaje.

CONTENIDO

	pág
INTRODUCCIÓN	1
PARTE I	
1. PRESENTACIÓN	3
1.1 DESCRIPCIÓN DEL DOCUMENTO	3
1.2 ANTECEDENTES	5
1.2.1 Metodología Seis Sigma.	5
1.2.2 Módulo de consultas del catálogo bibliográfico de la Biblioteca vía web.	7
1.3 OBJETIVOS	8
1.3.1 Objetivo General.	8
1.3.2 Objetivos Específicos.	8
2. GENERALIDADES DE SEIS SIGMA	10
2.1 HISTORIA Y EVOLUCIÓN	10
2.2 DEFINICIÓN DE SEIS SIGMA	10
2.3 PRINCIPIOS DE SEIS SIGMA	12
2.3.1 Prevenir defectos.	12
2.3.2 Reducir la Variación.	13
2.3.3 Orientación en el Cliente.	14
2.3.4 Decisiones basadas en hechos.	15
2.3.5 Trabajo en Equipo.	16
2.4 HERRAMIENTAS Y ENTRENAMIENTO	16
3. FASE DEFINIR	20
3.1 ENTENDER EL MODELO DEL NEGOCIO	21
3.2 DEFINIR EL PROBLEMA	22

3.2.1	Retroalimentación con el Cliente o VOC.	23
3.2.2	Oportunidades externas.	23
3.2.3	Experiencia.	24
3.3	DEFINIR ROLES Y RESPONSABILIDADES	26
3.3.1	Líderes Ejecutivos	26
3.3.2	Patrocinadores	27
3.3.3	Master Black Belts (MMB)	28
3.3.4	Black Belts (BB)	29
3.3.5	Green Belts (GB)	30
3.4	FORMAR EL EQUIPO DE TRABAJO	30
3.5	IDENTIFICAR LOS CLIENTES	32
3.6	IDENTIFICAR LO CRÍTICO PARA LA CALIDAD (CTQ'S)	34
3.7	DESARROLLAR EL PLAN DE PROYECTO	34
3.7.1	Definición del alcance y objetivos	35
3.7.2	Costo/ Beneficio	37
3.7.3	Presupuesto	42
3.7.4	Riesgos	45
3.7.5	Cronograma	50
3.8	DESARROLLAR EL CUADRO RESUMEN DEL PROYECTO	52
3.9	DEFINIR UN VOCABULARIO	53
3.10	IDENTIFICAR LOS PROCESOS ACTUALES	54
4.	FASE MEDIR	62
4.1	CAPTURAR LA VOZ DEL CLIENTE (VOC)	66
4.2	DETERMINAR QUE MEDIR	67
4.2.1	Tipos de variación	68
4.2.2	Medir lo que tenga valor	70
4.2.3	Exactitud de las medidas	70
4.3	PLAN DE COLECCIÓN DE DATOS	71
4.3.1	Sección 1: definir las metas y los objetivos	71
4.3.2	Sección 2: desarrollar las definiciones operativas y los procedimientos.	72

4.4	ELABORAR LAS GRÁFICAS DE LAS MEDIDAS	75
4.4.1	Diagrama de cajas	75
4.4.2	Diagrama de punto	77
4.4.3	Diagrama de torta.	78
4.4.4	Histograma con curva normal	79
4.4.5	Diagrama de Series de Tiempo	79
4.4.6	Diagrama de Corrido	80
4.4.7	Diagrama de dispersión	81
4.4.8	Diagrama de pareto	82
4.5	CALCULAR EL NIVEL SIGMA	83
4.6	BENCHMARKING	87
4.6.1	¿Qué es Benchmarking?	87
4.6.2	¿Por qué Benchmarking en Seis Sigma?	88
4.7	DESARROLLAR EL MODELO FINANCIERO	90
4.8	IDENTIFICAR Y PRIORIZAR LOS REQUERIMIENTOS DE LOS CLIENTES	92
4.9	INICIO DE DETERMINACIÓN DE MEJORAS	95
4.10	PLAN DE COMUNICACIONES	96
5.	FASE ANALIZAR.....	98
5.1	DETERMINAR QUE CAUSA LA VARIACIÓN	99
5.2	LLUVIA DE IDEAS PARA LA MEJORA DE PROCESOS	105
5.3	DETERMINAR CUALES MEJORAS TIENEN MAYOR IMPACTO EN LOS REQUERIMIENTOS DE LOS CLIENTES	107
5.4	ENCONTRAR LOS RIESGOS ASOCIADOS AL PROCESO REVISADO (AMFE)	110
5.5	DESARROLLAR EL PLAN DE CAPACITACIÓN	111
5.5.1	Evaluar las necesidades para la capacitación	111
5.5.2	Preparar un programa de capacitación.	112
5.5.3	Administrar la logística para la capacitación.	113
5.5.4	Evaluar y dar seguimiento al programa:	115

5.6	IDENTIFICAR LOS SISTEMAS AFECTADOS DE OTRAS DEPENDENCIAS	115
5.7	DESARROLLAR EL NUEVO MAPA DE PROCESOS	116
6.	FASE DISEÑAR.....	119
6.1	SELECCIONAR METODOLOGÍA DE DESARROLLO DE SOFTWARE	120
6.1.1	Cascada.	120
6.1.2	Espiral.	121
6.1.3	Proceso Unificado.	123
6.1.4	Modelo de Capacidad de Maduración (CMM).	125
6.1.5	Proceso Personal de Software (PSP).	127
6.1.6	Proceso de Software en Equipo (TSP).	128
6.1.7	Desarrollo Rápido de Aplicación (RAD).	130
6.1.8	Extreme Programing (XP).	131
6.2	DISEÑAR EL FORMATO DE DOCUMENTACIÓN DE PROGRAMAS	134
6.3	DEFINIR LA SEGURIDAD	135
6.4	ESPECIFICACIÓN DEL DISEÑO	137
6.5	INTEGRACIÓN DEL SISTEMA	141
6.5.1	Tipos de integración de sistemas	141
6.5.2	Estándares para conexión a la red	147
6.5.3	Interfaces de conexión a bases de datos.	148
6.6	REVISIÓN Y PERFECCIONAMIENTO DEL DISEÑO	148
6.7	SELECCIÓN DE HARDWARE Y SOFTWARE	151
6.8	CONSTRUCCIÓN DE PROTOTIPOS	154
6.8.1	Técnicas de cuarta generación.	156
6.8.2	Componentes de Software Reutilizable.	156
6.8.3	Especificaciones formales y entornos para prototipos.	156
6.9	DISEÑAR EL PLAN DE PRUEBAS	157
6.9.1	Prueba de Caja Blanca.	159
6.9.2	Prueba de Caja Negra.	159

6.9.3	Prueba de Entornos y Aplicaciones especializadas.	160
6.10	REVISAR QUE LOS PROTOTIPOS CUMPLAN CON LOS REQUERIMIENTOS	161
7.	FASE DE DESARROLLAR	162
7.1	ORGANIZACIÓN DEL CÓDIGO FUENTE	163
7.1.1	Árbol Jerárquico.	163
7.1.2	Enlaces Absolutos.	164
7.1.3	Enlaces Relativos.	165
7.2	CONSTRUCCIÓN	165
7.3	REVISAR QUE SE SATISFACIERON LOS REQUERIMIENTOS	166
7.4	DOCUMENTACIÓN DEL CÓDIGO	166
7.4.1	Legibilidad.	167
7.4.2	Ubicación de comentarios.	167
7.5	DISEÑAR EL PLAN DE IMPLANTACIÓN	168
7.5.1	Corte-Rápido.	169
7.5.2	Etapas.	170
7.6	EJECUTAR EL PLAN DE IMPLANTACIÓN	170
8.	FASE VALIDAR	171
8.1	EJECUTAR EL PLAN DE PRUEBAS.	171
8.2	CAPACITAR AL CLIENTE.	175
8.3	DETERMINAR LOS RIESGOS	177
8.4	DOCUMENTACIÓN DEL PROYECTO	178
8.4.1	Manuales del proyecto.	179
8.4.2	Libro del proyecto.	180
8.5	CREAR LOS MECANISMOS DE ADMINISTRACIÓN.	180
8.6	EVALUAR LOS BENEFICIOS ALCANZADOS	181
8.7	DOCUMENTAR LAS LECCIONES APRENDIDAS	182
8.7.1	Secciones de lecciones aprendidas.	182
8.7.2	Formato de las lecciones aprendidas.	183
8.8	COMUNICAR AL EQUIPO DE TRABAJO LOS RESULTADOS OBTENIDOS	183

8.9	CIERRE DEL PROYECTO	184
-----	---------------------	-----

PARTE II

9.	FASE DEFINIR	185
9.1	ENTENDER EL MODELO DEL NEGOCIO	185
9.2	DEFINIR EL PROBLEMA	187
9.3	DEFINIR ROLES Y RESPONSABILIDADES	188
9.4	FORMAR EL EQUIPO DE TRABAJO	189
9.5	IDENTIFICAR LOS CLIENTES	190
9.6	DETERMINAR LO CRÍTICO A ASEGURAR EN LA CALIDAD (CTQ).	191
9.7	DESARROLLAR EL PLAN DE PROYECTO	194
9.7.1	Alcance y objetivos del proyecto.	194
9.7.2	Costo/Beneficio.	194
9.7.3	Riesgos.	195
9.7.4	Cronograma.	197
9.8	CUADRO RESUMEN DEL PROYECTO	198
9.9	DEFINIR UN VOCABULARIO	199
9.10	IDENTIFICACIÓN DEL PROCESO ACTUAL	199
9.11	PRESENTAR AL COMITE PARA APROBAR	201
10.	FASE MEDIR	202
10.1	CAPTUAR LA VOZ DEL CLIENTE (VOC)	202
10.2	CTQ1	206
10.2.1	Determinar que medir.	206
10.2.2	Plan de Colección de Datos	207
10.2.3	Gráficas.	208
10.2.4	Calcular el nivel de sigma.	212
10.2.5	Benchmarking.	213
10.3	CTQ 2	214
10.3.1	Determinar qué medir.	214
10.3.2	Plan de colección de datos.	215

10.3.3	Elaborar las gráficas.	219
10.3.4	Calcular el nivel de sigma.	220
10.3.5	Benchmarking.	220
10.4	CTQ 3	225
10.4.1	Determinar que medir.	225
10.4.2	Plan de colección de datos.	226
10.4.3	ELABORAR LAS GRÁFICAS	227
10.4.4	Calcular el nivel de sigma.	229
10.4.5	Benchmarking.	230
10.5	DESARROLLAR EL MODELO FINANCIERO	231
10.6	IDENTIFICAR Y PRIORIZAR LOS REQUERIMIENTOS DE LOS CLIENTES	232
10.7	INICIO DE QFD	246
10.8	PLAN DE COMUNICACIONES	247
10.9	PRESENTAR AL COMITÉ PARA APROBAR	248
11.	FASE ANALIZAR.....	249
11.1	DETERMINAR QUE CAUSA LA VARIACIÓN	249
11.1.1	CTQ 1.	249
11.1.2	CTQ 2.	251
11.1.3	CTQ 3	251
11.2	SESIÓN DE LLUVIA DE IDEAS	253
11.3	DETERMINAR QUE SOLUCIONES TENDRAN MAYOR IMPACTO EN LOS REQUERIMIENTOS DEL CLIENTE	255
11.4	ANALISIS DE MODOS DE FALLO Y EFECTOS	258
11.5	DESARROLLAR EL PLAN DE CAPACITACIÓN	261
11.6	IDENTIFICAR LOS SISTEMAS AFECTADOS DE OTRAS DEPENDENCIAS	263
11.7	DESARROLLAR EL NUEVO MAPA DE PROCESOS	264
11.8	PRESENTAR AL COMITÉ PARA APROBAR	266
12.	FASE DISEÑAR.....	267

12.1	SELECCIONAR LA METODOLOGÍA DE DESARROLLO DE SOFTWARE	267
12.1.1	Principios de PSP.	269
12.1.2	Estructura del Proceso.	269
12.2	DISEÑO DE FORMATO DE DOCUMENTACIÓN DE PROGRAMAS	276
12.3	DEFINIR LA SEGURIDAD	278
12.4	ESPECIFICACIÓN DEL DISEÑO	279
12.4.1	Diseño de datos.	279
12.4.2	Diseño arquitectónico.	282
12.4.3	Diseño de interfaz	290
12.4.4	Diseño procedimental.	291
12.5	SELECCIÓN DE HARDWARE Y SOFTWARE	293
12.6	CONSTRUCCIÓN DE PROTOTIPOS	295
12.6.1	Búsqueda simple	295
12.6.2	Búsqueda Avanzada	295
12.6.3	Listado de Resultados	296
12.6.4	Subconsulta	297
12.6.5	Ver Detalles	298
12.6.6	Lista previa de referencias	299
12.6.7	Enviar referencias por correo electrónico	300
12.6.8	Imprimir listado de referencias	301
12.6.9	Detalles MARC	302
12.6.10	Resumen	303
12.7	DISEÑAR EL PLAN DE PRUEBAS	304
12.8	VERIFICAR QUE LOS PROTOTIPOS CUMPLAN CON LOS REQUERIMIENTOS	312
12.9	PRESENTAR AL COMITÉ PARA APROBAR	312
13.	FASE DESARROLLAR	313
13.1	ORGANIZACIÓN DEL CODIGO FUENTE	313
13.2	CONSTRUCCIÓN	314

13.2.1	Búsqueda simple	314
13.2.2	Búsqueda Avanzada	315
13.2.3	Listado de Resultados	316
13.2.4	Subconsulta	317
13.2.5	Ver Detalles	318
13.2.6	Lista previa de referencias	319
13.2.7	Enviar referencias por correo electrónico	322
13.2.8	Imprimir listado de referencias	323
13.2.9	Detalles MARC	325
13.2.10	Resumen	325
14.	CONCLUSIONES.....	327
15.	RECOMENDACIONES	329
	BIBLIOGRAFÍA	336
	BIBLIOGRAFÍA EN INTERNET	338
	ANEXOS	339

LISTADO DE FIGURAS

	pág
Figura 1. Representación de la variación.	14
Figura 2. Ciclo de análisis de riesgos proactivo.	48
Figura 3. Diagrama de GANTT.	52
Figura 4. Elementos utilizados en la representación de procesos.	58
Figura 5. Diagrama de alto nivel.	59
Figura 6. Diagrama de alto nivel con pasos intermedios.	59
Figura 7. Diagrama de nivel de detalle.	60
Figura 8. Probabilidades asociadas a una distribución normal.	65
Figura 9. Diagrama de cajas sencillo.	76
Figura 10. Diagrama de cajas avanzado.	76
Figura 11. Diagrama de punto.	78
Figura 12. Diagrama de torta.	78
Figura 13. Diagrama de histograma.	79
Figura 14. Diagrama de series de tiempo.	80
Figura 15. Diagrama de corrido.	81
Figura 16. Diagrama de dispersión.	82
Figura 17. Diagrama de pareto.	82
Figura 18. Retorno de inversión Vs tiempo.	91
Figura 19. Modelo de Kano.	94
Figura 20. Diagrama de corrido.	99
Figura 21. Ciclo de análisis del problema.	101
Figura 22. Pareto, causas de fallas del sistema.	102
Figura 23. Pareto, Causas de por que falla el código.	102
Figura 24. Diagrama causa – efecto.	104

Figura 25.	Ciclo de capacitación.	112
Figura 26.	Ejemplo diagrama de flujo del nuevo proceso.	118
Figura 27.	Modelo Cascada.	120
Figura 28.	Modelo de Espiral.	122
Figura 29.	Modelo proceso unificado.	124
Figura 30.	Modelo de CMM.	126
Figura 31.	Modelo proceso personal de software.	127
Figura 32.	Modelo Proceso de software en equipo.	129
Figura 33.	Modelo de desarrollo rápido de aplicación.	130
Figura 34.	Modelo Extreme Programming.	132
Figura 35.	Modelo del diseño.	138
Figura 36.	Integración orientada a datos.	142
Figura 37.	Integración orientada a interfaz de aplicación (API).	143
Figura 38.	Integración Orientada a Métodos.	144
Figura 39.	Integración Orientado a Portal.	145
Figura 40.	Integración Orientado a Procesos.	146
Figura 41.	Proceso de prototipos.	157
Figura 42.	Estructura de un árbol jerárquico.	164
Figura 43.	Administración del Riesgo.	178
Figura 44.	Modelo organizacional del proyecto.	190
Figura 47.	Proceso Actual	200
Figura 48.	Resultados de la VOC.	203
Figura 49.	Porcentaje de quejas del cliente.	203
Figura 50.	Arbol final soluciones	204
Figura 51.	Tabla de estadísticas de tiempo.	207
Figura 52.	Situación actual, y posición de captura de tiempo.	208
Figura 53.	Consultas vs. Tiempos de respuesta.	209
Figura 54.	Consultas defectuosas Vs No defectuosas.	209
Figura 55.	Número de consultas Vs hora.	210
Figura 56.	Consutas por hora vs. Consultas defectuosas por hora	210
Figura 57.	Usuarios que ven detalles vs no ven.	211

Figura 58.	Usuario que ven detalles Vs no ven detalles.	211
Figura 59.	Consulta simple sistema informix.	215
Figura 60.	Listado de resultados sistema informix.	216
Figura 61.	Subconsulta sistema informix.	217
Figura 62.	Detalles de un material bibliográfico sistema informix.	218
Figura 63.	Resumen sistema informix.	219
Figura 64.	Comparación de servicios.	219
Figura 65.	Búsqueda simple Benchmarking.	220
Figura 66.	Búsqueda avanzada Benchmarking.	221
Figura 67.	Listado de resultados Benchmarking.	221
Figura 68.	Detalles material bibliográfico Benchmarking.	222
Figura 69.	Detalles MARC Benchmarking.	222
Figura 70.	Reserva de material bibliográfico Benchmarking.	223
Figura 71.	Guardar referencias Benchmarking.	223
Figura 72.	Tablas de captura de datos estadísticos.	226
Figura 73.	Diagrama de Corrido	227
Figura 74.	Resultados de las Consultas.	228
Figura 75.	Gráficas de los tipos de errores.	228
Figura 76.	Caso de uso general.	235
Figura 77.	Diagrama de Casos de Uso	235
Figura 78.	Diagrama de actividades CU consultar.	237
Figura 79.	Diagrama de actividades CU subconsulta.	238
Figura 80.	Diagrama de actividades CU búsqueda simple.	239
Figura 81.	Diagrama de actividades CU búsqueda avanzada.	241
Figura 82.	Diagrama de actividades CU guardar referencia.	243
Figura 83.	Diagrama de actividades CU ver detalles.	245
Figura 84.	Diagrama de actividades CU navegar por los resultados.	246
Figura 85.	Gráfico de corrido CTQ 1.	249
Figura 86.	Tiempo de respuesta con cantidad de consultas por hora.	250
Figura 87.	Pareto causas de la velocidad.	250
Figura 88.	Pareto tipos de errores.	252

Figura 89. Pareto errores tipo 1.	253
Figura 90. Pareto requerimientos del cliente.	257
Figura 91. Nuevo mapa de procesos.	264
Figura 92. Estructura del proceso	270
Figura 93. Estructura del proceso	270
Figura 94. Estructura por niveles de psp.	271
Figura 95. Modelo dimensional biblioteca.	281
Figura 96. Arquitectura de red distribuida.	283
Figura 97. Capa de presentación.	285
Figura 99. Capa de datos.	289
Figura 100. Prototipo búsqueda simple.	295
Figura 101. Prototipo búsqueda avanzada.	296
Figura 102. Prototipo listado de resultados.	297
Figura 103. Prototipo subconsulta.	298
Figura 104. Prototipo ver detalles.	299
Figura 105. Prototipo lista previa de referencias.	300
Figura 106. Prototipo enviar referencia por e-mail.	301
Figura 107. Prototipo confirmación envío de e-mail.	301
Figura 108. Prototipo imprimir referencias.	302
Figura 109. Prototipo ventana de impresión.	302
Figura 110. Prototipo detalles MARC.	303
Figura 111. Prototipo resumen.	304
Figura 112. Árbol de organización del código fuente.	313
Figura 113. Árbol de organización de las librerías.	313
Figura 114. Construcción, Búsqueda simple.	314
Figura 115. Construcción, Búsqueda avanzada.	315
Figura 116. Construcción, Listado de resultados.	317
Figura 117. Construcción, Subconsulta.	318
Figura 118. Construcción, Ver detalles.	319
Figura 119. Construcción, Listado de referencias.	320
Figura 120. Construcción, Modificación de lista previa de referencias.	321

Figura 121. Construcción, Lista modificada de referencias.	321
Figura 122. Construcción, Envío de referencias por e-mail.	322
Figura 123. Construcción, Confirmación envío de referencias por e-mail.	323
Figura 124. Construcción, Imprimir referencias.	324
Figura 125. Construcción, Ventana de impresión.	324
Figura 126. Construcción, Detalles MARC.	325
Figura 127. Construcción, Resumen.	326

LISTADO DE TABLAS

	Pág
Tabla 1. Comparaciones de los niveles de Seis sigma	11
Tabla 2. Seis Sigma Vs Control de Calidad	13
Tabla 3. Estructura de la definición del problema.	25
Tabla 4. Métodos para clasificar a los clientes.	33
Tabla 5. Tipos de riesgos.	47
Tabla 6. Criterios de selección de medidas.	68
Tabla 7. Método de 1 P y 5 M's.	69
Tabla 8. Categoría de los riesgos en las medidas.	70
Tabla 9. Datos ejemplo.	77
Tabla 10. Conversión de DPMO a Nivel Sigma	86
Tabla 11. Categorías del diagrama causa-efecto.	104
Tabla 12. Costo/Tiempo/Impacto.	107
Tabla 13. Ranking de propuestas/requerimientos.	108
Tabla 14. Ejemplo QFD.	109
Tabla 15. Ejemplo de un plan de trabajo general.	113
Tabla 16. Ejemplo de un plan de sesión.	114
Tabla 17. Comparación metodologías de desarrollo de software	133
Tabla 18. Tipos de Pruebas.	173
Tabla 19. Registro de Pruebas	174
Tabla 20. Formas de Capacitación.	176
Tabla 21. Encuesta de Capacitación.	176
Tabla 22. Roles y responsabilidades.	189
Tabla 23. Identificación de clientes.	191
Tabla 24. Riesgos definidos en el plan.	195
Tabla 25. Principales características que debe tener el sistema.	233
Tabla 26. Listado de Requisitos Funcionales.	234

Tabla 27. CU consultar.	236
Tabla 28. CU subconsulta.	237
Tabla 29. CU búsqueda simple.	238
Tabla 30. CU búsqueda avanzada.	239
Tabla 31. CU guardar referencia.	241
Tabla 32. CU ver detalles	244
Tabla 33. CU navegar por los resultados.	245
Tabla 34. Plan de comunicaciones establecido.	247
Tabla 35. Sesión de lluvia de ideas.	253
Tabla 36. Puntaje de criterio para la selección de ideas.	255
Tabla 37. Análisis de Modos de Fallo y Efectos	259
Tabla 37. Análisis de Modos de Fallo y Efectos (Continuación)	260
Tabla 38. Formato uno de documentación de programas.	276
Tabla 39. Formato dos de documentación de programas.	277
Tabla 40. Formato tres de documentación de programas.	277
Tabla 41. Asignación de seguridad.	278
Tabla 42. Asignación de roles.	278
Tabla 43. Diferencias entre modelos dimensionales y E-R.	279
Tabla 44. Tabla de verdad para intersección.	293
Tabla 45. Tabla de verdad unión.	293
Tabla 46. Pruebas de Caja Blanca.	305
Tabla 49. Cuadro resumen proyecto.	337
Tabla 50. Métodos para capturar la VOC.	338
Tabla 51. Análisis de Modos de Fallo y Efectos	350
Tabla 52. Descripción de las fases de programación.	353
Tabla 53. Estándares de tipos de defectos.	356

LISTA DE ANEXOS

	Pág
ANEXO A. Cuadro resumen de proyecto	334
ANEXO B. Técnicas para capturar la voz del cliente.	338
ANEXO C. Captura de Requisitos con UML.	340
ANEXO D. Análisis de modos de fallo y efecto.	347
ANEXO E. Formatos utilizados en psp.	351

1. **TÍTULO:** SEIS SIGMA EN EL DESARROLLO DE PROYECTOS DE INFORMACIÓN TECNOLÓGICA DE LA DIVISIÓN DE SERVICIOS DE INFORMACIÓN Y SU APLICACIÓN EN UN MÓDULO DE LA INTERFACE WEB DEL SISTEMA DE LA BIBLIOTECA DE LA UIS*

2. **AUTORES:** ZAMBRANO OLARTE DIANA MARCELA, CONTRERAS NIGRINIS SERGIO ANDRES.**

3. **PALABRAS CLAVES:** Seis Sigma, Biblioteca, Crítico para la Calidad, Módulo de consulta, Catálogo Bibliográfico, Satisfacción del cliente, defectos, Project managment, Benchmarking.

4. DESCRIPCIÓN:

Durante mucho tiempo la División Servicios de Información (DSI) de la Universidad Industrial de Santander (UIS) ha construido una gran variedad de proyectos de información tecnológica que han sido elaborados utilizando diferentes metodologías de desarrollo de software, las cuales han logrado que los sistemas de información cumplan con las necesidades del cliente y de cierto modo alcanzando un alto nivel de satisfacción.

El presente proyecto se realizó con el propósito de integrar en un solo marco metodológico, un estándar en el que se tenga en cuenta una serie de fases y pasos para el desarrollo formal y eficiente de procesos y especialmente de software.

Así como Seis Sigma ayuda a las compañías a eliminar defectos, reducir costos y mejorar la satisfacción del cliente en sus procesos de manufactura, igualmente puede aumentar la efectividad en los tradicionales procesos de desarrollo de software.

El estudio se centrará en un modelo definido como DMADDV utilizado para el desarrollo de nuevos proyectos software. Además la definición y descripción de los pasos que el equipo de trabajo considere deban conformar cada una de las fases del modelo.

* Proyecto de Grado

** Facultad de Ciencias Físico-Mecánicas, Ingeniería de Sistemas, Dir. Ing. Enrique Torres López

1. **TITLE:** SIX SIGMA IN TECHNOLOGICAL INFORMATION PROJECT DEVELOPMENT OF THE INFORMATION SERVICE DIVISION AND ITS APPLICATION IN A MODULE OF THE WEB INTERFACE OF THE LIBRARY SYSTEM OF THE UIS.*
2. **AUTHORS:** Zambrano Olarte Diana Marcela, Contreras Nigrinis Sergio Andres.**
3. **KEY WORDS:** Library, Critical for Quality, Consultation Module, Bibliographic catalog, Customer Satisfaction, defects, project management, Six Sigma, Benchmarking

4. DESCRIPTION:

During a long time the Information Service Division (ISD) of the Industrial University of Santander (UIS) has built a variety of technological information projects that have been developed using different methodologies of software development, which have achieved that the information systems reach the client's needs and in some way obtain a high level of satisfaction.

The present project was done with the purpose of integrating in one single methodological frame, a standard in which is taken into account a series of phases and steps for the efficient and formal development of processes and specially software.

As well as Six Sigma helps companies eliminate defects, reduce costs and improve customer satisfaction in its manufactory processes, in this same way it can increase the effectiveness in traditional processes of software development.

The study will be centered in a defined model as DMADDV used for the development of new software. Also the definition and description of the steps that the work team considers that should conform each of the phases of the model.

** Physical-Mechanic Science Faculty, Systems Engineering, Dir. Eng. Enrique Torres López

GLOSARIO

ALCANCE: Define los límites del proceso o del proyecto de mejora; señala específicamente donde residen las oportunidades de mejora (puntos inicial y final); define dónde y qué hay que medir y analizar; tiene que estar dentro de la esfera de influencia y control del equipo que trabaja en el proyecto y, cuanto más amplio sea el alcance, más complejo y más tiempo requerirá el esfuerzo de mejora del proceso.

AMFE (Análisis de Modos de Fallo y Efectos): Es un proceso disciplinado que le permite anticiparse a los fallos, identificarlos y preverlos. El modo en que parte del proceso puede incumplir las especificaciones, creando un defecto y el impacto en el cliente si ese modo de fallo no se prevé o se corrige.

ASEGURAMIENTO DE LA CALIDAD O QA (*Quality Assurance*): Disciplina para mantenimiento de la conformidad de productos o servicios respecto a las especificaciones del cliente; las herramientas principales son inspección y control estadístico de procesos.

BB: (*Black Belt*): Líderes del equipo y jefes del proyecto, con habilidades de facilitador, responsables de dirigir un proyecto hasta la conclusión del mismo.

BENCHMARKING: Es un método para comparar procesos, utilizando estándares o las mejores prácticas como referencia, para luego identificar maneras de mejorar el proceso.

CHAMPION: Persona que representa a un equipo ante la alta dirección; da el aprobado final a las recomendaciones del equipo, y apoya su trabajo ante el consejo de calidad; facilita la obtención de recursos para el equipo según sean necesarios; ayuda al Black Belt y al equipo a superar los obstáculos; y es el mentor para el Black Belt.

CLIENTE: Cualquier persona y organización, interna o externa, que recibe el resultado (producto o servicio) del proceso; comprender el impacto del proceso tanto en los clientes internos como externos es fundamental para la gestión y mejora de los procesos.

CMM (Capability Maturity Model): Modelo de Madurez de Capacidad para el Software. Es un modelo para juzgar la madurez de los procesos de desarrollo y mantenimiento de software en una organización y para identificar las prácticas claves que se requieren para aumentar la madurez de estos procesos.

COSTO POR BAJA CALIDAD O COPQ (Cost Of Poor Quality): Costo total del personal, material y gastos generales atribuidos a las imperfecciones en los procesos que entregan productos o servicios que no cumplen con las especificaciones o expectativas. Estos costos incluirían la inspección, reproceso, trabajo duplicado, rechazos de material, repuestos y devoluciones, quejas, pérdida de clientes y el daño a la imagen.

CRÍTICOS PARA LA CALIDAD O CTQ (Critical To Quality): Elementos de un proceso que afectan significativamente la salida de ese proceso. Es vital identificar esos elementos para entender cómo hacer mejoras que reduzcan dramáticamente los costos y mejoren la calidad.

DEFECTO: Una característica medible del proceso o de su salida que no está dentro de los límites aceptables para el cliente, es decir, que no está conforme a las especificaciones.

DMADV: Es una estrategia de calidad para diseñar productos y procesos, es una parte integral de la iniciativa de la calidad Seis Sigma. Consiste en cinco fases: Definir, Medir, Analizar, Diseñar, Verificar.

DMAIC: Acrónimo del sistema de gestión y mejora de procesos que comprende las fases Definir, Medir, Analizar, Mejorar y Controlar.

DPMO o Defectos por millón de oportunidades: Cálculo utilizado en las iniciativas de mejora de procesos Seis Sigma que indica la cantidad de defectos de un proceso por millón de oportunidades; el número de defectos dividido por (el número de unidades por el número de oportunidades), por un millón = DPMO.

$$\text{DPMO} = \frac{\text{NUMERO DE DEFECTOS} * 1'000.000}{\text{NUMERO DE UNIDADES} * \text{NUMERO DE OPORTUNIDADES}}$$

FMEA (Failure Mode and Effect Analysis): Análisis de Modos de Fallo y Efectos (AMFE)

GB (Green Belt): Es un miembro de la compañía, que posee altas habilidades en una área específica en que desarrolla el proceso. Su participación en el proyecto puede ser parcial o de tiempo completo. Desarrollan destrezas en las herramientas estadísticas.

GESTIÓN DE LA CALIDAD TOTAL O TQM (Total Quality Management): Un enfoque de la gestión que se centra en la organización como un sistema, con énfasis en los equipos, los procesos, la estadística, la mejora continua y la entrega de productos y servicios que satisfacen y exceden las expectativas de los clientes.

MBB (Master Black Belt): Son los expertos en la metodología, y los responsables de las implantaciones estratégicas en la organización. Son los encargados del entrenamiento y los mentores de los Black Belts y Green Belt; ayudan a priorizar y a calcular el impacto de los proyectos.

REQUISITOS DE CLIENTE: Define las necesidades y expectativas del cliente, traducidas a términos medibles y utilizados en el proceso para garantizar la compatibilidad con las necesidades del cliente.

SIGMA: Letra del alfabeto griego utilizada en estadística para representar la desviación estándar de una población, un indicador del grado de variación en un conjunto de medidas o en un proceso.

SEIS SIGMA: Un concepto estadístico que mide un proceso en términos de defectos, en un nivel Seis Sigma sólo existen 3,4 defectos por millón de oportunidades. Seis Sigma es también una filosofía de gestión que enfoca

su atención en eliminar los defectos a través de prácticas que enfatizan la comprensión, la medida y la mejora de los procesos.

VOZ DEL CLIENTE O VOC (Voice of Customer): Datos (quejas, cuestionarios, comentarios, investigaciones de mercado, etc) que representan la perspectiva/necesidades de los clientes de la empresa; debe traducirse a requisitos medibles para el proceso.

X: Variable empleada para indicar factores o medidas en la entrada de un proceso o sistema empresarial.

Y: Variable empleada para indicar factores de medidas en los resultados de un proceso o sistema empresarial. Equivale a los resultados.

INTRODUCCIÓN

Cambiar la forma de hacer las cosas en dirección de encontrar la demanda de los clientes externos o internos, e incrementar los beneficios, ha venido siendo uno de los asuntos con mayor prioridad en las organizaciones hoy en día. En el pasado, el cambio en las organizaciones se daba realmente lento. Algunas compañías podrían sobrevivir o simplemente prosperar en el entorno y conservar los procesos estables. En nuestros días, sin embargo, la competencia en el negocio es mas fuerte cada día, convirtiéndose en un peligro continuar haciendo las cosas de la forma como se han realizado en el pasado.

Como resultado, la necesidad por el cambio a largo y corto plazo se incrementa más, donde muchas organizaciones sienten la necesidad por mejorar sus operaciones de cualquier forma en que se encuentren posibilitados en hacer cada día las cosas mucho mejor.

Si el cambio ha sido bien guiado y planeado, pueden llegar a resultar nuevas oportunidades, crecimientos, y un incremento en los beneficios para la organización. Compañías como General Electric, Motorola, y Allied Signal han hecho extraordinarias ganancias por el cambio cultural de los trabajadores gracias a la iniciativa de Seis Sigma.

Durante mucho tiempo la División Servicios de Información (DSI) de la Universidad Industrial de Santander (UIS) ha construido una gran variedad de proyectos de información tecnológica que han sido elaborados utilizando diferentes metodologías de desarrollo de software, las cuales han logrado que los sistemas de información cumplan con las necesidades del cliente y de cierto modo alcanzando un alto nivel de satisfacción.

Este proyecto se justifica debido a la necesidad de integrar en un solo marco metodológico, un estándar en el que se tenga en cuenta una serie de fases y pasos, para el desarrollo formal y eficiente de procesos, y especialmente de software. Así mismo permitirá crear una nueva alternativa en el desarrollo de proyectos de información tecnológica, que sea propio de la DSI, en donde se mejore la calidad del producto final, basado en las necesidades, satisfacción del cliente y en la eliminación de defectos.

El desarrollo de este proyecto traerá beneficios para toda la comunidad universitaria, y podrá ser asumido por el personal involucrado en la DSI dado que es un modelo que busca resultados finales satisfactorios tanto para los diferentes tipos de clientes como para los directores del proyecto.

El proyecto nace de la necesidad de buscar una solución a los problemas que se presentan en el desarrollo de proyectos de información tecnológica, como son: productos entregados en lapsos de tiempo mas largos de los esperados por los directores del proyecto y los clientes, el nivel de calidad de los servicios que presta el proyecto a los clientes finales, la no satisfacción total del cliente con el producto utilizado, entre otros. Esto conlleva a un mayor gasto de los recursos destinados al desarrollo del proyecto.

El estudio de la metodología “Seis Sigma” nos ofrece una nueva alternativa para desarrollar proyectos de información tecnológica, basados principalmente en la satisfacción total del cliente final y la mejora en los procesos y productos.

1. PRESENTACIÓN

1.1 DESCRIPCIÓN DEL DOCUMENTO

Esta sección servirá de guía al lector al momento de querer estudiar el proyecto paso a paso y a la vez facilita el entendimiento y comprensión del mismo.

La intención del documento es ser comprensible a cualquier tipo de lector, sin importar su grado de preparación en el tema, por lo cual, se muestra el proceso que se llevó a cabo en el desarrollo del proyecto dejando ver la fundamentación teórica utilizada como referencia.

Este libro está dividido en dos partes principalmente. La parte I contiene los capítulos del 1 al 8, donde se encuentra la metodología Seis Sigma en general, con todas las fases del modelo y cada uno de los pasos de cada fase. La parte II que contiene los capítulos del 9 al 15, que contiene los resultados que se hicieron del modelo en el proyecto piloto de consulta del material bibliográfico de la Biblioteca de la Universidad vía web, además de las conclusiones, recomendaciones, referencias bibliográficas y anexos.

En el capítulo 1 se presentan los antecedentes para la realización del proyecto, el objetivo general y los objetivos específicos descritos en el plan del proyecto.

El capítulo 2 presenta las generalidades de Seis Sigma y los principios esenciales.

El capítulo 3 describe la primera fase del modelo, la fase Definir, con todos los pasos, donde se presenta la descripción del problema y las necesidades del cliente.

El capítulo 4 describe la fase Medir del modelo y los pasos, donde se cuantifica el problema.

El capítulo 5 describe la fase Analizar del modelo y los pasos, donde se analizan y encuentran las mejores soluciones que van a resolver el problema.

El capítulo 6 describe la fase Diseñar y los pasos, que guiarán el diseño de las soluciones que se han considerado como las más óptimas.

El capítulo 7 describe la fase Desarrollar y todos los pasos, que guiarán el desarrollo de las soluciones.

El capítulo 8 describe la fase Validar y todos los pasos, que ayudan a verificar que el proyecto cumplió con los objetivos planteados.

El capítulo 9 describe los resultados de los pasos de la fase Definir del proyecto piloto de la biblioteca.

El capítulo 10 describe los resultados de los pasos de la fase Medir en el proyecto piloto.

El capítulo 11 describe los resultados de los pasos de la fase Analizar en el proyecto piloto.

El capítulo 12 describe los resultados de los pasos de la fase Diseñar en el proyecto piloto.

El capítulo 13, contiene los resultados de los pasos de la fase Desarrollar del proyecto piloto, hasta el paso de construcción, como se indicó en el objetivo específico.

El capítulo 14 y 15 contienen las conclusiones y recomendaciones para tener en cuenta en la continuidad del proyecto.

Finalmente se encuentra las referencias bibliográficas y los anexos que complementan al documento.

1.2 ANTECEDENTES

1.2.1 Metodología Seis Sigma.

Seis Sigma se desarrolló en Motorola a finales de los ochenta como un modo de proporcionar una orientación clara a la mejora de la dirección y el rendimiento empresarial. El concepto, las herramientas y el sistema Seis Sigma han evolucionado y se han expandido a lo largo de los años por empresas como GE y AlliedSignal / Honeywell, lo que ayudó a mantener un renovado interés y a redoblar los esfuerzos en la mejora de la calidad y de los procesos.

Seis Sigma mide y refleja estadísticamente la capacidad real de los procesos, correlacionándolos con características como los defectos por unidad y la probabilidad de éxito o de fallo. Su valor radica en transformar

una visión cultural basada en la complacencia, a una basada en el logro en toda una gama de sectores industriales.

Muchas compañías funcionan a un nivel de cuatro sigma, tolerando 6.210 defectos por un millón de oportunidades. El operar a un nivel Seis Sigma implica un ambiente prácticamente libre de defectos, permitiendo solo 3,4 defectos por millón de oportunidades: los productos y servicios son prácticamente perfectos (99.99936%) eliminando los defectos se eliminan las insatisfacciones.

Seis Sigma ha ayudado a Motorola a realizar grandes cambios en la línea baja de su organización. De hecho, ellos argumentan que los esfuerzos realizados de Seis Sigma en la compañía han significado un ahorro en costos de 16 billones de dólares. Desde entonces, cientos de compañías alrededor del mundo han adoptado Seis Sigma como la manera de hacer negocios. Los resultados hablan por sí solos

Por citar, compañías como Motorola, Allied Signal, General Electric, Honeywell y Ford, han reducido en costos mas de 9,8 Billones de dólares en promedio cada una en un periodo de 5 años.

La metodología Seis Sigma trabaja para reducir costos e incrementar la satisfacción del cliente, mediante la mejora de procesos de producción. Si observamos al departamento de información tecnológica como un departamento donde se produce, entonces podemos aplicar Seis Sigma, ya que poseemos una fábrica donde producimos sistemas de información, centros de datos, servidores, redes, clientes externos e internos.

1.2.2 Módulo de consultas del catálogo bibliográfico de la Biblioteca vía web.

La biblioteca de la Universidad Industrial de Santander UIS, actualmente no tiene un sitio Web, estructurado de primer nivel, donde los usuarios puedan acceder a todos sus servicios y a información general de la misma. En la actualidad la biblioteca cuenta con un espacio dentro del Web site institucional, pero esto ha generado ciertos inconvenientes, descritos a continuación:

- La estructura de menús del Web site institucional, esta diseñada de tal forma que la publicación de la información de la biblioteca está ubicada en una página de tercer nivel, sin ningún acceso directo desde la página principal. Lo anterior conduce a que la información no sea fácilmente accesible por el usuario.
- La información de la biblioteca está publicada de manera lineal, convirtiéndola en un sitio de difícil navegación, haciendo uso obligatorio de extensas barras de desplazamiento; además casi toda la información publicada es estática, con pocos accesos a servicios dinámicos.
- Debido a que la página es administrada por otra sección de la Universidad (DSI), todos los cambios que se deseen realizar a la información a publicar, deben ser autorizados por esta sección. Dicho procedimiento conlleva un trámite jerárquico haciendo que la información pierda vigencia.
- La biblioteca cuenta con una variedad de servicios que pueden implementarse de manera dinámica en un sitio Web, hoy en día se hacen altas inversiones en obtener información de calidad, pero no se han desarrollado los medios apropiados para ofrecer todos estos bancos de información al servicio de los usuarios.

- Actualmente existe una versión en Web del catálogo bibliográfico en línea, pero dicho desarrollo necesita una evaluación de sus procesos y decidir mejorar o diseñar una nueva aplicación con un nuevo enfoque.
- Existe alrededor de 30 Bancos Bibliográficos (Bases de Datos), en distintos formatos (CD-ROM en Internet), el problema que se presenta es que estos bancos están empaquetados independientemente, dificultando a los usuarios el acceso a ellos, esto conlleva a que los recursos no están siendo suficientemente aprovechados.
- Los avances tecnológicos hacen necesaria la construcción de un Web site propio, que le permita a la biblioteca estar a la vanguardia de la información y ofrecer a sus usuarios servicios ágiles, de acuerdo a las necesidades virtuales que el mundo actual exige.

1.3 OBJETIVOS

1.3.1 Objetivo General.

Analizar y definir los pasos de la metodología "Seis Sigma" para que sea otra alternativa de modelo de desarrollo de proyectos de información tecnológica de la División de Servicios de Información, y aplicar esta metodología en el desarrollo de un módulo de la nueva interface Web del sistema de información de la Biblioteca de la UIS, LIBRUIS.

1.3.2 Objetivos Específicos.

- Presentar una visión general de la metodología Seis Sigma.
- Definir y describir los pasos clave en nuestro modelo de desarrollo DMADDV para que sirva como alternativa guía en el desarrollo de los

proyectos de información tecnológica de la División de Servicios de Información de la UIS.

- Aplicar la metodología descrita en el modelo DMADDV a través de la elaboración de un módulo del proyecto: “Interfase Web del sistema de información de la biblioteca de la UIS”, desarrollando los pasos que sean aplicables al proyecto de las fases: Definir, Medir, Analizar, Diseñar y Desarrollar hasta finalizar el paso de construcción.
- Capacitar al personal de la División de Servicios de Información, en los métodos, herramientas y filosofía que utiliza la metodología Seis Sigma.

2. GENERALIDADES DE SEIS SIGMA

2.1 HISTORIA Y EVOLUCIÓN

La historia de Seis Sigma empieza en la década de los años ochenta en Motorola, donde primero fue desarrollado y probado. En 1983, el ingeniero Bill Smith concluyó que, si un producto era defectuoso y se corregía durante la producción, entonces otros defectos probablemente se estaban pasando por alto y posteriormente sería detectados por los clientes. En otras palabras, los índices de fallos en el proceso eran muy superiores a los indicados por los controles finales de productos. ¿Cuál era su argumento? Si los productos se montaban completamente libres de defectos, entonces probablemente no le fallarían más tarde a los clientes.

Este fue el punto de partida de Seis Sigma. El doctor Mikel Harry, fundador del instituto de investigación de Seis Sigma de Motorola, posteriormente pulió la metodología, para no solo eliminar las pérdidas en los procesos, sino también convertirla en moneda de crecimiento, sin importar el tipo específico de servicio, producto o sector del mercado.

2.2 DEFINICIÓN DE SEIS SIGMA

Seis Sigma se puede definir como una filosofía de gestión que enfoca sus esfuerzos en prevenir defectos, reducir la variación y aumentar la satisfacción del cliente. Seis sigma hace referencia a un objetivo de rendimiento que

mide un proceso en términos de defectos, en el cual un nivel Seis sigma implica tener 3,4 defectos por millón de oportunidades.

Las estadísticas en la metodología son pieza fundamental, ya que su objetivo principal es de reducir la variación y prevenir los defectos, además se han convertido en una nueva filosofía de administración que incluye: decisiones basadas en hechos, focalización en el cliente y trabajo en equipo.

Mientras otras iniciativas de producción y de calidad, que prometen grandes beneficios, frecuentemente fallan en las entregas del producto final, Seis Sigma si las entrega. Priorizar los procesos, significa menos horas de trabajo que da como resultado menos costos, mejorando así la satisfacción del cliente y creando un efecto positivo en los niveles más bajos del esquema organizacional.

Tabla 1. Comparaciones de los niveles de Seis sigma¹

Comparaciones de los niveles de Seis Sigma		
<i>Nivel Sigma</i>	<i>Porcentaje Correcto</i>	<i>Numero de defectos por un millón de oportunidades</i>
1	30.9	691.462
2	62.2	308.537
3	93.3193	66.807
4	99.3790	6.210
5	99.9767	233
6	99.99966	3.4

Estadísticamente Seis Sigma basa sus indicadores en la Tabla 1, donde un nivel 4 de Seis Sigma corresponde a mas del 99% de exactitud. En el estudio que se realiza, se tiene como expectativa llegar a un nivel de 3 sigma,

¹ Fuente: Libro “Six Sigma Software Development” por Tayntor Christine. Editorial AUERBACH, Cáp 1, Pág. 4.

considerando un éxito total con este nivel alcanzado, ya que empresas que han iniciado con esta nueva filosofía demoran dos o tres años en llegar a estos niveles de Sigma.

Aplicar esta metodología en la Universidad va mas allá de medir y eliminar defectos, implica una forma de hacer negocios, de tal forma que Seis sigma se incorpore en nuestra cultura organizacional y llegue a ser una de las cosas que nos haga diferentes en nuestro medio competitivo.

2.3 PRINCIPIOS DE SEIS SIGMA

Una organización Seis Sigma incluye los siguientes principios:

1. Prevenir Defectos
2. Reducir la Variación
3. Orientación en el cliente
4. Toma de decisiones basadas en hechos
5. Trabajo en equipo

2.3.1 Prevenir defectos.

Seis Sigma difiere de las demás metodologías de calidad total gracias a la insistencia de eliminar los defectos por medio de la prevención más que la corrección de estos.

Clásicos mecanismos de control de calidad se basan en inspeccionar productos para encontrar defectos y luego corregirlos. Seis sigma analiza los procesos para determinar qué causa estos defectos y luego los mejora con el fin de prevenir los errores. En la tabla 2 se muestra este efecto.

Tabla 2. Seis Sigma Vs Control de Calidad

	Acción a tomar	Acción en	El efecto es	Impacto en	Repetición
Control de Calidad	Inspeccionar	Producto	Corrección del Error	Un producto	Constata
Seis Sigma	Analizar	Proceso	Prevención del defecto	Todos los productos	Nunca

El objetivo de Seis Sigma es identificar los defectos antes que el producto salga de fábrica y sea entregado al cliente; el costo por baja calidad (COPQ²) puede incurrir en diferentes elementos como son:

- Pérdida: el producto no puede ser usado.
- Repetición de trabajo: esfuerzo requerido para corregir el error
- Inspección: costo de realizar rendimiento de control de calidad
- Reportes: esfuerzo involucrado en desarrollar y monitorear reportes.

Mientras muchas compañías se enfocan en la pérdida y la repetición, el costo por inspección y reportes puede superar a la pérdida o la repetición, particularmente en el servicio. En muchos casos, los empleados invierten gran parte de tiempo desarrollando métodos informales para corregir los defectos antes de que la auditoría del control de calidad los reporte.

2.3.2 Reducir la Variación.

El segundo principio de Seis Sigma es reducir la variación, en otras palabras aumentar la consistencia. Esta es una forma de prevenir defectos. La consistencia es importante por que puede ser predecible, y aquello que es predecible puede perfeccionarse.

² COPQ: Cost Of Poor Quality. Costos atribuidos a las imperfecciones en los procesos o servicios que se entregan.

Como se muestra en la figura 1, el jugador de la izquierda aparenta ser el mejor, porque apuntó al blanco en una ocasión, una empresa Seis sigma prefiere la consistencia del jugador de la derecha, aunque éste nunca golpeó en el centro, la puntería de este jugador es consistente.

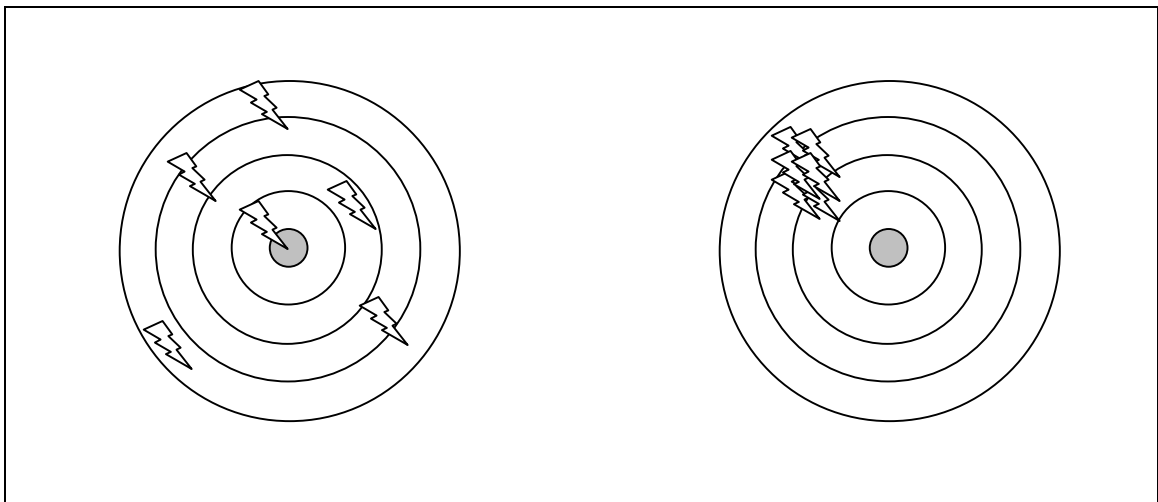


Figura 1. Representación de la variación.

2.3.3 Orientación en el Cliente.

El cliente es una pieza fundamental en el desarrollo de Seis Sigma, sean clientes externos (aquellos que compran los productos o servicios que la compañía vende), o clientes internos (que son los que usan los servicios que otro departamento les provee), estos son el punto central en todas las actividades. Compañías Seis Sigma saben que clientes satisfechos y aumento en las ganancias no son mutuamente excluyentes, por el contrario, aumentar las ganancias es un resultado directo en la fuerte orientación en el cliente. Los clientes quieren y desean productos y servicios perfectos.

Proyectos Seis Sigma comienzan con escuchar la voz de los clientes, lo que significa, entender sus necesidades y cuales son los requerimientos más importantes para él. La presencia del cliente no termina cuando los requerimientos son identificados y se da inicio al proyecto, durante todo el ciclo de vida del proyecto existe una constante comunicación entre las dos partes, y es mucho más que la entrega de reportes periódicamente, de esta forma se asegura que el cliente sea una parte del proceso y que los resultados garantizaran los requerimientos identificados.

2.3.4 Decisiones basadas en hechos.

Uno de los puntos que resaltan más a Seis Sigma de las otras metodologías, es la insistencia en que las decisiones sean basadas en hechos concretos.

Es importante entender exactamente como es que opera el proceso antes de realizar cualquier cambio. Suena algo básico y lo es; pero muchas veces en la prisa de mostrar el progreso en la solución del problema, las compañías fallan en entender exactamente qué es lo que están cambiando antes de empezar a implementar esta modificación. El resultado de esta falla pueden ser inesperados efectos secundarios o dicho de otra forma “defectos”.

Las compañías que utilizan Seis Sigma, saben que entender qué es lo que realmente el cliente quiere es fundamental. No simplemente la percepción del equipo de trabajo sobre las necesidades del cliente. Un producto o servicio defectuoso es algo que el cliente no utilizará o comprará.

La razón por la cual las decisiones deben ser basadas en hechos es simple; teniendo todos los hechos antes de realizar cualquier cambio la compañía puede eliminar la repetición del trabajo y la pérdida causada por soluciones a problemas incorrectos.

2.3.5 Trabajo en Equipo.

El trabajo en equipo es una de las características de compañías Seis Sigma. Para determinar la causa de la variación y poder desarrollar métodos para prevenir los defectos, se requiere la gente correcta trabajando unida para resolver el problema, compartiendo su conocimiento y experiencia.

Los miembros del equipo son seleccionados y valorados según su conocimiento y experiencia, no por su posición en el diagrama organizacional.

2.4 HERRAMIENTAS Y ENTRENAMIENTO

Uno de los principales objetivos de Seis Sigma es reducir la variación, para esto los proyectos de Seis Sigma usan procesos y herramientas estándar, mas allá de que cada equipo de trabajo desarrolle sus propios métodos y técnicas para solucionar los problemas. Esto no solo incrementaría la consistencia si no que reduciría el tiempo y los costos de los equipos en reinventar los procesos.

Seis Sigma reconoce que el entrenamiento es necesario para que los empleados entiendan la filosofía y la mejor forma de implementarla, para esto existen cinco niveles de programas de entrenamiento formal:

- *Green Belt*: Constan de más de dos semanas, y proveen a los participantes, conocimiento básico en los conceptos y herramientas.
- *Black Belt*: Es un entrenamiento más profundo, cerca de cuatro semanas, con más énfasis en herramientas de análisis estadístico.

- *Master Black Belt*: reciben un entrenamiento más especializado en herramientas estadísticas, y son los encargados de guiar y servir de mentores³ para los Black Belts en sus proyectos.
- *Champions*: son los encargados de que los recursos se encuentren disponibles para su uso, y se encuentran involucrados en las revisiones del proyecto.
- *Líderes Ejecutivos*: son los encargados de comprometerse con Seis Sigma y difundirla por toda la organización.

Los entrenamientos se van dando a medida que se va desarrollando el proyecto, la razón está en que es más efectivo estar aplicando en el mundo real lo que se está estudiando en teoría.

Seis Sigma se describe como una estrategia más que como una iniciativa, y aunque suenan similares, no lo son. Iniciativa se define como “el paso introductorio” mientras que estrategia es “un cuidadoso plan o método”, una diferencia realmente importante. Con énfasis en el análisis y en las decisiones basadas en hechos, se proveerá un método para el mejoramiento de nuestra organización.

Para realizar esto, tres cosas deben estar correctas. La gente correcta debe trabajar en el problema correcto, en la dirección correcta.

Seis sigma, basó sus fundamentos principalmente en el ámbito organizacional y de procesos, antes de ser utilizado para el desarrollo de software. El DMAIC⁴ se convirtió en la metodología utilizada por las compañías para aplicarlos tanto a la mejora como al diseño y rediseño de procesos.

³ Mentor: Consejero o guía

⁴ DMAIC: Acrónimo de Define, Measure, Analyze, Improve and Control.

Las fases que componen este modelo son las siguientes:

- *Definir* los proyectos, los objetivos y los entregables a los clientes.
- *Medir* el rendimiento actual del proceso
- *Analizar* y determinar lo que está mal, las causas raíces de los defectos y las posibles soluciones
- *Mejorar* los procesos para eliminar los defectos mediante la implementación de las soluciones
- *Controlar* el rendimiento del proceso mejorado asegurando que los cambios se mantengan

Como lo mencionamos anteriormente, este modelo aplica únicamente al mejoramiento de los procesos. En el área de desarrollo, Seis Sigma utiliza un modelo también de cinco fases, conocido como DMADV⁵, que específicamente se utiliza cuando un producto o un proceso no existe en la organización y debe ser desarrollado.

Estas cinco fases son las que a continuación se mencionan:

- *Definir* los proyectos, los objetivos y los entregables a los clientes.
- *Medir* y determinar las necesidades y especificaciones del cliente
- *Analizar* las opciones de los procesos para determinar las necesidades del cliente
- *Diseñar* detalladamente los procesos para satisfacer las necesidades del cliente
- *Verificar* el rendimiento del diseño y si se cumplieron los requerimientos del cliente.

⁵ DMADV: Acrónimo de Define, Measure, Analyze, Design and Verify

Así como Seis Sigma ayuda a las compañías a eliminar defectos, reducir costos y mejorar la satisfacción del cliente en sus procesos de manufactura, igualmente puede aumentar la efectividad en los tradicionales procesos de desarrollo de software.

El estudio se centrará en el modelo definido como DMADDV⁶ utilizado para el desarrollo de nuevos proyectos software. Además la definición y descripción de los pasos que el equipo de trabajo considere deban conformar cada una de las fases del modelo.

Las seis fases que componen el modelo son: Definir, Medir, Analizar, Diseñar, Desarrollar y Verificar, son las que este proyecto tiene el propósito de analizar y definir los pasos que las componen, como se verá en los capítulos siguientes.

⁶ DMADDV: Acrónimo de Define, Measure, Analyze, Design, Develop and Verify

3. FASE DEFINIR

La primera fase en el modelo DMADDV⁷ es *definir*. Esta fase establece el escenario de un proyecto Seis Sigma eficaz, ayudando a responder las siguientes cuatro (4) preguntas:

- ¿Cuál es el problema o la oportunidad que se va a tratar?
- ¿Cuál es el objetivo? Es decir ¿qué resultados se quieren conseguir y cuando?
- ¿Quién es el cliente al que sirve o sobre que impacta este proceso y su problema?
- ¿Cuál es el proceso que se está investigando?

Es de vital importancia iniciar un proyecto Seis Sigma definiendo de forma clara y específica el problema a tratar. Este es el propósito de la fase definir.

Los pasos claves que definimos para esta fase son:

1. Entender el modelo del negocio.
2. Definir el problema.
3. Definir los roles y las responsabilidades.
4. Formar el equipo de trabajo.
5. Identificar los clientes.
6. Identificar lo crítico para la calidad (CTQ's).
7. Desarrollar el plan del proyecto.

⁷ DMADDV: Siglas del acrónimo Definir, Medir, Analizar, Diseñar, Desarrollar y Verificar.

8. Establecer el cuadro resumen del proyecto.
9. Definir el vocabulario específico.
10. Identificar los procesos actuales.
11. Presentar al comité para aprobar.

3.1 ENTENDER EL MODELO DEL NEGOCIO

Cada organización para alcanzar su misión, visión y objetivos, organiza sus actividades por medio de un conjunto de procesos, de tal forma que se pueda entender la funcionalidad de ésta. Estas reglas tienen que permitir a la organización ser competitiva en la medida que los requerimientos del mercado lo van exigiendo.

Para que el proyecto que se vaya a construir pueda dar valor al negocio que lo utiliza y a sus clientes, se debe tener en cuenta la comprensión del contexto del sistema mediante un modelo del negocio. La finalidad del modelo del negocio es describir cada proceso del negocio, especificando sus datos, actividades, roles y reglas.

Las siguientes preguntas servirán de ayuda para definir los proyectos según el modelo del negocio:

¿Cuáles son las razones del negocio que obligan a emprender este proyecto?

¿El proyecto tiene una relación directa con las metas y objetivos del negocio?

¿Qué salidas dominantes de los procesos del negocio puede el proyecto solucionar y cómo?

3.2 DEFINIR EL PROBLEMA

En la organización, la gente clave de la alta dirección debe estar convencida de que el cambio es algo necesario. Además, deben creer en la iniciativa de Seis Sigma y procurar manejar los siguientes tres aspectos: la identificación de la nueva orientación que debe darse para que el cambio sea realizado, el cambio cultural que ocurrirá internamente, y la respuesta de las personas impactadas por el cambio.

La mayoría de las actividades en la etapa preliminar a la implementación de Seis Sigma serán realizadas por el Comité de dirección de proyectos. Las responsabilidades de éste comité son las siguientes:

1. Guiar y dirigir los pasos iniciales en el proceso de Seis Sigma.
2. Asegurar los recursos y el soporte para los proyectos de Seis Sigma.
3. Mantener enfocada a la organización en los esfuerzos de Seis Sigma y encontrar formas más eficientes y efectivas para llevar a cabo las actividades planeadas.
4. Establecer guías para resolver problemas interdepartamentales y anticiparse a los problemas que el cambio pueda generar en la organización. Por lo tanto deberá encargarse de involucrar a todo el personal para que se incorpore a la iniciativa de Seis Sigma.

Iniciar el cambio dentro de la organización implica que el comité también contemple los siguientes aspectos:

- Desarrollar y mantener claramente las especificaciones del comité, que ayudarán a guiarlo en su propio trabajo y proveerlo de la dirección para el proceso entero de Seis Sigma.
- Educar a la organización acerca de Seis Sigma, y las razones del por qué esta iniciativa es tomada.

- Comunicar los beneficios de Seis Sigma. Este punto es el más importante para que los empleados se comprometan con la iniciativa.
- Comunicar el costo y esfuerzo de dicha implementación, sin olvidar resaltar el margen costo/beneficio esperado.
- Determinar los indicadores a ser utilizados para medir los resultados logrados en el proceso.
- Coleccionar, analizar, y distribuir los resultados de los proyectos a través de la organización. Esta es otra función de comunicación con todas las partes involucradas.
- Adaptar los resultados de los proyectos de Seis Sigma a las estrategias y la planeación organizacional.

El principal rol que debe realizar el Comité de Dirección de Proyecto, es identificar los asuntos o problemas potenciales. Numerosas fuentes de información pueden ser usadas para identificar proyectos de Seis Sigma, algunas se mencionan a continuación:

3.2.1 Retroalimentación con el Cliente o VOC⁸.

Es una fuente de información acerca de la funcionalidad de los procesos, permitiendo identificar los problemas que estos procesos poseen y posteriormente identificar las oportunidades para mejorarlos. Aunque, VOC es una de las fuentes de información más complejas y difíciles de recolectar, documentar, y mantener, sus resultados son más efectivos.

3.2.2 Oportunidades externas.

El Benchmarking⁹ ofrece oportunidades a la alta gerencia que no se han considerado previamente. Esta información puede sugerir nuevos mercados para viejos o nuevos productos, o incluso nuevas líneas de negocio.

⁸ VOC: Siglas en ingles para Voz del Cliente (Voice Of the Customer).

3.2.3 Experiencia.

El comité debe sacar el máximo provecho de la experiencia y del conocimiento de las personas que se encuentran directamente involucrados en el proceso con el fin de identificar una oportunidad para la mejora o el rediseño de un proceso.

Para definir el problema, el comité deberá establecer las declaraciones de una forma EMART¹⁰, que significa:

- **Específica.** El problema tiene que ser cuantificado. Por ejemplo, se reducirá a un 12% la tasa de desempleo en el país.
- **Medible.** Los resultados obtenidos deben ser medibles. Para el ejemplo anterior, la tasa de desempleo debe haberse medido anteriormente en el país. De lo contrario sería imposible llegar a medir los resultados.
- **Alcanzable.** El objetivo debe ser realista. Para el ejemplo, sería imposible reducir la tasa de desempleo al 5%.
- **Relevante.** El mejoramiento debe poseer un impacto de satisfacción en el cliente. En el ejemplo, el objetivo es relevante si el país tiene un alto índice de desempleo.
- **Tiempo Límite.** El mejoramiento propuesto debe ser logrado en un periodo de tiempo específico. En el ejemplo, el tiempo límite debería ser medido en meses en lugar de años.

⁹ Benchmarking: Es una poderosa herramienta para ayudar a las organizaciones a comparar los procesos con sus competidores, y entender que hace a los productos o servicios de otras organizaciones superiores

¹⁰ EMART: En inglés SMART (specific, measure, attainable, relevant, timebound)

La tabla a continuación, representa la estructura de definición del problema.

Tabla 3. Estructura de la definición del problema.

<p>¿Qué?</p>	<ul style="list-style-type: none"> ▪ ¿Qué proceso está implicado? • ¿Qué no está funcionando? • ¿Cuál es la deficiencia u oportunidad?
<p>¿Dónde / cuándo?</p>	<ul style="list-style-type: none"> • ¿Dónde se observa el problema o deficiencia? <ul style="list-style-type: none"> - Departamento. - Región. - Sector. - etc. • ¿Cuándo se observa el problema o deficiencia? <ul style="list-style-type: none"> - Hora del día / mes / año. - Antes o después de. - Etc.
<p>¿De qué envergadura?</p>	<ul style="list-style-type: none"> • Qué envergadura tiene el problema/ deficiencia/oportunidad. • ¿Cómo medirlo?
<p>¿Impacto?</p>	<ul style="list-style-type: none"> • ¿Cuál es el impacto del problema/ oportunidad. • ¿Cuáles son los beneficios de actuar o las consecuencias de no actuar?

3.3 DEFINIR ROLES Y RESPONSABILIDADES

Una poderosa función de Seis Sigma es la creación de una infraestructura que asegura el recurso humano necesario para ejecutar las actividades del proceso. Estos individuos actúan como agentes generadores de cambio.

Una de las tareas que se deben realizar al iniciar Seis Sigma es definir los roles adecuados en la empresa y dejar claras sus responsabilidades.

Básicamente existen cinco (5) participantes claves en el equipo de trabajo:

1. Líderes ejecutivos.
2. Patrocinadores.
3. Master Black Belts (MBB's).
4. Black Belts (BB's).
5. Green Belts (GB's)

3.3.1 Líderes Ejecutivos

Son los presidentes ejecutivos responsables del desempeño de la empresa en su conjunto, encargados de alcanzar las metas estratégicas de su compañía a través de Seis Sigma.

El papel fundamental de los líderes ejecutivos es decidir la implementación de Seis Sigma en la compañía y promocionarla públicamente por toda la organización. A medida que comienza esta iniciativa de cambios dentro de la compañía, se debe ver con claridad el liderazgo, además de la convicción de que Seis Sigma y sus resultados objetivos son la mayor prioridad para la empresa.

A continuación se mencionan algunos aspectos importantes con que deben contar los líderes ejecutivos.

- **Determinación:** Deben creer firmemente que Seis Sigma tendrá éxito.
- **Confianza:** La confianza es un motivador poderoso, los líderes ejecutivos deben mostrar confianza no sólo en la metodología Seis Sigma, sino también en las personas encargadas de llevarlo a cabo.
- **Integridad:** Los líderes necesitan hacer lo que dicen que van a hacer. La integridad estimula la lealtad y el respeto, dos motivadores para los empleados de la organización.
- **Paciencia:** Los líderes ejecutivos son responsables de llevar todas las fases Seis Sigma con serenidad y bajo control.

Las funciones específicas de los líderes ejecutivos incluyen:

- Establecer los roles que desempeñarán los miembros del equipo e infraestructuras de la iniciativa Seis Sigma.
- Seleccionar los proyectos específicos y asignar los recursos.
- Revisar periódicamente el progreso de los proyectos en curso y aportar ideas.
- Ayudar a cuantificar el impacto Seis sigma en la empresa.
- Evaluar los progresos e identificar los puntos fuertes y débiles del esfuerzo.
- Compartir las mejoras.
- Actuar como eliminadores de obstáculos cuando los equipos identifiquen barreras.

3.3.2 Patrocinadores

Los patrocinadores de Seis Sigma son individuos de alto nivel jerárquico que entienden la iniciativa y están comprometidos con su éxito.

Las responsabilidades del patrocinador incluyen:

- Definir y mantener objetivos amplios para los proyectos a su cargo, lo que incluye definir la misión del proyecto, y garantizar que los objetivos están alineados con las prioridades de la empresa.
- Orientar y aprobar los cambios de dirección o alcance de un proyecto, si se requiere.
- Localizar y negociar recursos para los proyectos.
- Representar al equipo ante el grupo responsable y actuar como modelo a seguir por los miembros del equipo.
- Trabajar con los propietarios de los procesos para garantizar una transición sin dificultades hacia la conclusión de un proyecto.
- Son los encargados de costear el presupuesto de los proyectos.

3.3.3 Master Black Belts (MMB)

Los MBBs brindan liderazgo técnico al programa de Seis Sigma. En consecuencia deben saber todo lo que sabe un Black Belt y además entender la teoría matemática que sustenta los métodos estadísticos. Al mismo tiempo ayudan a los BBs a aplicar esos métodos en situaciones inusuales. Cuando sea posible la capacitación en estadística debe estar a cargo de un MBB, de lo contrario se produce el fenómeno conocido como “propagación del error”, es decir, los BBs transmiten errores a quienes tienen el grado de GB, quienes a su vez transmiten errores a los miembros del equipo. Además están encargados de identificar, desarrollar, implantar, actualizar, y eliminar las herramientas Seis Sigma a ser utilizadas.

3.3.4 Black Belts (BB)

Los Black Belts son personas que combinan las habilidades de liderazgo, técnicas, comunicación y algún conocimiento estadístico.

Los BBs con éxito normalmente comparten los siguientes rasgos:

- Trabajan bien tanto por su cuenta como en equipo.
- Conservan la serenidad bajo extrema presión.
- Se anticipan a los problemas y actúan sobre ellos inmediatamente.
- Respetan a sus compañeros de trabajo y son respetados por ellos.
- Inspiran a los demás.
- Son capaces de delegar tareas a otros miembros del equipo y de coordinar sus esfuerzos.
- Comprenden y reconocen las habilidades y limitaciones de sus compañeros de trabajo.
- Demuestran una preocupación sincera por los demás, por lo que necesiten o quieren.
- Aceptan las críticas correctamente.
- Están preocupados por los actuales procesos y resultados y quieren mejorar el sistema.

Sus responsabilidades son:

- Servir como mentores de los GBs.
- Definir las métricas de los proyectos.
- Ayudar a determinar las herramientas para poder medir estas métricas.

3.3.5 Green Belts (GB)

Los Green Belts son los líderes de proyectos seis sigma capaces de formar equipos, colaborar con ellos y manejar esos proyectos de principio a fin. Reciben una capacitación que incluye gestión de proyectos, herramientas de gestión y control de calidad, resolución de problemas y análisis descriptivo de datos estadísticos.

Trabajan en los proyectos a tiempo parcial, normalmente en áreas específicas y delimitadas. Aplican las herramientas Seis Sigma para examinar y solucionar los problemas crónicos en proyectos dentro de sus trabajos normales. Ayudan también a recoger y analizar datos, realizar experimentos o conducir otras tareas importantes en un proyecto.

3.4 FORMAR EL EQUIPO DE TRABAJO

Dado que la mayor parte del trabajo Seis Sigma se lleva a cabo mediante equipos, es necesario tener claridad sobre algunos aspectos que se deben tener en cuenta en la selección de los miembros del equipo del trabajo.

Cuando se trata de equipos, muchas organizaciones sobrecargan un equipo con todo tipo de personas cuyas habilidades puedan ser necesarias durante el proyecto. No es sorprendente que los grandes equipos se muevan más lentamente y que sus miembros tiendan también a comprometerse y a entusiasmarse menos. Existen numerosas reglas de oro muy diferentes sobre el tamaño del equipo, pero el número óptimo para casi cualquier proyecto es entre cinco y ocho. Un número mayor hace que la comunicación se haga demasiado compleja, que las decisiones sean más difíciles y que la cohesión se debilite.

Las siguientes preguntas ayudarán a la hora de seleccionar miembros para el equipo¹¹:

- ¿Quién posee los mayores conocimientos del proceso y mejores contactos con el cliente?
- ¿Quién tiene los mayores conocimientos del problema o el mejor acceso a los datos?
- ¿Qué habilidades o perspectivas clave serán necesarias durante el curso del proyecto?
- ¿Qué grupos se verán afectados más directamente por el proyecto?
- ¿Qué grado de representación de dirección/supervisión/primera línea es más probable que se necesite?
- ¿Qué habilidades, funciones o niveles organizativos se pueden obtener según sean necesarios durante el proyecto?

A menudo son necesarias diferentes habilidades y talentos para que las mejoras del proceso funcionen correctamente. Una vez la gente se halle dentro del programa Seis Sigma, el siguiente paso es darles los conocimientos y las habilidades en las herramientas necesarias.

Un miembro del equipo, realiza el rol de usuario final¹², con el objetivo principal de ayudar a determinar los requerimientos del proyecto a nivel de negocio.

Es importante conformar el equipo de trabajo con gente que no solamente tenga los conocimientos en un área específica, también es importante contar con gente que tenga un conjunto de características personales, como las que se muestran a continuación:

¹¹ Tomada del libro “The Six Sigma Way”

¹² Rol de usuario final: llamados en inglés Functional owners.

- *Compromiso:* el individuo debe creer en el proyecto y estar dispuesto a hacer lo que sea necesario para que tenga éxito.
- *Inclinado a actuar:* el miembro del equipo debe sentirse obligado a finalizar el proyecto exitosamente.
- *Flexibilidad:* los miembros del equipo deben estar dispuestos a adaptarse al cambio, y además involucrarse con él.
- *Innovación:* ser personas iniciadoras de ese cambio, buscando nuevas formas de lograr los objetivos.
- *Influencia Personal:* es altamente deseable que todos los miembros sean bien respetados dentro de sus propias comunidades.
- *Trabajo en equipo:* tener la habilidad de relacionarse con los demás miembros, compartiendo conocimientos y experiencias.
- *Tiempo disponible:* los individuos que tienen una carga de trabajo demasiado pesada no deben ser elegidos para el equipo de trabajo. No serán lo suficientemente eficaces sino que también crearán desacuerdo en el equipo por faltar a reuniones o por no poder entregar sus compromisos a tiempo.

3.5 IDENTIFICAR LOS CLIENTES

El pilar de la existencia de cada organización son en realidad sus clientes, ya sean externos o internos se llegan a considerar socios del proyecto. Por tal motivo, la identificación de los clientes se convierte en un paso fundamental de toda iniciativa y de Seis Sigma principalmente. Saber qué es lo que realmente los clientes quieren y no asumirlo es el punto de partida para que cualquier proyecto sea exitoso, por que Seis Sigma nunca trata de manejar el problema, simplemente trata de eliminarlo.

Los clientes de una organización juegan diferentes roles en la compañía, los cuales se pueden identificar en las siguientes categorías.

- Externos: es el grupo de clientes más fácil de identificar, ya que son los que pagan por usar el producto o utilizan el servicio.
- Internos: son los que ayudan a crear el producto o servicio. Aquellos que se encuentran involucrados en el proceso.

La segmentación de clientes le da una ventaja competitiva a la organización, ya que puede crear sus objetivos basados hacia donde se va a enfocar el mercado. La clave es equilibrar y diversificar los esfuerzos para aprender de diferentes grupos.

Los métodos más utilizados para encontrar la clasificación de los clientes se encuentra en la tabla 4. Los métodos de la nueva generación empiezan a adquirir más fuerza debido a las ventajas que dan con respecto a los tradicionales, ya que plantean ser métodos indirectos con los clientes. La combinación de estos métodos depende de los clientes, el mercado, y los recursos.

Los datos obtenidos son un arma fundamental para la toma de decisiones en el proyecto o la compañía, por eso es importante invertir la cantidad necesaria de recursos para obtener la información que brinde la ayuda necesaria para el proyecto.

Tabla 4. Métodos para clasificar a los clientes.

Tradicional	Nueva Generación
<ul style="list-style-type: none"> • Sondeos. • Grupos focales. • Entrevistas. • Sistemas de reclamaciones formales. • Investigaciones de mercado. • Programas de compradores. 	<ul style="list-style-type: none"> • Entrevistas y estudios específicos y de multinivel. • Fichas para sondeos de opinión del cliente. • Almacenamiento de datos (Data Warehousing) y explotación de datos (Data Mining). • Auditorias a clientes o proveedores. • Despliegue de la función de calidad.

3.6 IDENTIFICAR LO CRÍTICO PARA LA CALIDAD (CTQ'S)

Identificar las características críticas para la calidad es el aspecto más importante que se debe tratar para solucionar el problema que se está analizando en el proyecto y que influye directamente en la satisfacción del cliente.

Las CTQs representan las características que debe tener el producto o servicio y son definidas por los clientes, el BB, y los GB. Se convierten en los objetivos específicos del proyecto y deben tener las mismas características de la definición del problema: específico, medible, alcanzable, relevante, y en un tiempo específico.

Al finalizar el proyecto los resultados serán evaluados basados en las CTQ's, si se cumplieron las metas de las medidas se califica al proyecto como exitoso, de lo contrario, como fracaso. Debido a esta razón, se debe tener gran cuidado en el momento de hacer la declaración y sobretodo a lo que se espera obtener como resultado.

3.7 DESARROLLAR EL PLAN DE PROYECTO

La creación del Plan de Proyecto es un proceso dinámico que da lugar a un documento que dirige todo el diseño, desarrollo y futuro de un proyecto. Agrupa cada uno de los resultados, los procedimientos, los recursos necesarios para producirlos, y las medidas de calidad que deben cumplir para ser aceptado. Este documento puede crecer y cambiar durante el ciclo de vida de los proyectos. La planeación de proyectos aumenta las probabilidades de éxito con la detección a tiempo de los problemas mediante una constante supervisión.

El Plan de Proyecto se puede estructurar de muchas formas, pero principalmente debe contener los siguientes componentes:

- Definición del alcance y objetivos.
- Costo/ beneficio.
- Presupuesto.
- Riesgos.
- Cronograma.

Una de las habilidades que debe poseer el responsable de esta tarea es la administración de proyectos, disciplina que busca la aplicación de técnicas, conocimientos, y herramientas para abarcar proyectos que cumplan con los requerimientos planificados.

3.7.1 Definición del alcance y objetivos

El *alcance* es el que define y documenta las actividades específicas y los resultados de un proyecto particular. Declarar el alcance sirve como base para tomar decisiones futuras sobre el proyecto, este debe ser tan claro que debe permitir especificar lo que contiene y lo que no contiene el proyecto.

Todo los miembros del equipo del Proyecto deben trabajar juntos para desarrollar la declaración del alcance, usando como guía los siguientes ítems:

- La visión de equipo.
- Revisión del modelo de negocios, análisis de las necesidades y análisis de los procesos de nivel tecnológico existentes.
- Recursos humanos y financieros disponibles.
- Calidad deseada del proyecto.

Lo que debe quedar documentado por el comité es:

- La principal funcionalidad que será implementada.
- Los tipos de resultados que están en el alcance y los que no.
- Las fuentes de datos o bases de datos que están dentro y fuera del alcance.
- La cantidad de unidades que serán afectadas y/o se espera usen el sistema.
- Lo que realmente esta fuera del alcance.

Los *objetivos* del proyecto son los criterios cuantificables que se deben satisfacer para que el proyecto sea considerado acertado y a través de los cuales se mide el éxito del mismo. A partir de los objetivos los miembros del equipo, jefes del proyecto, usuarios finales, etc., determinan si el producto final cumple con los requerimientos y en que grado o nivel de satisfacción.

La estructura de la declaración de los objetivos se puede estandarizar bastante con los siguientes tres elementos:

- *Descripción de lo que hay que hacer.* La declaración de los objetivos debe empezar por un verbo (reducir, aumentar, eliminar), pero es conveniente evitar verbos que sean demasiado ambiguos.
- *Un objetivo medible para los resultados deseados.* El objetivo debe cuantificar el ahorro en costes perseguido, la eliminación de defectos o la reducción de tiempo, en porcentajes o cifras reales. Si es demasiado pronto incluso para realizar suposiciones, conviene dejar un marcador que indique donde se planea añadirlo más adelante. El objetivo medible es el que su equipo y los directivos de la empresa van a utilizar para evaluar el éxito del proyecto.
- *Un plazo o marco temporal para los resultados.* Es posible que tenga que revisar posteriormente la fecha definida en la primera parte del proyecto,

pero establecer un plazo ayuda a reunir recursos y compromisos y acorta el tiempo del ciclo del proyecto.

3.7.2 Costo/ Beneficio

EL análisis de costo/beneficio de cualquier proyecto juega un papel importante para la aprobación e inicio, convirtiéndose en uno de los fundamentos en cuales se basan los altos directivos para tomar la decisión de aprobarlo y dar luz verde a su comienzo.

Las oportunidades del proyecto deben ser evaluadas según los criterios establecidos por el equipo y la alta dirección. Los siguientes ítems se tienen en cuenta para identificar si no existe alguna oportunidad atractiva y si este proyecto no traerá beneficios.

- No estar de acuerdo con la voz del cliente.
- Tiempo de desarrollo no lógico.
- No entender claramente el proceso.
- Conflicto de intereses entre los patrocinadores del proyecto.
- Bajo retorno de inversión o poco valor agregado a la compañía.

Algunos de los métodos utilizados para el análisis del costo/beneficio son los siguientes:

- *Tiempo - valor del dinero:* Es considerado por los métodos mas sofisticados para realizar el análisis de costo/beneficio. También es conocido como el factor de descuento, es simplemente utilizado para convertir el valor futuro del dinero en valor presente. Se basa en que el dólar de hoy tiene mas valor que un dólar en unos años debido a los interés o a la ganancia que se pueda obtener.

- *Punto de equilibrio:* Es el tiempo que tomaría para que el total de los ingresos incrementados o la reducción de gastos sea igual al costo total. Sin embargo no toma en cuenta el Tiempo – Valor del dinero.

Los pasos para determinarlo son los siguientes:

1. Costo.
2. Ingresos incrementados totales o reducción de gastos.
3. Formula.

$$\text{Punto de Equilibrio} = (\text{Costo} / \text{Ingresos incrementados y/o reducción de gastos}) \times 12 \text{ (meses)}$$

- *Periodo de Devolución:* Es el tiempo requerido para recuperar el monto inicial de una inversión de capital. Este método calcula la cantidad de tiempo que se tomaría para lograr un flujo de caja positivo igual a la inversión total. Toma en cuenta beneficios como el valor asegurado. Este método indica esencialmente la liquidez del esfuerzo por mejorar un proceso en vez de su rentabilidad. Al igual que en el análisis de punto de equilibrio, el análisis del periodo de devolución no tiene en cuenta el valor del dinero en el tiempo.

Los pasos para determinarlo son los siguientes:

1. Costo.
2. Valor Asegurado.
3. Total ingresos incrementados y/o reducción de gastos.
4. Fórmula.

$$\text{Periodo de devolución} = [(\text{Costo} - \text{Valor Asegurado}) / \text{total de ingresos incrementados y/o reducción de gastos}] \times 12 \text{ (meses)}$$

- *Valor presente neto*¹³ (VPN): El VPN representa el Valor Presente (VP) de los flujos salientes de caja menos la cantidad de la inversión inicial (I).

$$VPN = VP - I$$

El valor presente del flujo de caja futuro es calculado utilizando el costo del capital como un factor de descuento. El propósito del factor de descuento es convertir el valor futuro del dinero en valor presente y se expresa como $1 +$ la tasa de interés (i).

Si la tasa de interés es del 25%, el factor de descuento es de $(1 + i)$. El factor de descuento puede expresarse como $(1 + 0.25)$ o 1.25

Los pasos para determinar el VPN son los siguientes:

1. Ingresos.
2. Valor Asegurado.
3. Factor de Descuento.
4. Inversión.
5. Formula.

$$VP = (\text{Ingresos} + \text{Valor Asegurado}) / (\text{Factor de Descuento})$$

$$VPN = VP - \text{inversión (I)}$$

- *Tasa interna de retorno*¹⁴ (TIR): Es la tasa de interés que hace la ecuación de la inversión inicial (I) con el valor presente (VP) de los futuros flujos de caja entrantes. Esto es, a la tasa interna de retorno $I = VP$ o $VPN = 0$.

¹³ VPN: Valor Presente Neto

¹⁴ TIR: Tasa Interna Retorno.

En el ejemplo anterior, la tasa era del 25%. Por lo tanto cualquier propuesta para mejorar un proyecto o procesos deberá difícilmente exceder esta tasa si se quiere considerar.

Cuando se calcula el TIR, el VPN se fija en cero y se resuelve para un interés (i). En este caso, el factor de descuento es $(1 + i)$ ya que no conocemos el interés verdadero, solamente conocemos el interés deseado.

Pasos para determina: el TIR

1. VPN.
2. Ingresos.
3. Valor Asegurado.
4. Factor de Descuento.
5. Inversión.
6. Formula.

$$VP = (\text{Ingresos} + \text{Valor Asegurado}) / (\text{Factor de Descuento})$$

$$VPN = VP - \text{Inversión (I)}$$

Para calcular el TIR, se lleva el VPN a cero y se resuelve para un interés (i)

Si alguno de estos tópicos no existe no se inicializa el proyecto o se debe definir la oportunidad nuevamente. Si existe la oportunidad, se dispone a determinar el incremento en la eficiencia, eficacia y la adaptabilidad para recuperar los costos de los recursos que consumirá el esfuerzo. Varias inquietudes deben ser tratadas como lo son:

- ¿Quiénes serán los más beneficiados por el mejoramiento del proceso?
- ¿Cuáles personas tienen interés en el rendimiento del proceso?
- ¿Éstas personas reconocerán el valor agregado añadido?
- ¿Los mejoramientos realizados a los procesos, convertirán a la compañía en una organización más competitiva, en términos de mejor rendimiento en el proceso, incremento en la porción del mercado, calidad e ingresos?

Finalmente para determinar los beneficios del proyecto que se presentarán al comité o alta dirección de la compañía, el equipo de trabajo debe plantear los siguientes puntos:

- **Impacto sobre los clientes y requisitos externos.** ¿Es esto una oportunidad importante para nuestros clientes?
- **Impacto financiero.** ¿Existe la reducción de costos, mejora de la eficiencia, aumentos en las ventas, o ganancia en la cuota del mercado?. ¿Qué ganancia monetaria puede existir a corto plazo? ¿Y a largo plazo? (es recomendable ser realista o exponer una cifra atractiva pero por debajo de lo que se estima, ya que en el momento de entregarse el proyecto si la cifra es superada, se obtendrá mayor éxito).
- **Tendencia.** ¿El problema empieza a tener un comportamiento de crecimiento o de decrecimiento en el tiempo futuro? ¿Que pasaría si no se hace nada?
- **Innovación.** ¿Qué nuevos conocimientos de la empresa, de los clientes, de los procesos o del sistema Seis Sigma podríamos obtener con este proyecto?
- **Gestión.** ¿Hasta qué punto ayudará este proyecto a dirigir los grupos de la organización y a crear una mejor gestión del proceso global?

Por lo general en compañías que usan Seis Sigma, existen grandes Ys es decir, Iniciativas más importantes que el negocio se encuentra trabajando, por lo general para aumentar ingresos, disminuir costos, mejorar el servicio o

algún otro tipo de beneficio. Si el proyecto no ataca una de estas grandes Ys, entonces no se hace.

Se hacen llamar grandes Y debido a que se utiliza la ecuación $Y = f(x)$ ¹⁵, donde X es un problema u oportunidad que se tiene. Si se modifica X, entonces la Y se ve afectada. Por ejemplo si para la organización la meta este año es reducir costos entonces Y es Costos. algunos ejemplos de X son:

- Inventario no esta siendo bien mejorado.
- Los sistemas actuales no están midiendo bien lo que se esta gastando.
- Eliminar 500,000 de gastos de arriendo al año haciendo lo siguiente.
- El departamento de información tecnológica quiere migrar de versión el servidor de correo electrónico por que la versión utilizada ha sido usada por mas de tres años.

El último de los cuatro proyectos es el que no se realizará y se desecha primeramente porque no afecta directamente la Y, que para el ejemplo es la reducción de costos.

3.7.3 Presupuesto

Nada se lleva a cabo sin dinero, y los proyectos de desarrollo de software no son la excepción. Existen muchas razones por la cuales se debe calcular los costos antes de incurrir en ellos, la principal es poder estimar una idea de cuanto costarán en términos de dinero los objetivos propuestos.

¹⁵ $Y = f(X)$. mayor profundización en el paso “Determinar que medir”, fase “Medir”

El presupuesto es una estimación que se realiza y es responsabilidad del líder del proyecto crear estándares para que el proyecto cumpla con las expectativas esperadas bajo los costos predecidos. Aspectos como cronogramas razonables, recursos y expectativas de la alta dirección deben ser tenidos en cuenta en la estimación que realice el equipo de trabajo. A continuación se describen las metodologías de estimación existentes para calcular el presupuesto de un proyecto.

Cada una de las metodologías existentes poseen fortalezas y debilidades por tal razón la utilización de varias metodologías como complemento unas de otras podría ser de gran herramienta para el proyecto.

- Analogía.

Busca proyectos similares realizados con anterioridad para compararlos, los datos son agregados al proyecto en curso para realizar la estimación. La analogía sólo puede ser utilizada por completo a nivel de sistema o de componentes.

La mayor fortaleza de este método es que se basa en la experiencia de proyectos ya ocurridos, las diferencias entre proyectos completados y proyectos propuestos pueden ser identificadas y se facilitaría estimar el impacto. Las debilidades de la metodología se observan al momento de encontrar las diferencias entre proyectos y en la limitación que existiría si no se encuentran proyectos similares o si los datos de los proyectos anteriores no son confiables.

- Abajo – Arriba.

La metodología identifica y estima cada componente individualmente y luego combina los resultados para producir un estimado del proyecto total.

Una de las dificultades que presenta es cuando la información no se encuentra disponible debido al proceso de ciclo de vida¹⁶ seleccionado. Además tiende a utilizar demasiado tiempo para llevarse a cabo.

- Arriba – Abajo.

Se basa en las características más sobresalientes del software. El proyecto subdivide los componentes y las fases del ciclo de vida desde un alto nivel hasta el más bajo nivel.

Las ventajas que presenta son las consideraciones que tiene la metodología con actividades a nivel del sistema como integración, documentación, control del proyecto, administración, etc. ya que la mayoría de las metodologías las ignora. Es considerada una de las más rápidas, fáciles de implementar y que requiere de pocos detalles para trabajar. Las desventajas son la falta de precisión y la tendencia a pasar por alto los niveles bajos de los componentes y los problemas técnicos, además ofrece muy pocos detalles en la justificación de las estimaciones.

- Juicio Experto.

Utiliza el talento humano para estimar los costos del proyecto, busca las personas con mayor conocimiento y experiencia en proyectos similares.

La ventaja se muestra en la experiencia que se utiliza para realizar la estimación, además de ofrecer un valor agregado para establecer el impacto del proyecto. Es la metodología que más se utiliza aunque siempre es complementada con otra. Las desventajas se presentan al momento de

¹⁶ Ciclo de Vida: Metodología para desarrollo de software, ver paso “SELECCIÓN DE METODOLOGÍA PARA DESARROLLO DE SOFTWARE”, fase “DISEÑAR”.

documentar y justificar los factores que ha considerado el experto para el proyecto.

- Algoritmo o Parametrizado.

Involucra el uso de ecuaciones para mejorar el rendimiento de la estimación, y su mayor utilización es en proyectos de desarrollo de software. Las ecuaciones son basadas en investigaciones y datos históricos que son las entradas como el conteo de líneas de código fuente (CLCF¹⁷), número de funciones, y otros tipos de costos como lenguajes, metodologías de diseño, habilidades, riesgos, etc.

Las ventajas de la metodología se presentan ya que la generación de resultados pueden repetirse, son fáciles de modificar, y las fórmulas son modificables para un mejor análisis sobre los costos. Sin embargo, los resultados son cuestionables cuando la estimación se realiza sobre proyectos que utilizan nuevas tecnologías; las ecuaciones finales no tienen la capacidad de trabajar bien con condiciones especiales, como personal que no se hubiese tenido en cuenta para realizar trabajos extras, aunque ninguna metodología es tan flexible para no sentir el impacto de sucesos de ese tipo.

3.7.4 Riesgos

Los riesgos son parte inherente de todos los proyectos, de hecho, son un factor esencial en el progreso. A pesar que algunos son inevitables, su identificación a tiempo y administración adecuada aumenta las posibilidades de éxito de un proyecto.

¹⁷ SLOC: Counting source lines of code.

Antes de analizar cómo se administran los riesgos, se define riesgo como la posibilidad de sufrir una pérdida. Para un proyecto específico, las consecuencias pueden ser: un producto terminado con menor calidad, aumento en los costos, retrasos en el cronograma de actividades, o no alcanzar el propósito y la intención del proyecto. En otras palabras, un riesgo es un problema en espera de ocurrir.

Para poder analizar los riesgos, se han considerado las siguientes categorías:

- *Riesgos del proyecto:* identifican los problemas potenciales de presupuesto, planificación temporal, personal, recursos, clientes y requisitos. Si estos riesgos se hacen realidad, es posible que la planificación del proyecto se retrase y que los costos aumenten.
- *Riesgos técnicos:* Identifican problemas potenciales de diseño, implementación, de interfaz, de verificación y de mantenimiento. Si estos riesgos se hacen efectivos, la implementación puede llegar a ser difícil o imposible.
- *Riesgos del negocio:* Amenazan la viabilidad del software a construir. Los principales riesgos del negocio son: construir un producto o sistema excelente que nadie quiere, construir un producto que no encaja en la estrategia comercial de la compañía, construir un producto que el departamento de ventas no sabe como vender, perder el apoyo de una gestión experta debido a cambios de enfoque o a cambios de personal, y perder presupuesto o personal asignado.

Para identificar los riesgos, suele ser efectivo crear una lista de comprobación de elementos de riesgo. Esta lista se enfoca en un

subconjunto de riesgos conocidos y predecibles, algunos de estos son listados a continuación en la tabla 5.

Tabla 5. Tipos de riesgos.

TIPO DE RIESGO	DEFINICIÓN	POSIBLES RIESGOS
<i>Tamaño del producto</i>	Riesgos asociados con el tamaño general del software a construir o modificar.	<ul style="list-style-type: none"> • El tamaño estimado del producto • Tamaño de las Bases de Datos creadas. • La cantidad de software reutilizado.
<i>Impacto en el negocio</i>	Riesgos asociados con las limitaciones de la gestión o del mercado.	<ul style="list-style-type: none"> • Efecto del producto en los ingresos de la compañía. • Número de clientes que usarán este producto. • Costos asociados por un retraso en la entrega. • Costos asociados por un producto defectuoso.
<i>Características del cliente</i>	Riesgos asociados con el cliente y la habilidad del desarrollador para comunicarse con él en los momentos oportunos.	<ul style="list-style-type: none"> • Los clientes tienen diferentes necesidades. • Los clientes tienen diferentes personalidades. • Los clientes se contradicen a menudo. • Los clientes están dispuestos a colaborar en el proceso.
<i>Definición del proceso</i>	Riesgos asociados con el grado de definición del proceso del software y su seguimiento por la organización de desarrollo.	<ul style="list-style-type: none"> • Está bien definido el proceso de software. • El análisis, diseño y pruebas se realizan sobre la marcha. • El equipo estima importante el concepto de calidad.
<i>Entorno de desarrollo</i>	Riesgos asociados con la disponibilidad y la calidad de las herramientas que se van a emplear en la construcción del producto.	<ul style="list-style-type: none"> • El código generado por las herramientas CASE es ineficiente. • No se pueden integrar algunas herramientas.
<i>Tecnología a construir</i>	Riesgos asociados con la complejidad del sistema a construir y la tecnología de punta que contiene el sistema.	<ul style="list-style-type: none"> • El software interactúa con hardware nuevo o no probado. • La base de datos usada no tiene el rendimiento que se esperaba.
<i>Tamaño y experiencia del grupo de Trabajo</i>	Son los riesgos asociados con la experiencia técnica y de proyectos de los ingenieros del software que van a realizar el trabajo.	<ul style="list-style-type: none"> • Se dispone de la mejor gente. • El personal tiene los conocimientos adecuados. • Se cuenta con suficiente personal.

Un equipo de proyecto que funciona eficazmente mide los riesgos continuamente y emplea la información para la toma de decisiones en todas las etapas del proyecto. Existen dos enfoques distintos para la administración de riesgos. Uno es *reactivo* y el otro es *proactivo*. La administración reactiva de riesgos significa que el equipo del proyecto reacciona a las consecuencias de los riesgos (los problemas reales) conforme ocurren. La administración proactiva de riesgos significa que el equipo del proyecto cuenta con un proceso visible para administrarlos. Este proceso se puede medir y repetir.

El proceso de administración proactiva de riesgos es el que se muestra a continuación en la figura 2.

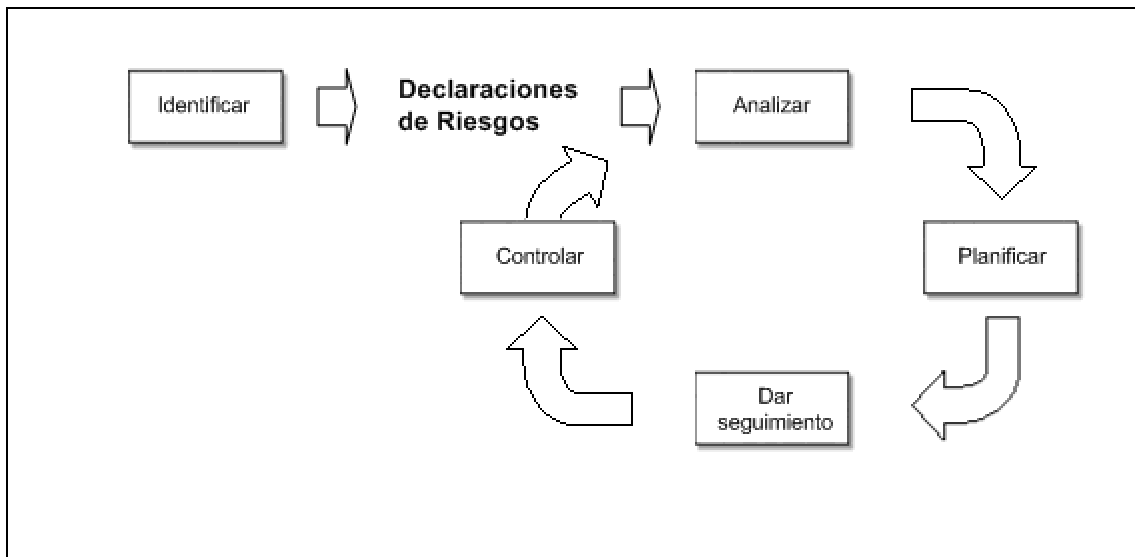


Figura 2. Ciclo de análisis de riesgos proactivo.

La *identificación* de riesgos es el primer paso en el proceso de la administración proactiva de riesgos. Los riesgos deben identificarse antes de que puedan administrarse. La identificación de riesgos proporciona al equipo del proyecto las oportunidades, indicios e información que le permiten ubicar los riesgos principales antes de que afecten adversamente al proyecto.

Cuando se declara un riesgo, el equipo de trabajo no debe considerar sólo un síntoma, sino también un resultado. Por esa razón, la declaración del riesgo debe incluir lo que provoca que surja la situación (esto es, la condición) y el resultado esperado (la consecuencia). Para documentar estos defectos y administrarlos eficientemente, se recomienda utilizar el cuadro Análisis de los modos de fallo y efectos (AMFE¹⁸).

El *análisis* de riesgos es el segundo paso en el proceso de administración proactiva de riesgos. Es la conversión de los datos de un riesgo a información para la toma de decisiones respectiva.

La *planificación* de acciones convierte la información sobre un riesgo en decisiones y acciones. La planificación implica desarrollar acciones para enfrentar los riesgos individuales, establecer prioridades en las acciones para un riesgo, y crear un plan integrado de administración de riesgos.

El *seguimiento* es la función de vigilancia del plan de acciones para riesgos, es esencial para la implementación de un plan de acciones eficaz. Esto implica establecer las unidades de medición del riesgo y los eventos de activación necesarios para asegurar que funcionan las acciones planificadas.

El *control* de riesgos es el último paso en el proceso de administración proactiva de riesgos. Se debe combinar con los procesos de administración de un proyecto para controlar los planes de acciones, corregir las variaciones de los planes, responder a los eventos de activación, y mejorar el proceso de administración de riesgos.

¹⁸ FMEA: Failure mode effect & analysis, ver anexo D.

3.7.5 Cronograma

El cronograma se convierte en un mecanismo utilizado en el proyecto para planear los tiempos, las tareas, recursos y costos que se involucrarán.

En el momento de planear un cronograma se debe ser realista con las fechas a establecer, la calidad del producto final, y el costo en que esto incurrirá.

Para desarrollar el cronograma se deben seguir los siguientes pasos:

1. Identificar las actividades.
2. Establecer la secuencia de las actividades.
3. Duración estimada de cada actividad.
4. Asignar los responsables.
5. Definir los recursos de cada actividad
6. Estimar los costos.

La identificación de actividades se lleva a cabo utilizando diferentes métodos, como la lluvia de ideas, entre otros. Las personas con más experiencia dentro del equipo en este tipo de proyectos, son de gran aporte en la identificación y en la depuración de éstas.

Después de determinar las actividades y su secuencia, se debe estar seguro de que la definición de entregables y logros posean una fecha estimada de cumplimiento.

En el momento de definir los logros se debe tener cuidado en diferentes aspectos, como lo son:

- Programar fechas de entregas irreales o no alcanzables, con el fin de impresionar al comité.
- No poseer una gran calidad en los objetivos.

- Mover las actividades que se ha programado para un alcance de un logro, hacia otro logro dentro del proyecto; sin tener en cuenta el impacto y el nuevo cronograma en todo el proyecto.

Para estimar la duración de cada actividad, se hace énfasis principalmente en la experiencia de proyectos anteriores. Si la actividad es totalmente nueva, se deben utilizar diferentes escenarios que puedan ocurrir, como la no predicción de recursos, adaptabilidad y capacidad de los miembros para esa actividad.

En el desarrollo de proyectos que involucran el desarrollo de aplicaciones informáticas existen modelos matemáticos que se utilizan para estimar la duración de las tareas o actividades del proyecto, modelos como Puntos de Función y COCOMO hacen parte del área de ingeniería del software

Los cronogramas sufren cambios en el transcurso del proyecto, convirtiéndolo en una actividad dinámica e iterativa.

Existen diferentes tipos de gráficos para representar los cronogramas, el tipo más utilizado y preferido por la mayoría de los líderes de proyecto es el diagrama de GANTT, ya que es más eficiente al momento de identificar actividades y tiempos. Permite comparar el cronograma con el de otros proyectos fácilmente. Las fases y actividades se distribuyen de arriba abajo, y el tiempo es representado de izquierda a derecha.

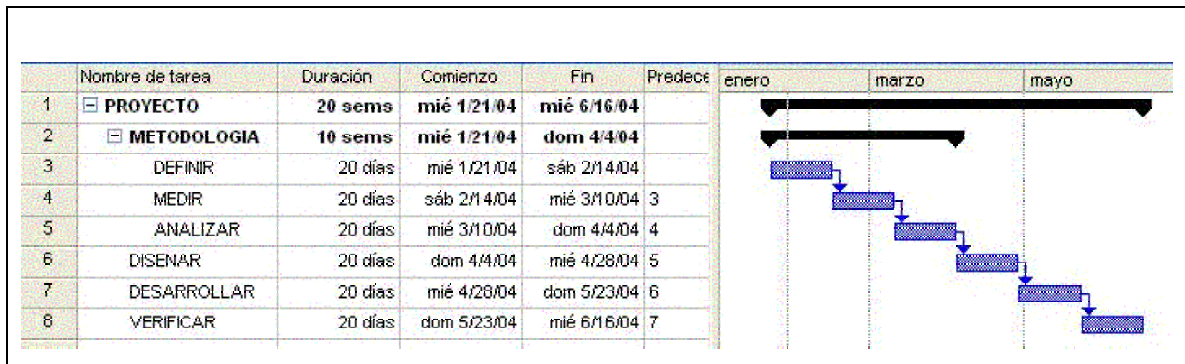


Figura 3. Diagrama de GANTT.

El diagrama de GANTT aunque es el gráfico que mejor se hace entender, posee algunas limitaciones, por citar algunas tenemos:

- No identifica las semanas críticas entre las fases.
- No posee coordinación entre los recursos o los requerimientos en el cronograma.

La decisión del tipo de gráfico a utilizar es responsabilidad del líder del equipo, y debe contar con la aprobación de los demás miembros.

3.8 DESARROLLAR EL CUADRO RESUMEN DEL PROYECTO

Este cuadro es uno de los documentos más importantes en un proyecto Seis Sigma, se utiliza como resumen de toda la información clave del proyecto. Su función principal es ayudar a los miembros del equipo a entender por que se forma el equipo, que metas se desean alcanzar, cuánto tiempo tomará el proyecto y cuánto tiempo el equipo dedicará a él.

Después de documentar esta información el líder del proyecto o el GB mantiene un control sobre lo que se definió de esta forma se evitarán los malentendidos y por el contrario existirá mayor focalización, en otras palabras: “menor variación”.

Este documento se inicia desde el primer encuentro con el equipo de trabajo y los campos que no se completan en éste, se actualizarán a lo largo del proyecto.

Existen muchas opciones para desarrollar y dar formato a un Cuadro de Proyecto. El modelo que se puede utilizar para el proyecto se observa en el anexo A.

3.9 DEFINIR UN VOCABULARIO

En todos los proyectos se encuentran personas con diferentes perspectivas con respecto a definiciones o conceptos de elementos que influyen directamente en el proyecto, creando un vocabulario. Por lo tanto, se debe realizar una investigación sobre los significados que tienen para los clientes cada uno de los términos del vocabulario utilizado y así crear un léxico común para todos los involucrados en el proyecto. Por ejemplo, identificar si un libro se encuentra en mal estado, puede significar para un estudiante (cliente 1) que el libro se encuentre subrayado en varias hojas, mientras que para otro estudiante (cliente 2) el que se encuentre subrayado no lo clasifica como libro en mal estado.

La metodología Seis Sigma busca la eliminación de los defectos y la satisfacción del cliente, para el ejemplo, las insatisfacciones se consideran como defectos. Conocer estas definiciones nos ayudarán a eliminar los defectos que causan insatisfacción del cliente.

Esta tarea puede ser realizada con la voz del cliente de una forma transparente para el cliente.

3.10 IDENTIFICAR LOS PROCESOS ACTUALES

Determinar los procesos críticos es uno de los pasos fundamentales dentro de la fase definir. Identificar cuales son los procesos críticos y cuales no, es responsabilidad del BB.

Al inicio de este paso, los siguientes interrogantes deben ser resueltos:

- ¿Qué productos o servicios produce el proceso?
- ¿Quién es el cliente?
- ¿Qué técnica esta disponible para medir la satisfacción del cliente?

Para determinar los procesos críticos, se deben contemplar los siguientes pasos:

- a. *Entradas del proceso.* Se refieren a los equipos, materiales, métodos, y el ambiente necesario para producir los productos o servicios.
- b. *Salidas del proceso.* Son los resultados del proceso, productos o servicios.
- c. *Objetivos del proceso.* Es la definición de los objetivos, metas, o logros.
- d. *Voz del proceso.* Es un mecanismo de realimentación en el cual la calidad del proceso puede medirse y examinarse en función de encontrar las metas propuestas, indicando los niveles en que se encuentra. Para entender la voz del proceso surgen las siguientes preguntas que deben ser resueltas.
 1. ¿Qué características críticas del proceso pueden ser mejoradas para que los servicios puedan encontrar las necesidades y las expectativas de los clientes?

2. ¿Cuáles metas a corto plazo pueden ser establecidas para que las características críticas de los procesos puedan encontrar las necesidades y expectativas de los clientes?
 3. Son necesarias otras fuentes para proveer información en busca de encontrar las nuevas metas?
 4. ¿Qué indicadores se utilizarán para medir dentro o durante el proceso?
 5. ¿Existe un método para recolectar datos del proceso?
 6. ¿El proceso actual se encuentra alcanzando los objetivos y metas propuestas que se establecieron desde un comienzo?
- e. *Cliente*. Son los que usan los productos o servicios resultado del proceso, convirtiéndose en los verdaderos jueces de la calidad del producto. El cliente se convierte en lo más importante en el proceso, es la razón de la existencia del proceso.
- f. *Necesidades del cliente y expectativas*. Ilustra los atributos que poseen los productos y servicios que los clientes requieren. Esta información puede ser suministrada por los patrocinadores o altos directivos.
- g. *Necesidades del cliente*. Se pueden entender como las expectativas específicas del cliente. Seleccionar atributos cuantificables brindan la oportunidad de alcanzar la calidad en servicios o en productos esperados.
- h. *Voz del cliente*. Son los datos que representan la perspectiva o necesidades de los clientes de la empresa, que deben traducirse a requisitos medibles para el proceso. Antes de utilizar alguna técnica de recolección de datos, el BB debe tener conocimiento en las siguientes preguntas:
- ¿Qué tanto satisface el proceso a los clientes?

- ¿Qué se intenta al salir a hablar con clientes y encontrar sus necesidades y expectativas?
- ¿Los indicadores por los cuales se medirán los datos son correctos?

Diferentes técnicas se encuentran disponibles para identificar la voz del cliente¹⁹.

Medir la satisfacción del cliente es importante para cualquier tipo de proyecto dentro de una organización. El mecanismo más usado es la realimentación; cualquier mecanismo es válido desde que los datos sean coherentes e íntegros, y ofrezcan la información que se busca. La voz del cliente es un punto que toma un largo periodo de tiempo en este paso, para compensar eso, la voz del proceso entra a jugar un papel importante debido a que se considera el espejo de la voz del cliente.

Para determinar cuáles son los procesos más críticos para el proyecto, se regresa a la ecuación de $Y = F(x)$, como se explicaba anteriormente las X se definen como los problemas que afectan directamente las Y , que son las iniciativas de la compañía. En la mayoría de las oportunidades se da que un Y es afectada por varias X , esto da como resultado lo siguiente:

$$Y = F(x_1, x_2, \dots, x_n)$$

Por ejemplo, si la Y es reducir costos, las X pueden ser, pago de impuestos (X_1), nómina (X_2), arriendo (x_3), servicios públicos (x_4), etc. Se obtiene la siguiente ecuación:

$$Y = 200 x_1 + 100 x_2 - 500 x_3 + 100 x_4$$

¹⁹ Técnicas para identificar la voz del cliente anexo B.

Por tal razón se selecciona a la x3 como el proceso más crítico y se deben enfocar todos los esfuerzos para solucionarlo, para el ejemplo, la x3 es el arriendo, lo cual hace que Y sea negativa, entonces la solución sería convertir x3 igual cero, para que la Y sea positiva, este sería el objetivo principal del proyecto Seis Sigma.

Diagrama de Flujo de Procesos (DFP)²⁰

El diseño de nuevos servicios, o el mejoramiento de los actuales, requieren que los miembros del equipo de trabajo tengan un claro entendimiento del proceso. La documentación además se convierte en pieza fundamental para la creatividad, el liderazgo, la definición, organización, control y cierre del proyecto. En la actualidad existen diferentes formas de documentar los procesos, los mas utilizados en las compañías Seis Sigma son los diagramas de flujo de procesos, ya que brindan una representación gráfica del proceso desde su comienzo hasta el final, pasando por las diferentes localizaciones por donde transcurre, ayudando a los miembros a identificar los puntos claves.

Los diagramas de proceso tienen como objetivos principales:

- Convertir un proceso realmente complejo en algo fácil de entender, dividiéndolo en componentes más pequeños.
- Documentar el flujo del proceso, mostrando la secuencia de las tareas.
- Entender las relaciones entre las tareas del proceso.
- Proveer argumentos para evaluar el proceso.
- Ofrecer un mayor entendimiento del proceso con el fin de prevenir los defectos antes que corregir los errores en el futuro.

²⁰ Process flow Diagram (PFD).

Para tener una herramienta bien formada se debe tener una estandarización en los elementos utilizados en todas las representaciones de procesos.

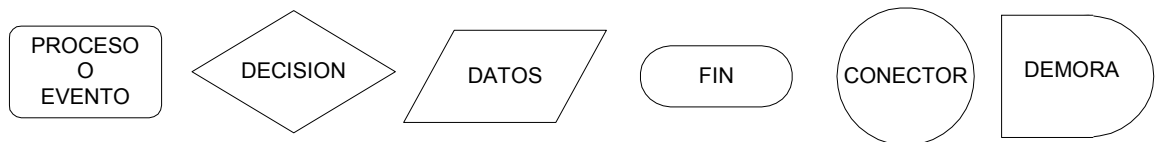


Figura 4. Elementos utilizados en la representación de procesos.

Los diagramas de flujo de procesos tienen la característica de ser iterativos, por lo tanto comienzan con un alto nivel de gráficos, llegando a un nivel tan detallado del proceso como se requiera. Los diferentes tipos de diagramas que se pueden utilizar son:

- *Diagrama de alto nivel.* Es el mapa inicial, que contiene el proceso en totalidad sin entrar en detalle en la trayectoria que sigue dentro de los estados por los que pasa; desde el principio hasta el fin. Al finalizar los mapas a este nivel y presentarse al grupo, se toma la determinación de ir al siguiente nivel si se considera necesario o si en este nivel los procesos son suficientemente comprensibles.

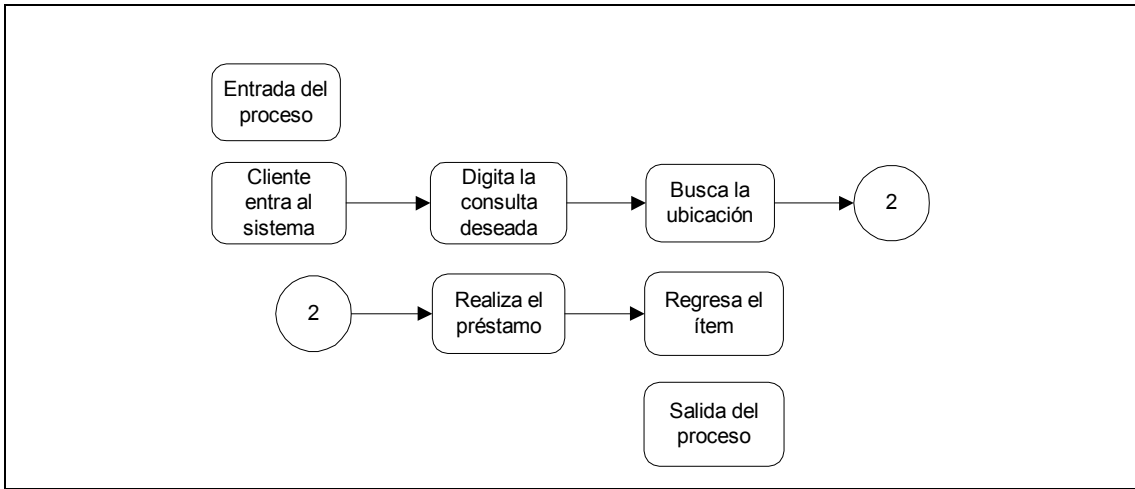


Figura 5. Diagrama de alto nivel.

- *Diagrama de alto nivel con pasos intermedios.* Constan de los mismos elementos del nivel alto, agregando los pasos intermedios que se llevan a cabo en cada departamento.

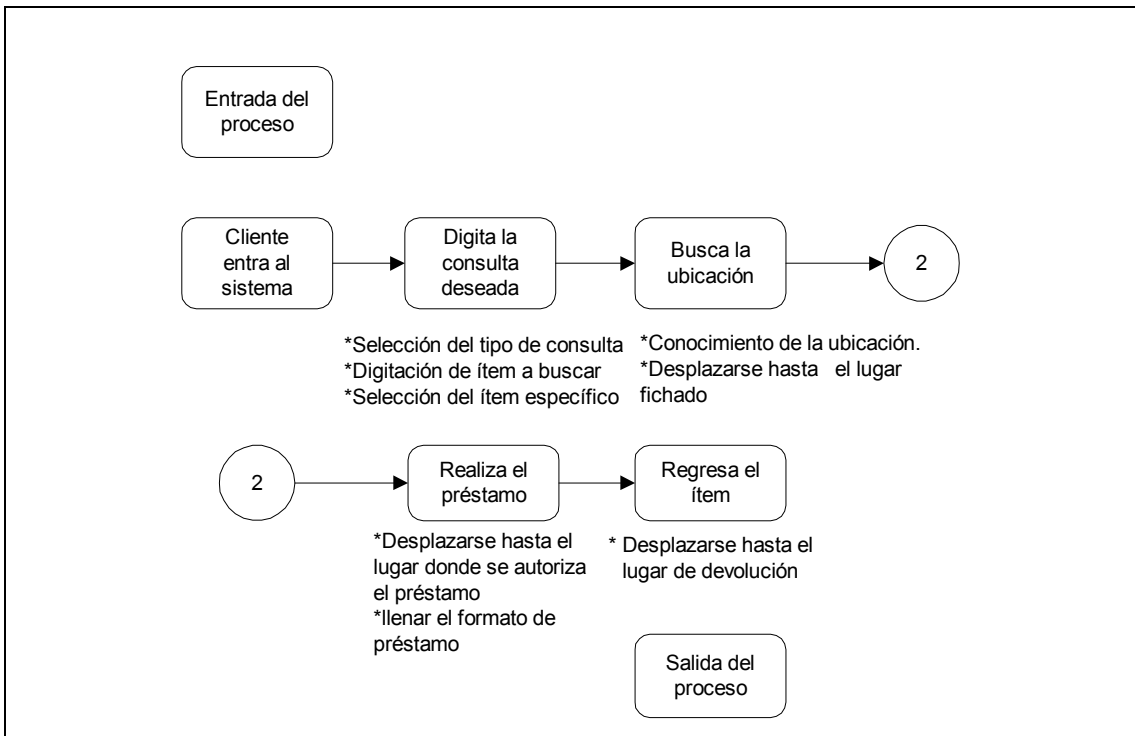


Figura 6. Diagrama de alto nivel con pasos intermedios.

- *Diagrama de nivel de detalle.* En este nivel el mapa contiene los pasos en una forma mucho más específica, incluyendo los pasos intermedios, y bloques de decisión.

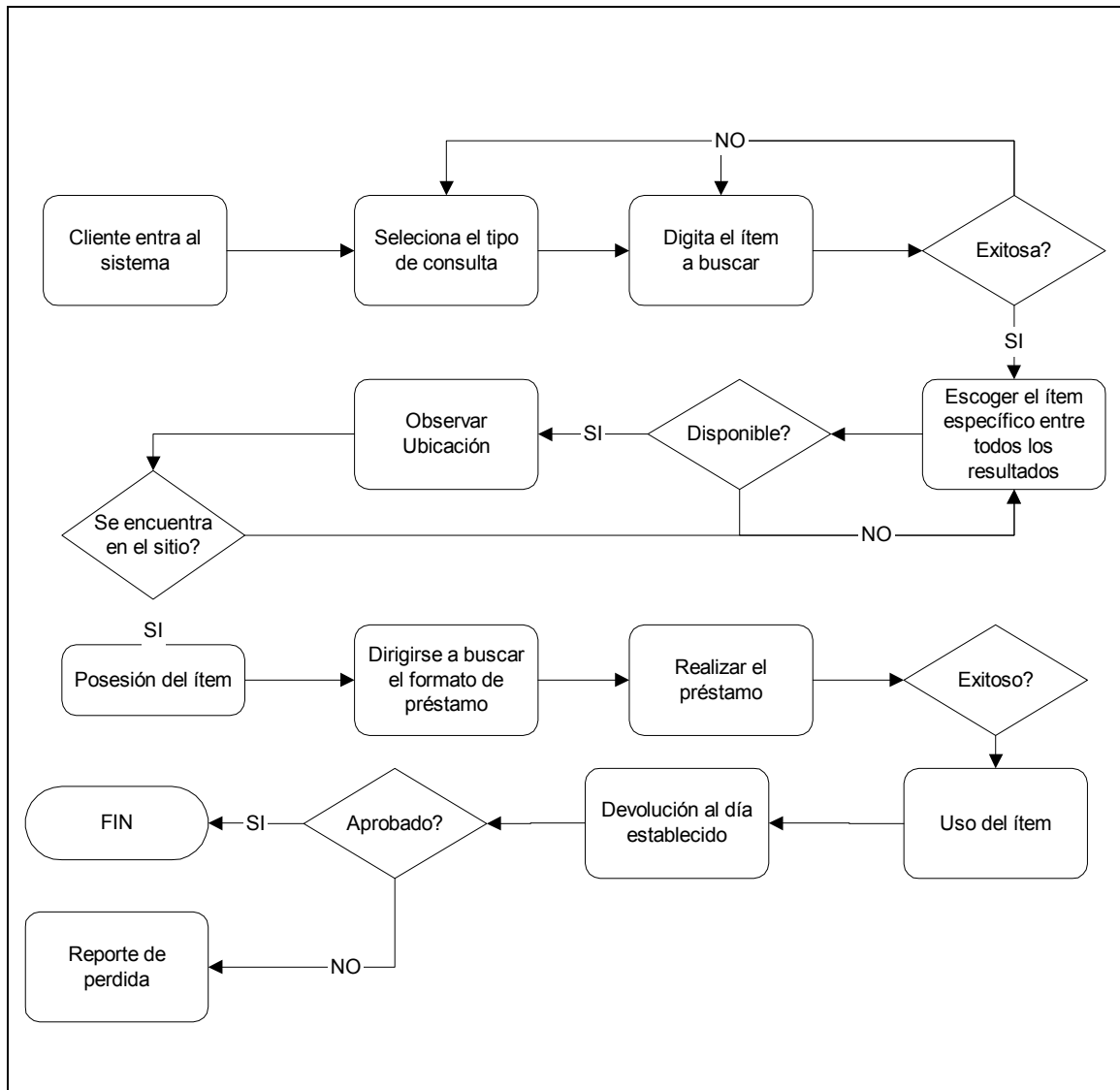


Figura 7. Diagrama de nivel de detalle.

- *Diagrama de nivel funcional.* La principal diferencia entre el nivel funcional y el nivel de detalle, es que el nivel funcional hace claridad en qué pasos del proceso se llevan a cabo en que departamento, mostrando quienes son los responsables de la entrega y resultado del producto final

o servicio. Ayuda a clarificar los límites del proceso y las dependencias entre departamentos.

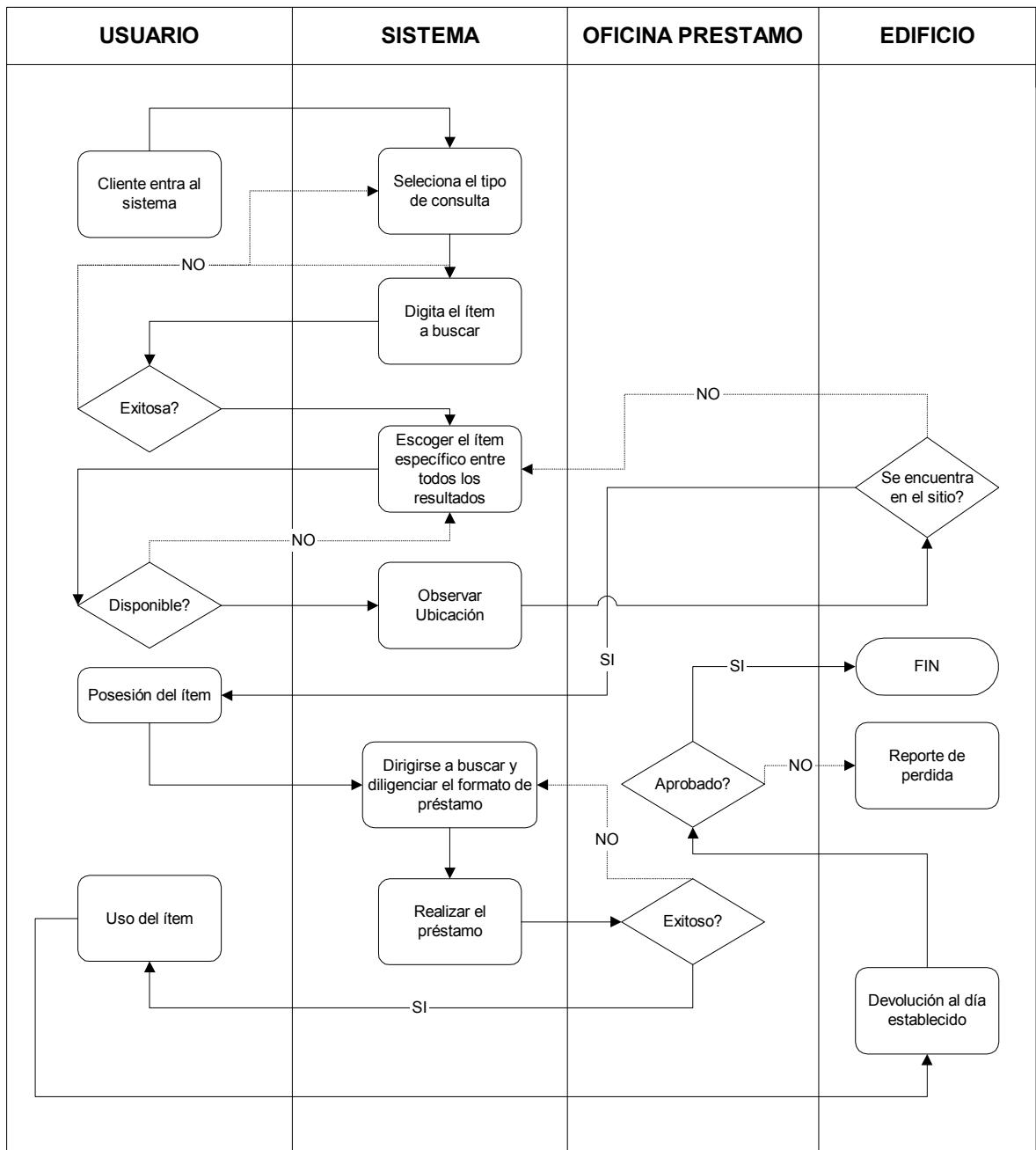


Figura 8. Diagrama de nivel funcional.

Desarrollar los diagramas de procesos, se convierte en una tarea que puede llegar a necesitar de mucho tiempo, análisis y estudio. Una mayor cantidad de procesos identificados y representados, servirán de soporte para las decisiones en su estrategia a futuro.

4. FASE MEDIR

Una vez el proyecto esta claramente definido, la segunda fase del modelo DMADDV es *MEDIR*. El objetivo es principalmente cuantificar el problema que se va a solucionar, de tal forma que se pueda obtener información del proceso actual y se puedan establecer parámetros válidos y confiables para monitorear el avance hacia las metas definidas en el paso anterior.

Gracias a los procesos internos cruciales que influyen en las medidas CTQ, se pueden medir los defectos generados en el proceso, de tal forma que si el cliente espera un cierto estándar, es necesario identificar los factores vitales que afectan esa expectativa, de tal forma podrá medirse el impacto de los defectos en esas áreas y que tanto afectan estos valores a la satisfacción del cliente, para esto es importante considerar los siguientes ítems:

- Sólo el 4% de todos los clientes con problemas se quejan.
- En promedio una persona con un problema se lo comunica habitualmente a otras 9 personas.
- Los pacientes y los clientes satisfechos le cuentan a otras 5 personas sobre la buena atención que han recibido.
- El coste para conseguir un cliente nuevo es generalmente de 5 a 7 veces mayor que el de mantener los clientes actuales.
- El coste de contratar y formar un empleado nuevo es hasta 10 veces más grande que el de mantener a los actuales.

Estos hechos resaltan la necesidad de satisfacer a sus clientes actuales para que permanezcan en su organización. Una excelente satisfacción del cliente es una de las pocas formas de lograr una ventaja competitiva sostenible.

Las estadísticas son pieza fundamental de Seis Sigma, ya que brindan un análisis, y un resultado de datos en el cual el equipo se puede fundamentar, utilizando la variación como los defectos ocurridos en el proceso. Las distribuciones existentes que arrojan los datos necesarios para identificar los defectos en el pasado, y predecirlos en el futuro, son varias, la distribución más utilizada en Seis Sigma, es la distribución normal. Las otras distribuciones se pueden utilizar dependiendo directamente si el proyecto lo requiere.

DISTRIBUCIÓN NORMAL

La distribución normal es una clase muy importante de las distribuciones estadísticas, se justifica en la frecuencia o normalidad con la que ciertos fenómenos tienden a parecerse en su comportamiento a esta distribución. Las distribuciones normales son simétricas y su curva es de forma campana con solo un pico (unimodal), tal como se observa en la figura 8.

Posee dos elementos que la afectan directamente, la media y la desviación estándar, la media se refiere al pico de la curva, donde se concentran la mayoría de los datos, y la desviación estándar es donde ocurre la dispersión de la campana. El símbolo utilizado para identificar la media es μ y la dispersión σ .

La importancia de la distribución normal se debe principalmente a que hay muchas variables asociadas a fenómenos naturales que siguen el modelo de la normal, por citar algunos tenemos.

- Caracteres morfológicos de individuos (personas, animales, plantas,...) de una especie, por ejemplo: tallas, pesos, envergaduras, diámetros, perímetros.
- Caracteres fisiológicos, por ejemplo: medicamentos.
- Caracteres sociológicos, por ejemplo: consumo de cierto producto por un mismo grupo de individuos, puntuaciones de examen.
- Caracteres psicológicos, por ejemplo: cociente intelectual, grado de adaptación a un medio.
- Errores cometidos al medir ciertas magnitudes.
- Valores estadísticos muestrales, por ejemplo: la media.
- Otras distribuciones como la binomial o la de Poisson son aproximaciones normales.

Debido a la simetría, y basándonos en el teorema de límite central, podemos determinar que cualquier variable aleatoria puede encontrarse bajo la curva de una distribución normal.

El **teorema del límite central** dice que si se tiene un grupo numeroso de variables independientes y todas ellas siguen el mismo modelo de distribución (cualquiera que éste sea), la suma de ellas se distribuye según una **distribución normal**.

Este teorema se aplica tanto a suma de variables discretas como de variables continuas.

Los parámetros de la distribución normal son:

Media : $n * \sigma$. (media de la variable individual multiplicada por el número de variables independientes).

Varianza : $n * \sigma_i^2$ (varianza de la variable individual multiplicada por el número de variables individuales).

En la figura 8 se puede visualizar las probabilidades asociadas a una distribución normal.

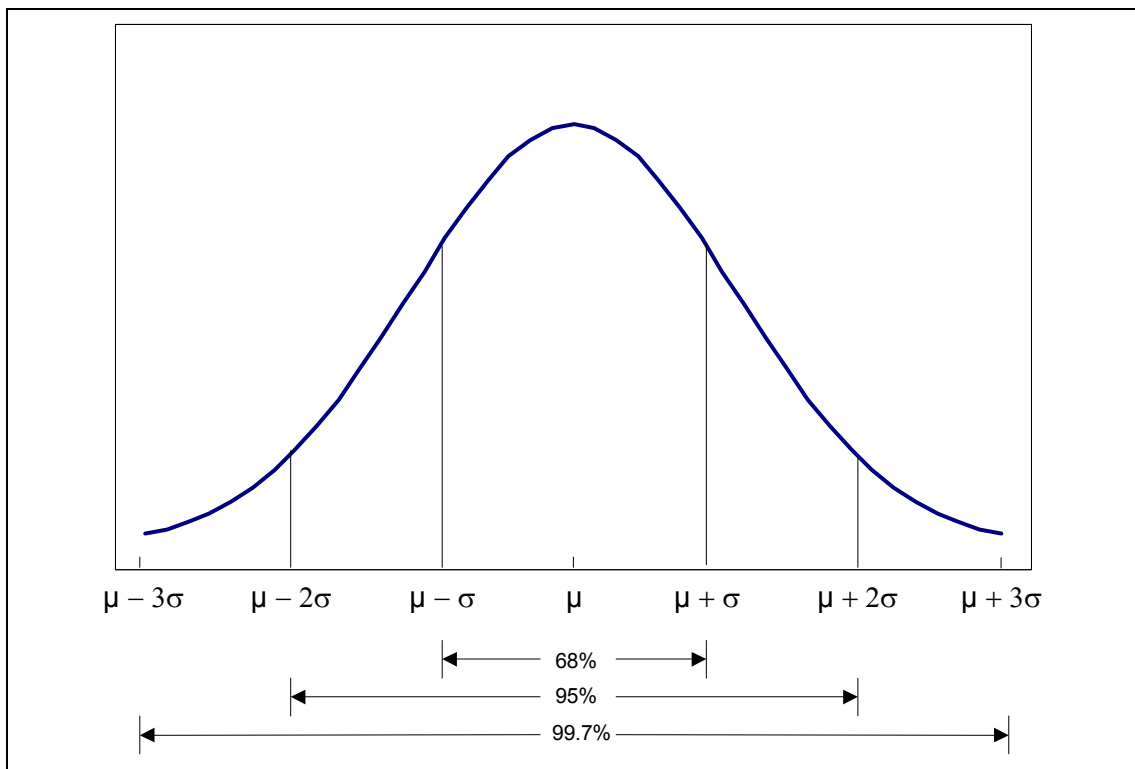


Figura 8. Probabilidades asociadas a una distribución normal.

Según la figura 8, se pueden clasificar los datos como:

- 50% del área, puede ser encontrado debajo de la mitad.
- 68,26% (34% a cada lado) se encuentra con mas o menos una desviación estándar.
- 95.44% puede ser encontrado con mas o menos dos desviaciones estándar.

- 99.74% puede ser encontrado con tres desviaciones estándar.
- 99,9937% cuatro desviaciones.
- 99,999943% cinco desviaciones.
- 99,99999998% seis desviaciones (meta de Seis Sigma).

Debido a que más del 0.9973 de la probabilidad de una distribución normal esta comprometida en el intervalo ($\mu - 3\sigma$, $\mu + 3\sigma$), a menudo se hace referencia a la cantidad 6σ como el ancho de una distribución normal.

Los pasos necesarios para llevar con éxito la fase de medir son:

1. Capturar la voz del cliente.
2. Determinar las métricas.
3. Plan de colección de datos.
4. Elaborar las gráficas de las medidas.
5. Calcular el nivel de sigma.
6. Benchmarking.
7. Desarrollar el modelo financiero.
8. Identificar y priorizar los requerimientos de los clientes.
9. Inicio de QFD.
10. Plan de comunicaciones.
11. Presentar al comité para aprobar.

4.1 CAPTURAR LA VOZ DEL CLIENTE (VOC)

La voz del cliente es uno de los pasos en los que Seis Sigma se diferencia de las otras metodologías de desarrollo de software, debido a la importancia que se le otorga al cliente durante todo el desarrollo del proyecto.

Existen diferentes mecanismos para realizarla (Ver anexo B) y cualquiera que sea utilizado debe mostrar datos lógicos y coherentes y ofrecer a los miembros del equipo la información correcta para tomar decisiones efectivas.

Tomar decisiones no basadas en hechos puede llegar a tener un costo muy alto que ninguna compañía esta dispuesta a correr, aunque algunas veces los proyectos deben correr el riesgo de asumir diferentes aspectos, la VOC ayuda a reducirlos.

4.2 DETERMINAR QUE MEDIR

Como Seis Sigma tiene sus fundamentos en el análisis estadístico, es importante hacer gran énfasis en las medidas. Uno de los principios de Seis Sigma es que las medidas, la precisión y exactitud del análisis estadístico y de las formulaciones matemáticas son la clave del éxito.

La ecuación $y = f(x)$ es una explicación matemática de la relación entre las entradas, las salidas y el proceso. En la ecuación “y” representa la salida final del proceso, es una función de las “x’s”, las variables o las entradas claves del proceso y “f” el proceso como tal. En resumen, las entradas son procesadas y se convierten en las salidas finales.

Aunque algunas veces no es tan simple, suelen existir varios pasos dentro del proceso y la ecuación se puede convertir en $y = f(x_1, x_2, x_3, \dots)$, donde las “x’s” representan las diferentes entradas en el proceso.

Durante el proceso, existe la posibilidad de encontrar defectos en cada paso que lo compone, la meta es eliminar los defectos tan pronto como sea posible. Para poder hacer esto, es necesario conocer las causas de la variación en cada paso, (o de las entradas), Para conocer las causas de la

variación es necesario medir el grado de variación, pero el primer paso es identificar las entradas y decidir que es lo que se va a medir.

La pregunta es ¿Cómo seleccionar de verdad unas métricas adecuadas?. Seleccionar solamente las medidas óptimas de rendimiento significa equilibrar dos elementos principales: 1) lo que es *factible*; 2) lo que es más útil o *valioso*.

La tabla 6 a continuación contiene una lista parcial de criterios a considerar, en las categorías de viabilidad y valor, cuando elija qué medir.

Tabla 6. Criterios de selección de medidas.

Valor / utilidad	Viabilidad
<ul style="list-style-type: none"> ▪ Enlaza con los requisitos prioritarios del cliente. ▪ Exactitud de los datos. ▪ Área de preocupación u oportunidad potencial. ▪ Se puede comparar con la de otras organizaciones. ▪ Permite obtener medidas de manera continuada. 	<ul style="list-style-type: none"> ▪ Disponibilidad de los datos. ▪ Plazo de entrega requerido. ▪ Coste de obtención de los datos. ▪ Complejidad. ▪ Posible resistencia o “factor miedo”.

4.2.1 Tipos de variación

Aunque el equipo de trabajo identifica las ordenes del cliente como la entrada principal a los procesos, también es necesario considerar todo lo que genere impacto en el proceso como una posible entrada o variable. Para identificar variables es muy útil comenzar con el método de las “1 P y 5 M’s”: persona, maquina, material, métodos, medidas y madre naturaleza.

La tabla 7 cuadro provee una definición del método:

Tabla 7. Método de 1 P y 5 M's.

Elemento	Explicación
Persona	Es el elemento humano, las diferencias que ocurren cuando más de una persona opera la misma parte del proceso.
Máquina	Son las clasificaciones de las piezas que conforman un equipo para trabajar.
Material	Son la materia prima o ingredientes .
Método	Son los procedimientos estándares para operar bien. Las variaciones son causadas cuando existe más de una forma diferente de realizar el proceso.
Medidas	El sistema de medidas puede arrojar resultados que realmente no son variaciones. Esto convierte a las medidas en un elemento de variación.
Madre naturaleza	Factores ambientales, incluyen temperatura, humedad y energía disponible.

Además de asignar las variables a un elemento, también es necesario mencionar que las variables se clasifican en *controlables* e *incontrolables* y a su vez las variables controlables se subdividen en *críticas* y *no-críticas*.

Variables controlables, son aquellas que las personas que miden el proceso pueden variar para determinar los efectos en las salidas. Variables críticas son las entradas controlables que tienen gran impacto en la salida.

Las variables incontrolables son aquellas que afectan la salida pero son difíciles de controlar.

4.2.2 Medir lo que tenga valor

Otro de los principios de Seis Sigma es medir lo que tenga valor. Al igual que los requerimientos pueden ser evaluados por el método EMART, existe un acrónimo que puede ser utilizado para medir las características de unas buenas medidas.

Las medidas pueden ser de tipo RAVF²¹:

Relevantes.

Adecuadas para detectar cambios en los procesos.

Valido y consistente a través del tiempo.

Fáciles.

4.2.3 Exactitud de las medidas

Las medidas son usadas para la toma de decisiones, por tal motivo es necesario hacer énfasis en tomar medidas que sean exactas. Es importante conocer que existen riesgos asociados a los sistemas de medidas, por ejemplo, la tabla 8 que se muestra a continuación.

Tabla 8. Categoría de los riesgos en las medidas.

	Alpha	Beta
Riesgo	Se rechazan buenos elementos.	Se aceptan malos elementos.
Grupo del riesgo	Proveedor.	Cliente.
Efecto en el proceso	Puede cambiar innecesariamente.	No se identifican los cambios que se necesitan.

²¹ RAVE: acrónimo de Relevant, Adequate to detect process changes, Valid and consistent from time to time, Easy.

Otros efectos	Aumento de costos innecesariamente.	Se crean defectos.
---------------	-------------------------------------	--------------------

4.3 PLAN DE COLECCIÓN DE DATOS

En proyectos de desarrollo de software, los datos son fundamentales para el equipo de trabajo. La colección de datos al igual que otros pasos dentro de la metodología, debe hacerse bajo una buena planeación, para asegurarse que los procesos y las medidas sean estables, confiables, y se puedan utilizar como soporte en la fase analizar.

Es importante destacar dos secciones principalmente. La primera donde se definen las metas y los objetivos de la colección de datos, y la segunda donde se desarrollan las definiciones operativas y los procedimientos.

4.3.1 Sección 1: definir las metas y los objetivos

En esta sección se debe definir claramente las metas y los objetivos de la colección de datos. Un buen plan de colección de datos debe incluir:

- Una breve descripción del proyecto.
- Los datos específicos que se necesitan (Nombre, Descripción).
- Las razones para recoger los datos.
- Qué información brindan esos datos al proceso estudiado y cómo ayudarán al equipo de trabajo.
- Qué se puede hacer con los datos una vez sean recogidos.

4.3.2 Sección 2: desarrollar las definiciones operativas y los procedimientos.

En esta sección se pueden desarrollar los siguientes pasos:

- Desarrollar las definiciones operativas.
 - Avanzar en el proceso de la colección de datos.
 - Comprobar la precisión y el valor de la medida.
 - Trabajar con los resultados.
-
- *Desarrollar las definiciones operativas.*

El equipo de trabajo debe hacer una descripción clara, comprensible y concreta de lo que hay que medir u observar, y como se debe hacer, para que todos los miembros del equipo de trabajo puedan operar o medir de forma coherente con base a la definición.

La fiabilidad e integridad de los datos es un factor importante de este punto, para encontrarla, los miembros del equipo deben hablar un mismo vocabulario, el cual ya ha sido establecido en la fase definir. Pasar por alto este paso puede terminar en resultados engañosos si los miembros del equipo los interpretan libremente y los definen en términos diferentes al momento de recoger datos. Los problemas serios pueden presentarse para la organización cuando se toman las decisiones económicas basado en estos datos potencialmente no fiables.

Existen diferentes fuentes de datos posibles en la organización. Es necesario garantizar que la fuente elegida, tenga datos precisos y que representen el proceso que se quiere medir. Si el equipo desea examinar datos históricos para incluirlos como parte del estudio, se debe prestar atención a que tan confiables han sido los datos y su fuente, y si es

recomendable continuar usando esos datos. Los datos que demuestran ser sospechados deben ser desechados.

- *Avanzar en el proceso de la colección de datos.*

Implica tener claridad en los procedimientos que se van a realizar para la recolección de los datos, el personal asignado a la toma de las medidas así como la fuente o la ubicación de los datos.

Una vez que se haya planeado y definido el proceso de colección de datos, se debe hacer un seguimiento a través del proceso, desde su comienzo hasta el final, para asegurarse que el plan se está ejecutando constantemente y exactamente. Si se asume que el Black Belt o el líder del equipo ha comunicado a los responsables de reunir los datos y los participantes de este proceso, cuáles son los datos que se deben reunir, posiblemente necesitarán una preparación adicional sobre las definiciones, procedimientos, pautas, etc.

Es buena idea que el Black Belt o el líder del proyecto estén presentes en el proceso de colección de datos, de esta manera los participantes sabrán enseguida si el plan se está ejecutando correctamente. La falta de supervisar el proceso desde las etapas iniciales, puede significar que más tarde una corrección necesite ser hecha, y muchos de los esfuerzos iniciales se perderán.

- *Comprobar la precisión y el valor de la medida.*

Existen varias formas de comprobar si las medidas son exactas y se mantienen precisas. La prueba más común para la efectividad de un sistema

de medida es el "Estudio de R&R"²², que consiste en repetir una medida en varios escenarios para comprobar cuatro importantes criterios: Precisión, Repetibilidad, Reproducibilidad y Estabilidad.

Los datos que son recogidos (y medidos) serán repetibles si el mismo equipo puede alcanzar esencialmente los mismos tiempos del resultado en un artículo particular cuando se mide más de una vez. Los datos serán reproducibles si otras personas o equipos que están midiendo los mismos artículos están alcanzando esencialmente los mismos resultados. El grado en las cuales el sistema de medidas es Preciso, es cuando no se encuentran diferencias entre una medida de media observada y el valor estándar asociado. El grado en el cual el sistema de la medida es estable, es expresado generalmente por la no variación resultado de medir el mismo artículo, con el mismo equipo, sobre un período extendido.

Los equipos de trabajo necesitan reconocer todos los factores posibles que causarían reducciones en la capacidad de repetición, la reproductibilidad, la precisión y la estabilidad, sobre cualquier longitud del tiempo y que alternadamente pueda rendir datos no fiables. Es buena práctica probar, quizás en una escala pequeña, cómo procederán la colección y las medidas de datos.

- *Trabajar con los resultados*

Las medidas contienen las características de Repetibilidad, reproducibilidad, Precisión y Estabilidad, el Black Belt o el líder del proyecto debe comprobar que los datos sean razonables y necesarios, de no ser así, ellos son los encargados de determinar donde ocurrió la falla, y de determinar el procedimiento a seguir con los datos erróneos. Se debe tener en cuenta las

²² R&R: Repetibilidad y reproducibilidad.

definiciones operacionales para resolver los malentendidos y las interpretaciones que pudieron haber causado los errores.

4.4 ELABORAR LAS GRÁFICAS DE LAS MEDIDAS

La variación que exista dentro de un proceso es vista como defecto, identificarla es una tarea indispensable dentro de esta fase con el propósito de eliminarla.

Los gráficos más utilizados en proyectos Seis Sigma son:

4.4.1 Diagrama de cajas

El diagrama de cajas representa varias características importantes sobre un conjunto de datos, tales como: la dispersión, la desviación de la simetría, la media, el rango, entre otros.

En la figura 9, obtenido con el paquete statgraphics 5.1, para los datos ejemplo de la tabla 9, se identifica la línea vertical que representa la media y la línea horizontal que se extiende por el gráfico representando el rango de opciones existentes.

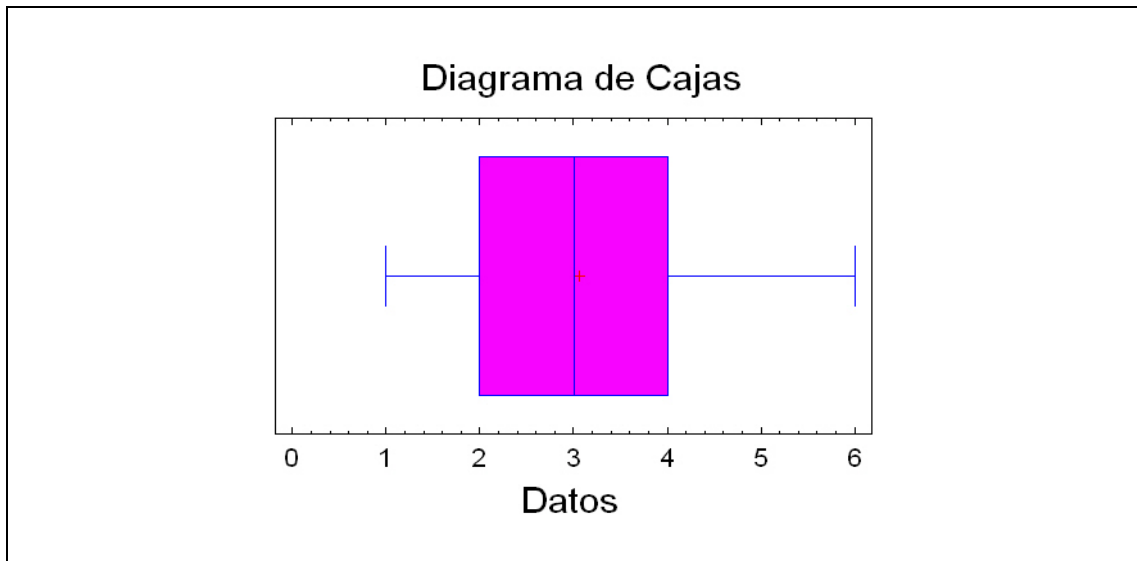


Figura 9. Diagrama de cajas sencillo.

Los diagramas son útiles cuando se necesitan hacer comparación sobre los datos obtenidos discriminados por grupos de personas o de fechas u otros como se observa en la figura 10. La desventaja es que no se conoce la cantidad de valores que se encuentran en cada posible valor, aunque se tiene la seguridad que alguna(s) observación(es) arrojaron ese valor.

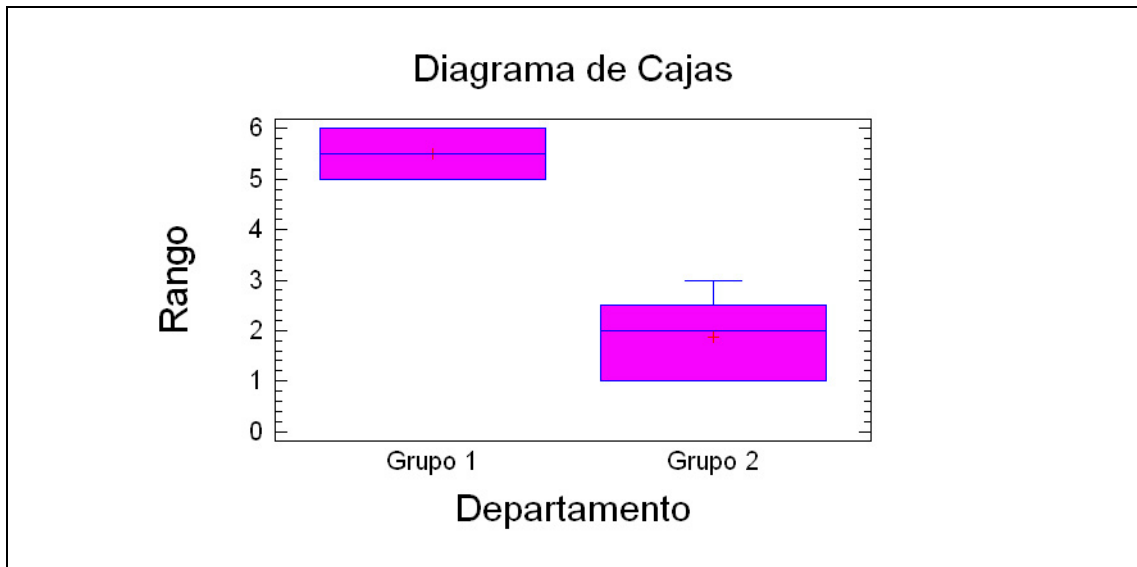


Figura 10. Diagrama de cajas avanzado.

Tabla 9. Datos ejemplo.

DATOS							
Grupo	Resultado	Grupo	Resultado	Grupo	Resultado	Grupo	Resultado
4	3	3	2	3	2	3	6
3	3	4	4	3	2	1	5
4	4	5	1	2	3	1	6
5	2	4	2	4	1	1	5
4	3	5	4	5	1	3	6
2	1	5	2	5	2	4	2
4	3	2	2	2	2	4	1
3	4	3	1	5	3	4	2
5	3	3	2	4	1	2	2
5	3	3	4	4	2	5	3
5	3	3	3	2	3	2	3
4	4	5	3	4	3	3	2
4	3	5	4	3	2	3	2
3	3	4	4	3	4	3	1
1	5	1	6	4	4	4	2
1	6	1	6	5	4	5	3
1	5	4	2	3	4	5	4
1	6	4	2	3	4		
2	1	5	3	5	4		
5	3	2	1	4	4		

4.4.2 Diagrama de punto

Es utilizado en conjuntos pequeños de datos. Las ventajas más sobresalientes son permitir ver con facilidad la ubicación o tendencia central de los datos, así como su dispersión o variabilidad. Elimina la dificultad de no tener conocimiento de los valores exactos en cada opción encontrada en el diagrama de cajas. La desventaja más sobresaliente se presenta en la dificultad de contar el número de puntos. Por otro lado, la media no es identificada fácilmente.

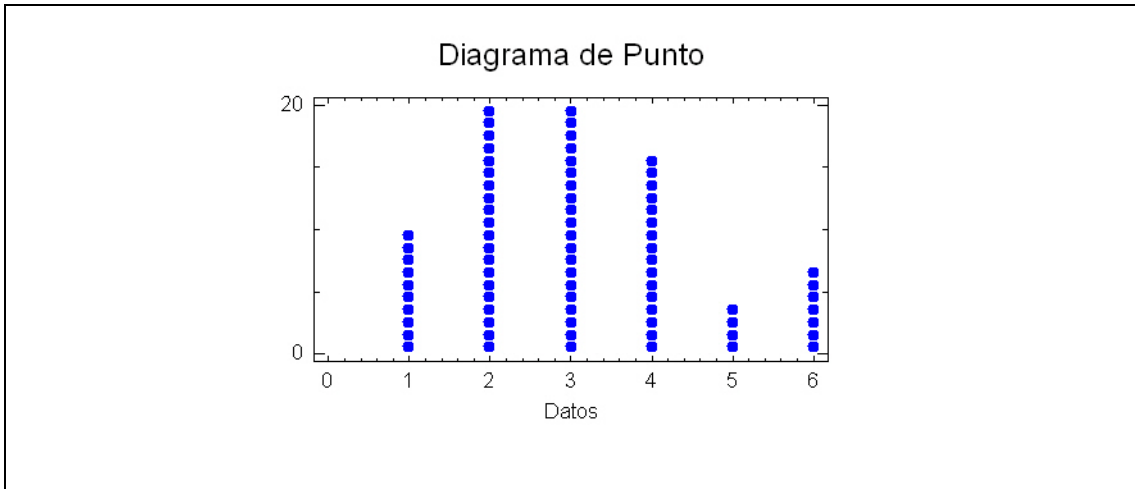


Figura 11. Diagrama de punto.

4.4.3 Diagrama de torta.

El diagrama de torta presenta la ventaja de mostrar claramente la cantidad de muestras obtenidas en una opción. Exponiendo también los porcentajes, y diferenciados por diferentes colores. Una de las desventajas es que no muestra la media de los datos.

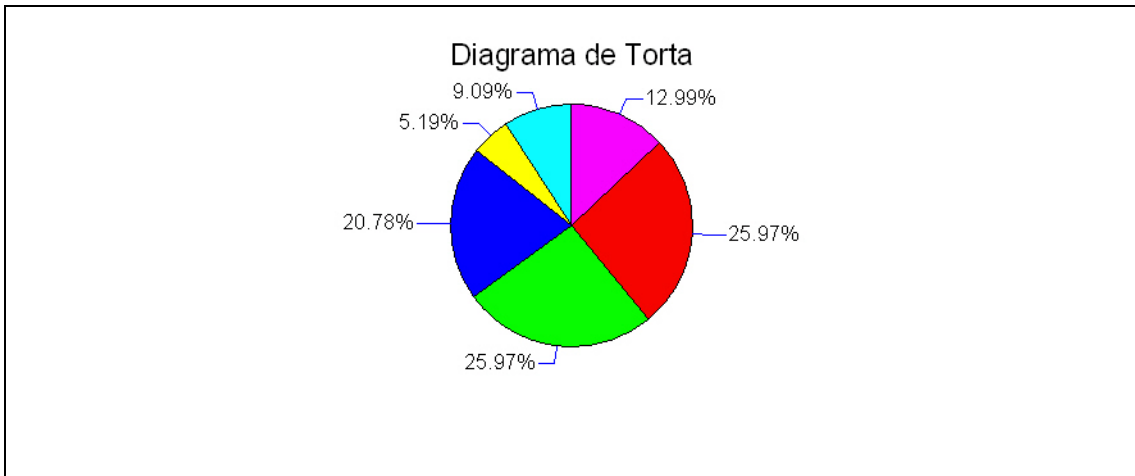


Figura 12. Diagrama de torta.

4.4.4 Histograma con curva normal

El histograma proporciona una impresión visual del aspecto que tiene la distribución de las mediciones, así como información sobre la dispersión de los datos. Es el diagrama más utilizado por la estadística, y agregándole la curva normal, se obtienen mayores ventajas.

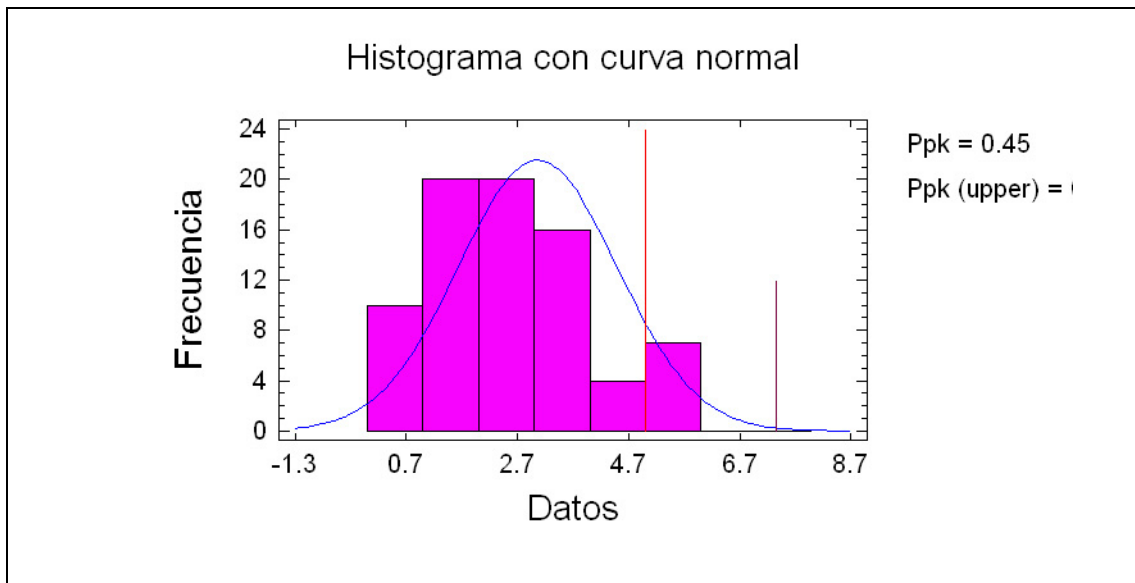


Figura 13. Diagrama de histograma.

4.4.5 Diagrama de Series de Tiempo

Es un diagrama en el que el eje vertical denota el valor observado, mientras que el eje horizontal denota el tiempo. Una de las ventajas es que a menudo se observan ciertas características importantes de los datos que generalmente pasan inadvertidas. Las desventajas más sobresalientes es la no identificación de la media.

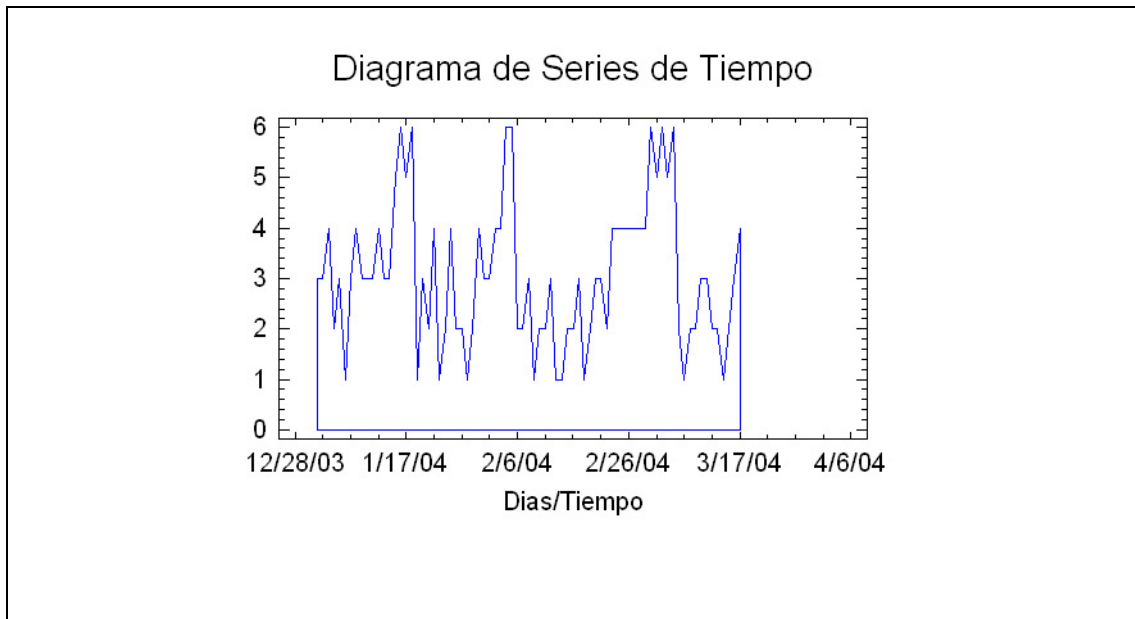


Figura 14. Diagrama de series de tiempo.

4.4.6 Diagrama de Corrido

El diagrama tiene las mismas características del diagrama de series de tiempo, más la representación de la media y algunos datos estadísticos básicos. Es otra manera útil de examinar la variabilidad en datos que dependen del tiempo.

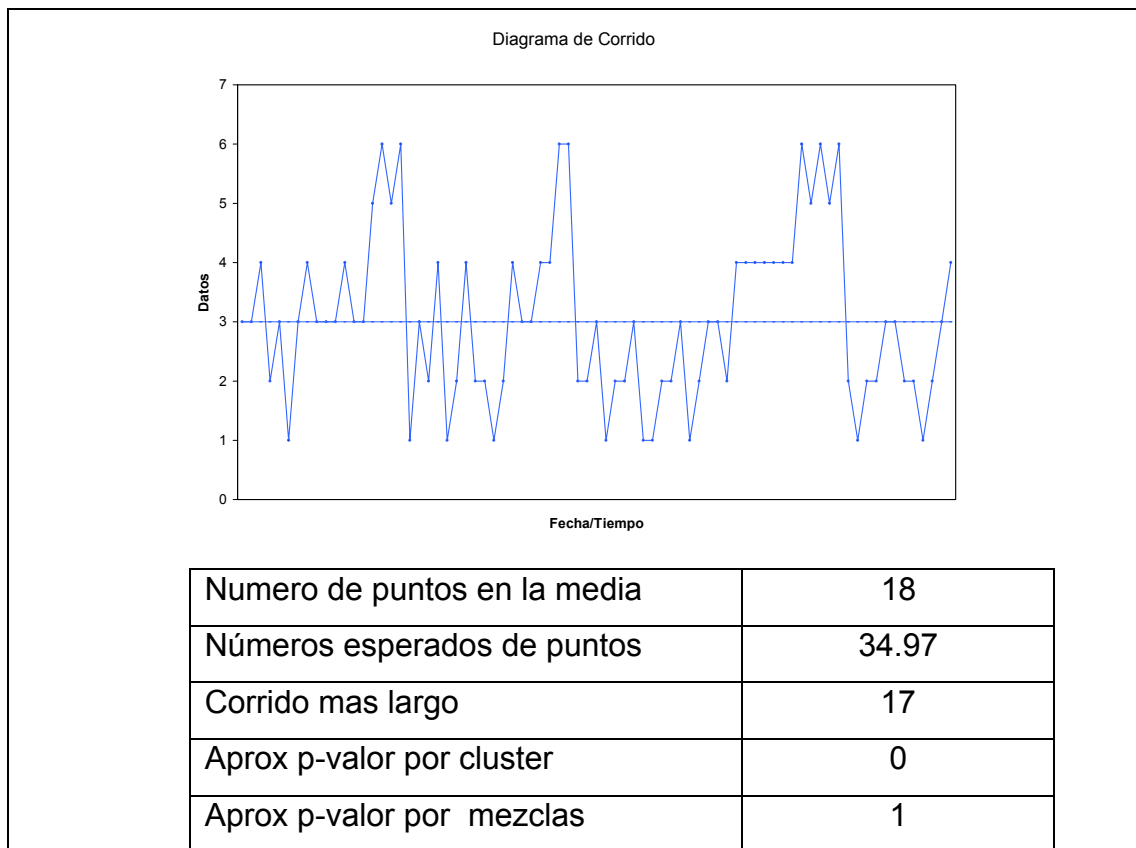


Figura 15. Diagrama de corrido.

4.4.7 Diagrama de dispersión

Es un diagrama en el que cada par (x, y) esta representado con un punto en un sistema de coordenadas bidimensionales. El análisis de este diagrama indica que si una curva no pasa exactamente por todos los puntos, existe una evidencia fuerte, de que los puntos están dispersos de manera aleatoria alrededor de una línea recta.

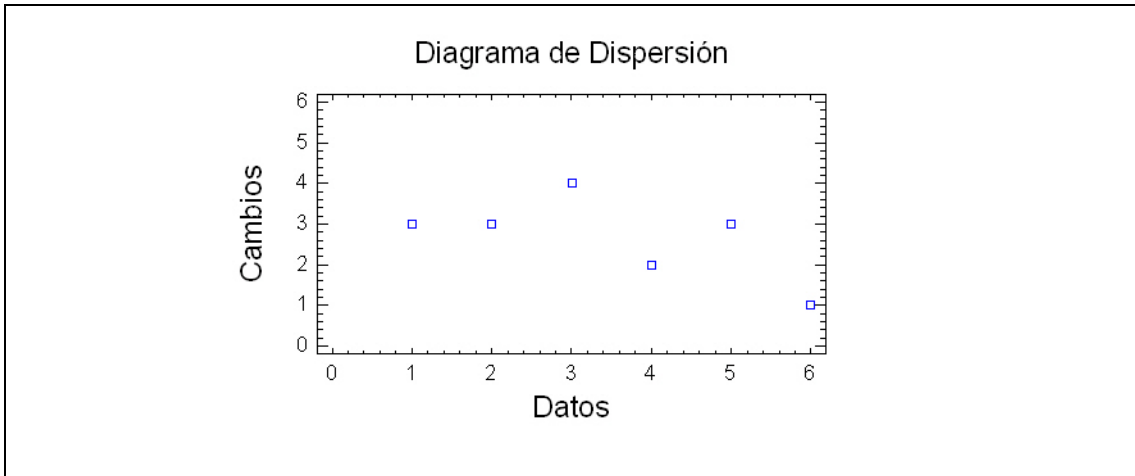


Figura 16. Diagrama de dispersión.

4.4.8 Diagrama de Pareto

Los diagramas de Pareto son herramientas poderosas para enfocar los proyectos en la dirección correcta. Basados en el principio de Pareto, donde solo pequeños factores trascendentales son los responsables de producir la mayoría de problemas. En otras palabras, el 80% de los problemas se derivan del 20% de las causas, los diagramas de Pareto separan los factores y los representan en orden descendente desde los más, hasta los menos problemáticos.

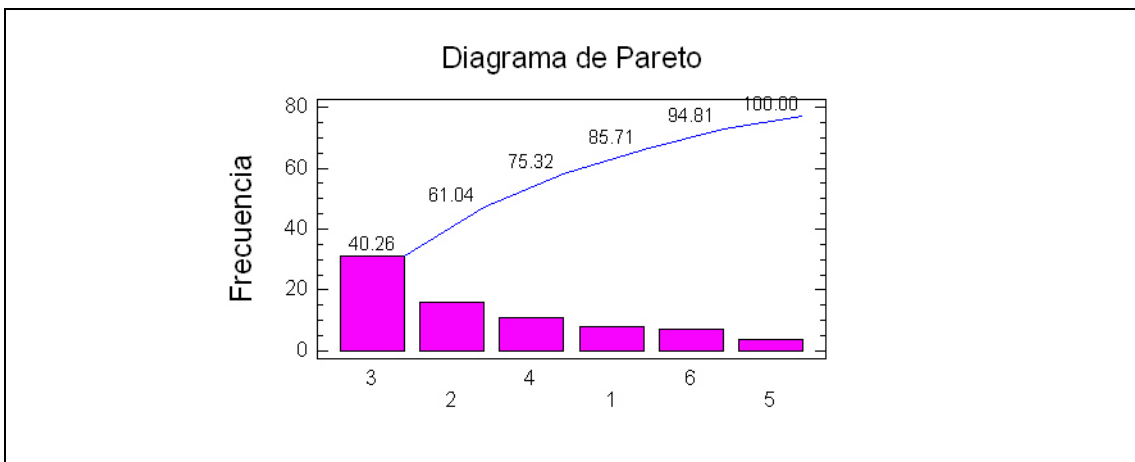


Figura 17. Diagrama de Pareto.

4.5 CALCULAR EL NIVEL SIGMA

Para calcular el nivel de sigma, se debe tener claridad en las siguientes definiciones:

- *Unidad*. Es el resultado del servicio prestado, o del producto final.
- *Defecto*. Es el fallo al no encontrar las especificaciones establecidas por el cliente o estándares de rendimiento.
- *Defectuoso*. Unidad con uno o varios defectos. Son identificados al final del proceso.
- *Oportunidad de defecto*. Son las oportunidades que tiene una unidad en ser defectuosa.
- *Defectos por unidad (DPU)*. El número total de defectos en una muestra, dividido en el número total de unidades de la misma muestra.
- *Rendimiento final*. Es la cantidad de unidades sin defectos.

Para determinar las oportunidades por defecto se debe tener perfectamente claro el método para obtener medidas basadas en oportunidades y luego como se expresan esas medidas. El reto del equipo está en identificar un número realista de oportunidades de defecto para cada producto o servicio. Determinar el número de oportunidades debe ser una responsabilidad ética del grupo, ya que el número de oportunidades es directamente proporcional al nivel de sigma. Los pasos a seguir son los siguientes:

1. Desarrollar una lista preliminar de tipos de defectos. ¿Cuántos tipos de defectos puede tener un servicio o producto?
2. Determinar cuáles son los defectos específicos reales, críticos para el cliente. Aquellos que más influyen en los requerimientos del cliente. La determinación debe tener características razonables, realistas, prácticas, y la más importante, consistentes.
3. Definir estándares para calcular el número de oportunidades.

Los servicios entre más complejos sean, más oportunidades tendrán, para estimar las oportunidades se establecen los siguientes pautas:

- Dirigirse a las áreas donde se generan los problemas o que mayor valor agregado dan al proceso.
- Encontrar la semejanza que poseen algunos defectos y agruparlos con el propósito de tener clasificaciones dentro de estos.
- Ser coherente.
- Estimar el número de oportunidades con cuidado; un número grande da un mejor nivel de sigma, pero es difícil medir los resultados al final del proyecto.
- Estar seguro de que el defecto si es realmente importante para el cliente o para el proceso.

Los pasos para calcular el nivel de sigma son los siguientes:

1. Calcular el número unidades. (NU)
Por ejemplo: 115
2. Calcular el número de unidades con defectos. (NUD)
Ej. 40
3. Calcular el número de oportunidades (NO)
Ej. 10
4. Calcular el defecto por unidad (DPU)

$$\text{DPU} = \frac{\text{Numero de Defectos (NUD)}}{\text{Numero de Unidades (NU)}}$$

$$\text{DPU} = \frac{40}{115} = 0,3478$$

5. Calcular el número de defectos por oportunidad (DPO)

$$\text{DPO} = \frac{\text{Defectos por Unidad (DPU)}}{\text{Numero de Oportunidades (NO)}}$$

$$\text{DPO} = \frac{0,3478}{10} = 0,03478$$

6. Calcular los defectos por millón de oportunidades (DPMO)

$$\text{DPMO} = \text{Defectos por Oportunidad (DPO)} \times 1'000.000 (10^6)$$

$$\text{DPMO} = 0,03478 \times 1'000.000 (10^6) = 34.783$$

7. Convertir el DPMO en el valor de Sigma establecido en tabla 10.

$$34783 = 3,30 \text{ Sigma}$$

Tabla 10. Conversión de DPMO a Nivel Sigma²³

Tabla de conversión DPMO con Sigma					
DPMO	Sigma	DPMO	Sigma	DPMO	Sigma
933,193	0.00	291,160	2.05	4,661	4.10
926,471	0.05	274,253	2.10	4,024	4.15
919,243	0.10	257,846	2.15	3,467	4.20
911,492	0.15	241,964	2.20	2,980	4.25
903,199	0.20	226,627	2.25	2,555	4.30
894,350	0.25	211,856	2.30	2,186	4.35
884,930	0.30	197,663	2.35	1,866	4.40
874,928	0.35	184,060	2.40	1,589	4.45
864,334	0.40	171,056	2.45	1,350	4.50
853,141	0.45	158,655	2.50	1,144	4.55
841,345	0.50	146,859	2.55	968	4.60
828,944	0.55	135,666	2.60	816	4.65
815,940	0.60	125,072	2.65	687	4.70
802,338	0.65	115,070	2.70	577	4.75
788,145	0.70	105,650	2.75	483	4.80
773,373	0.75	96,800	2.80	404	4.85
758,036	0.80	88,508	2.85	337	4.90
742,154	0.85	80,757	2.90	280	4.95
725,747	0.90	73,529	2.95	233	5.00
708,840	0.95	66,807	3.00	193	5.05
691,462	1.00	60,571	3.05	159	5.10
673,645	1.05	54,799	3.10	131	5.15
655,422	1.10	49,471	3.15	108	5.20
636,831	1.15	44,565	3.20	89	5.25
617,911	1.20	40,059	3.25	72	5.30
598,706	1.25	35,930	3.30	59	5.35
579,260	1.30	32,157	3.35	48	5.40
559,618	1.35	28,717	3.40	39	5.45
539,828	1.40	25,588	3.45	32	5.50
519,939	1.45	22,750	3.50	26	5.55
500,000	1.50	20,182	3.55	21	5.60
480,061	1.55	17,865	3.60	17	5.65
460,172	1.60	15,778	3.65	13	5.70
440,382	1.65	13,904	3.70	11	5.75
420,740	1.70	12,225	3.75	9	5.80
401,294	1.75	10,724	3.80	7	5.85
382,088	1.80	9,387	3.85	5	5.90
363,169	1.85	8,198	3.90	4	5.95
344,578	1.90	7,143	3.95	3	6.00
326,355	1.95	6,210	4.00		
308,537	2.00	5,386	4.05		

²³ Tomada del libro. "SIX SIGMA. The Breakthrough Management Strategy Revolutionizing the World's Top Corporations"

4.6 BENCHMARKING

La práctica del Benchmarking no es nueva realmente, lo que si es nuevo es el auge que empieza a tener. Es común escuchar en las organizaciones frases como: “Por qué deberíamos hacer nosotros Benchmarking?”, “Nosotros somos los mejores en nuestro negocio”. “Estamos solos. No podemos compararnos con otros en nuestro negocio”. Muchas de las compañías creen que son los mejores, debido a su larga trayectoria que les ha generado dicha reputación. Sin embargo, cuando se comienza con una iniciativa como Seis Sigma, y descubren su verdadero nivel de sigma entran en razón de que sucede realmente con la compañía.

Hoy en día la cantidad de información disponible para utilizar en la toma de decisiones ha aumentado notablemente, igual los competidores tienen acceso a esa misma información; por lo tanto compañías que sean capaces de transformar esos datos en conocimiento y logren utilizarlo para beneficio propio, tendrán ventajas sobre los competidores.

4.6.1 ¿Qué es Benchmarking?

“Benchmarking es una poderosa herramienta para ayudar a las organizaciones a comparar los procesos con sus competidores, y entender que hace a los productos o servicios de otras organizaciones superiores”.

La mayoría de las veces las compañías tienen conocimiento de quienes son sus competidores más fuertes en algún proceso específico, sea para un producto o servicio, por lo tanto es necesario conocer como nuestras fortalezas y limitantes afectan directamente la satisfacción del cliente y rentabilidad.

Los resultados obtenidos en la práctica del Benchmarking deben ser incorporados en las operaciones del día a día, consiguiendo con esto que las organizaciones piensen diferente acerca de cómo realizan su trabajo y en cómo se pueden solucionar los problemas.

La primera asignación que se debe plantear es que tipo de Benchmarking se utilizará:

Benchmarking Interno. Compara los procesos comunes entre diversas funciones con un competidor. Mide la precisión y la eficiencia de los procesos a través de los departamentos internos.

Benchmarking Competitivo. Observa directamente a los competidores, midiendo el nivel de fidelidad, y satisfacción de los clientes, y la porción del mercado que poseen. El Benchmarking competitivo ayuda a identificar los cambios y las preferencias en los clientes. Por otro lado revela cual es el valor que tienen las mercancías hacia ellos y en como piensan los clientes hacia la compañía.

Benchmarking funcional. Se enfoca en el proceso mismo, y en las organizaciones con procesos similares más allá de que se encuentren en otro negocio. Ayuda a identificar estrategias de producción, a calcular las horas promedio de trabajadores requeridos y a calcular el costo de retrabajo.

4.6.2 ¿Por qué Benchmarking en Seis Sigma?

El primer paso después de determinar que tipo de Benchmarking utilizar es identificar cuál es el campo o el porcentaje de defecto de un producto particular, servicio o transacción, que tenga gran valor para la organización. El segundo, revela el número de oportunidades para un defecto existente en algún producto, servicio o transacción. Basandose en la fórmula defectos

por oportunidad (DPO), la cual se utiliza para calcular el defecto por millón de oportunidades (DPMO), se concluye que el Benchmarking es pieza fundamental en Seis Sigma.

Debido a que el Benchmarking implica mostrar datos a los competidores, existe un código que se debe aplicar éticamente por los participantes. Este código contiene los siguientes aspectos.

- *Legalidad.* No entrar en discusiones o actos que podrían considerarse ilegales, entre la compañía y el socio.
- *Intercambio.* No preguntar a los competidores asuntos que la compañía no estaría dispuesta a responder.
- *Confidencialidad.* Utilizar la información de los competidores de la misma forma en la que la compañía deseara que la información fuese tratada. Toda información recibida no puede ser puesta al exterior sin previa autorización.
- *Uso de la información.* No utilizar la información del socio, para un uso diferente al que se haya acordado. En la mayoría de los casos se utiliza la información para el mejoramiento de procesos.
- *Contacto.* Iniciar el contacto a través de alguien designado por la compañía para el contacto. Respetar la cultura empresarial y trabajar dentro de las conductas previamente establecidas.
- *Preparación.* Demostrar eficiencia y eficacia con el contacto aprovechando la mayor parte del tiempo con el contacto. Demostrar profesionalismo y conocimiento del área.
- *Cumplimiento.* No realizar compromisos que no pueda realizar o guardar. Completar el trabajo asignado para cada uno de los participantes, como se acordó mutuamente.
- *Entendimiento.* Es la **regla de oro** del Benchmarking, tratar al socio y a la información del socio, como le gustaría a la compañía que fuera tratado y la información fuese utilizada.

4.7 DESARROLLAR EL MODELO FINANCIERO

Los proyectos que se desarrollan en cualquier compañía están acompañados de un presupuesto para su realización. El retorno de inversión es uno de los aspectos más importantes que el comité de proyectos utiliza para dar su aprobación.

El retorno de inversión para proyectos nuevos o de mejora por lo general es un porcentaje determinado por la alta dirección de la compañía, aunque puede ser establecido para un proyecto o área de trabajo específica. Por ejemplo, que cualquier proyecto en el área de IT debe tener un retorno de inversión mayor del 25%.

El modelo que se plantea es un pronóstico de beneficios en términos monetarios para el proyecto. La alta dirección de la compañía establece que tipos de proyectos deben presentar el modelo, basados en las características que la compañía determine. Por ejemplo, solo proyectos que tengan un presupuesto mayor a 20 millones de pesos de inversión debe presentar el modelo financiero.

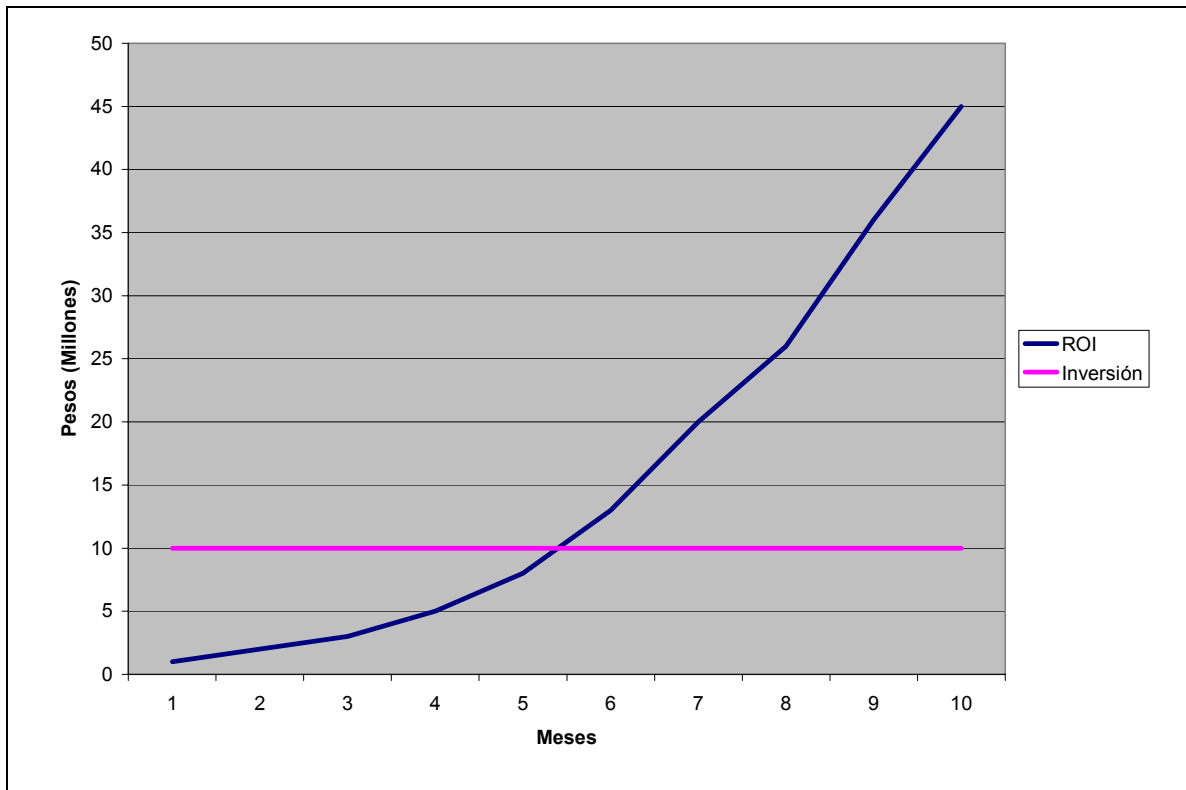


Figura 18. Retorno de inversión Vs tiempo.

El modelo debe ser coherente con la gran iniciativa de la compañía, aunque mostrar cifras de retorno de inversión altas y tiempo en periodos específicos es atractivo para la aprobación del proyecto y generará confianza para el equipo por parte de la alta dirección, es recomendable ser realista al analizar el tiempo de retorno o presentar una cifra menor a lo que realmente se puede alcanzar y que continúe siendo atractiva, si esta cifra es superada el impacto positivo que tendrá el comité y los otros miembros de la alta dirección hacia el equipo de trabajo será mayor.

4.8 IDENTIFICAR Y PRIORIZAR LOS REQUERIMIENTOS DE LOS CLIENTES

La captura de requisitos es uno de los pasos fundamentales y de mayor importancia en el proceso ya que definen las necesidades y las expectativas del cliente final.

En la mayoría de los casos la captura de requisitos suele ser compleja, muchas veces los usuarios no saben que parte de su trabajo puede transformarse en software, es decir, no saben cuáles son los requisitos ni tampoco cómo especificarlos de forma precisa. La voz del cliente se convierte en el mecanismo más recomendado para la captura.

En primer lugar es necesario establecer un estándar para que la definición de requisitos sea clara y concisa.

1. *Enlazar con un resultado específico:* No tendría sentido si el requisito no describe cuestiones relativas a un producto, servicio o evento específico.
2. *Describir un solo criterio o factor de rendimiento.* Debe quedar muy claro lo que el cliente busca o lo que va a evaluar ya sea rapidez, costo, etc., pero se debe evitar la tentación de agrupar varios factores.
3. *Expresarse mediante factores observables o medibles.* Si no puede observarse una forma de comprobar si el requisito se cumple o no, se concluye que este requisito es demasiado ambiguo. Cuando el requisito no es observable, se busca el mejor indicador para medir la solución.
4. *Permitir establecer un nivel de rendimiento “aceptable” o “no aceptable”.* El requisito debe ayudar a establecer la norma para el “defecto”.
5. *Ser detallada pero concisa.* Aunque uno de los mayores inconvenientes en la definición de requisitos es que resulte demasiado breve, al mismo tiempo si hay demasiadas palabras, nadie lo leerá. Lo correcto es lograr un equilibrio.

6. *Que coincida con la voz del cliente.* Lo más importante es que los requisitos o especificaciones deben ajustarse a las necesidades o expectativas del cliente.

Basándose en el modelo de Kano (figura 19) los requerimientos de los clientes se pueden clasificar en tres modalidades, que son:

- **Los que el cliente dice y desea.** Son aquellos que el cliente dice claramente, y que el servicio o producto debe cumplir para tener un nivel alto de calidad. Por ejemplo, cuando se ordena un café en una cafetería, el requisito es que el café este fresco y con la intensidad en que se ordenó, oscuro o claro.
- **Los que el cliente no dice y no generan satisfacción pero si insatisfacción.** Son aquellos que si el servicio o producto los cumple no genera absolutamente nada en la satisfacción de cliente, pero que si no los cumple generará una gran insatisfacción y hasta la pérdida del cliente, son ocultos pero están presentes. Para el ejemplo anterior, si el café se sirve caliente el cliente pasará por inadvertido ese hecho, pero si el café se sirve frío el cliente se sentirá insatisfecho y no regresará.
- **Los que el cliente no dice y que generan gran satisfacción en el cliente.** Son aquellos más difíciles de identificar, pero son los que hacen que un cliente regrese y se vuelva fiel a la compañía. En el ejemplo, si el café se sirve con galletas de chips de chocolate y varios barquillos para acompañar el café sin incrementar el costo, el cliente sentirá una satisfacción grande, si el café es servido sin acompañante, el cliente simplemente no sentirá nada y opinará acerca del café.

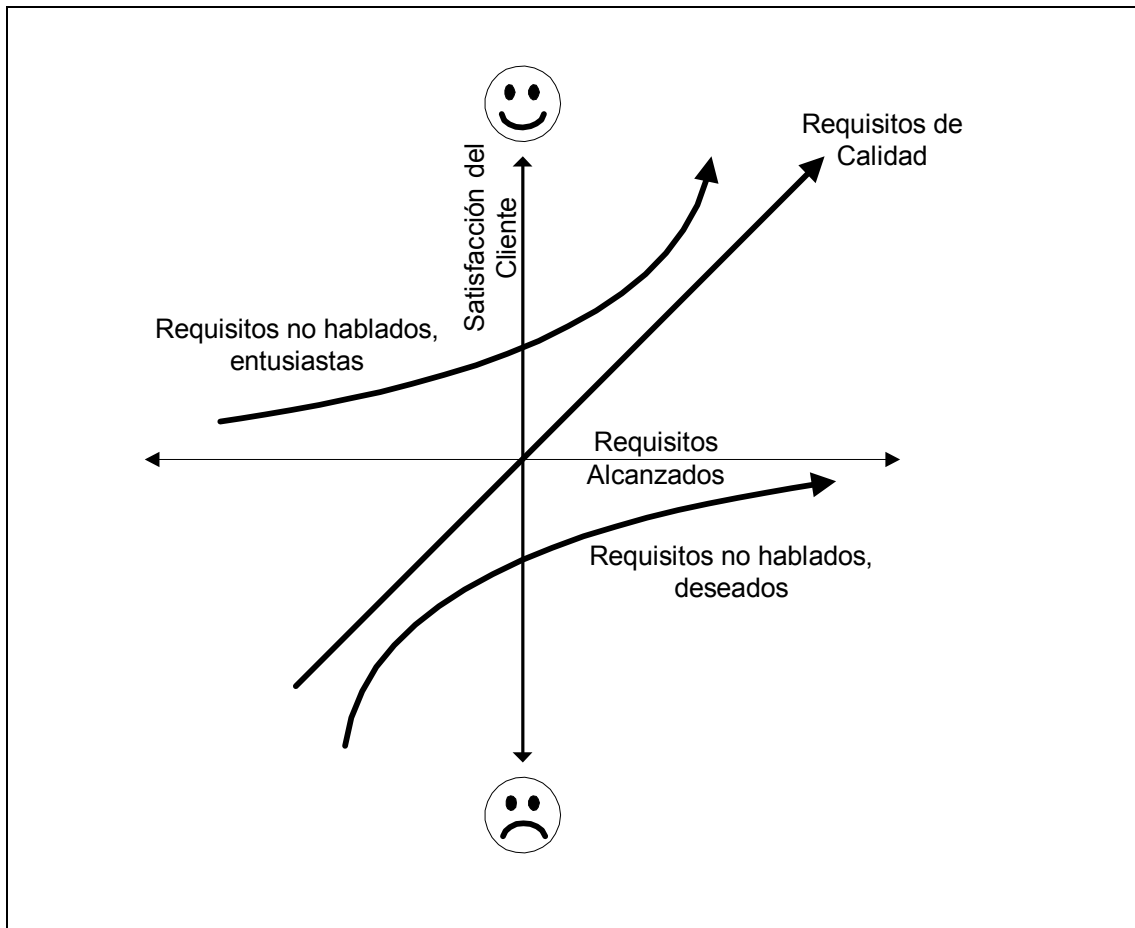


Figura 19. Modelo de Kano.

Después de ser implantados los requerimientos que no se dicen pero que generan gran satisfacción en el cliente (tipo 3), se convierten en requerimientos que el cliente no dice y no generan satisfacción pero si insatisfacción (tipo 2).

Diferentes formas se utilizan para identificar y dar prioridad a los requisitos, UML, QFD, entre otras. El auge que posee UML hoy en día es bastante grande ya que es una metodología sencilla y aplicable a la mayoría de los proyectos de desarrollo de software y se basa en la utilización de casos de uso (Ver anexo C). El desarrollo en función de la calidad (QFD)²⁴, ha sido

²⁴ Desarrollo en función de la calidad definición en Ingles de Quality Functin Deployment (QFD)

utilizada por más de 40 años a lo largo de las compañías para el análisis de datos sobre lo que los clientes realmente quieren, su principal propósito es mejorar la satisfacción del cliente mejorando la calidad de los productos o servicios, algo que los métodos tradicionales de captura de requisitos no realizan.

NADA INCORRECTO ≠ TODO CORRECTO

4.9 INICIO DE DETERMINACIÓN DE MEJORAS

Aunque la priorización de los requerimientos se puede realizar utilizando diferentes métodos, Seis Sigma utiliza la mayoría de las veces “Calidad en función del desarrollo (QFD)”.

Este paso es donde se inicia la matriz de QFD, aunque es solo el comienzo la calificación que se le proporcionará a los requerimientos del proyecto es un factor trascendental para hallar la solución verdadera al problema.

La escala puede variar dependiendo de los miembros del equipo, es recomendable utilizar la siguiente:

Tipo	Valor	Símbolo
Alto	5	o
Medio	3	o
Bajo	1	Δ

La calificación de los requerimientos debe estar directamente relacionado con los valores del árbol de soluciones que se determinó después de la voz del cliente, los requerimientos que afecten directamente a la CTQ con mayor porcentaje obtendrán la calificación alta y así descendientemente. Por

ejemplo. La CTQ 1 es la reducción de 60 segundos a 20 segundos en la búsqueda, y tiene una importancia del 70%, y la CTQ 2 es aumentar en un 30% la oferta del servicio en la biblioteca y tiene un 30% de importancia. Por lo tanto la clasificación es la siguiente:

Requerimiento	Importancia
Búsqueda ágil, eficaz, y efectiva.	5
Diseño gráfico amigable.	1
Disponibilidad de computadores para los usuarios.	3

4.10 PLAN DE COMUNICACIONES

El plan de comunicaciones es la estrategia usada para comunicar a todos los que participan en el proyecto, las actividades y el estado del proyecto, así como reportar el estado actual de sus tareas individuales y establecer los métodos para desarrollar registros históricos del proyecto. Un buen plan de comunicaciones ayuda a determinar y corregir a tiempo los errores que se pueden ir presentando en el transcurso del proyecto.

Es apropiado durante todo el proyecto mantener líneas de comunicación abiertas entre los miembros del equipo y los directores, pero también es importante mantener la comunicación con los usuarios finales del producto o servicio. El primer paso en el plan de comunicaciones es determinar los diferentes grupos que necesitarán información del proyecto. Los grupos que pueden existir son:

- Comité de Dirección.
- Miembros del equipo.
- Usuarios y gente que no esté directamente involucrada en las actividades del proyecto.

- Agentes externos.
- Patrocinadores del proyecto.
- Público.

Segundo, para cada uno de estos grupos, se debe determinar el tipo de información que necesitan, el nivel de detalle que requieren (Nivel muy específico, alto nivel y general) y qué tan seguido necesitan la información. Existen diferentes tipos de información que se necesita, dependiendo del grupo con el que se quiera comunicar, además existen muchas formas de comunicar la información. Las diferentes formas de comunicación que se pueden usar con cada persona que hace parte del equipo de trabajo son: reuniones (sea en grupo o individual), llamadas telefónicas, informes escritos, correo electrónico o la combinación de estos.

La siguiente es una guía para desarrollar un efectivo plan de comunicación:

- Se debe involucrar los miembros clave del equipo a la hora de desarrollar el plan de comunicaciones.
- Trabajar con cada uno de los miembros del equipo para definir cuando y como la comunicación tendrá lugar y como se trabajará en equipo para solucionar problemas que puedan surgir en el proyecto.
- Diseñar una estrategia en la que cada miembro del equipo ayude a asegurar que la información llegue a todos.
- Comenzar a desarrollar el plan de la comunicación tan pronto como se inicie un nuevo proyecto. Si el equipo de trabajo cambia con frecuencia, es necesario desarrollar nuevas estrategias de comunicación.

5. FASE ANALIZAR

Después de obtener la aprobación de la fase anterior, *Analizar* es el siguiente paso, para determinar el rendimiento de los procesos actuales, y comprender cuales son las causas de los defectos.

Al momento de iniciar la fase analizar, la ecuación $Y = F(x_1, x_2, \dots, x_n)$ establecida por los miembros del equipo ya debe tener claramente identificadas las X's que afectan negativamente el resultado. Se pasará entonces de las conclusiones estadísticas a determinar soluciones prácticas para crear acciones correctivas que generen una satisfacción en el cliente y una buena calidad en los productos o servicios.

Esta fase del proyecto tiene como principio no enfocarse en los síntomas del problema, sino en la *causa raíz* y en las *soluciones del problema*.

Para el desarrollo de esta fase se han identificado los siguientes pasos:

1. Determinar que causa la variación.
2. Lluvia de ideas para la mejora de procesos.
3. Determinar cuales mejoras tienen mayor impacto en los requerimientos del cliente.
4. Encontrar los riesgos asociados al proceso revisado (AMFE).
5. Desarrollar el plan de capacitación.
6. Identificar los sistemas afectados de otras dependencias.
7. Desarrollar el nuevo mapa de procesos.
8. Presentar al comité para aprobar.

rompería el objetivo de Seis Sigma²⁵. No se trata de ignorar la experiencia de algunos, ni la intuición de otros, pero no utilizar los datos obtenidos se convertiría en una pérdida de tiempo, y se corre el gran riesgo de tomar decisiones erróneamente, pasando por alto las verdades raíces del problema.

El ciclo del problema se enfoca en la elaboración y comprobación de las hipótesis planteadas acerca de la causa del problema como se muestra en la figura 21. Los pasos a seguir para hallar la raíz del problema son:

- Plantear las hipótesis cuidadosamente evitando escribirlas vagamente o haciéndolas demasiado breves, y nunca cambiar o modificar los datos para que las hipótesis sean ciertas.
- Analizar los datos que se han recopilado en la fase anterior o en esta fase, para discernir patrones, tendencias y otros factores que estén directamente relacionados con el problema, buscando la aceptación o el rechazo de las hipótesis.
- Analizar los procesos profundamente en su funcionamiento para identificar incoherencias, o áreas problemáticas que puedan contribuir al problema.

²⁵ Hacer suposiciones es un paso que se debe hacer más adelante, pero hacer realidad suposiciones sin fundamentos es un error. Algunas veces las suposiciones pueden resultar ciertas, pero no por esto se deben asumirlas ciertas desde el comienzo.

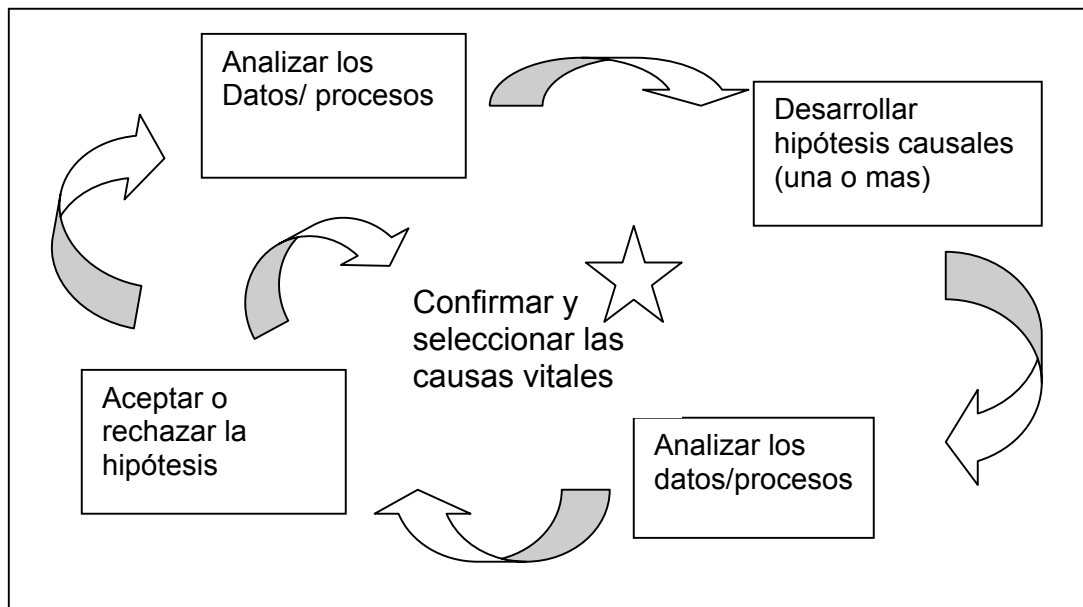


Figura 21. Ciclo de análisis del problema.

Los diagramas estadísticos ayudan a identificar las causas de la raíz de los problemas, el método complementario de este análisis es el planteamiento de los “5 porqués”?, que consiste en preguntar hasta cinco veces por qué, hasta llegar a una respuesta más concreta de cual es la raíz del problema. Por ejemplo:

- ¿Por qué falla el sistema? Por errores en el código de programación.
- ¿Por qué tiene errores el código de programación? Por que cada vez salen cosas nuevas.
- ¿Por qué cada vez salen cosas nuevas? Por que los clientes no explicaron claramente que se quería y los miembros del equipo no entendieron.
- ¿Por qué los clientes no explicaron claramente lo que se quería? Por que faltó planeación en el proyecto.
- ¿Por qué faltó planeación en el proyecto? Por que falló la metodología utilizada o fue obsoleta.

En este ejemplo se observa, que la causa de la raíz de los problemas en los sistemas no son las líneas de programación mal realizadas, sino el mal uso de las metodologías.

Para esta actividad los gráficos más utilizados son los diagramas de pareto, debido a que muestran claramente donde está la falla.

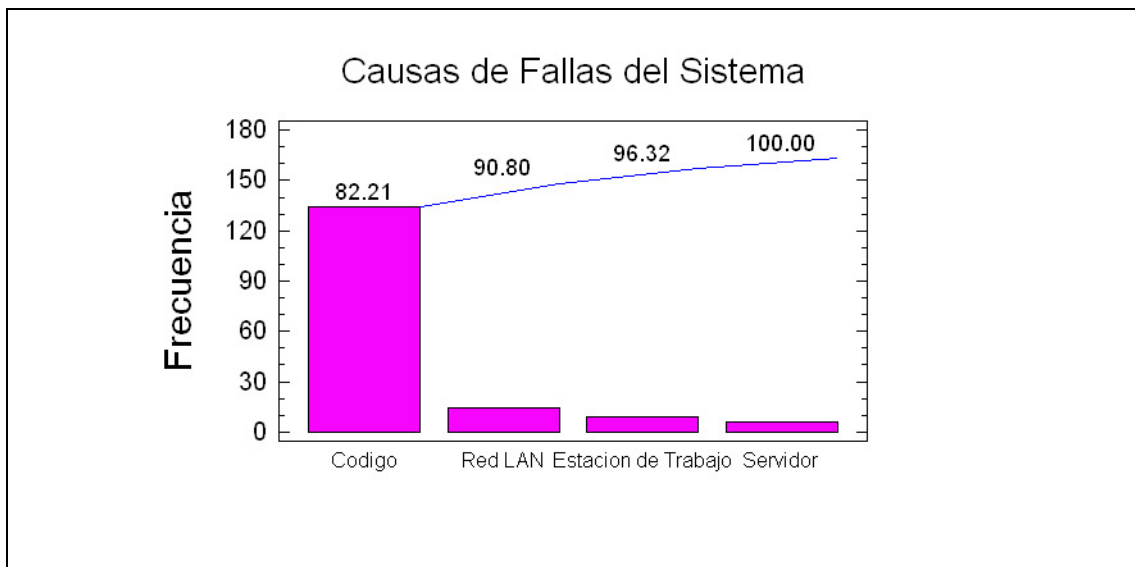


Figura 22. Pareto, causas de fallas del sistema.

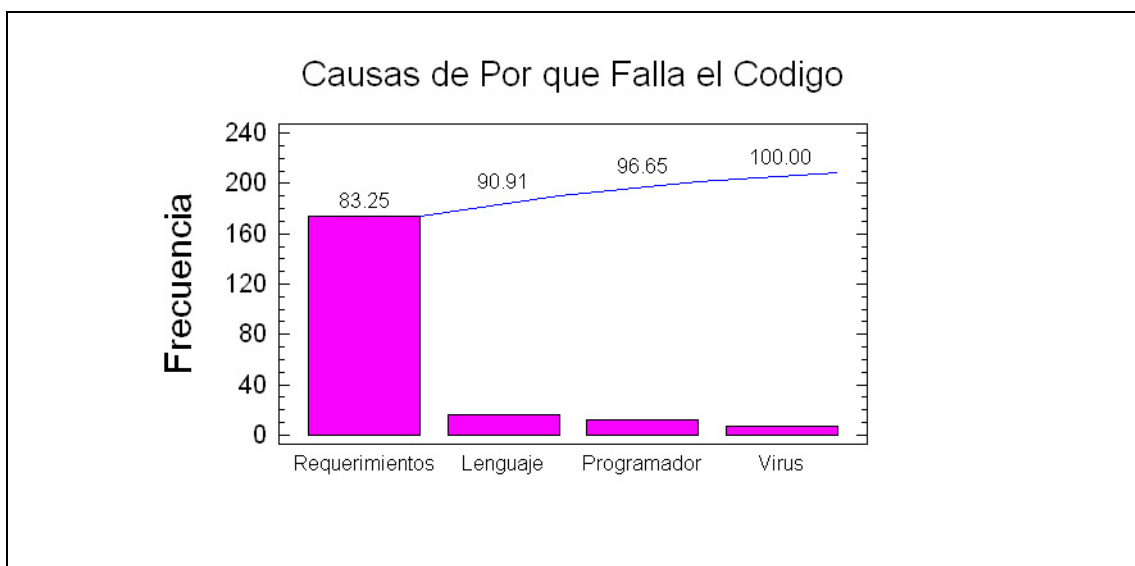


Figura 23. Pareto, Causas de por que falla el código.

La forma para elaborar los diagramas, es la metodología de arriba a abajo: desarrollar un nivel alto, y llegar hasta niveles bajos que son tan específicos como se necesite. Otros diagramas como el de cajas y el de corrido, son útiles para explicar más a fondo las causas de la raíz, pero su elaboración se realiza dependiendo si el proyecto lo requiere.

Otro diagrama que se utiliza frecuentemente para analizar la causa de la variación, es el diagrama causa-efecto (C-E) también conocido como el diagrama de esqueleto de pescado, el cual comienza con una lista estructurada de posibles causas, reuniendo un grupo de ideas y datos, mediante el método de lluvia de ideas²⁶. Una ventaja del diagrama C-E, es que al establecer categorías, el grupo piensa en diferentes posibilidades sin limitarse solo a unas pocas.

La explicación del diagrama C-E es la siguiente: las agujas o espinas arriba representan la variación de las entradas o procesos, denotándose con la X. Y las agujas abajo representan la variación de las salidas, denotándose por la Y.

Para construir el diagrama, se inicia indagándose el por qué de lo que se analizará, en el momento en que el problema establecido es aceptado por los miembros del equipo, la pregunta es ubicada en la cabecera del diagrama, seguidamente los miembros del equipo comienzan una lluvia de respuestas que se utilizará para encontrar la respuesta más acertada a la pregunta establecida. El diagrama se utiliza tanto para servicios como para producción, como se muestra en la tabla 11.

²⁶ Lluvia de ideas: Ver paso 2 de la fase analizar.

Tabla 11. Categorías del diagrama causa-efecto.

Servicios (Las 4 P's)	Producción (1 P y 5 M's)
<ul style="list-style-type: none"> • Políticas. • Procedimientos. • Personas. • Planta Tecnológica. 	<ul style="list-style-type: none"> • Maquinas. • Métodos. • Materiales. • Mediciones. • Madre naturaleza (ambiente). • Personas.

La figura 24 muestra el diagrama C-E, donde se ubican en las espinas principales del pescado o en las agujas las categorías, y las ramificaciones de cada categoría las causas.

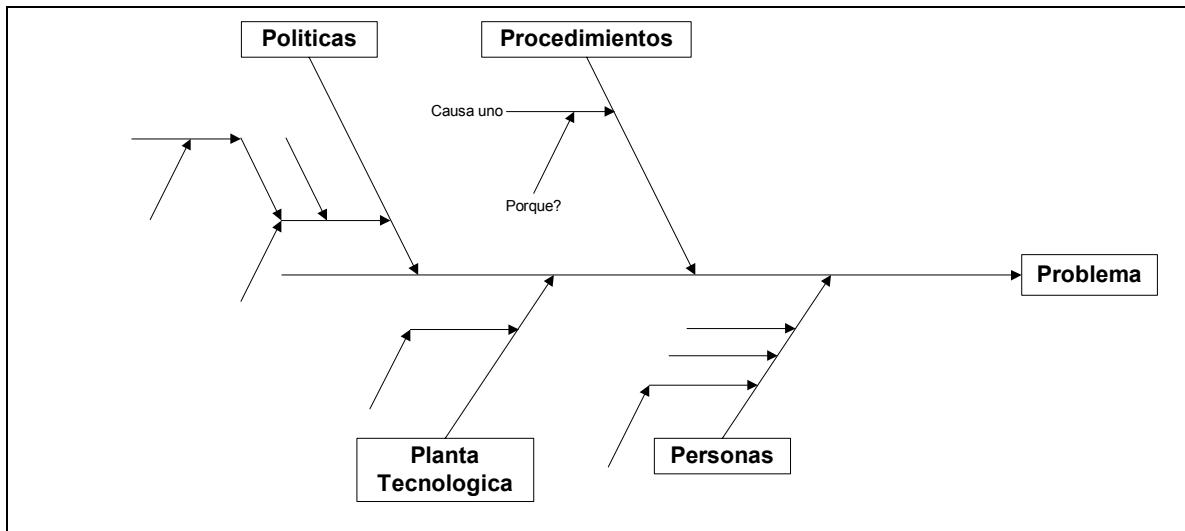


Figura 24. Diagrama causa – efecto.

Las categorías son una sugerencia, y pueden cambiar según el análisis que se deba llevar a cabo.

5.2 LLUVIA DE IDEAS PARA LA MEJORA DE PROCESOS

Una vez el equipo de trabajo tenga una lista de lo que ellos creen pueden ser las causas de la variación en los procesos, se comienza a considerar formas de eliminar esta variación. Por lo general se recurre al método de lluvia de ideas para introducir estas soluciones.

El método de lluvia de ideas, es una herramienta que se utiliza para generar la mayor cantidad de ideas o soluciones como sea posible a un problema específico. Antes de comenzar con una sesión efectiva de lluvia de ideas, es necesario establecer un conjunto de reglas que permitan establecer un código de conducta para poder interactuar entre los miembros del equipo.

La mejor forma de establecer estas reglas, es hacer que el mismo equipo de trabajo cree sus propias reglas. Una vez creadas, se debe asegurar que las reuniones se sigan según esas reglas.

Las siguientes son reglas que son comúnmente usadas en una sesión de lluvia de ideas:

- Mantener a todos los miembros del equipo involucrados. Que todos conozcan que es lo que se quiere lograr.
- No criticar las ideas de los otros. No es un debate, discusión o foro en el que se intenta demostrar la superioridad sobre otras personas.
- Construir sobre las ideas de otras personas. Generalmente una idea dada por una persona puede ser la base para otra gran idea.
- No pensar en calidad sino en cantidad.
- Eliminar las ideas repetidas o aquellas que no presentan relación con el fin de la reunión.

Las siguientes preguntas ayudan a preparar una sesión de lluvia de ideas.

- ¿Quién será el líder?
- ¿Quiénes participarán?
- ¿Dónde se realizará?
- ¿Qué deseo como resultado?
- ¿Cómo puedo escribir rápidamente para registrar las ideas sin retrasar al grupo?
- ¿Qué materiales necesito?

El resultado de la sesión será una lista de ideas, algunas pueden ser buenas otras no. Se debe evitar intentar analizar las ideas durante la sesión, después se podrán analizar y clasificar mediante los diagramas de C-E, o QFD. A menudo este es el siguiente paso después de la lluvia de ideas.

Para crear los diagramas se debe clasificar la lluvia de ideas, creando grupos con las ideas que se relacionen.

¿Cómo clasificar las ideas?:

- Agrupar las ideas que deben estar juntas. No es importante definir por qué deben ir juntas.
- Clarificar cualquier idea si no es lo suficientemente clara.
- Una idea puede ser ubicada en más de un grupo si se considera apropiado.
- Los grupos pequeños de ideas pueden pertenecer a otros más grandes.
- Los grupos grandes de ideas pueden hacerse más exactos o reducidos.
- Cuando la mayoría de las ideas se han agrupado, se puede comenzar a incorporar los nombres de los grupos que se crearon.

5.3 DETERMINAR CUALES MEJORAS TIENEN MAYOR IMPACTO EN LOS REQUERIMIENTOS DE LOS CLIENTES

Determinar con que se obtendrá el mayor impacto en los clientes, es una tarea de gran valor para el proyecto. En la mayoría de los casos, las opciones elegidas a resolver son aquellas que poseen un costo menor y un tiempo corto, sin establecer que tanto impacto generarán. Seis Sigma enfatiza que todas las decisiones deben ser basadas en hechos, y por lo tanto ésta es una herramienta que ayuda a la toma de decisiones.

Para la realización de este paso es recomendable realizar lo siguiente:

1. Listar los requerimientos de los clientes establecidos en la fase definir.
2. Cuadro de costo/tiempo/impacto.

Tabla 12. Costo/Tiempo/Impacto.

Situación	Costo	Tiempo	Impacto
Programar nuevamente.	Alto	Alto	Medio
Rediseñar.	Medio	Alto	Alto
Corregir errores.	Alto	Bajo	Bajo

El orden en que se deben atacar las situaciones es: primero trabajar sobre aquella que genere mayor impacto en el cliente, así su costo sea alto, se reflejará en la satisfacción del cliente después. Seguidamente atacar aquellas que tengan un costo bajo, y un impacto medio, y después atacar las restantes. Cuando el tiempo de proyecto es largo, se deben resolver todas las situaciones, si por el contrario el tiempo es relativamente corto, se debe atacar las de mayor impacto.

3. Crear el cuadro de ranking de las situaciones.

Tabla 13. Ranking de propuestas/requerimientos.

Ranking de Propuestas/Requerimientos						
		(3) Propuesta		(3) Propuesta		
Requerimiento (1)	importancia (2)	Efecto (4)	Impacto Cliente (5)	Efecto (4)	Impacto Cliente (5)	Impacto Total (6)
Grado de satisfacción por implementar la mejora, o el nuevo diseño			(7)		(7)	

El cálculo es el siguiente:

En la columna requerimiento(1) se ubican hacia abajo los requerimientos de los clientes.

- Calificar la importancia que posee cada requerimiento en la columna de Ranking (2), en una escala definida previamente. Este punto se realizó en el paso 9 de la fase medir.
- En la fila (3) Propuesta, se coloca una breve descripción de las propuestas planteadas para encontrar la satisfacción de los requerimientos de los clientes. Se ubican tantas como se hayan planteado.
- Calificar el efecto que tiene la propuesta (3) en satisfacer el requerimiento planteado, en la misma escala.
- Calcular el Impacto-Cliente(5), multiplicando el Efecto(4) con el Ranking de importancia (2)

$$\text{Impacto – Cliente (5) = Importancia (2) X Efecto (4)}$$

- El Impacto Total es calculado con la suma de todas las casillas de Impacto-Cliente (5) de cada propuesta.
- Los grados de satisfacción por la propuesta (7), son calculados con la suma de la columna (5) de cada Propuesta.

Tabla 14. Ejemplo QFD.

Ranking de Propuestas/Requerimientos						
		Propuestas				
		Utilizar otro Lenguaje de Programación		Adquirir Nuevos Equipos.		
Requerimientos	Importancia	Efecto	I.C.	Efecto	I.C.	Impacto Total
Búsqueda Ágil, Eficaz, y Eficiente.	10	10	100	1	10	110
Diseño Grafico Amigable.	4	4	16	1	4	20
Disponibilidad para la demanda de los usuarios.	7	1	7	10	70	77
Grado de satisfacción por implementar la mejora, o el nuevo diseño.			123		84	

Observando el ejemplo, se deduce que, utilizar otro lenguaje de programación tendría mayor impacto en todos los requerimientos, y principalmente se estaría haciendo énfasis en solucionar el requerimiento de los clientes que posee mayor puntuación (Búsqueda Ágil, Eficaz, y Eficiente).

5.4 ENCONTRAR LOS RIESGOS ASOCIADOS AL PROCESO REVISADO (AMFE)

Antes de presentar las propuestas para el cambio de los procesos al líder del proyecto, el equipo debe reconocer que esta propuesta no sería completa si no se determinan los riesgos asociados a la mejora o el nuevo de diseño de procesos.

Para realizar el análisis de los riesgos, existen diferentes métodos que se pueden utilizar, uno de ellos es una hoja de balance conocida como: Analisis de modos de fallo y efecto (AMFE)²⁷.

Los objetivos que plantea este método son:

- Identificar formas en las cuales un proceso no satisface los requerimientos del cliente.
- Determinar las fallas potenciales que tienen más efecto en el cliente.
- Evaluar los controles que se diseñaron para prevenir la falla.
- Documentar el plan de acción correctivo y sus resultados.

Este método provee un importante resultado para cada modo de fallo. Este resultado se denomina *Número de Prioridad del Riesgo (NPR)*²⁸. El mayor valor encontrado de NPR es el valor del impacto más serio en la falla. Los ítems con valores altos de NPR normalmente tienen planes de acción para disminuir los riesgos. Ver anexo D.

²⁷ Traducción de FMEA: Failure Modes and Effect Analysis

²⁸ RPN Risk Priority Number

5.5 DESARROLLAR EL PLAN DE CAPACITACIÓN

La capacitación se puede considerar como una poderosa herramienta con que cuenta una organización para alcanzar sus objetivos. Sin nuevos conocimientos, conceptos, herramientas y metodologías, todos los miembros (o el personal) que hacen parte de una compañía no podrán alcanzar un nivel de competencia que la organización exige para ganar participación de los clientes, rentabilidad y competitividad.

La capacitación implica un cambio de paradigma de las personas que operan los procesos anteriormente realizados. Se deben realizar prácticas adecuadas y contar con los instructores adecuados para enseñarle al personal a pensar y actuar según el cambio que se está implantando.

El objetivo principal de la capacitación, es lograr que los usuarios tengan el dominio necesario de las herramientas básicas acerca de los procesos que se están implantando.

El proceso de capacitación está formado por un ciclo constante de actividades como lo muestra el diagrama de la figura 25.

5.5.1 Evaluar las necesidades para la capacitación

La capacitación es necesaria sólo cuando un empleado carece del conocimiento que se requiere para que realice su trabajo.

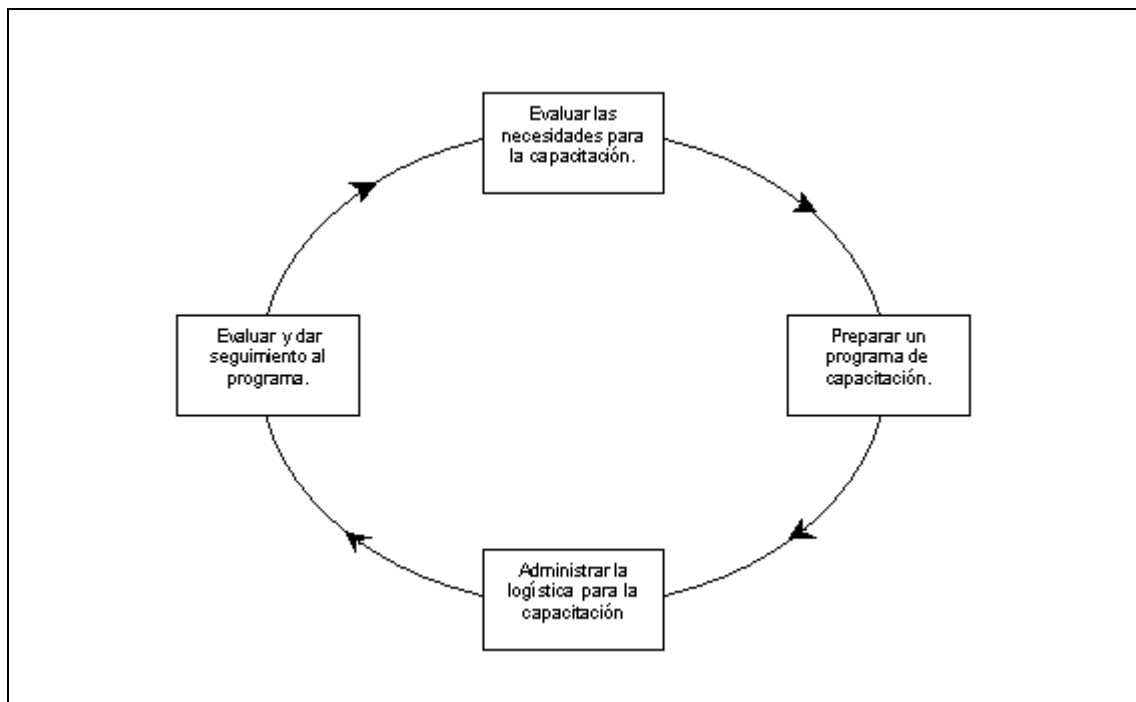


Figura 25. Ciclo de capacitación.

5.5.2 Preparar un programa de capacitación.

La planeación del programa de capacitación puede contener los siguientes ítem:

- Establecer objetivos generales del curso.
- Desarrollar un plan general de capacitación.
- Determinar la metodología, técnicas y enfoque de la capacitación.
- Desarrollar planes de sesiones de capacitación.
- Determinar los requerimientos de recursos.
- Desarrollar el presupuesto para actividades de capacitación.

5.5.3 Administrar la logística para la capacitación.

El éxito de un programa de capacitación depende, no sólo de la calidad de éste, sino también de la logística. Los participantes necesitan tener el ambiente adecuado para evitar tener problemas para concentrarse durante la capacitación.

Para asegurar que las cosas se efectúan correctamente, el responsable del programa de capacitación debe preparar un plan de trabajo que identifique:

- Todas las actividades que necesitan realizarse.
- Todos los materiales que se requieren para cada actividad.
- Los responsables de cada actividad.
- La fecha límite para terminar cada actividad.

Ejemplo de un plan de trabajo general:

Tabla 15. Ejemplo de un plan de trabajo general.

Tema				
Objetivo				
Metodología				
Dirigido a:				
Recursos	<input type="checkbox"/>		<input type="checkbox"/>	
	<input type="checkbox"/>		<input type="checkbox"/>	
	<input type="checkbox"/>		<input type="checkbox"/>	
Presupuesto			Responsables	
Ubicación			Fecha inicio	
			Fecha fin	

Cronograma				
Sesión	Tema	Fecha	Hora	Duración
Aprobado	<input type="checkbox"/>			
NOTAS				

Ejemplo de un plan de trabajo por sesión

Tabla 16. Ejemplo de un plan de sesión.

Tema			
Objetivos			
Metodología			
Lugar		Hora	
Fecha		Duración	
Cronograma			
Responsable			

Una vez que se tiene el plan de trabajo, se debe asegurar que cada quien tenga una copia. Debe utilizarse antes, durante y después del taller para comprobar que todo se realiza de acuerdo con el plan y el tiempo.

5.5.4 Evaluar y dar seguimiento al programa:

La evaluación es un proceso continuo que comienza con el desarrollo de los objetivos de capacitación. Algunas veces, es útil recolectar datos básicos de los participantes, tanto de su nivel de conocimientos y habilidades, como de sus expectativas respecto a la capacitación. Esto puede hacerse para determinar el nivel de habilidad de cada participante y para recibir información de lo que esperan aprender.

5.6 IDENTIFICAR LOS SISTEMAS AFECTADOS DE OTRAS DEPENDENCIAS

Tener conocimiento previo de los sistemas que afectarán directamente al nuevo sistema y entender su funcionamiento, ayudará a que el proyecto disminuya el riesgo de tener defectos durante su desarrollo o después de su implantación.

Los casos de uso analizados con anterioridad, son utilizados para saber que módulos específicos de que sistemas son a los que se les debe dar mayor prioridad.

Los aspectos más relevantes que se deben tener en cuenta en los sistemas son:

- Plataforma en que funcionan.
- Acceso a Datos (Si aplica).

- Motor(es) de base(s) de datos que utiliza(n).
- Modelo de datos, Esquema estrella o Entidad – Relación.
- Seguridad.
- Objetos de programación para la conexión (OLEDB, ODBC, JDBC, TCP/IP, EDI, etc.).
- Conexiones físicas.

La integración de los sistemas²⁹, es un paso posterior en la metodología y en el cual tener como mínimo el conocimiento de los anteriores aspectos optimizaría el trabajo.

5.7 DESARROLLAR EL NUEVO MAPA DE PROCESOS

Los mapas de proceso son unas de las herramientas esenciales de Seis Sigma. Lo básico de un mapa de proceso es simple: una serie de tareas y decisiones conectados mediante flechas para mostrar el flujo de trabajo.

El equipo de trabajo debe evaluar los mapas de procesos en los cuales se encuentra la causa raíz del problema. De esta forma se pueden analizar y observar algunas de las siguientes áreas problemáticas:

- *Desconexiones*: Puntos en los que el trabajo se pasa de un grupo a otro o donde se gestionan de forma incompleta en los que el proveedor y el cliente no se comunican con claridad los requisitos mutuos.
- *Cuellos de Botella*: Puntos del proceso en los que el volumen sobrepasa la capacidad, lo que hace que todo el flujo de trabajo sea cada vez más lento. Los cuellos de botella son puntos clave para entregar a tiempo los productos y servicios a los clientes y en la cantidad adecuada.

²⁹ Ver paso integración de sistemas, Fase “Diseñar”

- *Redundancias:* Actividades repetidas en dos puntos del proceso, también incluyen actividades paralelas que duplican los mismos resultados.
- *Bucles:* Lugares donde se retorna un alto volumen de trabajo al proceso para repararlo, corregirlo o solucionarlo.
- *Decisiones o inspecciones:* Puntos del proceso en que intervienen la elección de opciones, la evaluación o la verificación y crean retrasos potenciales. Estas actividades tienden a multiplicarse en la vida del proceso o de la empresa.

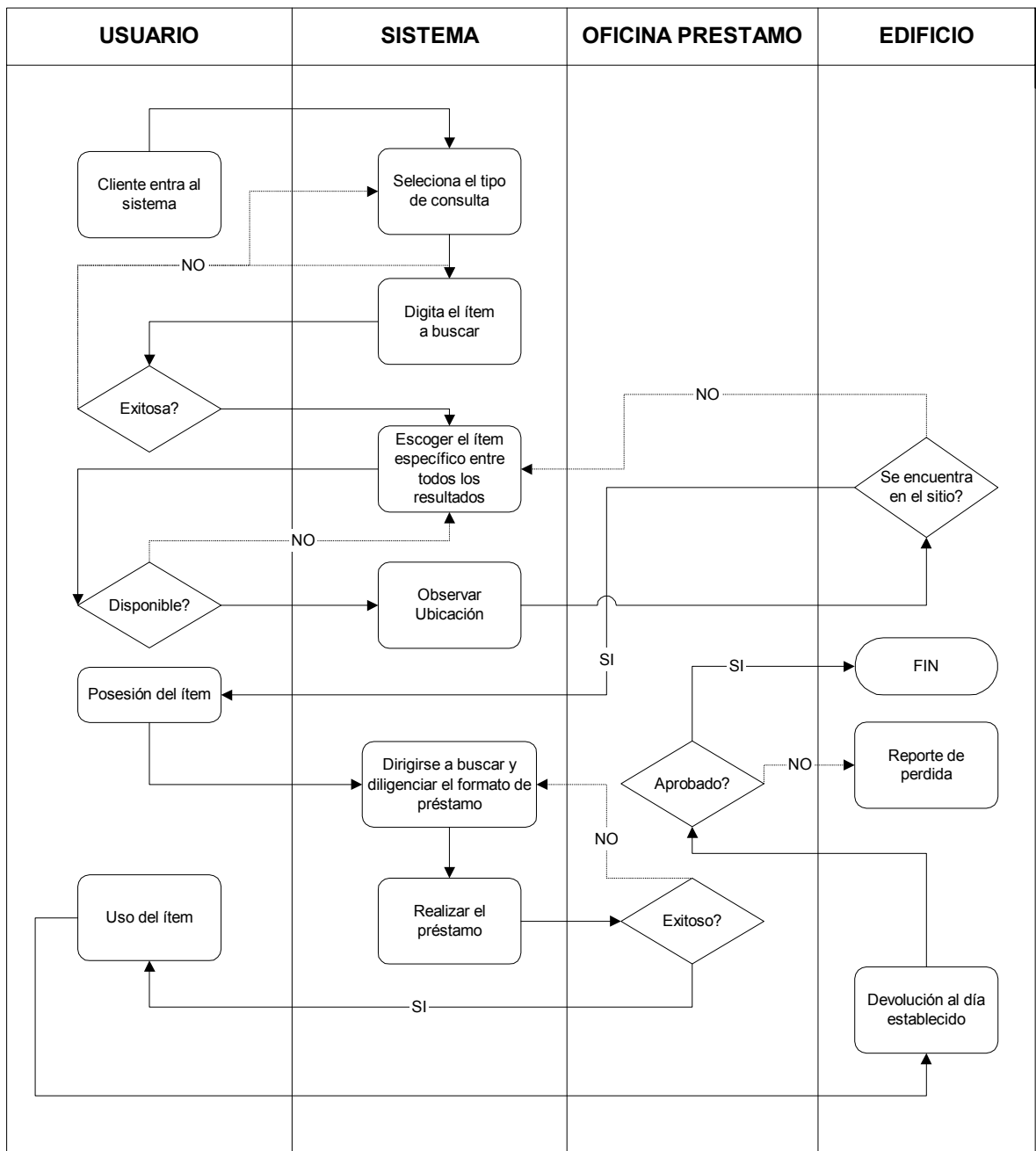


Figura 26. Ejemplo diagrama de flujo del nuevo proceso.

6. FASE DISEÑAR

En la fase de diseño se determina la arquitectura general del sistema y su comportamiento dinámico, adaptando la especificación realizada según los resultados de las soluciones obtenidas en la etapa anterior. En esta fase, se decide que tecnología se utilizará en el sistema aprovechando y adaptando sus ventajas. Los pasos que se identificaron para esta fase son los siguientes:

1. Seleccionar la Metodología de desarrollo de software.
2. Diseñar el formato de documentación de programas.
3. Definir la seguridad.
4. Especificación del diseño.
5. Integración del Sistema.
6. Revisión y perfeccionamiento del diseño.
7. Selección de Hardware y Software.
8. Construcción de prototipos.
9. Diseñar el plan de pruebas.
10. Revisar que los prototipos cumplan con los requerimientos.

6.1 SELECCIONAR METODOLOGÍA DE DESARROLLO DE SOFTWARE

6.1.1 Cascada.

En un modelo en cascada, un proyecto progresa a través de una secuencia ordenada de pasos partiendo del concepto inicial del software hasta la prueba del sistema. El proyecto realiza una revisión al final de cada etapa para determinar si está preparado para pasar a la siguiente etapa. Cuando la revisión determina que el proyecto no está listo para pasar a la siguiente etapa, permanece en la etapa actual hasta que esté preparado.

El modelo en cascada está dirigido por documentos; es decir, los productos principales del trabajo que se pasan de etapa en etapa son documentos.

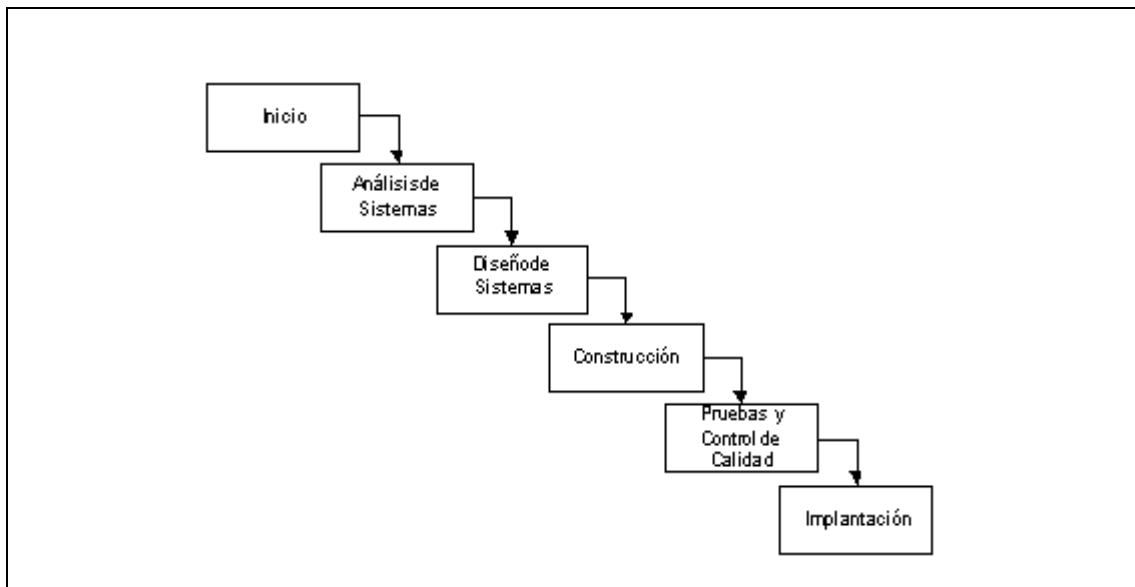


Figura 27. Modelo Cascada.

Ventajas

- Ayuda a minimizar los gastos de la planificación porque permite realizarla sin problemas.
- La secuencia de pasos y de las tareas que se realizan en cada paso se definen claramente.
- Se puede permitir que el personal se transfiera de un proyecto a otro.

Desventajas

- No permite flexibilidad en los cambios.
- Los requerimientos deben ser especificados completamente al principio del proyecto y no pueden sufrir cambios.
- El cliente es involucrado sólo periódicamente.
- Una etapa no comienza hasta no terminar por completo la anterior.
- No proporciona resultados tangibles en forma de software hasta el final del ciclo de vida.
- Requiere una cantidad excesiva de documentación.

6.1.2 Espiral.

El modelo de desarrollo en espiral, es un modelo manejador de riesgos durante el proyecto. Es utilizado para guiar a los miembros a diferentes niveles durante el desarrollo del sistema de información. Se diferencia de los otros modelos en dos características, la primera de ellas es su parte cíclica donde muestra durante el transcurso del proyecto un incremento en la definición e implementación del sistema y un decrecimiento en el riesgo. La otra diferencia es, que se enfoca en los puntos más importantes comprometiéndose a asegurar que las soluciones sean exitosas.

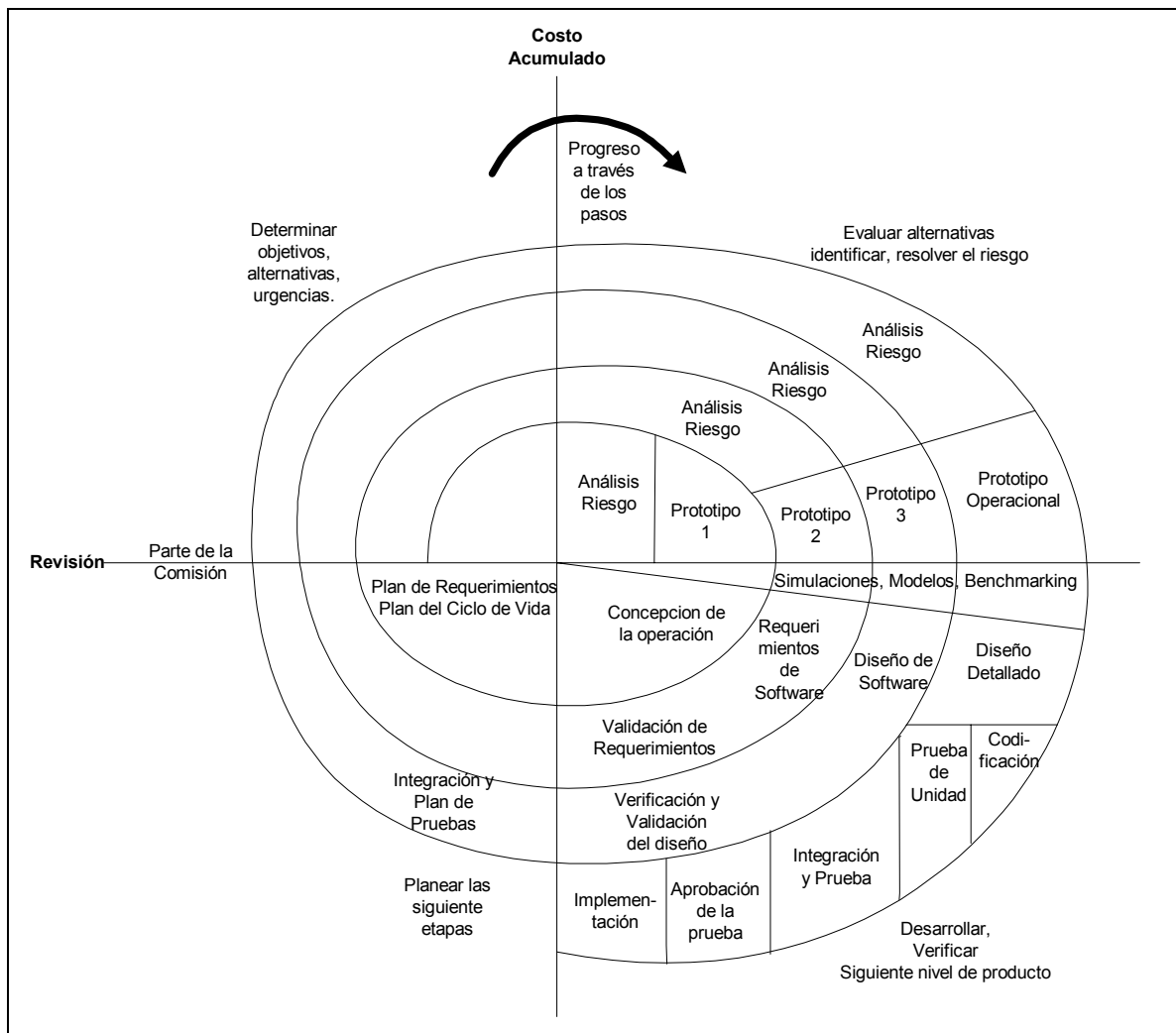


Figura 28. Modelo de Espiral.

Ventajas

- El riesgo es considerado desde muchos puntos de vista, lo que ayuda a garantizar no pasar por alto detalles que puedan hacer fracasar el proyecto.
- Se crean prototipos y se hacen simulaciones en cada fase, para minimizar el riesgo.
- Se observan diferentes alternativas de solución en cada fase.
- El diseño toma mayor importancia en las áreas donde existe mayor riesgo.

- Después de cada fase, se evalúa lo realizado y se planea la siguiente fase.
- Puede combinarse con otros modelos.
- El costo es inversamente proporcional al riesgo.

Desventajas

- Se puede llegar a perder la continuidad del proyecto cuando se presentan abandonos por parte de los miembros del equipo de trabajo.
- Se retrasa todo el proyecto cuando los resultados no son entregados según el cronograma.
- Si el riesgo no está bien identificado, el modelo falla.
- Requiere de más administración.

6.1.3 Proceso Unificado.

El proceso unificado, es un proceso de desarrollo de software basado en componentes, que adopta UML³⁰ como lenguaje de modelamiento para unificar todos los esquemas de un sistema software.

Más que un simple proceso de desarrollo, es un marco de trabajo que puede especializarse en gran variedad de sistemas software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto.

Los verdaderos aspectos definitorios del Proceso Unificado se resumen en las siguientes tres características, **dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental.**

³⁰ UML: Unified Modeling Language. Constituye una parte esencial del Proceso Unificado, pues sus desarrollos fueron paralelos.

Cada ciclo se desarrolla a lo largo de cuatro fases como se muestra en la figura 29. En la columna izquierda de la figura se muestran los flujos de trabajo fundamentales. Las curvas muestran hasta donde se llevan a cabo los flujos de trabajo en cada fase. Como cada fase se divide en iteraciones, éstas deben pasar por los flujos de trabajo, como se observa en la figura 29.

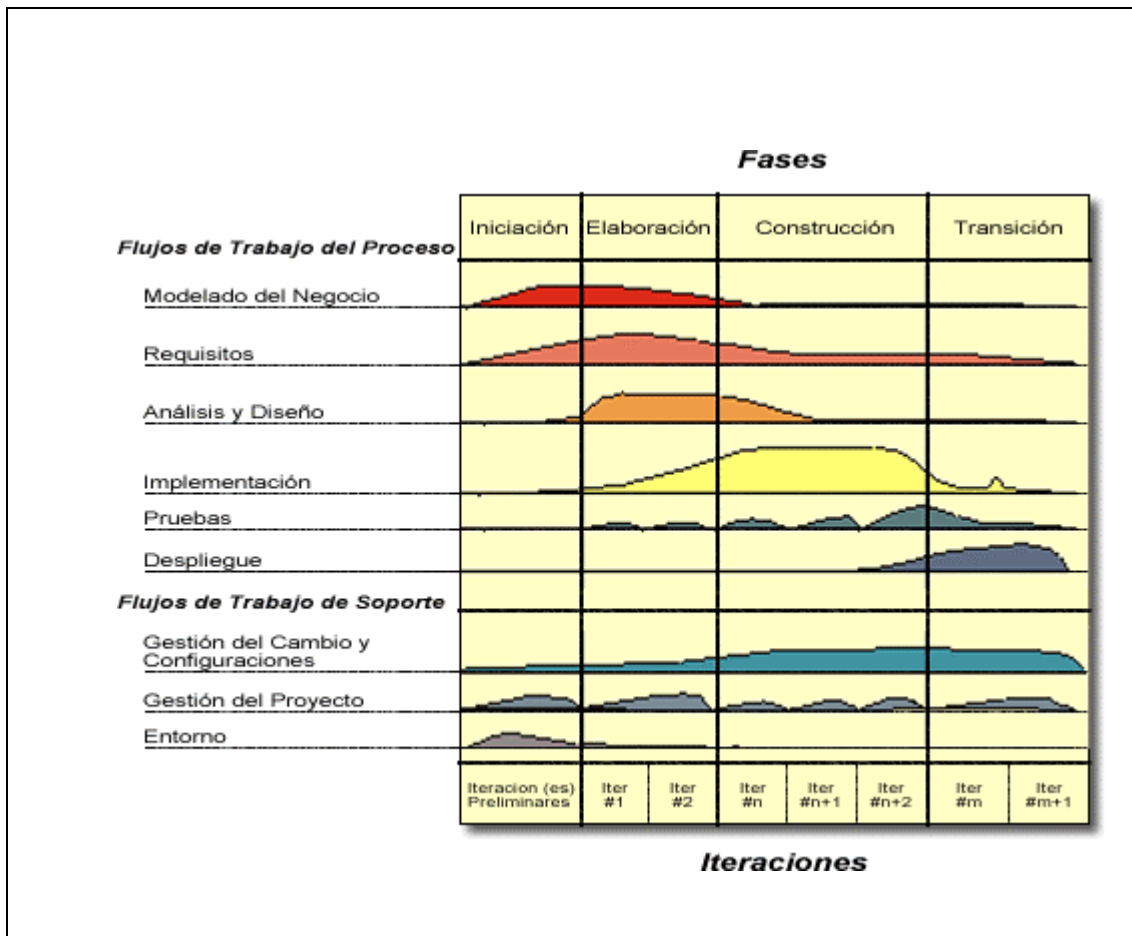


Figura 29. Modelo proceso unificado.

Ventajas

- Desarrollo basado en componentes.
- Utiliza UML como único lenguaje de modelamiento.
- Proceso Integrado.

- Permite ganar y retener control intelectual sobre el proyecto, para administrar su complejidad.
- Provee una base efectiva para el reutilización de software.
- Las iteraciones reducen el riesgo.

Desventajas

- Es un método enfocado en el proceso y no en el cliente.
- No se enfoca en el verdadero problema del cliente, si no en los requerimientos.

6.1.4 Modelo de Capacidad de Maduración (CMM³¹).

Es un programa especializado de calidad, enfocado al desarrollo de software. Fue creado por SEI³² como respuesta a los altos costos, cantidad de tiempo y la falta de programación que se presentaban en los proyectos de desarrollo de software.

CMM es una metodología para mejorar el desarrollo y las entregas de software haciendo los procesos predecibles.

³¹ CMM: Capability Maturity Model.

³² SEI: Software Engineering Institute

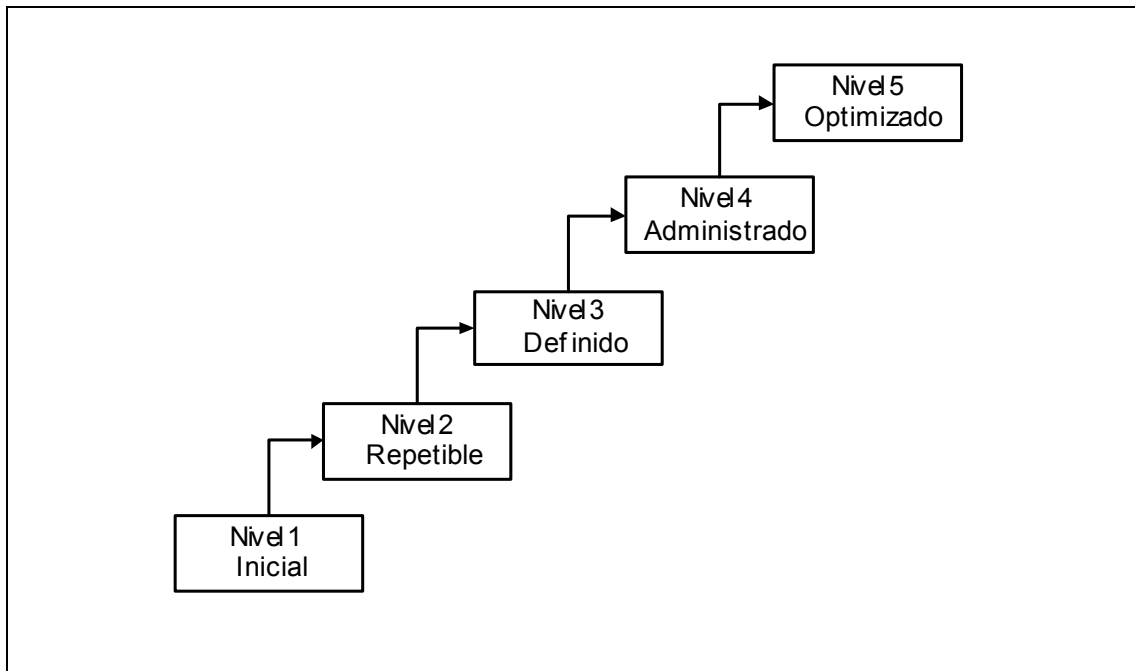


Figura 30. Modelo de CMM.

Ventajas

- Se centra en la reducción de la variación.
- Cuantifica el funcionamiento de mejora de procesos.

Desventajas

- No tiene una intensa preocupación por el cliente, ni hace énfasis en el trabajo en equipo ni en las decisiones basadas en hechos.
- Se aplica solamente a los procesos de software.
- CMM no asegura que el problema correcto esta siendo tratado.

6.1.5 Proceso Personal de Software (PSP³³).

El proceso personal de software, es considerado no solamente como un proceso de desarrollo de software, sino también como una metodología de mejoramiento de los procesos software, ayuda a ingenieros y desarrolladores individuales a mejorar su habilidad de escribir código, a planificar de forma más exacta los proyectos y a administrar la calidad de su producto.

PSP puede ser aplicado a muchas partes del proceso de desarrollo de software, desde el desarrollo de programas pequeños, la definición de requerimientos, escritura de documento, pruebas de sistema y mantenimiento.

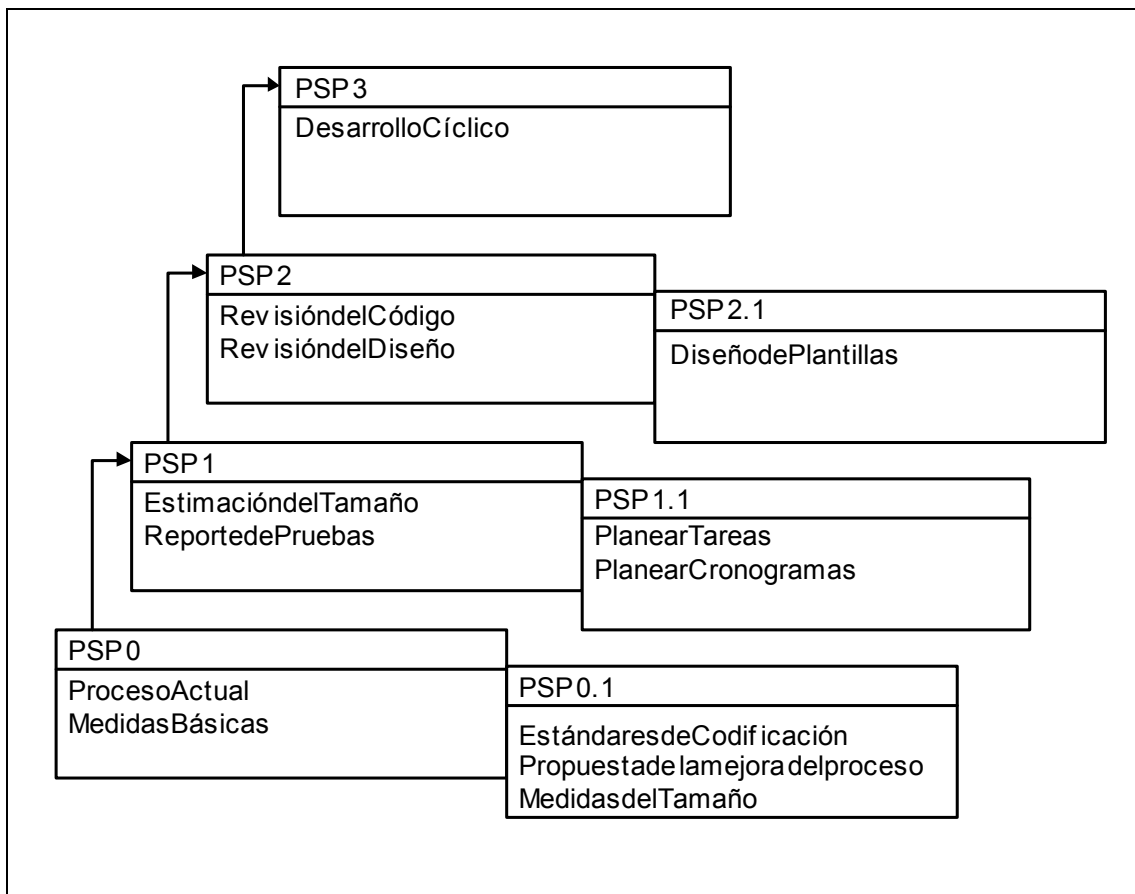


Figura 31. Modelo proceso personal de software.

³³ PSP: Personal Software Process

Ventajas

- Mejora la calidad de los productos y la eficiencia de los procesos.
- Las revisiones requeridas durante el proceso, aseguran que los problemas de diseño y codificación sean identificados.
- Produce cero defectos sobre el calendario y los costos deseados.

Desventajas

- La curva de aprendizaje es grande.
- No define como implementar un producto.

6.1.6 Proceso de Software en Equipo (TSP³⁴).

Es un modelo utilizado para guiar a los equipos de ingenieros a desarrollar intensamente software. Ayuda a los equipos a cumplir con los cronogramas planteados y los presupuestos acordados del proyecto. Maneja de 2 a 20 miembros por equipo y hasta 150 equipos.

³⁴ TSP: Team Software Process.

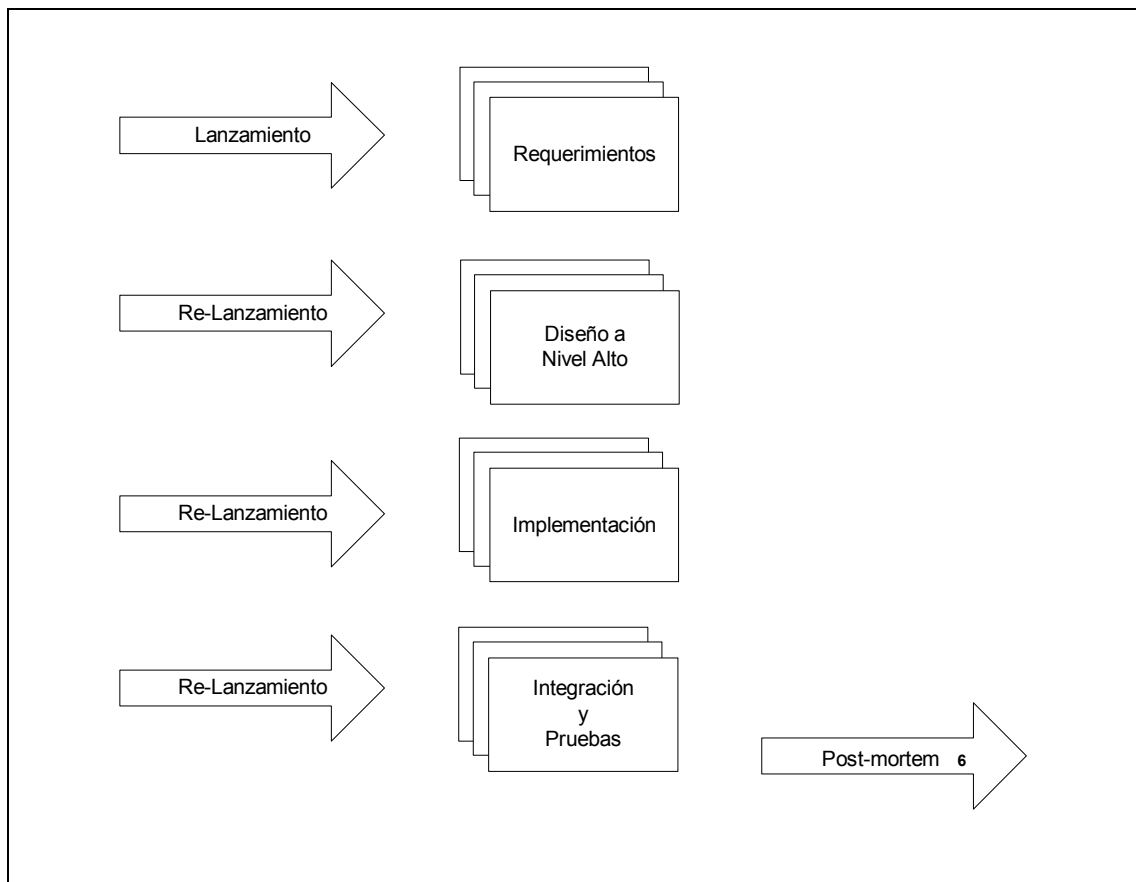


Figura 32. Modelo Proceso de software en equipo.

Ventajas

- Los proyectos TSP, pueden comenzar o finalizar en cualquier fase.
- Cada fase puede comenzar con un lanzamiento o relanzamiento.
- Ayuda a realizar productos de alta calidad, con costos y cronogramas agresivos.

³⁵ Post-mortem: se dice de una evaluación que se realiza a algún suceso o hecho que ha sucedido recientemente.

Desventajas

- Problemas del trabajo en equipo (liderazgo inefectivo, no compromiso, calidad pobre, diferencias en el nivel de conocimiento por los miembros).
- Diferentes objetivos establecidos, o no priorización.

6.1.7 Desarrollo Rápido de Aplicación (RAD³⁶).

Es un modelo de desarrollo de software lineal secuencial, enfocado en periodos cortos de tiempo. Se considera ágil y de desarrollo altamente veloz, basando su construcción en componentes.

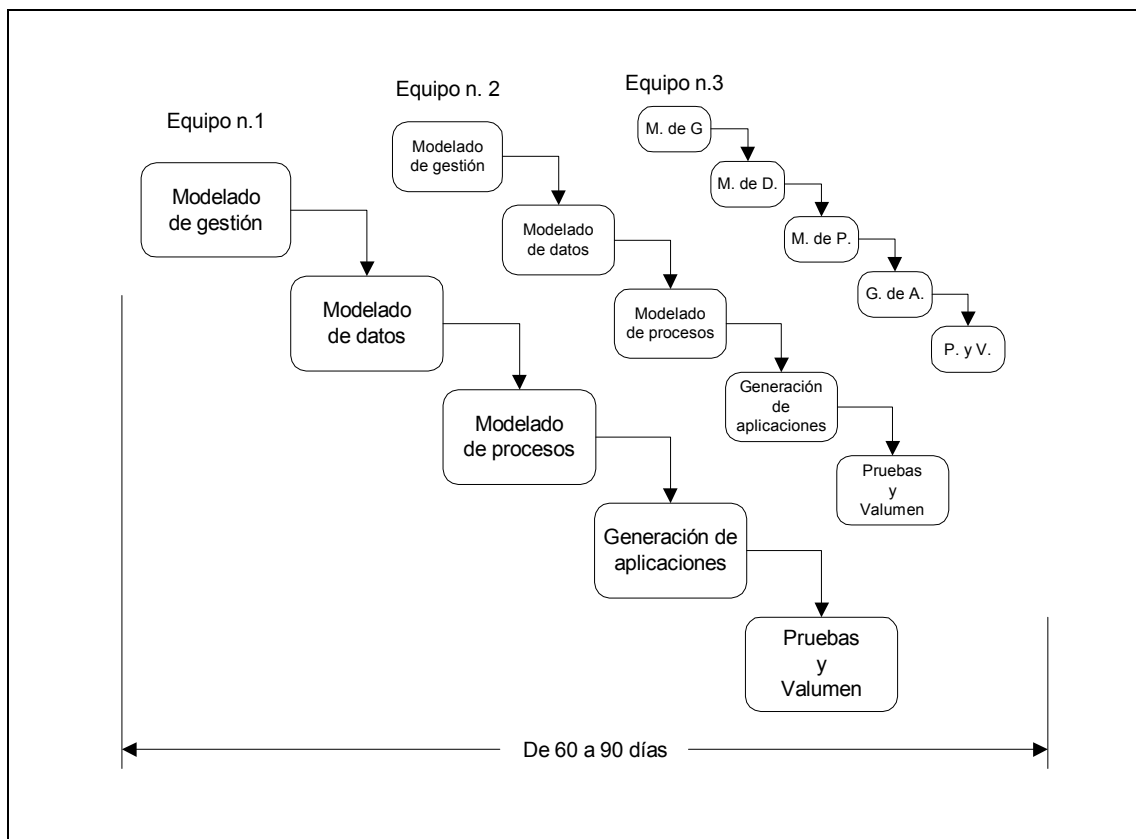


Figura 33. Modelo de desarrollo rápido de aplicación.

³⁶ RAD: Rapid Application Development.

Ventajas

- Es un proceso lineal, donde no se necesita tanta realimentación.
- Con el prototipado es fácil tomar riesgos tecnológicos.

Desventajas

- Las expectativas del cliente son altas con respecto al tiempo y la calidad del producto esperado.
- Tiene el riesgo de caer en “programar - corregir”.
- Requiere de recursos humanos suficientes en proyectos de gran escala.

6.1.8 Extreme Programing (XP).

Extreme Programming es una disciplina ágil de desarrollo de software, basada en la simplicidad, comunicación, y realimentación con el cliente. Trabaja enfocándose en partes simples, que individualmente no tienen sentido, pero integrándolas forman un producto software de alta calidad, que se entrega tan rápido como sea posible.

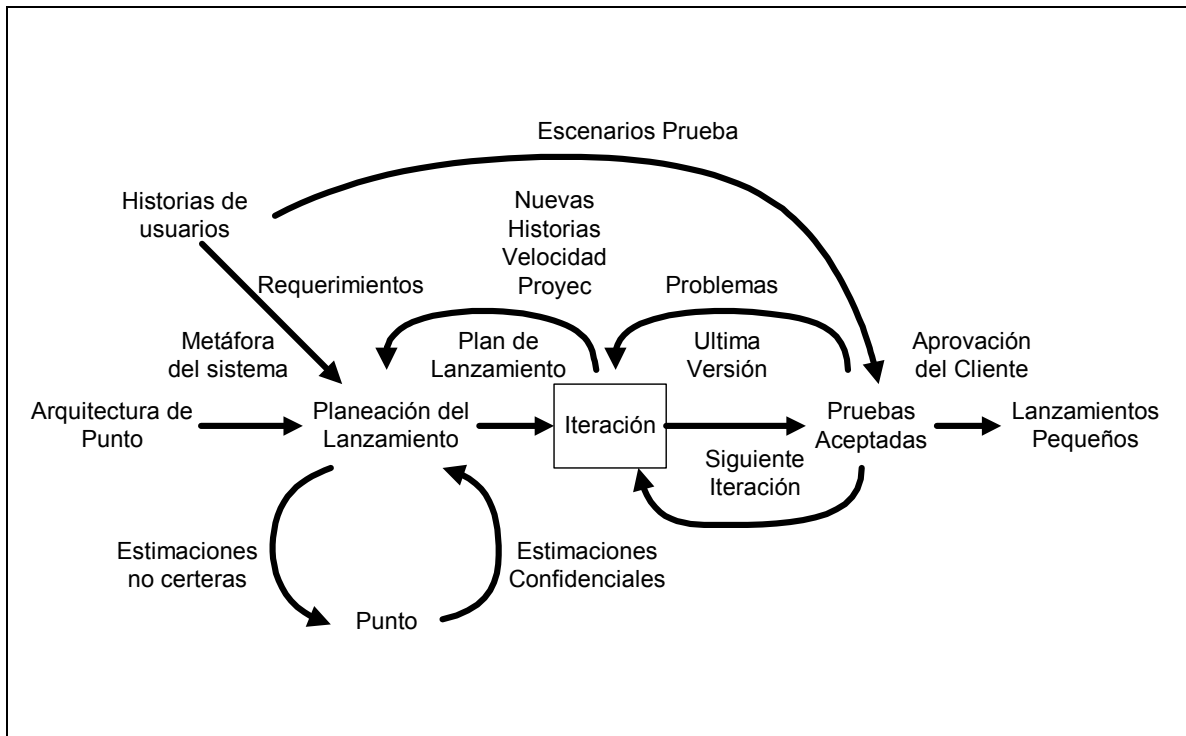


Figura 34. Modelo Extreme Programming.

Ventajas

- Altamente incremental e iterativo.
- Metodología de desarrollo ágil.
- Se desarrolla la versión más simple posible que resuelva el problema.
- Se ejecutan todas las pruebas todos los días.
- Se enfoca en la simplicidad.

Desventajas

- No hay requisitos explícitos si no que el cliente participa en el desarrollo.
- Se empieza por automatizar las pruebas.
- Se cambia el diseño aunque sea radicalmente cada vez que se necesite (después de la etapa del diseño).

Tabla 17. Comparación metodologías de desarrollo de software

METODOLOGÍA	ES RECOMENDABLE UTILIZAR	NO ES RECOMENDABLE UTILIZAR
CASCADA	<ul style="list-style-type: none"> • Cuando existe una definición estable del producto. • Proyectos complejos. • Proyectos largos con largos ciclos de desarrollo. 	<ul style="list-style-type: none"> • Cuando los requerimientos cambian constantemente. • Poco tiempo disponible.
ESPIRAL	<ul style="list-style-type: none"> • Cuando el cliente necesite observar los avances del desarrollo durante todo el proyecto, para hacer correcciones. 	<ul style="list-style-type: none"> • Proyectos pequeños. • Los sistemas de Información a desarrollar no son críticos.
RAD	<ul style="list-style-type: none"> • La rapidez del desarrollo es esencial. • Los sistemas existentes se han realizado con componentes. • Existe la motivación necesaria entre los miembros del equipo y los clientes. 	<ul style="list-style-type: none"> • No se pueden utilizar módulos. • Los riesgos tecnológicos son altos.
PROCESO UNIFICADO (UML)	<ul style="list-style-type: none"> • Cuando se desea modelar todo el proceso. 	<ul style="list-style-type: none"> • Cuando no es necesaria toda la documentación que utiliza UML ni se requiere ver el proceso modelado.
EXTREME PROGRAMMING (XP)	<ul style="list-style-type: none"> • Se buscan soluciones rápidas a problemas puntuales. • El tiempo es escaso y es factor fundamental. 	<ul style="list-style-type: none"> • Existen sistemas implantados y las mejoras rompen el diseño anterior. • Existe una cultura rígida entre los programadores. • Proyectos Grandes.
PERSONAL SOFTWARE PROCESS (PSP)	<ul style="list-style-type: none"> • En proyectos de desarrollo de software pequeños donde los desarrolladores trabajan individualmente. 	<ul style="list-style-type: none"> • Cuando el tamaño del proyecto es grande y el desarrollo implique la conformación de grandes equipos de trabajo.
TEAM SOFTWARE PROCESS (TSP)	<ul style="list-style-type: none"> • Existen grupos interdisciplinarios de trabajo. • Los miembros aceptan nuevas ideas. 	<ul style="list-style-type: none"> • Falta de liderazgo y compromiso en la organización. • El interés en usar diferentes metodologías es mínimo.
CAPABILITY MATURITY MODEL (CMM)	<ul style="list-style-type: none"> • Cuando se requiere exactitud en las entregas y los costos. • Cuando la calidad del producto es pieza fundamental en la organización. 	<ul style="list-style-type: none"> • Cuando la organización no esta preparada para combinar las herramientas de Información Tecnológica con administración de proyectos.

6.2 DISEÑAR EL FORMATO DE DOCUMENTACIÓN DE PROGRAMAS

Debido a la rotación de personal en las compañías, y teniendo en cuenta que los sistemas de información son una herramienta fundamental para realizar las tareas, se debe tener un formato estándar para documentar los programas que facilite la comprensión del código a todos los miembros del equipo debido a que la forma de programar es diferente en cada uno de los desarrolladores. La cantidad de tiempo que un integrante nuevo invierte en entender el código de los programas existentes es realmente alta y costosa. Por eso la documentación de programas toma más importancia en el desarrollo de software.

Los formatos pueden llegar a tener diferentes estilos, el líder del proyecto en conjunto con los miembros del equipo deben llegar a un consenso acerca del formato a utilizar.

Algunas características que debe tener el formato son:

- Nombre del proyecto.
- Nombre del programa.
- Breve descripción.
- Versión del software.(Opcional)
- Nombre del autor.
- Fecha de creación.
- Directorio donde se almacenará.
- Variables ambiguas.
- Directorio de archivos fuente.
- Bases de Datos a las que accesa, tablas de la Base de Datos que accesa, y Campos utilizados(Opcional).

6.3 DEFINIR LA SEGURIDAD

La seguridad en los sistemas de información juega un papel importante. En las fases anteriores no se mencionaba directamente el tema de seguridad, aunque al analizar los riesgos, la seguridad fue un punto a tener en cuenta. Es en este instante en el que el equipo de trabajo toma la decisión sobre que aspectos de seguridad deben ser tratados con gran importancia.

Debido a la variedad de los sistemas de información que posee una compañía, no todos los estándares de seguridad son aplicables para cualquier sistema, y aunque para muchos departamentos de Administración de Información la seguridad va a nivel de sistemas operativos únicamente, existen otros aspectos que van a nivel de diseño del sistema que pueden resultar menos costosos y más efectivos que otros.

Los roles y las responsabilidades de los miembros que forman parte de la seguridad se pueden describir como sigue:

- *Black Belt*: Líder del proyecto.
- *Jefe de seguridad de sistemas de información*: Es el encargado de desarrollar los estándares de seguridad que serán aplicados durante el ciclo de vida del software.
- *Asesor legal*: Es el responsable de asesorar a los miembros del equipo acerca de todos los asuntos legales en que pueda llegar a incurrir un sistema de información, como el contrato, licencias, incumplimientos, confidencialidad de información, derechos de autor, etc.

Las consideraciones que se deben tener en cuenta en aspectos de seguridad son las siguientes:

1. *Seguridad general de la información:* la responsabilidad de la seguridad del sistema debe ser asignada con nombres propios a los encargados, dependiendo del contrato que se haya realizado.
2. *Control de hardware y software:* se debe determinar quien puede instalar, agregar o cambiar software, y bajo que circunstancia es permitido. El control del software se hace con el fin de evitar virus, modificaciones al código, instalación de software no licenciado, o software que desproteja la entrada de intrusos.
3. *Control de la información y de los datos.* los miembros de equipo trabajarán con información y datos de la compañía que no tienen acceso libre, por lo tanto se deben establecer cláusulas para prevenir la divulgación de la información o datos durante el proyecto o después de su finalización.
4. *Seguridad en la documentación:* proporciona instrucciones a los usuarios acerca de las características de la seguridad en el sistema, además de convertirse en un soporte para documentar que los requerimientos se han cumplido.
5. *Asuntos legales:* se debe trabajar en conjunto con el departamento legal de la compañía, en los asuntos de tipo jurídico, para evitar violaciones de seguridad, riesgos y responsabilidades en el contrato.
6. *Personal:* se debe definir el personal que tendrá acceso físico y virtual a la información durante y después del proyecto.
7. *Seguridad física:* ubicación, medio ambiente, temperatura, riesgos de accidentes, etc.
8. *Características en los sistemas de información:* se refiere a las funciones específicas que se incorporan en los sistemas de información, dependiendo de variedad de factores como, sistema operativo, procesamiento de datos, transmisión de datos, riesgos, etc.
Por ejemplo:
 - a. Identificación y Autenticación.
 - b. Control de Acceso.

- c. Auditoria.
- d. Criptografía.
 - Autenticación de Datos.
 - Firma digital.
 - Llave de acceso.
 - Módulos de seguridad para la criptografía.
 - Validaciones.
- e. Integridad del Sistema.
- f. Arquitectura del Sistema

Estas consideraciones son importantes para cualquier sistema, aunque la realización de todas no es factor indispensable para que un sistema sea seguro totalmente. El tamaño y la importancia que el proyecto pueda representar a la compañía requerirá de estas o de otras consideraciones más.

6.4 ESPECIFICACIÓN DEL DISEÑO

El *diseño* es el paso más importante en el desarrollo de cualquier producto o servicio. Podría definirse como: "*el proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema con suficiente detalle como para permitir su realización física*"³⁷

El diseño del software se aplica independientemente de la metodología de desarrollo de software utilizado. El resultado de la creación de un modelo de diseño, comprende el diseño de las representaciones de datos, la arquitectura, interfaces y procedimientos.

³⁷ TAYLOR, E.S., An Interim Report on Engineering Design, Massachusetts Institute of Technology, Cambridge, MA, 1959.



Figura 35. Modelo del diseño.

En la figura 35, se representa el modelo del diseño en forma de una pirámide. Esto significa, que se busca crear un diseño de software que sea estable. Una base amplia en *el diseño de datos*, que transforma el modelo de la información en las estructuras de datos necesarias para crear software. Una región media estable, el *diseño arquitectónico*, que define la relación entre los principales elementos estructurales del programa. El *diseño de interfaz* que describe como se comunica el software consigo mismo, con los sistemas que operan con él, y con los usuarios finales. Y una parte superior, *diseño procedimental*, que transforma los elementos de la arquitectura en una descripción procedimental de los componentes del software.

Existe un conjunto de principios, sugeridos por Davis³⁸ para el diseño de software, algunos señalados en la siguiente lista:

- *Aumentar de tamaño los sistemas incrementalmente:* Una de las técnicas más efectivas para reducir los riesgos en la construcción de software, es hacerlos crecer incrementalmente. Comenzar con un sistema pequeño,

³⁸ Se comentan un pequeño conjunto de los principios de diseño de Davis. Para complementar esta información ver: Davis, A. 201 Principles of Software Development, McGraw Hill 1995.

que implemente solo unas pocas funciones, y luego incrementar su funcionalidad.

- *Hacerlo Simple*³⁹.: Una arquitectura o un algoritmo simple conduce a alcanzar mantenibilidad. Además, si se descompone el software en subcomponentes, se debe tener en cuenta que los humanos poseen dificultad para comprender más de siete cosas al tiempo.
- *Se debe poder seguir los pasos del diseño hasta el modelo de análisis*: Como un solo elemento del diseño se refiere a menudo a múltiples requisitos, es necesario tener los medios para hacer un seguimiento de cómo el modelo de diseño ha satisfecho los requisitos.
- *El diseño no debe inventar nada que ya esté inventado*: Los sistemas se construyen usando un conjunto de estructuras de diseño, muchas de las cuales ya se han utilizado anteriormente. Estas estructuras, a menudo denominadas componentes de diseño reutilizables, deben considerarse siempre antes que reinventar algo.
- *El diseño debería estructurarse para admitir cambios*: Durante el desarrollo de software, generalmente se encuentran errores, nuevos requerimientos, o el resultado de la falta de comunicación. Todo esto causa que el diseño cambie. Esto quiere decir que se deben seleccionar arquitecturas, componentes y especificaciones técnicas que permitan afrontar mejor el cambio. Los conceptos de Abstracción, Refinamiento, Modularidad, entre otros, permite conseguir este principio.
- *El diseño no es escribir código, y escribir código no es diseñar*: Incluso cuando se crean diseños procedimentales detallados para los componentes de un programa, el nivel de abstracción del modelo de diseño es mayor que el del código fuente. Las únicas decisiones hechas de diseño al nivel de código se refieren a los pequeños detalles de implementación que permiten codificar el diseño procedimental.

³⁹ Hacerlo Simple: Mantenerlo lo mas simple posible.

- *Analizar las causas de los errores:* Los errores son comunes en el desarrollo de software. Generalmente se invierten enormes cantidades de dinero y recursos detectando y eliminando. Como lo dice Seis Sigma, es más beneficioso prevenirlos antes que ocurran.
- *Se debe valorar la calidad del diseño mientras se crea, no después de terminarlo:* Existen variedad de conceptos de diseño y medidas del diseño disponibles para ayudar al diseñador en la valoración de la calidad.
- *El diseño debe presentar uniformidad:* Un diseño es uniforme si parece que sólo una persona desarrolló todo el conjunto. Se deberían definir normas de estilo y de formato para un equipo de diseño antes de comenzar el trabajo de diseño. Un diseño está integrado si se tiene cuidado en definir las interfaces entre los componentes del diseño.
- *Asignar a todos los productos un nombre y una versión:* Aunque existan muchas versiones intermedias de productos software, especificación de requerimientos, especificación de diseño, código, plan de pruebas, plan de administración, manuales de usuario, etc, cada producto se debe abastecer de un único nombre, número de versión y revisión y la fecha de creación.
- *Realizar el seguimiento de cada cambio:* Cada cambio puede generar problemas, tres de ellos son:
 1. El cambio realizado no solucionó el problema que se pretendía tratar.
 2. El cambio solucionó el problema, pero causó otros.
 3. En un futuro, cuando el cambio se percibe, nadie puede determinar por que fue hecho o por quien.

En cualquiera de los tres casos, es mejor hacer un seguimiento de cada cambio, llevando un registro de cada uno de los siguientes detalles:

- La petición del cambio.
- El proceso de la aprobación del cambio (Quién, cuándo, por qué).

- Los cambios a los productos intermedios.

6.5 INTEGRACIÓN DEL SISTEMA

La Integración de Sistemas consiste en integrar productos hardware y software estándar, así como productos desarrollados específicamente para construir un sistema que sea una solución completa que responda a las necesidades de los usuarios.

La integración de aplicaciones se debe hacer utilizando protocolos y tecnologías estándar disponibles. Existen diferentes métodos y caminos para integrar diferentes paquetes de aplicaciones y entornos de sistemas para crear una solución tecnológica interoperativa y homogénea. También es necesario tener en mente que la integración de soluciones debe ser lo bastante flexible como para ser fácilmente adaptable a los nuevos requerimientos de negocio.

6.5.1 Tipos de integración de sistemas

Actualmente existen 5 tipos de integración de sistemas. Aunque el objetivo de la metodología no es entrar en detalle sobre cada tipo, se describirá brevemente cada uno de ellos.

1. Integración orientada a datos.

Se basa en extraer datos de una base de datos para ubicarlos en otra, dentro de la misma organización o con otras organizaciones. Presenta las siguientes características:

- Agrega bases de datos completas, tablas o campos, si el sistema de información lo requiere.
- Se puede dar una transformación en los datos o en la lógica del negocio.
- Es la integración más económica.
- Se debe entender el metadata⁴⁰ entre las dos bases de datos.
- Se debe definir los eventos para que las señales sean capturadas cuando se necesita mover los datos.
- Los métodos para realizar este tipo de integración son:
 - Software para replicar la base de datos.
 - Message broker.
 - Construir una aplicación para sincronizarlas.

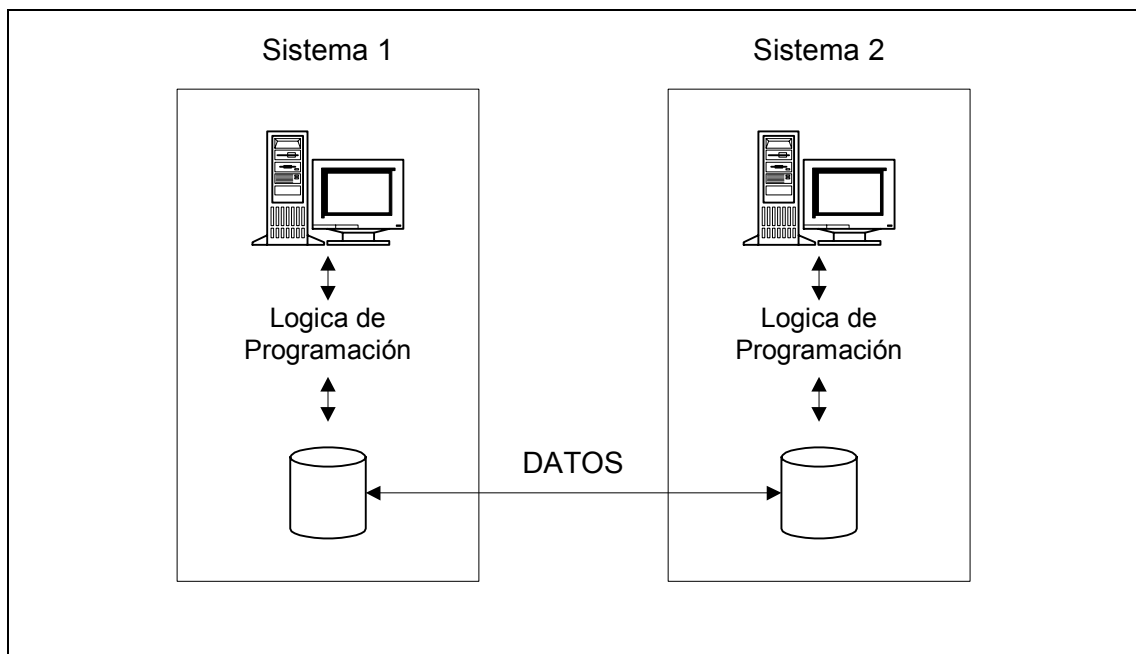


Figura 36. Integración orientada a datos.

⁴⁰ Metadata: Datos acerca de los datos.

2. Integración orientada a la interfaz de aplicación (API).

Es una interfaz que sirve de apalancamiento entre un paquete de software dentro de dos sistemas de información. Presenta las siguientes características:

- Utiliza los procesos del negocio y los elementos de datos más simples.
- Los messages broker son los más utilizados para realizar la API⁴¹.
- Su utilización puede generar varios niveles de acceso o de servicio sobre las otras aplicaciones.
- Se implementa escribiendo una función de llamado dentro de un programa el cual provee el enlace a una subrutina que necesita ser ejecutada.
- Algunos de los estándares para utilizar un API son:
 - Método de invocación remoto de Java (RMI).
 - CORBA.
 - Inter-ORB Protocol.
 - Microsoft Component Object Model COM+.

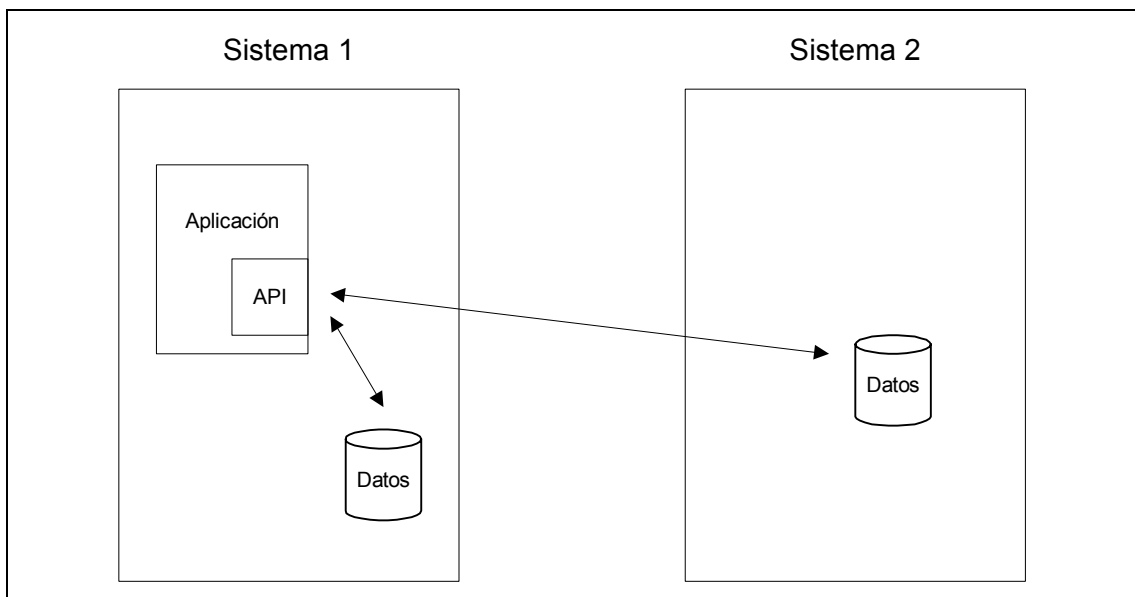


Figura 37. Integración orientada a interfaz de aplicación (API).

⁴¹ API: Application program interface.

3. Integración orientada a métodos.

Busca compartir la lógica del negocio mediante una aplicación compartida que entienda los métodos de los otros sistemas. Presenta las siguientes características:

- Se debe definir los métodos y estándares a compartirse a través de las aplicaciones.
- Existen las siguientes formas de realizarlo:
 - Utilizar un servidor Central.
 - Utilizar Objetos Distribuidos.
- Ofrece la reutilización de código.
- Presenta una desventaja cuando los sistemas tienen que sufrir cambios grandes para que se pueden entender con otros.

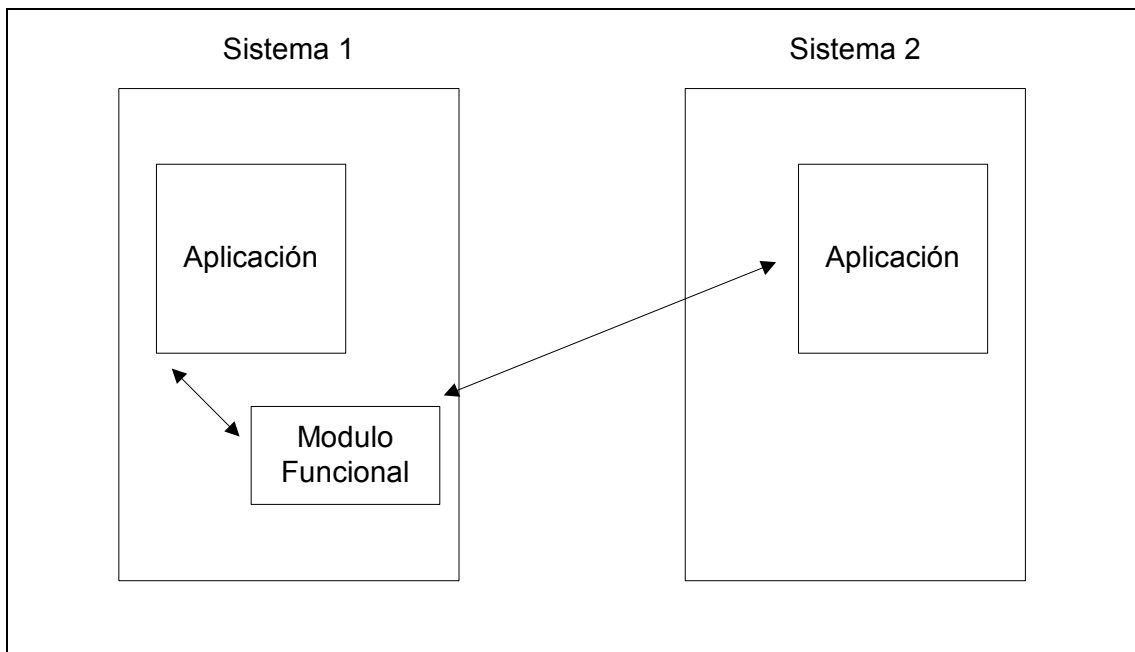


Figura 38. Integración Orientada a Métodos.

4. Integración orientado a portal.

Presenta la información de diferentes aplicaciones a través de una sola interfaz, la mayoría de las veces utilizando un navegador. Presenta las siguientes características:

- Implementación rápida.
- Aprovecha la maduración de tecnologías como: Web Services, Web Clients, Servidores de Bases de Datos, Servidores de Aplicación.
- Ofrece seguridad en la integridad de los datos de las aplicaciones.
- La información no fluye en tiempo real, necesita de la interacción humana para realizarse.
- Los sistemas no reaccionan automáticamente a los eventos del negocio.
- La información tiene que ser abstraída de una capa lógica de otra aplicación.

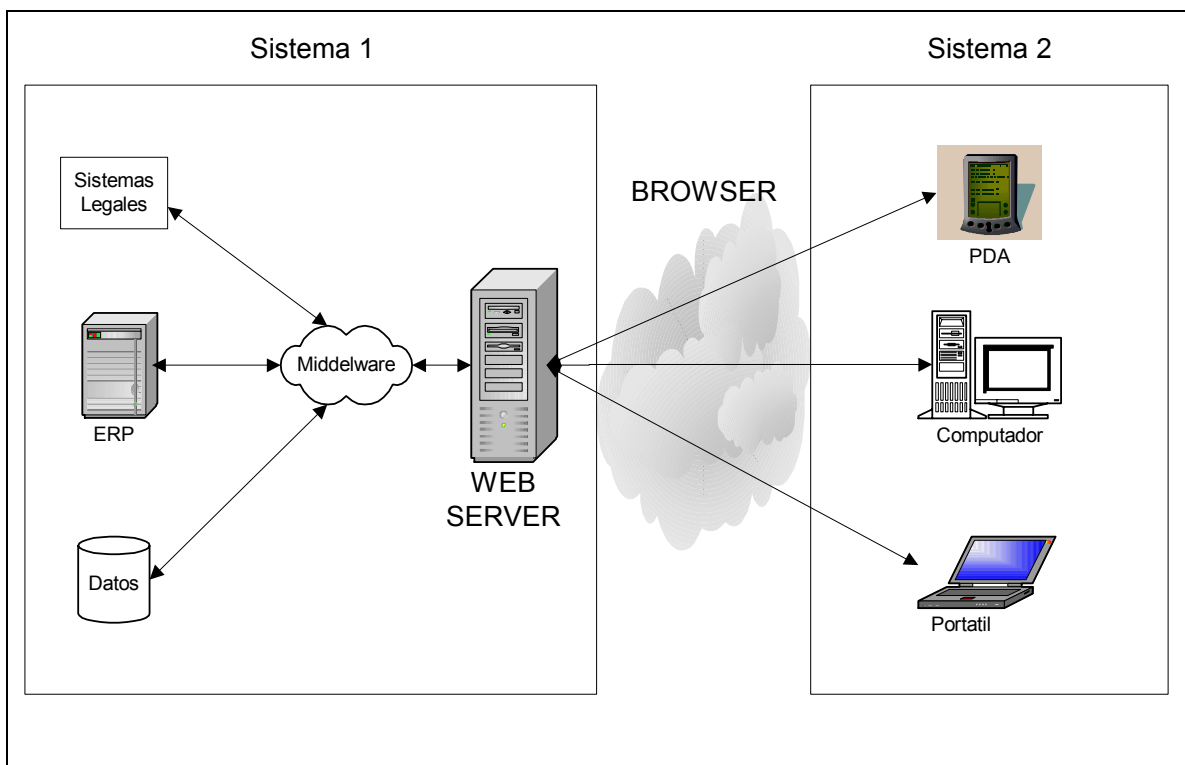


Figura 39. Integración Orientado a Portal.

5. Integración orientada a procesos.

Define modelos de procesos comunes para que se extiendan a través de los diferentes sistemas internos y externos. Presenta las siguientes características:

- Presenta una visibilidad confiable de los demás sistemas integrados.
- Los procesos son entendidos perfectamente por los otros sistemas.
- Usa estándares maduros, asemejando un sistema de workflow.
- Reaccionan en tiempo real a los eventos ocurridos en los otros sistemas
- Brinda la capacidad de redefinir procesos en cualquier momento.

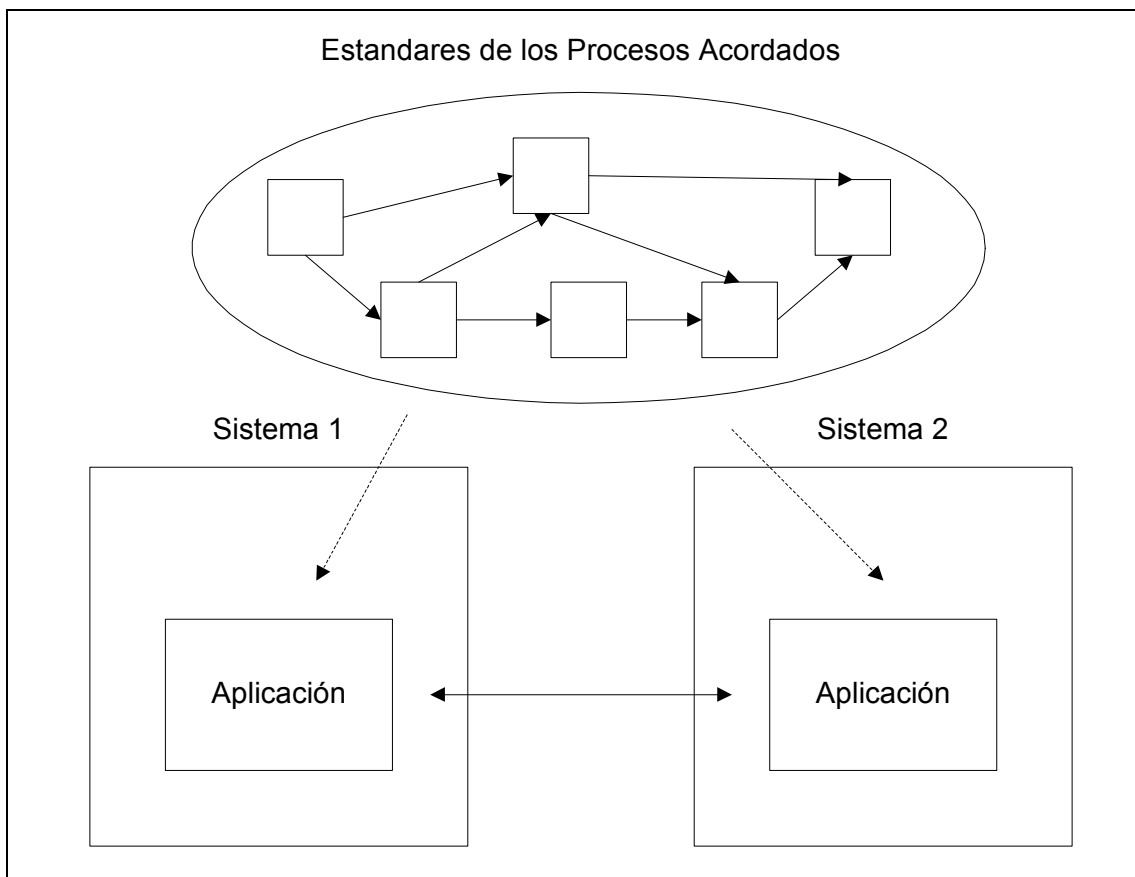


Figura 40. Integración Orientado a Procesos.

6.5.2 Estándares para conexión a la red

- **ASCII**⁴²: Representaciones numéricas de caracteres de texto.
- **TCP/IP**⁴³, **PPP**⁴⁴: Son los protocolos más conocidos de transmisión de datos, que indican cómo se efectúa la transferencia de información en la red. El protocolo TCP es el servicio de entrega segura y continua de información, en donde se especifica el formato de los datos y los reconocimientos que dos computadoras intercambian para lograr una transferencia segura y que asegura que los datos lleguen a su destino correctamente. El IP es el encargado de direccionar correctamente los paquetes de información que se enviaron a través de la capa TCP. PPP son versiones de TCP/IP diseñadas para establecer comunicación TCP/IP a través del puerto serial al que el módem está conectado cuando la conexión se hace vía telefónica.
- **FTP**⁴⁵: Estas siglas designan un método que permiten realizar el envío y recepción de archivos a través de Internet, de un computador a otro.
- **HTML**⁴⁶: Es un lenguaje que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces que conducen a otros documentos o fuentes de información y con inserciones multimedia.

⁴² ASCII: **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange.

⁴³ TCP/IP: **T**ransmission **C**ontrol **P**rotocol / **I**nternet **P**rotocol.

⁴⁴ PPP: **P**oint to **P**oint **P**rotocol.

⁴⁵ FTP: **F**ile **T**ransfer **P**rotocol.

⁴⁶ HTML: **H**yper**T**ext **M**arkup **L**anguage.

6.5.3 Interfaces de conexión a bases de datos.

- **JDBC**⁴⁷: El driver de conexión de java con las bases de Datos.
- **ODBC**⁴⁸: La conectividad abierta de las bases de datos, es un estándar o una interfaz de programación que se utiliza para acceder fuentes de datos (una base de datos que conecta con otros programas) que utilizan SQL como lenguaje de manipulación de datos, sin importar las interfaces propietarias de los sistemas manejadores de bases de datos.
- **OLEDB**: es una tecnología que tiene como objetivo resolver ciertas restricciones de ODBC. Autoriza el acceso a todo tipo de datos, permitiendo administrar el aspecto de tener distribuidas las fuentes de datos, y tiene en cuenta las restricciones de la Web. En un futuro esta tecnología reemplazará a la tecnología ODBC. ADO es el modelo de objetos que permite simplificar el acceso a esta tecnología.

6.6 REVISIÓN Y PERFECCIONAMIENTO DEL DISEÑO

Esta actividad, se hace como garantía de calidad del diseño del software. En ella intervienen analistas, líderes del proyecto y en algunos casos los clientes. Los principales objetivos son:

1. Descubrir errores en la lógica.
2. Verificar que cumpla con los requisitos.
3. Verificar la manejabilidad y claridad en el diseño.

⁴⁷ JDBC: **J**ava **D**atabase **C**onnectivity.

⁴⁸ ODBC, **O**pen **D**atabase **C**onnectivity.

Las siguientes características sirven de directrices para la evaluación de un buen diseño⁴⁹:

- El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acomodar todos los requisitos implícitos que desea el cliente.
- El diseño debe ser una guía que puedan leer y entender los que construyan el código, prueban y mantienen el software.
- El diseño debería proporcionar una completa idea de lo que es el software, enfocando los dominios de datos, funcional y de comportamiento desde la perspectiva de la implementación.

Existen muchas técnicas que ayudan a evaluar y perfeccionar el diseño inicial. A continuación se mencionan algunas:

- *Ensayos y simulaciones de un proceso*: Ensayar y discutir el funcionamiento del proceso, es una buena forma de validar el rendimiento de las cosas, conllevan a que los posibles problemas salten a la vista y determinar si es necesario mayor nivel de detalle.
- *Evaluación de momentos de la verdad*: Para esto es necesario identificar y evaluar las CTQ's del cliente en el proceso. Aunque se dispongan de los mejores métodos para proporcionar productos mejores y con mayor rapidez, siempre se debe tratar al cliente y contar con él, para que esto no repercuta en su insatisfacción.
- *Secciones de Información*: Disponer de mayor información del cliente o personas familiarizadas con los procesos pueden ayudar a tener una mejor comprensión de éstos. De esta forma se podrán prever cosas que

⁴⁹ McGlaughlin, R. Some Notes on Program Design, Software Engineering Notes, vol 16, n° 4, octubre 1991, págs 53 - 54.

nunca antes se han imaginado. La búsqueda de información en otras personas también proporcionará apoyo. Se debe tener cuidado de no limitarse a escuchar atentamente sugerencias, para luego no hacer nada con la información sugerida.

- *Análisis de Problemas Potenciales:* Todos los procesos tienen un gran número de problemas potenciales. El equipo de diseño de un proceso, puede tratar todos los problemas posibles, pero también puede tratar de identificar los más grandes y preparar acciones proactivas para eliminarlos o reducirlos. En el análisis de problemas potenciales, la estrategia fundamental es centrarse en las etapas críticas del proceso y preguntar: ¿Qué podría fallar?. Entonces hay que concentrarse en los problemas que tengan mayor probabilidad o mayor impacto y desarrollar acciones *preventivas*⁵⁰ o *reparadoras*⁵¹.
- *Análisis de consecuencias no deseadas:* Se debe analizar el impacto que conlleva el nuevo proceso y los diversos procedimientos, formularios, sistemas, etc. Poner a funcionar un nuevo proceso, trae un efecto en la gente y los demás procesos. Este cambio, puede crear problemas que nunca se habían pensado y que son potencialmente importantes. Comprender la interconectividad de los procesos es clave para realizar un buen análisis de las consecuencias. Puede hacer un seguimiento de los efectos de los nuevos procesos para ver quién se verá afectado y como.

Lo más importante en la revisión y el perfeccionamiento del diseño de los procesos, es obtener información de estos y adaptar o mejorar el proceso de manera que incorpore ese aprendizaje.

⁵⁰ *Acciones Preventivas:* Son las acciones encaminadas a reducir o bloquear el efecto del problema.

⁵¹ *Acciones Reparadoras:* Son las medidas diseñadas para contener o vencer las consecuencias del problema.

6.7 SELECCIÓN DE HARDWARE Y SOFTWARE

En la etapa de diseño se debe tener especial cuidado con la selección de las herramientas hardware y software que se van a utilizar para el desarrollo del proyecto.

En el proceso de selección se pueden mencionar dos pasos:

1. Identificar el “software” que mejor se ajuste a los requerimientos.
2. Identificar el “hardware” que mejor funcione con ese “software”.

El analista de sistemas, deberá evaluar el software junto con el líder del proyecto. La decisión más importante que se debe tomar es el tipo de software que se usará, existen diferentes propuestas de software libre, o con licencia que se pueden considerar. Para esta decisión se debe haber realizado un previo análisis de costo/beneficio. Se pueden evaluar desde las siguientes perspectivas:

- Modo de operación.
- Transición.
- Actualizaciones.
- Fabricación
- Valor (costo).

Elección de un lenguaje

La decisión más importante que se debe tomar al diseñar un sistema grande de software, es la de tomar en consideración el lenguaje de software que se va a utilizar en la aplicación del sistema. Un análisis inapropiado, puede ocasionar dificultades posteriores y aumentar el costo previamente planeado.

Este problema puede desaparecer si se selecciona el lenguaje de programación más apropiado para la solución y la infraestructura física existente.

Una buena elección de un lenguaje es importante porque reduce al mínimo las dificultades de codificar el diseño, reduce la cantidad de pruebas necesarias haciendo el programa más legible y más fácil de mantener.

La aplicación del sistema debe ser fácil de mantener, por lo tanto esto implica que debe codificarse en un lenguaje de alto nivel que proporcione la posibilidad de construir un sistema como varios módulos autónomos cooperativos.

Entre los criterios que se aplican para la evaluación de lenguajes disponibles están:

1. **Los requisitos del cliente del sistema.** La persona que contrata el sistema puede especificar que se utilice determinado lenguaje y se debe respetar, el líder del proyecto es el encargado de decidir cual es el lenguaje más apropiado para realizar el sistema.
2. **Disponibilidades de compiladores del lenguaje.** Si se realiza una aplicación por medio de la configuración de un sistema operativo o un hardware en particular, se debe disponer de un traductor del lenguaje eficiente para aplicarlo.
3. **Disponibilidad de instrumentos de software para apoyar el desarrollo de los programas.** Instrumentos de software, construcciones de referencia cruzada, sistemas para control de código, y analizadores de flujo de ejecución, son importantes en el

apoyo del proceso de programación, ya que facilitan la aplicación y confirmación del sistema.

4. **Conocimiento del personal de programación existente.** Aunque no es una dificultad para un desarrollador aprender un nuevo lenguaje, se necesita adquirir práctica en algún lenguaje antes de adquirir una verdadera competencia.
5. **Lenguajes de programación utilizados en proyectos previos.** Esto se utiliza cuando los desarrolladores han trabajado con un lenguaje en proyectos anteriores y están familiarizados con él.
6. **Necesidad de transportar el software.** Si esta orientado el software a una sola configuración de hardware y tiene un tiempo de vida limitado, los aspectos de su transporte no son limitados. Si el sistema esta destinado a operar en maquinas distintas es necesario un lenguaje de programación capaz de crear programas portátiles.
7. **La aplicación que se está programando.** Influye en gran medida respecto al lenguaje que se utilizará.

Lenguaje de programación e ingeniería de software

Independientemente de la metodología de desarrollo de software, el lenguaje de programación tendrá impacto en la planificación, el análisis, el diseño, la codificación, la prueba y el mantenimiento.

La calidad del resultado final se encuentra más fuertemente unida a las actividades de ingeniería de software que preceden y sigue a la codificación.

Para la selección del lenguaje se debe tener en cuenta cuales son las causas

principales que generan el problema, por ejemplo si la causa más grande es la velocidad con que se muestran los datos pero el cliente quiere que el sistema utilice demasiados gráficos, se debe seleccionar el lenguaje que garantice la velocidad antes que los gráficos, un segundo proyecto es como hacer que los gráficos alcancen los niveles de velocidad exigido, tarea realizada con la planeación multigeneracional de proyectos⁵².

Luego de la selección del software, se debe realizar la evaluación del hardware. El analista de sistemas necesita trabajar junto con el líder del proyecto para determinar que hardware será necesario. En el momento de adquirir el nuevo hardware, es importante tener conocimiento del inventario con el que se cuenta actualmente, y si no se dispone de uno, el analista necesita realizar uno rápidamente y apoyarse en él.

6.8 CONSTRUCCIÓN DE PROTOTIPOS

Después de aplicar los principios del análisis, e independientemente de la metodología de desarrollo de software que se aplique, se construye un modelo de software llamado *prototipo* que será valorado por el cliente y el desarrollador. La principal aplicación de los prototipos de un sistema es ayudar tanto a los clientes como a los desarrolladores a entender los requisitos del sistema y de esta forma verificar que se esta haciendo lo que el cliente quiere. También puede ser usado para entrenar al cliente antes que el sistema final sea entregado.

Los beneficios de desarrollar un prototipo son:

⁵² Planeación multigeneracional de proyectos, en inglés, multigeneration project management plan.

1. Identificar los malentendidos entre los desarrolladores y el cliente, mientras se demuestran las funciones del sistema.
2. Detectar a tiempo los servicios del usuario que hagan falta.
3. Identificar y mejorar los servicios que sean difíciles de usar o confusos para el cliente.
4. Identificar y completar los requisitos que estén incompletos o inconsistentes.
5. Servir de base para escribir una especificación de sistema de calidad.

Pueden darse dos enfoques al momento de crear un prototipo, cerrado o abierto. El enfoque de *prototipo cerrado* se denomina a menudo *prototipo desechable* y sirve únicamente como una demostración de los requisitos. Un enfoque de *prototipo abierto* se denomina prototipo evolutivo, y se emplea como una primera parte de una actividad de diseño a la que le sigue la construcción.

Antes de poder elegir un enfoque abierto o cerrado, es necesario determinar si se puede crear un prototipo del sistema a construir. A la hora de determinar esto, es necesario tener en cuenta los siguientes aspectos: área de aplicación, complejidad, características del cliente y del proyecto. Si una aplicación candidata con las características anteriormente mencionadas requiere el desarrollo de miles de líneas de código antes de poder realizar cualquier función demostrable, es muy probable que sea demasiado compleja para realizar un prototipo.

Para que la creación del prototipo de software sea efectiva, se debe desarrollar rápidamente, de tal forma que el cliente pueda valorar los resultados y recomendar cambios a tiempo. Existen tres métodos y herramientas para la creación de prototipos rápidos:

6.8.1 Técnicas de cuarta generación.

Estas técnicas de cuarta generación (4GT⁵³) comprenden una amplia gama de lenguajes de consulta e informes de bases de datos, generadores de programas, aplicaciones y de otros lenguajes no procedimentales de muy alto nivel. Como las técnicas 4GT permiten al desarrollador generar código ejecutable rápidamente, son ideales para la creación rápida de prototipos.

6.8.2 Componentes de Software Reutilizable.

Otro enfoque para crear componentes es ensamblar, más que construir, mediante un conjunto de componentes software existentes. Un componente software puede ser una estructura de datos (o una base de datos) o un componente arquitectónico de software (un programa) o un componente procedimental (un módulo). En todos los casos se debe diseñar el componente software de manera que permita ser reutilizado sin un conocimiento detallado de su funcionalidad.

6.8.3 Especificaciones formales y entornos para prototipos.

Los desarrolladores de los lenguajes de especificación, están desarrollando entornos interactivos que permitan al analista crear una especificación basada en lenguaje de un sistema o software y además invoquen herramientas automáticas que traducen la especificación basada en el lenguaje en código ejecutable y permitan al cliente usar el código ejecutable del prototipo para refinar los requisitos formales.

⁵³ 4GT: fourth generation techniques

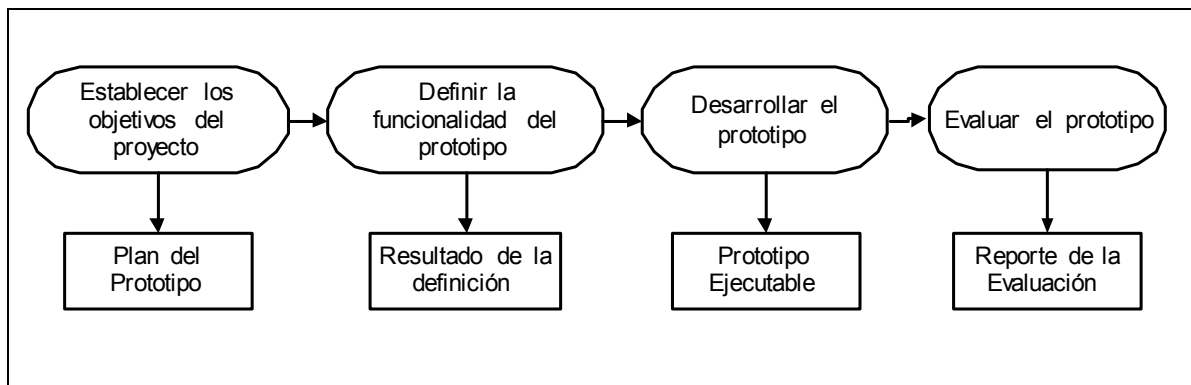


Figura 41. Proceso de prototipos.

6.9 DISEÑAR EL PLAN DE PRUEBAS

Las pruebas del software son elementos críticos para la garantía de la calidad del software y se presenta como una revisión final de las especificaciones, del diseño y de la codificación.

Según Glen Myers⁵⁴, la prueba tiene como objetivo los siguientes tres puntos:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

El cambio de paradigma que se plantea en estos objetivos, busca diseñar pruebas para descubrir errores y no para ignorarlos. Por otro lado, las pruebas son uno de los factores fundamentales para medir la calidad del software producido. Aunque si se tiene en cuenta que “La prueba no puede

⁵⁴ The Art of Software Testing, Wiley, 1979

asegurar la ausencia de defectos, solo puede demostrar que existen defectos en el software”⁵⁵, el equipo de trabajo estará más dispuesto a considerar diferentes aspectos que pueden generar errores.

Se deben tener en cuenta los siguientes aspectos:

- Hacer un seguimiento de las pruebas hasta los requisitos del cliente.
- Plantear y diseñar las pruebas antes de generar el código.
- El 80% de todos los errores se centran en sólo en el 20% de los módulos.
- Empezar las pruebas en módulos individuales y avanzar hasta probar el sistema entero.

Casos de prueba

El diseño de pruebas requiere de tanto conocimiento como el diseño del sistema en general. En muchas ocasiones es recomendado que los diseñadores de las pruebas, sean personas diferentes a los desarrolladores del sistema; ya que al no encontrarse involucrados en el proyecto, diseñaran pruebas que midan cualquier aspecto del software.

Los casos de prueba se pueden clasificar en tres partes, una parte procedimental denominada *prueba de caja blanca*, otra parte funcional operativa conocida como *prueba de caja negra*, y una tercera llamada *prueba de entornos y aplicaciones especializadas*.

⁵⁵ Ingeniería del Software, Roger S. PRESSMAN, Un Enfoque Práctico.

6.9.1 Prueba de Caja Blanca.

Evalúa que se hayan cumplido los requerimientos de los usuarios, enfocándose en la parte procedimental. Los casos de prueba dan como resultado los siguientes cuatro puntos:

1. Casos de prueba que garanticen la utilización de por lo menos una vez todos los caminos independientes de cada módulo.
2. Se ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
3. Se ejecuten todos los bucles en sus límites y con sus límites operacionales.
4. Se ejerciten las estructuras internas de datos para asegurar su validez.

Las justificaciones del porque de las pruebas de caja blanca son basadas en los estándares de la IEEE⁵⁶

- Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa.
- A menudo creemos que un camino lógico tiene pocas posibilidades de ejecutarse cuando, de hecho, se puede ejecutar de forma normal.
- Los errores tipográficos son aleatorios.

6.9.2 Prueba de Caja Negra.

Se enfoca en toda la parte funcional del software. A Diferencia de la prueba de la prueba de caja blanca, la prueba de caja negra busca errores en las

⁵⁶ JONES, T.C. Programming Productivity: Issues for the 80's, IEEE Computer Society Press, 1981.

funciones, en la interfaz, las estructuras de datos o en accesos a bases de datos externas, rendimiento, e inicialización o terminación.

Las pruebas de Caja Negra se diseñan para responder las siguientes preguntas:

1. ¿Cómo se prueba la validez funcional?
2. ¿Qué clases de entrada compondrán unos buenos casos de prueba?
3. ¿Es el sistema particularmente sensible a ciertos valores de entrada?
4. ¿De qué forma están aislados los límites de una clase de datos?
5. ¿Qué volúmenes y niveles de datos tolerará el sistema?
6. ¿Qué efectos sobre la operación del sistema tendrán combinaciones específicas de datos?

6.9.3 Prueba de Entornos y Aplicaciones especializadas.

Otras pruebas de importancia para la satisfacción del cliente aún más especializadas que las de caja blanca y caja negra son:

- Prueba de interfaces gráficas de usuario.
 - Ventanas.
 - Menús emergentes.
 - Operantes con el ratón.
 - Entrada de datos.
- Prueba de arquitectura Cliente/Servidor.
- Prueba de documentación y de ayuda.
- Prueba de sistemas de tiempo real.

6.10 REVISAR QUE LOS PROTOTIPOS CUMPLAN CON LOS REQUERIMIENTOS

Estas revisiones, lideradas por los Black Belts, sirven de filtro en el proceso de desarrollo, con el fin de detectar defectos para que puedan ser eliminados a tiempo y se pueda mejorar la calidad del software. Para llevar a cabo estas revisiones, se debe dedicar tiempo y esfuerzo por parte de todos los miembros del equipo.

7. FASE DE DESARROLLAR

La fase de *desarrollo* es la quinta fase en el modelo DMADDV. Esta fase se centra en el *Cómo*, es decir, cómo se implementa la función como una arquitectura del software, cómo han de implementarse los detalles procedimentales, cómo han de caracterizarse las interfaces y cómo ha de traducirse el diseño en un lenguaje de programación.

En la fase de desarrollo se implementa el sistema, es decir, se crea el código correspondiente al resultado de la fase de diseño, siguiendo los patrones y la arquitectura escogida. Uno de los puntos importantes a controlar en esta etapa consiste en la coordinación del equipo de desarrolladores del proyecto.

Los pasos que guiarán la fase de desarrollo son los siguientes:

1. Organización del código fuente.
2. Construcción.
3. Revisar que se satisficieron los requerimientos.
4. Documentación del código.
5. Diseñar el plan de implantación.
6. Ejecutar el plan de implantación.

7.1 ORGANIZACIÓN DEL CÓDIGO FUENTE

La organización de los archivos que contienen el código fuente son guardados utilizando una jerarquía establecida; aunque este paso es ignorado por muchos equipos de trabajo, es una tarea que convertirá al proyecto en un trabajo organizado, en el presente y en el futuro.

La organización del código fuente es independiente en cada proyecto, es recomendado tener en cuenta lo siguientes aspectos:

- Nombre de archivo: dar un nombre que al leerlo se entienda lo que contiene.
- Grupos o Subgrupos: clasificar en grupos o clases los archivos dependiendo de su objetivo o similitud, dividiéndolo las veces sea necesario hasta alcanzar un nivel de alto entendimiento.
- Gráficas: las gráficas utilizadas se deben almacenar en una carpeta diferente.

7.1.1 Árbol Jerárquico.

Un árbol jerárquico es la representación lógica de datos o archivos como se observa en la figura 42. Se grafica en forma inversa, la raíz se encuentra arriba y las hojas se despliegan hacia abajo. El símbolo “ / ” representa la raíz del directorio donde se encuentran los archivos, A, B son los dos grupos grandes de clases que existen, C es una subclase de A, en tanto que D, E, F, G, H, I, J representan los archivos.

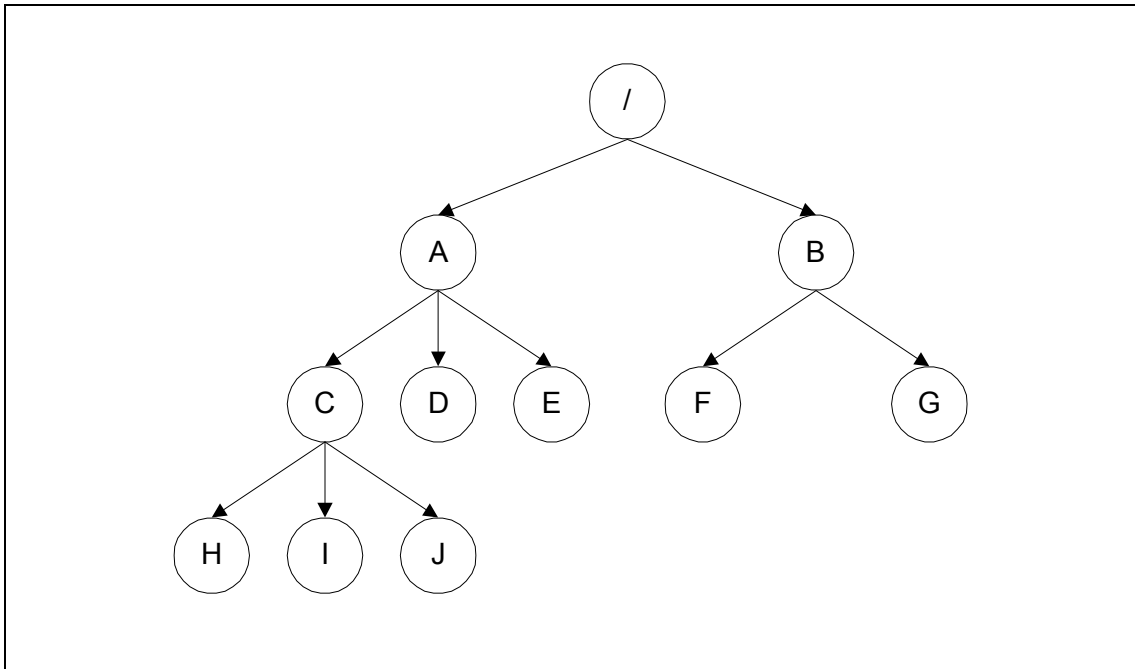


Figura 42. Estructura de un árbol jerárquico.

Después de definirse el árbol jerárquico de los archivos del código fuente, se debe entregar a cada uno de los miembros del equipo una copia impresa y realizar una breve explicación para evitar posibles confusiones.

De acuerdo con el árbol definido, se utilizan dos tipos de enlaces entre los archivos, los enlaces relativos o absolutos, siendo los más utilizados en proyectos que utilizan tecnologías de hipertexto.

7.1.2 Enlaces Absolutos.

Son aquellos enlaces que contienen la dirección absoluta de un archivo desde el directorio raíz hasta el archivo. Por ejemplo, utilizando la figura 42, y asumiendo que cada archivo son páginas jsp, se desea encontrar el archivo J.

<http://www.raiz.com/A/C/J.jsp>

7.1.3 Enlaces Relativos.

Son aquellos enlaces que contienen la dirección del archivo desde donde se encuentra ubicado hasta el archivo deseado. Para el ejemplo de la figura 42, ir de H -> D

../D.jsp

7.2 CONSTRUCCIÓN

Esta actividad incluye la codificación e integración de los módulos con técnicas de programación estructurada cumpliendo con los lineamientos especificados en la etapa del diseño.

La escritura del código debe ser sencilla, casi mecánica, debido a que se han tomado todas las decisiones durante el transcurso del proyecto.

Una vez generado el código fuente, la función de un módulo debe ser clara sin necesidad de referirse a ningún diseño, el código debe ser comprensible, es decir, debe mezclarse la simplicidad con la claridad.

Entre los elementos de estilo se encuentran la documentación interna (a nivel código fuente), los métodos de declaración de datos, enfoque de construcción de sentencias y las técnicas de entrada y salida.

7.3 REVISAR QUE SE SATISFACIERON LOS REQUERIMIENTOS

La revisión es conducida por el líder del proyecto y sirve de filtro en el proceso de desarrollo, con el fin de identificar defectos en el desarrollo para que puedan ser eliminados a tiempo y se pueda mejorar la calidad del software. Para llevar a cabo las revisiones, se debe dedicar tiempo y esfuerzo por parte de todos los miembros del equipo.

Otro tipo de documentos pueden ser utilizados de apoyo en la reunión de revisión, como por ejemplo:

- Definición del problema.
- Los requerimientos de los clientes (CTQ's).
- Determinar que medir.
- QFD.
- AMFE.
- Especificación del diseño.

7.4 DOCUMENTACIÓN DEL CÓDIGO

Este paso es propuesto para asegurar que el código si será documentado, debido a que la documentación permite a los desarrolladores comunicarse con otros, además también proporciona una clara guía de comprensión durante la última fase del proyecto (Validar).

Aunque el formato ya ha sido establecido en el paso de diseño de formato de documentación de programas, en la fase diseño, se deben tener en cuenta algunas características que se mencionan a continuación:

7.4.1 Legibilidad.

Es el primer aspecto que se debe tener en cuenta, para identificar variables y etiquetas. Aunque algunos lenguajes de programación limitan la longitud de los nombres de las variables o de las etiquetas a pocos caracteres, se debe buscar la mayor comprensión posible. Por ejemplo:

$$D = (H_D_C * V_H_N) + (H_D_D * V_H_D)$$

$$DEU = (HORS_D_NOC * VAL_HOR_NOC) + (HORS_D_DIA * VAL_HOR_DIA)$$

$$DEUDA = (HORAS DE DEMORA NOCHE * VALOR HORA NOCHE DEMORA) + (HORAS DE DEMORA DIA * VALOR HORA DIA DEMORA)$$

7.4.2 Ubicación de comentarios.

La ubicación se debe hacer de cierta forma que no ahogue el código pero tampoco lo ignore. Existen dos categorías, de prólogo y de descripción que poseen diferencias notables.

Los comentarios de prólogo se escriben al principio de cada módulo, indicando el propósito de la función del módulo o una explicación de los datos pertinentes como variables importantes, su uso, las restricciones y limitaciones.

Los comentarios descriptivos se incluyen en el cuerpo del código fuente y se usan para describir las funciones del procesamiento, proporcionando conocimiento extra que ofrezca un mayor entendimiento para los lectores

Además los comentarios descriptivos deben:

- Describir los bloques de código en vez de describir cada línea.
- Usar líneas en blanco o tabulaciones de forma que sean fácilmente distinguibles del código.
- Que sean correctos, un comentario incorrecto o que se pueda interpretar mal es peor que no ponerlo.

En Conclusión, la documentación de código fuente es una tarea que debe ser realizada. En su desarrollo surgen ciertas preguntas que el grupo debe resolver para un mejor trabajo en equipo como por ejemplo:

- ¿Cuántos comentarios son suficientes?
- ¿Dónde se deben situarse los comentarios?
- ¿Son los comentarios no mantenibles, y por lo tanto, no fiables?

7.5 DISEÑAR EL PLAN DE IMPLANTACIÓN

La implantación del sistema es uno de los últimos pasos de la metodología con gran importancia, por lo tanto se necesita de una planeación previa, ya que en la mayoría de los proyectos este paso se convierte en el más difícil de efectuar.

La complejidad de este paso depende directamente de la tecnología que se utilizó en el proyecto; si es un producto estándar será relativamente fácil y la familiarización con los usuarios pueda ser más rápida. Sin embargo, cuando la tecnología es nueva y las interfaces gráficas cambian notablemente, se debe tener demasiada precaución y una buena planeación para que el proyecto no falle al final.

La estrategia a escoger para la implantación es responsabilidad del líder del proyecto, los aspectos que se deben incluir son los siguientes:

- Descripción del ambiente (Servidores, Computadores, Bases de Datos, interfaces, etc.)
- Descripción de los datos necesarios.
- Programación de las fechas en que se realizará.
 - Fines de semana?
 - Última semana del mes?
 - Horas nocturnas?
 - Fecha definitiva de implantación completa.
- Orden de los pasos a seguir.

Las dos estrategias más conocidas para la implantación de proyectos de software son:

7.5.1 Corte-Rápido.

Busca una migración completa del sistema viejo al sistema nuevo.

Reemplazo Inmediato.

- Es el más rápido.
- Necesita un plan de reserva.
- Requiere de una fuerte planeación y pruebas.

Operación en Paralelo.

- Disminuye los riesgos.
- El corte ocurre cuando el sistema nuevo está completamente migrado.

7.5.2 Etapas.

Se implanta el sistema nuevo por etapas, de una en una, hasta completar el sistema. La etapa nueva que se implanta sustituye el funcionamiento de la vieja.

7.6 EJECUTAR EL PLAN DE IMPLANTACIÓN

La ejecución del plan de implantación, se puede realizar antes o después del paso de pruebas de la fase validar, según como el proyecto lo requiera. El paso de revisar que se satisficieron los requerimientos puede ser utilizado como prueba, para dar inicio a la implantación.

La estrategia de implantación escogida, debe ser entendible por el líder del proyecto para su ejecución. Los pasos a seguir marcarán el desenlace del proyecto.

- Conferir la responsabilidad de la implantación a un comité técnico y a uno administrativo.
- Recibir formalmente la versión definitiva del sistema.
- Armar y desplegar el sistema según se requiera.
- Seguir el orden de pasos programados en el plan.

Al terminar con la totalidad de los pasos programados para la implantación, se debe informar al líder del proyecto para proceder con la siguiente fase del proyecto.

8. FASE VALIDAR

En esta fase principalmente se validará el proyecto en su totalidad, el diseño, los procesos tratados además de establecer los controles que se deban tener para esos procesos, pero sobretodo verificar que se hayan cumplido las CTQ's y que la ecuación $Y = F(X)$ no contenga las variables que se han atacado durante todo el proyecto.

Los pasos que definen el desarrollo de esta fase son los siguientes:

1. Ejecutar el plan de pruebas.
2. Capacitar al personal.
3. Determinar los riesgos.
4. Entregar la documentación final.
5. Crear mecanismos de administración.
6. Evaluar los beneficios alcanzados.
7. Documentar las lecciones aprendidas.
8. Comunicar al equipo los resultados obtenidos.
9. Cierre del proyecto.

8.1 EJECUTAR EL PLAN DE PRUEBAS.

Este paso se realiza para culminar el plan de pruebas diseñado con anterioridad, y presentar los resultados al líder del equipo, los cuales deben ser manejados sistemáticamente. Las construcciones que posean defectos deben ser probadas nuevamente y posiblemente regresadas a la fase de diseño o construcción.

Como ayuda para guiar las pruebas se utilizan muchos de los casos de uso implementados para definir los requerimientos de los clientes, los cuales se convertirían en casos de prueba. El modelo de pruebas cambia constantemente debido a ciertos factores como:

- *La eliminación de casos de uso.* En la mayoría de los casos sucede esto por que el cliente no fue claro en la explicación. Otra posibilidad es que el proceso se haya cambiado durante el desarrollo del proyecto.
- *El refinamiento de algunos casos de uso.* Son adherencias que se les hacen a los casos de uso durante el transcurso del proyecto.
- *Creación de nuevos casos de uso.*

Para una mejor planeación del plan de pruebas, se recomienda asignarlas al equipo de trabajo, dependiendo de su tipo. Los diferentes tipos de pruebas que se pueden realizar se muestran en la tabla 18⁵⁷.

En la tabla 19 se propone una forma de registrar los resultados de las pruebas para presentar al líder del proyecto.

⁵⁷ Tabla 19: Tipos de Prueba: Tomado del libro "SIX SIGMA SOFTWARE DEVELOPMENT", Pág. 165.

Tabla 18. Tipos de Pruebas.

Tipos de Pruebas				
Tipo	Objetivo	¿Qué se Prueba?	¿Cuándo es realizado?	Responsables
Unidad.	Verificar que programa de cualquier modulo cumpla con los requerimientos.	Lógica de programación.	Tan pronto como el módulo es terminado.	Desarrolladores, control de calidad, clientes.
Sistema.	Verificar que todo el sistema opere de acuerdo con las especificaciones.	Interfaces con el sistema.	Cuando todos los módulos han sido probados y los errores hayan sido resueltos.	Desarrolladores, control de calidad, clientes.
Integración.	Se asegura que las interfaces entre la funcionalidad del sistema funcione correctamente según lo planeado y que el sistema encuentre los niveles de rendimiento bajo condiciones normales.	Interfaces desde y hacia sistemas externos; operan en el mundo real.	Cuando las pruebas de sistema son completadas.	Desarrolladores, control de calidad, clientes.
Resistencia.	Prueba los limites del sistema y el rendimiento bajo circunstancias extremas.	La robustez del sistema bajo altos volúmenes de datos y pocas fuentes disponibles.	Cuando las pruebas de integración son completadas.	Desarrolladores, control de calidad, clientes.
Aceptación.	Verificar que el sistema encuentre todos los requerimientos de los clientes definidos.	Funcionalidad y uso fácil.	Cuando las pruebas de integración son completadas.	Cientes, control de calidad.

La aceptación de las pruebas por el líder del proyecto es la última parte de este punto, en este momento el equipo debe estar completamente convencido de que el sistema corre correctamente bajo condiciones normales y extremas. Y que se hayan encontrado todos los requerimientos definidos por los clientes. La última prueba es realizada por clientes después de que el proyecto se encuentre en marcha y es que la documentación entregada a cada cliente este acorde con lo explicado en las capacitaciones.

8.2 CAPACITAR AL CLIENTE.

Capacitar al cliente es uno de los pasos más importante de la fase validar y uno de los más importantes de todo el modelo DMADDV de Seis Sigma. Es el momento en que el cliente compara sus expectativas sobre el proyecto con lo que esta siendo entregado, y determinan si fueron superadas o frustradas.

La capacitación se debe enfocar en los aspectos que son más relevantes para cierto grupo de clientes, con el fin de no invertir tiempo en aspectos que nunca se utilizarán.

La forma de capacitación que se utilice debe ser aprobada por el líder del proyecto y se debe ajustar al presupuesto. También es recomendable tener en cuenta el nivel de conocimiento de los clientes, por citar un ejemplo, no es recomendable utilizar una forma de capacitación basada en Web si los participantes han tenido un mínimo contacto con Internet. Las formas más comunes de realizar las capacitaciones son:

Tabla 20. Formas de Capacitación.

Formas de Capacitación				
Tipo	Descripción	Costo	Tiempo de preparación	Tiempo de ejecución
Presentación con video-beam	Realizar una presentación con diapositivas de power-point, flash, etc.	Bajo.	1 – 2 días.	1 – 2 horas.
Uno a uno	El guía va puesto por puesto de cada cliente explicando paso por paso el sistema	Medio	1 – 2 días	Meses
Clase.	Se programan clases con el grupo seleccionado.	Alto.	1 semana.	Mes.
Portal	Se crea un Web site con un manual mas descriptivo (gráficos, ejemplos dinámicos, etc.).	Medio-Alto.	1 Mes.	Indefinido

Es conveniente realizar una encuesta acerca del nivel de satisfacción después de efectuada la capacitación. Aunque resultado sea satisfactorio, esto no significa que el cliente haya retenido toda la información o que haya entendido todos los procedimientos del producto. En la tabla 21 se observa un ejemplo de encuesta que puede ser utilizado, la agregación o eliminación de preguntas es de acuerdo si el proyecto lo requiera.

Tabla 21. Encuesta de Capacitación.

Encuesta de Capacitación					
Ubicación:	Fecha:				
Instructor:					
Cliente: (opcional)					
Las siguientes preguntas son realizadas para medir el nivel de satisfacción del producto entregado, y la capacitación realizada.					
La calificación se realiza de 1 a 5, en donde 5 es el nivel mas alto de satisfacción y 1 el nivel mas bajo					
Sistema					
Tiempos de respuesta.	1	2	3	4	5
Integridad en los resultados.	1	2	3	4	5
Integración con otros sistemas.	1	2	3	4	5
Interfaz grafica					
Amigable.	1	2	3	4	5
Entendible.	1	2	3	4	5

Requerimientos					
Requerimientos encontrados.	1	2	3	4	5
Optimización de procedimientos.	1	2	3	4	5
Reportes Útiles.	1	2	3	4	5
Capacitación					
Material utilizado	1	2	3	4	5
Conocimiento del Instructor	1	2	3	4	5
Habilidades del Instructor	1	2	3	4	5

En la mayoría de los casos durante las capacitaciones los clientes identifican nuevos requerimientos para el sistema, o identifican más usos que el sistema podría tener y que pueden ofrecer un gran valor agregado al trabajo diario. Todos los comentarios realizados por los clientes deben ser tenidos en cuenta y ser registrados por el líder del proyecto, para una nueva versión del sistema.

8.3 DETERMINAR LOS RIESGOS

El riesgo que se maneja en este paso, es aquel que puede llegar a existir después de implantar el producto para uso de los clientes. La Administración de los riesgos en esta fase se puede hacer de la misma forma que fueron manejados los riesgos que puede existir durante el transcurso del proyecto, identificados en la fase Definir.

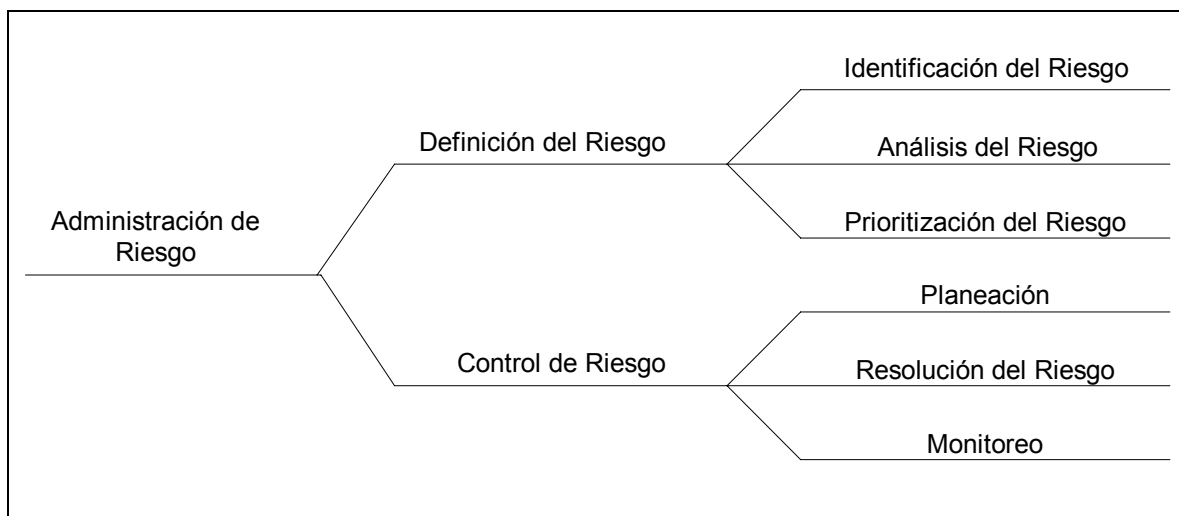


Figura 43. Administración del Riesgo.

Los tipos de riesgos que se pueden presentar son de tres categorías principalmente, riesgos del negocio, operacionales y técnicos. Sin importar cual sea su tipo, el trato debe ser el mismo para cada uno y debe ser comunicado al responsable del sistema en el momento más oportuno. Para establecer los posibles riesgos utilizamos el método de lluvia de ideas utilizado en la fase definir también.

8.4 DOCUMENTACIÓN DEL PROYECTO

La documentación es parte fundamental durante todo el desarrollo del proyecto. Ayuda principalmente a definir, planear, organizar, controlar y cerrar el proyecto. Aunque el exceso de documentación puede generar en algunos casos problemas, ésta se hace o no tan compleja dependiendo del tamaño y la importancia del proyecto.

Una buena documentación es una excelente herramienta de comunicación, así como también es una herramienta para analizar y guiar las revisiones del proyecto.

La documentación incluye los documentos que se han completado durante todo el desarrollo del proyecto, como son:

- Diagramas de Flujo.
- Cuadros de Proyecto.
- Plan de Proyecto.
- Documentos Técnicos (documentación del software, especificación de requisitos).

También incluye la documentación específica del cierre del proyecto, tales como:

8.4.1 Manuales del proyecto.

Es un libro esencial de referencia para tener cierta información disponible fácilmente. Sin embargo, más que proporcionar a sus lectores información útil, es también una herramienta de comunicación que le permite a la gente actuar eficientemente y efectivamente.

Para generar el manual de proyecto, es útil realizar los siguientes pasos:

1. Determinar el contenido, para ésto se pueden entrevistar a los miembros del equipo.
2. Organizar el contenido, se puede realizar ya sea por tópicos o fases.
3. Determinar el número de copias, usando el número de los miembros del equipo como base.
4. Asignar responsabilidades para el mantenimiento del manual.
5. Publicar y distribuir el manual.
6. Retroalimentación por parte de los usuarios del manual.

8.4.2 Libro del proyecto.

El libro del Proyecto, es como un historial de archivos y se encarga principalmente de almacenar información. Además de la información de carácter administrativo, contiene información de la compañía y sus políticas, procedimientos específicos del proyecto y la documentación técnica.

Al igual que para el manual del proyecto, es útil seguir los siguientes pasos para desarrollar el libro del proyecto:

1. Identificar el contenido.
2. Determinar la organización del contenido.
3. Controlar el retiro de los documentos, proporcionando una hoja de registro.
4. Determinar la localización de la biblioteca, proporcionando un sitio fácilmente accesible; también se determinan los procedimientos para tener acceso al material.

8.5 CREAR LOS MECANISMOS DE ADMINISTRACIÓN.

Una vez finalizado el proyecto, se deben tener en cuenta ciertos aspectos con el fin de administrar mejor el proyecto.

Formación: es preciso aprender como funciona este nuevo proceso y evitar los paradigmas del cambio para así romper con los hábitos antiguos.

Documentación: Se debe tener clara y documentada la forma en que se deben hacer las cosas y si es posible, disponer de las respuestas a las preguntas más frecuentes.

Solución de Problemas: Es conveniente dejar claras las responsabilidades respecto a quien o quienes se encargarán de tratar los aspectos que surjan.

Gestión del rendimiento: Mantener los ojos abiertos hacia la necesidad y oportunidad de revisar las descripciones de puestos de trabajo, incentivos y criterios de revisión del rendimiento.

Medidas: Se deben documentar siempre los resultados.

8.6 EVALUAR LOS BENEFICIOS ALCANZADOS

En la fase inicial del proyecto, se describió la forma de llevar a cabo el análisis de costo y beneficio. Es en este paso, donde se verificará que los beneficios planteados se han alcanzado.

Para llevar a cabo este paso, existen diferentes mecanismos, en la mayoría de los proyectos los Black Belts deciden utilizar los mismos mecanismos utilizados en la fase medir.

Primero volver a capturar los datos de la misma manera que en la fase dos, realizarles las mismas medidas y gráficos y observar si se alcanzó la meta propuesta.

Segundo, volver a capturar la voz del cliente, para dar fin al proyecto o plantear mejoras para alcanzar la satisfacción total del cliente.

El reporte de este paso es el primer reporte que el comité desea ver al presentar esta fase.

8.7 DOCUMENTAR LAS LECCIONES APRENDIDAS

Las lecciones aprendidas durante todo el proceso son en definitiva la salida final y se obtienen únicamente con el trabajo de solucionar problemas de la vida real. Su único propósito es documentar los logros y las fallas del proyecto, para evitar que se cometan los mismos errores en proyectos siguientes. Por ejemplo, las lecciones aprendidas documentan las razones del porque se han tomado ciertas acciones correctivas, o las causas de las variaciones del funcionamiento, la ocurrencia de riesgos no planeados, errores que sucedieron y que pudieron ser evitados, entre otros.

Desafortunadamente, los proyectos pueden fallar. También hay cosas que se pueden aprender de los proyectos que fallan, así como de los proyectos exitosos, y es precisamente esta información la que sirve de referencia en el futuro. Se debe tener en cuenta que aprender de las lecciones es una excelente oportunidad para obtener beneficios a todos los asociados en el proyecto.

Las lecciones aprendidas se deben implementar de las siguientes formas:

8.7.1 Secciones de lecciones aprendidas.

Además de comunicar el cierre del proyecto por escrito, es recomendable un mecanismo para revisión en grupo. Las secciones de lecciones aprendidas son una serie de reuniones en las que participan todo el equipo de trabajo, los directores y los dueños del proyecto.

Para que estas secciones sean exitosas, todos los problemas encontrados por el equipo de trabajo deben ser presentados abiertamente. Aquellos problemas que son encontrados deben priorizarse. No es necesario

documentar todo pequeño detalle que sucede, sin embargo las cosas importantes se deben discutir sea por solicitud de los clientes o los directores del proyecto.

8.7.2 Formato de las lecciones aprendidas.

Existen numerosos formatos para documentar las lecciones aprendidas. Generalmente, cada lección aprendida debe ser documentada en una sola página y debe contener principalmente: en el encabezado el nombre del proyecto, la fecha y el punto de contacto para la lección aprendida. El cuerpo debe describir la lección aprendida de la siguiente forma:

- Declaración del problema: Se debe describir el problema que ocurrió y proporcionar información suficiente para establecer que fue lo que sucedió.
- Discusión: Describir en detalle la causa y el impacto del problema.
- Referencias: Proporcionar las referencias usadas o cualquier fuente de información que pueda ser útil para entender el problema y tomar las acciones correctivas.
- Acciones correctivas: Identificar cuales acciones correctivas fueron tomadas y discutir los resultados.

8.8 COMUNICAR AL EQUIPO DE TRABAJO LOS RESULTADOS OBTENIDOS

La organización debe creer firmemente en la importancia de reconocer el resultado que el equipo alcanzó. Esta es la oportunidad para celebrar y reconocer oficialmente los esfuerzos y agradecer por su participación. Una celebración ayuda a los miembros del equipo a reconocer formalmente que el proyecto ha finalizado y se da por cerrado el trabajo que ellos han hecho.

Además, los anima a recordar lo que aprendieron durante el desarrollo del mismo y sobre las experiencias que pueden considerar beneficiosas tanto a ellos como a la organización en los proyectos siguientes.

8.9 CIERRE DEL PROYECTO

El cierre del proyecto es la última fase dentro del ciclo de vida del proyecto y es uno de los pasos más importantes durante todo el desarrollo del mismo. Comienza una vez el director del proyecto concluya que se alcanzaron los objetivos planteados y el cliente y los usuarios aceptan los resultados entregados. El cierre del proyecto es importante, pues toda la información del proyecto se recoge en este paso y es la que servirá de referencia en un futuro. La documentación recogida durante el cierre del proyecto puede ser reutilizada por otros proyectos para prevenir problemas en el futuro. También se ejecuta el cierre del contrato y la aceptación formal y aprobación por parte de los dueños del proyecto.

9. FASE DEFINIR

9.1 ENTENDER EL MODELO DEL NEGOCIO

El proyecto “Consulta del catálogo Bibliográfico vía Web” se enmarca dentro del contexto de la **Biblioteca de la Universidad Industrial de Santander**. Para entender mejor el modelo del negocio con el que se va a interactuar, es necesario conocer la misión, visión y objetivos de la Biblioteca.

MISIÓN

Ser un centro integral de información capaz de satisfacer y anticiparse a las necesidades de documentación de la comunidad universitaria, académica e investigativa a nivel regional, nacional e internacional, mediante la prestación de servicios de adquisición, procesamiento, recuperación y disseminación de información con criterios de calidad. Para ello se apoya en la utilización de tecnología moderna y talento humano idóneo, constituyéndose de esta forma en líder del desarrollo y promoción de actividades intelectuales que estimulen procesos de enseñanza y de aprendizaje

VISION

La Biblioteca de la Universidad Industrial de Santander sería un sistema conectado a la red mundial de información, mediante una infraestructura digital que permita nuevas formas de conocimiento que contribuyan a la formación integral de los usuarios. Así mismo, se espera lograr un posicionamiento local, regional e internacional para ofrecer servicios abiertos, dinámicos y oportunos, como soporte principal a la academia e investigación. El concurso de un equipo humano interdisciplinario, competente y comprometido con la institución, además de la utilización de una metodología innovadora, serán factores vitales para lograr un ambiente adecuado y garantizar la calidad de sus servicios

OBJETIVOS

- Ofrecer servicios de información con criterios de calidad para satisfacer las necesidades de los usuarios.
- Apoyar la docencia, la investigación y la extensión a través del suministro de información oportuna, utilizando tecnologías apropiadas que estimulen procesos de enseñanza y aprendizaje.
- Generar las condiciones adecuadas que permiten el manejo de la información acorde con los avances del siglo XXI.
- Generar en los usuarios de la biblioteca una cultura de lectura.
- Apoyar la política de regionalización de la universidad, en lo relacionado con las unidades de información

Según esto, se definió que la Biblioteca de la UIS, en función de esa misión, visión y objetivos, encamina todos sus esfuerzos en la iniciativa de prestar un mejor servicio a todos los usuarios.

Y = Mejoramiento de servicios

Los servicios que ofrece la biblioteca a sus usuarios en general, son los que se mencionan a continuación:

- Acceso y consulta a las colecciones.
- Catalogo de consulta en línea.
- Consulta de base de datos.
- Consulta de videos, audio-cassettes y microformas.
- Referencia.
- Préstamo.
- Préstamo ínter bibliotecario.
- Suministro de documentos y bibliografías.
- Conmutación bibliográfica.
- Reproducción de documentos.

9.2 DEFINIR EL PROBLEMA

La forma en que se puede hacer más fácilmente la definición del problema, es encontrando la respuesta a las preguntas que se hacen en los cuadros a continuación:

¿QUE?	
¿Qué proceso está implicado?	Catálogo Bibliográfico Web.
¿Qué no está funcionando?	Las búsquedas a través del sistema Web de Libruis.
¿Cuál es la deficiencia?	El sistema sólo tiene una forma de realizar la consulta, los servicios que presta son muy pocos, y los tiempos de respuesta son altos.

¿DONDE / CUANDO?	
¿Dónde se observa el problema o deficiencia?	Campus, red externa. En el catálogo existente.
¿Cuándo se observa el problema o deficiencia?	Actualmente.

IMPACTO	
¿Cuál es el impacto de la deficiencia?	Usuarios insatisfechos.
¿Cuáles son los beneficios de actuar o las consecuencias de no actuar?	La consecuencia de no actuar, va en contra del objetivo principal de la biblioteca: mejoramiento de servicios.

Según esto, la definición del problema se puede plantear de la siguiente forma:

“Actualmente, los usuarios de la biblioteca realizan las consultas del material bibliográfico por medio del catálogo bibliográfico, situado en la página Web e Intranet de la UIS. El problema es que los usuarios se encuentran insatisfechos por el servicio que presta y se quejan constantemente”.

9.3 DEFINIR ROLES Y RESPONSABILIDADES

Las personas que estarán directamente involucradas en el proyecto con sus distintos roles son:

Tabla 22. Roles y responsabilidades.

Responsabilidades	Personas Responsables
Director DSI (Patrocinador)	Ing. Enrique Torres López
Director Biblioteca (Cliente)	Ing. Gladys Hernández
Gerente del Proyecto	Ing. Enrique Torres López
Responsable del Proyecto en la DSI	Ing. Enrique Torres López
Responsable del Proyecto en la Biblioteca	Ing. Gilberto Rivas Ing. Yamile Barragán
Master Black Belt	Ing. Edwin Suárez
Black Belts	Sergio Contreras Diana Zambrano
Green Belts	Ing. Gilberto Rivas Ing. Yamile Barragán
Equipo de Trabajo (Analista de Sistemas, Arquitecto, Diseñador, Desarrollador)	Sergio Contreras Diana Zambrano Yamile Barragán
Representante de los usuarios finales.	Yamile Barragán.

Las responsabilidades de los GB, BB y MBB se encuentran documentadas en la primera parte, en la fase definir de la metodología.

9.4 FORMAR EL EQUIPO DE TRABAJO

Una vez identificados los responsables de cada rol, comienzan a ser parte del equipo de trabajo, cuya estructura organizacional está representada Gráficamente de la siguiente forma (figura 44):

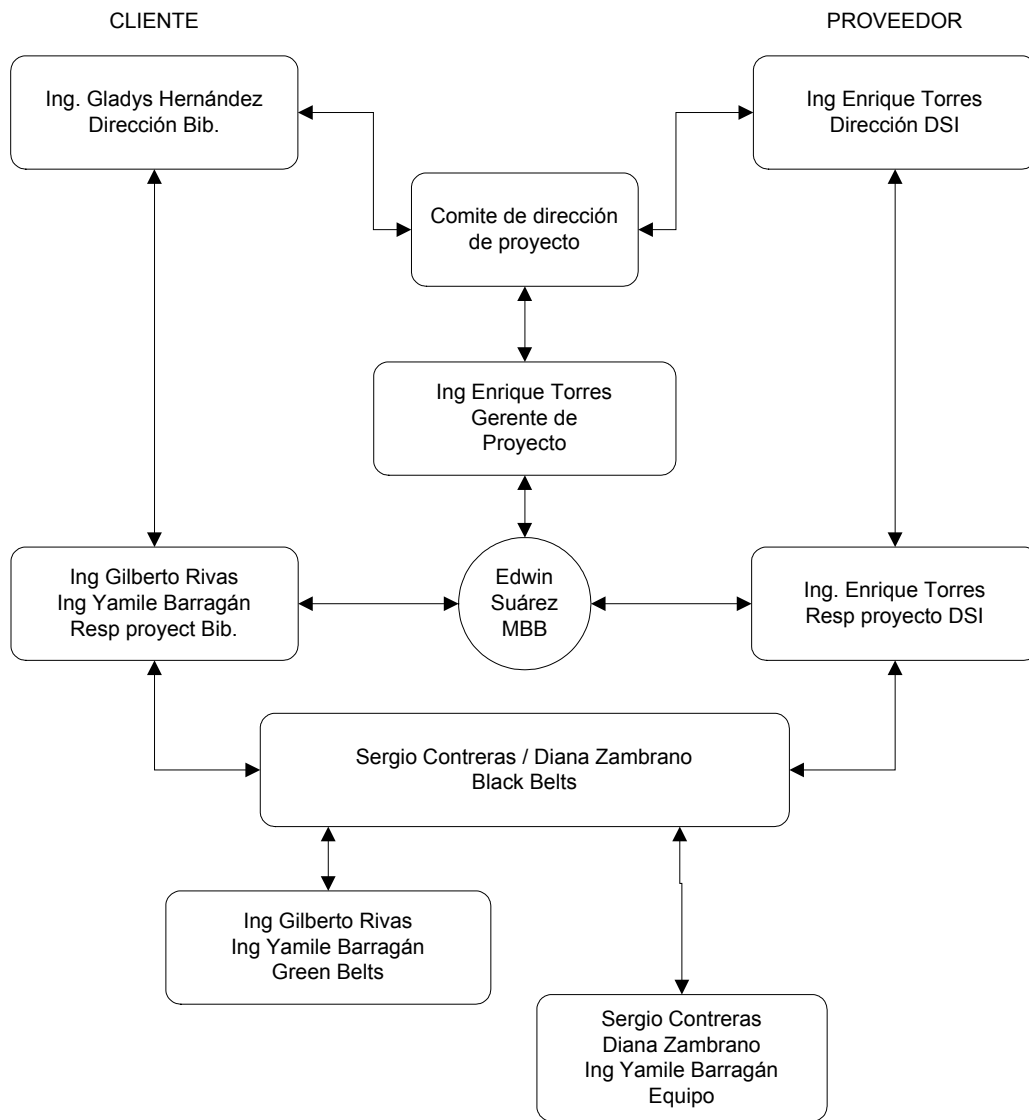


Figura 44. Modelo organizacional del proyecto.

9.5 IDENTIFICAR LOS CLIENTES

La identificación de los clientes fue realizada con el método tradicional de sondeo dentro de la universidad, y con la ayuda del participante de la biblioteca que está representando a los clientes en el proyecto. Los métodos de nueva generación no aplican debido a que la envergadura del proyecto es pequeña y no lo justifica.

Tabla 23. Identificación de clientes.

	Nombre	Descripción	Frecuencia
CLIENTES EXTERNOS	Estudiantes UIS.	Se encuentran los estudiantes de todas las carreras de la Universidad, de primer nivel hasta estudiantes en proyecto de grado.	Alto.
	Profesores.	Aquellos que tienen vínculos con la Universidad.	Medio.
	Empleados Administrativos.	Personal administrativo de la Universidad.	Bajo.
	Personas naturales.	Son los estudiantes, profesores y demás personas que no tienen vínculos directos con la UIS, pero utilizan el sistema para consultar si existe material que les pueda interesar.	Bajo.
CLIENTES INTERNOS	Directivas.	Personal administrativo de la biblioteca vinculado con el área de información tecnológica.	Bajo.

9.6 DETERMINAR LO CRÍTICO A ASEGURAR EN LA CALIDAD (CTQ).

La determinación de las CTQ's se basó en diferentes aspectos, el primero de ellos es la opinión realizada por los representantes de los clientes, el gerente del proyecto y el equipo de trabajo. Los miembros de los dos primeros roles se basaron en la experiencia y las constantes quejas realizadas por los usuarios finales acerca del sistema actual, los otros miembros participantes utilizaron la lógica y la experiencia en cosas específicas. Los otros aspectos son analizados más adelante en el proyecto.

El mecanismo utilizado fue el desarrollo del árbol de soluciones, el cual contó con más de tres versiones, realizados en sesiones diferentes hasta llegar a su aprobación o depuración para así alcanzar la versión más acertada. El árbol de soluciones es el de la figura 45.

Después de que el árbol llegó a su versión final, lo crítico para asegurar la calidad fue:

- El tiempo de respuesta.
- Los servicios que presta.
- Las formas de realizar la consulta.

Debido a que en el modelo DMADDV de Seis Sigma, las CTQ's deben ser definidas de una forma EMART⁵⁸, la definición de cada una es la siguiente:

- Disminuir el tiempo de respuesta actual en un 20%.
- Igualar el número de servicios que presta el catálogo bibliográfico Web (2) con el servicio de búsqueda de material bibliográfico informix (5).
- Disminuir en un 25% la cantidad de intentos que realiza un usuario para obtener un resultado en la consulta.

Los números utilizados en la definición de cada una de las CTQ's son estimados y pueden cambiar después de completarse la fase medir y ser presentada al comité para su aprobación.

⁵⁸ EMART: Especifica, Medible, Alcanzable, Relevante, Tiempo Limite.

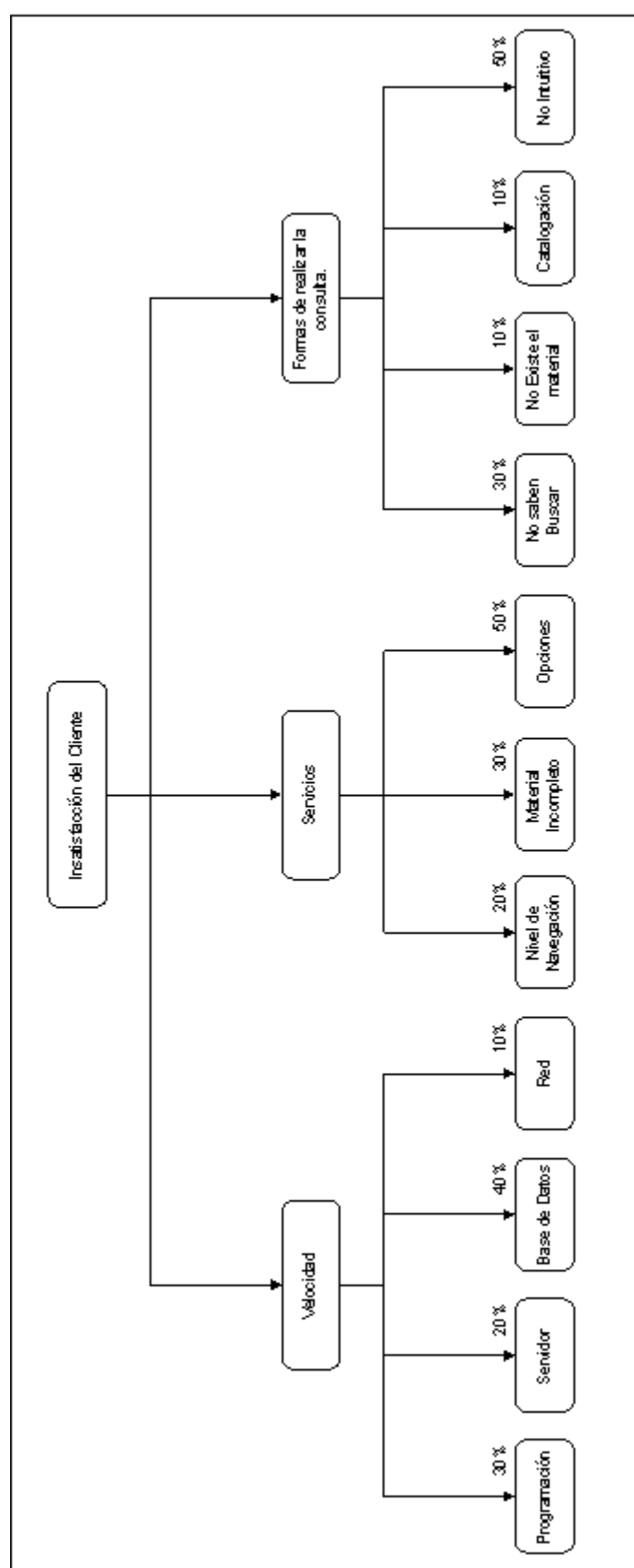


Figura 45. Diagrama de Arbol de soluciones.

9.7 DESARROLLAR EL PLAN DE PROYECTO

Los ítems más importantes que se deben definir en el desarrollo del plan son los siguientes:

9.7.1 Alcance y objetivos del proyecto.

Objetivo General

Desarrollar un prototipo del módulo de consulta de catálogo bibliográfico “LIBRUIS” a través de la interfaz del sistema de información web de la UIS.

Objetivos Específicos

- Disminuir el tiempo de respuesta actual en un 20%.

- Igualar el número de servicios que presta el catálogo bibliográfico Web (2) con el servicio de búsqueda de material bibliográfico informix (5).

- Disminuir en un 25% la cantidad de intentos que realiza un usuario para obtener un resultado en la consulta.

9.7.2 Costo/Beneficio.

El costo/beneficio del proyecto es de tipo cualitativo, debido a que se realizará dentro una dependencia que pertenece a una institución sin ánimo de lucro, por lo tanto los beneficios se analizan de la siguiente manera.

La gran iniciativa que tiene la Biblioteca de Universidad es el excelente servicio que presta a los usuarios. El catálogo bibliográfico entra a formar parte de la lista de proyectos que se desarrollarán para cumplir la iniciativa.

La ecuación $Y = F(x)$ ayuda a clarificar la decisión.

La Y definida del proyecto es el excelente servicio, la X que abarcaría el proyecto es la optimización del catálogo bibliográfico a través de una interfaz Web, al relacionarlas se obtiene.

Excelente Servicio = Optimización del catálogo bibliográfico.

En conclusión, el proyecto de optimización del catálogo bibliográfico de la universidad, si tiene un efecto sobre la gran iniciativa que se tiene en estos momentos por parte de la biblioteca.

9.7.3 Riesgos.

Tabla 24. Riesgos definidos en el plan.

TIPO DE RIESGO	DEFINICIÓN	POSIBLES RIESGOS
<i>Tamaño del producto.</i>	Riesgos asociados con el tamaño general del software a construir o modificar.	<ul style="list-style-type: none"> • Tamaño estimado del producto. • Tamaño de las Bases de Datos creadas o empleadas por el producto. • Cantidad de software reutilizado.
<i>Impacto en el negocio.</i>	Riesgos asociados con las limitaciones de la gestión o del mercado.	<ul style="list-style-type: none"> • Efecto del producto en los ingresos de la compañía. • Número de clientes que usarán este producto. • Costos asociados por un retraso en la entrega. • Costos asociados por un

		producto defectuoso.
<i>Características del cliente.</i>	Riesgos asociados con el cliente y la habilidad del desarrollador para comunicarse con él en los momentos oportunos.	<ul style="list-style-type: none"> • Clientes con diferentes necesidades. • Clientes con diferentes personalidades. • Contradicción en los Clientes. • Colaboración de los clientes en el proceso.
<i>Definición del proceso.</i>	Riesgos asociados con el grado de definición del proceso del software y su seguimiento por la organización de desarrollo.	<ul style="list-style-type: none"> • Está bien definido el proceso de software. • El análisis, diseño y pruebas se realizan sobre la marcha. • El equipo estima importante el concepto de calidad.
<i>Entorno de desarrollo.</i>	Riesgos asociados con la disponibilidad y la calidad de las herramientas que se van a emplear en la construcción del producto.	<ul style="list-style-type: none"> • El código generado por las herramientas CASE es ineficiente. • Incapacidad para integrar algunas herramientas.
<i>Tecnología a construir.</i>	Riesgos asociados con la complejidad del sistema a construir y la tecnología de punta que contiene el sistema.	<ul style="list-style-type: none"> • El software interactúa con hardware nuevo o no probado. • Base de datos utilizada no tiene el rendimiento esperado.
<i>Tamaño y experiencia del grupo de Trabajo.</i>	Son los riesgos asociados con la experiencia técnica y de proyectos de los ingenieros del software que van a realizar el trabajo.	<ul style="list-style-type: none"> • Disposición de la mejor gente. • El personal tiene los conocimientos adecuados. • Se cuenta con suficiente personal.

9.7.4 Cronograma.

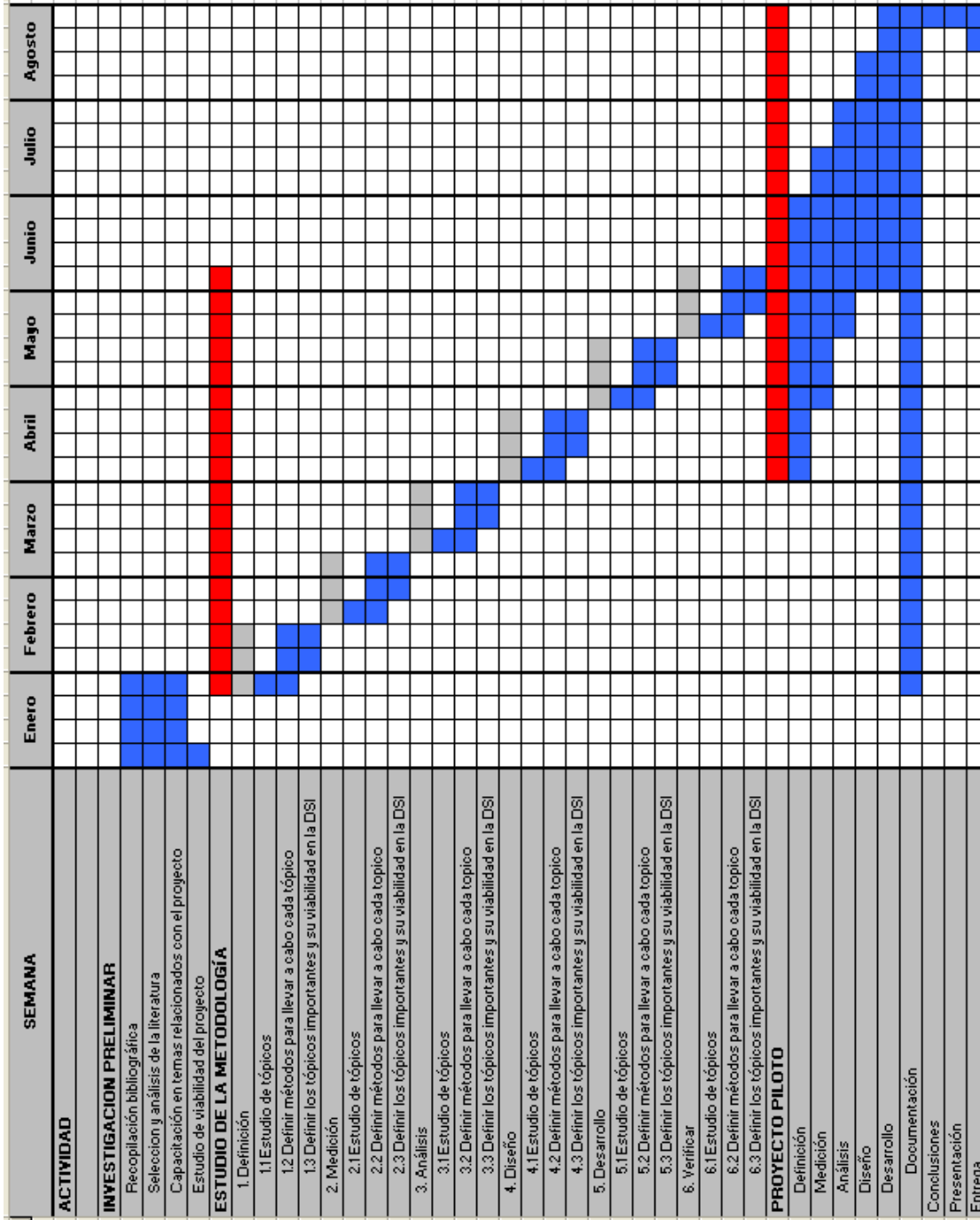


Figura 46. Cronograma.

9.8 CUADRO RESUMEN DEL PROYECTO

CUADRO RESUMEN DE PROYECTO					
Descripción del Proyecto					
Fecha Inicio:	Enero 2 / 2004	Fecha estimada fin:	Agosto 15 / 2004		
Título del Proyecto:	MODULO DE CONSULTA DEL CATÁLOGO BIBLIOGRÁFICO EN LA INTERFACE WEB DEL SISTEMA DE INFORMACIÓN DE LA BIBLIOTECA DE LA UIS, LIBRUIS.				
Definición del Problema:	"Actualmente, los usuarios de la biblioteca realizan las consultas del material bibliográfico por medio del catálogo, situado en la página Web e Intranet de la UIS. El problema es que los usuarios se encuentran insatisfechos por el servicio que presta y se quejan constantemente".				
Declaración de los objetivos	<ul style="list-style-type: none"> ▪ Disminuir el tiempo de respuesta actual en un 20%. ▪ Igualar el numero de servicios que presta el catálogo bibliográfico Web (2) con el servicio de búsqueda de material bibliográfico informix (5). ▪ Disminuir en un 25% la cantidad de intentos que realiza un usuario para obtener un resultado en la consulta. 				
Beneficios					
	Unidades	Medida Actual	Meta	Fecha proyectada	
Tiempo de Respuesta	Segundos	9 seg.	7.2 seg.	Ago/20/2004	
Numero de Servicios	SI / NO	2	5	Ago/20/2004	
Intentos de consulta	Intentos	2,07	1,55	Ago/20/2004	
Miembros del Equipo					
	Nombre	Rol	Departamento	% Tiempo	
	Enrique Torres López	Gerente	DSI	10%	
	Gladys Hernández	Gerente Cliente	Biblioteca	5%	
	Gilberto Rivas	Green Belt	DSI	40%	
	Sergio Contreras	Black Belt	DSI	100%	
	Diana Zambrano	Black Belt	DSI	100%	
	Yamile Barragán	Gerente IM	Biblioteca	20%	
	Edwin Suárez	Master Black Belt	Externo	70%	
Cronograma					
	Entregables	Fecha inicio	Fecha Fin	Responsables	Comentarios
	Fase Definir	Abr/05/2004	Jun/23/2004	D, S, G, Y, E.	
	Fase Medir	Abr/18/2004	Jul/17/2004	D, S	
	Fase Analizar	May/17/2004	Jul/31/2004	D, S	
	Fase Diseñar	Jun/01/2004	Ago/11/2004	D, S	
	Fase Desarrollar	Jun/01/2004	Ago/28/2004	D, S	
	Fase Verificar	N.A.	N.A.		
Aprobaciones					
	Rol/Título	Nombre	Fecha		

Revisiones		
Revisión número	Autor	Fecha

9.9 DEFINIR UN VOCABULARIO

La definición del vocabulario se realizó basada en los conceptos de los representantes de los clientes, el gerente del proyecto y los Black Belts.

Los conceptos finales más relevantes son los siguientes:

- Tiempo de respuesta de una consulta: es el tiempo que gasta el resultado de una consulta en ser enviado por el servidor de aplicaciones.
- Formas de realizar una consulta: es la sintaxis que el sistema acepta válida para realizar la consulta.
- Intuitivo: es la comprensión que el sistema tiene para entender al usuario final.
- Servicios: es lo que el sistema ofrece al usuario final para utilizar en el sistema.

9.10 IDENTIFICACIÓN DEL PROCESO ACTUAL

En el proceso actual, el usuario final inicia cuando selecciona los criterios de búsqueda, (Autor, Título o Materia), el tipo de documento (Libro, Tesis, Analítica) y digita las palabras clave a buscar.

El sistema muestra un listado de resultados basado en las palabras clave digitadas y los criterios seleccionados, el usuario observa los detalles

específicos de un material haciendo click en el título o en el número que se le asignó a la referencia en el listado.

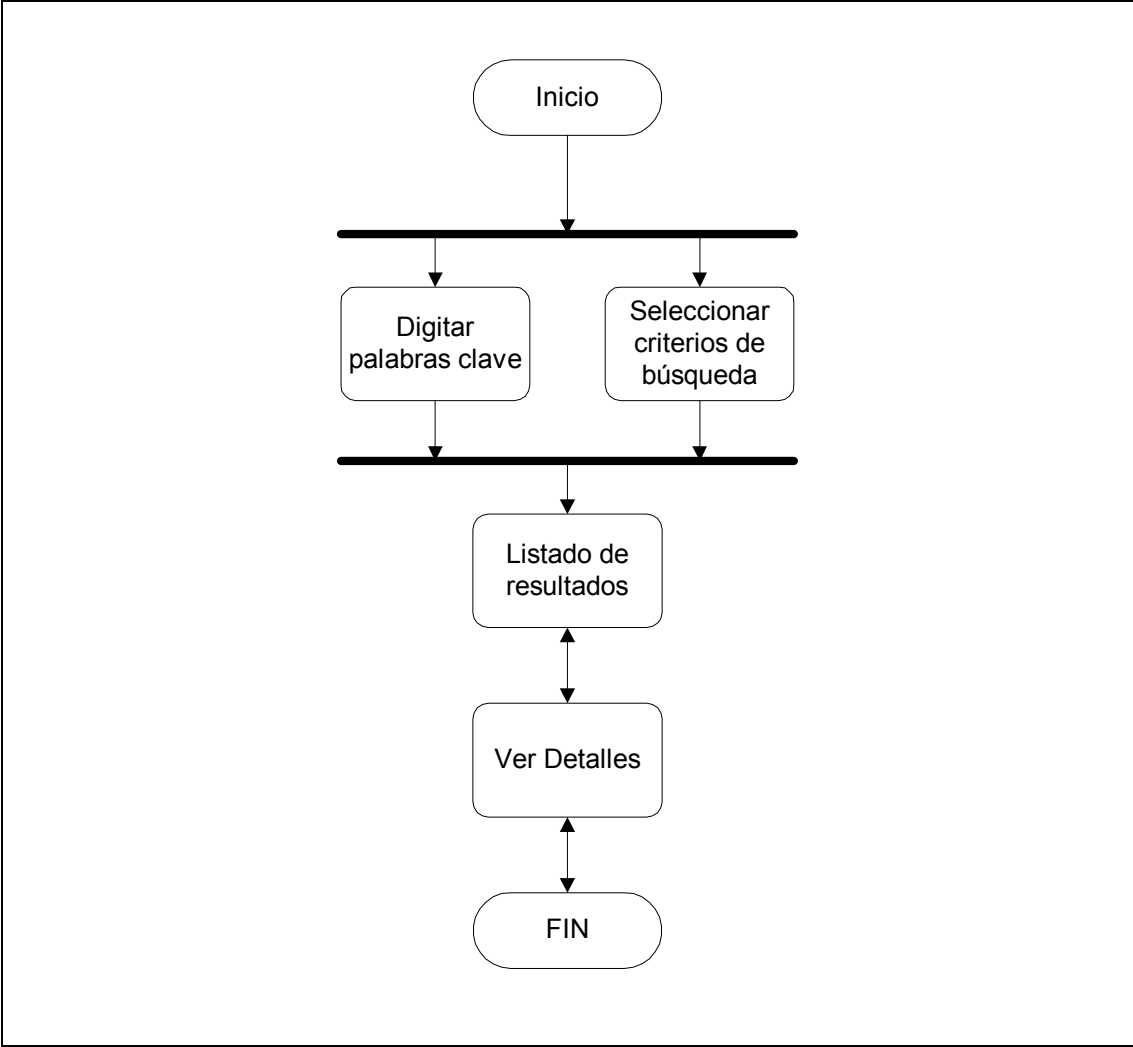


Figura 47. Proceso Actual

9.11 PRESENTAR AL COMITE PARA APROBAR

Los aspectos que el comité debe tener presente para aprobar el proyecto y dar luz verde al inicio de este, son:

- La declaración del problema.
- Las CTQ's.
- La ecuación $Y = F(x)$
- El costo/beneficio.
- El presupuesto.
- El árbol de soluciones.

10. FASE MEDIR

10.1 CAPTUAR LA VOZ DEL CLIENTE (VOC)

Los mecanismos utilizados para capturar la voz del cliente fueron la entrevista y la encuesta vía Web. Fueron seleccionados ya que se presentaban como los mecanismos que se acomodaban a las necesidades del proyecto en términos de presupuesto, tiempo y resultados.

La encuesta vía Web presenta como desventaja la demora en la obtención de los datos, aunque el promedio de encuestas recibidas diariamente fueron de 40 encuestas por día, considerado por el director del proyecto como un número aceptable para la captura de datos.

Los resultados de la VOC obtenidos según los mecanismos utilizados son los siguientes:

- **Satisfacción del cliente con el sistema**

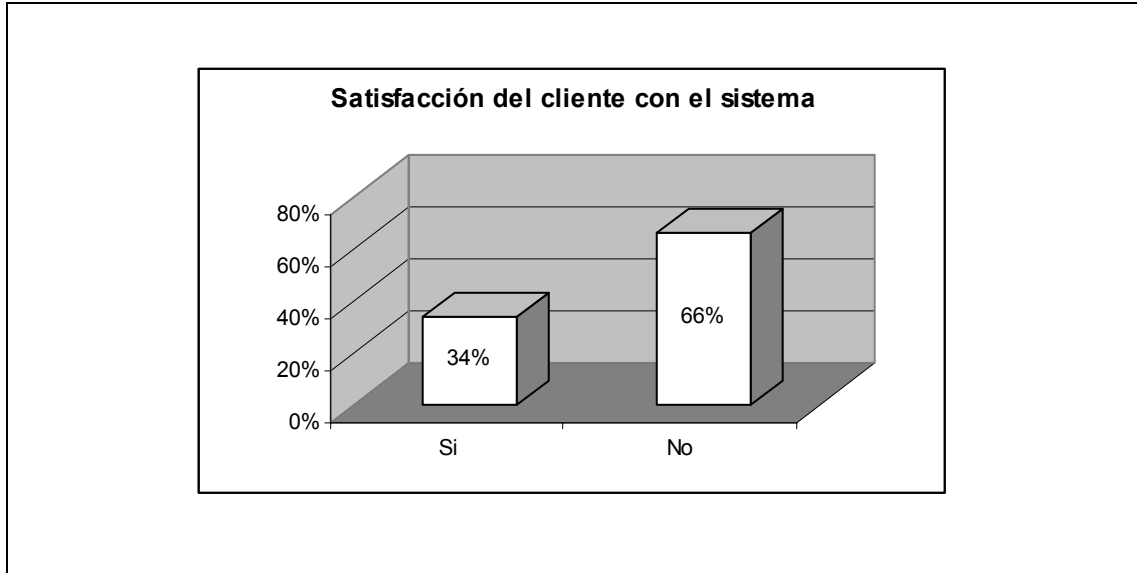


Figura 48. Resultados de la VOC.

- **Causas de la insatisfacción del cliente**

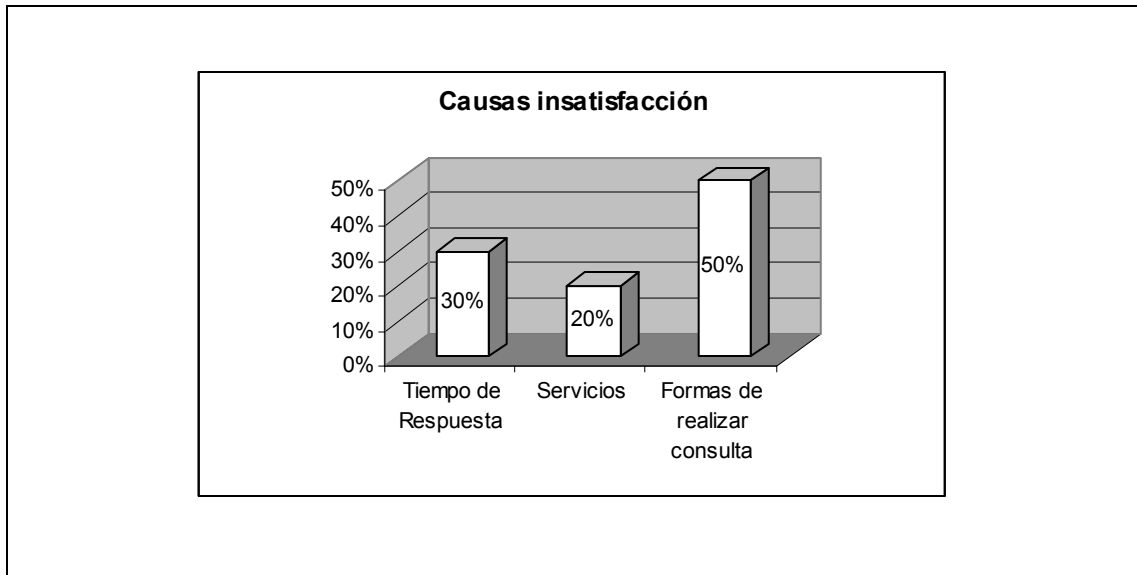


Figura 49. Porcentaje de quejas del cliente.

Después de tener en cuenta los resultados obtenidos por los representantes de los clientes y la encuesta realizada a los usuarios finales, el árbol de soluciones definitivo es observado en la figura 50.

El análisis de las métricas, el plan de colección de datos, las graficas, el cálculo del nivel de sigma y el Benchmarking se hace de manera individual para cada CTQ, como se muestra a continuación.

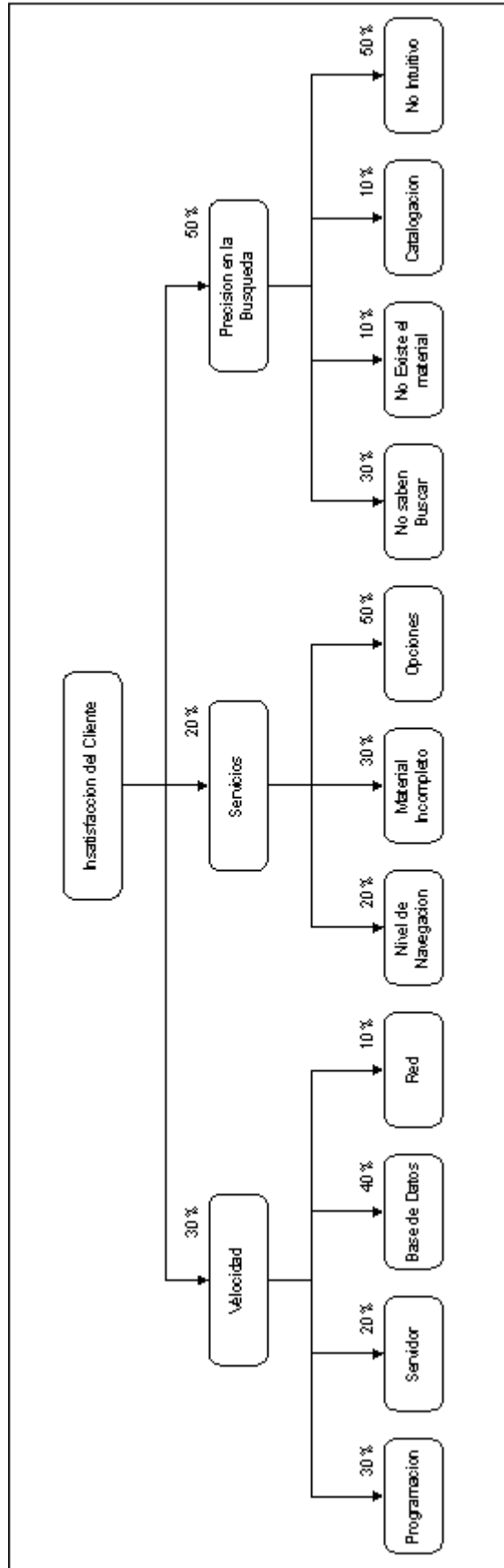


Figura 50. Árbol de soluciones.

10.2 CTQ1

10.2.1 Determinar que medir.

CTQ	Disminuir el tiempo de respuesta actual en un 20%.	Medida		Tiempo de Respuesta (seg.)	
		Actual	Meta		
Escenario	La medida es tomada para cualquier usuario ubicado en la LAN de la universidad o a través de un servidor de servicios de Internet externo por línea conmutada.	Media		9	
Comentario	El tiempo final es calculado al momento que la respuesta sale del servidor de aplicaciones, no se calcula cuando le aparece el resultado al usuario ya que no se puede garantizar que la velocidad del proveedor de servicios de Internet del usuario sea rápida y no es medible.			Wlbruis	
		NU		Bench	
				832	
		NUD		N/A	
		NO		245	
				N/A	
		Nivel sigma		26	
				N/A	
Plan de colección de datos					
Método	Tiempo	Fuente de los Datos	Recursos	Costo	Responsables
Control	1 mes	Web	1 servidor	0\$	Diana Zambrano Sergio Contreras
Descripción	Se calcula el tiempo de petición del usuario y el tiempo de respuesta de la petición por medio de dos variables que retornan el tiempo del servidor de aplicación y se insertan en una base de datos estadística.				

10.2.2 Plan de Colección de Datos

La utilización de variables de tiempo que se ejecutan en el servidor, ayudan a medir los tiempos de petición de la consulta, y los tiempos de respuesta del servidor. La función principal de la base de datos es medir las estadísticas necesitadas para el proyecto. La tabla que se diseñó para el tiempo de respuesta es la siguiente:



id_estadistica
ip_consulta
por_autor
por_titulo
por_materia
en_libro
en_tesis
en_analitica
tiempo_inicial
tiempo_final
fecha_consulta
ingreso_detalle

Figura 51. Tabla de estadísticas de tiempo.

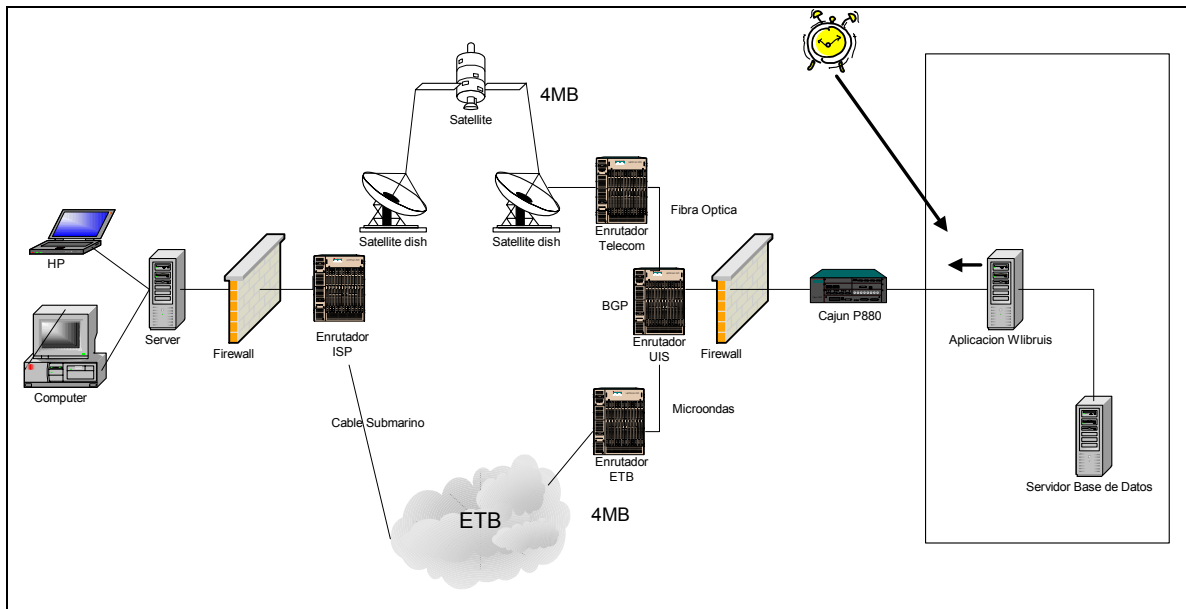


Figura 52. Situación actual, y posición de captura de tiempo.

El tiempo de respuesta se toma donde indica la flecha con dirección hacia la izquierda ←, debido a que es lo que con certeza se puede medir de una manera confiable, se garantiza que la velocidad de conexión hasta el otro proveedor de servicios de Internet ya sea por medio del satélite o de el cable subterráneo es rápida, pero no se puede garantizar después del proveedor de servicios del cliente.

10.2.3 Gráficas.

- Gráfica de corrido: representa el tiempo de respuesta en cada consulta realizada. Se puede observar la gran variación que existe respecto de la media.

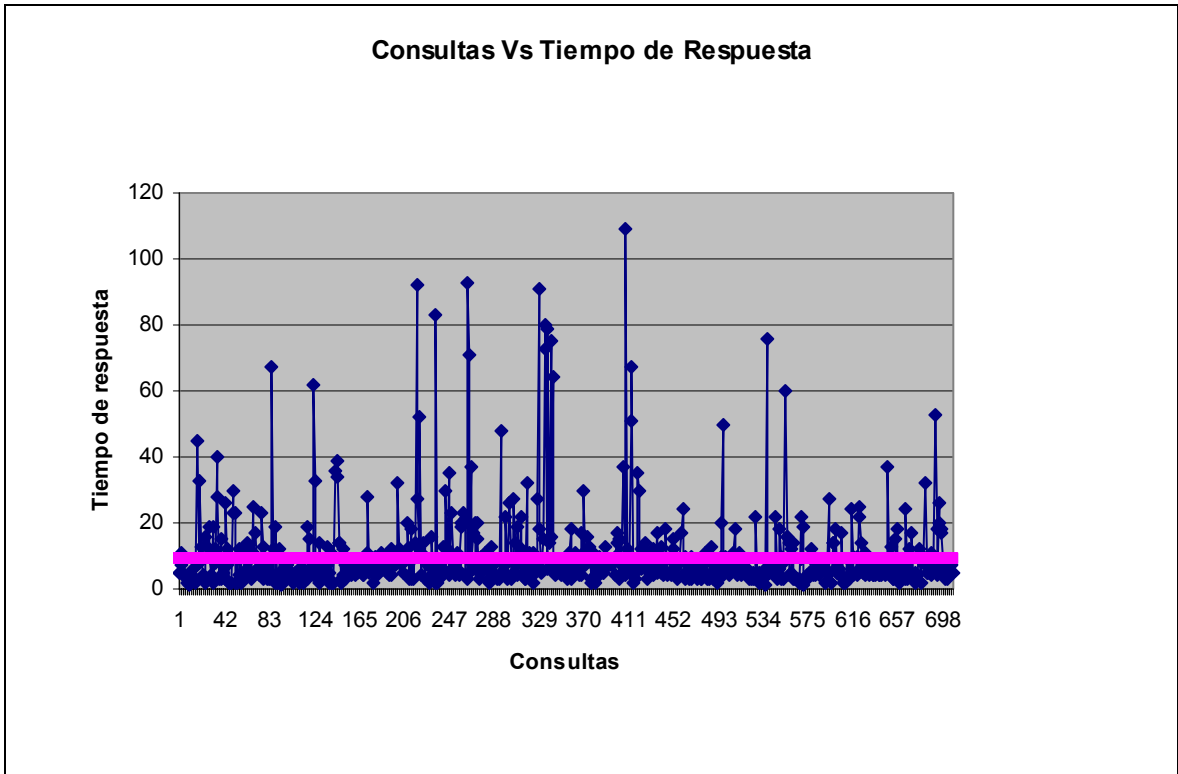


Figura 53. Consultas vs. Tiempos de respuesta.

- Diagrama de torta: representa los porcentajes de las consultas defectuosas y no defectuosas.

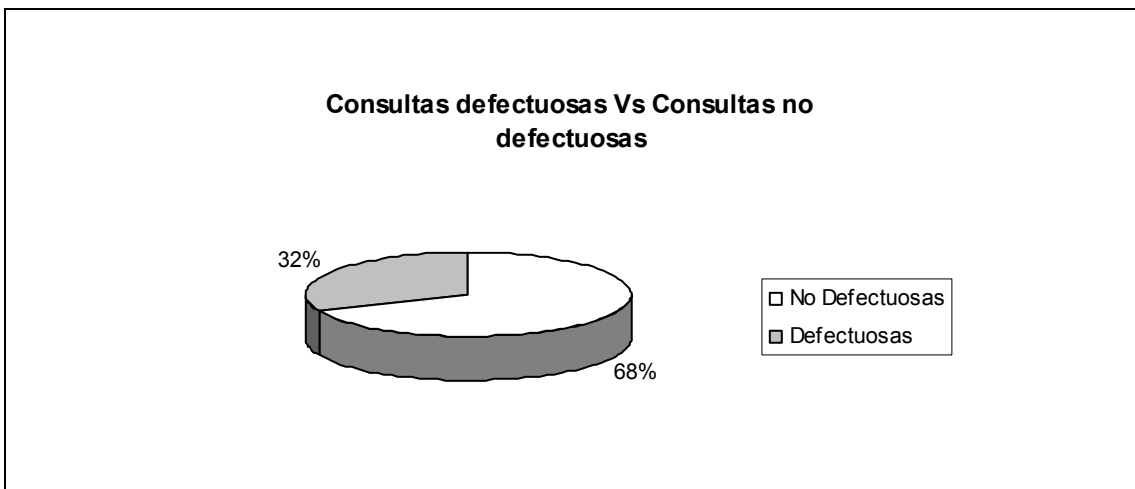


Figura 54. Consultas defectuosas Vs No defectuosas.

- Gráfico de barras: representa la cantidad de consultas que realizan durante cada hora.

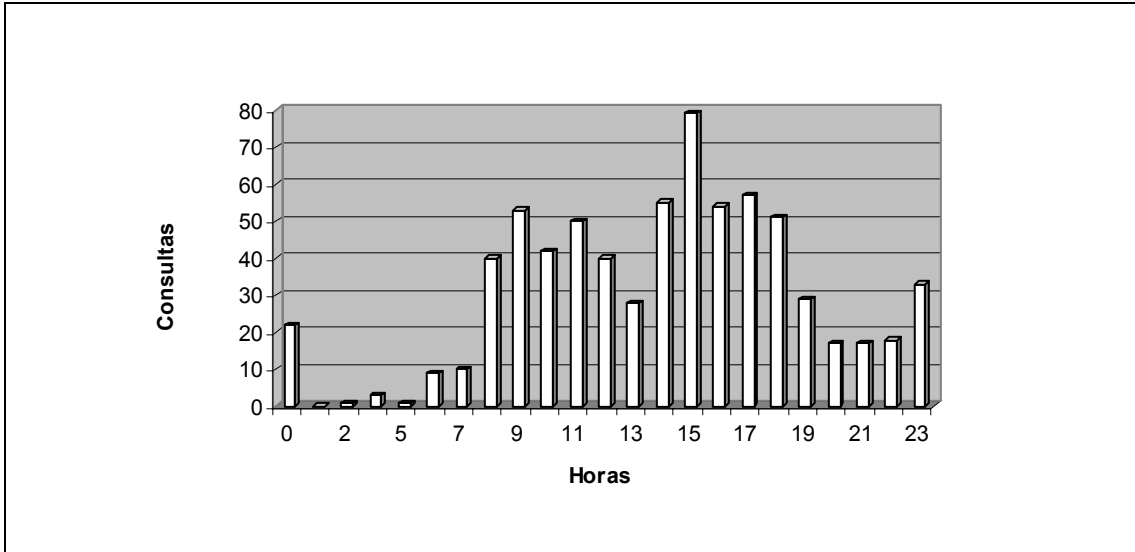


Figura 55. Número de consultas Vs hora.

- Gráfica de puntos: representa las consultas realizadas en su totalidad y las consultas defectuosas en cada hora.

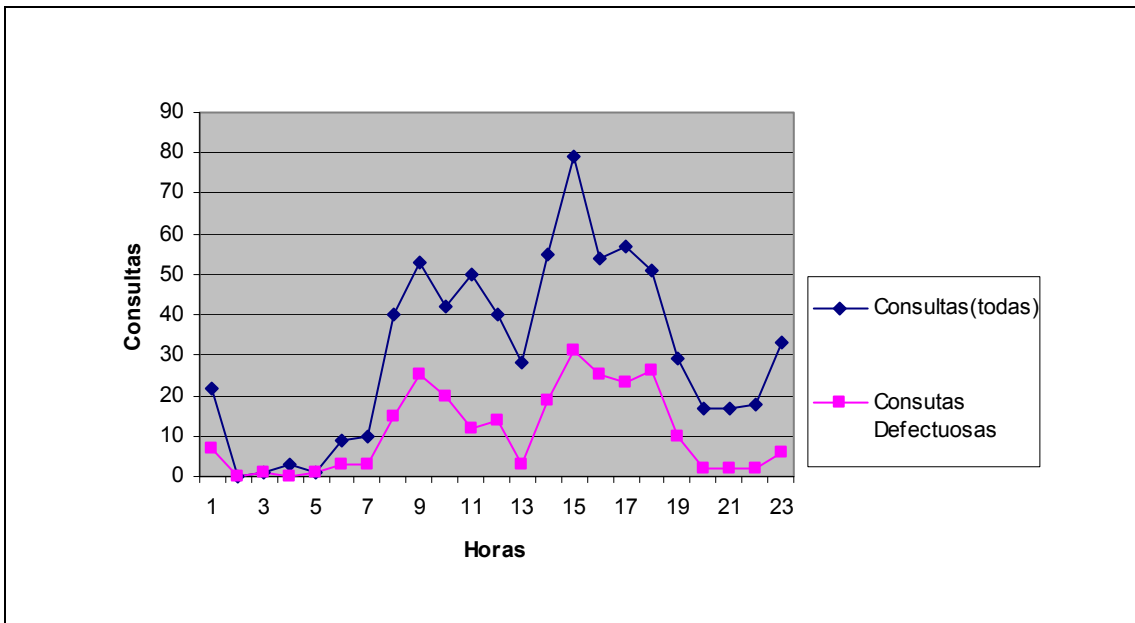


Figura 56. Consultas por hora vs. Consultas defectuosas por hora

- Diagrama de torta: representa la cantidad de usuarios que después de obtener el listado entran a ver los detalles de un documento.

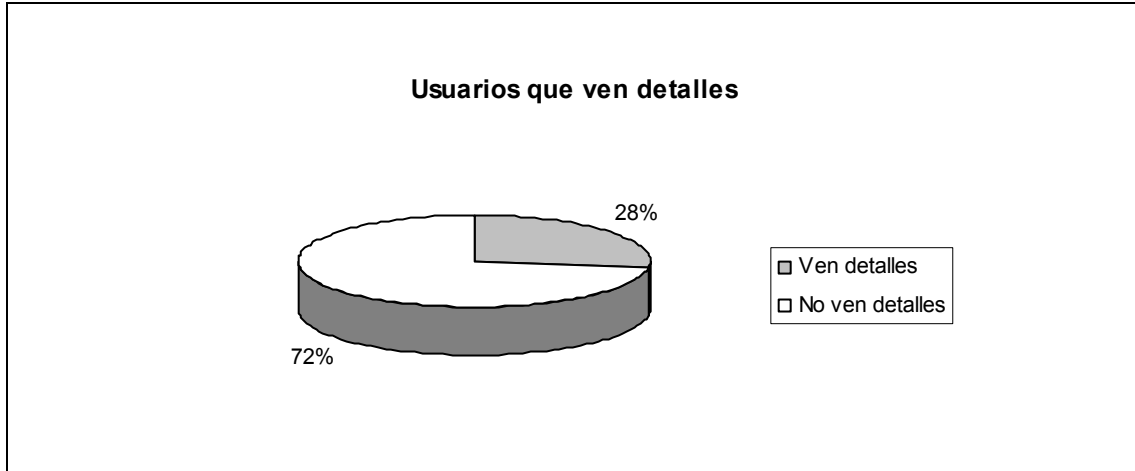


Figura 57. Usuarios que ven detalles vs no ven.

- Diagrama de torta: representa la cantidad de usuarios que consultan los detalles y los que no, cuando el tiempo de respuesta está por encima de la media.

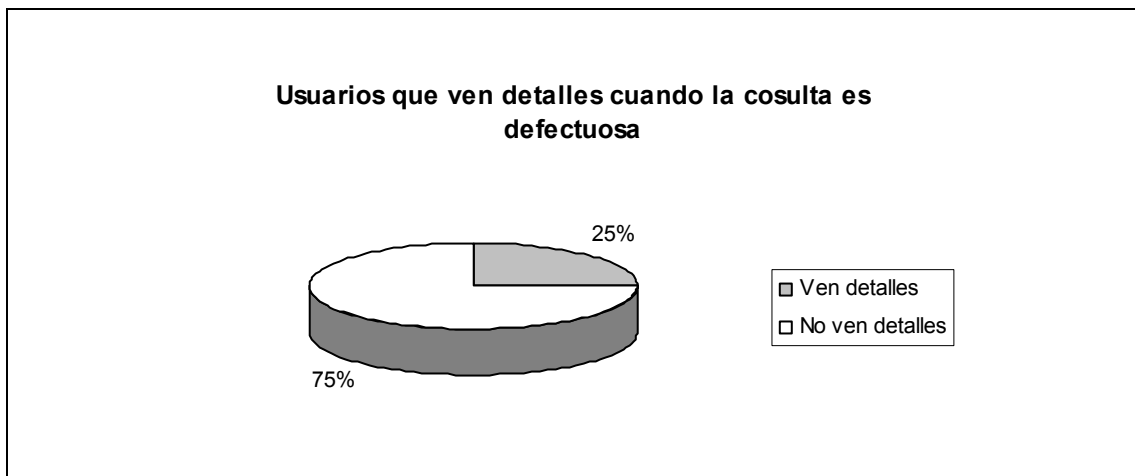


Figura 58. Usuario que ven detalles Vs no ven detalles.

10.2.4 Calcular el nivel de sigma.

Conceptos

- Unidad: Es una consulta. Inicia cuando el usuario oprime el botón de buscar hasta que el sistema envía el resultado de la consulta.
- Tiempo de Respuesta: Tiempo que gasta el servidor desde que recibe la petición del usuario hasta que envía la respuesta.
- Unidad defectuosa: Es una consulta que su tiempo de respuesta es mayor que la media de todas las unidades.
- Oportunidad: Son las consultas que se realizan en las horas críticas del servidor de la base de datos y que el tiempo de respuesta esta por encima de la media.

Número de unidades (NU) = 832

Número de unidades defectuosas (NUD) = 245

Número de oportunidades (NO) = 26

Cálculo

$$\text{DPU} = \frac{\text{NUD}}{\text{NU}} = \frac{245}{832} = 0.294471$$

$$\text{DPO} = \frac{\text{DPU}}{\text{NO}} = \frac{0.294471154}{26} = 0.011326$$

$$\text{DPMO} = \text{DPO} * 1000000 = 11325.81361$$

$$\text{Sigma} \quad 11325.81361 = 3,77$$

10.2.5 Benchmarking.

No Aplica.

10.3 CTQ 2

10.3.1 Determinar qué medir.

CTQ	Igualar el número de servicios que presta el catálogo bibliográfico Web (2) con el servicio de búsqueda de material bibliográfico informix (5).	Medida		Número de Servicios	
		Actual	Meta		
Escenario	Los servicios utilizados por los usuarios para consultar el material bibliográfico de la biblioteca.	2	5		
Comentario	Aunque se desean agregar mas servicios al sistema, la meta principal es igualar al sistema informix.	Wlibruis	Bench		
		NU			
		NUD			
		NO			
Plan de colección de datos					
Método	Tiempo	Fuente de los Datos	Recursos	Costo	Responsables
Comparación	1 semana	Catálogos disponibles	Internet	\$ 0	Diana Zambrano Sergio Contreras
Descripción de Metodología	Tener un conocimiento de los servicios que presta el sistema de consulta de catálogo bibliográfico de informix junto con el actual administrador del sistema. Enumerarlos y describirlos.				

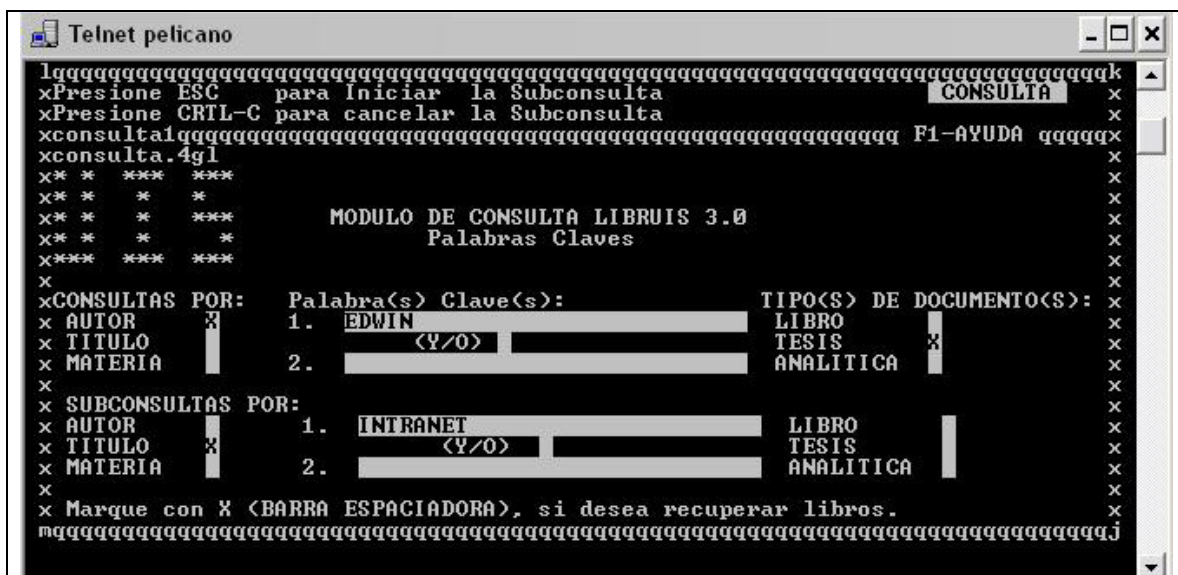


Figura 61. Subconsulta sistema informix.

Descripción:

- Los criterios anteriormente escritos, son utilizados nuevamente y no son modificables.
- Se debe seleccionar por lo menos un tipo de búsqueda (autor, titulo, materia), los tres pueden ser seleccionados.
- Se debe seleccionar por lo menos un tipo de documento a buscar (libro, tesis, revista o analítica) se pueden seleccionar los cuatro.
- Se debe escribir al menos una palabra clave a buscar.
- Para escribir dos palabras a buscar se debe escribir el conector lógico (y/o).
- El resultado arroja las similitudes con los criterios de búsqueda.

c. Detalles

Es un servicio que se agrega a la consulta, muestra los detalles del material seleccionado

10.3.4 Calcular el nivel de sigma.

Conceptos

- Unidad: Es un servicio que presta el sistema.
- Servicio: Una funcionalidad del sistema que interactúa con el usuario final para la búsqueda de un material bibliográfico.
- Unidad defectuosa: Es un servicio que no presta el sistema de búsqueda de catálogo bibliográfico Web, que si presta el sistema de búsqueda de catálogo bibliográfico en informix.
- Oportunidad: Es un servicio que es imposible realizar por Web.

10.3.5 Benchmarking.

La universidad complutense de Madrid, posee uno de los sistemas de búsqueda de catálogo bibliográfico vía Web más sorprendentes entre tantas universidades visitadas.

Presta diferentes servicios como

- Búsqueda simple.



The image shows a search interface titled "Buscar". It features a search bar with a dropdown menu set to "Palabra" and a search button labeled "Buscar". Below the search bar, there is a dropdown menu set to "Catálogo completo". Below the search bar, there is a list of search criteria:

- ◆ Autor
- ◆ Autor/Título
- ◆ Palabra
- ◆ Título
- ◆ Publicaciones Periódicas
- ◆ Título de la colección
- ◆ Materia
- ◆ Clasificación
- ◆ ISBN/ISSN

Figura 65. Búsqueda simple Benchmarking.

- Búsqueda Avanzada

Buscar en:

Autor Y

Materia Operadores

Cualquier campo

Idioma:

Tipo de Material:

Ubicación:

Editor:

Año: y Anterior

Ordenar según:

Figura 66. Búsqueda avanzada Benchmarking.

- Listado de resultado

(Extender) (Comenzar) (Buscar) (Modificar) (+ Catálogos) (Historial de búsqueda)

PALABRA CLAVE Ver Catálogo Completo

Ordenados por fecha

Núm	Marcar	PALABRAS CLAVE (1-5 de 5)
1	<input type="checkbox"/>	Control estadístico de la calidad / Douglas C. Montgomery ; traducción Rodolfo García Piña
2	<input type="checkbox"/>	Seis Sigma / Fermín Gómez Fraile, José Francisco Vilar Barrio, Miguel Tejero Monzón
3	<input type="checkbox"/>	Las claves de Seis Sigma : la implantación con éxito de una cultura que revoluciona el mundo empres
4	<input type="checkbox"/>	Seis Sigma : una iniciativa de calidad total / Enric Barba, Francesc Boix, Lluís Cuatrecasas ; prólo
5	<input type="checkbox"/>	El poder de Seis Sigma / Subir Chowdhury ; traducción de Gloria Méndez Sejjido

Figura 67. Listado de resultados Benchmarking.

- Detalles de Búsqueda

Autor	Chowdhury, Subir			
Título	El poder de Seis Sigma / Subir Chowdhury ; traducción de Gloria Méndez Seijido			
Publicac.	Madrid : Prentice Hall, D.L.2001			
Des.física	XI, 151 p. ; 21 cm			
ISBN	8420533610			
Clasific.	658.012.4			
Materia	Empresas -- Gestión			

UBICACION	SIGNATURA	ESTADO	TIPO DE PRESTAMO	NOTA
Económ. y Empr.-Sala	S658.012.4CHO	DISPONIBLE	PR. NORMAL	
Estadística-L. Acceso	L658CHO	DISPONIBLE	PR. NORMAL	

Figura 68. Detalles material bibliográfico Benchmarking.

- Ver detalles tipo MARC

```

008 020131s2001 sp 001 0 spa cnam i
017 D.L.M.33933-2001
020 8420533610
040 SpMaUCCE
080 658.012.4
100 1 Chowdhury, Subir
245 13 El poder de Seis Sigma /|cSubir Chowdhury ; traducción de
Gloria Méndez Seijido
260 Madrid :|bPrentice Hall, |cD.L.2001
300 XI, 151 p. ;|c21 cm
650 04 Empresas|xGestión
907 00 rs|b0
964 70 an|b0|c20020201

```

Figura 69. Detalles MARC Benchmarking.

- Reserva de material

Introduzca la siguiente información:

Su Nombre:

Código de Barras

Introduzca su PIN:

Cancelar si no se completa en: Día Mes

Figura 70. Reserva de material bibliográfico Benchmarking.

- Guardar referencias

Formato del Listado:	Enviar Listado a:
<input checked="" type="radio"/> Pantalla Completa <input type="radio"/> Presentación Abreviada <input type="radio"/> Pro-Cite <input type="radio"/> End-Note <input type="radio"/> MARC	<input checked="" type="radio"/> E-Mail Enviar a: <input type="text"/> Asunto: <input type="text" value="Búsqueda bibliográfica en Cisne, Catálogo c"/> <input type="radio"/> Pantalla <input type="radio"/> Disco Local
<input type="button" value="Enviar"/>	

Su Lista de Registros Guardados

Núm	Marcar	Exportar (1-2 de 2)
1	<input type="checkbox"/>	Las claves de Seis Sigma : la implantación con éxito de una Peter S. Pande, Robert P. Neuman, Roland R. Cavanagh ; tradu
2	<input type="checkbox"/>	El poder de Seis Sigma Subir Chowdhury ; traducción de Gloria Méndez Seijido

Figura 71. Guardar referencias Benchmarking.

- Búsqueda a:
 - Catálogo general: contiene todos los fondos y materiales adquiridos.
 - Subcatálogo de Fondo Antiguo: contiene una parte importante de los fondos del siglo IX a 1801 de la UCM.
 - Subcatálogo de la Biblioteca digital Dioscórides: permite consultar más de 3.000 libros digitalizados.
 - Subcatálogo de Tesis de la UCM: permite consultar más de 7.000 tesis digitalizadas.
 - Subcatálogo de recursos electrónicos: permite el acceso a bases de datos, monografías, tesis y revistas en texto completo.
 - Subcatálogo de Biblioteca Europea: documentación y legislación sobre la UE.
 - Catálogos especiales (nuevas adquisiciones, materiales especiales: DVD, videos).
 - Bibliografías recomendadas: asignatura / profesor.
 - Consultas conjuntas a catálogos de otras bibliotecas.
 - Opciones personalizadas: ver su registro de usuario (préstamos, cómo crear un PIN para acceder a recursos electrónicos desde casa) Información: horarios, fondos, ayuda, buzón de sugerencias, libros para adquirir.

Conclusión:

El catálogo de la universidad complutense cuenta con siete servicios. Los siguientes servicios se pueden tener en cuenta para el proyecto.

- Ver detalles MARC.
- Guardar referencias.
- Búsqueda avanzada.

El servicio de reserva de materiales no es tenido en cuenta por decisión de la alta dirección.

10.4 CTQ 3

10.4.1 Determinar que medir.

CTQ	Medida	Número de Intentos			
Disminuir en un 25% la cantidad de intentos que realiza un usuario para obtener un resultado en la consulta.	Actual	Meta			
Los resultados que arroja el sistema dependen del material que existe y está catalogado en el sistema de informix. El sistema W-Libraris permite únicamente 4 formas posibles al usuario al momento de hacer una consulta.	2,0791	1.5593			
Comentario	Wlibruis	Bench			
	NU	709			
	NUD	368			
	NO	5			
	Nivel sigma	2,7			
Plan de colección de datos					
Método	Tiempo	Fuente de los Datos	Recursos	Costo	Responsables
Encuesta electrónica.	1 mes	Datos que arroja el servidor.	1 servidor	\$ 0	Diana Zambrano Sergio Contreras
Descripción de Metodología	Se guarda en un campo de la tabla los datos o la cadena que un usuario digita al momento de realizar la consulta, y según la forma de realizar la consulta, el sistema arroja resultados o arroja un mensaje de error.				

10.4.2 Plan de colección de datos.

Para medir el número de intentos que un usuario hace al momento de realizar una consulta hasta obtener un resultado como respuesta, se crea una tabla en el servidor de la aplicación y se capturan los parámetros que el usuario introduce en el campo al momento de hacer la consulta, se guarda en otra variable el primer resultado que se genera en la lista de resultado y si la consulta no arroja un resultado, se guarda en otra variable el error que fue generado.

Las tablas diseñadas para medir el número de intentos hasta obtener un resultado son las siguientes:

The image displays two database table schemas. The first table, 'estadísticas_biblioteca', has a primary key 'id_estadística' and the following fields: ip_consulta, por_autor, por_titulo, por_materia, en_libro, en_tesis, en_analitica, tiempo_inicial, tiempo_final, fecha_consulta, ingreso_detalle, titulo_consulta, documento_consulta, autor_consulta, resultados, palabras, and materia_consulta. The second table, 'errores', has a primary key 'id_error' and the following fields: cadena_consulta, descripcion_error, fecha, and hora.

estadísticas_biblioteca	
id_estadística	
ip_consulta	
por_autor	
por_titulo	
por_materia	
en_libro	
en_tesis	
en_analitica	
tiempo_inicial	
tiempo_final	
fecha_consulta	
ingreso_detalle	
titulo_consulta	
documento_consulta	
autor_consulta	
resultados	
palabras	
materia_consulta	

errores	
id_error	
cadena_consulta	
descripcion_error	
fecha	
hora	

Figura 72. Tablas de captura de datos estadísticos.

10.4.3 ELABORAR LAS GRÁFICAS

- a. *Diagrama de corrido*: Esta gráfica representa la variación que existe del número de intentos al momento de realizar las consultas respecto de la media. La media de intentos que hace un usuario hasta obtener un resultado es 2,0791. Ver figura 73.

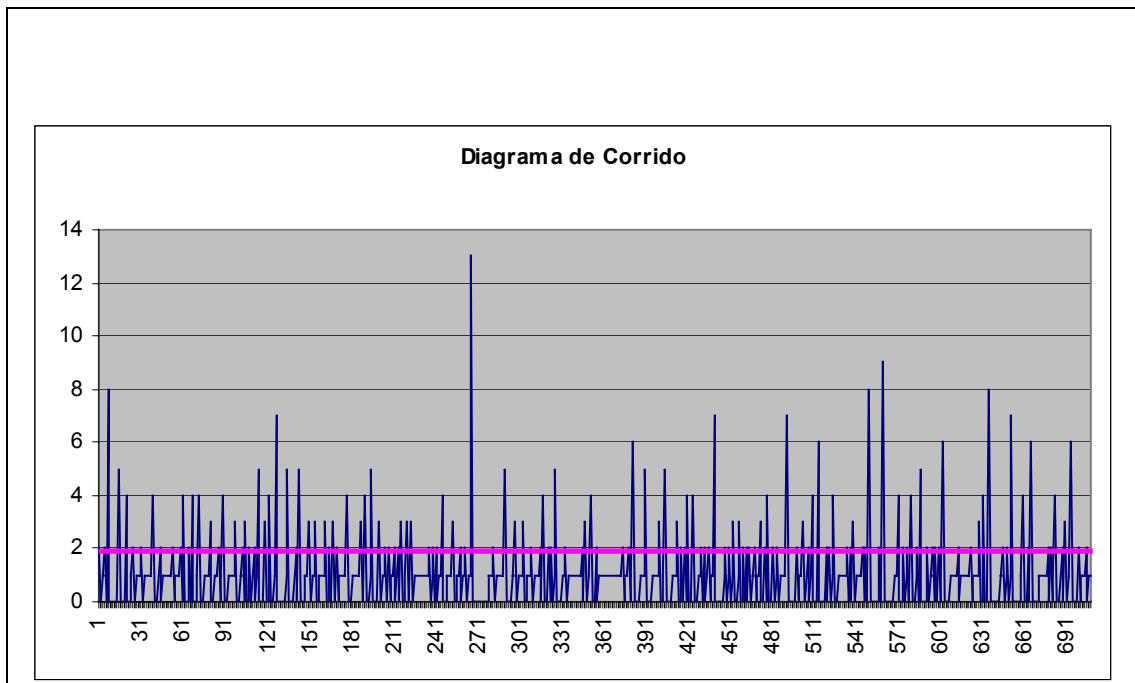


Figura 73. Diagrama de Corrido

- b. *Gráfica de los resultados de la consulta*: Del total de las consultas que se hicieron, se observa que el 48% de los datos arroja algún resultado de la consulta y el 52% son errores que el usuario comete al momento de realizar la consulta.

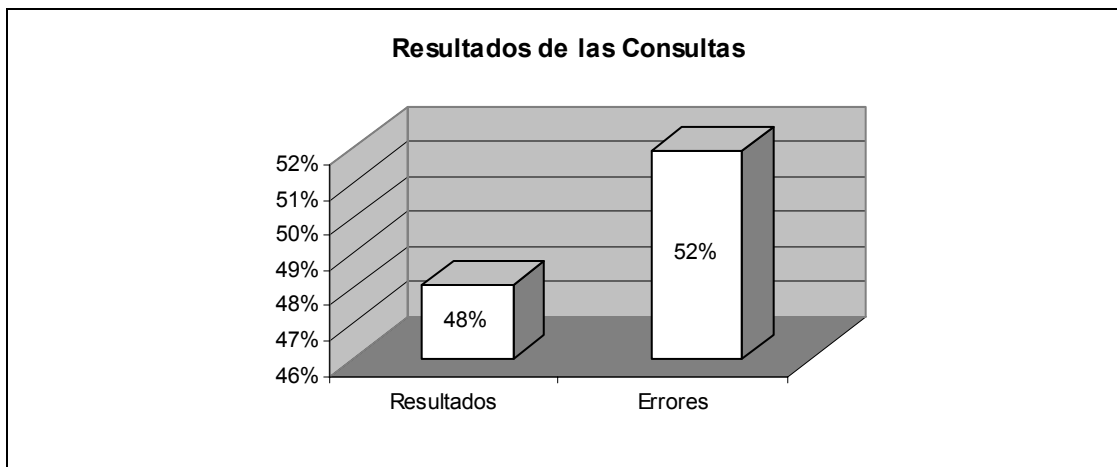


Figura 74. Resultados de las Consultas.

c. *Gráfica de los tipos de errores:* Esta gráfica muestra los tipos de errores más frecuentes que se presentan al momento de realizar las consultas. Los errores están clasificados de la siguiente forma:

- i. Error 1: Ha introducido caracteres inválidos en la consulta.
- ii. Error 2: No ha seleccionado el criterio de búsqueda.
- iii. Error 3: No ha seleccionado el tipo de documento.
- iv. Error 4: No ha introducido las palabras a buscar.

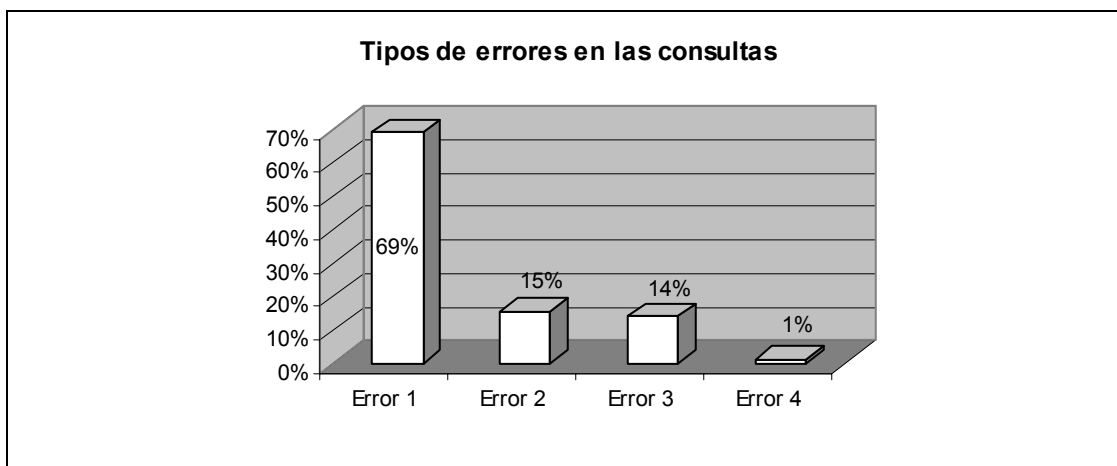


Figura 75. Gráficas de los tipos de errores.

10.4.4 Calcular el nivel de sigma.

Conceptos

- Unidad: es una consulta. Inicia cuando el usuario oprime el botón consultar hasta que el sistema arroja el resultado, si la consulta fue exitosa o hasta que el sistema arroja el mensaje de error resultado de una consulta mal realizada.
- Número de intentos: es el número de intentos que realiza un usuario en una consulta hasta que el sistema arroja algún resultado. El sistema puede arrojar o no resultados dependiendo de si existe el material en la biblioteca o si la consulta fue hecha con la sintaxis permitida.
- Unidad defectuosa: Cuando el sistema arroja algún mensaje de error tipo 1, 2 ó 3.
- Oportunidad: Las consultas que se realizan y que arrojan errores tipo 4.

Número de unidades (NU) = 709

Número de unidades defectuosas (NUD) = 368

Número de oportunidades (NO) = 5

Cálculo

$$\text{DPU} = \frac{\text{NUD}}{\text{NU}} = \frac{368}{709} = 0.5190$$

$$\text{DPO} = \frac{\text{DPU}}{\text{NO}} = \frac{0.5190}{5} = 0.1038$$

$$\text{DPMO} = \text{DPO} * 1000000 = 103800$$

$$\text{Sigma} \quad 103800 = 2,78$$

10.4.5 Benchmarking.

Existen muchos ejemplos de catálogos bibliográficos que se pueden tener en cuenta para comparar. Uno de los más visitados y más conocido es el buscador Google⁵⁹. Las ventajas que presta este buscador, comparado con la aplicación actual de consulta del material bibliográfico vía Web de la UIS son:

- Para realizar una consulta simple en google, se digitan las palabras descriptivas y se presiona la tecla ENTER o se hace clic en el botón de búsqueda en Google para ver la lista de resultados relevantes.
- Google sólo muestra aquellas páginas que incluyen *todos* los términos de la búsqueda. No es necesario incluir "and" entre sus términos.
- Para proporcionar resultados más exactos, no usa "búsquedas parciales" ni realiza búsquedas con "comodines". En otras palabras, busca exactamente los términos que ingresa en la casilla de búsqueda. Buscar "sal" o "sal*" no devolverá búsquedas que contengan "salero" o "salamandra".
- Las búsquedas **no** distinguen entre mayúsculas y minúsculas. Todas las letras, independientemente de como estén escritas, se consideran minúsculas.
- Las búsquedas Google en español no distinguen los acentos diacríticos, diéresis ni la letra ñe.
- Si no encuentra similitud de la palabra que ha utilizado el usuario, la aplicación aproxima la consulta según los registros que contenga.

Las características que se tuvieron en cuenta a la hora de realizar el Benchmarking fueron la medida del número de intentos que realiza un usuario al momento de realizar la consulta para que el sistema arroje algún

⁵⁹ GOOGLE: <http://www.google.com>

resultado. Se concluye, que este buscador, sea cual tipo de consulta que el usuario realice, arroja resultados al primer intento de búsqueda.

Al igual que en el análisis de Benchmarking de la CTQ 2, el catálogo bibliográfico de la Universidad Complutense de Madrid, es un gran acercamiento a las metas que se quieren alcanzar en este proyecto. Así como google, el catálogo de la biblioteca complutense, permite en su búsqueda simple, realizar las consultas por más de una palabra sin utilizar operadores.

La búsqueda por palabras clave es el tipo de consulta con más formas de realizarla. Si la búsqueda se realiza en cualquier campo, el sistema busca en todos los campos del registro, pero se puede limitar la consulta por Autores, Títulos, Materias, y Notas, y utilizar los operadores booleanos: AND, OR, AND NOT, NEAR y WITHIN #

Las palabras clave, además, pueden truncarse mediante los siguientes caracteres:

* Trunca hasta cinco caracteres a partir de la posición de la palabra en la que se encuentre.

** Trunca un número ilimitado de caracteres a partir de la posición de la palabra en la que se encuentre.

? Recupera la palabra especificada con cualquier carácter en la posición en la que se encuentre (anders?n recupera Anderson y Andersen).

10.5 DESARROLLAR EL MODELO FINANCIERO

No aplica para el proyecto, debido a su envergadura.

10.6 IDENTIFICAR Y PRIORIZAR LOS REQUERIMIENTOS DE LOS CLIENTES

Antes de iniciar la identificación y priorización de los requerimientos de los clientes, las CTQ's del proyecto ya deben estar claramente definidas. Según la definición del problema ***“Actualmente, los usuarios de la biblioteca realizan las consultas del material bibliográfico por medio del catálogo bibliográfico, situado en la página Web e Intranet de la UIS. El problema es que los usuarios se encuentran insatisfechos por el servicio que presta y se quejan constantemente”***, éstas CTQ's deben atacar directamente el problema con el fin de eliminarlo en su mayor parte.

Una vez establecidas las CTQ's se comienza a establecer la captura de Requisitos y la forma de satisfacerlos, centrado siempre en el cumplimiento de esas metas.

Cuando se comprende el contexto que enmarca el desarrollo del sistema, los clientes⁶⁰ a través de diversas fuentes de información, crean una primera lista de las características que debería tener el sistema, y son clasificadas basadas en el modelo de Kano. La tabla 25 muestra esta lista a continuación.

⁶⁰ Clientes: Biblioteca y los representantes de los usuarios. Los mecanismos que se utilizaron para la toma de requisitos candidatos fue la Voz del Cliente.

Tabla 25. Principales características que debe tener el sistema.

Número	Descripción	Tipo⁶¹
1	El sistema debe permitir realizar búsquedas del material bibliográfico que se encuentra incorporado en el sistema principal Libruis.	1
2	El resultado de la búsqueda debe permitir ver un listado con los datos más generales de cada referencia encontrada.	1
3	Se debe permitir seleccionar uno de estos resultados y ver los detalles generales de éste.	1
4	La referencia seleccionada se podrá ver en el formato MARC.	3
5	Se podrá realizar una subconsulta con los resultados arrojados en la consulta anterior.	2
6	Se podrá seleccionar las referencias que el usuario tenga a consideración, ya sea para imprimir un listado o enviar a una cuenta de correo electrónico.	3
7	El resultado que arroje debe ser en tiempos cortos.	1
8	La apariencia gráfica debe ser agradable al usuario.	2
9	El modo de navegación debe ser intuitivo sin importar el usuario.	2
10	El sistema debe ofrecer una ayuda que permita encontrar lo que el usuario está buscando.	2
11	El lenguaje de desarrollo debe adaptarse a los servidores que alojarán la aplicación.	1
12	Permitir consultar revistas.	1

Los requisitos pueden clasificarse de dos formas: requisitos funcionales y no funcionales. Los requisitos funcionales son aquellos que hacen todo lo que se supone debe hacer el sistema, es decir, las funcionalidades que el sistema pueda ofrecer a los diferentes tipos de usuarios. Los requisitos no funcionales son aquellos que no se encuentran afectando de una manera directa el objetivo principal del sistema.

⁶¹ Tipo: Clasificación establecida según el modelo de Kano.

El siguiente paso es identificar cuáles de los requisitos candidatos mencionados anteriormente se pueden representar por medio de casos de uso, es decir, establecer los requisitos funcionales.

Tabla 26. Listado de Requisitos Funcionales.

Nombre	Descripción
Realizar Búsqueda	Recuperar información acerca del material bibliográfico, utilizando una interfaz web que le permita realizar una búsqueda simple, con solo palabras, o una búsqueda avanzada, agregando operadores lógicos y otros criterios.
Ver Detalles	Ver las características más representativas de un libro, ya sea en un formato que contenga solamente la descripción general, o en uno más avanzado denominado MARC.
Subconsultas	Realizar una nueva búsqueda, teniendo en cuenta el resultado de una búsqueda antes realizada.
Guardar Referencia	Permitir seleccionar libros y guardarlos con la información general.
Imprimir Listado	Preparar una página con la información detallada de cada material de tal forma que se puede imprimir.
Enviar Listado	Generar un listado para enviarlos a una cuenta de correo electrónico.
Navegar por los resultados	Permitir navegar por los autores o materias, mostrados en los detalles de un material bibliográfico.

Los otros requisitos de la tabla 26, que no fueron analizados en la tabla 25 se consideran no funcionales.

CASOS DE USO

Los actores son los usuarios finales que interactuarán con el sistema de Búsqueda. Hacen parte de este grupo los estudiantes, profesores y los profesionales que pertenezcan a la comunidad universitaria.

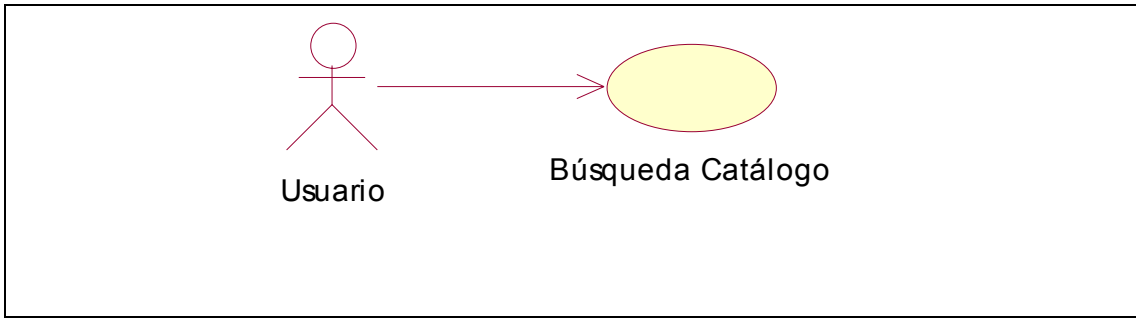


Figura 76. Caso de uso general.

Los casos de uso que se plantearon para este actor representa los requisitos funcionales por medio del diagrama de casos de uso presentado a continuación (Figura 77).

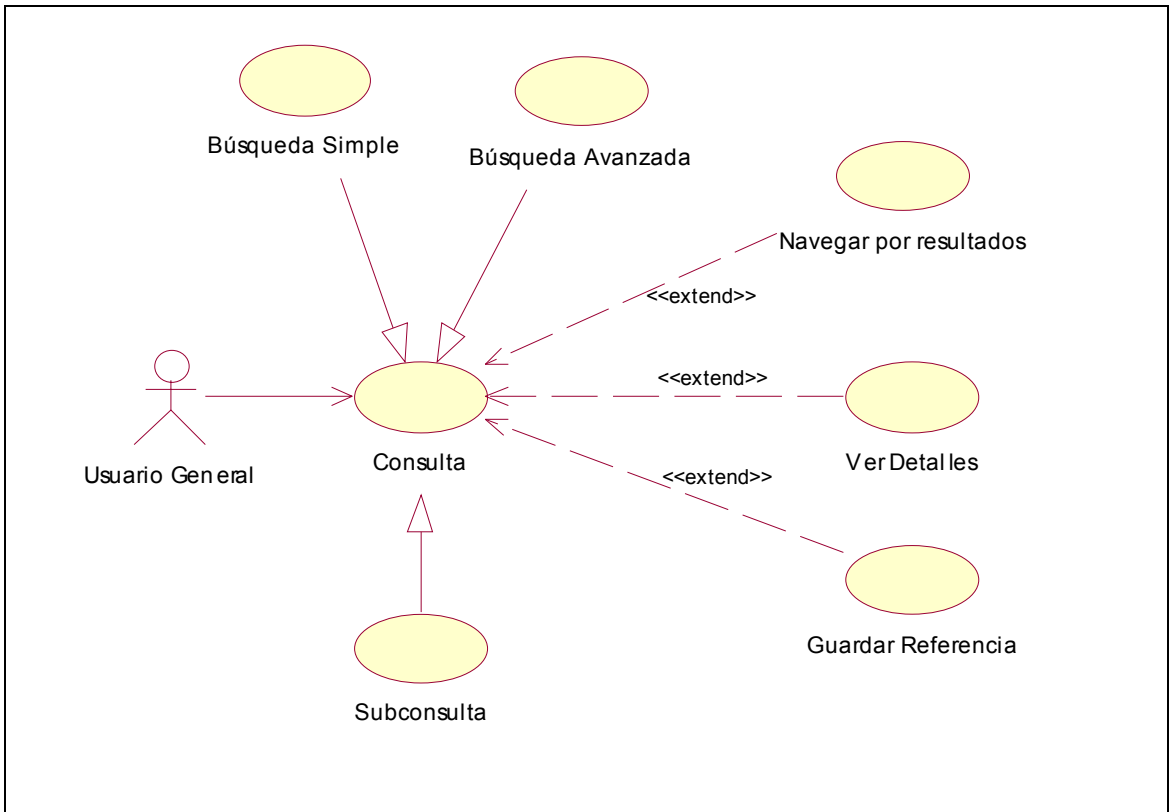


Figura 77. Diagrama de Casos de Uso

Descripción de los casos de uso (CU):

Tabla 27. CU consultar.

CU 1: consultar	
Nombre	Caso de uso de consultar.
Descripción	Este caso de uso describe como el usuario final utiliza el catálogo bibliográfico a través de la interfaz Web para consultar el material bibliográfico que tiene la Biblioteca de la Universidad.
Precondiciones	<ul style="list-style-type: none"> • La estación de trabajo debe estar conectada a la red. • El servidor de la aplicación este funcionando. • El servidor de la base de datos se encuentre funcionando. • El botón de buscar debe estar activo.
	<ol style="list-style-type: none"> 1. El sistema muestra las opciones de búsqueda. {avanzada o simple} 2. El sistema muestra cuadros de texto para las palabras, las opciones de tipos de documentos, y las opciones de tipos de material. {ingresa palabras; (Libro, Tesis, Revista); (Titulo, Autor, Materia)} 3. El sistema muestra las similitudes encontradas entre los criterios de búsqueda y los datos que posee.
	<ul style="list-style-type: none"> ▪ El sistema muestra los datos que tienen similitud con los criterios de búsqueda digitadas por el usuario y las opciones de subconsulta, ver detalles o guardar la referencia. ▪ El sistema muestra una página de resultados igual a cero y el usuario cierra la sesión. ▪ El sistema muestra una página de resultados igual a cero y el usuario realiza una búsqueda totalmente nueva.

Diagrama de actividades

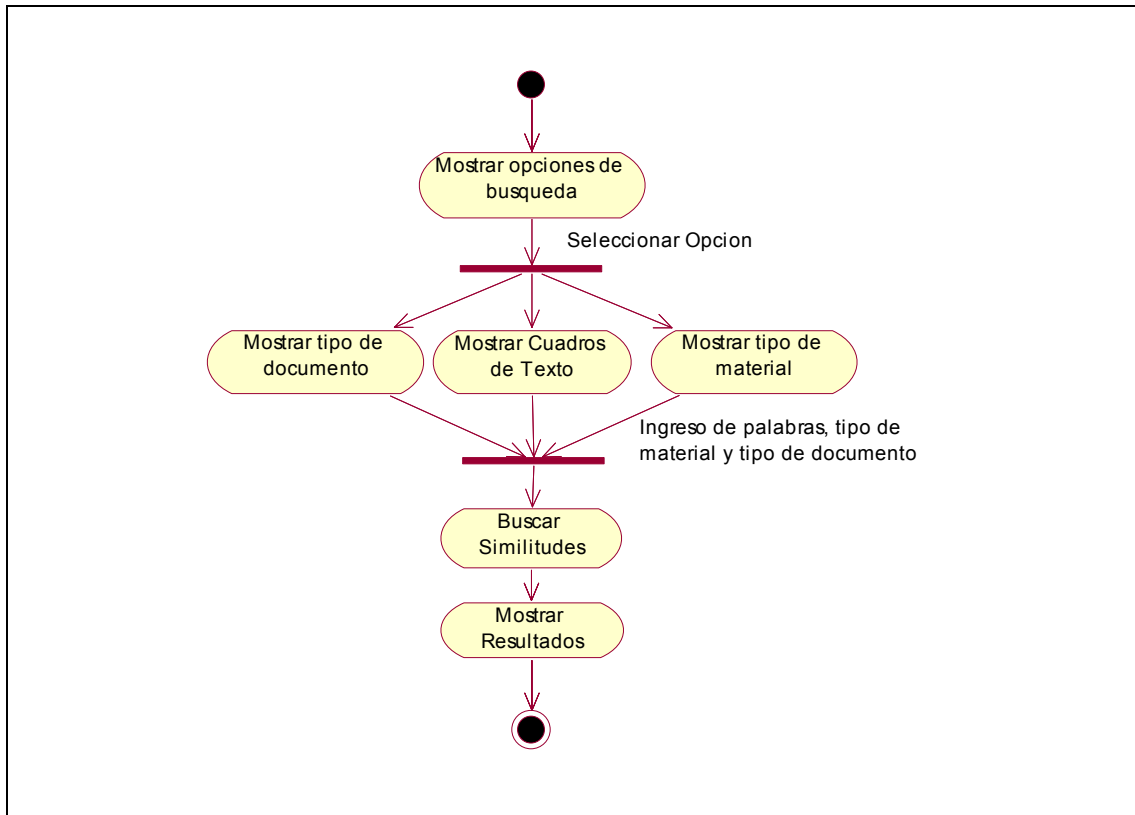


Figura 78. Diagrama de actividades CU consultar.

Tabla 28. CU subconsulta.

CU 2: subconsulta	
Nombre	Subconsulta
Descripción	Este caso de uso describe como el usuario final hace una búsqueda más filtrada para obtener resultados más precisos.
Precondiciones	<ul style="list-style-type: none"> ▪ La consulta debe haber mostrado al menos dos resultados. ▪ La sesión de la aplicación continúa abierta ⁶². ▪ Los criterios de búsqueda utilizados en la consulta anterior deben estar presentes.
Flujo Básico	<ol style="list-style-type: none"> 1. El sistema obtiene los criterios de búsqueda utilizados en la consulta anterior. 2. El sistema muestra los cuadros de texto, donde el usuario digitará los nuevos criterios que va a adicionar. 3. El sistema busca las similitudes entre los criterios de búsqueda y los datos que posee el sistema.

⁶² Sesión Abierta: Que la ventana del navegador no se haya cerrado.

	4. El sistema muestra los resultados de la búsqueda.
Poscondiciones	<ul style="list-style-type: none"> ▪ El sistema muestra resultados más precisos con los criterios de búsqueda del cliente, y la opción de ver detalles de cada uno de los resultados, guardar las referencias de cada uno de los resultados. ▪ El sistema muestra una página de resultados igual a cero y el usuario cierra la sesión. ▪ El sistema muestra una página de resultados igual a cero y el usuario realiza una búsqueda totalmente nueva.

Diagrama de actividades

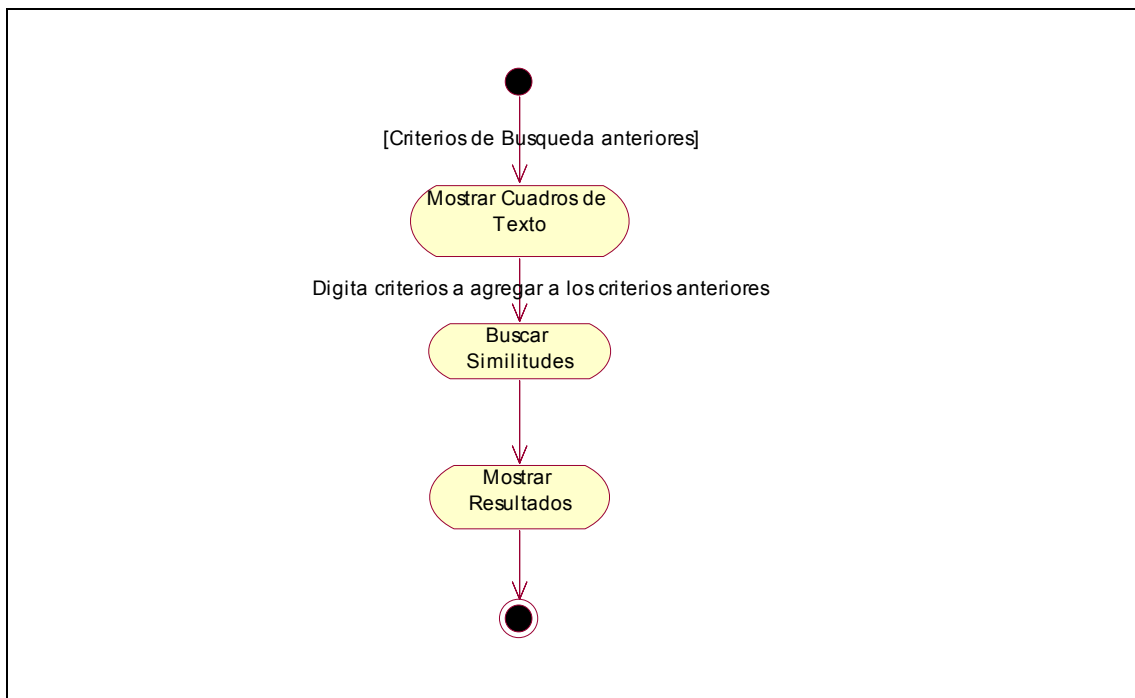


Figura 79. Diagrama de actividades CU subconsulta.

Tabla 29. CU búsqueda simple.

CU 3: búsqueda simple	
Nombre	Búsqueda simple
Descripción	Este caso de uso describe como un usuario utiliza el catálogo bibliográfico a través del Web de la Universidad para consultar el material bibliográfico que existe en la biblioteca.
Precondiciones	<ul style="list-style-type: none"> ▪ La estación de trabajo debe estar conectada a la red. ▪ El servidor de la aplicación debe estar funcionando. ▪ El servidor de la base de datos se encuentra funcionando. ▪ El botón de buscar debe estar activo.
	<ol style="list-style-type: none"> 1. El sistema muestra el cuadro de texto donde el usuario digitará las palabras por las que desea realizar la búsqueda. 2. El sistema comprueba que el número de palabras digitadas sea menor o igual a cuatro. 3. El sistema busca las similitudes de las palabras con los datos del sistema 4. El sistema muestra los resultados obtenidos de la búsqueda.
	<ul style="list-style-type: none"> ▪ El sistema muestra los datos que tienen similitud con los criterios de búsqueda digitadas por el usuario y las opciones de subconsulta, ver detalles o guardar la referencia. ▪ El sistema muestra una página de resultados igual a cero y el usuario cierra la sesión. ▪ El sistema muestra una página de resultados igual a cero y el usuario realiza una búsqueda totalmente nueva.

Diagrama de Actividades

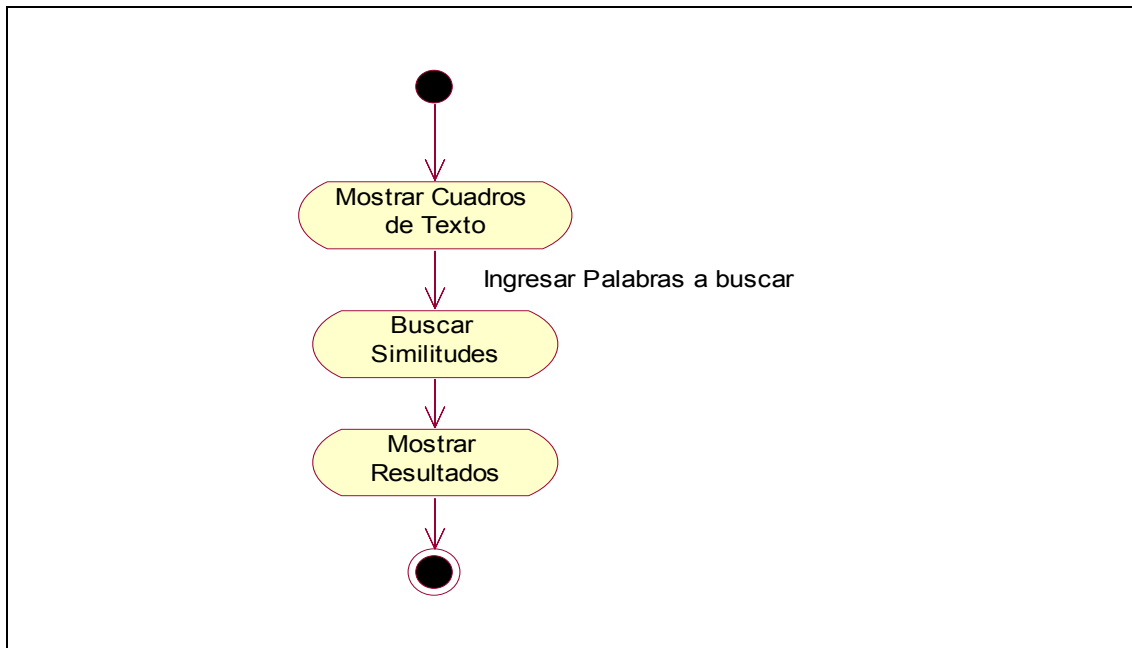


Figura 80. Diagrama de actividades CU búsqueda simple.

Tabla 30. CU búsqueda avanzada.

CU 4: búsqueda avanzada	
Nombre	Búsqueda avanzada
Descripción	Este caso de uso describe como un usuario final realiza una búsqueda avanzada al sistema de Catálogo Bibliográfico a través de la interfaz Web de la Universidad.
Precondiciones	<ul style="list-style-type: none"> ▪ La estación de trabajo debe estar conectada a la red. ▪ El servidor de la aplicación debe estar funcionando. ▪ El servidor de la base de datos se encuentra funcionando. ▪ La opción de buscar debe estar activa. ▪ Sólo se pueden ingresar cuatro palabras a buscar. ▪ Sólo se puede ingresar una palabra por cada cuadro de texto.
	<ol style="list-style-type: none"> 1. El sistema muestra cuatro cuadros de texto. {El usuario digita las palabras por las que desea hacer la búsqueda.} 2. El sistema muestra los tipos de opciones que puede ser una palabra. {El usuario selecciona el tipo de la(s) palabra(s) de ese cuadro de texto, autor, título, materia, libro, tesis, analítica, revista.} 3. El sistema verifica la cantidad de cuadros de texto llenos {uno solo} 4. El sistema verifica si el cuadro lleno tiene seleccionado su tipo. {Si} 5. El sistema busca las similitudes de la palabra y su tipo en los datos que posee el sistema. 6. El sistema muestra los resultados de la búsqueda.
Flujo Alternativo	<ol style="list-style-type: none"> 1. en 3, el sistema verifica la cantidad de cuadros de texto llenos {varias} 2. en 4, el sistema verifica si el cuadro de texto lleno tiene seleccionado su tipo. {No}
Poscondiciones	<ul style="list-style-type: none"> ▪ El sistema muestra resultados más precisos con los criterios de búsqueda del cliente, y la opción de ver detalles de cada uno de los resultados y guardar las referencias de cada uno de los resultados. ▪ El sistema muestra una página de resultados igual a cero y el usuario cierra la sesión. ▪ El sistema muestra una página de resultados igual a cero y el usuario realiza una búsqueda totalmente nueva.

Diagrama de Actividades

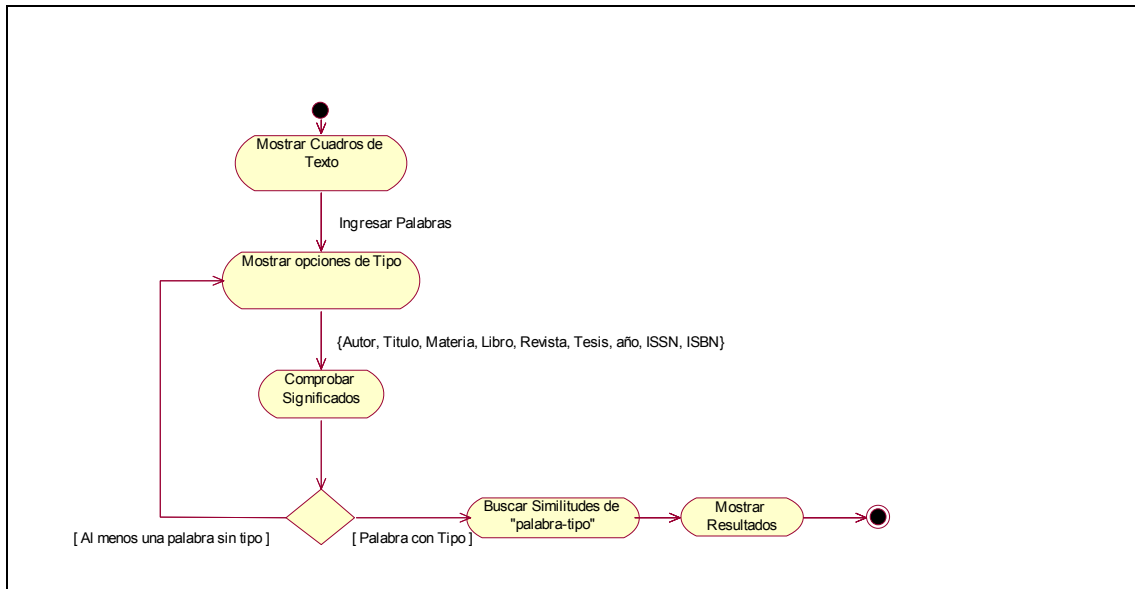


Figura 81. Diagrama de actividades CU búsqueda avanzada.

Tabla 31. CU guardar referencia.

CU 5: guardar referencia	
Nombre	Guardar los registros marcados.
Descripción	Este caso de uso describe como el usuario utiliza el sistema para seleccionar las referencias de preferencia para luego enviarlas a una cuenta de correo electrónico o imprimir un listado.
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haber realizado una consulta, y el sistema le ha arrojado los resultados organizados por páginas de hasta n registros. • El usuario ha obtenido al menos un resultado como respuesta del sistema. • La impresora debe estar funcionando. • La impresora debe tener papel para usar. • El servicio de envió en el servidor de correo debe estar activo.
	<ol style="list-style-type: none"> 1. El sistema muestra al usuario los resultados que le ha arrojado en respuesta a la consulta y escoge aquellos en los que él se encuentra interesado en recibir mayor información. 2. El usuario hace la selección de aquellos registros que son de su preferencia. 3. El sistema guarda los registros marcados. 4. El usuario expide el listado para ver los registros que ha guardado previamente. 5. El usuario solicita al sistema que la información adicional de los ítems que ha seleccionado sea enviada a una cuenta de

	<p>correo electrónico.</p> <ol style="list-style-type: none"> 6. El sistema basándose en la petición enviada por el usuario genera el cuerpo del mensaje del correo electrónico con la información adicional de los ítems seleccionados, y le presenta al usuario un formato de captura de datos. 7. El usuario ingresa los datos que son necesarios para enviar el correo electrónico, como la dirección a la cual se desea enviar la información, el título y un texto adicional que desee agregar al cuerpo del mensaje. 8. El sistema se conecta con el servidor de correo y envía la información al correo correspondiente y le muestra al usuario una notificación de que la operación fue llevada a cabo con éxito. 9. El caso de uso finaliza con éxito.
Flujo Alternativo	<ol style="list-style-type: none"> 1. En 2, el usuario puede solicitar al sistema ver los detalles de cualquier registro y desde ahí puede guardarlo. 2. En 3, el usuario puede seguir navegando por el sitio para agregar más registros al listado. 3. En 4, el usuario puede hacer modificaciones del listado si desea eliminar alguno de los registros guardados. 4. En 5, el usuario también puede solicitar al sistema imprimir el listado. Entonces el sistema muestra al usuario una página con la información adicional sobre los ítems solicitados por el usuario, la cual puede ser impresa por el usuario. 5. En 8, el sistema puede detectar un error en los datos en el momento de conectarse al servidor de correo por lo cual el sistema notifica al usuario el error y el caso de uso termina.
	<ul style="list-style-type: none"> • El sistema envía el correo electrónico al usuario y le notifica de este suceso, • El sistema envía el correo electrónico al usuario y presenta un error enviando el correo electrónico. • El sistema envía el listado en formato de impresión y la impresora imprime satisfactoriamente. • El sistema envía el listado en formato de impresión y se presenta un error en la impresión.

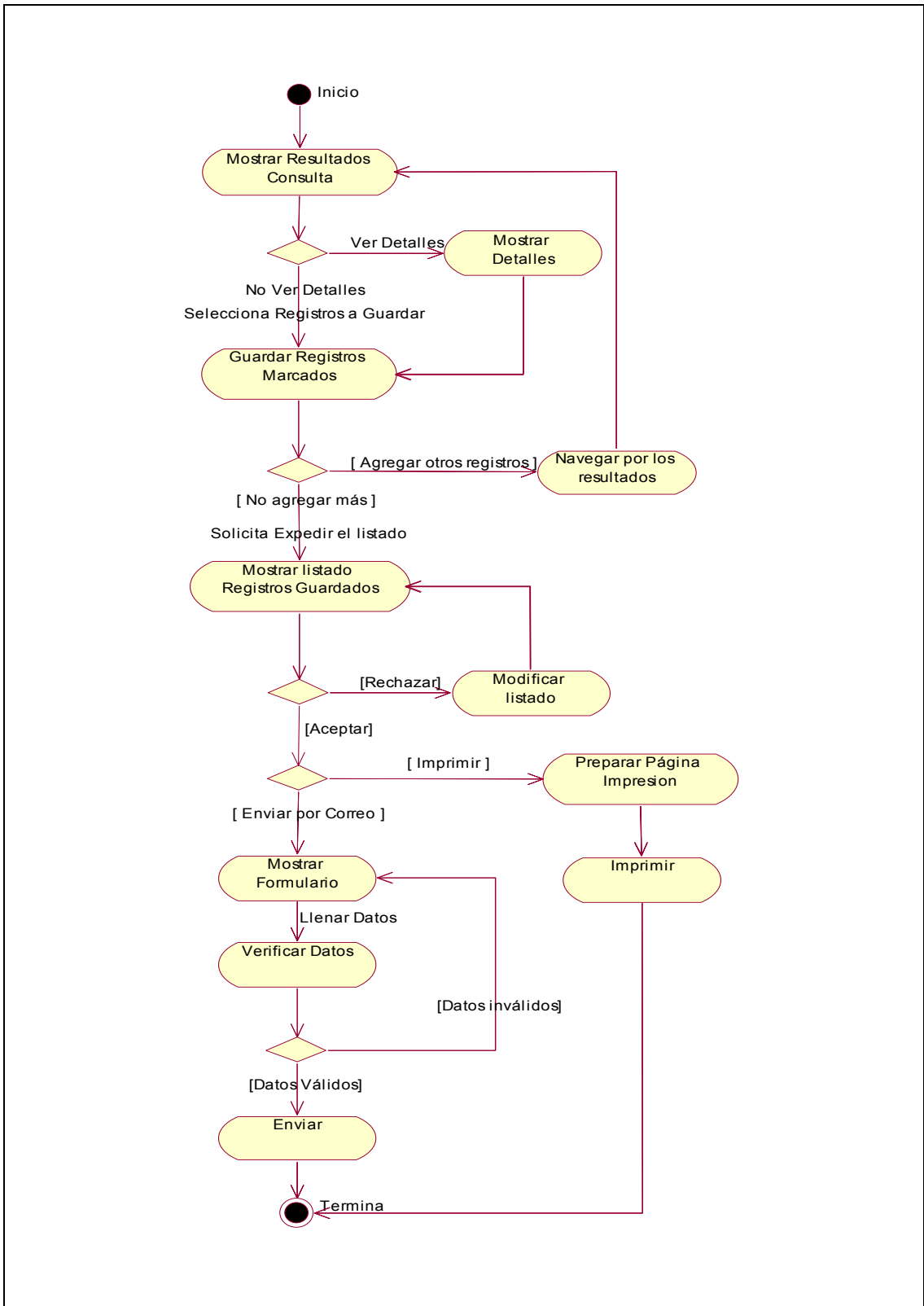


Figura 82. Diagrama de actividades CU guardar referencia.

Tabla 32. CU ver detalles

CU 6: ver detalles	
Nombre	Ver detalles.
Descripción	Este caso de uso describe como el sistema permite al usuario seleccionar las referencias de preferencia para luego ver los detalles más generales en formato Normal o en formato MARC.
	<ul style="list-style-type: none"> • El usuario debe haber realizado una consulta, y el sistema le ha arrojado al menos un resultado.
	<ol style="list-style-type: none"> 1. El sistema muestra al usuario los resultados que le ha arrojado en respuesta a la consulta y escoge aquellos en los que él se encuentra interesado en recibir mayor información. 2. El usuario selecciona el registro para el cual quiere ver la información general en formato Normal y envía la petición al sistema. 3. El sistema recibe la petición del usuario y muestra toda la información acerca del registro escogido en el formato Normal. 4. El caso de uso finaliza.
Flujo Alternativo	<ol style="list-style-type: none"> 1. En 2, el usuario selecciona mostrar la información en el formato MARC. 2. En 3, el usuario puede ver también la información en el formato opuesto al seleccionado en 2.
	<ul style="list-style-type: none"> • El usuario puede guardar la referencia. • El usuario puede ir a ver detalles MARC si esta en detalles normal, o ver detalles normal si esta en detalles MARC. • El usuario puede navegar por los autores o materias del material. • El usuario puede volver al inicio del sistema. • El usuario cierra la sesión.

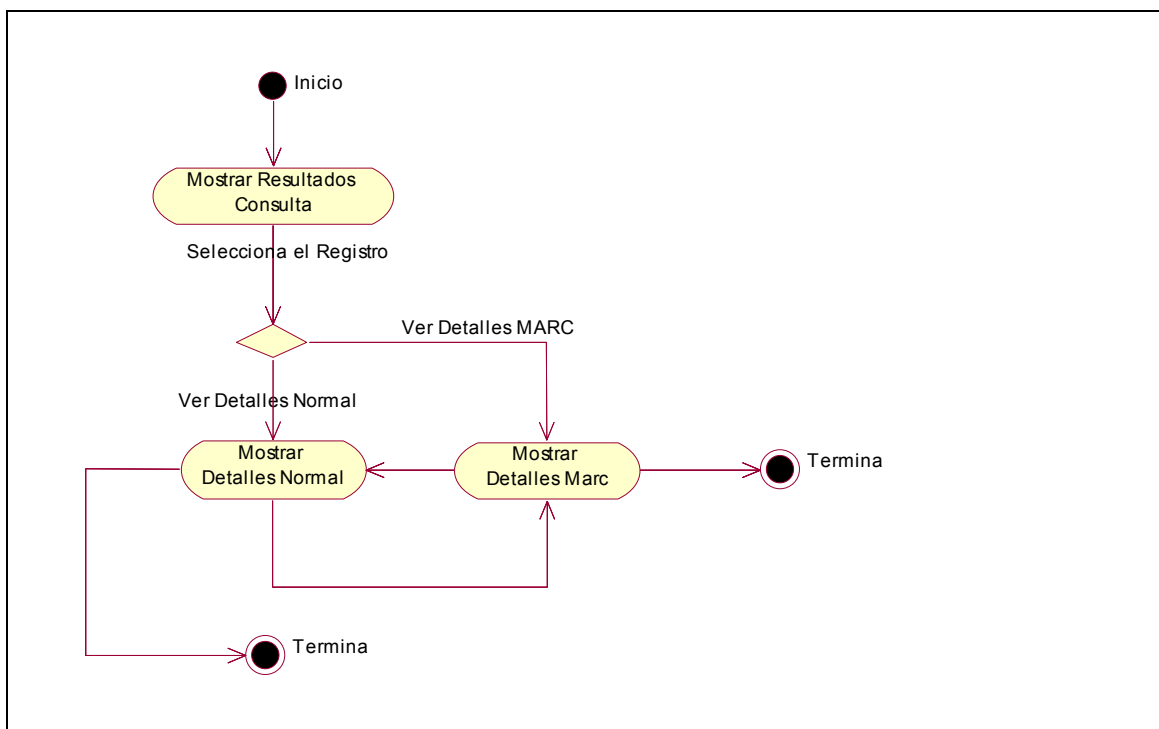


Figura 83. Diagrama de actividades CU ver detalles.

Tabla 33. CU navegar por los resultados.

CU 7: navegar por los resultados	
Nombre	Navegar por los resultados.
Descripción	Este caso de uso describe como el sistema permite al usuario navegar por los resultados obtenidos después de realizar cualquier consulta.
Precondiciones	<ul style="list-style-type: none"> • El sistema muestra al menos un autor. • El sistema muestra al menos una materia. • El sistema debe estar mostrando el detalle en formato normal.
	<ol style="list-style-type: none"> 1. El sistema muestra la información acerca del material bibliográfico escogido en el formato normal. 2. El usuario puede continuar navegando por una de las materias a las cuales pertenece el material. 3. El caso de uso finaliza.
	<ol style="list-style-type: none"> 1. En 2, el usuario puede continuar navegando por uno de los autores del material. 2. En 3, el usuario puede seguir navegando por los resultados de las consultas.
Poscondiciones	<ul style="list-style-type: none"> • El sistema muestra los resultados de la nueva consulta.

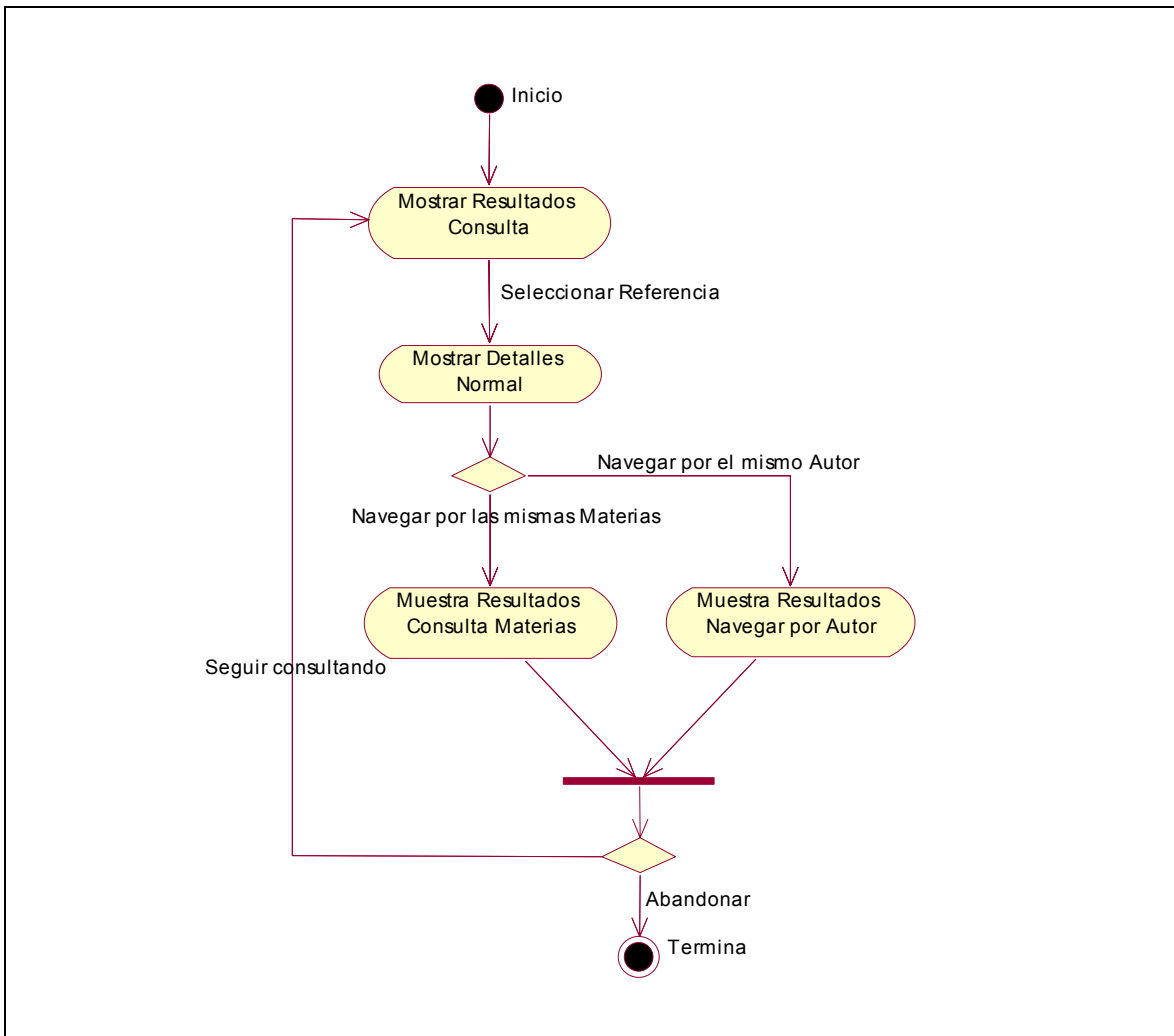


Figura 84. Diagrama de actividades CU navegar por los resultados.

10.7 INICIO DE QFD

La escala para determinar el grado de importancia es la siguiente:

Tipo	Valor	Símbolo
Alto	5	⊙
Medio	3	○
Bajo	1	△

Los porcentajes definitivos que se obtuvieron en el árbol de soluciones para las causas principales del problema son los siguientes:

Causa	Porcentaje
Tiempo de Respuesta.	30%
Servicios.	20%
Formas de realizar la consulta.	50%

La matriz de QFD es la siguiente:

Requerimientos	Importancia
Ofrecer más servicios	1
Material completo	1
Interfaz gráfica amigable	3
Intuitivo	5
Agilidad en la respuesta	3

10.8 PLAN DE COMUNICACIONES

Tabla 34. Plan de comunicaciones establecido.

GRUPO	TIPO DE INFORMACIÓN	NIVEL DE DETALLE	FRECUENCIA	METODO DE COMUNICACIÓN
Comité de Dirección	Estado del proyecto: Realizaciones, problemas o asuntos importantes que necesiten ser resueltos.	Alto nivel	Mensualmente.	Informes Escritos y reportes orales dados en reuniones.
Master Black Belt	Conceptos de la metodología Seis Sigma y el modelo	Nivel muy específico.	Quincenalmente.	Correo Electrónico.

	DMADDV.			
Equipo de Trabajo	Información detallada sobre el proyecto, cronograma, actividades, fechas límites, planes, publicaciones, riesgos y problemas.	Nivel muy específico.	Semanalmente.	Variedad: correo electrónico, reportes orales durante reuniones, informes escritos.
Usuarios	Actualizaciones generales del avance del proyecto.	General	Mensualmente	Reuniones.
Publico General	Opiniones generales sobre el proceso actual.	General	Diariamente	Encuestas sitio Web.

10.9 PRESENTAR AL COMITÉ PARA APROBAR

- Las graficas de la VOC.
- Cuadros de determinar las métricas.
- Modelo financiero (si aplica).
- Caso de uso principal, y según la envergadura del proyecto los casos de uso secundarios.
- Matriz de inicio de QFD.
- Plan de comunicaciones.

11. FASE ANALIZAR

11.1 DETERMINAR QUE CAUSA LA VARIACIÓN

Las medidas para cada una de las CTQ's, proporcionan una mejor visión sobre las causas del problema. La utilización de los gráficos de corrido y de barras para el tiempo de respuesta es de gran apoyo para encontrar la causa de la variación.

11.1.1 CTQ 1.

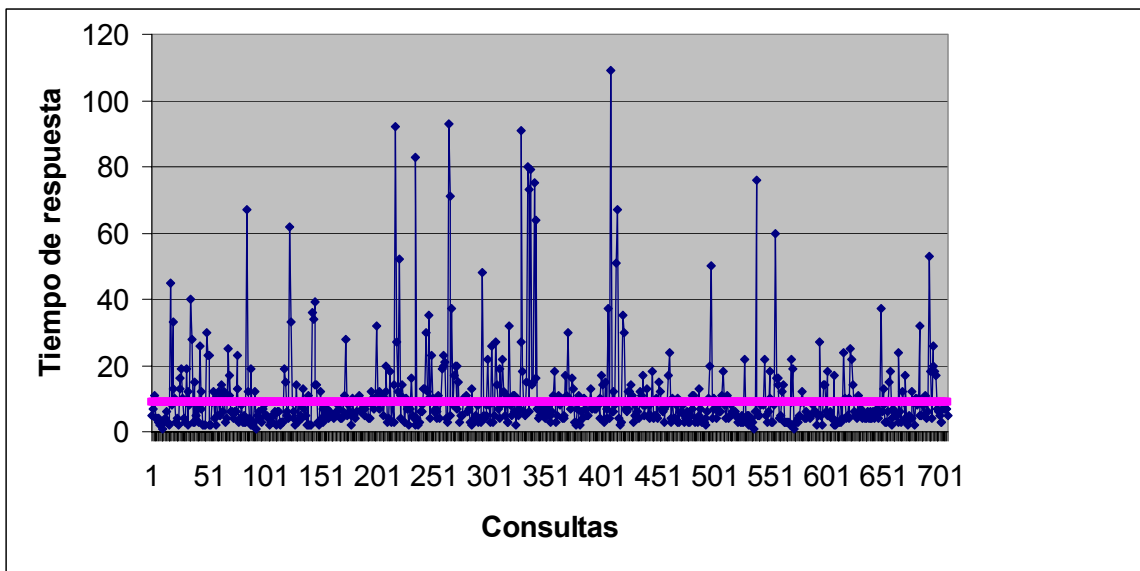


Figura 85. Gráfico de corrido CTQ 1.

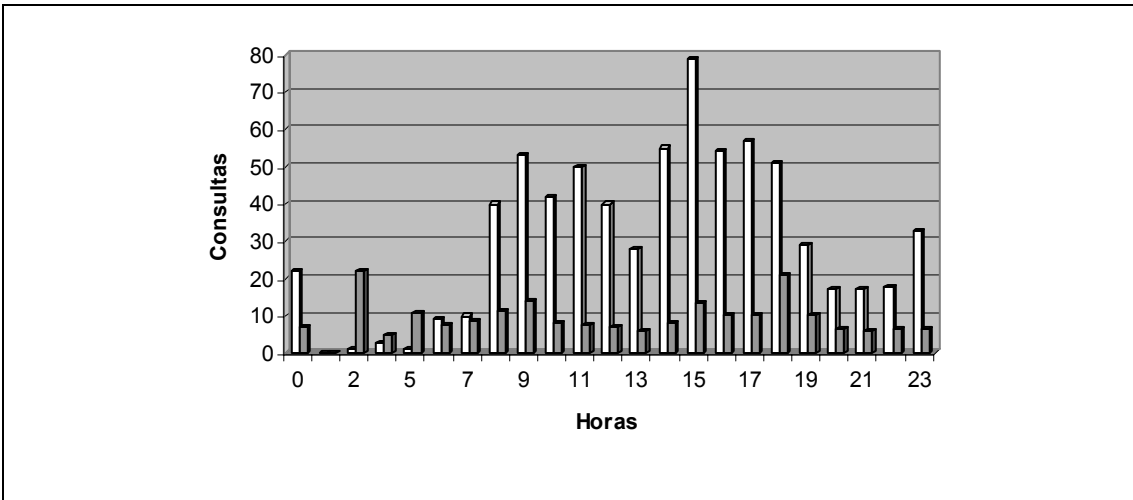


Figura 86. Tiempo de respuesta con cantidad de consultas por hora.

Las barras de color blanco muestran la cantidad de consultas realizadas por hora, y las barras de color azul muestran el promedio de tiempo de respuesta por hora.

Para determinar las causas se utilizan los diagramas de pareto.

Causas de la velocidad

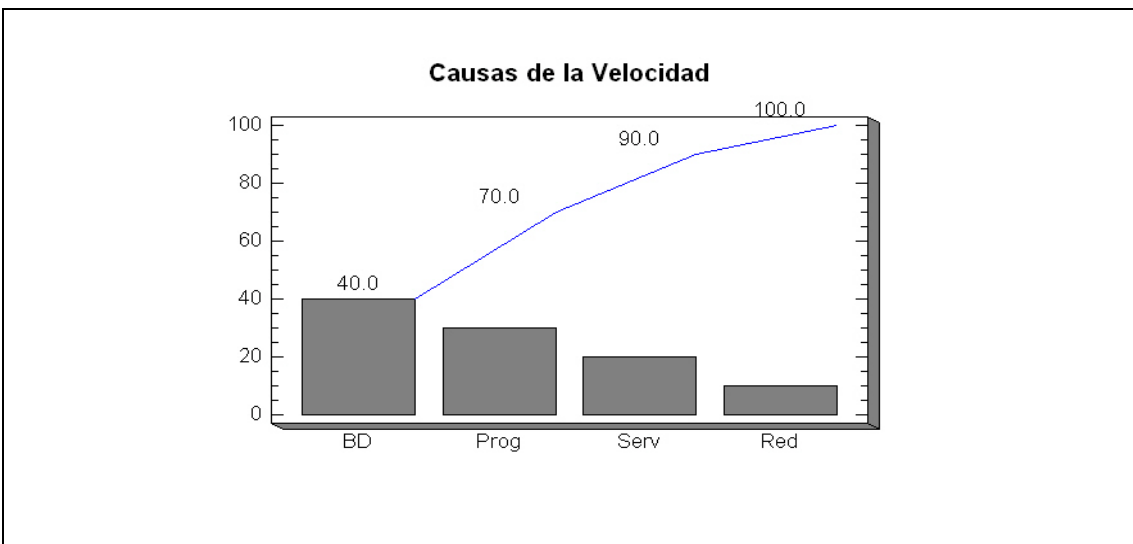


Figura 87. Pareto causas de la velocidad.

La causa del tiempo de respuesta alto se debe a varios factores que afectan directamente al sistema.

- Las consultas a la base de datos están alojadas en la dll, aunque es código compilado, no es óptimo para la base de datos.
- Existe mucha cantidad de volúmenes de datos, tanto que el modelo de la base de datos no es lo suficientemente rápido para satisfacer al cliente.
- La concurrencia de usuario en el servidor de aplicaciones y en el servidor de base de datos es alta.
- Los diferentes sistemas que se encuentran corriendo en el servidor de aplicaciones (Intranet, Cátedra, otros).
- Las diferentes bases de datos alojadas en el servidor de base de datos.

11.1.2 CTQ 2.

Por ser una variable discreta no se le aplica un análisis como el de las otras CTQ's.

11.1.3 CTQ 3

Dentro de los tipos de errores que se presentan al momento de realizar una consulta, se observan:

- Error 1: Ha introducido caracteres inválidos en la consulta.
- Error 2: No ha seleccionado el criterio de búsqueda.
- Error 3: No ha seleccionado el tipo de documento.
- Error 4: No ha introducido las palabras a buscar.

Como se muestra en el diagrama de Pareto, Figura 88, el error más frecuente fue el error tipo 1, que corresponde al 69,02% del total.

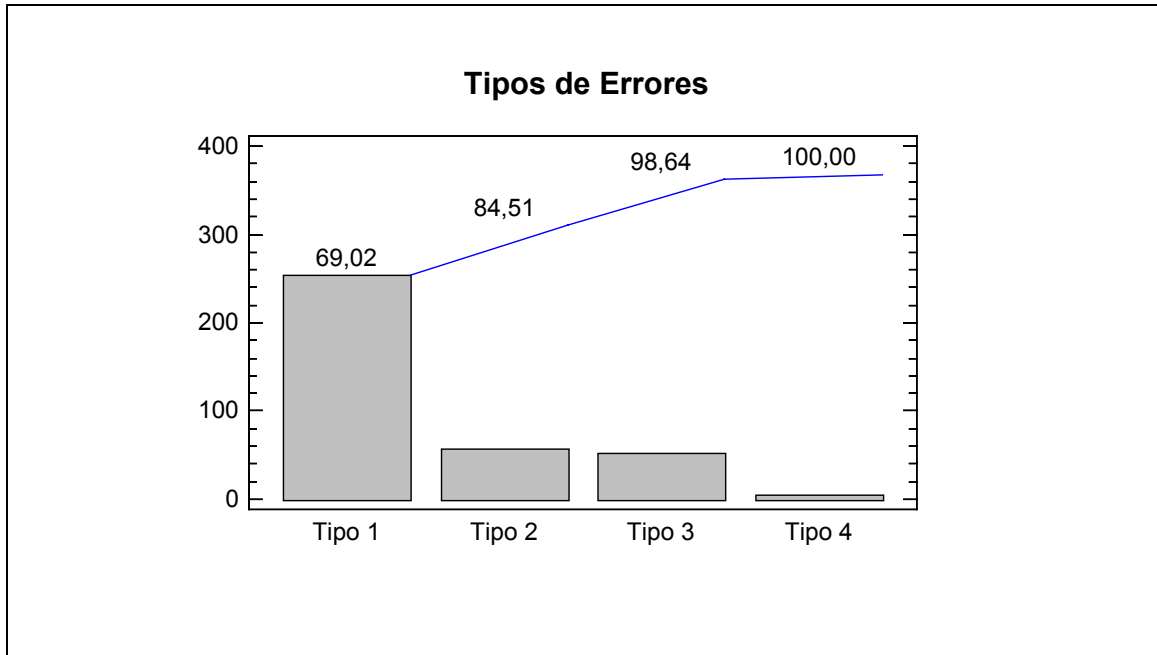


Figura 88. Pareto tipos de errores.

Para poder analizar el por qué se presenta el error tipo 1, se analizaron los datos que el usuario digita al momento de hacer las consultas. Los errores se presentaron por tres causas principalmente: Los espacios entre palabras, utilización equivocada de los operadores “+, %, Y, O”, y otros como tildes, palabras con la letra ñe y guiones.

La causa más frecuente del error tipo 1, es la escritura de palabras dejando espacios entre ellas. Ver figura 89.

Si se quiere reducir la variación del número de intentos que debe hacer un usuario al momento de realizar una consulta, se debe prestar mayor atención en evitar que los errores de tipo 1 se presenten, sobre todo por los espacios entre palabras.

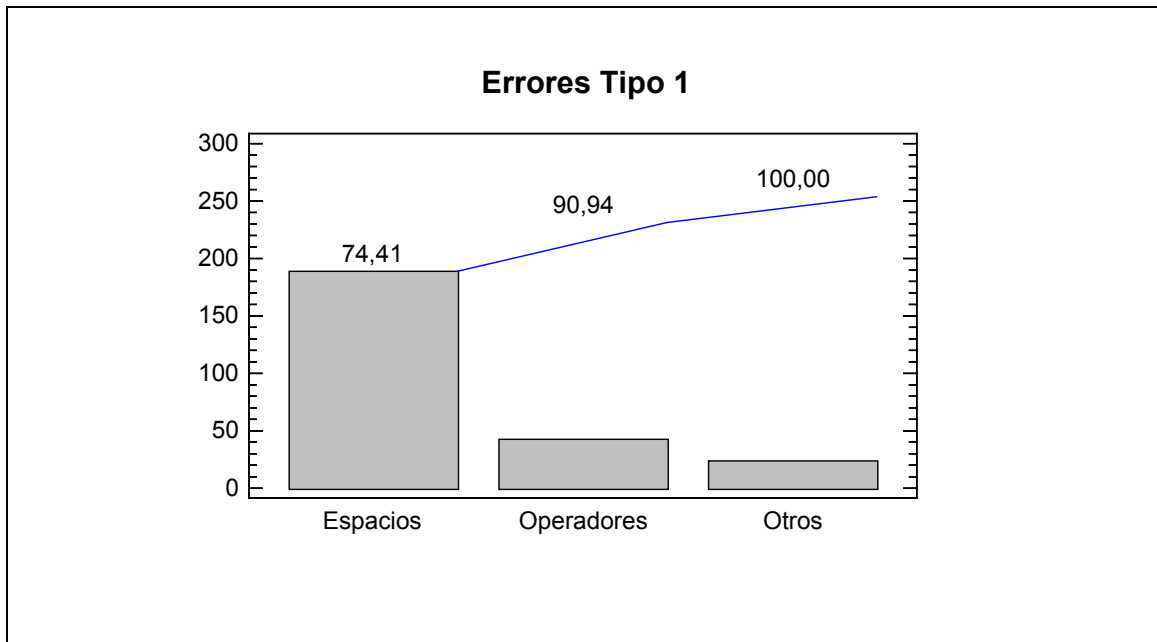


Figura 89. Pareto errores tipo 1.

11.2 SESIÓN DE LLUVIA DE IDEAS

La sesión de lluvia de ideas dio como resultado la siguiente lista de posibles soluciones, agrupadas según la CTQ que se intento atacar. Ver tabla 35.

Tabla 35. Sesión de lluvia de ideas.

IDEAS CTQ 1	PUNTAJE	TIPO
Hacer los queries de la forma procedimiento almacenado.	13	3
Crear un modelo de datos n-dimensional.	15	1
Gráficos tipo jpg.	11	3
Que haya una distribución del uso de memoria en el servidor.	6	2
Crear un campo entero que indiquela primera letra→ tabla de alfabeto.	7	1
Crear una tabla de más buscados.	8	1
Programar la ejecución de procesos a diferentes horas.	11	2

IDEAS CTQ 2	PUNTAJE	TIPO
Subconsultas.	11	3
Detalles MARC.	15	3
Búsqueda Avanzada.	11	3
Guardar Referencias.	13	3
Navegabilidad.	15	3
Ver Resumen.	11	3
Material Completo.	15	3 – 1

IDEAS CTQ 3	PUNTAJE	TIPO
Realizar las consultas sin utilizar operadores.	15	3
La búsqueda simple utilice todos los criterios de búsqueda.	15	3

Las ideas fueron agrupadas nuevamente en tres grupos, según un tipo de solución específico de la siguiente forma:

Tipo 1: Crear un nuevo modelo de datos.

Tipo 2: Optimizar el uso del servidor.

Tipo 3: Desarrollar un nuevo sistema.

Las ideas con un puntaje inferior a 9 puntos no son tenidas en cuenta. Los criterios para asignar el puntaje es el siguiente.

Tabla 36. Puntaje de criterio para la selección de ideas.

Criterio	Medida	Puntos		
		5	3	1
Viabilidad	Puntos del conjunto	15 – 20	10 - 14	4 – 9
Costo	Millón de pesos	0 – 1	1 – 3	3 o más
Recursos	Equipos	1 – 2	3 – 4	5 o más
Personal	Personas	1 – 2	3 – 6	7 o más
Tiempo	Meses	0 – 3	4 – 12	13 o más
Efectividad	Porcentaje que ataca la CTQ	100%	60%	10%
Originalidad	idea	Novedosa en la compañía	Antigua con mejoras	Antigua sin mejoras

11.3 DETERMINAR QUE SOLUCIONES TENDRAN MAYOR IMPACTO EN LOS REQUERIMIENTOS DEL CLIENTE

Las soluciones que se plantean son un paquete de pequeñas soluciones que deben realizarse.

1. Crear un nuevo modelo de datos.
 - 1.1. Crear un modelo Data Warehousing.

2. Optimizar el uso del servidor.
 - 2.1. Distribuir el uso de la memoria del servidor.
 - 2.2. Programar la ejecución de procesos a diferentes horas.

3. Desarrollar un nuevo sistema.
 - 3.1. Servicios.
 - 3.1.1. Subconsulta.
 - 3.1.2. Búsqueda avanzada.

- 3.1.3. Navegar por los resultados.
- 3.1.4. Guardar referencias.
- 3.1.5. Ver detalles MARC.
- 3.1.6. Ver resumen.
- 3.1.7. Material completo.
- 3.2. crear los gráficos tipo JPG.
- 3.3. Realizar consultas sin la utilización de operadores.
- 3.4. la búsqueda simple sea solo un cuadro, y los criterios de selección sean todos.
- 3.5. Los queries sean realizados por medio de procedimientos almacenados y no por código en la pagina.

La escala utilizada es:

Descripción	Valor
Alto	5
Medio	3
Bajo	1

La importancia de los requerimientos es dada según el porcentaje que se asignó a las causas del problema en el paso de definición del problema. Velocidad 30%, Servicios 20%, Formas de realizar la consulta 50%. Las expectativas del cliente son agrupadas en las causas del problema.

Cuadro QFD.

Como		Propuestas						
Requerimientos	Importancia	Desarrollar un nuevo sistema		Crear un nuevo modelo de datos		Optimizar el uso del servidor		TOTAL
		Ofrecer más servicios	1	5	5	1	1	
Material completo	1	3	3	5	5	1	1	9
Interfaz gráfica amigable	3	3	9	1	3	1	3	15
Intuitivo	5	5	25	1	5	1	5	35
Agilidad en la respuesta	3	1	3	5	15	5	15	33
TOTAL			45		29		25	99

Diagrama de Pareto

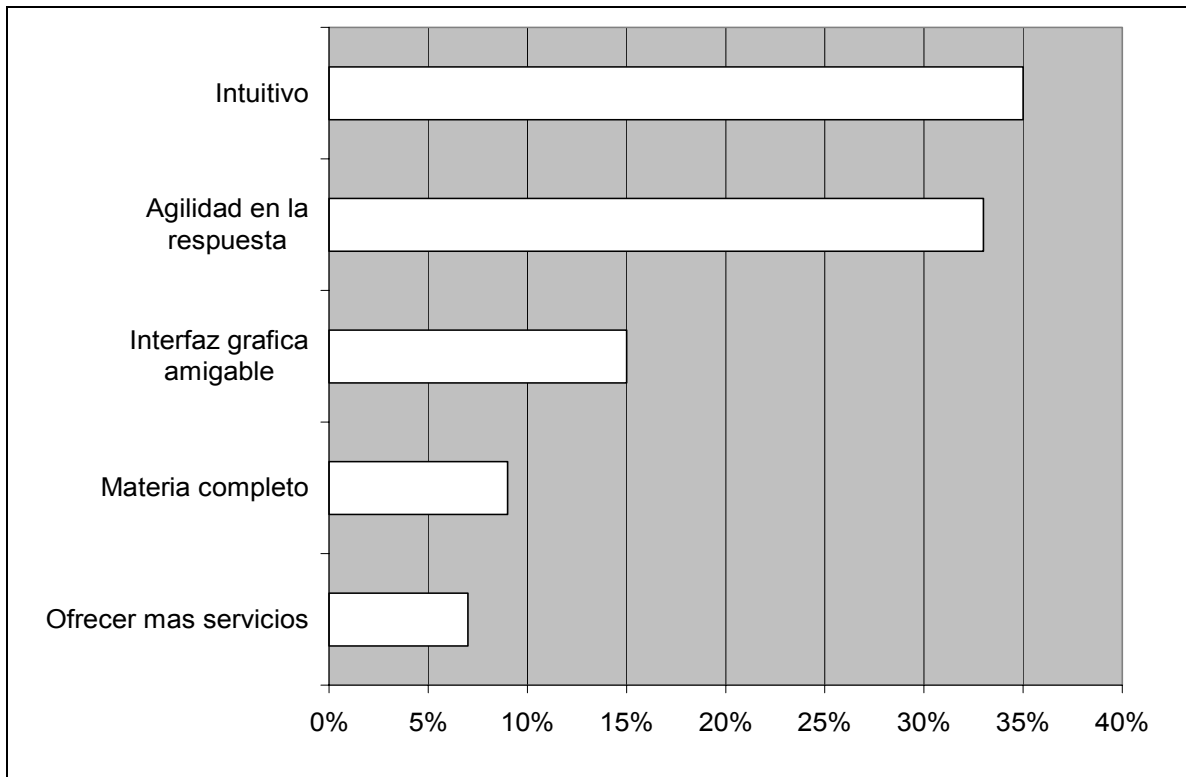


Figura 90. Pareto requerimientos del cliente.

Después del análisis de QFD y el diagrama de Pareto, la solución que se debe realizar primero, debido a que satisface en un mayor porcentaje a las expectativas del cliente, es desarrollar un nuevo sistema, la segunda solución es crear un nuevo modelo de datos de la base de datos, y la tercera solución es optimizar el servidor. Si en el ejemplo anterior las soluciones 1 y 2 tuvieran un porcentaje igual o similar, se analiza con el diagrama de Pareto que solución tiene mayor impacto sobre las expectativas del cliente en el orden de importancia, por ejemplo si la solución 3 tuviera un valor de 42 y la 2 de 43, el análisis debe decir que solución se realiza dependiendo del impacto que tenga en el requerimiento de mayor importancia en la matriz.

11.4 ANALISIS DE MODOS DE FALLO Y EFECTOS

Tabla 37. Análisis de Modos de Fallo y Efectos

ANÁLISIS DE MODOS DE FALLO Y EFECTOS:												
Nombre del Proceso			Fecha: 15/06/2004			Número revisión: 1						
Preparado por: Diana Zambrano Sergio Contreras			Revisado por:			Fecha de Revisión:						
Paso del Proceso	¿Que pudo pasar?		¿Por qué y con qué frecuencia?		¿Cómo se puede prevenir?		Plan de Acción		Resultados de la Acción			
	Modos de Fallo	Efectos de la Falla	Principales Causas	OCU	Controles Actuales	PROB	NPR	Acciones Recomendadas	Responsable	OCU	PROB	NPR
Desarrollo	Errores de código en la programación n.	El sistema puede arrojar resultados no integros.	No hay claridad en la sintaxis.	1	Ninguno.	10	100					
	Más de una persona esta modificando el mismo programa.	Se pierden las modificaciones realizadas por otros.	Fallas en la comunicación entre el equipo de trabajo.	10	Manejo de versiones en los archivos.	7	280					
Servicios	La aplicación web no esta disponible.	El usuario no puede utilizar la aplicación.	Servicio del servidor web detenido.	1	Chequeo manual del servidor.	10	10					
	La red fisica presenta problemas. Conexión con la base de datos no activa.	El tiempo de respuesta es muy alto. La aplicación no arrojará resultados.	Cambios inesperados en la red. El servicio de la base de datos esta detenido.	7	Monitoreo de red.	4	280					
			El servicio de la base de datos esta detenido.	10	Ninguno	10	1000	Verificar constantemente la conexión.	Administrador de servidores.	1	1	10

11.5 DESARROLLAR EL PLAN DE CAPACITACIÓN

La necesidad de la capacitación se presenta ya que la metodología que se está utilizando en el proyecto (Seis Sigma) es una metodología nueva para la División de Servicios de Información.

Tema	Introducción a la metodología Seis Sigma.			
Objetivo	Presentar a los asistentes los conocimientos básicos de un modelo de la metodología Seis Sigma para el desarrollo de Software.			
Metodología	Presentación			
Dirigido a:	Personal de la DSI y profesores de la Escuela de Ingeniería de Sistemas y de Ingeniería Industrial de la UIS.			
Recursos	<input type="checkbox"/>	video-Beam	<input type="checkbox"/>	Salón
	<input type="checkbox"/>	Computador	<input type="checkbox"/>	Manuales
	<input type="checkbox"/>	Tablero	<input type="checkbox"/>	Refrigerios
	<input type="checkbox"/>		<input type="checkbox"/>	
Presupuesto		60.000 \$	Responsables	Sergio Contreras
				Diana Zambrano
Ubicación		Auditorio INSED	Fecha inicio	14/07/2004
			Fecha fin	21/07/2004
Cronograma				
Sesión	Tema	Fecha	Hora	Duración
1	Definir-Medir	14/07/2004	10:00 AM	2 h
2	Analizar-Diseñar	16/07/2004	10:00 AM	2 h
3	Desarrollar-Validar	21/07/2004	10:00 AM	2 h
Aprobado	<input type="checkbox"/>			
NOTAS				

Plan de la Sesión uno.

Tema	Introducción y Fases Definir - Medir		
Objetivos	Hacer una introducción de la metodología Seis sigma y presentar los pasos de la fases definir y medir del modelo DMADDV.		
Metodología	Presentación.		
Lugar	Auditorio INSED	Hora	10 A.M.
Fecha	14/07/2004	Duración	2 horas
Cronograma			
Introducción a la Metodología Seis Sigma.			
Fase Definir.			
Fase Medir.			
Preguntas			
Responsables	Diana Zambrano		
	Sergio Contreras		

Plan sesión dos

Tema	Fases Analizar – Diseñar.		
Objetivos	Presentar los pasos de la fases Analizar y Diseñar del modelo DMADDV.		
Metodología	Presentación		
Lugar	Auditorio INSED	Hora	10 A.M.
Fecha	16/07/2004	Duración	2 horas
Cronograma			
Fase Analizar.			
Fase Diseñar.			
Preguntas.			
Responsables	Diana Zambrano		
	Sergio Contreras		

Plan sesión tres.

Tema	Fases Desarrollar – Validar.		
Objetivos	Presentar los pasos de la fases Desarrollar y Validar del modelo DMADDV. Y evaluar el alcance de la capacitación.		
Metodología	Presentación		
Lugar	Auditorio INSED	Hora	10 A.M.
Fecha	21/07/2004	Duración	2 horas
Cronograma			
Fase Desarrollar.			
Fase Validar.			
Preguntas.			
Evaluar Alcance de la capacitación.			
Responsables	Diana Zambrano		
	Sergio Contreras		

11.6 IDENTIFICAR LOS SISTEMAS AFECTADOS DE OTRAS DEPENDENCIAS

No existe un relación directa con otros sistemas, aunque existe una relación directa con las bases de datos de biblioteca (una que contiene el material de libros, tesis, y analíticas, y otra que contiene las revistas) que es utilizada también por el sistema Libruis. Debido a que las bases de datos se encuentran alojadas en otro servidor se debe realizar el análisis como si fuera un sistema.

Tópico	Descripción
Plataforma..	Unix
Motor de la base de datos.	Informix 4
Driver de conexión.	OLEDB
Modelo de datos.	

11.7 DESARROLLAR EL NUEVO MAPA DE PROCESOS

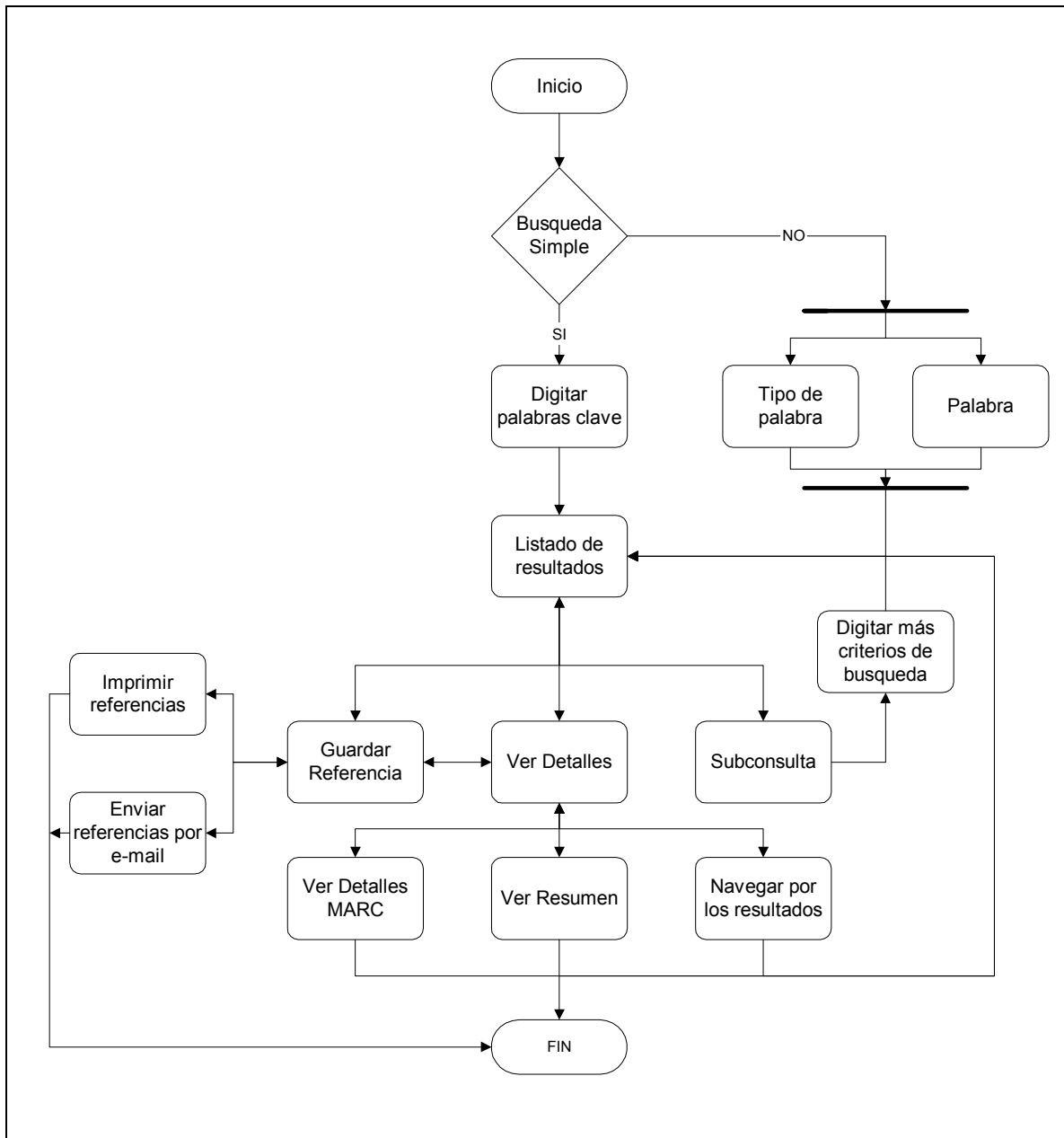


Figura 91. Nuevo mapa de procesos.

El sistema inicia con la opción de realizar una búsqueda simple, que la conforma un cuadro de texto donde el usuario final escribe las palabras clave a buscar con la restricción que sólo sean cuatro palabras como máximo y un botón de buscar. Presenta además la opción de realizar una búsqueda

avanzada donde se muestran cuatro listas donde cada una tiene las opciones de especificar más la búsqueda autor, título, materia, libro, revista, tesis o analítica, y cuatro cajas de texto para introducir la palabra clave de la opción.

Si la consulta es exitosa, se observan los resultados obtenidos, y se presentan tres: opciones guardar la referencia temporalmente, ver los detalles de una referencia y realizar una subconsulta de los datos.

La subconsulta se realiza para obtener resultados más precisos de la consulta anterior, y se realiza de igual forma que la simple.

Ver detalles muestra la información general del material, y tiene cuatro opciones que son:

- Ver Detalles MARC: Muestra la información del material en el formato MARC.
- Ver Resumen: Muestra el resumen del material, en la actualidad solo el material bibliográfico que es tesis cuenta con este servicio.
- Guardar referencia: Es la opción de guardar la referencia temporalmente, presentando dos opciones:
 - Enviar la lista de referencias a un e-mail.
 - Imprimir la lista en papel, la restricción es que la estación de trabajo donde se este realizando la consulta tenga conexión con una impresora.
- Navegar por los resultados: Cada autor y cada materia tienen un enlace para que se realicen nuevas consultas sobre la palabra escogida y arroja una nueva lista de resultados.

11.8 PRESENTAR AL COMITÉ PARA APROBAR

- Causas de la variación.
- Criterio de solución de lluvia de ideas.
- Impacto de mejoras. (QFD)
- AMFE.
- Plan de capacitación general.
- Nuevo mapa de procesos.

12. FASE DISEÑAR

12.1 SELECCIONAR LA METODOLOGÍA DE DESARROLLO DE SOFTWARE

Está claro que Seis Sigma no es un proceso de desarrollo de software, en lugar que eso, es más generalizado, es una metodología de mejoramiento de procesos y productos. Aunque algunos de los elementos de Seis sigma en el desarrollo de software son incluidos en el marco de psp, muchos otros están solamente disponibles en seis sigma y no son incorporados en psp. Es por esto, que seis sigma puede ser usado con cualquier modelo de desarrollo de software que sea estable, como PSP, que proporciona una excelente base para usar las herramientas de Seis Sigma en la ingeniería del software ya que son tecnologías complementarias.

La relación de Seis sigma en el software con psp, se puede caracterizar en la diferencia de sus objetivos, en donde los objetivos de psp son un subconjunto de los objetivos asociados al proyecto seis sigma para software.

- Los objetivos principales de psp son basados en el mejoramiento continuo del rendimiento del equipo de desarrollo de software en términos de costos, tiempo y calidad entregada.
- Los objetivos de Seis sigma incluye los objetivos de psp, pero también se usa para alcanzar otros muchos objetivos del negocio, como el mejoramiento del servicio del cliente, mejorar la satisfacción del cliente y muchos otros mencionados anteriormente.

La razón principal por la cual PSP es la metodología de desarrollo de software seleccionada en este proyecto, es por las dimensiones del proyecto y la conformación del equipo de trabajo, ya que PSP se concentra en las prácticas de trabajo de los ingenieros en una forma individual.

PSP se ensambla muy bien al modelo DMADDV de Seis Sigma, dado que proporciona un proceso con medidas que pueden ser implementado directamente con Seis sigma, y a su vez se complementa con la voz del cliente y con las herramientas que se utilizan para analizar, desarrollar y validar.

EL PROCESO PERSONAL DEL SOFTWARE

El proceso personal del software⁶³ es un proceso de mejoramiento diseñado para ayudar a controlar, administrar y mejorar la forma en que se trabaja individualmente. Está estructurado por formularios, guías y procedimientos para desarrollar software.

Así mismo les enseña a cómo planear y darle un seguimiento a su trabajo, a utilizar un proceso bien definido y medido, a establecer metas medibles, y finalmente a la utilización del rastreo constante para alcanzar dichas metas. PSP les demuestra a los ingenieros a cómo manejar la calidad desde el principio del trabajo, a cómo analizar los resultados de cada trabajo, y a cómo utilizar los resultados para mejorar el proceso de un proyecto siguiente.

⁶³ Traducción al inglés de PSP: Personal Software Process

12.1.1 Principios de PSP.

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

- Cada ingeniero es esencialmente diferente; para ser más precisos, los ingenieros deben planear su trabajo y basar sus planes en sus propios datos personales.
- Para mejorar constantemente su funcionamiento, los ingenieros deben utilizar personalmente procesos bien definidos y medibles.
- Para desarrollar productos de calidad, los ingenieros deben sentirse personalmente comprometidos con la calidad de sus productos.
- Cuesta menos encontrar y arreglar errores en la etapa inicial del proyecto que encontrarlos en las etapas subsecuentes.
- Es más eficiente prevenir defectos que encontrarlos y arreglarlos.
- La manera correcta de hacer las cosas es siempre la manera más rápida y más barata de hacer un trabajo.

12.1.2 Estructura del Proceso.

Una vez planteada la declaración de los requerimientos, el primer paso en el proceso PSP es el de la Planeación. Este proceso cuenta con una guía del proceso para ayudar en el desarrollo del trabajo y un resumen del plan para registrar los datos de la planeación. A medida que los ingenieros siguen la guía del proceso para realizar el trabajo, deben realizar un registro del tiempo y defectos. Al final del trabajo, durante la fase postmortem se compara el desempeño final con el desempeño planeado, también se abstraen los datos de tiempo y defectos de los registros y se debe completar el resumen del plan. Cuando este listo se entrega el producto junto con el formato. Figura 92 y 93.

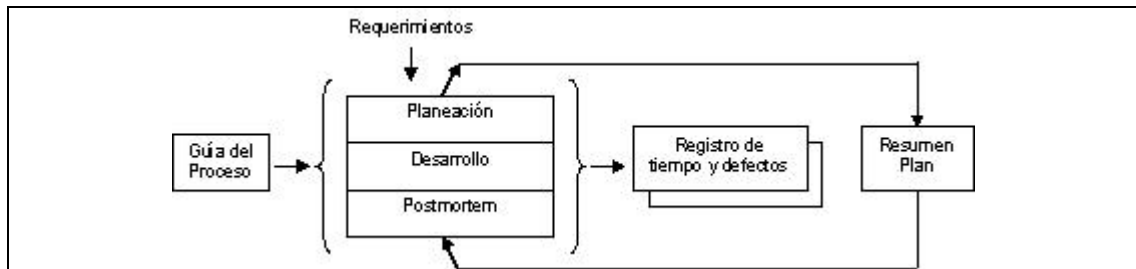


Figura 92. Estructura del proceso

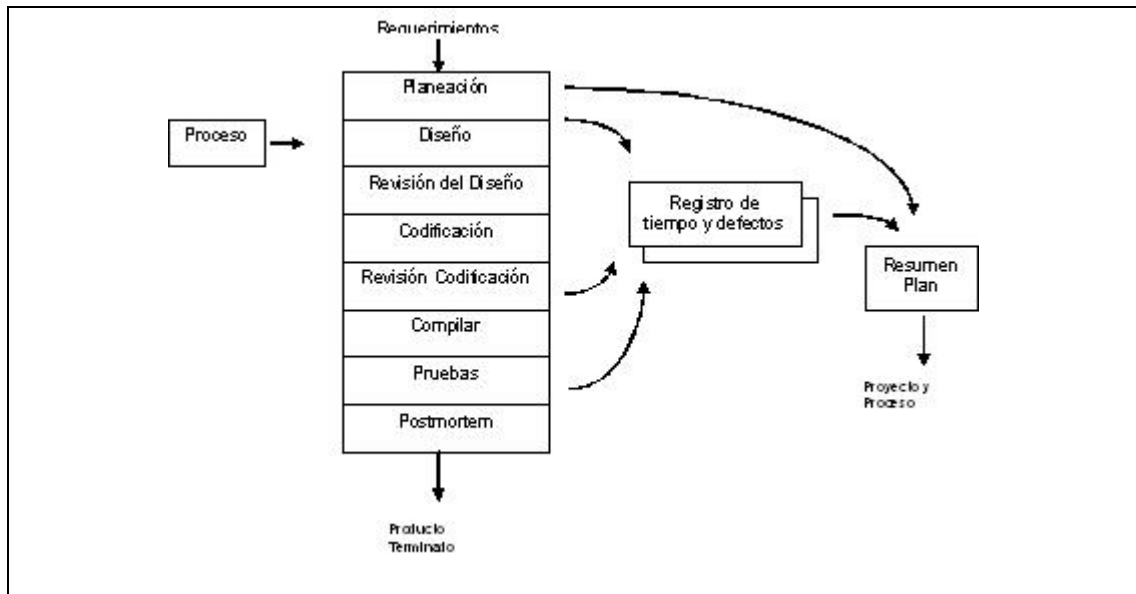


Figura 93. Estructura del proceso

PSP tiene un número de métodos que generalmente los ingenieros no practican, estos métodos se introducen a través de una serie de siete versiones del proceso completo. Estas versiones se etiquetan desde PSP 0 hasta PSP 3, y cada versión tiene un sistema similar de registros, de formatos, guías, y estándares. Las guías que intervienen en el proceso definen los pasos para cada parte del proceso.

Los registros y los formatos proporcionan las plantillas para almacenar los datos y los estándares ayudan a dirigir a los ingenieros durante todo el proyecto que desarrollan, desde el principio hasta el final de éste.

La evolución del modelo PSP se describe a continuación y es la que se muestra en la figura 94.

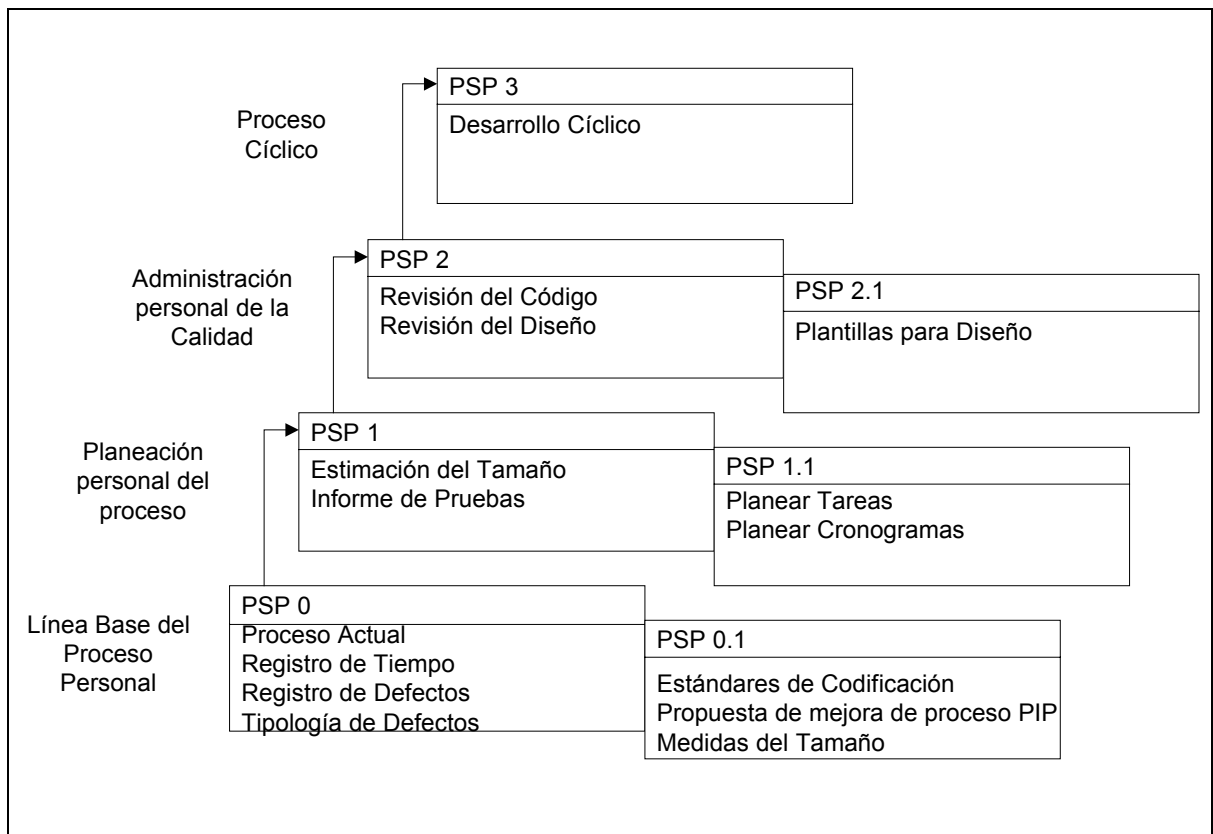


Figura 94. Estructura por niveles de psp.

LÍNEA BASE DEL PROCESO PERSONAL

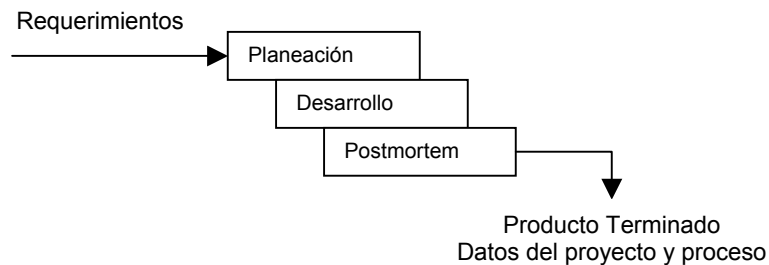
El objetivo principal del nivel inicial es el de proveer un marco bien definido para realizar la recolección de todos los datos que intervienen en el proyecto. Esta primera fase es muy importante, ya que proporciona un panorama general de qué métodos utilizar para "atacar" el problema.

PSP 0

Proceso Actual

Identifica el proceso de desarrollo de software sobre el que se va a trabajar, se pueden usar los siguientes pasos: Diseño, codificación, compilación y se deben realizar las siguientes tareas del flujo de trabajo:

- Planeación: Diseñar la forma como se hará el trabajo.
- Desarrollo: Desarrollo del actual proceso de software.
- Postmortem: Comparación del rendimiento actual con lo planeado.



Registro del Tiempo

Se usa para medir el tiempo que se gasta en cada parte del proceso PSP y así poder determinar en donde se consume la mayor cantidad de tiempo. El log de registros del tiempo se puede ver en el anexo xx.

Registro de Defectos

Se usa para mantener los datos de cada defecto encontrado y corregido y además poder determinar en donde ocurren la mayoría de las fallas. Para ver el log de registro de defectos vea el anexo xx.

Estándar de los tipos de defectos

Existen algunos estándares para describir los tipos de defectos. Estos se encuentran en el anexo E.

PSP 0.1

Estándares de Codificación

Se utilizan formatos que pueden contener: Encabezado, identificadores, comentarios, etc.

Medidas del Tamaño

Durante el proceso de planeación se estima el tamaño del trabajo en función de las líneas de código (LOC). Se debe desarrollar un estándar para determinar la forma de contar, pueden contener lo siguiente:

- Líneas modificadas o borradas.
- Comentarios o líneas en blanco.
- Líneas con varias declaraciones.
- Declaraciones
- Etiquetas
- Símbolos como { }, begin/end, then, else, case....

Propuesta de Mejoramiento del Proceso⁶⁴

Proporciona un registro de los problemas del proceso y las ideas para el mejoramiento.

PLANEACIÓN PERSONAL DEL PROCESO

Este proceso hace referencia y puntualiza los detalles finales de la estimación de recursos y cronograma, para continuar de lleno con la planeación de estos dos puntos. Una vez que se cuentan con estos planes, el programador tiene una vista general de su proyecto y puede tener un mejor juicio de la precisión de éste.

⁶⁴ PIP: Propose Improvement Process

PSP 1

Estimación del Tamaño

Cualquiera de los siguientes acercamientos pueden ser usados para estimar las Líneas de Código.

- Método PROBE⁶⁵.
- Modelo Cocomo⁶⁶

Informe de Pruebas

Es usado para mantener un registro de las pruebas hechas y los resultados obtenidos.

PSP 1.1

Planear Tareas

Implica la estimación del tiempo de desarrollo y los datos completos en cada tarea del proyecto.

Planear Cronograma

El cronograma se usa para registrar las horas actuales invertidas en un periodo calendario determinado. También se utiliza para relacionar las tareas previstas con el horario del calendario.

ADMINISTRACIÓN PERSONAL DE LA CALIDAD

Este proceso introduce las revisiones del diseño y el código que hacen parte del programa, así como las medidas de calidad y la evaluación.

⁶⁵ PROBE: PROxy-Based Estimating

⁶⁶ COCOMO: Constructive Cost Model

PSP 2

Revisión del Código

Estas revisiones pueden incluir:

- Verificar la inicialización de las variables y parámetros.
- Formatos de llamadas a las funciones.
- Verificar los nombres.
- Verificar cadenas.
- Verificar todos los archivos.
- Verificar punteros.
- Verificar los formatos de salida.
- Verificar los operadores lógicos.

En general verificar en cada línea de código las sintaxis de las instrucciones y la puntuación adecuada y asegurarse que hagan parte de los estándares de codificación.

Revisión del Diseño

- Asegurar que el diseño enmarque las especificaciones de los requerimientos.
- Verificar la lógica del programa.
- Verificar los casos especiales.
- Verificar el uso de las funciones.

PSP 2.1

Plantillas para Diseño

Introduce cuatro nuevas plantillas que proveen un marco ordenado que sirve para el registro correcto de los diseños que el programador realiza. Pueden incluir:

- Escenario operacional.
- Especificación funcional.
- Especificación del estado.
- Especificación lógica.

PROCESO CÍCLICO

Con este nuevo nivel se llega a un nuevo concepto y se introduce una nueva fase, la fase de realizar el proceso personal creado de una manera cíclica y uniforme.

PSP 3
Desarrollo Cíclico: Este nivel ayuda al desarrollador a desarrollar programas más largo en poco tiempo y con menos errores, esto quiere decir que el programador ahora tendría una forma de programar única y bien definida.

12.2 DISEÑO DE FORMATO DE DOCUMENTACIÓN DE PROGRAMAS

Los formatos presentados para la documentación de programas fueron presentados al líder del equipo, y luego el formato seleccionado fue entregado y explicado a cada uno de los desarrolladores del equipo.

Formato 1

Tabla 38. Formato uno de documentación de programas.

Programa		Proyecto	
Autor		Fecha	
Directorio de Ubicación		# versión	
Descripción		Base de Datos, Tablas	

Formato 2

Tabla 39. Formato dos de documentación de programas.

Proyecto			
Programa		Fecha versión 1	
Autor		Fecha versión 2	
Directorio de ubicación		Fecha versión 3	
Base de Datos, Tablas, Campos			
Descripción			

Formato 3

Tabla 40. Formato tres de documentación de programas.

Proyecto	
Programa	
Descripción	
Versión	
Autor	
Fecha	
Directorio de ubicación	
Variables	
Directorio de archivos fuente	
Base de datos, tablas, campos	

El formato seleccionado es el número uno, debido a que es el más corto, sencillo y se puede repetir sin importar las veces que exista una nueva

versión del programa, registrando los cambios históricamente sin llegar a ocupar demasiado espacio en el código del programa.

12.3 DEFINIR LA SEGURIDAD

En la definición de la seguridad muchos de los tipos a considerar ya se encuentran establecidos por la División de Servicios de Información, y se acomodaron a las exigencias del proyecto, algunos tipos de seguridad no aplicaban debido a la magnitud del proyecto.

Tabla 41. Asignación de seguridad.

SEGURIDAD		
Proyecto	Consulta de catálogo bibliográfico de la UIS vía Web.	
Tipo	Responsable	Fecha Asignación
Hardware / Software	Administrador de servidores de aplicaciones	06/2004
DATOS	Administrador de base de datos	06/2004
Documentación	Administrador de dominio	06/2004
Legal	N/A	-
Acceso de personal	N/A	-
Física	N/A	-
Sistema de Información	Auditoría	06/2004
	Identificación y autenticación	06/2004

Tabla 42. Asignación de roles.

Rol	Nombre
Administrador de base de datos	Ing. Benjamín Pico Merchán
Administrador de dominio	Ing. Walter Calderón
Administrador de servidores de aplicaciones	Ing. Benjamín Pico Merchán
	Ing. Walter Calderón
Auditoría	Ing. Leonilde Martínez
Identificación y autenticación	Sistema Informix
N/A	No Aplica

12.4 ESPECIFICACIÓN DEL DISEÑO

El diseño contiene cuatro etapas, según la pirámide en la figura 35, en el paso de especificación del diseño.

12.4.1 Diseño de datos.

Se basó en el concepto de modelos de datos dimensionales, debido a que se busca la rapidez en el tiempo de respuesta de la consulta y que existen grandes volúmenes de datos en la base de datos, donde el modelo n-dimensional tiene sus ventajas con los modelos entidad-relación.

Las diferencias de los modelos son las siguientes:

Tabla 43. Diferencias entre modelos dimensionales y E-R.

Entidad – Relación	Dimensional
1 tabla por entidad.	1 Tabla Fact por organización de datos.
Minimizar la redundancia.	Maximizar la comprensión.
Óptimo para actualizar datos.	Óptimo para recuperación de datos.
Modelo de Transacción.	Modelo Data Warehousing .

Los modelos dimensionales son modelos entidad relación desnormalizados, que buscan la fácil comprensión del lector, y la optimización para los reportes.

Los pasos para determinar el modelo dimensional son los siguientes:

- a. **Data Mart:** en los términos más simples es la trascendencia que tendrá la fuente de datos, en otras palabras, es la parte del modelo del negocio en que el equipo se centrará. Por ejemplo, órdenes de compra, envíos, ventas, pagos, etc.

“Consulta del catálogo bibliográfico”.

- b. **Granularidad de la tabla Fact:** es el nivel más bajo o el detalle más profundo de lo que significa un registro de la tabla de Fact, ya sea una transacción individual en un data mart de un banco, o una transacción de una venta en un sistema de un almacén.

“La información de un material bibliográfico dependiendo de una palabra”

- c. **Dimensiones:** son las tablas que contienen la información descriptiva de la tabla Fact, ya sea numérica o texto.

- Palabras (palabrasauttitmat).
- Autores (autores).
- Títulos (titulos).
- Materias (materias).
- Materiales (materialesbib).
- Tipos de documentos (tiposdoc).
- Formas (formascon).
- Notas (matbib_not).
- Ubicación del ejemplar (ubicacioneseje).
- Tiempo (tiempo).
- Dirección del resumen de las tesis (dir_resu_tesis).
- ISBN (isbn).
- ISSN(issn).
- Estado del ejemplar (estadoseje).
- Ejemplares (ejemplares).

12.4.2 Diseño arquitectónico.

Arquitectura de Aplicaciones Distribuidas

La Arquitectura de Aplicaciones Distribuidas en Internet (Distributed interNet Architecture - DNA) es un esquema que permite a los desarrolladores de software diseñar y construir aplicaciones distribuidas, orientada principalmente a las soluciones en Internet o en una Intranet. La arquitectura Windows DNA especifica como:

- Desarrollar aplicaciones distribuidas de múltiples capas, escalables y robustas utilizando la plataforma Windows.
- Extender datos existentes y aplicaciones externas para soportar el Internet.
- Soportar una amplia variedad de clientes, maximizando el alcance de las aplicaciones.

Windows DNA está basada en el concepto que dice que una aplicación distribuida debe estar separada lógicamente en tres partes o capas, como se explicaba en el modelo de tres capas de la arquitectura Cliente/Servidor. Esta separación incrementa la escalabilidad y hace que la aplicación sea más fácil de manejar y de actualizar.

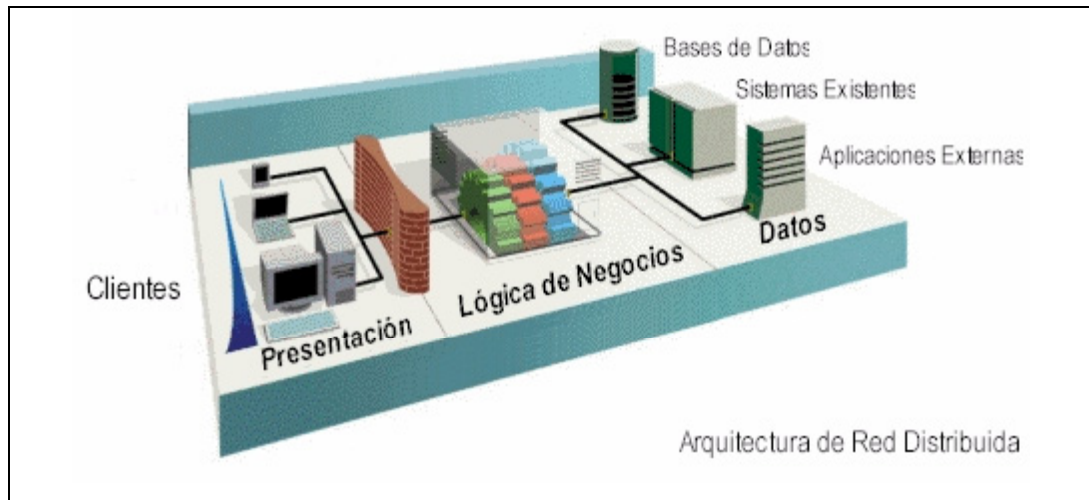


Figura 96. Arquitectura de red distribuida.

Los componentes de presentación se encargan de la interacción con el usuario y de solicitar servicios a la aplicación por medio de llamados a los componentes de la capa media que corresponde a la capa de las reglas del negocio. Los componentes en esta capa se encargan de realizar la lógica del negocio y de hacer solicitudes a la base de datos. La aplicación se hace más flexible porque los clientes pueden llamar a los componentes del servidor y mejorar la reutilización de código.

Las arquitecturas de múltiples capas, también son llamadas arquitecturas centradas en el servidor, debido a que ellas solamente permiten que la implementación de las reglas del negocio de una aplicación funcione en la capa intermedia en el servidor, independiente tanto de la interface de presentación como de la implementación de la base de datos.

Dentro de las ventajas que ofrece una arquitectura DNA se destacan las siguientes:

- Soporte de múltiples lenguajes. La implementación de la lógica de la aplicación puede realizarse en diferentes lenguajes de programación.

- Implementación centralizada. Los componentes o implementaciones de la lógica de la aplicación se encuentran localizadas en un mismo servidor facilitando el desarrollo, el mantenimiento y la implantación.
- Balance de carga. El balance de carga corresponde a la distribución adecuada de los componentes que hacen parte de una aplicación en diferentes servidores de modo que un computador no esté sobrecargado y otro esté sin carga. En la medida que la aplicación crece en número de componentes y en número de usuarios se debe procurar repartir los componentes en diferentes servidores permitiendo mayor escalabilidad⁶⁷.
- Eficiencia en el acceso a los datos. Las limitaciones por problemas de conexión a bases de datos son minimizadas debido a que solo el componente de aplicación accede a la conexión y no cada uno de los clientes. Además el cliente no necesita utilizar controladores ni conexiones a la base de datos.

Capa de presentación

Incluye la interfase de usuario y la lógica necesaria que determina como se le muestra al usuario la información manejada por la aplicación. La lógica encerrada en esta capa interactúa con la capa de negocios.

⁶⁷ La escalabilidad hace referencia a la capacidad de una aplicación de incrementar el número de usuarios sin que el desempeño de la misma disminuya.

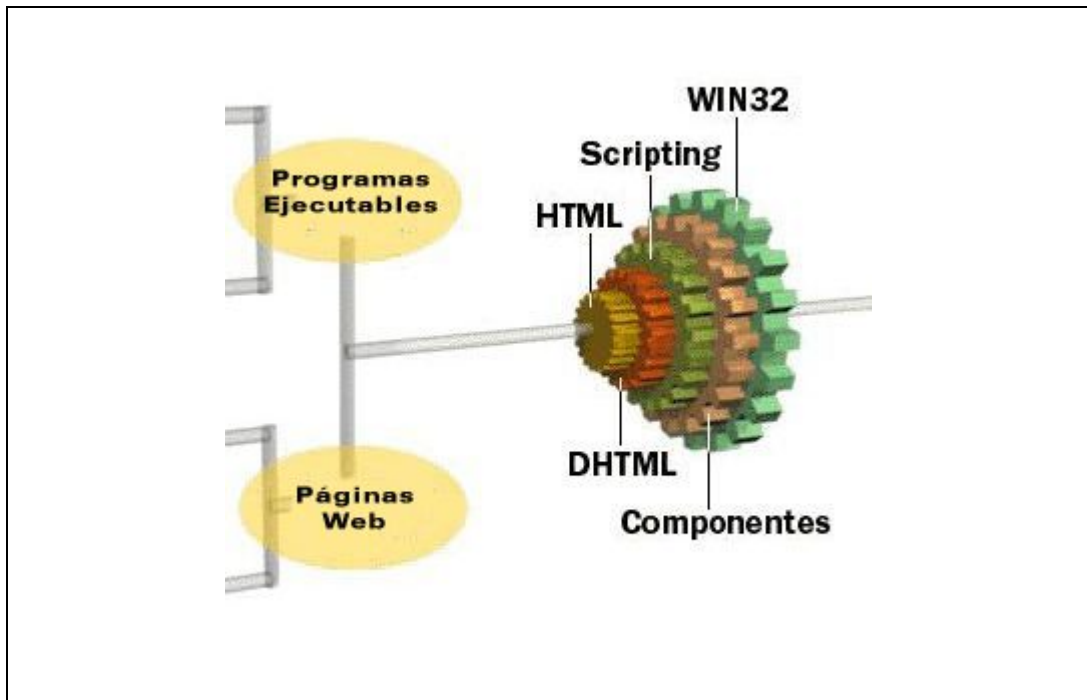


Figura 97. Capa de presentación.

Windows DNA ofrece una amplia variedad de interfaces de presentación o de usuario, permitiendo implementar una misma aplicación adecuada para las diferentes plataformas que utilizan los clientes. Los servicios disponibles para implementar la capa de presentación son:

- HTML: Permite que en cualquier ambiente de trabajo que soporte un navegador de Internet se pueda acceder a la aplicación.
- Scripting y HTML Dinámico (DHTML): Permiten añadirle a una aplicación basada en HTML una mayor funcionalidad en cuanto a la presentación. Se pueden diseñar páginas que respondan a eventos o que ejecuten una serie de acciones que pueden incluso modificar el contenido del documento HTML. Las acciones incluyen filtros gráficos, transiciones, posicionamiento de objetos, etc.

- Componentes: Algunas veces es necesario que algún segmento de la aplicación utilice el sistema operativo o la máquina sobre la cual se está ejecutando la aplicación mientras se mantiene una conexión activa a Internet. En esos casos se pueden aprovechar los componentes y los servicios de Internet para desarrollar una aplicación más robusta.
- Win32 API⁶⁸: Las aplicaciones escritas utilizando esta librería son las que mayor funcionalidad ofrecen, pero su alcance es limitado a las plataformas que soportan esta librería.

Capa de Reglas de Negocio

Implementa la combinación de políticas, reglas y algoritmos que constituyen los métodos que la empresa utiliza para realizar sus procesos de negocios.

La capa de lógica o reglas de negocios es el corazón de una aplicación distribuida y es donde se mantienen los procesos específicos de la aplicación y las reglas del negocio. La lógica de negocios implementada en los componentes hacen el puente entre la capa de presentación y la capa de datos.

Para desarrollar los componentes de la capa de la lógica de negocios de una aplicación se pueden utilizar los servicios Web, los servicios de mensajería y servicios de componentes.

⁶⁸ API, Application Programming Interface (Interfaz de Programación de Aplicaciones). Conjunto de librerías utilizadas para desarrollar y ejecutar aplicaciones sobre un sistema operativo u otra aplicación.

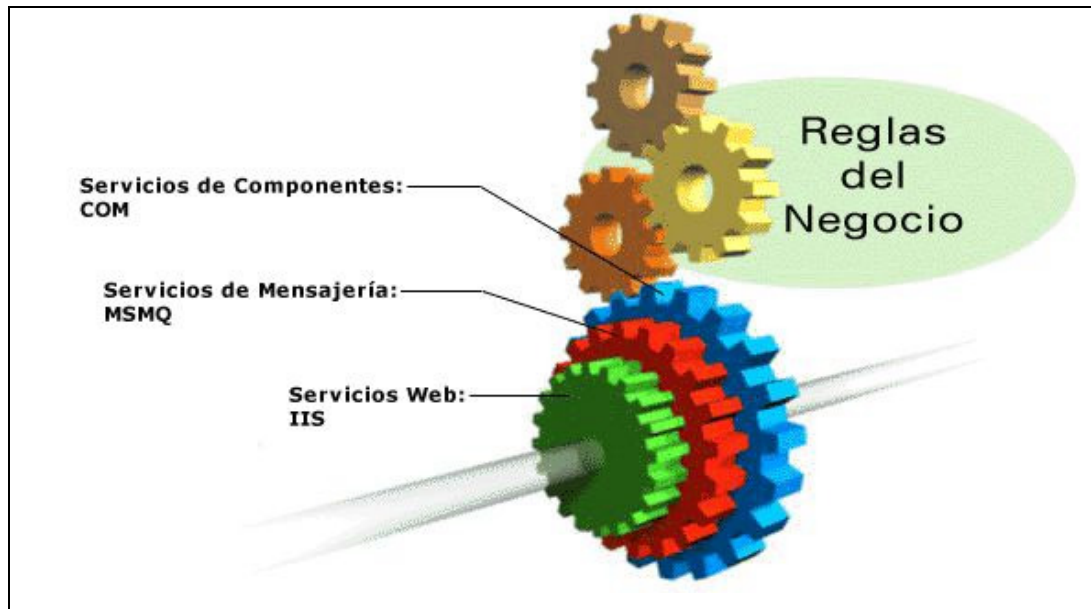


Figura 98. Capa lógica de negocios.

- **Servicios Web.** El Microsoft Internet Information Server o IIS permite el desarrollo de aplicaciones basadas en Web que pueden ser extendidas sobre el Internet o implementadas sobre una Intranet corporativa. El IIS hace uso de transacciones para hacer posible la ejecución de aplicaciones de una manera robusta y confiable. Las transacciones garantizan que cualquier procesamiento de información solicitado por un cliente en un sitio Web sea completado sin ningún error o de lo contrario los cambios generados por dicho proceso sean restablecidos en caso de error, permitiendo conservar la integridad de los datos. El IIS utiliza las Páginas de Servidor Activas o Active Server Pages (ASP) como lenguaje para generar Scripts los cuales permiten la implementación de páginas Web dinámicas de contenido interactivo.
- **Servicios de Mensajería.** Estos servicios son soportados por el Microsoft Message Queue Server o MSMQ, encargado de proveer mecanismos de comunicación a través de la red basado en un modelo de colas de mensajería e inclusive con la interoperabilidad con otros productos.

- **Servicios de Componentes.** El Microsoft Transaction Server o MTS es una utilidad que proporciona servicios para el desarrollo, implementación y administración de aplicaciones distribuidas basadas en componentes. El MTS administra todo lo relacionado con la reutilización de componentes, manejo de transacciones y conexiones a bases de datos realizadas desde los componentes. Todos los componentes utilizados por el MTS son creados bajo los parámetros de COM (Component Object Model).

Capa de Datos

Se encarga del almacenamiento y manejo de los datos. Estas funciones son responsabilidad de uno o varios motores de bases de datos. El Acceso Universal a Datos o Universal Data Access (UDA) proporciona acceso de alto desempeño a variadas fuentes de información incluyendo datos relacionales y no relacionales. Igualmente proporciona unas interfaces de programación de fácil uso, independientes de los lenguajes de programación. UDA está basado en especificaciones abiertas de modo que la mayoría de sistemas de bases de datos pueden utilizarlo.

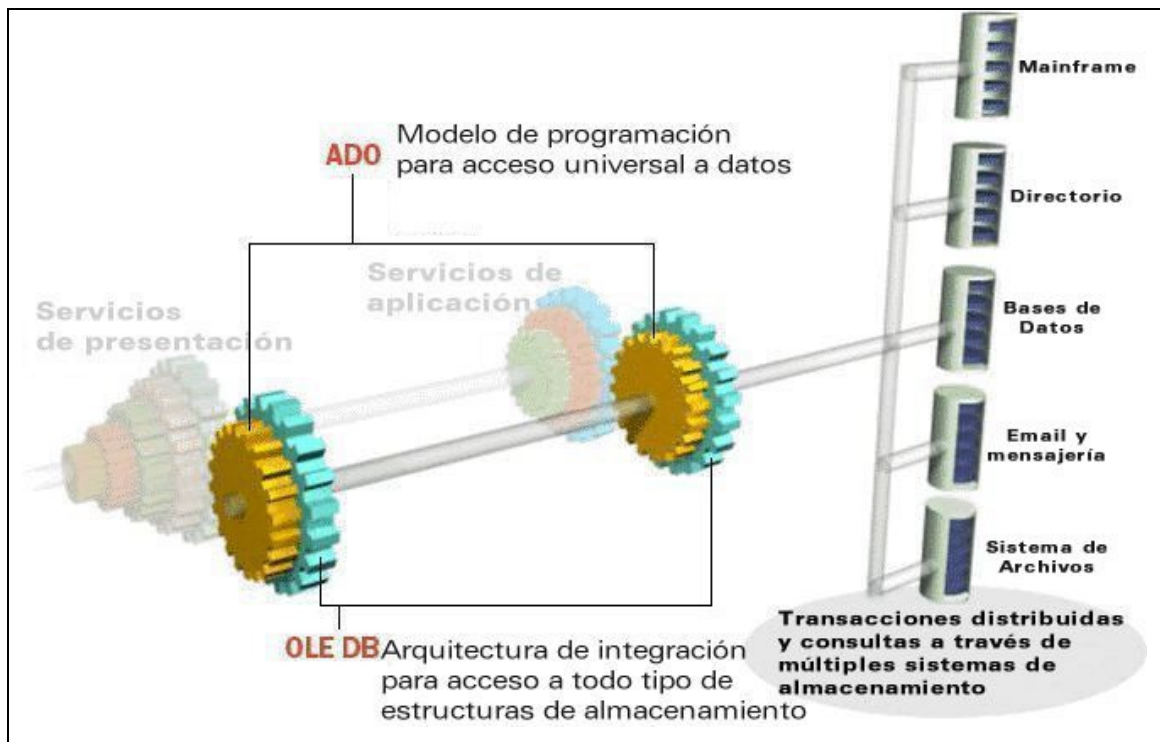


Figura 99. Capa de datos.

Componentes y Cooperación entre Componentes

Un componente es una unidad discreta de código construida con la tecnología ActiveX que define una serie de servicios o funcionalidad específica. Los componentes pueden ser de diferentes tipos y según se ubiquen en cualquiera de las capas reciben nombres diferentes en la arquitectura DNA. Así los componentes ubicados lógicamente en la capa de negocios se les conoce como *objetos de negocios*, construidos como componentes ActiveX, mientras que los que están ubicados lógicamente en la capa de presentación pueden estar implementados como Documentos Activex o como Controles ActiveX.

La cooperación o interoperabilidad entre componentes se basa en la especificación conocida como COM o Component Object Model, la cual define una serie de estándares que permiten la comunicación entre

objetos que la soportan. COM es el corazón de la arquitectura Windows DNA ya que permite la integración, la comunicación entre objetos, el acceso a una amplia variedad de clientes permitiendo hacer aplicaciones con un mayor alcance y promoviendo la escalabilidad.

COM provee una definición estándar de cómo ciertos objetos se pueden comunicar y compartir información. COM se caracteriza principalmente por ser Independiente del lenguaje, Independiente del sistema operativo y Capaz de trabajar con objetos en servidores remotos a través de una red. COM permite que diferentes componentes en diversos lenguajes de programación y que corran en diferentes sistemas operativos, interactúen con un mismo propósito en el ámbito de una misma aplicación, e igualmente permite hacer llamados a objetos remotos a través de DCOM o Distributed COM.

DCOM es un protocolo que permite que los componentes software se comuniquen directamente en una red de una manera confiable, segura y eficiente. Está diseñado para soportar diversos transportes de red lo cual lo hace compatible con casi todos los sistemas operativos y protocolos.

12.4.3 Diseño de interfaz

El diseño está basado en las líneas y formas de construcción del logotipo, este se forma a partir de una circunferencia en la que están inscritas las tres iniciales, manteniendo la curva en los bordes superior e inferior y eliminada a los 2 lados hallando un punto medio entre el recto de las 2 primeras letras (UI) y lo curvo de la tercera (S), estos elementos han servido como referencia.

Se utilizan los conceptos como limpieza visual, facilidad de navegación, rápida accesibilidad, que sirven como referencia para la página inicial del Web de la UIS.

12.4.4 Diseño procedimental.

El diseño procedimental de las búsquedas en el sistema se realizó basado en la teoría de conjuntos, especialmente utilizando la propiedad de la intersección. La utilización de la teoría es la siguiente:

Un conjunto es la reunión en un todo de objetos bien definidos y diferenciables entre sí, que se llaman elementos del mismo. Los objetos que forman al conjunto son nombrados elementos del conjunto o miembros del conjunto. Todo conjunto es una colección de objetos, pero no toda colección de objetos es un conjunto. Esta afirmación será demostrada más adelante.

La Teoría de Conjuntos es una teoría matemática, que estudia básicamente a un cierto tipo de objetos llamados conjuntos y algunas veces, a otros objetos denominados no conjuntos, así como a los problemas relacionados con estos.

Intuitiva e informalmente los objetos de estudio de la Teoría de Conjuntos quedan descritos de la siguiente manera:

1. Si x no tiene elementos, entonces x es un objeto de la Teoría de Conjuntos.
2. Si x es un conjunto, entonces x es un objeto de la Teoría de Conjuntos.
3. Los únicos objetos de la Teoría de Conjuntos son los descritos en 1 y 2.

Relación de Pertenencia: El ser elemento de es una relación binaria o de dos argumentos entre dos objetos de la Teoría de Conjuntos. Esta relación va de un objeto a otro, donde el segundo objeto es necesariamente un conjunto y el primero puede ser o no un conjunto.

Se dice que A está contenido en B (también que A es un **subconjunto** de B o que A es una parte de B), y se denota $A \subseteq B$, si todo elemento de A lo es también de B, es decir, $\forall x (x \in A \Rightarrow x \in B)$.

Intersección:

Se dice de dos conjuntos A y B al conjunto formado por objetos que son elementos de A y de B, es decir: $A \cap B := \{x \mid x \in A \wedge x \in B\}$.

Para explicar la intersección se utiliza la lógica matemática que dice lo siguiente:

La lógica matemática es la disciplina que trata de métodos de razonamiento. En un nivel elemental, la lógica proporciona reglas y técnicas para determinar si es o no válido un argumento dado. El razonamiento lógico se emplea en matemáticas para demostrar teoremas; ciertamente se usa en forma constante el razonamiento lógico para realizar cualquier actividad.

Una proposición o enunciado es una oración que puede ser falsa o verdadera pero no ambas a la vez. La proposición es un elemento fundamental de la lógica matemática.

Para el proyecto una proposición será descrita como una palabra que es buscada. Por ejemplo: "calculo integral"

- Proposición uno (P): calculo
- Proposición dos (Q): integral

La tabla de verdad de intersección para las dos proposición es la siguiente:

Tabla 44. Tabla de verdad para intersección.

P	Q	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0

Unión

Se dice **unión** de dos conjuntos A y B al conjunto formado por objetos que son elementos de A o de B, es decir: $A \cup B := \{x \mid x \in A \vee x \in B\}$.

Utilizando el ejemplo anterior, la tabla de verdad seria la siguiente:

Tabla 45. Tabla de verdad unión.

P	Q	$P \vee Q$
1	1	1
1	0	1
0	1	1
0	0	0

Debido a que en la unión se tienen muchas opciones verdaderas, no se utilizo en el diseño procedimental del sistema, para garantizar que el tiempo de respuesta en las búsquedas sea corto.

12.5 SELECCIÓN DE HARDWARE Y SOFTWARE

La selección del hardware se realizó basada en la infraestructura existente en la División de Servicios de Información de la Universidad Industrial de Santander.

- Servidor SILICON GRAPHICS modelo ORIGIN-2000, servidor de Base de Datos y tiene la siguiente configuración:
 - 4 procesadores RISC de 64 bits MIPS R12000 a 300 Mhz
 - 8,64 MB de memoria cache por procesador.
 - 4,32 GB de memoria RAM.
 - 52 GB en disco.
 - Controlador de dispositivos Ultra SCSI.
 - Interfase de red FastEthernet.
 - 2 unidades de cinta DAT 4 mm 12/24 GB.
- Servidor DELL Intranet DELL Poweredge 600SC.
 - Un procesador pentium 4 de 1,8 Ghz.
 - 1 MB de memoria cache.
 - 2 GB de memoria RAM.
 - 40 GB en disco.
 - Controlador de dispositivos Ultra SCSI.
 - Interfaz de red Intel 10/100.

La selección del software fue realizada principalmente en la experiencia con que contaban los desarrolladores del equipo de trabajo, el tiempo que se necesitaría para codificar todo el sitio Web, y las licencias del software que ya han sido adquiridas por parte de la División de Servicios de Información.

El lenguaje seleccionado fue active server pages (asp) con los scripts en javascript para el rendimiento de cada pagina. El software para desarrollar las paginas es Visual Interdev 6.0. El software para desarrollar las librerías dinámicas (dll) es Visual Basic 6.0. Los procedimientos almacenados se realizaran en el editor de informix.

12.6 CONSTRUCCIÓN DE PROTOTIPOS

12.6.1 Búsqueda simple

La búsqueda simple esta conformada por un cuadro de texto y un botón llamado consultar, el usuario puede digitar solamente como máximo cuatro palabras en el cuadro de texto. Las formas de realizar las consultas permiten que el usuario pueda utilizar los espacios en blanco para varias palabras, las tildes y la letra ñ. Los resultados que arroja el sistema se dan teniendo en cuenta todos los parámetros permitidos de consulta (Autor, Título, Materia, Libro, Tesis, Revistas, Analítica, ISSN o ISBN y año)



Sistema de Catalogo Bibliográfico

UNIVERSIDAD INDUSTRIAL DE SANTANDER

Ingeniería Software Pressman

Consultar

The image shows a simple search interface. At the top, it says 'Sistema de Catalogo Bibliográfico' and 'UNIVERSIDAD INDUSTRIAL DE SANTANDER'. Below this is a search input field containing the text 'Ingeniería Software Pressman' and a button labeled 'Consultar'.

Figura 100. Prototipo búsqueda simple.

12.6.2 Búsqueda Avanzada

La búsqueda avanzada cuenta con cuatro cuadros de texto donde el usuario digita las palabras clave por las que quiere buscar y cuatro listas donde cada una tiene los siguientes criterios de búsqueda (Autor, Título, Materia, Libro, Tesis, Revista, Analítica, ISBN, ISSN, Año). El usuario digita una palabra y al frente en la lista debe especificar cual es el tipo de palabra según los criterios

Sistema de Catalogo Bibliográfico

UNIVERSIDAD INDUSTRIAL DE SANTANDER

Ingeniería

Pressman

Software

Consultar

Título

Autor

Título

Autor

Título

Materia

Libro

Tesis

Revista

Analítica

ISBN

ISSN

Año

Detailed description: The image shows a web form for an advanced search system. At the top, it says 'Sistema de Catalogo Bibliográfico' and 'UNIVERSIDAD INDUSTRIAL DE SANTANDER'. There are four text input fields on the left containing the words 'Ingeniería', 'Pressman', 'Software', and an empty field. To the right of these fields are four dropdown menus. The first dropdown is set to 'Título'. The second is set to 'Autor'. The third is set to 'Título' and is currently open, showing a list of search criteria: 'Autor', 'Título', 'Materia', 'Libro', 'Tesis', 'Revista', 'Analítica', 'ISBN', 'ISSN', and 'Año'. The 'Título' option in this list is highlighted. Below the input fields is a blue 'Consultar' button.

Figura 101. Prototipo búsqueda avanzada.

12.6.3 Listado de Resultados

Este listado contiene los resultados de la consulta que el usuario realiza. Los datos que se muestran como resultado de la consulta, son los datos más generales de los documentos y son:

- El título.
- El tipo de documento y el autor.
- El año de publicación.

La lista general contiene unas casillas de verificación, las cuales permiten seleccionar los registros que el usuario desee guardar. En la parte superior se encuentran los botones de *guardar* (se usan para guardar las referencias seleccionadas), *ver listado* (para ver la lista previa de referencias guardadas), *subconsulta* y *siguientes registros*. Cada registro tiene un enlace para ver los detalles del material.

Sistema Consulta de Catalogo Bibliográfico

UNIVERSIDAD INDUSTRIAL DE SANTANDER

Palabras buscadas: **ingenieria software**

Num	Marcar	Palabras Claves
1	<input type="checkbox"/>	<p>Título : DISEÑO E IMPLEMENTACION DEL SOFTWARE PARA CALCULO ESTRUCTURAL EN INGENIERIA CIVIL / por Saul Prada Duarte e Isnardo Uribe Jaimes Autor : TESIS DE GRADO - PRADA DUARTE, SAUL; Universidad Industrial de Santander. Escuela de Ingenieria Civil. Tesis (pregrado); Universidad Industrial de Santander. Escuela de Ingenieria de Sistemas. Tesis (pregrado); Uribe Jaimes, Isnardo Año : 1985</p>
2	<input type="checkbox"/>	<p>Título : INGENIERIA DE SOFTWARE. Autor : LIBRO - FAIRLEY, RICHARD E. Año : 1987</p>
3	<input type="checkbox"/>	<p>Título : CASE IS SOFTWARE AUTOMATION. Autor : LIBRO - MCCLURE, CARMA L. Año : 1989</p>
4	<input type="checkbox"/>	<p>Título : DATABASE TECHNOLOGY: A SOFTWARE ENGINEERING APPROACH / John G. Hughes Autor : LIBRO - HUGHES, JOHN G. Año : 1988</p>
5	<input type="checkbox"/>	<p>Título : PROYECTO PILOTO EN LA APLICACION DE LA INGENIERIA DE SOFTWARE PARA EL MANEJO DE INFORMACION ESTADISTICA DEL LABORATORIO DE GRANULOMETRIA DEL INSTITUTO COLOMBIANO DEL PETROLEO / Juan Carlos Correa Correa y Guillermo Garcia Nino Autor : TESIS DE GRADO - CORREA CORREA, JUAN CARLOS; Garcia Nino, Guillermo; Universidad Industrial de Santander. Escuela de Ingenieria de Sistemas. Tesis (pregrado) Año : 1991</p>

Figura 102. Prototipo listado de resultados.

12.6.4 Subconsulta

La subconsulta contiene dos cuadros de texto, el primero muestra la información de la consulta realizada anteriormente y no se puede modificar y

el segundo donde el usuario puede introducir los nuevos parámetros de consulta.

Desde el listado general se puede realizar la subconsulta, cuya finalidad es filtrar la información para obtener resultados más precisos, por ejemplo: La primera consulta, la puede realizar para obtener un listado con todos los libros de un autor, y la subconsulta la realiza para obtener de ese listado sólo las publicaciones de un cierto año. Las formas de realizar las subconsultas utilizan las mismas limitaciones de la búsqueda simple.

Sistema de Catalogo Bibliográfico
UNIVERSIDAD INDUSTRIAL DE SANTANDER

Ingeniería software

Pressman

Consultar

Detailed description: The image shows a web form for a bibliographic catalog system. At the top, it reads 'Sistema de Catalogo Bibliográfico' and 'UNIVERSIDAD INDUSTRIAL DE SANTANDER'. Below this, there are two input fields. The first field contains the text 'Ingeniería software' and the second field contains 'Pressman'. To the right of these fields is a button labeled 'Consultar'.

Figura 103. Prototipo subconsulta.

12.6.5 Ver Detalles

Ver detalles presenta tres secciones de información a mostrar, la primera sección *Información del Documento*, muestra los detalles más generales del documento, como: Tipo de documento, Título (principal), Autor (principal), Edición, Área de publicación, y descripción física. La segunda sección *Información de los ejemplares*, muestra los ejemplares que existen de ese documento con la información de: Número del ejemplar, Número de inventario, ubicación, estado, fecha de devolución (si el estado es prestado) y

descripción. Y la tercera sección *Otra información* presenta todos los títulos, todos los autores y todas las materias que pertenece ese material.

Desde ver detalles también puede ir a ver resumen, ver detalles tipo MARC y guardar referencia.

Sistema Consulta de Catalogo Bibliográfico
UNIVERSIDAD INDUSTRIAL DE SANTANDER

Información del documento

Tipo de documento	LIBRO	Número de Clasificación	005.1 / P935I
Título	INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN.		
Autor(es)	PRESSMAN, ROGER S.		
Edición			
Area Publicación	MADRID : McGRAW-HILL, c1988.		
Descripción Física	628p. : il., diagrs., tablas		

Información de los ejemplares:

Ejem.	Num. Inv.	Ubicación	Estado	Fecha (A/M/D)	Descripción
1	79441	GENERAL - PISO 3	DISPONIBLE		

Otra información:

Títulos	INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN.
Autores	PRESSMAN, ROGER S.
Materias	INGENIERIA DEL SOFTWARE

Figura 104. Prototipo ver detalles.

12.6.6 Lista previa de referencias

La lista previa de referencias se hace para que el usuario observe la lista de referencias que ha venido guardando durante la sesión de consulta.

Muestra un índice, un cuadro para marcar y la información más general del material como lo es el título, autor, año, ubicación, número de clasificación y el estado.

El usuario cuenta con la posibilidad de actualizar la lista cuantas veces quiera, seleccionando el cuadro de la referencia que desea borrar y luego oprimiendo el botón de borrar referencias seleccionadas, o puede regresar a la página anterior, continuar navegando y grabando más referencias.

También tiene la posibilidad de enviar la lista a una cuenta de correo electrónico o a un pantallazo para imprimir en una impresora.

La página valida que la dirección escrita sea correcta y lo envía a una nueva ventana de confirmación. Estas dos actividades no se pueden realizar simultáneamente, aunque si se pueden utilizar cuantas veces el usuario lo necesite.

Sistema de Catalogo Bibliografico
UNIVERSIDAD INDUSTRIAL DE SANTANDER

1	<input type="checkbox"/>	Título : INGENIERIA DE SOFTWARE. Autor : LIBRO - FAIRLEY, RICHARD E. Año : 1987 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / F172I Estado: DISPONIBLE
2	<input type="checkbox"/>	Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN. Autor : LIBRO - PRESSMAN, ROGER S. Año : 1988 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / P935I Estado: DISPONIBLE

Enviar por e-mail

Imprimir en Pantanlla

Figura 105. Prototipo lista previa de referencias.

12.6.7 Enviar referencias por correo electrónico

El usuario tiene la posibilidad de enviar la lista de referencias a una cuenta de correo electrónico, seleccionado la opción “Enviar por e-mail” y digitando una dirección de válida.

Sistema de Catalogo Bibliografico

UNIVERSIDAD INDUSTRIAL DE SANTANDER

1	<input type="checkbox"/>	Título : INGENIERIA DE SOFTWARE Autor : LIBRO - FAIRLEY, RICHARD E. Año : 1987 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / F1721 Estado: DISPONIBLE
2	<input type="checkbox"/>	Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN Autor : LIBRO - PRESSMAN, ROGER S. Año : 1988 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / P935I Estado: DISPONIBLE

Enviar por e-mail
 Imprimir en Pantanilla

Figura 106. Prototipo enviar referencia por e-mail.

Después del envío se muestra una ventana de confirmación, y se da la opción enviar a otras direcciones de correo electrónico o regresar al listado.

Sistema de Catalogo Bibliografico

UNIVERSIDAD INDUSTRIAL DE SANTANDER

Lista de referencias enviadas a el mail **Sergio.Contreras@ou.edu**

1	Título : INGENIERIA DE SOFTWARE Autor : LIBRO - FAIRLEY, RICHARD E. Año : 1987 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / F1721 Estado: DISPONIBLE
2	Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN Autor : LIBRO - PRESSMAN, ROGER S. Año : 1988 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / P935I Estado: DISPONIBLE

Enviar a otras cuentas de correo

e-mail 1:

e-mail 2:

e-mail 3:

Figura 107. Prototipo confirmación envío de e-mail.

12.6.8 Imprimir listado de referencias

La otra opción que tiene el usuario con el listado de referencias es poderlas imprimir. El usuario debe seleccionar “Imprimir en pantalla” y oprimir el botón enviar.

Sistema de Catalogo Bibliografico

UNIVERSIDAD INDUSTRIAL DE SANTANDER

1	<input type="checkbox"/>	Título : INGENIERIA DE SOFTWARE Autor : LIBRO - FAIRLEY, RICHARD E. Año : 1987 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / F1721 Estado: DISPONIBLE
2	<input type="checkbox"/>	Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN Autor : LIBRO - PRESSMAN, ROGER S. Año : 1988 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / P935I Estado: DISPONIBLE

Enviar por e-mail
 Imprimir en Pantalla

Figura 108. Prototipo imprimir referencias.

El sistema muestra el listado en la forma como se imprimirá y el usuario sólo tiene que oprimir el botón imprimir.

Sistema de Catalogo Bibliografico

UNIVERSIDAD INDUSTRIAL DE SANTANDER

Lista de referencias

1	Título : INGENIERIA DE SOFTWARE. Autor : LIBRO - FAIRLEY, RICHARD E. Año : 1987 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / F1721 Estado: DISPONIBLE
2	Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN. Autor : LIBRO - PRESSMAN, ROGER S. Año : 1988 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / P935I Estado: DISPONIBLE

Figura 109. Prototipo ventana de impresión.

12.6.9 Detalles MARC

Detalles MARC presenta una vista de la información de la descripción general del material en formato MARC. El formato MARC es un estándar para la representación y la comunicación de la información bibliográfica. En la parte superior se encuentra el botón *volver al registro*, que lo lleva directamente a la vista normal de los detalles generales.

Sistema Consulta de Catalogo Bibliográfico

UNIVERSIDAD INDUSTRIAL DE SANTANDER

[Volver al registro](#)

```

001 x533714477
008 991102s1999 sp 001 esp nam a
017 M903-1998
020 84-7615-222-1
040 SpMaUCEI
080 005.1
100 1 Pressman, Roger S.
245 10 Ingeniería del Software :|bun enfoque práctico /|cRoger S.
Pressman
250 4a. ed.
260 Madrid [etc.] :|bMcGraw-Hill,|ccop. 1998
300 581 p. ;|c24 cm.
504 Indices
650 04 Ingeniería del software
907 00 1t|b0|c991102|dm|ew|ga|zA00124

```

Ejem.	Num. Inv.	Ubicación	Estado	Fecha (A/M/D)	Descripción
1	79441	GENERAL - PISO 3	DISPONIBLE		

Figura 110. Prototipo detalles MARC.

12.6.10 Resumen

Contiene el título y los autores del documento de trabajo de grado con su respectivo resumen.

RESUMEN

IMPLANTACION DEL SISTEMA CORPORATIVO DE INFORMACION Y SERVICIOS INTERNET Y DESARROLLO DE UN MODULO PARA LOS PROCESOS

BUITRAGO BARRETO, DIANA LUZ - Villamizar Sanclemente, Olga

Acerca de...

PALABRAS CLAVES Sitio Web, Sistema de Informaci3n Web, Aplicaciones Distribuidas en Internet, Aplicaci3nes Cliente-Servidor, JavaServer Pages.

DESCRIPCION

A partir del segundo semestre de 1997, la Universidad Industrial de Santander (UIS), comenz3 a implantar los primeros prototipos del Sitio Web de la Universidad para ponerlos a disposici3n de su comunidad y del mundo entero. Progresivamente el sitio se ha ido enriqueciendo en informaci3n y se han utilizado nuevas tecnologfas y herramientas de programaci3n orientadas al Web, logrando asf que los visitantes encuentren un sitio completo y f3cil de navegar. Para generar un nuevo prototipo del Sistema de Informaci3n Web de la UIS y llevar a cabo su implantaci3n se evalu3 la estructura de informaci3n existente en prototipos anteriores y se le dio un nuevo enfoque en cuanto a la arquitectura de la informaci3n y dise1o gr3fico para hacerlo m3s funcional prestando servicios m3s eficientes que se adaptaran a las necesidades de los usuarios.

La nueva versi3n del Sistema de Informaci3n Web de la UIS utiliza bases de datos para la presentaci3n de informaci3n din3mica, permitiendo la actualizaci3n de informaci3n a trav3s del mismo sistema, a su vez proporciona espacios para la publicaci3n de informaci3n de cada dependencia tanto administrativa como acad3mica y presta los servicios de admisi3n y prematr3cula a trav3s de Web, los cuales se basan en el funcionamiento actual de los procedimientos de inscripci3n y matr3cula para estudiantes de pregrado en las 3reas tecnol3gica y humanfstica. El proceso de prematr3cula a trav3s del Web se integra a un m3dulo personalizado de consulta de informaci3n para estudiantes orientado al manejo de

Figura 111. Prototipo resumen.

12.7 DISE1AR EL PLAN DE PRUEBAS

Tabla 46. Pruebas de Caja Blanca.

Registro de Pruebas									
Sistema		Catálogo Bibliográfico Libruis Web			Probado por				
Modulo		Consulta			Fecha				
Tipo de Prueba		Caja Blanca			07-06-04				
Prueba	Nombre	Que se prueba	Resultado esperado	Resultado Actual	Corrección	Resultados de corrección	RESP.	Fecha	Corrección nuevamente
1	Busqued a Simple	Los resultados de la búsqueda	Que los resultados no tengan relación con lo escrito en cuadro de búsqueda.						
2	Busqued a Avanzada	Los resultados de la búsqueda	Que el resultado no sea coherente con el criterio específico que el usuario selecciono por palabra.						
3	Ver Detalles (Información)	La información general del material seleccionado.	Que la información general, la información del ejemplar, otros autores, materias, y títulos no sea la del material seleccionado .						

Tabla 46. Pruebas de Caja Blanca (Continuación).

Registro de Pruebas									
Sistema	Catálogo Bibliográfico Libruis Web			Probado por					
Modulo	Consulta			Fecha	07-06-04				
Tipo de Prueba	Caja Blanca			Corrección					
Prueba	Nombre	Que se prueba	Resultado esperado	Resultado Actual	Corrección	Resultados de corrección	RESP.	Fecha	Corrección nuevamente
4	Ver Detalles (navegación)	La navegabilidad por los resultados.	autores, títulos, materias, no tengan un enlace que muestre un listado de resultados por el criterio seleccionado.						
5	Ver Resumen	El resumen de la tesis seleccionada.	Que no muestre el resumen de la tesis que se selecciono en un cuadro de texto.						
6	Ver Detalles MARC	Que los detalles estilo MARC estén completos.	Que no muestre los detalles estilo MARC del material seleccionado.						

Tabla 46. Pruebas de Caja Blanca (Continuación).

Registro de Pruebas									
Sistema	Catálogo Bibliográfico Libruis Web			Probado por					
Modulo	Consulta			Fecha	07-06-04				
Tipo de Prueba	Caja Blanca			Corrección					
Prueba	Nombre	Que se prueba	Resultado esperado	Resultado Actual	Corrección	Resultados de corrección	RESP.	Fecha	Corrección nuevamente
7	Guardar referencias	El listado de referencias.	Que el listado de referencias que el usuario vaya seleccionando durante la navegación no sea el que es.						
8	Imprimir referencias	La impresión del listado.	Que la impresión del listado de referencias no se el realizado por el usuario.						
9	Enviar referencias por e-mail	El envío por e-mail del listado de referencias.	Que el envío no sea satisfactorio del listado vía mail, en formato texto.						

Tabla 47. Pruebas de Caja Negra.

Registro de Pruebas									
Sistema	Catálogo Bibliográfico Libruis Web				Probado por				
Modulo	Consulta				Fecha	07-06-04			
Tipo de Prueba	Caja Negra								
Prueba	Nombre	Que se prueba	Resultado esperado	Resultado Actual	Corrección	Resultados de corrección	RESP.	Fecha	Corrección nuevamente
1	Palabras ingresadas	El ingreso de más de cuatro palabras en la búsqueda simple.	Que el sistema no muestre ningún error de ingreso máximo de palabras superado.						
2	Sintaxis tipo uno	El ingreso de varias palabras separadas por espacios.	Que el sistema no muestre ningún resultado a la búsqueda.						
3	Sintaxis tipo dos	El ingreso de palabras con tildes.	Que el sistema no muestre ningún resultado a la búsqueda.						

Tabla 48. Pruebas de entornos y aplicaciones especializadas.

Registro de Pruebas									
Sistema		Catálogo Bibliográfico Libruis Web			Probado por				
Modulo		Consulta			Fecha		07-06-04		
Tipo de Prueba		Prueba de entornos y aplicaciones especializadas.							
Prueba	Nombre	Que se prueba	Resultado esperado	Resultado Actual	Corrección	Resultados de corrección	RESP.	Fecha	Corrección nuevamente
1	Búsqueda avanzada (uno)	El listado de criterios de búsqueda avanzada.	Que no este completo.						
2	Búsqueda avanzada (dos)	Que no se pueda seleccionar el mismo criterio de búsqueda en las cuatro opciones.	Que el sistema muestre un error de búsqueda.						
3	Listado de resultados (uno)	La información que muestra al usuario.	Que la información no este completa.						

Tabla 48. Pruebas de entornos y aplicaciones especializadas (Continuación)

Registro de Pruebas									
Sistema		Catálogo Bibliográfico Libruis Web			Probado por				
Modulo		Consulta			Fecha		07-06-04		
Tipo de Prueba		Prueba de entornos y aplicaciones especializadas.							
Prueba	Nombre	Que se prueba	Resultado esperado	Resultado Actual	Corrección	Resultados de corrección	RESP.	Fecha	Corrección nuevamente
4	Listado de resultados (dos)	Botón regresar.	Si el listado fue obtenido de una búsqueda simple, cuando se oprima el botón regresar vaya a la búsqueda avanzada.						
5	Listado de resultados (tres)	Botón regresar.	Si el listado fue obtenido de una búsqueda avanzada, cuando se oprima el botón regresar vaya a la búsqueda simple.						
6	Subconsulta	Los criterios de búsqueda.	Que no muestre los criterios anteriores de la búsqueda realizada anteriormente.						

Tabla 48. Pruebas de entornos y aplicaciones especializadas (Continuación)

Registro de Pruebas									
Sistema		Catálogo Bibliográfico Libruvis Web			Probado por				
Modulo		Consulta			Fecha		07-06-04		
Tipo de Prueba		Prueba de entornos y aplicaciones especializadas.							
Prueba	Nombre	Que se prueba	Resultado esperado	Resultado Actual	Corrección	Resultados de corrección	RESP.	Fecha	Corrección nuevamente
7	Enviar referencias	La selección sólo de una forma de envío (e-mail o pantalla)	Que el sistema permita seleccionar ambas opciones simultáneamente.						
8	Guardar referencias (uno)	El botón regresar.	Que si se guardo una referencia desde el listado y se oprime regresar vaya al ítem.						
9	Guardar referencias (dos)	El botón regresar.	Que si se guardo una referencia desde el detalle del material y se oprime regresar vaya al listado.						

12.8 VERIFICAR QUE LOS PROTOTIPOS CUMPLAN CON LOS REQUERIMIENTOS

Se comprobó que cada caso de uso se satisface con los prototipos presentados.

12.9 PRESENTAR AL COMITÉ PARA APROBAR

- Justificación y desarrollo de los pasos de la metodología de desarrollo de software seleccionada.
- Formato de documentación seleccionado.
- Cuadro de seguridad.
- Especificación del diseño.
- Integración del sistema (si aplica).
- Selección del hardware y software.
- Prototipos.
- Plan de pruebas.

13. FASE DESARROLLAR

13.1 ORGANIZACIÓN DEL CODIGO FUENTE

El sitio Web sólo tiene un nivel, aunque las imágenes se ubican en una carpeta única.

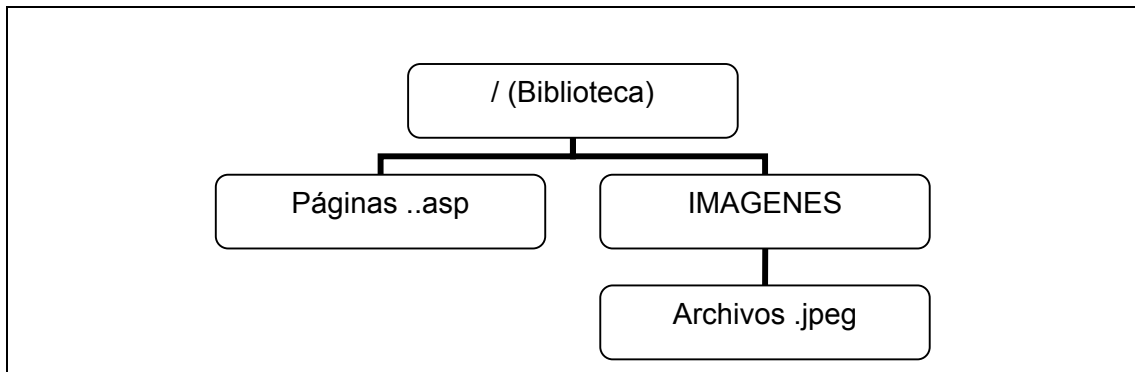


Figura 112. Árbol de organización del código fuente.

La librería también tiene un sólo nivel

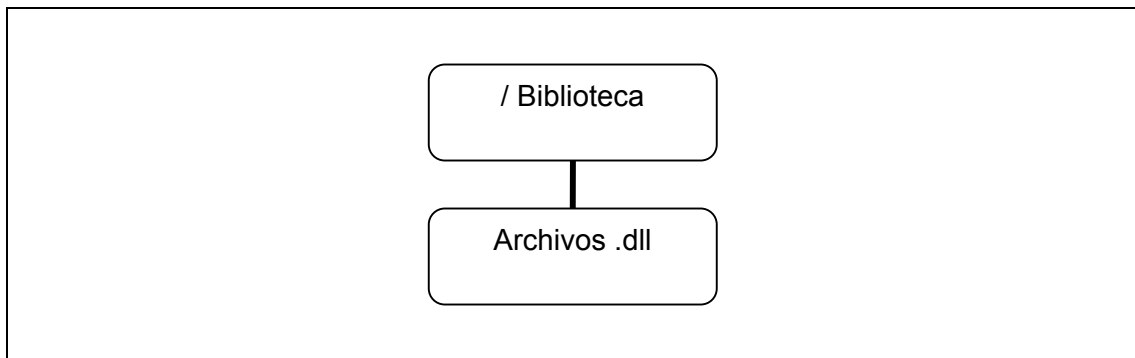


Figura 113. Árbol de organización de las librerías.

13.2 CONSTRUCCIÓN

13.2.1 Búsqueda simple

La búsqueda simple está conformada por un cuadro de texto y un botón llamado consultar, el usuario puede digitar como máximo solo cuatro palabras en el cuadro.

Las formas de realizar las consultas permiten que el usuario pueda utilizar los espacios en blanco para varias palabras, las tildes y la letra ñ.



Figura 114. Construcción, Búsqueda simple.

Tiene en la parte superior de la página la barra estándar que se tiene el sitio Web de la universidad, y tiene el acceso a la Intranet de la Universidad. También tiene dos botones regresar y ayuda. Regresar lo envía a la página anterior desde donde hizo el acceso si el usuario ingreso directamente al

catálogo bibliográfico no regresará a ningún lado, el botón de ayuda lleva al usuario la ayuda que tiene el sistema.

13.2.2 Búsqueda Avanzada

La búsqueda avanzada cuenta con cuatro cuadros de texto donde el usuario digita las palabras clave por las que quiere buscar y cuatro listas donde cada lista tiene los siguientes criterios de búsqueda (Autor, Título, Materia, Libro, Tesis, Revista, Analítica, ISBN, ISSN, Año). El usuario digita una palabra y al frente en la lista debe especificar que es la palabra según los criterios.

Palabras a consultar:	
Ingeniería	Título
Pressman	Autor
Software	Título
	Autor
	Título
	Materia
	Libro
	Tesis
	Revista
	Analítica
	Número de inv.
	ISBN o ISSN
	Año

Figura 115. Construcción, Búsqueda avanzada.

Tiene en la parte superior de la página la barra estándar que se tiene el sitio Web de la universidad, y tiene el acceso a la Intranet de la Universidad. También tiene dos botones regresar y ayuda. Regresar lo envía a la página anterior desde donde hizo el acceso si el usuario ingresó directamente al

catálogo bibliográfico no regresará a ningún lado, el botón de ayuda lleva a el usuario la ayuda que tiene el sistema.

13.2.3 Listado de Resultados

Este listado contiene los resultados de la consulta que el usuario realiza. Los datos que se muestran como resultado de la consulta, son los datos mas generales de los documentos y son:

- El título.
- El tipo de documento y el autor.
- El año de publicación.

La lista general contiene unas casillas de verificación en cada registro, las cuales permiten seleccionar los registros que el usuario desee guardar. En la parte superior se encuentran los botones de *guardar* (se usan para guardar las referencias seleccionadas), *ver listado* (para ver la lista previa de referencias guardadas), *subconsulta* y *siguientes registros*. Cada registro tiene un enlace para ver los detalles del material.

UNIVERSIDAD INDUSTRIAL DE SANTANDER

LIBRUIS Biblioteca

Regresar Ayuda

UNIVERSIDAD INDUSTRIAL DE SANTANDER

Palabras buscadas: **ingenieria software**

Subconsulta Guardar Ver listado

Siguientes Registros

Num	Marcar	Palabras Claves
1	<input type="checkbox"/>	<p>Título : <u>DISEÑO E IMPLEMENTACION DEL SOFTWARE PARA CALCULO ESTRUCTURAL EN INGENIERIA CIVIL / por Saul Prada Duarte e Isnardo Uribe Jaimes</u></p> <p>Autor : <u>TESIS DE GRADO - PRADA DUARTE, SAUL; Universidad Industrial de Santander. Escuela de Ingenieria Civil. Tesis (pregrado); Universidad Industrial de Santander. Escuela de Ingenieria de Sistemas. Tesis (pregrado); Uribe Jaimes, Isnardo</u></p> <p>Año : 1985</p>
2	<input type="checkbox"/>	<p>Título : <u>INGENIERIA DE SOFTWARE.</u></p> <p>Autor : <u>LIBRO - FAIRLEY, RICHARD E.</u></p> <p>Año : 1987</p>
3	<input type="checkbox"/>	<p>Título : <u>CASE IS SOFTWARE AUTOMATION.</u></p> <p>Autor : <u>LIBRO - MCCLURE, CARMA L.</u></p> <p>Año : 1989</p>
4	<input type="checkbox"/>	<p>Título : <u>DATABASE TECHNOLOGY: A SOFTWARE ENGINEERING APPROACH / John G. Hughes</u></p> <p>Autor : <u>LIBRO - HUGHES, JOHN G.</u></p> <p>Año : 1988</p>

Figura 116. Construcción, Listado de resultados.

13.2.4 Subconsulta

La subconsulta contiene dos cuadros de texto, el primero muestra la información de la consulta realizada anteriormente y no se puede modificar y el segundo donde el usuario puede introducir los siguientes parámetros de consulta.

Desde el listado general se puede realizar la subconsulta, cuya finalidad es filtrar la información para obtener resultados más precisos, por ejemplo: La primera consulta, la puede realizar para obtener un listado con todos los libros de un autor, y la subconsulta la realiza para obtener de ese listado solo las publicaciones de un cierto año. Las formas de realizar las subconsultas utilizan los mismos parámetros de la consulta.



Figura 117. Construcción, Subconsulta.

13.2.5 Ver Detalles

Ver detalles presenta tres secciones de información a mostrar, la primera sección *Información del Documento*, muestra los detalles más generales del documento, como: Tipo de documento, Título (principal), Autor (principal), Edición, Área de publicación, y descripción física. La segunda sección *Información de los ejemplares*, muestra los ejemplares que existen de ese documento con la información de: Número del ejemplar, Número de inventario, ubicación, estado, fecha de devolución (si el estado es prestado) y descripción. Y la tercera sección *Otra información* presenta todos los títulos, todos los autores y todas las materias a las que hace parte ese material.

Desde ver detalles también puede ir a ver resumen, ver detalles tipo MARC y guardar referencia.

The screenshot displays the LIBRUIS library system interface for the Universidad Industrial de Santander. At the top, there are logos for the university and the library, along with navigation buttons for 'MARC', 'Resumen', and 'Guardar'. The main content is divided into two sections: 'Información del documento' and 'Información de los ejemplares'.

Información del documento

Tipo de documento	LIBRO	Número de Clasificación	005.1 / P9351
Título	INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN.		
Autor(es)	PRESSMAN, ROGER S.		
Edición			
Area Publicación	MADRID : McGRAW-HILL, c1988.		
Descripción Física	628p. : il., diagrs., tablas		

Información de los ejemplares:

Ejem.	Num. Inv.	Ubicación	Estado	Fecha (A/M/D)	Descripción
1	79441	GENERAL -PISO 3	DISPONIBLE		

Otra información:

Títulos	INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN.
Autores	PRESSMAN, ROGER S.
Materias	INGENIERIA DEL SOFTWARE

Figura 118. Construcción, Ver detalles.

13.2.6 Lista previa de referencias

La lista previa de referencias se hace para que el usuario observe la lista de referencias que ha venido guardando durante la sesión de consulta.

Muestra un índice, un cuadro para marcar y la información más general del material como lo es el título, autor, año, ubicación, número de clasificación y el estado.

UNIVERSIDAD INDUSTRIAL DE SANTANDER

LIBRUIS Biblioteca

Regresar Ayuda

Listado de Referencias

NUM	MARCAR	DESCRIPCIÓN
1	<input type="checkbox"/>	<p>Título : INGENIERIA DE SOFTWARE Autor : LIBRO - FAIRLEY, RICHARD E. Año : 1987 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / F172I Estado: DISPONIBLE</p>
2	<input type="checkbox"/>	<p>Título : DATABASE TECHNOLOGY: A SOFTWARE ENGINEERING APPROACH / John G. Hughes Autor : LIBRO - HUGHES, JOHN G. Año : 1988 Ubicación: GENERAL - PISO 3 Número de Clasificación: 005.74/H893d Estado: DISPONIBLE</p>
3	<input type="checkbox"/>	<p>Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN Autor : LIBRO - PRESSMAN, ROGER S. Año : 1988 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / P935I Estado: DISPONIBLE</p>

Borrar referencias marcadas

Enviar por e-mail
 Imprimir en Pantalla

Enviar

Figura 119. Construcción, Listado de referencias.

El usuario cuenta con la posibilidad de actualizar la lista cuantas veces quiera, seleccionando el cuadro de la referencia que desea borrar y luego oprimiendo el botón de *borrar referencias seleccionadas*, o puede regresar a la página anterior, continuar navegando y grabando más referencias.

UNIVERSIDAD INDUSTRIAL DE SANTANDER

LIBRUIS Biblioteca

Regresar Ayuda

Listado de Referencias

NUM	MARCAR	DESCRIPCIÓN
1	<input type="checkbox"/>	Título : INGENIERIA DE SOFTWARE. Autor : LIBRO - FAIRLEY, RICHARD E. Año : 1987 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / F172I Estado: DISPONIBLE
2	<input checked="" type="checkbox"/>	Título : DATABASE TECHNOLOGY: A SOFTWARE ENGINEERING APPROACH / John G. Hughes Autor : LIBRO - HUGHES, JOHN G. Año : 1988 Ubicación: GENERAL - PISO 3 Número de Clasificación: 005.74/H893d Estado: DISPONIBLE
3	<input type="checkbox"/>	Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN. Autor : LIBRO - PRESSMAN, ROGER S. Año : 1988 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / P935I Estado: DISPONIBLE

Borrar referencias marcadas

Enviar por e-mail
 Imprimir en Pantanlla

Enviar

Figura 120. Construcción, Modificación de lista previa de referencias.

UNIVERSIDAD INDUSTRIAL DE SANTANDER

LIBRUIS Biblioteca

Regresar Ayuda

Listado de Referencias

Num	Marcar	Descripción
1	<input checked="" type="checkbox"/>	Título : INGENIERIA DE SOFTWARE. Autor : LIBRO - FAIRLEY, RICHARD E. Año : 1987 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / F172I Estado: DISPONIBLE
2	<input type="checkbox"/>	Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN. Autor : LIBRO - PRESSMAN, ROGER S. Año : 1988 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / P935I Estado: DISPONIBLE

Borrar referencias marcadas

Enviar por e-mail
 Imprimir en Pantanlla

Enviar

Figura 121. Construcción, Lista modificada de referencias.

También tiene las posibilidades de enviar la lista a una cuenta de correo electrónico o de visualizar en pantalla para imprimir en una impresora.

La página valida que la dirección escrita sea correcta y lo envía a una ventana de confirmación. Estas dos actividades no se pueden realizar simultáneamente, aunque si se pueden utilizar cuantas veces el usuario lo necesite.

13.2.7 Enviar referencias por correo electrónico

El usuario tiene la posibilidad de enviar la lista de referencias a una cuenta de correo electrónico, seleccionado la opción “Enviar por e-mail” y digitando una dirección válida.

The screenshot shows the 'Listado de Referencias' page in the LIBRUIS system. At the top, there is a header with the Universidad Industrial de Santander logo and the LIBRUIS logo. Below the header, there are buttons for 'Regresar' and 'Ayuda'. The main content area displays a table of references with columns for 'NUM', 'MARCAR', and 'DESCRIPCIÓN'. Two references are listed, both with checkboxes in the 'MARCAR' column. Below the table, there is a button labeled 'Borrar referencias marcadas'. At the bottom, there are two radio button options: 'Enviar por e-mail' (selected) and 'Imprimir en Pantalla'. A text input field contains the email address 'Sergio.Contreras@ou.edu', and an 'Enviar' button is positioned below it.

NUM	MARCAR	DESCRIPCIÓN
1	<input type="checkbox"/>	Título : INGENIERIA DE SOFTWARE Autor : <i>LIBRO</i> - FAIRLEY, RICHARD E. Año : 1987 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / F172I Estado: DISPONIBLE
2	<input type="checkbox"/>	Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN Autor : <i>LIBRO</i> - PRESSMAN, ROGER S. Año : 1988 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / P935I Estado: DISPONIBLE

Figura 122. Construcción, Envío de referencias por e-mail.

Después del envío se muestra una ventana de confirmación, y se da la opción enviar a otras direcciones de correo electrónico o regresar al listado.

UNIVERSIDAD INDUSTRIAL DE SANTANDER

LIBRUIS Biblioteca

Régresar Ayuda

Referencias enviadas

LISTA DE REFERENCIAS ENVIAS A EL MAIL: SERGIO.CONTRERAS@OU.EDU

1	Título : INGENIERIA DE SOFTWARE.	Autor : LIBRO - FAIRLEY, RICHARD E.	Año : 1987 Ubicación: RESERVA CENTRAL 3	Número de Clasificación: 005.1 / F172I	Estado: DISPONIBLE
2	Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN.	Autor : LIBRO - PRESSMAN, ROGER S.	Año : 1988 Ubicación: RESERVA CENTRAL 3	Número de Clasificación: 005.1 / P935I	Estado: DISPONIBLE

ENVIAR A OTRAS CUENTAS DE CORREO

e-mail 1:

e-mail 2:

e-mail 3:

Figura 123. Construcción, Confirmación envío de referencias por e-mail.

13.2.8 Imprimir listado de referencias

La otra opción que tiene el usuario con el listado de referencias es poderlas imprimir en papel. El usuario debe seleccionar “Imprimir en pantalla” y oprimir el botón enviar.

UNIVERSIDAD INDUSTRIAL DE SANTANDER

LIBRUIS Biblioteca

Régresar Ayuda

Listado de Referencias

NUM	MARCAR	DESCRIPCIÓN
1	<input type="checkbox"/>	<p>Título : INGENIERIA DE SOFTWARE. Autor : LIBRO - FAIRLEY, RICHARD E. Año : 1987 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / F172I Estado: DISPONIBLE</p>
2	<input type="checkbox"/>	<p>Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN. Autor : LIBRO - PRESSMAN, ROGER S. Año : 1988 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / P935I Estado: DISPONIBLE</p>

[Borrar referencias marcadas](#)

Enviar por e-mail
 Imprimir en Pantalla

[Enviar](#)

Figura 124. Construcción, Imprimir referencias.

El sistema muestra el listado en la forma como se imprimirá, el usuario sólo tiene que oprimir el botón imprimir.

UNIVERSIDAD INDUSTRIAL DE SANTANDER

LIBRUIS Biblioteca

Régresar Ayuda

Listado de Referencias

1		<p>Título : INGENIERIA DE SOFTWARE. Autor : LIBRO - FAIRLEY, RICHARD E. Año : 1987 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / F172I Estado: DISPONIBLE</p>
2		<p>Título : INGENIERIA DEL SOFTWARE : UN ENFOQUE PRACTICO / ROGER S. PRESSMAN. Autor : LIBRO - PRESSMAN, ROGER S. Año : 1988 Ubicación: RESERVA CENTRAL 3 Número de Clasificación: 005.1 / P935I Estado: DISPONIBLE</p>

[Imprimir](#)

Figura 125. Construcción, Ventana de impresión.

13.2.9 Detalles MARC

Detalles MARC presenta una vista de la información de la descripción general del material en formato MARC. El formato MARC es un estándar para la representación y la comunicación de la información bibliográfica. En la parte superior se encuentra el botón *volver al registro*, que lo lleva directamente a la vista normal de los detalles generales.

UNIVERSIDAD INDUSTRIAL DE SANTANDER

LIBRUIS Biblioteca

Regresar Ayuda

UNIVERSIDAD INDUSTRIAL DE SANTANDER

```
001 x533714477
008 991102s1999 sp 001 esp nam a
017 M903-1998
020 84-7615-222-1
040 SpMaUCEI
080 005.1
100 1 Pressman, Roger S.
245 10 Ingenieria del Software :|bun enfoque práctico /|cRoger S.
Pressman
250 4a. ed.
260 Madrid [etc.] :|bMcGraw-Hill,|ccop. 1998
300 581 p. ;|c24 cm.
504 Indices
650 04 Ingenieria del software
907 00 lt|b0|c991102|dm|ew|ga|zA00124
```

Ejem.	Num. Inv.	Ubicación	Estado	Fecha (A/M/D)	Descripción
1	79441	GENERAL - PISO 3	DISPONIBLE		

Figura 126. Construcción, Detalles MARC.

13.2.10 Resumen

Contiene el título y los autores del documento de trabajo de grado con su respectivo resumen.

RESUMEN

IMPLANTACION DEL SISTEMA CORPORATIVO DE INFORMACION Y SERVICIOS INTERNET Y DESARROLLO DE UN MODULO PARA LOS PROCESOS

BUITRAGO BARRETO, DIANA LUZ - Villamizar Sanclemente, Olga

Acerca de...

PALABRAS CLAVES Sitio Web, Sistema de Informaci=n Web, Aplicaciones Distribuidas en Internet, Aplicaci=nes Cliente-Servidor, JavaServerPages.

DESCRIPCION

A partir del segundo semestre de 1997, la Universidad Industrial de Santander (UIS), comenz= a implantar los primeros prototipos del Sitio Web de la Universidad para ponerlos a disposici=n de su comunidad y del mundo entero. Progresivamente el sitio se ha ido enriqueciendo en informaci=n y se han utilizado nuevas tecnologfas y herramientas de programaci=n orientadas al Web, logrando asf que los visitantes encuentren un sitio completo y ffcil de navegar. Para generar un nuevo prototipo del Sistema de Informaci=n Web de la UIS y llevar a cabo su implantaci=n se evalu= la estructura de informaci=n existente en prototipos anteriores y se le dio un nuevo enfoque en cuanto a la arquitectura de la informaci=n y dise±o grfcico para hacerlo mfs funcional prestando servicios mfs eficientes que se adaptaran a las necesidades de los usuarios.

La nueva versi=n del Sistema de Informaci=n Web de la UIS utiliza bases de datos para la presentaci=n de informaci=n dinfmica, permitiendo la actualizaci=n de informaci=n a travTs del mismo sistema, a su vez proporciona espacios para la publicaci=n de informaci=n de cada dependencia tanto administrativa como acadfmica y presta los servicios de admisi=n y prematrfcula a travTs de Web, los cuales se basan en el funcionamiento actual de los procedimientos de inscripci=n y matrfcula para estudiantes de pregrado en las reas

tecnol=gica y humanfstica. El proceso de prematrfcula a travTs del Web se integra a un m=duo personalizado de consulta de informaci=n para estudiantes orientado al manejo de

Figura 127. Construcci=n, Resumen.

14. CONCLUSIONES

- El modelo DMADDV de seis sigma, planteado para la División de Servicios de Información indica que los mayores esfuerzos en cualquier proyecto no consisten en encontrar la mejor solución, si no, en encontrar el problema indicado.
- Según lo anterior se debe romper el paradigma que actualmente los ingenieros tienen de invertir la mayor parte de los esfuerzos en encontrar la mejor solución al inicio del proyecto, en lugar de invertir la mayor parte de los esfuerzos en encontrar el problema correcto.
- Cambiar la forma de plantear los objetivos específicos de una frase con solo palabras, a una de manera medible utilizando números, ofrece a la organización saber con exactitud si se cumplieron los objetivos del proyecto propuestos y si éste concluye con éxito o fracasa.
- La idea central de Seis Sigma es que si puede descubrir como medir los defectos en un proceso, sistemáticamente se puede encontrar como eliminarlos y llevarlo tan cercano como se pueda a “cero defectos” .
- Desarrollar proyectos basados en hechos y enfocados en lo que quiere el cliente y no en lo que se cree que el cliente quiere, brinda la mayor confianza a las directivas de la División de Servicios de Información, de realizar lo que realmente se debe realizar, con un fundamento en mediciones estadísticas.

- Reducir los costos por tiempo que se invierte en corregir programas es un factor que genera un valor agregado para cada organización.
- La utilización de la estadística es una herramienta valiosa para identificar la situación actual del proceso, el cálculo de las medidas, la proyección de las medidas durante y después del proyecto.
- La utilización del modelo de Kano para la clasificación de requerimientos brinda a líderes del proyecto una forma más clara de entender lo que el sistema debe realizar, especificando claramente que requerimientos se deben hacer, cuales son indispensables pero que el cliente pasa desapercibido, y cuales generaran un impacto bueno en la satisfacción del cliente.
- Dentro del modelo existen diferentes formas de hacer las cosas, lo más importante es que se pueda encontrar la más adecuada según el problema que se vaya a solucionar y que no se deje de realizar algún paso dentro de las fases.
- El proceso personal del software, es un poderoso complemento con la metodología Seis Sigma, ya que sus finalidades son muy parecidas “resultados casi perfectos”. Además PSP con sus herramientas, brinda a los desarrolladores individuales la forma de mejorar su rendimiento personal a la hora realizar su trabajo.

15. RECOMENDACIONES

- Continuar trabajando en el mejoramiento del modelo DMADDV planteado para la División de Servicios de Información, donde se acomode mucho más a la cultura organizacional de ésta dependencia.
- Utilizar el modelo DMADDV inicialmente para los proyectos de desarrollo de software que se realicen en la División de servicios de información y que sean realizados por estudiantes en modalidad de práctica empresarial. Y progresivamente a los demás tipos de proyectos.
- Crear el rol de Master Black Belt dentro de la división de servicios de información para que sirva de apoyo a todos los proyectos de desarrollo de software con la metodología DMADDV, que se realicen a partir del momento en que se decida implantarse el modelo.
- Generar un alto grado de compromiso de la alta dirección en la División de Servicios de Información para la utilización de la metodología, logrando así un mayor apoyo con cualquier miembro de esta.
- Utilizar la herramienta QFD para la priorización de requerimientos, como indispensable en cualquier tipo de proyecto, sin importar que sea en forma complementaria.
- Utilizar el AMFE, para determinar los riesgos que se pueden presentar en el proyecto antes y después de su implantación, y manejar la administración de riesgos lo mejor posible durante el proyecto.

- Crear un espacio definido para todo el personal de la División de Servicios de Información donde se presenten capacitaciones de diferentes tipos, como por ejemplo, diseño, hardware, software.

BIBLIOGRAFÍA

JACOBSON, Ivar – BOOCH, Grady - RUMBAUGH, James. El Proceso Unificado de Desarrollo de Software. Editorial Addison Wesley Iberoamericana. Madrid, 2000.

_____ El Lenguaje Unificado de Modelado. Editorial Addison Wesley Iberoamericana Madrid, 1999.

PRESSMAN, Roger S. Ingeniería del software un enfoque práctico. Editorial McGraw Hill. Cuarta Edición. España. 1.998.

TAYNTOR, Christine B. Six Sigma Software Development. Auerbach Publications. USA, 2003.

PANDE, Peter S. – NEUMAN, Robert P. – CAVANAGH, Roland R. Las Claves de Seis Sigma. Editorial McGraw Hill. Edición en español , traducido de la primera versión en español de *The Six Sigma Way*.

BRUE, Greg. Seis Sigma para directivos. Editorial McGraw Hill. Edición en español, traducido de la primera edición en inglés de *Six sigma for Managers*.

LOWENTHAL, Jeffrey N. Six Sigma Project Management: A Pocket Guide. ASQ Quality Press publications, Wisconsin. 2002.

CHOWDHURY, Subir. The Power of Six Sigma. Editorial Dearborn Trade. USA. 2001

HARRY, Mikel – SCHROEDER, Richard. Six Sigma: The Breakthrough Management Strategy Revolutionizing the World's Top Corporations. Editorial Currency. New York. 2000.

FALKNER, Jayson – GALBRAITH, Ben – IRANI, Romin – KOCHMER, Casey – KUNNUMPURATH, Meeraj – PANDURANGA, Sathya – PERRUMAL, Krishnaraj – TIMNEY, John. Beginning ASP Web Development. Wrox Press LTD. USA. 2001.

EDWIN RAMON SUAREZ ALVAREZ. Proyecto de Grado "Sistema de información Intranet-Extranet para la seccional de la Universidad Industrial de Santander en Barrancabermeja"

BIBLIOGRAFÍA EN INTERNET

<http://www.ge.com/sixsigma>: Este es el sitio de General Electric, una de las principales compañías Seis Sigma del momento.

<http://www.isixsigma.com>. Este sitio esta dedicado a dar soporte a todos los practicantes de la metodología Seis Sigma.

<http://www.seissigma.com>. Este sitio es un complemento teórico de Seis Sigma.

<http://www.sei.cmu.edu>. Este es el sitio de Software Engineering Institute y contiene información sobre ingeniería del Software.

<http://www.pmi.org>. Este es el sitio de Project Management Institute y contiene información adicional sobre Project Management.

<http://www.construx.com/survivalguide>. Este sitio web es un complemento del libro *Software Project Survival Guide* de Steve McConnell (Microsoft Press, 1998)

ANEXO A. Cuadro resumen de proyecto

A continuación se describen las secciones que contiene el modelo del cuadro resumen de proyecto y la descripción de los campos que se deben llenar en cada una.

Sección descripción del proyecto

- **Fecha de inicio:** Es la fecha en la que se da inicio al proyecto, puede ser la fecha en la que se convoca al equipo de trabajo o la fecha del primer encuentro.
- **Fecha estimada fin:** La fecha en que planea que las fases sean completadas.
- **Título del proyecto:** debe ser claro y visualizar rápidamente el proyecto.
- **Definición del Problema:** Mencionar una breve descripción del proyecto (problemas a tratar). Tratar de mantener un problema por proyecto. Así las cosas son menos complicadas y se ejecutan más rápido!!!
- **Declaración de los objetivos:** Después de haberse mencionado el problema, se plantean los objetivos.

Sección beneficios

El propósito de esta sección es cuantificar los beneficios proyectados en el proyecto. En el modelo se mencionan algunos, como el nivel sigma, COPQ⁶⁹ y satisfacción del cliente, aunque se pueden describir otros beneficios.

⁶⁹ COQP: Cost of Poor Quality, Costo por Mala Calidad

- **Unidades:** A excepción del nivel sigma, todas las entradas en beneficios deben tener una unidad de medida específica, por ejemplo en COPQ las unidades pueden ser porcentaje, mientras que satisfacción del cliente se puede dar en una “escala del 1 al 5”.
- **Medida Actual:** Se da entrada al nivel actual en cada ítem.
- **Meta:** El nivel que se proyecta alcanzar en cada uno de los ítems.
- **Fecha proyectada:** la fecha en la que se proyecta se alcance el beneficio.

Sección miembros del equipo

El *cuadro de Proyecto* puede contener también la lista de personas implicadas en el proyecto, incluyendo a los miembros del equipo, al personal de soporte y consultoría así como al patrocinador del proyecto.

- **Nombre:** El nombre de cada uno de los miembros del equipo
- **Rol:** El papel que juega dentro del equipo.
- **Departamento:** La dependencia a la que pertenece.
- **% Tiempo:** El porcentaje de tiempo que se espera cada miembro del equipo invierta en el proyecto.

Sección cronograma

Esta sección sirve en gran escala como un plan de proyecto, mostrando los datos en donde cada fase del DMADDV planea ser completada. Proyectos largos suelen dividir las fases en fases más pequeñas y también documentan las fechas planeadas de finalización.

- **Entregables:** El nombre de la fase o entregable a seguir.
- **Fecha Inicio:** La fecha en la que se da inicio a la fase.
- **Fecha Fin:** La fecha en la que se espera se complete la fase.

- **Responsables:** La persona que tiene la responsabilidad sobre la fase. No siempre tiene que ser el líder del equipo.
- **Comentarios:** Donde se indica si se ha completado satisfactoriamente cada fase o si es necesario cambiar la fecha de fin.

Se establecen las fechas de presentar la fase al comité de proyecto, donde se evalúan y se da el visto bueno para seguir a la siguiente fase.

Sección de aprobaciones

- **Rol/Título:** Es desarrollado por un grupo seleccionado o por el comité de proyectos
- **Nombre:** Se escribe el nombre del funcionario.
- **Fecha:** La fecha en la que el funcionario aprobó el cuadro de proyecto.

Sección revisiones

- **Revisión número:** El numero de revisión, normalmente es secuencial.
- **Autor:** Los nombres de las personas que a la fecha han revisado el documento.
- **Fecha:** La fecha en la que se hace la revisión.

Tabla 49. Cuadro resumen proyecto.

CUADRO RESUMEN DE PROYECTO				
<i>Descripción del Proyecto</i>				
Fecha Inicio:		Fecha estimada fin:		
Título del Proyecto:				
Definición del Problema:				
Declaración de los objetivos				
<i>Beneficios</i>				
	Unidades	Medida Actual	Meta	Fecha proyectada
Nivel sigma				
COPQ				
Satisfacción del cliente				
Otro:				
<i>Miembros del Equipo</i>				
Nombre	Rol	Departamento	% Tiempo	
<i>Cronograma</i>				
Entregables	Fecha inicio	Fecha Fin	Responsables	Comentarios
Fase Definir				
Fase Medir				
Fase Analizar				
Fase Diseñar				
Fase Desarrollar				
Fase Verificar				
<i>Aprobaciones</i>				
Rol/Título	Nombre	Fecha		
<i>Revisiones</i>				
Revisión número	Autor	Fecha		

ANEXO B. Técnicas para capturar la voz del cliente.

Tabla 50. Métodos para capturar la VOC.

Método	Tiempo de recolección de datos	Datos	Costo	Ventajas	Desventajas
Encuesta telefónica.	2 semanas	Cuantificable (objetiva)	Alto	Si los datos son correctos, se aplican a una población grande.	Rígido, y poca flexibilidad.
Encuesta por correo electrónico.	Meses	Cuantificable (objetiva)	Bajo	Bajo costo, los datos se puede generalizar a poblaciones grandes.	Demoras en obtener respuesta.
Dirigirse a Grupos en persona.	Un día	Cualitativo (subjetivo)	Bajo-medio	Puede ser personalizado, es una oportunidad para profundizar en las respuestas. Se pueden usar ayudas visuales.	No se puede generalizar a grandes poblaciones, las respuestas profesionales por unos pocos se convierten en riesgo. Movilidad de asistentes.
Dirigirse a grupos en línea.	Un día	Cualitativo (subjetivo)	Bajo	Puede ser personalizado. No requiere de viajes en la movilidad de participantes, ayudas visuales pueden ser utilizadas.	No se puede generalizar a grandes poblaciones. Dificultad en grupos de mayor edad.
Entrevistas uno a uno.	Varios días	Cualitativo (subjetivo)	Bajo	Puede ser personalizado. Oportunidad para profundizar en algunas respuestas. Ayudas visuales pueden ser utilizadas.	Entrevistas desordenadas y falta de seriedad, genera poca colaboración para recolectar datos, requieren mucha movilidad de los entrevistadores. Se pierde el beneficio de la discusión en grupo.
Intercepciones.	Varios días	Ambos	Medio	Bajo especificas circunstancias, los datos pueden ser	No se puede generalizar a poblaciones

				cuantificados a una limitada población. Ayudas visuales pueden ser utilizadas.	grandes. Apropiado para pocos tópicos.
Pruebas a usuarios.	Un mes	Ambos	Medio	Provee amistad con los usuarios.	No es adecuado para la investigación. Apropiado para pocos tópicos.
Quejas de los clientes.	Varios Meses	Cualitativos (subjetivo)	Bajo	Provee entradas específicas.	No se puede generalizar. Apropiado para pocos tópicos.
Benchmarking.	Un Mes	Cualitativos	Medio	Compara nuestros procesos con los procesos de los mejores competidores.	Los procesos no se pueden implantar en cualquier organización.
Data warehousing.	Varios meses	Cuantitativos	Alto	Grandes volúmenes de datos históricos acerca del negocio.	Necesita de tiempo para recolectar una cantidad suficiente de datos.
Data mining.	Varios Meses	Ambos	Medio	Análisis de datos con relaciones ocultas.	Necesidad de tiempo para recolectar una suficiente cantidad de datos.

ANEXO C. Captura de Requisitos con UML.

Para guiar la captura de requisitos del sistema, se establece el siguiente flujo de trabajo:

1. Comprender el contexto del sistema.

En proyectos de desarrollo de software se encuentran implicados diferentes roles de personas que tienen asignadas diferentes responsabilidades. Cada uno de ellos debe tener un conocimiento claro acerca del negocio en que se encuentra y en como el sistema va a ser parte de él.

Los métodos que se utilizan para el entendimiento de la situación son explicados más a fondo en el paso de “Definición del problema”. Cuando el modelo es entendido y se convierte manejable por los miembros del equipo de trabajo, la comunicación con los clientes se hace fluida y sin obstáculos.

2. Crear un borrador de la definición de requisitos.

Durante el transcurso del proyecto los clientes, usuarios, analistas y desarrolladores van entendiendo mejor el contexto del sistema y su entorno; con una comprensión de alto nivel por parte de los desarrolladores, la cual los hace generar una gran cantidad de ideas acerca de que debería tener el sistema. Crear una lista de estas ideas puede mantener el proyecto bajo control, ya que muchas de estas ideas

se caracterizan como “algo bueno por tener”⁷⁰ y otras “algo indispensable a tener”⁷¹.

La lista se convierte en un fundamento indispensable para la planificación, transformándose en requisitos, y después en casos de uso.

Cada característica debe poseer ciertos atributos con los cuales los miembros del equipo se basaran para planificar el trabajo, tomar decisiones y clasificarlas.

- Nombre.
- Descripción o Definición.
- Breve justificación.
- Coste estimado de implementación.
- Prioridad.

3. Capturar los requisitos funcionales.

En la definición de requisitos del sistema, cada cliente se encuentra en la posición de que el sistema realice lo que el quiere que haga. En esta lista de requisitos encontramos dos tipos, los funcionales y los no funcionales; de modo que los miembros del equipo encargados de esta tarea deben estar preparados con fundamentos para identificar cuales son funcionales y cuales no. En el momento que los analistas puedan representar los requisitos mediante casos de uso y se encuentren con la capacidad de describir cada caso de uso, entonces saben lo que debe hacer el sistema.

Por otro lado, los clientes tienen demasiadas tareas que desearían que el sistema hiciera por ellos, por lo tanto, la voz del cliente⁷² juega un papel

⁷⁰ Algo bueno por tener: Siglas en ingles “NICE TO HAVE”.

⁷¹ Algo indispensable a tener: Siglas en ingles “HAVE TO HAVE”.

muy importante, ya que ayuda a comprender que es realmente necesario y que no.

4. Capturar los requisitos no funcionales.

Los requisitos no funcionales, son aquellos que no se encuentran afectando de una manera directa el objetivo principal del sistema, haciendo referencia a fenómenos del mundo real, que no son realmente complicados de identificar, como restricciones del entorno, la implementación, el rendimiento, las dependencias de las plataformas, facilidad de mantenimiento, extensibilidad y fiabilidad.

5. Validar los requisitos.

La validación incluye cualquier acción que pueda emprender para volver a comprobar los requisitos y garantizar que reflejan exactamente las necesidades y expectativas del cliente. Un método puede ser dar a los clientes un ejemplo basado en los requisitos y luego esperar sus reacciones o, simplemente preguntarles. Esta validación puede también incluir una comprobación con la gente que interviene en el proceso y que necesitará interpretar y cumplir esos requisitos.

6. Identificar el cliente o usuarios finales.

En el modelo de casos de uso se describe lo que hace el sistema para cada usuario. Estos usuarios se representan mediante “actores⁷³”. En este caso, usuario, representa al cliente final o a los otros sistemas con que interactúa.

⁷² Voz del Cliente: Ver anexo B.

⁷³ Actor: El conjunto de roles que los usuarios de casos de uso desempeñan cuando interactúan con estos casos de uso. Fuente libro: “El proceso unificado de desarrollo de software UML”.

Los actores y los casos de uso se identifican para delimitar el sistema de su entorno y para esbozar quién y que (actores) va a recibir el producto o servicio e interactuará con el sistema, y que funcionalidad (casos de uso) se espera.

La identificación de actores y de casos de uso es la actividad más decisiva para obtener adecuadamente los requisitos y es responsabilidad del analista de sistemas.

Esta actividad incluye 4 pasos:

- *Encontrar los actores.*

El analista de sistemas junto con el cliente identifica los usuarios e intenta organizarlos en categorías representadas por *actores*. Para elegir los candidatos a actores se tienen en cuenta los siguientes criterios: en primer lugar debe existir al menos un usuario que se puede representar con un actor, evitando crear actores que no son necesarios. El segundo criterio, no deben existir dos actores que posean un mismo rol, de ser así los actores deben ser generalizados en un nuevo rol que contenga las características de los actores que se repiten.

El analista de sistemas da nombre a los actores y describe brevemente los papeles de cada uno y para qué utiliza el sistema el actor.

- *Encontrar los Casos de uso.*

Los casos de uso deben ser planteados por los analistas del sistema. El modelo del negocio juega un papel fundamental ya que de él surgen muchos casos, en los cuales para cada actor se plantea mínimo un caso de uso.

Después de el análisis de los casos de uso, muchos de los que se han planteado, no llegan a considerarse como un caso uso; sin embargo se consideran parte de otros, ya que un caso puede invocar a otro y así sucesivamente. Los casos de uso deben ser lo menos complejos posibles, donde posean características de modificar, revisar, probar y manejar unitariamente.

- *Describir brevemente cada caso de uso.*

Mientras los miembros del equipo trabajan en la identificación de los casos de uso, su conocimiento y entendimiento acerca del proceso se incrementa a un alto nivel, por esta razón, los miembros tienden a clasificar diferentes asuntos importantes como obvios, y los casos comienzan a empobrecerse en el detalle, por tal motivo, se debe proponer una descripción breve de los pasos y las acciones que el sistema tiene con los actores.

- *Describir el modelo de caso de uso completo:*

Por medio de una breve descripción y diagramas se explica como se relacionan los casos de uso entre sí con sus actores.

7. Priorizar los casos de uso.

El principio de esta actividad es llevar a cabo una planeación del desarrollo en diferentes sentidos, económico, duración, recursos, o el negocio en general.

Cada caso de uso debe tener un identificador que describa el orden en que se debe desarrollar y el momento de realizarse. Pueden ubicarse en diferentes fases del proyecto análisis, medición, diseño, e implementación. La fase de verificación utiliza los casos de uso para comprobar el rendimiento, y el orden en que se prueba es el mismo orden que se les ha dado en este punto.

La aprobación de la priorización de los casos de uso, debe ser certificada por el líder del equipo.

8. Detallar los casos de uso.

Se debe hacer una descripción del flujo de sucesos de cada caso de uso, definiendo una secuencia de acciones, incluyendo como comienza, termina e interactúan con los actores. Esta actividad se debe realizar en conjunto con los usuarios reales de cada caso de uso, para saber que comprensión tienen sobre este y así poder discutir propuestas con ellos.

La descripción de un caso de uso puede incluir lo siguiente:

- Definir el estado inicial.
- Cómo y cuando comienza.
- El orden requerido en el que las acciones se deben ejecutar.
- Cómo y cuando terminan.
- Definición de los posibles estados finales.
- Los caminos de ejecución que no están permitidos.
- Las descripciones de los caminos alternativos, así como la descripción del camino básico.

- La interacción del sistema con los actores y qué cambios producen.
- La utilización de objetos, valores y recursos del sistema.
- La descripción completa de lo que hace el sistema.

A menudo la interacción entre los actores y los casos de uso puede transitar por tantos estados y tantas transiciones alternativas que es casi imposible mantener consistente la descripción textual de los casos de uso. Por tal circunstancia puede ser útil utilizar una técnica de modelado visual para describir los casos de uso, de esta forma se puede ayudar al analista de sistemas a mejorar la comprensión de los casos de uso.

Los diagramas que pueden utilizarse son:

- *Diagramas de estado*, para describir los estados de los casos de uso y las transiciones entre esos estados.
- *Diagramas de actividad*, para describir las transiciones entre estados con más detalle como secuencias de acciones.
- *Diagramas de interacción*, para describir cómo interactúa una instancia de casos de uso con la instancia de un actor.

ANEXO D. Análisis de modos de fallo y efecto. AMFE

El modelo de AMFE contiene cinco secciones principalmente:

- **¿Qué pudo pasar?**

Esta sección describe las posibles causas actuales y los efectos en los requerimientos del cliente.

- *Modo de fallo:* Describe las formas en que el proceso falla. Si existen varias se deben listar en filas separadas.
- *Efectos de la falla:* Describe el impacto que tienen las fallas sobre los requerimientos del cliente.
- *Intensidad (INT):* Cuantifica el impacto de la falla sobre los requerimientos del cliente. Se recomienda utilizar una escala que diferencie claramente la prioridad alta, media y baja.

- **¿Por qué y con qué frecuencia?**

Esta sección cuantifica las principales causas y la frecuencia con que estas fallan ocurren.

- *Principales causas:* Lista de las posibles causas de la falla. Si existen muchas causas se deben listar en filas diferentes.
- *Frecuencia de Ocurrencia:* Cuantifica la frecuencia con que ocurren estas posibles causas. Se debe utilizar la escala definida anteriormente

- **¿Cómo se puede prevenir?**

En esta sección se describen los procedimientos o los controles que se usan actualmente para prevenir la falla.

- *Controles*: Describe los procesos que se usan para prevenir o detectar los modos de fallo.
- *Probabilidad de fallo*: Cuantifica la probabilidad de que el control que se este utilizando falle. También se usa la escala.
- *Número de Prioridad del Riesgo*: Es el resultado de la multiplicación de la *intensidad, frecuencia de ocurrencia y la probabilidad de fallo*.

$$\text{NPR} = \text{INT} * \text{OCU} * \text{PROB}$$

- **Plan de Acción**

Describe las acciones que se siguen en la disminución de los ítems con alto riesgo.

- *Acción Recomendada*: Describe la acción que se toma para disminuir la frecuencia de ocurrencia y la probabilidad de fallo.
- *Responsable*: Contiene el nombre de la persona responsable del plan de acción.

- **Resultado del Plan de Acción**

En esta sección se miden los éxitos de cada acción correctiva, recalculando los campos de intensidad, frecuencia de ocurrencia y el resultado del valor NPR.

- *Resultado de la Intensidad (INT)*: Cuantifica el impacto de la falla sobre los requerimientos del cliente. Normalmente no presenta cambios frente a la acción correctiva. Se calcula usando la escala definida.

- *Resultado de la Frecuencia de Ocurrencia (OCU):* Cuantifica la frecuencia con la que la posible causa ocurre después que la acción correctiva se ha completado.
- *Resultado de la Probabilidad de Fallo (PROB):* Cuantifica la probabilidad de que los controles fallen después que se completa la acción correctiva. Se cuantifica con la escala definida anteriormente.
- *Resultado del NPR:* Se calcula nuevamente multiplicando los tres campos anteriores ($INT * OCU * PROB$). Si la acción correctiva es efectiva, este resultado debe ser menos que el obtenido anteriormente.

INSTRUCCIONES PARA LLENAR EL FORMATO DE REGISTRO DEL TIEMPO

Propósito:	El objetivo de este formulario es registrar el tiempo que gastan los ingenieros programadores en la asignaciones que se les hace a nivel individual.
Generalidades	El tiempo se registra en minutos, se puede redondear a los 5 minutos más próximos. Es importante mantener el formulario siempre a la mano, de tal forma que exista uno siempre disponible a la hora de comenzar a trabajar y completar el formulario de una forma limpia y ordenada. No es recomendable mantener el formato en forma electrónica, a menos que tenga permiso especial del director del proyecto.
Encabezado	Se digita el nombre y el nombre del proceso o módulo asignado para trabajar.
LOC Inicial	Si el trabajo implica un nuevo desarrollo, se digita cero. Si por el contrario se esta reanudando, modificando o ampliando un trabajo existente, se deben determinar las LOC que existen y digitar este número ahí.
LOC Final	Una vez se termina de desarrollar el módulo asignado, se calculan nuevamente las LOC y se digita ese número ahí.
Fecha	Se digita la fecha en que se realiza la entrada. Si existen muchas modificaciones en el mismo día, se puede dejar ese campo en blanco.
Hora Inicio	Se digita el tiempo en que se comienza a trabajar en la programación de un módulo.
Hora final	Se digita el tiempo en que se deja de trabajar en ese módulo.
Tiempo Interrupciones	Se registra cualquier interrupción de tiempo en la que no se trabajó e esa tarea. Se escribe la razón de la interrupción en la columna de "comentarios". Si existen muchas interrupciones se debe registrar de la siguiente forma: 5 + 2 (min.)
Delta Tiempo	Es el tiempo total que se gasto en la tarea menos el tiempo de las interrupciones. Por ejemplo: <i>Hora inicial: 7:43</i> <i>Hora final: 8:24</i> <i>T. Interrupciones: 7 min.</i> <i>Delta tiempo = 34 min.</i>

Actividad	Se digita el nombre de la fase de programación en la que se haya trabajado. En la tabla xx se encuentra las descripciones de las fases de programación. Ejemplo: codificar.
Comentarios	Se registran los comentarios pertinentes que sirvan para recordar cualquier detalle específico de esta actividad. Ejemplo: Revisión de la teoría de conjuntos.
Notas	Si por algún motivo alguna vez se olvida llegar algún registro de tiempo, es necesario escribir la mejor estimación posible, o si por algún motivo este formulario no esta a la mano del programador, se deben registrar los tiempos en un lugar aparte y luego llenar el formulario debidamente. Si olvida registrar un tiempo, ingrese la mejor estimación.

Tabla 52. Descripción de las fases de programación.

Descripción de las fases de programación	
Se pueden usar las siguientes categorías para completar la columna de "Actividad"	
Diseño	Es el tiempo que el programador gasta en pensar en como va a solucionar el problema y en diseñar el algoritmo. Incluye: diagramas, pseudocódigo, y otras tareas, cualquier cosa antes de comenzar a escribir el código es considerado diseño
Codificar	Es la traducción del algoritmo en código fuente.
Compilar	Compilar el código fuente. La fase de compilar se completa cuando el código corre limpiamente sin los errores de sintaxis reportados por el compilador.
Revisión	La fase de revisión implica que el código sea revisado por otra persona.
Pruebas	Implica las pruebas que se hacen al programa para identificar y reparar defectos. Si durante la fase de pruebas hace falta agregar algunas líneas de código, estas se agregan y la actividad que queda registrada es la de pruebas.

REGISTRO DE DEFECTOS

Nombre Programa					Fecha Programa #		
Fecha	Número	Tipo	Encontrado	Removido	Tiempo requerido	Defecto Arreglado	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Descripción: _____

Fecha	Número	Tipo	Encontrado	Removido	Tiempo requerido	Defecto Arreglado	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Descripción: _____

Fecha	Número	Tipo	Encontrado	Removido	Tiempo requerido	Defecto Arreglado	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Descripción: _____

Fecha	Número	Tipo	Encontrado	Removido	Tiempo requerido	Defecto Arreglado	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Descripción: _____

Fecha	Número	Tipo	Encontrado	Removido	Tiempo requerido	Defecto Arreglado	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Descripción: _____

INSTRUCCIONES PARA LLENAR EL FORMATO DE REGISTRO DE DEFECTOS

Propósito	Este formulario registra los datos de cada defecto encontrado y corregido. Estos datos son utilizados para completar el resumen del plan de proyecto.
Generalidades	Se deben registrar todos los defectos durante el desarrollo en este formato. Se registra cada defecto separado y completamente.
Encabezado	Se digita lo siguiente: <ul style="list-style-type: none">- Nombre.- Fecha- Nombre del módulo del programa.- El número del programa.
Fecha	Es la fecha en que fue encontrado el defecto
Número	Es el número de cada defecto. Para cada programa se usa un número secuencial, desde uno (1) hasta n.
Tipo	Se escribe el tipo de defecto según la tabla de estándares de tipos de defectos. Tabla 52. Se debe tener claridad para seleccionar cual tipo aplica.
Introducido	Es la fase en la que el defecto fue introducido. La mayoría de las veces es la fase de codificación.
Removido	Es la fase en la que se removió el defecto.
Tiempo requerido	Es la estimación de la medida del tiempo que se requirió en encontrar y corregir el defecto.
Defecto Arreglado (opcional)	Esta entrada puede ser ignorada. Si este defecto fue introducido al momento de corregir otro defecto, se registra el número del defecto corregido, si es muy difícil de identificar, se puede digitar "X" en ese espacio.
Descripción	Se escribe una breve descripción del defecto. Ésta debe ser clara de tal forma que después se pueda recordar el defecto que causo el error y el por qué se generó.

ESTÁNDAR DE TIPOS DE DEFECTOS

Tabla 53. Estándares de tipos de defectos.

Número	Nombre Tipo	Descripción
10	Sintaxis	Ortografía, puntuación, errores tipográficos, formato de la instrucción.
20	Asignación	Declaración, nombres duplicados, rango de datos, inicialización de datos.
30	Algoritmo.	Errores en el diseño del algoritmo
40	Interfase.	Errores en el módulo del diseño de la interfase: llamadas y referencias a los procedimientos, listas de parámetros.
50	Arquitectura.	Errores en el diseño de la arquitectura: modularización, estructura, acoplamiento, cohesión.
60	Datos	Errores en el diseño de los datos: estructura y contenido.
70	Comprobación	Fallas al momento de validar que los valores de los datos sean verdaderos antes de ser usados, mensajes de error.
80	Documentación	Comentarios del código fuente, mensajes y otra documentación externa.
90	Construcción, Empaquetamiento	Cambios de administración, librerías, control de versiones, creación de archivos de error, etc.
100	Entorno	Herramientas CASE, compiladores, pruebas, u otros problemas del sistema de soporte.
110	Sistema	Hardware y configuración de la plataforma, recursos, memoria compartida.