

SISTEMA WEB PARA LA GESTIÓN DE LA CALIDAD EN LOS PROCESOS DE LAMINACIÓN DE SIDOR.

Autor: Br. Angel Augusto Sierraalta.

Profesor Tutor: Prof. Milagro Rivero.

Profesor Cotutor: Prof. Mario Spinetti.

Asesor Industrial: Ing. Carlos Agostini.

www.bdigital.ula.ve

Proyecto de Grado presentado ante la ilustre Universidad de Los Andes
como requisito final para optar al Título de Ingeniero de Sistemas.

Mérida, Venezuela

Enero 2005



UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA

Universidad de Los Andes © Derechos Reservados 2005.

Resumen

Hoy en día los sistemas de información Web han ido ganando terreno, hasta llegar el punto de que han favorecido el éxito de las empresas. El uso de estos sistemas permite mantener un mejor control de las principales operaciones de las empresas, tales como: planeación, organización, dirección, gestión y control, ya que la adecuada administración de las mismas conlleva a poder tomar mejores decisiones que benefician a la empresa y a los objetivos de esta.

La toma de decisiones en una empresa es un proceso complejo que origina responsabilidades y riesgos que hay que asumir, por lo que es necesario contar con tecnología de punta y estar a la vanguardia para adaptar las nuevas técnicas que vayan surgiendo. Actualmente las empresas grandes como SIDOR cuentan con tecnologías y sistemas de punta que les permiten tomar decisiones de una manera más sencilla, que les ahorra tiempo y que les permite evaluar distintos escenarios y de ahí tomar la mejor decisión de acuerdo a sus necesidades u objetivos.

Una de estas nuevas tecnologías que se aplican en SIDOR y que facilitan la gestión de los procesos es el desarrollo de aplicaciones Web basadas en tecnología de punta tal como lo es .NET, ya que esta nueva herramienta, en conjunto con manejadores de bases de datos avanzados como por ejemplo SQL SERVER, permiten un amplio desarrollo de aplicaciones Web robustas y estables, las cuales están ligadas directamente a los procesos que ocurren en planta. Un ejemplo de esto es el sistema que aquí se presenta, cuya implementación se encuentra realizada en .NET. Además de esto, durante el análisis y diseño de este sistema se sigue como guía el Proceso Unificado de Desarrollo de Software, lo cual le da un soporte teórico al sistema bueno y bien detallado.

Índice

Resumen.....	ii
Índice.....	iii
Índice de Figuras.....	ix
Índice de Tablas	xii
Capítulo 1	1
1 Introducción	1
1.1 Descripción de la empresa	1
1.2 Proceso de Laminación	3
1.2.1 Laminador en Caliente.....	4
1.2.2 Línea de Decapado I y II.....	4
1.2.3 Laminador Tándem I y II.....	4
1.2.4 Línea de estañado y cromado electrolítico I.	5
1.2.5 Línea de estañado y cromado electrolítico II.....	5
1.3 Ubicación del proyecto dentro de SIDOR	5
1.4 Antecedentes	8
1.5 Definición del Problema	8
1.6 Objetivos.....	10
1.6.1 Objetivo General:.....	10
1.6.2 Objetivos Específicos:	10
1.7 Metodología	11
1.7.1 Proceso Unificado (UP).	11
1.8 Alcance del proyecto.....	11
1.9 Estructura del documento	12
Capítulo 2.....	14
Marco Teórico.....	14
2 Introducción	14

2.1	Sistemas de Información.....	14
2.1.1	Objetivos de los Sistemas de Información.....	17
2.2	El Proceso Unificado de Desarrollo de Software:	17
2.2.1	Ciclos de Proceso Unificado.....	18
2.2.2	Fases dentro de un Ciclo.....	19
2.3	Lenguaje de modelado unificado (UML)	21
2.4	Bases de Datos	22
2.4.1	Sistema de Gestión de Base de Datos (SGBD):.....	22
2.5	La Plataforma .NET	23
2.5.1	El .NET Framework.....	24
2.5.2	Componentes del .NET Framework	25
2.5.2.1	Common Language Rutine:	25
2.5.2.2	Biblioteca de Clases.....	26
2.5.3	Funcionamiento del .NET Framework.	26
2.6	Visual Basic .NET y ASP .NET:	27
2.6.1	Aplicaciones ASP.NET.....	27
2.6.1.1	Funcionamiento de una aplicación ASP.NET	28
2.7	XML Web Service	30
2.7.1	Funcionamiento de los XML Web Service.....	31
2.7.1.1	Simple Object Access Protocol (SOAP):.....	31
2.7.1.2	WSDL:	31
2.7.1.3	Web Service Discovery (DISCO):.....	32
2.7.1.4	UDDI Business Registry:.....	32
2.8	Microsoft SQL SERVER 2000	32
2.8.1	Motor de SQL SERVER.....	32
2.8.2	Procedimientos almacenados y funciones	33
Capítulo 3	34
Fase de Inicio	34
3	Introducción	34
3.1	VISIÓN GENERAL:	35
3.2	LISTA ACTOR-OBJETIVOS:	36

3.3	MODELO DE CASOS DE USO.....	37
3.3.1	Caso de Uso UC1: Revisar Defectos.	39
3.3.2	Caso de Uso UC2: Analizar Bobinas.....	41
3.3.3	Caso de Uso UC3: Consultar Cantidad Bobinas Asignadas.....	43
3.3.4	Caso de Uso UC4: Revisar Bobinas Asignadas.....	44
3.3.5	Caso de Uso UC5: Modificar datos Cualitativos.....	45
3.3.6	Caso de Uso UC6: Aprobar datos Cualitativos.....	46
3.3.7	Caso de Uso UC7: Asignar Responsables.	46
3.3.8	Caso de Uso UC8: Asignar Causas.....	47
3.4	ESPECIFICACIONES COMPLEMENTARIAS	48
3.4.1	Introducción:	48
3.4.2	Detección y gestión de errores.....	48
3.4.3	Seguridad	48
3.4.4	Facilidad de Uso	48
3.4.5	Rendimiento.....	49
3.4.6	Restricciones de Implementación.....	49
3.5	GLOSARIO	49
3.6	PLAN DE ITERACIÓN:	50
Capítulo 4	51
Fase de Elaboración	51
4	Introducción	51
4.1	ITERACIÓN I.....	52
4.1.1	MODELO DE CASOS DE USO.....	52
4.1.1.1	DSS 1: Escenario Revisar Defectos.....	54
4.1.1.2	DSS 2: Escenario Analizar Bobinas.	54
4.1.1.3	DSS 3: Escenario Consultar Cantidad Bobinas Asignadas.....	55
4.1.1.4	DSS 4: Escenario Revisar Bobinas Asignadas.	55
4.1.2	MODELO DEL DOMINIO.....	56
4.1.2.1	Diagrama de Clases Conceptuales.	56
4.1.3	MODELO DE DISEÑO.	59
4.1.3.1	Diagramas de Interacción (DI).....	60

4.1.3.1.1	DI 1: Escenario Revisar Defectos.	60
4.1.3.1.2	DI 2: Escenario Analizar Bobinas.	62
4.1.3.1.3	DI 3: Escenario Consultar Cantidad Bobinas Asignadas.	62
4.1.3.1.4	DI 4: Escenario Revisar Bobinas Asignadas.	63
4.1.3.2	Diagrama de Clases de Diseño (DCD).	64
4.1.4	IMPLEMENTACIÓN.	66
4.1.5	GLOSARIO.	68
4.1.6	PLAN DE ITERACIÓN:	70
4.2	ITERACIÓN II.	70
4.2.1	MODELO DE CASOS DE USO.	71
4.2.1.1	Descripción de los Casos de Uso (Requisitos funcionales).	73
4.2.1.1.1	Caso de Uso UC 9: Eliminar Responsable Asignado.	73
4.2.1.1.2	Caso de Uso UC 10: Revisar Acciones Asignadas a Causas.	75
4.2.1.1.3	Caso de Uso UC 11: Asignar Acciones a Causas.	76
4.2.1.2	Especificaciones Complementarias.	79
4.2.1.2.1	Introducción.	79
4.2.1.2.2	Detección y gestión de errores.	79
4.2.1.2.3	Restricciones de Implementación.	79
4.2.1.3	Elaboración de Diagramas de Secuencia.	79
4.2.1.3.1	DSS 5: Eliminar Responsable Asignado.	80
4.2.1.3.2	DSS 6: Revisar Acciones asignadas a Causas.	80
4.2.1.3.3	DSS 7: Asignar Acciones a Causas.	81
4.2.2	MODELO DEL DOMINIO.	81
4.2.2.1	Diagrama de Clases Conceptuales.	81
4.2.3	MODELO DE DISEÑO.	83
4.2.3.1	Diagramas de Interacción	83
4.2.3.1.1	DI 5: Eliminar Responsable Asignado.	83
4.2.3.1.2	DI 6: Revisar Acciones Asignadas a Causas.	84
4.2.3.1.3	DI 7: Asignar Acciones a Causas.	85
4.2.3.2	Diagrama de Clases de Diseño.	86
4.2.4	IMPLEMENTACIÓN.	87

4.2.5	GLOSARIO.	88
4.2.6	Plan de Iteración.	88
4.3	ITERACIÓN III.	89
4.3.1	MODELO DE CASOS DE USO.	90
4.3.1.1	Descripción de los Casos de Uso (Requisitos funcionales).	92
4.3.1.1.1	Caso de Uso UC 12: Gestión de Procesistas.	92
4.3.1.1.2	Caso de Uso UC 13: Gestión de Defectos.	94
4.3.1.1.3	Caso de Uso UC 14: Gestión Acciones Asignadas.	95
4.3.1.1.4	Caso de Uso UC 15: Ver Pareto Causas.	97
4.3.1.1.5	Caso de Uso UC 16: Ver Pareto Acciones.	98
4.3.1.2	Elaboración de Diagramas de Secuencia.	99
4.3.1.2.1	DSS 8: Gestión de Procesistas.	100
4.3.1.2.2	DSS 9: Gestión de Defectos.	100
4.3.1.2.3	DSS 10: Gestión Acciones Asignadas.	101
4.3.1.2.4	DSS 11: Ver Pareto Causas.	101
4.3.1.2.5	DSS 12: Ver Parreto Acciones.	102
4.3.2	MODELO DEL DOMINIO.	102
4.3.3	MODELO DE DISEÑO.	103
4.3.3.1	Diagramas de Interacción.	103
4.3.3.1.1	DI 8: Gestión de Procesistas.	103
4.3.3.1.2	DI 9: Gestión de defectos.	105
4.3.3.1.3	DI 10: Gestión Acciones Asignadas.	106
4.3.3.1.4	DI 11: Ver Pareto Causas.	106
4.3.3.1.5	DI 12: Ver Pareto Acciones.	106
4.3.3.2	Diagrama de Clases de Diseño.	107
4.3.4	IMPLEMENTACIÓN.	108
4.3.5	GLOSARIO.	109
4.3.6	PLAN DE ITERACIÓN.	110
Capítulo 5	111
Fase de Construcción.	111
5	Introducción.	111

5.1	DISEÑO DE LA INTERFAZ.....	111
5.1.1	DIAGRAMA DE FLUJO DE NAVEGACIÓN DEL SGC.....	113
5.2	CONEXIÓN ENTRE LAS CAPAS DE LA APLICACIÓN.	114
5.3	ESPECIFICACIONES COMPLEMENTARIAS.	117
5.4	GLOSARIO.	118
5.5	PLAN DE ITERACIÓN.	118
Capítulo 6	119
Fase de Transición.	119
6	Introducción.....	119
6.1	Implementación de las Pantallas.....	119
6.1.1	Pantalla Revisión de Defectos.....	120
6.2	Pruebas Finales del Sistema.....	122
6.2.1	Pruebas sobre la pantalla Revisión de defectos:.....	122
6.2.2	Pruebas sobre la pantalla Asignación de Responsables:.....	124
6.2.3	Pruebas sobre la pantalla Eliminación de Responsables:.....	126
6.2.4	Pruebas sobre la pantalla Cantidad de Bobinas Asignadas:.....	126
6.2.5	Pruebas sobre la pantalla Revisar Bobinas Asignadas:.....	127
6.2.6	Pruebas sobre la pantalla Revisar Acciones Asignadas a Causas:.....	127
6.2.7	Pruebas sobre la pantalla Gestión de Acciones Asignadas:.....	127
6.3	Puesta en Producción del Sistema.....	128
6.3.1	Diagrama de Implementación.	128
Conclusiones	131
Recomendaciones	133
Bibliografía	134
Anexos	136
Anexo A: Diccionario de Datos.....		136
Anexo B: Diccionario de Términos.....		143
Anexo C: Pantallas del Sistema.		145

Índice de Figuras

Figura 1 Ubicación Geográfica de SIDOR.....	2
Figura 2. Layout de la Empresa	2
Figura 3 Procesos Planos	3
Figura 4 Organigrama Gerencial de Sidor	6
Figura 5. Dirección Industrial	7
Figura 6. Fines del Sistema de Información	15
Figura 7. Proceso Entrada Salida	17
Figura 8. Ciclos del proceso Unificado.....	18
Figura 9. Fases del Proceso Unificado.....	19
Figura 10. Fases	19
Figura 11. Plataforma .Net.....	24
Figura 12. Componentes de .NET Framework	25
Figura 13. Aplicación ASP.NET	28
Figura 14. Funcionamiento de una aplicación ASP.NET	29
Figura 15. Funcionamiento de una aplicación ASP.NET con BD.....	30
Figura 16. Funcionamiento de un Web Service.....	31
Figura 17. Visión general del Sistema	35
Figura 18. Diagrama de casos de uso parcial.....	38
Figura 19. DSS 1 Revisar Defectos.	54
Figura 20. DSS 2 Analizar Bobinas.....	54
Figura 21. DSS 3 Consultar Cantidad Bobinas Asignadas.	55
Figura 22. DSS 4 Revisar Bobinas Asignadas.....	55
Figura 23. Diagrama del Dominio (Fase I).....	57
Figura 24. Diagrama del Dominio (Fase II).....	57
Figura 25. Diagrama de clases de Dominio - Iteración I.	58
Figura 26. DI 1 Revisar Defectos.	61

Figura 27. DS 2 Analizar Bobinas.	62
Figura 28. DI 3 Consultar Cantidad Bobinas Asignadas.	63
Figura 29. DI 4 Revisar Bobinas Asignadas.	64
Figura 30. Diagrama de Clases de Diseño - Iteración I.	65
Figura 31. Diagrama de Casos de Uso - Iteración II.	72
Figura 32. DSS 5: Eliminar Responsable Asignado.	80
Figura 33. DSS 6: Revisar Acciones Asignadas a Causas.	80
Figura 34. DSS 7: Asignar Acciones a Causas.	81
Figura 35. Diagrama de Clases de Dominio - Iteración II.	82
Figura 36 DI 5: Eliminar Responsable Asignado	84
Figura 37 DI 6: Revisar Acciones Asignadas a Causas.	85
Figura 38 DI 7: Asignar Acciones a Causas.	85
Figura 39 Diagrama de Clases de Diseño - Iteración II.	86
Figura 40 Diagrama de Casos de Uso - Iteración III.	91
Figura 41. DSS 8: Gestión de Procesistas.	100
Figura 42. DSS 9: Gestión de Defectos.	100
Figura 43. DSS 10: Gestión Acciones Asignadas.	101
Figura 44. DSS 11: Ver Pareto Causas.	101
Figura 45. DSS 12: Ver Pareto Acciones.	102
Figura 46. Diagrama de Clases de Dominio – Iteración III.	103
Figura 47. DI 8: Gestión de Procesistas.	104
Figura 48. DI 9: Gestión de Defectos.	105
Figura 49. DI 10: Gestión de Defectos.	106
Figura 50. DI 11: Ver Pareto causas.	106
Figura 51. DI 12: Ver Pareto Acciones.	107
Figura 52. Diagrama de clases de Diseño – Iteración III.	107
Figura 53. Diseño general de las pantallas.	112
Figura 54. Diagrama de flujo de navegación del sistema.	114
Figura 55. Conexión de la capa de UI, Negocio y Datos.	115
Figura 56. Menú.	119
Figura 57. Filtros de la pantalla Revisión de Defectos.	120

Figura 58. Datos a visualizar por el usuario.	121
Figura 59. Pantalla Revisión de defectos.	121
Figura 60. Rango de fechas inválido.....	123
Figura 61. Listado de Bobinas Defectuosas.....	124
Figura 62. Listado de bobinas a asignar responsables	125
Figura 63. Lista de responsables posibles a asignar.	125
Figura 64. Resultado de una asignación	126
Figura 65. Diagrama de Implementación.....	129
Figura 66. Diagrama de Requisitos de software.	130

www.bdigital.ula.ve

Índice de Tablas

Tabla 1. Artefactos de la fase de Inicio.....	35
Tabla 2. Lista Actor – Objetivos.....	37
Tabla 3. Glosario – Fase de Inicio	50
Tabla 4. Artefactos de la fase de Inicio.....	52
Tabla 5. Descripción de las clases del Modelo de Dominio.	59
Tabla 6. Definición de Procedimientos Almacenados.....	68
Tabla 7. Glosario – Fase de Elaboración – Iteración I.....	70
Tabla 8. Listado Actor – Objetivos.....	71
Tabla 9. Descripción de clases del Modelo de Dominio.	82
Tabla 10. Definición de Procedimientos Almacenados.....	88
Tabla 11. Glosario – Fase de Elaboración – Iteración II	88
Tabla 12. Lista Actor – Objetivos.....	90
Tabla 13. Definición de Procedimientos Almacenados.....	109
Tabla 14. Glosario – Fase de Elaboración – Iteración III	109
Tabla 15. Glosario – Fase de Construcción	118
Tabla 16. Datos de prueba 1 – pantalla Revisión de defectos.	123
Tabla 17. Datos de prueba 2 – pantalla Revisión de defectos.	123
Tabla 18. Prueba pantalla asignación de responsables.	124
Tabla 19. Prueba pantalla eliminación de responsable.	126
Tabla 20. Prueba pantalla cantidad de bobinas asignadas.	126
Tabla 21. Prueba pantalla revisar bobinas asignadas.....	127
Tabla 22. Prueba pantalla Acciones Asignadas a Causas.	127
Tabla 23. Prueba pantalla gestión de acciones.....	128

Capítulo 1

1 Introducción

En este capítulo se dará una breve descripción de la empresa, se mostrará el área dentro de SIDOR donde tuvo lugar la realización del proyecto, así como también, se dará una descripción corta de los procesos de laminación de SIDOR. Además, se presentan los antecedentes que sirven de base para la realización del proyecto, la definición del problema, los objetivos que tendrá que cumplir el proyecto, una breve descripción de la metodología a utilizar, el alcance del proyecto y finalmente la estructura del documento.

1.1 Descripción de la empresa

La Siderúrgica del Orinoco (SIDOR), C.A. Es una empresa privada dedicada a la fabricación de productos de acero, destinados tanto al mercado nacional como al de exportación. Su planta industrial se encuentra ubicada en Ciudad Guayana, en la zona industrial Matanzas, sobre el margen derecho del río Orinoco, a 17 km. de confluencia con el río Caroní, y a 300 Km. de la desembocadura del Orinoco en el Océano Atlántico. Está Conectada con el resto del país vía terrestre, y con el resto del mundo vía fluvial-marítima. Se abastece de Energía eléctrica generada en las represas de Macagua y Guri, sobre el río Caroní, así como de gas natural proveniente de los campos petroleros del oriente venezolano. Sus instalaciones se extienden sobre una superficie de 2.200 hectáreas, de las cuales 87 son techadas.



Fig. 1
Mapa de la ubicación geográfica de la Planta de SIDOR, C. A.

Figura 1 Ubicación Geográfica de SIDOR.

La ubicación de las principales instalaciones de SIDOR puede observarse en la figura 2, tomada del folleto SIDOR publicado por la Dirección de Relaciones Institucionales:

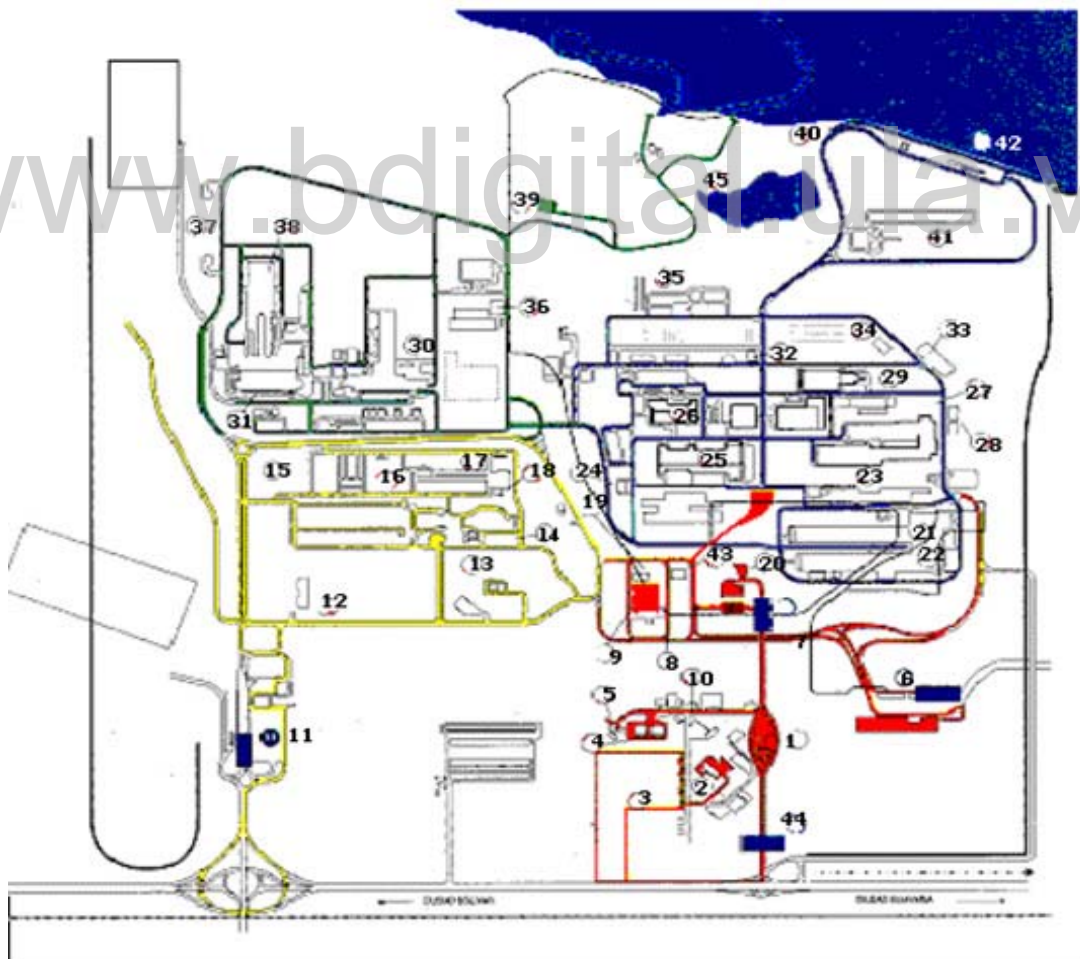


Figura 2. Layout de la Empresa

Leyenda:

- | | | | |
|---------------------------|-----------------------------|-----------------------------|-------------------------|
| 1. Edif. Administrativo I | 2. Centro de investigación | 3. Urb. Mapanare | 4. Fábrica de comida |
| 5. FAC | 6. Portón 3 | 7. Portón 2 | 8. Salud ocupacional |
| 9. Edif. Bomberos | 10. Edif. Administrativo II | 11. Portón 4 | 12. Serv. al personal |
| 13. Planta de aire IV | 14. Planta de pellas | 15. Edif. Midrex II | 16. Planta Briquetas |
| 17. Planta HyL II | 18. Edif. HyL II y Pellas | 19. Edif. centro de control | 20. Entrenamiento |
| 21. Planos en frío | 22. Planos recubiertos | 23. Planos en caliente | 24. Equipo móvil |
| 25. Fábrica de tubos | 26. Fundición | 27. Gasómetro | 28. Alm. Prod. Term. |
| 29. Almacén general | 30. Acería Planchones | 31. Acería Palanquillas | 32. Central termoelect. |
| 33. Planta de aire III | 34. Patio carbón | 35. Midrex II y HyL I | 36. Alm. prod. químicos |
| 37. Planta de chatarra | 38. Barras y alambrón | 39. Planta trat. aguas neg. | 40. Estación bombeo |
| 41. Planta de cal | 42. Muelle | 43. Edif. Producción | 44. Portón 1 |

1.2 Proceso de Laminación

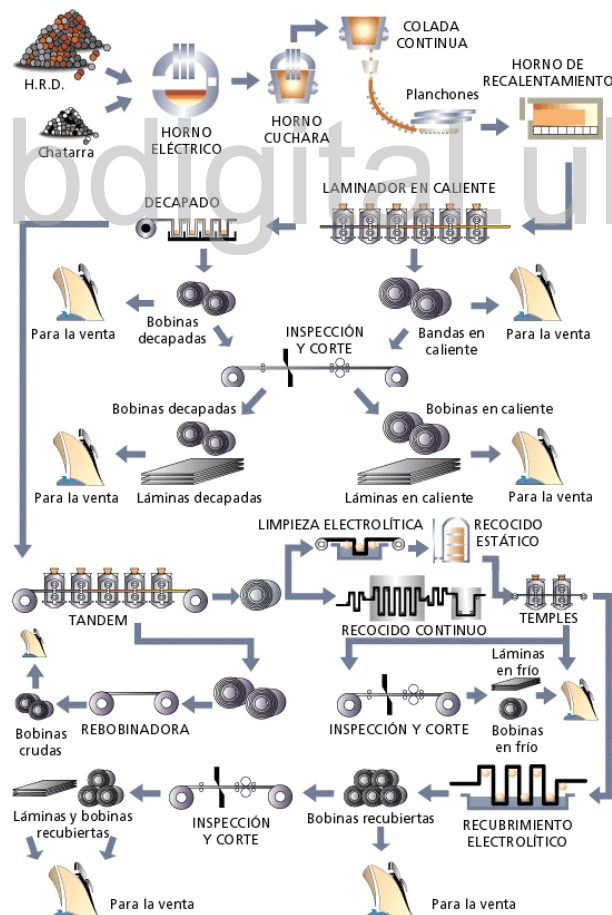


Figura 3 Procesos Planos

1.2.1 Proceso de Laminación en Caliente.

Descripción:

Los planchones producidos en las acerías de planchones de SIDOR, son colocados en la vía de rodillos de la mesa de los hornos de vigas galopantes y/o trasladadas mediante grúas a la mesa desapiladora de los hornos de empuje. De allí son trasladados a los hornos de calentamiento y una vez alcanzada la temperatura de laminación, los planchones son descargados en la vía de rodillos del Tren IV Reversibles donde de acuerdo al programa establecido, se le da la forma y dimensión a cada producto (bandas y chapas) en el laminador; pasado según su destino a la mesa de enfriamiento (chapas) y/o al Tren Continuo y enrolladores, desde donde salen convertidos en bobinas para posteriormente ser enviados a los patios de almacenamiento y/o plataformas para su enfriamiento.

1.2.2 Línea de Decapado I y II.

Descripción:

Las bobinas son tomadas por la grúa y colocadas en los bastidores desenrolladores, donde las bandas son cortadas en los extremos y soldadas entre sí (bobina entrante y saliente). De allí pasa a los tanques de desoxidación y enjuague donde la sometemos al corte de bordes y engrase. Por último, después de enrollado pasan a las áreas de almacenamiento.

1.2.3 Laminador Tándem I y II.

Descripción:

La sección de entrada al Tren de Laminación está equipada con una cadena transportadora tipo caballete, la cual está situada paralelamente a la línea y donde son quitados los flejes a las bobinas. Una vez en posición, las bobinas son llevadas mediante una viga transversal al centro de la línea de paso donde a través de un carro hidráulico es transportada a la estación de preparación y al cilindro de cambio. En este momento comienza el enhebrado, el cual consiste en pasar la banda a través de los 5 bastidores que componen el tren, controlando automáticamente los motores del laminador principal, a fin de asegurar una tensión uniforme en cada bastidor. La banda sobre un cilindro deflector recorre nuevamente desde el quinto (5º) bastidor a una mesa de banda y al cilindro de tensión estacionario. Seguidamente la correa regresa a su posición inicial

acelerando, posteriormente, el tren de laminación a su total velocidad enrolladora (1.800 Mtrs/min). El tren laminador se para cuando el enrollado finaliza y la banda completa está embobinada desde el mandril del cilindro de tensión y transferida por un carro de bobina accionado hidráulicamente a una cadena transportadora tipo caballete.

1.2.4 Línea de estañado y cromado electrolítico I.

Descripción:

Las bobinas son cargadas en el carro porta - bobinas y transportadas hasta el desenrollador, luego se procede a enhebrar la línea a baja velocidad y la banda va pasando por una serie de tanques y procesos químicos y físicos - químicos, de acuerdo al producto final (bobinas estañadas o cromadas), efectuando la limpieza, recubrimiento y enjuague de la misma. La banda continúa hasta el sistema de aceitado electrostático en donde es cubierta con una película de aceite para protegerla de los abrasivos. Por último la banda sigue hasta el enrollador de salida pasando por la cizalla en donde se le corta la cola a la banda y de esta manera obtener una bobina de peso requerido.

1.2.5 Línea de estañado y cromado electrolítico II.

Descripción:

Las bobinas son cargadas en el carro porta - bobinas y transportadas hasta el desenrollador, luego se procede a enhebrar la línea a baja velocidad y la banda va pasando por una serie de tanques y procesos químicos y físico - químicos, efectuándose la limpieza, recubrimientos y enjuague de la misma. La banda continúa hasta el sistema de aceitados electrostático en donde es cubierta con una película de aceites para protegerla de los abrasivos. Por último la banda sigue hasta el enrollador de salida pasando por la cizalla en donde se le corta la cola a la banda y de esta manera obtener una bobina de peso requerido.

1.3 Ubicación del proyecto dentro de SIDOR

La figura 4 muestra el Organigrama jerárquico de las gerencias de SIDOR. Dentro de la Dirección industrial se encuentra la Gerencia General de Ingeniería y Medio ambiente, donde a su vez esta localizada la gerencia de Automatización. (Ver Figura 5).

Conceptualmente esta gerencia (Automatización) mantiene un esquema piramidal de funcionamiento que se distribuye de la siguiente manera:

- Nivel 0.** Elementos de campo
- Nivel 1.** Red de control en campo
- Nivel 2A.** Red de supervisión y control.
- Nivel 2B.** Explotación de Datos.

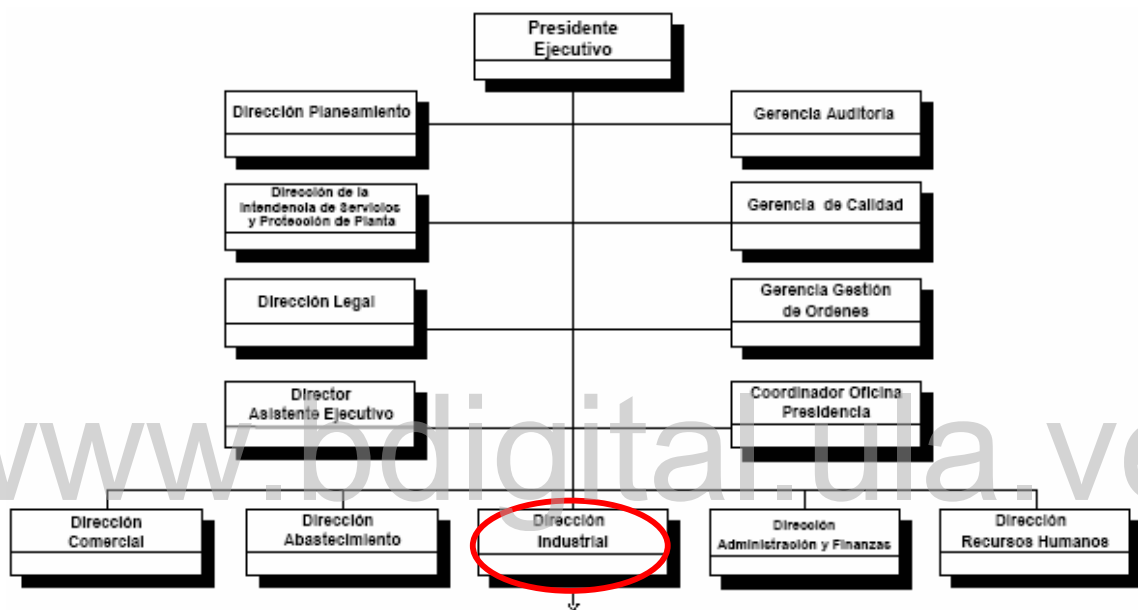


Figura 4 Organigrama Gerencial de Sidor

El Nivel 0, esta constituido por el conjunto de sensores y actuadores instalados en la planta. Este es el nivel mas bajo en la Pirámide de Automatización y solo obedece al nivel superior inmediato.

El Nivel 1, lo constituyen aquellos dispositivos en donde están programados los lazos de control, las condiciones de alarma, el manejo de los actuadores y la lógica secuencial según sea el caso del proyecto. A este nivel están conectados los instrumentos del Nivel 0.

El Nivel 2A, este nivel tiene como objetivo proporcionar sistemas o aplicaciones que permitan tener una visión clara del proceso al cual se presta este servicio. Para el logro de este objetivo se realiza el desarrollo de las aplicaciones que permiten la operación,

supervisión y monitoreo de variables de proceso, el ingreso de datos manuales de proceso y el almacenamiento de las variables en bases de datos para el control de la gestión del proceso.

El Nivel 2B, es el encargado de la explotación de datos del sistema disponibles en el Nivel 2A para su presentación por medio de tecnología WEB y para visualización de datos históricos de planta, así como también se encarga de publicar por medio de tecnología WEB las prácticas operativas y el manejo de los datos que tiene que ver con la gestión del proceso, seguridad y de calidad.

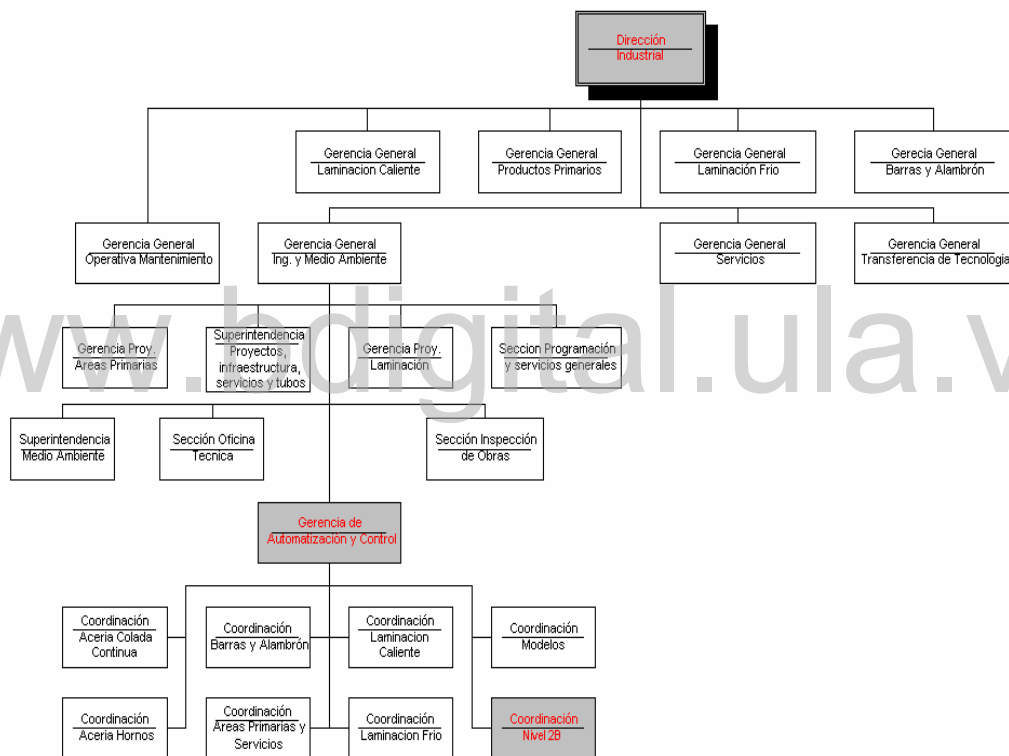


Figura 5. Dirección Industrial

El desarrollo del proyecto esta ubicado dentro del Nivel 2B, ya que como anteriormente se señaló, en esta área es donde se llevan a cabo los proyectos del tipo “Automatización vía Web”, los cuales vinculan gran numero de Aplicaciones accesibles a través de la Intranet.

1.4 Antecedentes

Actualmente en SIDOR, tanto la gerencia de laminación en frío, como la gerencia de laminación en caliente, no cuentan con un sistema de información, que les permita adquirir, almacenar y gestionar en forma estructurada, toda la información relacionada con los defectos presentes en las bobinas mientras se distribuyen por las diferentes líneas de laminación. Esto trae como consecuencia que los superintendentes, procesistas, inspectores, supervisores y jefes de plantas, no puedan de manera rápida y sencilla contar con toda esta información, lo que a su vez ocasiona dificultades a la hora de:

- analizar las causas de cada uno de los defectos,
- tomar acciones correctivas,
- realizar seguimientos sobre los defectos,
- ó tomar cualquier otro tipo de decisión.

Dadas todas estas circunstancias las respectivas gerencias de laminación, deciden solicitar a la gerencia de Automatización un sistema capaz de solventar todos los inconvenientes que vienen padeciendo las diferentes personas encargadas de velar por la buena producción de los productos laminados (Para este caso Bobinas). Es de esta solicitud de donde surge el SISTEMA DE GESTIÓN CUALITATIVA (SGC), el cual se encargará de llevar un control tanto de los defectos detectados en las bobinas como de las posibles causas de su producción.

1.5 Definición del Problema

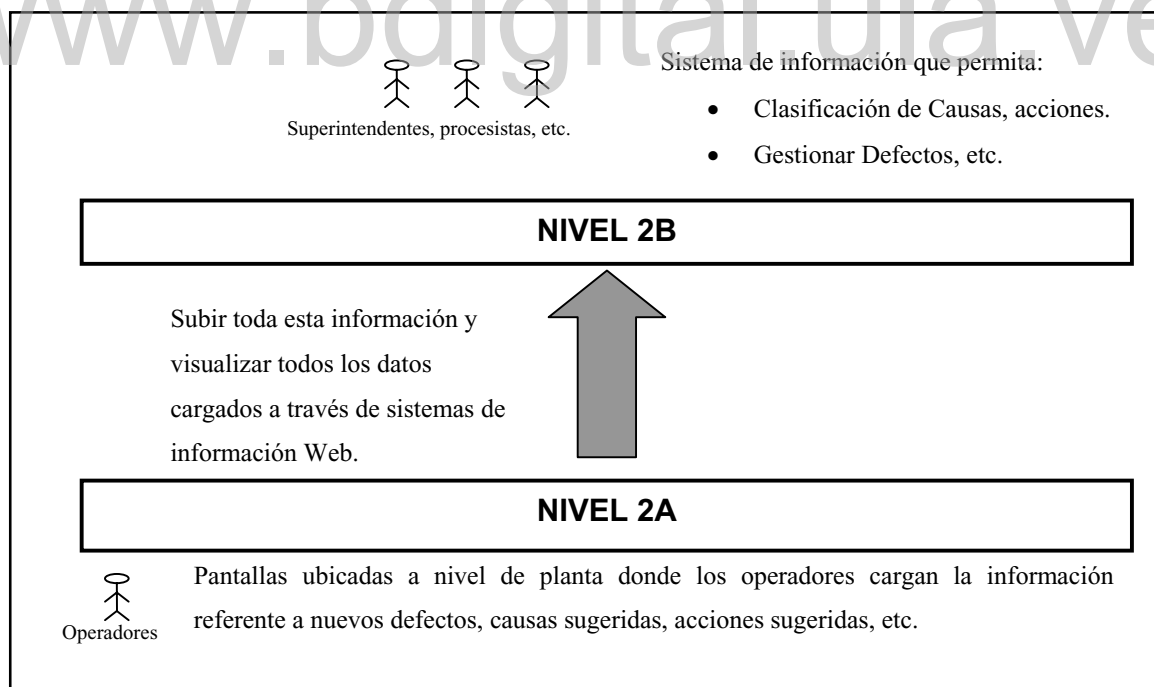
Toda la información que se genera al momento de producir una bobina a la que se le detecta un defecto, es de vital importancia para mejorar la calidad en los productos laminados. Por esta razón, es preocupante no contar con un sistema que permita almacenar toda esta información y luego utilizarla para mejorar fallas en los procesos, ya que para solucionar estas fallas y mejorar la calidad de producción es indispensable analizar el porque de estas fallas, asignar causas productoras de estos defectos y a su vez

asignarles acciones a esas causas, para que de esta manera se puedan evitar futuras ocurrencias del defecto.

Actualmente no existe un sistema que permite llevar todo este control de los defectos y evitar futuras ocurrencias de los mismos, lo cual es una de las posibles causas que puede estar ocasionando la ocurrencia de defectos similares en las bobinas. Por tal motivo lo que se quiere es un sistema que permita corregir ocurrencia de defectos similares en las bobinas a través de:

- Asignación de responsables por cada defecto que se detecte en las bobinas.
- Asignación de causas y acciones a cada defecto detectado.
- Cambio de cualquier estado de las acciones tomadas.
- Control de las causas y acciones asignadas a las bobinas por parte de los Superintendentes de las líneas.

El siguiente diagrama ilustra en líneas generales el problema:



1.6 Objetivos

Los Objetivos que se esperan alcanzar durante la realización de este proyecto, reflejan la característica del sistema.

1.6.1 Objetivo General:

Analizar, diseñar e implementar un sistema de información Web, que permita gestionar todos los defectos que se detectan a nivel de planta en las bobinas durante su producción.

1.6.2 Objetivos Específicos:

- **Utilizar el proceso Unificado de Desarrollo de Software** para llevar a cabo todo el análisis y diseño del sistema de información.
- **Implementar el sistema**, lo cual implica:
 - Desarrollar todas las pantallas las cuales permitan:
 - Revisión de cada una de las bobinas defectuosas por parte de los inspectores, supervisores o jefes de planta.
 - Asignar responsables a bobinas defectuosas para su procesamiento.
 - Asignar causas a los defectos.
 - Asignar acciones a las causas.
 - Llevar un control por parte de los superintendentes de las bobinas a las cuales se les fue asignado responsables.
 - Llevar un control por parte de los superintendentes de los defectos a los cuales se les fue asignado causas y acciones.
 - Programar todos los componentes incluidos en cada una de las pantallas, los cuales se encargan de la ejecución de la lógica de funcionamiento del sistema.
 - Realizar todos los procedimientos almacenados, funciones y vistas necesarias para el acceso a los datos, lo cual implica manejo de SQL.
 - Realizar todas las pruebas de aceptación.
 - Realización de manuales de Usuarios del sistema.

1.7 Metodología

Durante el desarrollo de este proyecto se utilizó como metodología de desarrollo, el Proceso Unificado (UP).

1.7.1 Proceso Unificado (UP).

El UP fomenta muchas buenas prácticas, pero la más importante es el **Desarrollo Iterativo**. En este enfoque, el desarrollo se analiza en una serie de mini-proyectos, llamados **iteraciones**; el resultado en cada uno es un sistema que puede ser probado, integrado y ejecutado. Cada iteración incluye sus propias actividades de análisis de requisitos, diseño, implementación y pruebas. UP incluye dentro de su estructura el **Lenguaje de Modelado Unificado (UML)**, para escribir análisis de requisitos, así como también establecer muchas etapas en el análisis y diseño del sistema. Un proyecto UP organiza el trabajo y las iteraciones en cuatro fases fundamentales [Larman 03]:

1. **Inicio:** visión aproximada, análisis del negocio, alcance, estimaciones imprecisas.
2. **Elaboración:** Visión refinada. Implementación iterativa del núcleo central de la arquitectura, resolución de los riesgos altos, identificación de mas requisitos y alcance, estimaciones mas realistas.
3. **Construcción:** implementación iterativa del resto de requisitos de menor riesgo y elementos más fáciles, preparación para el despliegue.
4. **Transición:** Pruebas beta, despliegue.

1.8 Alcance del proyecto

La finalidad de este proyecto es el de analizar, diseñar un sistema de información Web, el cual debe cumplir con una serie de requisitos relacionados con la gestión de la calidad de los productos laminados. Luego se debe implementar el diseño en un sistema accesible a todos los usuarios a través de la intranet de SIDOR. El sistema debe ser desarrollado utilizando para ello el siguiente software:

- ASP.NET para el desarrollo de las pantallas del sistema,

- Visual Basic. NET para la programación de cada uno de los componentes de usuario de las pantallas especificadas y,
- SQL SERVER 2000 como manejador de Base de Datos.

La base de datos a la cual esta ligada el desarrollo de este proyecto, debe ser de tipo relacional (desarrollada por personas de la empresa responsables del sistema). El sistema estará soportado sobre *Microsoft® Windows*.

1.9 Estructura del documento

El resto de la estructura de este documento será como sigue:

- **Capítulo 2: Marco Teórico**

En el que se describirá detalladamente la metodología (Proceso Unificado), conceptos básicos y demás temas necesarios para una comprensión clara del desarrollo del proyecto.

- **Capítulo 3: Fase de Inicio**

En este capítulo se comienzan a aplicar las fases del Proceso Unificado (UP), comenzando por la fase de inicio. En el se detallan algunos casos de uso, resultado de los primeros requerimientos de los usuarios del sistema.

- **Capítulo 4: Fase de Elaboración**

En este capítulo se realiza la mayor parte del análisis y diseño del sistema, se llevan a cabo nuevos talleres de requisitos, donde se obtienen como resultado nuevos requerimientos por parte de los usuarios, lo que implica la aparición de nuevos Casos de Uso. También en esta fase se construyen los diferentes diagramas que conforman el modelo de Casos de Uso, Modelo de Dominio y Modelo de diseño del sistema.

- **Capítulo 5: Fase de Construcción**

En este capítulo se lleva a cabo el diseño de la interfaz del sistema, esto implica diseño e implementación de las diferentes pantallas Web que conforman el sistema, también se especifica en forma general la interacción entre las tres capas que conforman la arquitectura de tres capas de la aplicación.

- **Capítulo 6: Fase de Transición**

En este último capítulo del documento se describe la implementación de las pantallas Web de sistema. También se describen algunas de las diferentes pruebas que se realizaron al sistema antes de la puesta en producción, y finalmente se especifica la puesta en producción del sistema.

- **Bibliografía.**
- **Anexos.**

www.bdigital.ula.ve

Capítulo 2

Marco Teórico

2 Introducción

En este capítulo se describen completamente todos los conceptos necesarios para el entendimiento y comprensión del proyecto. También se presenta a detalle la metodología de desarrollo (Proceso Unificado). De igual manera se mostrara un poco la plataforma .NET, sus componentes y funcionamiento. Se dará una breve descripción de ASP.NET, Visual Basic.NET y SQL Server, siendo estos el software utilizado durante el desarrollo del sistema.

2.1 Sistemas de Información

Un sistema de información es un conjunto de elementos que interactúan entre sí, con el fin de apoyar las actividades de una empresa o negocio. Un sistema de información realiza cuatro actividades básicas:

- Entrada de Información.
- Almacenamiento de Información.
- Procesamiento de Información.
- Salida de Información.

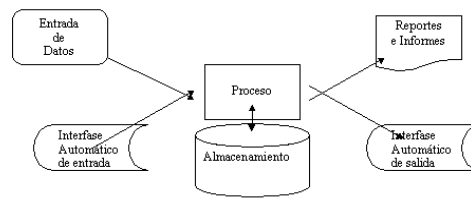


Figura 6. Fines del Sistema de Información

- Entrada de Información:** Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfases automáticas.
- Almacenamiento de información:** El almacenamiento es una de las actividades o capacidades más importantes que tiene un sistema de información, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en bases de datos ó en estructuras de información denominadas archivos.
- Procesamiento de Información:** Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información, la cual puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección a partir de los datos que contienen un estado de resultados o un balance general de un año base.
- Salida de Información:** La salida es la capacidad de un Sistema de Información de retornar la información procesada o bien la capacidad de explotar los datos de entrada al exterior. Es importante aclarar que la salida de un Sistema de Información puede

constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interfase automática de salida.

Los sistemas de información se desempeñan dentro del contexto del ambiente organizacional en el cual se desenvuelven, y transforman los diferentes elementos de entrada (datos, hechos o eventos) en información con un significado particular para los usuarios, permitiendo un mejor desenvolvimiento de las actividades y la toma de mejores decisiones en la organización. [Barrios 01]

Los Sistemas de Información pueden ser catalogados por sus componentes físicos como:

- **Tecnología Informática:** referida a los diferentes programas del sistema, de apoyo y de maquinas.
- **Recursos Humanos:** referida a las personas, tales como los usuarios, operadores, administradores y desarrolladores.
- **Programas de Aplicación:** referida al *software* o programas computacionales, la documentación y los estándares.
- **Datos Internos:** referida a los datos pertenecientes a la organización.
- **Datos Externos:** referida a los datos que provienen del entorno.

Los sistemas de información también pueden ser catalogados por sus componentes lógicos, relacionados con la recepción, procesamiento, almacenamiento, y validaciones de los datos como:

- **Procesamiento de transacciones:** se encarga de la localización de los datos y luego la colocación de los mismos, para realizar las transformaciones correspondientes.
- **Administración de Datos:** se encarga de la parte de mantenimiento, seguridad y control de los datos almacenados.
- **Producción de Información:** se encarga de la transformación de los datos, para presentarla a los usuarios y poder ejecutar las actividades de la organización de una manera más provechosa.

2.1.1 Objetivos de los Sistemas de Información

Los Sistemas de Información deben cumplir tres objetivos básicos dentro de las organizaciones:

1. Automatización de procesos operativos.
2. Proporcionar información que sirva de apoyo al proceso de toma de decisiones.
3. Lograr ventajas competitivas a través de su Implementación y uso.

Los Sistemas de Información que logran la automatización de procesos operativos dentro de una organización, son llamados frecuentemente **Sistemas Transaccionales**, ya que su función primordial consiste en procesar transacciones. Por otra parte, los Sistemas de Información que apoyan el proceso de toma de decisiones son los Sistemas de Soporte a la Toma de Decisiones, Sistemas para la Toma de Decisión de Grupo, Sistemas Expertos de Soporte a la Toma de Decisiones y Sistema de Información para Ejecutivos. El tercer tipo de sistema, de acuerdo con su uso u objetivos que cumplen, es el de los Sistemas Estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información.

2.2 El Proceso Unificado de Desarrollo de Software:

El proceso unificado es un proceso de desarrollo de software [JACOBSON 02]. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software (véase la figura 7)

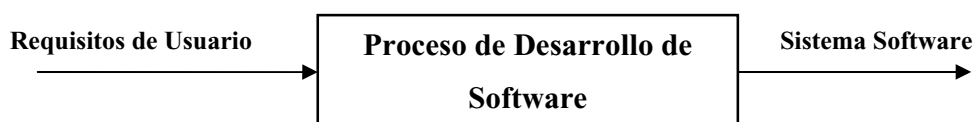


Figura 7. Proceso Entrada Salida

Sin embargo el proceso unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. El Proceso Unificado utiliza el *Lenguaje Unificado de Modelado* (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial del proceso unificado. No obstante, los verdaderos aspectos definitorios del Proceso Unificado se resumen en frases clave; dirigido por casos de usos, centrado en la arquitectura, e iterativo e incremental.

2.2.1 Ciclos de Proceso Unificado.

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema, como se muestra en la figura 8. Cada ciclo concluye con una versión del producto para los clientes.

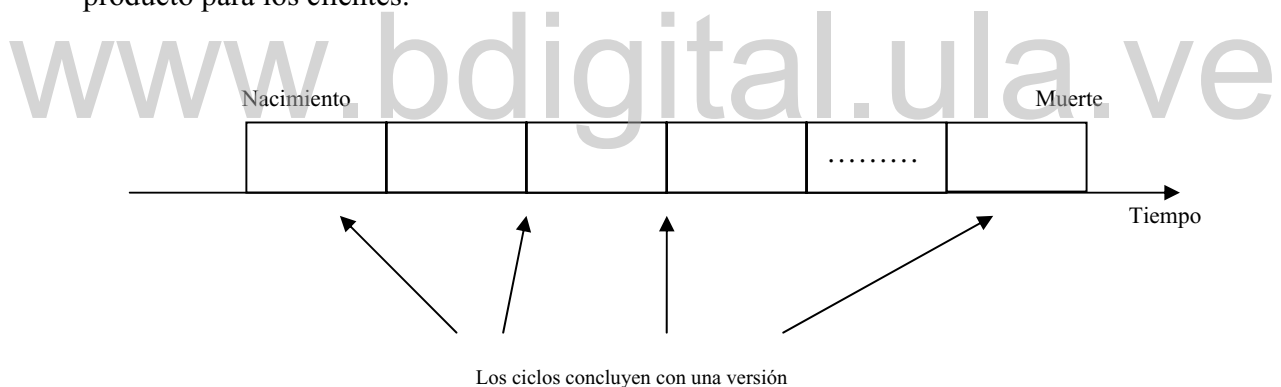


Figura 8. Ciclos del proceso Unificado.

Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones. Véase figura 9

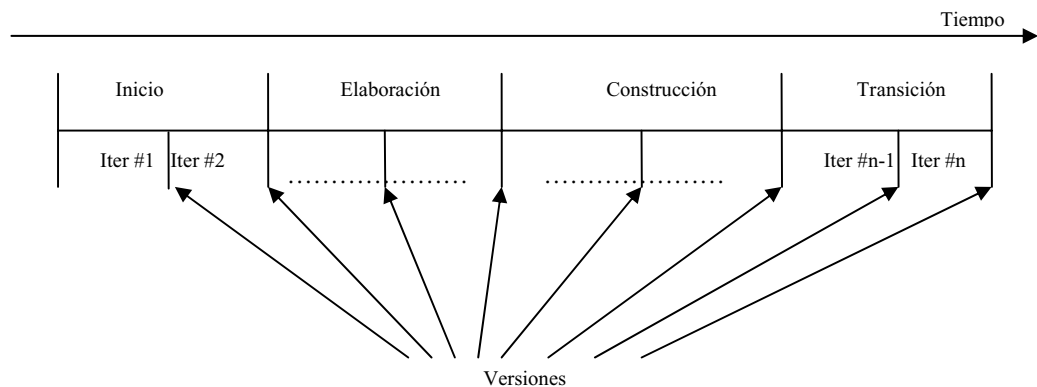


Figura 9. Fases del Proceso Unificado.

2.2.2 Fases dentro de un Ciclo.

Cada ciclo se desarrolla a lo largo del tiempo. Este tiempo a su vez se divide en fases, como se muestra en la figura 10. A través de una secuencia de modelos es posible visualizar lo que esta sucediendo en esas fases.

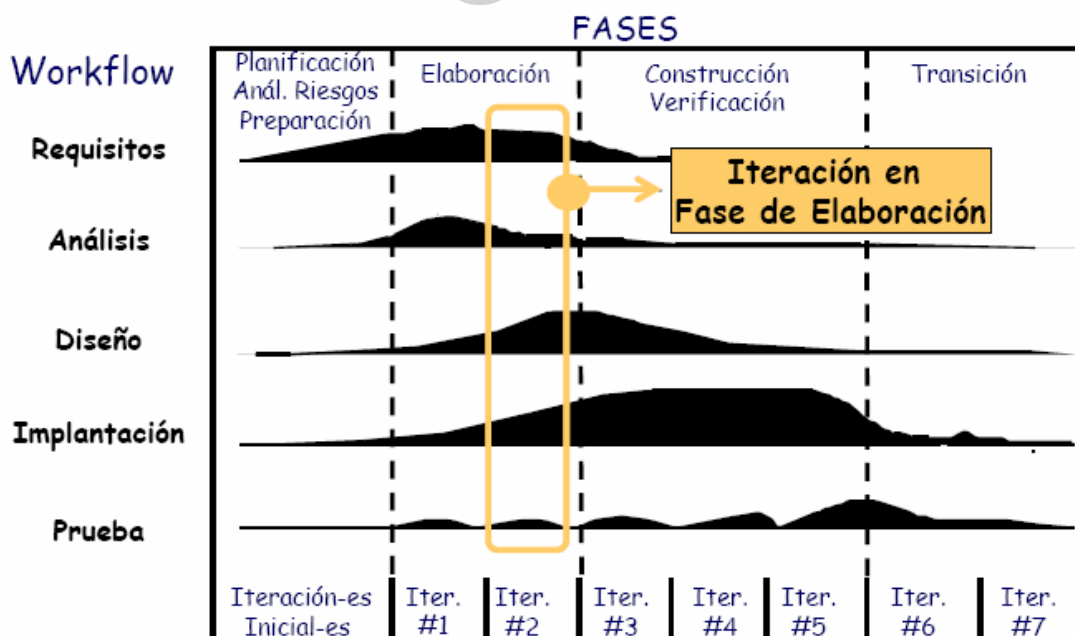


Figura 10. Fases

La figura 10 muestra en la columna izquierda los flujos de trabajo; requisitos, análisis, diseño, implementación y pruebas. Las curvas son una aproximación de hasta donde se llevan a cabo los flujos de trabajo en cada fase. Se debe recordar que cada fase se divide en iteraciones. Una iteración típica pasa por los cinco flujos de trabajo.

Durante la *Fase de Inicio*, se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el producto. En esta fase se responde a la pregunta; ¿Cuáles son las principales funciones del sistema para sus usuarios más importantes? La respuesta es que se encuentra en un modelo de casos de usos simplificados que contenga los casos de uso más críticos. En esta fase se identifican y priorizan los riesgos mas importantes, se planifica en detalle la fase de elaboración, y se estima el proyecto de manera aproximada.

Durante la *fase de elaboración*, se especifica en detalle la mayoría de los casos de uso y se diseña la arquitectura del sistema. La relación entre la arquitectura del sistema y el propio sistema es primordial. Por lo tanto, la arquitectura se expresa en forma de vistas de todos los modelos del sistema, los cuales juntos representan el modelo entero. Esto implica que hay vistas arquitectónicas del modelo de casos de uso, del modelo de análisis, del modelo de diseño, del modelo de implementación y el modelo de despliegue.

Durante la *fase de construcción*, se crea el producto. En esta fase la línea base de la arquitectura crece hasta convertirse en el sistema completo. La descripción evoluciona hasta convertirse en un producto preparado para ser entregado a la comunidad de usuarios. Al final de esta fase, el producto contiene todos los casos de uso que se han acordado para el desarrollo de esta versión. Sin embargo puede que no este completamente libre de defectos. Muchos de estos defectos se descubrirán durante la fase de transición.

La *fase de transición* cubre el periodo durante el cual el producto pasa de la versión beta a la versión final. Durante la versión beta un número reducido de usuarios prueba el sistema e informa de defectos y deficiencias. Esto hasta llegar a la versión final.

2.3 Lenguaje de modelado unificado (UML)

El lenguaje de modelado unificado, mejor conocido como UML (en inglés *Unified Modeling Software*), surgió a finales de 1995 por un grupo de personas entre las que destacan Grady Booch, Jim Rumbaugh e Ivan Jacobson. UML es un lenguaje visual estándar empleado para la especificación, construcción y documentación de *software* orientados a objetos, mediante la representación de los diversos elementos y procesos que interactúan de una u otra manera con dicho *software* [Quatrani 01] [OMG 03]. A continuación se listan una serie de definiciones utilizadas en UML:

- **Modelo de Casos de Uso:** El UP define el Modelo de Casos de Uso como el conjunto de todos los Casos de Uso; permite modelar la funcionalidad y entorno del sistema. [LARMAN 03]
- **Actor:** representa a un comportamiento o a una persona (identificada por un rol), sistema informatizado u organización. [LARMAN 03]
- **Escenario:** Es una secuencia específica de acciones e interacciones entre los usuarios y el sistema objeto de estudio; también se denomina *instancia de caso de uso*. [LARMAN 03]
- **Caso de Uso:** Es una colección de escenarios con éxitos y fallos relacionados, que describe a los actores utilizando un sistema para satisfacer un objetivo. Los casos de uso son requisitos; ante todo son requisitos funcionales que indican que hará el sistema. [LARMAN 03]
- **Diagramas de Casos de Uso:** En estos diagramas se muestran las distintas relaciones entre las diferentes funciones del sistema y los actores. [OMG 03] [Besembel 00].
- **Diagrama de secuencia del sistema (DSS):** Es un dibujo que muestra para un escenario específico de un caso de uso, los eventos que generan los actores externos, el orden y los eventos entre los sistemas. En estos tipos de diagramas los objetos son representados mediante rectángulos y son colocados en fila, cada uno de ellos posee una línea de tiempo vertical por debajo, y la relación entre los objetos (o mensajes) se expresa mediante una flecha que parte desde un punto de la línea de un objeto hasta llegar a la línea de tiempo de otro objeto [LARMAN 03] [OMG 03] [Quatrani 01] [Besembel 00]

- **Modelo del Dominio:** El Modelo del Dominio es la representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. A este tipo de modelos también se los denomina **Modelos Conceptuales, Modelo de Objetos del dominio y Modelos de Objetos de Análisis**.
- **Diagramas de Clases:** Los diagramas de clases expresan de manera general la estructura estática de un sistema, en términos de clases y de relaciones entre estas clases. Al igual que una clase describe un conjunto de objetos, una asociación describe un conjunto de enlaces; los objetos son instancias de clases y los enlaces son instancias de relaciones.

2.4 Bases de Datos

Las bases de datos son un conjunto de datos relacionados entre si, los cuales se almacenan o registran de manera ordenada con el fin de representar un aspecto del mundo real y poder ser manipuladas por los usuarios a quienes está dirigida. [Elmasri 97]. El conjunto de programas que permite crear y mantener la base de datos recibe el nombre de **Sistema de Gestión de Base de Datos (SGBD)**.

2.4.1 Sistema de Gestión de Base de Datos (SGBD):

Un sistema de gestión de base de datos (SGBD), consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. [Silberschatz 00] El primer objetivo de un SGBD es proporcionar un entorno que sea tanto práctico como eficiente de usar en la recuperación y el almacenamiento de la información de la base de datos. Los SGBD se diseñan para gestionar grandes cantidades de información. Los datos en una base de datos (BD), pueden estar localizados en una sola computadora (**BD centralizada**), o en varias computadoras (**BD distribuida**). Existen diferentes tipos de bases de datos:

- **Relacionales:** son las más utilizadas actualmente, su principal característica es que cada registro (llamado tupla o fila), se refiere a una información relacionada con un solo tema. En un Sistema de Gestión de Base de Datos Relacionales (SGBDR) se

gestionan todos los datos que tratan sobre un tema específico en tablas (llamadas relaciones), además permite cambiar datos (llamados atributos o columnas) provenientes de diferentes tablas. [Viescas 00].

- **Orientada a Objetos:** surgieron para darle respuesta a la necesidad de almacenar tipos de datos mas complejos como imágenes, sonido, video, y tipos personalizados, en aplicaciones mas complejas como las bases de datos de gráficos, sistemas geográficos, diseño y fabricación de ingeniería. Las características principales de este tipo de base de datos es que el diseñador de la misma puede especificar la estructura de los objetos complejos y las operaciones que se aplican sobre ellos, junto con las características propias de los lenguajes orientados a objetos [Elmasri 97]
- **Deductivas (o lógicas):** son bases de datos que a través de hechos y reglas buscan inferir o deducir nuevos conocimientos a partir de los datos existentes

En la actualidad existe un lenguaje para definir una base de datos (Lenguaje de definición de datos, o en ingles *data definition language* - DDL) y además manipular los datos que hay en ella (lenguaje de manipulación de datos, o en ingles *data manipulation language* - DML), este lenguaje se llama **Lenguaje Estructurado de Consulta**, mejor conocido como **SQL** (por sus siglas en ingles *Structured Query Language*).

2.5 La Plataforma .NET

Provee los cimientos para la nueva generación de software. Utiliza los Servicios Web como un medio para poder inter-operar a distintas tecnologías. Permite conectar distintos sistemas operativos, dispositivos físicos, información y usuarios. Da a los desarrolladores las herramientas y tecnologías para hacer rápidamente soluciones de negocios que involucran distintas aplicaciones, dispositivos físicos y organizaciones.

Que es la Plataforma .NET



Figura 11. Plataforma .Net

La idea central detrás de la plataforma .NET es la de servicio. Más concretamente **software como servicio** y de cómo construir, instalar, consumir, integrar o agregar (en federaciones) estos servicios para que puedan ser accedidos mediante Internet. El usuario de Internet puede con un explorador de Internet no solamente acceder a contenido como texto, imágenes o sonido, también puede hacer uso de servicios Web. Estos son los bloques de construcción o componentes sobre los cuales se basa el modelo de computación distribuida en Internet. La plataforma .NET permite usar Internet y su capacidad de distribución para que los usuarios accedan desde cualquier dispositivo, en cualquier sistema operativo y lugar a la funcionalidad que los servicios Web proveen. Los desarrolladores por su parte tienen la infraestructura y herramientas para crearlos y hacer uso de ellos en programas. Es decir, se trata de aprovechar la capacidad de distribución a gran escala de Internet para acceder a servicios de software.

2.5.1 El .NET Framework.

Es un conjunto de servicios de programación diseñados para simplificar el desarrollo de aplicaciones en el entorno altamente distribuido de Internet. Los componentes del .NET Framework proveen los "ladrillos" necesarios para construir las aplicaciones Web, los servicios Web y cualquier otra aplicación dentro de Visual Studio .NET.

2.5.2 Componentes del .NET Framework



Figura 12. Componentes de .NET Framework

2.5.2.1 Common Language Rutine:

Provee lo que se llama código administrado, es decir, un entorno que provee servicios automáticos al código que se ejecuta. Los servicios son variados:

- **Cargador de Clases:** Permite cargar en memoria las clases.
- **Compilador MSIL a nativo:** Transforma código intermedio de alto nivel independiente del hardware que lo ejecuta a código de máquina propio del dispositivo que lo ejecuta.
- **Administrador de Código:** Coordina toda la operación de los distintos subsistemas del Common Language Runtime.
- **Recolector de Basura:** Elimina de memoria objetos no utilizados.
- **Motor de Seguridad:** Administra la seguridad del código que se ejecuta.
- **Motor de Depuración:** Permite hacer un seguimiento de la ejecución del código aún cuando se utilicen lenguajes distintos.
- **Verificador de Tipos:** Controla que las variables de la aplicación usen el área de memoria que tienen asignado.
- **Administrador de Excepciones:** Maneja los errores que se producen durante la ejecución del código.

- **Soporte de multiproceso (threads):** Permite ejecutar código en forma paralela.
- **Empaquetador de COM:** Coordina la comunicación con los componentes COM para que puedan ser usados por el .NET Framework.

2.5.2.2 Biblioteca de Clases

La librería de clases base son las clases sobre las cuales se construyen todas las demás clases que utilizan los programas de Visual Studio .NET. La clase madre de todas es **System**. A partir de ella por un mecanismo llamado herencia de clases, se construyen las demás clases. Debido a que en la librería de clases base hay muchas clases, se utiliza para identificarlas un mecanismo llamado **espacio de nombres** (namespace). La parte del nombre de la clase que se encuentra a la derecha del último punto se llama **tipo de la clase**. Todo lo que resta se llama espacio de nombres. Por ejemplo: En la clase llamada **System.Runtime.InteropServices**, **InteropServices** es el tipo de la clase y **System.Runtime** es el espacio de nombre. El espacio de nombre es una manera de organizar en grupos las distintas clases. Esto hace más manejable y fácil su uso. La librería de clases base es independiente del lenguaje. Permite el uso y la depuración de otros lenguajes. Es extensible ya que por el mecanismo de herencia el usuario puede crear nuevas clases que usan las clases base como "ladrillos". También el usuario puede incorporarlas en bibliotecas para su utilización posterior. Es segura ya que es posible permitir o restringir su uso por medio de distintos mecanismos de seguridad.

2.5.3 Funcionamiento del .NET Framework.

Cuando se crea una aplicación en algún lenguaje compatible con la plataforma .NET, es posible utilizar cualquiera de los servicios que la biblioteca de clases de .NET. Cuando se compila la aplicación, se crea un código intermedio llamado MSIL. Este código es independiente de la plataforma de hardware. Una vez compilado, el ejecutor de lenguaje común administra la ejecución de la aplicación. Uno de los subsistemas del Common Language Runtime se llama compilación JIT, que transforma el código intermedio MSIL al código de máquina en el sistema donde la aplicación se va a ejecutar. Esta compilación a lenguaje de máquina lo hace en el momento de ejecución del código. Cuando un

dispositivo de cliente, ejecuta una aplicación hecha con Visual Studio .NET, se ejecuta en el código de máquina del sistema del cliente. La aplicación sin embargo puede interactuar con otras aplicaciones .NET y servicios independientemente del lenguaje en que fueron desarrollados.

2.6 Visual Basic .NET y ASP .NET:

Visual Basic .NET (algunas veces denominado VB. NET), es un lenguaje rediseñado Orientado a Objetos, que mejora el Visual Basic tradicional e incluso rompe la compatibilidad con los programas Visual Basic clásicos existentes. En resumen Visual Basic. NET es un lenguaje elegante, moderno e ideal para crear la siguiente generación de aplicaciones empresariales. En Visual Basic .NET a diferencia de lo que ocurría en las versiones anteriores de Visual Basic, sólo existe un tipo de fichero de código, el cual tiene la extensión **.vb**, en este tipo de fichero pueden coexistir distintos tipos de elementos, por ejemplo: un módulo de clase, un formulario, un módulo de código, un control, etc.; mientras que en las versiones anteriores de Visual Basic, cada uno de estos elementos tenían su propio tipo de fichero con su respectiva extensión. Por su parte El ASP.NET de Microsoft es una tecnología de scripts que corren en el servidor y pueden ser utilizados para crear aplicaciones dinámicas e interactivas en la Web. Una página ASP.NET es una página de HTML que contiene scripts que son procesados por un servidor Web antes de ser enviados al navegador del usuario. El código ASP.NET es más “compacto” que el código ASP; los scripts requeridos para realizar una función dada son más cortos en ASP.NET que en ASP. Debido a que los scripts que corren del lado del servidor están contruidos en una página regular de HTML, pueden ser entregados en casi cualquier navegador, Un archivo ASP.NET puede ser creado utilizando cualquier herramienta de edición, como el Notepad.

2.6.1 Aplicaciones ASP.NET

Al contrario de las aplicaciones de escritorios (que normalmente están en un único archivo .exe), las aplicaciones ASP.NET se dividen en varias páginas Web. Esta división implica que un usuario puede entrar en una aplicación ASP.NET desde diferentes puntos

y seguir un vínculo fuera de su aplicación aparte de su sitio Web u otro sitio Web. Cada aplicación ASP.NET individual comparte un conjunto de recurso y opciones de configuración común. Técnicamente hablando, cada aplicación ASP.NET se ejecuta dentro de un dominio de aplicación separado, lo que es burdamente similar a un <<proceso>> Windows en código no administrado normal. Los dominios de aplicación se aíslan en memoria, implicando que si una aplicación Web provoca un error fatal, no afectará el resto de aplicaciones actualmente en ejecución. Por consiguiente, la definición estándar de una aplicación ASP.NET se describe como una combinación de archivos, páginas, manejadores, módulos y código ejecutable que puede invocarse desde un directorio virtual (y opcionalmente sus subdirectorios), de un servidor Web (véase la figura 13).

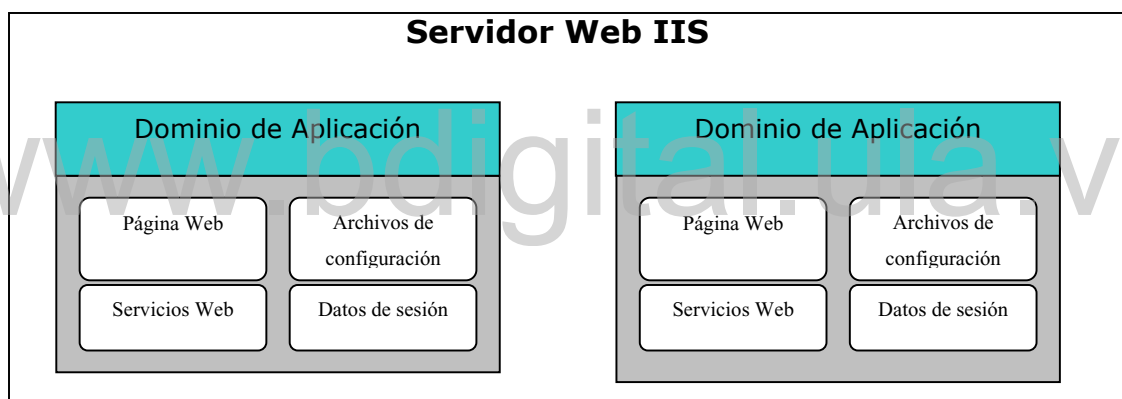


Figura 13. Aplicación ASP.NET

2.6.1.1 Funcionamiento de una aplicación ASP.NET

Cuando un servidor Web recibe una petición para mostrar una página Web estática, el servidor la envía directamente al navegador que la solicita. Cuando el servidor Web recibe una petición para mostrar una página dinámica, sin embargo, reacciona de forma distinta: transfiere la página a un software especial encargado de finalizar la página. Este software especial se denomina servidor de aplicaciones. El servidor de aplicaciones (en aplicaciones .NET se trata de .NET Framework), lee el código de la página (páginas .aspx para aplicaciones ASP.NET), finaliza la página en función de las instrucciones del código

y elimina el código de la página. El resultado es una página estática que el servidor de aplicaciones devuelve al servidor Web, que a su vez la envía al navegador solicitante. Lo único que el navegador recibe cuando llega la página es código HTML puro (ver figura 14). El .NET Framework, permite trabajar con recursos del lado del servidor, como una base de datos. Por ejemplo, una página dinámica (.aspx) puede indicar al .NET Framework que extraiga datos de una base de datos y los inserte en el código HTML de la página. La instrucción para extraer datos de una base de datos recibe el nombre de consulta de base de datos. Un servidor de aplicaciones no se puede comunicar directamente con una base de datos porque el formato propietario de esta última impide que se descifren los datos, el servidor de aplicaciones sólo se puede comunicar con la base de datos a través de un controlador que actúe de intermediario con la base de datos: el software actúa entonces como un intérprete entre el servidor de aplicaciones y la base de datos.

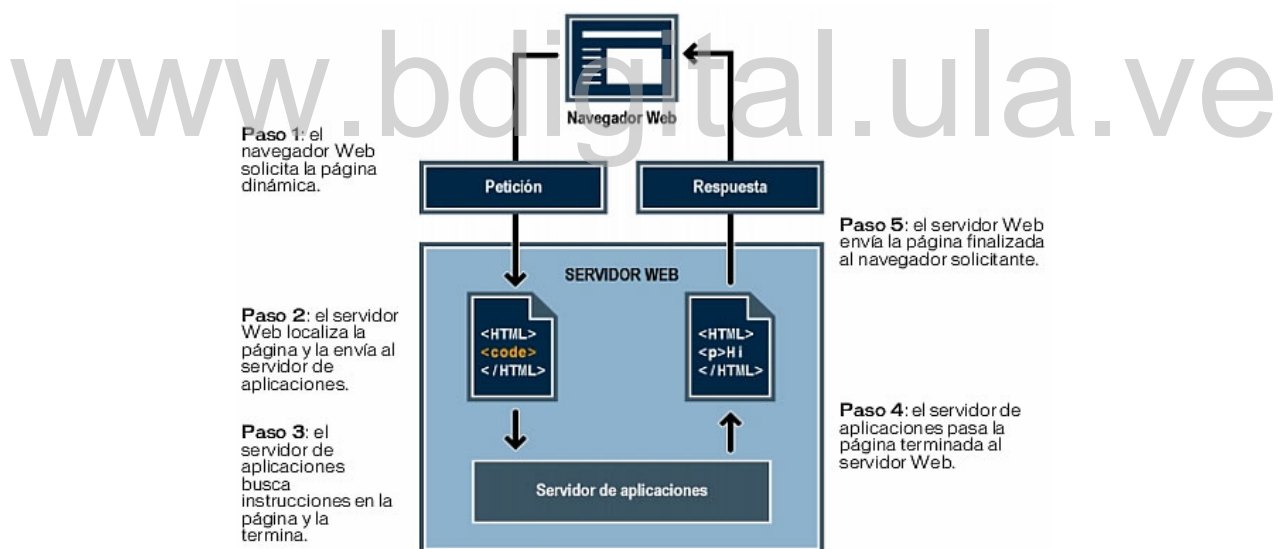


Figura 14. Funcionamiento de una aplicación ASP.NET

Una vez que el controlador establece la comunicación, la consulta se ejecuta en la base de datos y se crea un juego de registros. El juego de registros se devuelve al servidor de aplicaciones, que emplea los datos para completar la página (ver figura 15).

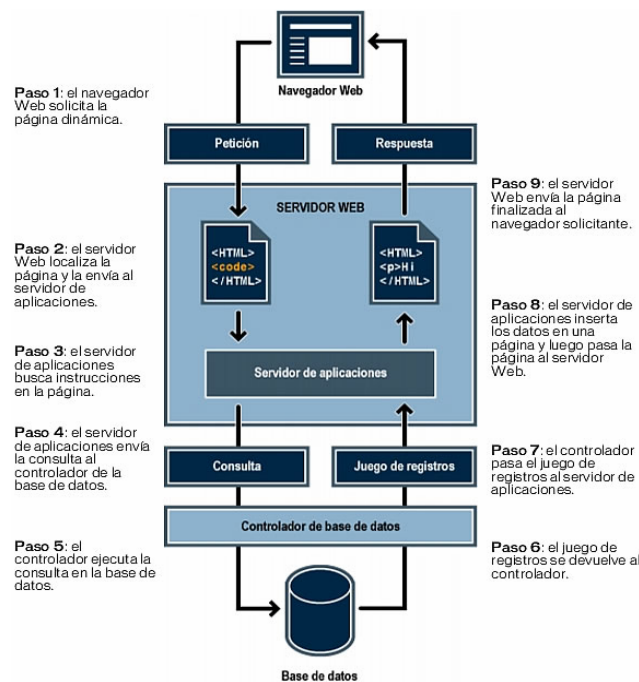


Figura 15. Funcionamiento de una aplicación ASP.NET con BD

2.7 XML Web Service

Un Web Service es una aplicación que proporciona datos y servicios a otras aplicaciones. Basado en XML, se apoya para su funcionamiento en una arquitectura de comunicaciones estándar entre el proveedor del servicio y el cliente. Esta comunicación se efectúa por medio de mensajes. Estos mensajes "empaquetados" mediante SOAP deben complementar las pautas definidas en el "contrato" de servicio entre el proveedor y el cliente (WSDL). Los Web Service, entre otras ventajas, son independientes de la plataforma y del lenguaje con el que se desarrollen, tanto las aplicaciones del proveedor del servicio como las del cliente del mismo. Servidores y clientes sólo deben entender los protocolos anteriormente mencionados y como estos fueron ampliamente aceptados en las distintas plataformas hasta convertirse en estándares, hoy es posible lograr una integración multiplataforma con implementaciones totalmente transparentes en cada extremo.

2.7.1 Funcionamiento de los XML Web Service

La siguiente figura ilustra el funcionamiento de un Web Service:

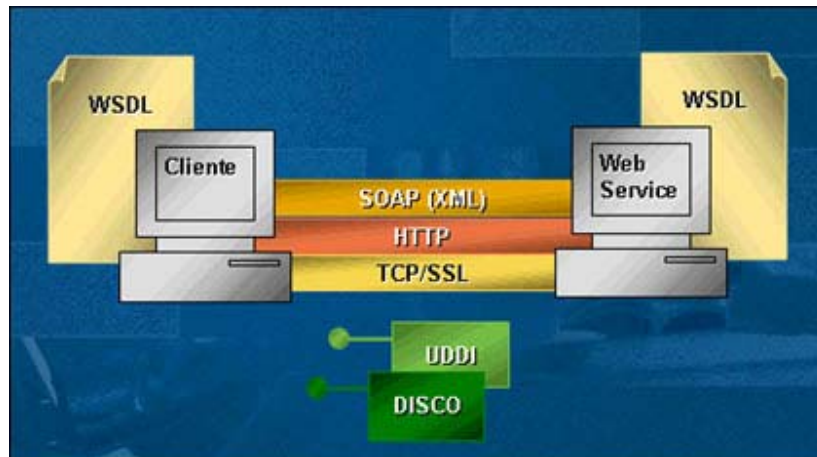


Figura 16. Funcionamiento de un Web Service

2.7.1.1 Simple Object Access Protocol (SOAP):

Protocolo de intercambio de información estructurada basado en XML. Entre otras cosas, SOAP puede representar llamadas a procedimientos remotos (RPC) y sus respuestas, proveyendo un mecanismo para intercambiar información y servicios entre aplicaciones. Funciona normalmente sobre HTTP así que trabaja con los *firewalls*, los *proxies* y otros obstáculos existentes.

2.7.1.2 WSDL:

Documento XML que describe completamente el formato de los mensajes intercambiados. Se puede consultar libremente por los clientes y representa el "contrato" de servicio.

2.7.1.3 Web Service Discovery (DISCO):

Es el proceso de localización y descubrimiento de las descripciones de servicios Web. A través de este proceso los clientes descubren la existencia del servicio, la funcionalidad ofrecida y como interactuar con él.

2.7.1.4 UDDI Business Registry:

Los nodos UDDI permiten que las organizaciones publiquen información relativa a los servicios que ofrecen. Constituyen el punto de entrada a los servicios Web, proporcionando un mecanismo sencillo de localización de los servicios disponibles. Como parte de la especificación UDDI, se dispone de una aplicación que permite interactuar programáticamente con el registro.

2.8 Microsoft SQL SERVER 2000

Microsoft SQL SERVER 2000 es un sistema de gestión de base de datos relacionales (SGBD o RDBMS; Relational Database Management System) cliente/servidor de alto rendimiento [Kalen Delaney].

2.8.1 Motor de SQL SERVER

El motor de SQL SERVER está diseñado para admitir una gran variedad de aplicaciones exigentes, como el procesamiento de transacciones en línea (OLTP; On Line Transaction Processing) y las aplicaciones de ayuda en la toma de decisiones. En el núcleo de estas capacidades de toma de decisiones está Transact-SQL, la versión de Microsoft del Lenguaje de consulta estructurado (SQL; Structured Query Language). Transact-SQL: Denominado a veces T-SQL, es un superconjunto potente del SQL estándar. El lenguaje Transact-SQL cumple con la norma SQL-92 del instituto Nacional Americano de Normalización (ANSI: American National Standards Institute) en su nivel de entrada, aunque ofrece mucha mas potencia gracias a sus extensiones exclusivas a la norma. Por ejemplo Transact-SQL proporciona construcciones de programación (como variables,

operaciones condicionales [IF-THEN-ELSE] y bucles) que pueden ayudar a simplificar en gran medida el desarrollo de aplicaciones.

2.8.2 Procedimientos almacenados y funciones

Dicho de manera sencilla, los procedimientos almacenados y las funciones son conjuntos de instrucciones SQL guardados dentro de una Base de Datos accesibles desde una aplicación frontal [Kalen Delaney]. Los procedimientos almacenados y las funciones se distinguen entre sí por la forma de utilizarlos en el código. Los procedimientos almacenados llevan a cabo acciones y suelen contener todas las comprobaciones de errores asociadas con dichas acciones. Por otra parte una función devuelve un valor, y dicho valor puede utilizarse en cualquier lugar en que una instrucción SQL pueda emplear un único valor. Existen tipos especiales de procedimientos almacenados los cuales se denominan **Desencadenadores (triggers)**; este tipo de procedimientos almacenados a diferencia de los comunes son invocados o desencadenados automáticamente por el SQL Server. Los desencadenadores están asociados con partes concretas de los datos y se les llama automáticamente cuando se realiza un intento de modificar esos datos, independientemente de lo que haya provocado la modificación (una entrada de usuario o una acción de una aplicación).

Capítulo 3

Fase de Inicio

3 Introducción

El propósito de la fase de inicio es establecer una visión común inicial de los objetivos del proyecto, determinar si es viable y decidir si merece la pena llevar a cabo algunas investigaciones serias en la fase de elaboración. De antemano (Por parte de la Empresa), fue decidido que este proyecto se llevara a cabo sin ninguna duda, lo que implica que muchos de los *Artefactos* que se construyen en esta fase, referente a riesgos, costos, viabilidad, y algunas otras decisiones, serán obviados, implicando esto que la fase sea relativamente breve. Incluye solamente el análisis de requisitos de los primeros talleres de requisitos¹, para luego pasar a la fase de elaboración. Los Artefactos que se crearon en esta fase se especifican en la tabla 1:

<i>Artefacto</i> ²	<i>Comentario</i>
Visión General.	Describe brevemente los objetivos generales del sistema y un bosquejo general de la estructura del mismo.
Modelo de Casos de Uso.	Describe los requisitos funcionales y los no funcionales relacionados.
Especificación Complementaria.	Describe otros requisitos.
Glosario.	Terminología clave del sistema.

¹ Indica reuniones de discusión de requisitos.

² Estos Artefactos se completan solo parcialmente en esta fase, luego se refinan de manera iterativa en la fase de elaboración.

Plan de Iteración	Describe que hacer en la primera iteración de la fase de elaboración.
-------------------	---

Tabla 1. Artefactos de la fase de Inicio.

3.1 VISIÓN GENERAL:

El **Sistema de Gestión Cualitativa (SGC)**, es un sistema de información Web, el cual servirá de apoyo a todo el personal encargado de la gestión diaria que se realiza a las *Bobinas Defectuosas**, producidas en las diferentes *líneas de Laminación**. Este sistema residirá dentro de SIDOR ubicándose en los servidores Cluster de producción, localizados dentro de la Organización. La interacción del sistema con los usuarios será a través de computadores clientes (uno por cada usuario del sistema), los cuales se comunican con los servidores de producción por medio de la intranet privada de SIDOR (ver figura 17).

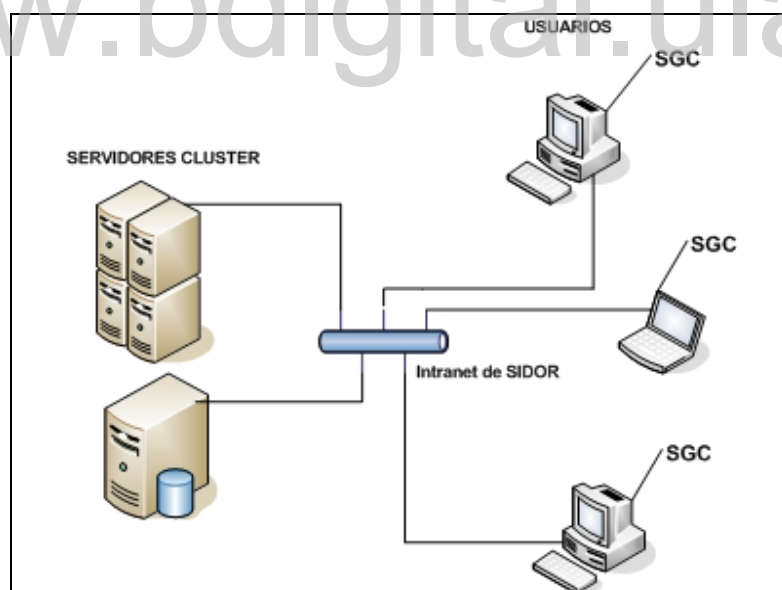


Figura 17. Visión general del Sistema

El sistema debe ser capaz de visualizar y permitir manipular los datos que se almacenan en Bases de Datos, y que proceden en tiempo real de las líneas de laminación. Estos datos

representan las *características cualitativas** de las bobinas defectuosas. Esto permitirá que todo el personal encargado pueda de una mejor manera llevar la gestión de los defectos de las bobinas. Además de esto podrán en tiempo real, ir gestionando cada una de las bobinas que se producen con algún defecto.

3.2 LISTA ACTOR-OBJETIVOS:

En esta lista se recogen los actores principales y sus objetivos, los cuales posteriormente se convierten en **Casos de Uso**. Se describe un Caso de Uso por cada objetivo de usuario, nombrándose de manera similar al objetivo de usuario.

ACTOR	OBJETIVOS
Jefe de Línea	<ul style="list-style-type: none"> • Revisar en un listado, bobinas defectuosas con sus respectivas características cualitativas. • Modificar algunas características cualitativas de las bobinas defectuosas. • Aprobar o rechazar como Jefe de Línea, las características cualitativas de las bobinas defectuosas.
Inspector	<ul style="list-style-type: none"> • Revisar en un listado, bobinas defectuosas con sus respectivas características cualitativas. • Modificar algunas características cualitativas de las bobinas defectuosas. • Aprobar o rechazar como Inspector de Línea, las características cualitativas de las bobinas

	defectuosas.
Supervisor	<ul style="list-style-type: none"> • Revisar en un listado, bobinas defectuosas con sus respectivas características cualitativas. • Modificar algunas características cualitativas de las bobinas defectuosas. • Aprobar o rechazar como Supervisor de Línea, las características cualitativas de las bobinas defectuosas.
Jefe de Procesos	<ul style="list-style-type: none"> • Analizar un listado de bobinas defectuosas. • Asignar responsables a cada una de las bobinas defectuosas.
Procesista	<ul style="list-style-type: none"> • Verificar cantidad de bobinas asignadas por Línea para su análisis. • Revisar listado con bobinas asignadas de una línea y defecto determinado. • Asignar causas para los defectos detectados en las bobinas defectuosas asignadas.

Tabla 2. Lista Actor – Objetivos.

3.3 MODELO DE CASOS DE USO.

El **Modelo de Casos de Uso** es básicamente el conjunto de todos los Casos de Uso que van surgiendo de la lista Actor – Objetivos; es un modelo de funcionalidad del sistema. Haciendo referencia a cada uno de los objetivos de usuario (tabla 2), se genera un

diagrama de Casos de Uso (Ver figura 18), el cual es una representación del contexto *Parcial* del sistema. Este diagrama se irá modificando a lo largo de las demás fases del UP, a medida que los requisitos vayan cambiando. Para la escritura de los casos de uso se usó la plantilla disponible en www.usecases.org la cual tiene un formato ampliamente extendido y además compartido. Esta plantilla se utiliza para los casos de uso principales, para los demás casos de uso se utiliza formato informal.

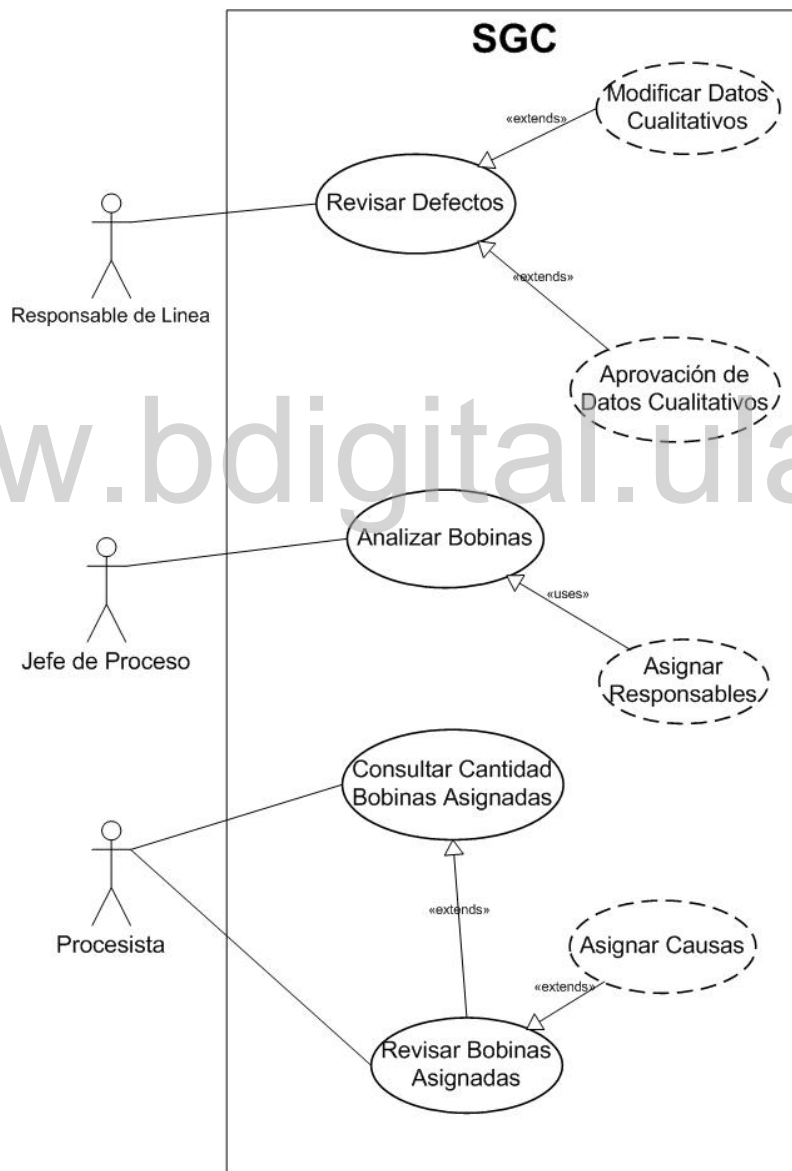


Figura 18. Diagrama de casos de uso parcial.

3.3.1 Caso de Uso UC1: Revisar Defectos.

Actores Principales:

- Responsable de Línea.

Personal Involucrado e Intereses:

- Inspector: Quiere revisar fácil y cómodamente, listas con las características cualitativas de las bobinas defectuosas, producidas en tiempo real en las diferentes líneas de laminación bajo su Inspección. Además quiere poder revisar bobinas defectuosas producidas en fechas anteriores.
- Supervisor: Quiere revisar fácil y cómodamente, listas con las características cualitativas de las bobinas defectuosas, producidas en tiempo real en las diferentes líneas de laminación bajo su Supervisión. Además quiere poder revisar bobinas defectuosas producidas en fechas anteriores.
- Jefe de Línea: Quiere revisar fácil y cómodamente, listas con las características cualitativas de las bobinas defectuosas, producidas en tiempo real en las diferentes líneas de laminación bajo su Coordinación. Además quiere poder revisar bobinas defectuosas producidas en fechas anteriores.

Precondiciones:

- El Actor debe estar previamente cargados en el sistema.

Garantías de Éxito (Poscondiciones):

- Se visualiza el listado correcto de bobinas defectuosas producidas, con sus respectivas características cualitativas.

Escenario Principal de Éxito (Flujo Básico):

1. El Actor selecciona la *Línea Origen* la cual quiere que tengan las bobinas defectuosas a mostrar.

2. El Actor selecciona un rango de fechas, dentro del cual desea que se encuentren las bobinas defectuosas, cuyas características cualitativas se visualizarán.
3. El Actor selecciona la *cuadrilla**, en la cual desea que estén las bobinas defectuosas a mostrar.
4. El Actor selecciona el o los *Orígenes**, para las bobinas defectuosas a mostrar.
5. El Actor selecciona el o los *Estados de Calidad**, para las bobinas defectuosas a mostrar.
6. Se Visualiza el listado de bobinas defectuosas con sus respectivas características cualitativas.

El Actor repite los pasos 2 – 6, cuantas veces considere necesarias.

Extensiones (o Flujo Alternativo):

*a En cualquier momento el sistema falla:

Se debe emitir un mensaje al usuario, y además de debe reportar el problema.

2a. El Actor selecciona un rango de fechas no válido:

1. El Sistema debe emitir un mensaje indicando el error y además deberá rechazar la selección de fechas y solicitar una nueva.

4a. El Actor no selecciona un Origen:

1. Emitir un mensaje indicando que se debe seleccionar un origen.

5a. El Actor no selecciona los Estados de Calidad:

1. Emitir un mensaje indicando que se debe seleccionar al menos un estado de calidad.

6a. No se visualizan los datos esperados:

1. Emitir un mensaje señalando que no existen datos para la entrada especificada.

Requisitos Especiales:

- Pantalla Web, con texto visible y claro al usuario, basada en los estándares de la empresa.
- Tiempos de respuestas para la visualización de los datos no mayor de 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

- 1-5a. Se usará el teclado o Mouse de computador para la selección de Línea Origen, Fechas, Cuadrilla, Origen y Estado de Calidad respectivamente.
- 6a. La pantalla Web se visualizará en monitores de 17” (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende de los *turnos*.

3.3.2 Caso de Uso UC2: Analizar Bobinas.**Actores Principales:**

- Jefe de Procesos.

Personal Involucrado e Intereses:

- Jefe de Procesos: Quiere visualizar en un listado bobinas defectuosas con sus respectivas características cualitativas para analizarlas.

Precondiciones:

- El jefe de procesos debe estar cargado en el sistema.

Garantías de Éxito (Poscondiciones):

- visualización correcta y efectiva del listado de bobinas defectuosas con todas las características cualitativas de las mismas.

Escenario Principal de Éxito (Flujo Básico):

1. El Jefe de procesos selecciona la Línea Origen.
2. El Jefe de procesos selecciona un rango de fechas dentro de los cuales desea que se encuentre el listado de bobinas defectuosas.
3. El Jefe de procesos selecciona el *defecto* * con el cual quiere que aparezcan las

bobinas del listado.

4. El jefe de procesos selecciona la(s) *aplicación*^{*}(es) que tendrán las bobinas del listado.
5. El Jefe de procesos selecciona el(los) estado(s) de calidad.
6. Se visualiza y analiza el listado deseado de bobinas defectuosas.

El Jefe de procesos repite los pasos 1 – 6 hasta que lo considere suficiente.

Extensiones (o Flujo Alternativo):

*a En cualquier momento el sistema falla:

1. Se debe emitir un mensaje al usuario y además se debe reportar el problema.
- 2a. El Jefe de Procesos selecciona un rango de fechas no válido:
 1. El Sistema debe emitir un mensaje indicando el error y además deberá rechazar la selección de fechas y solicitar una nueva.
- 4a. El jefe de procesos no selecciona ninguna aplicación:
 1. Automáticamente el sistema debe visualizar el listado sin tomar en cuenta la aplicación que tenga las bobinas defectuosas.
- 5a. El jefe de procesos no selecciona ningún estado de calidad:
 1. Automáticamente el sistema debe visualizar el listado sin tomar en cuenta el estado de calidad que tenga las bobinas defectuosas.
- 6a. No se visualiza el listado:
 1. Mostrar un mensaje indicando que no existen datos para la entrada especificada.

Requisitos Especiales:

- Pantalla Web, con texto visible y claro al usuario, basada en los estándares de la empresa.
- Tiempos de respuestas para la visualización de los datos no mayor de 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

- 1-5a. Se usará el Teclado o Mouse de computador para la selección de las entradas.

6a. La pantalla Web se visualizará en monitores de 17” (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende de los *turnos*.

3.3.3 Caso de Uso UC3: Consultar Cantidad Bobinas Asignadas.**Actores Principales:**

- Procesista.

Personal Involucrado e Intereses:

- Procesista: Quiere visualizar en un listado Línea – Defecto – Cantidad, el total de bobinas defectuosas que se le fueron asignadas para su gestión.

Precondiciones:

- El procesista debe estar cargado en el sistema

Garantías de Éxito (Poscondiciones):

- visualización correcta y efectiva del listado Línea – Defecto – Cantidad.

Escenario Principal de Éxito (Flujo Básico):

1. El procesista selecciona una línea de la cual quiere revisar la cantidad de bobinas asignadas.
2. El procesista visualiza el listado.

Extensiones (o Flujo Alternativo):

*a En cualquier momento el sistema falla:

1. Se debe emitir un mensaje al usuario, de manera tal que reporte el fallo.
- 2a. No se visualiza el listado:
1. Mostrar un mensaje indicando que no existen datos para la línea especificada.

Requisitos Especiales:

- Pantalla Web, con texto visible y claro al usuario, basada en los estándares de la empresa.
- Tiempos de respuestas para la visualización del listado no mayor de 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

- 1a. Se usará el Teclado o Mouse de computador para la selección de las entradas.
- 2a. La pantalla Web se visualizará en monitores de 17” (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende de los *turnos*.

3.3.4 Caso de Uso UC4: Revisar Bobinas Asignadas.**Actores Principales:**

- Procesista.

Personal Involucrado e Intereses:

- Procesista: Quiere revisar en un listado las características cualitativas de las bobinas las cuales le fueron asignadas, también quiere visualizar posibles causas responsables de los defectos presentados en las bobinas defectuosas.

Precondiciones:

- El procesista debe estar registrado en el sistema

Garantías de Éxito (Poscondiciones):

- Se visualizan las bobinas defectuosas asignadas con sus respectivas características cualitativas.
- Se visualizan causas anteriormente aplicadas (en caso de existir)

Escenario Principal de Éxito (Flujo Básico):

1. El procesista selecciona una línea específica de la cual quiere visualizar el listado de bobinas defectuosas.
2. El procesista selecciona un defecto específico, el cual quiere analizar y asignar causas.
3. se visualizan los listados de bobinas defectuosas asignadas y causas anteriormente aplicadas (en caso de que existan).

Extensiones (o Flujo Alternativo):

*a En cualquier momento el sistema falla:

1. Se debe emitir un mensaje al usuario, de manera tal que reporte el fallo.

Requisitos Especiales:

- Pantalla Web, con texto visible y claro al usuario, basada en los estándares de la empresa.
- Tiempos de respuestas para la visualización del listado no mayor de 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

- 1-2a. Se usará el Teclado o Mouse de computador para la selección de las entradas.
- 3a. La pantalla Web se visualizará en monitores de 17" (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende de los *turnos*.

3.3.5 Caso de Uso UC5: Modificar datos Cualitativos.**Actores Principales:**

- Responsable de Línea.

Escenario Principal de Éxito:

El Actor desea visualizar todos los datos cualitativos de una bobina defectuosa en específico, con el objeto de modificar algunos datos cualitativos a dicha bobina.

Escenarios Alternativos:

Si en cualquier momento el sistema falla; se debe emitir un mensaje al usuario y además se debe reportar el problema.

Si se detecta un error durante el registro de la modificación; Se debe emitir un mensaje de error notificando al usuario que no se registraron las modificaciones de los datos cualitativos.

3.3.6 Caso de Uso UC6: Aprobar datos Cualitativos.**Actores Principales:**

- Responsable de Línea.

Escenario Principal de Éxito:

El Actor firma (aprueba), los datos cualitativos de una bobina defectuosa en específico.

Escenarios Alternativos:

Si en cualquier momento el sistema falla; se debe emitir un mensaje al usuario y además se debe reportar el problema.

Si se detecta un error durante el registro de la firma; Se debe emitir un mensaje de error notificando al usuario que no se registró la firma de los datos cualitativos.

3.3.7 Caso de Uso UC7: Asignar Responsables.**Actores Principales:**

- Jefe de Procesos.

Escenario Principal de Éxito:

El Actor selecciona un responsable de una lista para cada una de las bobinas

defectuosas.

Escenarios Alternativos:

Si en cualquier momento el sistema falla; se debe emitir un mensaje al usuario y además se debe reportar el problema.

Si ocurre un error durante la asignación de los responsables; se debe emitir un mensaje indicando la ocurrencia del error, además de esto, el sistema debe impedir la asignación de responsables solicitada.

3.3.8 Caso de Uso UC8: Asignar Causas.**Actores Principales:**

- Procesista.

Escenario Principal de Éxito:

El Actor asigna correctamente a cada bobina una nueva causa o una anteriormente aplicada al mismo defecto, según sea el caso. Se registran las asignaciones de causas por *bobinas defectuosas** satisfactoriamente.

Extensiones (o Flujo Alternativo):

Si en cualquier momento el sistema falla; se debe emitir un mensaje al usuario y además se debe reportar el problema.

Si ocurre un error durante la asignación de las causas; se deberá mostrar un mensaje indicando el error. El sistema deberá rechazar la asignación y notificarlo. El sistema debe quedar disponible para intentar nuevamente la asignación de causas a defectos.

* Ver sección 3.5

3.4 ESPECIFICACIONES COMPLEMENTARIAS

3.4.1 Introducción:

Esta sección del documento es el repositorio de todos los requisitos del sistema que no se capturan en los Casos de Uso (Requisitos funcionales), es decir los **requisitos no funcionales**.

3.4.2 Detección y gestión de errores

Los errores ocurrentes en el sistema deben ocultarse al usuario, es decir se debe notificar de la ocurrencia del mismo, pero de manera agradable. El sistema deberá también notificar automáticamente al personal encargado del sistema la ocurrencia del error, de manera tal, que si no se corrige automáticamente, este investigue y lo solucione rápidamente.

3.4.3 Seguridad

Todo usuario debe estar previamente registrado en el sistema para su manejo.

3.4.4 Facilidad de Uso

El usuario será capaz de ver y manipular la información en Pantallas Web, visualizadas en monitores de 17". Por tanto:

- Se deben ver los listados de datos clara y fácilmente.
- Seguir los estándares de la empresa para la visualización de estas pantallas.

Velocidad, comodidad y procesamiento libre de errores, es lo más importante para los usuarios, durante el manejo del sistema. Procurar en lo posible emitir mensaje de errores y advertencias con sonidos, de manera que se evite cualquier despiste del usuario.

Fiabilidad

El sistema debe ser 100% fiable con respecto a la información que se visualiza en las listas de bobinas defectuosas, ya que dichas listas son de suma importancia para incrementar la calidad de las bobinas que se producen en las líneas de Laminación.


3.4.5 Rendimiento

Como se mencionó en el apartado Facilidad de Uso, los usuarios del sistema desean buen rendimiento al momento de visualizar listados de bobinas defectuosas como también al hacer cualquier tipo de registro y asignación de datos. El objetivo es conseguir esto en menos de 30 Segundos, el 90% de las veces.

3.4.6 Restricciones de Implementación

Debido a estándares de SIDOR, la solución del sistema se llevará a cabo usando tecnología .NET, en especial ASP.NET y Visual Basic. NET.

3.5 GLOSARIO

Término	Definición e Información	Alias
Bobina	 Producto plano de acero, suministrado en rollos.	
Bobina defectuosa	Bobina a la cual se le detecta un defecto a la hora de su producción.	
Defecto	Desperfecto que se producen en las bobinas a la hora de su producción. Todos los tipos de defectos que se manejan en las líneas de producción se encuentran estandarizados por una codificación especial.	C.D
Línea de Laminación	Maquinaria encargada de la producción y ajustes de las bobinas.	Línea
Línea Origen	Línea origen del defecto que se detecta.	L.O
Características cualitativas	Conjunto de datos característicos de las bobinas defectuosas	
Cuadrilla	Grupo de trabajo encargado de las líneas de laminación.	
Turno	Son horarios de trabajo establecidos por SIDOR para las cuadrillas.	

Origen	Es el origen del defecto detectado en la bobina defectuosa. Puede ser propio, derivado o asignado.	
Estado de Calidad	Es el estado que se le asigna a la bobina al momento de su producción.	E.C
Aplicación	Se refiere a la aplicación que se le da el personal de Calidad de SIDOR, a las bobinas que resultan defectuosas.	

Tabla 3. Glosario – Fase de Inicio

3.6 PLAN DE ITERACIÓN:

Con el Objeto de realizar un diseño e implementación “amplio y superficial” que incluyan elementos importantes del sistema, fue decidido abordar para la iteración I de la *fase de elaboración*, los escenarios principales de éxito de los casos de uso más importantes analizados en la fase de inicio. Estos son:

- Revisar defectos.
- Analizar bobinas.
- Consultar cantidad bobinas asignadas.
- Revisar bobinas asignadas.

Para ello se refinó un poco el Modelo de Casos de Uso, diseñando cada uno de los **Diagramas de Secuencias** para los escenarios principales de éxito de los Casos de Uso arriba especificados. De esta manera fue posible detallar los elementos de entrada y salida relacionados con el sistema en estudio.

Capítulo 4

Fase de Elaboración

4 Introducción

Para esta Fase del *Proceso Unificado*, se planificó tres iteraciones (según acuerdos con los líderes del proyecto dentro de la empresa), en las cuales se elaboraran las diferentes etapas que cada una de ellas conlleva; estas son: Análisis, Diseño e Implementación y pruebas. La etapa de Análisis esta comprendida por el refinamiento del Modelo de Casos de Uso, así como la elaboración del Modelo de Dominio. La etapa de Diseño esta representada por la construcción del Modelo de Diseño. Paralelamente a la elaboración de cada uno de los Modelos, estará la implementación respectiva de la aplicación.

En la **Iteración I** se analizan todos los casos de uso descritos en formato completo en la fase de inicio (Ver Capítulo 3), esto lleva al refinamiento del Modelo de Casos de Uso a través de la construcción de los **Diagramas de Secuencia**. Además se construye el Diagrama de Clases de Dominio, lo cual implica el comienzo del Modelado del Dominio. Una vez finalizada esta etapa de análisis, se comienza con el diseño de los Diagramas de Interacción, así como también la elaboración del Diagrama de Clases de Diseño, siendo estos dos últimos parte del Modelo de Diseño.

En la **Iteración II y III**, se llevan a cabo nuevos talleres de requisitos con el fin de capturar nuevos requerimientos de los usuarios del sistema, y por ende se refinan cada uno de los Modelos de Casos de Uso, Dominio y Diseño. Además del refinamiento de cada uno de los Modelos también se refina el Glosario del sistema con los nuevos

términos que fueron surgiendo. Los Artefactos³ que fueron creados en esta fase se especifican en la tabla 4:

<i>Artefacto</i> ⁴	<i>Comentario</i>
Modelo de Casos de Uso.	Describe los requisitos funcionales y los no funcionales relacionados.
Modelo de Dominio	Describe el Dominio del sistema.
Modelo de Diseño	Describe la interacción de cada uno de los objetos software del sistema y las clases software del mismo.
Glosario.	Terminología clave del sistema.
Plan de Iteración	Describe que hacer en las próximas iteración de la fase de elaboración.

Tabla 4. Artefactos de la fase de Inicio.

4.1 ITERACIÓN I.

4.1.1 MODELO DE CASOS DE USO.

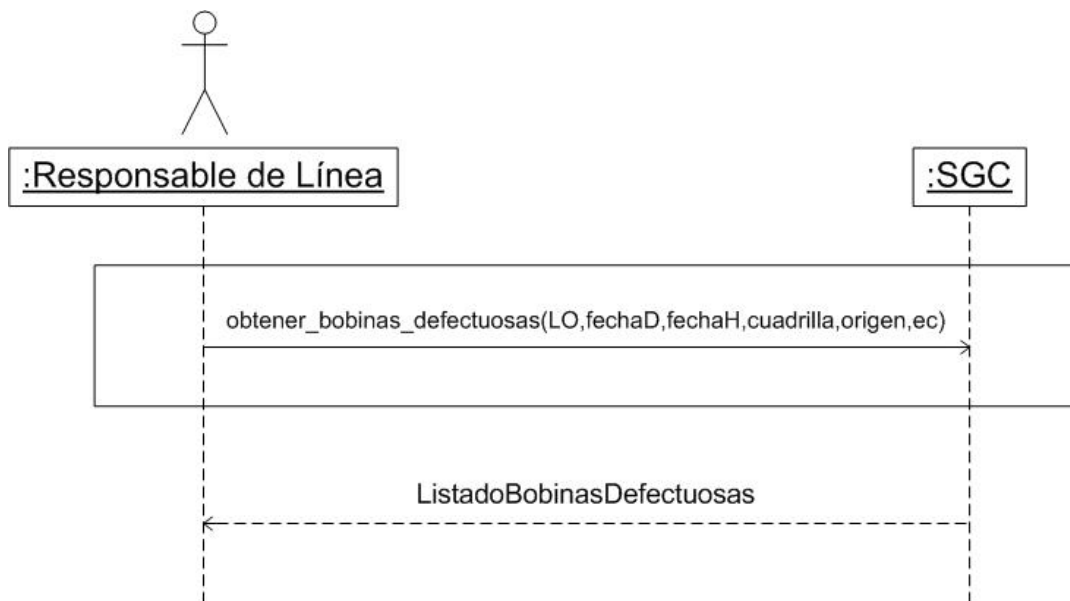
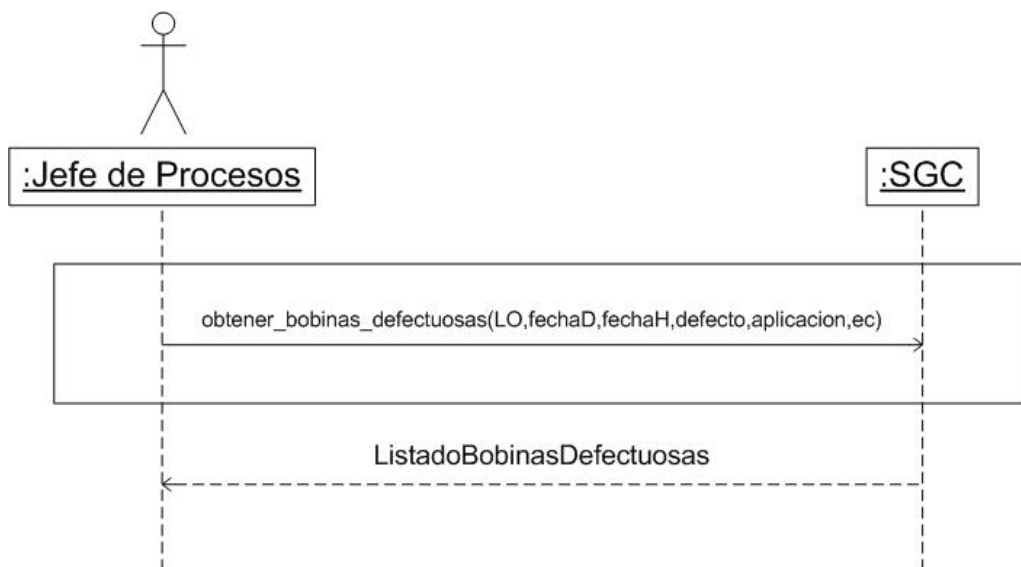
A continuación se especifican cada uno de los *Diagramas de Secuencias* (DSS), para los Casos de Uso seleccionados en el plan de iteración de la fase de Inicio (*Apartado 3.6 del capítulo 3*). El conjunto de estos diagramas ayuda al refinamiento de los Casos de Uso capturados en la fase de Inicio, al mejoramiento del Modelo de Casos de Uso y a mejorar la especificación del contexto parcial del sistema. Cada uno de estos diagramas ilustra la interacción del usuario (Definidos en la tabla 2) con el sistema, el cual desea satisfacer sus objetivos.

³ Etapas que se llevan a cabo dentro de una fase del UP.

⁴ Estos Artefactos se completan solo parcialmente en esta fase, luego se refinan de manera iterativa en la fase de elaboración.

- **DSS 1:** Este diagrama (figura 19) corresponde al escenario principal de éxito del *Caso de Uso UC 1: Revisar defectos (Apartado 3.3.1)*, donde el usuario introduce: línea origen, rango de fechas, cuadrilla, origen y estado de calidad, y el sistema como respuesta retorna un listado con las bobinas defectuosas deseadas.
- **DSS 2:** Este escenario (figura 20) es similar al del caso de uso *UC 1: Revisar Defectos*, con una diferencia respecto a los parámetros que debe introducir el usuario para cumplir con su objetivo. El diagrama ilustra el escenario principal de éxito para el *Caso de Uso UC 2: Analizar bobinas (Apartado 3.3.2)*. Aquí el usuario introduce la línea origen, rango de fechas, defecto, aplicación y estado de calidad, y el sistema retorna un listado cuyo contenido son bobinas defectuosas las cuales se analizarán y posteriormente asignarán responsables para su gestión.
- **DSS 3:** Este diagrama (figura 21) especifica como el usuario luego de introducir la línea origen, obtiene en un listado, la cantidad de bobinas defectuosas que les fueron asignadas para su gestión con sus respectivos defectos. Aquí se ilustra el escenario principal de éxito del *Caso de Uso UC 3: Consultar Cantidad Bobinas Asignadas (Apartado 3.3.3)*.
- **DSS 4:** Este diagrama (figura 22), corresponde al *Caso de Uso UC 4: Revisar Bobinas Asignadas (Apartado 3.3.4)*. Aquí el usuario especificando la línea y el defecto, podrá obtener un listado detallado con cada una de las bobinas asignadas para su gestión. Además de esto obtendrá causas anteriormente asignadas al defecto especificado en la solicitud (en caso de existir).

Las siguientes figuras corresponden a los diagramas de secuencia anteriormente descritos:

4.1.1.1 DSS 1: Escenario Revisar Defectos.**Figura 19. DSS 1 Revisar Defectos.****4.1.1.2 DSS 2: Escenario Analizar Bobinas.****Figura 20. DSS 2 Analizar Bobinas.**

4.1.1.3 DSS 3: Escenario Consultar Cantidad Bobinas Asignadas.

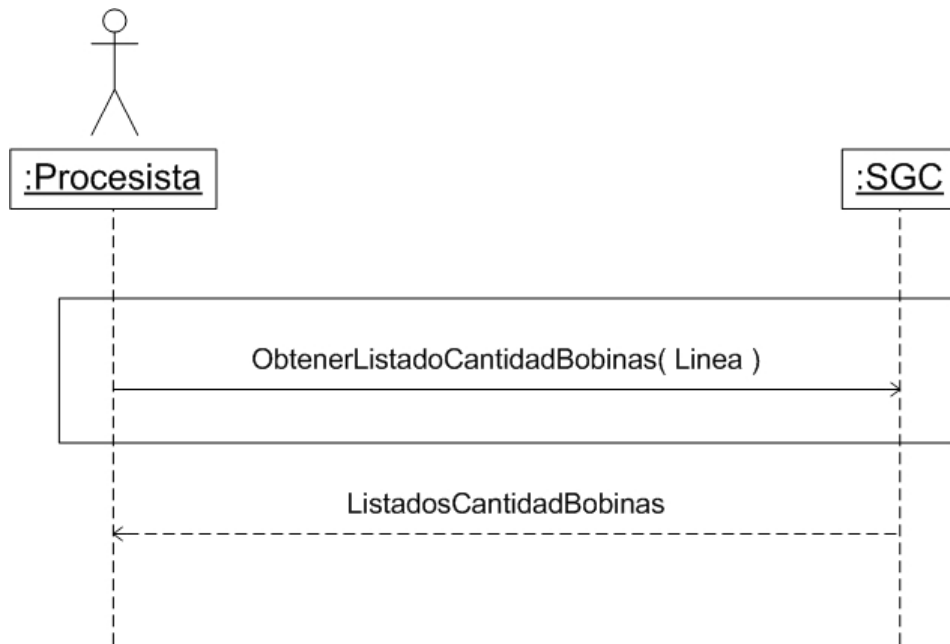


Figura 21. DSS 3 Consultar Cantidad Bobinas Asignadas.

4.1.1.4 DSS 4: Escenario Revisar Bobinas Asignadas.

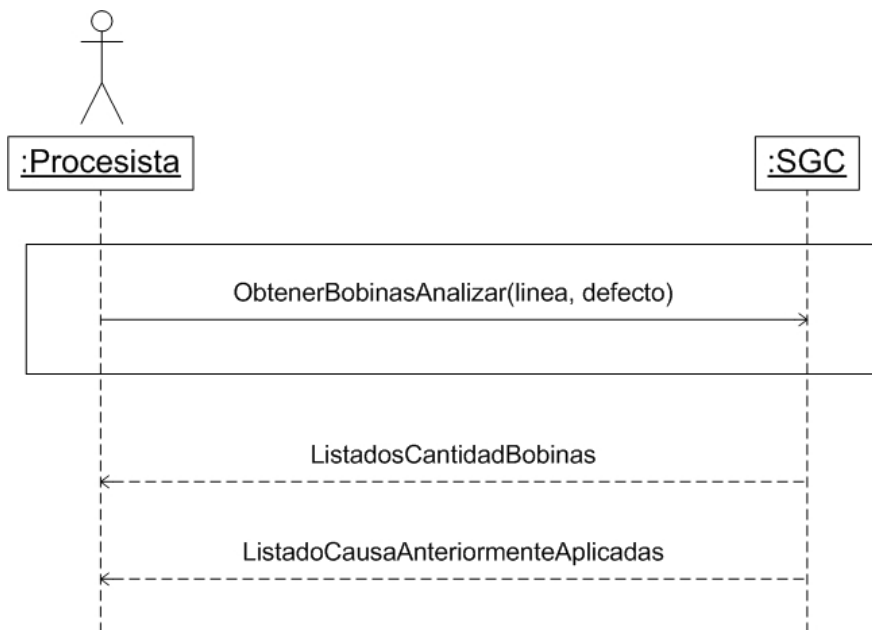


Figura 22. DSS 4 Revisar Bobinas Asignadas.

4.1.2 MODELO DEL DOMINIO.

Utilizando la notación UML, un Modelo del Dominio se representa con un conjunto de *diagramas de clases* en los que no se define ninguna operación. Este modelo puede mostrar:

- Objetos del dominio o clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de las clases conceptuales.

4.1.2.1 Diagrama de Clases Conceptuales.

- **Identificación de Clases Conceptuales:**

En esta etapa de análisis del Modelo del Dominio, se especifican las *Clases Conceptuales* relacionadas con los escenarios principales de éxito de los casos de uso en estudio (Previamente seleccionados en el plan de iteración – Apartado 3.6). Para la identificación de las *clases conceptuales* se aplicó la técnica de identificación de *frases nominales*⁵ [LARMAN-03], la cual se basa en la identificación de nombres y frases nominales de las descripciones textuales de un dominio, considerándolos como *clases conceptuales* o atributos candidatos. Este análisis se realiza a partir de los Casos de Uso en formato completo. La siguiente lista muestra las *clases conceptuales* extraídas del análisis de frases nominales:

- Bobina Defectuosa.
- Datos Cualitativos.
- Línea Origen.
- Estados Calidad.
- Defecto.
- Aplicación.
- Responsable.
- Causas.

⁵ Frase Nominal: Frase representativa del sistema extraída de los casos de uso.

La lista de clases conceptuales obtenida genera la Fase I del Diagrama de Dominio de la figura 23, al cual posteriormente se adicionarán asociaciones y atributos.

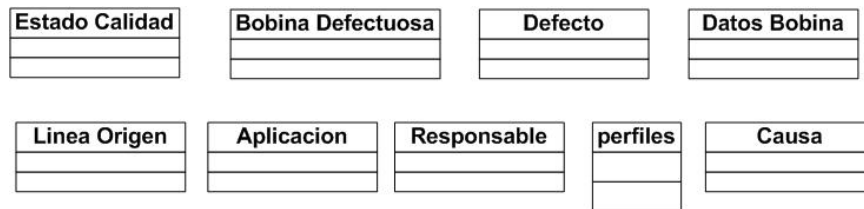


Figura 23. Diagrama del Dominio (Fase I).

• **Adición de Asociaciones:**

Una Asociación, es representada mediante una línea entre clases, con un nombre de asociación. La asociación es inherentemente bidireccional, lo que significa es que desde las instancias de cualquiera de las dos clases, es posible el recorrido lógico hacia la otra.

A continuación se definen las asociaciones entre las clases del diagrama mostrado en la figura 23.

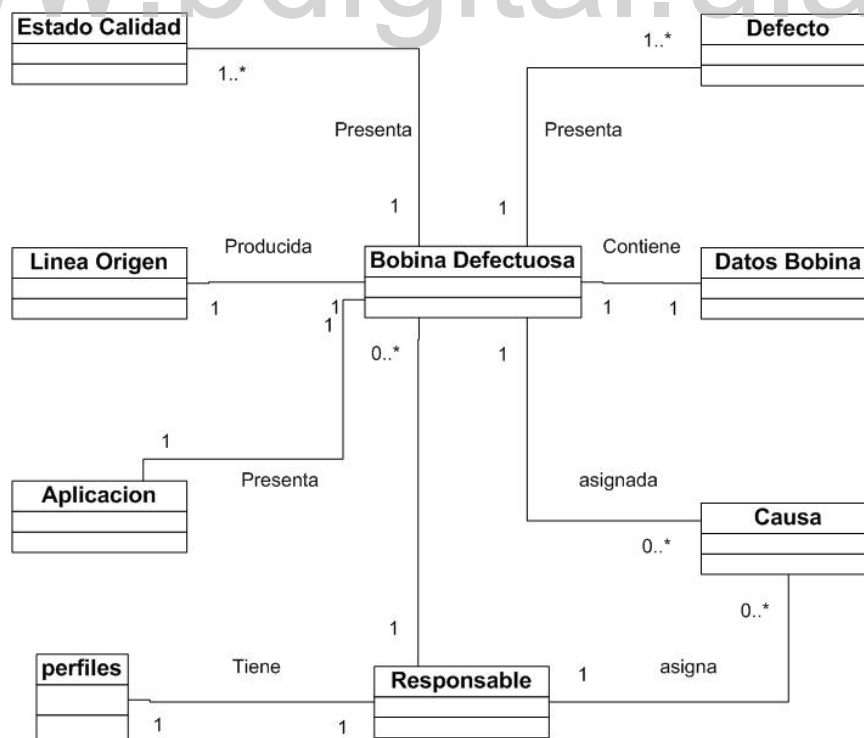


Figura 24. Diagrama del Dominio (Fase II).

- **Adición de Atributos:**

Los Atributos son valores de datos lógicos de un objeto. Se incluyen en el modelo del dominio aquellos atributos para los que los requisitos (Casos de Uso) sugieren o implican una necesidad de registrar la información. Los Atributos se muestran en el segundo compartimiento del rectángulo de la clase. La combinación de las clases conceptuales, asociaciones y atributos descubiertos, da lugar al Modelo del Dominio parcial del SGC, el cual se representa en la figura 25

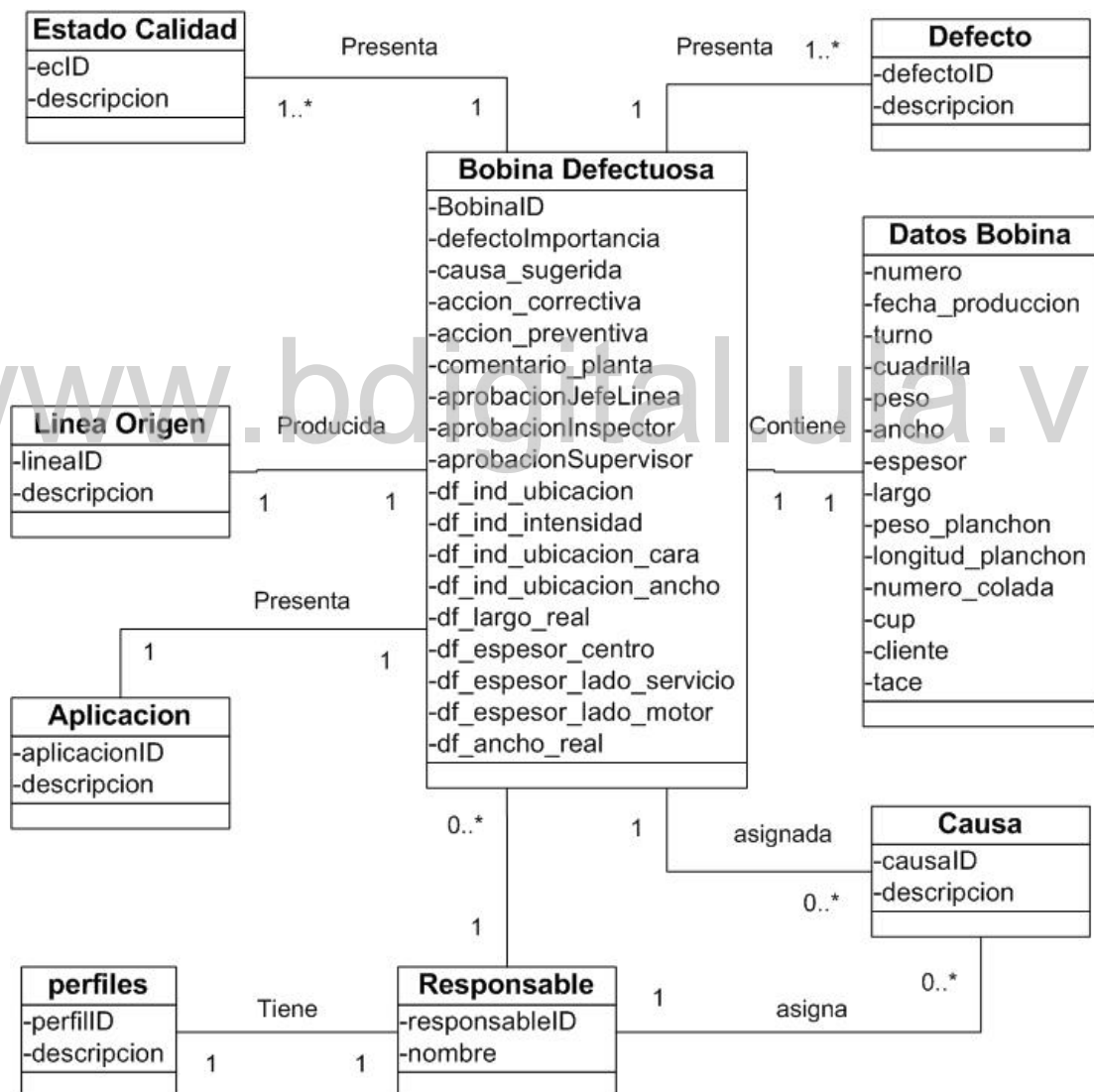


Figura 25. Diagrama de clases de Dominio - Iteración I.

En la tabla 5 se describen cada una de las clases conceptuales del diagrama de clases de la figura 25:

Nombre de la Clase	Descripción
Bobina Defectuosa	Representa el objeto físico Bobina defectuosas, que se produce en las diferentes líneas de Laminación
Datos Bobinas	Representa todos los datos generales propios de una bobina defectuosa
Defecto	Representa los diferentes defectos que puede presentar una bobina durante su producción.
Estado de Calidad	Representa los diferentes estados que se les puede asignar a una bobina que resulta defectuosa
Línea Origen	Representa las diferentes líneas de producción de bobinas.
Aplicación	Representa las diferentes aplicaciones que pueden ser asignadas a una bobina defectuosa.
Causa	Representa la causa de la aparición de un defecto en una bobina.
Responsables	Representa a las personas autorizadas para el manejo del sistema
Perfiles	Representa el perfil que cada una de las persona posee dentro de la Empresa.

Tabla 5. Descripción de las clases del Modelo de Dominio.

4.1.3 MODELO DE DISEÑO.

Lo esencial del Modelo de Diseño es la creación de los **Diagramas de Interacción**, los cuales representan el modo en que los objetos colaboran entre si para satisfacer los requisitos (casos de uso). Después de la elaboración de los *Diagramas de Interacción* se representa el **Diagrama de Clases** (del diseño), este es una abstracción del desarrollo que se debe implementar. Por lo que se refiere al UP, estos dos artefactos forman el Modelo

de Diseño. A continuación se especifican cada uno de los diagramas de interacción resultado de la etapa de Análisis de los Casos de Uso en estudio.

4.1.3.1 Diagramas de Interacción (DI).

Los diagramas de interacción ilustran el modo en que los objetos interaccionan por medio de mensajes. El término diagrama de interacción es una generalización de dos tipos de diagramas UML, cualquiera de ellos puede ser utilizado para representar interacciones de mensajes:

- Diagramas de colaboración.
- Diagramas de secuencias.

Los **diagramas de colaboración** ilustran las interacciones entre objetos en un formato de grafo o red, en el cual los objetos se pueden colocar en cualquier lugar del diagrama. Los **diagramas de secuencia** ilustran las interacciones en un tipo de formato con aspecto de valla, en el que cada objeto nuevo se añade a la derecha. Para la etapa de diseño presentado en este trabajo se seleccionaron los *diagramas de colaboración* para representar las interacciones entre los objetos del sistema, esto se debe a que los diagramas de secuencias crecen verticalmente a medida que se agregan los objetos y por el ancho de las páginas estarían limitados, lo que no sucede con los diagramas de colaboración.

4.1.3.1.1 DI 1: Escenario Revisar Defectos.

El diagrama de la figura 26 ilustra la manera de como los objetos del sistema interaccionan entre si, con la finalidad de satisfacer los requisitos especificados en el escenario principal de éxito del Caso de Uso *UC 1: Revisar Defectos*. En este caso (refiriéndonos al *DSS 1: Revisar defectos – Fig. 19*), el usuario hace una solicitud al sistema, el cual debe responder retornando un listado de bobinas defectuosas. Para ello todos los objetos del sistema deben interactuar a través de mensajes, los cuales se van pasando entre los objetos hasta cumplir el objetivo buscado. Para este caso, cuando el sistema recibe la solicitud *obtener_bobinas_defectuosas (...)*, debería responder de la siguiente manera:

1. La *colección*⁶ de **Bobinas Defectuosas** envía el mensaje *obtener_datos_bobinas* (*numero, fecha_produccion*) a la instancia **Datos Bobinas**. Esto lo hace por cada una de las bobinas defectuosas de la colección de manera tal que le sea posible obtener los datos generales de cada una de las bobinas de la colección.
2. De manera similar, la colección de **Bobinas Defectuosas** envía el mensaje *obtener_linea* (*LO*) a la instancia **Línea Origen**.
3. La colección de **Bobinas Defectuosas** envía el mensaje *obtener_ec* (*ec*) a la instancia **Estado Calidad**.
4. La colección de **Bobinas Defectuosas** envía el mensaje *obtener_defecto* (*defecto*) a la instancia **Defecto**.
5. La colección de **Bobinas Defectuosas** envía el mensaje *obtener_aplicacion* (*aplicacion*) a la instancia **Aplicación**.

De esta manera el sistema debe estructurar todo el listado con los datos requeridos por el usuario y retornarlo. Vale la pena señalar que los datos específicos de los diferentes listados que los usuarios requieren como respuesta del sistema se especificarán más a detalle en los subsiguientes capítulos del trabajo, a la hora de diseñar cada una de las pantallas, donde se visualizarán los mismos.

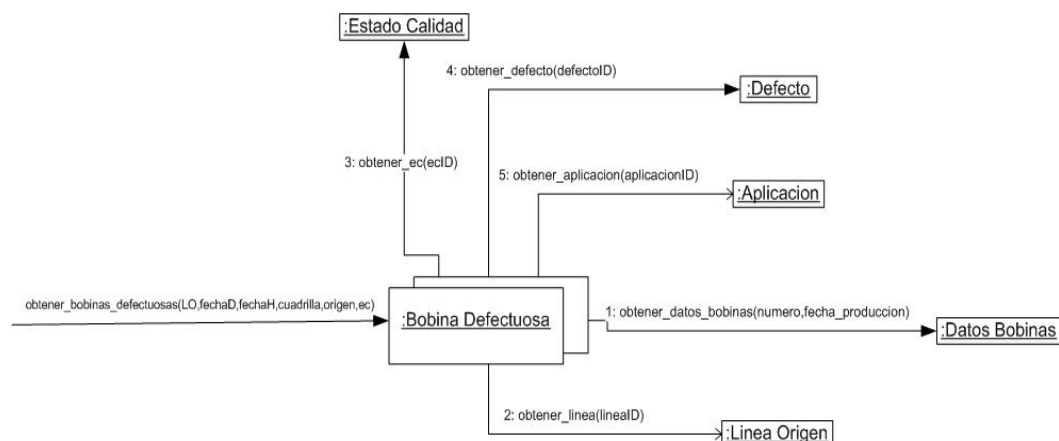


Figura 26. DI 1 Revisar Defectos.

⁶ Conjunto de instancias que representan a un Multiobjeto (Como una tabla o lista)

4.1.3.1.2 DI 2: Escenario Analizar Bobinas.

La descripción del diseño de este *Diagrama de Interacción* (figura 27), es similar al anterior, ya que el listado de bobinas defectuosas que desea visualizar el usuario para analizarlo y posteriormente asignarle responsables es prácticamente el mismo. La diferencia que existe es que los parámetros que suministra el usuario al sistema son diferentes a los del caso anterior, es decir las bobinas defectuosas que el usuario desea visualizar en este listado son aquellas que posean una *línea origen*, *rango de fechas*, *defecto*, *aplicación* y *ec* determinado, a diferencia del diagrama anterior, en el que las bobinas defectuosas deseadas son aquellas que tengan *línea origen*, *rango de fechas*, *cuadrilla*, *origen* y *ec* específico. Esta diferencia se hará más palpable a la hora de diseñar las *CONSULTAS SQL*⁷, que cumplan con estos requerimientos, ya que es allí donde se nota la diferencia entre una sentencia y la otra.

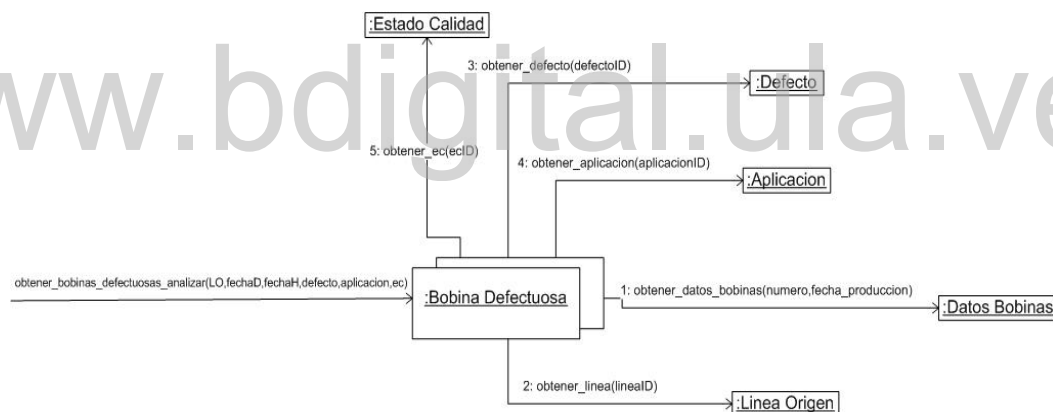


Figura 27. DS 2 Analizar Bobinas.

4.1.3.1.3 DI 3: Escenario Consultar Cantidad Bobinas Asignadas.

El diagrama de la figura 28 representa el diseño del escenario principal de éxito del Caso de Uso *UC 3: Consultar Cantidad Bobinas Asignadas*, este refleja la manera como el sistema debe responder al momento en que el usuario solicita el listado Línea – Defecto –

⁷ Revisar marco teórico

Cantidad, donde se reflejará el numero de bobinas que le fueron asignadas por Línea. El diagrama de la figura 28 se lee como sigue:

1. La colección recibe la solicitud *ObtenerListadoCantidadBobinas (LO, responsable)*
2. La colección envía el mensaje *obtener_datos_bobinas_defectuosas (BobinaID)* a la instancia **Bobina Defectuosa**.
3. La instancia **Bobina Defectuosa** envía el mensaje *obtener_defecto (defectoID)* a la instancia **Defecto**.
4. La instancia **Bobina Defectuosa** envía el mensaje *obtener_linea (lineaID)* a la instancia **Línea Origen**.

El resultado de los envíos de mensajes entre las diferentes instancias del diagrama es el listado Línea – Defecto – Cantidad solicitado, la figura 28 muestra el diagrama de objetos de este proceso



Figura 28. DI 3 Consultar Cantidad Bobinas Asignadas.

4.1.3.1.4 DI 4: Escenario Revisar Bobinas Asignadas.

El diagrama de la figura 29 refleja la realización del Caso de Uso *UC 4: Revisar Bobinas Asignadas*, donde el usuario además de visualizar un listado detallado con las bobinas asignadas, visualizará las causas anteriormente aplicadas a los defectos para su posible reutilización. El diagrama de la figura 29 se lee de la siguiente manera:

1. La colección **Responsable Bobinas Defectuosas** recibe el mensaje *ObtenerBobinasAnalizar (linea, defecto)* y paralelamente la colección **Causas** recibe la solicitud *ObtenerCausas (linea, defecto)*.

2. La colección **Responsable Bobinas Defectuosas** envía el mensaje *obtener_datos_bobinas_defectuosas (BobinaID)* a la instancia **Bobinas Defectuosas**.
3. La instancia **Bobinas Defectuosas** envía el mensaje *obtener_datos_bobinas (numeo, fecha_produccion)* a la instancia **Datos Bobina**.
4. La instancia **Bobinas Defectuosas** envía el mensaje *obtener_defecto (defectoID)* a la instancia **Defecto**.
5. La instancia **Bobinas Defectuosas** envía el mensaje *obtener_linea (lineaID)* a la instancia **Línea Origen**.

Como resultado se obtiene el listado deseado con los detalles de cada una de las bobinas asignadas al usuario. Paralelamente se obtiene el listado con las causas anteriormente aplicadas a las bobinas en la línea y defecto especificados.

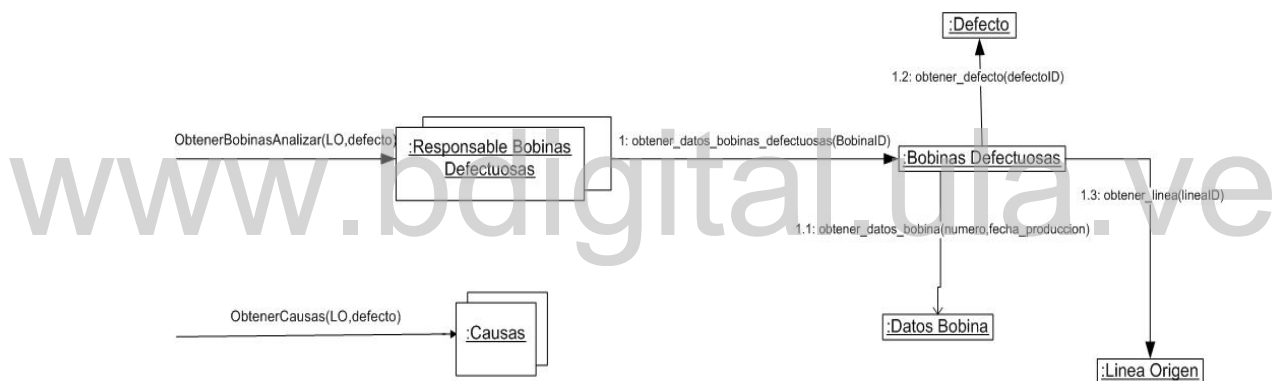


Figura 29. DI 4 Revisar Bobinas Asignadas.

4.1.3.2 Diagrama de Clases (De Diseño - DCD).

Un diagrama de clases de diseño (DCD) representa las especificaciones de las clases e interfaces software en el sistema. Entre la información general que se presenta en el diagrama de clase se encuentra:

- Clases
- Asociaciones
- Atributos y tipos
- Operaciones.

A diferencia de las clases conceptuales del Modelo del Dominio, las Clases de Diseño de los DCD muestran las definiciones de las clases software en lugar de los conceptos del mundo real. Sin embargo, el diseño del DCD está basado netamente en el Diagrama de Clases de Dominio (ver figura 25), para que de esta manera sea más fácil y más representativo pasar de la representación del mundo real del sistema (que se logró con el diagrama de dominio), al mundo del software. La figura 30 especifica el Diagrama de Clases de Diseño, donde se reflejan las diferentes clases de diseño con sus respectivos atributos, asociaciones y operaciones.

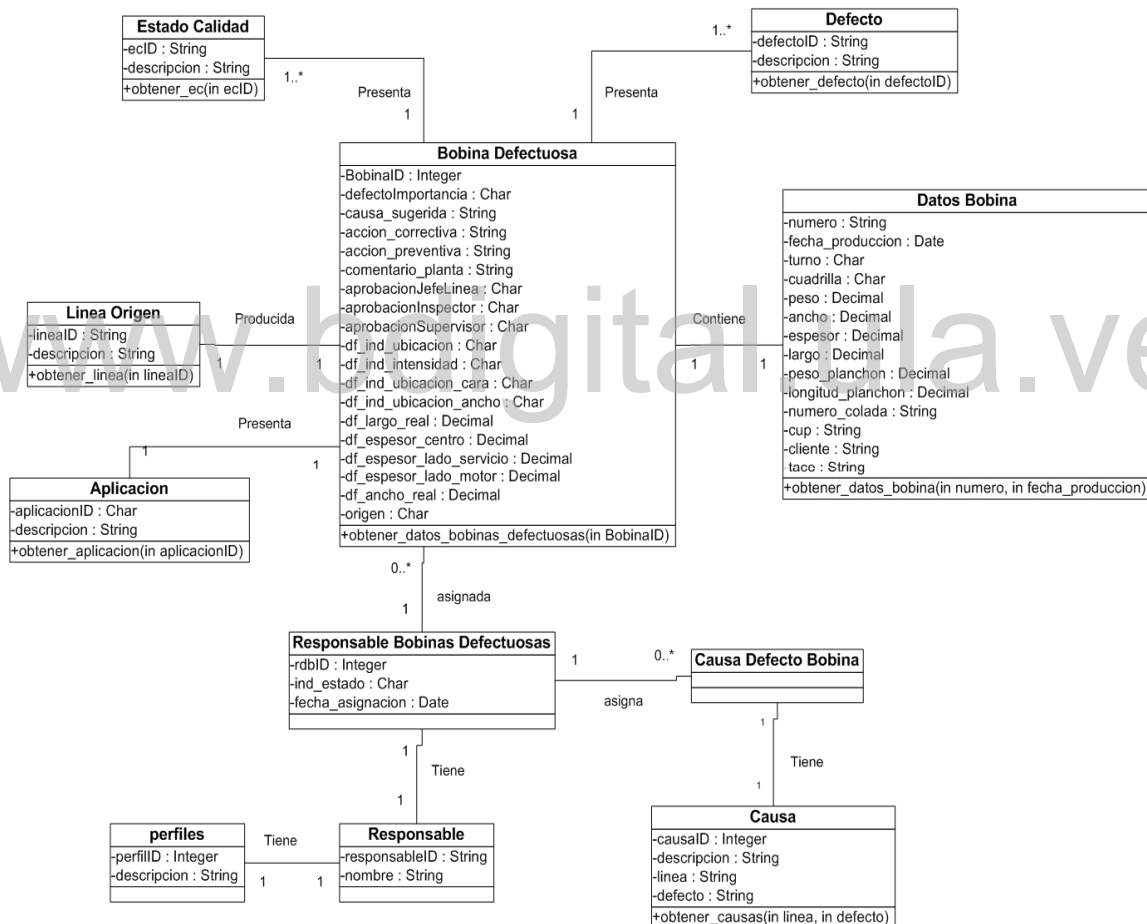


Figura 30. Diagrama de Clases de Diseño - Iteración I.

Una vez obtenido este diagrama se puede notar que es bastante similar al diagrama de clases conceptuales obtenido en el modelado de dominio (ver fig. 25). Si embargo, durante la etapa de diseño fue necesario agregar las clases **Responsable Bobinas**

Defectuosas y Causa Defecto Bobina y modificar algunas asociaciones. Esto aclara la idea de que no necesariamente se deben especificar en el DCD solo aquellas clases que se obtuvieron en el modelo de dominio, sino que también es posible incluir nuevas clases. Una vez obtenido este diagrama se procede a la etapa de diseño de la Base de Datos, la cual estuvo a cargo de una de las personas responsables del sistema designada por la Empresa. Cabe resaltar que el paso de este diagrama de clases de diseño a la elaboración e implementación de las tablas de la Base de Datos es bastante sencillo, ya que gracias a todo el trabajo de Análisis y Diseño que se llevó a cabo durante la elaboración del diagrama es prácticamente suficiente para pasar al diagrama relacional de la Base de Datos el cual se implementará en SQL SERVER.

4.1.4 IMPLEMENTACIÓN.

El trabajo de Diseño que se ha llevado a cabo hasta ahora no implica que no exista diseño durante la etapa de implementación, por el contrario, gran parte de los diagramas de interacción mas sencillos se plasmaron directamente en el código, sin dejar de tomar en cuenta el diseño que esto conllevaría. Esta decisión se tomo debido a que en el transcurso de las próximas iteraciones de la fase de elaboración fueron surgiendo nuevos requisitos, lo que implica que se tendría que diseñar un gran número de diagramas, y por conveniencia se prefiere implementar directamente los requisitos más simples. Cada uno de los *diagramas de interacción* que se diseñaron en el Modelo de Diseño, se transformaran en esta etapa en sentencias SQL, que son las responsables de que los diferentes listados solicitados por los usuarios se generen. Esto señala la importancia que tiene la creación de los diagramas de interacción durante el desarrollo del proyecto guiado por el UP, ya que es relativamente sencillo generar código cuando se tiene bien claro como el sistema debe interactuar para cumplir los requerimientos. Luego en el capítulo siguiente del trabajo se explicará la manera de cómo el SGC por medio de la *arquitectura multicapas* se comunica con la Base de Datos para ejecutar estas sentencias SQL y visualizar los listados obtenidos como resultado.

Todas estas sentencias SQL, se almacenan en *procedimientos almacenados*⁸ a nivel de Base de Datos, para luego codificar una clase general (Llamada Data_Access.vb) en código Visual Basic. Net que se encarga de hacer la comunicación entre la aplicación y estos procedimientos almacenados. Además, se deben codificar también a nivel de Base de Datos, todas las *funciones y vistas* necesarias para la implementación de los procedimientos almacenados. Luego de finalizar la implementación de la Base de Datos a cargo de la persona anteriormente mencionada, se procedió a implementar cada uno de los procedimientos almacenados, así como las funciones y vistas. Los procedimientos codificados se muestran en la tabla 6:

Procedimiento Almacenado	Parámetros de entrada	Resultado
web_sgc_obtener_bobinas_defectuosas	LO, fechaD, fechaH, cuadrilla, origen, ec.	Listado con bobinas defectuosas.
web_sgc_obtener_bobinas_defectuosas_analizar	LO, fechaD, fechaH, defecto, aplicación, ec.	Listado con bobinas defectuosas a analizar.
web_sgc_obtener_listado_cantidad_bobinas	LO, responsable	Listado Línea – Defecto – Cantidad
web_sgc_obtener_bobinas_asignadas	LO, defecto	Listado detallado con las bobinas defectuosas asignadas
web_sgc_obtener_causas	LO, defecto	Listado con causa anteriormente

⁸ Consulte Marco Teórico.

		aplicadas por línea – defecto-
--	--	-----------------------------------

Tabla 6. Definición de Procedimientos Almacenados.

Por estándares de la empresa todos los procedimientos almacenados que se crean en los servidores de BD, deben poseer el prefijo WEB en caso de que se trate de una aplicación Web, además se debe colocar seguido del prefijo el nombre del proyecto. Es por esto que todos los procedimientos almacenados comienzan por web_sgc. Todos los códigos implementados durante el desarrollo del proyecto, se almacenan en el archivo de codificaciones ubicado en el CD-ROM adjunto a este trabajo.

4.1.5 GLOSARIO.

Para la descripción del glosario en esta iteración se obviarán los términos ya descritos en el glosario de la fase de inicio (Apartado 3.5).

Término	Definición e Información	Alias
defectoImportancia	Representa la prioridad que tiene un defecto en una bobina defectuosa	def_imp
Causa_sugerida	Representa la causa sugerida que carga el operador en planta.	Cs
accion_preventiva	Representa la acción preventiva sugerida por el operador en planta para evitar el defecto.	Ap
accion_correctiva	Representa la acción correctiva sugerida por el operador en planta para corregir el defecto.	Ac
comentario_planta	Representa algún comentario que carga el operador en planta según su consideración.	com_p
aprobacionJefeLinea	Representa la firma del Jefe de Línea	firmaJP
aprobacionSupervisor	Representa la firma del Supervisor de Línea	firmaS
aprobacionInspector	Representa la firma del Inspector de Línea	firmaI

df_ind_ubicacion	Representa un código para la ubicación del defecto presentado en la bobina.	df_ind_ubi
df_ind_intensidad	Representa un código para el grado de intensidad del defecto presentado en la bobina.	df_ind_in
df_ind_ubicacion_cara	Representa un código para la ubicación cara del defecto presentado en la bobina.	df_ind_ubi_c
df_ind_ubicacion_ancho	Representa un código para la ubicación ancho del defecto presentado en la bobina.	df_ind_ubi_a
Df_largo_real	Representa el largo del defecto presentado en la bobina.	df_lr
df_espesor_centro	Representa el espesor del defecto presentado en la bobina.	df_ec
df_espesor_lado_servicio	Representa el espesor del lado del servicio del defecto presentado en la bobina.	df_els
df_espesor_lado_motor	Representa el espesor del lado del motor del defecto presentado en la bobina.	df_elm
Df_ancho_real	Representa el ancho del defecto presentado en la bobina.	df_ar
Numero	Representa el numero de cada una de las bobinas	nro_bobina
Fecha_produccion	Representa la fecha de producción de la bobina.	fecha_prod
Peso	Representa el peso de la bobina.	
Ancho	Representa el ancho de la bobina.	
Espesor	Representa el espesor de la bobina.	
Largo	Representa el largo de la bobina.	
Peso_planchon	Representa el peso del plancho de donde se produce una bobina.	
Longitud_planchon	Representa la longitud del plancho de donde	

	se produce una bobina.	
numero_colada	Representa el número de la colada donde se produjo el planchon.	nro_colada
Cup	Representa un código de especificaciones de la bobina producida.	
Cliente	Representa el cliente destinado a la bobina	
Tace	Representa el tipo de acero con que se produjo la bobina.	
Fecha_asignacion	Representa la fecha que le fue asignada la bobina a un responsable para su gestión.	FA
Perfil	Representa el perfil que puede tener un responsable en la Empresa.	

Tabla 7. Glosario – Fase de Elaboración – Iteración I

4.1.6 PLAN DE ITERACIÓN:

Para la segunda Iteración se planificó un segundo taller de requisitos, donde se tomaron nuevos requerimientos, también se refinan los ya existentes debido al surgimiento de nuevas peticiones sobre ellos. Una vez finalizado el taller de requisitos, se procedió con la escritura de los casos de uso surgidos, así como también se realizó toda la etapa de análisis, diseño e implementación que esto conlleva. Todo esto lleva al refinamiento de los Modelos de Casos de Uso, Modelo de Dominio y Modelo de Diseño respectivamente.

4.2 ITERACIÓN II.

Al comienzo de esta iteración fue llevado a cabo un segundo taller de requisitos, el cual trajo como resultado nuevos requerimientos por parte de los usuarios. Debido a esto es conveniente refinar el Listado Actor – Objetivos de la tabla 2, ya que para alguno de los usuarios los objetivos se incrementaron. En esta nueva lista se omitirán los objetivos que no se vieron afectados.

ACTOR	OBJETIVOS
Jefe de Procesos	<ul style="list-style-type: none"> • Analizar un listado de bobinas defectuosas. • Asignar responsables a cada una de las bobinas defectuosas. • Eliminar asignaciones erróneas de bobinas a procesistas.
Procesista	<ul style="list-style-type: none"> • Verificar cantidad de bobinas asignadas por Línea para su análisis. • Revisar listado con bobinas asignadas de una línea y defecto determinado. • Asignar causas para los defectos detectados en las bobinas defectuosas asignadas. • Revisar en un listado todas las causas con sus acciones asignadas anteriormente por línea y defecto. • Asignar nuevas acciones a las causas.

Tabla 8. Listado Actor – Objetivos.

4.2.1 MODELO DE CASOS DE USO.

De la lista anterior surgen tres nuevos casos de uso los cuales deben especificarse en formato completo, estos son:

- Eliminar Responsables a Bobinas.
- Revisar acciones asignadas a causas.
- Asignar acciones.

Debido al surgimiento de estos tres nuevos casos de uso el diagrama de casos de uso de la figura 2 queda modificado como sigue:

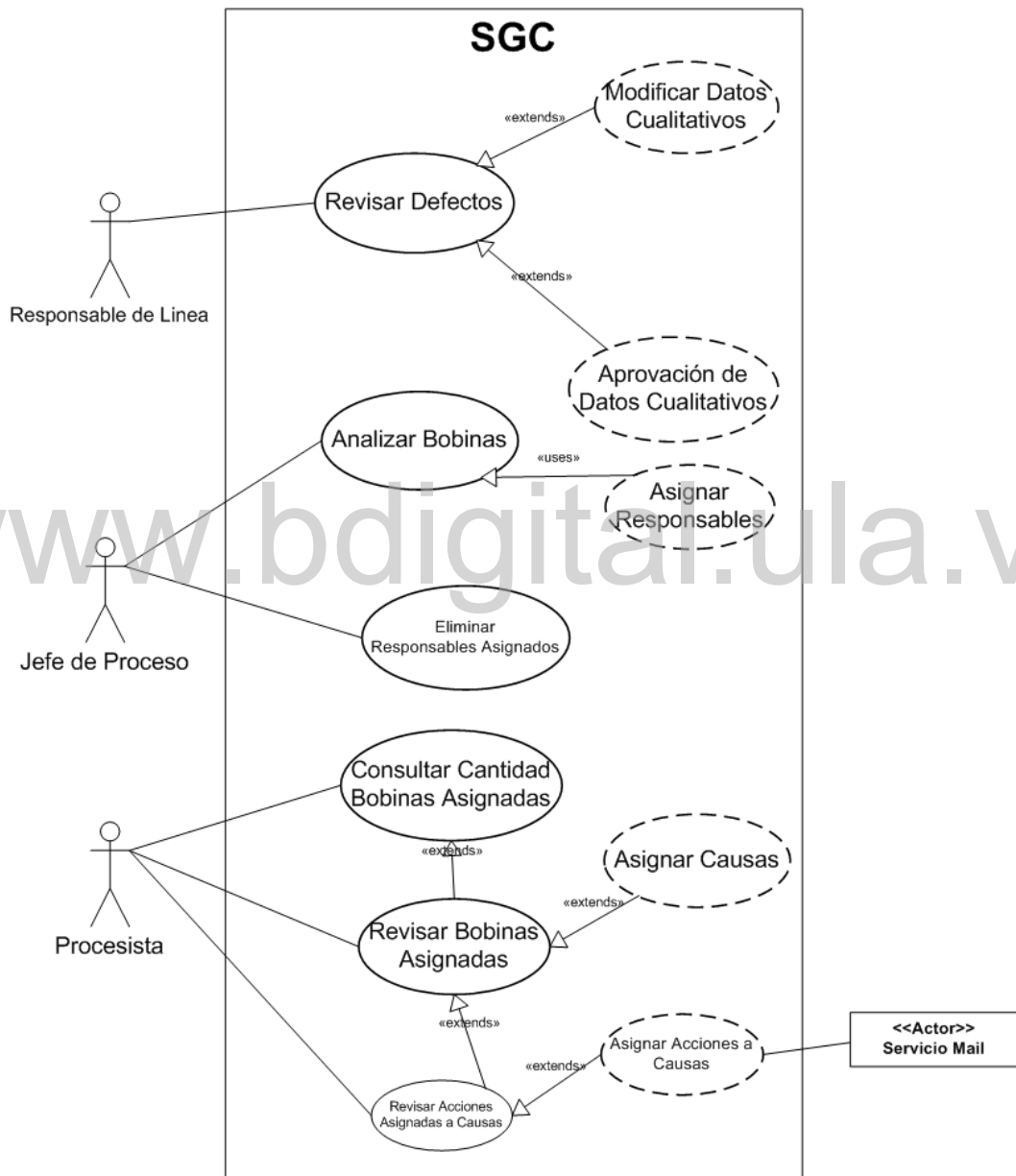


Figura 31. Diagrama de Casos de Uso - Iteración II.

4.2.1.1 Descripción de los Casos de Uso (Requisitos funcionales).

A continuación se describen cada uno de los nuevos Casos de Uso en formato completo:

4.2.1.1.1 Caso de Uso UC 9: Eliminar Responsable Asignado.

Actores Principales:

- Jefe de Proceso.

Personal Involucrado e Intereses:

- Jefe de Proceso: Quiere visualizar en un listado bobinas defectuosas (con sus respectivos datos cualitativos), a las cuales se les haya asignado responsables. Además de esto quiere poder eliminar asignaciones de responsables erróneas.

Precondiciones:

- El Jefe de procesos debe estar cargado previamente en el sistema, además de esto debe poseer el permiso requerido por el sistema para eliminar responsables.

Garantías de Éxito (Poscondiciones):

- Se visualiza el listado correcto de bobinas defectuosas a las cuales se les haya asignado responsable
- Se elimina satisfactoriamente los responsables deseados.

Escenario Principal de Éxito (Flujo Básico):

1. El Actor selecciona la *Línea Origen* la cual quiere que tengan las bobinas defectuosas con responsable asignado a visualizar.
2. El Actor selecciona un rango de fechas, dentro del cual desea que se encuentren las bobinas defectuosas con responsable asignado que desea visualizar.
3. El Actor selecciona el responsable para las bobinas defectuosas que desea visualizar.
4. Se Visualiza el listado de bobinas defectuosas para el responsable seleccionado.

Se repiten los pasos 1 – 4 cuantas veces el Actor lo considere.

5. Se procesa la eliminación de responsable a las bobinas seleccionadas por el Actor.

Extensiones (o Flujo Alternativo):

- *a En cualquier momento el sistema falla:

Se debe emitir un mensaje al usuario, y además de debe reportar el problema.

- 2a. El Actor selecciona un rango de fechas no válido:

2. El Sistema debe emitir un mensaje indicando el error y además deberá rechazar la selección de fechas y solicitar una nueva.

- 4a. No se visualizan los datos esperados:

2. Emitir un mensaje señalando que no existen datos para la entrada especificada.

- 5a. Ocurre un error durante la eliminación de los responsables:

1. Emitir un mensaje señalando la ocurrencia del error, reportar el error a las personas responsables del sistema.

Requisitos Especiales:

- Pantalla Web, con texto visible y claro al usuario, basada en los estándares de la empresa.
- Tiempos de respuestas para la visualización de los datos no mayor de 30 Seg.
- Tiempos de respuesta para la eliminación de los responsables no mayor a 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

- 1-3a. Se usará el teclado o Mouse de computador para la selección de Línea Origen, Fechas, defecto y responsables.
- 4a. La pantalla Web se visualizará en monitores de 17” (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende de los *turnos*.

4.2.1.1.2 Caso de Uso UC 10: Revisar Acciones Asignadas a Causas.**Actores Principales:**

- Procesista.

Personal Involucrado e Intereses:

- Procesista: Quiere visualizar las causas asignadas a defectos y además quiere por cada causa visualizar las acciones correctivas aplicadas a las causas. También quiere marcar como significativas aquellas acciones que considere que fueron buenas para corregir las causas.

Precondiciones:

- El Actor debe estar registrado en el sistema y poseer el permiso requerido por el sistema para asignar acciones a causas.

Garantías de Éxito (Poscondiciones):

- Se visualiza el listado correcto de causas asignadas a defectos por instalación y defecto.
- Se visualizan las acciones anteriormente aplicadas por causas.
- Se registran como significativas aquellas acciones que el Actor considere que lo fueron (en caso de considerarlo el Actor).

Escenario Principal de Éxito (Flujo Básico):

1. El Actor selecciona la *Línea Origen* para las causas a visualizar.
2. El Actor selecciona el defecto para las causas a visualizar.
3. Se visualizan las causas.
4. se visualizan las acciones asignadas a las causas (En caso de existir).

El Actor repite los pasos 1 – 4, cuantas veces considere necesarias.

5. Se marcan como significativas aquellas acciones que el Actor considere.
6. Se procesa el registro de acciones significativas.

Extensiones (o Flujo Alternativo):

*a En cualquier momento el sistema falla:

Se debe emitir un mensaje al usuario, y además de debe reportar el problema.

4a. No se visualizan las causas esperadas:

1. Emitir un mensaje señalando que no existen datos para la entrada especificada.

6a. Ocurre un error durante el registro de las acciones significativas:

1. Emitir un mensaje de error notificando al usuario y además informar a los responsables del mantenimiento del sistema de la ocurrencia del mismo.

Requisitos Especiales:

- Pantalla Web, con texto visible y claro al usuario, basada en los estándares de la empresa.
- Tiempos de respuestas para la visualización de los datos no mayor de 30 Seg.
- Tiempo de respuestas para el registro de acciones significativas no mayor de 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

1-2a. Se usará el teclado o Mouse de computador para la selección de Línea Origen y defecto.

6a. La pantalla Web se visualizará en monitores de 17” (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende de los *turnos*.

4.2.1.1.3 Caso de Uso UC 11: Asignar Acciones a Causas.

Actores Principales:

- Procesista.

Personal Involucrado e Intereses:

- **Procesista:** Quiere asignar acciones correctivas a causas productoras de los defectos. Quiere también asignar un responsable que ejecute la acción creada, además quiere asignar una fecha de inicio y una de culminación para la realización de la acción. Una vez editada la acción quiere que automáticamente se envíe un E-Mail a la persona responsable de ejecutar la acción, notificando dicha asignación para que la ejecute antes de la fecha de culminación. Finalmente quiere que el sistema ejecute un proceso diariamente de manera tal que las acciones cuya fecha de culminación sea menor a la del día actual se coloquen automáticamente en estado “Vencido” a la acción y se notifique a la persona que estaba responsable de ejecutarla.

Precondiciones:

- El Actor debe estar registrado en el sistema y poseer los permisos necesarios para asignar acciones a causas.
- Las personas a las cuales se les asignarán las acciones para su ejecución deben estar registradas en el sistema.

Garantías de Éxito (Poscondiciones):

- Se edita correctamente la acción.
- Se envía satisfactoriamente el E-Mail notificando al responsable de ejecutar la acción.

Escenario Principal de Éxito (Flujo Básico):

1. El Actor selecciona la fecha de inicio de la acción.
2. El Actor selecciona la fecha de culminación de la acción.
3. El Actor introduce el nombre de la acción.
4. El Actor introduce la descripción de la acción.
5. El Actor introduce el procedimiento a seguir para el cumplimiento de la acción.
6. El actor selecciona de una lista el responsable que ejecutará la acción.
7. El Actor asigna el estado “En Curso” a la acción.

8. Se registra la nueva acción.
9. Se envía el E-Mail al responsable de ejecutar la acción con los datos de la misma.

Extensiones (o Flujo Alternativo):

*a En cualquier momento el sistema falla:

Se debe emitir un mensaje al usuario, y además de debe reportar el problema.

1 - 2a. El Actor selecciona un rango de fechas no válido:

1. El Sistema debe emitir un mensaje indicando el error y además deberá rechazar la selección de fechas y solicitar una nueva.

8a. Ocurre un error durante el registro de la acción:

1. Emitir un mensaje señalando que no se registro la acción y notificar a los responsables del sistema.
2. Evitar el envío del E – Mail.

9a. Ocurre un error durante el envío del E – Mail:

1. Notificar al usuario que no se logró enviar el E – Mail.
2. Informar para que lo envíe manualmente.
3. Notificar al personal encargado del sistema.

Requisitos Especiales:

- Pantalla Web, con texto visible y claro al usuario, basada en los estándares de la empresa.
- Tiempos de respuestas para el registro de la acción no mayor de 30 Seg.
- Tiempo de respuesta para el envío de E – Mail no mayor de 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

- 1-7a. Se usará el teclado o Mouse de computador para la edición de la acción.
- 6a. La pantalla Web se visualizará en monitores de 17” (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende de los *turnos*.

4.2.1.2 Especificaciones Complementarias.

4.2.1.2.1 Introducción.

Esta sección del documento es el repositorio de todos los requisitos del SGC que no se capturan en los Casos de Uso anteriormente especificados (Requisitos no funcionales). Se excluirán aquellas especificaciones que ya fueron cubiertas en la fase de inicio (ver apartado 3.4).

4.2.1.2.2 Detección y gestión de errores.

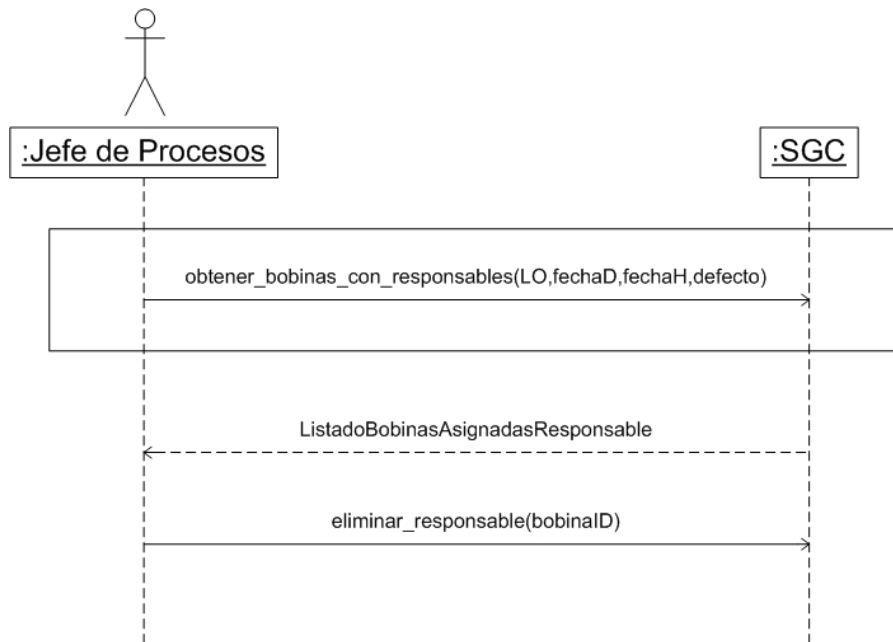
Para realizar la notificación de la ocurrencia de errores se usará un servicio diseñado en la empresa para el envío de E – Mail. De esta manera cada vez que ocurra un evento de error cada una de las personas encargadas de realizar mantenimiento y monitoreo al SGC será notificado de inmediato.

4.2.1.2.3 Restricciones de Implementación.

El proceso de actualización de estado de las acciones vencidas y envío de E-Mail a los responsables respectivos, será un proceso que corra en el servidor de BD, debido a que la empresa cuenta con servidores SQL SEVER que permiten el envío de E-Mail programados.

4.2.1.3 Elaboración de Diagramas de Secuencia.

En esta etapa se especifican cada uno de los *diagramas de secuencia* correspondientes a la realización de los *Casos de Uso* descritos en la etapa anterior. Se omiten aquellos diagramas que ya fueron creados en la Iteración I.

4.2.1.3.1 DSS 5: Eliminar Responsable Asignado.**Figura 32. DSS 5: Eliminar Responsable Asignado.****4.2.1.3.2 DSS 6: Revisar Acciones asignadas a Causas.****Figura 33. DSS 6: Revisar Acciones Asignadas a Causas.**

4.2.1.3.3 DSS 7: Asignar Acciones a Causas.

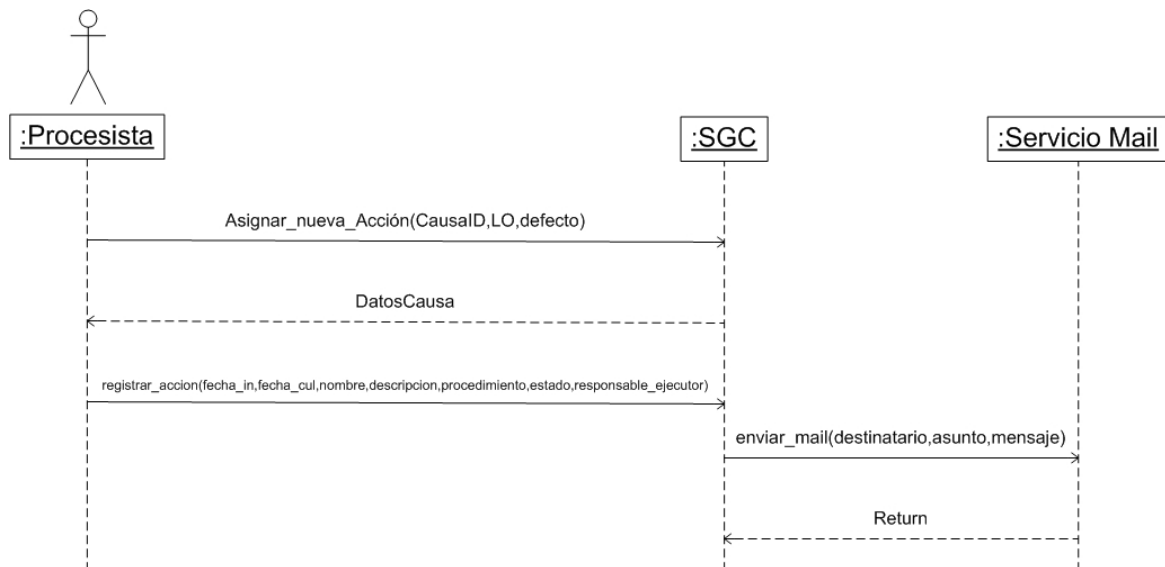


Figura 34. DSS 7: Asignar Acciones a Causas.

www.bdigital.ula.ve

4.2.2 MODELO DEL DOMINIO.

4.2.2.1 Diagrama de Clases Conceptuales.

Aplicando la técnica de *frases nominales* para los casos de uso descritos en la etapa anterior, surge la nueva clase **Acciones**, cuyos atributos y asociación se especifica en el diagrama de clases de dominio de la figura 35:

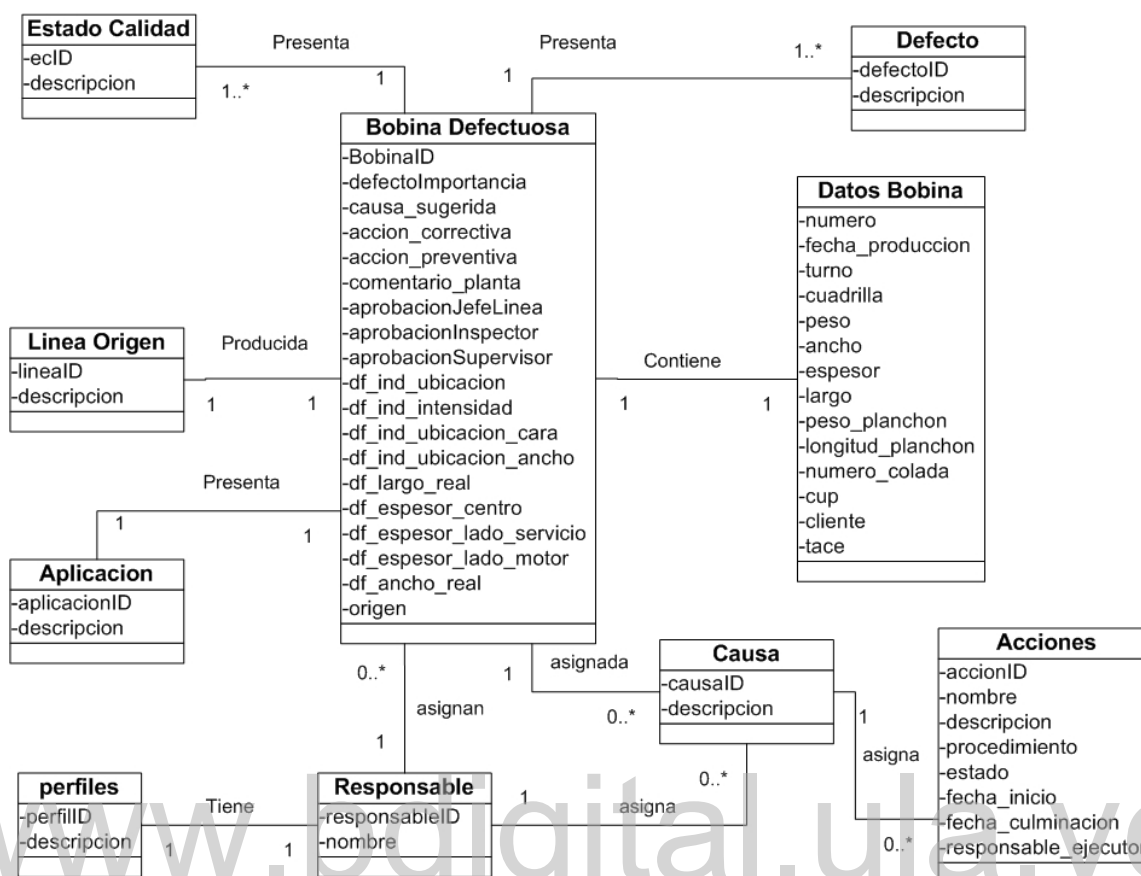


Figura 35. Diagrama de Clases de Dominio - Iteración II.

EL diagrama obtenido en esta etapa de análisis (figura. 35) corresponde a la continuación del modelado de dominio comenzado en la Iteración I de esta fase del UP (ver figura 25), allí se reflejan claramente los pasos para la obtención de cada una de las clases conceptuales. En la tabla 9 se describe la nueva clase Acciones, omitiéndose las anteriormente ya especificadas en la tabla 5:

Nombre de la Clase	Descripción
Acciones	Representa las acciones que se deberían ejecutar con la finalidad de corregir la existencia de las causas de los defectos.

Tabla 9. Descripción de clases del Modelo de Dominio.

4.2.3 MODELO DE DISEÑO.

En esta etapa de la Iteración II se procederá a diseñar los *diagramas de interacción*, los cuales ilustran la interacción entre los objetos del sistema con el fin de cumplir con los objetivos descritos por los casos de uso especificados en esta Iteración.

4.2.3.1 Diagramas de Interacción

A continuación se especifican cada uno de los diagramas de interacción para la realización de los casos de uso de esta Iteración.

4.2.3.1.1 DI 5: Eliminar Responsable Asignado.

El diagrama de la figura 36 refleja la realización del Caso de Uso UC 9: *Eliminar Responsable Asignado*, donde el usuario además de querer visualizar un listado detallado de bobinas defectuosas con responsable asignado, también quiere eliminar ciertas bobinas a responsables las cuales se las haya asignado erróneamente. El diagrama de la figura 36 se lee de la siguiente manera:

1. La colección **Bobina Defectuosa** recibe el mensaje *obtener_bobinas_con_responsables(LO,fechaD,fechaH,defecto)*.
2. La colección **Bobina Defectuosa** envía el mensaje *obtener_datos_bobinas(numero, fecha_produccion)* a la instancia **Datos Bobinas**.
3. La colección **Bobinas Defectuosas** envía el mensaje *obtener_linea (lineaID)* a la instancia **Linea Origen**.
4. La colección **Bobina Defectuosa** envía el mensaje *obtener_defecto (defectoID)* a la instancia Defecto.
5. La colección **Bobina Defectuosa** envía el mensaje *obtener_responsable_bobina (bobinaID)* a la instancia **Responsable Bobina Defectuosa**.
6. La instancia **Responsable Bobina Defectuosa** envía el mensaje *ontener_responsable(resonsableID)* a la instancia **Responsable**.

7. En caso de que se retorne el listado solicitado la colección Bobina Defectuosa, por cada bobina defectuosa seleccionada, envía el mensaje *eliminar_responsable(BobinaID)* a la instancia **Responsable Bobina Defectuosa**.

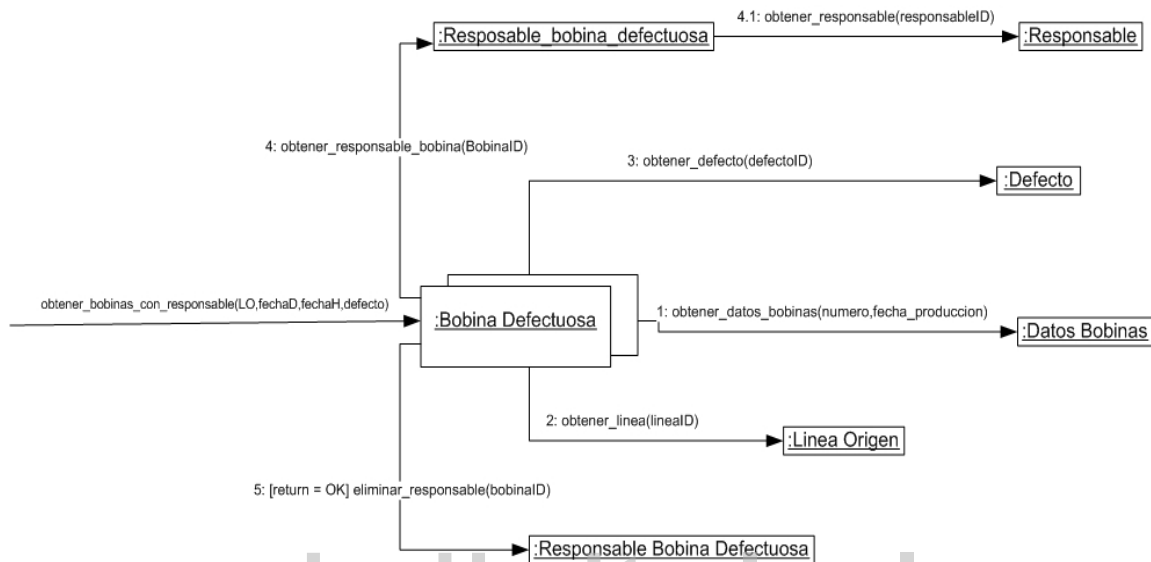


Figura 36 DI 5: Eliminar Responsable Asignado

4.2.3.1.2 DI 6: Revisar Acciones Asignadas a Causas.

El diagrama de la figura 37 refleja la realización del Caso de Uso *UC 10: Revisar Acciones Asignadas a Causas*, donde el usuario además de querer visualizar las causas asignadas por línea y defecto desea visualizar también la acciones asignadas a las causas. Por otro lado si lo considera puede marcar una acción como significativa. El diagrama de la figura 37 se lee de la siguiente manera:

1. La colección **Causa** recibe el mensaje *obtener_causas(LO,defecto)*.
2. La colección **Causa** envía el mensaje *obtener_acciones(causaID,linea,defecto)* a la colección **Acciones** por cada una de las instancias de su colección, de esta manera obtiene el conjunto de acciones asignadas a cada causa.
3. En caso de que se solicite, la colección **Causa** envía el mensaje *marcar_significativa(accionID)* a la instancia **Acción**.

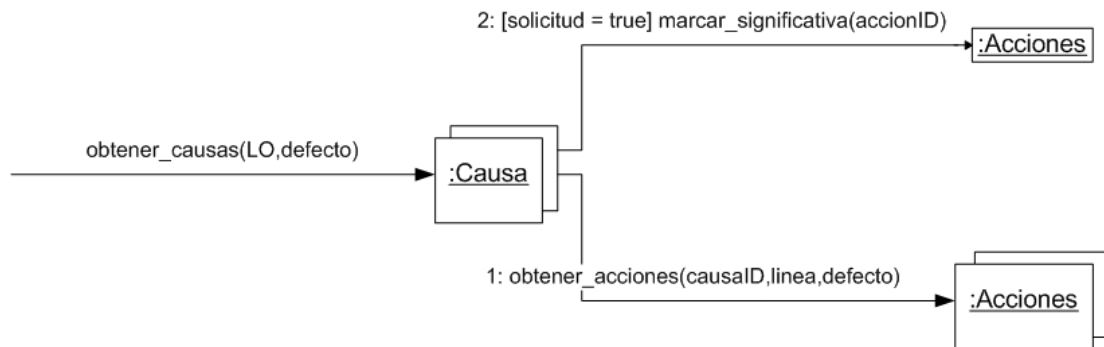


Figura 37 DI 6: Revisar Acciones Asignadas a Causas.

4.2.3.1.3 DI 7: Asignar Acciones a Causas.

El diagrama de la figura 38 refleja la realización del Caso de Uso *UC 11: Asignar Acciones a Causas*, donde el usuario editará una nueva acción para ser ejecutada por un responsable. Además a este responsable se le notificará automáticamente de la asignación. El diagrama de la figura nnn se lee de la siguiente manera:

1. La instancia **Causa** recibe el mensaje *asignar_nueva_acción (causalID,LO,defecto)*.
2. La instancia **Causa** envía el mensaje *enviar_mail(destinatario,asunto,mensaje)* al sistema externo **Servicio Mail** (Propio de la empresa).
3. Si el envío fue realizado con éxito, la instancia **Causa** envía el mensaje *registrar_accion(...)* con todos los atributos necesarios para la creación de la Acción.

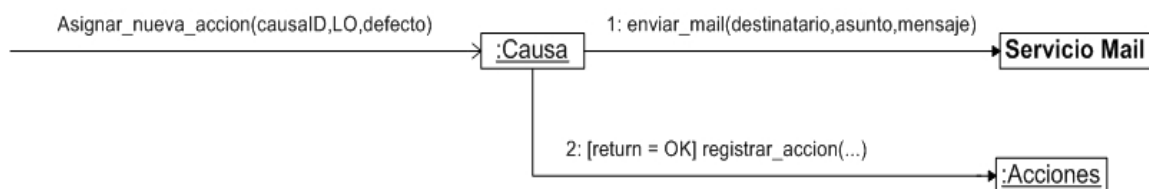


Figura 38 DI 7: Asignar Acciones a Causas.

4.2.3.2 Diagrama de Clases de Diseño.

En la figura 39 se detalla el diagrama de clase de diseño obtenido en esta Iteración. Este diagrama a diferencia del obtenido en la Iteración I (ver figura 30) tiene una nueva clase de diseño (Acciones), asociada a la clase causas, ya que por requerimientos de los casos de uso capturados, a cada una de las causas se les debería asignar entre 0 y muchas acciones.

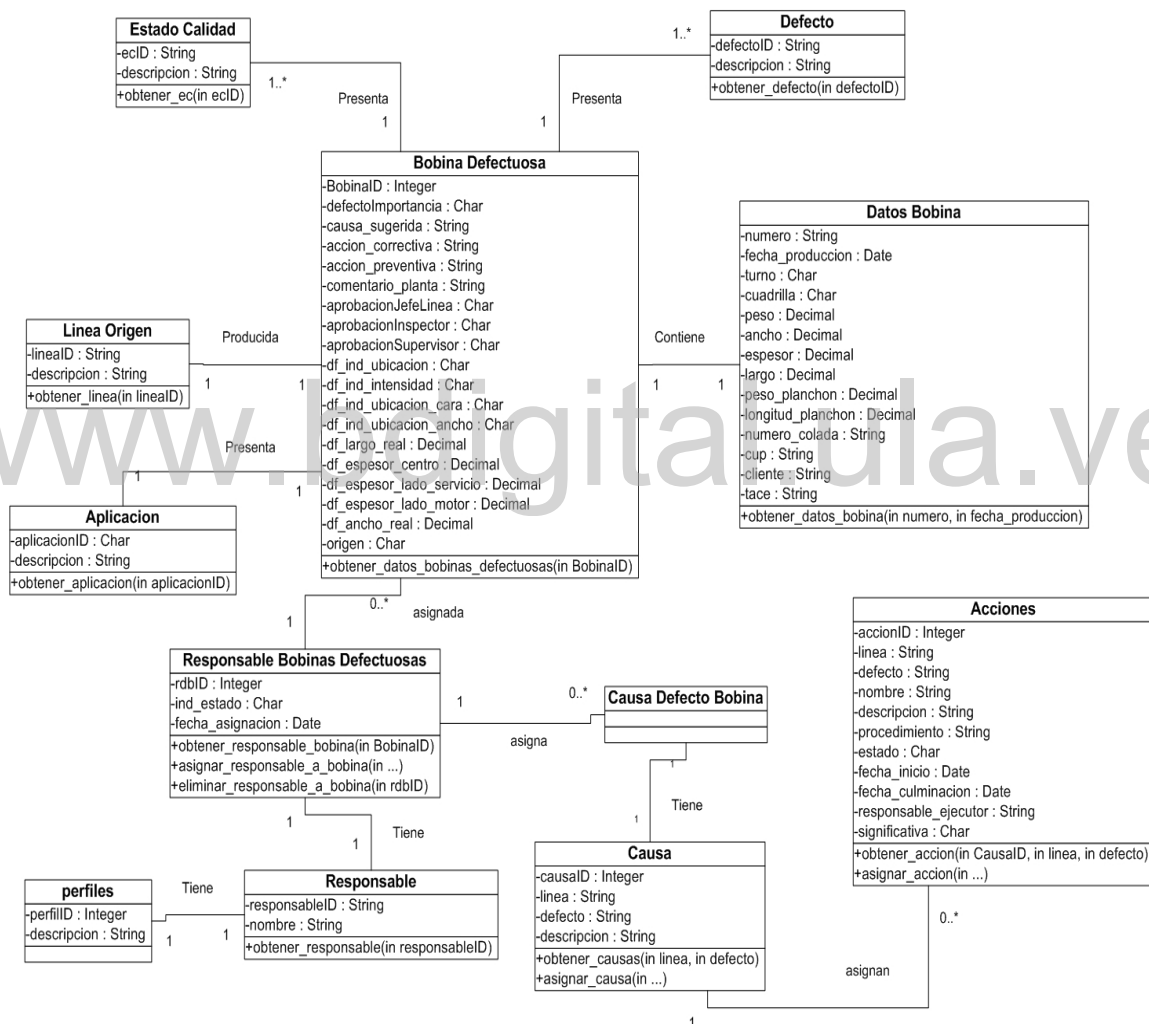


Figura 39 Diagrama de Clases de Diseño - Iteración II.

En la sección de anexos (Anexo A) se describen cada una de las clases del diagrama de clases de diseño.

4.2.4 IMPLEMENTACIÓN.

En esta etapa se procede a implementar cada uno de los procedimientos almacenados que corresponden con los mensajes provenientes de los diagramas de interacción. Estos mensajes son el resultado de la realización de los nuevos Casos de Uso capturados en el taller de requisitos. Estos procedimientos almacenados serán los encargados de obtener como de almacenar los datos en la BD. Los mismos se muestran en la tabla 10:

Procedimiento Almacenado	Parámetros de entrada	Resultado
web_sgc_obtener_bobinas_con_responsable	LO, fechaD, fechaH, defecto.	Listado con bobinas defectuosas con responsable asignado.
Web_sgc_eliminar_responsable	BobinaID	Eliminación del responsable a la bobina.
web_sgc_obtener_causas	LO, defecto.	Listado con causas por línea, defecto.
web_sgc_obtener_acciones	causaID, linea, defecto	Listado con acciones asignadas a causas.
web_sgc_marcar_significativas	accionad	Marcar como significativa una acción.
web_sgc_asignar_nueva_accion	causaID, linea, defecto	Retornar datos de la Causas
web_sgc_registrar_accion	fecha_inicio, fecha_culminacion, nombre, descripción,	Registro de una nueva acción.

	procedimiento, estado, responsable	
--	---------------------------------------	--

Tabla 10. Definición de Procedimientos Almacenados.

Recordemos que la comunicación de estos procedimientos almacenados a nivel de Base de Datos con la aplicación, será por medio de una clase general (Data_Access.vb) la cual se comenzará a desarrollar en la siguiente iteración. También debemos recordar que para visualizar cada uno de los código fuentes pertenecientes a esta clase y a los procedimientos almacenados podemos dirigirnos al CD (sección de codificaciones) adjunto al trabajo.

4.2.5 GLOSARIO.

Término	Definición e Información	Alias
Servicio Mail	Servicio Web ubicado en los servidores de producción encargado de enviar Mail, recibiendo como parámetro el destinatario del Mail, el Asunto y el mensaje (cuerpo del Mail)	

Tabla 11. Glosario – Fase de Elaboración – Iteración II

Los términos anteriormente especificados en los Glosarios de la Etapas anteriores fueron omitidos.

4.2.6 Plan de Iteración.

Para la tercera Iteración se planificó un tercer taller serio de requisitos, donde se tomaron nuevos requerimientos. También se refinan los ya existentes en caso debido a nuevas peticiones sobre ellos. Además en la siguiente Iteración se aumenta la comunicación con el usuario, ya que la etapa de **Construcción y Transición** se dedica mayormente a implementar más a detalle y a realizar pruebas rigurosas donde la interacción con los

usuarios es indispensable. En el transcurso de estas pruebas los usuarios van estableciendo nuevos requerimientos sencillos de implementar, los cuales se plasman directamente en el código. Para el análisis, diseño e implementación durante esta tercera iteración de la fase de elaboración se crean los mismos artefactos del UP que se han venido creando en las iteraciones anteriores.

4.3 ITERACIÓN III.

El taller de requisitos que se llevo a cabo para la elaboración de esta Iteración se centro más que todo en una visión netamente de gestión, donde lo primordial fue visualizar en pantallas todo el trabajo realizado por los procesistas para corregir los defectos presentados en las bobinas. Las personas encargadas de supervisar este trabajo son los Superintendentes de Laminación, debido a esto surge el nuevo Actor Superintendente, cuyos Objetivos a cumplir por el sistema se especifican a continuación:

ACTOR	OBJETIVOS
Superintendente	<ul style="list-style-type: none"> • Visualizar en una pantalla todo el trabajo realizado por los procesistas, es decir causas asignadas a las bobinas defectuosas y acciones correctivas asignadas a las causas. • Visualizar en una pantalla defectos a los cuales se les haya asignado causas y acciones correctivas. • Visualizar en gráficos de paretos, las veces que fueron asignadas las causas a los defectos. • Visualizar en gráficos de paretos las veces que se asignaron

	acciones a diferentes causas. <ul style="list-style-type: none"> • Visualizar en una pantalla las acciones asignadas a las causas con todos sus datos.
--	---

Tabla 12. Lista Actor – Objetivos.

Esta lista (tabla 12), corresponde a la continuación de la lista de Actor – Objetivos que se ha venido creando en la fase de inicio y en la Iteración II respectivamente.

4.3.1 MODELO DE CASOS DE USO

De la lista anterior de Actor - Objetivos (tabla 12) surgen nuevos casos de uso los cuales deben especificarse en formato completo, estos son:

- Gestión de Procesistas.
- Gestión de Defectos.
- Ver Paretos Causas Defectos.
- Ver Paretos Acciones Causas.
- Gestión de Acciones Asignadas.

Debido al surgimiento de estos nuevos casos de uso el Diagrama de Casos de Uso de la figura 31 queda modificado como sigue:

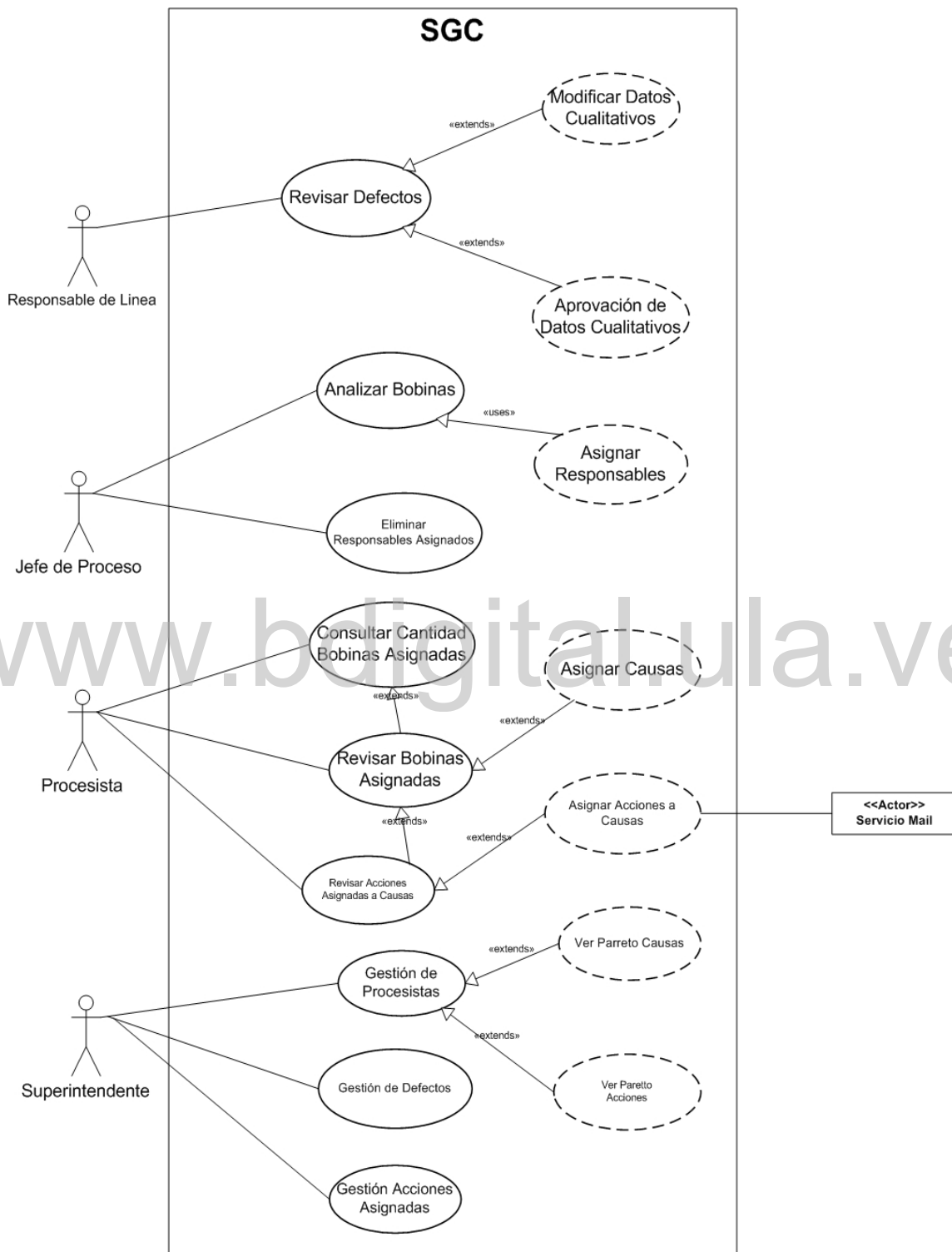


Figura 40 Diagrama de Casos de Uso - Iteración III.

4.3.1.1 Descripción de los Casos de Uso (Requisitos funcionales).

A continuación se describen cada uno de los nuevos Casos de Uso en formato completo:

4.3.1.1.1 Caso de Uso UC 12: Gestión de Procesistas.

Actores Principales:

- Superintendente.

Personal Involucrado e Intereses:

- Superintendente: Quiere visualizar en una pantalla clara y bien especificada cada uno de los responsables de bobinas defectuosas (Procesistas), además quiere ver por responsable las bobinas que se les fueron asignadas. Por cada una de las bobinas asignadas quiere visualizar las causas asignadas y por cada causa quiere visualizar las acciones.
- Procesista: Quiere que se visualice correctamente su trabajo realizado.

Precondiciones:

- El Superintendente debe estar cargado previamente en el sistema.

Garantías de Éxito (Poscondiciones):

- Se visualiza el listado con los responsables con bobinas defectuosas asignadas.
- Se visualizan las bobinas asignadas por responsables.
- Se visualizan las causas asignadas por bobina defectuosa (En caso de existir).
- Se visualizan las acciones asignadas por causa. (En caso de Existir).

Escenario Principal de Éxito (Flujo Básico):

1. El Actor selecciona la *Línea Origen* para los responsables que desea visualizar.
2. El Actor selecciona un rango de fechas, dentro del cual desea que se encuentre la fecha de asignación de las bobinas por responsable.
3. El Actor selecciona el responsable que desea visualizar.

4. El Actor selecciona el *tipo de estado* que quiere visualizar.
5. Se Visualizan los responsables con sus bobinas asignadas, causas y acciones.

Se repiten los pasos 1 – 5 cuantas veces el Actor lo considere.

Extensiones (o Flujo Alternativo):

*a En cualquier momento el sistema falla:

Se debe emitir un mensaje al usuario, y además de debe reportar el problema.

2a. El Actor selecciona un rango de fechas no válido:

1. El Sistema debe emitir un mensaje indicando el error y además deberá rechazar la selección de fechas y solicitar una nueva.

5a. No se visualizan los datos esperados:

1. Emitir un mensaje señalando que no existen datos para la entrada especificada.

Requisitos Especiales:

- Pantalla Web, con texto visible y claro al usuario, basada en los estándares de la empresa.
- Tiempos de respuestas para la visualización de los datos no mayor de 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

- 1-4a. Se usará el teclado o Mouse de computador para la selección de Línea Origen, Fechas, responsable y estado.
- 5a. La pantalla Web se visualizará en monitores de 17” (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende del Actor.

4.3.1.1.2 Caso de Uso UC 13: Gestión de Defectos.**Actores Principales:**

- Superintendente.

Personal Involucrado e Intereses:

- Superintendente: Quiere visualizar en una pantalla clara y bien especificada cada uno de los defectos a los cuales se les aya asignado causas, además en caso de que las causas tengan acciones asignadas quiere visualizarlas.

Precondiciones:

- El Superintendente debe estar cargado previamente en el sistema.

Garantías de Éxito (Poscondiciones):

- Se visualiza el listado de defectos con causas asignadas.
- Se visualizan las causas asignadas a los defectos.
- Se visualizan las acciones asignadas por causa. (En caso de Existir).

Escenario Principal de Éxito (Flujo Básico):

1. El Actor selecciona la *Línea Origen* para los defectos que desea visualizar.
2. El Actor selecciona un rango de fechas, dentro del cual desea que se encuentre la fecha de asignación de las causas al defecto.
3. El Actor selecciona el defecto que desea visualizar.
4. Se Visualizan los defectos con sus causas y acciones asignadas.

Se repiten los pasos 1 – 4 cuantas veces el Actor lo considere.

Extensiones (o Flujo Alternativo):

*a En cualquier momento el sistema falla:

Se debe emitir un mensaje al usuario, y además de debe reportar el problema.

2a. El Actor selecciona un rango de fechas no válido:

1. El Sistema debe emitir un mensaje indicando el error y además deberá rechazar la selección de fechas y solicitar una nueva.

4a. No se visualizan los datos esperados:

1. Emitir un mensaje señalando que no existen datos para la entrada especificada.

Requisitos Especiales:

- Pantalla Web, con texto visible y claro al usuario, basada en los estándares de la empresa.
- Tiempos de respuestas para la visualización de los datos no mayor de 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

1-3a. Se usará el teclado o Mouse del computador para la selección de Línea Origen, Fechas y defecto.

5a. La pantalla Web se visualizará en monitores de 17” (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende del Actor.

4.3.1.1.3 Caso de Uso UC 14: Gestión Acciones Asignadas.

Actores Principales:

- Superintendente.

Personal Involucrado e Intereses:

- Superintendente: Quiere visualizar en un listado todas las acciones asignadas a las causas, donde se refleje tanto el procesista que creo la acción como la persona responsable de ejecutarla.
- Procesista: Quiere que se visualicen correctamente las acción creadas por él, para que se vea el desarrollo de su trabajo.

Precondiciones:

- El Superintendente debe estar cargado previamente en el sistema.

Garantías de Éxito (Poscondiciones):

- Se visualiza el listado de acciones correctamente.

Escenario Principal de Éxito (Flujo Básico):

1. El Actor selecciona un rango de fechas dentro del cual desea que se encuentre las acciones.
2. El Actor selecciona el estado que desea que tengan las acciones a visualizar.
3. El Actor selecciona el responsable ejecutor de las acciones que considere.
4. Se visualiza el listado de acciones.

Se repiten los pasos 1 – 4 cuantas veces el Actor lo considere.

Extensiones (o Flujo Alternativo):

*a En cualquier momento el sistema falla:

Se debe emitir un mensaje al usuario, y además de debe reportar el problema.

1a. El Actor selecciona un rango de fechas no válido:

1. El Sistema debe emitir un mensaje indicando el error y además deberá rechazar la selección de fechas y solicitar una nueva.

4a. No se visualizan las acciones esperadas:

1. Emitir un mensaje señalando que no existen datos para la entrada especificada.

Requisitos Especiales:

- Pantalla Web, con texto visible y claro al usuario, basada en los estándares de la empresa.
- Tiempos de respuestas para la visualización de las acciones no mayor de 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

1-3a. Se usará el teclado o Mouse del computador para la selección de Fechas, estado y responsable.

4a. La pantalla Web se visualizará en monitores de 17” (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende del Actor.

4.3.1.1.4 Caso de Uso UC 15: Ver Pareto Causas.

Actores Principales:

- Superintendente.

Personal Involucrado e Intereses:

- Superintendente: Quiere visualizar en una gráfica las causas y la cantidad de veces que esas causas fueron asignadas a un defecto en específico.

Precondiciones:

- El Superintendente debe estar cargado previamente en el sistema.

Garantías de Éxito (Poscondiciones):

- Se visualiza la grafica correctamente.

Escenario Principal de Éxito (Flujo Básico):

1. El Actor selecciona el defecto al cual desea graficarle las causas asignadas.
2. Se visualiza la Gráfica correctamente.

Se repiten los pasos 1 – 2 cuantas veces el Actor lo considere.

Extensiones (o Flujo Alternativo):

*a En cualquier momento el sistema falla:

Se debe emitir un mensaje al usuario, y además se debe reportar el problema.

Requisitos Especiales:

- Pantalla Web basada en los estándares de la empresa para visualizar la gráfica.
- Tiempos de respuestas para la visualización de la gráfica no mayor de 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

- 1a. Se usará el teclado o Mouse del computador para la selección del defecto.
- 2a. Debe utilizarse para graficar las causas Chart FX para .NET (Software utilizado para realizar Gráficas de datos).
- 2b. La pantalla Web se visualizará en monitores de 17" (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende del Actor.

4.3.1.1.5 Caso de Uso UC 16: Ver Pareto Acciones.**Actores Principales:**

- Superintendente.

Personal Involucrado e Intereses:

- Superintendente: Quiere visualizar en una gráfica las acciones y la cantidad de veces que esas acciones fueron asignadas a una causa en específico.

Precondiciones:

- El Superintendente debe estar cargado previamente en el sistema.

Garantías de Éxito (Poscondiciones):

- Se visualiza la grafica correctamente.

Escenario Principal de Éxito (Flujo Básico):

1. El Actor selecciona la causa a la cual desea graficarle las acciones asignadas.

2. Se visualiza la Gráfica correctamente.

Se repiten los pasos 1 – 2 cuantas veces el Actor lo considere.

Extensiones (o Flujo Alternativo):

- *a En cualquier momento el sistema falla:

Se debe emitir un mensaje al usuario, y además de debe reportar el problema a los responsables del sistema.

Requisitos Especiales:

- Pantalla Web basada en los estándares de la empresa para visualizar la gráfica.
- Tiempos de respuestas para la visualización de la gráfica no mayor de 30 Seg.

Lista de Tecnologías y Variaciones de Datos:

- 1a. Se usará el teclado o Mouse del computador para la selección de la causa.
- 2a. Debe utilizarse para graficar las acciones Chart FX para .NET.
- 2b. La pantalla Web se visualizará en monitores de 17” (Estándar de la Empresa).

Frecuencia:

Podría ser casi continuo, depende del Actor.

4.3.1.2 Elaboración de Diagramas de Secuencia.

En esta etapa se especifican cada uno de los *diagramas de secuencia* correspondientes a la realización de los *Casos de Uso* descritos en la sección anterior. Se omiten aquellos diagramas que ya fueron presentados en la Iteración I y II respectivamente.

4.3.1.2.1 DSS 8: Gestión de Procesistas.

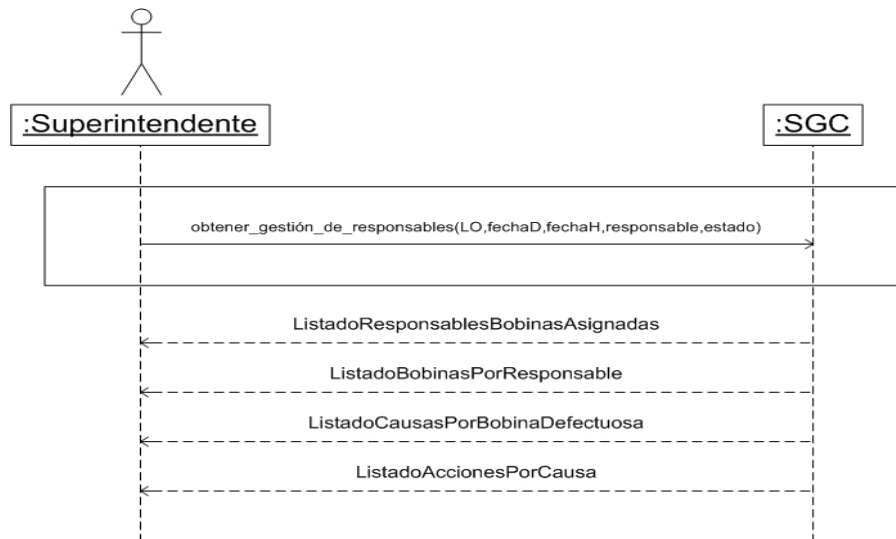


Figura 41. DSS 8: Gestión de Procesistas.

4.3.1.2.2 DSS 9: Gestión de Defectos.

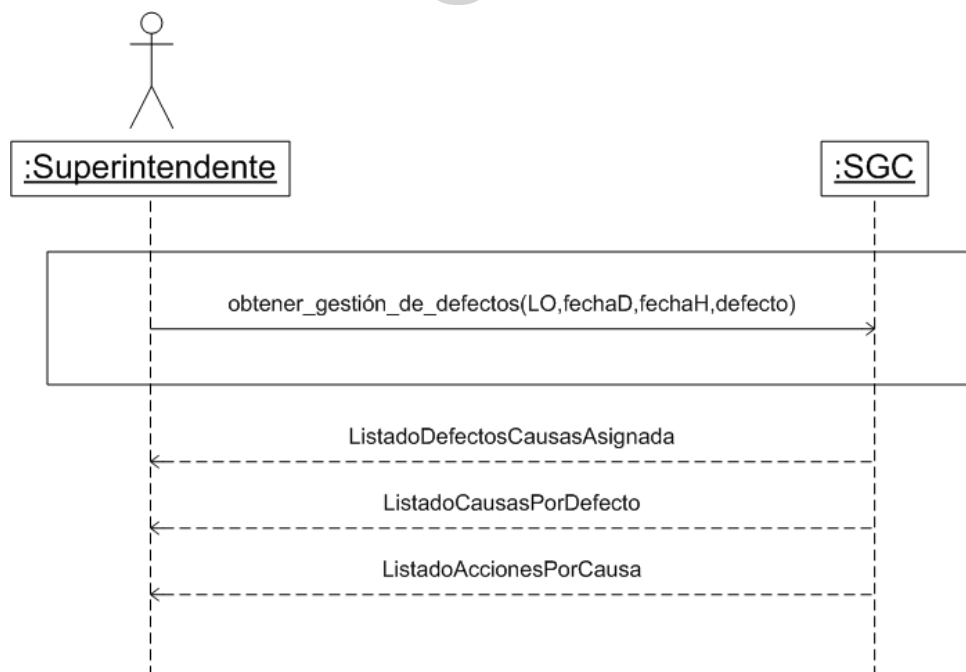


Figura 42. DSS 9: Gestión de Defectos.

4.3.1.2.3 DSS 10: Gestión Acciones Asignadas.

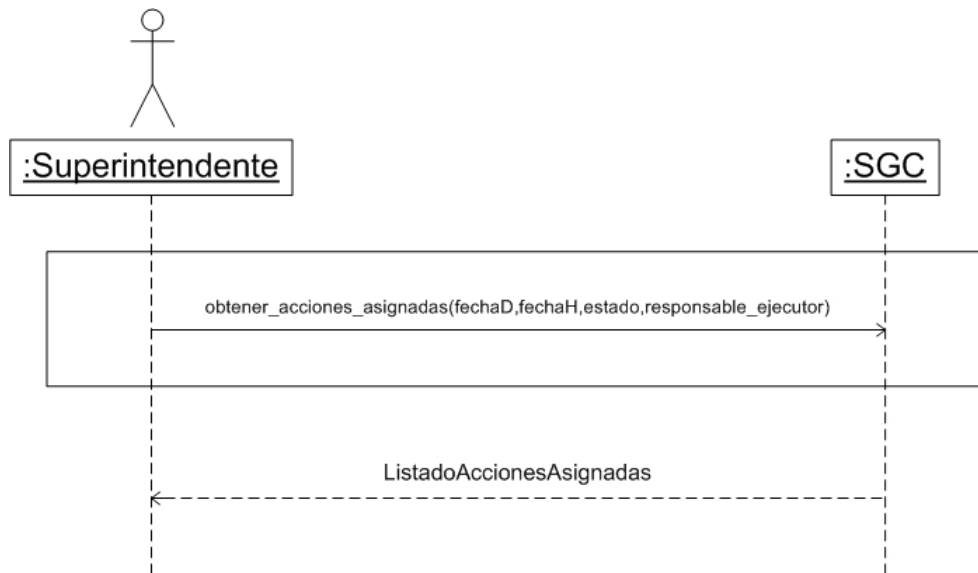


Figura 43. DSS 10: Gestión Acciones Asignadas.

4.3.1.2.4 DSS 11: Ver Pareto Causas.

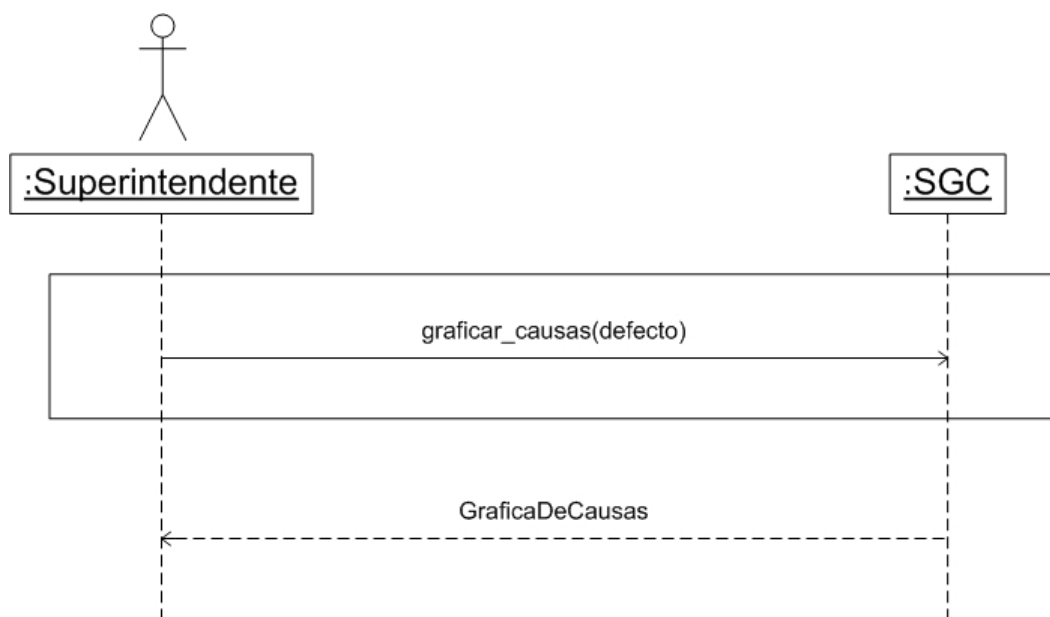


Figura 44. DSS 11: Ver Pareto Causas.

4.3.1.2.5 DSS 12: Ver Parreto Acciones.

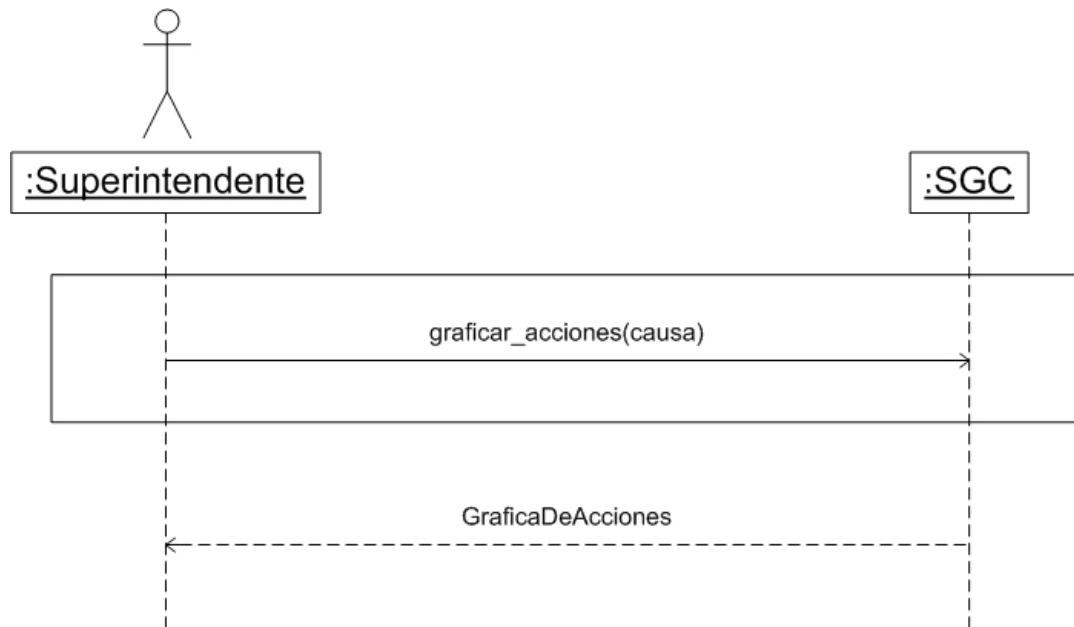


Figura 45. DSS 12: Ver Parreto Acciones.

www.bdigital.ula.ve

4.3.2 MODELO DEL DOMINIO.

En esta Iteración no surgen cambios sobre las clases conceptuales del dominio ya existentes, ya que los nuevos requerimientos son en base a reportes del trabajo que realizan los usuarios ya especificados, los nuevos Casos de Uso no inspiran nuevas frases nominales para identificar clases conceptuales de dominio. Debido a esto el diagrama de clases conceptuales queda igual al obtenido en la iteración anterior (Ver figura 35)

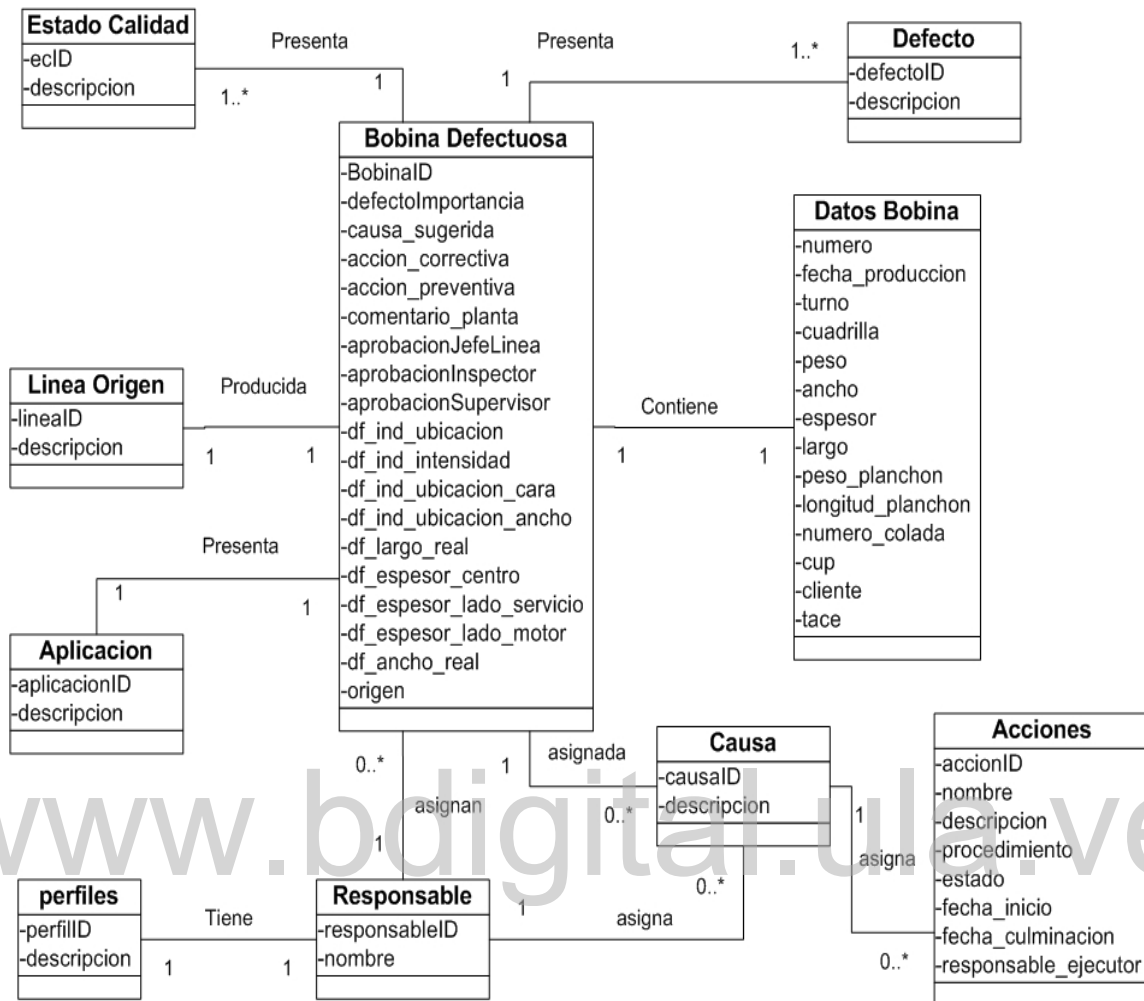


Figura 46. Diagrama de Clases de Dominio – Iteración III.

4.3.3 MODELO DE DISEÑO.

4.3.3.1 Diagramas de Interacción

A continuación se especifican cada uno de los diagramas de interacción para la realización de los casos de uso de esta Iteración.

4.3.3.1.1 DI 8: Gestión de Procesistas.

El diagrama de la figura 47 refleja la realización del Caso de Uso *UC 12: Gestión de Procesistas*, donde se obtendrá como resultado todo el trabajo realizado por los diferentes

procesistas que conforman el sistema. El diagrama de la figura 47 se lee de la siguiente manera:

1. La colección **Responsable Bobinas Defectuosas** recibe el mensaje *obtener_gestion_de_responsables(...)*
2. La colección **Responsable Bobinas Defectuosas** envía el mensaje *obtener_responsable(...)* a la instancia **Responsable**.
3. La colección **Responsable Bobinas Defectuosas** envía el mensaje *obtener_bobina_defectuosa(...)* a la Colección **Bobina Defectuosa**.
4. La colección **Bobina Defectuosa** envía el mensaje *obtener_datos_bobina_defectuosa(...)* a la instancia **Datos Bobina**.
5. La colección **Bobina Defectuosa** envía el mensaje *obtener_linea(...)* a la instancia **Linea Origen**
6. La colección **Responsable Bobinas Defectuosas** envía el mensaje *obtener_causas_bobinas_defectuosa(...)* a la Colección **Causas Defectos Bobina**.
7. La colección **Causas Defectos Bobina** envía el mensaje *obtener_causa(...)* a la instancia **Causa**.
8. La instancia **Causa** envía el mensaje *obtener_acciones(...)* a la colección **Acciones**.

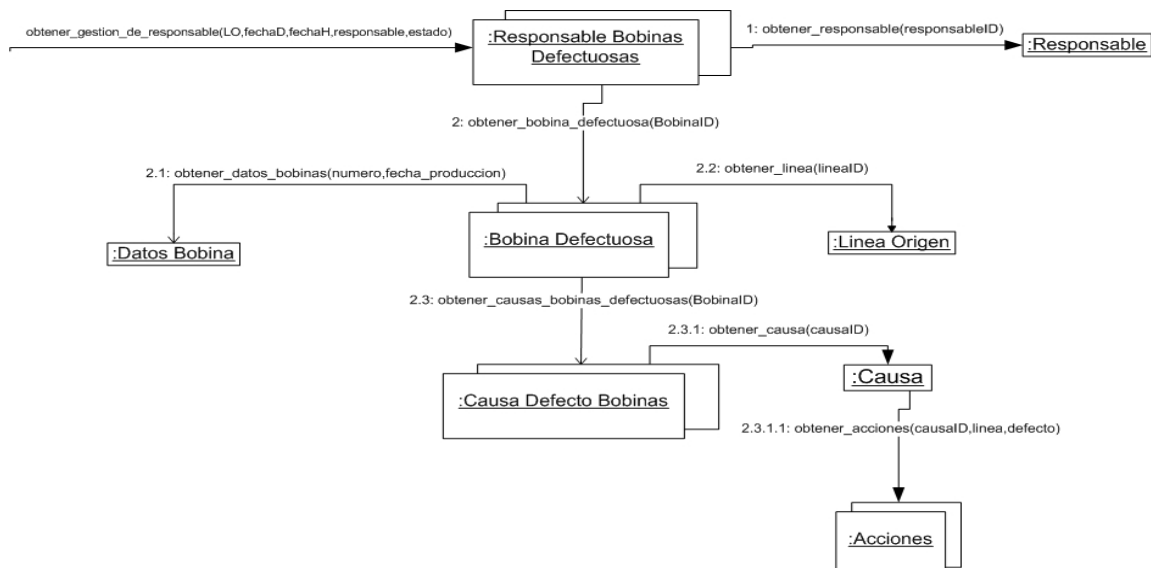


Figura 47. DI 8: Gestión de Procesistas.

4.3.3.1.2 DI 9: Gestión de defectos.

El diagrama de la figura 48 refleja la realización del Caso de Uso *UC 13: Gestión de Defectos*, donde se obtendrá como resultado las causas asignadas por defecto y las acciones asignadas por causa. El diagrama de la figura 48 se lee de la siguiente manera:

1. La colección **Responsable Bobinas Defectuosas** recibe el mensaje *obtener_gestion_defectos(...)*
2. La colección **Responsable Bobinas Defectuosas** envía el mensaje *obtener_bobina_defectuosa(...)* a la instancia **Bobina Defectuosa**.
3. La instancia **Bobina Defectuosa** envía el mensaje *obtener_linea(...)* a la instancia **Línea Origen**.
4. La instancia **Bobina Defectuosa** envía el mensaje *obtener_defecto(...)* a la instancia **Defecto**.
5. La instancia **Bobina Defectuosa** envía el mensaje *obtener_causas_bobinas_defectuosa(...)* a la Colección **Causas Defectos Bobina**.
6. La colección **Causas Defectos Bobina** envía el mensaje *obtener_causa(...)* a la instancia **Causa**.
7. La instancia **Causa** envía el mensaje *obtener_acciones(...)* a la colección **Acciones**.

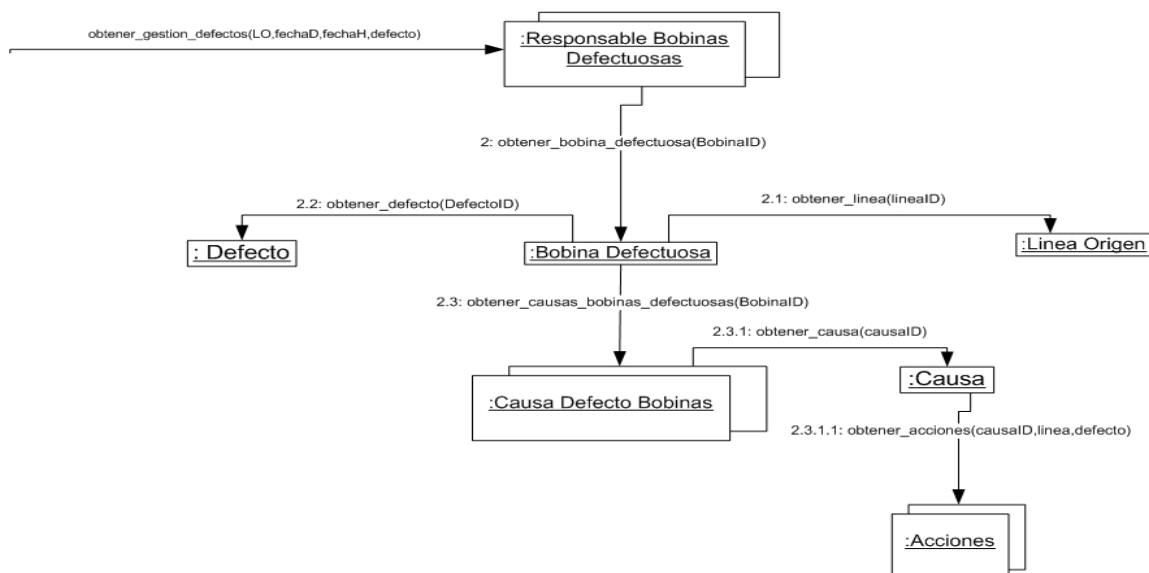


Figura 48. DI 9: Gestión de Defectos.

4.3.3.1.3 DI 10: Gestión Acciones Asignadas.

En este diagrama (figura 49), se ve reflejado la realización del caso de uso *UC 14: Gestión Acciones Asignadas*, donde se obtiene el listado de acciones asignadas a las causas de acuerdo a los parámetros de entrada del mensaje *obtener_acciones_asignadas(...)* que recibe la colección **Acciones**.

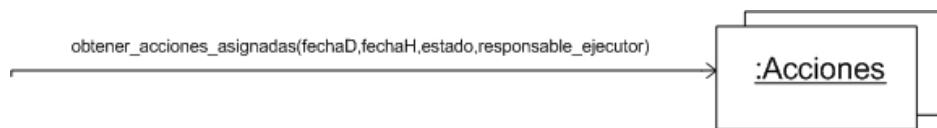


Figura 49. DI 10: Gestión de Defectos.

4.3.3.1.4 DI 11: Ver Pareto Causas.

El diagrama de la figura 50 se lee como sigue:

1. La colección **Causa Defecto Bobina** recibe el mensaje *obtener_cantidad_causas_por_defecto(...)*
2. La colección **Causa Defecto Bobina** envía el mensaje *obtener_cantidad_causas(...)* a la instancia **Causa**.

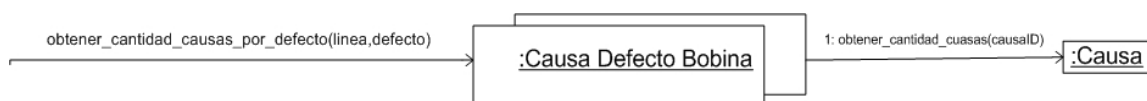


Figura 50. DI 11: Ver Pareto causas.

4.3.3.1.5 DI 12: Ver Pareto Acciones.

En el diagrama de la figura 51 se refleja la realización del caso de uso *UC 16: Ver Pareto Acciones*, donde se obtiene las veces que un conjunto de acciones son asignadas a una causa por causa, defecto y línea.

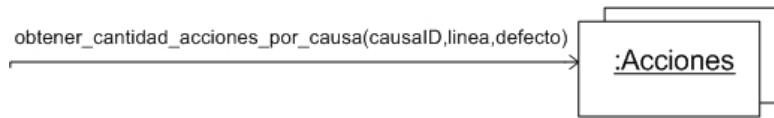


Figura 51. DI 12: Ver Pareto Acciones.

4.3.3.2 Diagrama de Clases de Diseño.

El Diagrama de Clases de Diseño correspondiente a esta iteración sufre modificaciones respecto al anterior (iteración II) en lo que respecta a operaciones, ya que surgieron nuevas operaciones para el cumplimiento de los objetivos del sistema. El nuevo diagrama se muestra en la figura 52:

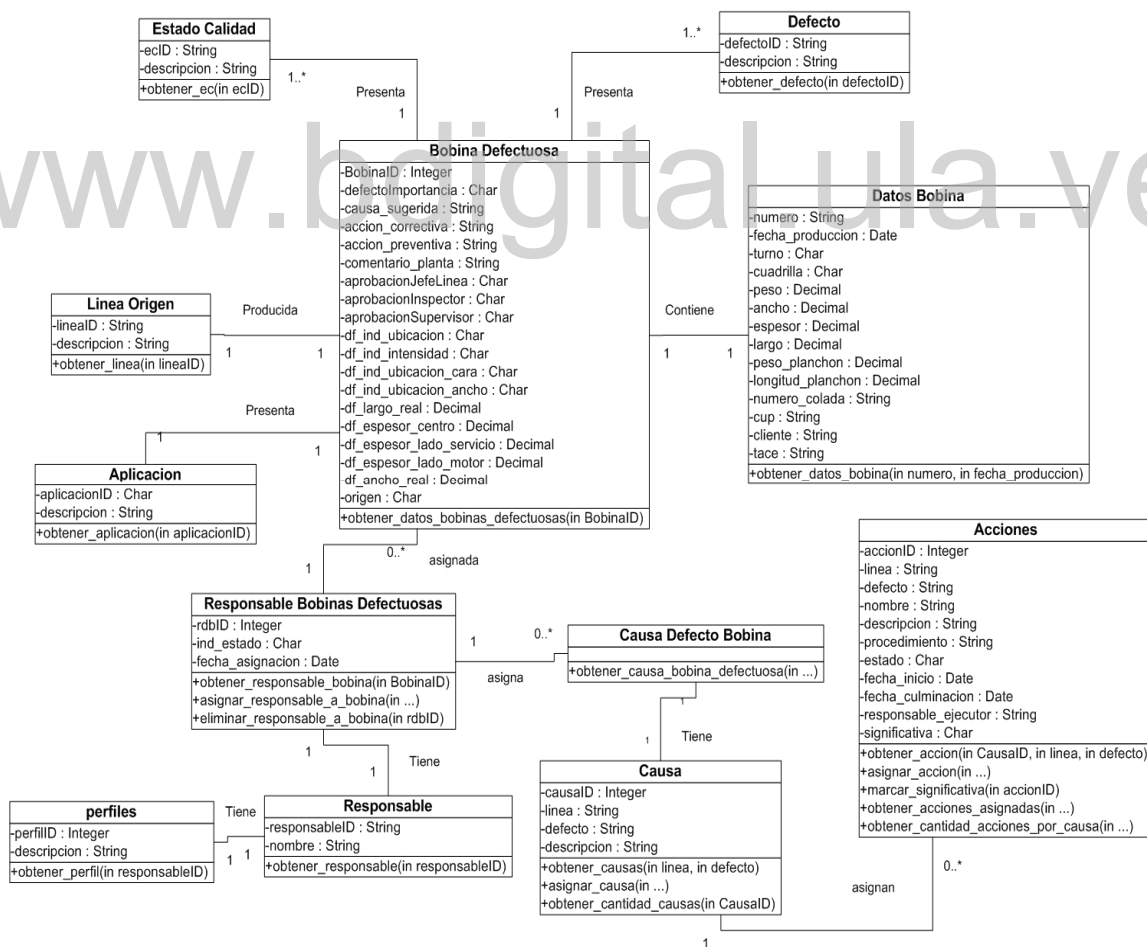


Figura 52. Diagrama de clases de Diseño – Iteración III.

4.3.4 IMPLEMENTACIÓN.

En esta etapa de la Iteración III al igual que en la Iteración anterior se procede a codificar y probar cada uno de los procedimientos almacenados que corresponden con los mensajes provenientes de los diagramas de interacción. Estos procedimientos almacenados se muestran en la tabla 13:

Procedimiento Almacenado	Parámetros de entrada	Resultado
web_sgc_obtener_gestion_de_responsable	LO, fechaD, fechaH, responsable, estado.	Responsables con sus bobinas asignadas causas y acciones.
web_sgc_obtener_gestion_defectos	LO, fechaD, fechaH, defecto	Defectos con sus causas asignadas y acciones por causas.
web_sgc_obtener_acciones_asignadas	fechaD, fechaH, estado, responsable_ejecutor	Listado con acciones asignadas a causas.
web_sgc_obtener_cantidad_causas_por_defecto	linea, defecto	Causas con la cantidad de veces que fueron asignadas a un defecto.
Web_sgc_obtener_cantidad_acciones_por_causa	causasID, linea,	Acciones con

	defecto	la cantidad de veces que fueron asignadas por causa.
--	---------	--

Tabla 13. Definición de Procedimientos Almacenados.

Vale la pena resaltar que para la construcción de cada uno de estos procedimientos almacenados se deben construir una serie de *funciones y vistas* necesarias para mejorar el rendimiento de los procedimientos, y lograr uno de los requisitos generales del sistema; **tiempos de respuestas cortos**. De igual manera se continuara con la codificación de la clase Data_Access.vb que se encargara de la comunicación con la base de datos para hacer posible la ejecución de estos procedimientos almacenados.

4.3.5 GLOSARIO.

Término	Definición e Información	Alias
Tipo de Estado	Corresponde al estado que se desea que posean las bobina a visualizar por responsabe, puede ser con Plan completo, Plan incompleto, sin Plan	estado
Plan	Corresponde a la asignación de causas y acciones	
Plan Completo	Cuando a una bobina se le asigna una causas a su defecto y a dicha causas se le asigna una acción a ejecutar	
Plan Incompleto	Cuando una bobina se le ha asignado solamente la causas del defecto	
Sin Plan	Bobinas sin causas ni acciones asignadas.	

Tabla 14. Glosario – Fase de Elaboración – Iteración III

4.3.6 PLAN DE ITERACIÓN.

Para la fase de construcción se implementó la mayor parte del sistema. Se realizó el diseño de las pantallas y de los diagramas donde se especifica claramente la interacción entre las capas de la Arquitectura del sistema WEB, es decir la interacción entre la capa IU, la Capa de Negocio y la Capa de Datos. Una vez que se finalizó la implementación de cada una de las pantallas con la respectiva comunicación entre Capas, se comenzó a realizar las pruebas finales previas a la puesta en producción del sistema. Paralelamente se realizó la redacción del Manual de Usuarios del Sistema.

www.bdigital.ula.ve

Capítulo 5

Fase de Construcción.

5 Introducción

El propósito de esta fase, se centra esencialmente en terminar de construir la aplicación, preparar el despliegue (diseño de pantallas Web) y la escritura y creación de guías de usuario y ayuda on-line. Mientras la elaboración se basó en la construcción del núcleo de la arquitectura, la fase de construcción se encarga de terminar de construir e implementar esta arquitectura.

5.1 DISEÑO DE LA INTERFAZ.

En esta etapa de la fase de construcción se muestra el diseño general de la interfaz gráfica del sistema. La interfaz de usuario corresponde al diseño visual con el que interactuará el *usuario interno del sistema*. Debido a que el sistema está diseñado para ser utilizado por personas con conocimientos básicos de computación, la interfaz debe ser presentada de manera clara, consistente, y con un lenguaje escrito y visual común para no confundir a los usuarios. Esta interfaz consistirá en un grupo de pantallas Web, las cuales contendrán diversos elementos que permitirán operar fácilmente el sistema. Entre estos elementos destacan tablas de datos (DataGrid en ASP.NET), listas desplegables de selección (DropDownList en ASP.NET), cajas de selección (CheckBox en ASP.NET), Listas de Selección fijas (ListBox en ASP.NET), Botones de acción (Button en ASP.NET) y cajas de texto (TextBox).

La figura 53 muestra el diseño general seguido para la elaboración de cada una de las pantallas Web del sistema. Este diseño es el que actualmente se implementa en los diferentes sistemas que se desarrollan en SIDOR (Estándares de la Empresa), por esta razón el sistema presentado en este trabajo se guiará por este diseño durante la construcción de las pantallas Web.

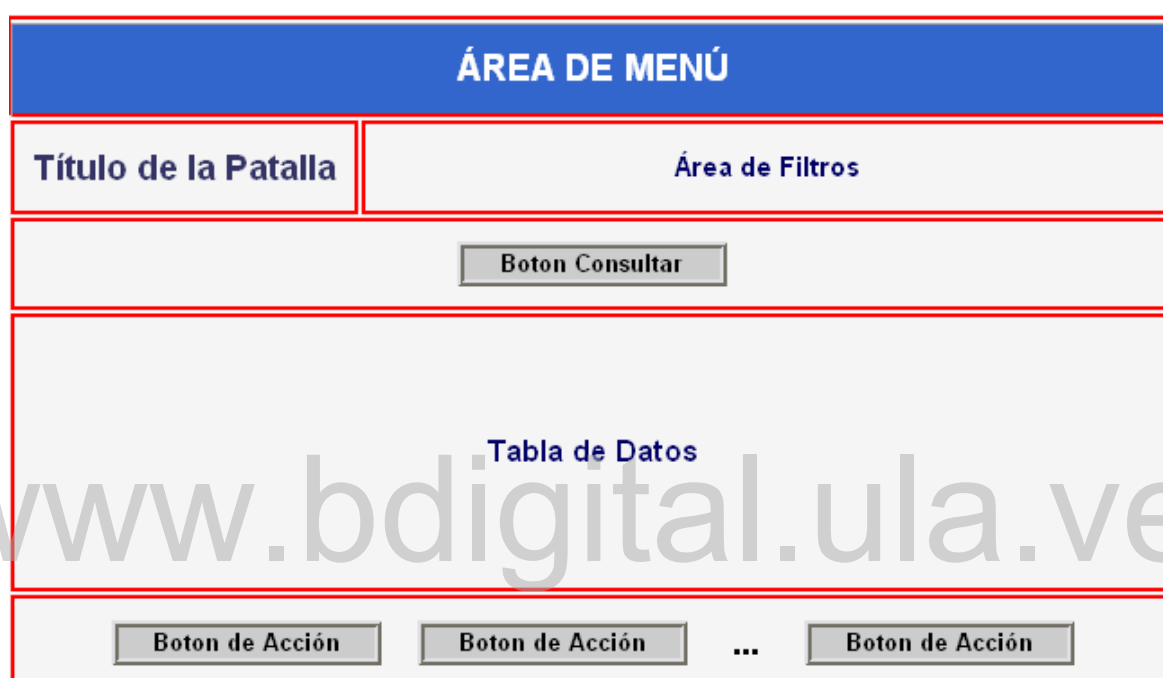


Figura 53. Diseño general de las pantallas.

Cada uno de los componentes de este diseño se especifica a continuación:

- **Área de Menú:** En esta sección de la pantalla se coloca un Menú configurable XML, en el cual se especifican cada una de las direcciones para acceder a las diferentes pantallas principales del sistema. Este menú corresponde a un control reutilizable diseñado en la empresa y utilizado en todos los sistemas Web que se desarrollan dentro de la misma.
- **Título de la pantalla:** En esta sección se coloca el título correspondiente a la pantalla que se está utilizando.

- **Área de Filtros:** En esta sección se ubican todos los tipos de controles utilizados para filtrar los pedidos que se hacen al sistema (Solicitudes de visualización de Datos). Estos controles pueden ser listas desplegables, cajas de selección múltiple, cajas de texto y calendarios.
- **Botón consultar:** Este control es utilizado en las pantallas para que el usuario haga la solicitud al sistema de visualización de datos.
- **Tabla de Datos:** En esta sección se ubican las tablas de datos (DataGrid) encargadas de visualizar los diferentes datos cualitativos característicos de las bobinas defectuosas.
- **Botones de Acción:** Estos botones tienen como objetivo enviar al sistema solicitudes tales como: Guardar, Volver a la Página anterior, Ir a la página siguiente, Enviar, etc.

5.1.1 DIAGRAMA DE FLUJO DE NAVEGACIÓN DEL SGC.

El figura 54 muestra el diagrama general de navegación entre pantallas del sistema. SGC corresponde a la pantalla Principal del sistema, una vez ingresado a esta pantalla es posible visualizar la ayuda On-Line, donde se podrá acceder a una serie de documentos que conforman la documentación del sistema (Manuales de Usuarios). El icono LAC o LAF, quiere decir que bien sea Laminación en caliente o Laminación en Frío (Áreas de Laminación), será posible acceder a cada una de las pantallas del sistema.

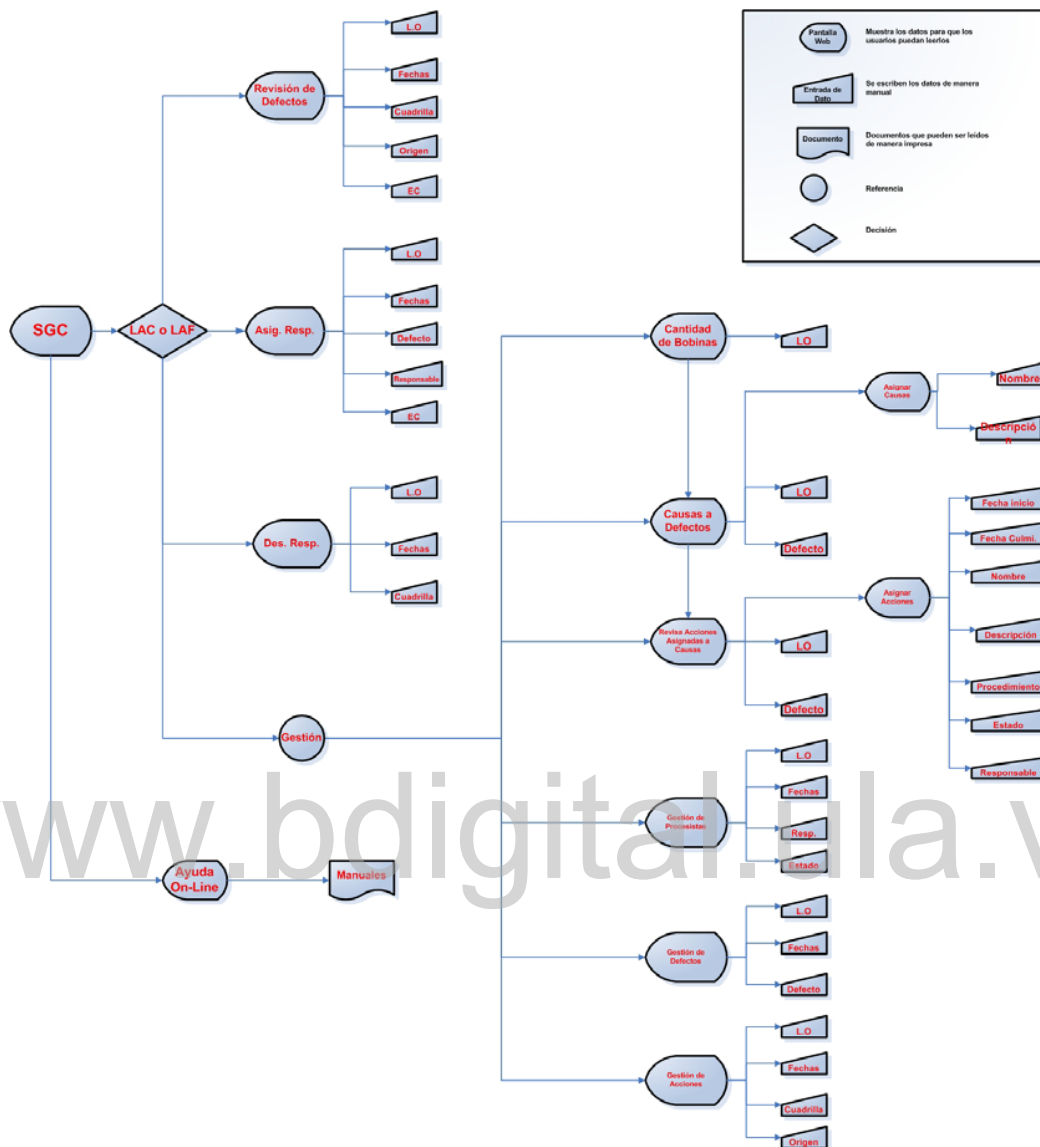


Figura 54. Diagrama de flujo de navegación del sistema.

5.2 CONEXIÓN ENTRE LAS CAPAS DE LA APLICACIÓN.

Las aplicaciones se organizan en capas lógicas que separan los aspectos más importantes de dicha aplicación, estas son: la Capa de UI (interfaz de Usuario), Capa de Negocios (o Capa de Dominio) y la Capa de Datos. A continuación se especifica de forma general como se realiza la comunicación entre la Capa de Interfaz de Usuario UI, la Capa de Negocio (o capa de dominio) y la Capa de Datos del sistema en cada una de las pantallas

del sistema, para cumplir con los objetivos de los diferentes usuarios (Realización de los Casos de Uso).

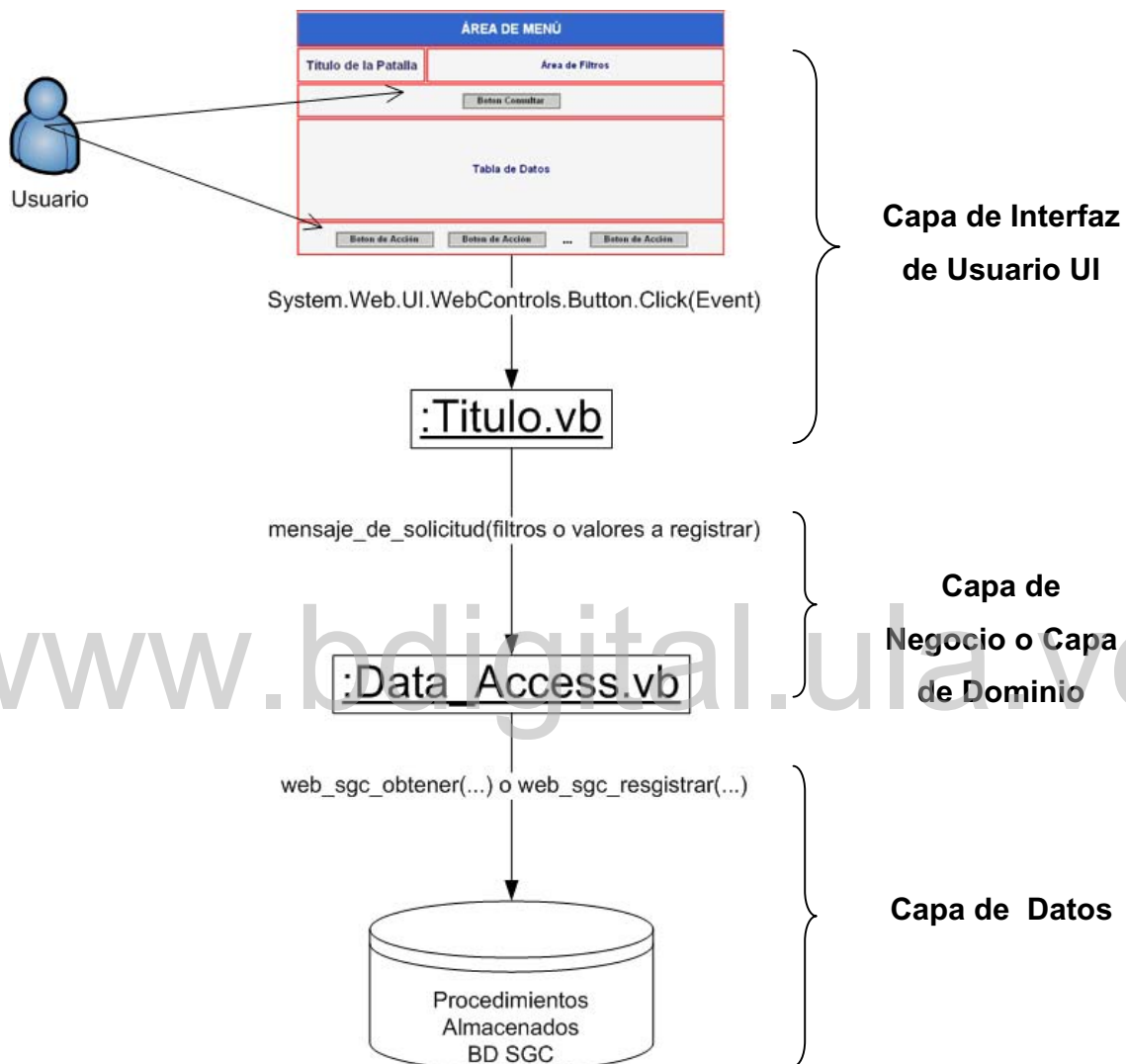


Figura 55. Conexión de la capa de UI, Negocio y Datos.

Cada vez que un usuario del sistema acciona cualquiera de los controles de acción de la una pantalla (Botón consultar o botón de acción) se genera un evento denominado click interno de la aplicación (Ver figura 55). Este evento es manejado por la clase `Titulo.vb` la cual es la encargada de manejar todos los eventos de los diferentes controles creados en la pantalla `Titulo`. Esta clase es instanciada automáticamente por el .Net al momento de la creación de la pantalla. Todos estos eventos que son generados por el usuario, se

encuentran dentro de la Capa de Interfaz de Usuario UI. Una vez desencadenados estos eventos entra en juego la Capa de Negocios, ya que es esta la que se encarga de decidir que hacer con las solicitudes de los usuarios (Manejar la lógica). Para ello está la Clase `Data_Access.vb` donde se encuentra albergada toda la lógica necesaria para realizar la comunicación con la Capa de Datos.

Para que la Capa de Negocio se comunice con la Capa de Datos y debido a que la Base de Datos se encuentra implementada en SQL SERVER es necesario la creación de cuatro Objetos propios de .Net, estos Objetos Son:

- `SqlConnection`: Al cual se le entrega la cadena de conexión (Ubicación del Servidor de Base de Datos, usuario para la conexión y password).
- `SqlCommand`: Al cual se le entrega la conexión (`SqlConnection`), nombre del procedimiento almacenado y los parámetros que este recibe.
- `System.Data.Dataset`: Donde se almacenará la respuesta al llamado del Procedimiento almacenado.
- `SqlDataAdapter`: Al cual se le entrega el `SqlCommand` y el `DataSet` para que solicite la ejecución del procedimiento almacenado y almacene la respuesta dentro del `DataSet`.

Una vez almacenada la respuesta de la Capa de Datos dentro del `DataSet` será posible volver a la Capa de Interfaz de Usuario y Visualizar la Respuesta de la Capa de Datos en la pantalla. Recordemos que el código fuente de la clase `Data_Access.vb` se encuentra albergado en la sección de codificación del CD adjunto al Proyecto. Antes de Implementar cada una de estas interfaces de comunicación en cada una de las pantallas del sistema, las mismas fueron sometidas a un amplio número de Pruebas, donde se obtuvieron resultados deseados y no deseados. Estos resultados sirvieron de retroalimentación para terminar el proceso de refinamiento de cada una de estas interfaces para cada pantalla del sistema. A pesar de estas pruebas, en la Fase de Transición, donde se realizan las últimas pruebas para la puesta en producción del sistema, se realizarán nuevas pruebas, las cuales se harán con ayuda de los usuarios del sistema y se tabularán los resultados obtenidos.

5.3 ESPECIFICACIONES COMPLEMENTARIAS.

Restricciones de Implementación. Manejo de Excepciones

El Manejo de Errores del sistema debe hacerse a través de instrucciones *Try ... Catch*, cuya sintaxis es la siguiente:

Try

....

Código a Ejecutarse

...

Catch

...

Manejo de Error (Variable Err)

End Try

Dentro del cuerpo **Código a Ejecutarse**, se coloca todo el código que se espera que se ejecute correctamente. En caso de que este falle, es posible realizar cierta acción dentro de la sección **Catch**. Para el caso de este sistema (SGC), la acción a ejecutar será redireccionar la pantalla a una en la cual se especifique un mensaje solicitando disculpas al usuario. Esta pantalla una vez se levante debe enviar automáticamente un Mail (Usando para esto el Sistema de Envíos de Mail de la Empresa), a las personas encargadas de realizar mantenimiento al sistema. Esto es para los casos donde el sistema pueda fallar aleatoriamente. Para los casos de entradas de datos incorrectas se deberá emitir un mensaje de ALERTA al usuario indicándole el error de entrada de datos y rechazando la entrada (En la Fase de Transición se especificarán estos mensajes de alerta).

5.4 GLOSARIO.

Término	Definición e Información	Alias
BD	Base de Datos	
SqlConnection	Objeto propio de .Net encargado de configurar la conexión para establecer comunicaciones con BD implementadas en SQL SERVER	
SqlCommand	Objeto propio de .Net encargado de configurar el comando que se ejecutará en la BD implementada en SQL SERVER	
System.Data.Dataset	Objeto propio de .Net encargado de almacenar respuestas a llamados de comandos SQL en las BD. Puede no ser SQL SERVER	
SqlDataAdapter	Objeto propio de .Net encargado de ejecutar la solicitud a la BD implementada en SQL SERVER para la ejecución del comando configurado en el SqlCommand.	
<i>Try ... Catch</i>	Instrucción usada para el manejo de excepciones.	

Tabla 15. Glosario – Fase de Construcción

5.5 PLAN DE ITERACIÓN.

En la Fase de Transición se tiene planificada la implementación de cada una de las pantallas del sistema utilizando para esto el diseño especificado en esta Fase. También se pretende realizar todas las Pruebas restantes antes de la implementación. Se construirá un diagrama de Implementación del sistema y finalmente se realizará la puesta en producción del mismo. Se refinarán y culminarán los Manuales de usuario para que una vez listos se realicen las diferentes jornadas de capacitación para los usuarios que no intervinieron durante el desarrollo del SGC.

Capítulo 6

Fase de Transición.

6 Introducción

Esta Fase del proceso unificado esta dedicada a la implementación final y a las pruebas finales del sistema, aquí se realiza la puesta en producción. Para esto se debe implementar las pantallas siguiendo el diseño especificado en el Capítulo 5 y finalizar los manuales respectivos de usuario. También se debe antes de culminar esta Fase realizar jornadas de capacitación a los usuarios, de manera tal que al momento de que salga el sistema a producción los mismos tenga el conocimiento necesario sobre cómo acceder a las diferentes pantallas y como manipularlas.

6.1 Implementación de las Pantallas.

En esta etapa de la fase de Transición se describe la implementación de cada una de las pantallas correspondientes a los casos de uso descritos en las Fases anteriores del UP. Antes de comenzar a especificar el diseño de cada de las pantallas, se muestra en la figura 56 el Menú encargado de proveer la ubicación de cada una de las pantallas principales del sistema.



Figura 56. Menú

Este menú estándar usado en todos los sistemas desarrollados dentro de SIDOR, es configurado a través de un archivo XML, el cual debe ser creado de forma tal que contenga la dirección URL de cada una de las pantallas principales que conforman el sistema en desarrollo. Además de esto por estándares de la empresa se debe colocar el logotipo del sistema en este menú. Haciendo referencia al formato especificado en la etapa de diseño de la interfaz del capítulo 5 (figura 53), este menú se coloca en todas las pantallas principales del sistema en la sección Menú.

6.1.1 Pantalla Revisión de Defectos

Esta pantalla corresponde a la implementación del *Caso de Uso UC1: Revisar Defectos* descrito en el capítulo 3. Siguiendo el formato general, se listan a continuación cada uno de los elementos que conforman la pantalla:

- **Título:** Para no perder la referencia al caso de uso UC1 se coloca como título de esta pantalla **Revisión de Defectos**.
- **Filtros:** La figura 57 muestra cada uno de los controles ubicados en la sección de filtros del Formato general.

The image shows a web interface for filtering defects. At the top, there's a search bar with 'L.O: TA 2'. To its right are two date pickers both set to '28/04/2005'. Further right is a dropdown menu labeled 'C: Todas'. Below these, there's a section for 'Origen' with checkboxes for 'P' (checked), 'A', and 'D'. To the right of this is a section for 'E.C.' with checkboxes for '30', '20', '22' (checked), '70', '80', '60', '50', and '00'.

Figura 57. Filtros de la pantalla Revisión de Defectos.

Todos estos controles que se ubican en esta sección de Filtros, corresponden a los parámetros que el usuario debe introducir manualmente para que el sistema pueda cumplir con su objetivo principal (Ver texto caso de uso UC1).

- **Botón Consulta:** a través de este botón el usuario podrá hacer la solicitud al sistema para que este cumpla su objetivo.
- **Datos:** Los datos que el usuario desea visualizar en esta pantalla fueron solicitados por este durante los talleres de requisitos para la escritura de los Casos de Uso. En

la figura 58 se especifican estos datos (Cada dato se encuentran definidos en el anexo A – clase bobina defectuosa)

Listado de Bobinas																			
Fecha prod.	C.D.	Bobina	EC	Línea Origen	Peso	CUP	Cliente	Tace	Prog.	T	C	O/V	Aplic.	C.S	A.P	A.C	I	S	J.P
28/4/2005	151	2537183	20	TA 2	13770	5742	INMETALSA	0303	734	1		SA04556							
28/4/2005	451	1564803	20	TA 2	10370	6012	ENVENSA	0005	734	1		SA06704							
28/4/2005	241	2537191	20	TA 2	13000	6042	SOCOMINTER	0003	734	2		MT04578							
28/4/2005	341	2537194	70	TA 2	8095	6012	DOMIN-VALE	0005	734	2		SA06396							
28/4/2005	135	2537195	20	TA 2	3395	6012	DOMIN-VALE	0005	734	2		SA06396							
1																			

Figura 58. Datos a visualizar por el usuario.

La siguiente figura muestra la implementación final para la pantalla Revisión de Defectos:

Listado de Bobinas																			
Fecha prod.	C.D.	Bobina	EC	Línea Origen	Peso	CUP	Cliente	Tace	Prog.	T	C	O/V	Aplic.	C.S	A.P	A.C	I	S	J.P
28/4/2005	151	2537183	20	TA 2	13770	5742	INMETALSA	0303	734	1		SA04556		i	i	i	?	?	×
28/4/2005	451	1564803	20	TA 2	10370	6012	ENVENSA	0005	734	1		SA06704		i	i	i	?	?	×
28/4/2005	241	2537191	20	TA 2	13000	6042	SOCOMINTER	0003	734	2		MT04578		i	i	i	?	?	?
28/4/2005	341	2537194	70	TA 2	8095	6012	DOMIN-VALE	0005	734	2		SA06396		i	i	i	?	?	?
28/4/2005	135	2537195	20	TA 2	3395	6012	DOMIN-VALE	0005	734	2		SA06396		i	i	i	?	?	?

Figura 59. Pantalla Revisión de defectos.

Al igual que en la pantalla anterior, las demás pantallas principales del sistema fueron implementadas siguiendo este mismo esquema de implementación, basado en el texto de los casos de uso y aplicando el formato descrito en el capítulo 5.

La implementación correspondiente a las demás pantallas principales del sistema: Analizar Bobina y Asignar Responsable, Eliminar Responsables Asignados, consultar

cantidad de bobinas asignadas, etc. (Ver figura 40) se encuentra en la sección de Anexos de este trabajo. (Anexo C).

6.2 Pruebas Finales del Sistema.

Las Pruebas Finales del sistema tienen como finalidad verificar si las diferentes interfaces diseñadas e implementadas corresponden con lo plasmado en los casos de uso descritos en los capítulos anteriores. Hay que recordar que todas estas pruebas serán realizadas en conjunto con los usuarios, de manera tal que se pueda ir ajustando paralelamente detalles y correcciones echas por los mismos. Existen diferentes tipos de pruebas automatizadas, las cuales pueden ser llevadas a cabo mediante el computador con el fin de asegurara y solventar el correcto funcionamiento de los diversos módulos existentes [Barrios 2001], y para el desarrollo de las pruebas del sistema se tomará en cuneta las siguientes:

- **Pruebas Caja Blanca:** Los datos de prueba son derivados del comportamiento interno del sistema, cubriendo de esta manera el comportamiento lógico del sistema.
- **Pruebas Caja Negra:** Los datos de prueba son colocados sin considerar el comportamiento interno del sistema, utilizando valores ubicados justo en los límites permitidos y fuera de ellos.

A continuación se muestran algunas de las distintas pruebas realizadas y los resultados obtenidos en cada una de ellas:

6.2.1 Pruebas sobre la pantalla Revisión de defectos:

- La tabla 16 Muestra los valores introducidos y el resultado obtenido para solicitar la consulta de un listado de bobinas, donde se introduce un rango de fechas no válido:

Parámetro	Valor
L.O	TA 2
FechaD	30/04/2005

FechaH	28/04/2005
Cuadrilla	Todas
Origen	P
EC	20,22,70,80,60
RESULTADO	Rango de fechas Inválido

Tabla 16. Datos de prueba 1 – pantalla Revisión de defectos.

La figura 60 muestra la manera como el sistema presenta el mensaje de error al usuario para que sea corregido el rango de fechas:



Figura 60. Rango de fechas inválido.

- La tabla 17 Muestra los valores introducidos y el resultado obtenido para solicitar la consulta de un listado de bobinas, donde se introduce valores correctos:
-

Parámetro	Valor
L.O	TA 2
FechaD	28/04/2005
FechaH	28/04/2005
Cuadrilla	Todas
Origen	P
EC	20,22,70,80,60
RESULTADO	Visualización del listado de Bobinas.

Tabla 17. Datos de prueba 2 – pantalla Revisión de defectos.

La figura 61 muestra como es presentado el resultado por el sistema:

Fecha prod.	C.D.	Bobina	EC	Línea Origen	Peso	CUP	Cliente	Tace	Prog.	T	C	O/V	Aplic.	C.S	A.P	A.C	I	S	J.P
28/4/2005	151	2537183	20	TA 2	13770	5742	INMETALSA	0303	734	1		SA04556		i	i	i	?	?	?
28/4/2005	451	1564803	20	TA 2	10370	6012	ENVENSA	0005	734	1		SA06704		i	i	i	?	?	?
28/4/2005	241	2537191	20	TA 2	13000	6042	SOCOMINTER	0003	734	2		MT04578		i	i	i	?	?	?
28/4/2005	341	2537194	70	TA 2	8095	6012	DOMIN-VALE	0005	734	2		SA06396		i	i	i	?	?	?
28/4/2005	135	2537195	20	TA 2	3395	6012	DOMIN-VALE	0005	734	2		SA06396		i	i	i	?	?	?

Figura 61. Listado de Bobinas Defectuosas.

6.2.2 Pruebas sobre la pantalla Asignación de Responsables:

- La tabla 18 Muestra los valores introducidos y el resultado obtenido al solicitar la consulta de un listado de bobinas defectuosas a asignar responsables:

Parámetro	Valor
Línea	Todas
FechaD	28/04/2005
FechaH	28/04/2005
Responsable	Todos
Aplicación	LPO
EC	20,22,70,80,60
RESULTADO	Visualización del listado de Bobinas.

Tabla 18. Prueba pantalla asignación de responsables.

La figura 62 ilustra el resultado de esta prueba.

Listado de Bobinas

Responsable	Fecha Prod.	C.D.	Bobina	EC	Línea	Peso	CUP	Tace	Prog.	T	C	O/V	C.S	A.P	A.C	I	S	J.P
	28/4/2005	170	1563956	22	TM 2	12136	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	1563956	22	TM 2	12136	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	170	1563960	22	TM 2	13236	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	1563960	22	TM 2	13236	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	170	2536963	22	TM 2	12972	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	2536963	22	TM 2	12972	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	170	1563955	22	TM 2	11550	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	1563955	22	TM 2	11550	6042	0004	469	2		SA05134	i	i	i	?	?	?

Figura 62. Listado de bobinas a asignar responsables

Una vez visualizado el listado de bobinas defectuosas, se procede a seleccionar de la primera columna de la tabla de datos el responsable a asignar a esta bobina y a asignarlo. La figura 63 ilustra el resultado de esta prueba:

Listado de Bobinas

Responsable	Fecha Prod.	C.D.	Bobina	EC	Línea	Peso	CUP	Tace	Prog.	T	C	O/V	C.S	A.P	A.C	I	S	J.P
	28/4/2005	170	1563956	22	TM 2	12136	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	1563956	22	TM 2	12136	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	170	1563960	22	TM 2	13236	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	1563960	22	TM 2	13236	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	170	2536963	22	TM 2	12972	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	2536963	22	TM 2	12972	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	170	1563955	22	TM 2	11550	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	1563955	22	TM 2	11550	6042	0004	469	2		SA05134	i	i	i	?	?	?

Figura 63. Lista de responsables posibles a asignar.

Como resultado se obtiene el mensaje ilustrado en la figura 64:



Figura 64. Resultado de una asignación

6.2.3 Pruebas sobre la pantalla Eliminación de Responsables:

- La siguiente prueba mostrada en la tabla 19 especifica la entrada y el resultado obtenido al eliminar un responsable de una bobina:

Parámetro	Valor
L.O	Todas
FechaD	28/04/2005
FechaH	28/04/2005
Responsable	SIERRAALTA Angel
RESULTADO	Responsable Eliminado con éxito.

Tabla 19. Prueba pantalla eliminación de responsable.

6.2.4 Pruebas sobre la pantalla Cantidad de Bobinas Asignadas:

- La Tabla 20 muestra los datos introducidos y el resultado obtenido en la pantalla para solicitar el listado de la línea – defecto - cantidad:

Parámetro	Valor
L.O	APL
RESULTADO	Listado Visualizado con éxito

Tabla 20. Prueba pantalla cantidad de bobinas asignadas.

6.2.5 Pruebas sobre la pantalla Revisar Bobinas Asignadas:

- La Tabla 21 muestra los datos introducidos y el resultado obtenido en la pantalla Revisar Bobinas Asignadas (Asignación de Causas a Defectos) para solicitar el listado detallado de bobinas asignadas:

Parámetro	Valor
L.O	APL
Defecto	Falta Certf. Recubrimiento – 705
RESULTADO	Listado con bobinas asignadas. No existen causas asignadas al defecto.

Tabla 21. Prueba pantalla revisar bobinas asignadas.

6.2.6 Pruebas sobre la pantalla Revisar Acciones Asignadas a Causas:

- La Tabla 22 muestra los datos introducidos y el resultado obtenido en la pantalla para solicitar el listado detallado de bobinas asignadas:

Parámetro	Valor
L.O	APL
Defecto	Mal corte de Borde - 376
RESULTADO	No existen Datos.

Tabla 22. Prueba pantalla Acciones Asignadas a Causas.

6.2.7 Pruebas sobre la pantalla Gestión de Acciones Asignadas:

- La Tabla 23 muestra los datos introducidos en la pantalla y el resultado obtenido para solicitar el listado de acciones asignadas a los responsables ejecutores:

Parámetro	Valor
FechaD	29/04/2005
FechaH	29/04/2005
Estado	En Curso
Responsable Ejecutor	SIERRAALTA Angel
RESULTADO	No existen Datos.

Tabla 23. Prueba pantalla gestión de acciones.

6.3 Puesta en Producción del Sistema.

6.3.1 Diagrama de Implementación.

Estos diagramas muestran la distribución de los recursos físicos de equipo (*Hardware*) y el *software* que reside en él, permitiendo así visualizar en que maquina se ejecutan los componentes del sistema. La figura 65 muestra el diagrama de implementación final del SGC, el cual se encontraba durante su etapa de desarrollo en los servidores de desarrollo de Automatización.

El Nodo SIRPRDSQLPPLAF corresponde a uno de los Servidores Cluster de Producción de SIDOR, donde se ubican los componentes de Bases de Datos pertenecientes a los diferentes Sistemas Web que se desarrollan y que están relacionados con Laminación.

Por su lado, el Nodo SIRWEBLAF, corresponde al Servidor donde se deben ubicar todos los componentes de las diferentes aplicaciones Web que se desarrollan dentro de Automatización. Una vez ubicadas tanto la Base de Datos como la Aplicación Web en los correspondientes servidores de producción, el usuario, desde los diferentes Nodos Cliente, podrá a través del Internet Explorer acceder al SGC puesto en producción.

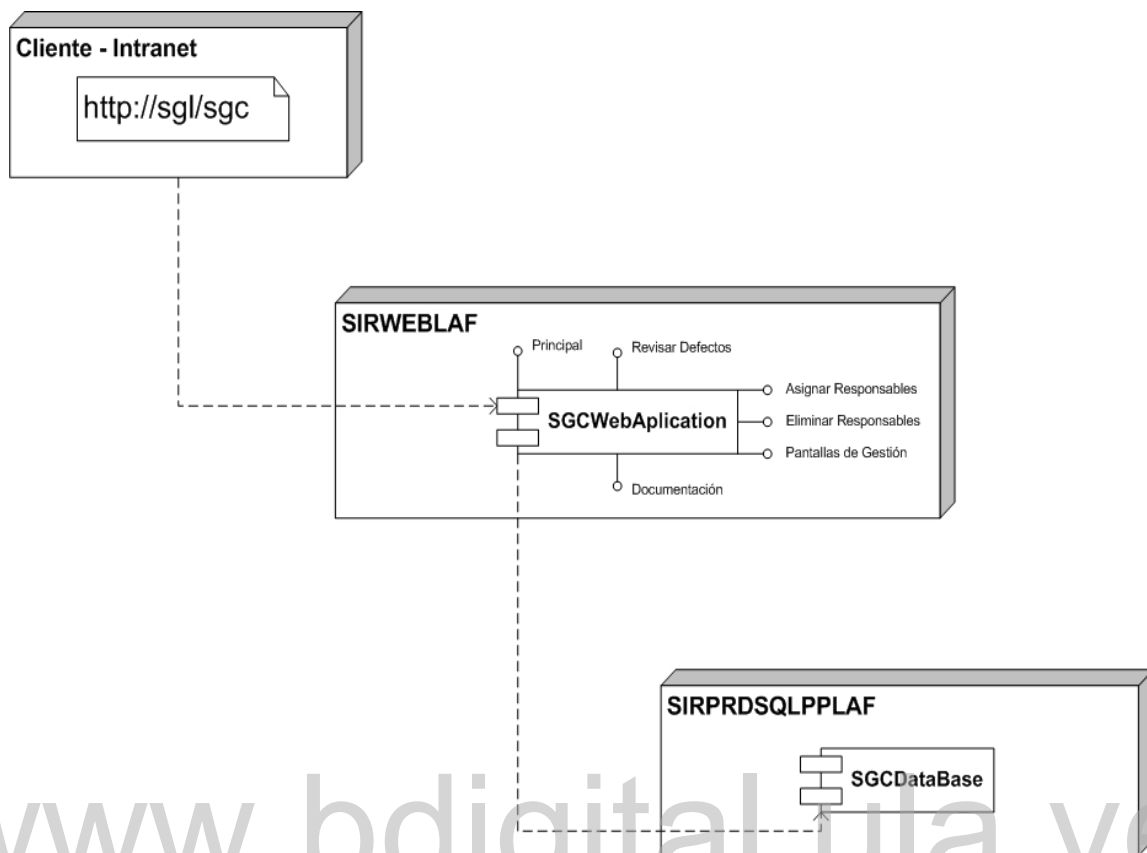


Figura 65. Diagrama de Implementación.

La figura 66 muestra un diagrama de redes de computadoras en el que se puede apreciar cuales son los requisitos por parte de los equipos clientes para poder interactuar con el sistema, es importante mencionar que tanto el Servidor Web SIRWEBLAF como el Servidor Cluster de Base de Datos SIRPRDSQLPPLAF ya están totalmente configurados para ejecutar este tipo de aplicaciones Web. En cuanto al equipo cliente solo es necesario que tenga instalado el software indicado en la figura para poder interactuar con el SGC a través de la intranet.

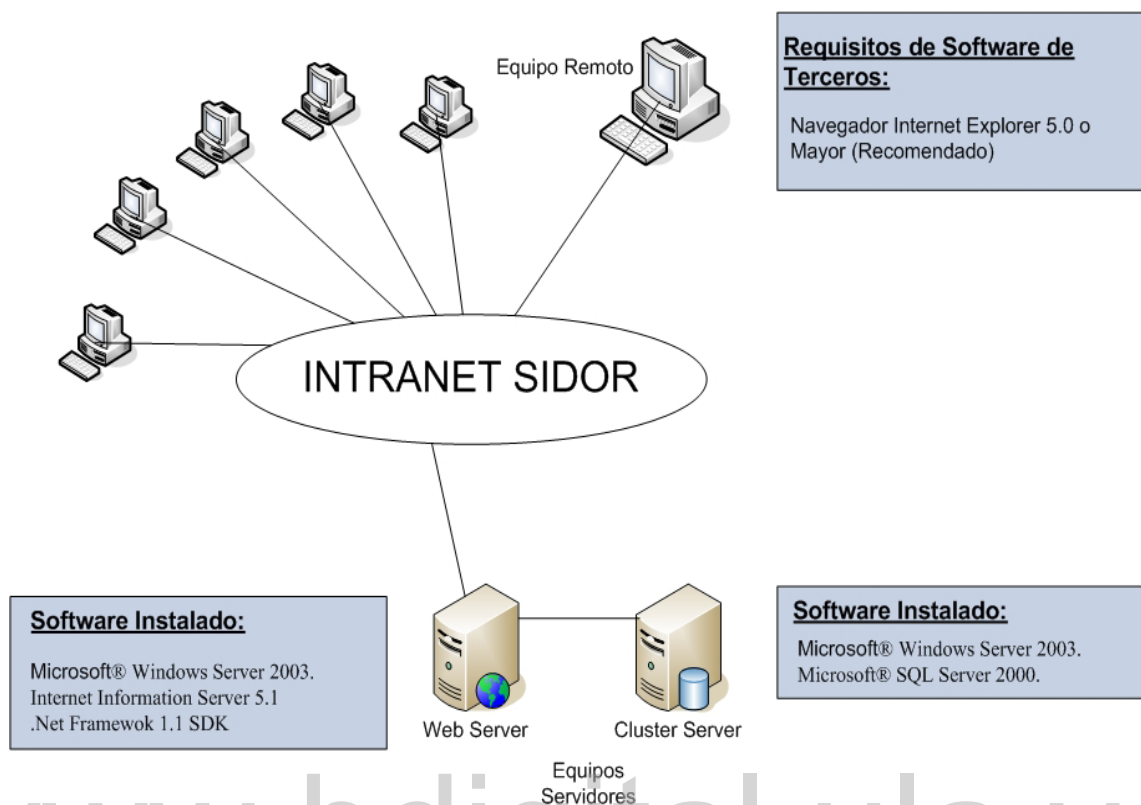


Figura 66. Diagrama de Requisitos de software.

Conclusiones

El aplicar un tipo de Metodología de Desarrollo como lo es El Proceso Unificado, proporciona un soporte teórico y robusto bien definido, que facilita y garantiza el desarrollo de todo el sistema. Debido a que el Proceso Unificado es una Metodología iterativa es posible avanzar en base a retroalimentaciones de cada una de las diferentes iteraciones, ayudando al refinamiento progresivo e incremental del sistema. Por otro lado, la posibilidad de ir implementando en base a lo que se va diseñando durante las diferentes fases del Proceso Unificado permitió obtener al final de cada fase parte del código fuente, el cual se fue probando y refinando hasta obtener la versión final del software.

Otra de las ventajas que tiene el aplicar el Proceso Unificado es la realización de diferentes talleres de requisitos, ya que estos permiten interactuar continuamente con los usuarios quienes son los que finalmente especifican el funcionamiento del sistema.

Desde el punto de vista de mantenimiento, actualización y supervisión del sistema, el haber utilizado UML (Lenguaje de Modelado Universal), será de gran ayuda, ya que los diferentes modelos y diagramas obtenidos durante las etapas de Análisis y Diseño ayudan al entendimiento claro y conciso del sistema, debido a que es relativamente sencillo comprender el funcionamiento del sistema a través de estos Modelos y diagramas.

Por último, las características del lenguaje de programación Visual Basic. Net, hace que la realización de un proyecto de este estilo sea sencilla y clara, ya que por ser un lenguaje Orientado a Objetos, la implementación de los diseños se hace casi mecánica. Además de esto, el entorno Gráfico para la implementación de pantallas Web es bastante elegante, lo que permite la creación de pantallas claras y bien entendibles al usuario.

En lo que al sistema se refiere, será de gran ayuda para los diferentes usuarios que interactúan con el mismo (presentados a lo largo del desarrollo de trabajo), ya que a través de el, será posible controlar todos los defectos presentados en las bobinas lo cual mejorará notablemente la calidad de las bobinas producidas, debido a que se podrán ir eliminando (gracias a la asignación de causas y acciones a los defectos), la ocurrencias de estos defectos en las bobinas.

www.bdigital.ula.ve

Recomendaciones

- Diseñar e implementar nuevas gráficas para facilitar un poco más la gestión realizada por los procesistas.
- Diseñar e implementar un módulo de ingreso y desactivación de usuarios al sistema.
- Agregar un nuevo usuario al sistema con perfil administrativo para que maneje este módulo.
- Realizar un taller de requisitos con los usuarios para definir este módulo de Administración.
- Sugerir a los encargados del sistema de envío de E-Mail por parte de la empresa, mejorara el mismo para que se permita enviar E-Mail con formato HTML.
- Realizar los Manuales de Usuarios para el Módulo de Administración.
- Capacitar al Usuario destinado para el manejo de este Módulo.

Bibliografía

[LARMAN 03] Craig, Larman. “UML y PATRONES. Una Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado”. Segunda Edición. Prentice Hall Madrid 2003.

[MULLER 97] Pierre – Alain, Muller “Modelado de Objetos con UML”. Eyrolles Barcelona 1997.

[SILBERSCHATZ] Abraham, Silberschatz; Henry F, Korth; S, Sudarshan “Fundamentos de Bases de Datos”. Tercera Edición. McGraw – Hill.

[PRESSMAN] Roger S, Pressman “Ingeniería del Software. Un Enfoque Práctico”. Cuarta Edición. McGraw - Hill.

[Groff – Weinberg 02] Groff, James; Weinberg, Paul. “Aplique SQL”. McGraw - Hill. Primera Edición. 2002.

[MATTHEW 02] MacDonald, Matthew. “ASP.NET Manual de referencia”. McGraw - Hill. Primera Edición. 2002.

[REBECCA 02] Riordan, Rebecca. “ADO.NET Aprenda ya”. McGraw - Hill. Primera Edición. 2002.

[FRANCESCO 03] Balena, Francesco. “Programación avanzada con Microsoft Visual Basic.NET”. McGraw - Hill. Primera Edición. 2003.

[RICK 02] Dobson, Rick. “Programación de Microsoft SQL SERVER 2000 con Microsoft Visual Basic.NET”. McGraw - Hill. Primera Edición. 2002.

[BARRIOS 01] Barrios, Judith. “Estudios de estructuras, componentes, interrelaciones metodologías y tecnologías asociadas a los sistemas de información”. Asignatura: Sistemas de Información. Facultad de Ingeniería, **USTEFI** código S-20, Universidad de los Andes. Mérida Venezuela. 2001.

[BESEMBEL 00] Besembel, Isabel. “Diseño de Software Orientado a Objetos”. Asignatura: Base de Datos. Facultad de Ingeniería. Universidad de los Andes. Mérida Venezuela 2000.

[MSDN] Microsoft. <http://msdn.microsoft.com>

[RAE] Real Academia Española. <http://www.rae.es>

www.bdigital.ula.ve

Anexos

Anexo A: Diccionario de Datos

CLASE BOBINAS DEFECTUOSAS

Atributos	Nombre del Atributo	Descripción	Tipo	Rango
	BobinalID	Identificador de la bobina defectuosa	Entero	
	DefectoImportancia	Importancia del defecto presentado por la bobina	Cadena	Longitud = 1
	Causa_sugerida	Causas Sugerida productora del defecto	Cadena	Máxima Longitud = 500
	Accion_correctiva	Acción Correctiva Sugerida para el defecto	Cadena	Máxima Longitud = 500
	Accion_preventiva	Acción Preventiva sugerida para evitar el defecto.	Cadena	Máxima Longitud = 500
	Comentario_planta	Comentario cargado en planta	Cadena	Máxima Longitud = 500
	Aprobación_JefeLinea	Firma Jefe de Línea	Cadena	Longitud = 1
	Aprobación_Inspector	Firma Inspector	Cadena	Longitud = 1
	Aprobación_Supervisor	Firma Supervisor	Cadena	Longitud = 1
	Df_ind_ubicacion	Indicador de ubicación del defecto	Cadena	Longitud = 1
	Df_ind_intensidad	Indicador de intensidad del defecto	Cadena	Longitud = 1
	Df_ubicacion_cara	Indicador de ubicación de lado de la cara de la bobina	Cadena	Longitud = 1
	Df_ubicacion_ancho	Indicador de ancho del defecto	Cadena	Longitud = 1
	Df_largo_real	Indicador de largo del	Flotante	

		defecto		
	Df_espesor_centro	Espesor desde el centro del defecto	Flotante	
	Df_espesor_lado_servicio	Espesor del lado del servicio de la bobina para el defecto	Flotante	
	Df_espesor_lado_motor	Espesor del lado del motor de la bobina para el defecto	Flotante	Longitud = 1
	Df_ancho_real	Ancho verdadero del defecto	Flotante	
	Origen	Origen del defecto (Propio, asignado o derivado)	Cadena	Longitud = 1
Operaciones	Obtener_datos_bobinas_defectuosas	Obtiene los datos relacionados con el defecto de una bobina.		

Tabla A.1 Descripción Clase Bobinas Defectuosas.

CLASE DATOS BOBINAS

www.biblioteca.ula.ve				
Atributos	Nombre del Atributo	Descripción	Tipo	Rango
	Número	Número de la Bobina	cadena	Longitud = 10
	Fecha_produccion	Fecha de Producción de la bobina	Fecha	
	Turno	Turno de producción de la bobina	Cadena	Longitud = 1
	Cuadrilla	Cuadrilla que produjo la bobina	Cadena	Longitud = 1
	Peso	Peso de la bobina.	Flotante	
	Ancho	Ancho de la bobina	Flotante	
	Espesor	Espesor de la bobina	Flotante	
	Largo	Largo de la Bobina	Flotante	
	Peso_planchon	Peso del Planchón de donde fue producida la bobina	Flotante	
	Longitud_pnachon	Longitud del Planchón de donde fue producida la bobina	Flotante	
	Numero_colada	Número de la colada en la	Cadena	Máxima

		que se produjo el planchón		Longitud = 10
	Cup	Indicador de calidad de la bobina	Cadena	Máxima Longitud = 10
	Cliente	Cliente a quien esta destinada la bobina	Cadena	Máxima Longitud = 20
	Tace	Tipo de Acero de la bobina	Entero	
Operaciones	Obtener_datos_bobina	Obtiene los datos relacionados con una bobina.		

Tabla A.2 Descripción Clase Datos Bobina.

CLASE LINEA ORIGEN

		Nombre del Atributo	Descripción	Tipo	Rango
Atributos		LineaID	Indicador de la línea origen del defecto	Cadena	Longitud = 5
		Descripción	Descripción de la Línea Origen del defecto	Cadena	Máxima Longitud = 50
Operaciones		Obtener_linea	Obtiene una línea en específico		

Tabla A.3 Descripción Clase Línea Origen.

CLASE DEFECTO

		Nombre del Atributo	Descripción	Tipo	Rango
Atributos		defectoID	Indicador del código del defecto	Cadena	Longitud = 5
		Descripción	Descripción del defecto	Cadena	Máxima Longitud = 50
Operaciones		Obtener_defecto	Obtiene un defecto en específico		

Tabla A.4 Descripción Clase Defecto.

CLASE ESTADO CALIDAD

		Nombre del Atributo	Descripción	Tipo	Rango
Atributos		ecID	Indicador del código del estado de calidad	Cadena	Longitud = 2
		Descripción	Descripción del estado de calidad	Cadena	Máxima Longitud = 50
Operaciones		Obtener_ec	Obtiene un estado de calidad en específico		

Tabla A.5 Descripción Clase Estado de Calidad

CLASE APLICACION

		Nombre del Atributo	Descripción	Tipo	Rango
Atributos		aplicacionID	Indicador del código de la aplicación	Cadena	Longitud = 1
		Descripción	Descripción del estado de calidad	Cadena	Máxima Longitud = 50
Operaciones		Obtener_aplicacion	Obtiene una aplicación en específico		

Tabla A.6 Descripción Clase Aplicación.

CLASE RESPONSABLE BOBINAS DEFECTUOSAS

		Nombre del Atributo	Descripción	Tipo	Rango
Atributos		rdbID	Indicador	Entero	
		Ind_estado	Indicador de si la bobina tiene causa o no asignada	Cadena	Longitud = 1
		Fecha_asignacion	Fecha en la que le fue asignada la bobina al responsable	Fecha	

Operaciones	Obtener_responsable_bobina	Obtiene un responsable para una bobina en específico		
	Asignar_responsable_a_bobina	Asigna un responsable a una bobina.		
	Eliminar_responsable_a_bobina	Elimina el Responsable de la bobina		

Tabla A.7 Descripción Clase Responsable Bobina Defectuosa.

CLASE RESPONSABLE

	Nombre del Atributo	Descripción	Tipo	Rango
Atributos	responsableID	Login de Dominio dentro de SIDOR	Cadena	Longitud = 10
	Nombre	Nombre del Responsable	Cadena	Máxima Longitud = 30
Operaciones	Obtener_responsable	Obtiene un responsable en específico		

Tabla A.8 Descripción Clase Responsable.

CLASE PERFILES

	Nombre del Atributo	Descripción	Tipo	Rango
Atributos	perfilID	Rol del responsable dentro de sidor	Cadena	Longitud = 10
	Descripción	Descripción del Perfil	Cadena	Máxima Longitud = 50

Operaciones	Obtener_responsable	Obtiene un responsable en específico		

Tabla A.9 Descripción Clase Perfiles.

CLASE CAUSA

	Nombre del Atributo	Descripción	Tipo	Rango
Atributos	CausaID	Indicador de la causa	Entero	
	Linea	Línea Origen asociada a la causa	Cadena	Longitud = 5
	Defecto	Defecto Asociado a la Causa	Cadena	Longitud = 3
	Descripción	Descripción de la Causa	Cadena	Máxima Longitud = 500
Operaciones	Obtener_causas	Obtiene un listado de causas asignadas por línea y defecto		
	Asignar_causas	Asigna una causas al defecto de una bobina		
	Obtener_cantidad_de_causas	Obtiene las veces que una causa fue asignada		

Tabla A.10 Descripción Clase Causa.

CLASE ACCIONES

	Nombre del Atributo	Descripción	Tipo	Rango
Atributos	accionID	Indicador de la acción	Entero	
	linea	Línea asociada a la acción	Cadena	Longitud = 5
	Defecto	Defecto asociado a la acción	Cadena	Longitud = 3
	nombre	Nombre de la acción	Cadena	Máxima Longitud = 30
	descripcion	Descripción de la acción.	Cadena	Máxima Longitud = 500
	Procedimiento	Procedimiento a cumplir	Cadena	Máxima

		para ejecutar la acción		Longitud = 500
	Estado	Estado de la acción	Cadena	Longitud = 1
	Fecha_inicio	Fecha de inicio de la acción	Fecha	
	Fecha_culminacion	Fecha de Culminación de la acción	Fecha	
	Responsable_ejecutor	Persona a ejecutar la acción	Cadena	Máxima Longitud = 10
	significativa	Marca de si la acción es significativa o no	Binario	
Operaciones	Obtener_accion	Obtiene todas las acciones para una causa en específico.		
	Asignar_accion	Asigna una acción a una causa		
	Marcar_significativa	Registra si una acción es significativa o no.		
	Obtener_acciones_asignadas	Obtiene todas las acciones asignadas a un responsable ejecutor		
	Obtener_cantidad_de_acciones_por_causa	Obtiene las veces que varias acciones fueron aplicadas a una misma causa.		

Tabla A.11 Descripción Clase Acciones.

Anexo B: Diccionario de Términos

B.1 Aplicación Web: El Word Wide Web Consortium (W3C) define el término “Web” como: *“el universo de información accesible a través de la red”*. Una aplicación Web es un sistema que permite a un usuario final acceder a una parcela de información contenida en el universo al que hace referencia la anterior definición del W3C. Las aplicaciones Web son aplicaciones basadas en el muy extendido paradigma “cliente/servidor”. Este paradigma consiste en un servidor que sabe cómo proporcionar un servicio y un cliente que desea acceder al servicio.

B.2 Base de Datos Relacional: es una base de datos que contiene múltiples tablas que comparten datos.

B.3 Consulta de Base de Datos: es la operación mediante la cual se extrae un juego de registros de una base de datos. Una consulta consta de criterios de búsqueda expresados en un lenguaje de base de datos denominado SQL.

B.4 Controlador de base de datos: es un software que actúa como intérprete entre una aplicación Web y una base de datos. Los datos de una base de datos se almacenan en un formato propio de dicha base de datos. Un controlador de base de datos permite a la aplicación Web leer y manipular datos que, de otro modo, resultarían indescifrables.

B.5 IIS: Internet Information Server es un servidor Web, que incluye los servicios de: HTTP, HTTPS, FTP, SMTP (correo saliente) y NNTP (grupos de noticias). Además es capaz de ejecutar varios motores de script como: ASP, ASP.NET PHP, Cold Fusion, etc.

B.6 JOB: Son tareas programadas en el SQL Server, con la finalidad de ejecutar acciones específicas sobre las bases de datos de forma automática, en instancias específicas del tiempo.

B.7 Juego de registros: es un conjunto de datos extraídos de una o varias tablas de una base de datos.

B.8 Página dinámica: es una página Web personalizada por el servidor de aplicaciones antes de que la página se envíe a un navegador

B.9 Página estática: es una página Web que el servidor de aplicaciones no modifica antes de enviarla a un navegador.

B.10 Servidor de aplicaciones: es un software que ayuda al servidor Web a procesar las páginas que contienen scripts o etiquetas del lado del servidor. Cuando se solicita al servidor una página de este tipo, el servidor Web pasa la página al servidor de aplicaciones para su procesamiento antes de enviarla al navegador. Entre los servidores de aplicaciones más habituales se encuentran Microsoft .NET Framework, IBM WebSphere y Apache Tomcat.

B.11 Servidor Web: es un software que suministra páginas Web en respuesta a las peticiones de los navegadores Web. La petición de una página se genera cuando un visitante hace clic en un vínculo de una página Web en el navegador, elige un marcador en el navegador o introduce un URL en el cuadro de texto Dirección del navegador. Entre los servidores Web más utilizados se encuentran Microsoft Internet Information Server (IIS), Apache HTTP Server, Netscape Enterprise Server y Sun ONE Web Server.

B.12 Sistema de Administración de Base de Datos (DBMS o sistema de base de datos): es un software que se utiliza para crear y manipular bases de datos. Entre los sistemas de bases de datos más habituales figuran Microsoft SQL SERVER, Oracle y MySQL.

B.13 SOAP: SOAP es un protocolo basado en XML, diseñado específicamente para transportar llamadas a procedimiento utilizando XML.

www.bdigital.ula.ve

Anexo C: Pantallas del Sistema.

- Pantalla Principal del sistema:



Figura C.1 Principal.

- Pantalla Revisión de Defectos:

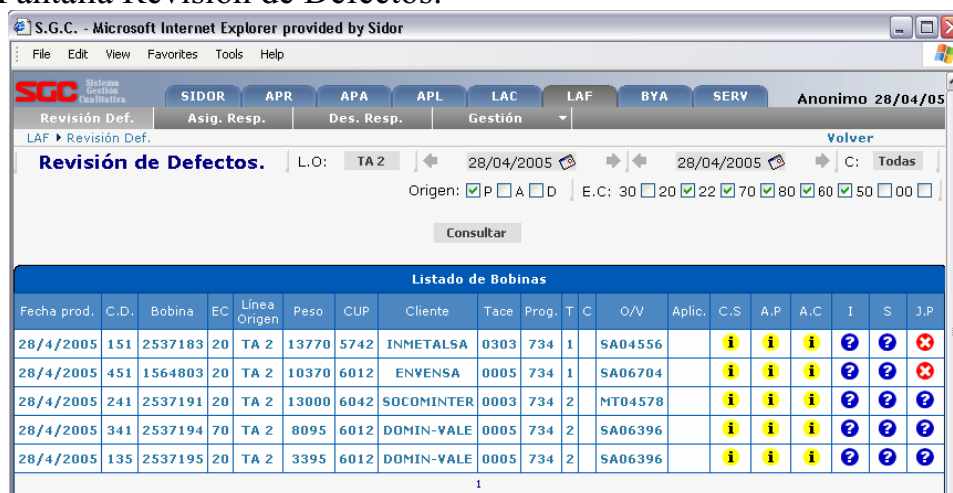


Figura C.2 Revisión de defectos.

- Pantalla Firmar bobinas:

S.G.C. - Microsoft Internet Explorer provided by Sidor

File Edit View Favorites Tools Help

SGC Sistema Gestión Calidad

SIDOR APR APA APL LAC LAF BYA SERV Anonimo 28/04/05

Revisión Def. Asig. Resp. Des. Resp. Gestión Volver

LAF ▶ Revisión Def.

Datos de la Bobina

Numero de bobina: 2537183

Linea Act./Linea Ret.	TA 2	TA 2	Esp. Ord.	0.22	Aplicación Línea		Tolerancias
Defecto	151-BORDES ROTOS (INIC Y FINAL)	Anc. Ord.	842.60	Aplicación		Esp+	0.01 Anch 1
Peso	13770	Anc. Real	857	Linea Origen	TA 2	Esp-	0.01 Pmin 1900
Fecha/Prod.	28/4/2005 05:49:09 AM	Material/reproceso		Estado de Calidad Línea	20	Largo +	0 Pmax 2050
Cup	5742	Turno	1	Estado de Calidad	20	Defectos	
Cliente	INMETALSA	Linea Previa	DC 2	Linea Destino	RCC	UB.	C IN. F
Tace	0303	Cuadrilla		O/Venta	SA04556 - 01	LA.	O CA. N
Bobina Madre Frio	1564925	Bobina Madre LCC		Uso final O/V	TAPAS CORONA PRY OFF (CERVEZA).		
Programa	734 -0110	Colada	051615	Ruta O/V	DC 2-TA 2-RCC-TM 2-PB 1-EC 1-CH 1		
F/Retención.	28/4/2005	F/Aplicación		F/Derivación			
Comentarios a PCP	Nuevos O/VI renglón bobina						
Firma SI/No	No						

Causas/ Acciones/ Bobina

Causa Sugerida	Acción Preventiva
Acción Correctiva	Comentarios

Guardar Atrás

Figura C.3 Firmar bobinas.

- Pantalla Asignación de Responsables:

S.G.C. - Microsoft Internet Explorer provided by Sidor

File Edit View Favorites Tools Help

SGC Sistema Gestión Calidad

SIDOR APR APA APL LAC LAF BYA SERV Anonimo 28/04/05

Revisión Def. Asig. Resp. Des. Resp. Gestión Volver

LAF ▶ Asig. Resp.

Asig. de Resp.

Linea: Todas 28/04/2005 28/04/2005 Responsable: Todos

Aplicación: lpo ☒ sov ☐ lpoo ☐ 2da ☐ ch ☐ en ☐ db ☐ rp ☐ E.C: 30 ☐ 20 ☒ 22 ☒ 70 ☒ 80 ☒ 50 ☐

Consultar

Listado de Bobinas

Responsable	Fecha Prod.	C.D.	Bobina	EC	Línea	Peso	CUP	Tace	Prog.	T	C	O/V	C.S	A.P	A.C	I	S	J.P
	28/4/2005	170	1563956	22	TM 2	12136	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	1563956	22	TM 2	12136	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	170	1563960	22	TM 2	13236	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	1563960	22	TM 2	13236	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	170	2536963	22	TM 2	12972	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	2536963	22	TM 2	12972	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	170	1563955	22	TM 2	11550	6042	0004	469	2		SA05134	i	i	i	?	?	?
	28/4/2005	494	1563955	22	TM 2	11550	6042	0004	469	2		SA05134	i	i	i	?	?	?

1 2 3

Guardar

Figura C.4 Asignar Responsables.

- Pantalla Eliminación de Responsables:

Des/Responsables. L.O: Todas 28/04/2005 28/04/2005 Resp: Todos

Listado de Bobinas

Responsable	Fecha prod.	Fecha Asig.	C.D.	Bobina	EC	Línea	Peso	CUP	Tace	Prog	T	C	O/V	C.S	A.P	A.C
SIERRALTA Angel	28/4/2005	28/4/2005	494	1563964	20	TM 2	13144	6042	0004	469	2		SA05134	i	i	i
SIERRALTA Angel	28/4/2005	28/4/2005	494	1563947	20	TM 2	13306	6042	0004	469	1		SA05134	i	i	i
SIERRALTA Angel	28/4/2005	28/4/2005	341	2537194	70	TA 2	8095	6012	0005	734	2		SA06396	i	i	i
SIERRALTA Angel	28/4/2005	28/4/2005	241	2537191	20	TA 2	13000	6042	0003	734	2		MT04578	i	i	i
SIERRALTA Angel	28/4/2005	28/4/2005	494	1563954	20	TM 2	13070	6042	0004	469	1		SA05134	i	i	i
SIERRALTA Angel	28/4/2005	28/4/2005	494	1563959	20	TM 2	13080	6042	0004	469	1		SA05134	i	i	i
SIERRALTA Angel	28/4/2005	28/4/2005	135	2537195	20	TA 2	3395	6012	0005	734	2		SA06396	i	i	i
SIERRALTA Angel	28/4/2005	28/4/2005	494	1563959	20	TM 2	13080	6042	0004	469	1		SA05134	i	i	i
SIERRALTA Angel	28/4/2005	28/4/2005	494	1563954	20	TM 2	13070	6042	0004	469	1		SA05134	i	i	i
SIERRALTA Angel	28/4/2005	28/4/2005	135	2537195	20	TA 2	3395	6012	0005	734	2		SA06396	i	i	i

1

* Las bobinas con los combos deshabilitados, significan que el responsable de esa bobina ya le asignó una causa.

Guardar

Figura C.5 Eliminación de responsables.

- Pantalla Cantidad de Bobinas Asignadas:

Cantidad de Bobinas Asignadas por Defectos L.O: APL

Listado de Bobinas y Defectos

Línea	ID Defecto	Descripción del defecto	Nro. de Bobinas
APL	705	FALTA CERTIF. RECUBRIMIENTO	1
APL	713	IMPRESIONES LCH	1

1 2

NOTA: Seleccione el defecto para asignarle las causas

Figura C.6 Cantidad de Bobinas Asignadas.

- Pantalla Detalle Bobinas Asignadas:

Asignación Causas / Defecto L.O: APL Defecto: FALTA CERTIF. RECUBRIMIENTO-705

Listado de Bobinas

Linea Origen	Bobina	Fecha Prod.	C.D	EC	Peso	CUP	Tace	Prog.	T	C	C.S	A.P	A.C	I	S	J.P
APL	T251582	8/3/2005	705	20	2130	5742	0303	250 -0020	3	C	i	i	i	?	?	?

Causas Anteriormente Aplicadas

NOTAS:
Al asignar la(s) causa(s), podrá asignar la acción inmediatamente. Seleccione la(s) bobina(s) y la(s) causa(s) a asignar.

Acciones: Nueva Causa, Guardar, Siguiente

Figura C.7 Bobinas Asignadas.

- Pantalla Asignar Causas:

Nueva Causa

Defecto Seleccionado: FALTA CERTIF. RECUBRIMIENTO-705

Titulo: Causa de Ejemplo

Ingresar la Nueva Causa: Esta es una Causa de Ejemplo

Acciones: Guardar, Cerrar

Figura C.8 Asignar Causas.

- Pantalla Acciones Asignadas a Causas:

Acciones por Causas / Defecto L.O: APL Defecto: FALTA CERTIF. RECUBRIMIENTO-705

Causa: Causa de Ejemplo

Descripción de la Causa: Esta es una Causa de Ejemplo

Acciones: Guardar Significativas

Figura C.9 Acciones asignadas a causas

- Pantalla Asignar Acciones:

S.G.C. - Microsoft Internet Explorer provided by Sidor

File Edit View Favorites Tools Help

SGC Sistema Gestión Cualitativa

SIDOR APR APA APL LAC LAF BYA SERV Anonimo 29/04/05

Revisión Def. Asig. Resp. Des. Resp. Gestión Volver

LAC Gestión

Nueva Acción

Fecha Inicio: 29/04/2005

Fecha de Culminación: 29/04/2005

Defecto Seleccionado: FALTA CERTIF. RECUBRIMIENTO-705

Causa Seleccionada: Causa de Ejemplo

Nombre de la acción: Acción de Prueba

Descripción de la Acción: Descripción de la Acción de Prueba

Procedimiento: Procedimiento de la acción de prueba

Estado: En Curso

Responsable: SIERRAALTA Angel

Guardar

Figura C.10 Asignar Acciones

- Pantalla Gestión de Procesistas:

S.G.C. - Microsoft Internet Explorer provided by Sidor

File Edit View Favorites Tools Help

SGC Sistema Gestión Cualitativa

SIDOR APR APA APL LAC LAF BYA SERV Anonimo 29/04/05

Revisión Def. Asig. Resp. Des. Resp. Gestión Volver

LAC Gestión Gestión/Resp.

Gestión Procesistas L.O: Todas 25/04/2005 29/04/2005 Resp: SIERRAALTA Angel Todas

Consultar

Responsable: SIERRAALTA Angel G: 0%

Responsable con porcentaje de gestión < 80

Responsable con porcentaje de gestión >= 80

Figura C.11 Gestión de Procesistas

- Pantalla Gestión de Defectos:

S.G.C. - Microsoft Internet Explorer provided by Sidor

File Edit View Favorites Tools Help

SGC Sistema Gestión Cualitativa

SIDOR APR APA APL LAC LAF BYA SERV Anonimo 29/04/05

Revisión Def. Asig. Resp. Des. Resp. Gestión Volver

LAC Gestión Gestión/Def.

Gestión/Def L.O: Todas F.F: F.D: Fecha Prod. 25/04/2005 29/04/2005 Def: Todos

Consultar

APL:	477-ZONAS DE PRESION (SUP)	
APL:	701-MAL ENROLLADO RECUPERABLE (COS)	
LCC:	035-DIFEREN. DE REND. METALICO (LDD)	
LCC:	100-ANCHO INFERIOR (DIM)	
LCC:	133-POCO PESO X BORDES AGRIETAD (LDD)	
LCC:	136-BORDES AGRIETADOS (3 MM) (COS)	
LCC:	-2-SIN DEFECTO	
LCC:	241-BORDES RIZADOS (FOR)	
LCC:	251-PORO DEL TIPO BANDING (SUP)	
LCC:	321-VARIACION DE ANCHO (DIM)	
LCC:	331-VARIACION DE ANCHO ESTIRAMI (DIM)	

Figura C.12 Gestión de defectos.

- Pantalla Gráfico de Causas:

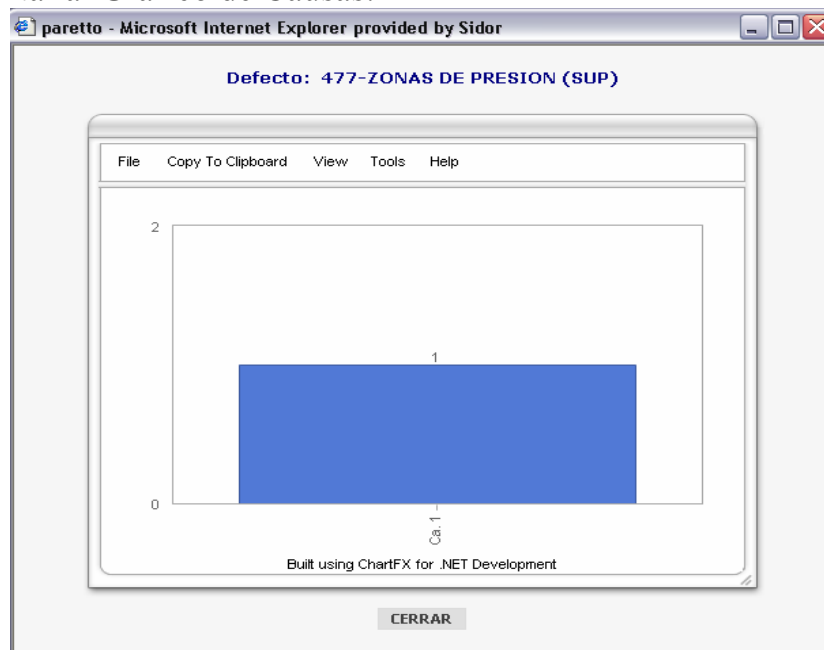


Figura C.13 Gráfico de Causas a Defectos.

- Pantalla Gráfico de Acciones a Causas:

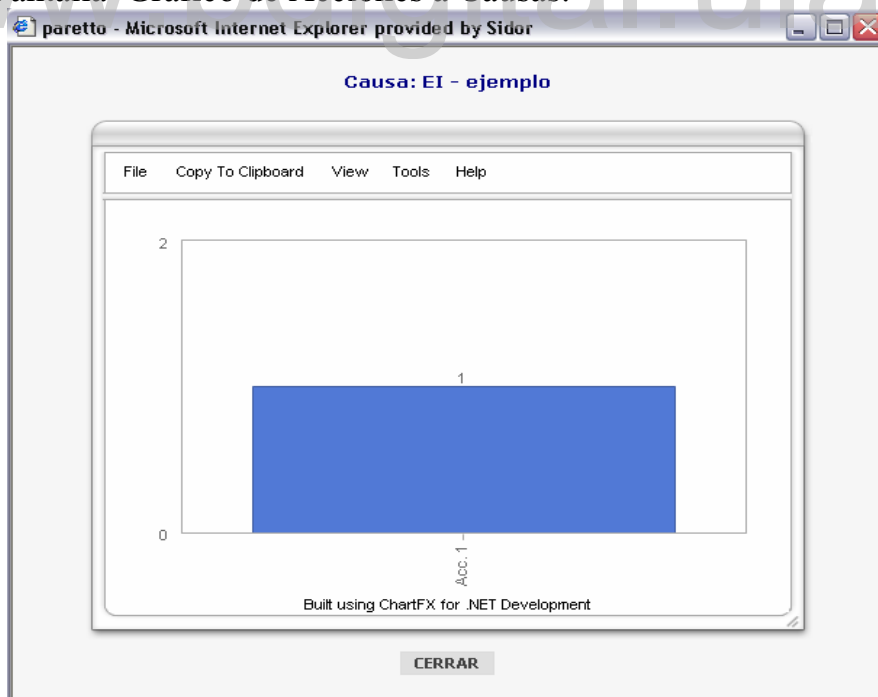


Figura C.14 Gráfico de Acciones a Causas.