



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS

**SISTEMA WEB PARA EL CONTROL DE
CALIDAD EN EL PROCESO DE
LAMINACIÓN DE SIDOR, Y LIBRO DE
NOVEDADES**

Elaborado por:

Br. Claudia Devora Rivas López

Tutor Académico:

Dr. Milagro Rivero

Tutor Industrial:

Ing. Carlos Agostini.

MÉRIDA, SEPTIEMBRE DE 2005

UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

www.bdigitalula.ve



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS

**SISTEMA WEB PARA EL CONTROL DE
CALIDAD EN EL PROCESO DE
LAMINACIÓN DE SIDOR, Y LIBRO DE
NOVEDADES**

www.bdigital.ula.ve

**TRABAJO FINAL DE GRADO PRESENTADO ANTE LA
ILUSTRE UNIVERSIDAD DE LOS ANDES COMO
REQUISITO FINAL PARA OPTAR AL TÍTULO DE
INGENIERO DE SISTEMAS**

MÉRIDA, SEPTIEMBRE DE 2005

UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

SISTEMA WEB PARA EL CONTROL DE CALIDAD EN EL PROCESO DE LAMINACIÓN DE SIDOR, Y LIBRO DE NOVEDADES

POR: Claudia Devora Rivas López

**Trabajo final de grado presentado como requisito obligatorio
para optar al título de Ingeniero De Sistemas.**

UNIVERSIDAD DE LOS ANDES

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA DE SISTEMAS

www.bdigital.ula.ve

APROBADO:

Dra. Milagro Rivero

CI: _____

Tutor

Dra. Flor Narciso

CI: _____

Jurado

Dr. Domingo Hernández

CI: _____

Jurado

**Universidad de los Andes
Facultad de Ingeniería
Escuela de Ingeniería de Sistemas**



Dedicatoria:

Madre has sido un apoyo incondicional durante mi vida, en todos los proyectos que me he propuesto me has infundido confianza, y voluntad para la realización de los mismos. Me has brindado tus virtudes, las cuales me han permitido superar los obstáculos que se me han presentado a lo largo del camino hacia el éxito. Gracias por ser el motor de mi vida, te amo madre mía.

Padre, has sido un ejemplo de perseverancia, quien me ha enseñado a salir airoso frente a las dificultades, y continuar en el camino que nos hemos trazado en la vida. Gracias padre mío por infundirme el deseo de superación, y de crecimiento personal. Sin tus consejos y tu apoyo no hubiera logrado esta meta.

A mi hermano, para quien debo ser un ejemplo y apoyo fundamental. Gracias por existir y hacer mi vida hermosa, teniendo con quien compartirla.

Agradecimientos:

Mis más sinceros agradecimientos para:

La Universidad de los Andes, por ser la casa de estudio que me brindo sus conocimientos durante el trayecto de mi carrera que me permitieron la realización de este trabajo.

La Escuela de Sistemas, que me brindo la formación académica y profesional que tengo en la actualidad.

A mi tutor industrial Carlos Agostini, por haberme brindado la oportunidad de la realización de este proyecto en la gerencia de automatización. Gracias por ser un apoyo importante en el desarrollo y realización de este proyecto.

A mi tutora Dr. Milagro Rivero quien ha sido un apoyo en el fortalecimiento de los conocimientos necesarios para la realización de este sistema.

A Hendryx Infante, quien ha sido un fiel compañero y me ha brindado su conocimiento y ayuda para la realización de este proyecto, has sido una parte importante de mi vida profesional y personal. Gracias por ser mi apoyo incondicional en todos los momentos que te necesite.

A todos aquellos compañeros y amigos que participaron de una u otra manera en mi desarrollo profesional que me han permitido llegar a este punto de mi carrera. Gracias por permanecer a mi lado en los buenos y malos momentos.



Resumen

Un sistema Web, es un conjunto de archivos electrónicos y paginas Web referentes a un tema en particular, los cuales interactúan para lograr un objetivo específico. Estos sistemas son ampliamente usados en las empresas públicas y privadas, tanto en la intranet de tales empresas como en el Internet. Para el proyecto presentado en este trabajo, se realizó un Sistema Web el cual estará accesible solo en la intranet de la empresa privada SIDOR.

El objetivo primordial de este sistema es poder realizar el control de calidad en el proceso de laminación de la empresa, así como mantener informado al personal de los acontecimientos ocurridos en las plantas de producción. A este respecto, es necesario aclarar el concepto de calidad que se manejará. En las diferentes líneas de laminación, la calidad de los productos se mide de acuerdo los defectos visuales que el producto pueda presentar, así que el sistema esta diseñado para que se realice un circuito de control sobre tales defectos visuales observados al momento de la producción.

La metodología de desarrollo que se utilizó fue el Proceso Unificado "Rational" (RUP), el cual comprende cuatro fases donde se fue recolectando la información y los requerimientos funcionales del software basándose en la realización de diagramas de casos de uso, de secuencia, de clases y el modelado físico de la base de datos. Se utilizo una arquitectura cliente – servidor, la aplicación se desarrollo en el lenguaje de programación Visual Basic .Net y manejador de base de datos Sql Server 2000. Para ello existirán dos servidores, uno en el cual se encontrará la aplicación Web, y el otro como servidor de base de datos.

La aplicación se realizó en dos módulos principales, el área de gestión de defectos, y el área de información de sucesos en planta o libro de novedades.

Palabras Claves: RUP, WEB, Visual Basic.NET, SQL SERVER.

Índice

Dedicatoria:	5
Agradecimientos:	6
Índice	7
Índice de figuras	9
Introducción	11
Antecedentes:	14
Descripción del problema:	16
Objetivos	20
Objetivo General:	20
Objetivos secundarios:	20
Metodología.	20
Fase de Inicio.	22
Fase de Elaboración.	22
Fase de construcción.	23
Fase de Transición.	23
Alcance:	23
Estructura del documento.	24
Capítulo II:	25
Marco Teórico:	25
Prácticas Claves del proceso de desarrollo de software	25
Desarrollo iterativo	25
Orientación al manejo del riesgo	26
Orientación al cliente	27
Desarrollo evolutivo	27
Proceso Unificado:	28
Etapas del proceso unificado:	30
1. Etapa de ingeniería	30
2. Etapa de producción	31
ASP.NET	31
Sistemas de información:	34
Clasificación de los sistemas:	35
Sistema de información informatizado:	36
Capítulo III	37
Fase de inicio:	37
Visión general:	37



Obtención del modelo de casos de uso:	39
Requisitos funcionales:	39
Actores y casos de uso:	41
Modelos de Casos de Uso:	43
Descripción de casos de uso	44
Especificaciones complementarias:	64
Glosario:	65
Capítulo IV	67
Fase de elaboración	67
Iteración I:	68
Iteración II	101
Iteración III	123
Capítulo V	139
Diseño e interfaz de usuario:	139
Diseño de reportes del sistema para el SGD	140
Diseño de páginas del LN.	141
Conexión entre capas y aplicación del sistema:	143
Diagrama de implementación	143
Pruebas Finales del Sistema.	144
Conclusiones	145
Bibliografía	146
Anexos	147
Corporación SIDOR	147
Soporte de plataforma de .NET	161
Framework:	161
Arquitectura de Framework:	162
Lenguaje SQL	165
UML (Unified Modeling Language)	166

Índice de figuras

Figura 1: Pirámide de Automatización	13
Figura 2: Pantalla de carga de datos	16
Figura 3: Gestión de Defectos	19
Figura 4: Gestión del Libro de Novedades	19
Figura 5: Fases del desarrollo	25
Figura 6: Perfiles de Riesgo	26
Figura 7: Diagrama de sistema.	35
Figura 8: Sistema de gestión	35
Figura 9: Sistema de información	35
Figura 10: Diagrama de casos de uso SGD	43
Figura 11: Diagrama de casos de uso LN	44
Figura 12: UCS Gestión de indicadores	52
Figura 13: UCS Gestión de novedades generales	58
Figura 14: DSS Revisión de defectos	68
Figura 15: DSS Visualización de defectos asignados.	69
Figura 16: DSS Evolución de defectos en el tiempo	69
Figura 17: DSS Evolución de defectos ante acciones correctivas asignadas.	70
Figura 18: DSS Verificación de carga de causas, acciones por bobina defectuosa.	71
Figura 19: DSS Gestión de indicadores de planta	72
Figura 20: DSS Visualizar productividad efectiva	72
Figura 21: DSS Visualizar efectividad	73
Figura 22: DSS Visualizar puesta a mil	74
Figura 23: DSS Gestión de material defectuoso.	75
Figura 24: DSS Gestión de equipos defectuosos	75
Figura 25: DSS Gestión de novedades generales.	76
Figura 26: DSS Generación de Reportes	77
Figura 27: DSS Gestión de novedades jerárquicas	78
Figura 28: Modelo de dominio de la iteración 1 para el SGD	79
Figura 29: Modelo de Dominio, iteración 1 para el LN.	80
Figura 30: Modelo de Dominio, iteración 1 para el SGD	80
Figura 31: Modelo de Dominio para el LN	81
Figura 32: Clases Conceptuales del SGD	81
Figura 33: Clases Conceptuales del LN.	82
Figura 34: DI Revisión de toneladas defectuosas.	84
Figura 35: DI Defectos Asignados	85
Figura 36: DI Verificación de causas, acciones, por bobina defectuosa	86
Figura 37: DI Escenario Visualizar productividad efectiva	86
Figura 38: DI Escenario Visualizar efectividad	87
Figura 39: DI Escenario Visualizar puesta a mil	88
Figura 40: DI Escenario Visualizar calidad	88
Figura 41: DI Escenario visualizar equipos defectuosos	89
Figura 42: DI Escenario Visualización de novedades	90
Figura 43: DI Escenario Crear/Modificar Novedad General	90
Figura 44: DI Escenario Crear/Modificar Novedad Jerárquica	91
Figura 45: DCD del sistema de gestión de defectos	93
Figura 46: DCD del Libro de novedades	94
Figura 47: Navegación de las pantallas LN	96
Figura 48: Menú XML para navegación del sistema	97
Figura 49: Navegación en las pantallas SGD	97
Figura 50: Menú XML para navegación del sistema	98
Figura 51: Visualización del Menú Estándar de Automatización para el LN	98
Figura 52: Visualización del Menú Estándar de Automatización para el SGD	98

Figura 53: Diagrama de casos de uso SGD Iteración 2	103
Figura 54: Diagrama de casos de uso LN Iteración 2	104
Figura 55: UCS Gestión de comentarios	108
Figura 56: DSS Visualización de defectos	111
Figura 57: DSS Verificar gestión de bobinas defectuosas	111
Figura 58: Asignar responsables	112
Figura 59: DSS Gestión de comentarios	113
Figura 60: Modelo de domino del SGD para la iteración II	115
Figura 61: Modelo de domino del LN para la iteración II	116
Figura 62: DI Escenario Visualización de defectos:	118
Figura 63: DI Escenario Verificar gestión de bobinas defectuosas	118
Figura 64: DI escenario asignación de responsables	119
Figura 65: DI escenario crear o modificar comentarios	119
Figura 66: DCD para el SGD, iteración II	120
Figura 67: DCD para el LN, iteración II	121
Figura 68: Navegación de las pantallas Iteración II	122
Figura 69: Menú XML para navegación del sistema	122
Figura 70: Modelo de casos de uso para el SGD, iteración III	124
Figura 71: DSS Visualizar bobinas afectadas por comité para PCP	128
Figura 72: DSS Visualizar bobinas para SICOP	128
Figura 73: DSS Datos bobinas	129
Figura 74: Modelo de dominio SGD para la iteración III	130
Figura 75: Modelo de dominio LN para la iteración III	131
Figura 76: DI Visualizar bobinas afectadas por comité para PCP	132
Figura 77: DI Visualizar bobinas para SICOP	133
Figura 78: DI Datos Bobinas	133
Figura 79: DCD del SGD para la iteración III	134
Figura 80: DCD para el LN, iteración III	135
Figura 81: Navegación de las pantallas SGC Iteración III	136
Figura 82: Menú XML para navegación del sistema SGD.	136
Figura 83: Navegación de las pantallas LN Iteración III	137
Figura 84: Menú XML para navegación del sistema LN.	138
Figura 85: Diseño de reportes del SGD.	141
Figura 86: Diseño de páginas de datos del LN	141
Figura 87: Crear novedad	142
Figura 88: Modificar novedad	142
Figura 89: Configuración de capas con la aplicación del sistema	143
Figura 90: Diagrama de implementación.	144
Figura 91 : Mapa de la ubicación geográfica de la planta	148
Figura 92: Muelle SIDOR	148
Figura 93: Plano general o layout de SIDOR	149
Figura 94: Organización SIDOR	150
Figura 95: Dirección Industrial	151
Figura 96: Laminador en Caliente	153
Figura 97: Bobina Terminada	153
Figura 98: Enrolladores	155
Figura 99: Material Decapado	155
Figura 100: Limpieza electrolítica	160
Figura 101: Recocido	160
Figura 102: Arquitectura de .Net Framework	162
Figura 103: Biblioteca de Clases .Net	164
Figura 104: Historia del UML.	167

Introducción

El desarrollo de sistemas de información con tecnología de punta forma parte vital de toda organización. La necesidad de actualizarse hoy en día es imprescindible para empresas que compiten en mercados agresivos, ya que cada vez las mismas se encuentran ante niveles de exigencia mayores de parte del cliente a quien se ofrecen servicios. Aunado a ello se requiere que la industria se consolide en el mercado al cual le toca competir. Como la prioridad de una empresa es su negocio, la tecnología va asociado a ello ya que esta le permitirá alcanzar sus objetivos de una forma más eficiente y confiable, entre los cuales se lista la rentabilidad, productividad y eficiencia de la empresa.

Una prioridad que la empresa debe cubrir implica mantener la calidad en cuanto a la producción, a este respecto un sistema de información robusto y bien diseñado puede ofrecerse como una herramienta clave en la gestión empresarial. Como este campo es preciso cubrir, el presente proyecto trata de un sistema de calidad para la gestión empresarial de SIDOR (Siderúrgica del Orinoco). Comenzaremos a adentrarnos en el mismo, haciendo una descripción de la empresa y el proceso productivo que es necesario mantener de acuerdo a estándares de calidad establecidos por la corporación.

Gestión Empresarial

Para consolidarse en el mercado como una empresa productora de acero de primer nivel, define una estrategia con visión de largo plazo y crecimiento, utilizando tres vectores principales de desarrollo: la inversión en tecnologías, equipos y expansión; el desarrollo de recursos humanos; y la integración con la cadena venezolana del hierro y el acero.

La visión de SIDOR, comprende tener estándares de competitividad similares a los productores de acero más eficientes y con la finalidad de estar ubicada entre las mejores siderúrgicas del mundo.

La empresa tiene el compromiso de satisfacer las necesidades de sus clientes y mantener estándares mundiales de calidad en sus productos bajo la norma ISO 9001, que aseguran su competitividad en los mercados internacionales.

Hoy, la industria privatizada tiene la obligación de mantener y superar los estándares de calidad y eficiencia productiva que exige la industria.

Para cumplir con estos objetivos anteriormente expuestos, SIDOR ha puesto en práctica un sistema de calidad que le permite cumplir con las exigencias establecidas y ocupar una posición destacada en el área donde realiza sus operaciones.

El sistema se basa en la participación de todo el personal y en la búsqueda de la excelencia en los procesos y productos. Esta dedicación se traduce en un esfuerzo continuo que asegura la entrega al mercado de productos siderúrgicos certificados.

A este respecto juega un papel fundamental el área de automatización de la empresa, la cual tiene la responsabilidad de mantener estándares de calidad y optimización de los procesos, así como coordinar el flujo de información de los mismos. En esta área se mantiene un esquema de trabajo piramidal, el cual se distribuye de la siguiente manera:

Nivel 0, Elementos de Campo: 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.

Nivel 1, Red de Control de Campo: 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.

Nivel 2a, Red de Supervisión y Control: 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.

Nivel 2b, Explotación de Datos: 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, y planificación a nivel gerencial.

Nivel 3, Papel Electrónico: es el encargado 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.

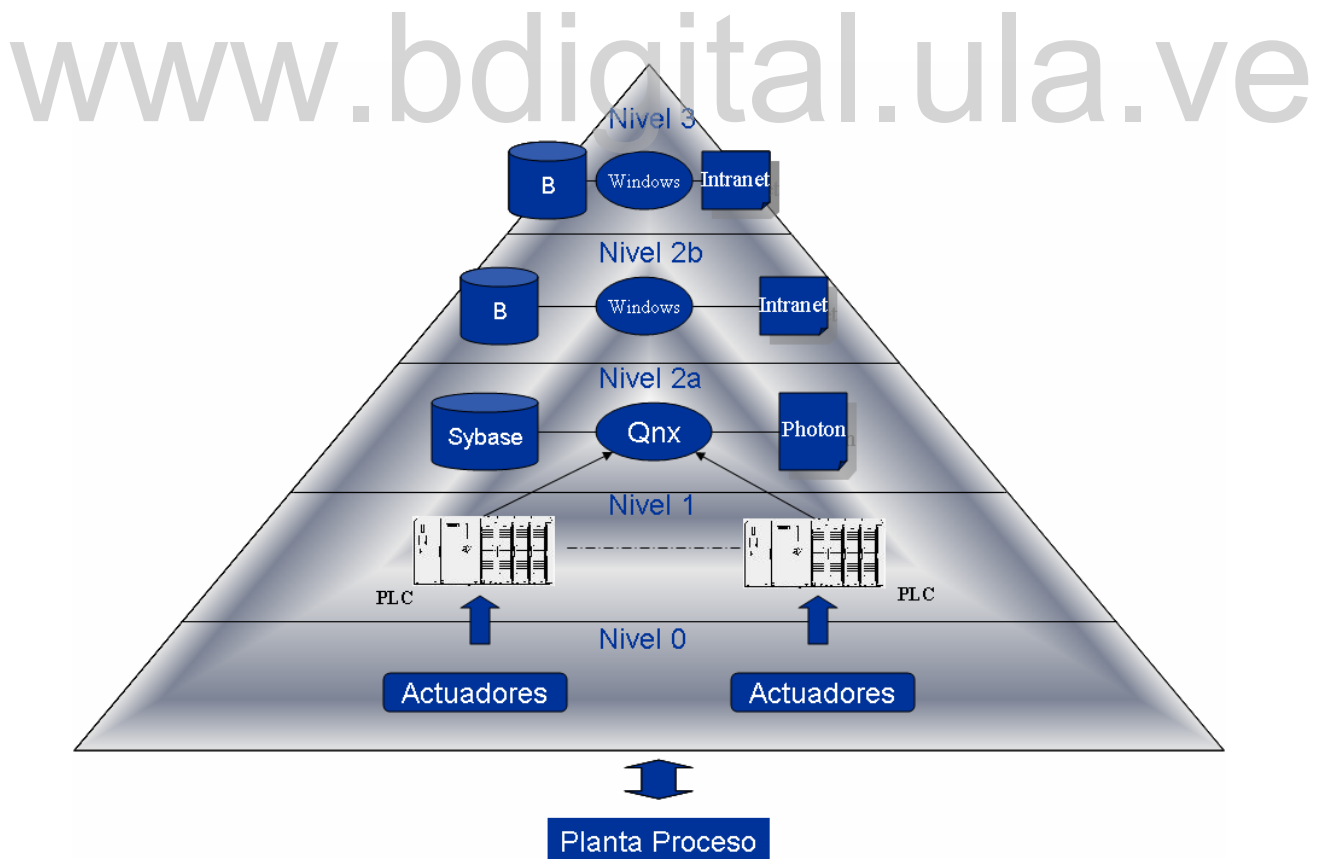


Figura 1: Pirámide de Automatización

Uno de los campos que contribuye al buen funcionamiento de esta gestión de calidad, lo constituye el área de Nivel 2B, quien deberá permitir que la información adecuada se encuentre de forma accesible en la intranet de la empresa, con la finalidad de agilizar el proceso de gestión de calidad en la misma. A continuación describimos la situación actual de esta área.

Antecedentes:

Con la idea de automatizar todas las áreas y líneas de producción bajo un mismo estándar, nace el proyecto piso de planta, que originalmente estaba más enfocado a realizar las actividades propias de la automatización de la planta, entre las cuales podemos contar:

- Adquirir y mostrar programa de producción y datos del producto a procesar.
- Mostrar mímicos y pantallas de operación.
- Correr modelos para setup y control.
- Mostrar y almacenar alarmas, eventos y tendencias.
- Adquirir y almacenar datos relativos al producto y proceso.
- Enviar la información de producción al N2B.
- Adquirir y almacenar datos manuales y semiautomáticos.
- Sistema de justificación de demoras e interrupciones.
- Sistema de información de novedades y desvíos de proceso.
- Mostrar Prácticas Operativas y Especificaciones de Elaboración Adquirir y enviar la información de Retenciones de Calidad.
- Adquirir y enviar la información de manejo de materiales.
- CIPP Local: Implementar modelos de tracking tecnológico.

Actualmente, cada uno de estos sistemas, junto con su Nivel 1, constituye una “isla de automatización” estas islas funcionan en forma autónoma, para lograr tal funcionamiento de manera adecuada (modelos, programa de producción, etc.) es necesario que dicha “isla” este conectada al resto de la planta. La base de datos sybase sirve como repositorio temporal de datos para

mantener la operatividad en caso de desconexiones temporales de la red global. Como referencia, el tiempo que una línea debería poder operar sin conexión a la red global es de 4 horas.

La plataforma de estos sistemas es actualmente QNX 4 con Photon como ambiente grafico, y Sybase SQL Anywhere como base de datos.

No en todas las líneas existe un Nivel 2 A con adquisición y control. En estos casos existen servicios mínimos que ofrece el proyecto piso de planta, entre los cuales están:

- Ingreso / consolidación de datos manuales de proceso o producción.
- Detección y justificación de demoras / interrupciones
- Visualización de prácticas
- Transferencia de datos hacia N2B y Ciclo Activo
- Bitácoras o Libros de novedades

Disponiendo así de tal información proveniente del nivel 2 A, se esta desarrollando modularmente un sistema consolidado de información disponible a través de la intranet.

Contiene en un solo ambiente los datos e indicadores de un conjunto de líneas que definen una planta.

Dicho sistema es denominado con las siglas SGL, las cuales significan, Sistema de Gestión en Línea. Podemos denominar al SGL como el antecedente fundamental ya que al iniciarse esta ideología de desarrollo, se ha extendido hacia todos los proyectos que se están realizando en la gerencia de automatización.

Descripción del problema:

En el proceso productivo de las bobinas diariamente ocurren desperfectos en la fabricación de las mismas, por lo que es necesario llevar un control con la finalidad de disminuir la ocurrencia de tales fallas en la producción, y por ende disminuir los costos que se agregan al fabricar un producto defectuoso.

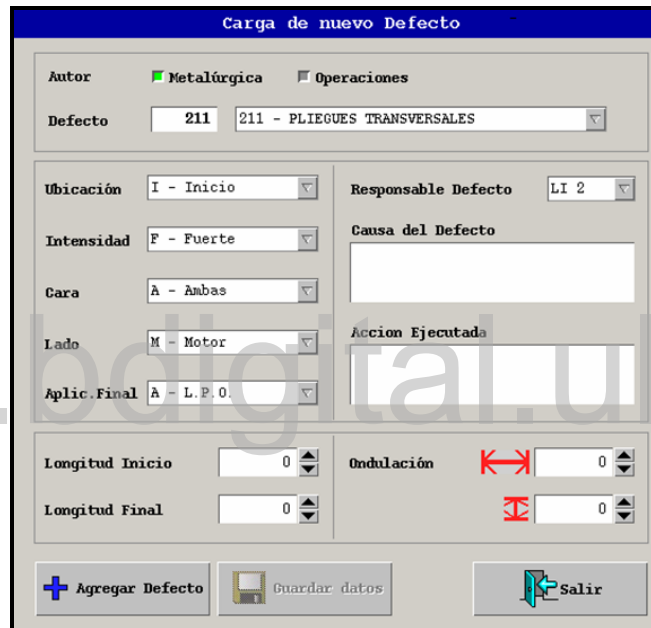


Figura 2: Pantalla de carga de datos

Hoy en día el operador carga desde planta los defectos que se observan en las líneas por cada bobina que es producida, podemos observar tal pantalla a continuación:

Esta información es guardada en SICOP, y posteriormente procesada por el personal autorizado para dictaminar hacia donde se debe direccionar la bobina producida. Esta información cargada por los operadores se mantiene actualmente en el nivel 2A bajo QNX.

Sin embargo la información provista por la pantalla anterior no es suficiente para efectuar el proceso de toma de decisión de manera efectiva y rápida, ya que existe falta de control sobre las

fallas presentadas por las bobinas, lo cual puede ser solventado a través de un sistema de reportes que permita visualizar las toneladas caídas por defecto/instalación así como gráficos que reflejen la ejecución de las acciones propuestas por los encargados de disminuir la ocurrencia de los defectos de las bobinas y el efecto obtenido a través de un periodo de tiempo especificado. Aunado a ello es necesario que al hacer cambios a nivel gerencial, y los mismos afecten la producción se cuente con el medio para la ejecución de estas actualizaciones.

El principal causante de esta situación radica en la ausencia de un sistema de información donde los gerentes, procesistas, inspectores, supervisores y jefes de plantas (personal autorizado), puedan acceder al mismo y así obtener la información en línea, a través de la intranet.

Por lo tanto, es necesario el desarrollo de un sistema de información que permita manejar los datos y a su vez, facilitar la gestión empresarial.

Debemos acotar que en el proceso productivo también existen otros factores que afectan la calidad, entre ellos podemos incluir indicadores como la productividad efectiva, efectividad y puesta a mil. Incluso si hay equipos defectuosos, es necesario abarcar estos rubros. A este respecto hoy en día se maneja la información correspondiente a los indicadores de calidad, equipos defectuosos, y bobinas defectuosas en diferentes sistemas, bien sea manual o en línea. Como medida de control de estos datos en particular se desea concentrar toda la información en un mismo sistema que agrupe la información diariamente y en los turnos de producción de la planta. También sería de interés contemplar en tal sistema un histórico de las novedades referentes a la productividad de acuerdo a tales sucesos de planta.

La coordinación de nivel 2B (quienes están encargados de realizar los desarrollos Web) junto con la gerencia de laminado (lo cual incluye laminación en frío, y laminación en caliente), han determinado que es necesario solventar la situación actual, y así aunar al sistema de gestión en línea (SGL), un nuevo sistema denominado sistema de gestión cualitativa (SGC) el cual mantendrá el control sobre los defectos presentados por las bobinas, y aunado al mismo el Libro de novedades, que concentrara la información de los sistemas que afectan la calidad de la

producción. Realmente ambos módulos de trabajo forman parte de un sistema de calidad para SIDOR, el cual inicialmente se implementará en el área de laminación, y a futuro en toda la planta.

Este sistema de gestión debe cumplir con los siguientes requisitos para la gestión adecuada de las bobinas:

- Clasificar todas las causas y acciones establecidas sobre las bobinas a las cuales se les detecta un defecto a nivel de planta.
- Asignar responsables por cada defecto que se detecte en las bobinas.
- Asignar causas y acciones a cada defecto detectado.
- Poder crear y asignar nuevas Causas por defecto detectado.
- Poder crear y asignar nuevas Acciones por defecto detectado.
- Cambiar cualquier estado de las acciones tomadas.
- A nivel de reuniones de comité, permitir cambiar estados de calidad de las bobinas de manera que algunas bobinas defectuosas puedan ser utilizadas para otros fines.
- Mostrar reportes en el tiempo con respecto a Defectos/Causas/Acciones.
- Mostrar reportes referentes a las mejores Acciones tomadas por defecto detectado.
- Mostrar reportes de evolución en el tiempo en cuanto a las toneladas caídas por defectos.
- Mantener un área de control sobre la justificación de los defectos ocurridos, las acciones correctivas y preventivas a tomar.
- Actualizar la reprogramación de los productos de laminado.

En cuanto a los requerimientos para el libro de novedades podemos mencionar:

- Información por turnos sobre los defectos observados en las líneas.
- Revisión detallada sobre el material descartado por las bobinas o puesta a mil (PAM).
- Medición de la efectividad.
- Medición de la productividad efectividad.
- Control sobre equipos en observación.

Sistema de Gestión de Defectos.

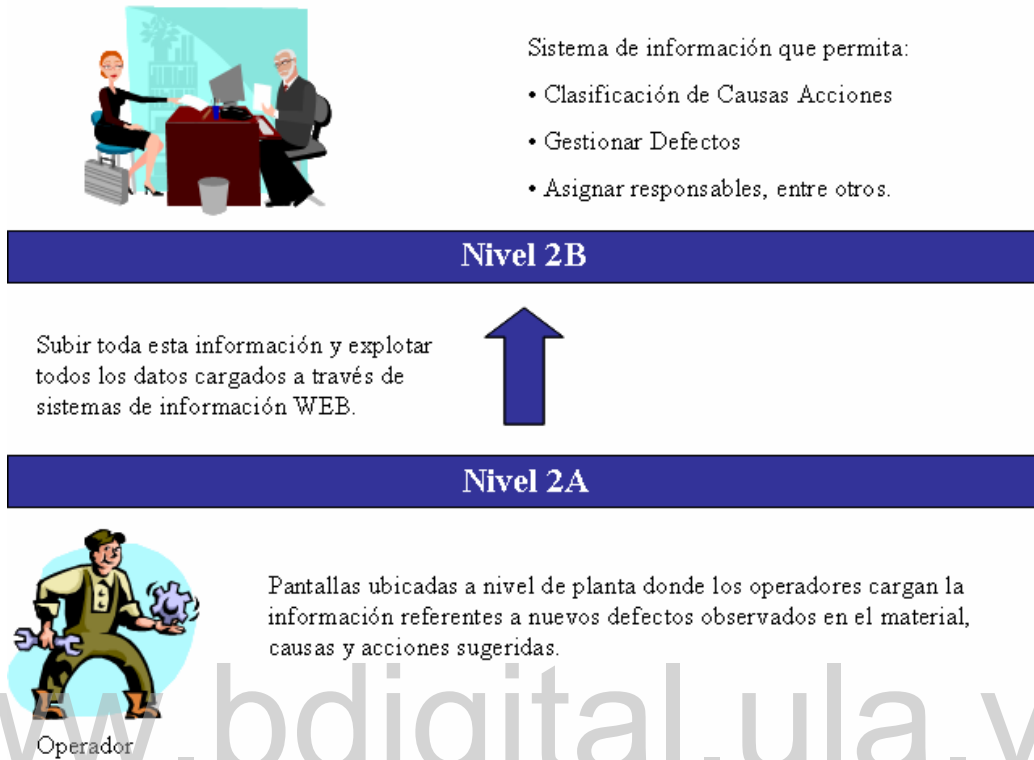


Figura 3: Gestión de Defectos

Sistema de Libro de novedades



Figura 4: Gestión del Libro de Novedades

Objetivos

Objetivo General:

Desarrollo e implementación del sistema de gestión de calidad, usando la tecnología Web.

Objetivos secundarios:

- Desarrollar un prototipo para el área de reportes en el sistema de gestión cualitativo y en el área de novedades ocurridas en planta.
- Desarrollar reportes en el tiempo con respecto a Defectos/Causas/Acciones.
- Desarrollar reportes referentes a las mejores acciones tomadas por defecto detectado sobre las bobinas.
- Desarrollar reportes que permitan el control del llenado de datos del sistema, para mantener su buen funcionamiento.
- Desarrollar reportes con revisión detallada sobre los defectos observado en los productos por turno.
- Desarrollar reportes detallados sobre el material descartado por bobina.
- Medición de la efectividad por turno/línea y justificación sobre las interrupciones acaecidas, a través de reportes diarios.
- Control sobre equipos en observación.
- Permitir el acceso a los manuales del sistema vía WEB.

Metodología.

Cuando se va a construir un sistema software es necesario conocer un lenguaje de programación, pero con eso no basta. Si se quiere que el sistema sea robusto y mantenible es necesario que el problema sea analizado y la solución sea cuidadosamente diseñada. Se debe seguir un proceso robusto, que incluya las actividades principales. Si se sigue un proceso de desarrollo que se ocupa de plantear cómo se realiza el análisis y el diseño, y cómo se relacionan los productos de ambos, entonces la construcción de sistemas software va a poder ser planificable y repetible, y la

probabilidad de obtener un sistema de mejor calidad al final del proceso aumenta considerablemente.

Durante mucho tiempo se ha utilizado el tradicional modelo en cascada, el cual ha demostrado que no refleja adecuadamente la complejidad inherente al proceso de desarrollo de software.

Los problemas que presenta este modelo nacen de su propia estructura, al ser una secuencia de grandes etapas que requieren como hitos la documentación completa antes de continuar con la siguiente etapa.

Para solucionar este problema se debe hacer uso de métodos iterativos e incrementales, que unidos a otras prácticas claves como la orientación al manejo de riesgos y la planeación adaptable, permiten de forma natural guiar adecuadamente el proceso de desarrollo de software.

Se ha escogido seguir este proceso debido a que aplica los últimos avances en Ingeniería del Software, y a que adopta un enfoque eminentemente práctico, aportando soluciones a las principales dudas y/o problemas con los que se enfrenta el desarrollador. Su mayor aportación consiste en atar los cabos sueltos que anteriores métodos dejan.

La notación que se usa para los distintos modelos, tal y como se ha dicho anteriormente, es la proporcionada por UML, que se ha convertido en el estándar de facto en cuanto a notación orientada a objetos. El uso de UML permite integrar con mayor facilidad en el equipo de desarrollo a nuevos miembros y compartir con otros equipos la documentación, pues es de esperar que cualquier desarrollador versado en orientación a objetos conozca y use UML (o se esté planteando su uso).

Se va a abarcar todo el ciclo de vida, empezando por los requisitos y acabando en el sistema funcionando, proporcionando así una visión completa y coherente de la producción de sistemas software. El enfoque que toma es el de un ciclo de vida iterativo incremental, el cual permite una gran flexibilidad a la hora de adaptarlo a un proyecto y a un equipo de desarrollo específicos. El

ciclo de vida está dirigido por casos de uso, es decir, por la funcionalidad que ofrece el sistema a los futuros usuarios del mismo. Así no se pierde de vista la motivación principal que debería estar en cualquier proceso de construcción de software: el resolver una necesidad del usuario/cliente.

Tal y como lo establece el Proceso Unificado, a través de su Lenguaje Unificado de Modelado (UML), el desarrollo del software se llevará a cabo en cuatro fases: inicio, elaboración, construcción y transición. Las tareas que se realizarán en cada una de estas fases se describen a continuación.

Fase de Inicio.

Con el propósito de hacer una descripción del producto software que se desarrollará, durante esta fase se procederá a analizar todos los requerimientos necesarios para lograr la funcionalidad del sistema. De esta manera, los requisitos propios del sistema se verán reflejados en un modelo de casos de uso, en donde se representarán aquellos módulos o subsistemas que permitan obtener un esbozo inicial de la arquitectura.

Una vez establecido el dominio del software, dado por los requisitos funcionales, se procederá a identificar los diferentes eventos que pudiesen afectar el desarrollo de la aplicación, mitigándolos en la medida de lo posible.

Por último, y habiendo analizado, estructurado y refinado los requerimientos del software, se procederá a identificar la posible arquitectura del software, delineando un modelo de diseño de la misma, en donde se muestren las funciones principales del sistema. La definición de la arquitectura, así como la línea base para su desarrollo, formarán parte de la siguiente fase de desarrollo del software: la fase de elaboración.

Fase de Elaboración.

Las actividades en esta fase se centran en el establecimiento de una arquitectura sólida, y el refinamiento de la mayor parte de los requisitos para desarrollar el software. Es aquí donde se

presentan vistas de todos los modelos del sistema, haciendo uso de diagramas de clase de análisis, de diseño y de colaboración.

Fase de construcción.

Para completar el software que se ha venido modelando en las fases de inicio y elaboración se realizará, como parte de la creación, el diseño de la interfaz de usuario, según los requisitos de rendimiento y eficiencia que pudieron ser establecidos en un comienzo. Por otro lado, se codificarán todos los procedimientos elaborados en la etapa anterior, utilizando una herramienta de programación orientada a objeto, considerando las características propias del software.

Posterior a la codificación, se integrarán todos los módulos pertenecientes al software de control, realizando todas las pruebas pertinentes, tales como las pruebas de unidad, para evaluar la eficiencia de los algoritmos utilizados, y corregir, en lo posible, aquellos errores que pudiesen ser detectados.

Finalmente, se integrará el software preparado y evaluado con el hardware (robot móvil), completando así la descripción y creación del producto propuesto, preparado ya para las últimas pruebas, antes de su entrega al usuario.

Fase de Transición.

En esta fase, se harán todas las pruebas necesarias al software desarrollado, con el fin de verificar si el mismo cumple con los requisitos que fueron establecidos en fases anteriores.

Para finalizar, se elaborarán los manuales de usuario y mantenimiento, que facilitarán tanto la manipulación del sistema por parte del usuario, como la posterior modificación y reutilización del mismo.

Alcance:

La finalidad de este proyecto es hacer el diseño, desarrollo e implementación, de un *Sistema de Calidad* el cual este compuesto por dos módulos principales, Gestión de Defectos, y Libro de

Novedades. En el primer modulo se pretende cubrir las necesidades en cuanto a la gestión de los defectos acaecidos en las bobinas a través de reportes que cubran la explotación de los datos que se están obteniendo desde planta, y de la gestión en si del sistema. En el segundo modulo se concentrara la información de diferentes sistemas con la finalidad de llevar un histórico de las novedades de planta para tener presente reportes específicos de los acontecimientos mas relevantes que afectan la productividad. Ambos módulos están dirigidos hacia las gerencias tanto de laminación como de calidad, quienes están encargados de la toma de decisiones y regulación de los sucesos de planta. Se prevé que el sistema estará disponible en la intranet de la empresa, lo cual permitirá que los usuarios accedan al mismo. El sistema estará soportado Microsoft Windows y Microsoft SQLSERVER 2000.

Estructura del documento.

Los capítulos siguientes seguirán la siguiente estructura:

Capitulo II: Marco Teórico.

En el se detallara la metodología a seguir, así como los conceptos necesarios para llevar a cabo a feliz termino la solución del problema planteado.

Capitulo III: Análisis de Requisitos

En esta sección se describirán de manera formal la evolución de los requerimientos del sistema exigidos por los usuarios del mismo a la par con la evolución del diseño teórico, y de las interfaces que el sistema.

Capitulo IV: Evolución, implementación y pruebas:

En esta sección se mostrará y describirá la evolución del sistema hasta llegar al producto final, junto con varias pruebas realizadas.

Capitulo V: Conclusiones y Recomendaciones.

Bibliografía.

Capítulo II:

Marco Teórico:

Este capítulo tiene la finalidad de sentar las bases teóricas fundamentales que se aplicaran durante el diseño, e implementación del sistema. Se detallará la metodología usada, así como las ventajas que ofrece el proceso unificado en el diseño, desarrollo del mismo. A su vez esta sección pretende informar al lector de este trabajo sobre algunos conceptos que no le sean familiares, y colaborar en que la información presentada sea comprensible.

Prácticas Claves del proceso de desarrollo de software

Desarrollo iterativo

El desarrollo iterativo es un método de construcción de productos cuyo ciclo de vida está compuesto por un conjunto de iteraciones, las cuales tienen como objetivo entregar versiones del software. Cada iteración se considera un subproyecto que genera productos de software y no sólo documentación, permitiendo al usuario tener puntos de verificación y control más rápidos e induciendo un proceso continuo de pruebas y de integración desde las primeras iteraciones. Las iteraciones están compuestas por el conjunto de disciplinas o actividades ya conocidas en el proceso de desarrollo de software. Estas son la especificación de requerimientos, el análisis y diseño, las pruebas, la administración de la configuración y el proceso de gerencia de proyectos.

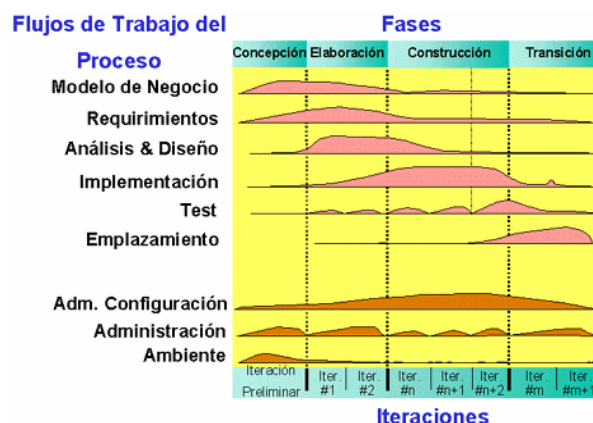


Figura 5: Fases del desarrollo

En la Figura 5 se muestra gráficamente la relación existente entre las disciplinas o actividades y las iteraciones.

Orientación al manejo del riesgo

Cada proyecto tiene asociado intrínsecamente un conjunto de riesgos que requieren un plan de manejo claramente establecido, documentado y con una implementación eficaz. De esta manera se pretende evitar posibles retrasos en los tiempos de entrega, problemas de calidad en el producto o en el peor de los casos, que puedan afectar la culminación del proyecto. Estos procesos pueden ser tan complejos y elaborados como la importancia del proyecto lo requiera. En las etapas iniciales se implementan las funcionalidades con mayor exposición al riesgo y las de mayor complejidad, mejorando la posibilidad de éxito del proyecto.

En la fase inicial del proyecto, el nivel de exposición al riesgo en ambos modelos es casi igual, pero en las fases siguientes es completamente diferente para cada modelo

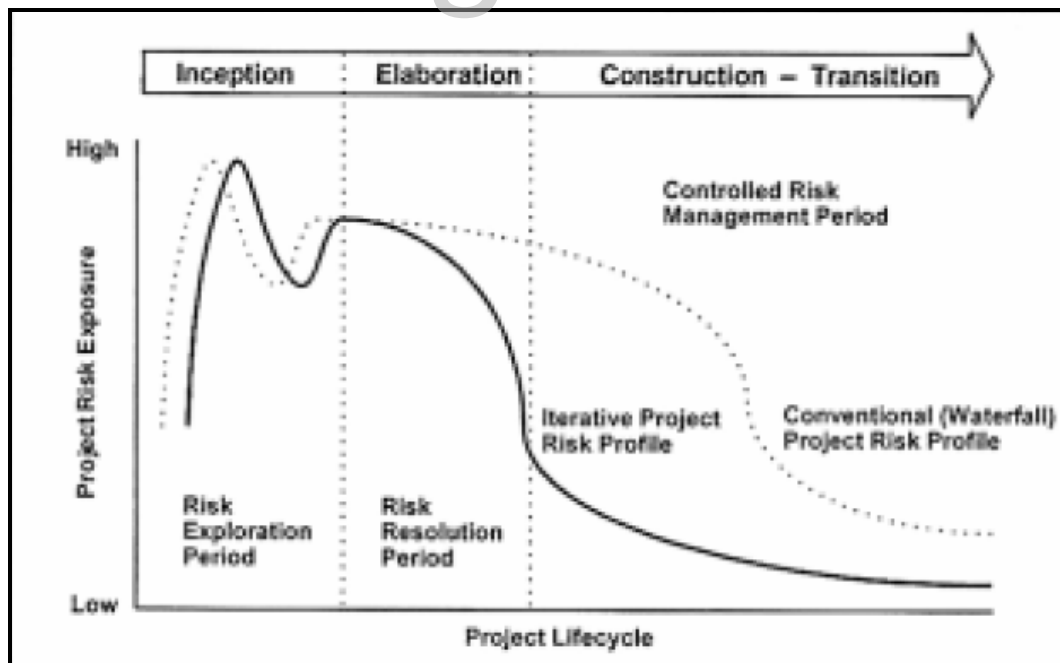


Figura 6: Perfiles de Riesgo

Este comportamiento se debe al período de exploración de riesgos del modelo en espiral, donde se identifican los riesgos, se priorizan y se define un plan de manejo para mitigarlos. Se procede a la fase de elaboración donde se implementan aquellos casos de uso que atacan los riesgos de más alta prioridad, lo cual se denomina período de resolución de riesgos. Al final de esta fase se debe tener definida la arquitectura del sistema, así como la infraestructura en la que se soportará.

Orientación al cliente

Cuando se inicia un proyecto de desarrollo de software se conoce la importancia de la participación del cliente para lograr su terminación exitosa, pero usualmente cometemos el error de olvidar esta norma básica, lo que implica que la participación del cliente se restringe al inicio y finalización del proyecto, lo que en la mayoría de los casos produce un alto grado de insatisfacción en el usuario, al no obtener el producto con las especificaciones esperadas.

El cliente es quien realmente conoce el valor que aportará el producto que está siendo desarrollado y puede definir las prioridades desde la perspectiva organizacional. Esto quiere decir que es necesario contar con su participación en el proceso de planificación de las fases y de las iteraciones. Posteriormente se requiere su participación en cada iteración para proveer retroalimentación temprana al equipo de desarrolladores, garantizando el cumplimiento de las expectativas que tiene, además de ofrecerle una visibilidad permanente del estado del proyecto, asegurando su compromiso para terminarlo exitosamente. Se debe tener en cuenta que el cliente no se interesa por los aspectos técnicos de alta complejidad y riesgo, razón por la cual se debe combinar esta práctica con una orientación al manejo del riesgo.

Desarrollo evolutivo

Cuando se trabaja con una especificación de requerimientos monolítica, se cae en el error de creer que se comprende completamente el concepto del producto sin haberlo validado con el cliente con algo más que documentos y modelos abstractos. Este proceso inicia con un concepto poco claro del producto a construir, y sólo se tiene claridad en la medida que se vaya desarrollando y verificando el producto con el cliente. Este tipo de proyectos se asemejan más al patrón que siguen los proyectos de investigación y desarrollo de nuevos productos.

Proceso Unificado:

Un proceso de desarrollo de software describe un enfoque para la construcción, desarrollo y posiblemente mantenimiento del software. El proceso unificado se ha convertido en un proceso de desarrollo de software de gran éxito para la a construcción de sistemas orientados a objetos.

El proceso unificado combina las practicas comúnmente aceptadas, tales como el ciclo de vida iterativo, y desarrollo dirigido por el riesgo. La concepción de un sistema de información va mucho más allá de levantar los requerimientos, elaborar un conjunto de modelos y comenzar a programar.

En este punto podemos considerar que la definición de la arquitectura del software se convierte en el eje orientador que permite controlar el desarrollo iterativo e incremental del sistema, a través de su ciclo de vida. Esta arquitectura se define en las primeras fases del proyecto, básicamente en la de elaboración, y se refina a través de todo el proyecto.

El RUP se fundamenta en seis prácticas: el desarrollo iterativo, la administración de requerimientos, la arquitectura basada en componentes, en el modelamiento visual, en la verificación continua de la calidad y la administración del cambio. Estas seis prácticas orientan el modelo y con ellas se pretende solucionar muchos de los problemas asociados al software.

Adicionalmente hay muchos aspectos de diseño que son bien conocidos, pero que en realidad han sido muy poco implementados en los proyectos de software; estos son: facilidad de uso, modularidad, encapsulamiento y facilidad de mantenimiento. Es necesario entonces definir una arquitectura sólida basada en componentes, para construir mejores y más flexibles soluciones de software para las necesidades organizacionales.

Los cambios en un proyecto no pueden ser detenidos dado que la evolución del entorno de cada organización es continua, pero sí pueden ser administrados de manera que su impacto pueda ser estimado para determinar si dicho cambio se incluye o no y si el proyecto debe ser reajustado. Cada cambio en el proyecto debe tener especificado cuándo y cómo se va a realizar, quién lo va a hacer y qué productos se ven involucrados en ese cambio. En ese punto es donde el control de

cambios y la trazabilidad de los componentes a través de los diversos modelos adquieren una gran importancia. Existen algunos aspectos que se deben tener en cuenta para desarrollar exitosamente un proyecto.

A continuación se enumeran algunos de ellos:

- Se debe tener definida claramente la metodología de trabajo de cada fase del proceso del desarrollo de software, en especial las fases de administración de requerimientos y control de cambios, los cuales son los eslabones más débiles del proceso de desarrollo de software en nuestras organizaciones.
- La participación activa de los usuarios y los acuerdos en los tiempos pactados, teniendo en cuenta los datos generados de los procesos de estimación y planificación, son responsabilidad del jefe del proyecto, pero deben ser elaborados con integrantes claves del equipo del proyecto.
- El Grupo debe definir, documentar y actualizar el proceso de aseguramiento de la calidad del software, gestionar los recursos necesarios para que sea operativo desde el comienzo del proyecto, entregar el plan de calidad y velar por su cumplimiento a lo largo del ciclo de vida del proyecto.
- El proceso de incorporación y utilización de nuevas tecnologías es quizás uno de los aspectos más críticos dentro del proyecto y de mayores riesgos. La definición de una metodología de administración del cambio tecnológico, clara y muy práctica, facilitaría considerablemente el trabajo realizado en la fase de elaboración, lo cual permitiría determinar la viabilidad de la incorporación de dicha tecnología en el proyecto. Teniendo en cuenta los aspectos mencionados previamente, Rational que recientemente fue comprada por IBM, elaboró un marco de referencia para el proceso de desarrollo de software basado en el modelo en espiral. Este método se conoce como RUP “Racional Unified Process”. Para una mejor organización, el RUP agrupa las iteraciones en etapas y fases que facilitan la administración del proyecto

Etapas del proceso unificado:

1. Etapa de ingeniería

Esta etapa agrupa las fases de concepción y de elaboración, lo que básicamente le da por objetivos la conceptualización del sistema y el diseño inicial de la solución del problema. Se inicia el proceso de administración de los requerimientos con la identificación y especificación de casos de usos, así como el proceso de aseguramiento de la calidad a través de los casos de prueba. Se identifican los riesgos y se establece su plan de manejo, se ajusta ese plan según la tabla de prioridad de riesgos y la de casos de usos vs. riesgos, para determinar en qué orden y en qué iteraciones se desarrollarán los artefactos de software que son la solución a los casos de uso. Se identifican los recursos necesarios, tanto económicos como humanos, acordes con las necesidades del proyecto. Se da comienzo al proceso de estimación y planificación inicial a un nivel macro para todo el proyecto y posteriormente se realiza una estimación detallada de tiempos y recursos de las fases de concepción y elaboración.

1.1. Fase de concepción

Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

1.1.1 Planeación de las fases y de las iteraciones

A partir del modelo de casos de uso y de la lista de riesgos, se puede determinar qué casos de uso deben implementarse primero para atacar los riesgos de mayor exposición. Con base en la información previa se realiza el proceso de planificación general y un plan de trabajo detallado para la siguiente fase, así como el plan para la siguiente iteración. Se debe establecer una relación clara y directa entre los casos de uso y los casos de prueba para facilitar que el proceso de aseguramiento de la calidad del software se ejecute adecuadamente. El plan de pruebas debe planearse en esta fase, ejecutarse desde la primera iteración de la fase de elaboración y refinarse sucesivamente durante el ciclo de vida del proyecto.

1.2. Fase de elaboración

Los casos de uso seleccionados para desarrollarse en esta fase permiten definir la arquitectura del sistema, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar del problema y comienza la ejecución del plan de manejo de riesgos, según las prioridades definidas en él.

Al final de la fase se determina la viabilidad de continuar el proyecto y si se decide proseguir, dado que la mayor parte de los riesgos han sido mitigados, se escriben los planes de trabajo de las etapas de construcción y transición y se detalla el plan de trabajo de la primera iteración de la fase de construcción.

2. Etapa de producción

En esta etapa se realiza un proceso de refinamiento de las estimaciones de tiempos y recursos para las fases de construcción y transición, se define un plan de mantenimiento para los productos entregados en la etapa de ingeniería, se implementan los casos de uso pendientes y se entrega el producto al cliente, garantizando la capacitación y el soporte adecuados.

2.1. Fase de construcción

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar el cambio de los artefactos construidos, ejecutar el plan de administración de recursos y mejoras en el proceso de desarrollo para el proyecto.

2.2. Fase de transición

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto al inicio del mismo.

ASP.NET

ASP.NET es un ambiente de programación construido sobre el entorno NGWS New Generation Windows Services, que permite crear poderosas aplicaciones de Internet ASP.NET ofrece varias

ventajas importantes sobre los modelos previos de desarrollo para Internet, entre las cuales podemos nombrar:

Mejor Eficiencia: ASP.NET corre código compilado sobre el entorno NGWS en el servidor. Distinto a sus predecesores interpretados, ASP.NET usa amarres tempranos ("early binding"), así como compilación justo a tiempo ("just-in-time compilation"), optimización nativa, y servicios de caché, sin configuración adicional. Para los desarrolladores, esto significa eficiencia dramáticamente superior antes de escribir la primera línea de código.¹

Herramientas superiores de desarrollo: ASP.NET tiene una "caja de herramientas" rica : el ambiente de desarrollo integrado de Visual Studio.NET. La edición WYSIWYG, la creación de controles mediante "drag-and-drop", y la publicación automática son varias ventajas.

Poder y Flexibilidad: Porque ASP.NET está basado en el Entorno Común de Ejecución de Lenguajes (Common Language Runtime, o "CLR"), el poder y la flexibilidad de la plataforma completa está disponible para los desarrolladores. Las librerías de Clases del CLR, la Mensajería, y las soluciones de Acceso a Datos, son accesibles al través del Internet. ASP.NET permite el uso de una gran variedad de lenguajes de programación y, por tanto, usted puede escoger el mejor lenguaje para su aplicación, o particionar su aplicación en varios lenguajes. Mas aún, la interoperabilidad del CLR garantiza que su inversión en el desarrollo de aplicaciones COM es preservada cuando se migra a ASP.NET

Simplicidad: ASP.NET hace fácil el ejecutar tareas comunes, desde el simple envío de un formulario o la autenticación de un cliente, hasta el despliegue y la configuración de un Web. Por ejemplo, el entorno de paginado de ASP.NET le permite construir interfases de usuario que separan limpiamente la lógica de su aplicación del código de su presentación, y maneja eventos con un modelo sencillo de procesamiento de formularios al estilo de Visual Basic. Adicionalmente, el CLR simplifica el desarrollo con servicios de código gerenciado, como el

¹ Tomado del libro "ASP.NET Manual de referencia"

conteo automático de referencias y la limpieza automática de la memoria utilizada por su aplicación.

Gerenciabilidad: ASP.NET usa un sistema jerárquico de configuración, basado en archivos de texto, que simplifica la aplicación de parámetros de configuración al servidor y sus aplicaciones. Porque la información de configuración es almacenada como texto, nuevos parámetros pueden ser configurados sin recurrir a herramientas de administración locales. Esta filosofía de "cero administración local" también se extiende al despliegue de aplicaciones de ASP.NET. Una aplicación de ASP.NET se despliega a un servidor simplemente copiando los archivos necesarios al servidor. No hay que reiniciar el servidor, ni siquiera para reemplazar código compilado que ya está en servicio.

Escalabilidad y Disponibilidad: ASP.NET ha sido diseñado para la escalabilidad con características específicamente dirigidas a mejorar el funcionamiento de servidores racimados (clustered) y de servidores con procesadores múltiples. Los procesos del servidor son vigilados y gerenciados por el entorno del ambiente de ejecución de ASP.NET, así que si algún proceso se entorpece o se detiene, un nuevo proceso puede ser creado para reemplazarlo, lo cual ayuda a mantener la disponibilidad de su aplicación para manejar solicitudes de servicio.

Personalización y Extensibilidad: ASP.NET entrega una arquitectura bien formada que permite que los desarrolladores "enchufen" su código al nivel apropiado. De hecho, es posible el extender o reemplazar cualquier sub.-componente del ambiente de ejecución de ASP.NET con un componente personalizado. La implementación de autenticación personalizada o de servicios de mantenimiento de estado nunca ha sido tan sencillo.

Seguridad: Una parte importante de muchas aplicaciones Web radica en la capacidad de identificar usuarios y controlar el acceso a los recursos. Se conoce como autenticación al acto de determinar la identidad de la entidad solicitante. Por lo general, el usuario deberá presentar sus credenciales, como el par nombre de usuario y contraseña, para ser autenticado. Cuando se encuentre disponible una identidad autenticada, deberá determinarse si esa identidad puede tener

acceso a un recurso específico. Este proceso se conoce como autorización. ASP.NET e IIS colaboran para proporcionar servicios de autenticación y autorización a las aplicaciones.

Una característica importante de los objetos COM es la capacidad de controlar la identidad con la que se ejecuta el código de objeto COM. Se conoce como representación al hecho de que un objeto COM ejecute código con la identidad de la entidad solicitante. Opcionalmente, las aplicaciones ASP.NET Framework pueden representar solicitudes.

Es posible que algunas aplicaciones también personalicen dinámicamente el contenido en función de la identidad del solicitante o de un conjunto de funciones al que pertenece la identidad del solicitante. Las aplicaciones ASP.NET Framework pueden comprobar dinámicamente si la entidad solicitante actual participa en una función concreta. Por ejemplo, es posible que una aplicación intente comprobar si el usuario actual pertenece a la función del administrador a fin de generar condicionalmente contenido para los administradores.

Sistemas de información:

"Un sistema es un conjunto de elementos materiales, o inmateriales, (hombres, maquinas, métodos, reglas, etc.) relacionados que transforman, mediante procesos, elementos (entradas) en otros nuevos elementos (salidas)" ²

En nuestro tema sólo nos interesan aquellos sistemas que estén constituidos por organizaciones (empresas, departamentos de empresas, organismos públicos, etc.) que *funcionen con el fin de obtener unos objetivos* determinados, y dentro de ellos, nos centraremos en los sistemas de información informatizados, es decir que emplean ordenadores.

Podemos tener así una nueva definición de sistema: "*conjunto de elementos que interactúan entre si, orientados a la consecución de un objetivo común*".

Un sistema puede formar parte de uno más general, que seria su entorno, y/o estar formado por otros sistemas, que se llamarían subsistemas.

Cualquier empresa u organización es en sí un sistema. Su entorno es el sistema productivo en el que se inserta, del que recibe una serie de entradas, en forma de demandas, y al que entrega una serie de salidas, u oferta. A su vez dentro de él existen varios subsistemas formados por los distintos departamentos o áreas en las que se divide la empresa.

² Tomado de <http://monografias.com>

Clasificación de los sistemas:

Sistema físico: o sistema operativo: transforma un flujo físico de entradas (materias primas, flujos financieros,) en un flujo físico de salidas (productos terminados, flujos financieros,)

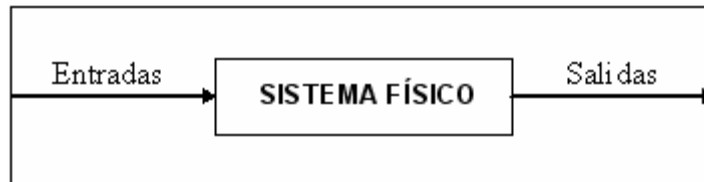


Figura 7: Diagrama de sistema.

Sistema de gestión: o sistema de control que se encarga de controlar el funcionamiento del sistema físico, se puede equivaler con la dirección de la empresa.

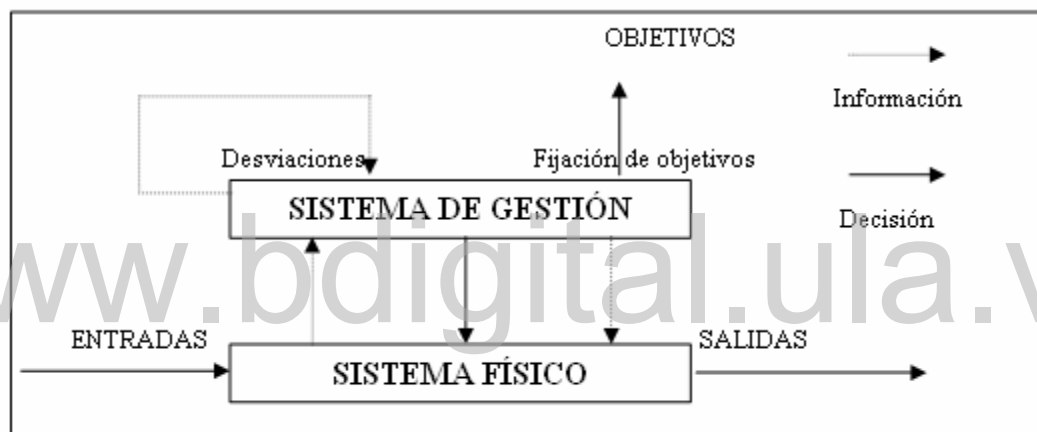


Figura 8: Sistema de gestión

El sistema de información: Es un sistema que actúa de intermediario entre el sistema operativo y el sistema de gestión. Vendría a ser el departamento de informática de la empresa.

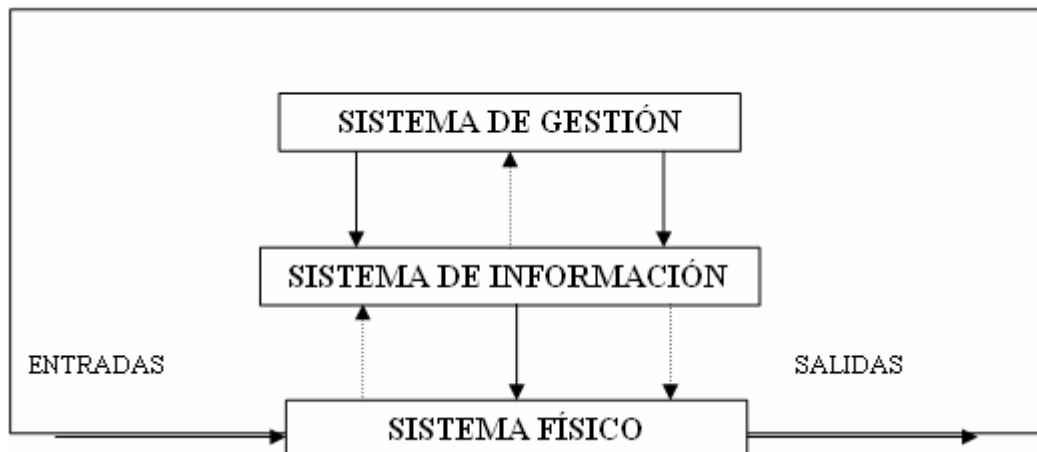


Figura 9: Sistema de información

Para que los subsistemas que forman una organización funcionen coordinadamente es necesario que exista este subsistema, que tiene como misión asegurar que la información necesaria fluya, dentro del sistema, de unos subsistemas a otros. Toda organización evolucionada debe disponer, hoy en día, de un buen sistema de información, ya que de ello dependerá en gran parte el éxito de la misma a la hora de alcanzar los objetivos propuestos.

Sistema de información informatizado:

Cuando un sistema de información cuenta entre sus recursos con ordenadores, en los que se basa parcial o totalmente, puede denominarse sistema de información informatizado. También se les suele llamar sistemas informáticos.

Dos razones justifican la informatización de un sistema de información:

- **Simplificación y mejora del trabajo administrativo** (contabilidad, facturación, nómina, etc.) por la automatización de procesos repetitivos y tediosos de simple ejecución.
- **Ayuda a la toma de decisiones:** En ocasiones el propio ordenador puede tomar decisiones, de acuerdo con unas pautas, o reglas. Cuando la decisión la tienen que tomar las personas, este sistema le permite disponer el máximo de información posible, ya que el ordenador puede seleccionar a gran velocidad entre una gran masa de datos almacenados, aquellas informaciones útiles para la toma de decisiones.

También es una herramienta útil para hacer simulaciones (por ejemplo con hojas de cálculo) que permiten a los gestores medir rápidamente cuales serían las consecuencias de, determinadas decisiones con el fin de encontrar progresivamente las mejores decisiones posibles

Capítulo III

Fase de inicio:

Esta fase comprende una serie de flujos de trabajo que permitirán construir el sistema gradualmente. Dichos flujos de trabajo están definidos en las fases principales del proceso unificado, al cual ya hemos hecho referencia en el capítulo 2.

Este capítulo tiene como finalidad describir las actividades realizadas en la fase de inicio del sistema de gestión de calidad, haciendo énfasis especial en los flujos de trabajo de los requisitos, análisis y diseño del mismo.

En si el esfuerzo principal implica desarrollar un modelo de sistema que se va a construir, y usar caso de uso de forma adecuada para la creación de este modelo. Esto es debido a que los requisitos funcionales se estructuran mejor a través de los casos de uso y a que la mayoría de requisitos no funcionales son específicos de un solo caso de uso, y pueden tratarse en este contexto. Este modelo que se describe en la fase de inicio, nos permitirá tener una visión inicial de los objetivos del proyecto. En nuestro caso no haremos un estudio de viabilidad, ya que la empresa SIDOR, anteriormente decidió que se llevaría a cabo el diseño, desarrollo e implementación de este sistema, por lo que este capítulo será relativamente breve a este respecto.

Cuestiones a abordarse en la fase de inicio:

- Visión y análisis del negocio: En esta parte se describirán los objetivos del sistema, así como un bosquejo del mismo.
- Modelos de casos de uso: Describe los requisitos funcionales y no funcionales relacionados.
- Glosario: Terminología clave del dominio del sistema.
- Plan de iteración: Describe que hacer en la primera iteración de la elaboración.

Visión general:

Como se ha mencionado en el capítulo 1, el sistema de gestión de calidad para SIDOR, comprende dos sistemas bien definidos para colaborar con la gestión de calidad de la empresa.

Estos sistemas son: El área de reportes del Sistema de Gestión de Defectos, y el Libro de Novedades. A continuación haremos una descripción general de ambos.

El **Sistema de Gestión de Defectos (SGD)** es un sistema de información, el cual permitirá a todo el personal autorizado y capacitado, llevar a cabo todas las labores de gestión de defectos que se detectan en los productos de las diferentes líneas de laminación. El área de reportes de este sistema permitirá ver la evolución de los defectos en el tiempo para las diferentes áreas de proceso de los productos, con dichos reportes, el personal capacitado podrá hacer estudios cualitativos que permita completar el ciclo de gestión, y analizar a largo plazo el efecto de las decisiones tomadas con referencia a la aparición de defectos en los productos.

El **Libro de Novedades (LN)** es un sistema de información que tiene como finalidad condensar en un solo site la información mas relevante obtenida en los turnos de trabajos de la empresa. Esta información comprende la visualización de los datos de los siguientes rubros:

- Calidad
- Indicadores.
- Equipos.
- Reporte general.
- Novedades Generales.

El objetivo principal del libro de novedades radica en que los supervisores y gerentes puedan tener acceso a esta información fundamental de las líneas de producción de la empresa, y manejar novedades específicas que los gerentes deseen informar a los supervisores vía Web.

Calidad: Partiendo de la información encontrada en el **sistema de gestión de defectos (SGD)**, se condensará la información correspondiente a los defectos ocurridos en el material producido. Se desea presentar la información del material defectuoso agrupado por defecto, mostrando los kilogramos afectados así como la cantidad de productos que presentaron el defecto.

Indicadores: El área de indicadores incluirá tres partes las cuales contendrán básicamente los indicadores de la producción registrada para las líneas de planta, tal registro se deberá llevar diariamente, y por turnos. Así mismo para cada indicador, se debe almacenar las novedades generadas a modo informativo. Estos indicadores son, la productividad efectiva, la efectividad, y puesta a mil. La información será extraída de dos sistemas que mantienen registros sobre tales datos, estos sistemas son el Sistema de Interrupciones y Puesta a Mil.

Reporte general: este reporte comprenderá la información que hoy en día se maneja a través de Excel, mostrando un reporte general de los datos que ayudan a gestionar el control de la producción.

Novedades Generales: desde esta parte del sistema LN, los gerentes podrán cargar novedades que deberán ser aplicadas a las áreas que ellos deseen que los superintendentes se informen.

Ambos sistemas estarán ubicados en los servidores de producción, los cuales se encuentran en el laboratorio de automatización de la empresa. El acceso al sistema por parte de los usuarios se hará a través de la intranet de SIDOR, ya que el sistema será desarrollado con tecnología Web. Es decir, cada usuario deberá tener un computador cliente con acceso a la intranet privada de la organización, para poder interactuar con estos sistemas.

Obtención del modelo de casos de uso:

El modelo de casos de uso proporciona la entrada fundamental para el análisis, el diseño y las pruebas. Este modelo contiene ciertos artefactos que permiten describir el sistema, entre ellos se encuentran los actores, casos de uso y sus relaciones. Es importante resaltar que este modelo de casos de uso permite tener una visión inicial del sistema, y para describirlo adecuadamente, primero hablaremos de los requisitos previos, y luego culminaremos con la descripción de los casos de uso derivados de los requisitos del sistema.

Requisitos funcionales:

Implican todas las acciones que el sistema debe realizar y que son determinadas una vez entendido el dominio del software, estos requisitos van a la par con la identificación de los casos

de uso. Para el área de reportes del SGD, podemos identificar los siguientes requisitos funcionales:

- Generación de reportes de toneladas afectadas por defecto e instalación de cada gerencia de laminación.
- Visualización de bobinas con defecto específico para asignación de responsables, y línea afectada.
- Visualización grafica de evolución de defectos por toneladas caídas en el tiempo.
- Visualización grafica de evolución de defectos de acuerdo a las acciones aplicadas en el tiempo.
- Visualización de material defectuoso a los cuales no se les ha cargado la información de correspondiente a las causas posibles del defecto, acciones a tomar para prevención de los mismos.
- Visualización de bobinas asignadas por diferentes líneas de producción a la línea propia.
- Generación de reportes de los defectos gestionados, y los responsables de llevar a cabo las acciones adecuadas para tal gestión.

Para el LN los requisitos funcionales son:

- Revisión de material defectuoso por línea de producción.
- Revisión de material descartado por bobina en cada línea de producción.
- Revisión de indicadores de producción por línea de producción.
- Revisión de equipos en estado de observación.
- Crear y/o modificar novedades generales.
- Crear y/o modificar comentarios particulares por registros de las revisiones anteriores.
- Crear y/o modificar novedades generales provenientes de los gerentes hacia los supervisores de planta.
- Generación de reportes diarios de defectos, equipos defectuosos y material defectuoso

Actores y casos de uso:

Después de haber recopilado los requisitos funcionales, se determinaron todos aquellos entes externos al sistema que interactúan con el y que son conocidos como actores. Los actores son el entorno del sistema y se comunican con el mismo mediante el envío y recepción de mensajes desde y hacia el sistema, según éste efectúe los casos de uso.

Actores:

Los actores que intervendrán en el SGD y LN fueron escogidos analizando los usuarios y sistemas que lo usan, tomando como base la información proporcionada por la visión general del sistema así como los requisitos funcionales anteriormente descritos.

A continuación presentaremos una tabla que contiene el actor junto con la descripción del mismo:

ACTOR	OBJETIVOS
Jefe de Línea	<ul style="list-style-type: none"> • Revisar como Jefe de línea la carga de causas de defectos del material, la carga de acciones correctivas y preventivas de a ese defecto. • Visualización grafica de evolución de defectos en el tiempo.
Inspector	<ul style="list-style-type: none"> • Revisar como Inspector de línea la carga de causas de defectos del material, la carga de acciones correctivas y preventivas de ese defecto. • Visualización grafica de evolución de defectos en el tiempo.
Supervisor	<ul style="list-style-type: none"> • Revisar como Supervisor de Línea la carga de causas de defectos del material, la carga de acciones correctivas y preventivas de ese defecto. • Visualización grafica de evolución de defectos en el tiempo.
Jefe de Procesos	<ul style="list-style-type: none"> • Visualizar toneladas afectadas por defecto, en cada línea de la gerencia. • Análisis de reportes de los defectos gestionados, y los responsables de llevar a cabo las acciones adecuadas para tal gestión.



Procesista	<ul style="list-style-type: none">• Análisis de reportes de los defectos gestionados, y los responsables de llevar a cabo las acciones adecuadas para tal gestión.• Análisis de evolución de defectos de acuerdo a las acciones aplicadas en el tiempo.
------------	--

Listado de actores para el modulo Libro de novedades:

Actor	Objetivos
Supervisores	<ul style="list-style-type: none">• Revisar el material defectuoso por línea de producción.• Revisar el material descartado por bobina en cada línea de producción.• Revisar los indicadores de producción por línea de producción.• Revisar los equipos en estado de observación..• Crear y/o modificar novedades generales.
Gerentes	<ul style="list-style-type: none">• Revisar el material defectuoso por línea de producción.• Revisar el material descartado por bobina en cada línea de producción.• Revisar los indicadores de producción por línea de producción.• Revisar los equipos en estado de observación.• Crear y/o modificar novedades generales provenientes para los supervisores de planta.

Modelos de Casos de Uso:

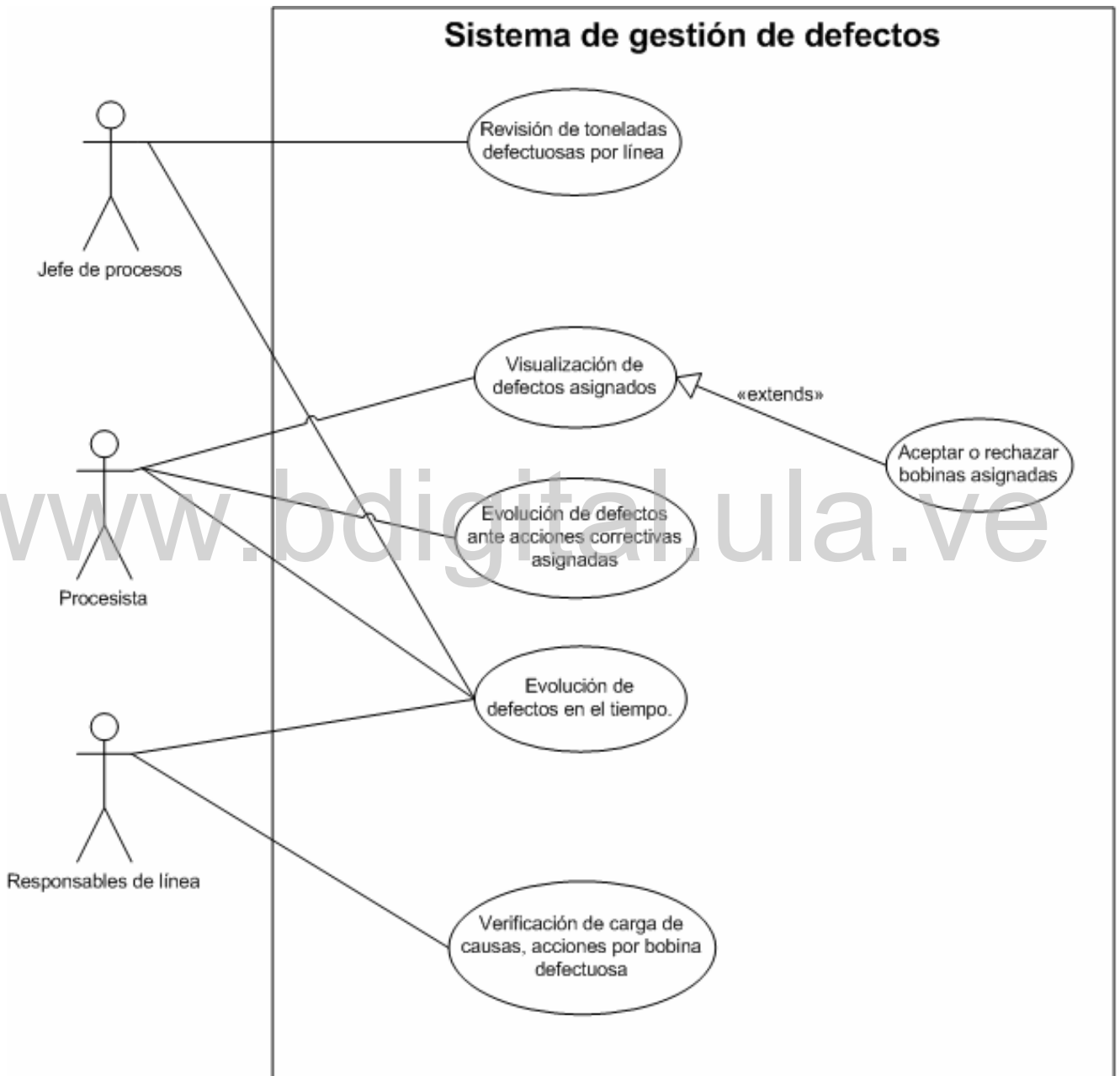


Figura 10: Diagrama de casos de uso SGD

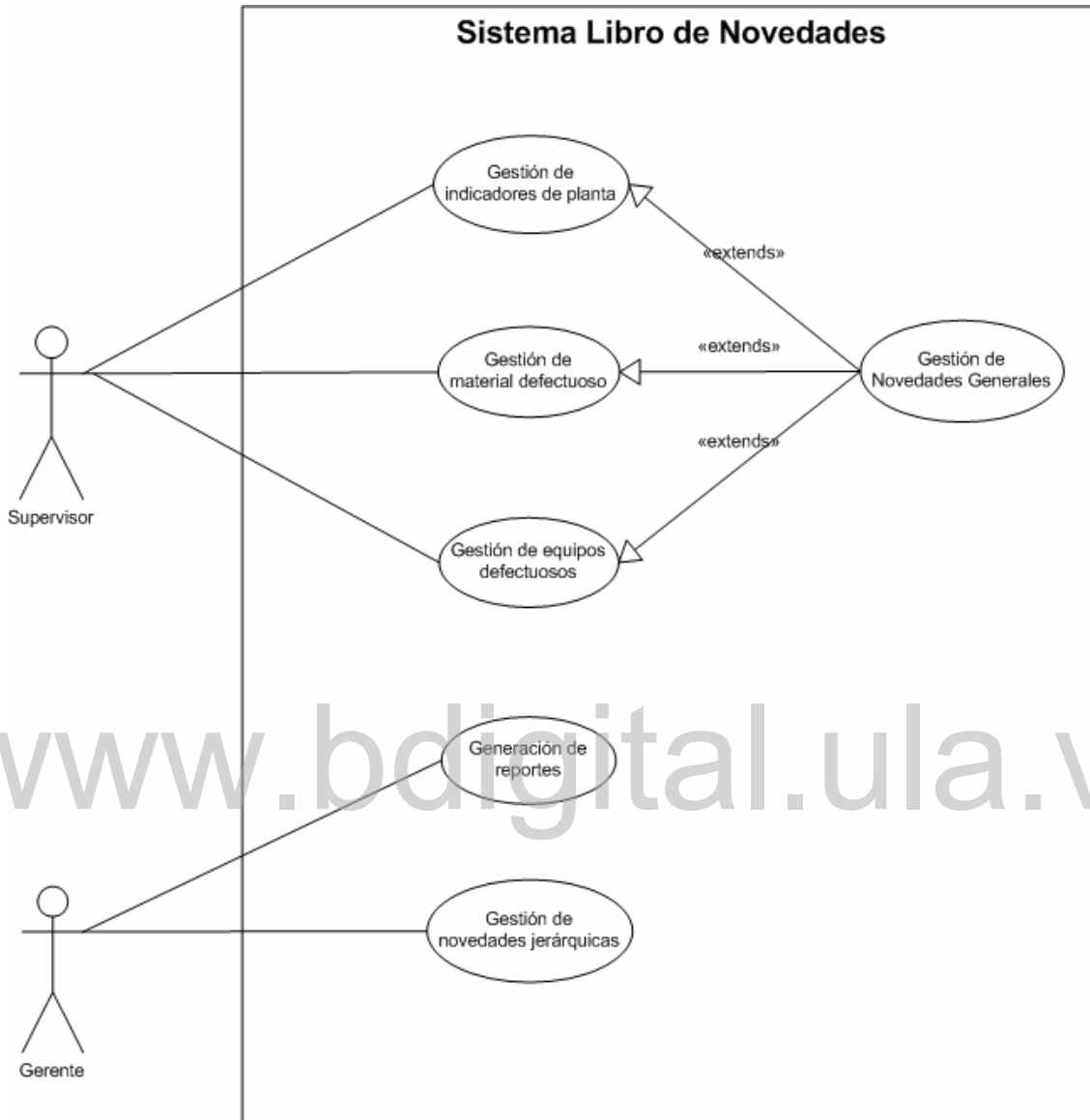


Figura 11: Diagrama de casos de uso LN

Descripción de casos de uso

Casos de uso del subsistema de gestión de defectos.

Revisión de toneladas defectuosas por línea

Actor Principal:

Jefe de procesos: Visualizar la cantidad de defectos presentados en su línea de producción, y las toneladas de material que presentaron tal defecto.

<p>Poscondiciones: Visualización de defectos presentados por línea, junto con la cantidad de material afectado por tal defecto.</p>	
<p>Escenario principal de éxito:</p>	
<ul style="list-style-type: none"> • Usuario accede al sistema, selecciona el rango de fechas dentro del cual se mostrará las toneladas afectadas por instalación. • El usuario selecciona el o los estados de calidad, para las bobinas defectuosas a mostrar. • El usuario selecciona la o las aplicaciones, para las bobinas defectuosas a mostrar. 	<ul style="list-style-type: none"> • Sistema toma la gerencia desde donde se accede al sistema. • Se visualiza un listado de defectos por instalación indicando la cantidad en toneladas que han presentado el defecto en cuestión.
<p>Escenarios o flujos alternativos:</p>	
<ul style="list-style-type: none"> • Usuario escoge un rango de fechas incoherente. • Usuario no selecciona el estado de calidad de las bobinas a 	<ul style="list-style-type: none"> • El sistema genera un mensaje de error.

mostrar.	<ul style="list-style-type: none"> El sistema genera un mensaje de error indicando al usuario que debe cargar el dato correspondiente.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Visualización de defectos asignados:

Actor Principal:	
Procesista: Visualizar las bobinas asignadas en su línea de producción.	
Precondiciones: El usuario debe estar cargado en el sistema como procesista.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> Usuario accede al sistema, selecciona el rango de fechas. El usuario selecciona el o los estados de calidad, para las bobinas defectuosas a mostrar. El usuario selecciona las cuadrillas en las cuales se produjo la bobina. El usuario selecciona la línea de producción que desea consultar. El usuario puede repetir los pasos anteriormente, para cualquier cambio en la consulta de los datos. 	<ul style="list-style-type: none"> Se visualiza un listado de bobinas defectuosas, que han sido asignadas.
Escenarios o flujos alternativos:	

<ul style="list-style-type: none"> • Usuario escoge un rango de fechas incoherente. 	<ul style="list-style-type: none"> • El sistema genera un mensaje de error.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Aceptar o rechazar bobinas asignadas.

Actor Principal:	
Jefe de procesos: Analiza las bobinas asignadas y decide firmarlas o aceptarlas definitivamente.	
Poscondiciones: Aceptación o rechazo definitivo de bobinas asignadas.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> • Usuario chequea cada bobina que desee aceptar. 	<ul style="list-style-type: none"> • El sistema guarda los datos asignados.

Evolución de defectos ante acciones correctivas asignadas:

Actor Principal:	
Procesista: Visualizar gráficamente la evolución de un defecto y las acciones aplicadas en un intervalo de tiempo.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> • Usuario accede al sistema, selecciona el rango de fechas. • El usuario selecciona el o los estados de calidad del defecto. • El usuario selecciona el defecto que 	

<p>desea analizar.</p> <ul style="list-style-type: none"> • El usuario selecciona la línea de producción que desea consultar. • El usuario puede repetir los pasos anteriormente, para cualquier cambio en la consulta de los datos. 	<ul style="list-style-type: none"> • Generación de un gráfico reportando las toneladas afectadas en un periodo de tiempo, y un indicador de las acciones aplicadas durante tal periodo.
<p>Escenarios o flujos alternativos:</p>	
<ul style="list-style-type: none"> • Usuario escoge un rango de fechas incoherente. 	<ul style="list-style-type: none"> • El sistema genera un mensaje de error.
<p>Requisitos especiales:</p>	
<p>Respuesta del sistema para la visualización de datos no mayor a 30 segundos.</p>	

Evolución de defectos en el tiempo:

<p>Actor Principal:</p> <p>Jefe de procesos: Visualizar gráficamente la evolución de un defecto en un intervalo de tiempo, para verificar el efecto de la gestión de defectos.</p> <p>Procesista: Visualizar gráficamente la evolución de un defecto en un intervalo de tiempo.</p> <p>Responsable de línea: Visualizar gráficamente la evolución de un defecto en un intervalo de tiempo.</p>	
<p>Escenario principal de éxito:</p>	
<ul style="list-style-type: none"> • Usuario accede al sistema, selecciona 	



<p>el rango de fechas.</p> <ul style="list-style-type: none"> • El usuario selecciona el o los estados de calidad del defecto. • El usuario selecciona el defecto que desea analizar. • El usuario selecciona la línea de producción que desea consultar. <ul style="list-style-type: none"> • El usuario puede repetir los pasos anteriormente, para cualquier cambio en la consulta de los datos. 	<ul style="list-style-type: none"> • Generación de un gráfico reportando las toneladas afectadas en un periodo de tiempo determinado por el usuario.
<p>Escenarios o flujos alternativos:</p>	
<ul style="list-style-type: none"> • Usuario escoge un rango de fechas incoherente. 	<ul style="list-style-type: none"> • El sistema genera un mensaje de error.
<p>Requisitos especiales:</p>	
<p>Respuesta del sistema para la visualización de datos no mayor a 30 segundos.</p>	

Verificación de carga de causas, acciones por bobina defectuosa:

<p>Actor Principal:</p> <p>Responsable de línea: visualizar listado de bobinas defectuosas a las cuales no se les ha cargado los datos correspondientes a las causas del defecto y acciones para corregirlos.</p>	
<p>Escenario principal de éxito:</p>	
<ul style="list-style-type: none"> • Usuario accede al sistema, selecciona 	

<p>el rango de fechas.</p> <ul style="list-style-type: none"> • El usuario selecciona el o los estados de calidad, para las bobinas defectuosas a mostrar. • El usuario selecciona la aplicación de las bobinas defectuosas. • El usuario selecciona las cuadrillas en las cuales se produjo la bobina. • El usuario selecciona la línea de producción que desea consultar. • El usuario selecciona el origen del defecto. • El usuario selecciona el campo que no han llenado los operadores de planta de un listado de campos posibles. 	<ul style="list-style-type: none"> • Se visualiza un listado de bobinas defectuosas, que han sido registradas en el proceso productivo, y los operadores no le han cargado el campo seleccionado por el usuario.
<p>Escenarios o flujos alternativos:</p>	
<ul style="list-style-type: none"> • Usuario escoge un rango de fechas incoherente. • Usuario escoge no escoge el origen del defecto seleccionado. 	<ul style="list-style-type: none"> • El sistema genera un mensaje de error, indicando el campo inadecuado.

Requisitos especiales:
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.

Casos de uso para el subsistema libro de novedades:

Gestión de indicadores de planta:

Actor Principal:	
Supervisor: Desea visualizar el estado general de planta, visualizando los indicadores correspondientes a la productividad y efectividad de la planta.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> • Usuario accede al sistema, y selecciona el indicador que desea consultar, bien sea la productividad efectiva, la efectividad o puesta a mil. 	<ul style="list-style-type: none"> • De acuerdo al indicador seleccionado el sistema presenta los datos correspondientes a tal indicador, de acuerdo a la fecha y el turno actual.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Dentro de la gestión de indicadores de planta, cabe mencionar que se generan otros casos de uso de acuerdo al indicador seleccionado.

A continuación los detallamos:

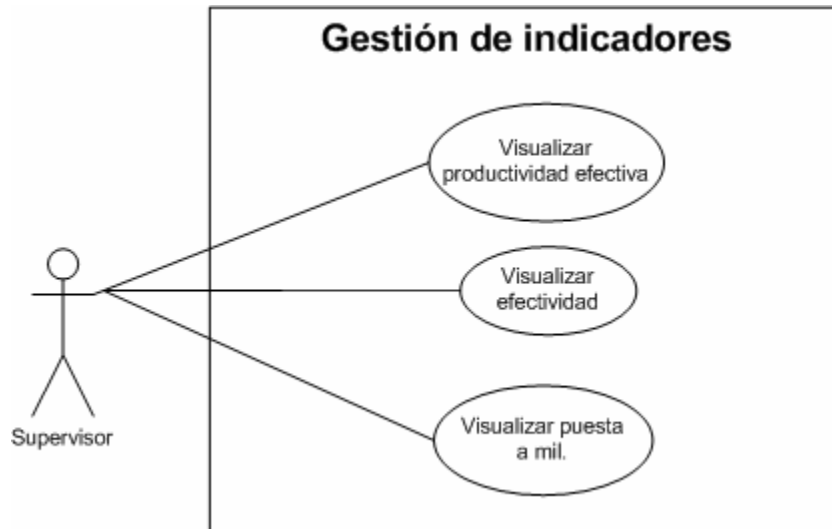


Figura 12: UCS Gestión de indicadores

Visualizar productividad efectiva:

<p>Actor Principal:</p> <p>Supervisor: Visualizar los registros de las interrupciones acaecidas en planta, que afectaron el indicador de productividad efectiva.</p>	
<p>Poscondiciones: Visualización de las interrupciones ocurridas en los sistemas definidos para cada línea de producción.</p>	
<p>Escenario principal de éxito:</p>	
<ul style="list-style-type: none"> • Usuario accede al sistema. 	<ul style="list-style-type: none"> • El sistema toma la fecha actual, y el turno correspondiente a la fecha, y la línea de producción. • El sistema genera los registros correspondientes a las interrupciones ocurridas en planta agrupadas por los sistemas pertenecientes a la línea de producción consultada, para cada agrupación el sistema desplegará los



	<p>detalles de los registros de tales interrupciones.</p> <ul style="list-style-type: none"> • El sistema genera los datos de la productividad efectiva real, estándar, y la desviación estándar obtenida para tal turno de producción.
Escenarios o flujos alternativos:	
<p>Usuario elige una nueva fecha de consulta, o elige una línea específica de producción, o elige un turno diferente de producción.</p>	<ul style="list-style-type: none"> • El sistema toma el dato actualizado, y se visualizan los datos requeridos por el usuario.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Visualizar efectividad:

<p>Actor Principal:</p> <p>Supervisor: Visualizar los registros de las interrupciones acaecidas en planta, que afectaron el indicador de efectividad</p>	
<p>Poscondiciones: Visualización de las interrupciones ocurridas en los sistemas definidos para cada línea de producción.</p>	
<p>Escenario principal de éxito:</p>	
<ul style="list-style-type: none"> • Usuario accede al sistema. 	<ul style="list-style-type: none"> • El sistema toma la fecha actual, y el turno correspondiente a la fecha, y la



	<p>línea de producción.</p> <ul style="list-style-type: none"> • El sistema genera los registros correspondientes a las interrupciones ocurridas en planta agrupadas por tipo de interrupción, zona, sistema, subsistema donde se presentó la misma. • El sistema genera los datos de la efectividad real, estándar, y la desviación estándar obtenida para tal turno de producción.
<p>Escenarios o flujos alternativos:</p>	
<p>Usuario elige una nueva fecha de consulta, o elige una línea específica de producción, o elige un turno diferente de producción.</p>	<ul style="list-style-type: none"> • El sistema toma el dato actualizado, y se visualizan los datos requeridos por el usuario.
<p>Requisitos especiales:</p>	
<p>Respuesta del sistema para la visualización de datos no mayor a 30 segundos.</p>	

Visualizar puesta a mil:

<p>Actor Principal:</p>
<p>Supervisor: Visualizar los productos que sufrieron descarte de material por defecto.</p>
<p>Poscondiciones: Visualización de toneladas descartadas por defecto encontrado en las bobinas producidas.</p>
<p>Escenario principal de éxito:</p>

<ul style="list-style-type: none"> • Usuario accede al sistema. 	<ul style="list-style-type: none"> • El sistema toma la fecha actual, y el turno correspondiente a la fecha, y la línea de producción. • El sistema genera los datos correspondientes a la cantidad de bobinas afectadas por defecto. • El sistema genera los datos de puesta a mil estándar, real y la desviación a modo de comparación para el usuario.
<p>Escenarios o flujos alternativos:</p>	
<p>Usuario elige una nueva fecha de consulta, elige una línea específica de producción, elige un turno diferente de producción.</p>	<ul style="list-style-type: none"> • El sistema toma el dato actualizado, y se visualizan los datos requeridos por el usuario.
<p>Requisitos especiales:</p>	
<p>Respuesta del sistema para la visualización de datos no mayor a 30 segundos.</p>	

Gestión de material defectuoso:

<p>Actor Principal: Supervisor de línea.</p>
<p>Personal involucrado e intereses:</p> <ul style="list-style-type: none"> • Supervisor: Desea revisar un listado de los defectos ocurridos en su línea de producción de la cual es responsable así como los defectos que fueron asignados por otra líneas, y ver

<p>la cantidad de material afectado para cada defecto en particular. Aunado a esta lista, desea ver las características de cada uno de los elementos afectados en su línea de producción.</p> <ul style="list-style-type: none"> • Gerente: Revisión general de defectos ocurridos en la línea a su cargo. 	
<p>Poscondiciones: Visualización de defectos ocurridos en la línea de producción.</p>	
<p>Escenario principal de éxito:</p>	
<ul style="list-style-type: none"> • Usuario accede al sistema 	<ul style="list-style-type: none"> • Sistema toma los datos correspondientes una de las líneas de la gerencia consultada, el turno actual y la fecha del día. • Se visualiza los datos de los defectos encontrados en el material producido en la correspondiente línea.
<ul style="list-style-type: none"> • El usuario analiza los datos visualizados. 	
<p>Escenarios o flujos alternativos:</p>	
<ul style="list-style-type: none"> • El actor desea visualizar datos correspondientes a otras líneas. • El actor selecciona la línea de producción que desea consultar. • El usuario escoge la fecha de consulta deseada. • El usuario escoge el turno de producción deseado con respecto a la fecha actual. 	<ul style="list-style-type: none"> • Visualización de los datos correspondientes a la selección

	realizada.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Gestión de equipos defectuosos:

Actor Principal: Supervisor de línea.	
Personal involucrado e intereses: <ul style="list-style-type: none"> Supervisor: Desea revisar un listado de los equipos defectuosos, o equipos que deben ser cambiados cuyo estado es defectuoso. 	
Poscondiciones: Visualización del equipos catalogados como en observación, la ubicación del equipo y el estado presentado.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> Usuario accede al sistema. 	<ul style="list-style-type: none"> Sistema toma los datos correspondientes una de las líneas de la gerencia consultada, el turno actual y la fecha del día. Visualización de equipos en observación, la ubicación del equipo y el estado presentado.
<ul style="list-style-type: none"> El usuario analiza los datos visualizados. 	
Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> El actor desea visualizar datos correspondientes a otras líneas. El actor selecciona la línea de 	

<p>producción que desea consultar.</p> <ul style="list-style-type: none"> • El usuario escoge la fecha de consulta deseada. • El usuario escoge el turno de producción deseado con respecto a la fecha actual. 	<ul style="list-style-type: none"> • Visualización de equipos en observación, la ubicación del equipo y el estado presentado.
<p>Requisitos especiales:</p>	
<p>Respuesta del sistema para la visualización de datos no mayor a 30 segundos.</p>	

Gestión de novedades generales:

En la gestión de novedades generales esta incluido la creación de novedades, y la modificación de las mismas, a continuación detallamos los casos de uso que están implícitos en dicha gestión.

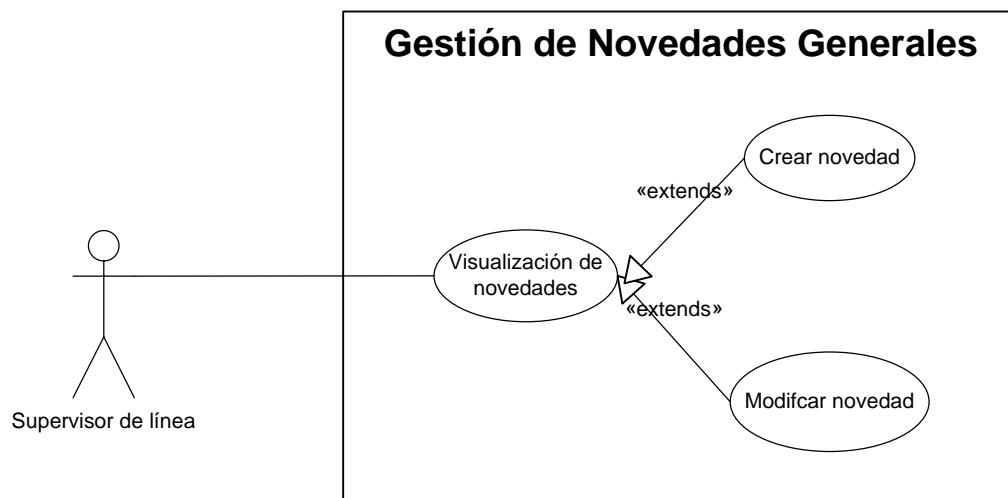


Figura 13: UCS Gestión de novedades generales

Visualización de novedades:

<p>Actor Principal: Supervisor de línea.</p>
<p>Personal involucrado e intereses:</p>

<ul style="list-style-type: none"> Supervisor: Desea revisar un listado las novedades que se han presentado en planta, con respecto a los indicadores de planta, el material hallado como defectuoso o los equipos que han presentado defectos y están en observación. 	
Poscondiciones: Visualización de datos de novedades generales.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> Usuario accede al sistema. 	<ul style="list-style-type: none"> Sistema toma los datos correspondientes una de las líneas de la gerencia consultada, el turno actual y la fecha del día. Visualización de novedades.
Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> El actor desea visualizar datos correspondientes a otras líneas. El actor selecciona la línea de producción que desea consultar. El usuario escoge la fecha de consulta deseada. El usuario escoge el turno de producción deseado con respecto a la fecha actual. 	<ul style="list-style-type: none"> Visualización de novedades generales para los datos consultados.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Crear novedad general:

Actor Principal: Supervisor de línea.	
Personal involucrado e intereses: <ul style="list-style-type: none"> Supervisor: Desea agregar una novedad para la visualización de las mismas. 	
Poscondiciones: Guardar satisfactoriamente los datos correspondientes a las novedades generales.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> Usuario introduce el título de la novedad general. Usuario introduce la descripción de la novedad general. 	<ul style="list-style-type: none"> El sistema guarda los datos correspondientes a la novedad general.
Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> El usuario trata de almacenar una novedad sin incluir título, o descripción. 	<ul style="list-style-type: none"> El sistema genera un mensaje indicando que ha ocurrido un error en la carga de datos.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Modificar novedad general:

Actor Principal: Supervisor de línea.	
Personal involucrado e intereses: <ul style="list-style-type: none"> Supervisor: Desea modificar una novedad para la visualización de las mismas. 	
Poscondiciones: Guardar satisfactoriamente los datos correspondientes a las novedades generales.	

Escenario principal de éxito:	
<ul style="list-style-type: none"> • Usuario introduce el título modificado de la novedad general. • Usuario introduce la descripción modificada de la novedad general. 	<ul style="list-style-type: none"> • El sistema guarda los datos correspondientes a la novedad general.
Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> • El usuario trata de almacenar una novedad sin incluir título, o descripción. 	<ul style="list-style-type: none"> • El sistema genera un mensaje indicando que ha ocurrido un error en la carga de datos.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Generación de reportes:

Actor Principal: Gerente de planta.
Personal involucrado e intereses: <ul style="list-style-type: none"> • Gerente: Desea visualizar el estado de planta, analizando el estado de los indicadores de la misma, el material que ha presentado defectos, y los equipos que presentan defectos en planta.
Poscondiciones: Visualización de datos principales que le permiten al usuario tener una idea general del estado de planta.
Escenario principal de éxito:

<ul style="list-style-type: none"> • Usuario accede al sistema. • El usuario analiza los datos visualizados. 	<ul style="list-style-type: none"> • Sistema toma los datos correspondientes una de las líneas de la gerencia consultada, la fecha del día. • Visualización de indicadores de planta, material defectuoso y equipos defectuosos.
<p>Escenarios o flujos alternativos:</p>	
<ul style="list-style-type: none"> • El actor desea visualizar datos correspondientes a otras líneas. • El actor selecciona la línea de producción que desea consultar. • El usuario escoge la fecha de consulta deseada. • El usuario escoge el turno de producción deseado con respecto a la fecha actual. 	<ul style="list-style-type: none"> • Visualización de indicadores de planta, material defectuoso y equipos defectuosos para los datos elegidos.
<p>Requisitos especiales:</p>	
<p>Respuesta del sistema para la visualización de datos no mayor a 30 segundos.</p>	

Gestión de novedades jerárquicas:

Actor Principal: Gerente de planta.	
Personal involucrado e intereses:	
<ul style="list-style-type: none"> Gerente: Tiene la finalidad de almacenar novedades generales, que deberán aparecer en la visualización de novedades generales durante un periodo de tiempo indicado por el gerente. 	
Poscondiciones: Gestión satisfactoria de novedades generales.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> Usuario accede al sistema, escoge un rango de fechas de vigencia de la novedad. Escoge el título, descripción y lugares del sistema donde debe verse reflejada la novedad escogida. Y las líneas de producción a las cuales es aplicable tal novedad. 	<ul style="list-style-type: none"> El sistema almacena los datos respectivos.
Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> El usuario escoge rango de fechas inadecuadas. El usuario no asocia la novedad general, a alguna línea o sección del sistema. El usuario no coloca datos de la novedad. 	<ul style="list-style-type: none"> El sistema genera un mensaje de error indicando al usuario el campo que debe corregir.

Requisitos especiales:

Respuesta del sistema para la visualización de datos no mayor a 30 segundos.
--

Especificaciones complementarias:

Para el sistema de gestión de defectos, y el libro de novedades hay ciertas especificaciones que no pudieron ser capturadas mediante los casos de uso del sistema, así que en esta área se tomará en cuenta tales requisitos, y serán expuestos para el diseño del sistema.

Seguridad:

El sistema debe detectar el rol del usuario para darle el acceso necesario al manejo de información que le este permitido.

Facilidad de uso:

El sistema debe ofrecer un ambiente amigable mediante el cual el usuario pueda interactuar. De acuerdo al hardware que se manipulara, contaremos con monitores de 17' con una resolución de 1024*768, por lo tanto la disposición de la información debe estar adecuadamente distribuida para que sea legible y cumpla con los estándares de la empresa.

Aunado a ello, el sistema debe estar diseñado para responder rápidamente a los requerimientos del usuario, se exige un buen y rápido procesamiento de la información.

Fiabilidad:

La información contenida en los reportes debe ser totalmente veraz ya que la misma constituye parte primordial en el ciclo de gestión de calidad, desde el punto de vista de producción y desde el punto de vista de defectos de los productos procesados.

Rendimiento:

Se desea conseguir respuestas del sistema por debajo de los 30 segundos, con la finalidad de que la información sea manipulada rápidamente, por ello el rendimiento del sistema debe cumplir con

este requisito que fue incluido en los casos de uso, y constituye un valor importante en el desempeño del mismo.

Implementación:

De acuerdo con el software manejado por la empresa, el sistema deberá implantarse con tecnología .Net, y uso de bases de datos con SqlServer 2000.

Glosario:

Término	Definición e Información	Alias
Bobina	Acero procesado como producto plano y entregado en rollos.	
Bobina defectuosa	Bobina a la cual se le detecta un defecto a la hora de su producción.	
Defecto	Característica detectada en las bobinas producidas que incumplen los requerimientos de calidad. Estos defectos son manejados a través de una codificación previa.	C.D
Línea de Laminación	Maquinaria encargada de la producción y ajustes de las bobinas.	Línea
Línea Origen	Línea causante de aparición del defecto.	L.O
Línea de retención.	Línea en la cual se observa el material defectuoso.	
Características cualitativas	Datos característicos de las bobinas defectuosas.	
Cuadrilla	Grupo de trabajo encargado de las líneas de laminación.	
Turno	Horarios de trabajo establecidos por la empresa.	
Origen	Es el origen del defecto detectado en la bobina defectuosa. Puede ser propio, derivado o asignado.	
Estado de Calidad	Característica cualitativa del material la cual es asignada en el momento de su producción.	E.C
Aplicación	Se refiere a la aplicación que se le da el personal de Calidad de la empresa, a las bobinas que resultan defectuosas.	

Indicadores	Son parámetros que permiten al área de industrial determinar el estado de la producción en planta.	
Productividad Efectiva:	Es un indicador de la producción de planta, el cual muestra la relación entre las toneladas de material producido, con relación al tiempo efectivo en el cual se produjo tal tonelaje.	
Efectividad:	Es un indicador de la producción, el cual muestra la relación entre el tiempo efectivo (tiempo durante el cual la línea trabajó), y el tiempo disponible para la producción, excluyendo el tiempo de demoras ocurridas en las líneas de producción.	
Puesta a mil:	Indicador del tonelaje de descarte ocurrido debido a la presencia de defectos en las bobinas procesadas.	
Calidad:	En el área del proceso productivo este termino abarca una amplia gama de análisis a nivel físico, químico de los materiales usados en la producción. Sin embargo en el contexto al cual le referimos, el término calidad esta acotado a los defectos visibles presentados por las bobinas que se producen en las líneas de laminado.	

Capítulo IV

Fase de elaboración

En esta fase del proyecto se pretende construir el núcleo central del sistema de calidad, resolver los elementos que se consideren de alto riesgo, y definir la mayoría de los requisitos del mismo. Los objetivos principales que debemos conseguir comprenden la recopilación de los requisitos que aún queden pendientes, establecer una arquitectura sólida para guiar el trabajo durante la fase de construcción y transición. Se debe conseguir un esquema de la descripción de una arquitectura candidata, que perfila las vistas de los modelos de casos de uso, análisis, diseño e implementación. El desarrollo de esta parte del proyecto, se realizará en tres iteraciones, las cuales han sido planificadas y detalladas para el cumplimiento de los objetivos en la fase de elaboración.

Iteración 1: se realizará un análisis más detallado de los casos de uso del sistema, así como una aclaración de los eventos de entrada y salida del sistema las cuales serán presentadas a través de diagramas de secuencia UML. Aunado a ello, se dará inicio al modelo de dominio, haciendo uso diagramas de clases, posteriormente se procederá a realizar diagramas de elaboración e iteración para la obtención de un modelo de diseño del sistema.

Iteración 2 e iteración 3: en esta parte se realizarán nuevos talleres de requisitos con la finalidad de completar los requerimientos del sistema, y mejorar los encontrados en la fase de inicio, como consecuencia de ello, se deberá actualizar el modelo de casos de uso, el modelo de dominio y el modelo de diseño del sistema.

Es importante acotar que al realizar los modelos anteriormente mencionados para la arquitectura del sistema, se verá afectado nuestro glosario de términos el cual será actualizado, para que el lector comprenda los términos que usaremos en el desarrollo.

Iteración I:

Modelo de Casos de uso.

Como describimos anteriormente, en esta parte refinaremos el modelo de casos de uso previamente encontrado en la fase de inicio a través del planteamiento de diagramas de secuencia (DSS), los cuales son artefactos que muestran los eventos de entrada y salida relacionados con los eventos que surgen en los casos de uso del sistema. Siguiendo la recomendación de Craig Larman³, se realizará un DSS para el escenario de éxito y los escenarios alternativos de los casos de uso previamente definidos en la fase de inicio.

Diagramas de secuencia para el sistema de gestión de defectos

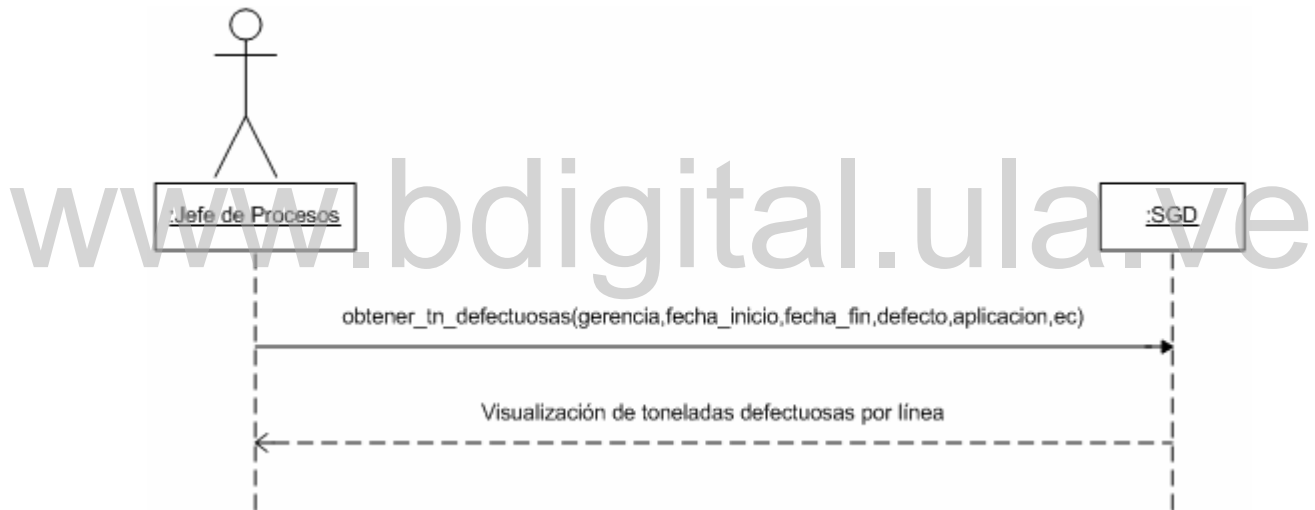


Figura 14: DSS Revisión de defectos

DSS Revisión de defectos: este diagrama de secuencia corresponde con el caso de uso, *Revisión de toneladas defectuosas por gerencia*, el cual representa el escenario de éxito del mismo. Se observa que el usuario introduce la fecha de inicio y fin para la consulta, el defecto a revisar, aplicación y estados de calidad del material defectuoso, y el sistema retorna como respuesta una tabla que contiene la contabilización del peso en toneladas de los defectos presentados en cada línea de las gerencias.

³ UML Y PATRONES. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Autor: Craig Larman.

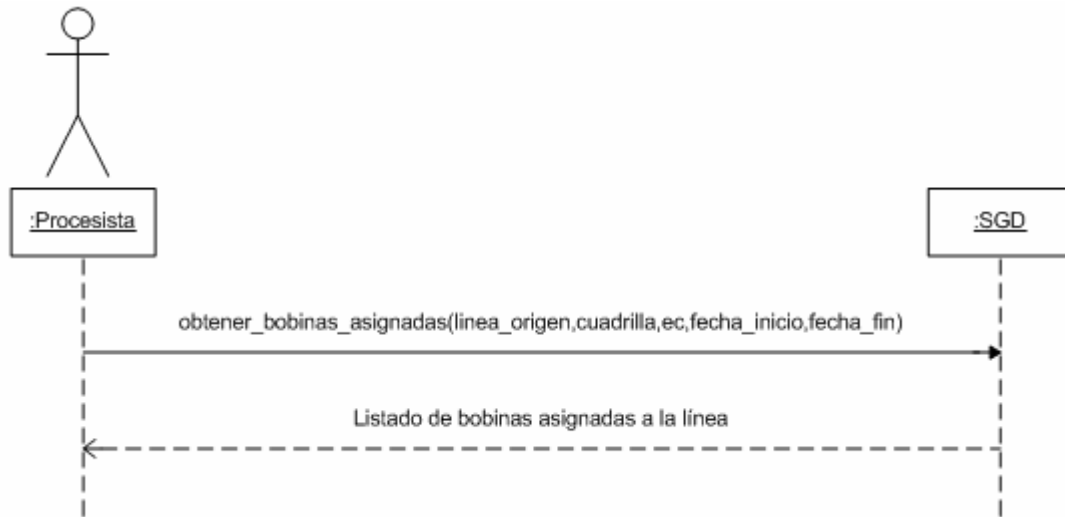


Figura 15: DSS Visualización de defectos asignados.

DSS Visualización de defectos asignados: este diagrama de secuencia corresponde con el caso de uso, *Visualización de defectos asignados*, el cual representa el escenario de éxito de del mismo. Se observa que el usuario introduce la línea origen, la cuadrilla, estado de calidad del material defectuoso, la fecha de inicio y fin para la consulta, entonces el sistema retorna como respuesta un listado de las bobinas defectuosas que han sido asignadas a la línea de la cual es responsable el procesista.

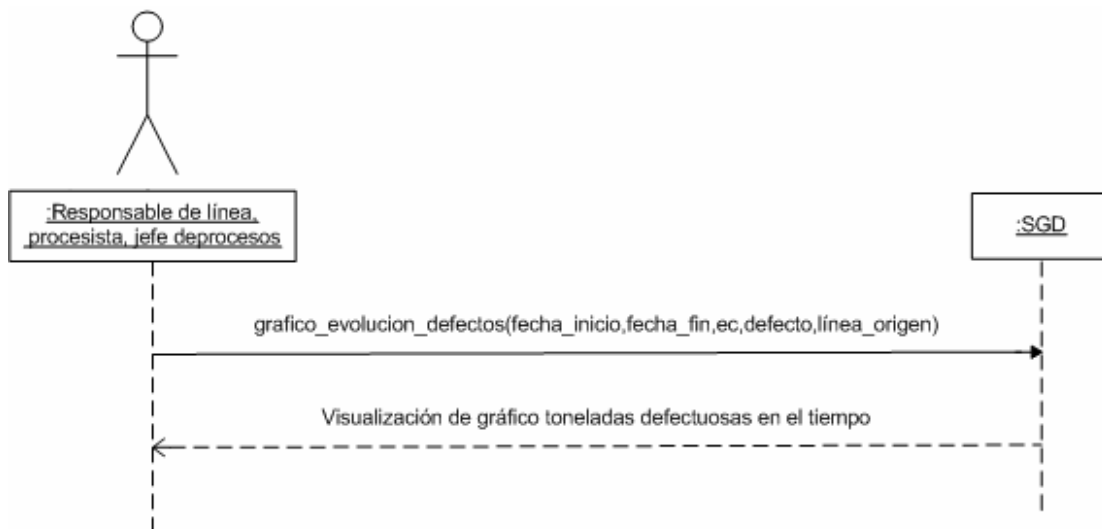


Figura 16: DSS Evolución de defectos en el tiempo

DSS Evolución de defectos en el tiempo: este diagrama representa el escenario de éxito del caso de uso *Evolución de defectos en el tiempo*, el cual permitirá a los usuarios analizar la progresión del defecto en un intervalo de tiempo deseado. Para ello el actor, escoge el intervalo de tiempo, el estado de calidad de las bobinas que presentaron el defecto, el defecto a analizar, y la línea origen en donde se presenta tal defecto. La respuesta del sistema se corresponde a la visualización de un gráfico de las toneladas de material que presento el defecto diariamente para el intervalo de tiempo elegido.



Figura 17: DSS Evolución de defectos ante acciones correctivas asignadas.

DSS Evolución de defectos ante acciones correctivas asignadas: este diagrama representa el escenario de éxito del caso de uso *Evolución de defectos ante acciones correctivas asignadas*, el cual permitirá al procesista analizar la progresión del defecto en un intervalo de tiempo deseado en comparación con las acciones aplicadas por los procesistas en el periodo de tiempo asignado por el responsable de defectos. Para ello el actor, escoge el intervalo de tiempo, el estado de calidad de las bobinas que presentaron el defecto, el defecto a analizar, y la línea origen en donde se presenta tal defecto. La respuesta del sistema se corresponde a la visualización de un gráfico de las toneladas de material que presento el defecto diariamente para el intervalo de tiempo elegido, así como el intervalo de aplicación de las acciones asignadas.

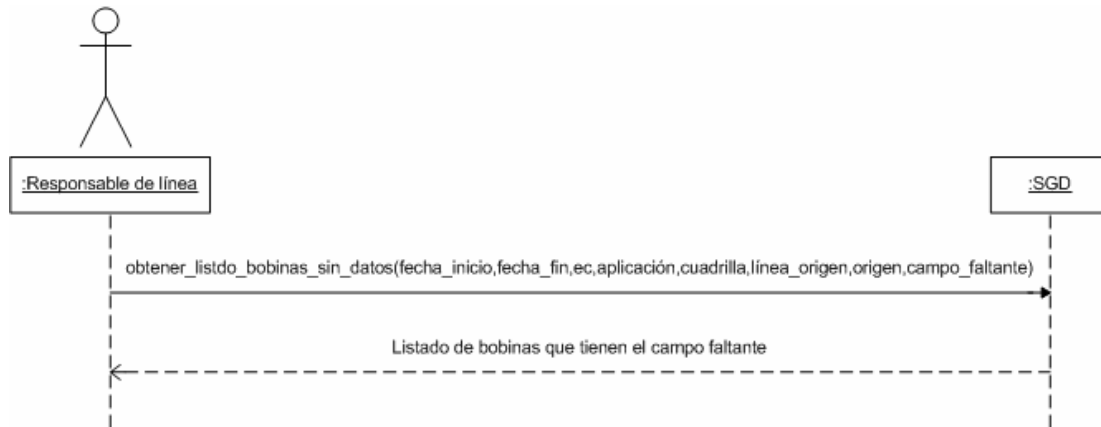


Figura 18: DSS Verificación de carga de causas, acciones por bobina defectuosa.

DSS Verificación de carga de causas, acciones por bobina defectuosa: el diagrama anterior corresponde al caso de uso *Verificación de carga de causas, acciones por bobina defectuosa*, como vemos el usuario especifica la fecha inicio, fecha fin, estado de calidad, aplicación, línea origen, origen de las bobinas, y el campo faltante, es decir, puede especificar ver bobinas a las cuales no se cargo en planta las causas del defecto, o bobinas a las cuales no se cargo las acciones correctivas o a las acciones preventivas para que no ocurra mas tal defecto. Luego el sistema responderá trayendo el listado de bobinas que cumplan con las características especificadas por el usuario.

Diagramas de secuencia para el sistema libro de novedades:

Para el libro de novedades, es interesante hacer notar que los casos de uso previamente definidos en la fase de inicio poseen casos de uso que se derivan de acuerdo a la decisión del actor. Como esto afecta el modelado de los diagramas de secuencia, trataremos de englobar o detallar los mismos siguiendo la misma lógica que se planteo en la fase de inicio.

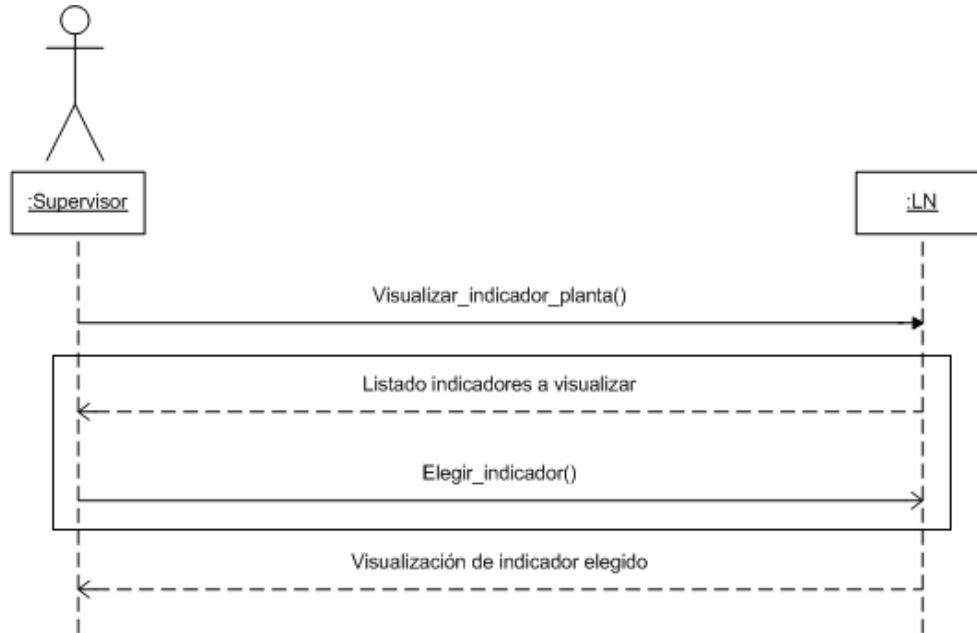


Figura 19: DSS Gestión de indicadores de planta

DSS Gestión de indicadores de planta: Este diagrama de secuencia está relacionado con el caso de uso *Gestión de indicadores de planta*, en esta parte el supervisor solo tiene la responsabilidad de elegir el indicador de planta que desea analizar, y de acuerdo a su elección se mostrarán los datos adecuados.

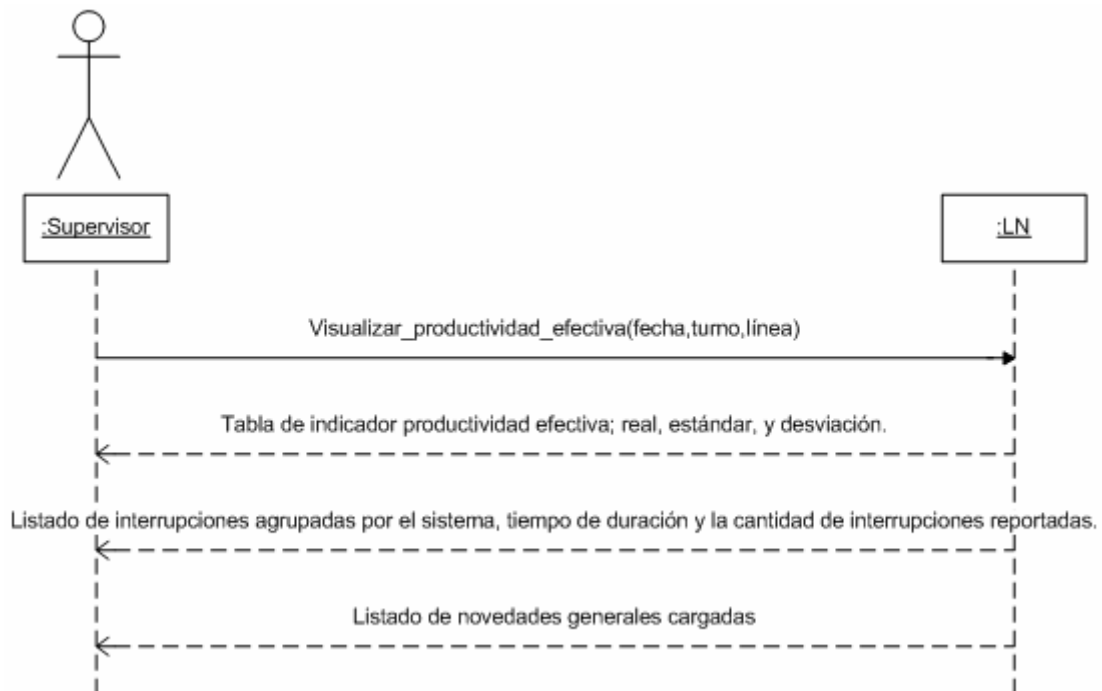


Figura 20: DSS Visualizar productividad efectiva

DSS Visualizar productividad efectiva: el diagrama de la figura, representa el escenario de éxito del caso de uso *Visualizar productividad efectiva*, el cual es consecuencia de la elección previa mostrada en el DSS Gestión de Indicadores. Al acceder a esta área el sistema tomará la fecha y turno del día, así como la línea seleccionada, luego se generará los datos correspondientes a la productividad efectiva real, estándar y la desviación entre ambas. Como la productividad efectiva es calculada a partir de las interrupciones ocurridas en planta, el sistema presenta los datos de las interrupciones detectadas, las cuales son agrupadas por el sistema donde ocurrió la interrupción. Esta parte le permite al supervisor detectar que área de la planta esta produciendo un bajo rendimiento. Aunado a tal detalle, se muestra las novedades que se han cargado en planta.



Figura 21: DSS Visualizar efectividad

DSS Visualizar efectividad: el diagrama de la figura, representa el escenario de éxito del caso de uso *Visualizar efectividad*. Al acceder a esta área el sistema tomará la fecha y turno del día, así como la línea seleccionada, luego se generará los datos correspondientes a la efectividad real, estándar y la desviación entre ambas. Como la efectividad es calculada a partir del tiempo efectivo de trabajo, y el mismo de ve afectado por las demoras ocurridas en planta, el sistema presenta los datos de las interrupciones detectadas, las cuales son agrupadas por

Tipo/Zona/Sistema/Subsistema donde ocurrió la demora. En esta área también se visualizan las novedades generales que han sido cargadas previamente, para información de los usuarios del sistema.

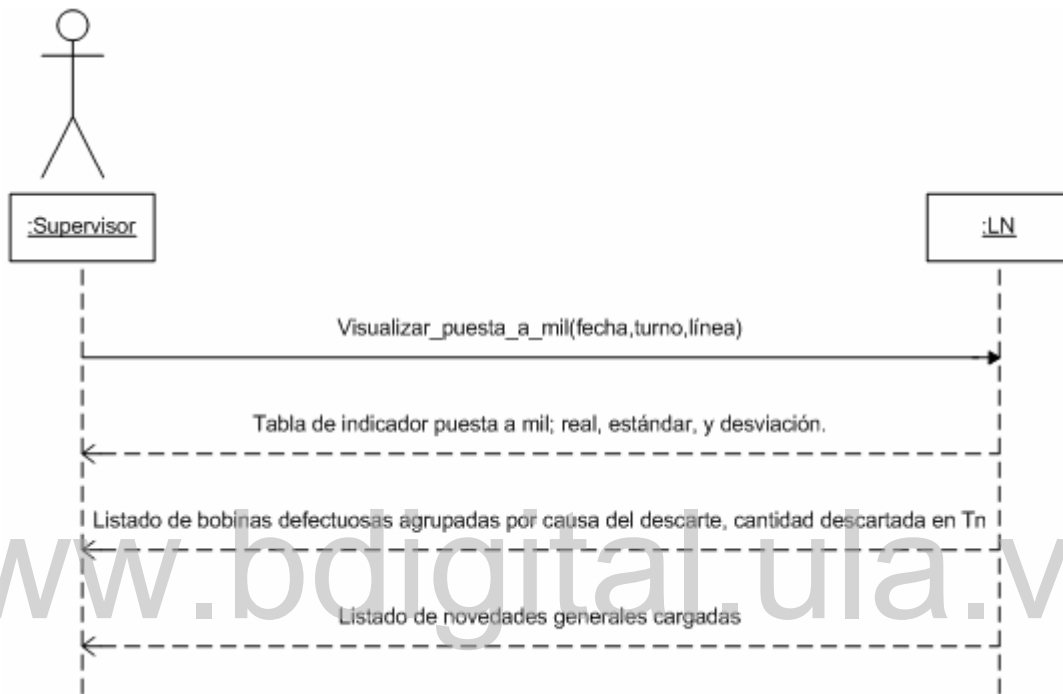


Figura 22: DSS Visualizar puesta a mil

DSS Visualizar puesta a mil: Este DSS está relacionado con el caso de uso Visualizar puesta a mil, con el cual el supervisor desea ver la cantidad de toneladas perdidas debido a la detección de defectos en las mismas. El sistema toma la fecha y turno del día de consulta, y la línea cargada, el sistema procede a generar la tabla de indicadores de planta de puesta a mil, que contiene la PAM real, estándar y la desviación respectiva. El sistema genera un listado correspondiente a todas las bobinas que han sufrido descarte de material debido a la presencia de defectos; tal listado es agrupado por la causa del descarte, y se mostrará la cantidad de toneladas descartadas por defecto presentado. Siguiendo la misma línea de presentación se visualizará el listado correspondiente a las novedades generales cargadas para ser observadas en esta área.

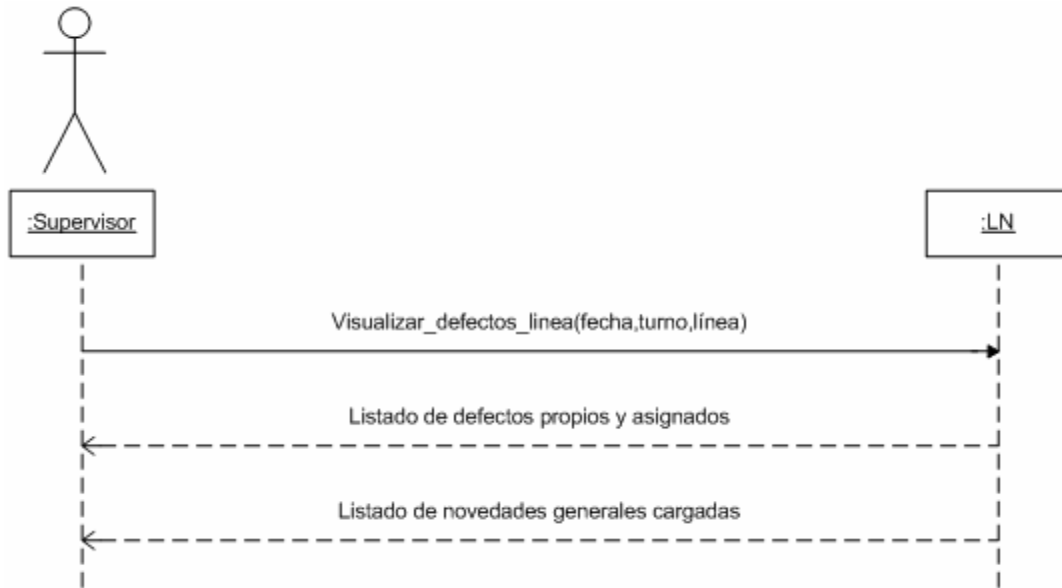


Figura 23: DSS Gestión de material defectuoso.

DSS Gestión de material defectuoso: Este diagrama está relacionado con el caso de uso *Gestión de material defectuoso*, en el cual, el sistema toma los datos correspondientes a la fecha y turno del día de consulta, así como la línea de producción correspondiente; en consecuencia con tal información se genera un listado con el contenido de los defectos propios, y los defectos asignados a tal línea de producción. Paralelamente se genera el listado de novedades generales correspondientes a esta área.



Figura 24: DSS Gestión de equipos defectuosos

DSS Gestión de equipos defectuosos: el diagrama anterior, esta relacionado con el caso de uso *Gestión de equipos defectuosos*, en el cual al acceder al sistema se genera el listado de equipos defectuosos encontrados en una línea de planta, para la fecha y turno del día en el que se accede a ubicar tal información. Para dicha selección se presenta el listado de novedades generales referentes a esta área.

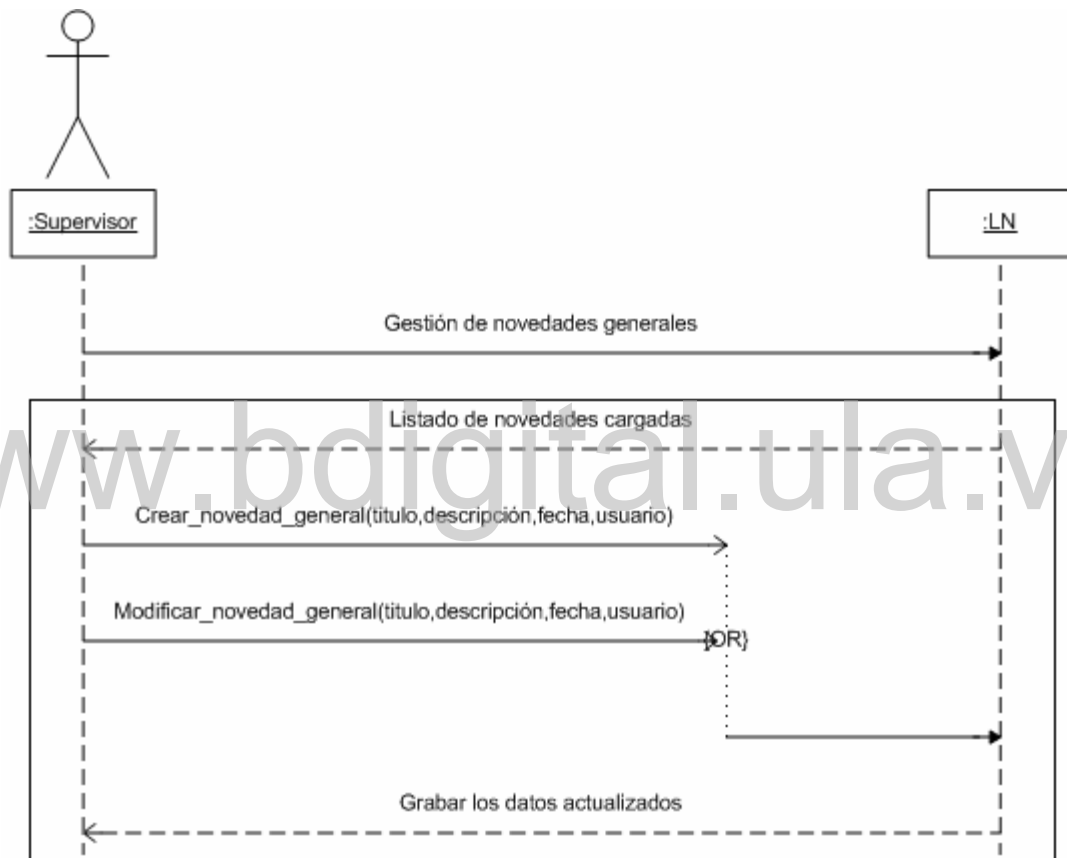


Figura 25: DSS Gestión de novedades generales.

DSS Gestión de novedades generales: este diagrama de secuencia está relacionado con el caso de uso *Gestión de novedades generales*, el cual contiene varios pasos de acuerdo a lo que el usuario desee realizar a este respecto. Como vemos el actor puede observar las novedades generales creadas por los diferentes usuarios del sistema; el actor podrá crear una nueva novedad o modificar una novedad previamente cargada por el. Para ambos casos el sistema requiere el título de la novedad, la descripción de la novedad, la fecha de inserción o modificación, así como el usuario del sistema.

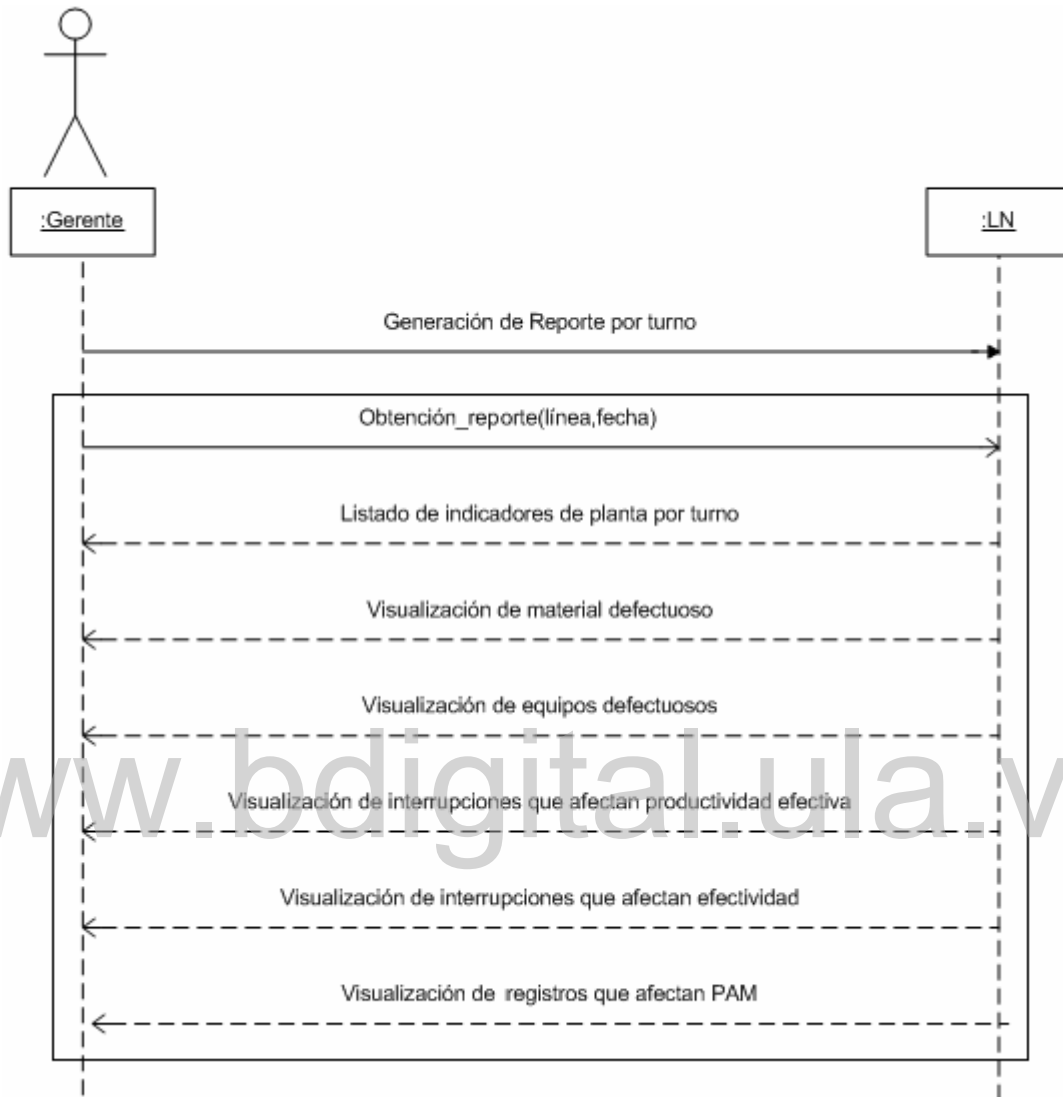


Figura 26: DSS Generación de Reportes

DSS Generación de Reportes: este diagrama parte del caso de uso *Generación de Reportes*, con el cual el gerente accede al sistema con la finalidad de ver un resumen de la información primordial de planta. Este reporte recibe como parámetros de entrada la línea y la fecha de consulta. Con tales datos el sistema genera una tabla resumen de los indicadores de la planta, los cuales vienen dado por turno y por cuadrilla. Paralelamente se visualiza un listado del material que ha contenido defectos en producción, los equipos que poseen características defectuosas; las interrupciones que afectan tanto la productividad efectiva como la efectividad y las bobinas que han sufrido descarte por motivo de presencia de defectos.

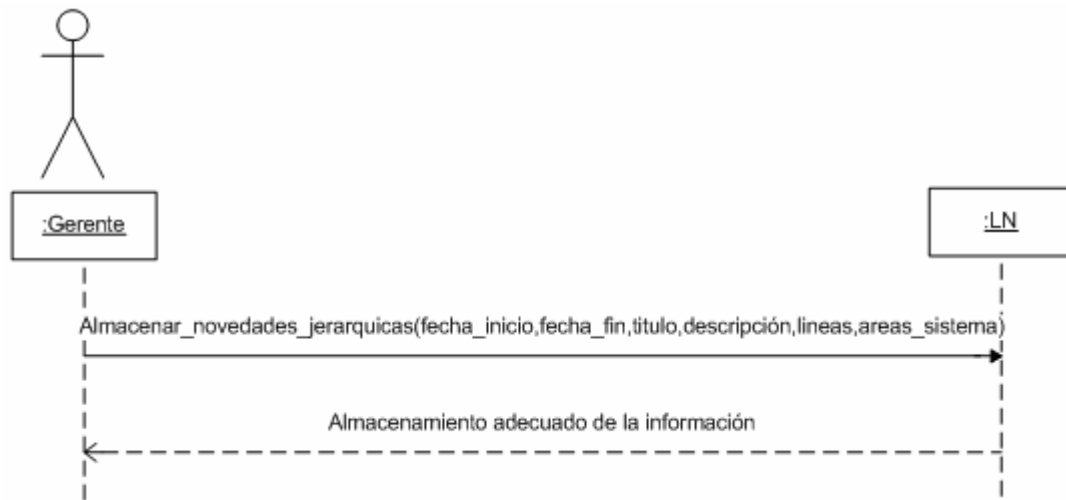


Figura 27: DSS Gestión de novedades jerárquicas

DSS Gestión de novedades jerárquicas: Como vemos el gerente de planta es el encargado de estimular el sistema, enviando el periodo de vigencia de la novedad, el título, la descripción de la misma, las líneas de producción donde aparecerá, así como las áreas del sistema donde serán visualizadas en el listado de novedades generales presentado en las mismas. Este diagrama se corresponde al escenario de éxito del caso de uso *Gestión de novedades jerárquicas*.

Modelo de dominio

El modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los eventos que suceden en el entorno en el que trabaja el sistema. Para la obtención de las clases se usan las especificaciones de requisitos.

Hay tres formas típicas de encontrar las clases del sistema, estas son:

- Objetos del negocio que representan cosas que se manipulan en el mismo.
- Objetos del mundo real y conceptos de los que el sistema debe hacer un seguimiento.
- Sucesos que ocurrirán o han ocurrido.

El modelo del dominio se describe mediante diagramas de clases UML.

Para la obtención del modelo de dominio de esta iteración es necesaria la identificación de las clases conceptuales del sistema, lo cual parte de las frases usadas en la descripción de los casos de uso de formato completo que hemos desarrollado previamente.

Del sistema se extrajeron las siguientes clases conceptuales para el SGD:

- Bobina defectuosa.
- Línea origen.
- Estado de calidad.
- Defecto.
- Aplicación.
- Responsable.
- Perfiles.

Para el LN tenemos las siguientes clases conceptuales:

- Novedades generales.
- Novedades jerárquicas.
- Secciones.
- Línea.
- Material defectuoso (Bobinas).
- Equipos.
- Interrupciones.
- Responsable.
- Perfiles.

Este listado nos genera el primer diagrama de dominio del sistema:

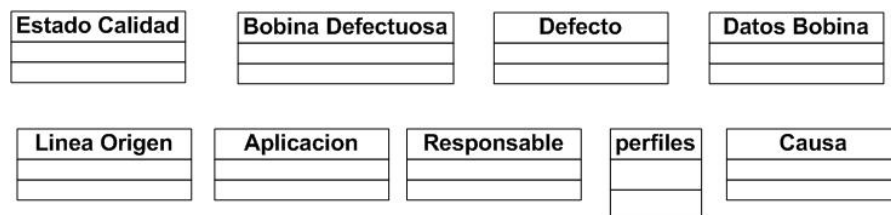


Figura 28: Modelo de dominio de la iteración 1 para el SGD

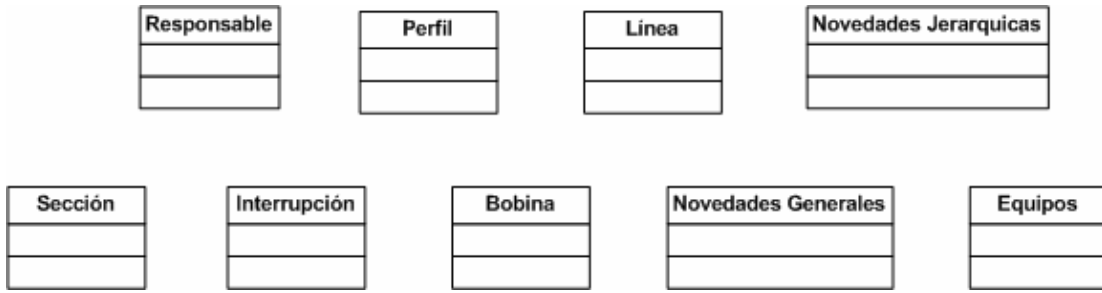


Figura 29: Modelo de Dominio, iteración 1 para el LN.

Estas clases conceptuales están relacionadas entre si por asociaciones de instancias, a continuación presentamos el diagrama de dominio con las mismas.

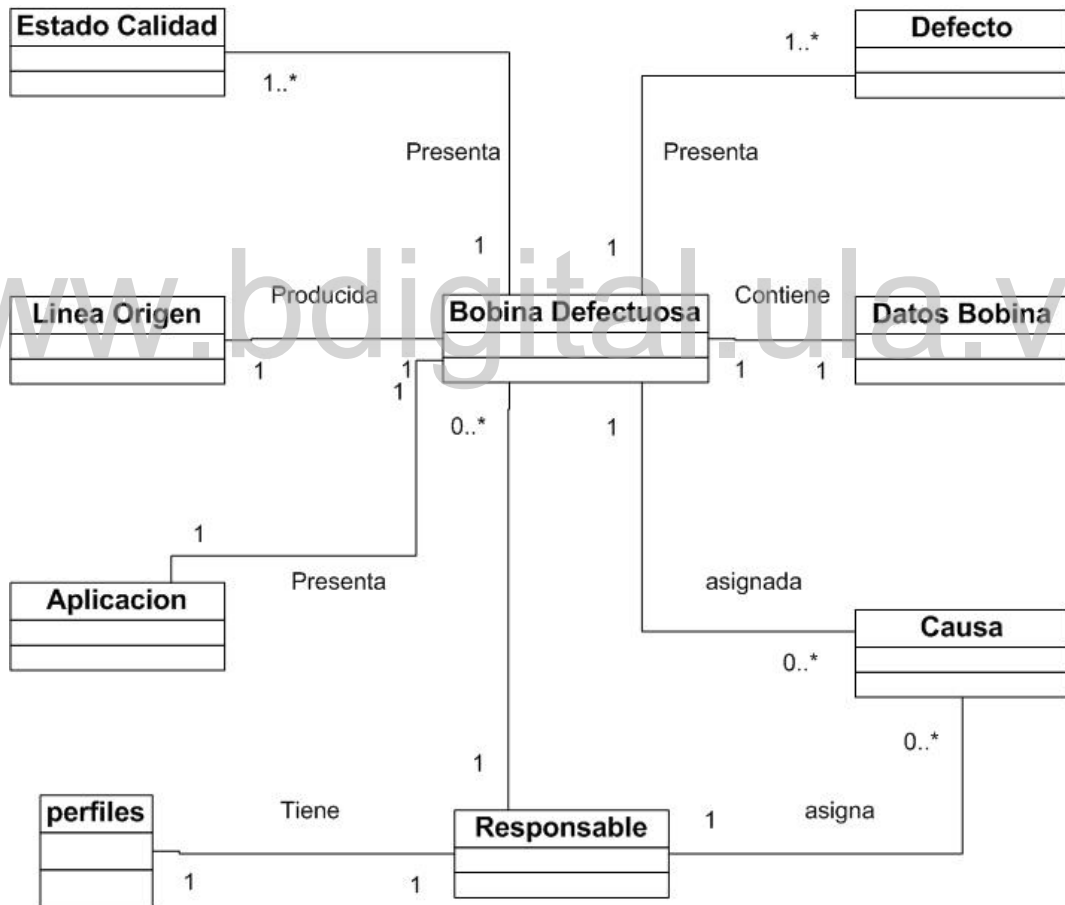


Figura 30: Modelo de Dominio, iteración 1 para el SGD

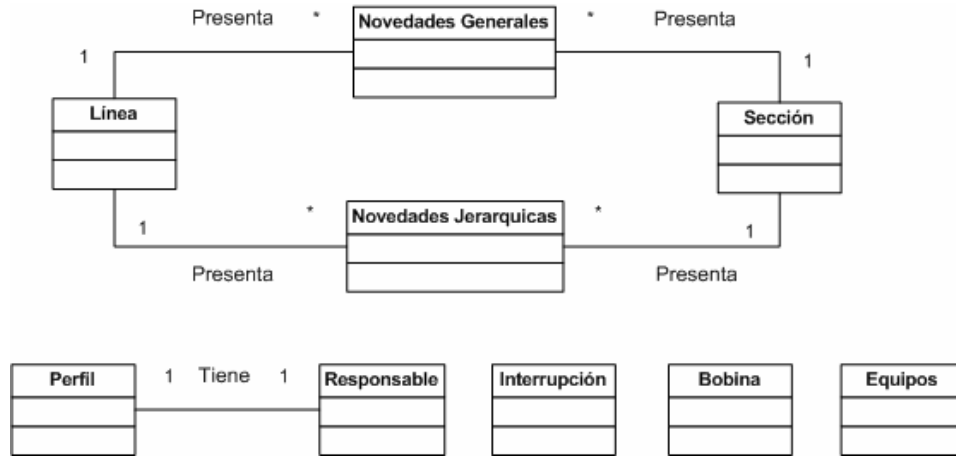


Figura 31: Modelo de Dominio para el LN

Luego de tener el modelo de dominio con sus respectivas asociaciones, podemos definir los atributos que consideramos que las clases del sistema deben poseer para satisfacer los requisitos y objetivos trazados.

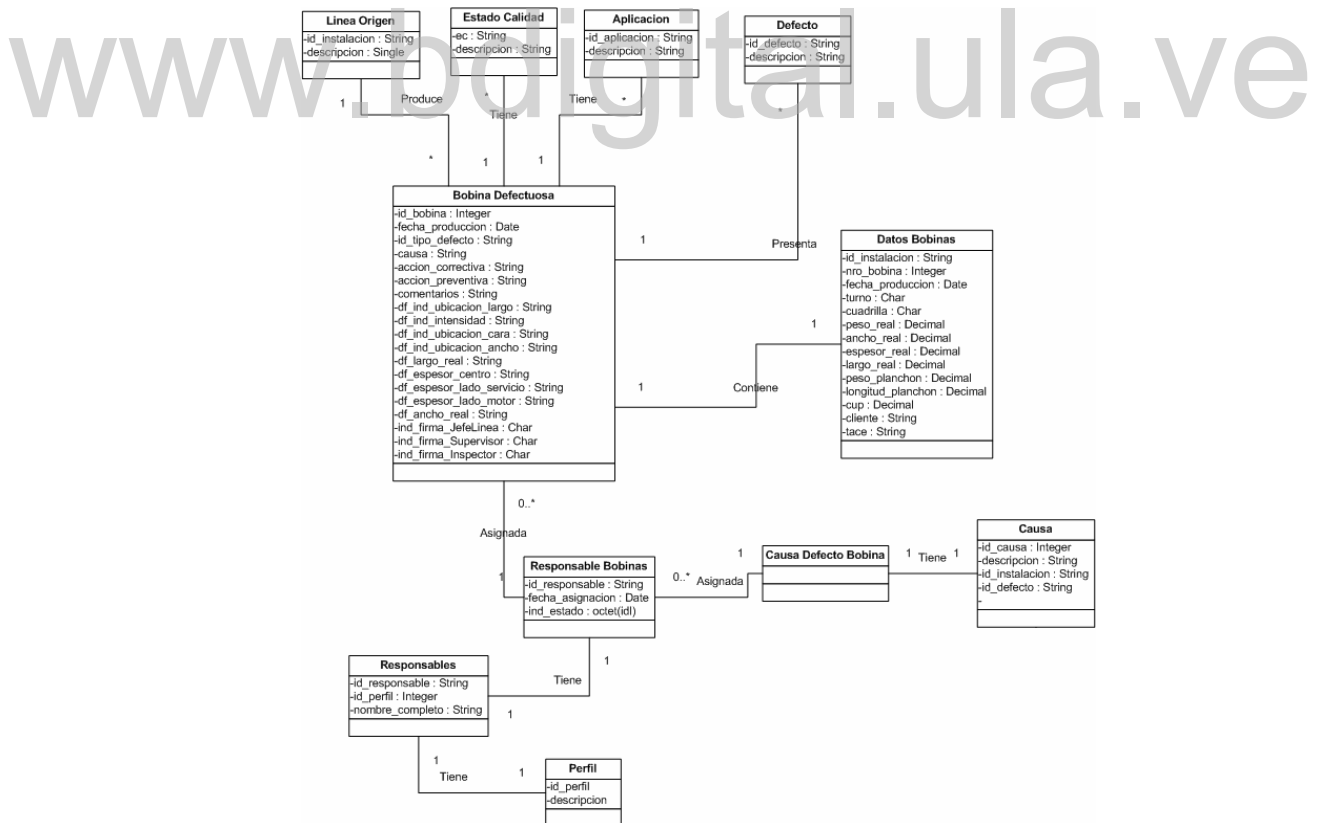


Figura 32: Clases Conceptuales del SGD

Podemos notar en el diagrama de clases, que surgen varias clases las cuales debemos describir detalladamente:

Bobina defectuosa: Objeto físico producido en las líneas de laminación de la empresa.

Datos bobinas: Representa los datos generales propios de las bobinas producidas en las líneas de laminación.

Estado de calidad: Atributo presentado por las bobinas producidas, tal atributo es asignado a las bobinas.

Aplicación: atributo asignado a la bobina, de acuerdo al defecto presentado.

Defecto: característica indeseable presentada por las bobinas, que puede aparecer en cualquier etapa de su producción.

Perfil: es el identificador del cargo ocupado de un usuario dentro del sistema.

Responsable: Personas autorizadas, que de acuerdo al perfil de la empresa poseen ciertos privilegios en el uso del sistema.

Causa: Representa el evento que da origen a la aparición de un defecto en las bobinas.

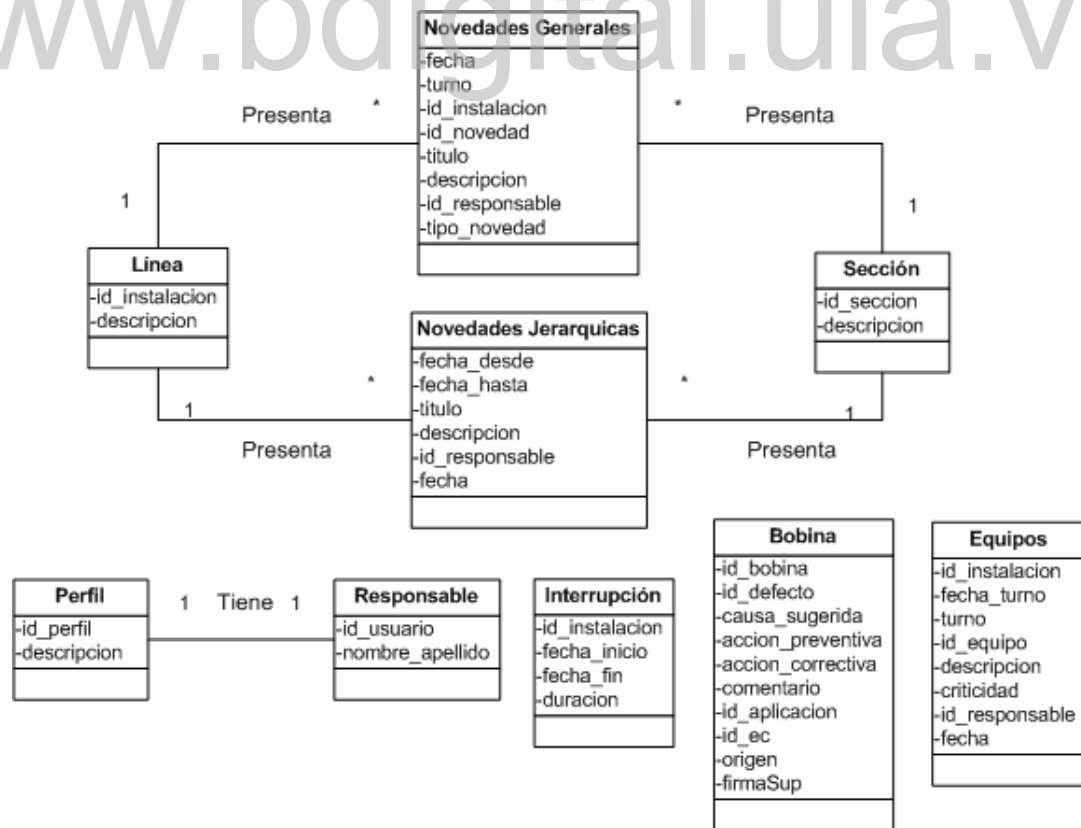


Figura 33: Clases Conceptuales del LN.

Del diagrama de clases del Libro de novedades obtenemos las clases que debemos definir para este módulo:

Interrupción: Intervalo de tiempo durante el cual una línea de producción está parada. Este objeto es el principal factor que afecta los indicadores de planta, relacionados con la efectividad de la línea, así como la productividad de la misma.

Equipos: Mecanismo usado para la realización de un trabajo, el cual posee un periodo de vida útil, luego del cual debe ser reemplazado.

Novedad General: es un comentario que debe mantenerse durante un turno de producción, para la línea que se ha especificado.

Novedad Jerárquica: es un comentario proveniente de la gerencia, que tiene un periodo de vigencia, el cual debe estar habilitado para el tiempo determinado por ellos, para las secciones y líneas específicas.

Estas clases se agregan a las definidas en el SGD, para nuestro sistema de calidad.

Modelo del diseño:

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación tienen impacto en el sistema. Este modelo sirve de abstracción de la implementación del sistema, y es una de las actividades fundamentales en la implementación.

Con la finalidad de obtener un adecuado modelo de diseño, realizaremos diagramas de iteración que nos permitirán ilustrar las iteraciones entre los objetos en un formato de grafo o red. Los diagramas de iteración son una generalización de los diagramas de colaboración y secuencia que presentan de forma más especializada la iteración de mensajes.

DI Escenario Revisión de toneladas defectuosas:

Este diagrama de iteración permite visualizar la forma en que los objetos del sistema actúan para solventar la solicitud del usuario de retornar las toneladas de bobinas con defectos detectadas en las líneas de producción. Cuando el sistema recibe la solicitud “obtener defectos por instalaciones”, el sistema genera la siguiente secuencia de pasos:

1. La colección de *bobinas defectuosas* solicita los datos de la bobina a la instancia *datos bobinas*, bajo el mensaje *obtener_datos_bobina()*
2. La colección de *bobinas defectuosas* solicita *obtener_linea()*, al objeto Línea Origen.
3. La colección de *bobinas defectuosas* solicita *obtener_estado_calidad()*, al objeto Estado de calidad.
4. La colección de *bobinas defectuosas* solicita *obtener_aplicacion()*, al objeto Aplicación.
5. La colección de *bobinas defectuosas* solicita *obtener_defecto()*, al objeto Defecto.

Luego de ello el sistema agrupa por defecto encontrado al listado de bobinas, sumando las toneladas correspondientes a cada bobina que presentan el mismo defecto particular, y se listan por línea de producción del material afectado.

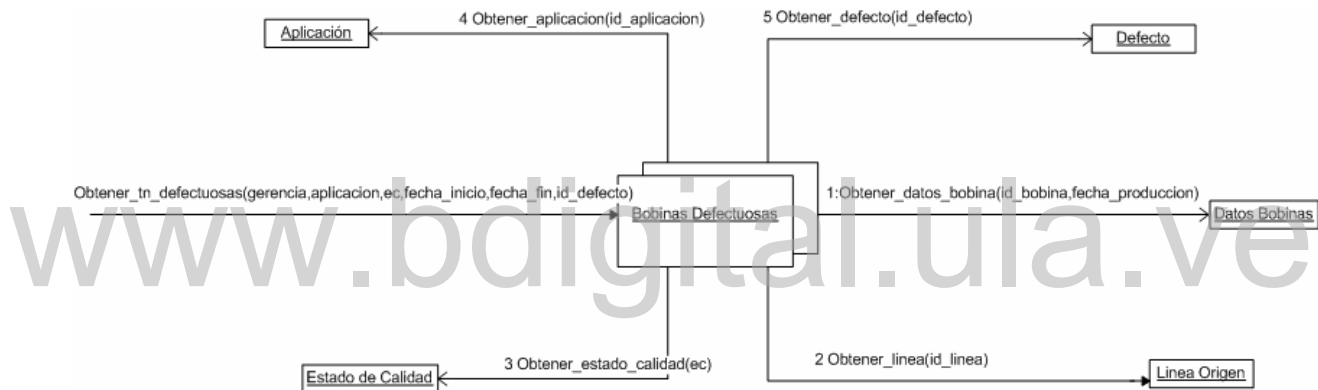


Figura 34: DI Revisión de toneladas defectuosas.

DI Escenario Defectos Asignados:

El diagrama de iteración permite observar como los objetos del sistema SGD colaboran entre si para satisfacer el caso de uso: *Visualización de defectos asignados*, a continuación vemos la secuencia de pasos que el sistema debe realizar:

1. La colección de *bobinas defectuosas* solicita los datos de la bobina a la instancia *datos bobinas*, bajo el mensaje *obtener_datos_bobina()*
2. La colección de *bobinas defectuosas* solicita *obtener_linea()*, al objeto Línea Origen.
3. La colección de *bobinas defectuosas* solicita *obtener_estado_calidad()*, al objeto Estado de calidad.

Luego de tener la información solicitada internamente por el sistema, se debe armar el listado de de bobinas defectuosas correspondientes a la solicitud realizada al sistema.

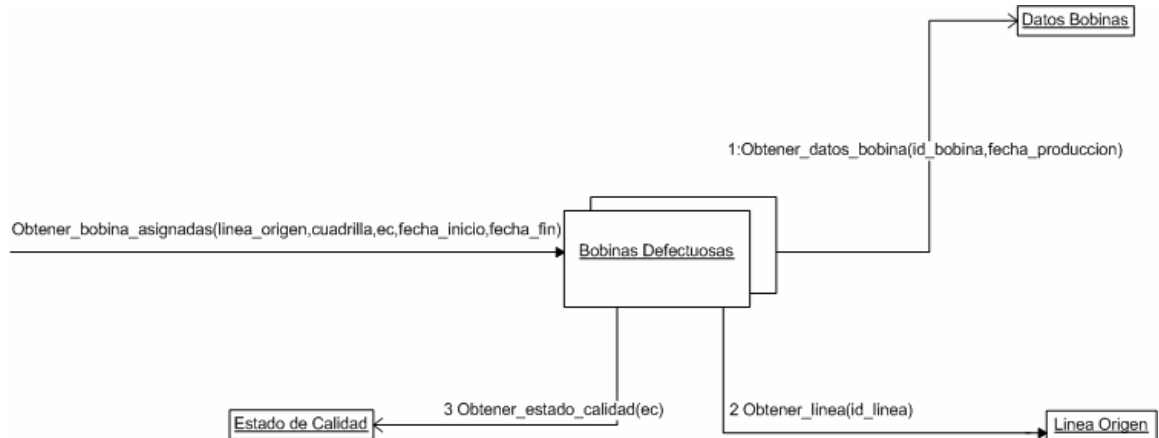


Figura 35: DI Defectos Asignados

DI Escenario Verificación de carga de causas, acciones, por bobina defectuosa

Similarmente a los escenarios anteriormente expuestos podemos notar la interacción entre la colección de bobinas defectuosas, en relación con las clases definidas previamente. Esta respuesta del sistema se genera cuando el usuario entra a revisar aquellas bobinas de planta a las cuales no se les ha cargado datos importantes para la gestión de defectos.

1. La colección de *bobinas defectuosas*, procede a solicitar del objeto *datos bobinas* *Obtener_datos_bobinas(id_bobina, fecha_produccion)*, donde se debe traer de la colección aquellas bobinas que tenga como cualidad ausencia de causas, acciones, de acuerdo a lo que el sistema halla solicitado a la colección.
2. La colección de *bobinas defectuosas* solicita *obtener_linea()*, al objeto Línea Origen.
3. La colección de *bobinas defectuosas* solicita *obtener_estado_calidad()*, al objeto Estado de calidad.
4. La colección de *bobinas defectuosas* solicita *obtener_aplicacion()*, al objeto Aplicación.

Tales solicitudes se realizan por cada bobina que presenta las cualidades especificadas, lo cual dará como resultado un listado de bobinas defectuosas.

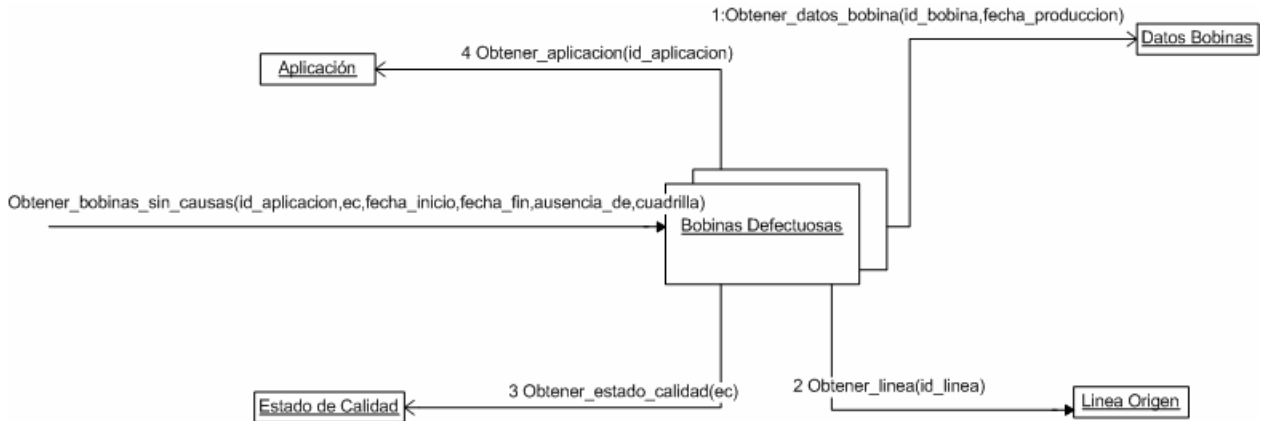


Figura 36: DI Verificación de causas, acciones, por bobina defectuosa

Como observamos para los diagramas de iteración del modulo de reportes del sistema de gestión de defectos se sigue una lógica similar, ya que al mostrar los diferentes reportes los objetos interactúan haciendo solicitudes equivalentes, por ello procederemos a especificar los diagramas de colaboración correspondientes al libro de novedades.

DI Escenario Visualizar productividad efectiva:

En el diagrama de la figura se observa la iteración entre los objetos del modulo del libro de novedades para la obtención de los resultados propuestos en UCS Visualizar productividad efectiva. Al hacer la solicitud al sistema *Obtener_datos_productividad_efectiva* (turno, fecha, línea, gerencia) se generan dos llamadas a la colección de interrupciones:

1. Se hace la llamada *Obtener_cantidad_interrupciones* (turno, fecha, línea, gerencia) hacia una colección de interrupciones que proviene del sistema de interrupciones, de allí se realiza la agrupación de interrupciones presentadas que afectan la productividad efectiva de la línea de producción.
2. Se hace la llamada *Obtener_detalle_interrupciones* (turno, fecha, línea, gerencia), con la cual se obtienen los datos referentes a las interrupciones ocurridas en el turno, fecha y línea elegidos.

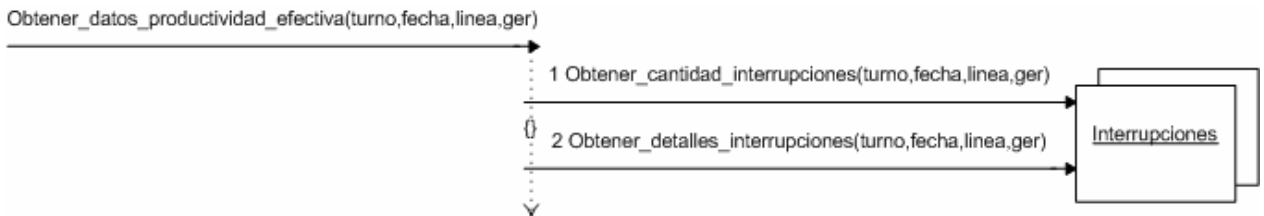


Figura 37: DI Escenario Visualizar productividad efectiva

DI Escenario Visualizar efectividad:

En el diagrama de la figura se observa la iteración entre los objetos del modulo del libro de novedades para la obtención de los resultados propuestos en UCS Visualizar efectividad. Al hacer la solicitud al sistema *Obtener_datos_efectividad* (*turno, fecha, línea, gerencia*) se generan dos llamadas a la colección de interrupciones:

1. Se hace la llamada *Obtener_cantidad_interrupciones* (*turno, fecha, línea, gerencia*) hacia una colección de interrupciones que proviene del sistema de interrupciones, de allí se realiza la agrupación de interrupciones presentadas que afectan la efectividad de la línea de producción.
2. Se hace la llamada *Obtener_detalle_interrupciones* (*turno, fecha, línea, gerencia*), con la cual se obtienen los datos referentes a las interrupciones ocurridas en el turno, fecha y línea elegidos, y corresponden a las interrupciones agrupadas en la primera solicitud.

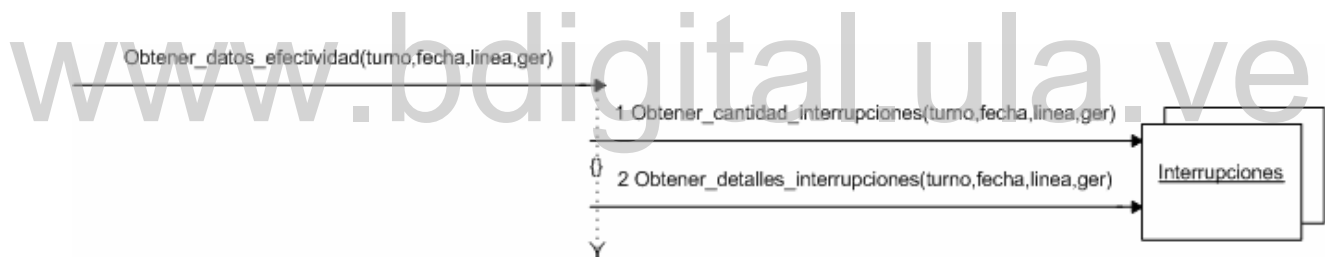


Figura 38: DI Escenario Visualizar efectividad

DI Escenario Visualizar puesta a mil:

En el diagrama de la figura se observa la iteración entre los objetos del modulo del libro de novedades para la obtención de los resultados propuestos en UCS Visualizar puesta a mil. Al hacer la solicitud al sistema *Obtener_pam* (*turno, fecha, línea, gerencia*) se generan dos llamadas a la colección de bobinas pertenecientes al PAM:

1. Se hace la llamada *Obtener_defectos* (*turno, fecha, línea, gerencia*) a la colección de bobinas pertenecientes al sistema puesta a mil, lo cual genera un listado de los defectos causantes de pérdida de material que presentaron las bobinas producidas en el turno, fecha y línea de producción indicada.

- Se hace la llamada *Obtener_bobinas* (*turno, fecha, línea, gerencia*), con la cual se obtienen los datos referentes a las bobinas producidas en el turno, fecha y línea elegidos, y corresponden a los defectos obtenidos en la llamada anterior.

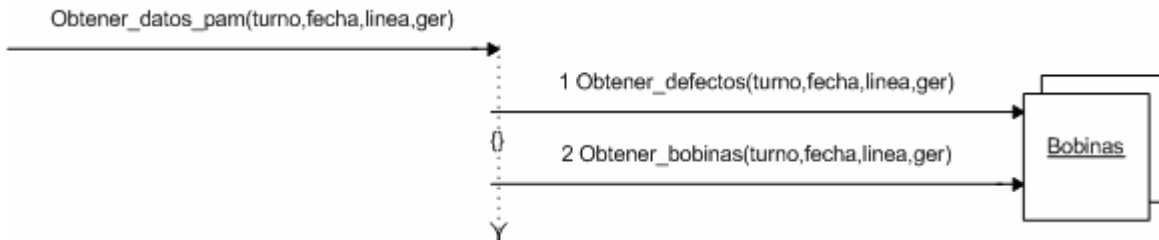


Figura 39: DI Escenario Visualizar puesta a mil

DI Escenario Visualizar calidad:

En el diagrama de la figura se observa la iteración entre los objetos del modulo del libro de novedades para la obtención de los resultados propuestos en UCS Gestión de material defectuoso.

Al hacer la solicitud al sistema *Obtener_datos_calidad* (*turno, fecha, línea, gerencia*) se generan dos llamadas a la colección de bobinas pertenecientes al SGD:

- Se hace la llamada *Obtener_defectos* (*turno, fecha, línea, gerencia*) a la colección de bobinas pertenecientes al sistema de gestión de defectos lo cual genera un listado de los defectos que presentaron las bobinas producidas en el turno, fecha y línea de producción indicada.
- Se hace la llamada *Obtener_bobinas* (*turno, fecha, línea, gerencia*), con la cual se obtienen los datos referentes a las bobinas producidas en el turno, fecha y línea elegidos, y corresponden a los defectos obtenidos en la llamada anterior.

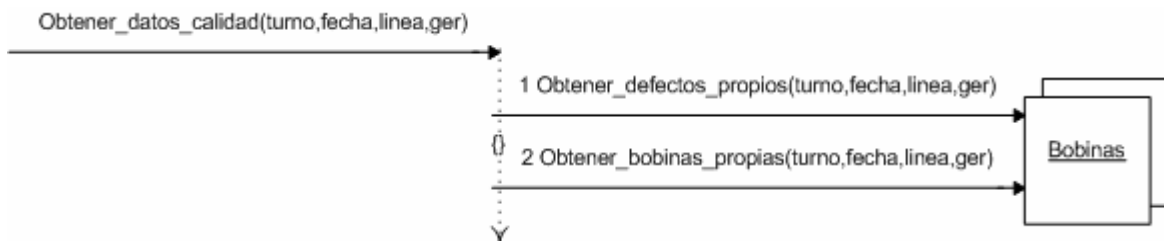


Figura 40: DI Escenario Visualizar calidad

DI Escenario Visualizar equipos defectuosos:

Este escenario corresponde al caso de uso Gestión de equipos defectuosos, con el cual se procede a generar un listado de los equipos que han presentado defectos. Se observa en el diagrama que se

hace una solicitud a la colección de equipos, el cual internamente genera un listado de los equipos que se encuentran en observación.

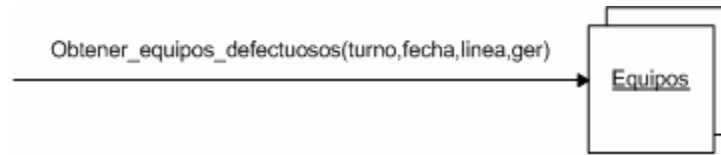


Figura 41: DI Escenario visualizar equipos defectuosos

DI Escenario Visualización de novedades.

Este diagrama se corresponde con el caso de uso visualización de novedades, el cual está asociado al caso de uso gestión de novedades generales, el cual se desglosa en visualización de novedades, y crear/modificar novedad tanto general como jerárquica. El sistema responde al realizarse el pedido de listar novedades.

1. Se solicita a la colección de novedades generales el *listado_novedad_general(turno, línea, fecha, id_sección)*. A su vez este objeto interactúa con el objeto línea, sección, y turno para hacer las solicitudes *obtener_linea(id_instalacion)*, *obtener_seccion(id_seccion)*, *obtener_turno(turno)*, respectivamente. Por cada objeto de la colección de novedades se realiza la llamada correspondiente.
2. Se solicita a la colección de novedades jerárquicas el *listado_novedad_jerarquica(turno, línea, fecha, id_sección)*. A su vez este objeto interactúa con el objeto línea, sección, y turno para hacer las solicitudes *obtener_linea(id_instalacion)*, *obtener_seccion(id_seccion)*, *obtener_turno(turno)*, respectivamente. Por cada objeto de la colección de novedades se realiza la llamada correspondiente.

Es de notar que para mostrar el listado de novedades generales, se toma en cuenta la actuación del objeto novedades jerárquicas debido a que estas deben visualizarse en el sitio donde se muestren las novedades generales.

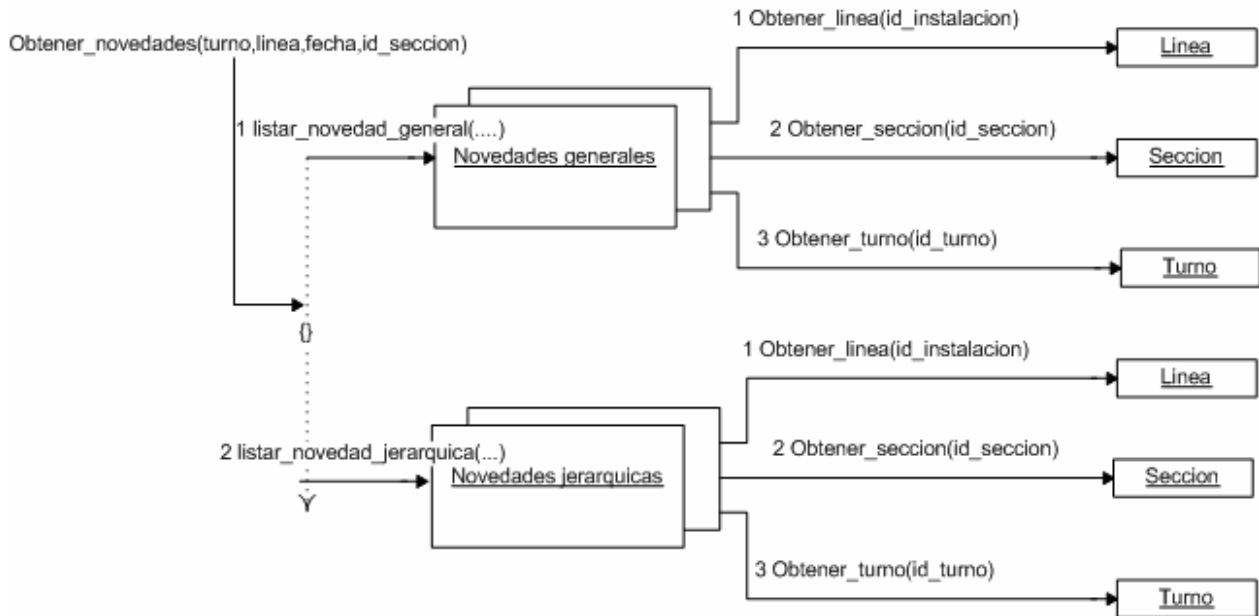


Figura 42: DI Escenario Visualización de novedades

DI Escenario Crear/Modificar Novedad General

Este escenario representa dos casos de uso Crear y Modificar novedad general. Se usa solo un diagrama ya que la actuación de los objetos del sistema es exactamente igual. La solicitud se hace al sistema con la llamada `Guardar_novedad_general(turno, fecha, línea, título, descripción, autor, sección)`, la secuencia es la siguiente:

1. Se verifica la existencia de la instalación en el objeto línea.
2. Se verifica la existencia de la sección, que identifica el tipo de bobina.
3. Se verifica la existencia del turno.
4. Luego de verificar los datos claves de la colección, se verifica que el ítem a agregar no exista, para crearlo en la colección. En caso de que el objeto exista, se hará la modificación de los campos solicitados.



Figura 43: DI Escenario Crear/Modificar Novedad General

DI Escenario crear, modificar novedad jerárquica.

Este escenario representa dos casos de uso Crear y Modificar novedad jerárquica. Se usa solo un diagrama ya que la actuación de los objetos del sistema es exactamente igual. La solicitud se hace al sistema con la llamada `Guardar_novedad_jerarquica(fecha_inicio, fecha_fin, turno, titulo, descripción, autor, id_linea, id_sección)`, la secuencia es la siguiente:

5. Se verifica la existencia de la instalación en el objeto línea.
6. Se verifica la existencia de la sección, que identifica el tipo de bobina.
7. Se verifica la existencia del turno.
8. Luego de verificar los datos claves de la colección, se verifica que el ítem a agregar no exista, para crearlo en la colección. En caso de que el objeto exista, se hará la modificación de los campos solicitados.

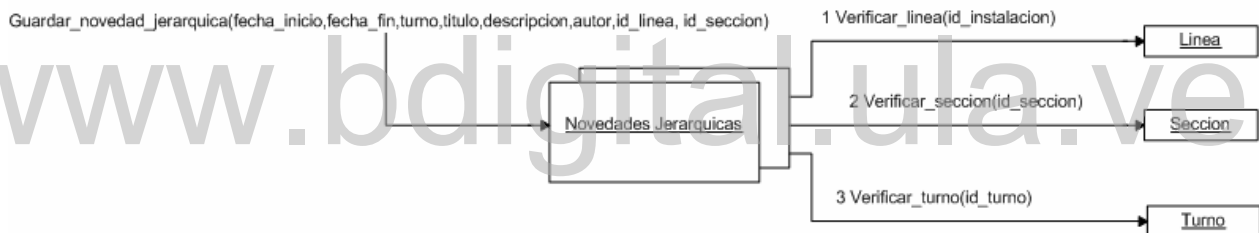


Figura 44: DI Escenario Crear/Modificar Novedad Jerárquica

Diagramas de clases de diseño (DCD)

Un diagrama de clases de diseño representa las especificaciones de las clases e interfaces software. Este tipo de diagramas nos provee información general del sistema con respecto a clases, asociaciones, y atributos, así como interfaces, con sus operaciones y constantes, métodos, información acerca del tipo de atributos, navegabilidad y dependencias.

A partir de los diagramas expuestos anteriormente, elegimos las siguientes entidades como las clases del modelo de diseño:

Clases SGD

- Clase bobina defectuosa.
- Clase defecto.
- Clase estado de calidad.

- Clase datos bobinas.
- Clase línea.
- Clase aplicación.
- Clase responsable de bobinas.
- Clase responsable.
- Clase causa.

Clases LN

- Clase novedades jerárquicas
- Clase novedades generales
- Clase turnos
- Clase secciones
- Clase líneas

Similarmente como se realizó el diagrama del modelo de dominio, se realiza el modelo de diseño de clases, en el cual estarán presentes las clases con sus correspondientes atributos. El siguiente paso en el diseño corresponde a la adición de las operaciones o los nombres de los métodos, los cuales son identificados a través de los diagramas de iteración.

Al igual que en el modelo de dominio, las entidades poseen relaciones entre sí, las cuales representan una interacción directa entre ellas. En principio, debe seleccionarse e identificarse las relaciones del modelo de dominio y aplicarlas al diagrama de clases de diseño.

Básicamente, el modelo de Clases de Diseño es una modificación del modelo de dominio, y la principal diferencia radica en el nivel de abstracción, debido a que el modelo de clases tiene como finalidad la especificación de las Clases Software.

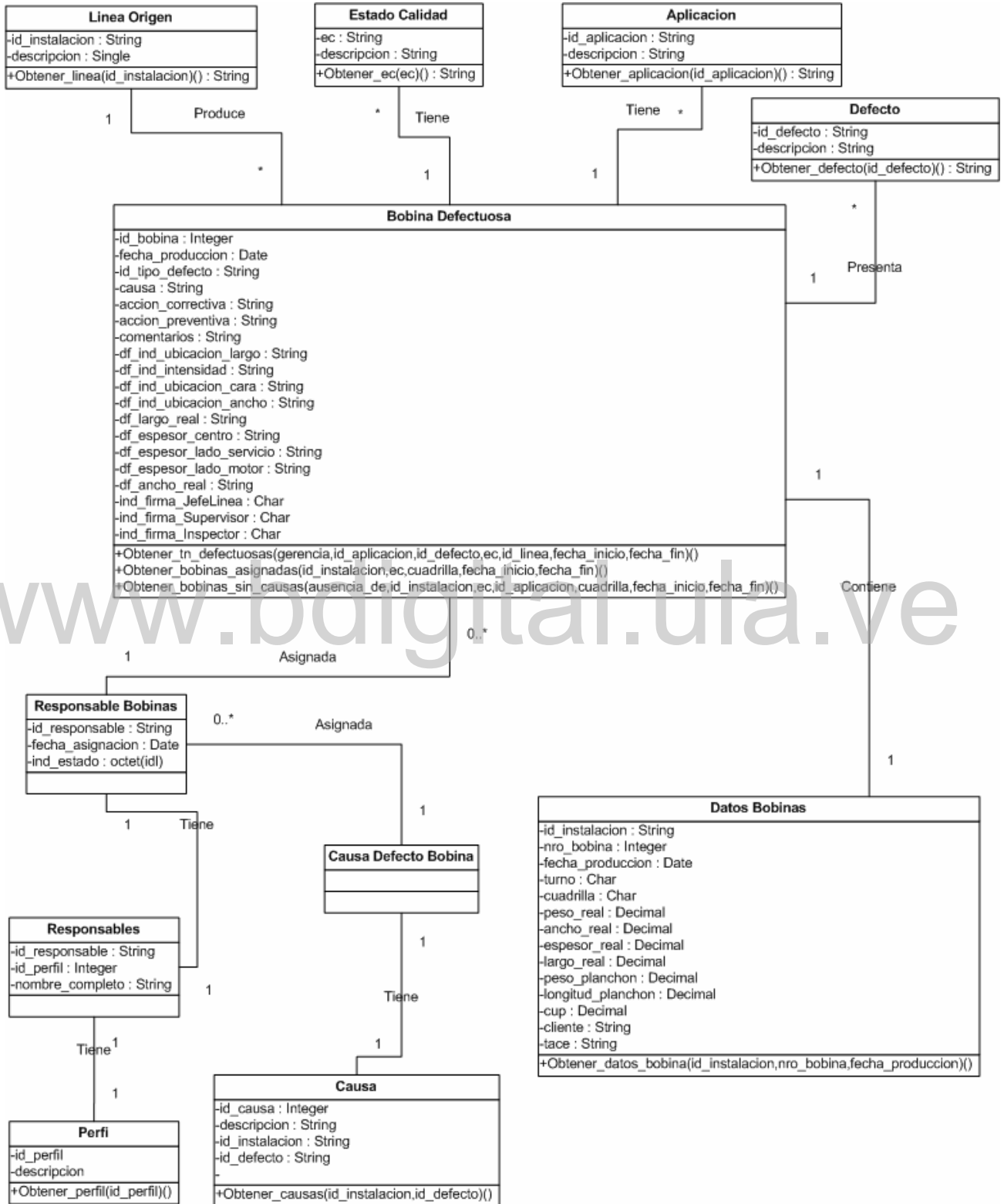


Figura 45: DCD del sistema de gestión de defectos

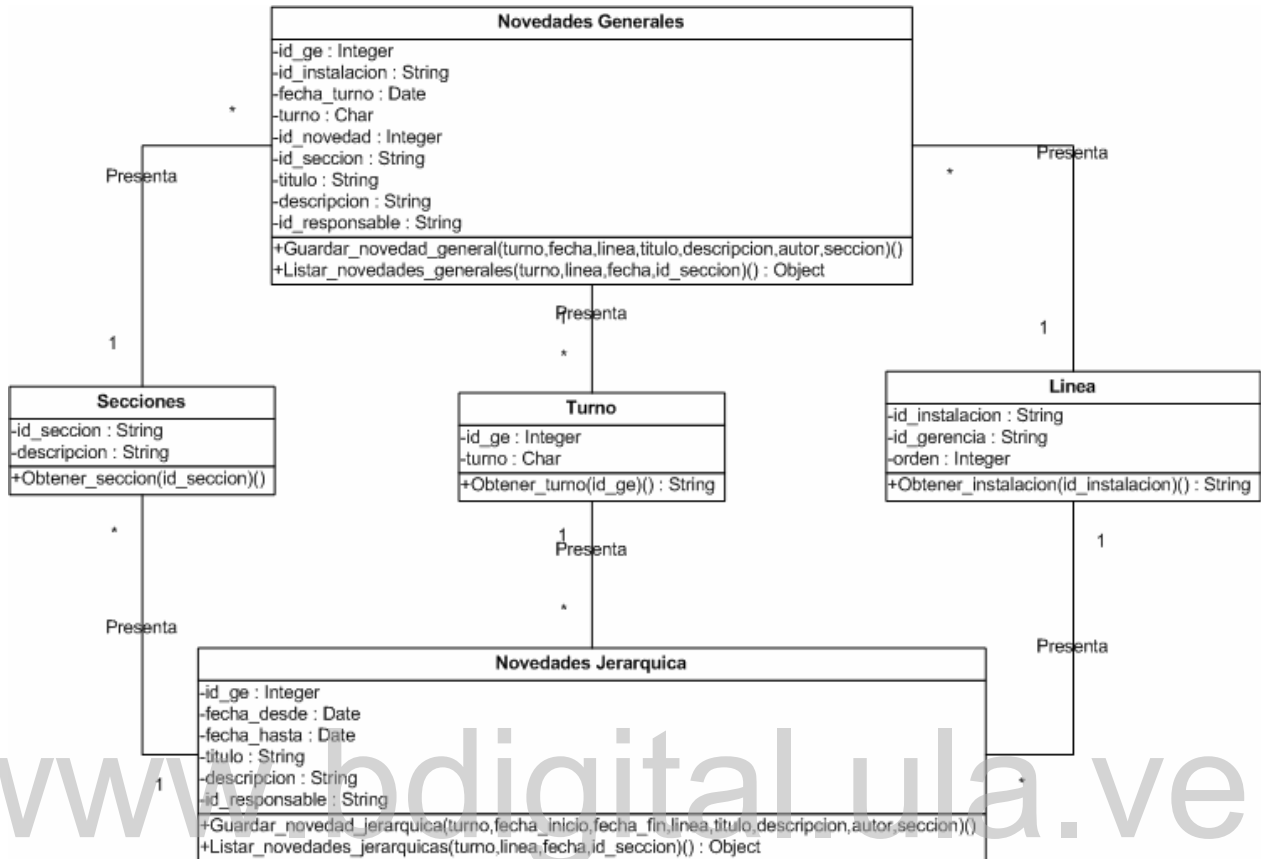


Figura 46: DCD del Libro de novedades

Una vez obtenido este diagrama se procede a la etapa de diseño de la Base de Datos, la cual estuvo a cargo del líder del proyecto designado por la Empresa.

Sin embargo es importante acotar que teniendo como base las clases de diseño presentadas, es material suficiente para pasar al diagrama relacional de la Base de Datos el cual se implementará en SQL SERVER.

Implementación

La implementación consiste en la elaboración del producto parcial definido por el análisis y diseño durante esta primera iteración. Dicho producto inicial esta conformado por la base de datos, las rutinas o procedimientos almacenados, las rutinas en visual Basic y las páginas Web.

Rutinas y procedimientos almacenados:

Luego de tener el diseño de base de datos, siguiendo los estándares de la empresa para la obtención de los datos se procedió a crear procedimientos almacenados, que contienen las

consultas deseadas, funciones y vistas. Estas consultas serán accedidas desde .Net a través de una clase que se encarga de la conexión entre el código Vb.Net, y las consultas almacenadas en SQL SERVER. A continuación mostraremos un tablero con los parámetros de entrada de los procedimientos almacenados, los datos de entrada, y los resultados obtenidos.

Procedimientos	Parámetros de entrada	Resultados obtenidos
web_sgc_reporte_defectos_instalaciones	<ul style="list-style-type: none"> • @fecha_desde • @fecha_hasta • @ger • @id_tipo_defecto • @ind_aplicacion • @ec • @tipo_fecha 	Listado de toneladas defectuosas por instalación
web_sgc_listar_bobinas_asignadas	<ul style="list-style-type: none"> • @fecha_desde • @fecha_hasta datetime • @linea_defecto • @ec • @gerencia • @cuadrilla 	Listado de bobinas asignadas
web_sgc_listar_bobinas_sin_causas_acciones	<ul style="list-style-type: none"> • @id_instalacion • @id_tipo_defecto • @gerencia • @fecha_desde • @fecha_hasta • @ausencia_de 	Listado de bobinas sin que tienen ausencia de causas, o defectos

Como vemos en el listado se muestran el estándar que se sigue en el desarrollo, la notación usada para la llamada a los store procedures consta del prefijo Web, luego el nombre del proyecto, y finalmente la indicación del listado de datos que traerá la llamada. Como las llamadas son

equivalentes, el resto de los procedimientos almacenados, y el código fuente del mismo se encuentra en el CD adjunto a este proyecto.

Pantallas:

Al tener diseñada la base de datos, el siguiente paso es el diseño y elaboración de las pantallas de acuerdo con las características visuales respecto de los estándares de la empresa y la lógica de la programación necesaria para la implementación de los casos de uso.

La navegación entre las pantallas Web diseñadas se realiza a través de un menú configurable por un archivo XML, el cual posee una estructura ramificada, similar a una estructura de árbol, el cual permite la organización adecuada de la información, y así mismo permite que el sistema mantenga el estándar de acceso y visualización a los sistemas Web.

A continuación presentamos la estructura manejada en el Libro de novedades, en cuanto a la navegabilidad en las pantallas y la estructura XML correspondiente.

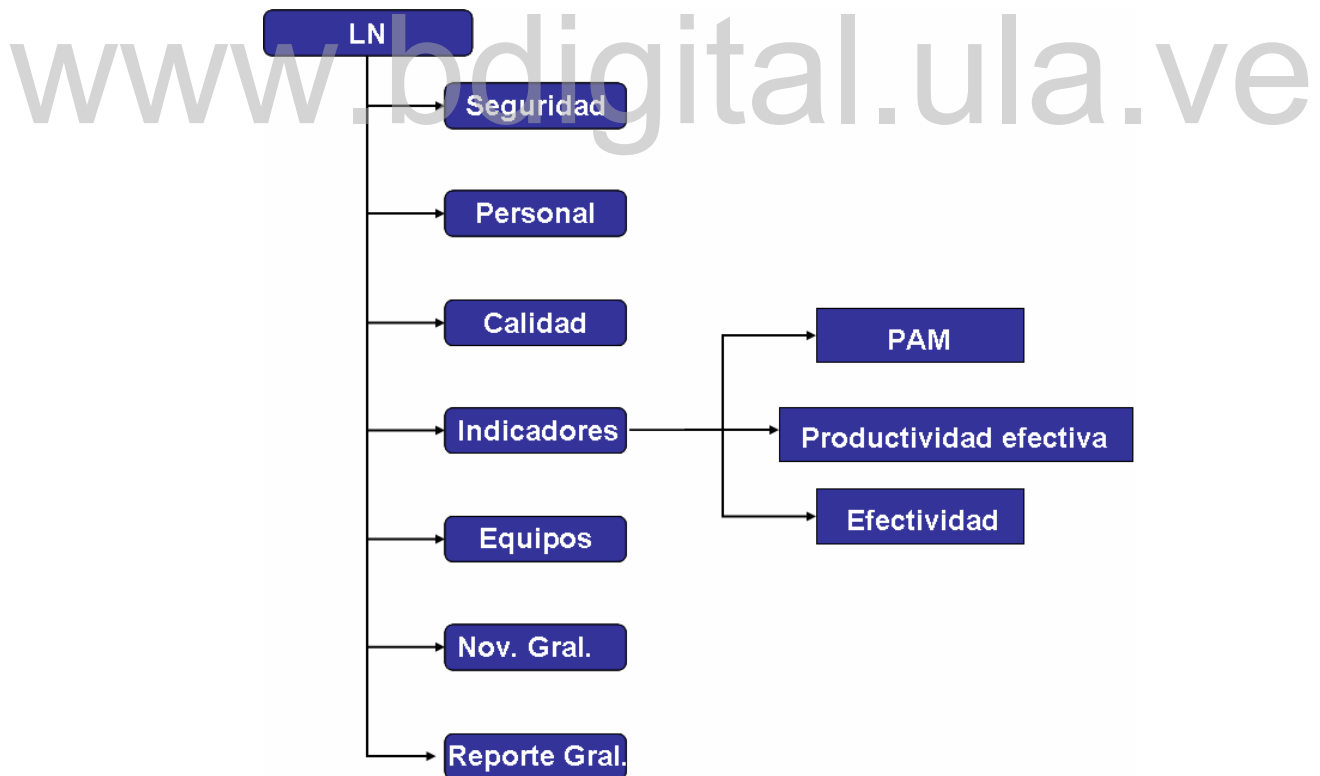


Figura 47: Navegación de las pantallas LN


```

- <Menu>
- <item>
+ <item caption="SIDOR" id="home_In" default="home_url_In">
  <item caption="APR" id="apr" target="_self" url="/sgl_In/en_construccion.aspx" default="serv" />
  <item caption="APA" id="apa" target="_self" url="/sgl_In/en_construccion.aspx" default="serv" />
  <item caption="APL" id="apl" target="_self" url="/sgl_In/en_construccion.aspx" default="serv" />
+ <item caption="LAC" id="lac" default="lac_reporte_general">
- <item caption="LAF" id="laf" default="laf_reporte_general">
  <item caption="Seguridad" id="laf_In_seg" url="/sgl_In/novedades_inicio.aspx?g=laf&tipo_novedad=seg"
    target="_self" />
  <item caption="Personal" id="laf_In_per" url="/sgl_In/novedades_inicio.aspx?g=laf&tipo_novedad=per" target="_self" />
  <item caption="Calidad" id="laf_In_cal" url="/sgl_In/calidad.aspx?g=laf" target="_self" />
- <item caption="Indicadores" id="laf_In_ind" default="laf_pam">
  <item caption="PAM" id="laf_pam" url="/sgl_In/pam.aspx?g=laf" target="_self" />
  <item caption="Productividad Efectiva" id="laf_produc_efec" url="/sgl_In/productividad_efectiva.aspx?g=laf"
    target="_self" />
  <item caption="Efectividad" id="laf_efectividad" url="/sgl_In/efectividad.aspx?g=laf" target="_self" />
</item>
<item caption="Programa" id="laf_In_prg" url="/sgl_In/novedades_inicio.aspx?g=laf&tipo_novedad=prg"
  target="_self" />
<item caption="Insumos" id="laf_In_ins" url="/sgl_In/novedades_inicio.aspx?g=laf&tipo_novedad=ins" target="_self" />
<item caption="Equipos" id="laf_In_equ" url="/sgl_In/equipos.aspx?g=laf" target="_self" />
<item caption="ING." id="laf_ing" url="/sgl_In/novedades_inicio.aspx?g=laf&tipo_novedad=ing" target="_self" />
<item caption="Mantto" id="laf_mant" url="/sgl_In/novedades_inicio.aspx?g=laf&tipo_novedad=mantto" target="_self" />
- <item caption="Reportes" id="laf_reporte" default="laf_reporte_general">
  <item caption="General" id="laf_reporte_general" url="/sgl_In/reporte.aspx?g=laf" target="_self" />
  <item caption="Novedades" id="laf_reporte_novedades" url="/sgl_In/reporte_novedades.aspx?g=laf"
    target="_self" />
</item>
<item caption="Nov. Gral." id="laf_novedad_gral" url="/sgl_In/novedades_generales.aspx?g=laf" target="_self" />
</item>
<item caption="BYA" id="bya" target="_self" url="/sgl_In/en_construccion.aspx" default="serv" />
<item caption="SERV" id="serv" target="_self" url="/sgl_In/en_construccion.aspx" default="serv" />
</item>
</Menu>

```

Figura 48: Menú XML para navegación del sistema

Estructura del área de reportes del SGD usada:

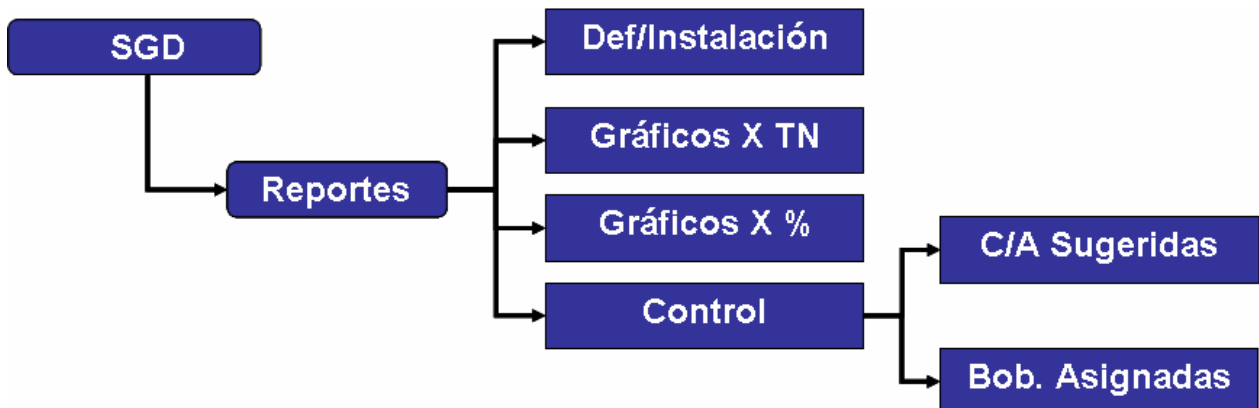


Figura 49: Navegación en las pantallas SGD

```

<?xml version="1.0" encoding="UTF-8" ?>
+ <!-- -->
- <Menu>
- <item>
+ <item caption="SIDOR" id="home_sgc" default="home_url_sgc">
  <item caption="APR" id="apr" target="_self" url="/sgl_defectos/browse/en_construccion.aspx" default="apr" />
  <item caption="APA" id="apa" target="_self" url="/sgl_defectos/browse/en_construccion.aspx" default="apa" />
  <item caption="APL" id="apl" target="_self" url="/sgl_defectos/browse/en_construccion.aspx" default="apl" />
- <item caption="LAC" id="lac" target="_self" default="lac_SGC_reportes_def">
  <item caption="Revisión Def." id="lac_SGC_casugeridas" url="/sgl_defectos/browse/bobinas_qnx.aspx?g=LAC"
    target="_self" />
+ <item caption="Comité" id="lac_comite" default="lac_SGC_Comite">
+ <item caption="Respons." id="lac_SGC_resp" default="lac_SGC_asig_resp">
+ <item caption="Gestión" id="lac_SGC_Gestion" default="lac_SGC_Gestion_basig" target="_self">
- <item caption="Reportes" id="lac_SGC_reportes" default="lac_SGC_reportes_def">
  <item caption="Def/Instalacion" id="lac_SGC_reportes_def"
    url="/sgl_defectos/browse/reportes_defecto_instalacion.aspx?g=LAC" target="_self" />
+ <item caption="Graficos x Tn" id="lac_SGC_reportes_graficos_tn" target="_self"
  default="lac_SGC_reportes_tn_EV_Tiempo">
+ <item caption="Graficos x %" id="lac_SGC_reportes_graficos_porc" target="_self"
  default="lac_SGC_reportes_porc_EV_Tiempo">
- <item caption="CONTROL" id="lac_SGC_control" target="_self" default="lac_SGC_Control_casug">
  <item caption="C/A Sugeridas" id="lac_SGC_Control_casug"
    url="/sgl_defectos/browse/reporte_ca_sugeridas.aspx?g=LAC" target="_self" />
  <item caption="Bobinas Asig" id="lac_sgc_Bobinas_Asig" url="/sgl_defectos/browse/Bobinas_Asignadas.aspx?
    g=LAC" target="_self" />
  <item caption="Eventos" id="lac_sgc_eventos" url="/sgl_defectos/browse/eventos.aspx?g=LAC"
    target="_self" />
  </item>
</item>
</item>
<item caption="PCP" id="lac_SGC_PCP" url="/sgl_defectos/browse/reporte_pcp_laf.aspx?g=LAC" target="_self" />
<item caption="SICOP" id="lac_SGC_SICOP" url="/sgl_defectos/browse/sicop.aspx?g=LAC" target="_self" />

```

Figura 50: Menú XML para navegación del sistema

En la figuras siguientes se muestra la visualización correspondiente al menú utilizado para los proyectos en Nivel 2B del área de automatización, el cual es construido siguiendo la estructura definida por el archivo XML correspondiente. El menú muestra solo dos niveles del árbol, mientras que el resto de los elementos son mostrados según se despliegue temporalmente cada una de las ramas. El usuario es identificado y mostrado al igual que la fecha del acceso a la página Web.



Figura 51: Visualización del Menú Estándar de Automatización para el LN

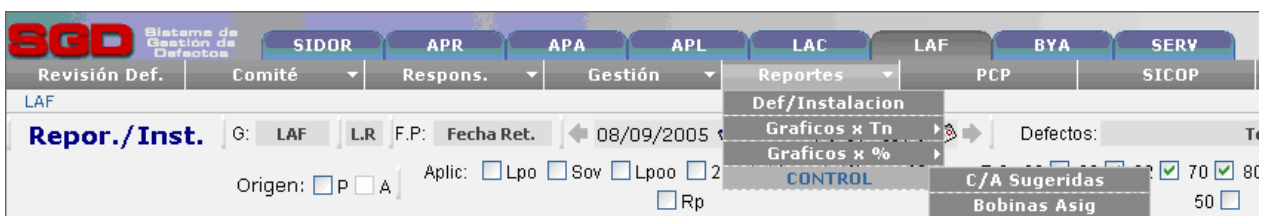


Figura 52: Visualización del Menú Estándar de Automatización para el SGD

Glosario de términos:

Término	Definición e Información
Causa_sugerida	Representa la causa sugerida que carga el operador en planta.
accion_preventiva	Representa la acción preventiva sugerida por el operador en planta para evitar el defecto.
accion_correctiva	Representa la acción correctiva sugerida por el operador en planta para corregir el defecto.
comentarios	Representa algún comentario que carga el operador en planta según su consideración.
ind_firma_JefeLinea	Representa la firma del Jefe de Línea
ind_firma_Supervisor	Representa la firma del Supervisor de Línea
ind_firma_Inspector	Representa la firma del Inspector de Línea
df_ind_ubicacion	Representa un código para la ubicación del defecto presentado en la bobina.
df_ind_intensidad	Representa un código para el grado de intensidad del defecto presentado en la bobina.
df_ind_ubicacion_cara	Representa un código para la ubicación cara del defecto presentado en la bobina.
df_ind_ubicacion_ancho	Representa un código para la ubicación ancho del defecto presentado en la bobina.
df_largo_real	Representa el largo del defecto presentado en la bobina.
df_espesor_centro	Representa el espesor del defecto presentado en la bobina.
df_espesor_lado_servicio	Representa el espesor del lado del servicio del defecto presentado en la bobina.
df_espesor_lado_motor	Representa el espesor del lado del motor del defecto presentado en la bobina.
df_ancho_real	Representa el ancho del defecto presentado en la bobina.
Fecha_produccion	Representa la fecha de producción de la bobina.
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.
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.
Perfil	Representa el perfil que puede tener un responsable en la Empresa.
Interrupción	Es el periodo de tiempo durante el cual las líneas de producción se detienen, por diferentes causas y provocan efectos negativos en la efectividad de las líneas y la productividad efectiva de las mismas.

www.bdigital.ula.ve

Iteración II

Luego de haber presentado el sistema obtenido en la iteración 1, los usuarios del sistema incrementaron los requisitos del mismo, por lo que se realizó un nuevo taller de requisitos para la definición de los nuevos casos de uso generados con los requerimientos agregados por los usuarios. A continuación mostramos la modificación realizada en el listado Actor – Objetivos para el SGD y el LN.

Listado Actor – Objetivo para el modulo SGD:

ACTOR	OBJETIVOS
Jefe de Procesos	<ul style="list-style-type: none"> • Visualizar toneladas afectadas por defecto, en cada línea de la gerencia. • Visualizar un listado de bobinas defectuosas por línea y defecto. • Asignar responsables a cada una de las bobinas defectuosas. • Análisis de reportes de los defectos gestionados, y los responsables de llevar a cabo las acciones adecuadas para tal gestión.

Como vemos en el listado de actor – objetivos, se agregan dos requerimientos más al área del sistema de gestión de defectos.

Listado Actor – Objetivo para el modulo LN:

ACTOR	OBJETIVOS
Supervisores	<ul style="list-style-type: none"> • Revisar el material defectuoso por línea de producción. • Revisar el material descartado por bobina en cada línea de producción. • Revisar los indicadores de producción por línea de

	<p>producción.</p> <ul style="list-style-type: none">• Revisar los equipos en estado de observación.• Crear y/o modificar novedades generales.• Crear y/o modificar comentarios particulares para los datos visualizados.
--	---

Como vemos en el listado de actor – objetivos, se agregan un nuevo objetivo para el supervisor quien deseará cargar comentarios para cada registro del sistema.

Estos nuevos requerimientos afectan el modelo de casos de uso planteado en la fase de inicio. A continuación planteamos el nuevo modelo de casos de uso modificado para ambos módulos del sistema.

www.bdigital.ula.ve

Modelo de Casos de Uso:

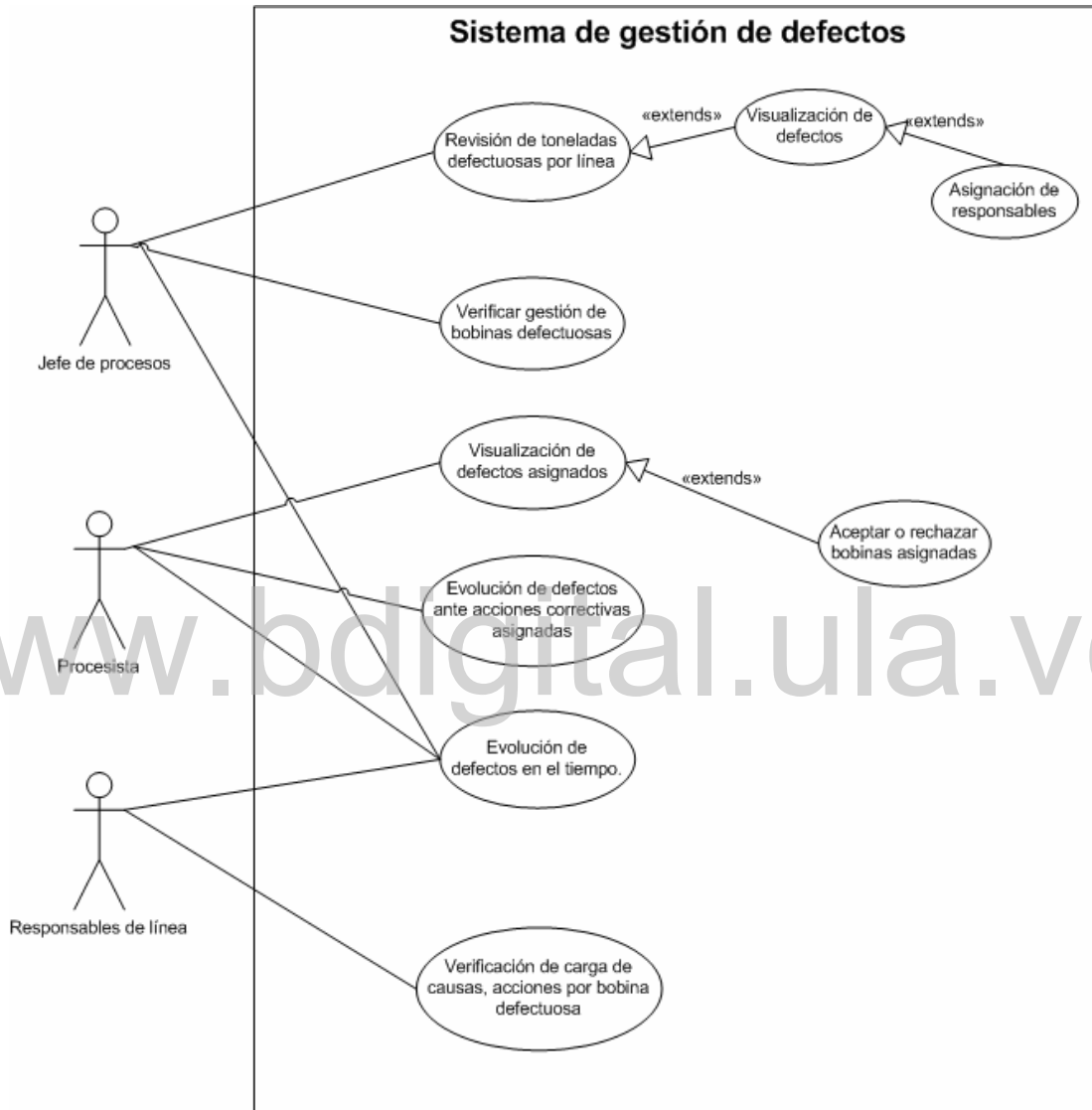


Figura 53: Diagrama de casos de uso SGD Iteración 2

Como vemos los nuevos casos de uso generados corresponden a:

- Visualización de defectos, el cual pretende cubrir la necesidad de ver el listado de bobinas afectadas por un defecto en particular. Esto permitirá al usuario corroborar que el tonelaje de las bobinas afectadas por defecto se corresponda con los datos reales.

- Asignación de responsables: en esta área se pretende dar permiso al procesista de la carga de responsables de la gestión de bobinas, esto lo podrá hacer a partir del listado que se presentará por defectos encontrados en cada línea.

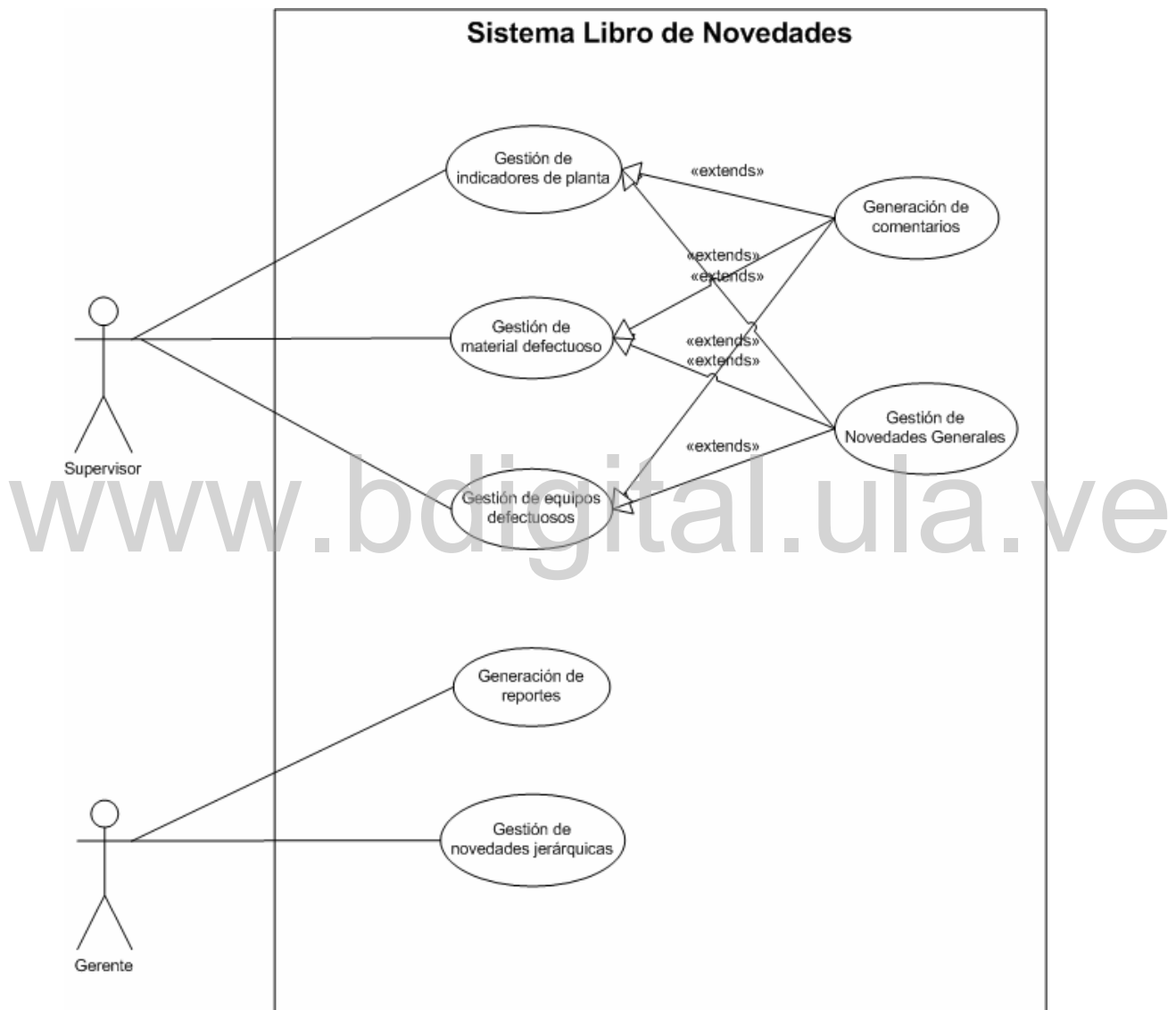


Figura 54: Diagrama de casos de uso LN Iteración 2

En esta parte el nuevo caso de uso generado corresponde a la gestión de comentarios, con el supervisor podrá cargar comentarios de los registros visualizados en cada área del libro de novedades. Esto es a modo de justificación de los sucesos de planta.

A continuación describimos tales casos de uso con mayor detalle:

Visualización de defectos:

<p>Actor Principal:</p> <p>Jefe de procesos: Desea revisar listas que contengan las características de las bobinas que han presentado un defecto en común y han sido producidas en las diferentes líneas de laminación a su cargo.</p>	
<p>Precondiciones:</p> <p>El usuario debe figurar en el sistema como jefe de procesos.</p>	
<p>Poscondiciones: Visualización de bobinas con un defecto en común.</p>	
<p>Escenario principal de éxito:</p>	
<ul style="list-style-type: none"> • Usuario accede a visualizar los detalles del defecto mostrado en el caso de uso Revisión de defectos por línea. 	<ul style="list-style-type: none"> • El sistema toma la selección fecha en el caso de uso Revisión de defectos por línea, y se visualizan los datos correspondientes a cada material afectado y sus características cualitativas.
<p>Requisitos especiales:</p>	
<p>Respuesta del sistema para la visualización de datos no mayor a 30 segundos.</p>	

Verificar gestión de bobinas defectuosas:

<p>Actor Principal:</p> <p>Jefe de línea: Desea verificar que las bobinas a las cuales se les ha asignado responsables hayan sido gestionadas.</p>
<p>Poscondiciones: Visualización de responsables asignados, y la gestión de bobinas que ha realizado.</p>

Escenario principal de éxito:	
<ul style="list-style-type: none"> • Usuario accede al sistema, selecciona el rango de fechas dentro del cual se mostrará las toneladas afectadas por instalación. 	<ul style="list-style-type: none"> • El sistema muestra un listado de responsables. Por cada responsable se lista la cantidad de bobinas que el responsable le ha asignado causas, y las que no le ha asignado causas. Por este listado se visualizan la cantidad de causas diferentes asociadas a las bobinas procesadas. También se visualiza si el responsable de defectos ha asignado responsables de acciones que gestionen los defectos de las bobinas. Por cada responsable de acción se observa si ha asignado acciones para corregir los defectos de las bobinas, y el estado de dichas acciones.
Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> • Usuario escoge un rango de fechas incoherente. 	<ul style="list-style-type: none"> • El sistema genera un mensaje de error.

Requisitos especiales:
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.

Asignación de responsables.

Actor Principal:	
Jefe de procesos: Visualizar la cantidad de defectos presentados en su línea de producción, y las toneladas de material que presentaron tal defecto.	
Poscondiciones: Asignación satisfactoria de responsables..	
Escenario principal de éxito:	
<ul style="list-style-type: none"> • Usuario escoge el responsable de defectos de una lista por cada bobina defectuosa. 	<ul style="list-style-type: none"> • El sistema guarda los datos, y genera un mensaje mostrando que se ha hecho la carga de datos satisfactoriamente.
Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> • Usuario no selecciona el responsable e intenta guardar los datos. 	<ul style="list-style-type: none"> • El sistema genera un mensaje de error.

Gestión de comentarios:

En este UCS se derivan otros casos de uso que deben ser especificados individualmente, y a continuación lo mostramos en el diagrama:

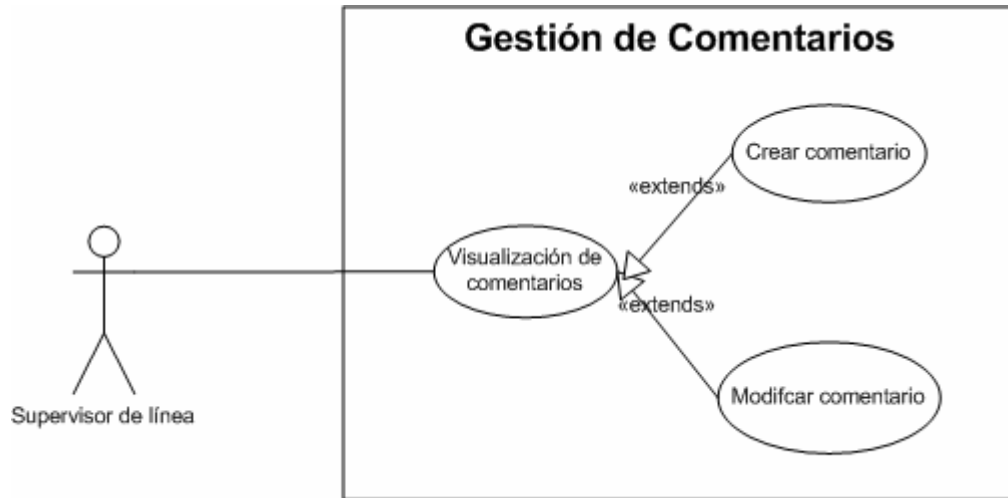


Figura 55: UCS Gestión de comentarios

Visualización de comentarios:

Actor Principal: Supervisor de línea.	
Personal involucrado e intereses: <ul style="list-style-type: none"> Supervisor: Desea revisar un listado los comentarios que justifican la ocurrencia de registros particulares, con respecto a los indicadores de planta, el material hallado como defectuoso o los equipos que han presentado defectos y están en observación. 	
Precondiciones: Comentarios inicialmente cargados.	
Poscondiciones: Visualización de comentarios.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> Usuario accede al sistema. 	<ul style="list-style-type: none"> Sistema toma la instalación, la fecha, el turno, el identificador del registro. Con tales datos se genera la visualización de comentarios.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Crear comentario:

Actor Principal: Supervisor de línea.	
Personal involucrado e intereses: <ul style="list-style-type: none"> Supervisor: Desea agregar un comentario para justificar la aparición de los registros. 	
Poscondiciones: Guardar satisfactoriamente los datos correspondientes al comentario.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> Usuario introduce el título del comentario. Usuario introduce la descripción del comentario. 	<ul style="list-style-type: none"> El sistema guarda los datos correspondientes al comentario, y la relación con el área desde donde es ejecutado..
Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> El usuario trata de almacenar un comentario sin incluir título, o descripción. 	<ul style="list-style-type: none"> El sistema genera un mensaje indicando que ha ocurrido un error en la carga de datos.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Modificar novedad general:

Actor Principal: Supervisor de línea.
Personal involucrado e intereses: <ul style="list-style-type: none"> Supervisor: Desea modificar una novedad para la visualización de las mismas.

Poscondiciones: Guardar satisfactoriamente los datos correspondientes a las novedades generales.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> • Usuario introduce el título modificado de la novedad general. • Usuario introduce la descripción modificada de la novedad general. 	<ul style="list-style-type: none"> • El sistema guarda los datos correspondientes a la novedad general.
Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> • El usuario trata de almacenar una novedad sin incluir título, o descripción. 	<ul style="list-style-type: none"> • El sistema genera un mensaje indicando que ha ocurrido un error en la carga de datos.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Elaboración de diagramas de secuencia:

En esta parte de la elaboración realizaremos los diagramas de secuencia generados de los nuevos casos de uso definidos en el taller de requisitos del sistema, se omitirán los DSS que previamente se ha definido en la iteración I.

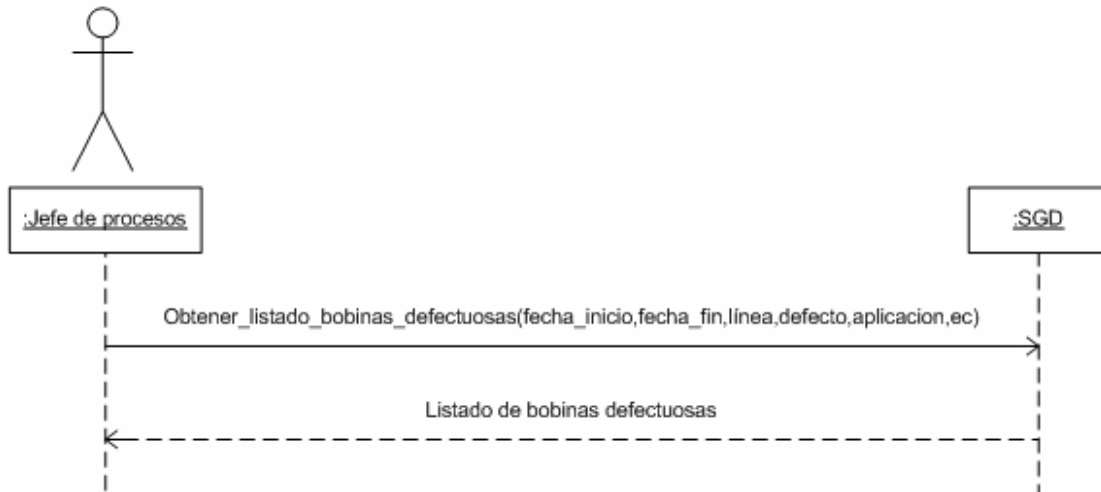


Figura 56: DSS Visualización de defectos

DSS Visualización de Defectos: este diagrama se obtuvo a partir del escenario de éxito del UCS *Visualización de defectos*, con el cual el usuario al acceder al sistema, se realiza una búsqueda de la información, para ello el usuario previamente deberá haber cargado el intervalo de tiempo en el cual se produjo las bobinas, la línea donde se originó el defecto, el defecto encontrado, la aplicación y estado de calidad de la misma. El sistema genera el listado de bobinas que cumple los requisitos del usuario.

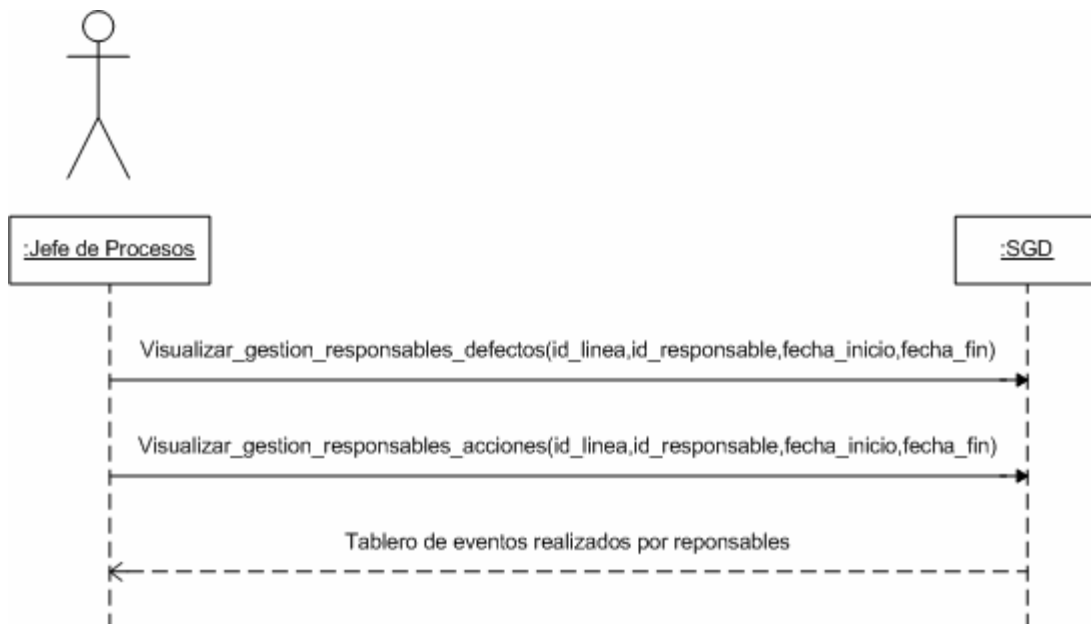


Figura 57: DSS Verificar gestión de bobinas defectuosas

DSS Verificar gestión de bobinas defectuosas: Como observamos este diagrama surge del caso de uso *Verificar gestión de bobinas defectuosas*, con el cual el usuario hace una petición al sistema, proveyendo los datos necesarios para crear un listado de los sucesos realizados por responsable de bobinas defectuosas y por responsable de acciones.

Al obtener ambos listados, el sistema debe generar un tablero con los datos informativos que indican las bobinas que han sido gestionadas por los responsables. A su vez se debe generar un listado de los responsables de acciones mostrando el estado de las acciones aplicadas por causas que se debe gestionar.



Figura 58: Asignar responsables

DSS Asignar responsables: este diagrama se deriva del *UCS Asignar Responsables*, mediante el cual el usuario escoge el responsable de defectos obtenido del listado de bobinas defectuosas y procede a almacenarlo, por cada registro se debe almacenar el responsable de defecto asociado a una bobina, por ello es necesario que el sistema reciba el identificador del objeto para la correspondiente asociación.

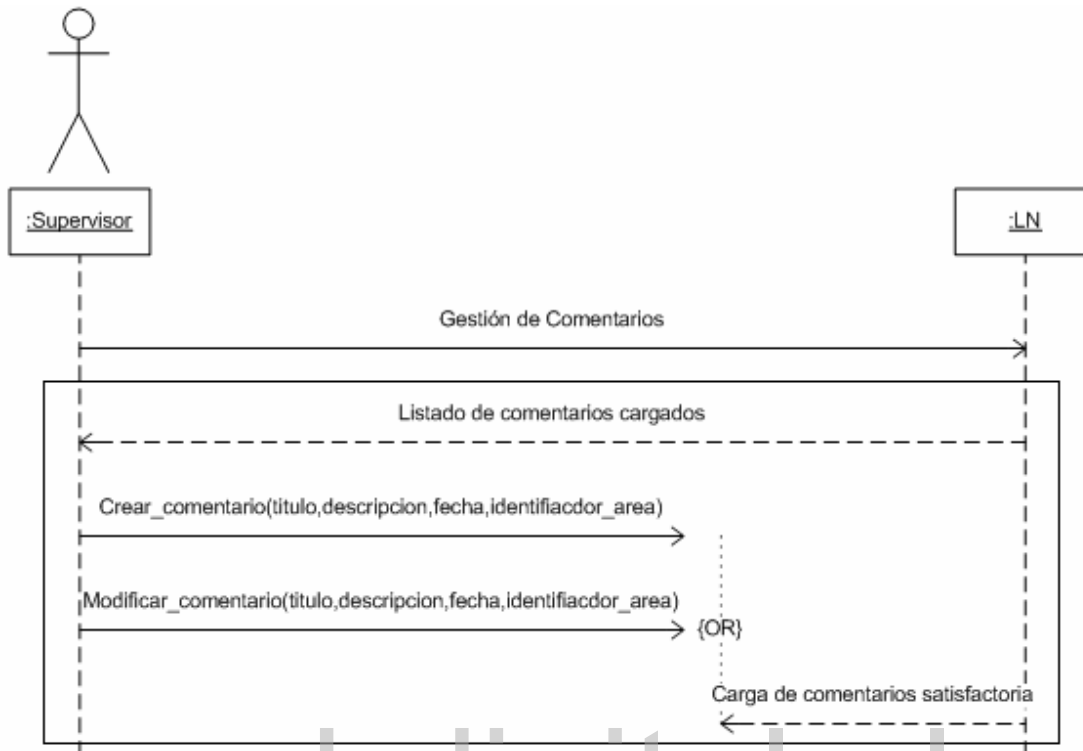


Figura 59: DSS Gestión de comentarios

DSS Gestión de comentarios: este diagrama está relacionado con el caso de uso de gestión de comentarios, el cual contiene internamente tres casos de uso asociados, ya que al acceder al área de gestión de comentarios, el sistema genera el listado de comentarios previamente cargados, luego el usuario decide si crea un nuevo comentario, o modifica un comentario previamente cargado por él.

Modelo de dominio

En este punto, se deben extraer los nombres o frases nominales del dominio en consideración, específicamente de la descripción textual de los casos de uso. Es recomendable utilizar los casos de uso en formato completo, puesto que constituyen una descripción excelente para la extracción de la información. De acuerdo a la aplicación de esta técnica partiendo de los casos de usos agregados al sistema observamos la inclusión de una nueva clase como lo son **Acciones**, para el SGD. Y en el caso del LN se incluye lo siguiente:

- Efectividad
- Productividad efectiva

- Puesta a mil
- Calidad
- Comentarios

A continuación especificamos la función de las nuevas clases incluidas:

Clase	Descripción
Acciones	Representa las acciones que se deberían ejecutar con la finalidad de evitar la aparición de defectos en las bobinas por causas previamente especificadas en el sistema.
Efectividad	Representan las interrupciones cargadas en el sistema que afectan este indicador de planta, y los usuarios le han agregado comentarios específicos en el sistema.
Productividad Efectiva	Representan las interrupciones cargadas en el sistema que afectan este indicador de planta, y los usuarios le han agregado comentarios específicos en el sistema.
Puesta a mil.	Representan los defectos de las bobinas que han sufrido cortes por la aparición de los mismos, y los usuarios le han agregado comentarios específicos en el sistema.
Calidad	Representan los defectos encontrados en el sistema de gestión de defectos, y afectan la calidad de la producción.

Agregando las clases conceptuales a los diagramas previamente realizados en la iteración 1, y aplicando las relaciones de asociación de clases podemos obtener los nuevos diagramas de esta etapa del desarrollo.

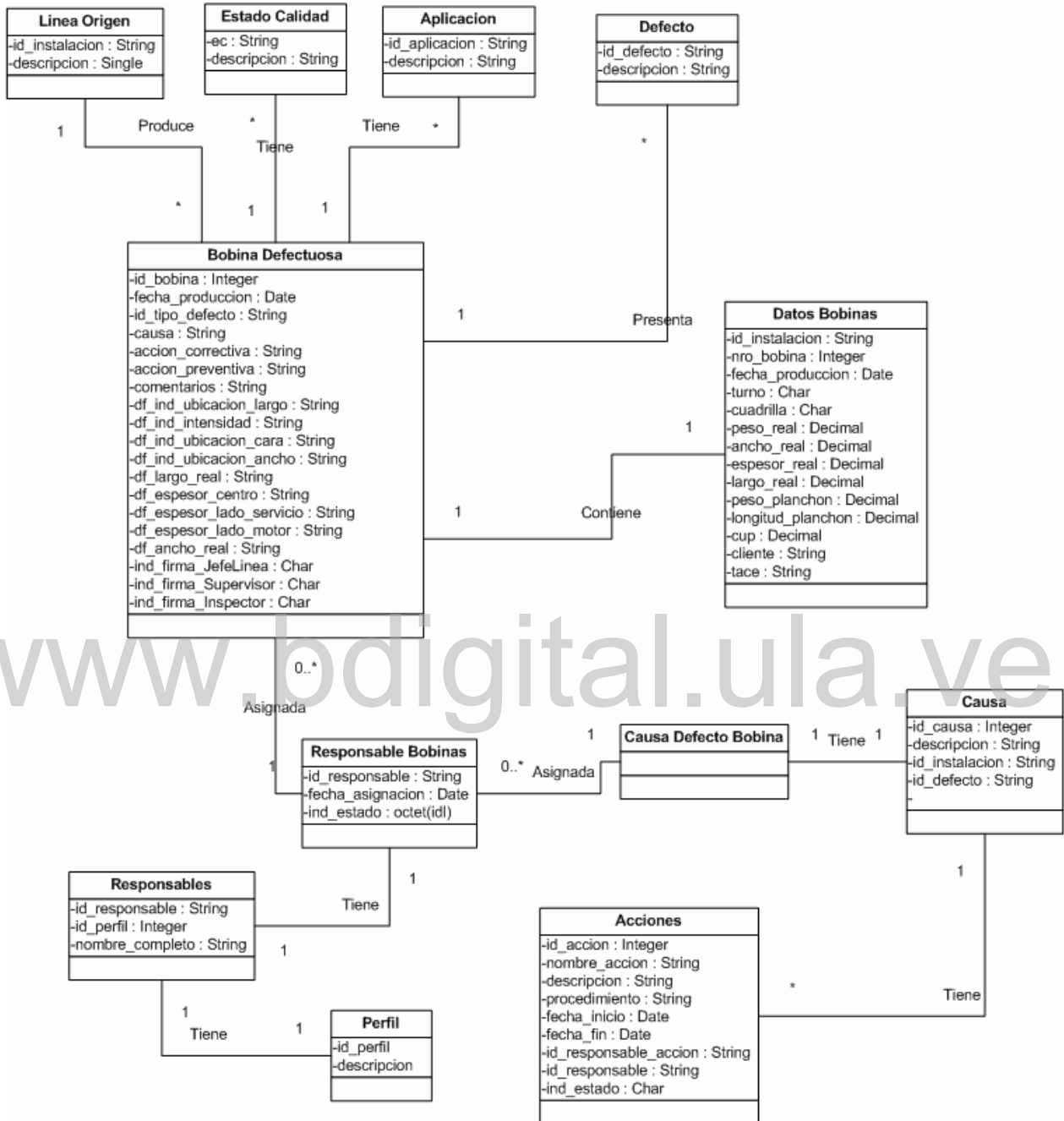


Figura 60: Modelo de domino del SGD para la iteración II

Para la obtención de este modelo de dominio se realizaron los mismos pasos que se siguieron en la iteración I. Con la finalidad de no redundar en la presentación de los modelos, solo se esta presentando el resultado obtenido a través del análisis y diseño del sistema.

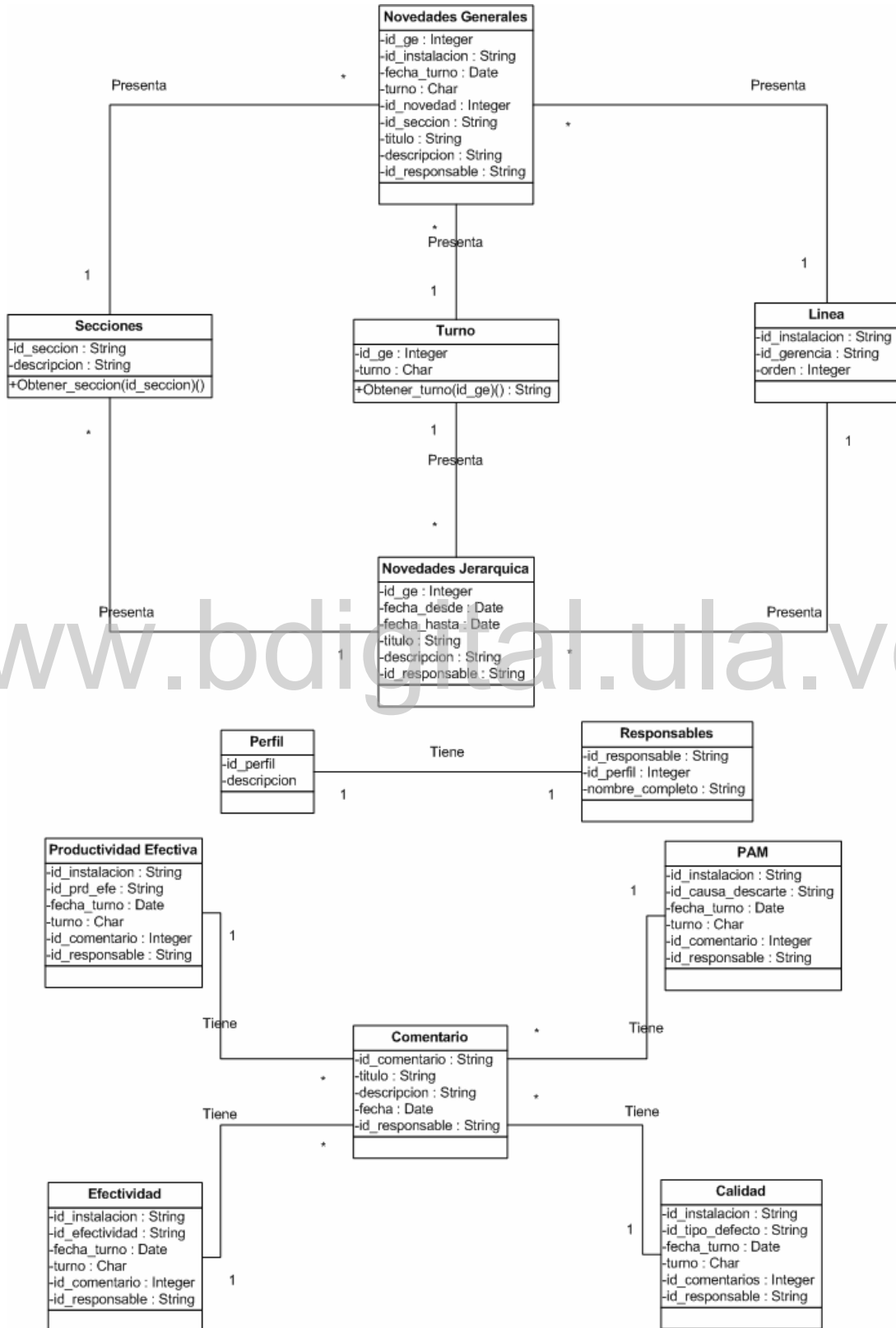


Figura 61: Modelo de domino del LN para la iteración II

Modelo del diseño:

Para esta iteración 2, se tiene un modelo de diseño un poco más amplio, el cual define una gran parte del sistema y es en esta sección es donde se explora realmente la funcionalidad del mismo. Nuevamente debe se estudia el comportamiento interno del sistema, pero solo de los nuevos elementos agregados al sistema.

DI Escenario Visualización de defectos:

El diagrama 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: Visualización de defectos*. En este caso se hace una solicitud al sistema, el cual debe responder retornando un listado de bobinas defectuosas. Para este caso, cuando el sistema recibe la solicitud *obtener_listado_bobinas_defectuosas (...)*, debería responder de la siguiente manera:

1. La *colección* 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 estructura todo el listado con los datos requeridos.

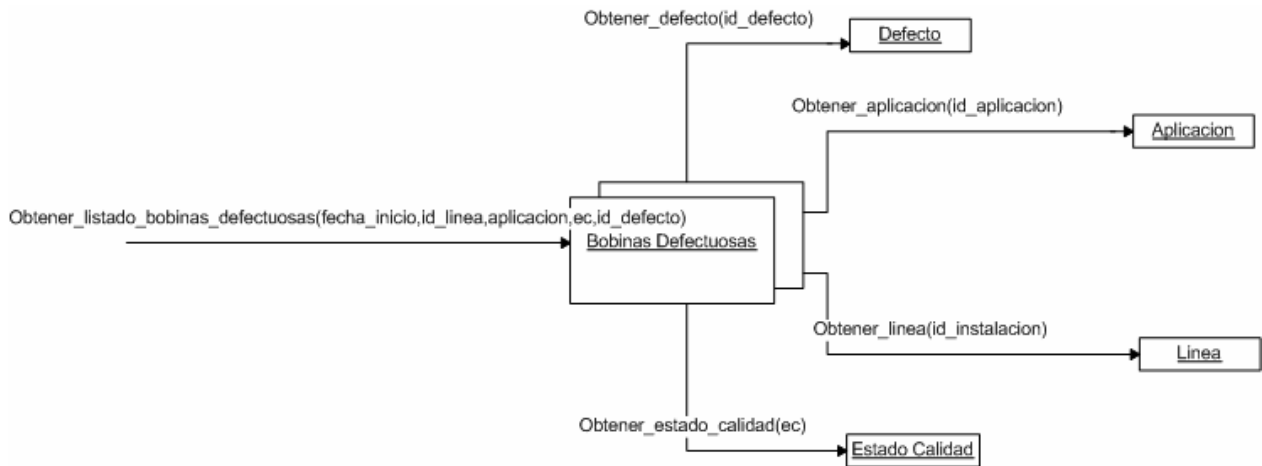


Figura 62: DI Escenario Visualización de defectos:

DI Escenario Verificar gestión de bobinas defectuosas

El usuario hace la solicitud para la visualización de las bobinas que han sido gestionadas en el sistema SGD. El sistema responde con la siguiente secuencia de acciones:

1. La colección de bobinas defectuosas envía la llamada *Obtener_responsables_bobina(...)* al objeto Responsables Bobinas.
2. A su vez por cada responsable de bobinas, se envía la solicitud a la instancia Causa defecto bobinas *Obtener_causas_aplicadas(...)*.
3. Posteriormente se hace la llamada al objeto Causa, solicitando *obtener_causas(...)*.
4. Finalmente, se solicita a las acciones *Cantidad_acciones_gestionadas_por_estado(...)*

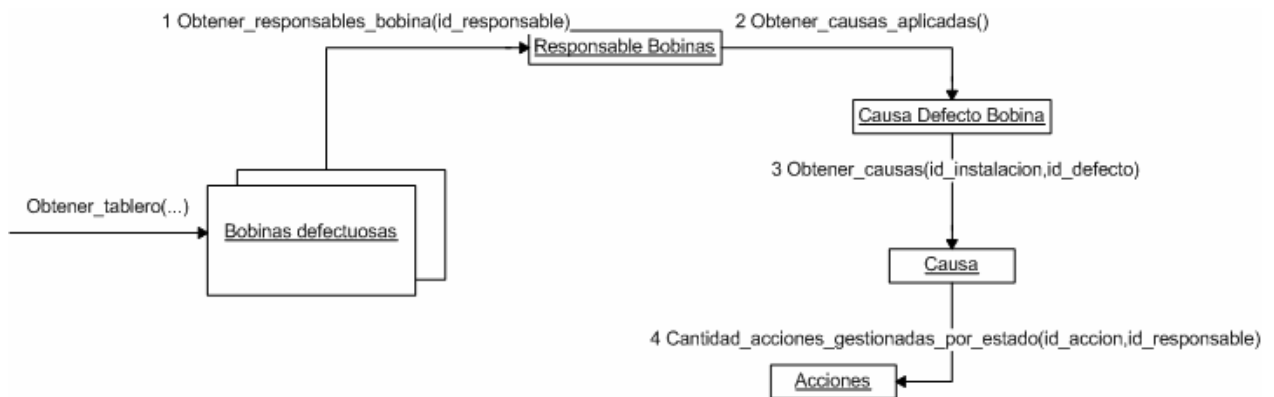


Figura 63: DI Escenario Verificar gestión de bobinas defectuosas

DI escenario asignación de responsables:

Se realiza la solicitud de almacenamiento de responsable de bobina a la colección de bobinas, donde esta instancia procede a verificar el responsable, si el responsable existe, se agrega en la colección de responsables bobinas, como vemos en la siguiente figura:

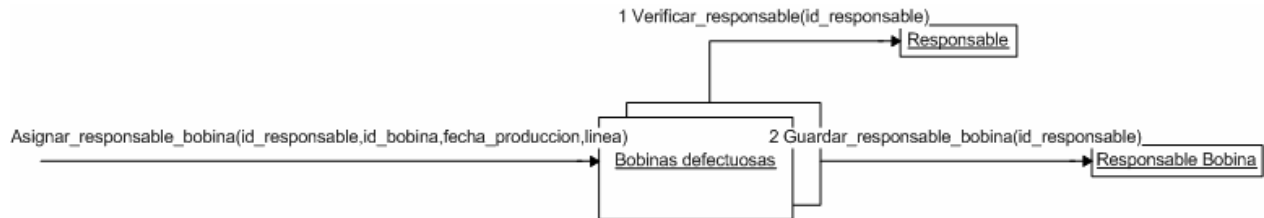


Figura 64: DI escenario asignación de responsables

DI escenario crear o modificar comentarios

Este escenario es llamado por diferentes instancias, las cuales son: *productividad efectiva*, *efectividad*, *calidad*, *PAM*, de acuerdo al tipo de comentario que se vaya a almacenar en el sistema. Como la secuencia de interacción entre los objetos del sistema es la misma, se presenta un solo diagrama representativo del proceso:

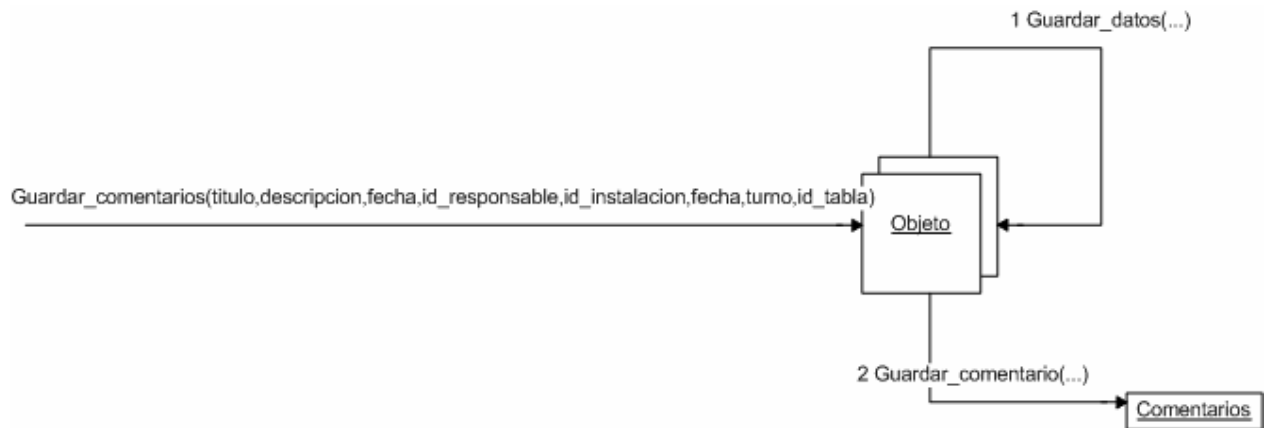


Figura 65: DI escenario crear o modificar comentarios

1. Se genera la llamada a la instancia deseada (: *productividad efectiva*, *efectividad*, *calidad*, *PAM*), a su vez esta instancia procede a guardar los datos recibidos, en caso de que existan previamente solo se modificaran, en caso de que no existan se almacenara un nuevo item del objeto.

2. El objeto, realiza la llamada a la instancia comentarios, la cual verifica la existencia del comentario, en caso de que exista, se modifican los datos, si no existe se crea un nuevo comentario asociado al objeto desde el cual se realiza la llamada correspondiente.

Diagramas de clases de diseño (DCD)

El diagrama de clases inicial para cada iteración coincide con el modelo de dominio de la misma. Los atributos de las entidades, se extraen en su mayoría del texto de los casos de uso, sin embargo una gran parte es extraída del análisis de requisitos posterior.

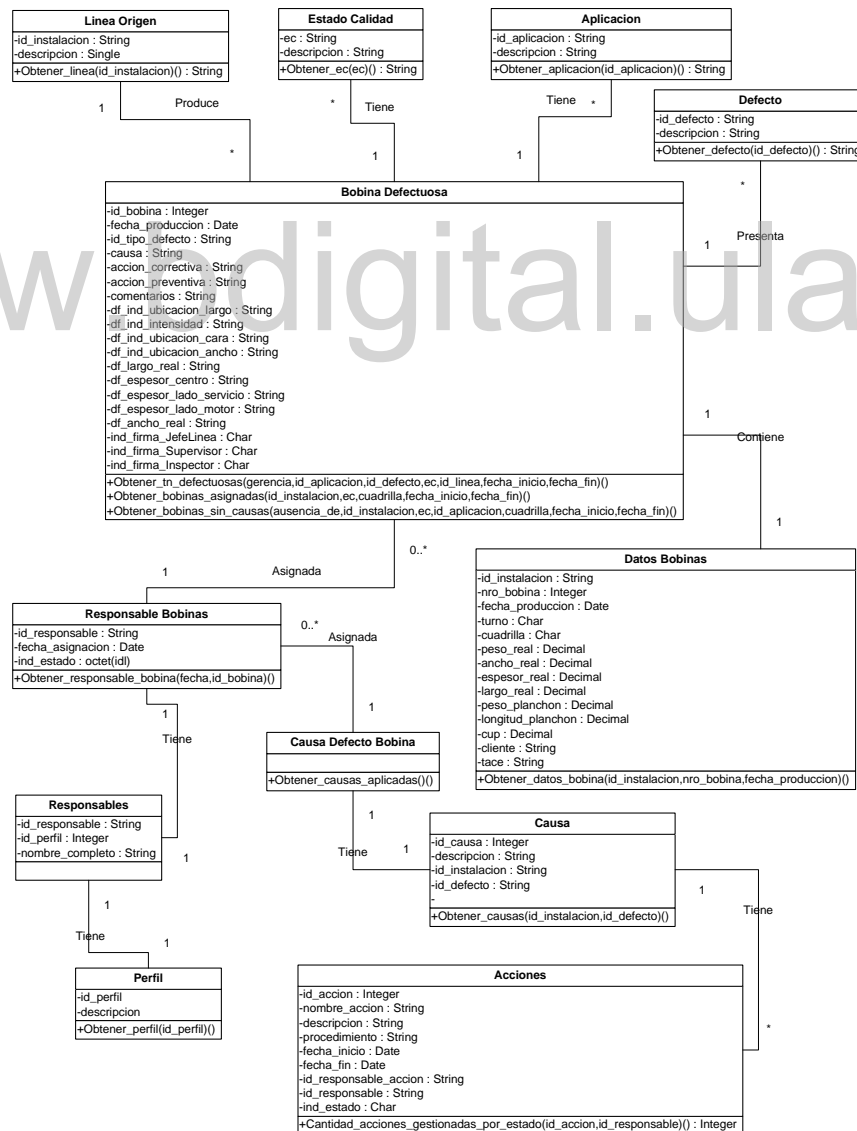


Figura 66: DCD para el SGD, iteración II

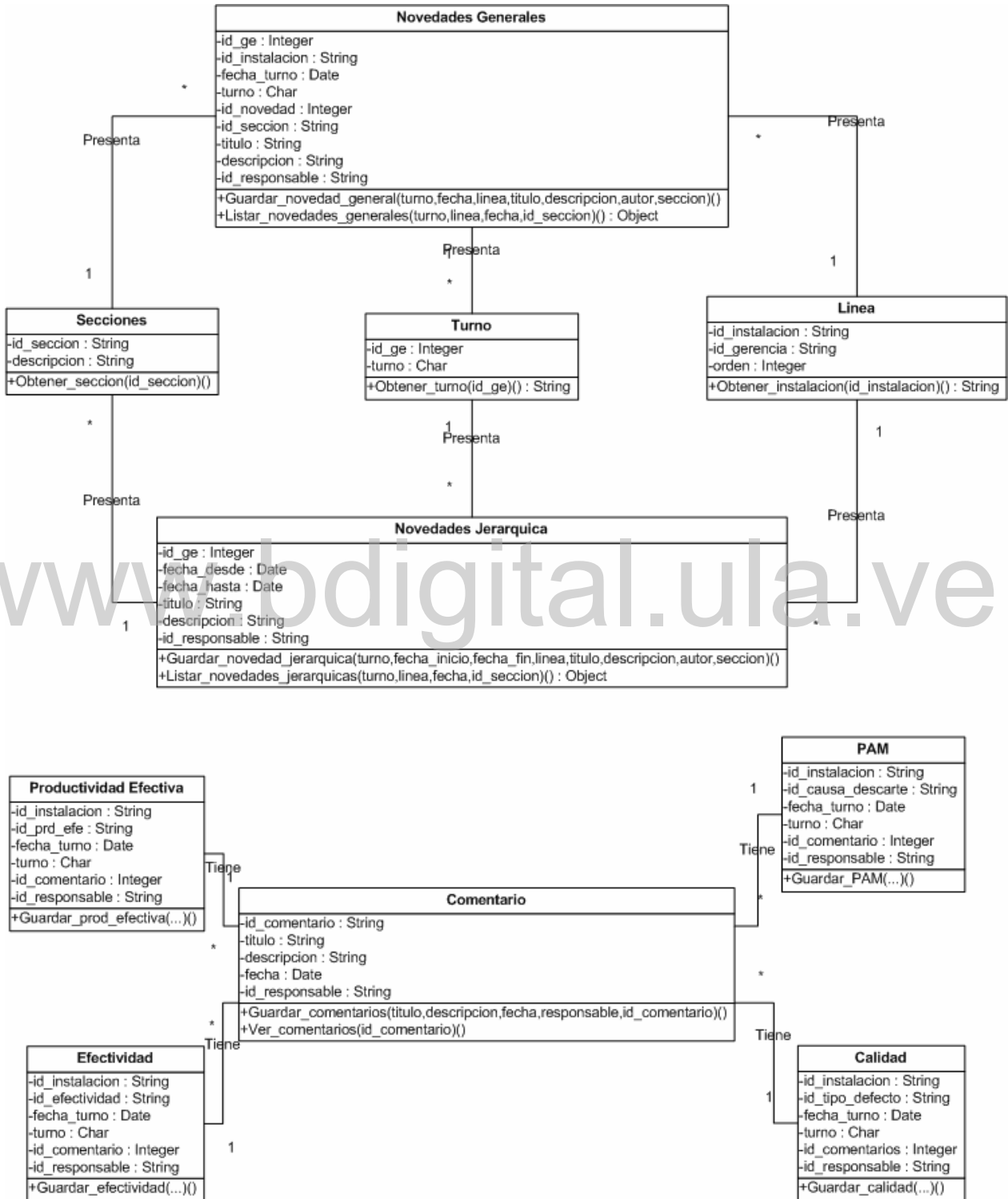


Figura 67: DCD para el LN, iteración II

Implementación

En cuanto a la implementación para esta iteración se procedieron a crear los procedimientos almacenados necesarios en SQL SERVER para la visualización posterior en las páginas Web correspondientes.

Pantallas:

Elaboración de las pantallas para la representación de los casos de uso. Como vimos en el caso del Libro de novedades no se adicionaron pantallas, en ves de ello se adicionaron funcionalidades a las pantallas existentes. Por ello el diagrama de navegabilidad presentado en la iteración I se mantiene. En el caso del SGD, se adiciono una nueva pantalla, que podemos ver en la figura:

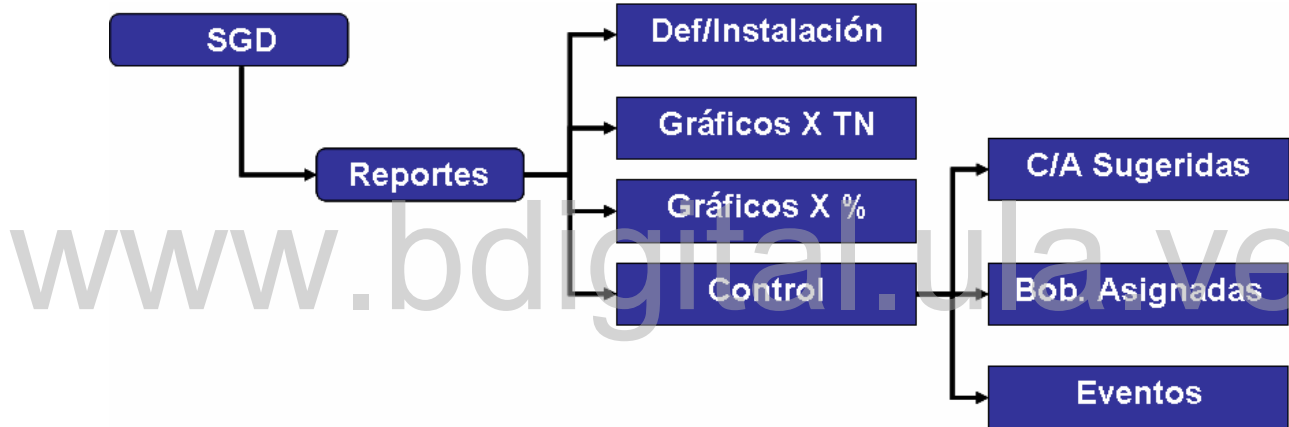


Figura 68: Navegación de las pantallas Iteración II

```

<?xml version="1.0" encoding="UTF-8" ?>
+ <!-- -->
- <Menu>
- <item>
+ <item caption="SIDOR" id="home_sgc" default="home_url_sgc">
  <item caption="APR" id="apr" target="_self" url="/sgl_defectos/browse/en_construccion.aspx" default="apr" />
  <item caption="APA" id="apa" target="_self" url="/sgl_defectos/browse/en_construccion.aspx" default="apa" />
  <item caption="APL" id="apl" target="_self" url="/sgl_defectos/browse/en_construccion.aspx" default="apl" />
- <item caption="LAC" id="lac" target="_self" default="lac_SGC_reportes_def">
  <item caption="Revisión Def." id="lac_SGC_casugeridas" url="/sgl_defectos/browse/bobinas_qnx.aspx?g=LAC"
    target="_self" />
+ <item caption="Comité" id="lac_comite" default="lac_SGC_Comite">
+ <item caption="Respons." id="lac_SGC_resp" default="lac_SGC_asig_resp">
+ <item caption="Gestión" id="lac_SGC_Gestion" default="lac_SGC_Gestion_basig" target="_self">
- <item caption="Reportes" id="lac_SGC_reportes" default="lac_SGC_reportes_def">
  <item caption="Def/Instalacion" id="lac_SGC_reportes_def"
    url="/sgl_defectos/browse/reportes_defecto_instalacion.aspx?g=LAC" target="_self" />
+ <item caption="Gráficos x Tn" id="lac_SGC_reportes_graficos_tn" target="_self"
  default="lac_SGC_reportes_tn_EV_Tiempo">
+ <item caption="Gráficos x %" id="lac_SGC_reportes_graficos_porc" target="_self"
  default="lac_SGC_reportes_porc_EV_Tiempo">
- <item caption="CONTROL" id="lac_SGC_control" target="_self" default="lac_SGC_Control_casug">
  <item caption="C/A Sugeridas" id="lac_SGC_Control_casug"
    url="/sgl_defectos/browse/reporte_ca_sugeridas.aspx?g=LAC" target="_self" />
  <item caption="Bobinas Asig" id="lac_sgc_Bobinas_Asig" url="/sgl_defectos/browse/Bobinas_Asignadas.aspx?
    g=LAC" target="_self" />
  <item caption="Eventos" id="lac_sgc_eventos" url="/sgl_defectos/browse/eventos.aspx?g=LAC"
    target="_self" />
</item>
</item>
<item caption="PCP" id="lac_SGC_PCP" url="/sgl_defectos/browse/reporte_pcp_laf.aspx?g=LAC" target="_self" />
<item caption="SICOP" id="lac_SGC_SICOP" url="/sgl_defectos/browse/sicop.aspx?g=LAC" target="_self" />
  
```

Figura 69: Menú XML para navegación del sistema

Iteración III

Para el desarrollo del Sistema de Gestión de Calidad se plantearon 3 iteraciones, durante las cuales se han obtenido productos parciales de la totalidad del sistema. En esta última iteración se espera concluir con el desarrollo, por lo cual incluye tanto el subsistema actual, como el refinamiento de los anteriores.

El taller de requisitos que se llevo a cabo para la elaboración de esta Iteración se centro en la presentación de dos nuevos reportes que permitirán cerrar el ciclo de gestión de bobinas. Estos son:

- PCP: Donde se muestra un listado de los cambios realizados por el comité de calidad en las bobinas, con el cual el personal de programación y gestión de órdenes podrá realizar los cambios solicitados por bobinas en las diferentes líneas.
- SICOP: En esta página se muestra el listado de bobinas que han sufrido cambios en el sistema y son actualizados en la base de datos general de la empresa.
- Datos de bobinas: visualización de los datos de una bobina en particular.

En estos casos procederemos a realizar los casos de uso correspondientes, los diagramas de secuencia e iteración para la compresión del sistema.

Para el libro de novedades, se sugiere la carga de novedades desde nuevas secciones, las cuales son:

- Ingeniería
- Mantenimiento
- Programación
- Insumos

A este particular, la incidencia con respecto al sistema ya presentado afecta la instancia secciones, en la cual se deberán agregar estos ítems, para la correcta carga de novedades, tanto jerárquicas como generales.

Como esta modificación no tiene incidencia en la aparición de nuevos objetos en el modelo de dominio, ni de clases del sistema, no se realizaran análisis en el diseño ya presentado. En otras palabras, para el libro de novedades, permaneceremos con el sistema presentado en la iteración II.

Modelo de Casos de Uso:

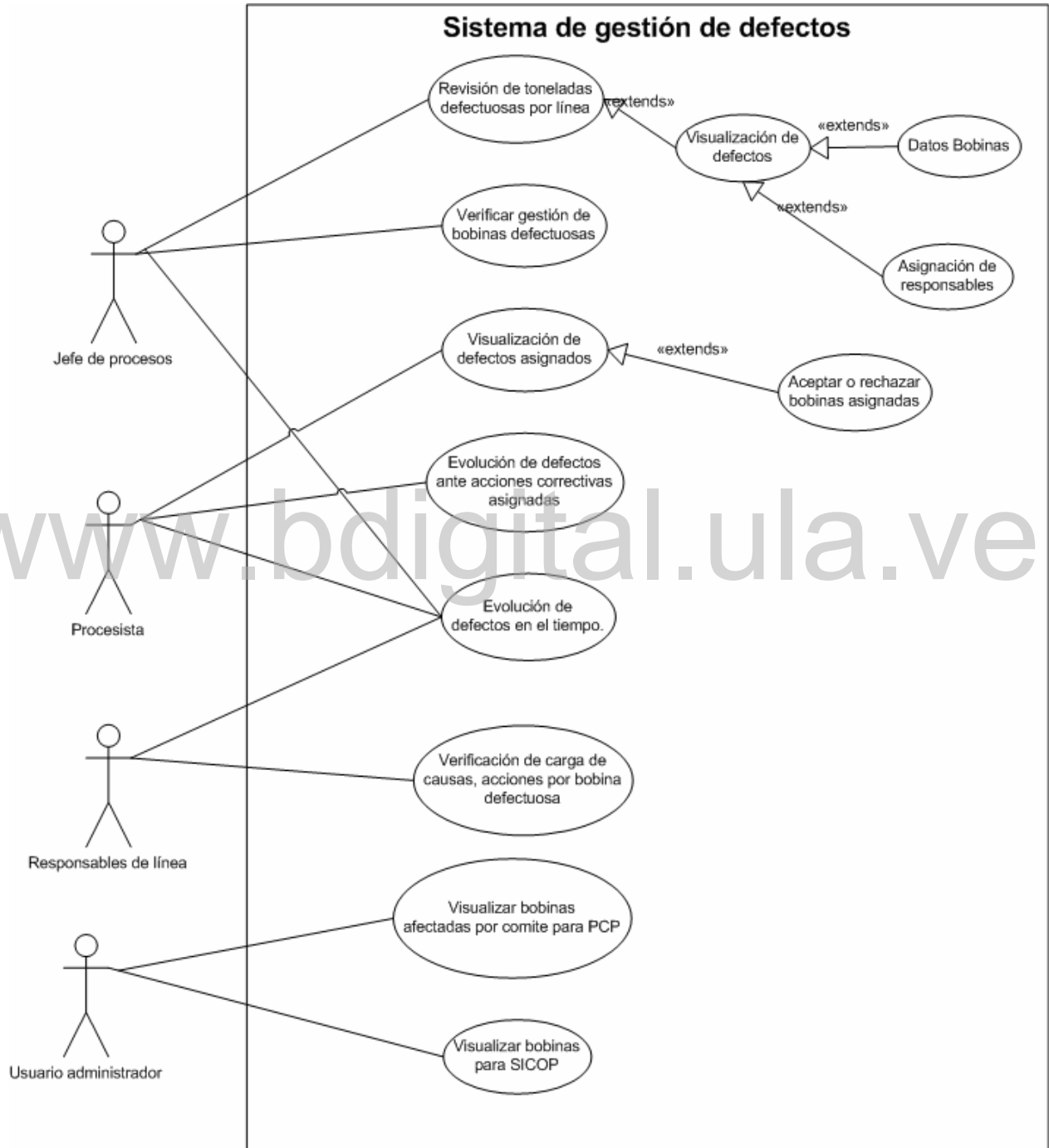


Figura 70: Modelo de casos de uso para el SGD, iteración III

Caso de uso: Datos bobinas

Escenario principal de éxito:	
<ul style="list-style-type: none"> El usuario accede al área a través del listado presentado en el caso de uso de defectos. Este provee al sistema de identificador de la bobina, la línea de origen. 	<ul style="list-style-type: none"> El sistema muestra los datos correspondientes a la bobina seleccionada.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Visualizar bobinas afectadas por comité para PCP

<p>Actor Principal: Usuario Administrador: Visualizar las bobinas que recibieron una nueva programación desde el comité de calidad.</p>	
Escenario principal de éxito:	
<ul style="list-style-type: none"> Usuario accede al sistema, selecciona el rango de fechas. El usuario selecciona el o los estados de calidad, para las bobinas defectuosas a mostrar. El usuario selecciona la línea de producción que desea consultar. El usuario puede repetir los pasos anteriormente, para cualquier cambio en la consulta de los datos. 	<ul style="list-style-type: none"> Se visualiza un listado de bobinas defectuosas, que han sido cambiadas por el comité.

Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> • Usuario escoge un rango de fechas incoherente. 	<ul style="list-style-type: none"> • El sistema genera un mensaje de error.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Visualizar bobinas para SICOP

<p>Actor Principal:</p> <p>Usuario administrador: Visualizar las bobinas a las cuales les han realizado cambios en la línea de producción, fecha de aplicación u otros.</p>	
Escenario principal de éxito:	
<ul style="list-style-type: none"> • Usuario accede al sistema, selecciona el rango de fechas. • El usuario selecciona las líneas. • El usuario puede repetir los pasos anteriormente, para cualquier cambio en la consulta de los datos. 	<ul style="list-style-type: none"> • Se visualiza un listado de bobinas defectuosas, que sufrieron cambios y han sido enviadas a SICOP para su actualización.
Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> • Usuario escoge un rango de fechas incoherente. 	<ul style="list-style-type: none"> • El sistema genera un mensaje de error.

Requisitos especiales:
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.

Visualizar manuales del SGD

Actor Principal:	
Todos los usuarios.	
Escenario principal de éxito:	
<ul style="list-style-type: none"> • Usuario accede al sistema. • El usuario escoge el manual a consultar. 	<ul style="list-style-type: none"> • Sistema genera un listado de los manuales disponibles del sistema. • El sistema abre el documento deseado.
Escenarios o flujos alternativos:	
<ul style="list-style-type: none"> • El usuario no tiene instalado los programas para apertura de documentos (Acrobat reader, office) 	<ul style="list-style-type: none"> • El sistema falla.
Requisitos especiales:	
Respuesta del sistema para la visualización de datos no mayor a 30 segundos.	

Elaboración de diagramas de secuencia:

En esta parte de la elaboración realizaremos los diagramas de secuencia generados de los nuevos casos de uso definidos en el taller de requisitos del sistema, se omitirán los DSS que previamente se ha definido en la iteración I, y en la iteración II.

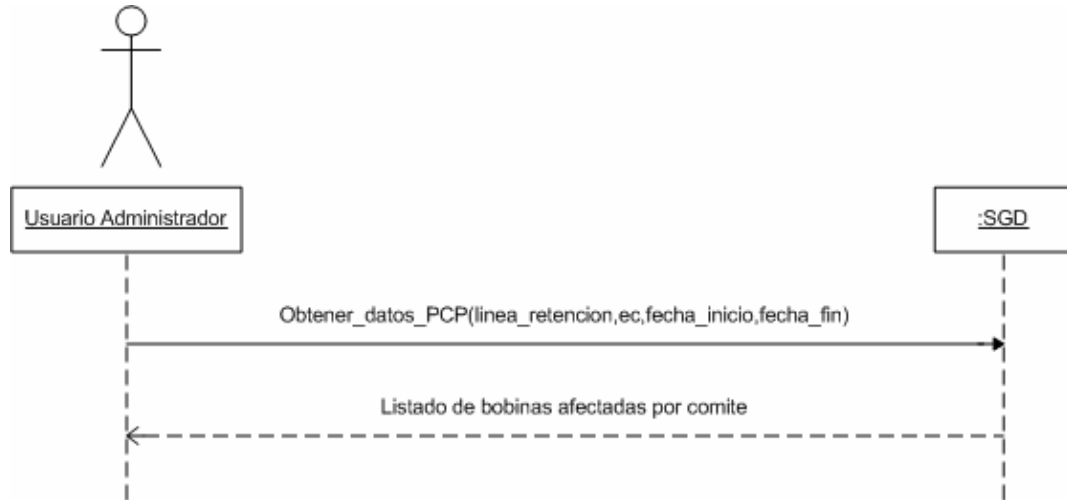


Figura 71: DSS Visualizar bobinas afectadas por comité para PCP

DSS Visualizar bobinas afectadas por comité para PCP

En este diagrama de secuencia, vemos la solicitud realizada por el usuario, en la que el escoge un periodo de tiempo, una línea de producción, el estado de calidad de las bobinas a revisar y el sistema genera el listado esperado.

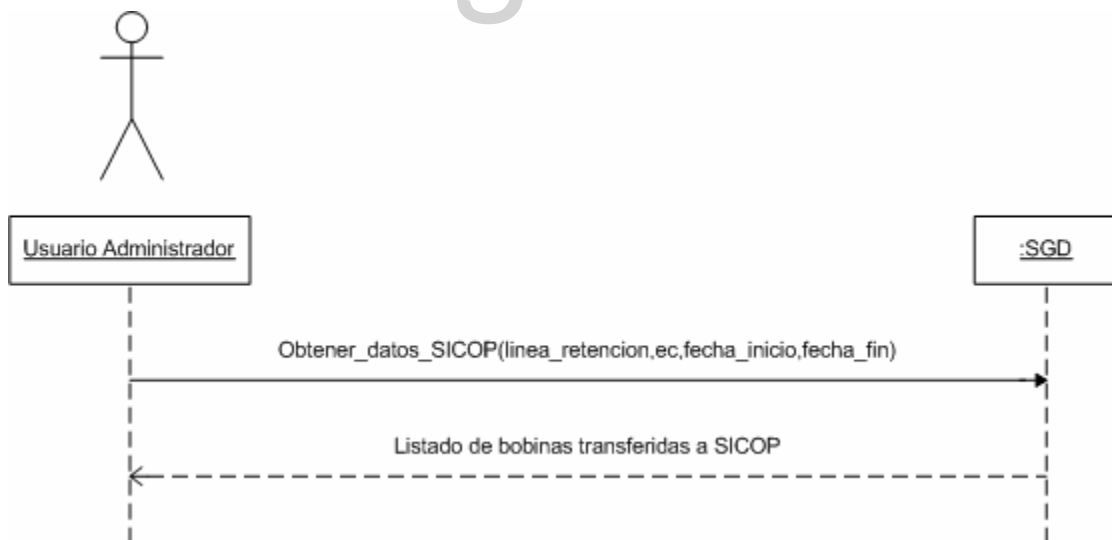


Figura 72: DSS Visualizar bobinas para SICOP

DSS Visualizar bobinas para SICOP

En este diagrama de secuencia el usuario escoge un periodo de tiempo (fecha inicio y fecha fin), una línea de producción, el estado de calidad de las bobinas a revisar y el sistema genera el listado de bobinas que han sido transferidas a SICOP.

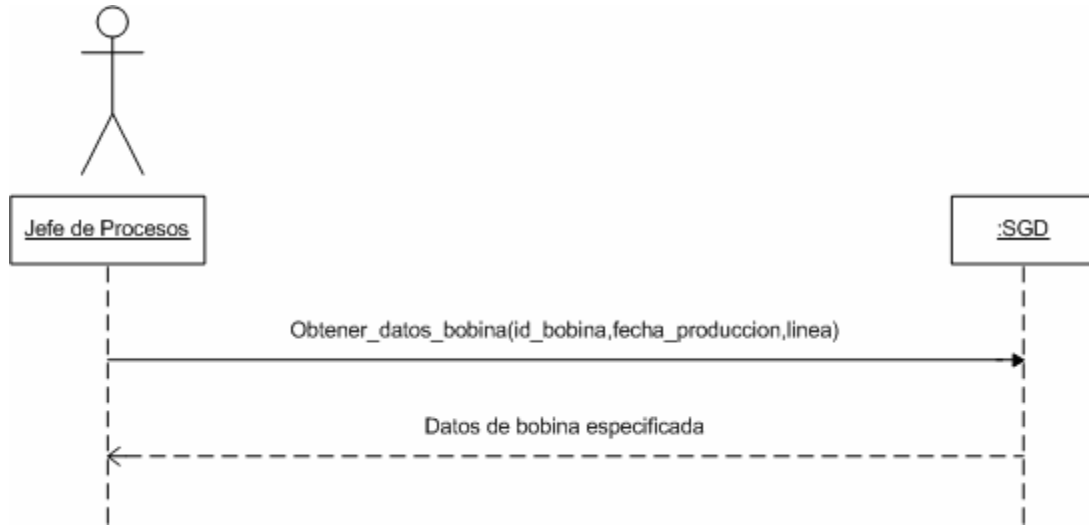


Figura 73: DSS Datos bobinas

DSS Datos bobinas:

Acá, la solicitud del usuario se hace indirectamente a través del DSS Visualización de defectos, desde el cual se accede a ver los datos de una bobina en particular, haciendo la solicitud al sistema con los identificadores de la bobina. El resultado obtenido es un listado de los datos de la bobina.

Modelo de dominio

Como especificamos anteriormente en esta etapa del diseño no es necesario la inclusión de nuevas clases, así que el modelo de la iteración II varia.

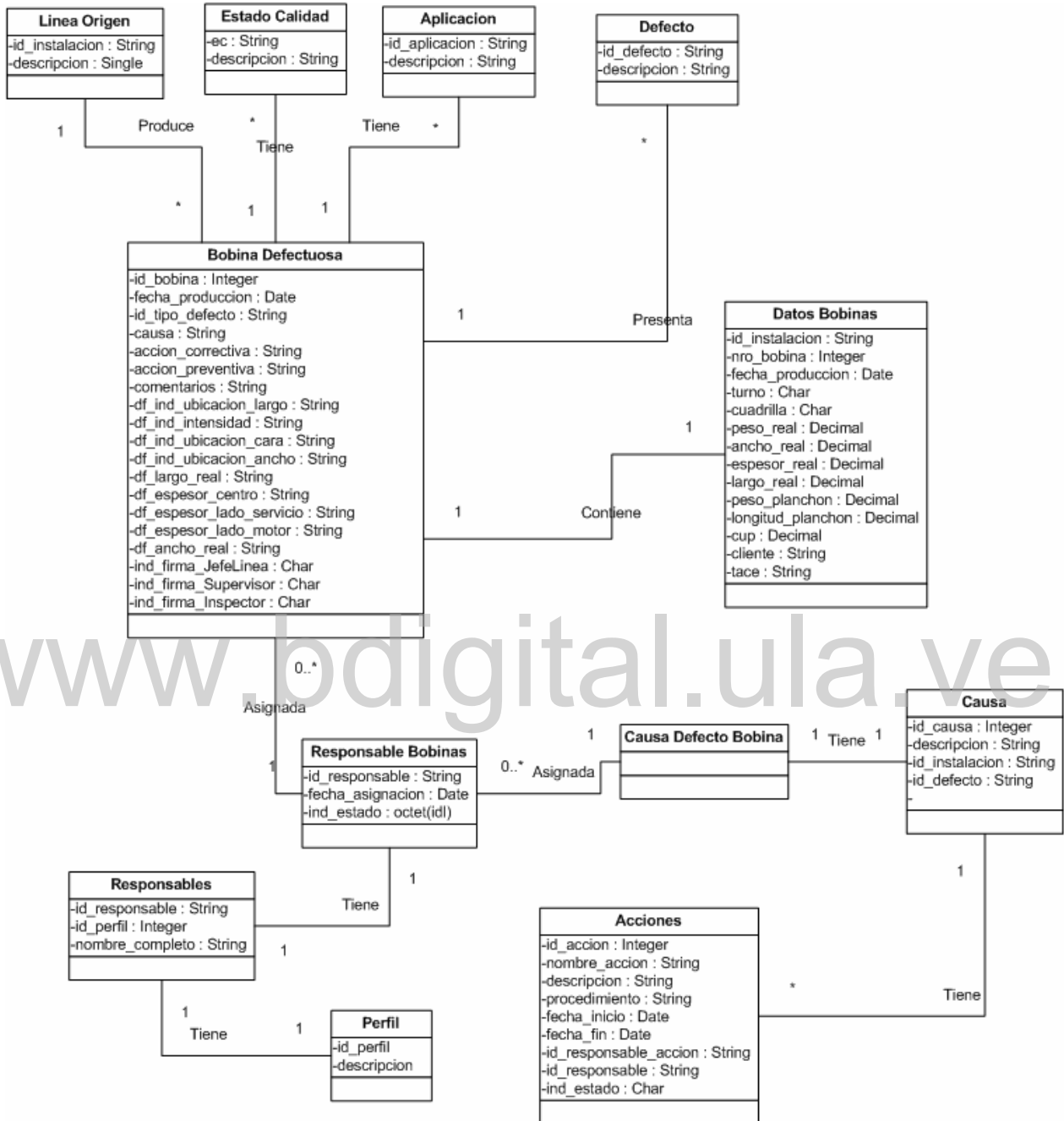


Figura 74: Modelo de dominio SGD para la iteración III

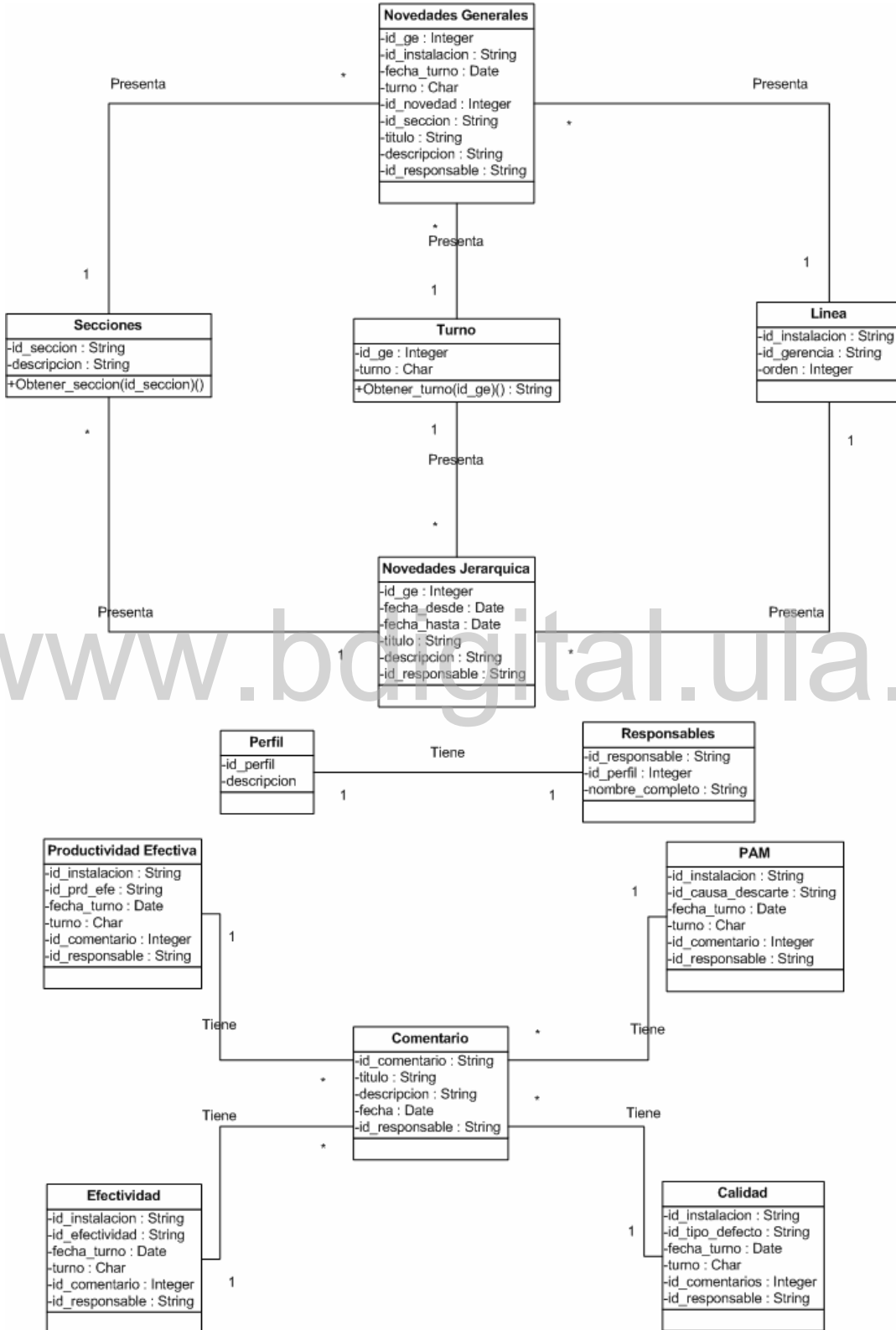


Figura 75: Modelo de dominio LN para la iteración III

Modelo del diseño:

Generalmente el objetivo de la última iteración del UP es el refinamiento del sistema, esto incluye además de la estructura básica de cada iteración, la revisión y la unificación el material elaborado.

DI Visualizar bobinas afectadas por comité para PCP

La colección recibe la llamada obtener datos PCP y genera la siguiente secuencia de mensajes:

1. La *colección de bobinas* envía el mensaje obtener datos bobina, del objeto *datos bobinas*.
2. La *colección de bobinas* envía el mensaje obtener línea, al objeto *Línea origen*.
3. La *colección de bobinas* envía el mensaje obtener estado de calidad, al objeto *estado de calidad*.

Finalmente, se obtiene el listado de bobinas afectadas por el comité para PCP.

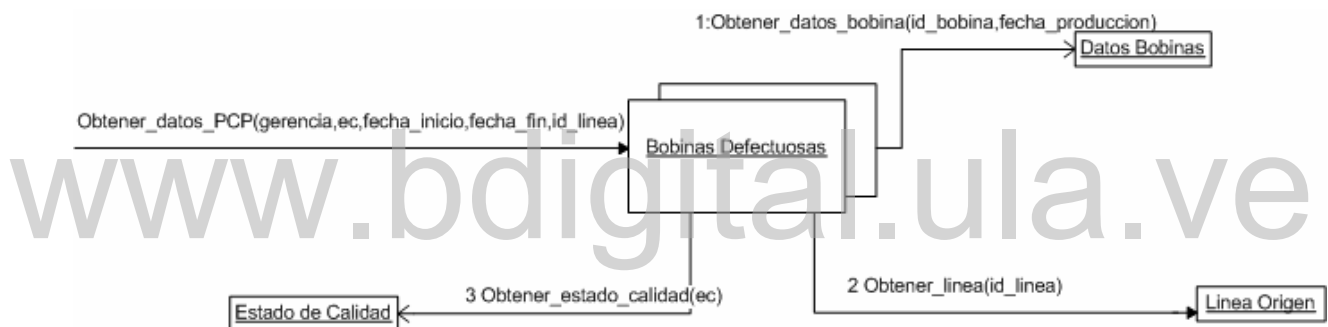


Figura 76: DI Visualizar bobinas afectadas por comité para PCP

DI Visualizar bobinas para SICOP

La colección recibe la llamada obtener datos SICOP y genera la siguiente secuencia de mensajes:

1. La *colección de bobinas* envía el mensaje obtener datos bobina, del objeto *datos bobinas*.
2. La *colección de bobinas* envía el mensaje obtener línea, al objeto *Línea origen*.
3. La *colección de bobinas* envía el mensaje obtener estado de calidad, al objeto *estado de calidad*.

Finalmente, se obtiene el listado de bobinas enviadas a SICOP.

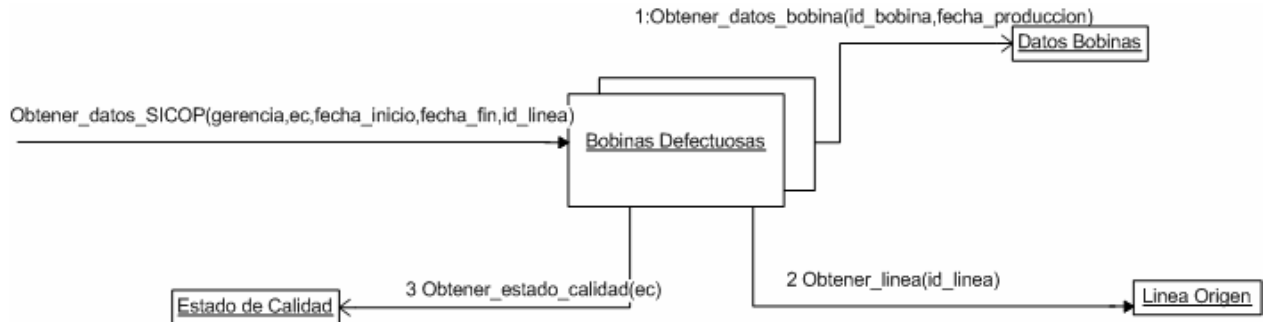


Figura 77: DI Visualizar bobinas para SICOP

DI Datos bobinas

En este diagrama se realiza la solicitud al objeto Bobinas Defectuosas, `Obtener_datos_bobina(...)`, donde el mismo realiza la llamada a la colección de los datos particulares por bobina, y se obtiene como respuesta el listado de todas las características que identifican una bobina producida en las líneas.

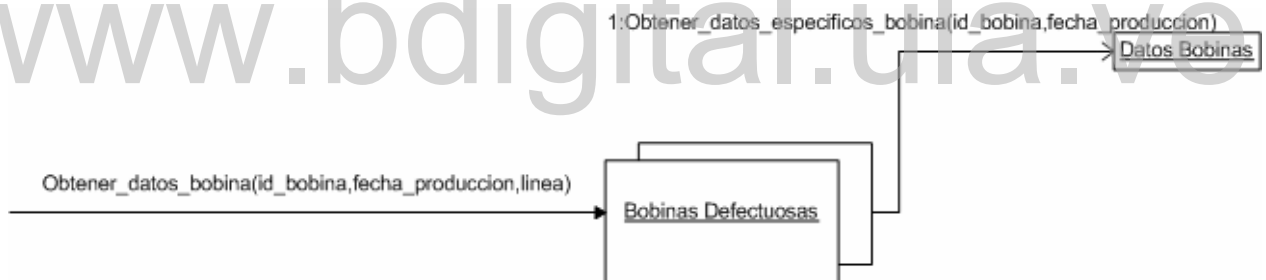


Figura 78: DI Datos Bobinas

Diagramas de clases de diseño (DCD)

Para este caso la incidencia sobre las clases previamente definidas se basa en las nuevas llamadas realizadas a cada instancia del sistema.

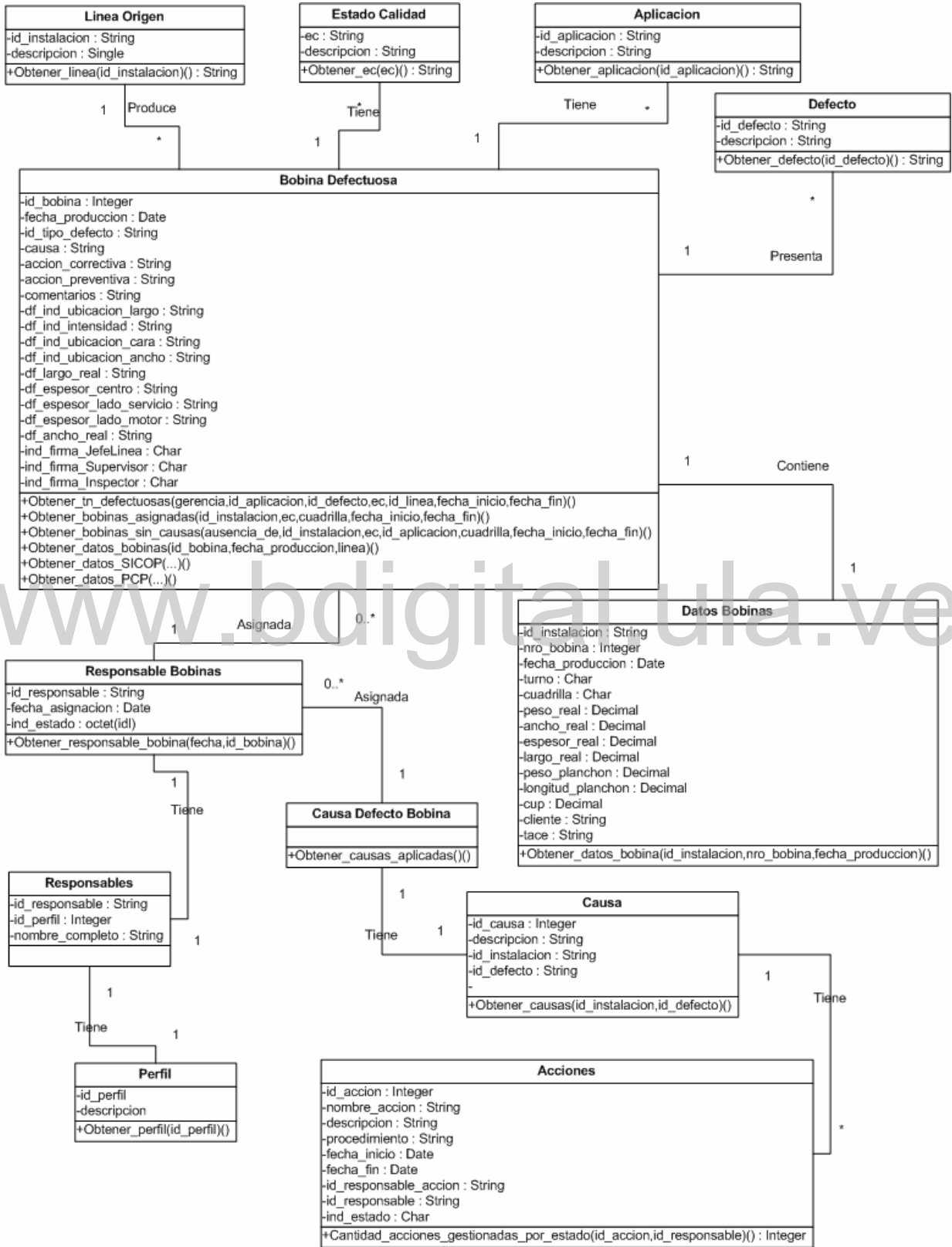


Figura 79: DCD del SGD para la iteración III

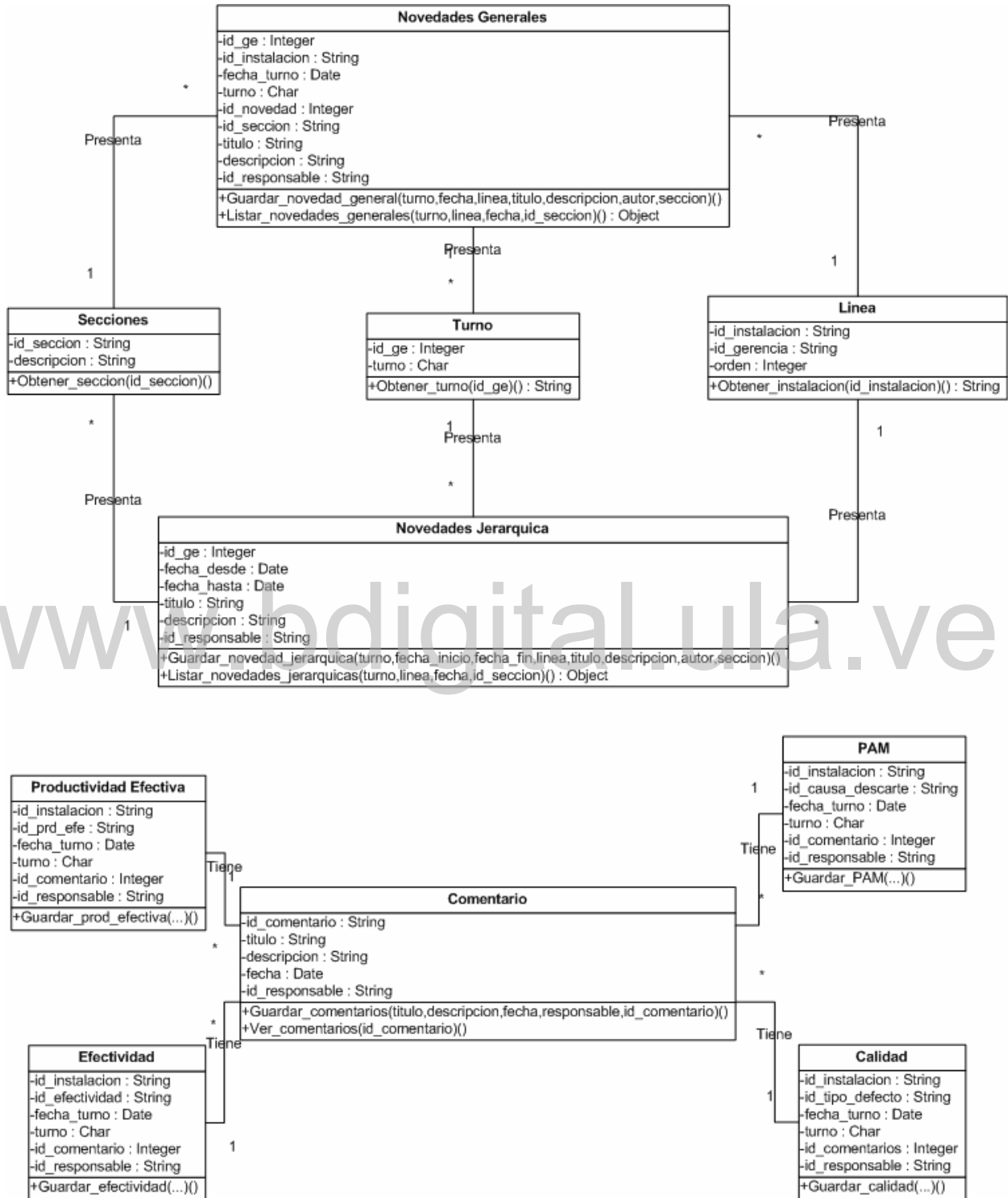


Figura 80: DCD para el LN, iteración III

Implementación

En cuanto a la implementación para esta iteración se procedieron a crear los procedimientos almacenados necesarios en SQL SERVER para la visualización posterior en las páginas Web correspondientes, tal como se realizó para la iteración I, y II.

Pantallas:

Elaboración de las pantallas para la representación de los casos de uso. En el caso del SGD, se adiciono dos pantallas, que podemos ver en la figura:

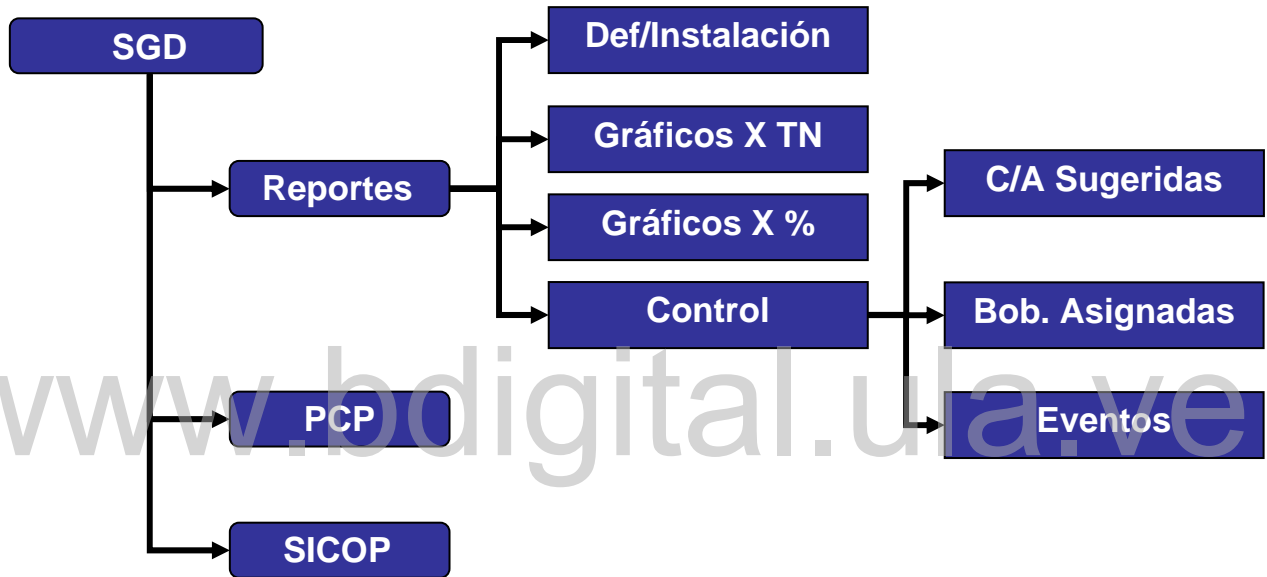


Figura 81: Navegación de las pantallas SGC Iteración III

```

<?xml version="1.0" encoding="UTF-8" ?>
+ <!-- -->
- <Menu>
- <item>
+ <item caption="SIDOR" id="home_sgc" default="home_url_sgc">
<item caption="APR" id="apr" target="_self" url="/sgl_defectos/browse/en_construccion.aspx" default="apr" />
<item caption="APA" id="apa" target="_self" url="/sgl_defectos/browse/en_construccion.aspx" default="apa" />
<item caption="APL" id="apl" target="_self" url="/sgl_defectos/browse/en_construccion.aspx" default="apl" />
- <item caption="LAC" id="lac" target="_self" default="lac_SGC_reportes_def">
<item caption="Revisión Def." id="lac_SGC_casugeridas" url="/sgl_defectos/browse/bobinas_qnx.aspx?g=LAC"
target="_self" />
+ <item caption="Comité" id="lac_comite" default="lac_SGC_Comite">
+ <item caption="Respons." id="lac_SGC_resp" default="lac_SGC_asig_resp">
+ <item caption="Gestión" id="lac_SGC_Gestion" default="lac_SGC_Gestion_basig" target="_self">
- <item caption="Reportes" id="lac_SGC_reportes" default="lac_SGC_reportes_def">
<item caption="Def/Instalacion" id="lac_SGC_reportes_def"
url="/sgl_defectos/browse/reportes_defecto_instalacion.aspx?g=LAC" target="_self" />
+ <item caption="Gráficos x Tn" id="lac_SGC_reportes_graficos_tn" target="_self"
default="lac_SGC_reportes_tn_EV_Tiempo">
+ <item caption="Gráficos x %" id="lac_SGC_reportes_graficos_porc" target="_self"
default="lac_SGC_reportes_porc_EV_Tiempo">
- <item caption="CONTROL" id="lac_SGC_control" target="_self" default="lac_SGC_Control_casug">
<item caption="C/A Sugeridas" id="lac_SGC_Control_casug"
url="/sgl_defectos/browse/reporte_ca_sugeridas.aspx?g=LAC" target="_self" />
<item caption="Bobinas Asig" id="lac_sgc_Bobinas_Asig" url="/sgl_defectos/browse/Bobinas_Asignadas.aspx?
g=LAC" target="_self" />
<item caption="Eventos" id="lac_sgc_eventos" url="/sgl_defectos/browse/eventos.aspx?g=LAC"
target="_self" />
</item>
</item>
<item caption="PCP" id="lac_SGC_PCP" url="/sgl_defectos/browse/reporte_pcp_laf.aspx?g=LAC" target="_self" />
<item caption="SICOP" id="lac_SGC_SICOP" url="/sgl_defectos/browse/sicop.aspx?g=LAC" target="_self" />
  
```

Figura 82: Menú XML para navegación del sistema SGD.

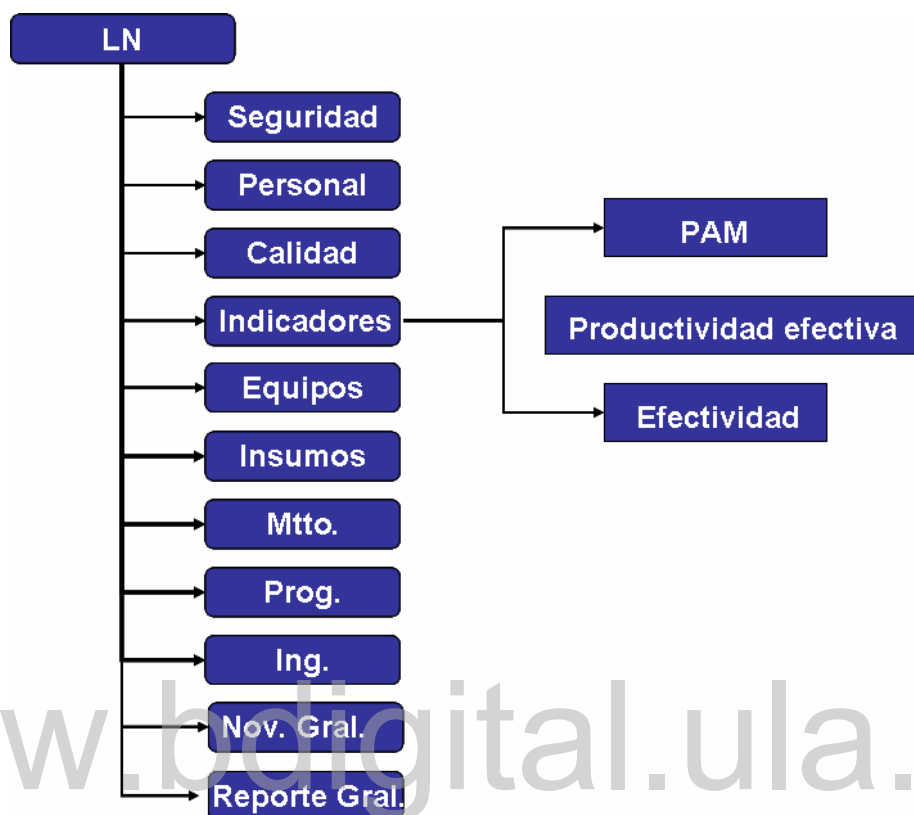


Figura 83: Navegación de las pantallas LN Iteración III

www.digital.ula.ve

```

- <Menu>
- <item>
+ <item caption="SIDOR" id="home_ln" default="home_url_ln">
  <item caption="APR" id="apr" target="_self" url="/sgl_ln/en_construccion.aspx" default="serv" />
  <item caption="APA" id="apa" target="_self" url="/sgl_ln/en_construccion.aspx" default="serv" />
  <item caption="APL" id="apl" target="_self" url="/sgl_ln/en_construccion.aspx" default="serv" />
+ <item caption="LAC" id="lac" default="lac_reporte_general">
- <item caption="LAF" id="laf" default="laf_reporte_general">
  <item caption="Seguridad" id="laf_ln_seg" url="/sgl_ln/novedades_inicio.aspx?g=laf&tipo_novedad=seg"
    target="_self" />
  <item caption="Personal" id="laf_ln_per" url="/sgl_ln/novedades_inicio.aspx?g=laf&tipo_novedad=per" target="_self" />
  <item caption="Calidad" id="laf_ln_cal" url="/sgl_ln/calidad.aspx?g=laf" target="_self" />
- <item caption="Indicadores" id="laf_ln_ind" default="laf_pam">
  <item caption="PAM" id="laf_pam" url="/sgl_ln/pam.aspx?g=laf" target="_self" />
  <item caption="Productividad Efectiva" id="laf_produc_efec" url="/sgl_ln/productividad_efectiva.aspx?g=laf"
    target="_self" />
  <item caption="Efectividad" id="laf_efectividad" url="/sgl_ln/efectividad.aspx?g=laf" target="_self" />
</item>
<item caption="Programa" id="laf_ln_prg" url="/sgl_ln/novedades_inicio.aspx?g=laf&tipo_novedad=prg"
  target="_self" />
<item caption="Insumos" id="laf_ln_ins" url="/sgl_ln/novedades_inicio.aspx?g=laf&tipo_novedad=ins" target="_self" />
<item caption="Equipos" id="laf_ln_equ" url="/sgl_ln/equipos.aspx?g=laf" target="_self" />
<item caption="ING." id="laf_ing" url="/sgl_ln/novedades_inicio.aspx?g=laf&tipo_novedad=ing" target="_self" />
<item caption="Mantto" id="laf_mant" url="/sgl_ln/novedades_inicio.aspx?g=laf&tipo_novedad=manto" target="_self" />
- <item caption="Reportes" id="laf_reporte" default="laf_reporte_general">
  <item caption="General" id="laf_reporte_general" url="/sgl_ln/reporte.aspx?g=laf" target="_self" />
  <item caption="Novedades" id="laf_reporte_novedades" url="/sgl_ln/reporte_novedades.aspx?g=laf"
    target="_self" />
</item>
<item caption="Nov. Gral." id="laf_novedad_gral" url="/sgl_ln/novedades_generales.aspx?g=laf" target="_self" />
</item>
<item caption="BYA" id="bya" target="_self" url="/sgl_ln/en_construccion.aspx" default="serv" />
<item caption="SERV" id="serv" target="_self" url="/sgl_ln/en_construccion.aspx" default="serv" />
</item>
</Menu>

```

Figura 84: Menú XML para navegación del sistema LN.

Capítulo V

La Fase de Construcción es la tercera fase del Proceso Unificado para el Desarrollo del Software, cuyo propósito primordial es el de obtener un producto software en su versión operativa inicial, denominada “versión beta”. Dicho producto deberá contar la calidad adecuada para su aplicación y cumplir con todos los requisitos establecidos previamente.

En esta fase se realizó básicamente la codificación y pruebas al sistema, para ello se refinó la arquitectura del sistema a medida que se construía, es decir, se llevó a cabo las etapas de implementación y pruebas.

Una vez que los subsistemas fueron programados y probados, se procedió a realizar la integración de los mismos, hasta que se obtuvo el sistema completo.

El flujo de pruebas fue realizado tanto a nivel de unidad como a nivel de integración, con el fin de encontrar errores y poder corregirlos, y así obtener una versión del sistema apropiada para funcionar en la LAN de la empresa.

También se realizó la documentación del sistema desarrollado y al manejo del mismo, obteniendo en esta fase un software construido y su documentación.

Diseño e interfaz de usuario:

Uno de los aspectos más importantes del proceso de desarrollo de software es el diseño de la interfaz de usuario, su importancia radica en que, por medio de esta se hace posible la interacción entre el usuario y el sistema.

Es importante señalar que el proceso de diseño de interfaz de usuario fue realizado con el fin de facilitar la interacción usuario – máquina, de manera que dichos usuarios puedan utilizar el sistema de manera natural y sencilla, es por esto que en el diseño de la interfaz se tomaron en

consideración aspectos como: las personas que van a manejar el software. La interfaz será un fracaso si el sistema no hace lo que los usuarios necesitan, en otras palabras, el sistema debe permitir a los usuarios cumplir con sus tareas (el diseño se centra en las tareas del usuario).

También es importante entender quien usará el sistema para hacer que, lo que se conoce como “**análisis de usuarios y tareas**”, en donde la necesidad de análisis de tareas se debe a que si se construye un gran sistema que no hace lo que el usuario necesita, muy probablemente será un fracaso. Pero, más aún que un simple “haz lo que se necesita”, un sistema exitoso debe de integrarse fácilmente al mundo en donde se encuentra el usuario y a su trabajo.

La interfaz de usuario se compone de un grupo de ventanas, cada una permite una opción específica, facilitando de esta manera el uso del sistema, se presentan como respuesta a eventos provocados por el usuario, es decir, al ser accionadas por el uso del Mouse o el teclado.

A continuación se describe todos los aspectos relacionados con la interfaz de usuario adoptada para el sistema, se especifican todos los detalles pertinentes sobre el diseño de las pantallas de la aplicación, mensajes de error provocados por el usuario, notificación de fallas del sistema, mensajes de diálogos entre otros.

Diseño de reportes del sistema para el SGD

Para el desarrollo de la interfaz del Sistema de Gestión de Calidad debe regirse al igual que todos los proyectos de del área de Automatización Nivel 2B por el conjunto de estándares relacionados a la navegación y visualización de las pantallas Web desarrolladas en ASP.NET mediante la herramienta Microsoft Visual Studio .NET. Entre estos se tiene en primer lugar la utilización de un menú creado a tiempo de ejecución mediante un control de usuario y utilizando un archivo de configuración XML. En segundo lugar se tiene todo lo relacionado a la visualización de los controles tales como botones, cajas de texto, listas desplegables, tablas, entre otros.

Siguiendo los estándares, se colocan los botones de filtros para el usuario en el área superior, y en el área inferior de la pantalla se muestran los resultados de los datos consultados.

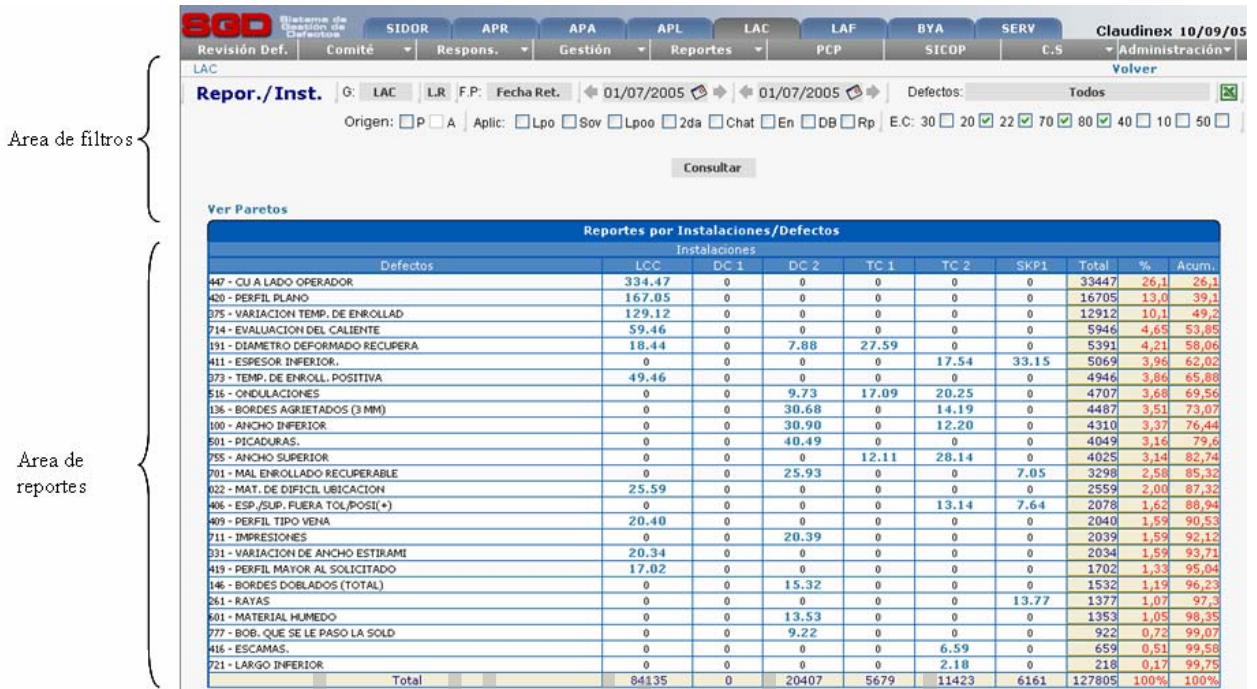


Figura 85: Diseño de reportes del SGD.

Diseño de páginas del LN.

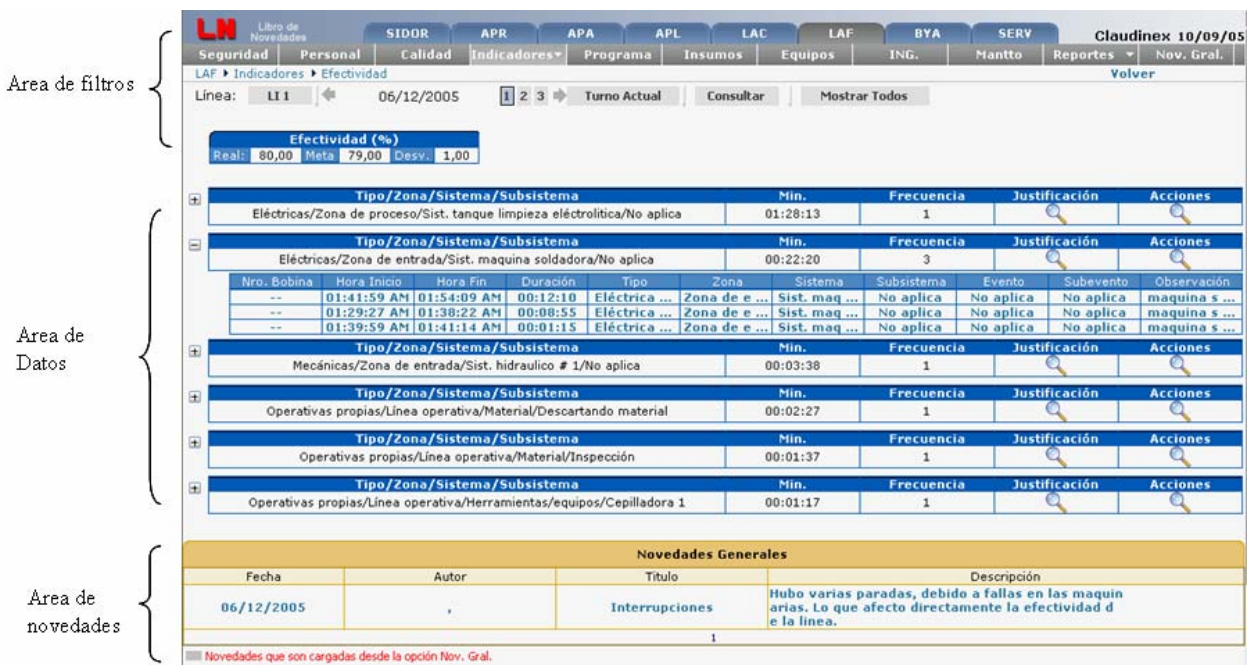
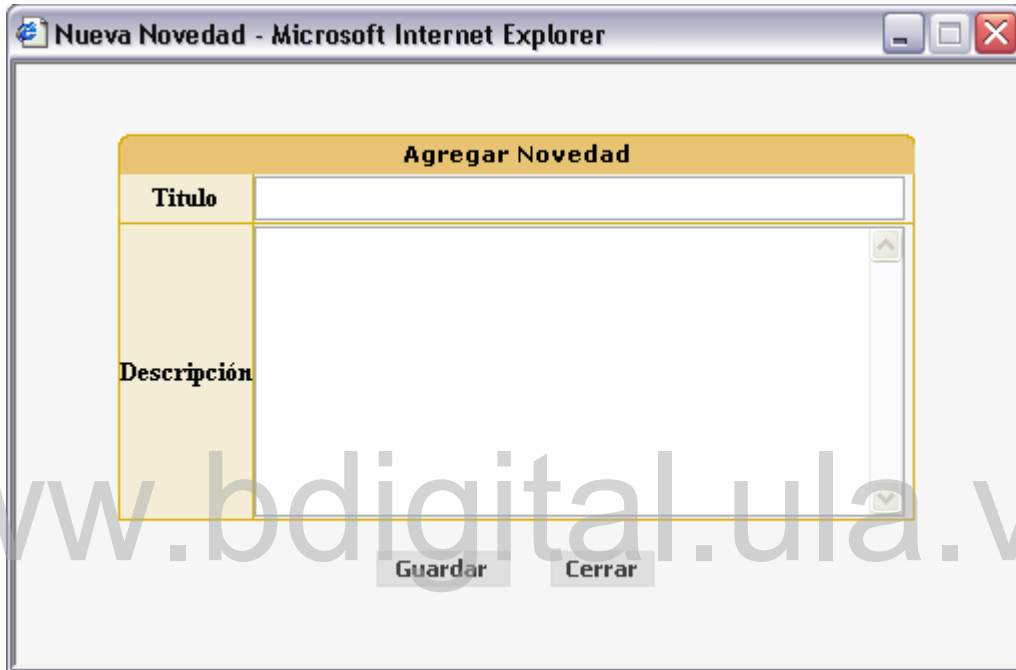


Figura 86: Diseño de páginas de datos del LN

Las pantallas del LN presentan la misma estructura de acuerdo al estándar de la empresa, por lo que solo presentamos el diseño de esta página, donde vemos la colección de los datos obtenidos, y la forma de visualización de las novedades generales. Otras pantallas diferentes del sistema, lo corresponden la pantalla de carga y modificación de novedades que presentamos a continuación:



Nueva Novedad - Microsoft Internet Explorer

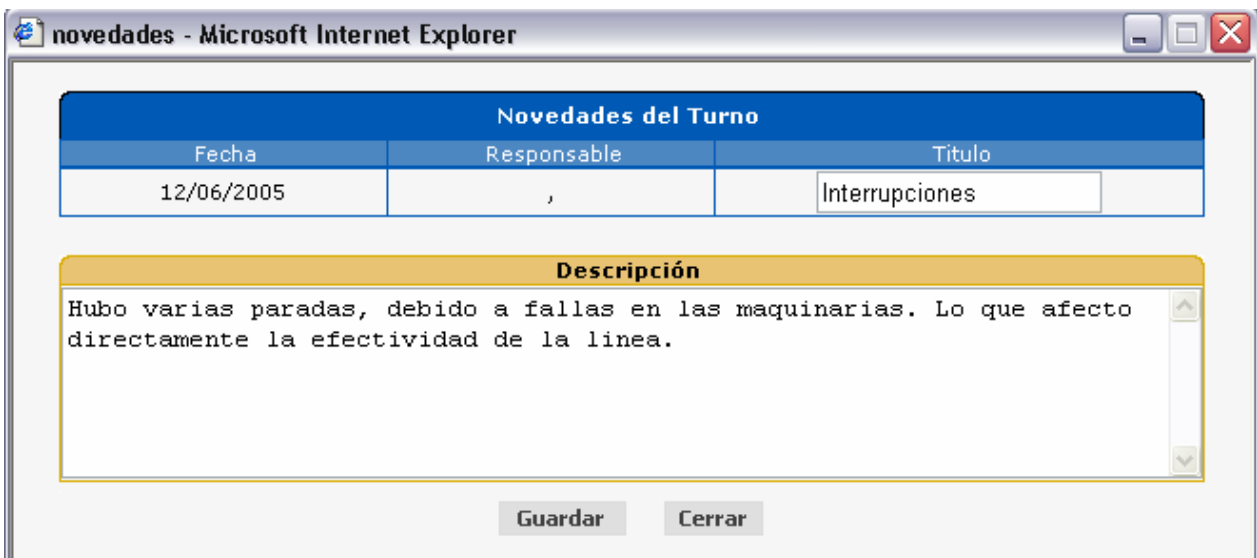
Agregar Novedad

Titulo

Descripción

Guardar **Cerrar**

Figura 87: Crear novedad



novedades - Microsoft Internet Explorer

Novedades del Turno

Fecha	Responsable	Titulo
12/06/2005	,	Interrupciones

Descripción

Hubo varias paradas, debido a fallas en las maquinarias. Lo que afecto directamente la efectividad de la linea.

Guardar **Cerrar**

Figura 88: Modificar novedad

Conexión entre capas y aplicación del sistema:

Básicamente el sistema se desenvuelve en tres capas, la capa del cliente, la capa de negocios, y la capa de datos.

La capa del cliente corresponde a la interfaz a través de la cual el usuario se conecta al sistema e interactúa con el mismo para realizar las peticiones deseadas. En esta capa se manejan todos los eventos de los elementos de la página, como por ejemplo el evento clic de los botones, u otros, que al ser solicitados ejecutan el código desarrollado en Vb.Net.

Capa de negocios, esta concierne a la comunicación entre el código en .net y la capa de datos. Como estándar en la empresa se mantiene una clase que posee las llamadas a los procedimientos almacenados usados para la presentación de la información.

La capa de datos, es aquella donde se encuentra la información almacenada. Tanto procedimientos, como vistas y funciones.

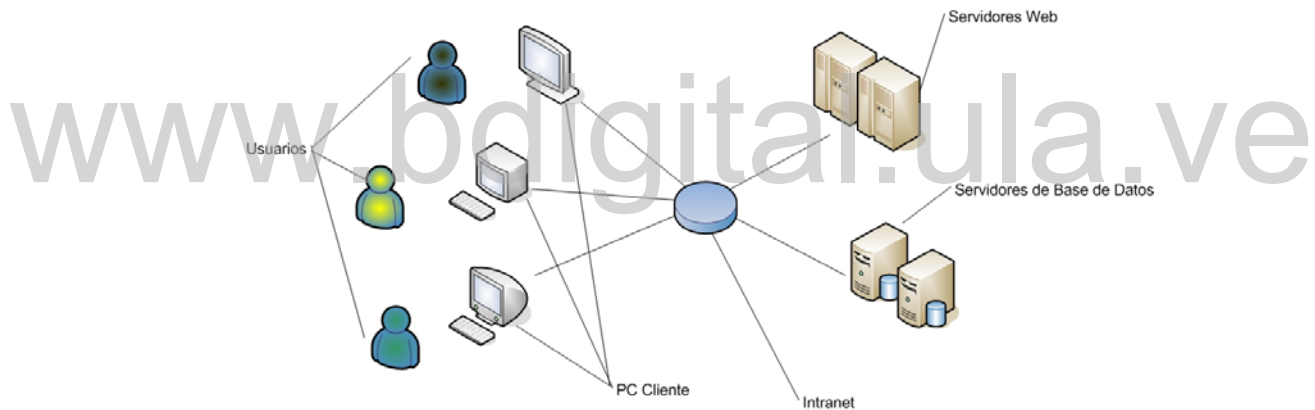


Figura 89: Configuración de capas con la aplicación del sistema

Diagrama de implementación

El diagrama de implementación representa la distribución física de los equipos, acá nuevamente podemos ver la fisiología de las tres capas, en las que la capa de presentación o cliente (Intranet) es comunicada a la capa de datos, la cual está ubicada en los servidores de bases de datos, lo cual se realiza mediante la capa de negocios, lo cual está representado por la capa de negocios que está ubicada en el servidor Web.

A continuación presentamos un diagrama que permite ver la relación de las tres capas:

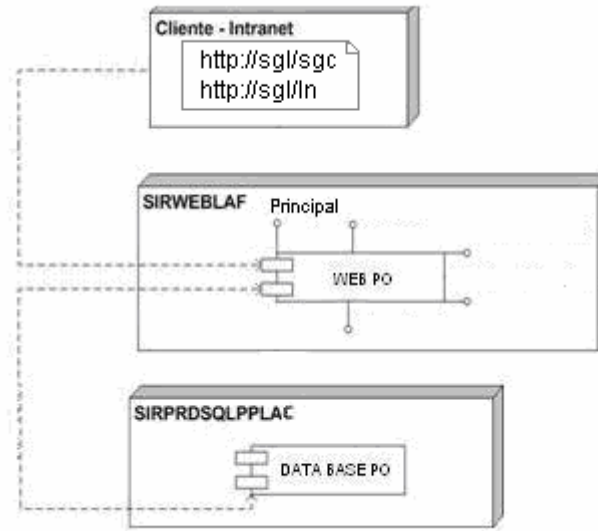


Figura 90: Diagrama de implementación.

Pruebas Finales del Sistema.

Las Pruebas Finales del sistema tienen como finalidad el refinamiento del sistema, mediante la verificación de los resultados esperados. La idea de realizar estas pruebas finales es el de hacer correcciones relativos a la puesta en marcha del proyecto. Para nuestro sistema se realizaron las siguientes pruebas:

Pruebas Caja Blanca: Los datos de prueba son derivados del comportamiento interno del sistema, puede decirse, con resultados esperados satisfactorios.

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, donde probablemente ocurran excepciones.

La ventaja de la realización de estas pruebas permite que se depure el código implementado para una mayor satisfacción de los clientes.

Conclusiones

La utilización del proceso unificado en el desarrollo de sistemas de información permite disponer de un marco de fases, iteraciones y evaluaciones que aportan los puntos de apoyos claves en los cuales los usuarios puede generar ideas, e interactuar con el sistema que posteriormente van a usar. Esta forma de trabajo permite que se prevean y corrijan situaciones de riesgo durante la arquitectura, diseño y desarrollo del mismo.

Usando los modelos visuales expresados en el Lenguaje Unificado de Modelado, nos permite hacer reutilización de software y esquemas de software. El uso del proceso unificado y de un lenguaje estándar dota del soporte financiero para herramientas más complejas, lo que hace que el proceso sea más efectivo y adaptable ya que no es un proceso rígido.

A este respecto durante el desarrollo del área de reportes del sistema de gestión de defectos, y el libro de novedades hubo interacción directa con el cliente a lo largo del desarrollo, lo que permitió que los mismos tuvieran conocimiento del progreso del sistema, y aportaran ideas al mismo y al final de la aplicación estuvieran satisfechos con los resultados obtenidos.

A su vez, como se hizo amplio uso de los diferentes modelos visuales permite que el sistema se pueda entender por otros desarrolladores, y los mismos puedan realizar el mantenimiento necesario e incluso hacer extensiones en caso de que los usuarios lo requieran.

Para el contexto del desarrollo del software, este método iterativo permitió que se llevara a buen término los objetivos inicialmente planteados.

Bibliografía

[ELMASRI 2000] Ramez Elmasri. “Sistemas de Base de Datos. Conceptos Fundamentales”. Pearson Educación. Segunda Edición. 2000.

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

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

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

[PRESSMAN 1998] Roger S. Pressman “Ingeniería del Software. Un Enfoque Práctico”. Cuarta Edición. McGraw - Hill.

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

Anexos

Corporación SIDOR

En la historia del desarrollo industrial venezolano, es importante recordar la fecha en que se decidió invertir en la creación de una planta siderúrgica de gran magnitud, en Guayana, ya que la meta a seguir en aquellos principios era dar un aporte significativo a la consolidación de Guayana como alternativa económica del país. El desarrollo del hierro y del acero se convirtió en el objetivo principal de la industria básica nacional. El plan industrial permitiría la diversificación de nuestra base productiva al agregar valor a los recursos naturales y vincular sus productos a la construcción, la manufactura y el petrolero, entre otros sectores.

En el presente aquellas metas se convierten en una realidad, ya que hoy en día se puede mostrar a la Siderúrgica del Orinoco (SIDOR) como una planta en operación cuya meta es mantener y superar los estándares de calidad y eficiencia productiva que exige la industria, por una parte; y por la otra, contribuir al crecimiento y expansión del producto territorial. De tal forma que cada uno de los integrantes de la familia SIDOR tiene la responsabilidad de cumplir con dichos objetivos.

Ubicación Geográfica de SIDOR:

La empresa se encuentra ubicada en la Zona Industrial Matanzas, Ciudad Guayana, Estado Bolívar - Venezuela, sobre la margen derecha del Río Orinoco, a 17 kilómetros de su confluencia con el Río Caroní y a 300 kilómetros de la desembocadura del Río Orinoco en el Océano Atlántico. Está conectada con el resto del país por vía terrestre, y por vía fluvial - marítima con el resto del mundo. Se abastece de energía eléctrica generada en las represas de Macagua y Gurí, ubicadas sobre el Río Caroní, así como de gas natural, proveniente de los campos petroleros del Oriente Venezolano.



Figura 91 : Mapa de la ubicación geográfica de la planta
Instalaciones principales:



Figura 92: Muelle SIDOR

SIDOR cuenta con una red de comunicaciones conformada por 74 Km. de carreteras pavimentadas, 132 Km. de vías férreas y acceso al mar por un Terminal portuario con capacidad para atracar hasta 6 barcos de 20.000 t de peso muerto cada uno. En el muelle se cargan y descargan los productos siderúrgicos así como también los materiales e insumos con o sin almacenamiento. El muelle dispone de un área de almacenes de 3000 m², donde se ubican diez naves techadas provistas de grúas puente de 25 t, así como también de un área de patios de 60.000 m² para almacenamiento de carga general y un depósito de aduana de 2.500 m² dividido en cuatro naves techadas.

Por otra parte, SIDOR cuenta con edificaciones en las cuales se desarrollan las áreas administrativas y de soporte al personal, tales como los edificios administrativos I y II, edificio de recursos humanos, comedores, servicio médico y talleres. En el siguiente cuadro se muestran las principales instalaciones productivas.

La ubicación de las principales instalaciones de SIDOR puede observarse en el layout siguiente, tomado del folleto SIDOR publicado por la Dirección de Relaciones Institucionales:

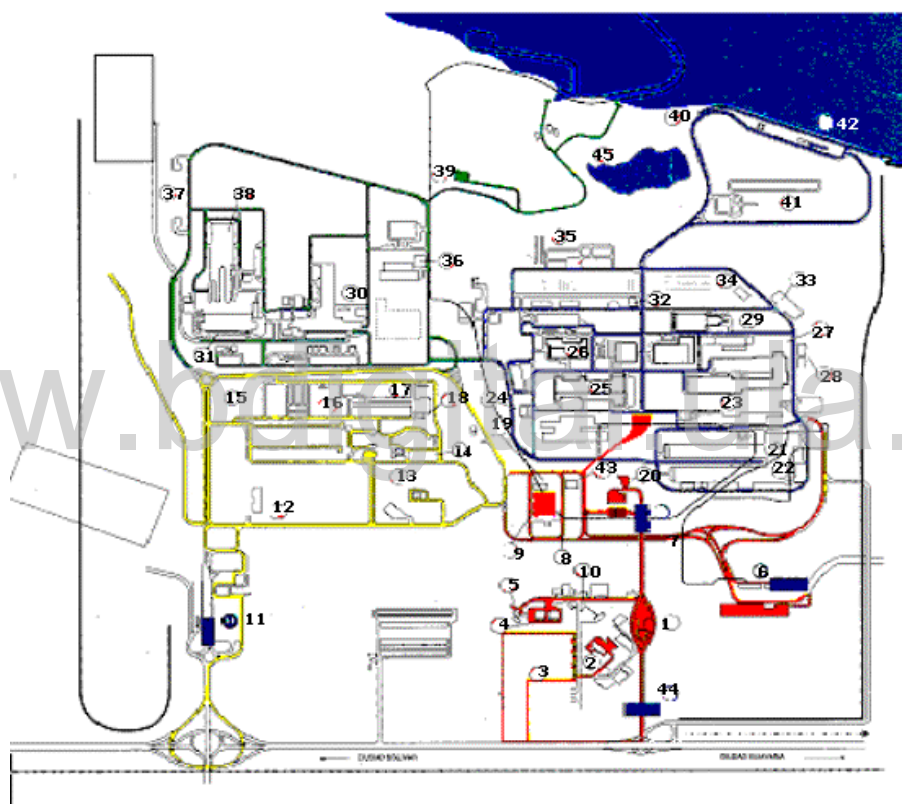


Figura 93: Plano general o layout de SIDOR

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 |

Organización de la Empresa

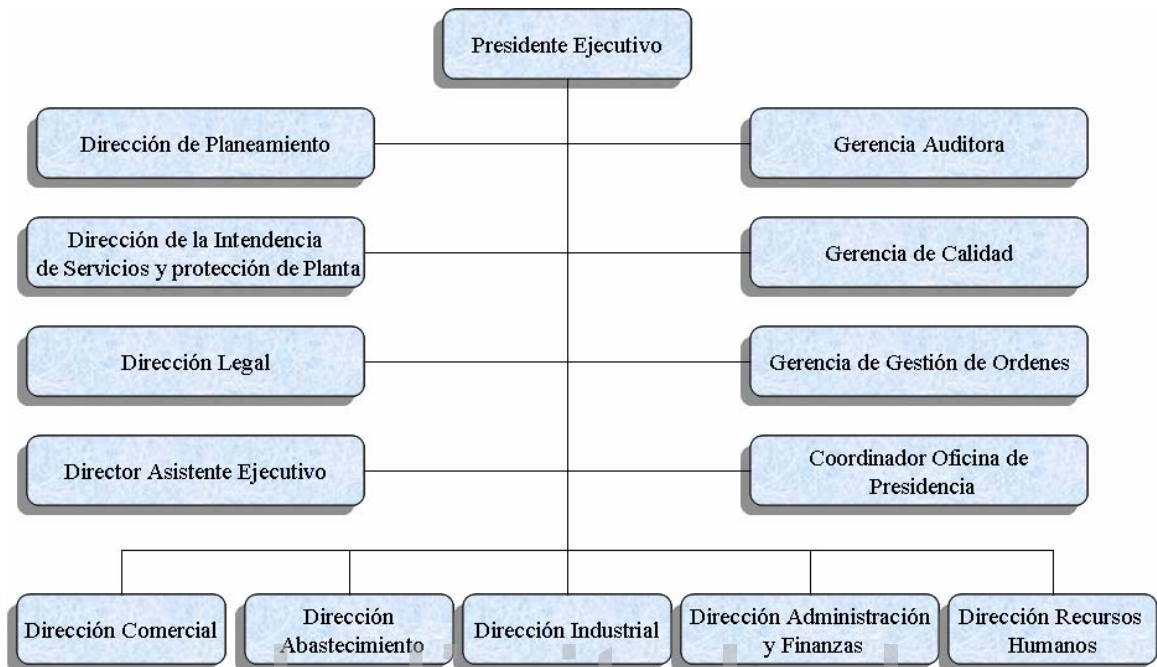


Figura 94: Organización SIDOR

La organización Siderúrgica del Orinoco cuenta actualmente con una Presidencia Ejecutiva, ocho (8) Direcciones Ejecutivas encargadas de lo siguiente:

- Dirección de Administración y Finanzas:** Administrar y asegurar el adecuado rendimiento de los recursos financieros de la empresa y prestar los servicios de contabilidad, auditoría y sistema de información
- Dirección de Recursos Humanos:** Formular y aplicar las políticas y estrategias corporativas en el ámbito socio – laboral, comunicacional y de servicios al personal
- Dirección de Planeamiento:** Formular e impulsar las políticas y estrategias corporativas, en materia comercial, operativa, financiera y de control de gestión.

- **Dirección de Asuntos Legales:** Garantizar la actuación de la compañía dentro del marco legal vigente y representarla ante terceros. En todos los aspectos jurídicos en los que estén involucrados sus derechos e interés.
- **Dirección de la Intendencia de Servicios y Protección de Planta:** Promover la imagen institucional de la empresa ante su público y entorno relevantes.
- **Dirección Comercial:** Comercializar y despachar los productos siderúrgicos en condiciones de calidad y oportunidad competitivas.
- **Dirección de Abastecimiento:** Obtener y suministrar materiales, insumos y servicios, requeridos por la compañía para sus operaciones.
- **Dirección Industrial:** Fabricar productos siderúrgicos y prestar los servicios industriales requeridos de manera competitiva y rentable.

La Dirección Industrial esta conformada por un conjunto de ocho (8) gerencias generales, las cuales ejecutan sus funciones de acuerdo al área específica de cada una de ellas. Cada Gerencia General ejecuta su respectiva labor dentro del marco de su área laboral, dicho de otro modo, solo para el entorno de su proceso productivo.

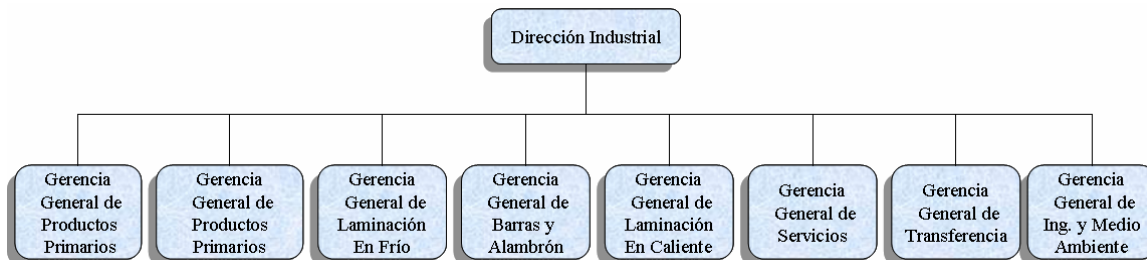


Figura 95: Dirección Industrial

Proceso productivo:

Fabricación de acero

La fabricación de acero en SIDOR se cumple mediante procesos de Reducción Directa y Hornos Eléctricos de Arco, complementados con Metalurgia Secundaria en los hornos de cuchara que garantizan la calidad interna del producto.

Finos de mineral, con alto contenido de hierro, se aglomeran en la Planta de Peletización. El producto resultante —las pellas— es procesado en dos plantas de Reducción Directa, una HyL II (dos módulos de lecho fijo) y otra Midrex (cuatro módulos de lecho móvil), que garantizan la obtención de Hierro de Reducción Directa (HRD). El HRD se carga a los Hornos Eléctricos de Arco para obtener acero líquido.

El acero líquido resultante, con alta calidad y bajos contenidos de impurezas y residuales, tiene una mayor participación de HRD y una menor proporción de chatarra (20% máximo). Su refinación se realiza en las Estaciones de Metalurgia Secundaria, donde se le incorporan las ferroaleaciones. Posteriormente, pasa a las máquinas de Colada Continua para su solidificación, obteniéndose semielaborados —Planchones o Palanquillas— que se destinan a la fabricación de Productos Planos y Productos Largos, respectivamente.

Fabricación de productos planos

Los planchones son cargados en Hornos de Recalentamiento y llevados a temperaturas de laminación. Este tratamiento permite, por medio de la oxidación que se genera, remover pequeños defectos superficiales y ablandar el acero para ser transformado mecánicamente en el Tren de Laminación en Caliente, en Bandas, con ancho y espesor definidos. Las Bandas pueden ser suministradas como tales o como Bobinas o Láminas, sin decapar o decapadas, en función de los requerimientos del cliente en el uso y forma.

Las bandas también pueden ser sometidas a deformación a temperatura ambiente (Laminación en Frío) para reducir el espesor y obtener Bobinas Laminadas en Frío (LAF). Estas últimas pueden ser entregadas al mercado como crudas (Full Hard), o continuar su procesamiento en los Hornos de Recocido y en los Trenes de Laminación de Temple, con el objetivo de modificar sus características metalúrgicas, mecánicas y, muy ligeramente, las geométricas. De esta manera, se obtienen Bobinas recocidas y/o procesadas en el Laminador de Temple, que podrán ser

proporcionadas en Bobinas, cortadas a longitudes específicas (Láminas), o continuar procesos posteriores con recubrimiento electroquímico de cromo o estaño.

Fabricación de productos largos

Las palanquillas son cargadas en Hornos de Recalentamiento y llevadas a temperatura de laminación. Este tratamiento permite, por medio de la oxidación generada, remover pequeños defectos superficiales y ablandar el acero para ser transformado mecánicamente en los Laminadores de Alambroón y de Barras, para obtener el Alambroón y las Barras con Resaltes (Cabillas), respectivamente.

Descripción detallada:

En este campo profundizaremos más sobre el proceso productivo al cual compete el desarrollo de este proyecto, lo cual incluye el área de laminación.

Laminación en Caliente:



Figura 96: Laminador en Caliente



Figura 97: Bobina Terminada

Función:

Obtener a partir de la deformación de planchones previamente calentados a una temperatura de 1200-1300 °C, planchas gruesas destinadas para la venta y bobinas que pueden ser vendidas o enviadas hacia Laminación en Frío.

Materiales Entrantes:

Para la producción de planchas gruesas, planchones con las siguientes dimensiones:



Longitud:	1.640 - 2.800 mm.
Ancho:	1.240 mm.
Espesor:	175 y 200 mm.

Para la producción de bobinas, planchones con las siguientes dimensiones:

Longitud:	6.100 - 12.000 mm.
Ancho:	450 -1.260 mm.
Espesor:	175 - 200 mm.

Materiales Salientes:

Planchas gruesas con las siguientes dimensiones:

Longitud:	6.100 - 2.400 mm.
Ancho:	1.500 -3.000 mm.
Espesor:	6 - 75 mm.

Bobinas con las siguientes dimensiones:

Ancho:	600-1.250 mm.
Espesor:	2 mm a 12 mm.
Peso:	9.6 a 20.00 Ton.

Descripción:

Los planchones 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 trasladado 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.

Enfriamiento

Al salir del último bastidor, la banda pasa por un sistema de enfriamiento laminar con duchas de agua. La forma y propiedades mecánicas de la banda son controladas, ajustando el flujo del agua.

Enrolladores



Figura 98: Enrolladores

Luego del enfriamiento, la banda se enrolla alrededor de un mandril, colocándose posteriormente un volcador para su flejado e identificación, y es transportada al área de almacenamiento.

Corte y tajado

Las bandas ya producidas pueden ser cortadas transversalmente a longitud específica y/o desbordeadas, transformándose en Bobinas. Ambos procesos pueden ser aplicados al mismo tiempo o separadamente en líneas posteriores de Inspección y Corte. Cuando se cortan de manera transversal, se ofrecen como Láminas y el material desbordeado se ofrece como tal o continúa al decapado como paso previo a la Laminación en Frío.

Línea de Decapado I y II



Figura 99: Material Decapado

Para eliminar las bandas el oxido de laminación, las bandas se hacen pasar de manera continua por tanques con ácido clorhídrico, bajo condiciones controladas de concentración y temperatura. Posteriormente, son lavadas con agua para eliminar el ácido remanente en la superficie. Luego se protegen con aceite mineral de fácil remoción.

Función:

La línea de decapado o desoxidación, se utiliza para eliminar el óxido o escamas que se hayan formado en la banda metálica, durante su paso por el laminador en caliente o cuando son alimentadas en los patios de almacenamiento.

Materiales Entrantes:

Banda de acero en forma de bobinas provenientes del patio de bobinas, con las siguientes características:

$6.35 \text{ mm} \geq \text{espesor} \geq 1.53 \text{ mm}.$

$1.270 \text{ mm} \geq \text{ancho} \geq 457 \text{ mm}.$

$\text{Diámetro} \leq 1.850 \text{ mm}.$

Materiales Salientes:

Bandas de acero decapadas en forma de bobinas.

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 si (bobina entrante y saliente). De allí pasa a los tanques de desoxidación y enjuague donde la someten al corte de bordes y engrase. Por último, después de enrollado pasan a las áreas de almacenamiento.

Laminación en frío:

Sidor cuenta con dos laminadores en frío (Tándems) para la fabricación de productos de alta calidad. En su producción utiliza bobinas en caliente – de propia fabricación – previamente decapadas, con un espesor máximo de 4,30 mm. Las cuales son laminadas en frío a un espesor máximo de 2,00 mm. Siguiendo practicas metalúrgicas y operativas que le garantizan productos con calidad de exportación.

Secuencia de producción:



La Laminación en Frío se realiza mediante el procesamiento de bobinas decapadas, a través de cinco (5) bastidores colocados en serie. Cada bastidor, de alta potencia, es capaz de impartir esfuerzos de compresión suficiente para reducir el espesor de la bobina hasta 0,17 mm. Durante la laminación se utiliza una emulsión para lubricar y disminuir la fricción generada por el roce entre los cilindros y la lámina. La bobina se alimenta en un extremo en el mandril desenrollador y, progresivamente, se le va reduciendo el espesor en cada bastidor hasta lograr el espesor deseado a la salida del último. El material es enrollado y se presenta en forma de Bobinas. Bajo estas condiciones, el material se encuentra altamente deformado en frío, perdiendo su ductilidad, con usos restringidos, por lo que debe ser tratado térmicamente (Recocido) para ablandarlo y recuperar sus características mecánicas

Laminador Tandem I y II

Función:

Procesar bobinas laminadas en caliente, provenientes de las líneas de Decapado, con la finalidad de reducir el espesor de la misma. Para estas reducciones se utilizan emulsiones de concentraciones de aceite.

Materiales Entrantes:

Bobina laminadas en caliente decapadas, con las siguientes características:

1.5 mm. \leq espesor \leq 4.75 mm.

457 mm. \leq ancho \leq 1050 mm.

Peso máximo= 17.100 Kg.

Velocidad de entrada de la banda al laminador: 60 Mtrs/min.

Materiales Salientes:

Bobinas laminadas en frío crudas con las siguientes características:

0.18 mm. \leq espesor \leq 2.0 mm.

457 mm. \leq espesor \leq 1050 mm.

Peso máximo= 17.100 Kg.

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 (5°) 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.

Línea de Estañado y Cromado Electrolítico I

Función:

Estañar o cromar bandas de acero mediante electrólisis.

Materiales Entrantes:

Bandas de acero preparadas en forma de bobinas provenientes de las líneas de preparación de bobinas.

Materiales Salientes:

Bandas de acero estañadas o cromadas en forma de bobinas.

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.

Línea de Estañado Electrolítico II

Función:

Estañar bandas de acero mediante electrólisis.

Materiales Entrantes:

Bandas de acero preparadas en forma de bobinas provenientes de las líneas de preparación de bobinas.

Materiales Salientes:

Bandas de acero estañadas en forma de bobinas.

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.

Limpieza electrolítica



Figura 100: Limpieza electrolítica

Durante la laminación, el material arrastra una pequeña película de emulsión, la cual requiere ser removida y/o se evapora durante el proceso de recocido, sin dejar restos de carbono sobre la superficie de la chapa. Para la remoción de la película de aceite, el material se procesa a través de la línea de Limpieza Electrolítica.

Se procesará posteriormente en la línea de Recocido Estático o en la línea de Recocido Continuo. Para la limpieza se utilizan soluciones calientes de detergentes alcalinos mediante reacciones de electrólisis.

Recocido:



Figura 101: Recocido

El material laminado en frío es duro y poco flexible, tiende a regresar a su forma original, lo cual limita su uso. Para ampliar su campo de utilización, éste debe ser sometido al Recocido, es decir, a un tratamiento térmico con temperaturas inferiores a la de transformación.

Este tratamiento se cumple bajo dos modalidades principales:

1. Recocido Estático, método tipo “batch” (Recocido en caja o en campana).
2. Recocido Continuo.

El Recocido Estático se ejecuta colocando varias bobinas laminadas en frío sobre una base que se cubre posteriormente con una campana protectora que protege el material con una atmósfera no oxidante. Sobre esta campana se coloca un horno que permite someter las piezas a un ciclo de calentamiento prescrito en las prácticas.

En el Recocido Continuo, las bobinas se desenrollan y se hacen pasar de manera continua por un horno a temperatura apropiada. Luego pasa por una fase de enfriamiento para volver a ser enrollados en bobinas nuevamente. Para mantener el flujo de material, a la salida del desenrollador se sueldan el final de una bobina con el inicio de la siguiente. Ello se facilita con sistemas de compensación instalados a la entrada y salida de la línea.

Durante el Recocido, se utiliza una atmósfera protectora no oxidante, compuesta por hidrógeno y nitrógeno que, además de proteger el acero de una reoxidación, limpia el acero de los remanentes de la emulsión utilizada durante la Laminación en Frío.

Indistintamente del método utilizado, las características del acero, luego del Recocido, dependerán de la composición química, la temperatura de laminación en caliente, la reducción en frío, el tiempo y temperatura de recocido.

Soporte de plataforma de .NET

Framework:

Cada cierto tiempo se produce una revolución en la tecnología empleada en el desarrollo de aplicaciones. Ahora estamos viviendo la revolución, lo que Microsoft llama .NET Framework o más comúnmente .NET.

Microsoft lanza esta nueva tecnología como respuesta a tecnología Java de Sun. El Framework .NET tiene grandes similitudes con la plataforma Java. El concepto de Framework de .Net es un marco donde las aplicaciones correrán, es decir ya no se hará directamente con el sistema operativo sino bajo esta armazón.

Los frameworks orientados al objeto son la piedra angular de la moderna ingeniería del software.

Los frameworks son los Generadores de Aplicación que se relacionan directamente con un dominio específico.

Los frameworks deben generar las aplicaciones para un dominio entero. Los frameworks no son ejecutables. Algunas de las características del framework no son mutables ni tampoco pueden ser alteradas fácilmente

Arquitectura de Framework:

La nueva tecnología de Microsoft ofrece soluciones a los problemas de programación actuales, como son la administración de código o la programación para Internet. Para aprovechar al máximo las características de .Net es necesario entender la arquitectura básica en la que esta implementada esta tecnología y así beneficiarse de todas las características que ofrece esta nueva plataforma.

El Framework de .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. Los principales componentes de este entorno son:

- Lenguajes de compilación
- Biblioteca de clases de .Net
- CLR (Common Language Runtime)

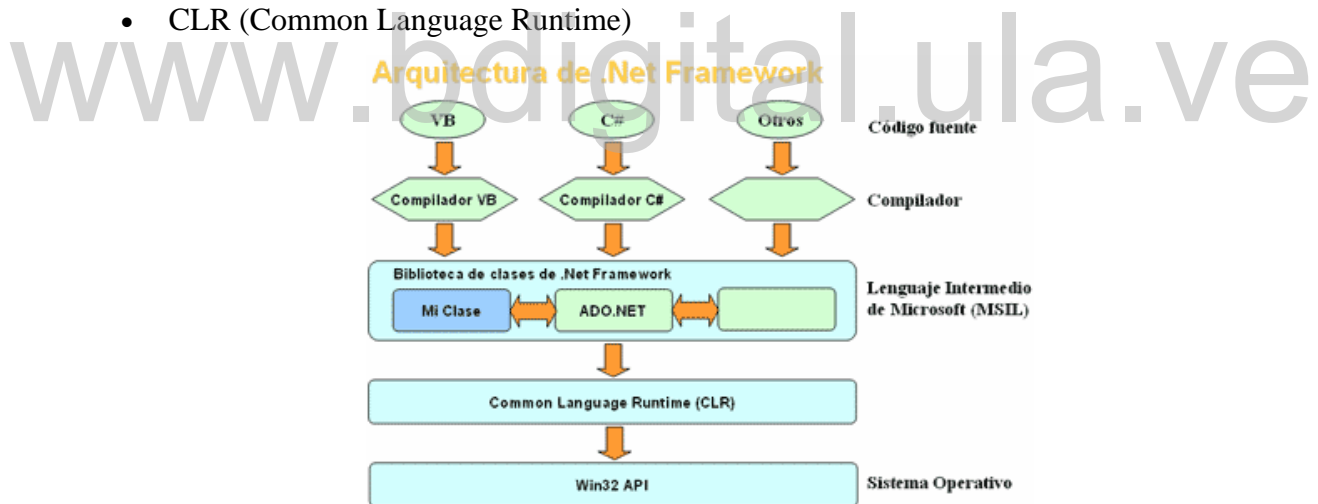


Figura 102: Arquitectura de .Net Framework

.Net Framework soporta múltiples lenguajes de programación y aunque cada lenguaje tiene sus características propias, es posible desarrollar cualquier tipo de aplicación con cualquiera de estos lenguajes. Existen más de 30 lenguajes adaptados a .Net, desde los más conocidos como C# (C Sharp), Visual Basic o C++ hasta otros lenguajes menos conocidos como Perl o Cobol.

Common Language Runtime (CLR):

El CLR es el verdadero núcleo del Framework de .Net, ya que es el entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios que ofrece el sistema operativo estándar Win32.

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .Net en un mismo código, denominado código intermedio (MSIL, Microsoft Intermediate Lenguaje). Para generar dicho código el compilador se basa en el Common Language Specification (CLS) que determina las reglas necesarias para crear código MSIL compatible con el CLR.

De esta forma, indistintamente de la herramienta de desarrollo utilizada y del lenguaje elegido, el código generado es siempre el mismo, ya que el MSIL es el único lenguaje que entiende directamente el CLR. Este código es transparente al desarrollo de la aplicación ya que lo genera automáticamente el compilador.

Sin embargo, el código generado en MSIL no es código máquina y por tanto no puede ejecutarse directamente. Se necesita un segundo paso en el que una herramienta denominada compilador JIT (Just-In-Time) genera el código máquina real que se ejecuta en la plataforma que tenga la computadora. De esta forma se consigue con .Net cierta independencia de la plataforma, ya que cada plataforma puede tener su compilador JIT y crear su propio código máquina a partir del código MSIL.

La compilación JIT la realiza el CLR a medida que se invocan los métodos en el programa y, el código ejecutable obtenido, se almacena en la memoria caché de la computadora, siendo recompilado sólo cuando se produce algún cambio en el código fuente.

Biblioteca de clases de .Net

Cuando se está programando una aplicación muchas veces se necesitan realizar acciones como manipulación de archivos, acceso a datos, conocer el estado del sistema, implementar seguridad, etc. El Framework organiza toda la funcionalidad del sistema operativo en un espacio de nombres

jerárquico de forma que a la hora de programar resulta bastante sencillo encontrar lo que se necesita.

Para ello, el Framework posee un sistema de tipos universal, denominado Common Type System (CTS). Este sistema permite que el programador pueda interactuar los tipos que se incluyen en el propio Framework (biblioteca de clases de .Net) con los creados por él mismo (clases). De esta forma se aprovechan las ventajas propias de la programación orientada a objetos, como la herencia de clases predefinidas para crear nuevas clases, o el polimorfismo de clases para modificar o ampliar funcionalidades de clases ya existentes.

Biblioteca de clases de .NET Framework

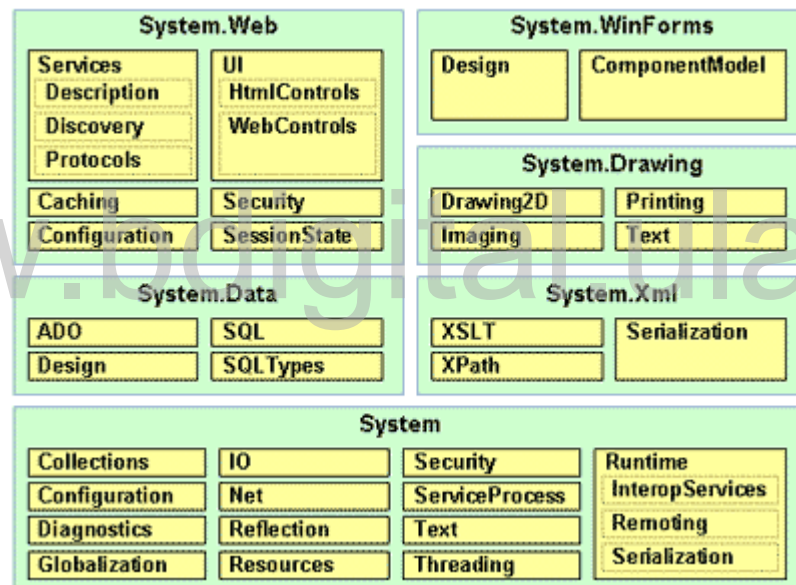


Figura 103: Biblioteca de Clases .Net

La biblioteca de clases de .Net Framework incluye, entre otros, tres componentes clave:

- ASP.NET para construir aplicaciones y servicios Web.
- Windows Forms para desarrollar interfaces de usuario.
- ADO.NET para conectar las aplicaciones a bases de datos.

La forma de organizar la biblioteca de clases de .Net dentro del código es a través de los espacios de nombres (namespaces), donde cada clase está organizada en espacios de nombres según su funcionalidad. Por ejemplo, para manejar ficheros se utiliza el espacio de nombres System.IO y si

lo que se quiere es obtener información de una fuente de datos se utilizará el espacio de nombres System. Data.

La principal ventaja de los espacios de nombres de .Net es que de esta forma se tiene toda la biblioteca de clases de .Net centralizada bajo el mismo espacio de nombres (System). Además, desde cualquier lenguaje se usa la misma sintaxis de invocación, ya que a todos los lenguajes se aplica la misma biblioteca de clases.

Ensamblados:

Uno de los mayores problemas de las aplicaciones actuales es que en muchos casos tienen que tratar con diferentes archivos binarios (DLL's), elementos de registro, conectividad abierta a bases de datos (ODBC), etc. Para solucionarlo el Framework de .Net maneja un nuevo concepto denominado ensamblado. Los ensamblados son ficheros con forma de EXE o DLL que contienen toda la funcionalidad de la aplicación de forma encapsulada. Por tanto la solución al problema puede ser copiar todos los ensamblados en el directorio de la aplicación. Con los ensamblados ya no es necesario registrar los componentes de la aplicación. Esto se debe a que los ensamblados almacenan dentro de sí mismos toda la información necesaria en lo que se denomina el manifiesto del ensamblado. El manifiesto recoge todos los métodos y propiedades en forma de meta-datos junto con otra información descriptiva, como permisos, dependencias, etc. Para gestionar el uso que hacen las aplicaciones de los ensamblados .Net utiliza la llamada caché global de ensamblados (GAC, Global Assembly Cache). Así, .Net Framework puede albergar en el GAC los ensamblados que puedan ser usados por varias aplicaciones e incluso distintas versiones de un mismo ensamblado, algo que no era posible con el anterior modelo COM.

Lenguaje SQL

Las aplicaciones en red son cada día más numerosas y versátiles. En muchos casos, el esquema básico de operación es una serie de scripts que rigen el comportamiento de una base de datos. Debido a la diversidad de lenguajes y de bases de datos existentes, la manera de comunicar entre unos y otras sería realmente complicada a gestionar de no ser por la existencia de estándares que nos permiten realizar las operaciones básicas de una forma universal. Es de eso de lo que trata el Structured Query Language que no es más que un lenguaje estándar de comunicación con bases de datos. Hablamos por tanto de un lenguaje normalizado que nos permite trabajar con

cualquier tipo de lenguaje (ASP o PHP) en combinación con cualquier tipo de base de datos (MS Access, SQL Server, MySQL).

El hecho de que sea estándar no quiere decir que sea idéntico para cada base de datos. En efecto, determinadas bases de datos implementan funciones específicas que no tienen necesariamente que funcionar en otras.

Aparte de esta universalidad, el SQL posee otras dos características muy apreciadas. Por una parte, presenta una potencia y versatilidad notables que contrasta, por otra, con su accesibilidad de aprendizaje.

SQL es una herramienta para organizar, gestionar y recuperar datos almacenados en una base de datos informática. El nombre "SQL" es una abreviatura de *Structured Query Language* (Lenguaje de consultas estructurado). Como su propio nombre indica, SQL es un lenguaje informático que se puede utilizar para interactuar con una base de datos y más concretamente con un tipo específico llamado *base de datos relacional*.

SQL es a la vez un lenguaje fácil de aprender y una herramienta completa para gestionar datos. Las peticiones sobre los datos se expresan mediante sentencias, que deben escribirse de acuerdo con unas reglas sintácticas y semánticas de este lenguaje.

SQL es un lenguaje estándar por haberse visto consolidado por el Instituto Americano de Normas (ANSI) y por la Organización de Estándares Internacional (ISO).

UML (Unified Modeling Language)

Es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa:

Booch, OMT y OOSE. UML ha puesto fin a las llamadas “guerras de métodos” que se han mantenido a lo largo de los 90, en las que los principales métodos sacaban nuevas versiones que incorporaban las técnicas de los demás. Con UML se fusiona la notación de estas técnicas para formar una herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos.

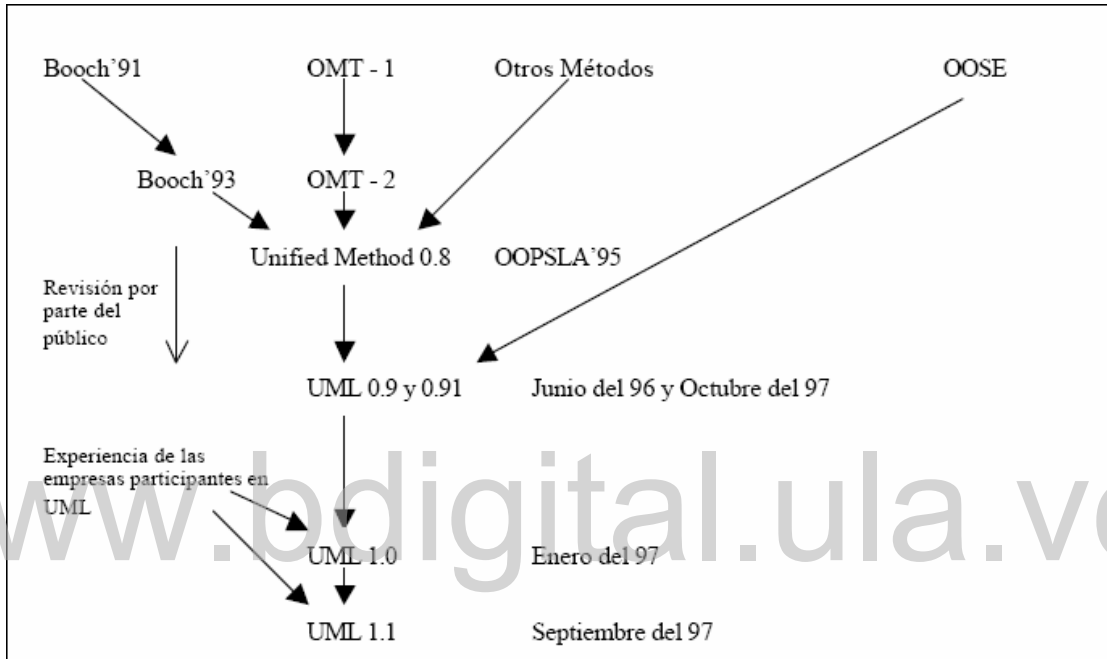


Figura 104: Historia del UML.

El objetivo principal cuando se empezó a gestar UML era posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello era necesario definir una notación y semántica común. En la Figura 17 se puede ver cuál ha sido la evolución de UML hasta la creación de UML 1.1.

Hay que tener en cuenta que el estándar UML no define un proceso de desarrollo específico, tan solo se trata de una notación.

UML contiene conceptos principales de la orientación orientada a objetos a continuación se mostrara una breve definición de estos conceptos básicos:

Modelos: Un modelo representa a un sistema software desde una perspectiva específica. Cada modelo nos permite fijarnos en un aspecto distinto del sistema. Los modelos de UML que se tratan en esta parte son los siguientes:

- Diagrama de Estructura Estática.
- Diagrama de Casos de Uso.
- Diagrama de Secuencia.
- Diagrama de Colaboración.
- Diagrama de Estados.

Diagramas de estructura estática: Con el nombre de diagramas de estructura estática se engloba tanto al Modelo Conceptual de la fase de Análisis como al Diagrama de Clases de la fase de Diseño. Ambos son distintos conceptualmente, mientras el primero modela elementos del dominio el segundo presenta los elementos de la solución software. Sin embargo, ambos comparten la misma notación para los elementos que los forman (clases y objetos) y las relaciones que existen entre los mismos (asociaciones).

Diagramas de caso de uso: Un diagrama de casos de uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Diagramas de iteración:

En los diagramas de interacción se muestra un patrón de interacción entre objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: Diagramas de Secuencia y Diagramas de Colaboración.

Diagrama de estados: Un diagrama de estados muestra la secuencia de estados por los que pasa un caso de uso o un objeto a lo largo de su vida, indicando qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.