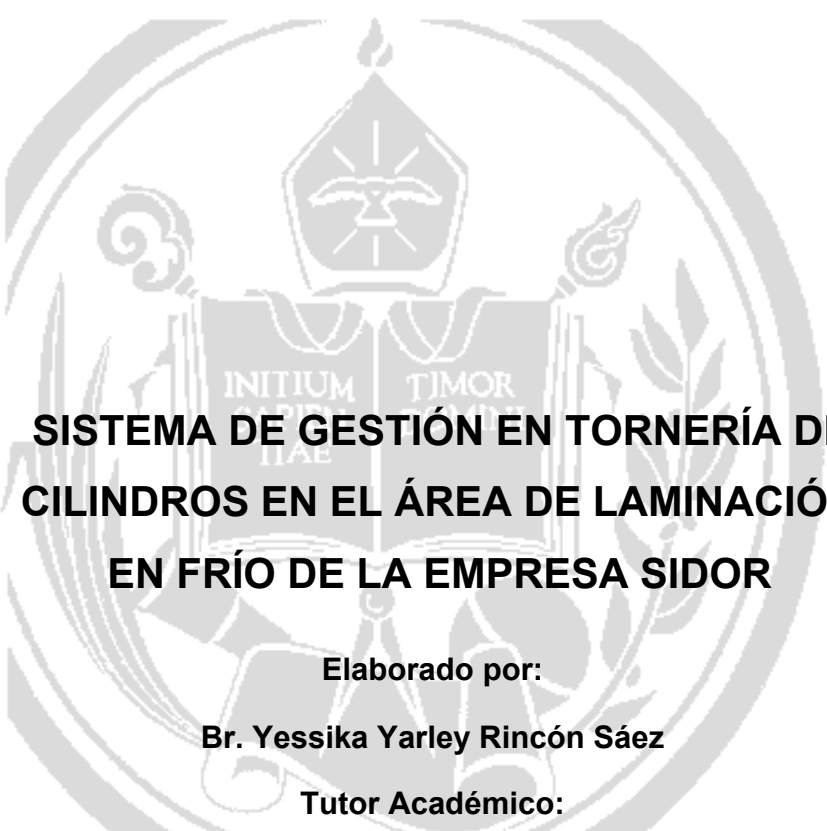




UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA

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



**SISTEMA DE GESTIÓN EN TORNERÍA DE
CILINDROS EN EL ÁREA DE LAMINACIÓN
EN FRÍO DE LA EMPRESA SIDOR**

Elaborado por:

Br. Yessika Yarley Rincón Sáez

Tutor Académico:

Dr. Wladimir Rodríguez

Tutor Industrial:

Ing. Carlos Agostini.

UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA

MÉRIDA, ABRIL DE 2006



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

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

**SISTEMA DE GESTIÓN EN TORNERÍA DE
CILINDROS EN EL ÁREA DE LAMINACIÓN
EN FRÍO DE LA EMPRESA SIDOR**

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

UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

MÉRIDA, ABRIL DE 2006



**PROYECTO DE GRADO
CALIFICACIÓN FINAL**

Titulo del Proyecto de Grado: **“SISTEMA DE GESTIÓN EN TORNERÍA DE CILINDROS EN EL ÁREA DE LAMINACIÓN EN FRIO DE LA EMPRESA SIDOR”.**

Bachiller: **YESSIKA YARLEY RINCÓN SÁEZ** C.I. **14.916.973**

- | | | |
|---|----|-------|
| 1) Calificación del (de la) Profesor(a) Tutor(a): | 20 | (20%) |
| 2) Calificación del manuscrito final: | 20 | (40%) |
| 3) Calificación de la defensa oral: | 20 | (40%) |

Calificación final: 20 **puntos**

Los suscritos miembros del Jurado asignan como calificación final del Proyecto de Grado la nota de:

20 (puntos)
Números

Veinte (puntos)
Letras

Observaciones:

Prof. Wladimir Rodriguez
Tutor

Prof. Isabel Besembel
Jurado

Prof. Dulce Rivero
Jurado

Mérida, 21 de Abril de 2006





Dedicatoria

A Dios, por permitirme vivir cada momento lleno de alegría, por iluminar mi camino y guiarme hacia un futuro lleno de éxitos en compañía de los seres que más amo.

A mi Madre, por estar siempre a mi lado apoyándome incondicionalmente, brindándome todo su amor, confianza y sabiduría. En los momentos donde sentía que ya no podía seguir adelante me diste tu mano y me llenaste de fuerza y valentía, siempre fuiste y serás mi modelo a seguir. Gracias madre, porque todo lo que soy te lo debo a ti.



Agradecimientos

Existen personas que me han brindado su apoyo y me han ayudado a alcanzar esta meta tan anhelada en mi vida. Por esto quiero expresar mis sinceros agradecimientos para:

Mi padre, mis abuelos, mis tías, mis primos y toda mi familia, que siempre han estado presentes y me han ofrecido su apoyo incondicional.

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 actualmente y de la cual me siento muy orgullosa.

A mi tutor Dr. Wladimir Rodríguez quien ha sido un apoyo en el fortalecimiento de los conocimientos necesarios para la realización de este sistema.

A la Sra. Judit por su paciencia y por estar pendiente de cualquier eventualidad y mostrar la mejor disposición en ayudarme.

A todo el personal de Automatización en SIDOR, por haberme brindado la oportunidad de realizar éste proyecto, que ha fortalecido enormemente mi formación profesional, así como también, me ha llenado de experiencias que me hicieron crecer a nivel personal.

A todos aquellos amigos y compañeros que vivieron junto a mí muchos momentos, tanto de alegrías, como de tristezas, brindándome mucha confianza para seguir siempre adelante.

Y a todas aquellas personas que de una u otra manera hicieron posible la realización de este trabajo.



Resumen

El Sistema de gestión en tornería de cilindros en el área de laminación en frío, se enmarca dentro de la intención de la empresa Siderúrgica del Orinoco (SIDOR C.A.) de centralizar la información, unificar la carga de datos, garantizar la exactitud y confiabilidad de la información, eliminar el uso de papel, globalizar el acceso a los datos y disponer de información en tiempo real. En la actualidad el manejo de la información es un factor determinante en el éxito o fracaso de cualquier empresa. En SIDOR, todas las aplicaciones a nivel de planta están siendo migradas a QNX, que es un sistema operativo de tiempo real, flexible, eficiente y modular, utilizando Sybase SQL Anywhere como sistema manejador de base de datos; su característica principal es la optimización de los recursos (uso de memoria RAM, disco rígido y CPU) para su buen desempeño.

Durante el análisis, diseño e implementación de este sistema se sigue como guía el Proceso Unificado de Desarrollo de Software, proporcionando un soporte teórico al proyecto. Debido a la naturaleza de los proyectos de desarrollo en el área de Automatización, se han de generar productos intermedios que satisfagan las necesidades del cliente, es decir, un desarrollo iterativo, razón por la cual se afianza aún más la utilización del proceso unificado como metodología de desarrollo. Se llevaron a cabo en cuatro fases: inicio, elaboración, construcción y transición.

Durante la fase de inicio se procedió a analizar todos los requerimientos necesarios para lograr la funcionalidad del sistema. De esta manera, los requisitos propios del sistema fueron reflejados en un modelo de casos de uso, en donde se representaron aquellos que permitieron obtener un esbozo inicial de la arquitectura.

La fase de elaboración se centra 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.

La fase de construcción tuvo como propósito primordial obtener el producto software en su versión operativa. Dicho producto cuenta con la calidad adecuada para su aplicación y cumple con todos los requisitos establecidos previamente.

En la fase de transición se realizaron 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 elaboraron 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.



Tabla de Contenido

Dedicatoria.....	i
Agradecimientos	ii
Resumen.....	iii
Tabla de Contenido	v
Índice de Figuras.....	viii
Índice de Tablas	xii
Glosario.....	1
I INTRODUCCION	4
I.1 Descripción de la Empresa	4
I.2 Ubicación Geográfica de SIDOR:	5
I.3 Organización.....	5
I.4 Antecedentes	8
I.5 Definición del Problema	10
I.6 Plataforma Estándar De Automatización - SIDOR	15
I.7 Objetivos Generales.....	16
I.8 Objetivos Específicos	17
I.9 Metodología a utilizar.....	18
I.10 Alcance	19
I.11 Estructura del Documento.	20
II MARCO TEÓRICO	22
II.1 Proceso Unificado.....	22
II.1.1 Características del Proceso Unificado	23
II.1.2 Etapas del Proceso Unificado:	25
II.1.3 Fase de Inicio.	25
II.1.4 Fase de Elaboración	26
II.1.5 Fase de Construcción	26
II.1.6 Fase de Transición.	26



II.2	Lenguaje de Modelado Unificado.....	27
II.3	Sistema Operativo QNX.....	31
II.3.1	Características:.....	31
II.4	Sybase SQL Anywhere para QNX.....	32
II.5	XML (Lenguaje de Marcas Extensible).....	34
II.6	¿Qué es un Servicio Web?.....	35
II.7	SOAP.....	36
III	INICIO.....	39
III.1	Visión General:.....	39
III.2	Modelo de Casos de Uso:.....	40
III.2.1	Límites del Sistema.....	40
III.2.2	Actores Principales.....	41
III.2.3	Actores – Actividades.....	42
III.2.4	Requisitos Funcionales.....	43
III.2.5	Casos de Uso.....	44
III.2.6	Modelos de Casos de Uso:.....	46
III.2.7	Definición de Casos de Uso.....	47
III.3	Especificación Complementaria.....	58
IV	ELABORACION.....	60
IV.1	Iteración I:.....	60
IV.1.1	Diagramas de Secuencia para el Sistema De Gestión En Tornería De Cilindro.....	61
IV.1.2	Modelo de Dominio.....	80
IV.1.3	Modelo de Diseño.....	84
IV.1.4	Implementación.....	93
IV.1.5	Plan de Iteración.....	96
IV.2	Iteración 2:.....	97
IV.2.1	Casos de Uso.....	97
IV.2.2	Modelo de Casos de Uso.....	100
IV.2.3	Diagramas de Secuencia.....	101
IV.2.4	Modelo de Dominio - Iteración 2.....	106



IV.2.5	Modelo de Diseño	109
IV.2.6	Diagramas de Clases	113
IV.2.7	Implementación	116
IV.2.8	Plan de Iteración	117
IV.3	Iteración 3:	118
IV.3.1	Casos de Usos	118
IV.3.2	Modelo de Casos de Uso	122
IV.3.3	Diagramas de Secuencia	123
IV.3.4	Modelo de Dominio - Iteración 3	127
IV.3.5	Modelo de Diseño	131
IV.3.6	Diagramas de Clases	134
IV.3.7	Implementación	136
V	CONSTRUCCION Y TRANSICIÓN	138
V.1	Interfaz de Usuario	138
V.1.1	Iv-Desktop del STC:	138
V.1.2	Catálogos del STC:	140
V.1.3	Pantallas del STC:	141
V.2	Diagrama de Implementación	143
V.3	Pruebas Finales en la Fase de Transición	144
	Conclusiones	146
	Recomendaciones	148
	Bibliografía	149
	APÉNDICE A: Caso de Uso Completo (Procesar Rectificado)	150
	APÉNDICE B: Pantallas del Sistema	154
	APÉNDICE C: Código de Conectores	163



Índice de Figuras

Figura 1: Mapa de la Ubicación Geográfica de la Planta	5
Figura 2: Organización de la Presidencia Ejecutiva	6
Figura 3: Organización de la Dirección Industrial.....	6
Figura 4: Estructura de la Gerencia General de Ingeniería y Medio Ambiente	7
Figura 5: Niveles de Automatización	8
Figura 6: Sistema de Gestión Actual en el Taller de Cilindros	9
Figura 7: Ciclo Productivo en el Taller de Cilindros.....	14
Figura 8: Plataforma Estándar Automatización - SIDOR	15
Figura 9: Un Proceso de Desarrollo de Software.	22
Figura 10: Estructura Completa del Proceso Unificado	23
Figura 11: Estructura Básica del Proceso Unificado	24
Figura 12: Historia del UML	28
Figura 13: Conexión de Aplicaciones a Servidores de Bases de Datos Sybase bajo QNX..	33
Figura 14: Servicios Web en Acción.....	36
Figura 15: Estructura del SOAP	36
Figura 16: Diagrama de Casos de Uso del STC	46
Figura 17: DSS Procesar y Reportar Ultrasonido.....	61
Figura 18: DSS Visualizar Lista de Cilindros a Inspección de Ultrasonido.....	62
Figura 19: DSS Visualizar Información del Cilindro Seleccionado en Ultrasonido	62
Figura 20: DSS Montar Cilindro en Máquina de Ultrasonido.....	63
Figura 21: DSS Inserción y Modificación de Datos de Ultrasonido	64
Figura 22: DSS Visualización de Datos de Ultrasonido.....	65
Figura 23: DSS Avanzar Secuencia de Ultrasonido	65
Figura 24: DSS Obtener Ultrasonido Procesado	66
Figura 25: DSS Reportar Ultrasonido.....	67
Figura 26: DSS Sacar Secuencia de Ultrasonidos	67
Figura 27: DSS Eliminar Ultrasonido.....	68
Figura 28: DSS Procesar y Reportar Rectificado	69



Figura 29: DSS Actualizar Lista de Cilindros a Rectificar.....	70
Figura 30: DSS Obtener Información de Cilindro a Rectificar	71
Figura 31: DSS Montar Cilindro en Máquina de Rectificado	71
Figura 32: DSS Inserción y Modificación de Datos de Rectificado.....	72
Figura 33: DSS Visualización de Datos de Rectificado	73
Figura 34: DSS Avanzar Secuencia de Rectificado.....	73
Figura 35: DSS Reportar Rectificado	74
Figura 36: DSS Obtener Rectificado Procesado.....	75
Figura 37: DSS Justificación de Datos Fuera de Patrón en Rectificado.....	76
Figura 38: DSS Eliminar Rectificado	77
Figura 39: DSS Procesar Granallado	78
Figura 40: DSS Reportar y Procesar Pareja.....	79
Figura 41: Modelo de Dominio – Asociaciones – Iteración 1	82
Figura 42: Modelo de Dominio – Atributos – Iteración 1	83
Figura 43: DI – Procesar Ultrasonido	85
Figura 44: DI – Obtener Cilindros a Rectificar	86
Figura 45: DI – Procesar Rectificado	87
Figura 46: DI – Procesar Granallado	88
Figura 47: DI – Procesar Pareja.....	89
Figura 48: Diagrama de Clases Parcial del STC – Iteración 1	91
Figura 49: Diagrama de Clases del STC – Iteración 1	92
Figura 50: Esquema de Pantallas.....	94
Figura 51: Modelo de Casos de Uso del STC – Iteración 2	100
Figura 52: Modelo de Casos de Uso del STC – Iteración 2	101
Figura 53: DSS Insertar Defectos de Ultrasonido	102
Figura 54: DSS Modificar Defectos de Ultrasonido.....	102
Figura 55: DSS Eliminar Defectos de Ultrasonido.....	103
Figura 56: DSS Procesar y Reportar Montaje	104
Figura 57: DSS Reportar y Procesar Desmontaje.....	105
Figura 58: Modelo de Dominio – Atributos – Iteración 2	107
Figura 59: Modelo de Dominio – Asociaciones – Iteración 2.....	108



Figura 60: DI – Obtener Cilindros a Montar	109
Figura 61: DI – Procesar Montaje.....	110
Figura 62: DI – Cambio Ciclo	111
Figura 63: DI – Procesar Desmontaje.....	112
Figura 64: Clases del STC – Iteración 2	113
Figura 65: Diagrama de Clases Parcial del STC – Iteración 2	114
Figura 66: Diagrama de Clases del STC – Iteración 2	115
Figura 67: Pantalla de Desmontaje – Iteración 2.....	116
Figura 68: Modelo de Casos de Uso del STC – Iteración 3	122
Figura 69: Modelo de Casos de Uso del STC – Iteración 3	123
Figura 70: DSS Procesar y Reportar Montaje Mueas.....	124
Figura 71: DSS Procesar y Reportar Desmontaje Mueas	124
Figura 72: DSS Transporte de Cilindros.....	126
Figura 73: DSS Procesar Agregado de Granalla	126
Figura 74: DSS Modificar Agregado de Granalla	127
Figura 75: DSS Eliminar Agregado de Granalla	127
Figura 76: Modelo de Dominio – Atributos -Iteración 3.....	129
Figura 77: Modelo de Dominio – Asociaciones -Iteración 3.....	130
Figura 78: DI – Montaje de Muela	131
Figura 79: DI – Desmontaje de Muela.....	132
Figura 80: DI – Transporte de Cilindros.....	133
Figura 81: DI – Control de Granalla Agregada	134
Figura 82: Diagrama de Clases del STC – Iteración 3	135
Figura 83: Pantalla de Control de Mueas	136
Figura 84: Iv-Desktop del STC.....	139
Figura 85: Pantallas de Catálogos del STC	140
Figura 86: Pantalla de Ultrasonido del STC	141
Figura 87: Pantalla de Ingreso de Datos del Módulo de Rectificado	142
Figura 88: Pantalla de Ingreso de Pareja	142
Figura 89: Diagrama de Implementación del STC en el Taller 2.....	143
Figura 90: Pantalla Ingreso de Datos de Ultrasonido	154



Figura 91: Pantalla Principal de Rectificado	155
Figura 92: Pantalla del Patrón para el Rectificado	155
Figura 93: Pantalla Para el Ingreso de Defectos de Cilindros	156
Figura 94: Pantalla Para el Ingreso de Datos en Granallado	156
Figura 95: Pantalla Para el Ingreso de Pases de Granalla.....	157
Figura 96: Pantalla Para Ingreso de Granalla Agregada a la Máquina Granalladora	157
Figura 97: Pantalla Para el Control de Muelas	158
Figura 98: Pantalla Para el Emparejamiento De Cilindros de Trabajo.....	158
Figura 99: Pantalla Para el Ingreso de Datos del Montaje.....	159
Figura 100: Pantalla Para el Ingreso de Datos del Desmontaje.....	159
Figura 101: Pantalla de Transporte de Cilindros	160
Figura 102: Pantalla Para Actualizar Información en el STC-QNX	160
Figura 103: Pantalla de Control de Acceso	161
Figura 104: Pantalla de Cambio de Cilindros en el Laminador (Línea Temple III).....	161
Figura 105: Pantalla Web del STC	162
Figura 106: Pantalla Web de Planificación del STC	162



Índice de Tablas

Tabla 1: Diccionario de datos	1
Tabla 2: Lista de Actores – Actividades del Sistema	42
Tabla 3: Lista de Casos de Uso Clasificados por Prioridad.....	45
Tabla 4: Lista de Categorías de Clases Conceptuales – Iteración 1	80



Glosario

Tabla 1: Diccionario de datos

Ítem	Descripción	Tipo
Id_Cilindro	Identificación única de cada registro de la tabla Ficha_cilindro	Entero
Numero_Cilindro	Identificación única cada cilindro.	Entero
Id_Linea_Actual	Identificador de la línea donde actualmente se encuentra el cilindro.	Cadena (4)
Id_Tipo_Cilindro	Identificador para el tipo de cilindro.	Entero
Nro_Bastidor_Actual	Número del bastidor en el cual el cilindro va o está laminando.	Entero
Nro_ciclo	Número de veces que el cilindro ha pasado por el laminador.	Entero
Indicativo_posición	Posición donde será colocado el cilindro en el laminador (Superior ó Inferior)	Carácter
Kilometraje_Total	Cantidad de kilómetros que ha laminado el cilindro en todo su tiempo operativo.	Real
Kilometraje_Ciclo	Cantidad de Kilómetros que laminó el cilindro durante un ciclo específico.	Real
Tonelaje_Total	Cantidad de toneladas laminadas por el cilindro en todo su tiempo operativo.	Real
Tonelaje_Ciclo	Cantidad de toneladas laminadas por el cilindro durante un ciclo específico.	Real
Rugosidad	Conjunto de irregularidades de la superficie del cilindro.	Real
Corona	Forma geométrica convexa, cóncava o cilíndrica dada a la tabla del cilindro a través del rectificado.	Real



Estante	Lugar utilizado para colocar los cilindros durante el mantenimiento en el taller.	Entero
Id_Area	Identificación única del taller donde se realizo cualquier proceso al cilindro (Taller 1, Taller 2)	Entero
Dureza	Resistencia ofrecida por un cuerpo a ser deformado.	Real
Grado	Es el ángulo donde se encuentra ubicado el defecto, tomando un punto de referencia 0° en la superficie del cilindro.	Real
Distancia	Es la medida en milímetros de la separación entre el lado operador del cilindro (referencia 0) y el defecto.	Real
Profundidad	Es la medida de la hondura del defecto en el cilindro.	Real



CAPÍTULO I: INTRODUCCION



I INTRODUCCION

Los sistemas y tecnologías de información han cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos, suministran una plataforma de información necesaria para la toma de decisiones y, lo más importante, su implantación logra ventajas competitivas o reducir la ventaja de los rivales.

El desarrollo de sistemas de información destinados a satisfacer las necesidades de SIDOR, forma parte de la constante inversión y la búsqueda de una mejora en el nivel competitivo de la misma.

El sistema de gestión en tornería de cilindros es un ejemplo claro de la necesidad de SIDOR de actualizar sus sistemas de información, absorbiendo gran cantidad de sistemas no formales (fuera de estándares) y globalizando el acceso a los datos, permitiendo así contar con información real, para una toma de decisiones eficiente.

1.1 Descripción de la Empresa

La Siderúrgica del Orinoco, Sidor C.A., es un complejo siderúrgico productor de acero integrado de Venezuela, cuyos procesos se inician con la fabricación de pellas y culmina con la entrega de productos terminados largos (barras y alambrón) y planos (laminados en frío, laminados en caliente y recubiertos). Produce planchones, bobinas, laminados, barras, varillas y productos revestidos.



1.2 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 1).



Figura 1: Mapa de la Ubicación Geográfica de la Planta

1.3 Organización

La Siderúrgica del Orinoco cuenta actualmente con una Presidencia Ejecutiva, la cual está conformada por Direcciones Ejecutivas encargadas de realizar diferentes tareas administrativas o de gestión según sea el campo de aplicación o áreas de desempeño. Véase Figura 2.

La Presidencia Ejecutiva delega a las Direcciones Generales, las funciones relacionadas con el ámbito interno de la empresa.

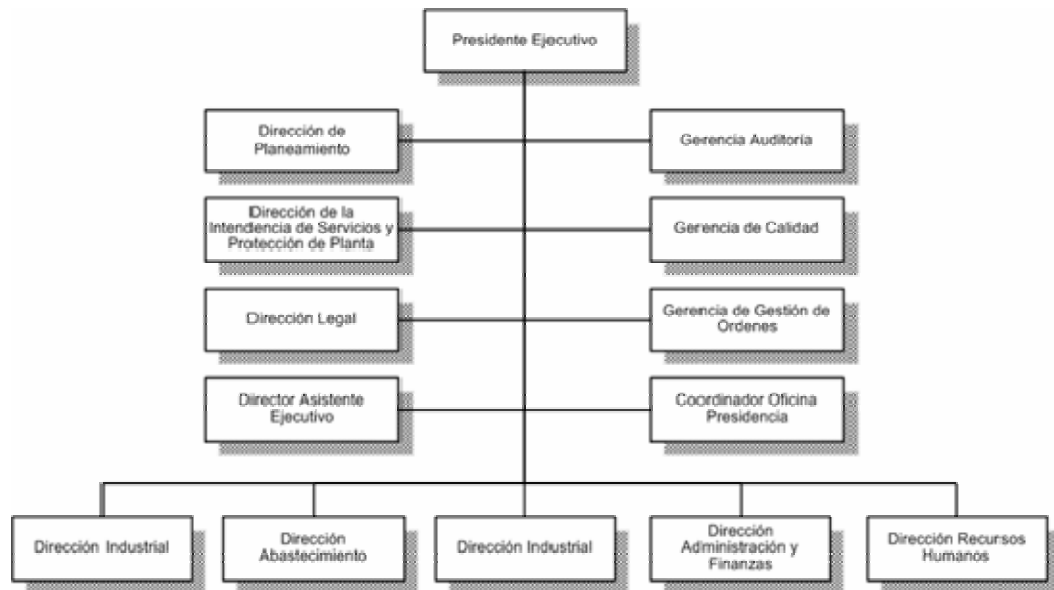


Figura 2: Organización de la Presidencia Ejecutiva

La Dirección Industrial tiene como objetivo primordial el fabricar productos siderúrgicos así como prestar los servicios industriales requeridos de manera competitiva y rentable. Esta dirección está conformada por un conjunto de Gerencias Generales, ejecutando cada una de ellas sus funciones de acuerdo al área específica o campo de aplicación. Véase Figura 3



Figura 3: Organización de la Dirección Industrial

Las Gerencias Generales representan de alguna forma los principales procesos productivos de la empresa, organizados de acuerdo a sus productos o servicios.

Los proyectos de automatización, aunque sean implementados en cualquier planta o gerencia general, son diseñados y elaborados en la Gerencia Automatización y Control, ubicada dentro de la Gerencia General Ingeniería y Medio Ambiente. Véase Figura 4.

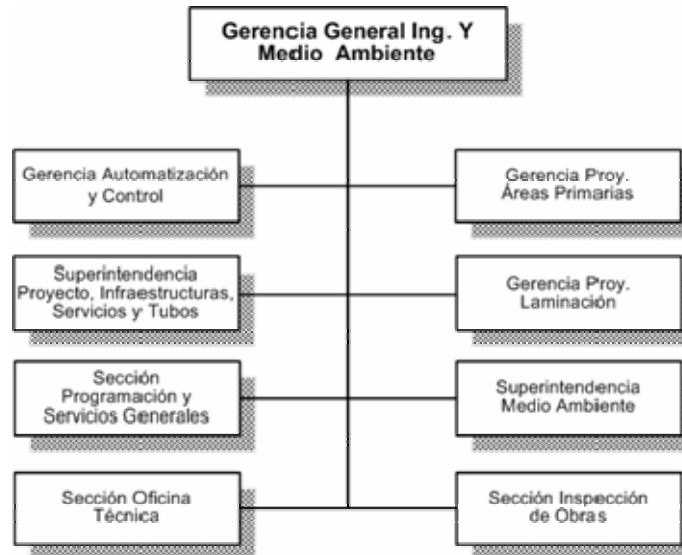


Figura 4: Estructura de la Gerencia General de Ingeniería y Medio Ambiente

Para la automatización y control de procesos en Sidor C.A. se ha desarrollado una plataforma que permite controlar todos los servicios, variables, comandos, eventos (alarmas) y visualizar en su totalidad el proceso productivo, la cual esta conformada por 4 niveles (Figura 5):

- **Nivel 0:** Se encuentran los elementos de campo (sensores, actuadores, etc.).
- **Nivel 1:** Se encuentran los elementos de control (PLC's) y monitoreo de las variables de campo como por ejemplo la temperatura. El PLC recibe toda la información de los elementos de campo.
- **Nivel 2:** Presenta divisiones:
 - **El nivel 2A**, en donde se realiza el monitoreo y control en tiempo real del proceso. Es aquí donde intervienen los operadores de cabina a visualización y seguimiento en tiempo real del proceso.
 - **En el nivel 2B**, basado en tecnología Web se lleva a cabo el monitoreo de datos a través de los sitios de las distintas líneas de producción.



- **Nivel 3:** Nivel Gerencial es donde se desarrolla la explotación de los datos, análisis estadístico del proceso y control de gestión. Intervienen las gerencias de las líneas.

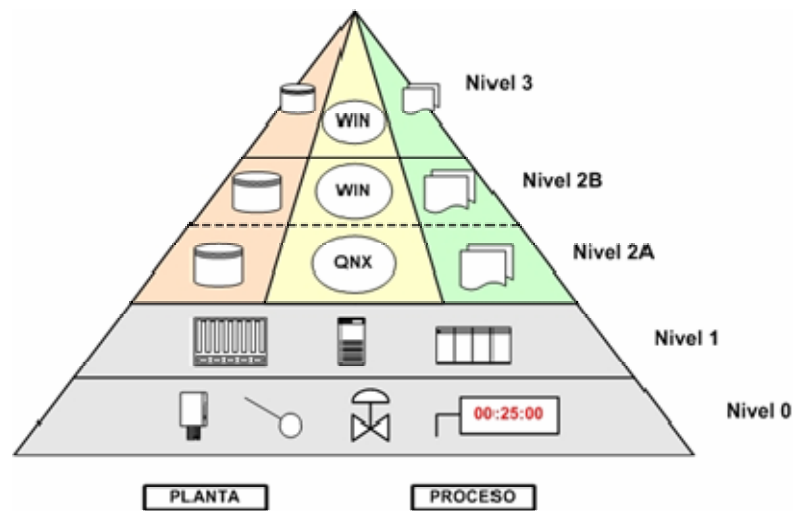


Figura 5: Niveles de Automatización

1.4 Antecedentes

En los talleres de Cilindros, la captura de información para la toma de decisiones proviene de una gran diversidad de Sistemas de Información, tales como el SAP, Planillas, SISCOT (Sistema actual de Gestión), base de datos independientes (ver Figura 6).

Todos estos métodos de almacenamiento de la información resultan ser incongruentes entre sí. Esto obliga a los usuarios (personal del taller de cilindros) a destinar gran parte de su tiempo en recopilar y corregir la información que necesitan, restándole tiempo al análisis, resolución de los problemas y gestión de la información.

La mayor parte de los reportes son construidos realizando cambios directamente en la base de datos y exportando a Excel la información, esto trae como consecuencia una manipulación descontrolada en la misma haciéndola menos confiable.

El sistema de gestión actual (SISCOT), está realizado bajo el lenguaje de programación DELPHI haciéndolo poco portable, y creando la necesidad de su instalación para consultar información.

La base de datos no cuenta con un sistema de replicación adecuado, colocando los datos del sistema en riesgo de pérdida.

En cuanto al hardware, el servidor en la actualidad se encuentra en un área no adecuada para éste, y los equipos utilizados en el área de Rectificación son equipos PC y no máquinas industriales, generando una gran cantidad de tiempo dedicada al mantenimiento de las mismas.

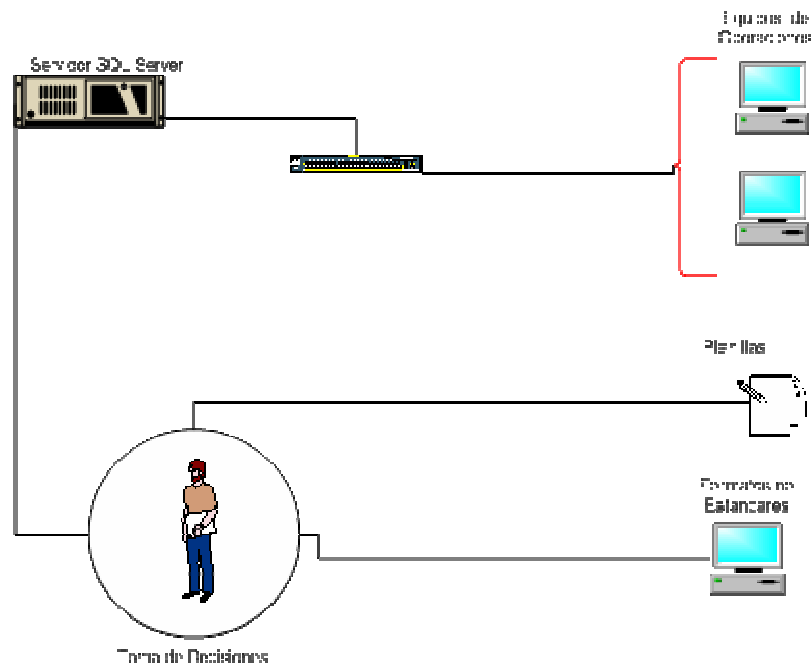


Figura 6: Sistema de Gestión Actual en el Taller de Cilindros



1.5 Definición del Problema

El Sistema de Gestión en Tornería de Cilindros en el área de Laminación en Frío de la empresa SIDOR, es un proyecto de gran envergadura, cuyo objetivo principal es automatizar todos los procesos productivos del taller de cilindros, como son: ultrasonido, rectificado, granallado, montaje, transporte, desmontaje de cilindros y mantenimiento de cajas.

El ciclo productivo del Taller de Cilindros comienza cuando un cilindro nuevo (de apoyo o de trabajo) llega al taller, pasa por una inspección de ultrasonido, donde se verifica su condición y la posible existencia de defectos; si al cilindro se le detecta algún defecto grave, pasa a disposición de control, donde los analistas del taller verifican el defecto y estudian que se debe hacer con el cilindro. El cilindro puede llevarse al taller foráneo, donde se trata de eliminar el defecto, en caso de recuperar el cilindro, éste quedará disponible en la planificación. Cuando la inspección de ultrasonido genera resultados positivos (cilindro sin defectos), el cilindro (en caso de ser de trabajo) pasa a buscar pareja, tomando en consideración ciertos parámetros, como el diámetro, tipo de material a laminar, rugosidad y dureza. Posteriormente el cilindro queda disponible en la planificación de rectificado.

En la planificación de rectificado, los cilindros también pueden ser planificados a granallar, esto depende de los requerimientos del laminador, el proceso de granallado sólo se le aplica a los cilindros de trabajo. Cuando el cilindro sale del proceso de rectificado, sin defectos, pasa al proceso de granallado (opcional), y luego queda disponible para el proceso de montaje.

Todos los cilindros disponibles a montaje deben tener una planificación previa, realizada por el supervisor. Una vez que los cilindros son planificados al montaje, se seleccionan las cajas que serán utilizadas por los cilindros, posteriormente dicho conjunto es enviado al



laminador; en el laminador se certifica la recepción del mismo. Durante el proceso de laminación, el cilindro cumple la función de darle el acabado y el espesor al material que debe laminar, durante este proceso el cilindro sufre ciertos daños que serán rectificadas en el taller, al concluir la laminación los cilindros son desmontados y separados de sus cajas, estos pasan a ser rectificadas y las cajas pasan a mantenimiento, si así es requerido según el kilometraje y tonelaje que tengan hasta ese momento.

A continuación se describe cada proceso por separado, para obtener una visión más amplia y detallada del proceso de los cilindros en el Taller.

- **Ultrasonido:** Mediante este proceso son detectados los defectos que posee el cilindro, utilizando para ello un palpador, que es pasado sobre la superficie del mismo, especificando el grado, distancia, profundidad e intensidad del daño. El ultrasonido se le puede realizar a cualquier cilindro excepto si este ya paso por el proceso de granallado. Posterior al proceso de ultrasonido un cilindro puede ser dirigido a rectificado, emparejamiento (en caso de los cilindros de trabajo sin pareja), o inclusive volver al proceso de ultrasonido.
- **Rectificado:** Cuando el cilindro se encuentra laminando sufre daños sobre la superficie del mismo, por ejemplo, grietas, latigazos, material pegado, entre otros, para eliminar todos estos defectos se le realiza un rectificado al cilindro, donde una rueda abrasiva (Muela) desgasta la superficie del cilindro, con el fin de consumir milímetros de su superficie y dejarlo liso. Posterior al proceso de rectificado un cilindro puede ser dirigido a ultrasonido, granallado, montaje (si el cilindro es de trabajo debe poseer pareja), emparejamiento (en caso de los cilindros de trabajo sin pareja), o inclusive volver al proceso de rectificado.
- **Granallado:** Los cilindros de trabajo pueden laminar diferentes tipos de materiales, los cuales poseen diferentes acabados, es por esto que es necesario adecuar la rugosidad de los cilindros, agregándole a la superficie del mismo un material denominado granalla, la cantidad de granalla depende de la rugosidad que



se desee para el cilindro. Posterior al proceso de granallado un cilindro puede ser dirigido a montaje o puede volverse a rectificar en caso de encontrar un defecto.

- **Emparejamiento:** Los cilindros de trabajo son agrupados por parejas, para ello deben ser compatibles, es decir, poseer una rugosidad, dureza y diámetro similar, las diferencias máximas permitidas entre los cilindros están definidas según patrones para las diferentes líneas y bastidores.
- **Montaje:** En esta fase al cilindro se le agregan a los lados unas cajas, quedando listo para ir a laminar.
- **Transporte:** El operador envía a los cilindros con sus respectivas cajas al laminador, certificando la calidad de los mismos.
- **Desmontaje:** En este proceso el cilindro es separado de las cajas, las cuales pueden ser enviadas o no a mantenimiento, dependiendo de una inspección visual realizada por el operador y/o supervisor. Los cilindros son enviados a rectificado.

Teniendo ya un conocimiento de como se maneja el Taller de Cilindro, se define el problema como el desarrollo de un sistema de gestión que permita controlar todos los procesos del Taller de Cilindros, que establezca una mayor sincronización a la hora de ejecutar las tareas por parte de cada uno de los usuarios que laboran. Los mismos tendrán la posibilidad de realizar una pronta toma de decisiones, debido al gran flujo de datos en línea.

El manejo de información deberá tener un mayor grado de confiabilidad, se requiere la eliminación del papel, absorbiendo la gran cantidad de sistemas de información no formales y globalizar el acceso a los datos, cerrando toda posibilidad de inconsistencias y/o errores de duplicidad en los mismos.



El desarrollo de aplicaciones y módulos bajo el sistema operativo QNX, es una de las actividades iniciales del proyecto. El Sistema De Gestión En Tornería De Cilindros, contará con los siguientes módulos:

- Módulo de ultrasonido
- Módulo de rectificado
- Módulo de granallado
- Módulo de parejas de cilindros
- Módulo de montaje
- Módulo de transporte de cilindros
- Módulo de desmontaje
- Módulo de montaje de muelas
- Módulo de agregado de granalla
- Aplicación de mantenimiento de cajas para los cilindros

en los cuales es indispensable la ejecución de las siguientes actividades:

- Análisis de requerimientos en Tornería De Cilindros en el área de Laminación en Frío.
- Levantamiento de la información que deberá manejar el sistema y que es indispensable para el personal que va a interactuar con la aplicación.
- Diseño de la base de datos.
- Diseño de todos los sub-módulos que contendrá la aplicación (Diseño de diagramas de flujo de datos)
- Diseño y validación de prototipos.
- Implementación de la base de datos en Sybase.
- Desarrollo de la aplicación: Se utilizará QNX, tomando las virtudes de un S.O de tiempo real para la adquisición de datos de las máquinas, mejorando la integridad y confiabilidad de datos.
- Desarrollo de conectores que permitirán el flujo de información entre los niveles de planta (nivel 1) y niveles de explotación de datos (nivel 2), con el fin de poder

1.6 Plataforma Estándar De Automatización - SIDOR

En la Figura 8 se visualiza la plataforma que manejará el STC

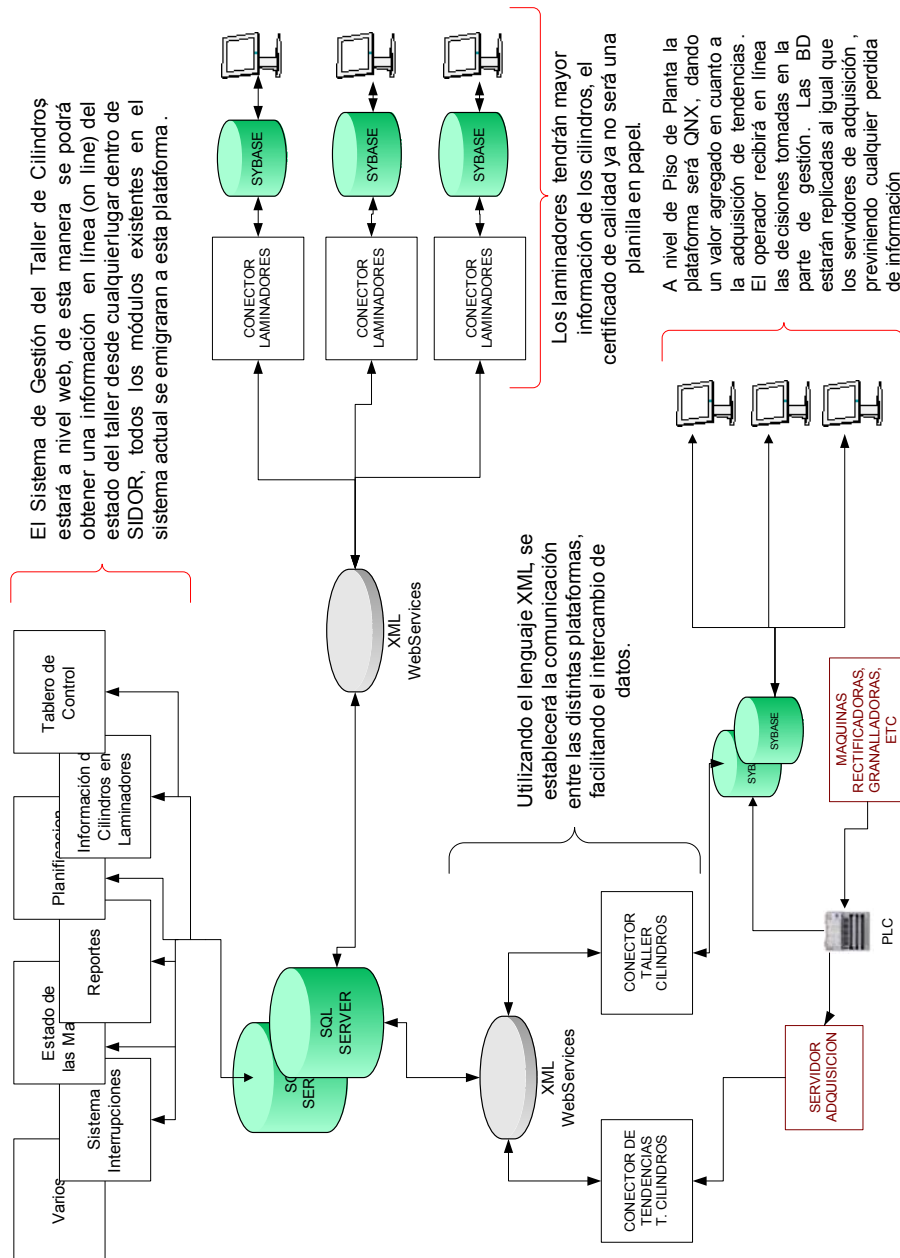


Figura 8: Plataforma Estándar Automatización - SIDOR



La instalación de esta plataforma permitirá:

La estandarización de esquema de servidores y equipos:

Se instalará un esquema de replicación de datos acorde con los estándares de SIDOR, así como también PC industriales en todos los lugares de operaciones

La unificación la información en una base de datos centralizada SQL Server:

Permitiendo realizar la explotación de datos de una manera estándar y única.

Sustitución de la red:

Dando mayor capacidad de transferencia de datos y seguridad de respuesta.

Cambiar la plataforma Windows en los sitios de operaciones:

Implantando QNX, tomando las virtudes de SO de tiempo real para la adquisición de datos de las máquinas, mejorando la integridad y confiabilidad de datos

1.7 Objetivos Generales

Desarrollar un Sistema de Gestión que permita registrar toda la información de los procesos dentro del Taller de Cilindros en el área de Laminación en Frío, absorbiendo todos los sistemas independientes y no estandarizados que funcionan en la actualidad. Eliminar el uso de planillas, que contribuirá a tener información congruente y confiable. Centralizando la información en una única base de datos para facilitar la explotación de datos necesaria para la toma de decisiones.



1.8 Objetivos Específicos

1. Analizar el proceso productivo del taller de cilindros, con el fin de obtener un conocimiento amplio del mismo, que sirva de base para el desarrollo del sistema.
2. Automatizar de manera eficiente los diferentes procesos del Taller de Cilindros, como lo son: ultrasonido, rectificado, granallado, emparejamiento, montaje y desmontaje de cajas, transporte, montaje y desmontaje de ruedas abrasivas.
3. Reducir el tiempo empleado por los supervisores del taller en la planificación de cilindros, por medio de la implementación de una programación continua de los cilindros a las diferentes máquinas.
4. Prolongar la vida útil de los cilindros, por medio de una gestión eficiente, facilitada por el sistema.
5. Centralizar la explotación de datos, para ello es necesario la creación de conectores que permiten el flujo de información entre los diferentes niveles de automatización, con la finalidad de generar reportes y gráficos en Web, para ser visualizados desde cualquier máquina en la Intranet.



1.9 Metodología a utilizar

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 completo, 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. [Serrano 1996]

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 el Proceso Unificado 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, 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 de se llevará a cabo en cuatro fases: inicio, elaboración, construcción y transición. [LARMAN 2003]

1.10 Alcance

El proyecto consiste en la implementación de un Sistema de Gestión para el Taller de Cilindros en el área de Laminación en Frío, este funcionará en dos plataformas diferentes, QNX para el desarrollo de las interfaces que serán manejadas por los operadores y que interactúan directamente con el proceso productivo (Nivel 2a); Windows para el desarrollo de reportes Web, introducción de datos al sistema y desarrollo del módulo de planificación (Nivel 2b). Está última es de vital importancia para el funcionamiento del sistema, pero no forma parte del trabajo realizado para proyecto de grado.



I.11 Estructura del Documento.

- Capítulo I: ***Introducción***
- Capítulo II: ***Marco Teórico.*** Se describe detalladamente los conceptos relacionados a la metodología de desarrollo, así como aquellos relacionados intrínsecamente al sistema.
- Capítulo III: ***Especificación de Requerimientos y Diseño.*** Se describen las técnicas de evaluación de requerimientos de diseño de acuerdo a las especificaciones exigidas por el cliente.
- Capítulo IV: ***Elaboración.*** Especifica toda la evolución o “ciclo de vida” del sistema.
- Capítulo V: ***Construcción y transición.*** Se describen los pasos realizados durante la fase de construcción del sistema y la fase de transición que cubre el período durante el cual el sistema llega a su versión operativa inicial, denominada versión beta.
- Conclusiones.
- Recomendaciones.
- Bibliografía.
- Apéndices.



CAPITULO II: MARCO TEÓRICO



II.1 Proceso Unificado

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software (véase Figura 9). Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.



Figura 9: Un Proceso de Desarrollo de Software.

El proceso unificado está basado en componentes, lo cual quiere decir que el sistema de software en construcción está formado por **componentes** de software interconectados a través de **interfaces bien** definidas.

El proceso unificado utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial del Proceso Unificado – sus desarrollos fueron paralelos. [LARMAN 2003]



II.1.1 Características del Proceso Unificado

Entre las características más resaltantes del Proceso Unificado se encuentran:

- **Centrado en los modelos:** Los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.
- **Guiado por los casos de uso:** Los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.
- **Centrado en la arquitectura:** Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.
- **Iterativo e incremental:** Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo (ver Figura 10).

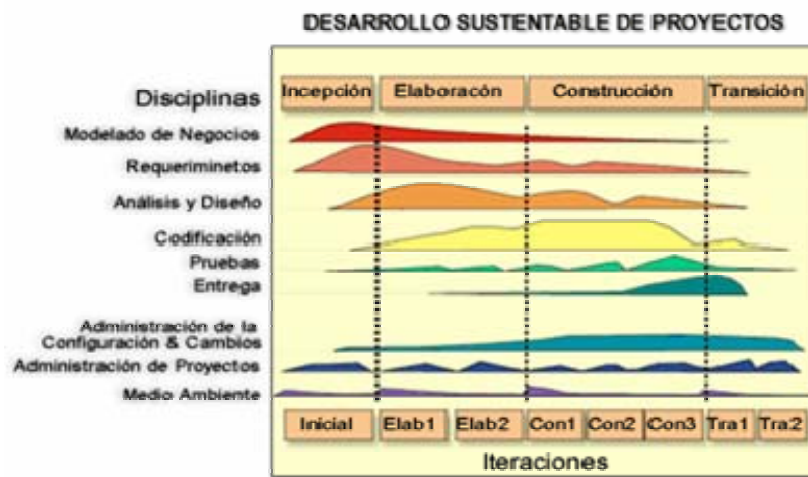


Figura 10: Estructura Completa del Proceso Unificado

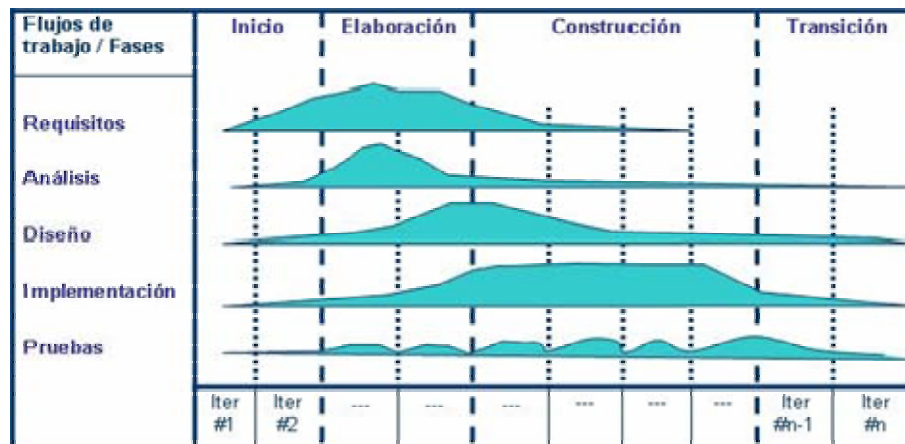


Figura 11: Estructura Básica del Proceso Unificado

El Proceso Unificado organiza los proyectos en términos de flujos de trabajo y fases, consistiendo cada una en una o más iteraciones (ver Figura 11). Con la aproximación iterativa, el énfasis de cada flujo de trabajo variará a través del ciclo de vida.

Cada ciclo de iteración en el proyecto comienza con un plan para lo que será logrado y concluye con una evaluación para definir si los objetivos han sido alcanzados. Esto facilita la comunicación entre el equipo, un factor crítico para el éxito de cualquier proyecto, permitiéndoles a los equipos de desarrollo reducir el tiempo, incrementando la predictibilidad del software que producen.

Fue desarrollado lado a lado con el estándar industrial UML y le da a cada miembro del equipo parámetros prescriptivos para una implementación exitosa de las seis mejores prácticas que permiten un desarrollo eficiente de aplicaciones empresariales de alta calidad:

- Desarrollar iterativamente para mitigar el riesgo tempranamente en el proyecto
- Administrar efectivamente los requerimientos
- Modelar visualmente para manejar la complejidad
- Usar arquitecturas por componentes para construir arquitecturas resistentes
- Verificar la calidad a través de todo el ciclo de vida
- Controlar los cambios en el software



II.1.2 Etapas del Proceso Unificado:

Las fases del ciclo de vida del software son: inicio, elaboración, construcción y transición. La concepción es definir el alcance del proyecto y definir el caso de uso. La elaboración es proyectar un plan, definir las características y cimentar la arquitectura. La construcción es crear el producto y la transición es transferir el producto a sus usuarios.

II.1.3 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.



II.1.4 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.

II.1.5 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, completando así la descripción y creación del producto propuesto, preparado ya para las últimas pruebas, antes de su entrega al usuario.

II.1.6 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.

II.2 Lenguaje de Modelado Unificado

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.

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 12 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.

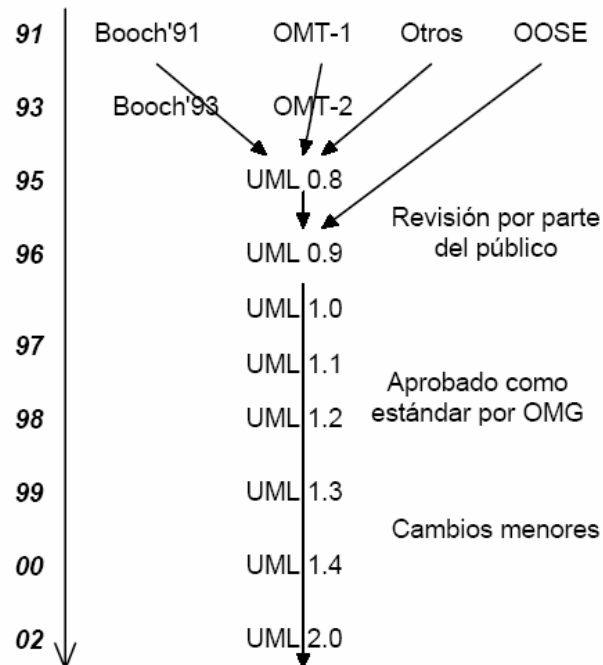


Figura 12: Historia del UML

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:

Los casos de uso son una técnica para especificar el comportamiento de un sistema: “Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios, dicho de otra manera, un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario”. Todo sistema de software ofrece a su entorno una serie de servicios. Un caso de uso es una forma de expresar cómo alguien o algo externo a un sistema lo usa.

Actores:

Un actor es una agrupación uniforme de personas, sistemas o máquinas que interactúan con el sistema que se está construyendo. Los actores son externos al sistema a desarrollar. Por lo tanto, al identificar actores también se está empezando a delimitar el sistema, y a definir su alcance.

Caso de uso:

Gráficamente, los casos de uso se representan con un óvalo, con el nombre del caso en su interior. Los casos de uso tienen las siguientes características:

- Están expresados desde el punto de vista del actor.
- Se documentan con texto informal.
- Describen tanto lo que hace el actor como lo que hace el sistema cuando interactúa con él, aunque el énfasis está puesto en la interacción.
- Son iniciados por un único actor.



- Están acotados al uso de una determinada funcionalidad claramente diferenciadas del sistema.

Alternativas:

Durante la ejecución de un caso de uso, suelen aparecer errores o excepciones. Por ejemplo, mientras se ingresa un pedido, el cliente puede solicitar un producto que está discontinuado. El sistema deberá en este caso informar esta situación al empleado que ingresa el pedido. Esas desviaciones del curso normal del caso de uso se llaman alternativas. Las alternativas tienen las siguientes características:

- Representan un error o excepción en el curso normal del caso de uso.
- No tienen sentido por sí mismas, fuera del contexto del caso de uso en el que ocurren.

Relaciones de extensión:

Muchas veces, la funcionalidad de un caso de uso incluye un conjunto de pasos que ocurren sólo en algunas oportunidades. Las extensiones tienen las siguientes características:

- Representan una parte de la funcionalidad del caso que no siempre ocurre.
- Son un caso de uso en sí mismas.
- No necesariamente provienen de un error o excepción. En su libro, Jacobson ejemplifica los casos de uso con ir a cenar a un restaurante. Para él, tomar café después de cenar es un ejemplo de una extensión. [JACOBSON 2000]

Relaciones de uso:

Es común que la misma funcionalidad del sistema sea accedida a partir de varios casos de uso. Este concepto no es novedoso, es simplemente el concepto de la subrutina o subprograma usado en un nivel más alto de abstracción. Las características de las relaciones de uso son:

- Aparecen como funcionalidad común, luego de haber especificado varios casos de uso.
- Los casos usados son casos de uso en sí mismos.



- El caso es usado siempre que el caso que lo usa es ejecutado. Esto marca la diferencia con las extensiones, que son opcionales.

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.

II.3 Sistema Operativo QNX

QNX es un sistema operativo de tiempo real, flexible, eficiente, modular basado en dos principios:

- **Arquitectura de MICROKERNEL.**
- **IPC basado en mensajes.**

QNX consiste en un microkernel y un conjunto de procesos cooperantes. El microkernel de QNX ocupa menos de 8Kbytes, y solo cumple funciones de IPC y scheduling, en contraposición de otros sistemas basados en “kernel’s”, donde este toma tantas funciones que el sistema operativo completo se encuentre dentro de este “kernel”.

II.3.1 Características:

- **Multitareas (Multitasking):** Ejecución de múltiples procesos sirviendo a múltiples usuarios en múltiples máquina cooperantes. Implica IPC, prioridades, sistema



preemptivo, etc.

- **Transportable (POSIX):** Posix es un standard desarrollado por la IEEE (Institute of Electric and Electronic Engineers). Asegura la reutilización de código fuente entre los sistemas operativos que cumplan la norma. Especifica la interfaz que el S.O. ofrece a las aplicaciones y usuarios.
- **Basado en Administradores:** Modular. Un microkernel a cargo de la coordinación general, y una serie de administradores de los diferentes recursos del sistema (Ej. Fsys: discos, Proc: CPU, Dev: Dispositivos)
- **Distribuido:** Indica la posibilidad de conectar una serie de máquinas en una red local y distribuir la ejecución de las aplicaciones a través de ellas.
- **Tiempo Real:** Multitasking - Priority Driven Preemptive Sheduler - Fast Context Switching.
- **Comunicación basado en Pasaje de Mensajes:** Bloqueo de procesos (sincronización), IPC.

II.4 Sybase SQL Anywhere para QNX

Sybase SQL Anywhere es un sistema manejador de base de datos bajo el enfoque objeto-relacional (procesamiento de transacciones de sentencias SQL). Su característica principal es la optimización de los recursos (uso de memoria RAM, disco rígido y CPU) para su buen desempeño.

Sybase SQL Anywhere para QNX posee un alto rendimiento en cuanto a la atención concurrente de un amplio número de usuarios con tiempos de respuesta realmente cortos.

El siguiente diagrama representa la conexión de las aplicaciones de usuarios a los servidores de bases de datos Sybase SQL Anywhere bajo QNX.

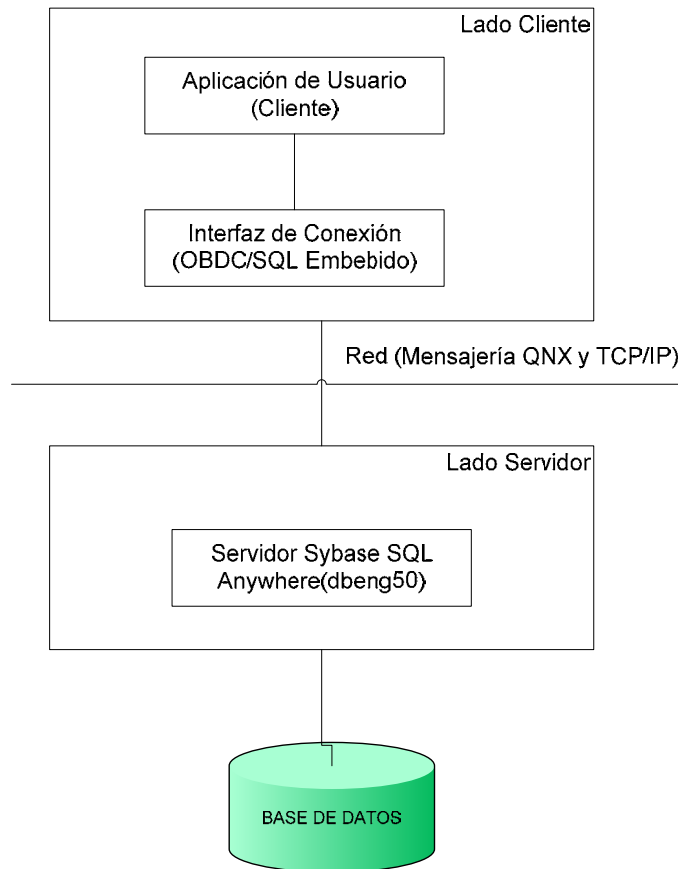


Figura 13: Conexión de Aplicaciones a Servidores de Bases de Datos Sybase bajo QNX

En este esquema se muestra la aplicación de usuario con la interfaz de conexión a la base de datos empleada: ODBC o SQL Embebido. La aplicación de usuario no manipula directamente la base de datos y el log de transacciones, estos archivos son solo manipulados por el servidor de base de datos (dbsrv05). La comunicación entre la aplicación cliente y el servidor es mensajería QNX o TCP/IP.

La aplicación de usuario que se conecta a la base de datos no requiere de una aplicación adicional para la conexión a la base de datos, ya que esta forma parte de la misma aplicación al ser compilada y enlazada con las librerías correspondientes, según el tipo de interfaz de conexión de base de datos.



II.5 XML (Lenguaje de Marcas Extensible)

XML fue desarrollado por un grupo de trabajo bajo los auspicios del consorcio World Wide Web (W3C) a partir de 1996. Este fue constituido en 1994 con el objetivo de desarrollar protocolos comunes para la evolución de Internet. Se trata de un consorcio de la industria internacional con sedes conjuntas en el Instituto Tecnológico de Massachussets, de Estados Unidos, el Instituto Nacional de Investigación en Informática y Automática europeo y la Keio University Shonan Fujisawa Campus de Japón. El W3C tiene como misión la publicación para uso público de protocolos o estándares globales de uso libre.

La mayoría de la gente ha oído hablar de XML (Lenguaje de Marcas Extensible Markup Language), siendo muy corriente que se tenga la idea de que es una especie de HTML (Hypertext Markup Language) avanzado, por lo que siempre que sale el tema de XML surgen cuestiones fundamentales como: ¿qué es exactamente XML?, ¿es XML una nueva versión de HTML?, ¿para que sirve?, ¿cuáles fueron los motivos de su creación?, ¿tiene alguna aplicación en la actualidad?

Como es sabido HTML se ha convertido en un lenguaje de inmensa popularidad durante los últimos años. También debemos notar que nos hemos encontrado con sus propias limitaciones, que algunas de ellas se han querido subsanar con scripts, javascripts, Active X, HTML dinámico, etc; pero en la realidad todas estas herramientas no aportan una solución global a las limitaciones del HTML. Quizá su diseño estaba previsto para Windows NT y no funciona en plataformas UNIX, o hacía uso de Perl pero no se integraba con Java. La implantación de XML viene a eliminar este tipo de situaciones permitiendo la creación de herramientas más estructuradas.

XML es un lenguaje de marcas que ofrece un formato para la descripción de datos estructurados, el cual conserva todas las propiedades importantes del SGML. Es decir, XML es un metalenguaje, dado que con él podemos definir nuestro propio lenguaje de presentación y, a diferencia del HTML, que se centra en la representación de la

información, XML se centra en la información en si misma. La particularidad más importante del XML es que no posee etiquetas prefijadas con anterioridad, ya que es el propio diseñador el que las crea a su antojo, dependiendo del contenido del documento. De esta forma, los documentos XML con información sobre libros deberían tener etiquetas como <AUTOR>, <EDITORIAL>, <Nº_DE_PÁGINAS>, <PRECIO>, etc., mientras que los documentos XML relacionados con educación incluyen etiquetas del tipo de <ASIGNATURA>, <ALUMNO>, <CURSO>, <NOTA>, etc.

XML es un formato basado en texto, subconjunto de SGML, específicamente diseñado para almacenar y transmitir datos. Un documento XML se compone de elementos XML, cada uno de los cuales consta de una etiqueta de inicio, de una etiqueta de fin y de los datos comprendidos entre ambas etiquetas. Al igual que los documentos HTML, un documento XML contiene texto anotado por etiquetas. Sin embargo, a diferencia de HTML, XML admite un conjunto ilimitado de etiquetas, no para indicar el aspecto que debe tener algo, sino lo que significa. Por ejemplo: un elemento XML puede estar etiquetado como precio, número de pedido o nombre. El autor del documento es quien decide qué tipo de datos va a utilizar y qué etiquetas son las más adecuadas. [VILLATE 2001]

II.6 ¿Qué es un Servicio Web?

Un **servicio Web** es una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes y ejecutadas sobre cualquier plataforma pueden utilizar los servicios Web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

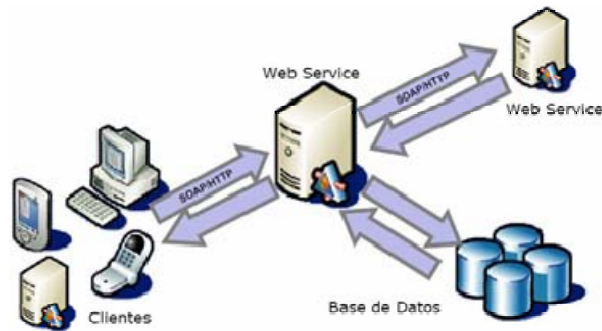


Figura 14: Servicios Web en Acción

II.7 SOAP

SOAP es un protocolo basado en XML para el intercambio de información de una forma descentralizada sobre entornos distribuidos.

Define un mecanismo para el paso de instrucciones (comandos) y parámetros entre clientes y servidores. Es totalmente independiente de la plataforma, el modelo de datos y el lenguaje de programación usado.

Estructura de SOAP

- Requiere un cuerpo (body) soap y un sobre (envelop).
- Cabeceras (header) (opcionales).



Figura 15: Estructura del SOAP



Algunas de las ventajas de SOAP son:

No se encuentra fuertemente asociado a ningún protocolo de transporte: La especificación de SOAP no describe como se deberían asociar los mensajes de SOAP con HTTP. Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.

No está atado a ninguna infraestructura de objeto distribuido La mayoría de los sistemas de objetos distribuidos se pueden extender, y ya lo están alguno de ellos para que admitan SOAP.

Aprovecha los estándares existentes en la industria: Los principales contribuyentes a la especificación SOAP evitaron, intencionadamente, reinventar las cosas. Optaron por extender los estándares existentes para que coincidieran con sus necesidades. Por ejemplo, SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema de tipo que ya están definidas en la especificación esquema de XML. Y como ya se ha mencionado SOAP no define un medio de transporte de los mensajes; los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.



CAPITULO III: ESPECIFICACIÓN DE REQUERIMIENTOS Y DISEÑO



III INICIO

El proceso unificado establece que durante la etapa de inicio debe hacerse un análisis de requerimientos, que permita definir algunas características generales importantes del proyecto. Sin embargo, el objetivo de la fase de inicio no es definir todos los requisitos, o generar una estimación creíble o plan de proyecto. Aún a riesgo de simplificar demasiado, la idea es hacer la investigación justa para formar una opinión racional y justificable del propósito global y la viabilidad del sistema, y decidir si merece la pena invertir en un estudio más profundo.

III.1 Visión General:

En el Taller de Cilindros de Laminación en Frío de SIDOR, la captura de datos que es utilizada para la toma de decisiones se realiza en diversos sistemas de información, como el SAP, Planillas, BDI, SISCOT (Sistema actual de Gestión), dichos sistemas son incongruentes entre sí. Esto obliga a los usuarios (personal del taller de Cilindros) a destinar gran parte de su tiempo en recopilar y corregir la información que necesitan, restándole tiempo al análisis, resolución de los problemas y gestión de la información.

Por lo tanto se requiere la implementación de un sistema automatizado que facilite la toma de decisiones y permita mantener información fiable de cada uno de los procesos que se ejecutan en el mismo. Es indispensable la eliminación de las planillas y evitar el manejo manual de las bases de datos.

El Sistema de Gestión de Taller de Cilindros deberá contener diversos módulos, entre los que se encuentran:

- Módulo de ultrasonido
- Módulo de rectificado



- Módulo de granallado
- Módulo de parejas de cilindros
- Módulo de montaje
- Módulo de transporte de cilindros
- Módulo de desmontaje
- Módulo de montaje de muelas
- Módulo de agregado de granalla
- Aplicación de mantenimiento de cajas para los cilindros

Estas aplicaciones serán desarrolladas para llevar un registro automático de cada proceso por la cual pasan los cilindros en su estadía en el Taller de Cilindros, serán manejadas directamente por los operadores de la planta, es por ello que su programación es en el Sistema Operativo QNX (nivel 2 de la pirámide de automatización). Además se requiere la explotación de estos datos, para que sirvan de base para la toma de decisiones y permitir una gestión más eficiente y confiable, para ello se hace indispensable el desarrollo de aplicaciones y reportes web. Es vital que la información que se maneje en el nivel operativo (nivel 2 A) sea congruente con la información del nivel superior (nivel 2 B), para tal fin, es necesario el desarrollo de conectores, que por medio del lenguaje XML se pueda transmitir la información entre dichos niveles.

III.2 Modelo de Casos de Uso:

El procedimiento básico para definir los casos de uso se rige por los siguientes pasos:

III.2.1 Límites del Sistema

A nivel de piso de planta la plataforma será QNX, dando un valor agregado en cuanto a la adquisición de tendencias y valores tomados directamente de las máquinas, el operador recibirá en línea las decisiones de gestión y los tiempos de producción tendrán un grado mas de confiabilidad. A nivel de gestión la plataforma será Windows, todos los sistemas



independientes serán llevados a aplicaciones Web según los estándares de SIDOR. La planificación de cilindros para los diferentes procesos de planta también será implantada a nivel Web. La explotación de datos se realizará por medio de reportes, que servirán de herramientas para los analistas a la hora de toma de decisiones, implicando una disminución en los tiempos de producción, esta información se podrá acceder por la intranet en cualquier parte dentro de SIDOR.

El sistema de gestión en tornería de cilindros tendrá iteración con el Sistema de cambio de cilindros en los laminadores, ya que esta última requiere información de los cilindros que son enviados del taller al laminador; cuando se efectúa un cambio de cilindros en el laminador, dicho sistema envía información del kilometraje y tonelaje al STC.

III.2.2 Actores Principales

Existe un conjunto de actores que interactúan con el sistema, estos se pueden clasificar según el rol que desempeñan dentro del mismo. Estos pueden ser:

OPERADORES: Son los encargados de realizar la carga de datos en el sistema, interactúan directamente con todos los módulos de piso de planta. Este accede a la aplicación por medio de un sistema de validación de usuarios. No poseen permisos especiales.

SUPERVISORES: Son los encargados de justificar la ocurrencia de ciertas acciones dentro del proceso, es decir, en el momento de la carga de datos pueden existir valores fuera de los patrones preestablecidos y es necesario que quede una observación registrada en el sistema y un usuario responsable de dicho evento.

ANALISTAS: Son los encargados de la toma de decisiones en cada uno de los diferentes procesos productivos del Taller de Cilindros. Tienen acceso tanto a las aplicaciones de piso de planta como a las aplicaciones web, poseen permisos especiales.



III.2.3 Actores – Actividades

Tabla 2: Lista de Actores – Actividades del Sistema

Actor	Actividades
OPERADOR	<ul style="list-style-type: none">▪ Registrar y reportar el proceso de ultrasonido.▪ Registrar y reportar el proceso de rectificado.▪ Registrar y reportar el proceso de granallado.▪ Registrar y reportar el montaje de cilindros.▪ Registrar y reportar el emparejamiento de cilindros.▪ Registrar y reportar el proceso de transporte de cilindros.▪ Registrar y reportar la recepción de cilindros en el laminador.▪ Registrar y reportar el proceso de desmontaje de cilindros.▪ Registrar y reportar el proceso de agregado de granalla a la máquina granalladora.▪ Registrar y reportar el proceso de montaje y desmontaje de muelas.▪ Registrar y reportar el mantenimiento de cajas.
SUPERVISOR	<ul style="list-style-type: none">▪ Justifica valores fuera de patrón en el proceso de rectificado.▪ Justifica valores fuera de patrón en el mantenimiento de cajas.▪ Planifica los cilindros para los diferentes procesos y para las diferentes máquinas, escogiendo previamente el patrón de proceso para dicho cilindro.
ANALISTA	<ul style="list-style-type: none">▪ Analiza los datos provenientes de los diferentes procesos y toman decisiones sobre eventos imprevistos y/o cilindros defectuosos.▪ Carga la información con la cual se alimenta el sistema: situaciones, defectos, patrones, etc.



III.2.4 Requisitos Funcionales

Los requisitos definen las características, capacidades y seguridad que el sistema debe contener, 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. En el Sistema de Gestión para el Taller de Cilindros, podemos identificar los siguientes requisitos funcionales:

- Permitir llevar un registro del proceso de ultrasonido para los cilindros de trabajo y cilindros de apoyo. Donde se especifiquen datos como las durezas del cilindro, el aparato utilizado, el palpador y el tipo de ensayo realizado.
- Permitir llevar un registro del proceso de granallado para los cilindros de trabajo. Donde se especifiquen datos como la rugosidad, el rugosímetro y la placa patrón utilizada, pases de granalla y la velocidad aplicada, incluyendo datos de ubicación del cilindro, como el área, estante y posición dentro del estante.
- Permitir llevar un registro del proceso de rectificado para los cilindros de trabajo y de Apoyo, se deben guardar datos tales como, temperatura, diámetro, rugosidad, durezas, excentricidad, corona, medidas de chaflán, defectos del cilindro con su grado y distancia, micrómetro y rugosímetro utilizados, línea y bastidor a donde se envía el cilindro, incluyendo datos de ubicación del cilindro, como el área, estante y posición dentro del estante. Todos estos datos deben ser validados contra un patrón que debe ser elegido previo al proceso de rectificado.
- Permitir formar parejas de cilindros de trabajo, los cuales podrán ser unidos dependiendo de las características de los mismos, tales como, diámetro, rugosidad, datos de línea y bastidor.
- Se requiere llevar un control del proceso de montaje de cilindros, para ello se deben escoger las cajas dependiendo si son cilindros de apoyo o de trabajo.



- Se requiere un control de los cilindros mandados del taller al laminador.
- Permitir llevar un registro de los cilindros que son desmontados del laminador y las cajas que están libres para ser usadas por otros cilindros o que debido a su tonelaje o kilometraje deben ser mandadas al mantenimiento de cajas.
- Se requiere llevar un control de la cantidad y tipo de granalla agregada a la máquina granalladora.
- Se requiere llevar un control de la muela utilizada para los rectificadores, el momento en que fue montada y desmontada, y el consumo (disminución de diámetro) de la misma.
- Es indispensable que el proceso utilizado para el mantenimiento de cajas sea totalmente automático, para lo cual, se requiere la eliminación de todas las planillas utilizadas para tal fin, así como también, permitir llevar un control exacto del momento en que las cajas requieren mantenimiento (calculado por medio del tonelaje y kilometraje de cada caja).

III.2.5 Casos de Uso

Los casos de uso pueden extraerse del conjunto de ítems de la relación actores – objetivos, tomando a cada objetivo como un caso de uso específico. Una actividad práctica útil es el establecer prioridades o niveles de riesgo, véase Tabla 3; esto se refiere a seleccionar un subconjunto de casos de uso más importantes y comenzar a desarrollarlos progresivamente hasta llegar a los considerados menos importantes.



Tabla 3: Lista de Casos de Uso Clasificados por Prioridad

Caso de Uso	Descripción	Prioridad
Procesar Ultrasonido	Registro de toda la información concerniente al proceso de ultrasonido.	1
Procesar Rectificado	Registro de toda la información concerniente al proceso de rectificado.	1
Procesar Granallado	Registro de toda la información concerniente al proceso de granallado.	1
Formar parejas de Cilindros	Emparejamiento de los cilindros de trabajo tomando en consideración los diámetros y durezas,	1
Procesar Montaje	Registro de los datos del proceso de montaje de cajas en los cilindros de apoyo y cilindros de trabajo.	2
Transporte de cilindros	Listado de los cilindros que son enviados del taller al laminador.	3
Desmontaje de los Cilindros.	Registro de la información del proceso de desmontaje de las cajas de los cilindros tanto de apoyo como de trabajo.	2
Montar Muela	Control de Montaje de Muela en la máquina rectificadora.	3
Desmontar Muela	Control del desmontaje de Muela de la máquina rectificadora.	3
Procesar Agregado de Granalla	Control de la cantidad de granalla agregada a la máquina granalladora.	3
Procesar Perfil de dureza	Registro de los datos correspondientes a las mediciones de durezas realizadas sobre los cilindros.	3
Mantenimiento de Cajas	Registro de los datos de la verificación del estado de las cajas, registro de los datos tomados en el mantenimiento de cajas. Control de cajas que deben ser sacadas de funcionamientos. Control de los rodamientos de las cajas.	1

III.2.6 Modelos de Casos de Uso:

En la Figura 16 se visualiza el modelo de casos de uso de la fase de inicio.

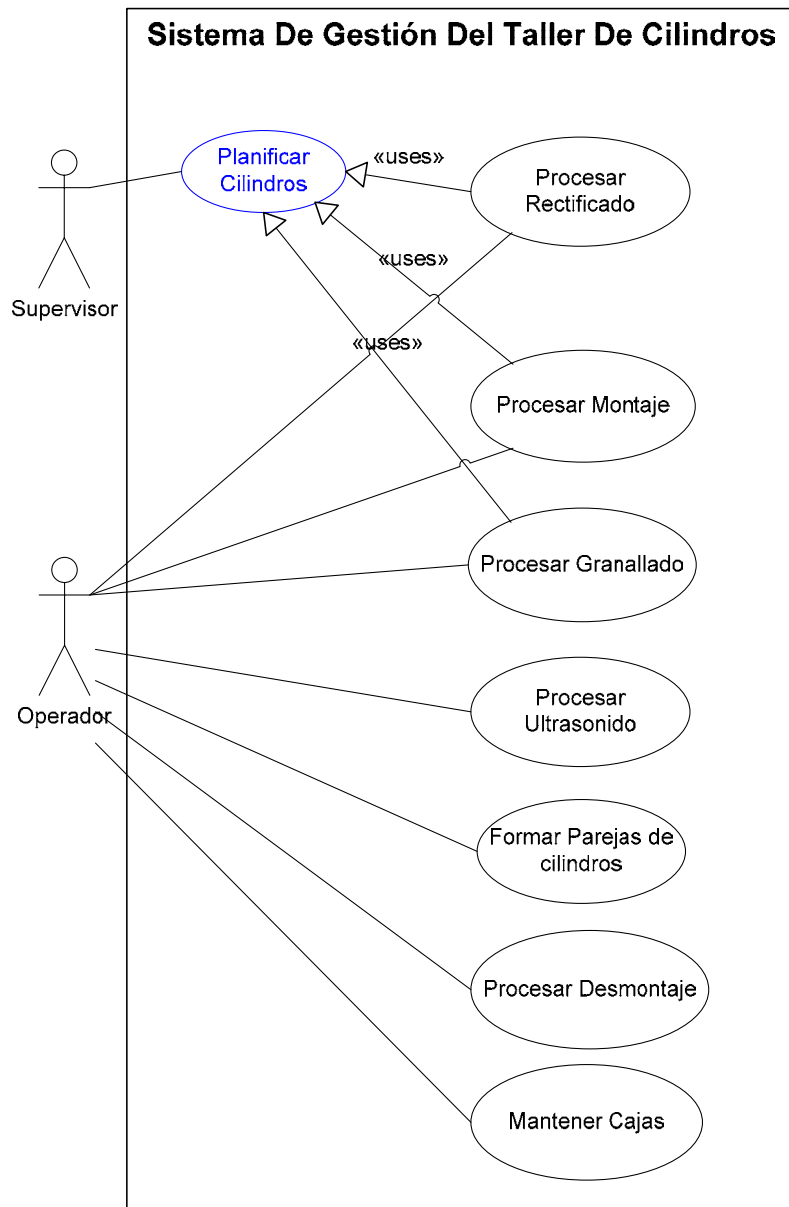


Figura 16: Diagrama de Casos de Uso del STC



III.2.7 Definición de Casos de Uso

Caso de uso: Procesar Ultrasonido
<i>Actor: Operador</i>
<i>Precondiciones:</i> El operador se identifica y autentica.
<i>Garantías de éxito:</i> Se registra el proceso de ultrasonido. Se actualiza la información en el registro del cilindro.
<i>Escenario principal de éxito:</i> <ol style="list-style-type: none">1. El operador escoge de la lista que se presenta el cilindro que desea procesar.2. El operador visualiza información básica del cilindro y pasa a montar el cilindro en la pantalla principal.3. El cilindro se monta correctamente, es decir, está listo para registrar el proceso de ultrasonido.4. El sistema genera una pantalla donde serán introducidos los datos del ultrasonido.5. El operador inserta el aparato y el palpador utilizado en el ultrasonido.6. El operador inserta el tipo de ensayo realizado en proceso.7. El operador guarda todos los datos.8. El operador pasa el cilindro a estado procesado (desmonta el cilindro).9. El cilindro es visualizado en la lista de ultrasonidos procesados.10. El operador escoge la opción de reportar el ultrasonido.11. El sistema hace una llamada al conector de reportar ultrasonidos.12. Se arma un archivo XML con toda la información de ultrasonido y del cilindro.13. Por medio del SOAP, la información es enviada al “web service.”14. El “web service” la recibe y manda un código de éxito o de error, dependiendo de si la información está correcta o no.15. En caso de que el “web service” mande un código de error, el operador deberá volver intentar el envío de información.
<i>Escenario Alternativo:</i>



- 2.a El operador trata de montar un cilindro ya estando montado otro
1. El sistema genera un mensaje de error.
- 3.a El operador elige un cilindro equivocado.
1. El operador selecciona “sacar secuencia”.
 2. El cilindro desaparece de la zona de montaje en ultrasonido.
 3. El Sistema reestablece todas las condiciones iniciales.
- 9.a Luego de procesar el cilindro, el operador puede elegir eliminar el registro de ultrasonido.
1. El sistema restaura las todas las situaciones iniciales.
 2. Se elimina el cilindro de la lista de cilindros procesados.
 3. Se visualiza el cilindro en la lista de cilindros listos para realizar Ultrasonido.
- 9.b Luego de procesar el cilindro, el operador puede elegir actualizar los datos de ultrasonido.
1. El operador modifica cualquier dato.
 2. El operador guarda la información nuevamente.
 3. El sistema actualiza todos los datos.

Requisitos Especiales:

- Interfaz en QNX amigable y acorde con los estándares de visualización de la empresa.
- Tiempos de respuesta pequeños, con la finalidad de agilizar la búsqueda de información de manera eficiente, visualización de datos no mayor a 10 segundos.

Caso de uso: Procesar Rectificado

Actor: Operador

Precondiciones: Programar los cilindros que serán procesados en la máquina rectificadora. El operador se identifica y autentica.

Garantías de éxito: Se registra el proceso de rectificado. Se actualiza la información



en el registro del cilindro.

Escenario principal de éxito:

1. El sistema presentará una lista de los cilindros planificados para el proceso de rectificado, se encuentran ordenados por prioridad, es decir, los primeros de la lista serán los primeros que deben ser procesados.
2. El operador escoge de la lista que se presenta el cilindro que debe procesar, el cual siempre se encuentra de primero en la lista, si se escoge un cilindro diferente, el operador debe justificar la selección realizada.
3. El operador visualiza información básica del cilindro y pasa a montar el cilindro en la pantalla principal.
4. El cilindro se visualiza en la zona de producción de rectificado.
5. El sistema permite levantar una ventana donde serán introducidos todos los datos de rectificado (temperatura, diámetro medido, rugosidad salida, longitud y profundidad de chaflán, micrómetro, rugosímetro, corona, grado, bastidor, línea, excentricidades, durezas, tipo de material.)
6. El operador puede visualizar el patrón escogido en la planificación para el rectificado de dicho cilindro.
7. El operador puede actualizar los perfiles de durezas.
8. El operador guarda todos los datos.
9. El sistema compara todas las mediciones introducidas por el operador con el patrón de rectificado escogido para dicho cilindro en el momento de la planificación y valida que todos los datos estén correctos (que concuerden con el patrón).
10. El operador pasa el cilindro a estado procesado (desmonta el cilindro).
11. El cilindro es visualizado en la lista de rectificadores procesados.
12. El operador escoge la opción de reportar el rectificado.
13. El sistema hace una llamada al conector de reportar rectificadores.
14. Se arma un archivo XML con toda la información del rectificado y del cilindro.
15. Por medio del SOAP, la información es enviada al “web service”.
16. El “web service” la recibe y manda un código de éxito o de error, dependiendo



de si la información esta correcta o no.

17. En caso de que el “web service” mande un código de error, el operador deberá volver intentar el envío de información.

Escenario Alternativo:

3.a El operador trata de montar un cilindro ya estando montado otro.

1. El sistema genera un mensaje de error.

4.a El operador elige un cilindro equivocado.

1. El operador selecciona “sacar secuencia”.
2. El cilindro desaparece de la zona de montaje en rectificado.
3. El Sistema reestablece todas las condiciones iniciales.

9.a Si existe algún dato fuera del patrón

1. El sistema bloquea el salir de la aplicación.
2. Coloca en color rojo todos los datos fuera de patrón.
3. El sistema genera un mensaje explicativo de los errores.
4. Los datos pueden ser corregidos por el operador.
5. Sino se desea corregir, el supervisor deberá justificar porque estos datos fuera de patrón están siendo introducidos al sistema, para ello, deberá introducir su login y clave.

11.a Luego de procesar el cilindro, el operador puede elegir eliminar el registro de rectificado.

1. El sistema restaura las todas las situaciones iniciales.
2. Se elimina el cilindro de la lista de cilindros procesados.
3. Se visualiza el cilindro en la lista de cilindros listos para realizar rectificado.

11.b Luego de procesar el cilindro, el operador puede elegir actualizar los datos de rectificado.

1. El operador modifica cualquier dato.
2. El operador guarda la información nuevamente.
3. El sistema actualiza todos los datos.

Requisitos Especiales:



- Interfaz en QNX amigable y acorde con los estándares de visualización de la empresa.
- Tiempos de respuesta pequeños, con la finalidad de agilizar la búsqueda de información de manera eficiente, visualización de datos no mayor a 10 segundos.

Caso de uso: Procesar Granallado

Actor: Operador

Precondiciones: El operador se identifica y autentica.

Garantías de éxito: Se registra el proceso de granallado. Se actualiza la información en el registro del cilindro.

Escenario principal de éxito:

1. El operador escoge de la lista que se presenta el cilindro de Trabajo que desea procesar.
2. El operador visualiza información básica del cilindro y pasa a montar el cilindro en la pantalla principal.
3. El cilindro se visualiza en la zona de producción de granallado.
4. El sistema permite levantar una ventana donde serán introducidos todos los datos de granallado.
5. El operador debe introducir la ubicación destino del cilindro.
6. El operador debe introducir el rugosímetro utilizado en el proceso.
7. El operador debe introducir diez mediciones de rugosidad.
8. El operador puede introducir observaciones sobre el proceso de granallado.
9. El sistema permitirá introducir los diferentes pases para agregarle granalla al cilindro y cambiar así la rugosidad del mismo, permitiendo registrar el tipo de granalla por pase y la velocidad utilizada en el proceso.
10. El operador guarda todos los datos.
11. El operador pasa el cilindro a estado procesado (desmonta el cilindro).
12. El cilindro es visualizado en la lista de granallados procesados.
13. El operador escoge la opción de reportar el granallado.
14. El sistema hace una llamada al conector de reportar granallado.



15. Se arma un archivo XML con toda la información del granallado y del cilindro.
16. Por medio del SOAP, la información es enviada al “web service”.
17. El “web service” la recibe y manda un código de éxito o de error, dependiendo de si la información esta correcta o no.
18. En caso de que el “web service” mande un código de error, el operador deberá volver intentar el envío de información.

Escenario Alternativo:

- 2.a El operador trata de montar un cilindro ya estando montado otro.
 1. El sistema genera un mensaje de error.
- 3.a El operador elige un cilindro equivocado.
 1. El operador selecciona “sacar secuencia”.
 2. El cilindro desaparece de la zona de montaje en granallado.
 3. El sistema reestablece todas las condiciones iniciales.
- 12.a Luego de procesar el cilindro, el operador puede elegir eliminar el registro de granallado.
 1. El sistema restaura las todas las situaciones iniciales.
 2. Se elimina el cilindro de la lista de cilindros procesados.
 3. Se visualiza el cilindro en la lista de cilindros listos para realizar granallado.
- 12.b Luego de procesar el cilindro, el operador puede elegir actualizar los datos de granallado.
 1. El operador modifica cualquier dato.
 2. El operador guarda la información nuevamente.
 3. El sistema actualiza todos los datos.

Requisitos Especiales:

- Interfaz en QNX amigable y acorde con los estándares de visualización de la empresa.
- Tiempos de respuesta pequeños, con la finalidad de agilizar la búsqueda de información de manera eficiente, visualización de datos no mayor a 10 segundos.



Caso de uso: Formar Pareja de Cilindros
<i>Actor: Operador</i>
<i>Precondiciones:</i> El operador se identifica y autentica.
<i>Garantías de éxito:</i> Se registra la pareja de cilindros. Se actualiza la información en el registro de los cilindros.
<i>Escenario principal de éxito:</i> <ol style="list-style-type: none">1. El operador escoge de la lista que se presenta el cilindro que desea emparejar.2. El operador visualiza información básica del cilindro y pasa a montar el cilindro en la pantalla principal.3. El cilindro se visualiza en la zona de producción de emparejamiento.4. El sistema permite levantar una ventana donde aparece una lista de las posibles parejas de dicho cilindro, tomando en consideración los diámetros, tipo de material a laminar y rugosidad de ambos cilindros.5. La pareja óptima del cilindro es el primero de la lista, sin embargo el operador puede escoger cualquier cilindro de la lista.6. El operador guarda todos los datos de la pareja.7. El operador pasa el cilindro a estado procesado (desmonta el cilindro).8. La pareja es visualizada en la lista de parejas procesadas.9. El operador escoge la opción de reportar la pareja.10. El sistema hace una llamada al conector de reportar parejas.11. Se arma un archivo XML con toda la información de la pareja de cilindros.12. Por medio del SOAP, la información es enviada al “web service”.13. El “web service” la recibe y manda un código de éxito o de error, dependiendo de si la información esta correcta o no.14. En caso de que el “web service” mande un código de error, el operador deberá volver intentar el envío de información.
<i>Escenario Alternativo:</i> <ol style="list-style-type: none">2.a El operador trata de montar un cilindro ya estando montado otro.<ol style="list-style-type: none">1. El sistema genera un mensaje de error.



3.a El operador elige un cilindro equivocado.

1. El operador selecciona “sacar secuencia”.
2. El cilindro desaparece de la zona de cilindros en producción.
3. El Sistema reestablece todas las condiciones iniciales.

8.a Luego de procesar, el operador puede elegir eliminar la pareja previamente creada.

1. El sistema restaura las todas las situaciones iniciales.
2. Se elimina los cilindros de la lista de parejas procesadas.
3. Se visualizan los cilindros en la lista de cilindros listos para ser emparejados.

8.b Luego de procesar, el operador puede elegir actualizar la pareja, escogiendo otro cilindro para formar la pareja.

Requisitos Especiales:

- Interfaz en QNX amigable y acorde con los estándares de visualización de la empresa.
- Tiempos de respuesta pequeños, con la finalidad de agilizar la búsqueda de información de manera eficiente, visualización de datos no mayor a 10 segundos.

Caso de uso: Procesar Montaje

Actor: Operador

Precondiciones: Programar las parejas de cilindros de trabajo o los cilindros de apoyo que serán procesados en montaje. El operador se identifica y autentica.

Garantías de éxito: Se registra el proceso de montaje. Se actualiza la información en el registro del cilindro.

Escenario principal de éxito:

1. El sistema genera la lista de cilindros de apoyos o parejas de trabajo programadas para el proceso de montaje, esta lista esta organizada por prioridad, es decir, el primer cilindro o pareja será lo primero en procesarse.
2. El operador escoge de la lista que se presenta el cilindro de Apoyo o la pareja



de cilindros de trabajo que debe procesar.

3. El operador visualiza información básica del cilindro de apoyo o pareja de cilindros de trabajo y pasa a montar en la pantalla principal.
4. El sistema permite levantar una ventana donde será escogido el juego de cajas que será montado con los cilindros y la posición de los cilindros (inferior o superior).
5. El operador pasa el cilindro de apoyo o pareja de trabajo a estado procesado.
6. El proceso de montaje es visualizado en la lista de elementos procesados.
7. El operador escoge la opción de reportar el montaje.
8. El sistema hace una llamada al conector de reportar montaje.
9. Se arma un archivo XML con toda la información del montaje y de los cilindros.
10. Por medio del SOAP, la información es enviada al “web service”.
11. El “web service” la recibe y manda un código de éxito o de error, dependiendo de si la información esta correcta o no.
12. En caso de que el “web service” mande un código de error, el operador deberá volver intentar el envío de información.

Escenario Alternativo:

- 3.a El operador trata de montar un cilindro ya estando montado otro.
 1. El sistema genera un mensaje de error.
- 4.a El operador elige un cilindro o pareja equivocado.
 1. El operador selecciona “sacar secuencia”.
 2. El cilindro o pareja desaparece de la zona de cilindros en producción
 3. El sistema reestablece todas las condiciones iniciales.
- 6.a Luego de procesar, el operador puede elegir eliminar el proceso de montaje.
 1. El sistema restaura las todas las situaciones iniciales.
 2. Se elimina los cilindros o parejas de la lista de montajes procesados.
 3. Se visualizan los cilindros en la lista de cilindros listos para ser procesados en el módulo de montaje.
- 6.b Luego de procesar, el operador puede elegir actualizar el proceso de montaje.



1. El operador puede elegir otras cajas para ser montadas en los cilindros.
2. El operador guarda nuevamente.
3. El sistema actualiza la información.

Requisitos Especiales:

- Interfaz en QNX amigable y acorde con los estándares de visualización de la empresa.
- Tiempos de respuesta pequeños, con la finalidad de agilizar la búsqueda de información de manera eficiente, visualización de datos no mayor a 10 segundos.

Caso de uso: Procesar Desmontaje

Actor: Operador

Precondiciones: El operador se identifica y autentica.

Garantías de éxito: Se registra el proceso de desmontaje. Se actualiza la información en el registro del cilindro.

Escenario principal de éxito:

1. El sistema actualiza la lista de cilindros a desmontar.
2. El operador escoge de la lista que se presenta el cilindro de apoyo o la pareja de cilindros de Trabajo de que desea procesar.
3. El operador visualiza información básica del cilindro de apoyo o pareja de cilindros de trabajo y pasa a montar en la pantalla principal.
4. El sistema permite levantar una ventana donde se deberá especificar si algún defecto fue encontrado e introducir algunas observaciones.
5. El operador pasa el cilindro de apoyo o pareja de trabajo a estado procesado.
6. El operador escoge la opción de reportar el desmontaje.
7. El sistema hace una llamada al conector de reportar desmontaje.
8. Se arma un archivo XML con toda la información del desmontaje y de los cilindros.
9. Por medio del SOAP, la información es enviada al “web service”.
10. El “web service” la recibe y manda un código de éxito o de error, dependiendo



de si la información esta correcta o no.

11. En caso de que el “web service” mande un código de error, el operador deberá volver intentar el envío de información.

Escenario Alternativo:

- 3.a El operador trata de montar un cilindro ya estando montado otro.
 1. El sistema genera un mensaje de error.
- 4.a El operador elige un cilindro o pareja equivocado.
 1. El operador selecciona “sacar secuencia”.
 2. El cilindro o pareja desaparece de la zona de cilindros en producción
 3. El sistema reestablece todas las condiciones iniciales.
- 5.a Luego de procesar, el operador puede elegir eliminar el proceso de desmontaje.
 1. El sistema restaura las todas las situaciones iniciales.
 2. Se elimina los cilindros o parejas de la lista de desmontajes procesados.
 3. Se visualizan los cilindros en la lista de cilindros listos para ser procesados en el Módulo de desmontaje.
- 5.b Luego de procesar, el operador puede elegir actualizar el proceso de desmontaje.
 1. El operador puede actualizar los datos de desmontaje.
 2. El operador guarda nuevamente.
 3. El sistema actualiza la información.

Requisitos Especiales:

- Interfaz en QNX amigable y acorde con los estándares de visualización de la empresa.
- Tiempos de respuesta pequeños, con la finalidad de agilizar la búsqueda de información de manera eficiente, visualización de datos no mayor a 10 segundos.



III.3 Especificación Complementaria

Los casos de uso permiten la recolección de la mayoría de los requisitos del sistema, sin embargo hay algunos de ellos que no pueden ser recolectados por medio de los casos de uso. Estos requisitos son llamados Requisitos No Funcionales.

Entre los requisitos no funcionales se tienen:

- El sistema debe generar archivos donde se lleve un registro de cada una de las acciones realizadas por los usuarios, así como también, registro de los errores del sistema.
- El sistema debe ser fácil de usar.
- Autenticación de usuarios, para poder acceder al sistema, se debe introducir un usuario y una clave con permisos que dependen del tipo de usuario que este ingresando al sistema.
- En caso de ocurrir una falla eléctrica u otra causa de apagado de las máquinas, el sistema debe guardar todas las operaciones que se habían realizado hasta ese momento, es decir, que la pérdida de información ante cualquier evento imprevisto sea mínima, permitiéndole a los usuarios seguir realizando el proceso ya iniciado, sin necesidad de comenzar de nuevo.
- El sistema debe ser completamente configurable, de esta manera no existe ningún inconveniente a la hora de instalar nuevamente el sistema.
- El sistema debe estar montado en máquinas industriales y no PC's, los monitores ubicados en planta deben ser de 17 pulgadas, ya que el texto debe ser de fácil visualización con un poco de distancia.