Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería

Escuela de Ingeniería Civil en Informática

IMPLEMENTACION DE UN "SISTEMA DE RECEPCIÓN, EMPAQUE Y FRIGORÍFICO" UTILIZANDO ARQUITECTURA DE CAPAS ORIENTADAS AL SERVICIO

Tesis para optar al Título de:
Ingeniero Civil en Informática

Profesor Patrocinante:
Sr. Martin Solar Monsalves
Docente Instituto de Informática.

LUIS MAURICIO LÓPEZ SOTO

VALDIVIA – CHILE

2006

Valdivia, 21 de Junio de 2006

De : Martín Gonzalo Solar Monsalves

A : Director Escuela Ingeniería Civil en Informática

Ref. : Informe Calificación Trabajo de Titulación

Nombre Trabajo de Titulación:

"IMPLEMENTACIÓN DE UN SISTEMA DE RECEPCIÓN, EMPAQUE Y FRIGORÍFICO UTILIZANDO ARQUITECTURA DE CAPAS ORIENTADAS AL SERVICIO."

Nombre Alumno:

Luis Mauricio López Soto.

Evaluación:

Cumplimiento del objetivo propuesto	7.0
Satisfacción de alguna necesidad	7.0
Aplicación del método científico	6.5
Interpretación de los datos y obtención de conclusiones	7.0
Originalidad	6.5
Aplicación de criterios de análisis y diseño	6.6
Perspectivas del trabajo	7.0
Coherencia y rigurosidad lógica	7.0
Precisión del lenguaje en la exposición, composición, redacción e ilustración	6.5
Nota Final	**6.8**

Sin otro particular, atte.:

Martín Solar Monsalves

VALDIVIA, 29 de Junio de 2006.-

DE: Oscar Mapocho Salazar Cerna
A: DIRECCIÓN ESCUELA INGENIERÍA CIVIL EN INFORMÁTICA

MOTIVO: INFORME TRABAJO DE TITULACIÓN

Nombre Trabajo de Titulación: Implementación de un "Sistema de Recepción, Empaque y Frigorífico" Utilizando Arquitectura de capas Orientadas al Servicio

Nombre del Alumno: Mauricio López Soto

6,5 (seis coma cinco)

Fundamentos de la Nota: Considero que el trabajo cumple sus objetivo y el estudiante demuestra sus aptitudes de ingeniero, y en ese aspecto el proyecto abordado presenta una original utilización en ambientes industriales de servicios Web, logrando además incorporarlos a una plataforma Web.
Sí noté cierta ausencia de una autoevaluación del estudiante en su rol de ingeniero civil informático, algo en lo cual indicara si las herramientas entregadas por la universidad le fueron suficientes o notó ciertas falencias que se pueden mejorar en la educación de las futuras generaciones.

Valdivia, 29 Junio del 2006

DE : Miguelina Vega Rosales

Profesor Instituto Informática

A: Dirección Escuela Ingeniería Civil en Informática

Informo a usted que el Proyecto de Título "IMPLEMENTACION DE UN SISTEMA DE RECEPCIÓN, EMPAQUE Y FRIGORÍFICO, UTILIZANDO ARQUITECTURA DE CAPAS ORIENTADAS AL SERVICIO"., presentado por el señor Luis Mauricio López Soto, cumple con el objetivo general propuesto, que es la implementación de un sistema de recepción, empaque y frigorífico., bajo una plataforma de desarrollo distribuida en capas y utilizando las tecnologías de los Web Services.

La metodología de trabajo y el lenguaje utilizado es el adecuado, sin embargo, no queda claro como el modelo propuesto influye en los costos de desarrollo y futuros costos de integración. Por lo anteriormente expuesto, califico este proyecto de título con nota 6,5 (seis, cinco).

Atentamente

Miguelina Vega R.

... A Mis padres, Luis y Carmen, por su apoyo y cariño, a Gloria por acompañarme todos estos años e Ignacio, mi hijo ...

INDICE

1. Introducción. 12

1.1 Principales Características de los WebServices. 13

1.2 Utilización de los WebServices para el análisis y desarrollo de los sistemas a realizar 14

2. Objetivos generales y específicos. 17

2.1 Objetivos generales. 17

2.2 Objetivos Específicos 17

3. Análisis de la Empresa, manejo de clientes, funcionamiento e identificación de problemas. 18

3.1 Servicios que presta la empresa a sus clientes. 18

3.2 Clientes de la empresa 20

3.3 Descripción "Recepción de materia prima" 21

3.4 Descripción "Empacado de Productos" 22

3.5 Descripción "Almacenamiento en frigorífico y gestión de bodega" 25

3.6 Hardware Involucrado en el Sistema 28
3.6.1 Dispositivos de pesaje electrónicos 28

3.7 Identificación de problemas específicos 29
3.7.1 Comunicación de dispositivos de pesaje con el sistema. 29
3.7.2 Agrupamiento de productos en bodegas o frigoríficos 30
3.7.3 Servicio responsable de la comunicación de los dispositivos con el sistema. 31
3.7.4 Etiquetado de las cajas de empaque dependiendo del propietario de la materia prima. 32

4. Análisis de tecnologías a utilizar y aplicación de ellas. 33

4.1 Modelo de N-Capas 33

4.2 Evolución del modelo de N-Capas a WebServices 35

4.3 WebServices 36
4.3.1 Definición de los WebServices 37
4.3.2 Requisitos que debe cumplir un WebServices 37
4.3.3 Elementos que componen los WebService 39
4.3.3.1 XML (eXtended Markup Language) 39
4.3.3.2 SOAP 42
4.3.3.3 WSDL (WebService Description Language) 44
4.3.3.4 Disco 45
4.3.3.5 UDDI 45
4.3.3.6 SOAP, WSDL, UDDI y WebServices 46
4.3.3.7 Posibles riesgos de los WebServices 47
4.3.3.8 Calidad de los WebServices 47
4.3.3.9 Estandarización de los WebServices 49

5. Diseño e implementación de módulo de "Recepción de Materia Prima" 50

5.1 Modelo de datos vinculado con el módulo de Recepción de materia Prima 50

5.2 Diagrama Casos de Uso para módulo de "Recepción de Materia Prima" 51

5.3 Plantillas Casos de Uso para "Módulo de Recepción de Materia Prima" 52
5.3.1 Caso de Uso "Crear y gestionar Recepción de Lote de Materia Prima" 52
5.3.2 Caso de Uso "Administrar Información de Propietario" 52
5.3.3 Caso de Uso "Administrar materia prima recepcionada" 53
5.3.4 Caso de Uso "Administrar dependencias del Lote de Materia prima" 53
5.3.5 Caso de Uso "Administrar Control de Calidad" 54

5.4 Diagramas de Secuencias 54
5.4.1 Diagrama de Secuencia "Crear y gestionar Recepción de Lote de Materia Prima" 55
5.4.2 Diagrama de Secuencia "Administrar Información de Propietario" y "Administrar materia prima recepcionada" 56
5.4.3 Diagrama de Secuencia "Administrar dependencias del Lote de Materia prima" 56
5.4.4 Diagrama de Secuencia "Administrar Control de Calidad" 57

5.5 Diagramas de Colaboración 57
5.5.1 Diagrama de Colaboración "Crear y gestionar Recepción de Lotes de Materia Prima" 58
5.5.2 Diagrama de Colaboración "Administrar Información de Propietario" y "Administrar materia prima recepcionada" 58
5.5.3 Diagrama de Colaboración "Administrar dependencias del Lote de Materia prima" 59
5.5.4 Diagrama de Colaboración "Administrar Control de Calidad" 59

5.6 Descripción de los WebServices a implementar 59

5.7 Principales pantallas del Módulo de Recepción de Materia Prima 64
5.7.1 Pantalla "Lista Lotes de Materia Prima" 64
5.7.2 Pantalla "Agregar y Actualizar Lotes de Materia Prima" 65
5.7.3 Pantalla "Información de Productor de Lote de Materia Prima" 65
5.7.4 Pantalla "Control de Calidad para Lote de Materia Prima" 66
5.7.5 Pantalla "Dependencias para Lote de Materia Prima" 66

6. Diseño e implementación de módulo de "Empaque de Productos" 68

6.1 Modelo de datos vinculado con el módulo de Empaque de Productos 68

6.2 Diagrama Casos de Uso para módulo de "Empaque de Productos" 70

6.3 Plantillas Casos de Uso para "Módulo de empaque de Productos" 71
6.3.1 Caso de Uso "Administrar Ordenes de Empaque" 71
6.3.2 Caso de Uso "Administrar Detalles de Orden de Empaque" 71
6.3.3 Caso de Uso "Inicializar Líneas de Producción" 72
6.3.4 Caso de Uso "Ejecutar Órdenes de Empaque" 72
6.3.5 Caso de Uso "Recuperar Información de Dispositivos de Pesaje" 73

6.4 Diagramas de Secuencias 74
6.4.1 Diagrama de Secuencia "Administrar Órdenes y Detalles de Ordenes de Empaque" 74
6.4.2 Diagrama de Secuencia "Inicializar Líneas de Producción" 74
6.4.3 Diagrama de Secuencia "Ejecutar Órdenes de Empaque" 75
6.4.4 Diagrama de Secuencia "Recuperar Información de Dispositivos de Pesaje" 75

6.5 Diagramas de Colaboración 76
6.5.1 Diagrama de Colaboración "Administrar Órdenes y Detalles de Ordenes de Empaque" 76
6.5.2 Diagrama de Colaboración "Inicializar Líneas de Producción" 76
6.5.3 Diagrama de Colaboración "Ejecutar Órdenes de Empaque" 77
6.5.4 Diagrama de Colaboración "Recuperar Información de Dispositivos de Pesaje" 77

6.6 Descripción de los WebServices a implementar 77

6.7 Principales pantallas del Módulo de Empaque de Productos 81
6.7.1 Pantalla "Lista Ordenes de Empaque" 81
6.7.2 Pantalla "Agregar y Actualizar Ordenes de Empaque" 82
6.7.3 Pantalla "Agregar y Actualizar Detalles de Ordenes de Empaque" 83
6.7.4 Pantalla "Inicialización de Línea de Producción" 84

6.7.5 Pantalla “Ejecución Orden de Empaque” 85

7. Diseño e implementación de módulo de “Frigorífico y Despacho” 87

7.1 Modelo de datos vinculado con el módulo de Frigorífico y Despacho 87

7.2 Diagrama Casos de Uso para módulo de “Frigorífico y Despacho” 89

7.3 Plantillas Casos de Uso para “Módulo de Frigorífico y Despacho” 90
7.3.1 Caso de Uso “Ingresar Pallets a Bodega” 90
7.3.2 Caso de Uso “Administrar Pallets en Bodega” 90
7.3.3 Caso de Uso “Administrar Instrucciones de Salida” 90
7.3.4 Caso de Uso “Administrar Detalles de Instrucciones de Salida” 91
7.3.5 Caso de Uso “Ejecutar Instrucciones de Salida” 91

7.4 Diagramas de Secuencias 92
7.4.1 Diagrama de Secuencia “Ingresar Pallets a Bodega” 92
7.4.2 Diagrama de Secuencia “Administrar Pallets en Bodega” 93
7.4.3 Diagrama de Secuencia “Administrar Instrucciones y Detalles de Instrucciones de Salida” 93
7.4.4 Diagrama de Secuencia “Ejecutar Instrucciones de Salida” 94

7.5 Diagramas de Colaboración 94
7.5.1 Diagrama de Colaboración “Ingresar Pallets a Bodega” 94
7.5.2 Diagrama de Colaboración “Administrar Pallets en Bodega” 95
7.5.3 Diagrama de Colaboración “Administrar Instrucciones y Detalles de Instrucciones de Salida” 95
7.5.4 Diagrama de Colaboración “Ejecutar Instrucciones de Salida” 96

7.6 Descripción de los WebServices a implementar 96

7.7 Principales pantallas del Módulo de Frigorífico y Despacho 102
7.7.1 Pantalla “Lista de Ingresos a Frigorífico” 102
7.7.2 Pantalla “Armado de Nuevo Pallet” 102
7.7.3 Pantalla “Lista de Pallets en Mantención” 103
7.7.4 Pantalla “Mover Cajas de Pallet en Mantención” 104
7.7.5 Pantalla “Ingresar Cajas Pallet en Mantención” 104
7.7.6 Pantalla “Mover Pallet en Mantención” 105
7.7.7 Pantalla “Fusionar Pallet en Mantención” 106
7.7.8 Pantalla “Listar Instrucciones de Salida” 106
7.7.9 Pantalla “Ver y Agregar Instrucción de Salida” 107
7.7.10 Pantalla “Nuevo Detalle de Instrucción de Salida” 108
7.7.11 Pantalla “Ejecución de Instrucción de Salida” 108

8. Conclusiones 110

9. Bibliografía 113

9.1 Documentos 113

9.2 Direcciones en Internet 113

INDICE DE TABLAS

Tabla 1. Tabla comparativa de gramática entre HTML y XML 40

Tabla 2. Planilla Caso de Uso: Creación y gestión de Recepción de Lote MP 52

Tabla 3. Planilla Caso de Uso: Administrar Información de Propietario 53

Tabla 4. Planilla Caso de Uso: Administrar materia prima recepcionada 53

Tabla 5. Planilla Caso de Uso: Administrar dependencias del Lote de Materia prima 54

Tabla 6. Planilla Caso de Uso: Administrar Control de Calidad 54

Tabla 7. Planilla Caso de Uso: Administrar Ordenes de Empaque 71

Tabla 8. Planilla Caso de Uso: Administrar Detalles de Orden de Empaque 72

Tabla 9. Planilla Caso de Uso: Inicializar Líneas de Producción 72

Tabla 10. Planilla Caso de Uso: Ejecutar Ordenes de Empaque 73

Tabla 11. Planilla Caso de Uso: Recuperar Información de Dispositivos de Pesaje 73

Tabla 12. Planilla Caso de Uso: Ingresar Pallets a Bodega 90

Tabla 13. Planilla Caso de Uso: Administrar Pallets en Bodega 90

Tabla 14. Planilla Caso de Uso: Administrar Instrucciones de Salida 91

Tabla 15. Planilla Caso de Uso: Administrar Detalles de Instrucciones de Salida 91

Tabla 16. Planilla Caso de Uso: Ejecutar Instrucciones de Salida 92

INDICE DE FIGURAS

Figura 1, Planteamiento de la solución. 16

Figura 2, Servicios que presta la empresa 19

Figura 3, Diagrama de la distribución de procesos que ofrece la empresa. 20

Figura 4, Balanza Electrónica 29

Figura 5, Dispositivo Conversor RS232 a TCP/IP 30

Figura 6, Modelo de N-Capas 34

Figura 7, Solicitud Respuesta SOAP 43

Figura 8, Vinculación entre UDDI, Cliente, SOAP y WebService 47

Figura 9, Modelo físico de Datos módulo de Recepción de Materia Prima 50

Figura 10, Diagrama de casos de uso de Módulo de "Recepción de Materia Prima" 51

Figura 11, Diagrama de secuencia "Crear y gestionar Recepción de Lote de Materia Prima" 55

Figura 12, Diagrama de secuencia "Administrar Información de Propietario" 56

Figura 13, Diagrama de secuencia "Administrar dependencias del Lote de Materia prima" 56

Figura 14, Diagrama de secuencia "Administrar Control de Calidad" 57

Figura 15, Diagrama de colaboración "Crear y gestionar Recepción de Lote de Materia Prima" 58

Figura 16, Diagrama de colaboración "Administrar Información de Propietario" 58

Figura 17, Diagrama de colaboración "Administrar dependencias del Lote de Materia prima" 59

Figura 18, Diagrama de colaboración "Administrar Control de Calidad" 59

Figura 19, Pantalla "Lista Lotes de Materia prima" 64

Figura 20, Pantalla "Agregar y Actualizar Lotes de Materia Prima" 65

Figura 21, Pantalla "Información de Productor de Lote de Materia Prima" 66

Figura 22, Pantalla "Control de Calidad para Lote de Materia Prima" 66

Figura 23, Pantalla "Dependencias para Lote de Materia Prima" 67

Figura 24, Modelo físico de datos módulo Empaque de Productos 69

Figura 25, Diagrama de casos de uso de Módulo de "Empaque de Productos" 70

Figura 26, Diagrama de Secuencia "Administrar Órdenes y Detalles de Ordenes de Empaque" 74

Figura 27, Diagrama de Secuencia "Inicializar Líneas de Producción" 75

Figura 28, Diagrama de Secuencia "Ejecutar Órdenes de Empaque" 75

Figura 29, Diagrama de Secuencia "Recuperar Información de Dispositivos de Pesaje" 75

Figura 30, Diagrama de Colaboración "Administrar Órdenes y Detalles de Ordenes de Empaque" 76

Figura 31, Diagrama de Colaboración "Inicializar Líneas de Producción" 76

Figura 32, Diagrama de Colaboración "Ejecutar Órdenes de Empaque" 77

Figura 33, Diagrama de Colaboración “Recuperar Información de Dispositivos de Pesaje” 77
Figura 34, Pantalla “Listar Ordenes de Empaque” 82
Figura 35, Pantalla “Agregar y Actualizar Ordenes de Empaque” 83
Figura 36, Pantalla “Agregar y Actualizar Detalles de Ordenes de Empaque” 84
Figura 37, Pantalla “Inicialización de Línea de Producción” 85
Figura 38, Pantalla “Ejecución Orden de Empaque” 86
Figura 39, Pantalla “Modelo Físico Módulo Frigorífico y Despacho” 88
Figura 40, Diagrama Casos de Uso: Módulo de Frigorífico y Despacho. 89
Figura 41, Diagrama de Secuencia “Ingresar Pallets a Bodega” 92
Figura 42, Diagrama de Secuencia “Administrar Pallets en Bodega” 93
Figura 43, Diagrama de Secuencia “Administrar Instrucciones y Detalles de Instrucciones de Salida” 93
Figura 44, Diagrama de Secuencia “Ejecutar Instrucciones de Salida” 94
Figura 45, Diagrama de Colaboración “Ingresar Pallets a Bodega” 94
Figura 46, Diagrama de Colaboración “Administrar Pallets en Bodega” 95
Figura 47, Diagrama de Colaboración “Administrar Instrucciones y Detalles de Instrucciones de Salida” 95
Figura 48, Diagrama de Colaboración “Ejecutar Instrucciones de Salida” 96
Figura 49, Pantalla “Lista de ingreso a frigorífico” 102
Figura 50, Pantalla “Armado de Nuevo Pallet” 103
Figura 51, Pantalla “Lista de Pallets en Mantención” 103
Figura 52, Pantalla “Mover Cajas de Pallet en Mantención” 104
Figura 53, Pantalla “Ingresar Cajas Pallet en Mantención” 105
Figura 54, Pantalla “Mover Pallet en Mantención” 105
Figura 55, Pantalla “Fusionar Pallet en Mantención” 106
Figura 56, Pantalla “Listar Instrucciones de Salida” 107
Figura 57, Pantalla “Ver y Agregar Instrucción de Salida” 107
Figura 58, Pantalla “Nuevo Detalle de Instrucción de Salida” 108
Figura 59, Pantalla “Ejecución de Instrucción de Salida” 109

RESUMEN

Uno de los costos de inversión más grandes en que deben incurrir las organizaciones corresponde a los gastos para la innovación e integración de sistemas. Dado estos factores, es de vital importancia instaurar un modelo de construcción de software que establezca las bases para agregar nuevos servicios a un costo menor que los actualmente conocidos.

En el intento de disminuir los costos asociados a las TI, las empresas han empezado a optar por plataformas homogéneas para el desarrollo de sus sistemas y como resultado de su búsqueda la plataforma Web se entabla como principal candidata para el desarrollo de sistemas, esto debido a las grandes ventajas que conlleva su utilización, entre las cuales se destaca: masificación a nivel mundial, facilidad de acceso, costos de implementación bajos y un escenario uniforme, independiente de la tecnología implantada en la empresa.

Las nuevas tendencias de arquitectura de software para la plataforma Web se basan en un nuevo modelo de construcción que incorporan conceptos y tecnologías denominados WebServices (Servicios Web), que pueden contribuir a lograr las metas establecidas en las organizaciones actuales, de acuerdo a los criterios señalados anteriormente. Éstos presentan características innovadoras debido a que incorporan ideas que durante años han trabajado en líneas distintas para lograr la integración de las aplicaciones distribuidas.

El presente documento tiene como objetivo presentar la implementación de un sistema de recepción, empaque y frigorífico, bajo una plataforma de desarrollo distribuida en capas y utilizando las tecnologías de los WebServices, se presentará un análisis de la forma en que opera la empresa, en lo referente a sus servicios que

brinda a sus clientes y se identificarán problemas específicos concernientes a la operación de sus diferentes áreas.

Por último, se describirán los beneficios obtenidos por la utilización de la plataforma de desarrollo en el sistema propuesto, el aporte de escalabilidad y modularidad.

ABSTRACT

One of the costs of investment the institutions should incur in bigger corresponds to the expenses for the invention and integration of systems. Die these factors, it is by vital importance establishing a model of the construction of software that lays the foundations to add new services to a minor cost than at present acquaintances.

In the attempt of decreasing the costs associated to them T.I., the companies have started to choose homogeneous platforms for the development of his systems and as a result of their quest the platform establishes Web itself like principal developmental candidate of systems, this due to the big advantages that you bear your utilization, between the ones that you stand out between: Worldwide masification, facility of access, low costs of implementation and an uniform, independent scene of the technology established at the company.

The new architectural tendencies of software for the platform base Web on a new model of the construction that they incorporate concepts and named technologies WebServices (Services Web), that the goals established in the present-day organizations, according to the criteria indicated previously can contribute to achieve themselves. These present innovative characteristics because they incorporate ideas than they have worked at different lines to achieve the integration of the distributed applications during years.

I document the present you aim at presenting the implementation of a system of reception, packing and cold storage room, under a platform of development distributed stratified and using the WebServices technologies, an analysis will show up of the form in operating the company, referring to his

services that you make a toast to his customers and that will identify specific problems cared about to the operation of his different areas.

Finally, they will describe the benefits obtained by the utilization of the platform of development in the proposed system, the contribution of scalability and modularity.

Capítulo 1

1. Introducción.

Debido a la alta competitividad y las exigencias del nuevo mercado, la empresa actual, que se caracteriza por ofrecer algún tipo de servicio a sus clientes, se ha visto forzada a buscar formas para destacar de entre sus iguales, buscando constantemente nuevos medios de llegada al cliente que ofrezcan servicios innovadores, de mejor calidad y alta disponibilidad para satisfacer las nuevas necesidades del cliente y asegurar una dinámica de trabajo en conjunto para el futuro.

Por otra parte, existe la necesidad de comunicar e integrar sistemas de tecnologías de información con agentes intermedios o proveedores, quienes poseen sistemas con aplicaciones propias especializadas en distintas materias y generalmente funcionando en plataformas heterogéneas, con el objetivo de disminuir tiempos de respuesta y asegurar así la calidad de la información.

Dado lo anterior, una solución viable es aplicar innovaciones tecnológicas y de servicios para cubrir las necesidades tanto internas como externas de la empresa, asegurando así la vigencia en el mercado.

Más, la constante incorporación de tecnologías de información se traduce directamente en inversión de tiempo y recursos, el cual, mediante estrategias organizacionales, se busca reducir al mínimo para mantener precios competitivos y márgenes de ganancia aceptables. Por esta razón, muchas empresas prefieren esperar y no invertir en tecnología hasta que sea estrictamente necesario, demora

que puede conducir a una notable pérdida de mercado (estancamiento tecnológico). Entonces, el objetivo de las empresas actuales es encontrar una formula que permita mantener un equilibrio entre las necesidades planteadas, aumentando la implantación de innovaciones tecnológicas proporcionando una integración de sus sistemas dentro del mercado competitivo y a la vez disminuyendo los respectivos costos de tiempo y recursos asociados.

1.1 Principales Características de los WebServices.

Los WebServices abarcan una visión de integración total para redes computacionales que incluyan computadores, servidores, PDA, programas y equipos de redes, todos trabajando juntos e integrados. Este conjunto de redes de dispositivos podrán trabajar en forma distribuida con un mejor acoplamiento entre dispositivos para cada tarea y otorgue información en tiempos acordes según las solicitudes del usuario. [Gov01]

Un gran conjunto de empresas, actualmente, están lejos de otorgar homogeneidad en sus formas de trabajo para sus distintas soluciones informáticas implementadas, ya sea por distintas políticas adoptadas en sus departamentos de T.I.[1] al momento de escoger plataformas de desarrollo o por trabajar con plataformas que al paso del tiempo se vuelven obsoletas, imposibilitando la integración de sus distintas áreas productivas y por lo tanto, se ven restringidas al momento de realizar cruces de información y dificultando una real consolidación de la información.

El principal objetivo que se logra en la utilización de los WebServices es la integración e interoperabilidad entre sistemas. Posibilitando así, compartir

[1] Tecnologías de Información

información y software, tanto internamente en la empresa o bien con sus clientes. Esta ventaja permitirá a la empresa escalar en su plataforma de negocios e imprimir un ahorro en desarrollo y mantenimiento de software. Al realizar la integración de aplicaciones se posibilitará obtener resultados en tiempo real y así agregar un plus al momento de realizar toma de decisiones.

En resumen, las ventajas que los webservices podrán aportar son las siguientes [Tab02]:

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Aportan una ventaja económica, ya que al utilizar estándares abiertos, se protege la inversión realizada en T.I.
- Baja complejidad en su utilización, debido a la gran cantidad de herramientas disponibles para una directa interacción.

1.2 Utilización de los WebServices para el análisis y desarrollo de los sistemas a realizar

La utilización de los webservices cumple un rol fundamental en la solución propuesta para el análisis y desarrollo del "Sistema de Recepción, Empaque y Frigorífico", esto debido a que la solución para los diferentes sistemas comprende la utilización de una arquitectura de tres capas [URL 1]:

- Capa de Datos: La cual contendrá la base de datos física que soporta el sistema y procedimientos almacenados que apoyan las distintas funcionalidades del mismo.
- Capa de Negocios: La cual contendrá la lógica de negocio del sistema, reglas de validación y el desglose de las funcionalidades que permitirán una óptima utilización de la solución.
- Capa de Presentación: Los servicios de esta capa se encargan de la presentación de los datos, reciben los eventos del usuario y controlan la interfaz cliente. Estos servicios serán implementados con HTML, ASP.NET y Flash MX.

Entendiéndose que la capa de negocios es la interlocutora para los distintos flujos de información entre el repositorio de datos y la aplicación frontal, capa con la cual interactuaran los usuarios finales, la utilización de los WebServices se torna fundamental en el sentido que permitirá encapsular en una sola capa todas las reglas de negocios que conforma el sistema, posibilitando así una independencia de la plataforma final utilizada por el usuario para la interacción con el sistema

A continuación se presenta el diagrama de flujo que describe el planteamiento de la solución.

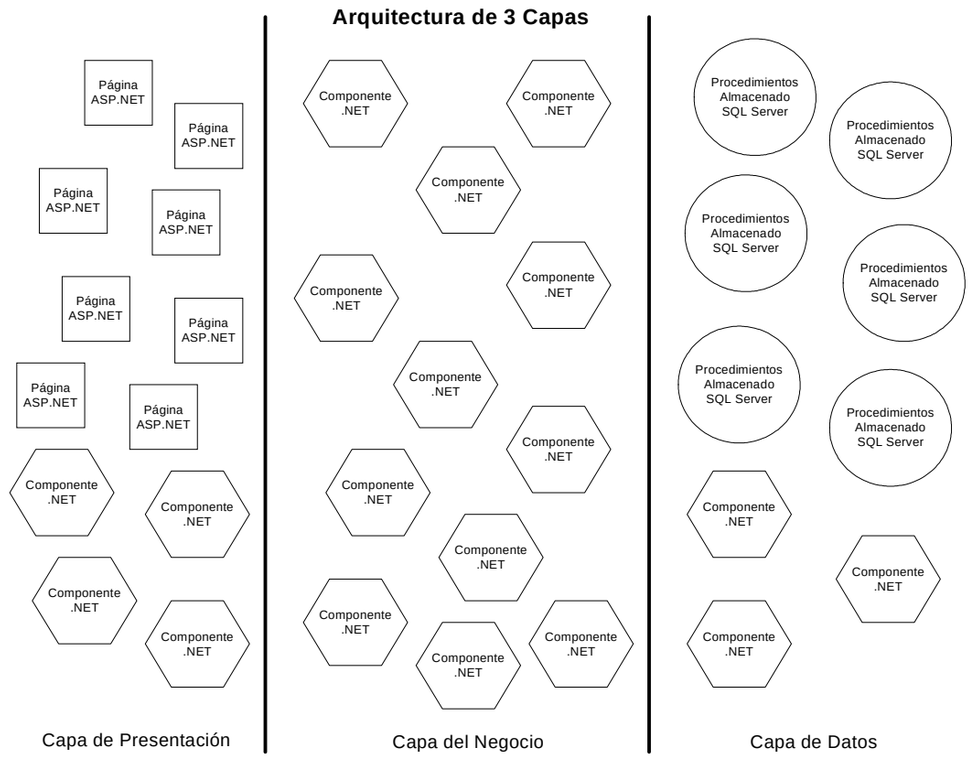


Figura 1, Planteamiento de la solución.

Capítulo 2

2. Objetivos generales y específicos.

2.1 Objetivos Generales.

Proponer y construir un modelo de capas orientadas al servicio que permita desarrollar en forma modular, escalable y a un bajo costo, un sistema de Recepción, Empaque y Frigorífico para satisfacer actuales y futuras necesidades de una empresa maquiladora de salmones y sus negocios involucrados.

2.2 Objetivos Específicos

- Comprender e implementar un modelo de múltiples capas orientadas al servicio (capa de datos, capa de negocios y capa de presentación) con la utilización de WebServices para la implementación de un sistema de recepción de materia prima, empacado, almacenamiento y administración en frigorífico.
- Validar el modelo planteado mediante la construcción de al menos dos aplicaciones front-end[2], lo cual permitirá medir la versatilidad de la solución para la interoperabilidad frente a diversos clientes.
- Determinar cómo el modelo propuesto ha influido en los costos de desarrollo y en futuros costos de integración del cliente del sistema.

[2] Aplicaciones que interactúan directamente con el usuario final.

Capítulo 3

3. Análisis de la Empresa, manejo de clientes, funcionamiento e identificación de problemas.

En el presente capítulo se describirá el funcionamiento de la empresa, servicios que presta hacia sus clientes, forma de que sus servicios operan, las distintas áreas que comprende y la función de cada una de ellas dentro del proceso de negocios de ésta.

3.1 Servicios que presta la empresa a sus clientes.

La empresa se caracteriza por ofrecer servicios de maquila (absorción de diferentes áreas de producción para la generación de productos terminados o semi-terminados) a los diferentes clientes que requieran sus servicios, comenzando con la recepción de materia prima a procesar, empaquetado, almacenamiento en frigorífico y su posterior despacho.

- Recepción de materia prima: recopilación y administración de los diferentes orígenes de la materia prima recepcionada, destacando el tipo de proceso a seguir para la materia prima, propietario de la misma, guías de despacho involucradas en la recepción y controles de dependencias de la materia prima en el caso que se encuentre vinculada a otra recepción de materia prima.
- Empacado de productos: Gestión y creación de las diferentes cajas de productos, enfocado a los requerimientos y necesidades de cada cliente, atendiendo a parametrizaciones de tolerancia, calibre, etiqueta a utilizar para el empacado y administración de los insumos utilizados en el empacamiento de él o los diferentes productos.

- Almacenamiento y gestión de bodegas: Gestión y creación de pallets, unidad básica que se compone por el agrupamiento de un conjunto de cajas de empaque, para realizar el ingreso de las mismas en las distintas bodegas que se administran en la empresa y posteriormente la definición de las salidas de bodega, según los criterios que establezca el cliente.
- Gestión de información: Conjunto de reportes enfocados para cada cliente, con los cuales controlan el estado proceso de su materia prima y/o productos maquilados por la empresa.

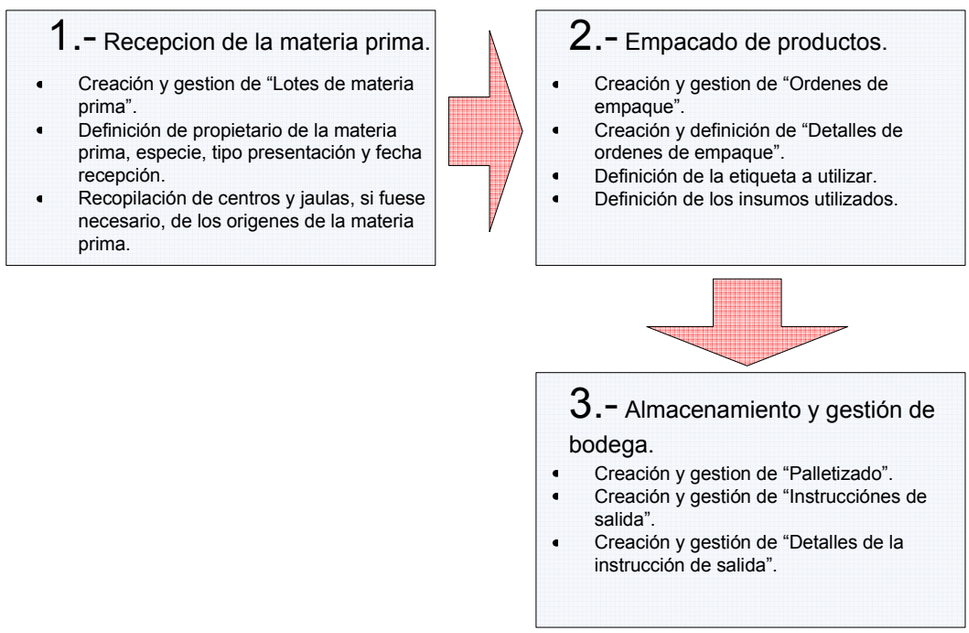


Figura 2, Servicios que presta la empresa

Para comprender mejor la dimensión de la solución es necesario visualizar los módulos de procesos que la empresa maquiladora ofrece a sus clientes.

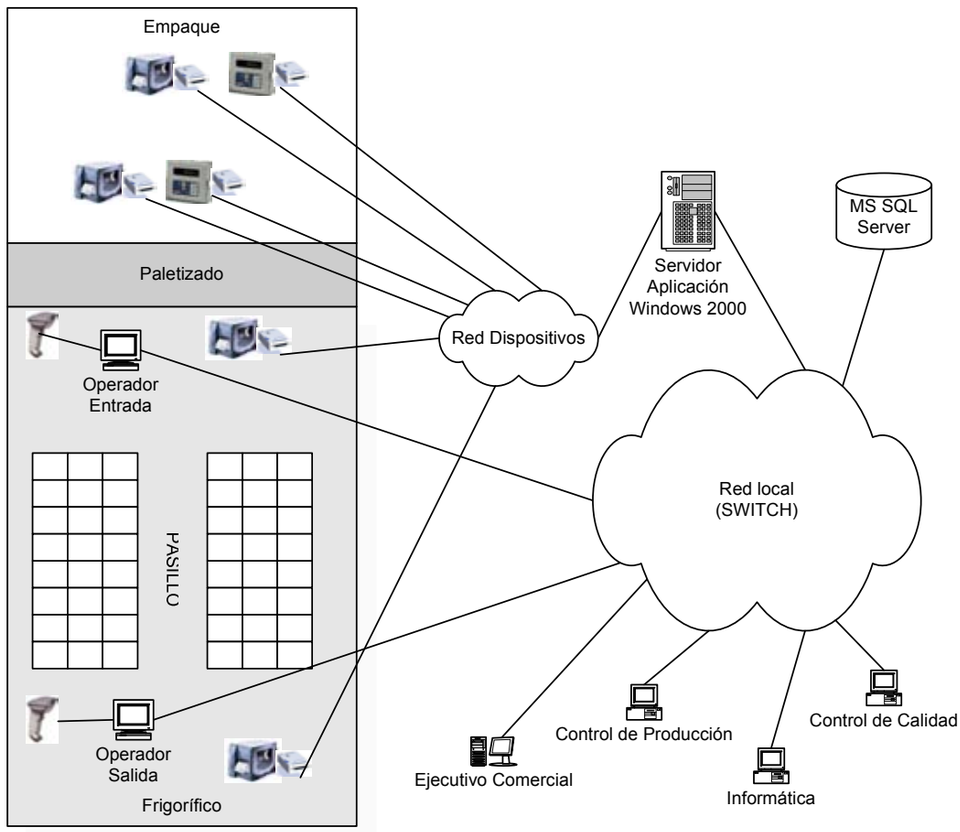


Figura 3, Diagrama de la distribución de procesos que ofrece la empresa.

3.2 Clientes de la empresa

Los clientes que posee la empresa maquiladora son empresas que generalmente obtienen excedentes de producción de materia prima (por lo tanto se encuentran imposibilitadas de procesarla) o directamente externalizan un conjunto de procesos que se aplican a sus materia prima.

Estos clientes son identificados a través de un código único, y es directamente vinculado con la recepción de la materia prima, imposibilitando el cruce de información referente a varios clientes, lo cual garantiza la confiabilidad de la información.

3.3 Descripción “Recepción de materia prima”

La recepción de materia prima es el comienzo de cualquier ciclo productivo que se aplique a una materia prima que un cliente desee maquilar. Es así que cualquier labor que se realice dentro de la planta tiene que tener como referente obligatorio un identificador de lote de materia prima.

Toda recepción que se realice debe cumplir un conjunto de características básicas que la individualiza del resto de las recepciones, entre las que se encuentran:

- Identificador de la recepción del lote de materia prima: Número único que identifica la recepción de la materia prima, el cual se transforma en el identificador del lote que se recepciona.
- Fecha de recepción: Fecha específica en la cual se ingresa la materia prima al sistema.
- Tipo de proceso: Corresponde al proceso industrial que le es aplicado a una materia prima para obtener un producto específico, ejemplo: proceso normal, cambio de envase,…, etc.
- Propietario: Identifica a la empresa a la cual se le esta prestando el servicio para procesar el lote de materia prima.
- Especie: Identifica a la familia o grupo que pertenece la materia prima a procesar, ejemplo: Salmón, Moluscos,…, etc.
- Presentación de la materia prima: Identifica el estado en que se presenta la materia prima la momento de ser recepcionada, ejemplo: entera, porción, pulpa,…, etc.

Para cumplir con normas de trazabilidad de la información, es importante definir el o los orígenes de la materia prima recepcionada, por lo tanto es un requisito fundamental describir los proveedores de la materia prima a procesar y

señalar la procedencia de esta. Para lograr ese objetivo se especifican las siguientes características:

- Número de la guía de despacho: Documento que envía el proveedor y con el cual se realiza el ingreso de la materia prima.
- Número de lote interno del proveedor: Identifica el lote proveniente de la materia prima.
- Proveedor: Especifica la empresa que traslada la materia prima.
- Centro: Nombre del lugar de origen de la materia prima.
- Jaula: Si la materia prima proviene de una jaula específica, se declara el número de jaula correspondiente.
- Peso y cantidad declarada por el proveedor: Se declara la cantidad de unidades y peso involucrado en la recepción según lo declarado por el proveedor.

3.4 Descripción "Empacado de Productos"

Una vez realizado la recepción de la materia prima, esta es empacada de acuerdo a los requerimientos que el cliente solicite, para poder mantener una estabilidad de la información generada en esta etapa de producción, es necesaria una clasificación de las ordenes de empaque que se generen en el sistema, para poder cumplir con los objetivos de gestión, las ordenes de empaque de productos son generadas especificando puntualmente que tipo de producto se esta trabajando, individualizando el origen del empacado y cumpliendo las parametrizaciones mínimas requeridas por cada cliente.

La realización de las "ordenes de empaque" se definen de acuerdo a la individualización de un conjunto de características únicas que las identifica, entre las cuales se encuentran:

- Identificador de Orden de Empaque: Numeración única que identifica a la orden de empaque específica.

- Propietario: Identifica a la empresa a la cual se le esta prestando el servicio para procesar el lote de materia prima.
- Código de Producto a procesar: Se identifica el producto de origen, el cual, de acuerdo a los procedimientos que se aplican, producen una serie de subproductos que se identifican como productos extendidos.
- Cliente: Cada orden de empaque que se define en la solución, pertenece a un cliente exclusivo.
- Unidad de Medida: Cada cliente podrá solicitar que el empacado de sus productos se encasillen de acuerdo a diferentes unidades de medida, por ejemplo: libras, kilogramos, onzas, etc.
- Etiqueta: La etiqueta a utilizar en la orden de empaque es suministrada por un mantenedor de etiquetas.
- Especie: Identifica a la familia o grupo que pertenece la orden de empaque.

A la definición de las “órdenes de empaque” se suma la generación de “detalles de órdenes de empaque”, que cumple con el objetivo de procesar un conjunto de productos resultantes del producto seleccionado en la “orden de empaque”, esto debido a que el módulo de empaque debe ser lo suficientemente flexible para permitir crear y gestionar diferentes productos vinculados con el detallado en la “orden de empaque”. Estos productos resultantes de esta gestión se les denominan “productos extendidos”, ya que, aunque son derivados de un producto padre, pueden tener diferentes tratamientos de acuerdo a una serie de parámetros, los cuales son:

- Calidad: La calidad a producir en una orden de empaque puede variar dentro de su ejecución, esto debido a que todo el producto ingresado no necesariamente se presenta en una forma homogénea.
- Calibre Inferior y Superior: Las piezas de productos que son empacadas generalmente deben cumplir con rangos de calibres preestablecidos y cada

rango de calibres debe ser manejado como un nuevo producto resultante o "producto extendido" del producto original.

- Unidad de Medida: Los "detalles de ordenes de empaque" pueden producir cajas de empaque que se encuentren en diferentes unidades de medida, por lo tanto, por ejemplo, es posible generar en un principio cajas de empaque con unidad de medida en kilogramos y finalizar con detalles de ordenes de empaque en libras.
- Color Inferior y Superior: Parámetro de medida por el cual el producto extendido es empacado de acuerdo a un rango de colores.
- Envase primario y secundario: Es necesario gestionar los envases en donde el producto resultante de la orden de empaque es almacenado.

En la definición de cada detalle de orden de empaque es posible establecer un conjunto de taras preestablecidas para las cajas que resulten del detalle de orden de empaque, la administración de las taras de empaque a través del sistema tiene como misión evitar la interacción de los operarios de empaque con las balanzas electrónicas, esto debido a intentar evitar posibles complicaciones al momento de ejecutar una orden de empaque.

Un punto importante en la gestión de las ordenes de empaque es la administración del personal involucrado en la ejecución de la misma, para cumplir con responsabilidades durante la ejecución de una orden de empaque se definen inicializaciones de líneas de producción, en cada línea de producción se ven involucrados una estación de trabajo con su respectiva balanza electrónica y etiquetadora, las inicializaciones administran al personal que realiza el empaque además del turno de trabajo y fecha/hora de la ejecución de la orden de empaque. Las inicializaciones de las órdenes de producción son un paso obligatorio antes de realizar cualquier tipo de operación de empaque.

3.5 Descripción “Almacenamiento en frigorífico y gestión de bodega”

Una vez realizado el empaque de productos, es necesario un almacenamiento de las cajas resultantes de las ordenes de empaque en frigorífico y/o bodegas, para realizar esta labor las cajas de empaque son agrupadas en cubos que se denominan “pallets de cajas de empaque”, estos pallets son formados previamente al ingreso de las cajas a bodega y cumplen la función de facilitar la administración del contenido de las bodegas, ya que, para poder realizar el armado de un pallet las cajas que son agrupadas deben cumplir con un conjunto de condiciones, las cuales son:

- Propietario: El propietario de todas las cajas que pertenezcan al pallet debe ser el mismo, la definición del propietario dominante del pallet se determina mediante la obtención del propietario de la primera caja que es ingresada al pallet.
- Especie: La especie de todas las cajas que pertenezcan al pallet debe ser la misma, la definición de la especie dominante del pallet se determina mediante la obtención del propietario de la primera caja de empaque.

Una vez realizada la creación del pallet de cajas de empaque, hay que definir su ubicación o posicionamiento dentro de una bodega, de acuerdo a la bodega seleccionada el sistema propondrá una ubicación tentativa para el almacenamiento, el conjunto de reglas que rige la selección de la ubicación son descritas en el apartado de “Identificación de problemas específicos”.

Al igual que en las ejecuciones de las ordenes de empaque, previo al momento de realizar la creación del pallet y su ingreso a bodega, se debe realizar una inicialización de la línea de producción que ejecuta el ingreso a bodega, ya que los resguardos en los procesos críticos de producción deben permanecer

durante todos los procesos que se apliquen a los productos de un propietario específico.

Es importante aclarar que los pallets almacenados en bodegas pueden sufrir alteraciones durante el transcurso de su almacenamiento, para cumplir con normas de gestión, estas alteraciones de los palles son administradas a través de un "mantenedor de pallets", las diferentes operaciones que agrupa el mantenedor de pallet se detallan a continuación:

- Mover cajas entre pallets: Para desplazar cajas entre pallets es necesario que ambos pallets se encuentren en estado de "mantención" y el movimiento es realizado seleccionando cajas de empaque desde un pallet de origen y trasladarlas hacia un pallet de destino.
- Ingresar nuevas cajas al pallet: Es posible agregar cajas de empaque que no han sido ingresadas a ningún pallet, para realizar estos movimientos es necesario realizar una inicialización de línea de producción que se encarga de realizar la mantención.
- Mover pallet: Es posible modificar la ubicación y posición del pallet dentro de cualquier bodega que se encuentre definida en el sistema.
- Fusionar pallets: Es posible la realización de fusiones de pallets que se encuentren en estado de "mantención", para realizar esta acción es necesario definir al pallet que adsorberá las cajas de empaque del pallet eliminado.

Dado el termino de permanencia de los pallets en bodega, o a petición del cliente para el cual se destinan el producto maquilado, se generan "instrucciones de salida" para realizar el despacho de las cajas de empaque que se encuentran agrupadas en pallets, la realización de una instrucción de salida debe cumplir con una serie de características, las cuales se detallan a continuación:

- Identificador de la instrucción de salida: Numeración única que genera el sistema para gestionar una instrucción de salida.
- Propietario: Identifica a la empresa a la cual se le esta prestando el servicio para procesar el lote de materia prima.
- Tipo de salida: Identifica el motivo que provoca la generación de la instrucción de salida.
- Especie: Identifica a la familia o grupo que pertenece la instrucción de salida.
- Cliente: Identifica al cliente destino al cual se le realizará la entrega de los productos maquilados.
- País: Define el país destino de los productos.
- Fecha Despacho: Fecha en la cual se realizará el despacho de los productos.
- Empresa de Transporte: Empresa que se encarga de realizar el transporte de los productos.
- Guía de despacho: Número de la guía de despacho con la cual se traslada los productos.

De forma similar que las órdenes de empaque, la ejecución de las instrucciones de salida deben realizarse a través de detalles de instrucciones de salida, los cuales definen una serie de parámetro que regirá el despacho, los cuales son:

- Definición del "producto extendido a despachar": Es obligatorio la definición del producto extendido por detalle de instrucción de salida que se desea despachar.
- Cantidad de cajas objetivos: Número límite de cajas a despachar por detalle de instrucción de salida.

- Peso Objetivo: Cantidad límite de peso a despachar por detalle de instrucción de salida.
- Rango de piezas o unidades por caja: Condición de filtra las cajas despachadas por la cantidad de piezas que contienen.
- Rango de fecha de congelamiento: Rango de fechas durante la cual se realizo el ingreso a frigorífico.
- Lote de Materia Prima: Identifica el Lote de Materia del cual se realice el despacho de productos.

Al igual que en el ingreso a frigorífico y en la ejecución de las ordenes de empaque, la ejecución de las instrucciones de salida son resguardadas a través de inicializaciones de líneas de producción. Estableciendo así que el tratamiento de las áreas de proceso y manipulación de productos deben existir responsables que amparen los procesos aplicados.

3.6 Hardware Involucrado en el Sistema

Son diferentes dispositivos de hardware que interactúan en forma continua con el sistema, al existir este tipo de relaciones es fundamental abarcar los diferentes dispositivos y a su vez incluirlos dentro de la solución propuesta, esto debido a obtener una administración globalizada de los elementos de software y hardware en una misma solución.

3.6.1 Dispositivos de pesaje electrónicos

Los dispositivos de pesaje electrónico o balanzas electrónicas, cumplen un rol fundamental para el funcionamiento del sistema, debido a que interactúan directamente en áreas criticas en el control de empaquetado y ofreciendo resguardo de los diferentes productos resultantes del mismo.

Estos dispositivos se caracterizan, generalmente, por ofrecer comunicaciones de tipo serial o protocolo de comunicaciones RS232, la cual contiene un conjunto de limitaciones al momento de realizar sus transmisiones.[URL 2]

Figura 4, Balanza Electrónica

3.7 Identificación de problemas específicos

Para la realización de la solución se tienen que resolver un conjunto de problemas, tanto para el buen funcionamiento de la solución o para otorgar funcionalidades específicas que acomodan de mejor forma la utilización del sistema, las dificultades a resolver son las siguientes:

3.7.1 Comunicación de dispositivos de pesaje con el sistema.

Los dispositivos de pesaje trabajan, generalmente, utilizando protocolos de comunicaciones que no resultan ser los más adecuados para poder lograr una interacción fluida con ellos, siendo que trabajan en base de comunicaciones seriales, RS232, resulta necesario trasladar el tipo de comunicaciones con que trabajan estos dispositivos al protocolo de comunicaciones TCP/IP, esta resolución se debe a las siguientes razones:

- Al realizarse las transmisiones de información por protocolo TCP/IP ofrece mejores garantías en el traslado de la información, esto debido que el mismo protocolo dispone de chequeos de errores.

- Las distancias de transmisión para comunicaciones en RS232 representan una limitante ya que solo pueden alcanzar, en el mejor caso, una distancia de 50 metros.[URL 3]

Para poder lograr la conversión en la transmisión de información se utilizan dispositivos conversores de comunicación, estos dispositivos conversores son adjuntados directamente a las balanzas electrónicas y así ofrecer una conversión transparente para el usuario final, los resultados del cambio del protocolo de comunicaciones de los dispositivos de pesaje ofrecen un conjunto de garantías, las cuales son:[URL 4]

- Al convertir el tipo de comunicación con que vienen los dispositivos de pesaje, estos dispositivos se transforman inmediatamente en puntos de red visibles a través de la intranet o extranet de la empresa.
- Se posibilita una mejor gestión de estos dispositivos en el sistema, ya que cada dispositivo de pesaje es individualizado como punto de red y su administración radica directamente en la configuración del sistema.

Figura 5, Dispositivo Conversor RS232 a TCP/IP

3.7.2 Agrupamiento de productos en bodegas o frigoríficos

Para lograr una mejor gestión de las bodegas y/o frigoríficos que administra la empresa, es necesario crear una lógica de agrupamiento de los diferentes productos resultantes del módulo de empaque, para eso se especificará

un conjunto de reglas jerárquicas, tras las cuales el sistema propondrá una ubicación tentativa para el usuario, el posicionamiento planteado ofrecerá la garantía de agrupar en islas de productos que se concentran bajo el siguiente esquema de reglas:

1. Por “Espacio disponible”: Al realizarse la cubicación de las bodegas, es importante tener un control de los espacios ocupados dentro de la misma.
2. Por “Propietario”: Para realizar el agrupamiento de productos en bodega es necesaria la concentración de productos que pertenezcan a un mismo Propietario, este Propietario es declarado al momento de recepcionar una materia prima.
3. Por “Especie”: Todos los productos resultantes del módulo de empaque pertenecen a una especie específica, por lo tanto es relevante agruparlas en la distribución de la bodega y/o frigorífico.
4. Por “Proceso”: Todos los productos resultantes del módulo de empaque son tratados mediante un proceso específico, por lo tanto es relevante agruparlas en la distribución de la bodega y/o frigorífico.
5. Por “Calidad”: Todos los productos resultantes del módulo de empaque contienen una calidad específica, por lo tanto es relevante agruparlas en la distribución de la bodega y/o frigorífico.

3.7.3 Servicio responsable de la comunicación de los dispositivos con el sistema.

Al existir comunicación de los dispositivos de pesaje con el sistema, es necesaria la creación de un servicio que se encargue de recepcionar esa información y administrarla dentro del sistema, para ello se crea un servicio del sistema operativo, el cual se encarga de ofrecer la atención necesaria a cada dispositivo de pesaje que se encuentre transmitiendo información.

Para garantizar la confiabilidad de la transmisión de información desde los dispositivos de pesaje hacia el sistema, el servicio levantará un hilo de proceso diferente por cada dispositivo de pesaje que se encuentre “ACTIVO”, estado necesario del dispositivo para que el servicio lo considere, y así ofrecer una cobertura total a las diferentes transmisiones de los dispositivos de pesaje.

3.7.4 Etiquetado de las cajas de empaque dependiendo del propietario de la materia prima.

El sistema, al trabajar con un conjunto de clientes diferentes, administrará en forma distintiva las etiquetas específicas de cada cliente, ofreciendo así, un servicio distintivo para cada uno de ellos, pero cumpliendo el resguardo que la administración del conjunto de etiquetas no se transforme en un perjuicio para el sistema. Para lograr este objetivo se crearán mantenedores específicos para las etiquetas, las cuales al ser declaradas, son guardadas directamente en el servidor del sistema y el código que las conforma es administrado mediante la utilización de “campos especiales”. La utilización de “campos especiales” para la generación de etiquetas ofrecerá la variabilidad necesaria para trabajar con ella, permitiendo así la reutilización de la etiqueta en las diferentes ordenes de empaque o ingreso a frigorífico

El conjunto de campos especiales a utilizar en el sistema es completamente administrable por el usuario y por lo tanto podrá ser tan extenso como el usuario desee.

Capítulo 4

4. Análisis de tecnologías a utilizar y aplicación de ellas.

4.1 Modelo de N-Capas

El modelo de N-Capas surge como la arquitectura predominante a la hora de realizar aplicaciones o soluciones multiplataforma en la mayor parte de las empresas de hoy, debido a su carácter de multiplataforma y a un conjunto de beneficios que se adhieren a su utilización, entre los que se cuentan:

- Desarrollos paralelos entre las capas: Esto es posible dado que las funcionalidades de cada capa se encuentran identificadas y diferenciadas, permitiendo desarrollar conjuntamente las diferentes capas que compone la solución.
- Soluciones más robustas: Debido a que cada capa encapsula una lógica específica de la solución en si, estableciéndose una diferenciación de roles que se adjudican a cada capa del desarrollo.
- Mantenimiento más sencillo: Debido a la misma encapsulación de las funcionalidades de la solución, el mantenimiento de cualquier aplicación desarrollada en un ambiente distribuido en n-capas se torna más simple, reduciendo tiempos de actualizaciones y mantenciones de la solución.
- Mayor flexibilidad de la solución: Debido a la distribución de la lógica de la solución, es posible añadir nuevas funcionalidades o módulos a la solución en forma más transparente, sin interferir en los desarrollos ya realizados y así acelerando las nuevas implementaciones.
- Mayor escalabilidad: Una de las principales ventajas de una aplicación distribuida diseñada correctamente es su buen escalado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente

añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir una buena escalabilidad.

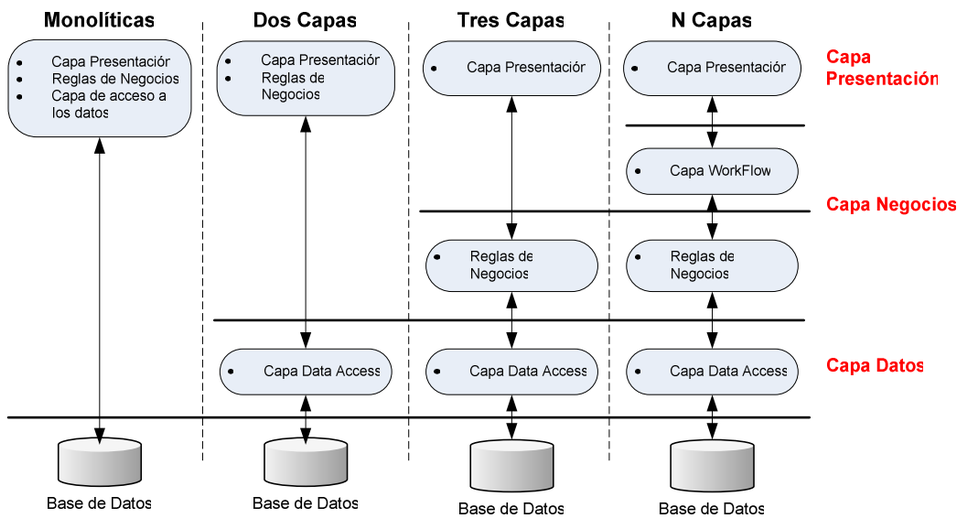


Figura 6, Modelo de N-Capas

A diferencia de lo que se pudiera pensar, el desarrollo en "n capas" no es un producto o un estándar, se encapsula en un concepto estratégico que ayuda a la construcción y despliegue lógico de un sistema distribuido.

Los sistemas de "n capas" subdivididos ayudan a facilitar el desarrollo rápido de aplicaciones y su posterior despliegue, con beneficios incrementales fruto de los esfuerzos del desarrollo en paralelo coordinado, resultando un decremento del tiempo de desarrollo y de sus costos involucrados.

Los sistema de "n-capas" apuestan a la utilización de clientes ligeros, clientes que no necesitan de instalación personalizada o de actualizaciones de las aplicaciones de lado del cliente, desplazando la capa de presentación de la aplicación en el lado del cliente o aplicaciones front-end, mientras que la lógica de

negocio y los datos residen en las capas intermedias o middleware[3] y los servidores de back−end[4]. El diseño para clientes ligeros minimiza los problemas de despliegue de las aplicaciones, mientras que maximiza la accesibilidad a la misma desde una amplia variedad de plataformas heterogéneas, abriendo así una gama de potenciales clientes que trabajen con la solución desarrollada.

La separación de la presentación, lógica de negocio y datos es realizada en un número indefinido de capas lógicas, permitiendo a cada capa ser desarrollada, potenciada, gestionada y desplegada de forma independiente. Esta es precisamente la base para el modelo de informática de red en "n capas". Las plataformas multicapas funcionan consistentemente a lo largo de un variado conjunto de hardware, permitiendo escalar las operaciones del negocio desde hardware no tan potente a prestadores de servicios robustos con un mínimo de impacto.

4.2 Evolución del modelo de N-Capas a WebServices

En la actualidad, las aplicaciones desarrolladas en "n-capas" se han tornado un referente para la nueva generación de proyectos de tecnologías de información para las empresas. Tomando este punto de partida, las aplicaciones tienden a ser divididas lógicamente, enfocando en cada división un grupo de funcionalidades especificas de cada solución. Lo más común es que la elección de una división en tres partes (presentación, lógica del negocio y finalmente almacenamiento y comunicación con repositorio de datos), aunque también existen otras posibilidades dependiendo del nivel de encapsulación que se pretende imprimir a

---

[3] Software de conectividad que permite ofrecer un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas

[4] Software encargado de realizar el procesamiento de las entradas Front-End

la solución. A pesar de que las aplicaciones de N niveles surgieron en un principio para solucionar algunos de los problemas relacionados con las aplicaciones cliente/servidor tradicionales, con la llegada del Web esta arquitectura ha pasado a desempeñar un papel predominante en la nueva programación.

La arquitectura en la que se basan los WebServices, es una arquitectura que se maneja en forma distribuida, dada esta característica, los webservices logran una vinculación con el modelo de aplicaciones de "n-capas". Esto se debe a que gran parte de su desarrollo se encuentra en diferentes plataformas y pueden estar en diferentes ubicaciones.

4.3 WebServices

Los WebServices o Servicios Web surgen ante la necesidad de establecer una estandarización de las comunicaciones entre las distintas plataformas y lenguajes de programación. Dada esta problemática, anteriormente se habían realizado intentos de crear estándares, pero no lograron un éxito adecuado, alguno de ellos como DCOM[5] (Microsoft) o CORBA (ORB) no lograron una penetración significativa ya que dependían directamente de la implementación del vendedor. [Liz02]

Otro problema que surgió era la utilización de RCP[6] para realizar las comunicaciones entre diferentes nodos. Esto debido a problemas de conectividad que se presentaban al momento de intentar establecer comunicaciones entre nodos, ya que la seguridad impuesta por los firewall bloqueaba este tipo de

---

[5] Remote Procedure Call o llamadas a procedimientos remotos

[6] Arquitectura de Negocios que permite la transferencia de servicios 1 a 1

comunicaciones. Es por eso que en 1999 se comenzó a plantear un nuevo estándar, el cual terminaría utilizando XML, SOAP, WSDL y UDDI.

4.3.1 Definición de los WebServices

Los WebServices son un sistema de software diseñado para soportar la interoperabilidad máquina-máquina a través de una red de comunicaciones. Este sistema tiene una interfaz descrita en un formato que puede ser procesado por una máquina (específicamente WSDL) y otros sistema podrán interactuar con el WebService utilizando mensajes SOAP, los cuales se encuentran establecidos previamente.[URL 5]

Dado esto, un WebService es un componente de software que se comunica con otras aplicaciones codificando los mensajes en XML y enviando estos mensajes a través de protocolos estándares de Internet, como por ejemplo HTTP[7]. Intuitivamente, un WebServices en similar a un sitio web que no cuenta con una interfaz para usuarios y que en el fondo da servicios o funcionalidades a las aplicaciones que lo requieran. Un WebServices, en vez de obtener solicitudes desde un navegador y retornar código HTML como respuesta, lo que realiza es recibir solicitudes a través de un mensaje formateado en XML desde una aplicación, realiza la operación por la cual fue diseñado y retorna un mensaje de respuesta también formateado en XML.

4.3.2 Requisitos que debe cumplir un WebServices

Los WebServices deben cumplir un conjunto de requisitos mínimos para ser denominados como tal, los cuales son: [URL 5]

---

[7] Hyper Text Transfer Protocol

a.- **Interoperabilidad**: Un servicio remoto debe permitir su utilización por clientes de distintas plataformas.

b.- **Amigabilidad con Internet**: La solución debe poder funcionar para soportar a clientes que accedan a los WebServices desde Internet.

c.- **Interfaces fuertemente tipadas**: No debe existir ambigüedad acerca del tipo de dato enviado y recibido desde un WebService, más aún, los tipos de datos definidos en el WebService debe corresponder a los tipos de datos de la mayoría de los lenguajes de programación procedural.

d.- **Posibilidad de aprovechar los estándares de Internet existentes**: La implementación del WebService deberá aprovechar estándares de Internet existentes y así evitar la generación de problemas que ya se han resuelto. Una solución construida sobre un estándar de Internet consolidado, podrá aprovechar un conjunto de herramientas y productos existentes creados para dicha tecnología.

e.- **Soporte para cualquier lenguaje**: El WebService no deberá comprometerse con algún lenguaje de programación específico. Un cliente deberá ser capaz de implementar un WebService independiente del lenguaje de programación en el que se haya realizado.

f.- **Soporte para cualquier infraestructura de componente distribuida**: El WebService no debe estar fuertemente ligado a una infraestructura de componentes en particular.

g.- **Descubrimiento**: La aplicación cliente que necesite acceder a una funcionalidad que ofrece un WebService necesita una forma de resolver la ubicación del WebService. Esto se logra mediante un proceso llamado normalmente “descubrimiento” (discovery). El “descubrimiento” se puede proporcionar mediante un directorio centralizado o por otros métodos pertinentes.

h.- Descripción: una vez que se ha resuelto el extremo de un WebService dado, el cliente necesita recopilar información suficiente como para poder operar en forma adecuada con el mismo. La descripción de un WebService implica meta datos estructurados sobre la interfaz que intenta utilizar la aplicación cliente, así como documentación escrita sobre el WebService, incluyendo ejemplos de uso (typelib).

i.- Formato del mensaje: Para poder realizar el intercambio de datos, tanto el cliente como el servidor tienen que estar de acuerdo en un mecanismo común de codificación y formato del mensaje. El uso de un mecanismo estándar de codificar los datos asegura que los datos que codifica el cliente los interpreta correctamente el servidor.

j.- Codificación: Los datos que se transmiten entre el cliente y el servidor necesitan codificarse en un cuerpo de mensaje.

k.- Transporte: Una vez que se ha dado el formato al mensaje y se han serializado los datos en el cuerpo del mensaje, se debe transferir entre el cliente y el servidor utilizando algún protocolo de transporte, por ejemplo: TCP/IP.

4.3.3 Elementos que componen los WebService

4.3.3.1 XML (eXtended Markup Language)

Es un lenguaje de marcas que ofrece un formato para la descripción de datos estructurados, es decir, XML es un metalenguaje, dado que con él podemos definir un lenguaje propio de presentación y a diferencia de HTML, que es centrado en la representación de la información, XML se centra en la información en si misma. La particularidad más importante de XML es que no posee etiquetas predefinidas con anterioridad, ya que es el propio diseñador el que las va creando de acuerdo a su necesidad, dependiendo directamente del contenido del documento. [Har05]

En el siguiente ejemplo se despliega una tabla con un código típico HTML y una versión equivalente en XML, se puede apreciar que es más interpretable la información que se intenta representar la versión en XML.

HTML	XML
<table> <tr> <td>Titulo</td> <td>Autor</td> <td>Precio</td> </tr> <td>La novia</td> <td>Juan Perez</td> <td>10000</td> </tr> <td>Clavel</td> <td>Pedro Reyes</td> <td>15000</td> </tr> <table>	<LIBROS> <libro> <titulo>La novia</titulo> <autor>Juan Perez</autor> <precio>10000</precio> </libro> <libro> <titulo>Clavel</titulo> <autor>Pedro Reyes</autor> <precio>15000</precio> </libro> </LIBROS>

Tabla 1. Tabla comparativa de gramática entre HTML y XML

XML fue desarrollado por un grupo de trabajo bajo los auspicios del consorcio World Wide Web (W3C). Este fue construido 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, Estados Unidos, el Instituto Nacional de Investigación en Informática 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. Al comenzar el proyecto, los objetivos planteados por el grupo de desarrollo del XML fueron diez puntos:

1. XML debe ser directamente utilizable sobre Internet.
2. XML debe soportar una amplia variedad de aplicaciones.
3. XML debe ser compatible con SGML (Standard Generalized Markup Language) o lenguaje de marcación generalizado.
4. Debe ser de fácil escritura de programas que procesen documentos XML.

5. El número de características opcionales en XML debe ser absolutamente mínimo, idealmente cero.
6. Los documentos XML deben ser legibles por los usuarios de este lenguaje y razonablemente claros.
7. El diseño de XML debe ser formal, conciso y preparado rápidamente.
8. XML debería ser simple pero perfectamente formalizado.
9. Los documentos XML deben ser fáciles de crear.
10. La brevedad en las marcas XML es la mínima importancia.

La meta fundamental del XML es hacer la cooperación y la interoperabilidad más fáciles entre módulos que pertenezcan a diferentes sistemas e inclusive, a diferentes organizaciones. Entonces, ¿Por qué XML se ha convertido en una tecnología de alto nivel tan rápidamente?, la respuesta es simple, XML es texto simple, pero a la vez también es elegante. Un lenguaje de marcas hace fácil el trabajo de identificar segmentos específicos de información con significados especiales. Usar texto simple y mantener el contenido universalmente interpretable son dos condiciones para implementar una tecnología exitosamente entre sistemas heterogéneos.

La principal fortaleza del XML es también, actualmente, su debilidad principal. El XML es demasiado genérico para ser utilizado sin definir externamente la sintaxis exacta de un documento. Tal estandarización es más simple alcanzar dentro de una sola aplicación corporativa, mas conseguir el mismo resultado en un contexto más amplio requiere un enorme esfuerzo para llegar a algunas definiciones estándares.

Desgraciadamente, cuando se trata de integrar sistemas de diferentes organizaciones, especialmente de una manera abierta, el XML muestra algunas

limitaciones. El XML trata de la descripción de los datos, pero solamente es útil cuando las personas están de acuerdo en la manera en que los datos deben ser descritos.

4.3.3.2 SOAP

Es un estándar propuesto por Microsoft, IBM y otros al consorcio WWW (W3C) para el intercambio de mensajes entre servicios Web y los consumidores de este servicio.

El protocolo SOAP esta construido sobre XML y solo describe el formato de los mensajes dejando abierta la posibilidad de usar varios transportes, aunque actualmente el transporte usado es HTML. La elección de HTML como transporte se debe a que es el transporte más utilizado por las organizaciones a nivel actual para establecer cualquier tipo de comunicación con su entorno.

El protocolo define un "sobre" en el que se empaqueta el requerimiento donde se especifica el destinatario de la llamada, el nombre del método que se invoca y opcionalmente una serie de parámetros con tipos definidos. La respuesta a este requerimiento se empaqueta de la misma forma, en un "sobre" que contiene el resultado del método invocado.

La utilidad de SOAP consiste en que con un conjunto de servicios simples se puede implementar aplicaciones que entreguen funcionalidades valiosas mediante la integración de estos servicios básicos o mediante el uso programático de estos. [URL 6]

SOAP se enmarca dentro de lo que se llama "component based arquitectures", o sea arquitecturas basadas en componentes. La idea principal

es que es más fácil identificar ciertas tareas que se programan en reiteradas ocasiones y en vez de implementarlas cada vez que se requieran, es más eficiente realizarlas una sola vez, de manera que puedan ser invocadas desde una variedad de aplicaciones clientes o consumidores de servicios. De esto se desprende una de las cualidades fundamentales que debe poseer la tecnología necesaria para la creación de componentes y así ser flexible, para permitir la incorporación de estas funcionalidades desde plataformas diversas, por ejemplo, una aplicación Web desarrollada en PHP o JSP o una aplicación de escritorio desarrollada en Visual Basic o C#, por nombrar algunas.

Otras tecnologías que persiguen estos objetivos son por ejemplo la arquitectura COM de Microsoft o la arquitectura CORBA de OMG, que permite la invocación de métodos de objetos que pueden ser remotos. La ventaja directa de SOAP frente a estos esquemas de desarrollo es que más sencillo de implementar y dado que es XML vía HTTP, el vocabulario y el método de transporte son generalizados o cien por ciento difundidos en la actualidad.

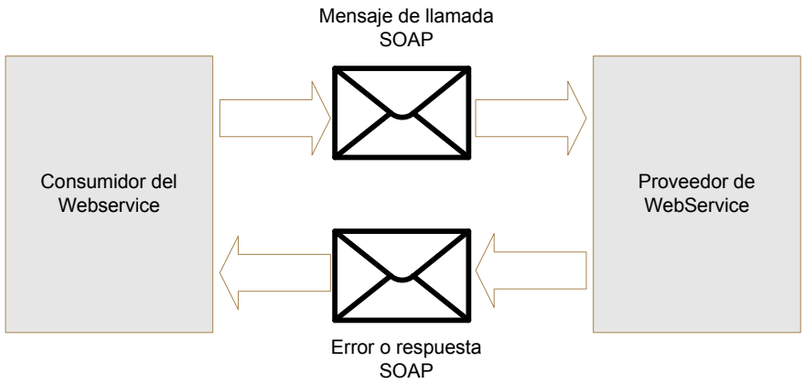


Figura 7, Solicitud Respuesta SOAP

4.3.3.3 WSDL (WebService Description Language)

Una de las ideas centrales detrás de los WebServices o Servicios Web es que las aplicaciones futuras estarán conformadas de una colección de servicios habilitados en una red. Mientras exista dos servicios equivalentes que se publiciten a la red de una forma estándar y neutra, en teoría una aplicación podrá seleccionar una de ellas en base a criterios establecidos de antemano como el precio o rendimiento del servicio. Además algunos servicios podrán permitir que fueran replicados entre máquinas, facilitando así que una aplicación que se ejecuta en una máquina mejore el rendimiento al replicar servicios a unidades de almacenamiento locales. [Chr01]

Se podría realizar una analogía de esta situación con la de prestaciones de servicios de una compañía que ofrece un conjunto de asistencias a productos, la compañía en si ofrece una gama de prestaciones a productos específicos y por contraparte se tiene a compañías deseosas de contratar servicios para aplicar a sus productos fabricados. Si se encuentra una buena opción, las partes interesadas intentan negociar condiciones aceptables para ambas, si se logra un acuerdo, la empresa solicitadora del servicio específico selecciona la funcionalidad o funcionalidades que le ofrezcan mayor garantía de una gama de empresas prestadoras del mismo servicio.

El Lenguaje de Descripción de Servicios Web (WSDL) es el equivalente de un resumen en XML describiendo los WebServices, donde se ubican y como poder invocarlos.

Dentro del documento WSDL se encuentran las siguientes definiciones:

- **<definitions>**: El elemento <definitions> contiene la definición de uno o más servicios. En la mayoría de los casos, un archivo WSDL se define un

servicio únicamente. Seguido de la etiqueta de definición se encontrarán declaraciones de algunos atributos, los cuales son:

- **<message>** y **<portType>:** Describe que operaciones provee el servicio.
- **<binding>:** Describe como se invocan las operaciones.
- **<service>:** Describe donde se ubica el servicio.
- **<documentation>:** Cualquier elemento WSDL puede contener información del servicio para el usuario.

4.3.3.4 Disco

DISCO (Discovery of WebService), es una especificación creada para ayudar a determinar como pueden encontrar los servicios Web en un servidor. Un archivo DISCO contiene el Identificador de recursos uniforme (URI[8]), de cada servicio Web disponible en ese equipo. El URI apunta normalmente al documento WSDL, que a su vez apunta al servicio Web en cuestión.

4.3.3.5 UDDI

UDDI[9] es un WebService en línea que puede ser utilizado desde las aplicaciones para descubrir de forma dinámica otros servicios en línea, todos ellos perfectamente integrados en una interfaz XML simple.

La especificación UDDI ha sido desarrollada por IBM, Microsoft y Ariba y está soportada por más de trescientas empresas, incluyendo a Oracle, Sun Microsystem y Nortel Networks. La última versión extiende la funcionalidad UDDI para mejorar el soporte con objetos para desarrollos de registros públicos y

---

[8] Uniform Resource Identifier

[9] Universal Description Discovery Integration.

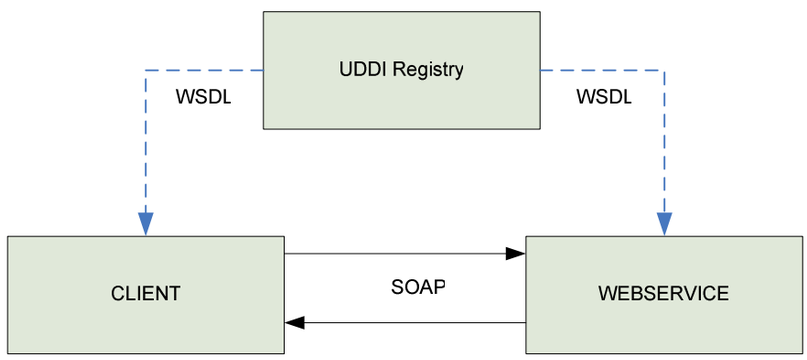
privados que gestionen los servicios Web internos empleando las especificaciones UDDI.

Estas empresas actúan como catalizadores iniciales para desarrollar la plataforma, así como sistemas relacionados. Los sistemas e-buniness de las compañías miembro, por lo tanto, pueden conectarse sin dificultad y combinarse en la creación de nuevas aplicaciones.

La especificación UDDI, junto con XML (eXtensible Markup Language), SOAP (Simple Object Access Protocol) y WSDL (Web Services Description Language) están ganando un amplio soporte en el marco de trabajo de los WebServices. En fin, existe un número mas que importante de empresas que se han registrado en el directorio UDDI.

4.3.3.6 SOAP, WSDL, UDDI y WebServices

SOAP a diferencia de XML-RCP[10], incluye una infraestructura a su alrededor. No es un simple protocolo de comunicaciones entre computadores, sino que además se rodea de términos como WSDL y UDDI, para comprender mejor esta afirmación obsérvese la siguiente figura:



[10] eXtensible Markup Language-Remote Call Procedure.

Figura 8, Vinculación entre UDDI, Cliente, SOAP y WebService

Las repercusiones que pueden ofrecer los WebServices a los usuarios actuales de Internet son mínimas, ya que los WebServices son herramientas que están directamente involucradas con los programadores de funcionalidades que intentas ofrecer métodos estandarizados a funcionalidades que son comunes a muchos usuarios. [URL 7]

4.3.3.7 Posibles riesgos de los WebServices

Las expectativas que se presentan frente a esta tecnología son amplias, mas hay que considerar que la aplicación de esta tecnología presenta ciertos riesgos, los cuales son:

- Los WebServices hacen uso de las mismas tecnologías que han sido atacadas en innumerable de ocasiones. Utilizando WebServices la seguridad de una empresa puede verse comprometida debido a la ausencia de técnicas de seguridad estándar para poder aplicar este tipo de soluciones.
- La calidad de los WebServices es un parámetro que no queda especificado con la suficiente claridad, ya que se puede caer en una expectativa de servicio la cual no cumple con los requisitos preestablecidos.
- Al ser una tecnología en desarrollo, la mayoría de los protocolos que se ven involucrados no han alcanzado un nivel de madurez tecnológico para ofrecer las garantías de seguridad que algunos clientes esperan recibir.

4.3.3.8 Calidad de los WebServices

Actualmente ya existen en el mercado herramientas específicamente diseñadas para medir la calidad de los WebServices, pero siguen siendo necesaria una estandarización sobre el tema. Los resultados sobre la calidad de los diferentes

WebServices servirán como parámetro referencial de comparación y aportarán al cliente de estos servicios a buscar un servicio específico que le ofrezca las garantías necesarias para operar con él. [Dan01]

Para que un WebServices se ejecute en la forma esperada y cumpla las expectativas para la cual ha sido creado, habrá que tener en cuenta una serie de parámetros como por ejemplo que los resultados obtenidos por el mismo sean los adecuados o simplemente que la dificultad de su uso sea la mínima. Otro elemento a tener en cuenta es el nivel de integración que debe ofrecer el servicio, ya que simplemente deberá cumplir con las expectativas genéricas de un sin fin de clientes que utilizaran su servicio, tratando de abarcar la mayoría de los casos posible para lo cual fue diseñado.

Teóricamente, los WebServices ofrecen conectividad en forma transparente para cualquier software que cumpla con los requisitos mínimos para operar con él, cada proveedor de servicios puede adoptar soluciones diferentes que resulten más o menos adecuadas para el cliente que requiera el servicio. Al adoptar especificaciones para dar soluciones a medida a un cliente puntual, puede atentar a la escalabilidad del mismo servicio ya que las necesidades de un cliente no siempre son compartidas por el resto de los usuarios del mismo servicio.

Por último, ha esta altura no existen herramientas que ofrezcan garantías de calidad sobre el WebService que puede ofrecer un proveedor de servicios, ya que se actúa en medida a las expectativas ofrecidas por el servicios, mas esto no garantiza que cumpla en un cien por ciento de acuerdo a lo que se ha planteado en un comienzo.

4.3.3.9 Estandarización de los WebServices

Los WebServices, como se ha explicado anteriormente, están basado en XML, que ha sido universalmente aceptado en un sin fin de protocolos y sistemas que operan actualmente. Pero la situación para el resto de los protocoles que se ven involucrados en la tecnologías de los WebServices es distinta, esto debido a que se encuentran en permanente desarrollo y por lo tanto son objetos a ofrecer cierta evolución. Dada esta situación, muchas empresas están a la expectativa de lo que pueda suceder con las determinaciones finales para cada protocolo, mas bien esperando una estandarización generalizada para invertir en esta tecnología.

Actualmente, los protocolos que se ven involucrados en esta tecnología se ven trabados en la estandarización debido a las técnicas de seguridad que las envuelven, solo SOAP es considerado por el World Wide Web Consortium (W3C) y se encuentra en tramites de estandarización. SOAP y WSDL son a esta altura ampliamente utilizados pero por el momento UDDI no ha tenido el mismo nivel de penetración de los protocolos anteriormente descritos.

Capítulo 5

5. Diseño e implementación de módulo de “Recepción de Materia Prima”

5.1 Modelo de datos vinculado con el módulo de Recepción de materia Prima

Como parte del diseño del módulo, se debe analizar el modelo de datos a utilizar, con el fin de describir los datos a almacenar y el flujo que habrá entre ellos.

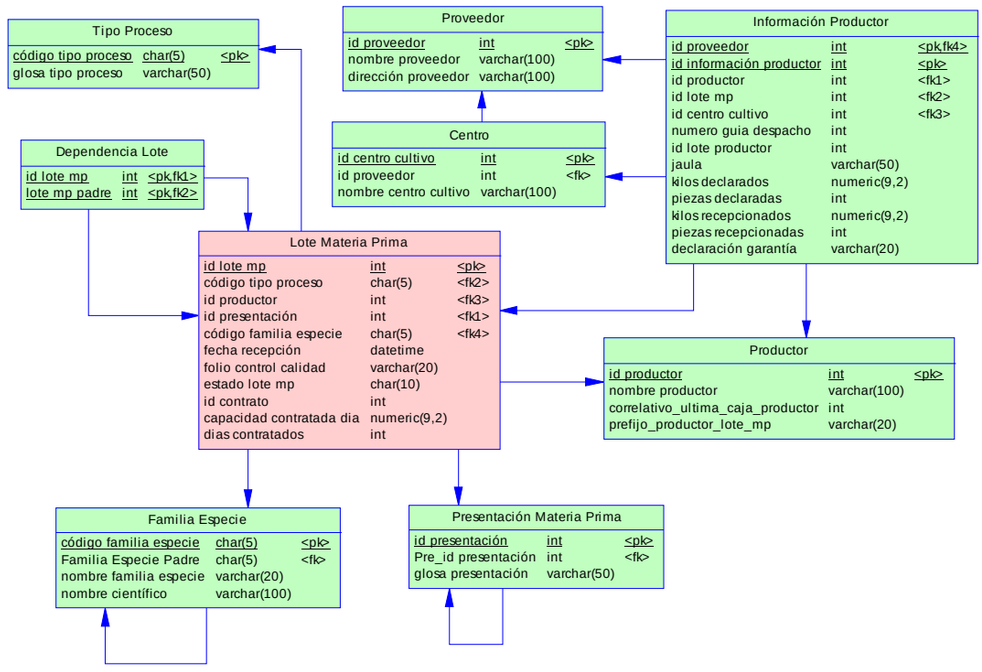


Figura 9, Modelo físico de Datos módulo de Recepción de Materia Prima

5.2 Diagrama Casos de Uso para módulo de “Recepción de Materia Prima”

A continuación se presentan los actores, diagrama y el detalle de los Casos de Uso definidos según los requerimientos establecidos para el módulo de “Recepción de Materia Prima”.

Los Actores que se presentan son:

- **Control de Producción:** Son los usuarios del departamento de control de producción, encargados de administrar la información básica del sistema.
- **Control de Calidad**: Es el actor que se encargará de ingresar y generar la información correspondiente al Departamento de Control de Calidad de la empresa.

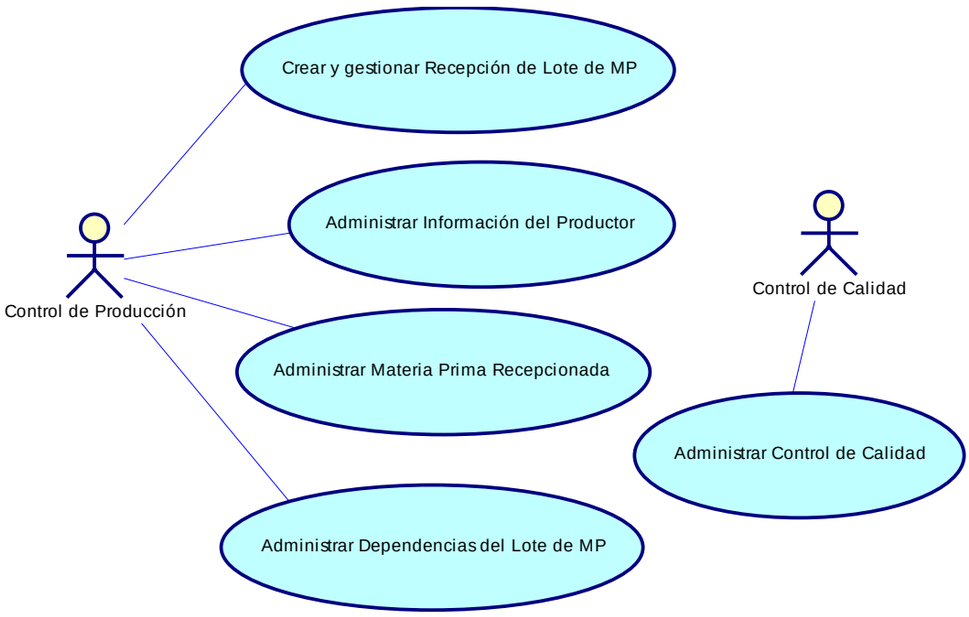


Figura 10, Diagrama de casos de uso de Módulo de “Recepción de Materia Prima”

5.3 Plantillas Casos de Uso para “Módulo de Recepción de Materia Prima”

5.3.1 Caso de Uso “Crear y gestionar Recepción de Lote de Materia Prima”

Actor: *Control de Producción*	**Sistema**
1. Despliega la lista de lotes de materia prima existentes pudiendo para ello utilizar un filtro.	
2. Indica mediante un botón que desea agregar un nuevo lote.	3. Despliega una pantalla con los campos vacíos en la cual se requiere la información básica para definir un lote de materia prima.
4. Ingresa la información básica del lote a definir, menos su identificador porque será asignado automáticamente por el Sistema.	
5. Guarda los cambios realizados presionando un botón para ello.	6. Guarda en la base de datos la información ingresada por el actor.
	7. Sistema retorna identificador del Lote de Materia Prima
8. Podrá editar la información de un lote de materia prima existente mediante la opión de "Modificar".	9. Despliega la misma pantalla de (3) pero con los campos autocompletados de acuerdo al lote seleccionado.
Excepciones:	

Tabla 2. Planilla Caso de Uso: Creación y gestión de Recepción de Lote MP

5.3.2 Caso de Uso “Administrar Información de Propietario”

Actor: *Control de Producción*	**Sistema**
1. Selecciona pantalla de ingreso de información de producción una vez ya creado el lote de materia prima.	2. Completa los controles maestros de proveedores y centros desde el repositorio de datos.
3. Indica mediante un botón que desea agregar una nueva "información de Propietario".	4. Despliega una pantalla con los campos vacíos en la cual se requiere la información básica para definir la información del Propietario.
5. Ingresa la información básica a definir.	7. Guarda en la base de datos la información ingresada por el actor.
6. Guarda los cambios realizados presionando un botón para ello.	
8. Si existe más información a ingresar, retorna al punto (3).	

9. Podrá editar la información del Propietario mediante la opción de "Modificar".	10. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	

Tabla 3. Planilla Caso de Uso: Administrar Información de Propietario

5.3.3 Caso de Uso “Administrar materia prima recepcionada”

Actor: *Control de Producción*	**Sistema**
1. Selecciona pantalla de "Materia Prima Recepcionada" una vez ya creado el lote de materia prima.	2. Recupera la "información de Propietario" que se vincula a la recepción de materia prima.
3. Selecciona e ingresa la "información de Propietario a completar".	
4. Ingresa la información básica a definir.	
5. Guarda los cambios realizados presionando un botón para ello.	6. Guarda en la base de datos la información ingresada por el actor.
7. Podrá editar la información del Propietario mediante la opción de "Modificar".	8. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	

Tabla 4. Planilla Caso de Uso: Administrar materia prima recepcionada

5.3.4 Caso de Uso “Administrar dependencias del Lote de Materia prima”

Actor: *Control de Producción*	**Sistema**
1. Selecciona pantalla de "Dependencias del Lote de Materia Prima" una vez ya creado el lote de materia prima.	2. Recupera información de dependencias previas, si existiesen.
3. Ingresa el identificador de Lote de Materia Prima del cual depende.	
5. Guarda los cambios realizados presionando un botón para ello.	6. Guarda en la base de datos la información ingresada por el actor.
7. Si existiese más de un Lote de Materia Prima del cual depende se retorna al punto (3).	
8. Si se desea heredar la información del Propietario, el actor selecciona la opción "Heredar Información del Lote"	9. Busca la información de Propietario del lote que se está heredando y la duplica para el lote de materia prima.

10. Podrá quitar dependencias del lote de materia prima seleccionando el lote que genera la dependencia y presionando el botón "Quitar dependencia".	11. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	
1. El actor no podrá heredar información del Propietario si el lote a que se hace referencia no es del mismo Propietario y especie definido en el lote de materia prima.	

Tabla 5. Planilla Caso de Uso: Administrar dependencias del Lote de Materia prima

5.3.5 Caso de Uso "Administrar Control de Calidad"

Actor: *Control de Calidad*	**Sistema**
1. Mira la lista de lotes de materia prima existentes con la posibilidad de acotar el rango de lotes a través de un filtro..	
2. Ingresa a la información específica de un lote de materia prima haciendo clic en su identificador.	3. Despliega una pantalla con la información del lote de materia prima. Toda la información desplegada no será modificable, salvo la encontrada en la página de "Control de Calidad".
4. Ingresa o modifica el número de folio asignado a este lote de materia prima y las declaraciones de garantía para cada una de sus guías de despacho asociadas.	
5. Guarda los cambios realizados presionando un botón para ello.	6. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	
1. Bloquea un lote de materia prima activo presionando el botón "Bloquear" en la fila del lote seleccionado.	2. Cambia el estado del lote de materia a "Bloqueado".
3. Desbloquea un lote de materia prima activo presionando el botón "Desbloquear" en la fila del lote seleccionado.	4. Cambia el estado del lote de materia a "Activo".

Tabla 6. Planilla Caso de Uso: Administrar Control de Calidad

5.4 DIAGRAMAS DE SECUENCIAS

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

5.4.1 Diagrama de Secuencia “Crear y gestionar Recepción de Lote de Materia Prima”

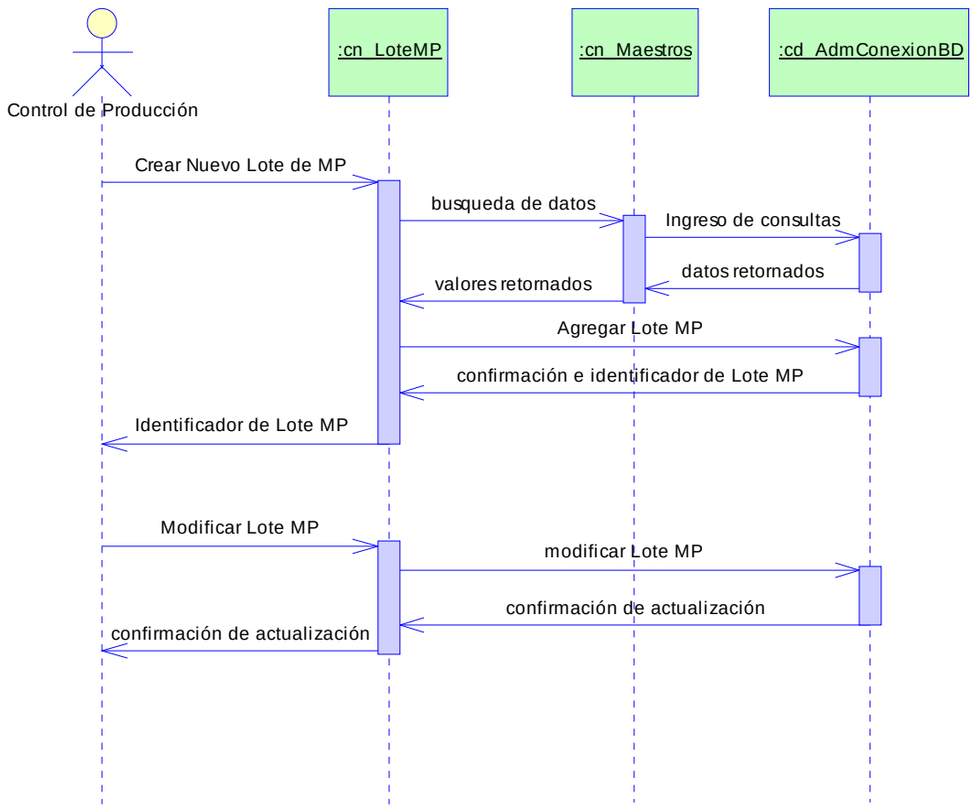


Figura 11, Diagrama de secuencia “Crear y gestionar Recepción de Lote de Materia Prima”

5.4.2 Diagrama de Secuencia “Administrar Información de Propietario” y “Administrar materia prima recepcionada”

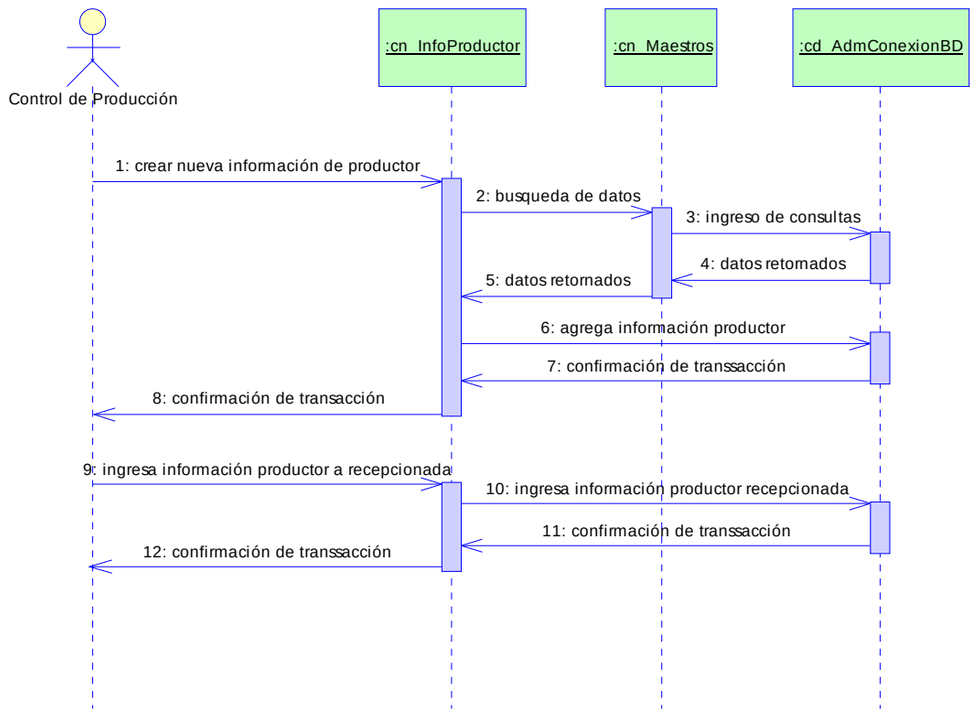


Figura 12, Diagrama de secuencia “Administrar Información de Propietario”

5.4.3 Diagrama de Secuencia “Administrar dependencias del Lote de Materia prima”

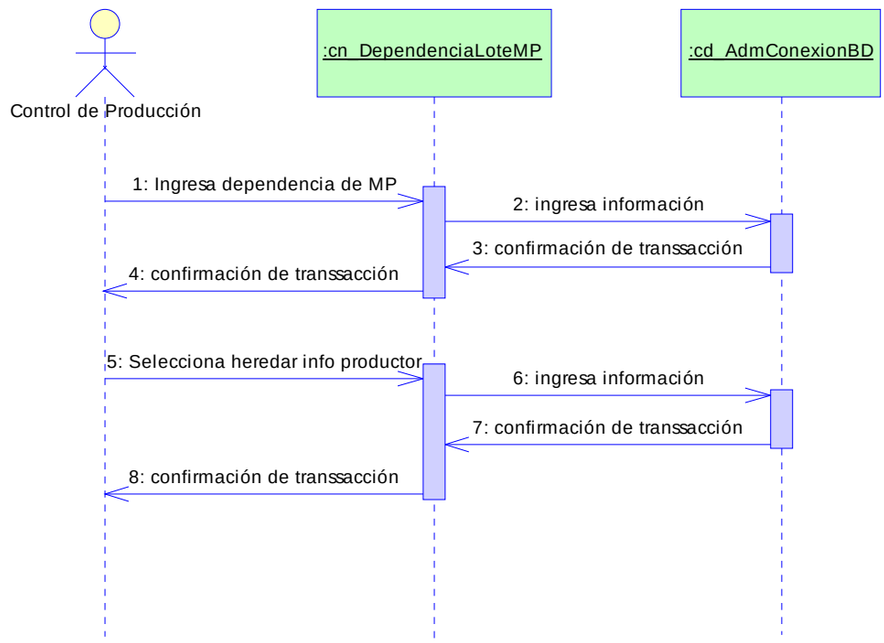


Figura 13, Diagrama de secuencia “Administrar dependencias del Lote de Materia prima”

5.4.4 Diagrama de Secuencia “Administrar Control de Calidad”

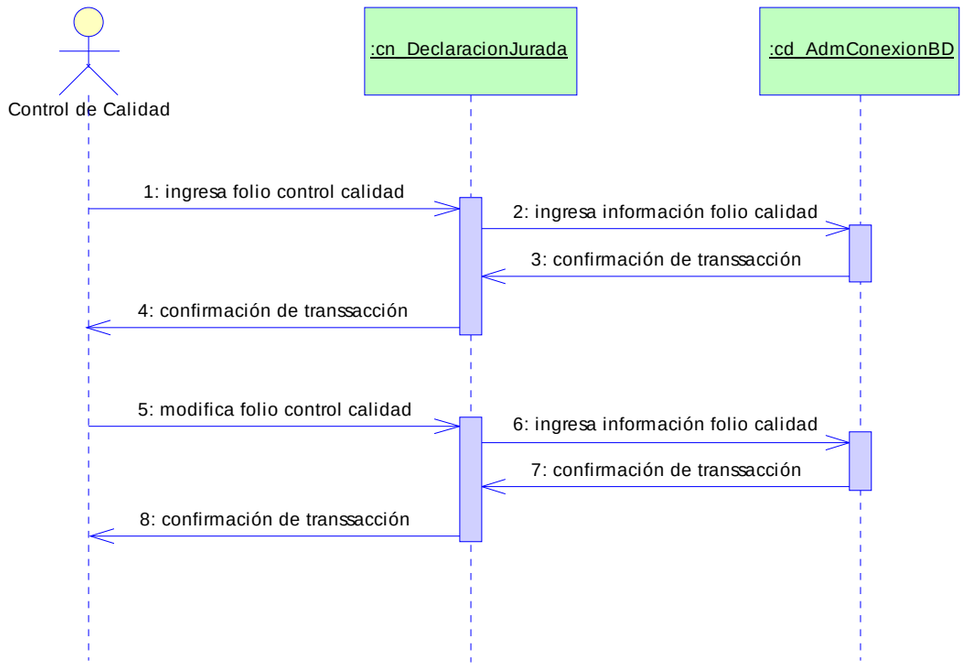


Figura 14, Diagrama de secuencia “Administrar Control de Calidad”

5.5 Diagramas de Colaboración

Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia. [URL 8]

5.5.1 Diagrama de Colaboración “Crear y gestionar Recepción de Lotes de Materia Prima”

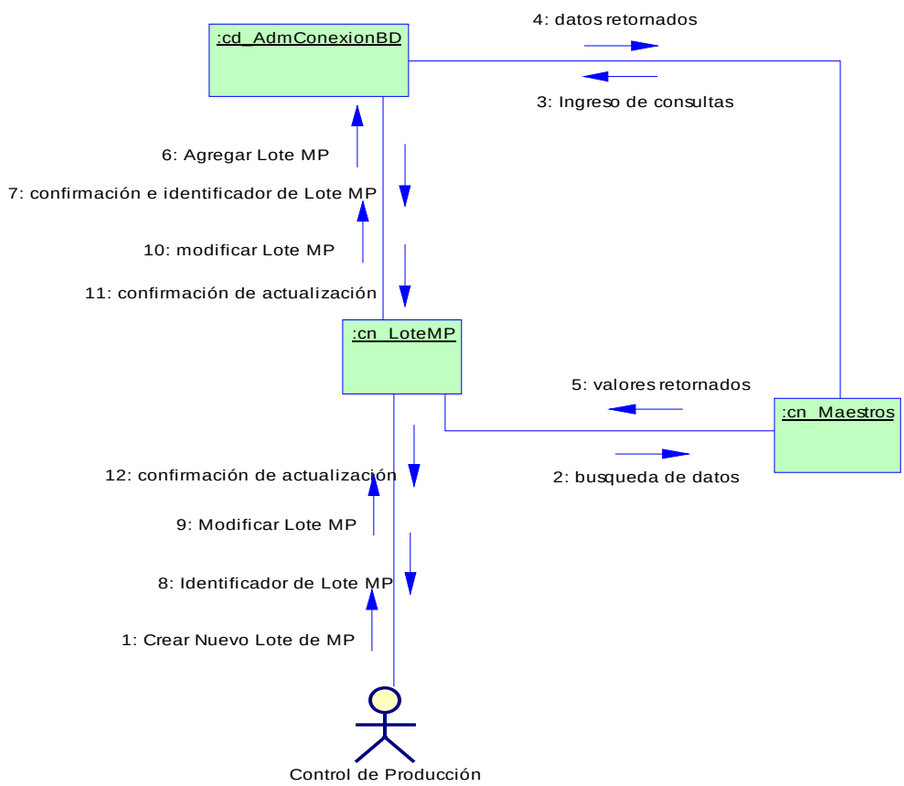


Figura 15, Diagrama de colaboración “Crear y gestionar Recepción de Lote de Materia Prima”

5.5.2 Diagrama de Colaboración “Administrar Información de Propietario” y “Administrar materia prima recepcionada”

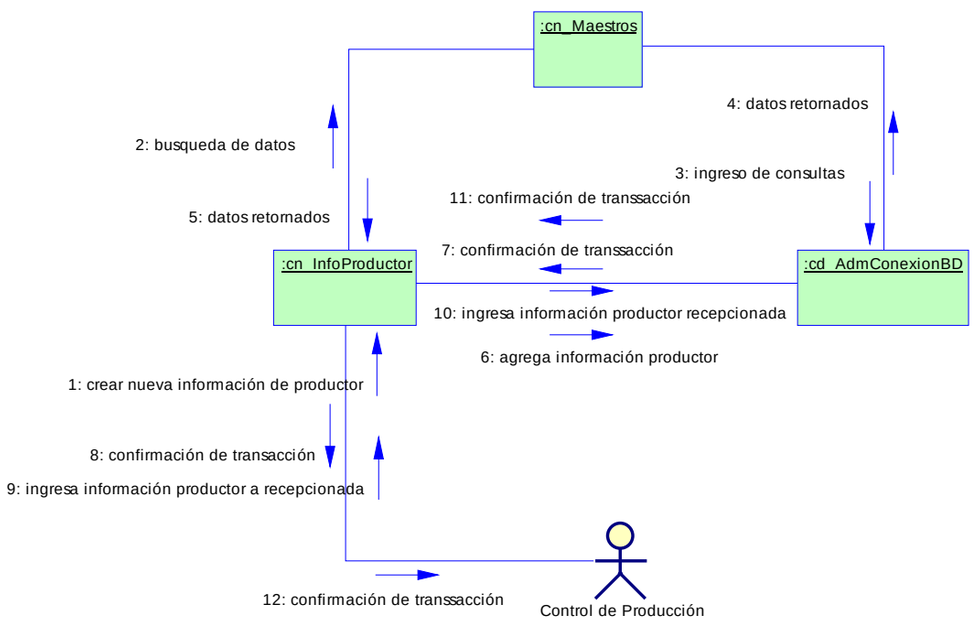


Figura 16, Diagrama de colaboración “Administrar Información de Propietario”

5.5.3 Diagrama de Colaboración “Administrar dependencias del Lote de Materia prima”

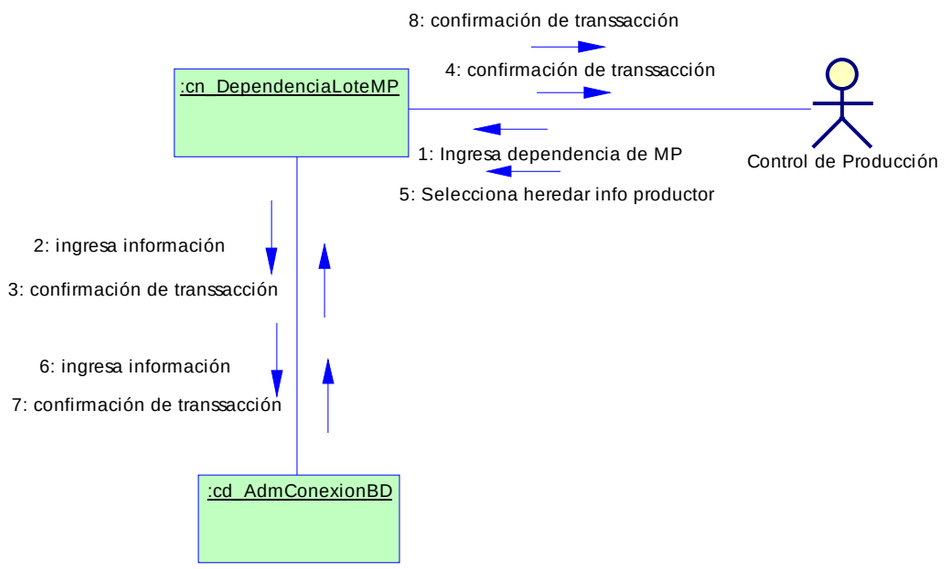


Figura 17, Diagrama de colaboración “Administrar dependencias del Lote de Materia prima”

5.5.4 Diagrama de Colaboración “Administrar Control de Calidad”

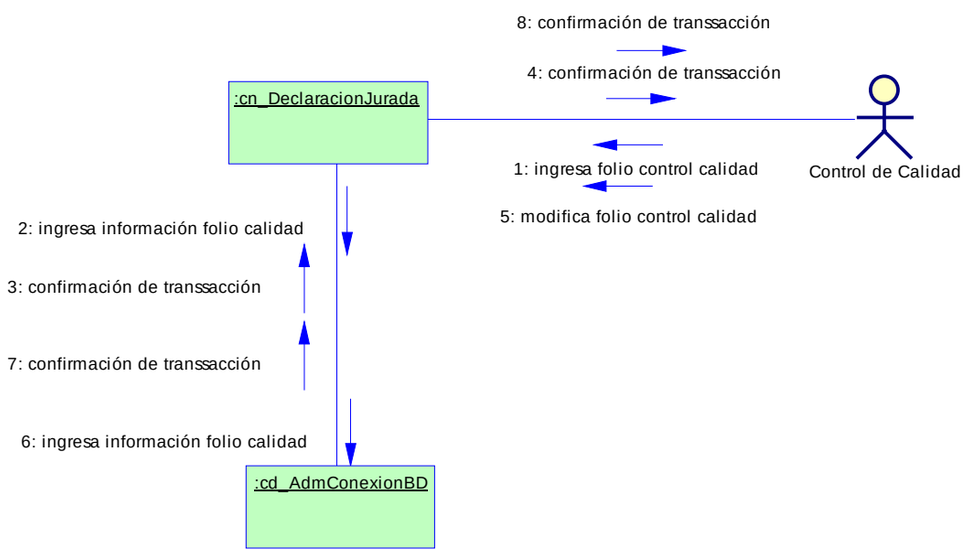


Figura 18, Diagrama de colaboración “Administrar Control de Calidad”

5.6 Descripción de los WebServices a implementar

Los WebServices y sus métodos a implementar para le módulo de “Recepción de Materia Prima” serán los siguientes:

a.- **cd_AdmConexiónBD**: WebServices encargado de administrar la conectividad del sistema con la base de datos, el cual proveerá de conexiones no permanente debido a la plataforma Web en el cual se desenvuelve el sistema; los métodos desarrollados para este WebServices son:

   i. GetDataset: Método empleado para la recuperación de información desde el repositorio de información, retorna un booleano con la confirmación de ejecución exitosa.

   ii. ExecQuery: Método empleado para la ejecución de instrucciones SQL las cuales no retornan información, el método retornará un booleano con la confirmación de ejecución exitosa.

b. **cn_LoteMP**: WebServices encargado de gestionar y administrar toda la información relevante para la generación de lotes de materia prima y su posterior administración., los métodos desarrollados para este WebServices son:

   i. boListarLoteMP: Método encargado de retornar un listado de lotes de materia prima de acuerdo a un listado de parámetros, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.

   ii. boAgregarLoteMP: Método encargado de insertar un lote de materia prima, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.

   iii. boActualizarLoteMP: Método encargado de actualizar un lote de materia prima, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.

c. **cn_Maestros**: WebServices encargado de recuperar toda la información relevante de acuerdo a los datos básicos o tablas maestras que el sistema necesita conocer, los métodos desarrollados para este WebServices para el módulo de recepción de materia prima son:

   i. boListarTiposProceso: Método encargado de recuperar el listado de los tipos de procesos a aplicarse a un lote de materia prima, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.

   ii. boListarPresentacion: Método encargado de recuperar el listado de los tipos de presentaciones a aplicarse a la materia prima recepcionada, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.

   iii. boListarEspecies: Método encargado de recuperar el listado de especies definas en el sistema, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.

   iv. boListarProductores: Método encargado de recuperar el listado de productores definidos en el sistema, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.

   v. boListarProveedor: Método encargado de recuperar el listado de proveedores definidos en el sistema, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.

   vi. boListarCentros: Método encargado de recuperar el listado de centros vinculados a un proveedor de materia prima definidos en el

sistema, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.

d. **cn_InfoProductor**: WebServices encargado de gestionar toda la información relevante de acuerdo a la información de los productores de la materia prima, los métodos desarrollados para este WebServices son:
    i. boListarInfoProductor: Método encargado de recuperar el listado de información de los productores de materia prima declarados en el sistema, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.
    ii. boAgregarInfoProductor: Método encargado de insertar la información correspondiente a un origen de materia prima de acuerdo a un proveedor, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.
    iii. boActualizarInfoProductor: Método encargado de actualizar la información correspondiente a un origen de materia prima de acuerdo a un proveedor, si la ejecución fue exitosa retornará un booleano de valor "true" confirmando su ejecución, en su defecto retornará un "false" junto con un mensaje de error.
e. **cn_DeclaraciónJurada**: WebServices encargado de gestionar toda la información relevante de los controles de calidad de acuerdo a la información de los productores de la materia prima, los métodos desarrollados para este WebServices son:
    i. boListarDeclaracionJurada: Método encargado de recuperar el listado de declaraciones juradas definidas por control de calidad, si la ejecución fue exitosa retornará un booleano de valor "true"

confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

ii. boInsertarDeclaracionJurada: Método encargado de insertar una declaración jurada definida por control de calidad, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

iii. boActualizarDeclaracionJurada: Método encargado de actualizar una declaración jurada definida por control de calidad, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

f. **cn_DependeciaLoteMP**: WebServices encargado de gestionar toda la información relevante de acuerdo a las posibles dependencias que pueda definirse a un lote de materia prima, los métodos desarrollados para este WebServices son:

i. boListarDependenciaLoteMP: Método encargado de recuperar el listado de dependencias de un lote de materia prima definidas por los encargados de control de calidad, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

ii. boAgreagarDependenciaLoteMP: Método encargado de insertar una dependencia de un lote de materia prima, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

iii. boBorrarDependenciaLoteMP: Método encargado de eliminar una dependencia de un lote de materia prima, si la ejecución fue exitosa

retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

5.7 Principales pantallas del Módulo de Recepción de Materia Prima

5.7.1 Pantalla “Lista Lotes de Materia Prima”

Pantalla en la cual se realiza el despliegue histórico de los lotes de materia prima definidos en el sistema, posibilitando al usuario la selección de un grupo de ellas mediante la especificación de un rango de lotes validos, para ello la aplicación Front-end establecerá comunicación con el WebServices de “cn_LoteMP” para el intercambio de información y confirmación de transacciones realizadas.

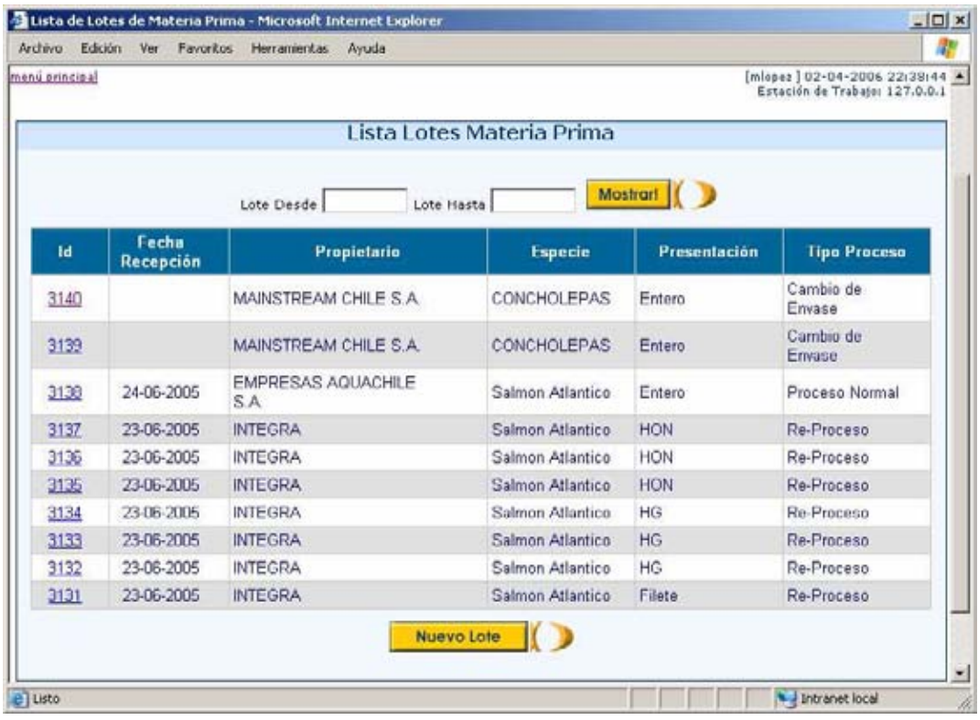


Figura 19, Pantalla “Lista Lotes de Materia prima”

5.7.2 Pantalla “Agregar y Actualizar Lotes de Materia Prima”

Pantalla en la cual se realiza la inserción de nuevos lotes de materia prima para el sistema y su posterior modificación, para ello la aplicación Front-end establecerá comunicación con el WebServices de “cn_LoteMP” y “cn_Maestros” para el intercambio de información y confirmación de transacciones realizadas.

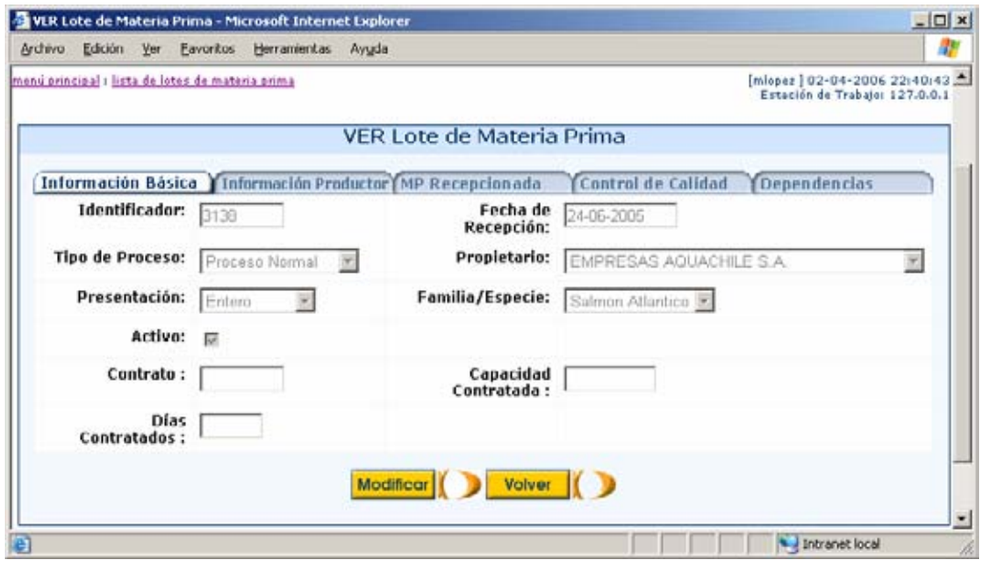


Figura 20, Pantalla “Agregar y Actualizar Lotes de Materia Prima”

5.7.3 Pantalla “Información de Productor de Lote de Materia Prima”

Pantalla en la cual se realiza la inserción de nuevas informaciones de productor para un lote de materia prima y su posterior modificación, para ello la aplicación Front-end establecerá comunicación con los WebServices de “cn_InfoProductor” y “cn_Maestros” para el intercambio de información y confirmación de transacciones realizadas.

Figura 21, Pantalla "Información de Productor de Lote de Materia Prima"

5.7.4 Pantalla "Control de Calidad para Lote de Materia Prima"

Pantalla en la cual se realiza la inserción de los controles de calidad por el departamento de Calidad para un lote de materia prima y su posterior modificación, para ello la aplicación Front-end establecerá comunicación con el WebServices de "cn_DeclaraciönJurada" para el intercambio de información y confirmación de transacciones realizadas.

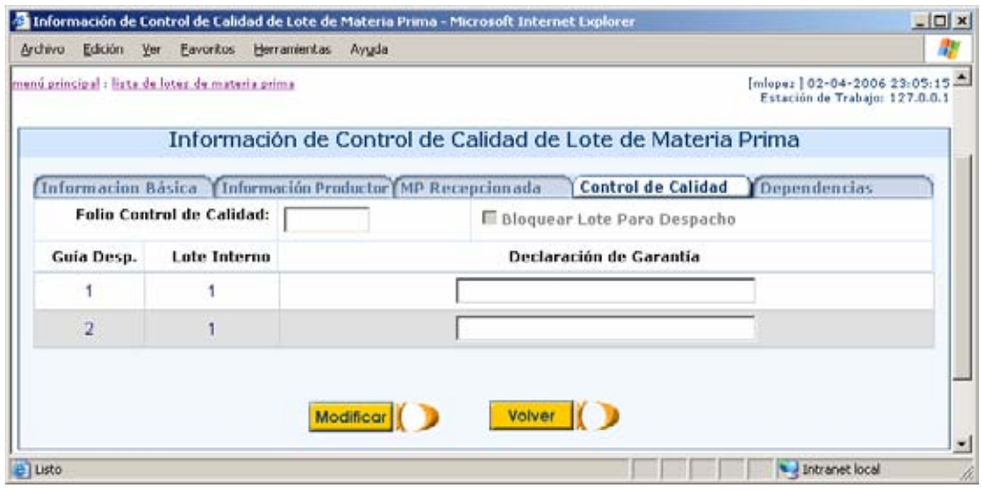


Figura 22, Pantalla "Control de Calidad para Lote de Materia Prima"

5.7.5 Pantalla "Dependencias para Lote de Materia Prima"

Pantalla en la cual se realiza la inserción de las dependencias para un lote de materia prima y su posterior modificación, para ello la aplicación Front-end establecerá comunicación con el WebServices de "cn_DependenciasLoteMP" para el intercambio de información y confirmación de transacciones realizadas.

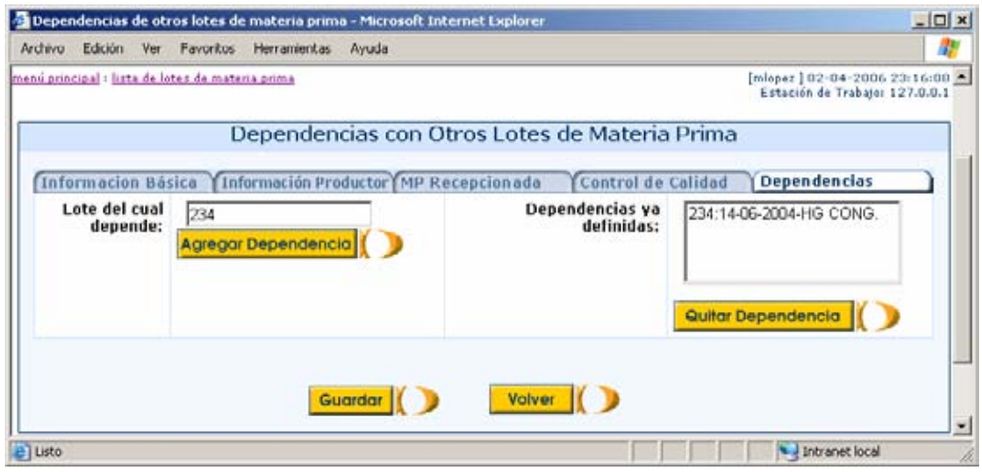


Figura 23, Pantalla "Dependencias para Lote de Materia Prima"

Capítulo 6

6. Diseño e implementación de módulo de “Empaque de Productos”

6.1 Modelo de datos vinculado con el módulo de Empaque de Productos

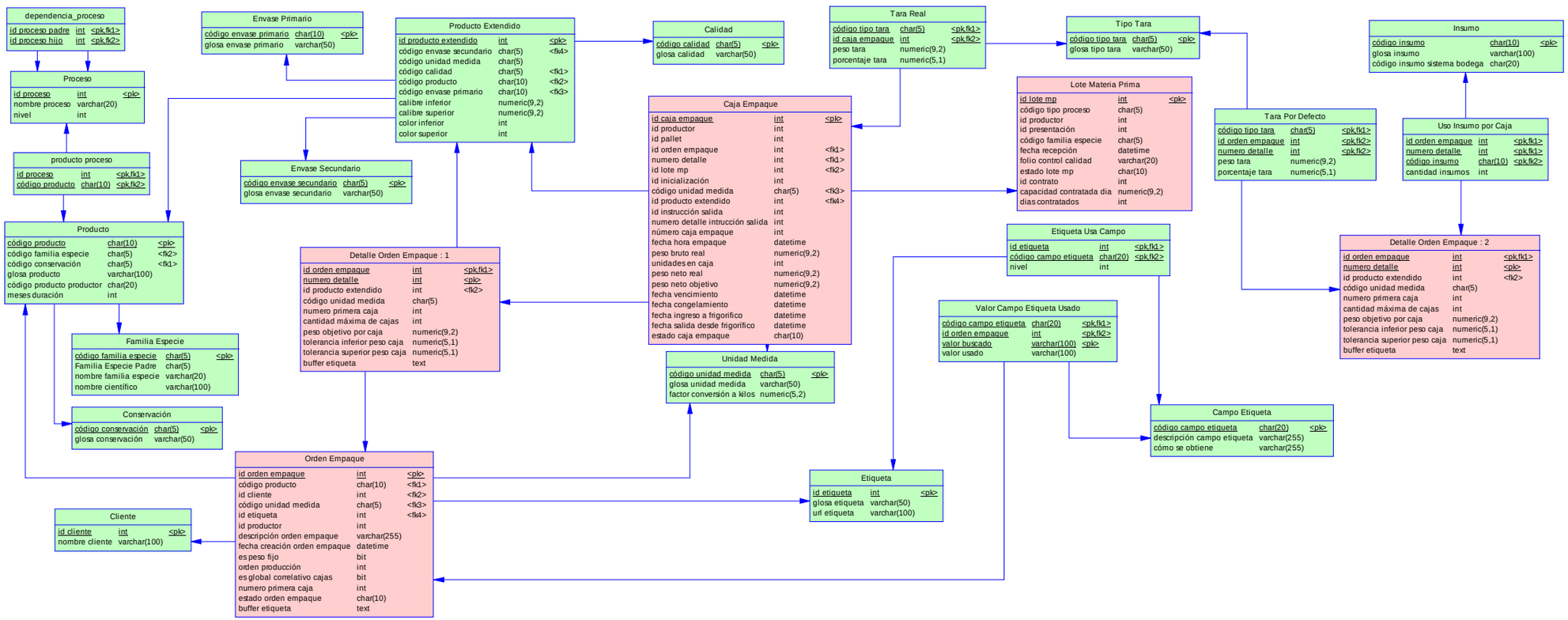


Figura 24, Modelo físico de datos módulo Empaque de Productos

6.2 Diagrama Casos de Uso para módulo de “Empaque de Productos”

A continuación se presenta los actores, diagrama y el detalle de los Casos de Uso definidos según los requerimientos establecidos para el módulo de “Empaque de Productos”.

Los Actores que se presentan son:

- **Operador de Empaque**: Es la persona que se encontrará en la línea de empaque y empacará cajas de acuerdo a las especificaciones realizadas en la orden de empaque y sus detalles.
- **Control de Producción**: Son los usuarios del departamento de control de producción, encargados de administrar la información básica del sistema.
- **Servicio Dispositivos**: Servicio instalado en el sistema operativo del servidor, encargado de realizar las recepciones de información de los diferentes dispositivos de pesaje y su posterior administración.

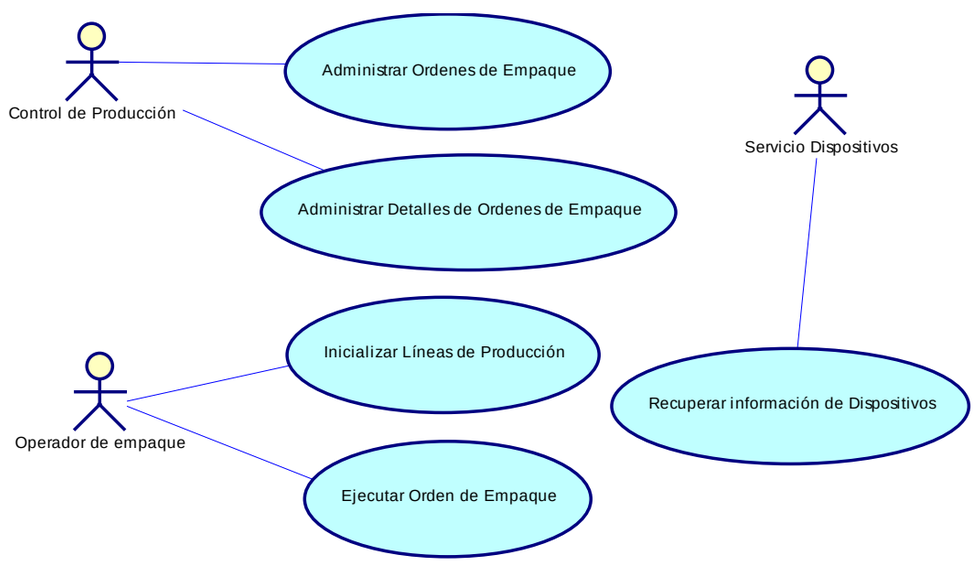


Figura 25, Diagrama de casos de uso de Módulo de “Empaque de Productos”

6.3 Plantillas Casos de Uso para “Módulo de empaque de Productos”

6.3.1 Caso de Uso “Administrar Ordenes de Empaque”

Actor: *Control de Producción*	**Sistema**
1. Mira la lista de lotes de órdenes de empaque existentes pudiendo para ello utilizar un filtro.	
2. Indica mediante un botón que desea agregar una nueva orden.	3. Despliega una pantalla, con los campos vacíos, en la cual se requiere la información básica para definir el encabezado de una orden de empaque.
4. Ingresa la información del encabezado de la nueva orden de empaque.	
5. Guarda los cambios realizados presionando un botón para ello.	6. Guarda en la base de datos la información ingresada por el actor.
7. Selecciona la opción “Campos Especiales”.	8. Despliega una pantalla, con los campos vacíos, la cual permite agregar y quitar reemplazo de valores para ciertos campos precalculados en la base de datos.
9. Guarda los cambios realizados presionando un botón para ello.	10. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	

Tabla 7. Planilla Caso de Uso: Administrar Ordenes de Empaque

6.3.2 Caso de Uso “Administrar Detalles de Orden de Empaque”

Actor: *Control de Producción*	**Sistema**
1. Selecciona la opción “Lista Detalles”.	2. Despliega una pantalla con la lista de todos los detalles definidos para la orden de empaque seleccionada en (1).
3. Indica mediante un botón que desea agregar un nuevo detalle.	4. Despliega una pantalla, con los campos vacíos, en la cual se requiere el detalle de la información para definir el nuevo detalle de la orden de empaque seleccionada en (1).
5. Guarda los cambios realizados presionando un botón para ello.	6. Guarda en la base de datos la información ingresada por el actor.
7. Selecciona la opción “Insumos” para indicar qué insumos utilizará en el empaque del detalle definido en (5).	8. Despliega una pantalla, con los campos vacíos, la cual permite agregar y quitar insumos utilizados en el detalle de la orden de empaque,

	definido en (5).
9. Guarda los cambios realizados presionando un botón para ello.	10. Guarda en la base de datos la información ingresada por el actor.
11. Puede editar la información de un detalle, de la orden de empaque definida en (1), existente mediante un clic en la fila específica mostrada en (4).	12. Despliega la pantalla de detalles de orden de empaque pero con los campos completados del detalle de la orden seleccionada.
14. Puede editar la información de un detalle de orden de empaque existente mediante un clic en la fila específica (del detalle) mostrada en (1).	15. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	

Tabla 8. Planilla Caso de Uso: Administrar Detalles de Orden de Empaque

6.3.3 Caso de Uso “Inicializar Líneas de Producción”

Actor: *Operador de Empaque*	**Sistema**
1. Selecciona del listado de órdenes de empaque una orden a ejecutar.	2. Despliega una pantalla con la información básica a recuperar para inicializar la línea de producción dirigida a la orden de empaque seleccionada en (1).
	3. Predefine en pantalla la última inicialización realizada en la línea de producción.
4. Determina si se inicializa la línea de producción con los mismos datos de la última inicialización o reparametriza la inicialización.	5. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	

Tabla 9. Planilla Caso de Uso: Inicializar Líneas de Producción

6.3.4 Caso de Uso “Ejecutar Órdenes de Empaque”

Actor: *Operador de Empaque*	**Sistema**
1. Selecciona del listado de órdenes de empaque una orden a ejecutar.	2. Inicializa la línea de producción (descrito en el caso de uso 6.3.3).
	3. Despliega una pantalla con los valores definidos en la orden de empaque y sus detalles.

4. Selecciona el detalle de la orden de empaque a ejecutar, ingresa el lote proveniente, la clave de producción, unidades que contiene la caja y la cantidad de cajas a crear.	5. Guarda en la base de datos la información ingresada por el actor.
	6. Genera la etiqueta correspondiente de la caja de empaque.
7. Repite el paso (4) hasta finalizar la creación de cajas de la orden de empaque.	
8. Finaliza la ejecución de la orden de empaque.	9. Cambia de estado a la orden de empaque.
Excepciones:	
a.- Es sistema solo creará cajas de empaque si las especies de la orden de empaque corresponde a la declarada en el lote de materia prima.	

Tabla 10. Planilla Caso de Uso: Ejecutar Ordenes de Empaque

6.3.5 Caso de Uso “Recuperar Información de Dispositivos de Pesaje”

Actor: *Servicio Dispositivo*	**Sistema**
1. Crea un hilo de proceso por cada balanza disponible en el sistema.	2. Almacena en buffer del dispositivo de pesaje la última pesada.
3. Por cada hilo de proceso verifica que la balanza se encuentre "Activa" y busca en el buffer de la balanza el último peso.	
4. Si existe algún valor en el buffer del dispositivo lo almacena en la base de datos.	5. Guarda en la base de datos la información ingresada por el actor.
6. Espera un tiempo y vuelve al punto (3).	
Excepciones:	

Tabla 11. Planilla Caso de Uso: Recuperar Información de Dispositivos de Pesaje

6.4 DIAGRAMAS DE SECUENCIAS

6.4.1 Diagrama de Secuencia “Administrar Órdenes y Detalles de Ordenes de Empaque”

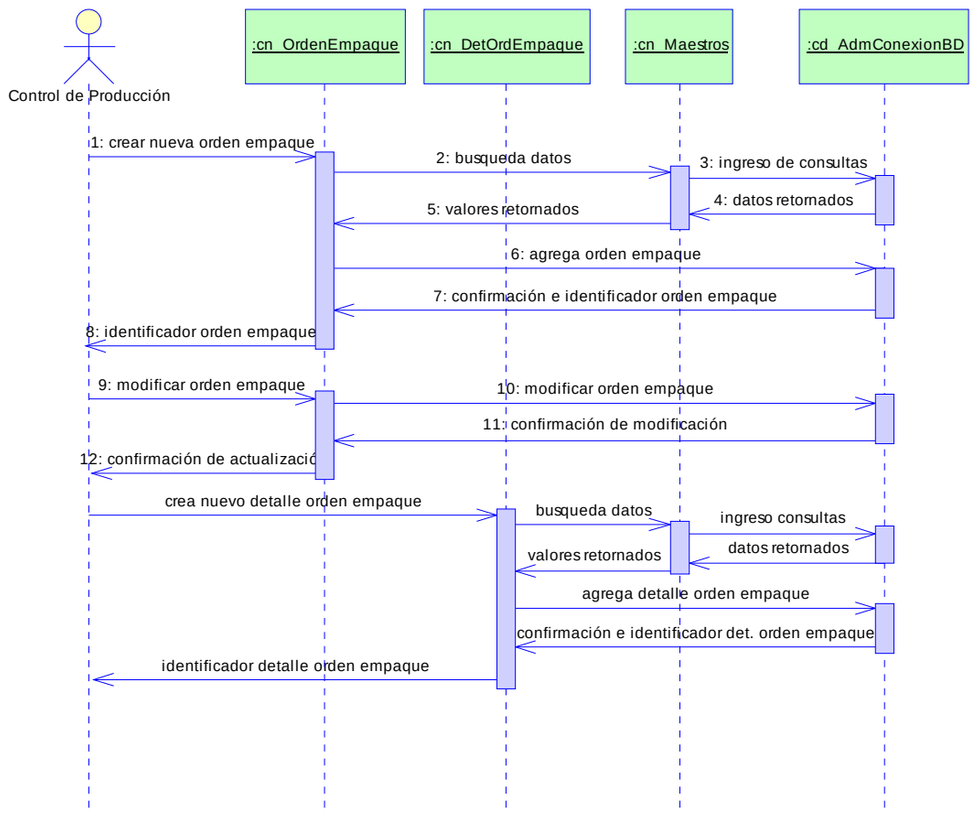


Figura 26, Diagrama de Secuencia “Administrar Órdenes y Detalles de Ordenes de Empaque”

6.4.2 Diagrama de Secuencia “Inicializar Líneas de Producción”

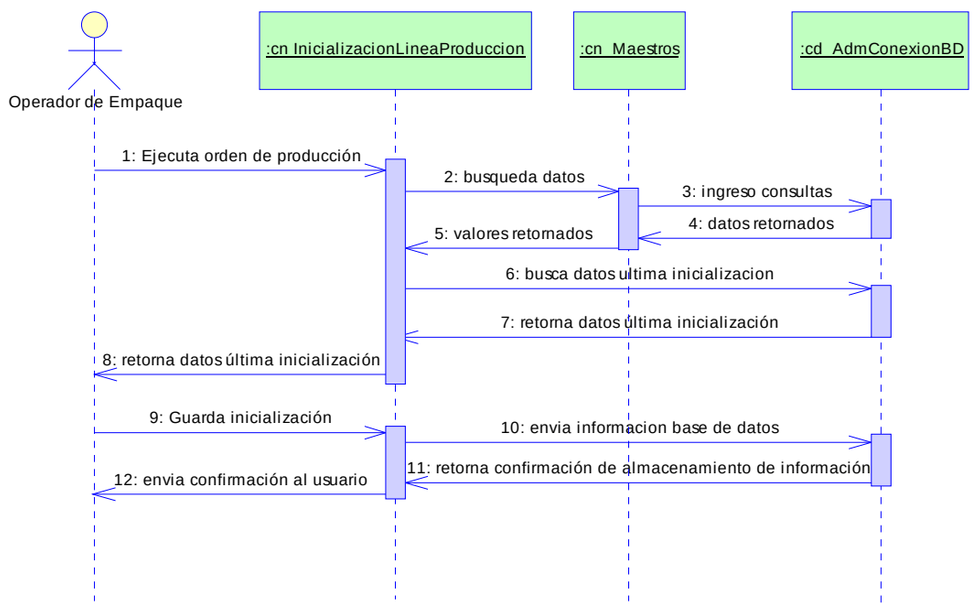


Figura 27, Diagrama de Secuencia "Inicializar Líneas de Producción"

6.4.3 Diagrama de Secuencia "Ejecutar Órdenes de Empaque"

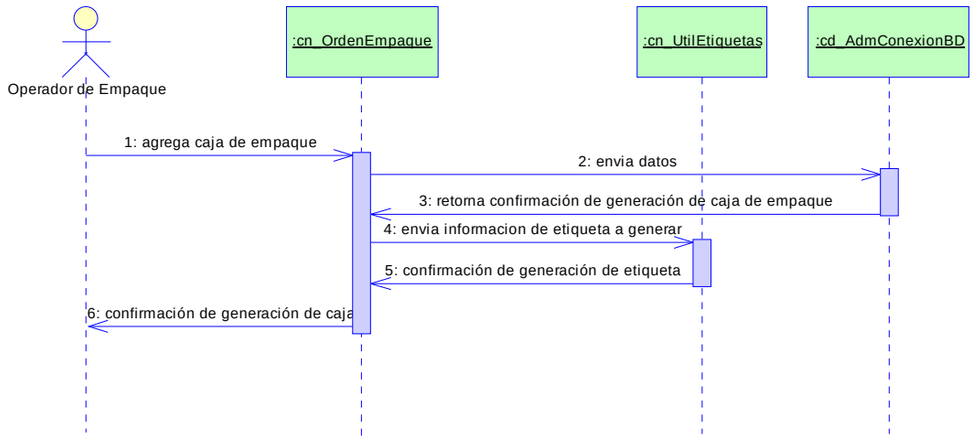


Figura 28, Diagrama de Secuencia "Ejecutar Órdenes de Empaque"

6.4.4 Diagrama de Secuencia "Recuperar Información de Dispositivos de Pesaje"

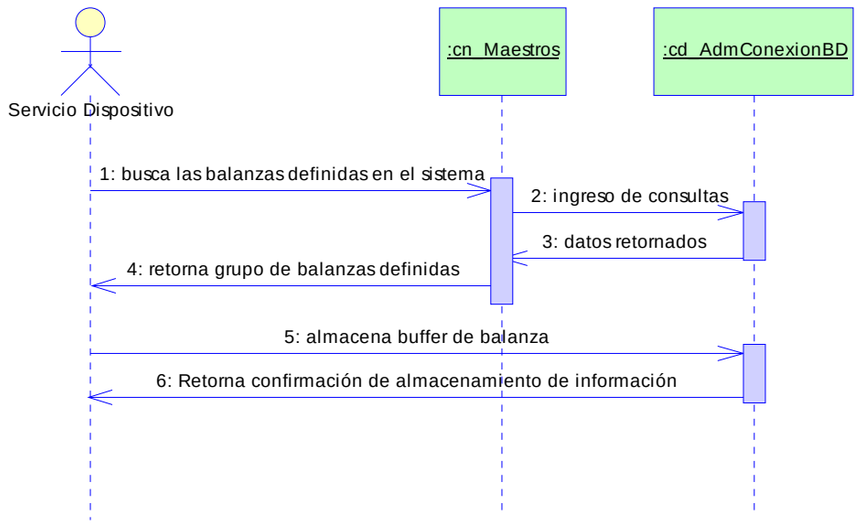


Figura 29, Diagrama de Secuencia "Recuperar Información de Dispositivos de Pesaje"

6.5 Diagramas de Colaboración

6.5.1 Diagrama de Colaboración “Administrar Órdenes y Detalles de Ordenes de Empaque”

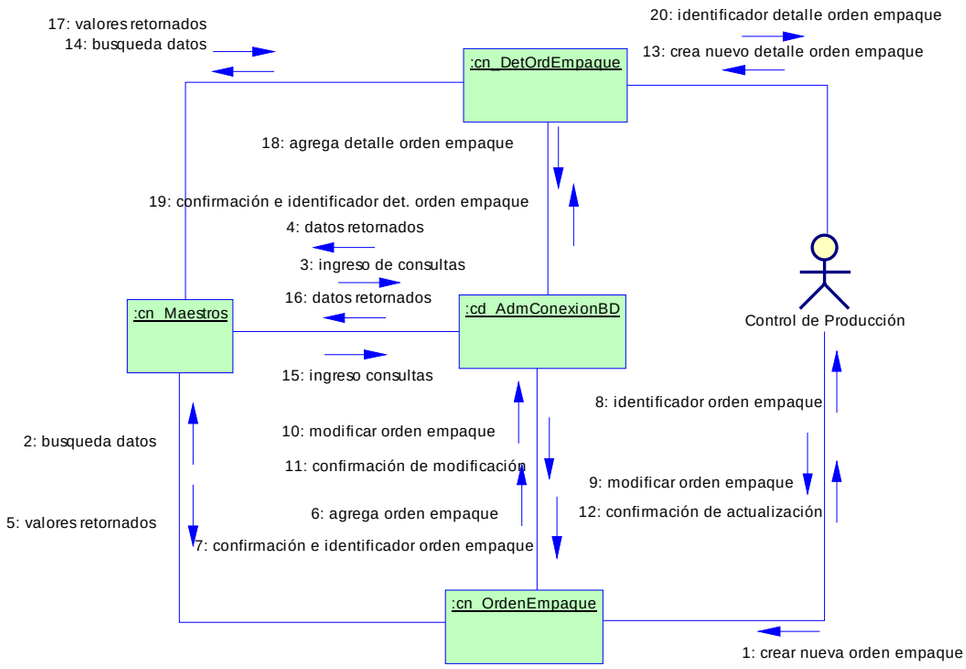


Figura 30, Diagrama de Colaboración “Administrar Órdenes y Detalles de Ordenes de Empaque”

6.5.2 Diagrama de Colaboración “Inicializar Líneas de Producción”

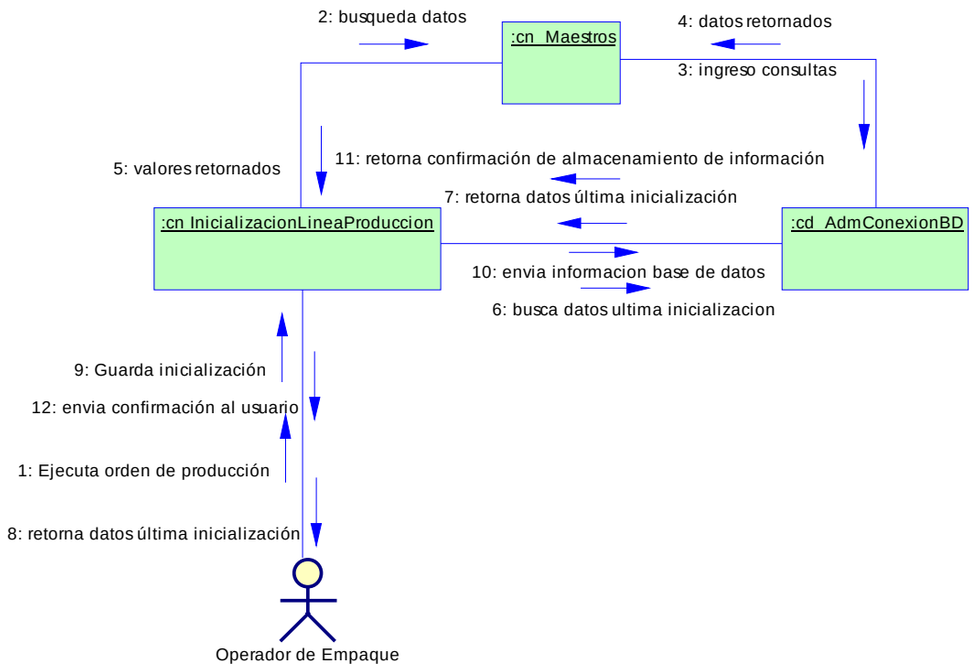


Figura 31, Diagrama de Colaboración “Inicializar Líneas de Producción”

6.5.3 Diagrama de Colaboración “Ejecutar Órdenes de Empaque”

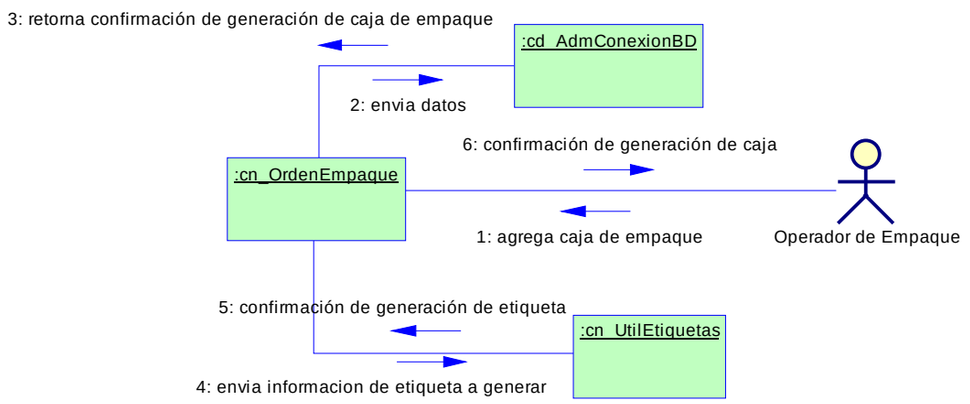


Figura 32, Diagrama de Colaboración “Ejecutar Órdenes de Empaque”

6.5.4 Diagrama de Colaboración “Recuperar Información de Dispositivos de Pesaje”

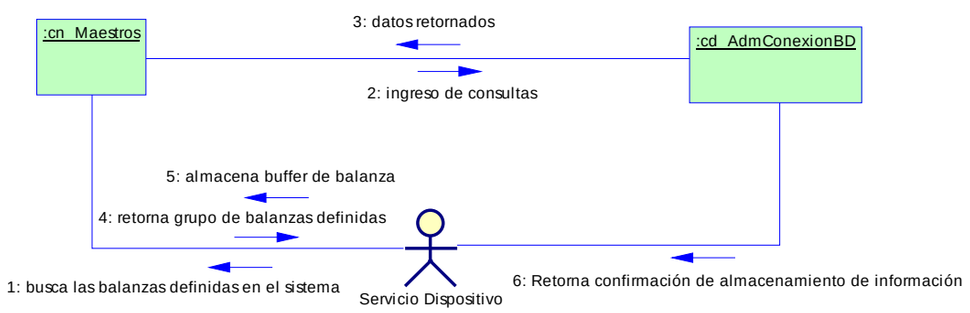


Figura 33, Diagrama de Colaboración “Recuperar Información de Dispositivos de Pesaje”

6.6 Descripción de los WebServices a implementar

Los WebServices y sus métodos a implementar para le módulo de “Empaque de Productos” serán los siguientes:

a. **cn_OrdenEmpaque**: WebServices encargado de gestionar toda la información relevante de acuerdo a las ordenes de empaque, los métodos desarrollados para este WebServices son:

   i. boListarOrdEmp: Método encargado de recuperar el listado de órdenes de empaque declarados en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su

ejecución, en su defecto retornará un “false” junto con un mensaje de error.

ii. boAgregarOrdEmp: Método encargado de insertar una nueva orden de empaque, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

iii. boActualizarOrdEmp: Método encargado de actualizar una orden de empaque, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

b. **cn_DetOrdEmp**: WebServices encargado de gestionar toda la información relevante de acuerdo a los detalles de las ordenes de empaque, los métodos desarrollados para este WebServices son:

i. boListarDetOrdEmp: Método encargado de recuperar el listado de detalles de órdenes de empaque declarados en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

ii. boAgregarDetOrdEmp: Método encargado de agregar un nuevo detalle de orden de empaque declarado en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

iii. boActualizarDetOrdEmp: Método encargado de actualizar un detalle de orden de empaque declarado en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

iv. boBorrarDerOrdEmp: Método encargado de eliminar un detalle de orden de empaque declarado en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

c. **cn_Maestros**: WebServices encargado de recuperar toda la información relevante de acuerdo a los datos básicos que el sistema necesita conocer, los métodos desarrollados para este WebServices para el módulo de empaque de productos son:

   i. boListarProductos: Método encargado de recuperar el listado de productos declarados en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

   ii. boListarPropietario: Método encargado de recuperar el listado de propietarios declarados en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

   iii. boListarCliente: Método encargado de recuperar el listado de clientes declarados en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

   iv. boListarEtiqueta: Método encargado de recuperar el listado de etiquetas declaradas en el sistema de acuerdo a un propietario específico, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

   v. boListarUnidadMedida: Método encargado de recuperar el listado de las unidades de medida declaradas en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su

ejecución, en su defecto retornará un “false” junto con un mensaje de error.

vi. boListarBalanzas: Método encargado de recuperar el listado de balanzas asociadas a las líneas de producción declaradas en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

d. **cn_InicializacionLineaProduccion:** WebServices encargado de recuperar toda la información relevante de acuerdo a las inicializaciones de líneas de producción, los métodos desarrollados para este WebServices para el módulo de empaque de productos son:

   a. boAgregarInicializacion: Método encargado de agregar una nueva inicialización de línea de producción declarada en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

   b. boListarUltimaInicializacion: Método encargado de listar la última inicialización realizada en una línea de producción declarada en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

   c. boListarAsignacionPersonas: Método encargado de listar las personas vinculadas a una inicialización declarada en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

   d. boListarLineaProduccion: Método encargado de listar las líneas de producción presentes en el sistema, si la ejecución fue exitosa

retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

e. boAgregarAsignacionPersona: Método encargado de agregar un listado de personas vinculadas a una línea de producción en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

f. boBorrarAsignacionPersona: Método encargado de eliminar un listado de personas vinculadas a una línea de producción en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

e. **cn_UtiEtiquetas**: WebServices encargado de gestionar toda la información relevante de acuerdo a las etiquetas declaras en el sistema, los métodos desarrollados para este WebServices para el módulo de empaque de productos son:

i. boImprimirEtiquetaOrdEmp: Método encargado de realizar la impresión de una etiqueta en una impresora declarada en una línea de producción, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

6.7 Principales pantallas del Módulo de Empaque de Productos

6.7.1 Pantalla “Lista Ordenes de Empaque”

Pantalla en la cual se realiza el despliegue histórico de las ordenes de empaque definidas en el sistema, posibilitando al usuario la selección de un grupo

de ellas mediante la especificación de un rango de ordenes de empaque validos, para ello la aplicación Front-end establecerá comunicación con el WebServices de "cn_OrdenEmpaque" para el intercambio de información y confirmación de transacciones realizadas.

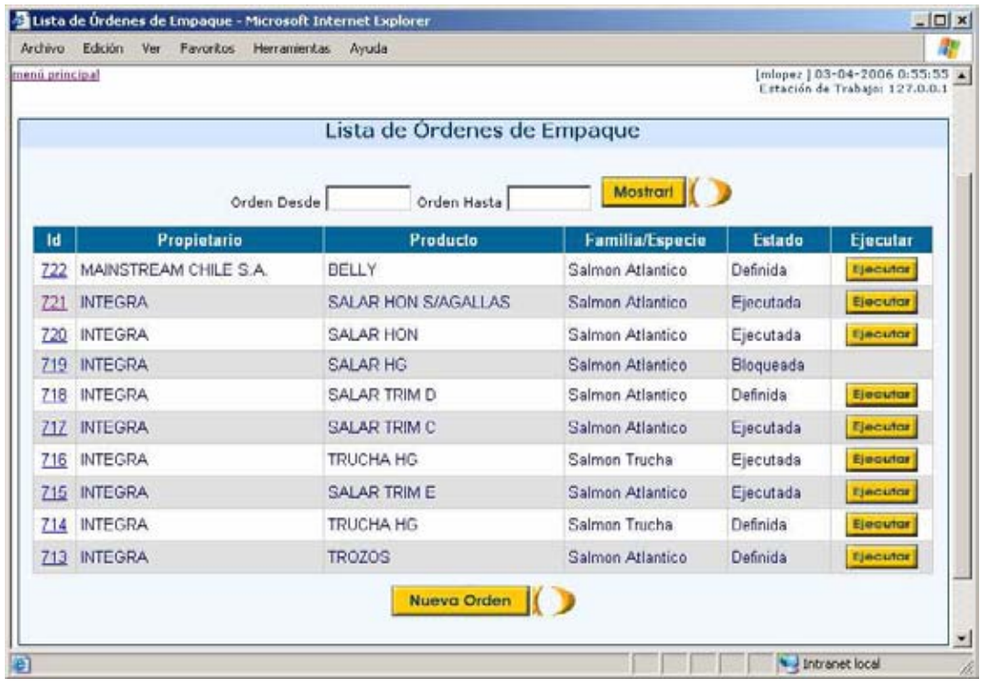


Figura 34, Pantalla "Listar Ordenes de Empaque"

6.7.2 Pantalla "Agregar y Actualizar Ordenes de Empaque"

Pantalla en la cual se realiza la inserción de nuevas ordenes de empaque para el sistema y su posterior modificación, para ello la aplicación Front-end establecerá comunicación con el WebServices de "cn_OrdenEmpaque" y "cn_Maestros" para el intercambio de información y confirmación de transacciones realizadas.

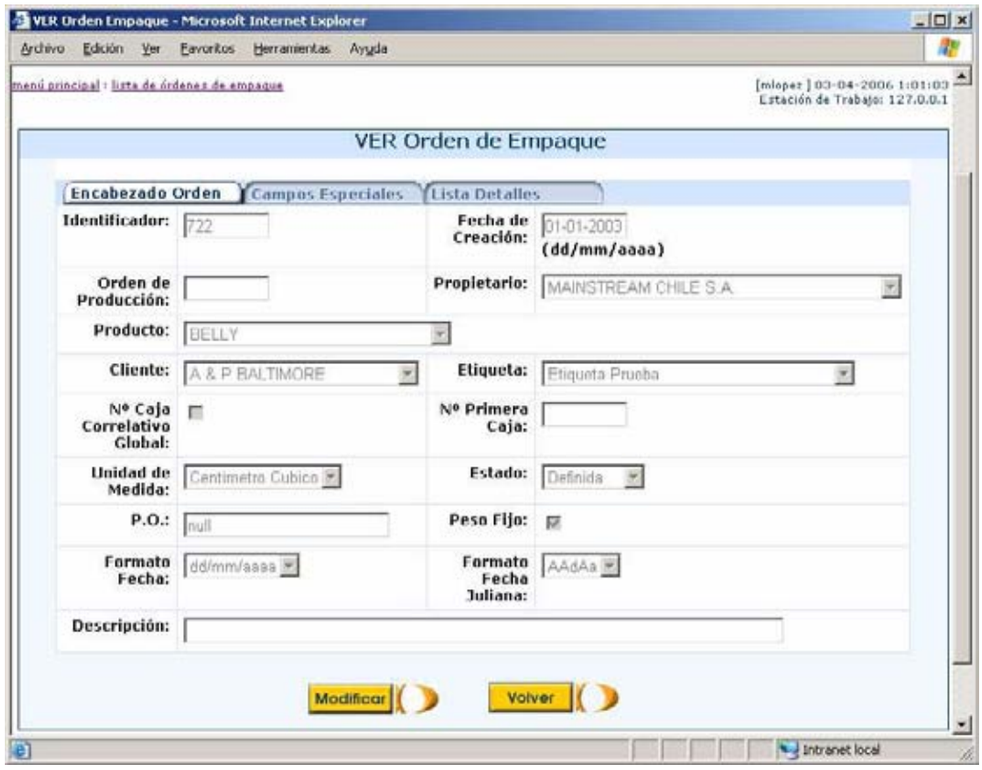


Figura 35, Pantalla "Agregar y Actualizar Ordenes de Empaque"

6.7.3 Pantalla "Agregar y Actualizar Detalles de Ordenes de Empaque"

Pantalla en la cual se realiza la inserción de nuevos detalles de ordenes de producción para el sistema y su posterior modificación, para ello la aplicación Front-end establecerá comunicación con el WebServices de "cn_DetOrdEmpaque" y "cn_Maestros" para el intercambio de información y confirmación de transacciones realizadas.

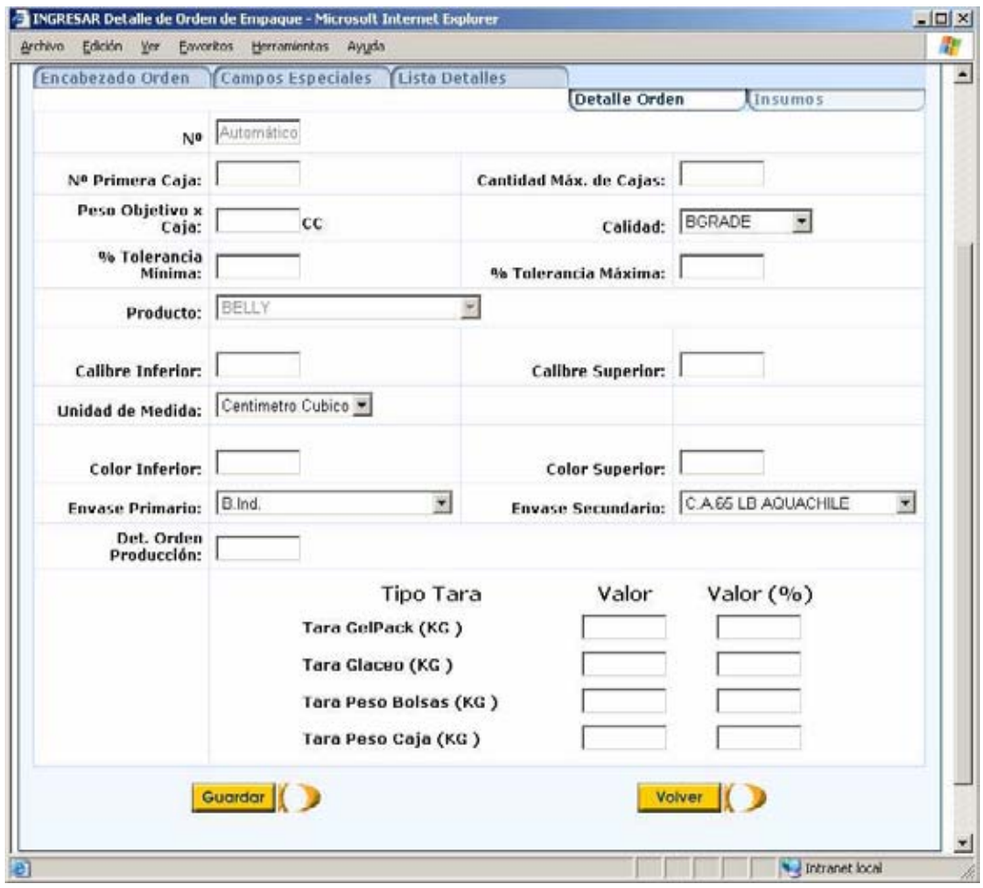


Figura 36, Pantalla "Agregar y Actualizar Detalles de Ordenes de Empaque"

6.7.4 Pantalla "Inicialización de Línea de Producción"

Pantalla en la cual se realiza la inicialización de las diferentes líneas de producción definidas en el sistema, para ello la aplicación Front-end establecerá comunicación con el WebServices de "cn_InicializacionLineaProduccion" y "cn_Maestros" para el intercambio de información y confirmación de transacciones realizadas.

Figura 37, Pantalla "Inicialización de Línea de Producción"

6.7.5 Pantalla "Ejecución Orden de Empaque"

Pantalla en la cual se realiza la ejecución de las ordenes de empaque definidas en el sistema, para ello la aplicación Front-end establecerá comunicación con el WebServices de "cn_OrdenEmpaque" y "cn_Maestros" para el intercambio de información y confirmación de transacciones realizadas.

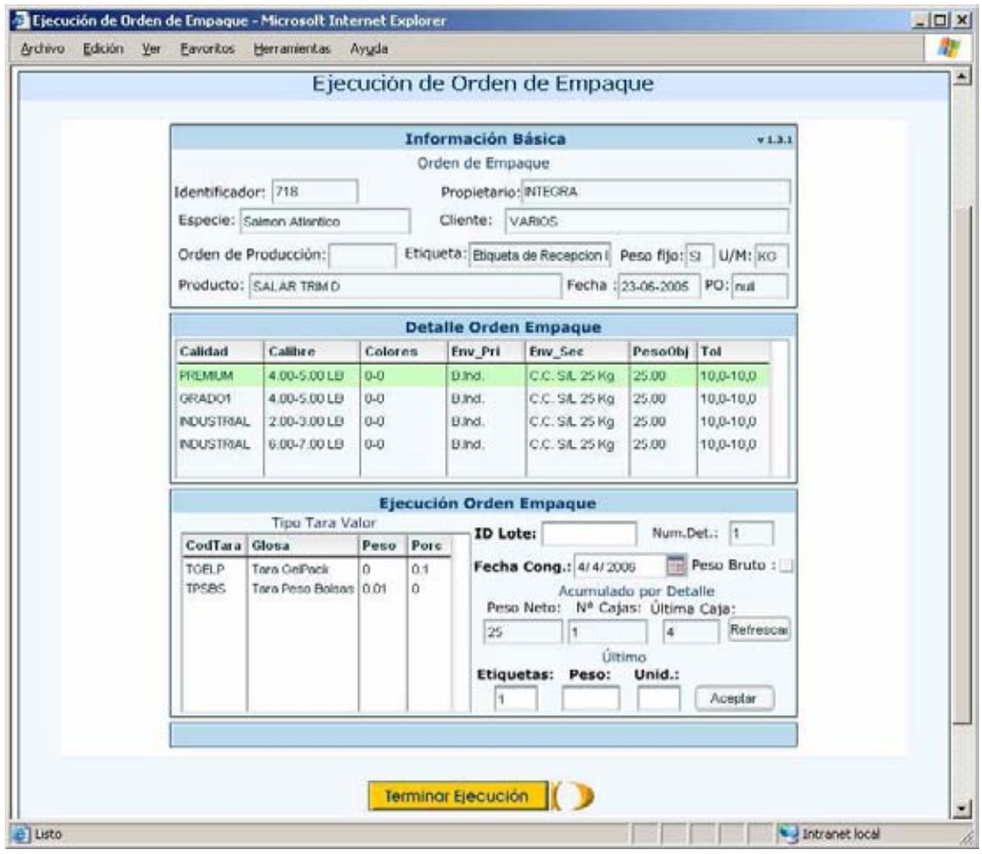


Figura 38, Pantalla "Ejecución Orden de Empaque"

Capítulo 7

7. Diseño e implementación de módulo de “Frigorífico y Despacho”

7.1 Modelo de datos vinculado con el módulo de Frigorífico y Despacho

Como parte del diseño del módulo, se debe analizar el modelo de datos a utilizar, con el fin de describir los datos a almacenar y el flujo que habrá entre ellos.

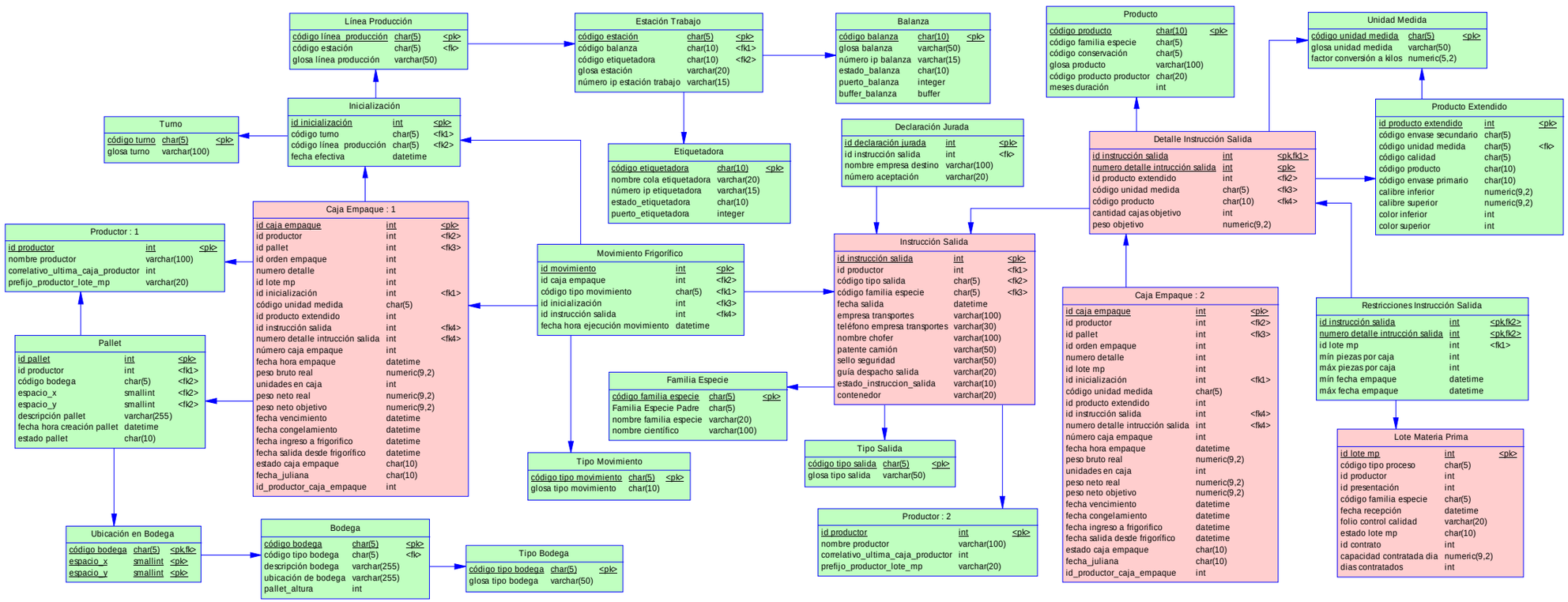


Figura 39, Pantalla "Modelo Físico Módulo Frigorífico y Despacho"

7.2 Diagrama Casos de Uso para módulo de “Frigorífico y Despacho”

A continuación se presenta los actores, diagrama y el detalle de los Casos de Uso definidos según los requerimientos establecidos para el módulo de “Frigorífico y Despacho”.

Los Actores que se presentan son:

- **Operador de Frigorífico**: Es la persona que se encontrará en la línea de ingreso y salida de frigorífico, creará los pallets y administrará su contenido dentro de las distintas bodegas que se declaren en el sistema..
- **Control de Producción**: Son los usuarios del departamento de control de producción encargados de administrar la información básica del sistema.
- **Servicio Dispositivos**: Servicio instalado en el sistema operativo del servidor, encargado de realizar las recepciones de información de los diferentes dispositivos de pesaje y su posterior administración.

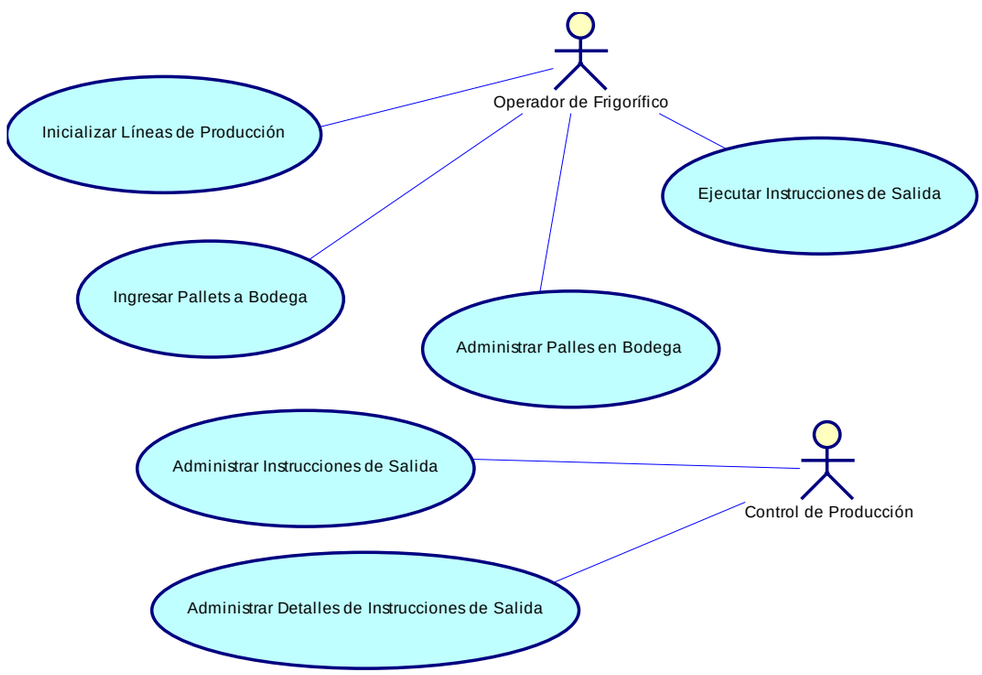


Figura 40, Diagrama Casos de Uso: Módulo de Frigorífico y Despacho.

7.3 Plantillas Casos de Uso para “Módulo de Frigorífico y Despacho”

7.3.1 Caso de Uso “Ingresar Pallets a Bodega”

Actor: *Operador de Frigorífico*	**Sistema**
1. Agrupa un conjunto de cajas que comparten características básicas de creación y define una bodega para su posicionamiento final.	
2. Ingresa el identificador de caja.	3. Si es primera caja recupera la información relevante para la creación del pallet.
4. Ingresa el resto de las cajas que conformarán al pallet.	5. De acuerdo a la información recuperada en el punto (3), valida la inserción de las restantes cajas.
6. Finaliza la creación de cajas, presionando un botón para ello.	7. Crea el Pallet y propone una ubicación del pallet de acuerdo a la bodega seleccionada por el actor.
8. Acepta o modifica posicionamiento propuesto por el sistema.	9. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	
El sistema solo admitirá a cajas de empaque que concuerden entre si de acuerdo a un conjunto de características.	

Tabla 12. Planilla Caso de Uso: Ingresar Pallets a Bodega

7.3.2 Caso de Uso “Administrar Pallets en Bodega”

Actor: *Operador de Frigorífico*	**Sistema**
1. Selecciona el o los pallets que necesiten ser mantenidos.	2. Cambia estado de el o los pallets.
3. Selecciona un pallet del listado de pallets en mantención y realiza las modificaciones necesarias para el pallet.	4. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	
a.- Es sistema solo permitirá la mantención de palles que se encuentren activos en el sistema.	

Tabla 13. Planilla Caso de Uso: Administrar Pallets en Bodega

7.3.3 Caso de Uso “Administrar Instrucciones de Salida”

Actor: *Control de Producción*	**Sistema**
1. Mira la lista de instrucciones de salida existentes, posibilitado para ello utilizar un filtro.	
2. Indica mediante un botón que desea agregar una instrucción de salida.	3. Despliega una pantalla, con los campos vacíos, en la cual se requiere la información básica para definir el encabezado de una instrucción de salida.
4. Ingresa la información del encabezado de la instrucción de salida.	
5. Guarda los cambios realizados presionando un botón para ello.	6. Guarda en la base de datos la información ingresada por el actor.
7. Guarda los cambios realizados presionando un botón para ello.	8. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	

Tabla 14. Planilla Caso de Uso: Administrar Instrucciones de Salida

7.3.4 Caso de Uso “Administrar Detalles de Instrucciones de Salida”

Actor: *Control de Producción*	**Sistema**
1. Selecciona la opción “Lista Detalles”.	2. Despliega una pantalla con la lista de todos los detalles definidos para la instrucción de salida seleccionada en (1).
3. Indica mediante un botón que desea agregar un nuevo detalle.	4. Despliega una pantalla, con los campos vacíos, en la cual se requiere el detalle de la información para definir el nuevo detalle de la instrucción de salida seleccionada en (1).
5. Guarda los cambios realizados presionando un botón para ello.	6. Guarda en la base de datos la información ingresada por el actor.
7. Puede editar la información de un detalle de la instrucción de salida definida en (1), existente mediante un clic en la fila específica mostrada en (4).	8. Despliega la pantalla de detalles de instrucción de salida pero con los campos completados con los datos del detalle de la instrucción seleccionada.
9. Realiza las modificaciones necesarias y confirma presionando un botón para ello.	10. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	

Tabla 15. Planilla Caso de Uso: Administrar Detalles de Instrucciones de Salida

7.3.5 Caso de Uso “Ejecutar Instrucciones de Salida”

Actor: *Operador de Frigorífico*	**Sistema**
1. Selecciona del listado de instrucciones de salida una instrucción a ejecutar.	2. Inicializa la línea de producción (descrito en el caso de uso 6.3.3).
	3. Despliega una pantalla con los valores definidos en la instrucción de salida.
4. Selecciona la unidad a despachar (pallet o caja) e ingresa identificador de la unidad.	5. Verifica que la unidad seleccionada por el actor cumpla con las condiciones necesarias para ser despachada.
6. Presiona botón para indicar el término de la ejecución de la instrucción de salida.	7. Guarda en la base de datos la información ingresada por el actor.
Excepciones:	
El sistema solo permitirá la salida de cajas o pallets del sistema que cumplan con los requisitos definidos en la instrucción de salida.	

Tabla 16. Planilla Caso de Uso: Ejecutar Instrucciones de Salida

7.4 DIAGRAMAS DE SECUENCIAS

7.4.1 Diagrama de Secuencia “Ingresar Pallets a Bodega”

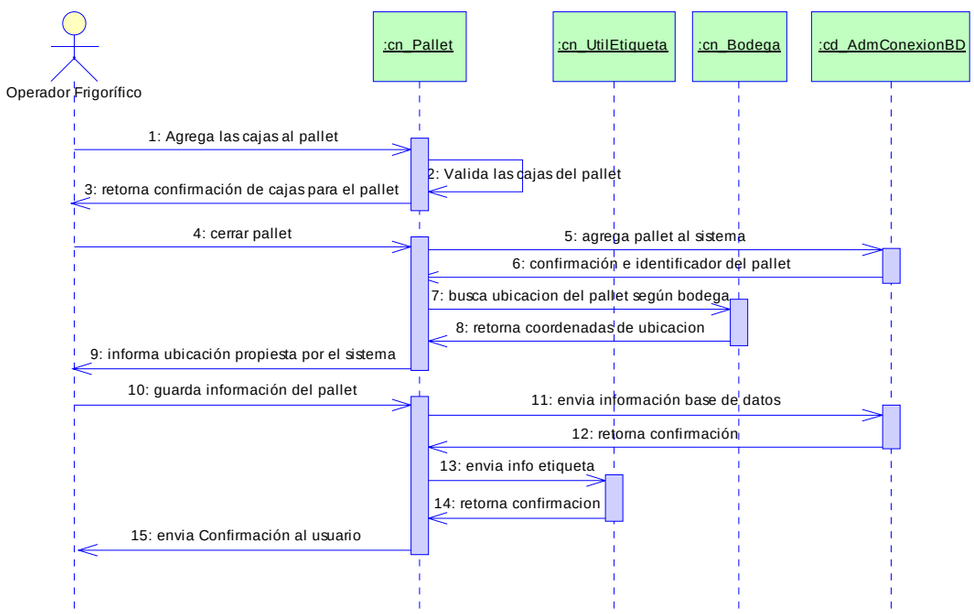


Figura 41, Diagrama de Secuencia “Ingresar Pallets a Bodega”

7.4.2 Diagrama de Secuencia “Administrar Pallets en Bodega”

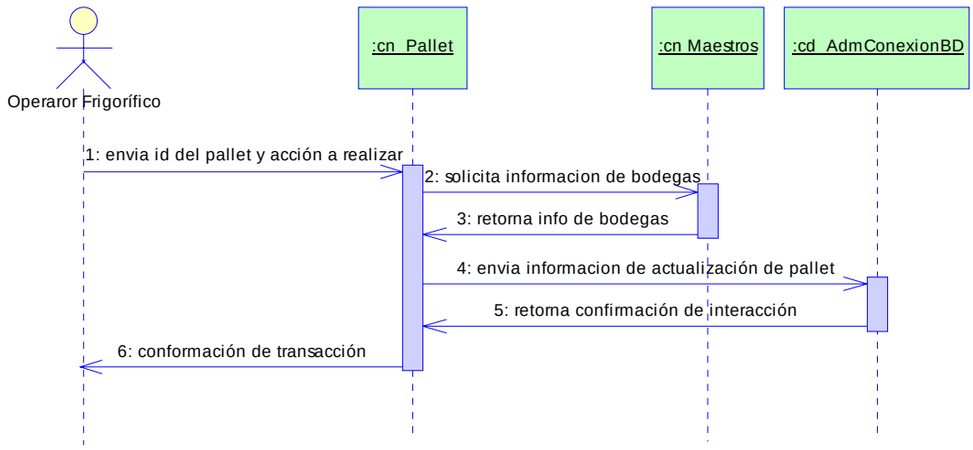


Figura 42, Diagrama de Secuencia “Administrar Pallets en Bodega”

7.4.3 Diagrama de Secuencia “Administrar Instrucciones y Detalles de Instrucciones de Salida”

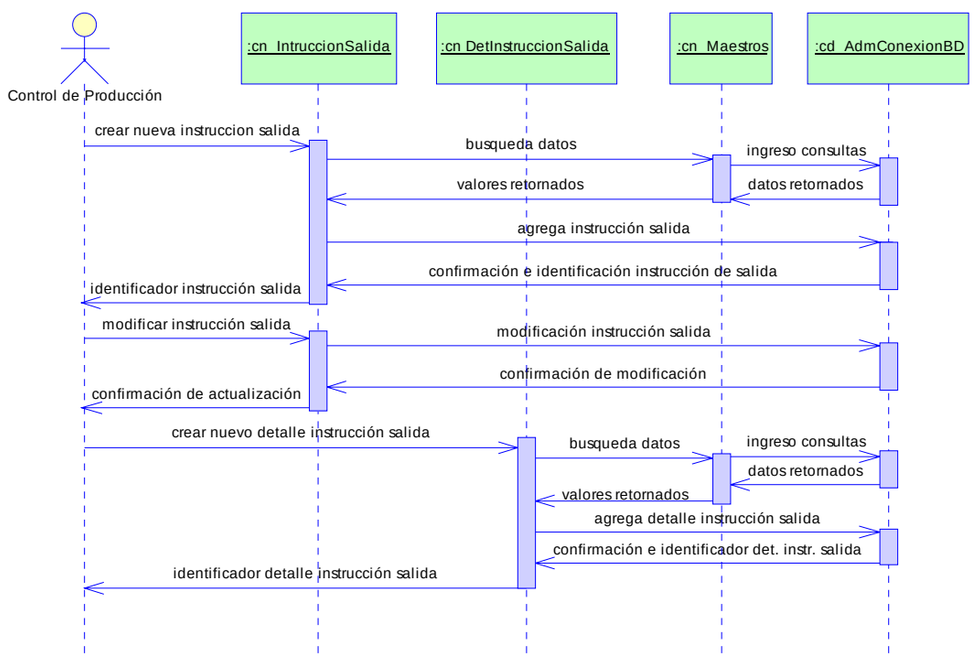


Figura 43, Diagrama de Secuencia “Administrar Instrucciones y Detalles de Instrucciones de Salida”

7.4.4 Diagrama de Secuencia “Ejecutar Instrucciones de Salida”

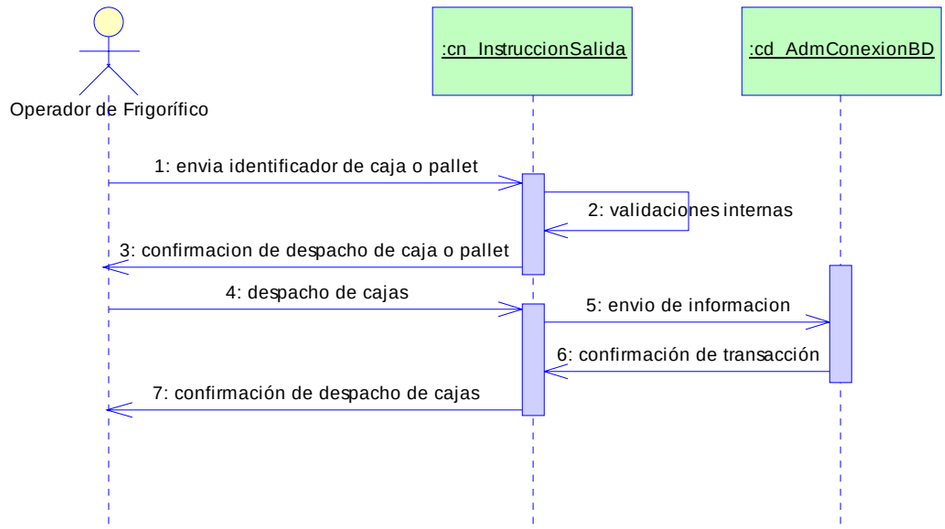


Figura 44, Diagrama de Secuencia “Ejecutar Instrucciones de Salida”

7.5 DIAGRAMAS DE COLABORACIÓN

7.5.1 Diagrama de Colaboración “Ingresar Pallets a Bodega”

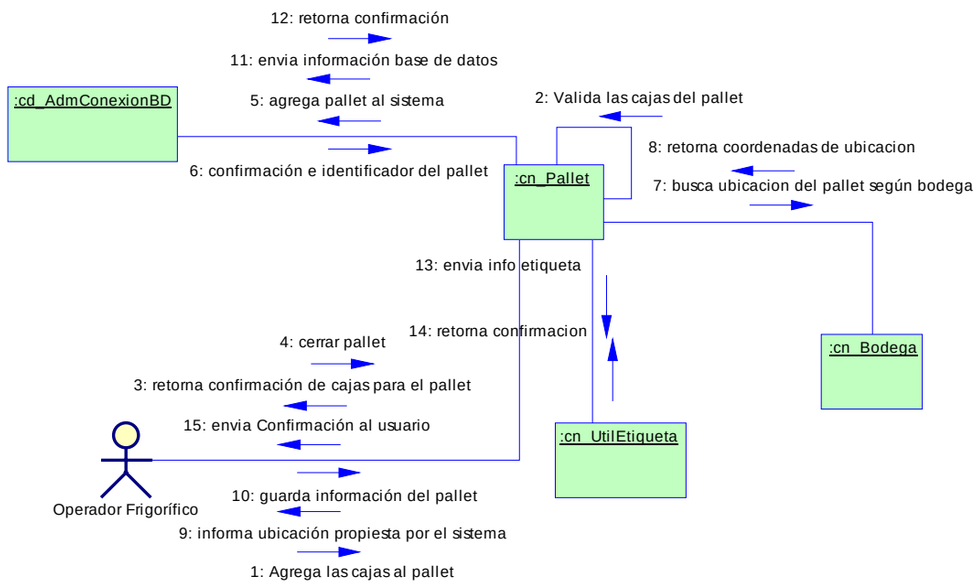


Figura 45, Diagrama de Colaboración “Ingresar Pallets a Bodega”

7.5.2 Diagrama de Colaboración “Administrar Pallets en Bodega”

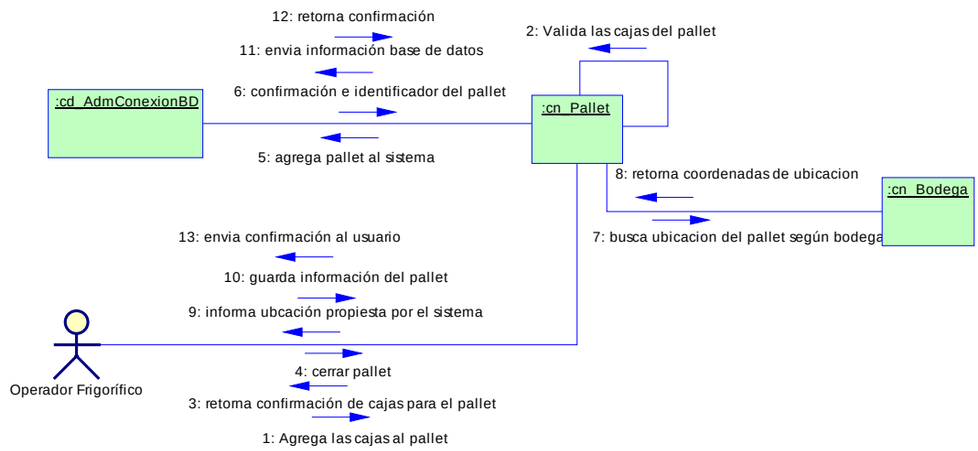


Figura 46, Diagrama de Colaboración “Administrar Pallets en Bodega”

7.5.3 Diagrama de Colaboración “Administrar Instrucciones y Detalles de Instrucciones de Salida”

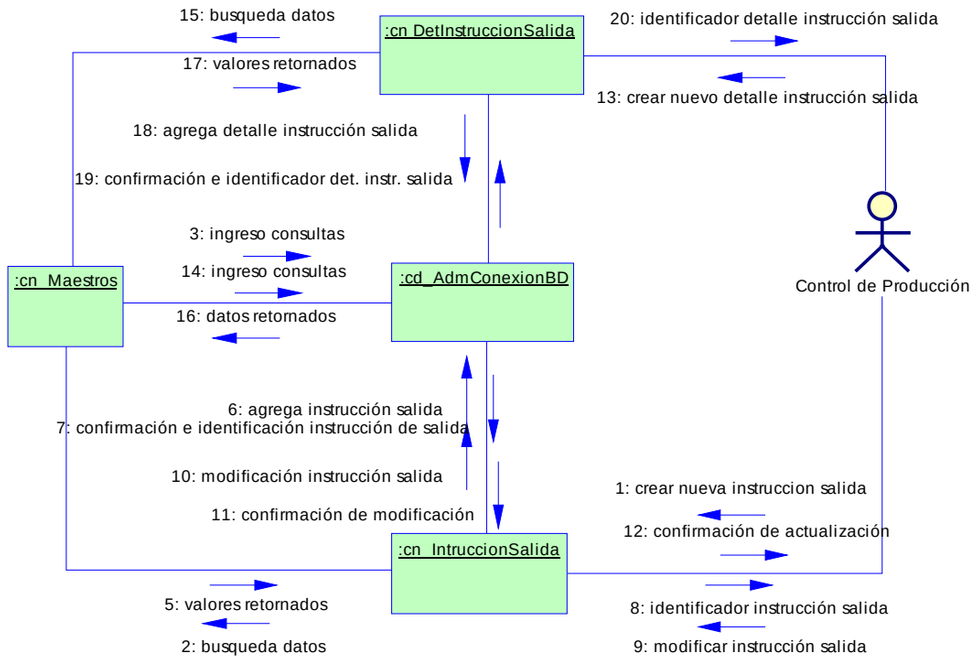


Figura 47, Diagrama de Colaboración “Administrar Instrucciones y Detalles de Instrucciones de Salida”

7.5.4 Diagrama de Colaboración “Ejecutar Instrucciones de Salida”

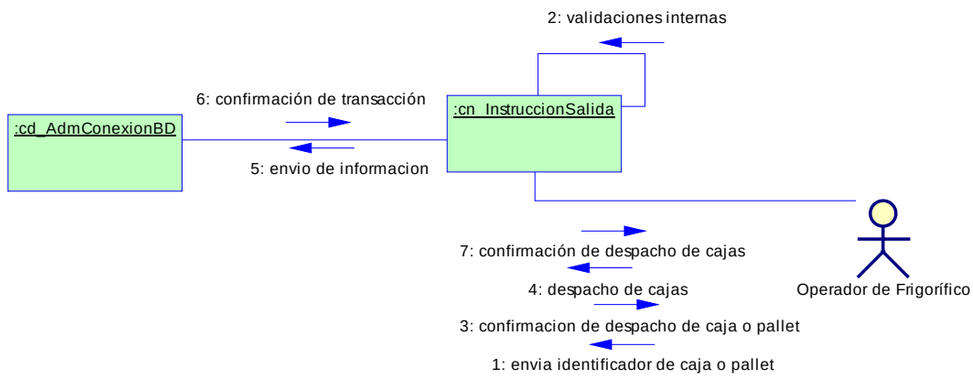


Figura 48, Diagrama de Colaboración “Ejecutar Instrucciones de Salida”

7.6 Descripción de los WebServices a implementar

Los WebServices y sus métodos a implementar para le módulo de “Frigorífico y Despacho” serán los siguientes:

a. **cn_Pallet:** WebServices encargado de recuperar y gestionar toda la información relevante de acuerdo a los pallets que se definen en el sistema, los métodos desarrollados para este WebServices para el módulo de frigorífico y despacho son:

   i. boListarPallet: Método encargado de retornar un pallet o conjunto de estos de acuerdo a una serie de parámetros que son enviados, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

   ii. boAgregarPallet: Método encargado de realizar la inserción de un nuevo pallet al sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

   iii. boListarCajasPallet: Método encargado de retornar todas las cajas pertenecientes a un pallet específico, si la ejecución fue exitosa

retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

iv. boAgregarCajasPallet: Método encargado de insertar un conjunto de cajas de empaque a un pallet, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error

v. boActualizarPallet: Método encargado de actualizar los datos vinculados a un pallet, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error

vi. boAgregarCajasPalletMantencion: Método encargado de agregar cajas de empaque a un pallet que se encuentre en mantención, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error

vii. boMoverCajasPalletMantencion: Método encargado de mover cajas de empaque entre pallets que se encuentren en estado de mantención, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error

viii. boMoverPalletUbicacion: Método encargado de trasladar la ubicación de un pallet que se encuentra en estado de manteción, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error

ix. boFusionarPallet: Método encargado de fusionar pallets que se encuentren en estado de mantención, si la ejecución fue exitosa

retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error

b. **cn_UtiEtiqueta:** WebServices encargado de gestionar toda la información relevante de acuerdo a las etiquetas declaras en el sistema, los métodos desarrollados para este WebServices para el módulo de empaque de productos son:

    i. boImprimirEtiquetaPallet: Método encargado de realizar la impresión de una etiqueta en una impresora declarada en una línea de producción, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

c. **cn_Bodega**: WebServices encargado de recuperar y gestionar toda la información relevante a las bodegas o frigoríficos definidos en el sistema, los métodos desarrollados para este WebServices para el módulo de frigorífico y despacho son

    i. boListarContenidoBodega: Método encargado de retornar la matriz de ocupación de una bodega, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

    ii. boPesoDelEspacio: Método encargado de calcular la factibilidad de agregar un pallet en un espacio específico de acuerdo a un conjunto de reglas especificadas en el punto (3.7.2), si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

    iii. boEspacioOptimoPallet: Método encargado de retornar una ubicación tentativa para un pallet dentro de una bodega o frigorífico, si la ejecución fue exitosa retornará un booleano de valor “true”

confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

d. **cn_Maestros**: WebServices encargado de recuperar toda la información relevante de acuerdo a los datos básicos que el sistema necesita conocer, los métodos desarrollados para este WebServices para el módulo de empaque de productos son:

    i. boListarPais: Método encargado de retornar un listado de los paises definidos en el sistema, estos datos son utilizados para la generación de las instrucciones de salida, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

    ii. boTipoSalida: Método encargado de retornar un listado de los tipos de salidas definidos en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

e. **cn_InstruccionSalida**: WebServices encargado de recuperar y gestionar toda la información relevante de acuerdo a las instrucciones de salida que se definen en el sistema, los métodos desarrollados para este WebServices para el módulo de frigorífico y despacho son:

    i. boListarInstruccionSalida: Método encargado de retornar un listado de instrucciones de salidas definidas en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

    ii. boAgregarInstruccionSalida: Método encargado de agregar una nueva instrucción de salida en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

iii. boActualizarInstruccionSalidaTrans: Método encargado de actualizar los datos vinculados al transporte de una instrucción de salida, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

iv. boListarInstruccionSalidaAcum: Método encargado de retornar el listado de cajas aculumadas que se encuentran despachadas para una instrucción de salida, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

v. boEjecInstruccionSalidaPallet: Método encargado de ejecutar una instrucción de salida en el sistema de acuerdo a un pallet específico, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

vi. boEjecInstruccionSalidaCaja: Método encargado de ejecutar una instrucción de salida en el sistema mediante el despacho de una caja individual, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

vii. boCerrarInstruccionSalida: Método encargado de confirmar el despacho de las cajas de empaque para una instrucción de salida, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

viii. boListarCajasEmpPreDespacho: Método encargado de retornar un listado de cajas de empaque que se han despachado en el sistema pero no se ha confirmado su despacho, si la ejecución fue exitosa

retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

ix. boDevolverCajasEmpInstruccionSalida: si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

f. **cn_DetInstruccionSalida**: WebServices encargado de recuperar y gestionar toda la información relevante de acuerdo a los detalles de instrucciones de salidas que se definen en el sistema, los métodos desarrollados para este WebServices para el módulo de frigorífico y despacho son

i. boListarDetInstruccionSalida: Método encargado de retornar un listado de detalles de instrucciones de salidas definidas en el sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

ii. boAgregarDetInstruccionSalida: Método encargado de agregar un nuevo detalle de instrucción de salida al sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

iii. boActualizarDetInstruccionSalida: Método encargado de actualizar un detalle de instrucción de salida del sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su ejecución, en su defecto retornará un “false” junto con un mensaje de error.

iv. boBorrarDetInstruccionSalida: Método encargado de eliminar un detalle de instrucción de salida del sistema, si la ejecución fue exitosa retornará un booleano de valor “true” confirmando su

ejecución, en su defecto retornará un “false” junto con un mensaje de error.

7.7 Principales pantallas del Módulo de Frigorífico y Despacho

7.7.1 Pantalla “Lista de Ingresos a Frigorífico”

Pantalla en la cual se realiza el despliegue histórico de los pallets definidos en el sistema, posibilitando al usuario la selección de un grupo de ellas mediante la especificación de un conjunto de parámetros válidos, para ello la aplicación Front-end establecerá comunicación con el WebServices de “cn_Pallet” para el intercambio de información y confirmación de transacciones realizadas.

Figura 49, Pantalla “Lista de ingreso a frigorífico”

7.7.2 Pantalla “Armado de Nuevo Pallet”

Pantalla en la cual se realiza el armado y ubicación de un pallet del sistema, para ello la aplicación Front-end establecerá comunicación los WebServices de “cn_Pallet” para la creación del pallet, “cn_Bodega” para gestionar el posicionamiento del pallet y “cn_UtilEtiqueta” para la generación de la etiqueta del pallet.

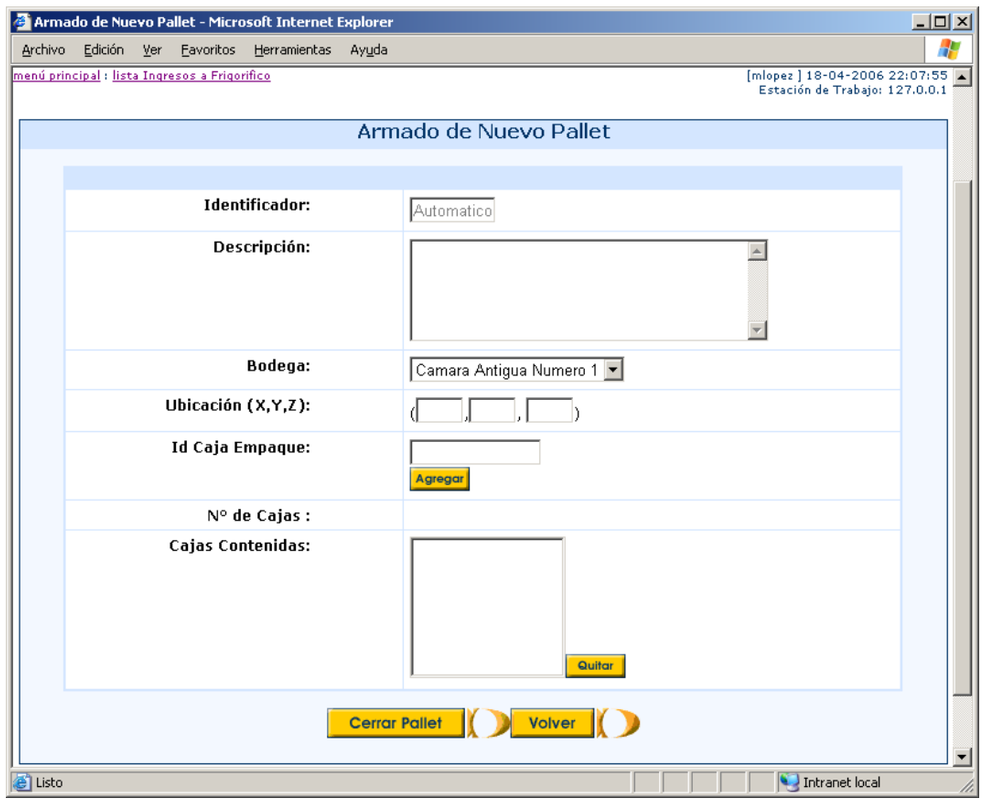


Figura 50, Pantalla "Armado de Nuevo Pallet"

7.7.3 Pantalla "Lista de Pallets en Mantención"

Pantalla en la cual se realiza el despliegue histórico de los pallets definidos en mantención en el sistema, posibilitando al usuario la selección de un pallet mediante el ingreso del identificador del pallet, para ello la aplicación Front-end establecerá comunicación con el WebServices de "cn_Pallet" para el intercambio de información y confirmación de transacciones realizadas.

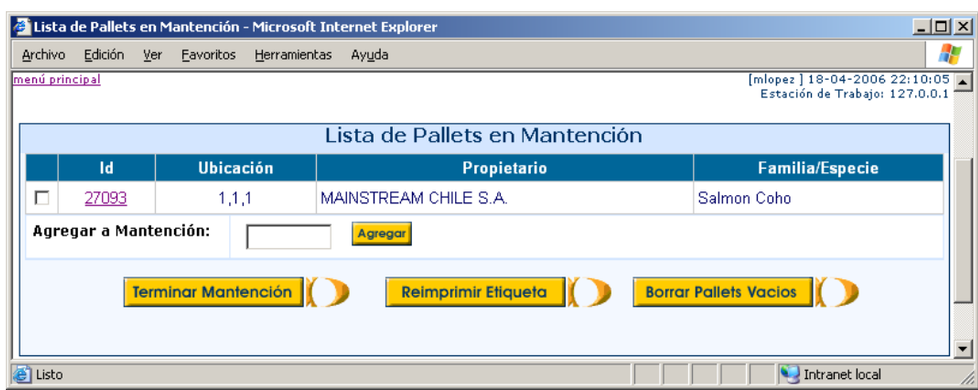


Figura 51, Pantalla "Lista de Pallets en Mantención"

7.7.4 Pantalla “Mover Cajas de Pallet en Mantención”

Pantalla en la cual se realiza el movimiento de cajas pertenecientes a pallets en mantención, para ello la aplicación Front-end establecerá comunicación con el WebServices de “cn_Pallet” para el intercambio de información y confirmación de transacciones realizadas.

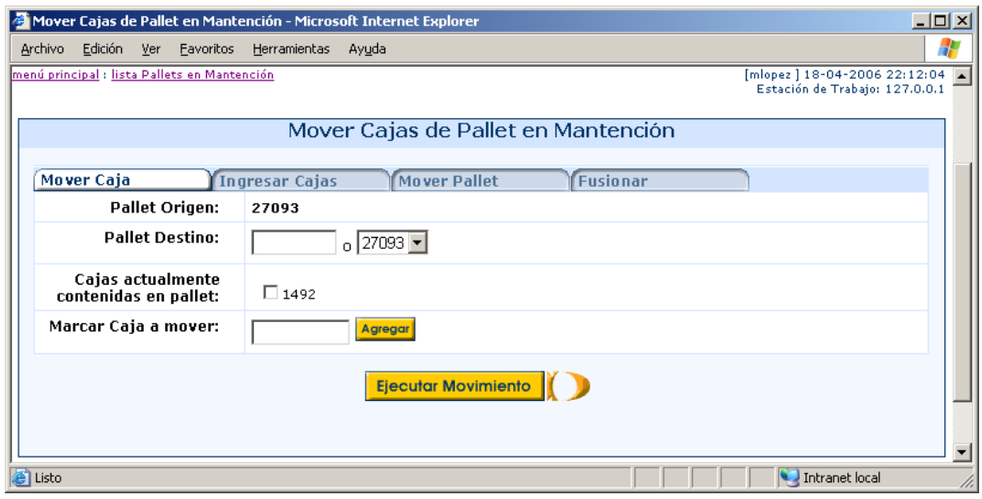


Figura 52, Pantalla “Mover Cajas de Pallet en Mantención”

7.7.5 Pantalla “Ingresar Cajas Pallet en Mantención”

Pantalla en la cual se realiza la inserción de cajas empacadas aún no ingresadas a frigorífico a un pallet que se encuentra en estado de mantención, para ello la aplicación Front-end establecerá comunicación con el WebServices de “cn_Pallet” para el intercambio de información y confirmación de transacciones realizadas.

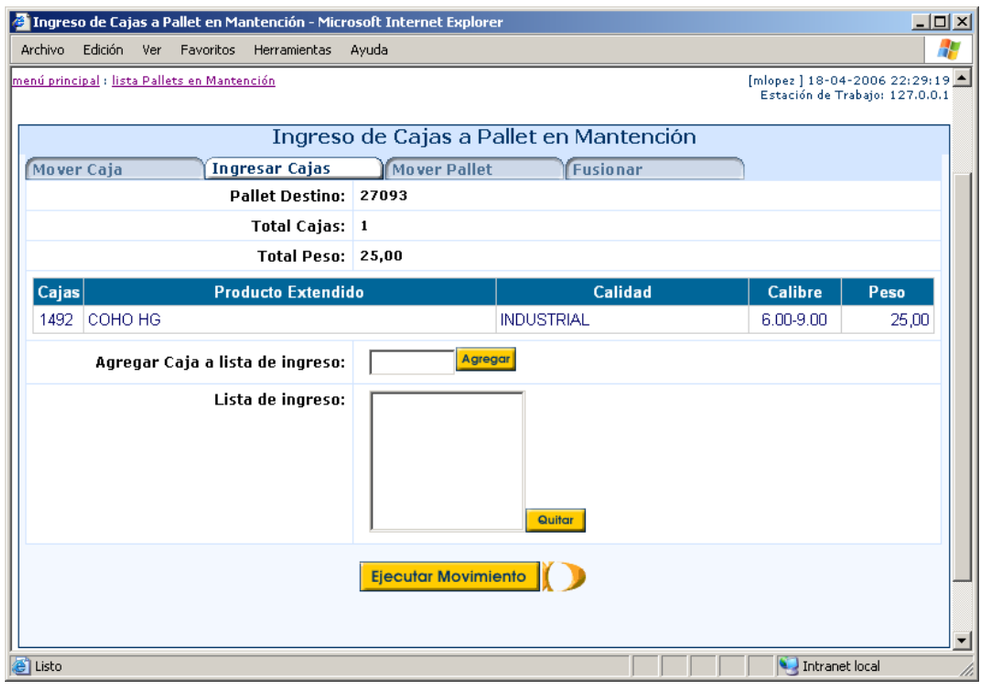


Figura 53, Pantalla "Ingresar Cajas Pallet en Mantención"

7.7.6 Pantalla "Mover Pallet en Mantención"

Pantalla en la cual se realiza el desplazamiento del pallet a una nueva ubicación de acuerdo a la bodega seleccionada, para ello la aplicación Front-end establecerá comunicación con el WebServices de "cn_Pallet" para el intercambio de información y confirmación de transacciones realizadas.

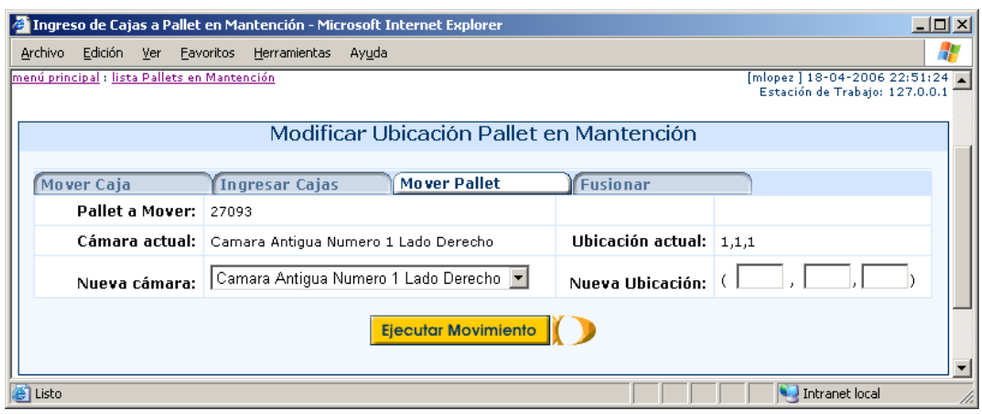


Figura 54, Pantalla "Mover Pallet en Mantención"

7.7.7 Pantalla “Fusionar Pallet en Mantención”

Pantalla en la cual se realiza la fusión de dos pallet que se encuentran en estado de mantención, para ello la aplicación Front-end establecerá comunicación con el WebServices de “cn_Pallet” para el intercambio de información y confirmación de transacciones realizadas.

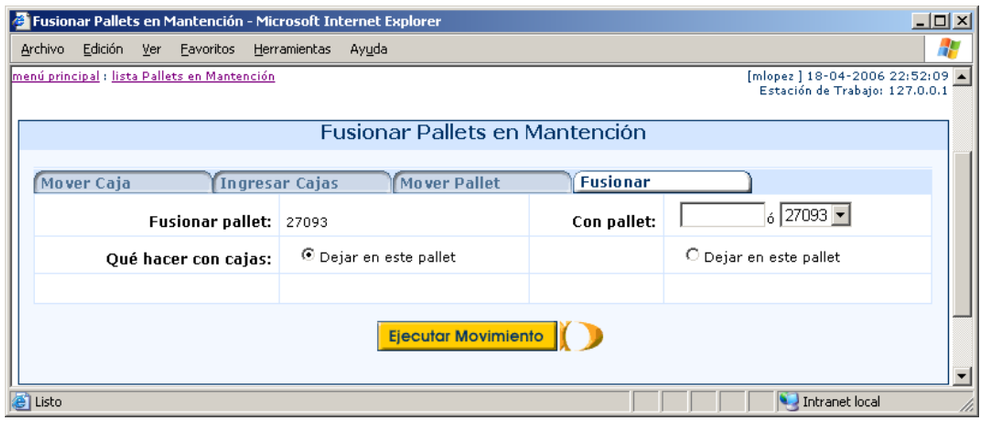


Figura 55, Pantalla “Fusionar Pallet en Mantención”

7.7.8 Pantalla “Listar Instrucciones de Salida”

Pantalla en la cual se realiza el despliegue histórico de las instrucciones de salida definidas en el sistema, posibilitando al usuario la selección de un grupo de ellas mediante la especificación de un rango de instrucciones de salida válidas, para ello la aplicación Front-end establecerá comunicación con el WebServices de “cn_InstruccionSalida” para el intercambio de información y confirmación de transacciones realizadas.

Figura 56, Pantalla "Listar Instrucciones de Salida"

7.7.9 Pantalla "Ver y Agregar Instrucción de Salida"

Pantalla en la cual se realiza la inserción de una nueva instrucciónd e salida en el sistema, para ello la aplicación Front-end establecerá comunicación con los WebServices de "cn_InstrucciónSalida" y "cn_Maestros" para el intercambio de información y confirmación de transacciones realizadas.

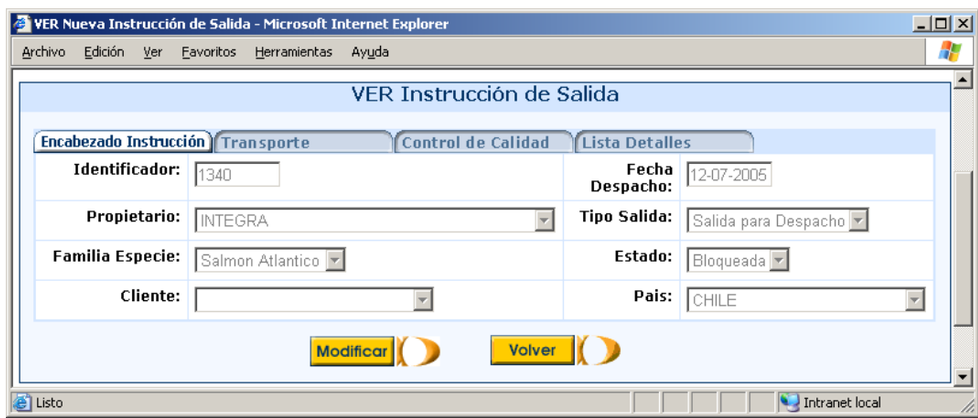


Figura 57, Pantalla "Ver y Agregar Instrucción de Salida"

7.7.10 Pantalla “Nuevo Detalle de Instrucción de Salida”

Pantalla en la cual se realiza la inserción de un nuevo detalle de instrucción de salida, para ello la aplicación Front-end establecerá comunicación con el WebServices de “cn_DetInstruccionSalida” para el intercambio de información y confirmación de transacciones realizadas.

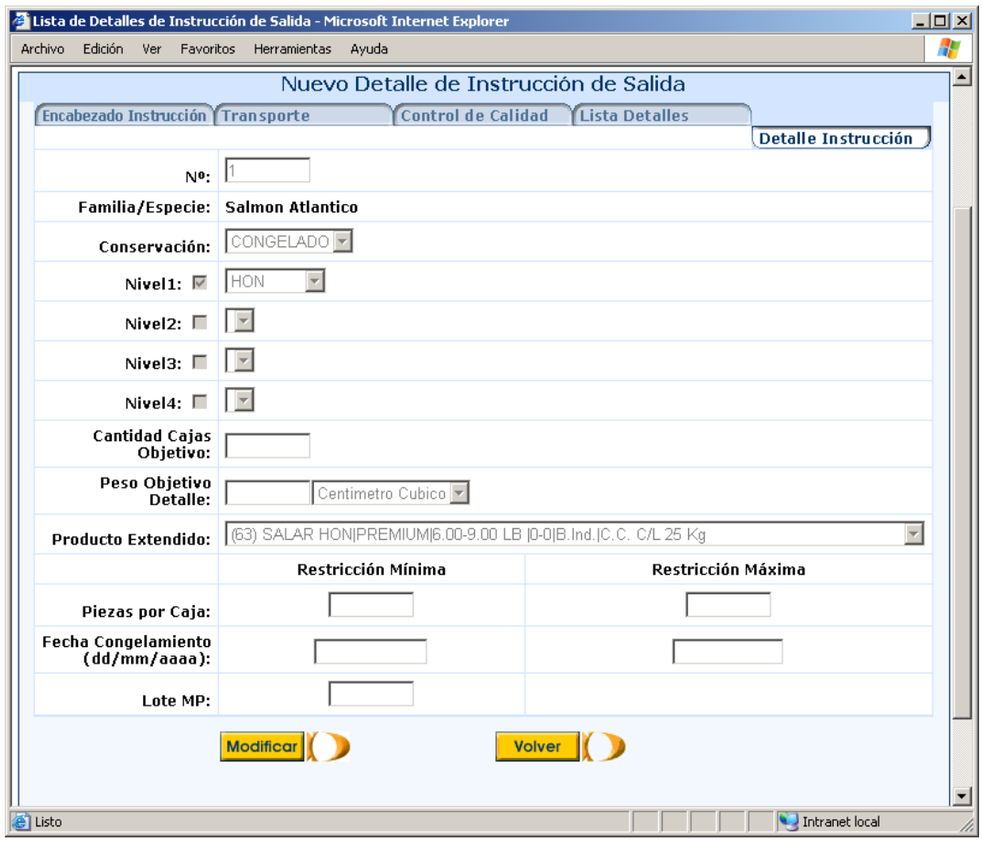


Figura 58, Pantalla “Nuevo Detalle de Instrucción de Salida”

7.7.11 Pantalla “Ejecución de Instrucción de Salida”

Pantalla en la cual se realiza la ejecución de una instrucción de salida, el usuario se encontrará posibilitado de despachar pallets completos o cajas individuales que cumplan los requisitos señalados en los detalles de la instrucción de salida, para ello la aplicación Front-end establecerá comunicación con el WebServices de “cn_InstrucciónSalida” para el intercambio de información y confirmación de transacciones realizadas.

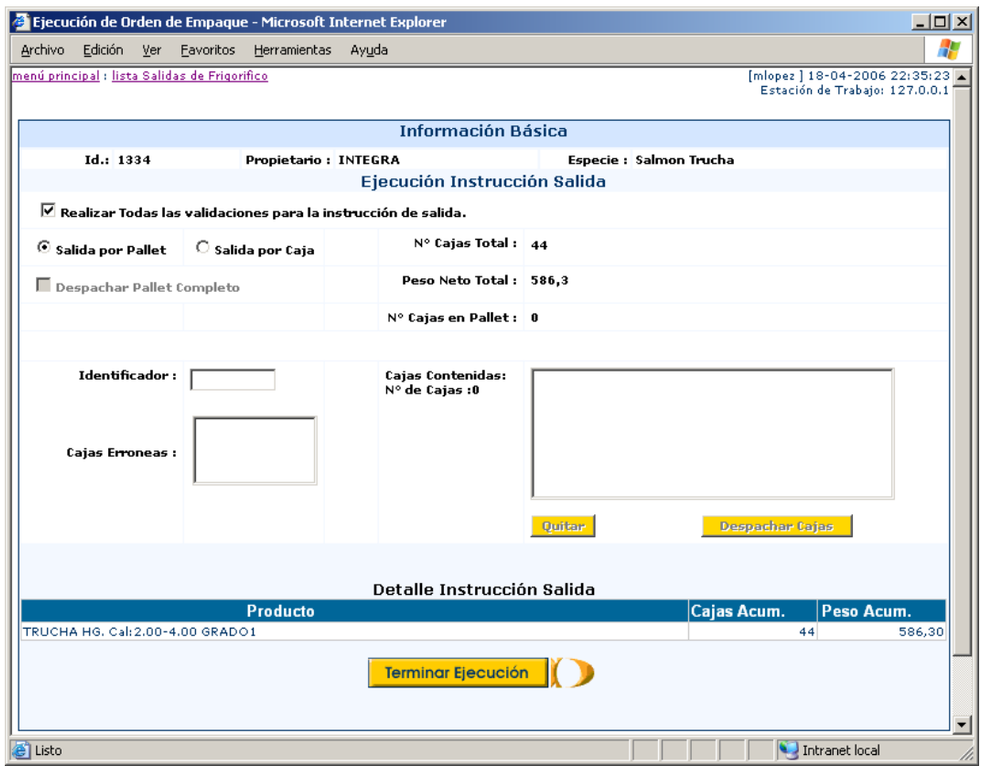


Figura 59, Pantalla "Ejecución de Instrucción de Salida"

Capítulo 8

8. Conclusiones

En el presente proyecto se analizó y diseño una solución integral para una empresa específica que ofrece servicios de maquilado, planteado en una plataforma que facilitará la integración de las diferentes áreas de negocios que comprende a la empresa y orientado a su utilización por diferentes clientes. El diseño y desarrollo de la solución en una arquitectura distribuida en capas permitió encapsular globalmente la lógica de negocios que se presentaba y así lograr una distribución de funcionalidades dentro de la solución planteada.

La encapsulación de las reglas de negocios dentro de una arquitectura distribuida en capas no bastó para ofrecer una solución integral para las diferentes empresas que podrían utilizar el sistema, esto debido a que no fue posible homogeneizar las diferentes plataformas de trabajo que distintivamente utilizaba cada cliente.

Entendiendo que finalmente una empresa maquiladora se transforma en una empresa prestadora de servicios, es vital que la solución planteada por la misma para compartir y administrar su información se hubiese planteado en una forma de trabajo que posibilitara la integración de las diferentes plataformas que utilizaban sus clientes. Dada esta problemática, la utilización de los WebServices se presentaba como la solución más viable de implementar, esto debido a que ofrecía ventajas de modularidad e independencia de plataformas y lenguajes, garantizando así una cobertura completa a los diferentes clientes presentes y futuros que utilizarán el sistema.

La empresa, al prestar así sus servicios, entrega un valor agregado a la gestión de la información generada, marcando diferencias puntuales con el resto de sus competidores y escalando una posición en la utilización de las plataformas de trabajo.

Entre los resultados directos a rescatar una vez finalizado el desarrollo de la solución con las plataformas y herramientas seleccionadas, destacaba la escalabilidad del mismo, debido a la modularización de las lógicas de negocios involucradas en el sistema desarrollados.

La versatilidad de la solución con diferentes aplicaciones front-end logró un nivel de transparencia satisfactoria de acuerdo a los objetivos planteados. Esto debido a la utilización de plataformas de desarrollo completamente distintas a las que originalmente se desenvuelve la solución, pero logrando un sincronismo adecuado que permitió avalar la utilización de los WebServices como agente interlocutor entre la lógica de la solución y clientes de la misma.

Dentro de los inconvenientes a encontrados durante el desarrollo de la solución, bajo la utilización de los WebServices, fue la implementación de un sistema de seguridad acorde a la información que se maneja. Esto debido a que el flujo de información entre el sistema y sus participantes se realiza mediante un intercambio de información plana o información sin encriptar, no pudiendo garantizar así la confiabilidad necesaria para el dialogo entre sus integrantes.

Una segunda problemática presentada durante la puesta en marcha de la solución fue la imposibilidad de contar con herramientas de testing para supervisar el buen funcionamiento de los WebServices, dificultando así la disminución de los tiempos de respuestas a las diferentes incidencias presentadas durante la

implementación. Para sobrellevar estas dificultades se debió realizar replicas de condiciones de trabajo en ambientes locales o simplemente asistir al sistema directamente en donde se desempeñaba.

El diseño e implementación de la solución planteada resultó un desafió no menor, debido a la alta conceptualización de ciclos productivos y manejo de áreas de negocios que deberían ser precisadas en el desarrollo de una solución estable y robusta. Para lograr este objetivo se debió alcanzar un nivel de comprensión del negocio importante, lo cual permitió legitimar las funcionalidades que abarcaba la solución.

La solución en si, aunque estableció puntos de control para las áreas más relevantes para la empresa, no involucró a subprocesos que podrían ser aplicados al conjunto de materias primas recepcionadas, como por ejemplo: control de las degradaciones de las materias primas, órdenes de planificación de las mismas y controles de calibración. Es importante destacar, aunque estos puntos de producción no se encontraban involucrados en la solución original, el diseño implementado permitió generar una solución escalable.

9. Bibliografía

9.1 DOCUMENTOS

- [Chr01] Christensen E., Curbera F., Meredith G., Weerawarana S. (2001), *Web Services Description Language (WSDL)* http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

- [Dan01] Daniel A. Menascé, Virgilio A. F. Almeida (2001), *Capacity Planning for Web Services*

- [Har05]Harold, Elliotte Rusty, Scout Jeans (2005) *XML Imprescindible*

- [Liz02]Lizarraga Celaya, Carlos.(2002). *Servicios Web*. Universidad de Sonora.

- [Gov01]Govindarajan K., Karp A., Kuno H., Beringer D.,Banerji A. (2001), *Conversation Definitions: defining interfaces of web services*

- [Tab02] Tabor, R.(2002). *Servicios Web XML*. España. Prentice Hall

9.2 Direcciones en Internet

- [URL 1] Desarrollo en N Capas http://www.microsoft.com/spanish/msdn/articulos/archivo/081102/voices/dncapas.asp

- [URL 2] Dispositivo de pesaje electrónico http://www.gse-inc.com

- [URL 3] Transmisión Serial

http://www.wut.de/e-8wwww-16-apes-000.php3

- [URL 4] Dispositivo conversor de comunicaciones

http://www.lantronix.com/products/

- [URL 5] Microsoft XML WebServices

http://msdn.microsoft.com/webservices/

- [URL 6] SOAP

http://www.w3.org/TR/soap/

- [URL 7] SOAP, WSDL, UDDI y WebServices

http://www.webservices.org/

- [URL 8] Introducción a UML

http://programacion.com/tutorial/uml/