

Desarrollo de Soluciones Customizadas para el Sistema de Gestión de Almacenes (WMS) de la
empresa NetLogistik

Autores: Luis Alejandro Linares Tejedor y Juan Diego Álvarez Sánchez

Trabajo de Grado para Optar al Título de Ingeniero de sistemas

Director: Juan Ramon Pernaletе Maldonado Doctor en ciencias de la educación

Tutor: Johanny Cecilia Navarro Motta Especialista en Evaluación y Gerencia de Proyectos

Entidad interesada: NET LOGISTIK COLOMBIA S.A.S.

Universidad Industrial de Santander

Facultad de ingenierías fisicomecánicas

Escuela de ingeniería de sistemas e informática

Ingeniería de Sistemas

Bucaramanga

2025

Tabla de contenido

Introducción	14
1. Planteamiento y Justificación del Problema.....	16
2. Objetivos	20
2.1. Objetivo General.....	20
2.2. Objetivos Específicos	20
3. Marco de Referencia	21
4. Metodología	25
4.1. Diseño	25
4.2. Implementación.....	25
4.3. Capacitación y pruebas de usuario.....	26
4.4. Puesta en marcha (Go-Live).....	27
4.5. Optimización y soporte.....	27
5. Desarrollo del proyecto	28
5.1. Trailer Release Page	29
5.1.1. <i>Diseño</i>	30
5.1.2. <i>Implementación</i>	32
5.1.3. <i>Capacitación y pruebas de usuario</i>	40
5.1.4. <i>Puesta en marcha (Go-Live)</i>	40
5.1.5. <i>Optimización y soporte</i>	40
5.1.6. <i>Análisis de resultado</i>	41
5.2. View inbound shipments	42
5.2.1. <i>Diseño</i>	43
5.2.2. <i>Implementación</i>	46
5.2.3. <i>Capacitación y pruebas de usuario</i>	53
5.2.4. <i>Puesta en marcha (Go-Live)</i>	54
5.2.5. <i>Optimización y soporte</i>	54
5.2.6. <i>Análisis de resultado</i>	54
5.3. Packing List Report.....	56

5.3.1.	<i>Diseño</i>	57
5.3.2.	<i>Implementación</i>	59
5.3.3.	<i>Capacitación y pruebas de usuario</i>	64
5.3.4.	<i>Puesta en marcha (Go-Live)</i>	65
5.3.5.	<i>Optimización y soporte</i>	65
5.3.6.	<i>Análisis de resultado</i>	65
5.4.	Bill Of Landing Report	67
5.4.1.	<i>Diseño</i>	68
5.4.2.	<i>Implementación</i>	71
5.4.3.	<i>Capacitación y pruebas de usuario</i>	75
5.4.4.	<i>Puesta en marcha (Go-Live)</i>	76
5.4.5.	<i>Optimización y soporte</i>	76
5.4.6.	<i>Análisis de resultado</i>	76
5.5.	EMS Bill Of Landing	78
5.5.1.	<i>Diseño</i>	79
5.5.2.	<i>Implementación</i>	81
5.5.3.	<i>Capacitación y pruebas de usuario</i>	84
5.5.4.	<i>Puesta en marcha (Go-Live)</i>	84
5.5.5.	<i>Optimización y soporte</i>	84
5.5.6.	<i>Análisis de resultado</i>	85
5.6.	Interface Inventory Reconciliation by Location	86
5.6.1.	<i>Diseño</i>	87
5.6.2.	<i>Implementación</i>	89
5.6.3.	<i>Capacitación y pruebas de usuario</i>	94
5.6.4.	<i>Puesta en marcha (Go-Live)</i>	94
5.6.5.	<i>Optimización y soporte</i>	94
5.6.6.	<i>Análisis de resultado</i>	95
5.7.	Job Release Hold	96

5.7.1. <i>Diseño</i>	97
5.7.2. <i>Implementación</i>	99
5.7.3. <i>Capacitación y pruebas de usuario</i>	101
5.7.4. <i>Puesta en marcha (Go-Live)</i>	101
5.7.5. <i>Optimización y soporte</i>	102
5.7.6. <i>Análisis de resultado</i>	102
6. Conclusiones	103
7. Recomendaciones	105
Referencias Bibliografías	106

Lista de Tablas

Tabla 1: Comparativa de resultados por customización 104

Lista de Figuras

Figura 1: Ejemplo de Reporte Generado con Jaspersoft	18
Figura 2: Ejemplo de Notificación Generada por EMS	19
Figura 5: Arcadia Cold Storage & Logistics	28
Figura 6: Trailer Release Page	29
Figura 7: Diagrama de casos de uso para la Trailer Release Page	31
Figura 8: Diagrama de flujo para la Trailer Release Page.....	32
Figura 9: Lógica del comando " list var receipt transport "	34
Figura 10: Lógica del comando " change var transport reference "	35
Figura 11: .resource de la Trailer Release Page	36
Figura 12: .action para listar de la Trailer Release Page	37
Figura 13: .action para modificar de la Trailer Release Page	37
Figura 14: .json de la Trailer Release Page.....	38
Figura 15: Criterios de búsqueda de la Trailer Release Page.....	39
Figura 16: Acción para modificar tiempo de liberación	39
Figura 17: Matriz de pruebas de la Trailer Release Page	40
Figura 18: View Inbound Shipment.....	43
Figura 19: Diagrama de flujo para la View Inbound Shipment	45
Figura 20: <i>Diagrama de Casos de usos para View Inbound Shipment</i>	46
Figura 21: Lógica del comando " list var inbound shipment to lpn"	49
Figura 22: .resource para Page Builder “View Inbound Shipment”	50
Figura 23: .action para Page Builder “View Inbound Shipment”	51
Figura 24: .json para Page Builder “View Inbound Shipment”	52
Figura 25: Criterios de búsqueda de la View Inbound Shipment.....	53
Figura 26: Matriz de pruebas del page View Inbound Shipment	53
Figura 27: Reporte Packing List	56
Figura 28: Diagrama de flujo para el reporte Packing List	58
Figura 29: Diagrama Entidad-Relación para el reporte Packing List.....	59
Figura 30: Lógica del comando "list var report packing list header"	60
Figura 31: Lógica del comando "list var report packing list detail"	61
Figura 32: Editor de consulta Jaspersoft para el reporte Packing List	62

Figura 33: Diseño del reporte principal Packing List en Jaspersoft	62
Figura 34: Diseño del sub-reporte Packing List en Jaspersoft	63
Figura 35: Vista previa del reporte Packing List en la interfaz web UI.....	63
Figura 36: Matriz de pruebas del reporte Packing List.....	64
Figura 37: Reporte Bill Of Landing	68
Figura 38: Diagrama de flujo para el reporte Bill Of Landing	70
Figura 39: Lógica del comando " list var report bill of lading"	72
Figura 40: Lógica del comando " list var detail report bill of lading"	72
Figura 41: Editor de consulta Jaspersoft para el reporte Bill Of Landing	73
Figura 42: Diseño del reporte principal Bill Of Landing en Jaspersoft	74
Figura 43: Diseño del subreporte Bill Of Landing en Jaspersoft.....	74
Figura 44: Matriz de pruebas del reporte Bill Of Landing	75
Figura 45: Reporte "Bill Of Landing" generado y enviado por el EMS.....	78
Figura 46: Diagrama de flujo para el EMS del reporte BOL	80
Figura 47: Lógica del comando "Create var EMS costum bol report"	81
Figura 48: Lógica del comando "raise var EMS for bol report client"	82
Figura 49: Políticas generadas para el EMS del reporte BOL.....	82
Figura 50: Lógica encargada de correr el workflow	83
Figura 51: Matriz de pruebas para el EMS del reporte Bill Of Landing.....	84
Figura 52: Interface Inventory Reconciliation by Location.....	86
Figura 53: Diagrama de flujo para la interface Inventory Reconciliation	88
Figura 54: Diagrama Entidad-Relación para la interface Inventory Reconciliation	89
Figura 55: Lógica del comando "retrieve var inventory reconciliation"	90
Figura 56: Flujo de datos en las transacciones de salida	91
Figura 57: Evento de salida de la interface Inventory Reconciliation.....	92
Figura 58: Método de recuperación de la interface Inventory Reconciliation	92
Figura 59: Job automatizado para la interface Inventory Reconciliation	93
Figura 60: Archivo CSV generado por la interface Inventory Reconciliation.....	93
Figura 61: Matriz de pruebas de la interface Inventory Reconciliation.....	94
Figura 62: Vista del Job Release Hold en la consola.....	96
Figura 63: Diagrama de flujo para el job release hold	98

Figura 64: Lógica del comando release var inventory hold.....	99
Figura 65: Lógica de configuración del Job Release Hold.....	100
Figura 66: Matriz de pruebas para el job release var inventory hold	101

Lista de Apéndices

Apéndice A: Descripción de la empresa	108
Apéndice B: Certificado de Finalización de Prácticas Empresariales Juan Alvarez	109
Apéndice C: Certificado de Finalización de Prácticas Empresariales Luis Linares	110

Glosario

3PL (Third-Party Logistics): Proveedor externo que gestiona operaciones logísticas para otras empresas, común en cadenas de suministro especializadas (ej: cadena fría).

ATOS: Metodología utilizada en implementaciones de sistemas de información y transformación digital, caracterizada por su enfoque estructurado y división en fases claras para gestionar proyectos complejos.

BlueYonder: Proveedor líder de soluciones tecnológicas para la gestión de la cadena de suministro y almacenes (WMS), con capacidades predictivas y de personalización.

EMS (Express Mail Service): Sistema de notificaciones automáticas integrado en el WMS para enviar reportes, alertas o documentos (ej: PDF) a usuarios o sistemas externos.

Jaspersoft: Herramienta de código abierto para la generación de reportes personalizados, integrada con bases de datos SQL.

Jobs (Tareas Programadas): Procesos automatizados ejecutados en horarios específicos para realizar actividades recurrentes (ej: generación de reportes).

MOCA (McHugh Open Component Architecture): Entorno técnico basado en Java para desarrollar componentes modulares y personalizaciones en sistemas como BlueYonder.

PAGE BUILDER: Extensión del WMS que permite diseñar interfaces gráficas personalizadas con funciones CRUD y conectividad a bases de datos.

Pallet: Plataforma plana, generalmente fabricada en madera, plástico o metal, utilizada para apilar, almacenar y transportar mercancías de manera eficiente en entornos logísticos. En el contexto del WMS, cada pallet se identifica con un LPN (License Plate Number).

Picking: Proceso de selección y recolección de productos específicos en un almacén para cumplir con los pedidos de los clientes, optimizando tiempos y reduciendo errores mediante el uso de tecnologías y rutas eficientes.

Stock: Cantidad total de mercancías o productos disponibles en un almacén o centro de distribución, gestionada y monitoreada a través del WMS.

WMS (Warehouse Management System): Sistema de gestión de almacenes que optimiza procesos como recepción, almacenamiento y despacho de mercancías.

Resumen

Título: Desarrollo de Soluciones Customizadas para el Sistema de Gestión de Almacenes (WMS) de la empresa NetLogistik

Autores: Luis Alejandro Linares Tejedor y Juan Diego Álvarez Sánchez.

Palabras Clave: WMS, SQL, MOCA, BlueYonder, NetLogistik, Customización, Automatización, Jaspersoft, PAGE BUILDER, EMS.

Descripción: Las operaciones de cualquier almacén dentro de una cadena de suministro requieren actividades como planeación, gestión y control del almacenamiento, así como el envío y recepción de bienes. Estas actividades siguen procedimientos y estándares que permiten construir una trazabilidad efectiva del producto, reduciendo errores humanos.

Para facilitar y optimizar estas operaciones, se desarrollaron los Sistemas de Gestión de Almacenes (WMS), como el de BlueYonder. Este tipo de software permite controlar de manera ordenada el ciclo de vida del producto en el almacén, desde su entrada hasta su salida, incluyendo procesos clave como recepción, almacenamiento, picking y despacho.

El propósito del trabajo es documentar la planeación, creación e implementación de soluciones digitales en el WMS de BlueYonder para Netlogistik, adaptadas a las necesidades específicas de cada cliente. Estas soluciones incluyen tareas programadas (JOBS), reportes personalizados con datos procesables, envío de mensajes automáticos (EMS), interfaces para gestionar transacciones y comandos desarrollados en SQL y MOCA.

Con estas adaptaciones, se busca mejorar la eficiencia operativa y aumentar la satisfacción del cliente desde la implementación del sistema.

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Ingeniería de Sistemas. Director: Juan Ramon Pernaletе Maldonado. MSc, Escuela de Ingeniería de Sistemas.

Abstract

Title: Development of customized solutions for NetLogistik's Warehouse Management System (WMS).

Authors: Luis Alejandro Linares Tejedor and Juan Diego Álvarez Sánchez.

Keywords: WMS, SQL, MOCA, BlueYonder, NetLogistik, Customization, Automation, Jaspersoft, PAGE BUILDER, EMS.

Description: The operations of any warehouse within a supply chain require activities such as planning, management and control of storage, as well as shipping and receiving of goods. These activities follow procedures and standards that allow building an effective traceability of the product, reducing human errors.

To facilitate and optimize these operations, Warehouse Management Systems (WMS), such as BlueYonder's, have been developed. This type of software allows to control in an orderly manner the life cycle of the product in the warehouse, from its entry to its exit, including key processes such as receiving, storage, picking and dispatch.

The purpose of the work is to document the planning, creation and implementation of digital solutions in the BlueYonder WMS for Netlogistik, tailored to the specific needs of each customer. These solutions include scheduled tasks (JOBS), customized reports with actionable data, automatic message sending (EMS), interfaces to manage transactions and commands developed in SQL and MOCA.

With these adaptations, we seek to improve operational efficiency and increase customer satisfaction from the implementation of the system.

* Degree Work

** Faculty of Physical-Mechanical Engineering. School of Systems and Computer Engineering. Systems Engineering. Director: Juan Ramon Pernalete Maldonado. MSc, School of Systems Engineering.

Introducción

El campo de la logística se ha convertido en un elemento fundamental para llevar a cabo las actividades operacionales que requiere cualquier almacén de cadena, desde la planeación, gestión y control del almacenamiento hasta el envío y recepción de bienes dentro de una cadena de suministro; cada una de estas actividades representa un conjunto de procedimientos y estándares alrededor de los cuales es posible construir una trazabilidad que permite conocer el histórico, la ubicación y la trayectoria de un producto o lote de productos a lo largo de la cadena de suministro en un momento dado; con el fin de facilitar este procedimiento y reducir la ocurrencia de errores humanos en el seguimiento de productos.

Ante tales perspectivas se hizo necesario la creación de un software especializado, dando lugar a los sistemas de gestión de almacenes (WMS, por sus siglas en inglés Warehouse Management System). Ante todo lo expuesto este trabajo busca documentar la planeación, creación e implementación de soluciones digitales dentro del sistema de gestión de almacenes de BlueYonder para la empresa Netlogistik, que sean aptas para llevar a cabo las actividades operacionales de un cliente de acuerdo a necesidades y/o requerimientos específicos.

Un WMS abarca todo el proceso de ciclo de vida de un producto de manera ordenada y controlada, desde que entra hasta que sale del almacén, incluyendo el envío, recepción, entrada de stock y picking, además, con la ayuda de este software se permite dar seguimientos a temas como cumplimiento de perdidos, garantizar que cada proceso se realice de manera adecuada, facilitar y gestionar operaciones de logística y evitar demoras, pérdidas o desperdicios; sin embargo es propicio modificar o desarrollar funcionalidades únicas según las operaciones que realice el cliente para lograr un óptimo y eficaz uso del software.

La realización de esta práctica empresarial parte del desarrollo de diversas customizaciones tales como tareas programadas (JOBS) para ejecutar procesos operativos únicos, generación de reportes con base en datos procesables y que permiten el monitoreo del

rendimiento operativo, la creación y modificación de interfaces de comunicación que faciliten el manejo de la información por medio de transacciones de entrada y salida, además del desarrollo de diferentes comandos en los lenguajes SQL y MOCA (por sus siglas en inglés McHugh Open Component Architecture) que modifiquen el funcionamiento estándar del sistema de acuerdo a los requerimientos solicitados. Gracias a estas soluciones se espera mejorar de forma clara y oportuna la operación logística y la satisfacción del cliente desde el primer momento que se implementa el WMS.

1. Planteamiento y Justificación del Problema

Netlogistik enfrenta el desafío de gestionar operaciones logísticas complejas en almacenes que requieren soluciones personalizadas para optimizar la eficiencia y reducir errores en la gestión de inventario. Con el fin de impulsar una solución completa, la empresa ofrece la integración de un Sistema de Gestión de Almacenes (WMS, por sus siglas en inglés), desarrollado por BlueYonder, el cual tiene como base la ejecución de funcionalidades estándar, pero que con alta frecuencia requiere ser adaptado a las necesidades particulares de cada cliente.

Este software brinda un beneficio útil al momento de controlar y administrar el inventario dentro del almacén, permitiendo así gestionar procesos como el recibo, almacenamiento, surtido (picking) y despacho de mercancías. Desde su implementación y puesta en funcionamiento se contempla la necesidad de adaptarse a las operaciones logísticas modernas de cada cliente, donde toma relevancia la creación de funcionalidades adicionales que no están disponibles en la versión estándar del sistema, como la falta de personalización adecuada en el WMS lo que ha desencadenado ineficiencias operativas, errores en la administración del inventario y dificultades en la trazabilidad de la información.

Partiendo de este contexto, Netlogistik contempla como una prioridad de alto valor lograr una adaptación oportuna del WMS según los requerimientos específicos de cada cliente, bien sea a través de la optimización de la visualización de datos, la mejora en la precisión y eficiencia del flujo de información, así como también la creación de interfaces que permitan realizar operaciones de tipo CRUD directamente en la base de datos. Para lograr esto, se requiere el uso de herramientas como comandos SQL y MOCA, ya que el sistema se desarrolló basado en estas tecnologías, así también se debe tener dominio de Jaspersoft para la generación de reportes y PAGE BUILDER para el desarrollo de pantallas personalizadas.

Cualquier solución que se pueda plantear debe estar enfocada en el diseño e implementación de comandos personalizados para diferentes customizaciones y permitir generar

información confiable al momento de evaluar la precisión del inventario, optimizar los tiempos de procesamiento de transacciones, y de igual forma ha de impulsar una mayor eficiencia operativa. Esto ha de incluir el desarrollo de algoritmos que permitan automatizar tareas recurrentes y la integración de reportes personalizados, teniendo como fin último mejorar la calidad de la información disponible para la toma de decisiones.

La generación de reportes personalizados facilitará la identificación de patrones y tendencias en la administración del inventario, a partir de la ejecución de comandos para la consulta y procesamiento de información clave, como las características físicas del inventario o el tratamiento del inventario recibido (ver Figura 1).

Figura 1: Ejemplo de Reporte Generado con Jaspersoft

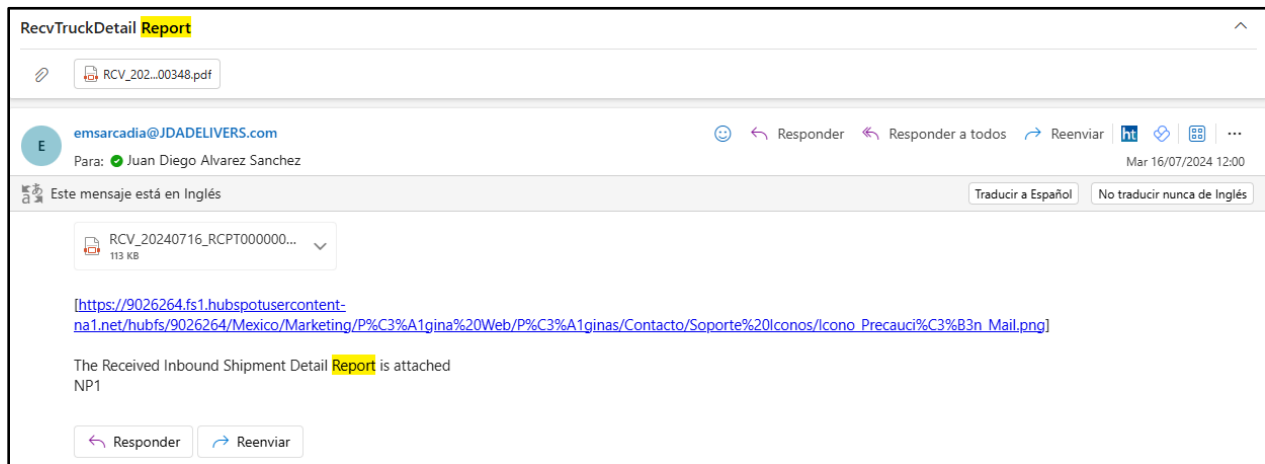
The screenshot displays the TIBCO JasperSoft Studio interface. On the left, the 'Outline' pane shows a tree structure for 'Varrcpttruckdetails', including 'Parameters' and 'Fields' such as 'wh_id', 'trknum', 'client_id', 'supnum', 'invnum', 'po_num', 'rcvtrk_stat_dsc', 'prtnum', 'prt_dsc', 'prt_client_id', 'untqty', 'stkuom', 'catch_qty', 'lotnum', 'clsdte', 'expire_dte', 'mandte', 'lodnum', 'rec_loc', and 'trlr_num'. Below the Outline is the 'Palette' with various design elements like 'Text Field', 'Image', 'Table', etc. The main design area shows a report titled 'Received Inbound Shipment Detail Report'. The report includes a header table with fields like Client ID, Equipment, Carrier, Shipment, Shipment Status, Close date, and Supplier. Below this is a main data table with columns LPN, Unit Qty, Catch Qty, Lot, Exp date, and Man date. The footer section contains shipping details, a 'Printed:' field, and a 'Page Footer' with the text 'Page X of Y'. A large black box with the text 'THIS IS NOT AN INVOICE' is overlaid on the report design.

Fuente: Los Autores (2025)

El proyecto también busca desarrollar pantallas de visualización de tipo PAGE BUILDER que permitan a los usuarios interactuar directamente con la base de datos, realizando operaciones como la inserción, actualización o eliminación de registros de manera intuitiva y eficiente. Esto no solo mejorará la usabilidad del sistema, sino que también reducirá la dependencia de los equipos técnicos para realizar modificaciones en la base de datos.

Finalmente, para la administración óptima y eficiente de la información destinada a los clientes se configurarán sistemas Express Mail Service (EMS), incorporando algoritmos personalizados. Los sistemas EMS permiten enviar notificaciones y reportes automáticos a los clientes, lo que mejora la comunicación y la transparencia en la gestión de los almacenes, además de garantizar que la información sea relevante y oportuna (*ver Figura 2*).

Figura 2: Ejemplo de Notificación Generada por EMS



Fuente: Los Autores (2025)

2. Objetivos

2.1. Objetivo General

Implementar soluciones customizadas al Sistema de Gestión de Almacenes (WMS), que permita elevar el nivel de eficiencia de los procesos en la cadena de suministros como, precisión de inventario, tiempos de procesamiento de órdenes y la generación de reportes personalizados para distintos niveles administrativos y operativos.

2.2. Objetivos Específicos

- Identificar las limitaciones actuales del WMS en la gestión de datos (inventario, tiempos de procesamiento y generación de reportes)
- Evaluar un conjunto de posibles soluciones técnicas mediante el uso de SQL y MOCA, orientadas a mejorar la precisión del inventario, la eficiencia del procesamiento de órdenes y la visualización de datos en tiempo real.
- Desarrollar la solución óptima mediante la creación de prototipos funcionales.
- Validar la Puesta en Marcha de las Soluciones customizadas, a través de Pruebas que garanticen su funcionamiento en un entorno productivo.

3. Marco de Referencia

La logística y la gestión de la cadena de suministro han evolucionado significativamente con la incorporación de tecnologías avanzadas que optimizan la planificación, ejecución y control de operaciones. Empresas especializadas como Netlogistik han desarrollado soluciones innovadoras como los sistemas de gestión de almacenes (WMS) y de transporte (TMS), que permiten a las organizaciones mejorar la eficiencia, visibilidad y trazabilidad de sus procesos logísticos.

Netlogistik es una empresa especializada en soluciones integrales de logística y cadena de suministro que se destaca por su enfoque vanguardista en la gestión de operaciones comerciales. Con años de experiencia en el sector, la empresa se ha destacado por la implementación de sistemas de Blue Yonder, como lo son WMS y TMS, los cuales apoyan a grandes industrias dedicadas a la cadena de suministro en la optimización de costos y la gestión eficiente del inventario y la distribución de productos.

En particular, para un proveedor logístico de terceros (3PL) especializado en la cadena de frío, donde la tecnología juega un papel crucial en la trazabilidad, gestión de inventarios y monitoreo del estado del inventario, factores esenciales para preservar la calidad y seguridad de los productos (Smith & Johnson, 2020). La creciente complejidad de la cadena de suministro ha generado la necesidad de herramientas más avanzadas, como Blue Yonder, que garantizan una respuesta ágil y adaptativa en un entorno altamente dinámico.

El WMS, siendo una excelente solución para la gestión de la cadena de suministro y almacenes, se implementa como software clave en este tipo de operaciones. Esta herramienta avanzada no solo automatiza los flujos de trabajo, sino que además permite realizar desarrollos que logren adaptarse al modelo de negocio de cada cliente. Al integrar tecnologías de análisis predictivo y capacidades de aprendizaje automático, Blue Yonder puede anticipar demandas,

gestionar eficientemente la rotación de inventarios bajo metodologías como FEFO (First Expired, First Out) y minimizar el desperdicio de productos perecederos (Blue Yonder, 2023).

MOCA (McHugh Open Component Architecture) es una tecnología clave en la estructura de Blue Yonder. Su arquitectura modular permite la integración, personalización y escalabilidad de soluciones empresariales. Basado en Java, MOCA hereda características clave como la portabilidad, permitiendo que su código se ejecute en diferentes sistemas operativos sin modificaciones significativas.

Gracias a su base en Java, MOCA aprovecha la robustez y flexibilidad del lenguaje, permitiendo la creación de aplicaciones modulares con componentes aislables que pueden liberarse de manera independiente, facilitando el mantenimiento y la evolución de la plataforma sin afectar el resto del sistema. Estas customizaciones permiten que Blue Yonder no solo se integre perfectamente con sistemas existentes como sensores IoT, software ERP y plataformas de transporte, sino que también optimice procesos operativos mediante soluciones hechas a la medida. Por ejemplo, los comandos personalizados y los jobs automatizados agilizan tareas repetitivas, mientras que las etiquetas y reportes adaptados garantizan la precisión y claridad en la comunicación interna y externa.

La arquitectura modular de MOCA proporciona una visibilidad de extremo a extremo en la cadena de suministro, asegurando que todas las áreas de la operación trabajen de manera sincronizada. Esto permite a las operaciones logísticas adaptarse rápidamente a cambios en las demandas del mercado o a requisitos regulatorios, garantizando una respuesta ágil y efectiva (Jones, 2019). Al combinar la flexibilidad de MOCA con las capacidades avanzadas de Blue Yonder, las empresas logran un nivel de personalización sin precedentes que impulsa su competitividad y asegura el cumplimiento de altos estándares de calidad.

Un ejemplo de un caso de éxito en la implementación de Blue Yonder por parte de NetLogistik es el caso de Alkosto Hiperahorro, una empresa colombiana con más de 60 años en

el mercado de distribución y 20 años en el sector retail. Alkosto, conocida por ser líder en la venta de electrodomésticos y productos electrónicos, enfrentó desafíos relacionados con el crecimiento sostenido de sus ventas, la apertura de nuevos almacenes y la creciente demanda de un mejor servicio al cliente. Para abordar estas necesidades, la empresa implementó Blue Yonder como su solución de gestión de almacenes, obteniendo resultados impresionantes.

Entre los logros más destacados, Alkosto logró aumentar la exactitud del inventario al 97%, reducir el tiempo promedio de acopio de 2.5 horas a solo 50 minutos por carga, y disminuir las mermas de un 0.20% a un 0.007%. Además, incrementaron los conteos fiscales de 2 a 4 veces al año y aumentaron la productividad en los procesos operativos en un 30%. La solución también les permitió medir la productividad del personal y mantener estándares de calidad y servicio sobresalientes, cumpliendo con las exigencias de un mercado altamente competitivo.

Otro caso relevante en la implementación de tecnología para la optimización logística es el trabajo de grado para optar a un título del estudiante Hayder Alexander Rodriguez Diaz de la Universidad Autónoma de Bucaramanga donde se desarrolló un sistema de gestión de información logística basado en servicios web para la empresa Transportes Exprecar S.A.S. De acuerdo con Rodríguez Díaz (2020), la empresa enfrentaba dificultades en la administración de sus operaciones debido a la falta de integración y automatización en sus procesos. Para solucionar este problema, se diseñó e implementó un sistema que permitió mejorar la trazabilidad de los envíos, la gestión de recolección y entrega de mercancías, y la eficiencia en la operación.

Este sistema incorporó módulos para la administración del inventario, el monitoreo de rutas y la generación automatizada de reportes, facilitando la toma de decisiones y reduciendo costos operativos. Además, la solución permitió la integración con tecnologías como IoT y GPS para un control en tiempo real de los envíos (Rodríguez Díaz, 2020).

La experiencia de Transportes Exprecar S.A.S. refleja cómo las soluciones tecnológicas avanzadas, como Blue Yonder y MOCA, pueden ser clave en la modernización de la logística. Al

igual que en el caso de Alkosto Hiperahorro, la implementación de sistemas de gestión eficientes no solo mejora la visibilidad de las operaciones, sino que también permite optimizar costos, aumentar la productividad y garantizar un servicio de calidad en un mercado cada vez más exigente.

Todas las soluciones gestionadas serán el resultado de un análisis detallado de la necesidad o del problema, la búsqueda técnica de alternativas viables, el desarrollo adecuado de la solución, su implementación en el WMS, el desarrollo de pruebas y validación con los usuarios finales, para finalmente, proveer un monitoreo continuo del desempeño de la solución implementada asegurando su cumplimiento en relación de las expectativas del cliente.

Entre las principales herramientas utilizadas en este proceso se encuentran:

- SQL y MOCA: Para la implementación de comandos personalizados que permitan desarrollar consultas avanzadas, algoritmos de validación de datos y funciones CRUD.
- PAGE BUILDER: Interfaces diseñadas para visualizar y manipular datos según los requerimientos del cliente.
- JasperSoft: Para la creación de reportes personalizados y la optimización de la visualización de datos.
- Express Mail Service (EMS): Sistemas de notificación automática que permitan una administración eficiente de la información destinada a los clientes.
- Diagramas de flujo y documentación técnica: Para garantizar que cada solución desarrollada esté bien estructurada, sea fácil de comprender y pueda ser utilizada como referencia en futuras actualizaciones del sistema.

4. Metodología

El desarrollo de este proyecto se realizó siguiendo la serie de lineamientos establecidos bajo la metodología ATOS, la cual promueve Netlogistik. Esta metodología es ampliamente utilizada en implementaciones de sistemas de información, transformación digital o servicios de TI. El uso de esta metodología nace de la necesidad de buscar un alcance holístico que permita evaluar y dimensionar proyectos complejos, escalables y que requieran la integración de sistemas, además de apoyar una gestión organizada, controlada y predecible del proyecto gracias a su enfoque estructural y división por fases claras.

4.1. Diseño

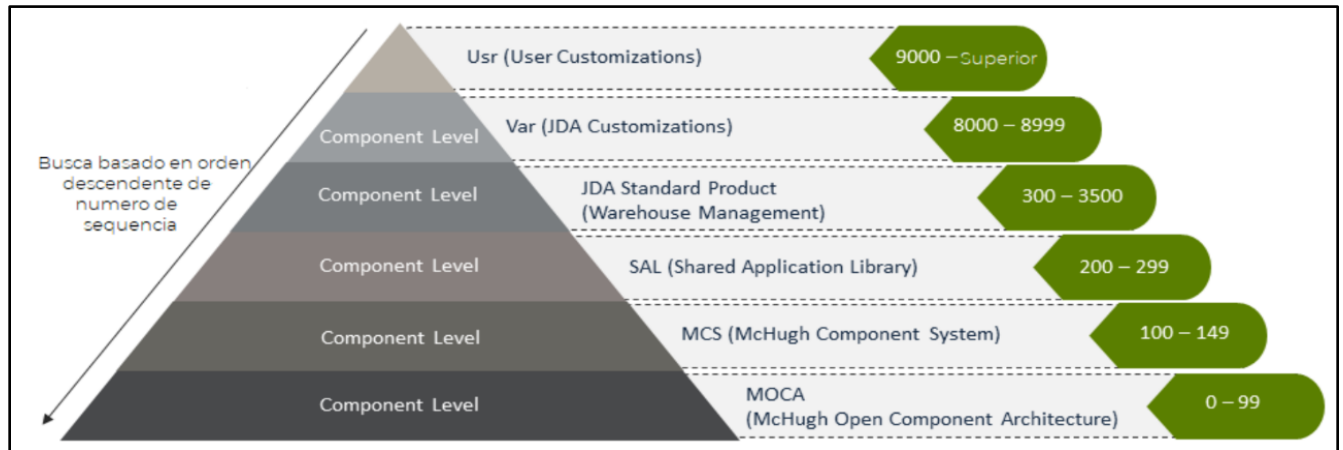
Considerando los requerimientos funcionales y técnicos del cliente se establecen las tecnologías implicadas en la solución, considerando el alcance de su ejecución en el WMS y si es necesario la integración de sistemas externos. Para visualizar y comunicar la customización que se va a desarrollar se usan diagramas de caso de uso, historias de usuario, diagramas de despliegue o diagramas de flujo de datos según sea el caso, así como también documentación técnica donde se especifica los estándares, módulos o configuraciones implicados. Una vez que se validan y comprenden los diseños y procesos a seguir con el cliente se da inicio a la implementación.

4.2. Implementación

De acuerdo al nivel de complejidad se realizan cambios a la configuración básica del WMS, se escriben y prueban scripts SQL y MOCA para modificaciones en la base de datos y/o se desarrollan módulos personalizados de acuerdo al manejo y control de la información; En la ejecución de todas las pruebas y corrección de resultados se hace uso de ambientes de prueba que facilitan seguir un enfoque iterativo a la hora de garantizar que las modificaciones sean funcionales y mejoren el rendimiento del sistema sin interrumpir las operaciones diarias.

La integración de los algoritmos MOCA se ubica según su nivel de componente, con lo que se mantiene claridad en el origen y propósito de cada comando, así como favorece su mantenimiento al verificar solo el nivel correspondiente sin necesidad de revisar la base del sistema.

Figura 3: Nivel de jerarquía de los componentes del WMS



Fuente: BY Basic to Advanced MOCA for development (2024)

Cada componente de la customización es validado por medio de pruebas unitarias, de las cuales se efectúa un seguimiento conforme avanza el versionamiento de la customización, de igual manera, cuando en el flujo de datos y los procesos modificados se ven involucrados diferentes módulos del WMS, se evalúa la interacción o posible impacto de las personalizaciones utilizando pruebas integrales.

4.3. Capacitación y pruebas de usuario

Una vez completo el desarrollo preliminar de la solución se valida su nivel de cumplimiento de acuerdo a los requerimientos solicitados en una sesión con el cliente, donde se preparan escenarios que simulen operaciones reales con el fin de interactuar con las nuevas funcionalidades, verificar que cumplen con sus necesidades y confirmar que las modificaciones funcionan según lo esperado. De igual manera, para familiarizar tanto a los usuarios finales como

a los administradores o equipos técnicos sobre los cambios introducidos se elaboran manuales de usuario y documentación técnica. En estos documentos se aclaran las guías y procedimientos a seguir, así como las modificaciones realizadas a nivel de bases de datos, configuraciones, interfaces de usuario, entre otros aspectos relevantes.

4.4. Puesta en marcha (Go-Live)

El proceso de lanzamiento de cualquier customización inicia con el establecimiento de un cronograma donde se detallan las fechas para la instalación y monitoreo de funcionalidades en el ambiente de producción que sea requerido. En la instalación se prioriza que el sistema pueda realizar operaciones reales del almacén sin interrupciones, por lo cual se ejecuta únicamente durante ventanas de tiempo fuera de la actividad operacional. En el momento en que se completa la instalación, se procede a validar en tiempo real que las configuraciones y personalizaciones funcionan correctamente. Asimismo, se habilitan opciones manuales en caso de que la transición lo requiera.

4.5. Optimización y soporte

En esta fase, se afinan los procesos y se resuelven problemas residuales. Para ello, se revisa constantemente el desempeño del WMS para identificar posibles áreas de mejora, ya se analizando métricas como tiempo de ciclo, precisión del inventario o velocidad en la ejecución de procesos. Según las necesidades reales observadas durante las primeras semanas de operación, se ajustan configuraciones para garantizar una experiencia positiva para los usuarios finales.

La solución de incidencias se enfoca en proporcionar soporte durante la operación diaria y evitar en lo mínimo que las actividades se interrumpan. Cuando se reporta un problema, inicia una investigación con el fin de determinar las causas y factores que lo desencadenaron. Lo cual permite mitigar su impacto en otros componentes del sistema mientras se implementan las correcciones necesarias.

5. Desarrollo del proyecto

Arcadia

Se realizaron 7 customizaciones para la empresa Arcadia Cold Storage & Logistics, la cual es un 3PL especializado en la cadena de frío, dedicado a brindar servicios de almacenamiento en instalaciones de última generación, gestión de inventarios, transporte y logística integrada para optimizar la cadena de suministro de sus clientes. Su red de almacenes estratégicamente ubicados y su tecnología avanzada garantizan la eficiencia, trazabilidad y seguridad de los productos, ayudando a empresas de la industria alimentaria a mantener la calidad y frescura de sus productos mientras reducen costos operativos. Esta empresa cuenta con 6 ambientes productivos operados bajo la el WMS.

Figura 3: Arcadia Cold Storage & Logistics

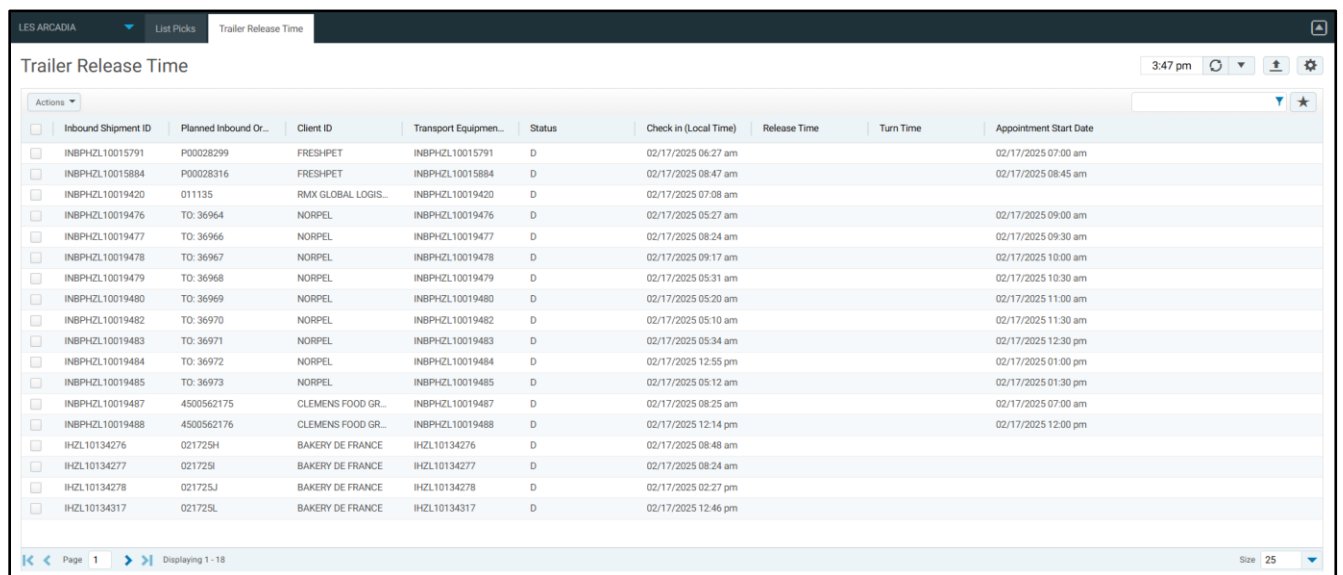


Fuente: BlueYonder (2024)

5.1. Trailer Release Page

Esta interfaz personalizada desarrollada con la ayuda de la extensión Page Builder tiene como principal objetivo permitir a los usuarios actualizar el tiempo de liberación (release time) de los equipos de transporte que operan en el recibo. Cuando un envío entrante es planificado en la mayor parte de casos se asigna una fecha de llegada esperada (Appointment Start Date), el cliente desea que se lleve un control sobre el tiempo transcurrido desde que el equipo de transporte llega al almacén hasta que es despachado. Esta customización proporciona una interfaz intuitiva que permite a los usuarios actualizar el tiempo de liberación de manera rápida y precisa, reduciendo los errores manuales y mejorando la visibilidad en tiempo real de los procesos logísticos.

Figura 4: *Trailer Release Page*



Inbound Shipment ID	Planned Inbound Dr...	Client ID	Transport Equipmen...	Status	Check in (Local Time)	Release Time	Turn Time	Appointment Start Date
INBPHZL10015791	P00028299	FRESHPET	INBPHZL10015791	D	02/17/2025 06:27 am			02/17/2025 07:00 am
INBPHZL10015884	P00028316	FRESHPET	INBPHZL10015884	D	02/17/2025 08:47 am			02/17/2025 08:45 am
INBPHZL10019420	011135	RMX GLOBAL LOGIS...	INBPHZL10019420	D	02/17/2025 07:08 am			
INBPHZL10019476	TO: 36964	NORPEL	INBPHZL10019476	D	02/17/2025 05:27 am			02/17/2025 09:00 am
INBPHZL10019477	TO: 36966	NORPEL	INBPHZL10019477	D	02/17/2025 08:24 am			02/17/2025 09:30 am
INBPHZL10019478	TO: 36967	NORPEL	INBPHZL10019478	D	02/17/2025 09:17 am			02/17/2025 10:00 am
INBPHZL10019479	TO: 36968	NORPEL	INBPHZL10019479	D	02/17/2025 05:31 am			02/17/2025 10:30 am
INBPHZL10019480	TO: 36969	NORPEL	INBPHZL10019480	D	02/17/2025 05:20 am			02/17/2025 11:00 am
INBPHZL10019482	TO: 36970	NORPEL	INBPHZL10019482	D	02/17/2025 05:10 am			02/17/2025 11:30 am
INBPHZL10019483	TO: 36971	NORPEL	INBPHZL10019483	D	02/17/2025 05:34 am			02/17/2025 12:30 pm
INBPHZL10019484	TO: 36972	NORPEL	INBPHZL10019484	D	02/17/2025 12:55 pm			02/17/2025 01:00 pm
INBPHZL10019485	TO: 36973	NORPEL	INBPHZL10019485	D	02/17/2025 05:12 am			02/17/2025 01:30 pm
INBPHZL10019487	4500562175	CLEMENS FOOD GR...	INBPHZL10019487	D	02/17/2025 08:25 am			02/17/2025 07:00 am
INBPHZL10019488	4500562176	CLEMENS FOOD GR...	INBPHZL10019488	D	02/17/2025 12:14 pm			02/17/2025 12:00 pm
IHZL10134276	021725H	BAKERY DE FRANCE	IHZL10134276	D	02/17/2025 08:48 am			
IHZL10134277	021725I	BAKERY DE FRANCE	IHZL10134277	D	02/17/2025 08:24 am			
IHZL10134278	021725J	BAKERY DE FRANCE	IHZL10134278	D	02/17/2025 02:27 pm			
IHZL10134317	021725L	BAKERY DE FRANCE	IHZL10134317	D	02/17/2025 12:46 pm			

Fuente: Los Autores (2025)

5.1.1. Diseño

Entre los requerimientos funcionales solicitados por el cliente se encuentran los siguientes:

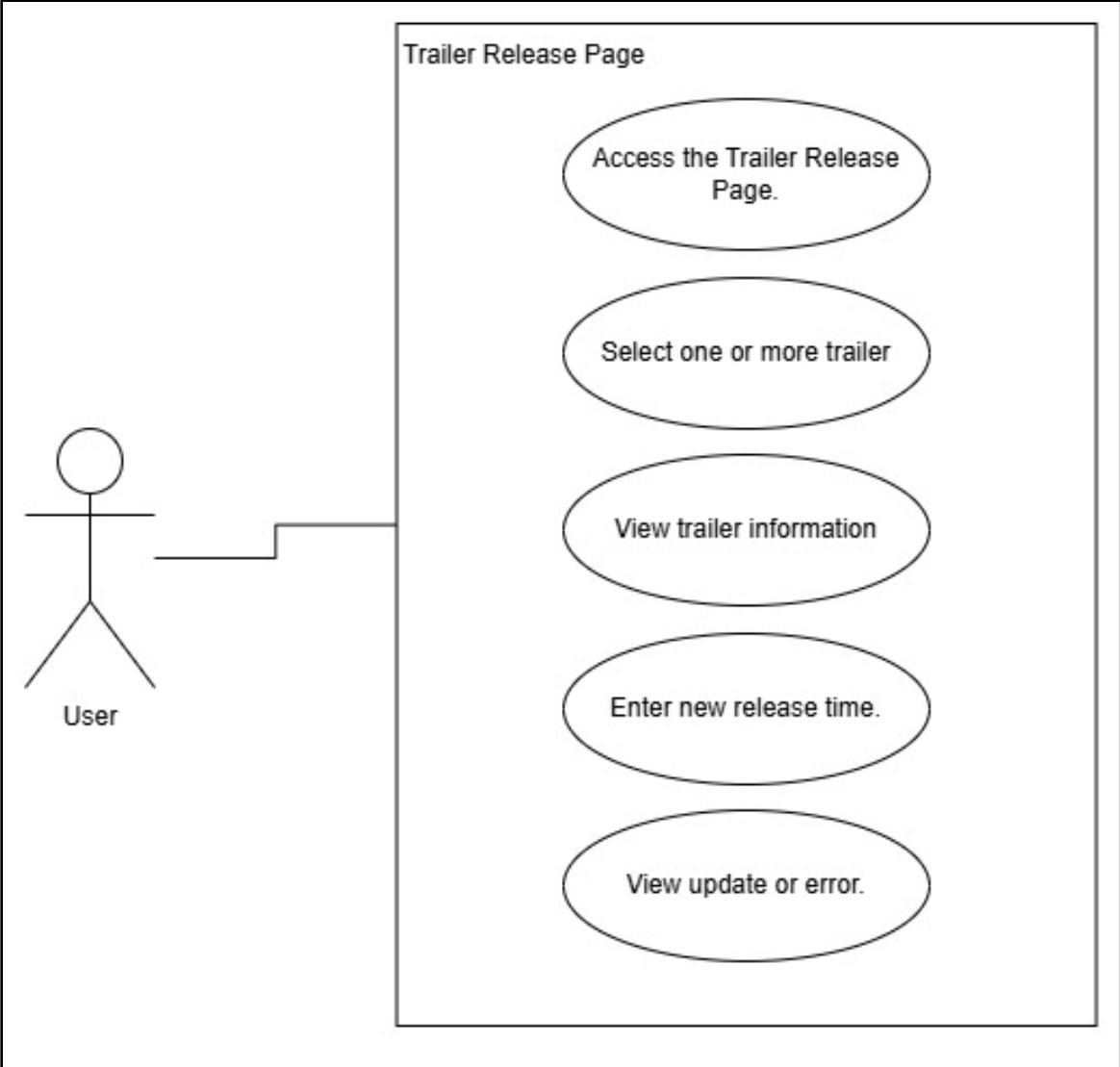
- **Actualización del Release Time:** La página debe permitir a los usuarios actualizar el tiempo de liberación de los trailers.
- **Visualización de Datos:** Mostrar información relevante como el ID del trailer, el estado actual, la hora de llegada y el tiempo de liberación actual.
- **Validación de Datos:** Implementar validaciones para asegurar que los datos ingresados sean correctos y estén dentro de los rangos permitidos.
- **Interfaz Intuitiva:** La interfaz debe ser fácil de usar, con botones claros y un diseño que facilite la navegación.

Para los requerimientos técnicos se encuentra imprescindible la creación de un comando personalizado MOCA que incluya consultas SQL para encontrar la información correspondiente, así como un comando que permita actualizar el tiempo de liberación de cada trailer.

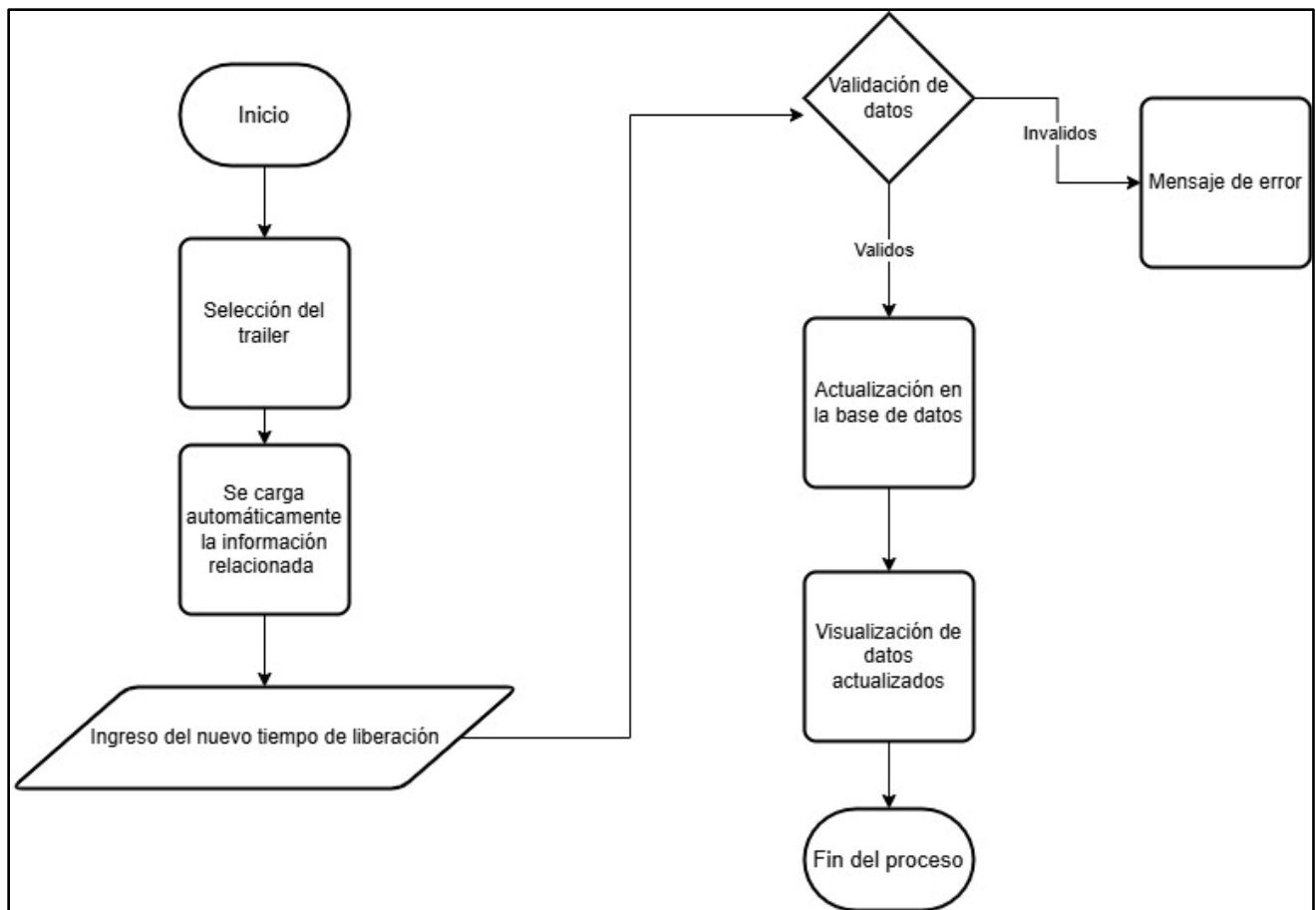
El desarrollo de la interfaz gráfica se lleva a cabo usando la extensión Page builder que facilita la lectura y ejecución de páginas definidas por usuario a través de la generación de un servicio API que se conecta a la interfaz web del WMS.

Diagrama de casos de uso y diagrama de flujo presentados al cliente:

Figura 5: Diagrama de casos de uso para la Trailer Release Page



Fuente: Los Autores (2025)

Figura 6: Diagrama de flujo para la Trailer Release Page

Fuente: Los Autores (2025)

5.1.2. Implementación

La creación de la interfaz comienza definiendo los campos se van a incluir, tales como:

- ID del envió entrante
- Numero de la orden entrante
- ID del cliente
- Numero de equipo de transporte
- Status del equipo de transporte

- Hora de llegada
- Tiempo de liberación
- Tiempo transcurrido (Turn Time)
- Fecha de inicio de la cita

El tiempo transcurrido representa la diferencia del tiempo de liberación con respecto a la hora de llegada o a la fecha de inicio de la cita dependiendo del caso, para determinar su valor se tomaron 3 casos posibles:

1. Si la hora de llegada es menor a la fecha de inicio de la cita, el tiempo transcurrido será diferencia entre la fecha de inicio de la cita y el tiempo de liberación.
2. Si la hora de llegada es mayor o igual a la fecha de inicio de la cita, el tiempo transcurrido será diferencia entre el tiempo de llegada y el tiempo de liberación.
3. Si no se programó una cita y no existe fecha de inicio de la cita, el tiempo transcurrido será diferencia entre la hora de llegada y el tiempo de liberación.

Además, la interfaz debe contar con un botón de acción que permita actualizar el tiempo de liberación. Al momento de validar el nuevo valor es preciso negar la entrada de una fecha que sea menor a la hora de llegada ya que no es coherente y generaría un valor negativo en el cálculo del tiempo transcurrido.

Desarrollo de Comandos SQL y MOCA

La lógica del comando MOCA que realiza la consulta requiere la aplicación de filtros para consultar la información apropiada, como por ejemplo el código del equipo de transporte, siendo únicamente de recibo. Por otro lado, con el esquema de la base de datos del sistema fue necesario relacionar 4 tablas que se asocian con el ID del trailer, el número de envió entrante y el ID de la cita programada. Una vez que la consulta sea correcta se procede a crear el comando como un componente de nivel "VAR" bajo el nombre de "list var receipt transport".

Figura 7: Lógica del comando "list var receipt transport "

```

278 [select rv.trknum,
279     rv.invnum,
280     rv.client_id,
281     trlr.trlr_id,
282     trlr.trlr_num,
283     trlr.trlr_stat,
284     trlr.arrdte,
285     trlr.pend_dte,
286     appt.start_dte,
287     case when trlr.arrdte is not null
288         and trlr.pend_dte is not null
289         and trlr.arrdte < appt.start_dte then CONCAT(CONCAT(DATEDIFF(MINUTE, appt.start_dte, trlr.pend_dte) / 60, ' hours, '), CONCAT(DATEDIFF(
MINUTE, appt.start_dte, trlr.pend_dte) % 60, ' minutes '))
290     when trlr.arrdte is not null
291     and trlr.pend_dte is not null
292     and trlr.arrdte >= appt.start_dte then CONCAT(CONCAT(DATEDIFF(MINUTE, trlr.arrdte, trlr.pend_dte) / 60, ' hours, '), CONCAT(DATEDIFF(MINUTE
, trlr.arrdte, trlr.pend_dte) % 60, ' minutes '))
293     when trlr.arrdte is not null
294     and trlr.pend_dte is not null
295     and appt.start_dte is null then CONCAT(CONCAT(DATEDIFF(MINUTE, trlr.arrdte, trlr.pend_dte) / 60, ' hours, '), CONCAT(DATEDIFF(MINUTE, trlr.
arrdte, trlr.pend_dte) % 60, ' minutes '))
296     else ''
297     end as turn_time
298 from trlr
299 join rovtrk
300     on trlr.trlr_id = rovtrk.trlr_id
301 join rovinv rv
302     on rovtrk.trknum = rv.trknum
303 left
304 join appt
305     on trlr.appt_id = appt.appt_id
306 where trlr.trlr_cod = /*#nobind*/ 'RCV' /*#bind*/
307 and stoloc_wh_id = nvl(@wh_id, @@wh_id)
308 and @trlr_num=raw

```

Fuente: Los Autores (2025)

Para la actualización del tiempo de liberación se creó otro comando que valida la información ingresada, primero al convertir el tiempo de liberación ingresado como una cadena de caracteres al formato de fecha, para luego compararlo con la hora de llegada y en el caso de que no se cumpla la validación devolver un error para informar al usuario. Otro factor importante es que la tabla relacionada a la información de los trailers no tiene por defecto un campo específico para el tiempo de liberación, por lo cual fue conveniente usar un campo vacío del que no se hace uso. Este comando también es un componente de nivel "VAR" y lleva el nombre de "change var transport reference".

Figura 8: Lógica del comando "change var transport reference "

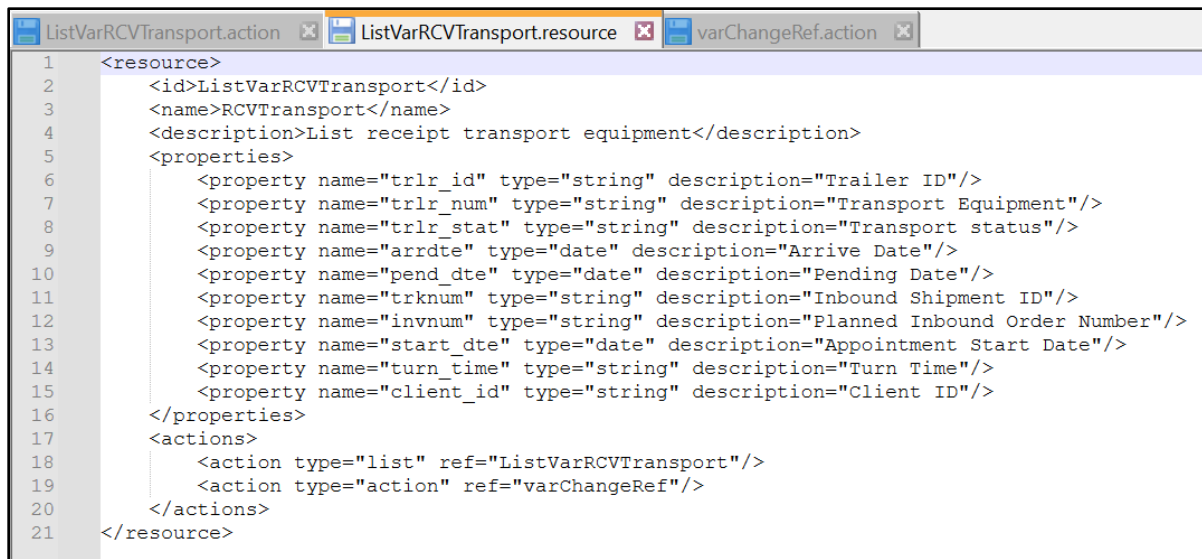
```
15 [select arrdte
16   from trlr
17   where trlr_id = @trlr_id]
18 |
19 [select @pend_dte:date release_time
20   from dual]
21 |
22 if (@release_time > @arrdte)
23 {
24   [update trlr
25     set pend_dte = @pend_dte:date
26     where trlr_id = @trlr_id]
27 }
28 else
29 {
30   set return status
31   where status = 8990910
32   and error_message = 'Release time cannot be before Check in time'
33 }]]>
34 </local-syntax>
35 </command>
```

Fuente: Los Autores (2025)

Continuando con la creación de la interfaz de programación de aplicaciones API, que se usa para comunicar dos sistemas, en este caso entre la base de datos MOCA/WMS y la interfaz gráfica Web UI, se consta de 3 tipos de archivos de definición:

- Archivos de definición de acciones (.action)
- Archivos de definición de mapeo de columnas (.json)
- Archivos de definición de recursos (.resource)

En el archivo .resource se incluye la representación del objeto que va interpretar la información recibida, por lo cual se definen los campos que se van a enlazar con la consulta a la base de datos además de incluir las acciones o funciones con las que contara la interfaz.

Figura 9: *.resource de la Trailer Release Page*

```
1 <resource>
2   <id>ListVarRCVTransport</id>
3   <name>RCVTransport</name>
4   <description>List receipt transport equipment</description>
5   <properties>
6     <property name="trlr_id" type="string" description="Trailer ID"/>
7     <property name="trlr_num" type="string" description="Transport Equipment"/>
8     <property name="trlr_stat" type="string" description="Transport status"/>
9     <property name="arrdte" type="date" description="Arrive Date"/>
10    <property name="pend_dte" type="date" description="Pending Date"/>
11    <property name="trknum" type="string" description="Inbound Shipment ID"/>
12    <property name="invnum" type="string" description="Planned Inbound Order Number"/>
13    <property name="start_dte" type="date" description="Appointment Start Date"/>
14    <property name="turn_time" type="string" description="Turn Time"/>
15    <property name="client_id" type="string" description="Client ID"/>
16  </properties>
17  <actions>
18    <action type="list" ref="ListVarRCVTransport"/>
19    <action type="action" ref="varChangeRef"/>
20  </actions>
21 </resource>
```

Fuente: Los Autores (2025)

Las acciones necesarias para llevar a cabo el funcionamiento de la interfaz son 2, una para listar y mostrar los resultados de la consulta a partir del método de comunicación GET, y otra para modificar el tiempo de liberación en los registros seleccionados a partir del método de comunicación PUT. En la acción de listar se ejecuta el comando “list var receipt transport” previamente creado, mientras que en la acción de modificar se ejecuta el comando “change var transport reference” enviando los argumentos requeridos.

Figura 10: .action para listar de la Trailer Release Page

```

1 <action>
2   <id>ListVarRCVTransport</id>
3   <name>RCVTransport</name>
4   <description>List receipt transport equipment</description>
5   <type>LIST</type>
6   <ws>
7     <method>GET</method>
8     <href>/ListVarRCVTransport</href>
9   </ws>
10  <listConfig>
11    <columnConfig>VarFilterRCVTransport</columnConfig>
12    <filterable/>
13    <pageable/>
14    <sortable/>
15  </listConfig>
16  <impl>
17    <command>list var receipt transport</command>
18  </impl>
19  <auth>
20    <opt>optOpen</opt>
21  </auth>
22 </action>

```

Fuente: Los Autores (2025)

Figura 11: .action para modificar de la Trailer Release Page

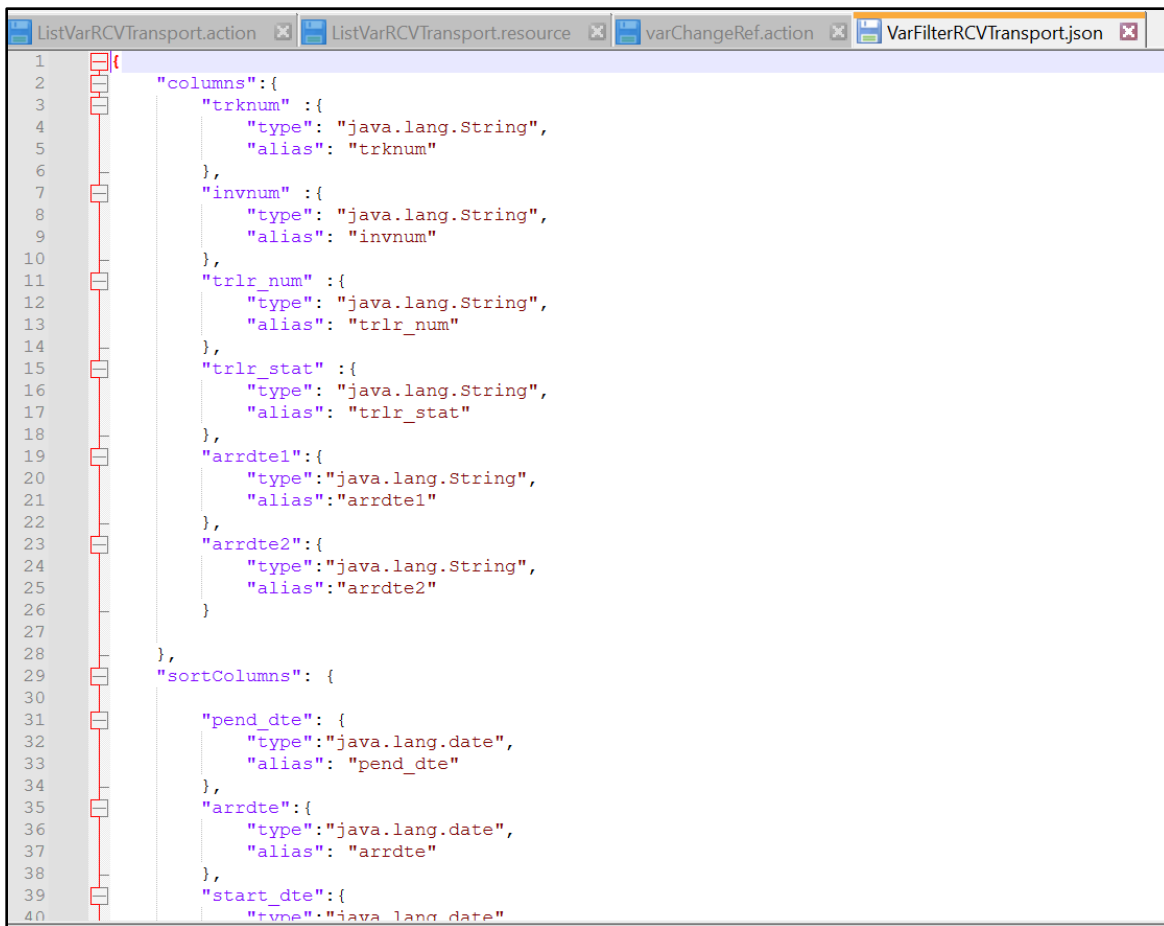
```

1 <action>
2   <id>varChangeRef</id>
3   <name>varChangeRef</name>
4   <description>Updates the transport reference number</description>
5   <type>ACTION</type>
6   <ws>
7     <method>PUT</method>
8     <href>/varChangeRef</href>
9   </ws>
10  <impl>
11    <command>change var transport reference</command>
12    <arguments>
13      <argument name="trlr_id" type="string" required="true" description="Id transport equipment"/>
14      <argument name="pend_dte" type="date" required="true" description="Pending Date"/>
15    </arguments>
16  </impl>
17  <auth>
18    <opt>optOpen</opt>
19  </auth>
20 </action>

```

Fuente: Los Autores (2025)

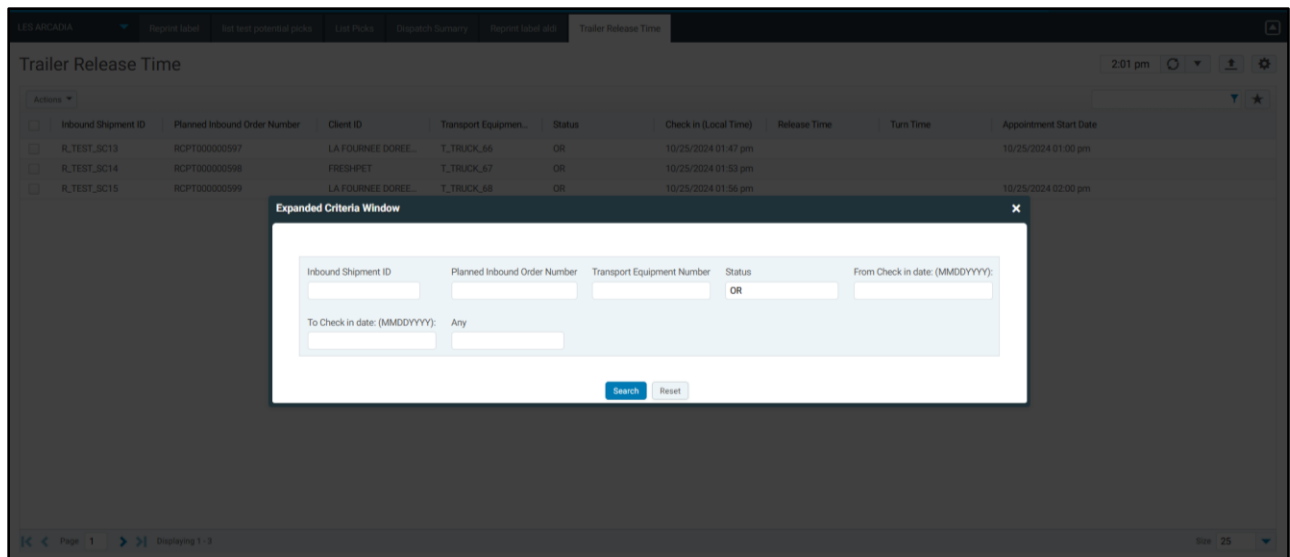
Por último, se crea el mapeo de columnas, donde se definen las columnas que se mostrarán en el grid y que servirán para el mantenimiento del filtrado y ordenamiento

Figura 12: *.json de la Trailer Release Page*

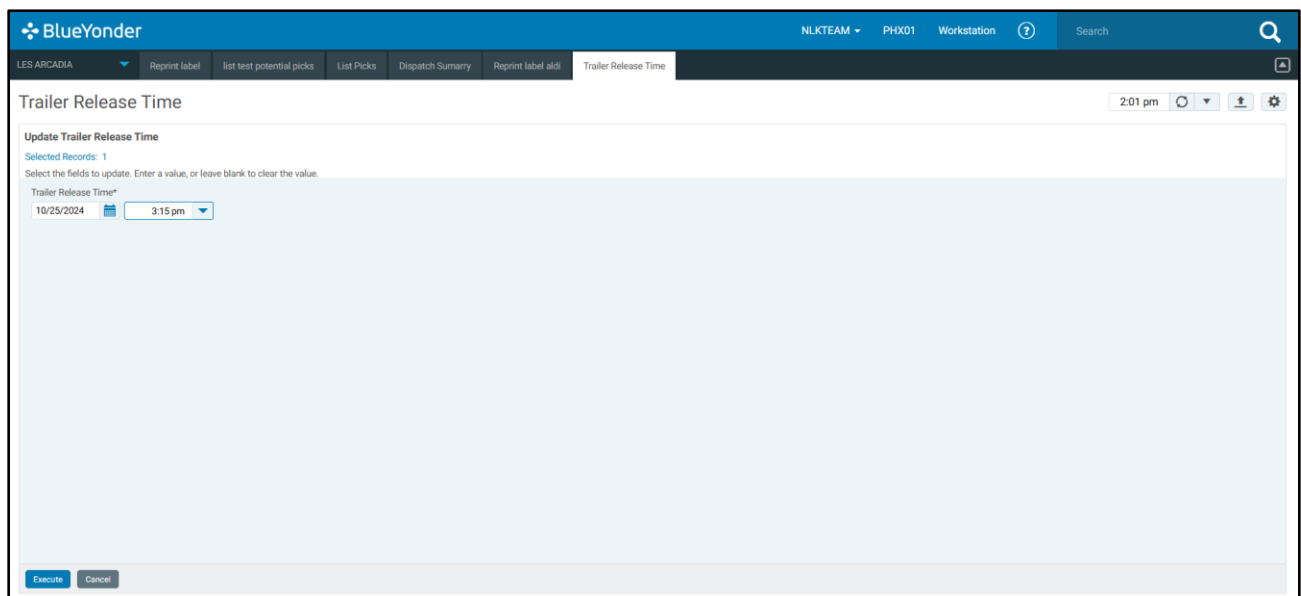
```
1  {
2  "columns":{
3      "trknum" :{
4          "type": "java.lang.String",
5          "alias": "trknum"
6      },
7      "invnum" :{
8          "type": "java.lang.String",
9          "alias": "invnum"
10     },
11     "trlr_num" :{
12         "type": "java.lang.String",
13         "alias": "trlr_num"
14     },
15     "trlr_stat" :{
16         "type": "java.lang.String",
17         "alias": "trlr_stat"
18     },
19     "arrdte1":{
20         "type": "java.lang.String",
21         "alias": "arrdte1"
22     },
23     "arrdte2":{
24         "type": "java.lang.String",
25         "alias": "arrdte2"
26     }
27 },
28 "sortColumns": {
29     "pend_dte": {
30         "type": "java.lang.date",
31         "alias": "pend_dte"
32     },
33     "arrdte":{
34         "type": "java.lang.date",
35         "alias": "arrdte"
36     },
37     "start_dte":{
38         "type": "java.lang.date"
39     }
40 }
```

Fuente: Los Autores (2025)

Una vez que se suben al sistema los 3 archivos de definición en conjunto, es posible crear el grid en la interfaz WEB UI del WMS, obteniendo como resultado la interfaz gráfica esperada, contando con las características de filtrado y modificación de registros correspondientes:

Figura 13: Criterios de búsqueda de la Trailer Release Page

Fuente: Los Autores (2025)

Figura 14: Acción para modificar tiempo de liberación

Fuente: Los Autores (2025)

Para completar la etapa de implementación se realizaron pruebas unitarias para asegurar que cada componente de la customización funcionara correctamente.

Figura 15: Matriz de pruebas de la Trailer Release Page

	A	B	C	D	E	F	G
1							
2	Client	Arcadia Cold		Testing Summary			
3	Date	3-Jun-24		Scenarios			
4			Review	Test In Review	0	0	
5			Pending	Test Pending Evaluation	0	0,00%	
6			Completed	Test Completed Successfully	10	100,00%	
7			Error	Test In Error	0	0,00%	
8			Quality	Test Pending Client Evaluation	0	0,00%	
9			Not Applicable	Test NOT Applies in the Process Model	0	0,00%	
10			Escalated	Cases Escalated for Blue Yonder	0	0	
11				Totals	10	100,00%	
12							
13	No.	Process	Status	Activity	Details	Environment	Result
14	1	Check filters functionality	Completed	Trailer Equipment ID: TRL0000298	Check the Trailer Equipment ID, Status and Trailer Equipment Number filters	NP1 WH: PHX01	successfully
15	2	Update trailer release time	Completed	Trailer Equipment ID: TRL0000345	Use the action Update trailer Release Time	NP1 WH: PHX01	successfully
16	3	Update trailer release time	Completed	Trailer Equipment ID: TRL0000597	Use the action Update trailer Release Time	NP1 WH: PHX01	successfully
17	4	Update trailer release time	Completed	Trailer Equipment ID: TRL0000610	Use the action Update trailer Release Time	NP1 WH: PHX01	successfully
18	5	Update trailer release time	Completed	Trailer Equipment ID: TRL0002877	Use the action Update trailer Release Time	NP1 WH: PHX01	successfully

Fuente: Los Autores (2025)

5.1.3. Capacitación y pruebas de usuario

Se elaboró un manual de usuario que explica cómo utilizar la **Trailer Release Page** y se realizaron sesiones con los usuarios, que interactuaron con la interfaz, realizando actualizaciones del tiempo de liberación y verificando que los cambios se reflejaran correctamente en el sistema.

5.1.4. Puesta en marcha (Go-Live)

Se estableció la fecha de instalación y por el horario operativo de los ambientes de producción se consideró prudente que la instalación se realizara en horas de la madrugada. Una vez instalada la customización, se validó en tiempo real que los comandos se cargaron en el sistema y las funcionalidades funcionaran correctamente.

5.1.5. Optimización y soporte

Se monitoreó el desempeño de la customización durante las primeras semanas de operación y se proporcionó soporte técnico continuo para resolver cualquier incidencia reportada por los usuarios.

Como nuevas modificaciones, se incluyó una nueva validación para que solo fuera visible el equipo de transporte que había llegado al almacén con una diferencia de las 24 horas al momento de consultar con el fin de reducir la información que no era imprescindible y mejorar los tiempos de carga. Además, se modificó el mensaje de error para guardarlo en la tabla correspondiente a errores del sistema, permitiendo obtener una mayor trazabilidad de posibles fallos.

5.1.6. Análisis de resultado

La customización de la **Trailer Release Page** utilizando **Page Builder** conlleva beneficios en la eficiencia operativa relacionada al manejo de trailers dentro del almacén, por medio del control de los tiempos que se programan para la llegada de un envío entrante y el tiempo que tarda un camión en ser liberado, es posible tener datos que apoyan estadísticas de rendimiento para cada tipo de equipo de transporte, el transportista implicado o los operarios que reciben la carga. De igual manera, la nueva validación de visualización de equipos de transporte ha optimizado los tiempos de carga de información, asegurando que los usuarios accedan solo a datos relevantes. La definición oportuna de requerimientos y expectativas por parte del cliente ha permitido proporcionar una interfaz más intuitiva, mayor trazabilidad de fallos y un soporte robusto para la mejora continua de la operación.

5.2. View inbound shipments

El View Inbound Shipment es una interfaz desarrollada con la extensión de Page Builder, la cual permite que el cliente pueda visualizar y gestionar sus envíos entrantes a las bodegas. En esta vista, se presenta una tabla con información clave sobre los embarques recibidos, en la cual se incluyen el número de embarque, el número de orden entrante, el número de línea, el número maestro del Bill of Lading, el identificador LPN, el número de lote y detalles del producto, como su número de ítem, entre otros.

Además, se incluyen datos del cliente asociado, el estado del embarque y fechas relevantes como la fecha de recepción, la fecha de inserción del registro, la fecha de finalización y la fecha de despacho. También cuenta con un filtro, que permite filtrar los resultados según criterios específicos como el cliente, fechas de recibo, fecha de cierre, fecha de despacho y número de Inbound Shipment, lo que facilita la búsqueda de información específica.

Esta vista es fundamental para el monitoreo y auditoría de la recepción de mercancías, asegurando el cumplimiento de los procesos logísticos y permitiendo una mejor gestión del inventario. Los usuarios pueden utilizarla para verificar discrepancias, rastrear embarques y tomar decisiones operativas basadas en datos en tiempo real, entre otros usos.

Figura 16: *View Inbound Shipment*

Inbound Shipment	Planned Inbound Cr.	Line	Mat Bld	LPN	Lot Number	LPN Quantity	Unit Quantity	Pallet Quantity	Item Number	Item Description	Client	Inbound Shipment S.	Received Date	Invoiced Date	Completion Date	Dispatch Date
ARMADA_CATCH01	RCF10000034	0002		0001200148000022	845474	1.0	40	1.0	CATCH15	MIX SPY BIRD-CHN S.	ARMADA	C	08-23-2024	08-23-2024	08-23-2024	08-26-2024
ARMADA_CATCH01	RCF10000034	0002		0001200148000022	845474	1.0	40	1.0	CATCH15	MIX SPY BIRD-CHN S.	ARMADA	C	08-23-2024	08-23-2024	08-23-2024	08-26-2024
ARMADA_CATCH01	RCF10000034	0002		0001200148000022	84546	1.0	40	1.0	CATCH15	MIX SPY BIRD-CHN S.	ARMADA	C	08-23-2024	08-23-2024	08-23-2024	08-26-2024
ARMADA_CATCH01	RCF10000034	0001		0001200148000022	845454	1.0	1	0.005	CATCH15	MIX SPY BIRD-CHN S.	ARMADA	C	08-20-2024	08-23-2024	08-23-2024	08-26-2024
INB_ARM14	0251400043	0003		0001200148000022	1234	1.0	54	1.0	001036	Potato, Hash Brown	ARMADA	C	08-20-2024	08-20-2024	08-20-2024	08-20-2024
INB_ARM3	0251400043	0003		0001200148000022	1234	1.0	72	1.000	001036	Potato, Hash Brown	ARMADA	C	08-20-2024	08-20-2024	08-20-2024	08-20-2024
INB_ARM3	0251400043	0001		0001200148000022	123	1.0	72	1.000	001036	Potato, Hash Brown	ARMADA	C	08-20-2024	08-20-2024	08-20-2024	08-20-2024
INB_ARM3	0251400043	0002		0001200148000022	123	1.0	72	1.000	001036	Potato, Hash Brown	ARMADA	C	08-20-2024	08-20-2024	08-20-2024	08-20-2024
INB_ARMADA	025140009	0001		0001200148000022	12	0.5	54	1.0	001058	Potato, Waffle Fries	ARMADA	C	10-08-2024	10-08-2024	10-08-2024	10-08-2024
INB_ARMADA	025140009	0001		0001200148000022	12	0.5	54	1.0	001058	Potato, Waffle Fries	ARMADA	C	10-08-2024	10-08-2024	10-08-2024	10-08-2024
INB_ARMADA_PRR...	RCF10000034	0002		0001200148000022	123	0.5	54	1.0	001058	Potato, Waffle Fries	ARMADA	C	10-08-2024	10-08-2024	10-08-2024	10-08-2024
INB_ARM15	RCF10000026	0001		0001200148000022	A	1.0	54	1.0	001036	Potato, Hash Brown	ARMADA	C	10-20-2024	10-20-2024	10-20-2024	10-20-2024
COMFEST03	COMFEST03	0002		0001200148000022	A	1.0	54	1.0	001036	Potato, Hash Brown	ARMADA	C	11-06-2024	11-06-2024	11-06-2024	09-19-2024
LPN1E5788	RCF10000044	0002		0001200148000022	123	1.0	54	1.0	001036	Potato, Hash Brown	ARMADA	R	11-06-2024	11-06-2024		
LPN1E5788	RCF10000044	0001		0001200148000022	123	1.0	54	1.0	107085	POTATO, WAFLE F...	ARMADA	R	11-06-2024	11-06-2024		
CATCH_R_01	RCF10000015	0001		0001200148000022	A	1.0	5	0.125	CATCH15	MIX SPY BIRD-CHN S.	ARMADA	R	11-14-2024	11-14-2024		
CATCH_R_01	RCF10000015	0001		0001200148000022	A	1.0	5	0.125	CATCH15	MIX SPY BIRD-CHN S.	ARMADA	R	11-14-2024	11-14-2024		
LPN1E5788	RCF10000044	0003		0001200148000022	---	0.5	54	1.0	001036	Potato, Waffle Fries	ARMADA	R	11-18-2024	11-18-2024		
CATCH_R_01	RCF10000015	0002		0001200148000022	A	1.0	5	0.0102	DOL_CATCH	DOL_ZDY TURKEY R.	ARMADA	R	11-19-2024	11-19-2024		
CATCH_R_01	RCF10000015	0002		0001200148000022	A	1.0	4	0.0140	DOL_CATCH	DOL_ZDY TURKEY R.	ARMADA	R	11-19-2024	11-19-2024		
CATCH_R_01	RCF10000015	0002		0001200148000022	A	1.0	5	0.0102	DOL_CATCH	DOL_ZDY TURKEY R.	ARMADA	R	11-19-2024	11-19-2024		
CATCH_R_01	RCF10000015	0002		0001200148000022	A	1.0	4	0.0096	DOL_CATCH	DOL_ZDY TURKEY R.	ARMADA	R	11-21-2024	11-21-2024		
INB_270424	RCF10000063	0002		0001200148000024	123	0.5	54	1.0	001058	Potato, Waffle Fries	ARMADA	C	01-02-2025	01-02-2025	01-02-2025	01-02-2025
INB_ARMADA_PRR...	RCF10000034	0001		0001200148000024	12	1.0	40	0.5	000105	POTATO, WAFLE F...	ARMADA	C	01-01-2025	01-01-2025	01-01-2025	01-01-2025
COMFEST02	COMFEST02	0001		0001200148000024	L10245/7097234E...	0.5	54	1.0	001036	Potato, Waffle Fries	ARMADA	C	01-01-2025	01-01-2025	01-01-2025	01-19-2024

Fuente: Los Autores (2025)

5.2.1. Diseño

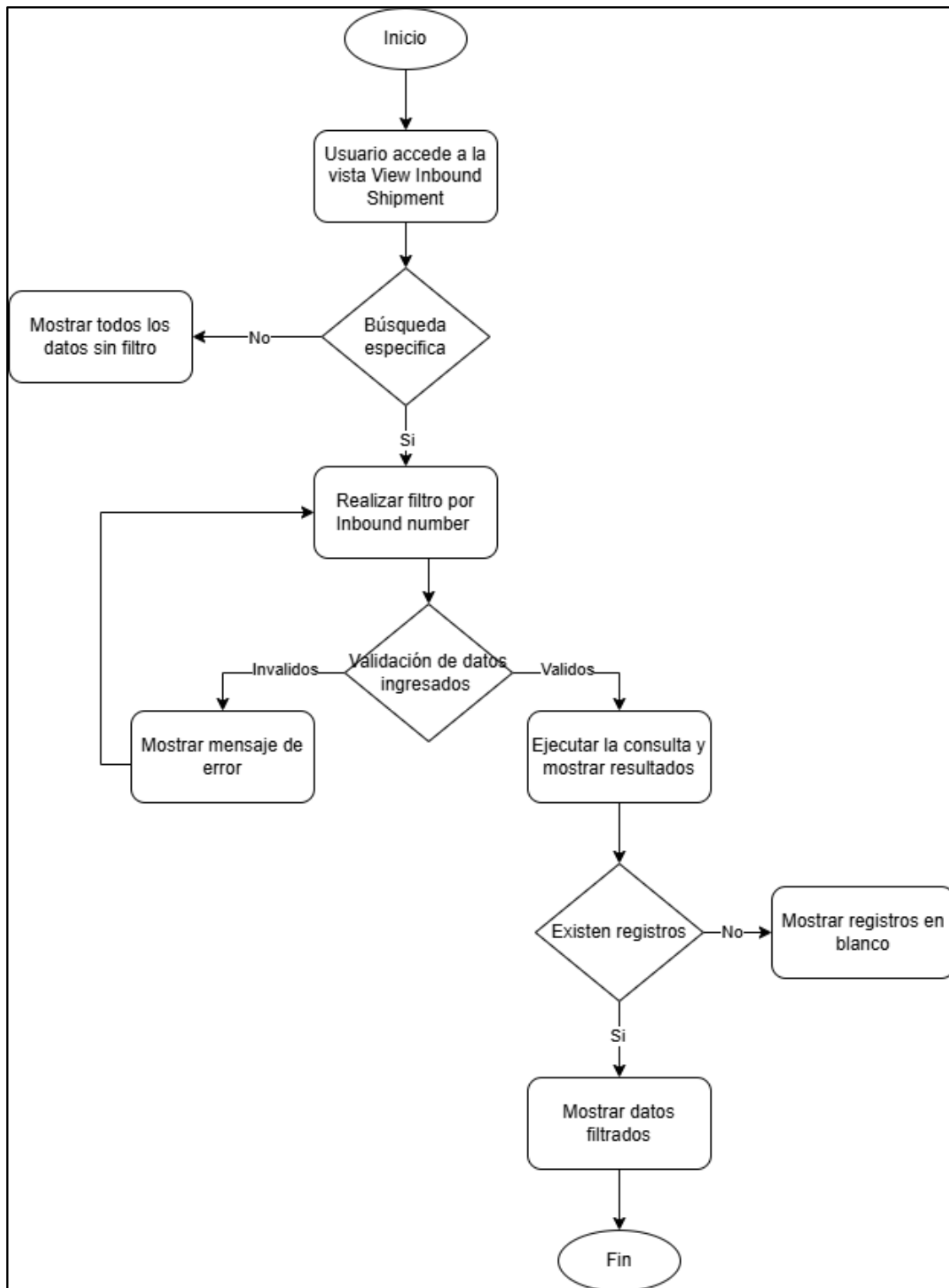
Entre los requerimientos funcionales solicitados por el cliente se encuentran los siguientes:

- **Visualización de envíos entrantes:** La página debe permitir a los usuarios consultar información detallada sobre los embarques recibidos, incluyendo número de embarque, número de orden entrante, línea, número maestro del Bill of Lading, identificador LPN, número de lote y detalles del producto.
- **Filtrado dinámico de datos:** Implementar un filtro que permita filtrar los registros por cliente, fechas de recepción, fecha de cierre, fecha de despacho y número de inbound shipment.
- **Validación de datos:** Asegurar que la información presentada sea precisa y consistente con los registros del sistema WMS, evitando datos duplicados o incorrectos.
- **Interfaz intuitiva y responsiva:** La interfaz debe ser clara y de fácil navegación, con opciones de búsqueda, ordenamiento y exportación de datos.

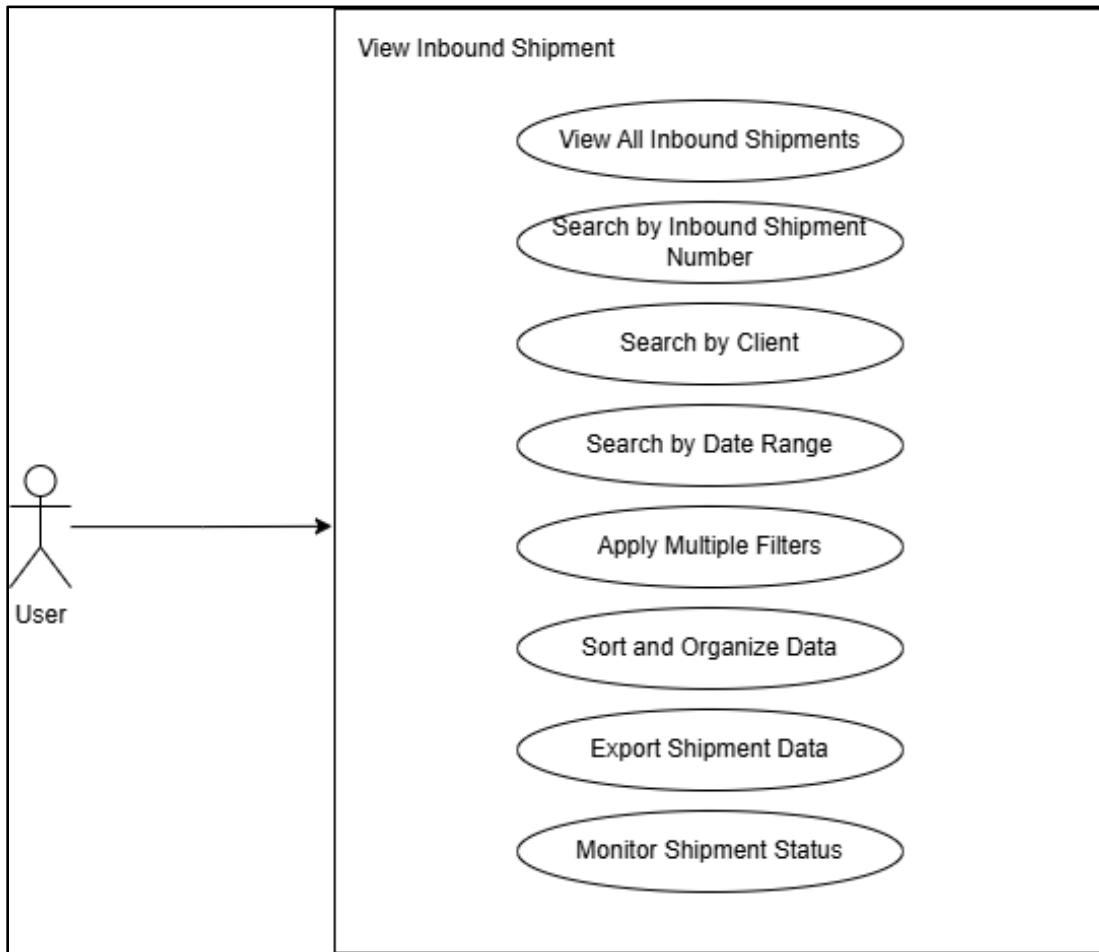
- Auditoría y monitoreo: Permitir a los usuarios auditar la recepción de mercancías, asegurando el cumplimiento de los procesos logísticos y detectando posibles discrepancias.
- Gestión y rastreo de embarques: Brindar herramientas para verificar el estado de los embarques y tomar decisiones operativas basadas en datos en tiempo real.

Requerimientos técnicos:

- Comando MOCA personalizado: Implementar un comando MOCA con consultas SQL optimizadas para extraer la información relevante de los embarques recibidos.
- Integración con el WMS: Asegurar la conectividad con la base de datos del WMS para obtener datos actualizados y confiables en tiempo real.
- Extensión Page Builder: Desarrollar la interfaz utilizando Page Builder, facilitando la personalización y ejecución de la vista dentro del sistema.

Figura 17: Diagrama de flujo para la View Inbound Shipment

Fuente: Los Autores (2025)

Figura 18: Diagrama de Casos de usos para View Inbound Shipment

Fuente: Los Autores (2025)

5.2.2. Implementación

Creación e implementación de la vista “View Inbound Shipment” se llevó a cabo utilizando la extensión Page Builder, lo cual permitió generar una interfaz web personalizada para el propio sistema del WMS. Dicho desarrollo se enfocó en proporcionar una herramienta eficiente para la consulta y gestión de los envíos entrantes a los almacenes

Análisis e interpretación de los requerimientos

Una vez proporcionados e identificados los requerimientos funcionales y técnico, fue muy importante interpretar como estos se traducían al desarrollo de la vista. No solo se trataba de satisfacer las especificaciones, sino de diseñar una solución eficiente, intuitiva y alineada con las necesidades operativas del cliente.

Dentro del análisis y la interpretación de los requerimientos funcionales y técnicos se identificaron los siguientes puntos:

- **Estructura de la información**

- Se identificó que toda la información debía organizarse en una tabla con columnas claramente definidas para lograr una fácil lectura de los datos a mostrar.
- Se priorizó la inclusión de datos esenciales como número de embarque, cliente, fechas clave y detalles del producto, asegurando que los usuarios tuvieran toda la información relevante en un solo lugar.
- Fue relevante entender que los datos estarían disponibles en tiempo real, los cuales serían obtenidos directamente de la base de datos del WMS.

- **Manejo de Volumen de Datos**

- Ya que los embarques pueden llegar a ser muy numerosos, se optó por incluir mecanismos de paginación y carga progresiva, permitiendo que los usuarios accedan rápidamente a la información sin sobrecargar el sistema.
- Se establecieron criterios de ordenamiento y búsqueda para que los usuarios puedan encontrar datos específicos sin necesidad de recorrer grandes volúmenes manualmente.

- **Interacción con el Usuario**
 - Se implementó un filtro con varios criterios los cuales permiten al usuario personalizar su consulta según su necesidad operativa.
 - Se añadieron opciones de exportación a formatos como Excel y PDF, facilitando la integración con otros procesos logísticos o reporte.
- **Seguridad y Control de Acceso**
 - Se establecieron restricciones de acceso para que solo los usuarios con permisos pudieran visualizar ciertos datos sensibles, como información de clientes y detalles de órdenes.

Desarrollo de Comandos SQL y MOCA

Para garantizar que la vista "View Inbound Shipments" pudiera mostrar los datos de manera eficiente y en tiempo real, se desarrolló el comando MOCA "list var inbound shipment to lpn", el cual realiza consultas SQL optimizadas para extraer información relevante desde la base de datos del WMS.

Estructura y Diseño del Comando

El desarrollo del comando se enfocó en tres aspectos principales:

1. Consulta eficiente de datos

Se diseñó una consulta SQL optimizada con filtros dinámicos.

2. Integración con Page Builder

El resultado debía presentarse en una tabla estructurada con información relevante.

3. Validaciones y Seguridad

Se implementaron mecanismos para evitar errores en los datos ingresados.

Definición de Parámetros de Entrada

El comando MOCA fue diseñado para aceptar múltiples parámetros de búsqueda, lo que permite filtrar la información de acuerdo con las necesidades del usuario. Algunos de los parámetros clave incluyen:

- `inbound_shipment_id` → Identificador único del embarque.
- `client_id` → Código del cliente para filtrar envíos específicos.
- `received_date` → Fecha de recepción del embarque.
- `lpn_number` → Identificador del pallet o contenedor dentro del WMS.
- `order_number` → Número de orden entrante vinculada al embarque.

Estos parámetros permiten que el usuario realice consultas específicas o visualice todos los embarques sin necesidad de aplicar filtros.

Figura 19: Lógica del comando "list var inbound shipment to lpn"

```
7 | publish data
8 | where sortOrderClause = nvl(@sortOrderClause, 'wh_id asc')
9 |
10 | publish data
11 | where filterByClause = nvl(@filterByClause,
12 |                          [1 = 1])
13 |    and wh_id = nvl(@wh_id, @@wh_id)
14 |
15 | if (@filterByClause like '%>=%')
16 | {
17 |     [select replace(tmp.filterByClause, '<=', '=') filterByClause
18 |       from (select replace(@filterByClause, '>=', '=') filterByClause
19 |            from dual) tmp] catch(-1403)
20 | }
21 |
```

Fuente: Los Autores (2025)

El archivo `.resource` juega un papel clave, ya que define la estructura del objeto que procesará la información obtenida. En este archivo se especifican los campos que estarán

vinculados a la consulta en la base de datos, además de incluir las acciones o funciones que estarán disponibles dentro de la interfaz para la manipulación de los datos.

Figura 20: .resource para Paga Builder “View Inbound Shipment”

```

1 <resource>
2   <id>varInbShipment</id>
3   <name>Inb Shipment to LPN</name>
4   <description>List Each Item for a Planned Inbound Order</description>
5   <properties>
6     <property name="wh_id" type="string" description="Warehouse"/>
7     <property name="trknum" type="string" description="Inbound Shipment"/>
8     <property name="invnum" type="string" description="Planned inbound order"/>
9     <property name="invlin" type="string" description="Line"/>
10    <property name="mst_bol" type="string" description="Load Document number"/>
11    <property name="lodnum" type="string" description="LPN"/>
12    <property name="lotnum" type="string" description="Lot Number"/>
13    <property name="no_pallet" type="string" description="Pallet Number"/>
14    <property name="no_case" type="string" description="Case Number"/>
15    <property name="no_lpn" type="string" description="LPN Number"/>
16    <property name="prtnum" type="string" description="Item Number"/>
17    <property name="lngdsc" type="string" description="Item description"/>
18    <property name="client_id" type="string" description="Client"/>
19    <property name="rcvtrk_stat" type="string" description="Inbound Shipment Status"/>
20    <property name="rcvdt" type="string" description="Received Date"/>
21    <property name="ins_dt" type="string" description="Inserted Date"/>
22    <property name="clsdt" type="string" description="Completion Date"/>
23    <property name="dispatch_dte" type="string" description="Dispatch Date"/>
24    <property name="ins_user_id" type="string" description="Inserted User"/>
25    <property name="rcvqty" type="integer" description="Quantity"/>
26    <property name="uomcod" type="string" description="Unit Of Measure"/>
27    <property name="subnum" type="string" description="Sub-LPN"/>
28    <property name="revlvl" type="string" description="Revision Level"/>
29    <property name="grswgt" type="string" description="Gross Weight"/>
30    <property name="netwgt" type="string" description="Net Weight"/>
31    <property name="catch_qty" type="string" description="Catch Quantity"/>
32  </properties>
33  <actions>
34    <action type="list" ref="varListInbShipment"/>
35  </actions>
36 </resource>

```

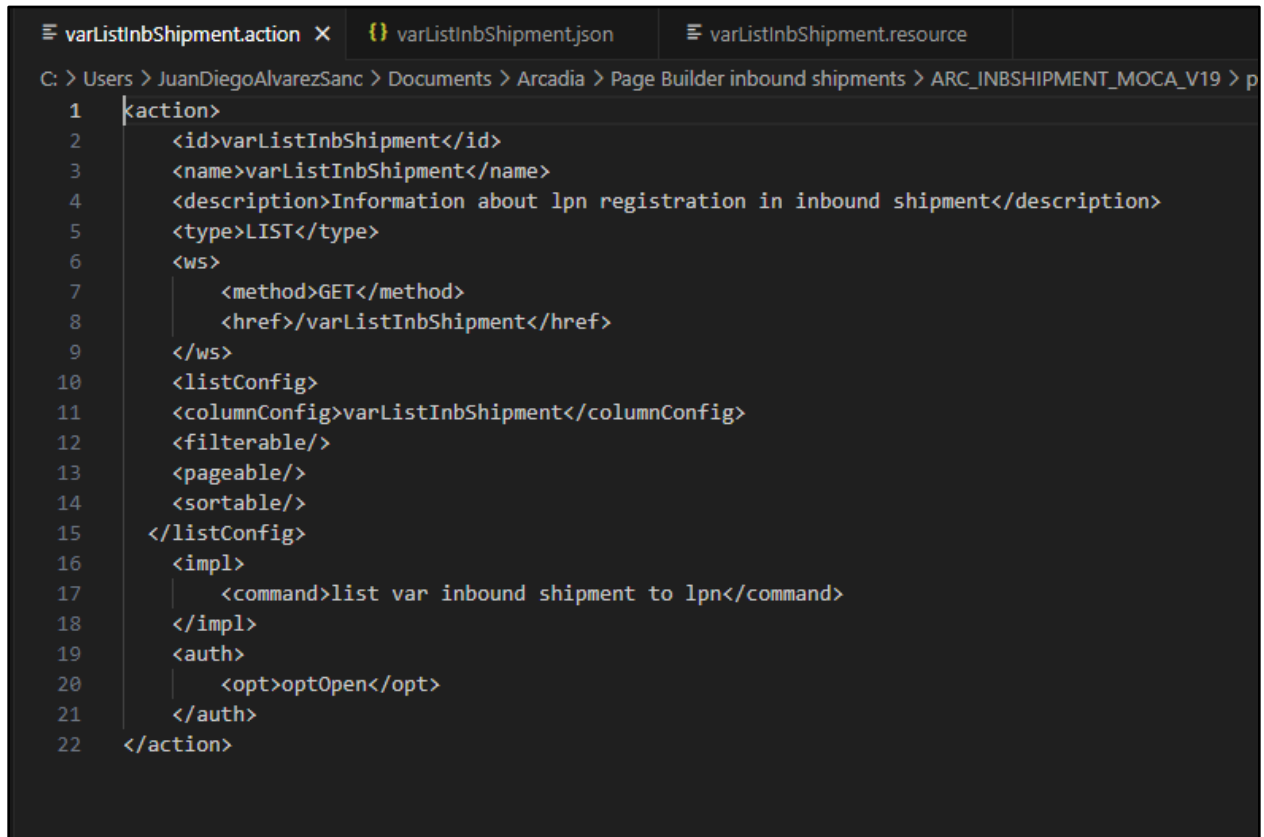
Fuente: Los Autores (2025)

El archivo .action es fundamental en la configuración de la API, ya que define las acciones que la interfaz podrá ejecutar al interactuar con la base de datos MOCA/WMS. Su propósito principal es servir como un puente entre la interfaz de usuario y los comandos que se ejecutarán en el sistema backend.

- Especifica qué operación se ejecutará cuando un usuario realice una acción en la interfaz.

- Permite mapear los datos de entrada y salida para garantizar que la información fluya correctamente entre la UI y la base de datos.
- Facilita la integración con comandos MOCA personalizados, permitiendo consultas y actualizaciones específicas en el sistema WMS.

Figura 21: *.action para Page Builder “View Inbound Shipment”*



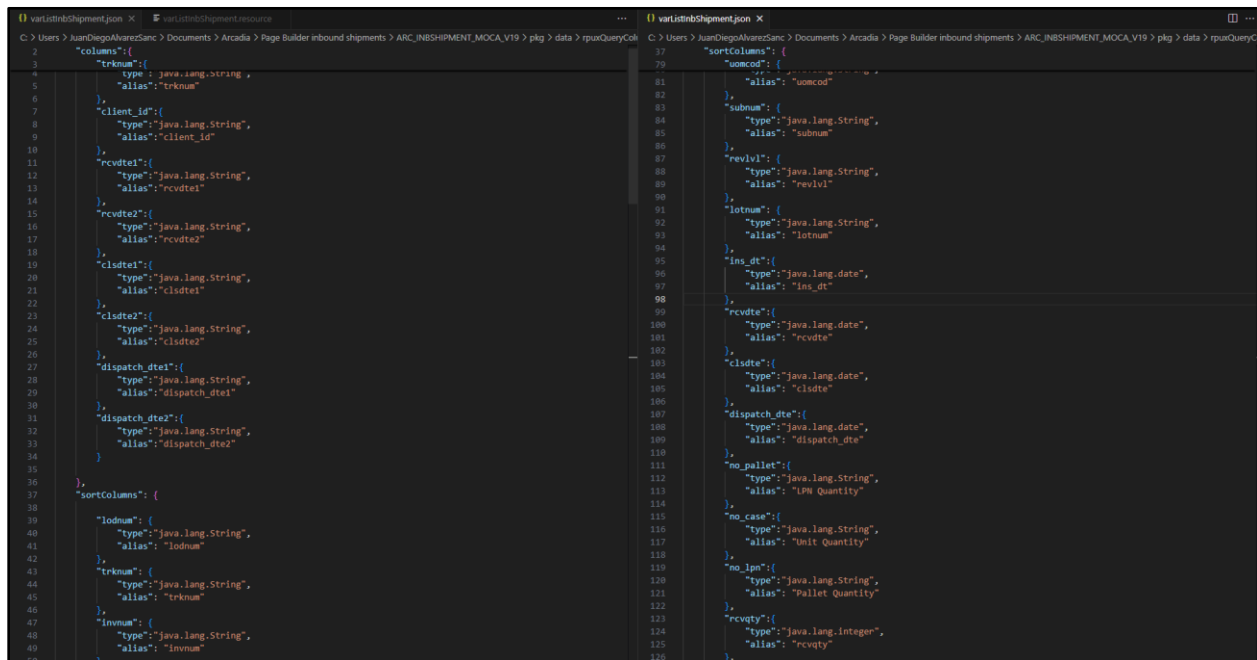
```
1 <action>
2   <id>varListInbShipment</id>
3   <name>varListInbShipment</name>
4   <description>Information about lpn registration in inbound shipment</description>
5   <type>LIST</type>
6   <ws>
7     <method>GET</method>
8     <href>/varListInbShipment</href>
9   </ws>
10  <listConfig>
11    <columnConfig>varListInbShipment</columnConfig>
12    <filterable/>
13    <pageable/>
14    <sortable/>
15  </listConfig>
16  <impl>
17    <command>list var inbound shipment to lpn</command>
18  </impl>
19  <auth>
20    <opt>optOpen</opt>
21  </auth>
22 </action>
```

Fuente: Los Autores (2025)

El archivo .json es muy importante ya que es el que se encarga de definir el mapeo de las columnas entre la base de datos y la interfaz web, asegurando la correcta interpretación y visualización de los datos. Especifica las columnas extraídas, sus tipos de datos y sus alias con los que se van a representar en la web. Adicionalmente establece criterios de ordenamiento permitiendo organizar los datos por atributos como número de tracking, cliente, fechas y

cantidades recibidas. Además, maneja valores numéricos clave como cantidad de productos (rcvqty), peso bruto (grswgt) y peso neto (netwgt).

Figura 22: *.json para Page Builder “View Inbound Shipment”*



The image shows two side-by-side JSON files in a code editor. The left file, named 'varlistInbShipment.json', defines columns for a data table. The right file, named 'varlistInbShipment.resource', defines sort columns for the same data table. Both files specify data types (e.g., java.lang.String, java.lang.Integer) and aliases for each field.

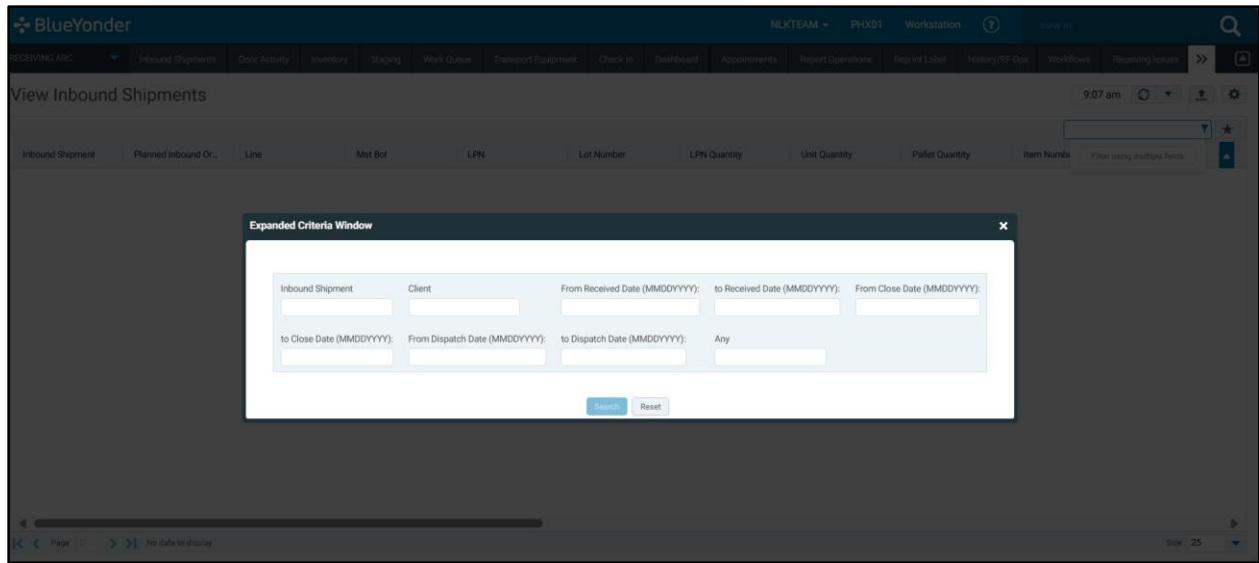
```
2      "columns": {
3        "trknum": {
4          "type": "java.lang.String",
5          "alias": "trknum"
6        },
7      },
8      "client_id": {
9        "type": "java.lang.String",
10       "alias": "client_id"
11     },
12     "rcvdte1": {
13       "type": "java.lang.String",
14       "alias": "rcvdte1"
15     },
16     "rcvdte2": {
17       "type": "java.lang.String",
18       "alias": "rcvdte2"
19     },
20     "clsdtel": {
21       "type": "java.lang.String",
22       "alias": "clsdtel"
23     },
24     "clsdtel2": {
25       "type": "java.lang.String",
26       "alias": "clsdtel2"
27     },
28     "dispatch_dte1": {
29       "type": "java.lang.String",
30       "alias": "dispatch_dte1"
31     },
32     "dispatch_dte2": {
33       "type": "java.lang.String",
34       "alias": "dispatch_dte2"
35     }
36   },
37   "sortColumns": {
38     "lodnum": {
39       "type": "java.lang.String",
40       "alias": "lodnum"
41     },
42     "trknum": {
43       "type": "java.lang.String",
44       "alias": "trknum"
45     },
46   },
47   "invnum": {
48     "type": "java.lang.String",
49     "alias": "invnum"
50   }
51 }
```

```
37   "sortColumns": {
38     "uomcod": {
39       "type": "java.lang.String",
40       "alias": "uomcod"
41     },
42     "subnum": {
43       "type": "java.lang.String",
44       "alias": "subnum"
45     },
46     "revlvl": {
47       "type": "java.lang.String",
48       "alias": "revlvl"
49     },
50     "lotnum": {
51       "type": "java.lang.String",
52       "alias": "lotnum"
53     },
54     "ins_dt": {
55       "type": "java.lang.date",
56       "alias": "ins_dt"
57     },
58     "rcvdte": {
59       "type": "java.lang.date",
60       "alias": "rcvdte"
61     },
62     "clsdte": {
63       "type": "java.lang.date",
64       "alias": "clsdte"
65     },
66     "dispatch_dte": {
67       "type": "java.lang.date",
68       "alias": "dispatch_dte"
69     },
70     "no_pallet": {
71       "type": "java.lang.String",
72       "alias": "IPN Quantity"
73     },
74     "no_case": {
75       "type": "java.lang.String",
76       "alias": "Unit Quantity"
77     },
78     "no_lpe": {
79       "type": "java.lang.String",
80       "alias": "Pallet Quantity"
81     },
82     "rcvqty": {
83       "type": "java.lang.Integer",
84       "alias": "rcvqty"
85     }
86   }
87 }
```

Fuente: Los Autores (2025)

Una vez que se suben al sistema los 3 archivos de definición en conjunto mediante un rollout con las rutas definidas para cada archivo, es posible crear el grid en la interfaz WEB UI del WMS, obteniendo como resultado la interfaz gráfica esperada, contando con las características de filtrado:

Figura 23: Criterios de búsqueda de la View Inbound Shipment



Fuente: Los Autores (2025)

Para completar la etapa de implementación se realizaron pruebas unitarias para asegurar que cada componente de la customización funcionara correctamente.

Figura 24: Matriz de pruebas del page View Inbound Shipment

Client	Arcadia Cold		Testing Summary			
Date	11-Apr-24		Scenarios			
	Review	Test In Review	0	0		
	Pending	Test Pending Evaluation	0	0,00%		
	Completed	Test Completed Successfully	4	100,00%		
	Error	Test In Error	0	0,00%		
	Quality	Test Pending Client Evaluation	0	0,00%		
	Not Applicable	Test NOT Applies in the Process Model	0	0,00%		
	Escalated	Cases Escalated for Blue Yonder	0	0		
	Totals		4	100,00%		

No.	Process	Status	Activity	Details	Environment	Result	Date
1	Check filters functionality	Completed	Filter test: filtering by inbound shipment	inbound shipment: INB_ARMADA_PRUEBA2	NP1 WH: PHX01	successfully	11/04/2024
2	Update trailer release time	Completed	Filter test: filtering by client	client: ARMADA	NP1 WH: PHX01	successfully	11/04/2024
3	Update trailer release time	Completed	Filter test: filter by date range	09/04/2024 - 11/04/2024	NP1 WH: PHX01	successfully	11/04/2024
4	Update trailer release time	Completed	Test of the relationship of data obtained with data in the database.	inbound shipment: INB_ARMADA_PRUEBA2	NP1 WH: PHX01	successfully	11/04/2024

Fuente: Los Autores (2025)

5.2.3. Capacitación y pruebas de usuario

En este aspecto de capacitación al usuario sobre la utilización del Page Builder se realizó un manual con las especificaciones técnicas y como usar dicho desarrollo. Adicionalmente se llevaron a cabo sesiones con el usuario en donde ellos pudieron interactuar y familiarizarse con el

desarrollo llevando a cabo consultas de inbound shipments, realizando búsquedas específicas mediante los filtros y verificando los datos obtenidos con los datos en la base de datos del sistema WMS.

5.2.4. Puesta en marcha (Go-Live)

Se dialogó con el cliente y se logró establecer fecha y hora para poder proceder con la instalación en los ambientes productivos, este proceso se llevó a cabo en horas de la madrugada en la cual la operación logística se encontraba detenida. Una vez finalizada la instalación de la customización, se realizó una validación en tiempo real de el comando que se cargó al sistema y la correcta funcionalidad de la vista.

5.2.5. Optimización y soporte

Para garantizar un buen funcionamiento del desarrollo se tuvo durante tres semanas en monitoreo y atendiendo cualquier inconveniente o mal funcionamiento del mismo, este proceso se llevó a cabo mediante el sistema de tickets de atlassian ‘Jira Service Management’ en donde el cliente reportaba estos incidentes o solicitudes de mejora. Una vez finalizado este seguimiento de tres semanas se procedió con la transición del desarrollo al equipo de soporte, el cual queda como responsable de cualquier incidente que se presente en adelante.

5.2.6. Análisis de resultado

El desarrollo de la customización de la “View Inbound Shipment” utilizando Page Builder ha demostrado una mejora significativa en la eficiencia de la operación de gestión de embarques entrantes. Gracias a esta implementación, a las consultas optimizadas y a los filtros dinámicos, los usuarios pueden acceder de forma rápida a información relevante sobre envíos recibidos, reduciendo el tiempo empleado en la búsqueda y validación de datos. La capacidad de visualizar detalles clave como número de embarque, cliente, fechas de recepción y cantidades recibidas ha permitido una mejor trazabilidad de los productos desde su llegada hasta su almacenamiento.

Además, la estandarización de formatos de datos ha optimizado la carga y procesamiento de información, garantizando tiempos de respuesta mucho más rápidos y una interfaz más intuitiva. La definición de requerimientos por parte del cliente ha sido clave para ofrecer una solución alineada a sus necesidades específicas, proporcionando una mayor visibilidad en la operación, minimizando errores y facilitando la toma de decisiones basada en datos en tiempo real.

5.3. Packing List Report

El Reporte Packing List es una solución personalizada desarrollada para el WMS, que permite generar un reporte detallado de las características físicas e información relevante correspondientes a cada pallet que se envía en una orden saliente. Este reporte es esencial para garantizar la trazabilidad y precisión en la gestión de inventarios, especialmente en operaciones logísticas que involucren productos de terceros.

El objetivo de esta customización es proporcionar un reporte que incluya información como el número de pallet, el código de producto, la cantidad, el peso bruto, el peso neto, el código de lote, la fecha de expiración y el código de retención (hold code), entre otros datos relevantes. Este reporte se genera a partir dos comandos MOCA que integran múltiples tablas de la base de datos del WMS.

Figura 25: Reporte Packing List

QTY SHIPPED		ITEM	DESCRIPTION		PALLET NUMBER	DATE CODE	HOLD CODE	GROSS WEIGHT	NET WEIGHT	CATCH WEIGHT
ORDERED QTY			LOT NUMBER	WAREHOUSE LOT						
		000000010000037407 7 PA	JMYD FC TKY SAU LINK 48CNT 12/38.			E				
20	CS		LOT01	----	L000000004B4	01-23-2025		652	576	
20	CS		LOT02	----	L000000004B5	02-06-2025		652	576	
20	CS		LOT03	----	L000000004B6	02-07-2025		652	576	
60	CS							1956	1,728	

November 06:42		Page 1 of 8
----------------	--	-------------

Fuente: Los Autores (2025)

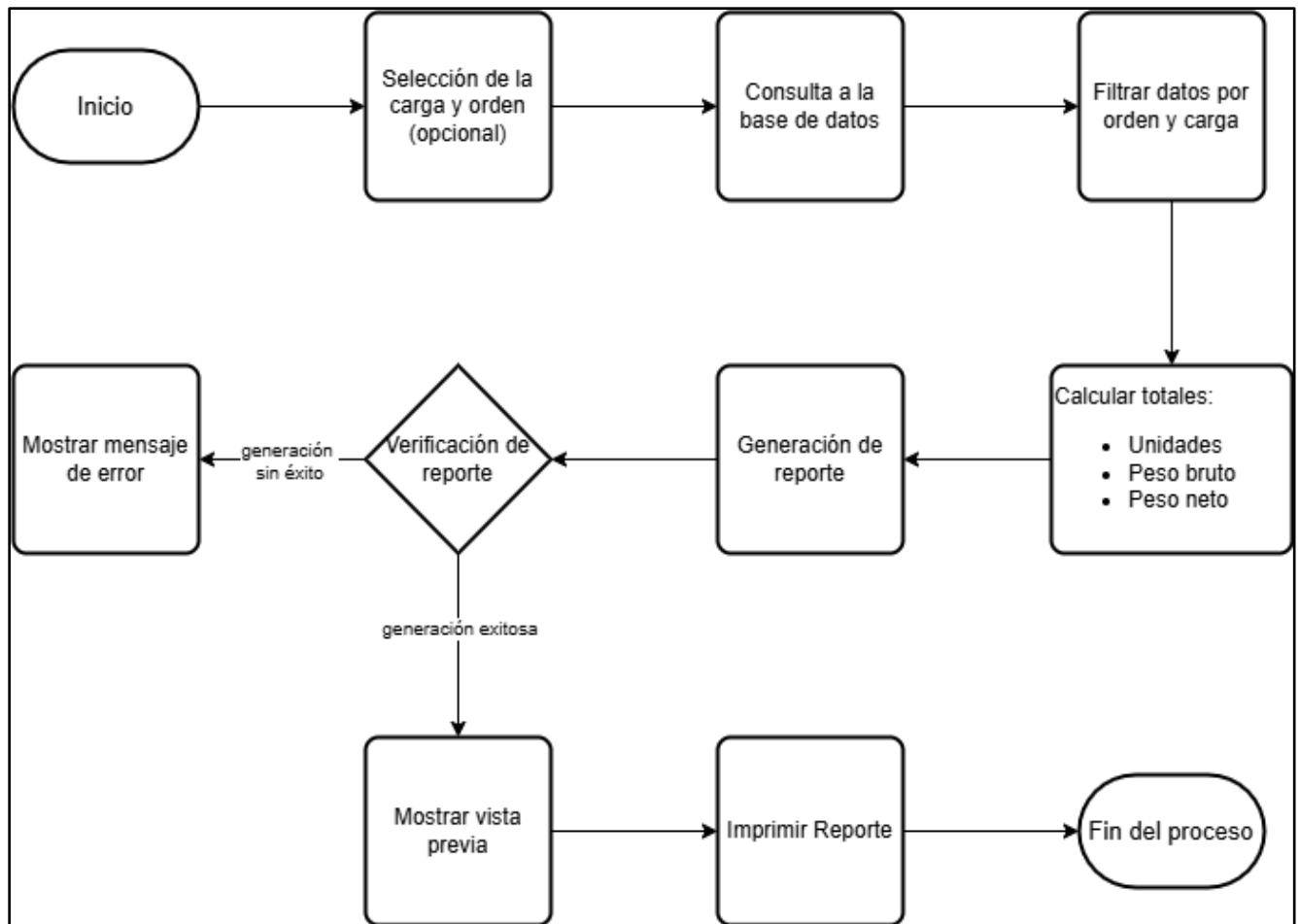
5.3.1. Diseño

Para los requerimientos funcionales, se establecieron características teniendo en cuenta que debía ser un reporte para transacciones de salida, tales como:

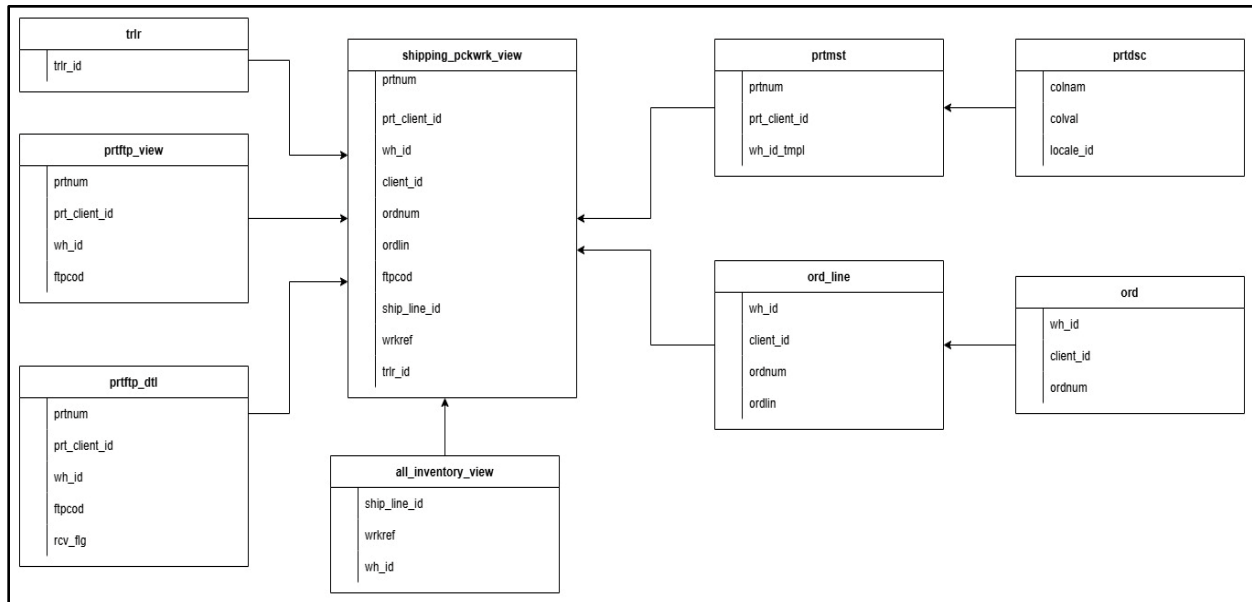
- La inclusión detallada de cada pallet surtido que forme parte de una orden saliente y que contara con una carga asignada para su envío.
- El encabezamiento incluirá toda la información de la orden saliente, como las direcciones de envío y recepción, así como las de fechas relevantes.
- El reporte debe permitir filtrar la información por número de orden y carga asignada, ya que una carga puede tener una o varias órdenes adjuntas. Además, si no se especifica la orden se deben visualizar todas las ordenes asociadas.
- El reporte debe calcular automáticamente las cantidades totales de unidades, así como el peso bruto y neto por cada ítem surtido.

En los requerimientos técnicos, se encuentra como herramienta para la generación de reportes, el software Jaspersoft, que permite una integración flexible con la base de datos MOCA/WMS, haciendo uso de uno o varios comandos que incluyen consultas SQL para obtener la información requerida. Además, el reporte debe integrarse sin problemas con el sistema WMS, proporcionando una vista previa de su contenido en la interfaz web UI antes de su impresión.

A continuación se presenta el diagrama de flujo para visualizar el proceso de generación del reporte, así como el diagrama Entidad-Relación, que sirve de guía para los comandos MOCA:

Figura 26: Diagrama de flujo para el reporte Packing List

Fuente: Los Autores (2025)

Figura 27: Diagrama Entidad-Relación para el reporte Packing List

Fuente: Los Autores (2025)

5.3.2. Implementación

Con base en la información solicitada, la consulta de la misma se dividió en 2 comandos, uno para obtener la información correspondiente al encabezado y otro para obtener los detalles de cada pallet. Esta decisión se toma debido a que en la estructura del reporte se incluye un subreporte, con el fin de listar los datos específicos por orden.

Para el comando del encabezado se relacionaron 14 tablas para buscar en su mayoría las direcciones e información adjunta al cliente (entidad que recibe los servicios del 3PL), comprador (entidad que adquiere los bienes) y al almacén que despacha la orden. Se usan como filtros opcionales el número de orden y la carga asociada. Una vez completada la consulta inicial se crea el comando como un componente de nivel “VAR” bajo el nombre de “list var report packing list header”.

Figura 28: Lógica del comando "list var report packing list header"

```

<!--[CDATA[
[select ord.stcust,
        cstmst.client_id,
        cstmst.adr_id,
        admst_cst.first_name as customer_first_name,
        admst_cst.last_name as customer_last_name,
        admst_cst.adrnam as customer_address_name,
        admst_cst.adrln1 as customer_address_1,
        admst_cst.adrln2 as customer_address_2,
        admst_cst.adrcty as customer_city,
        admst_cst.ctr_name as customer_country,
        admst_cst.adrstc as customer_state,
        admst_cst.adrpsz as customer_adrpsz,
        admst_cst.phnnum as customer_phone,
        admst_client.adrnam as client_address_name,
        admst_client.first_name as client_first_name,
        admst_client.last_name as client_last_name,
        admst_client.adrln1 as client_address_1,
        admst_client.adrln2 as client_address_2,
        admst_client.adrcty as client_city,
        admst_client.ctr_name as client_country,
        admst_client.adrstc as client_state,
        admst_client.adrpsz as client_adrpsz,
        admst_client.phnnum as client_phone,
        admst_wh.adrnam as wh_address_name,
        admst_wh.first_name as wh_first_name,
        admst_wh.last_name as wh_last_name,
        admst_wh.adrln1 as wh_address_1,
        admst_wh.adrln2 as wh_address_2,
        admst_wh.adrcty as wh_city,
        admst_wh.ctr_name as wh_country,
        admst_wh.adrstc as wh_adrstc,
        admst_wh.phnnum as wh_phone,
        admst_wh.email_adr as wh_email,
        admst_wh.adrstc as wh_adrstc,
        admst_wh.adrpsz as wh_adrpsz,
        ord.ordnum,
        ord.dlvnum,
        ord.cponum,
        to_char(ord.moddte, 'MM-DD-YYYY HH24:MI:SS') moddte,
        ord.dlvnum,
        ord.wh_id,
        ord_note.nottxt,
        car_move.carcod,
        car_move.car_move_id,
        trlr.trlr_num,
        appt.appt_id,
        appt.stoloc,
        trlr.trlr_broker,
        to_char(trlr.dispatch_dte, 'MM-DD-YYYY HH24:MI:SS') dispatch_dte,
        ord.rmanum,
        ord.csr_phnnum,
        stop.stop_seal,

```

Fuente: Los Autores (2025)

Por otro lado, en el comando del subreporte se relacionaron 11 tablas, donde se precisa buscar la configuración inicial de cada ítem recibido en la orden, esto con el fin de determinar, por ejemplo, qué tipo de fecha de seguimiento se debe asignar, o cuál es la unidad de medida usada para su almacenamiento, así como los pesos neto y bruto que se establecen por unidad recibida. Los parámetros que toma esta consulta son enviados directamente por la consulta del

reporte principal, siendo de igual manera el número de orden y la carga asociada. Este comando se crea como un componente de nivel “VAR”, denominado “list var report packing list detail”.

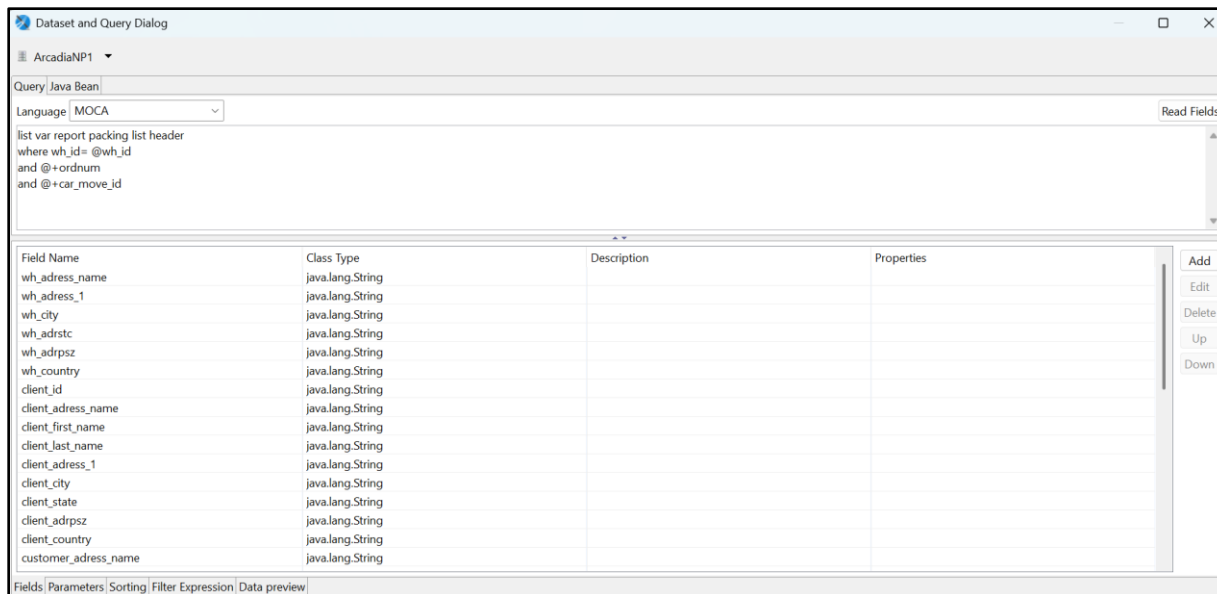
Figura 29: Lógica del comando "list var report packing list detail"

```
[select shipping_pckwrk_view.ftpcod,
shipping_pckwrk_view.pckqty,
shipping_pckwrk_view.car_move_id,
shipping_pckwrk_view.stop_id,
shipping_pckwrk_view.wh_id,
prtmst.prtnum,
prtmst.stkuom,
ptd.lngdsc,
ord_line.ordnum,
ord_line.ordqty,
ord_line.ordlin,
ord.dlvnum,
ord.cponum,
prtftp_view.grswgt,
prtftp_view.netwgt,
prtftp_view.rcvuom uomcod,
prtftp_dtl.untqty,
CASE WHEN (prtmst.catch_cod) is null THEN '0'
ELSE all_inventory_view.catch_qty
END AS catch_qty,
CASE WHEN (prtmst.catch_cod) is null THEN '0'
ELSE SUM(all_inventory_view.catch_qty)
END AS catch_ll,
all_inventory_view.lodnum,
all_inventory_view.lotnum,
all_inventory_view.revlvl,
sum(all_inventory_view.untqty) untqty_1,
sum(shipping_pckwrk_view.pckqty) inloadqty,
to_char(trlr.arrdte, 'MM-DD-YYYY') arrdte,
to_char(trlr.dispatch_dte, 'MM-DD-YYYY') dispatch_dte,
max(round(prtftp_view.grswgt * (shipping_pckwrk_view.pckqty), 3)) as
gross_weight,
max(round(prtftp_view.netwgt * (shipping_pckwrk_view.pckqty), 3)) as
net_weight,
CASE WHEN (prtmst.catch_cod) is not null THEN '0'
ELSE max(round(prtftp_view.grswgt * (shipping_pckwrk_view.pckqty)
, 3))
END AS grswgt_2,
CASE WHEN (prtmst.catch_cod) is not null THEN '0'
ELSE max(round(prtftp_view.netwgt * (shipping_pckwrk_view.pckqty)
, 3))
END AS netwgt_2,
CASE WHEN (prtmst.catch_cod) is not null THEN '0'
ELSE max(round(prtftp_view.grswgt * (shipping_pckwrk_view.pckqty)
, 3))
```

Fuente: Los Autores (2025)

Una vez que se integran los comandos al sistema, el siguiente paso es realizar el diseño del reporte en Jaspersoft. Para obtener los campos correspondientes, se ejecuta el comando como una consulta y se enlaza al ambiente de pruebas por medio de un adaptador de datos personalizado. Es importante aclarar que por buenas prácticas de desarrollo se recomienda usar un comando por reporte.

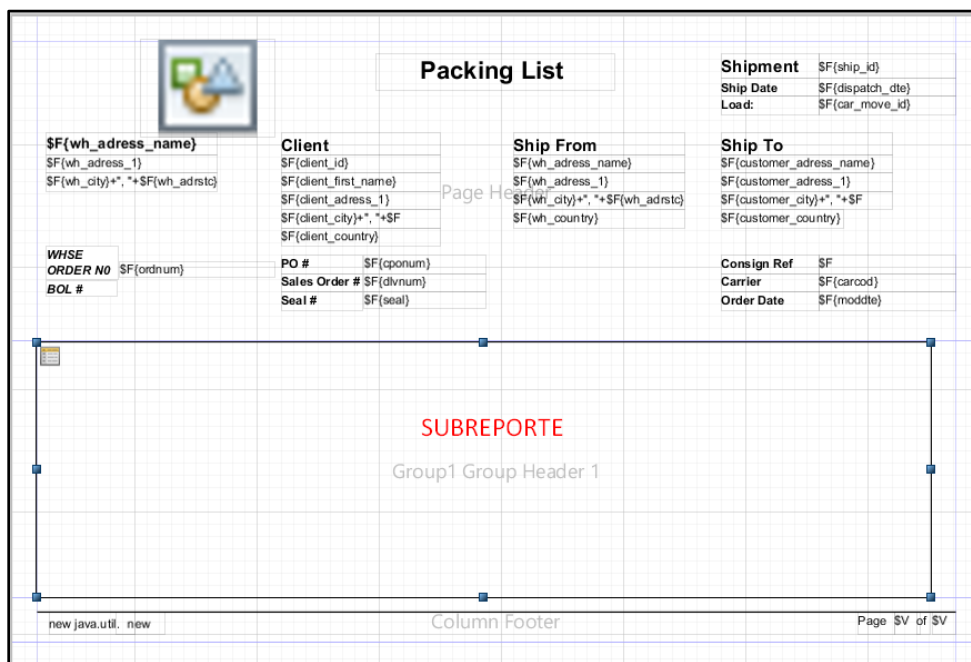
Figura 30: Editor de consulta Jaspersoft para el reporte Packing List



Fuente: Los Autores (2025)

En el reporte principal se incluye los datos del encabezado y un elemento de tipo subreporte, por el cual se envían los parámetros necesarios para el comando del reporte anidado.

Figura 31: Diseño del reporte principal Packing List en Jaspersoft



Fuente: Los Autores (2025)

El cálculo de las cantidades totales se lleva a cabo en el subreporte, agrupando los pallets recibidos por ítem y calculando en el compendio final las cantidades totales de la orden.

Figura 32: Diseño del sub-reporte Packing List en Jaspersoft

QTY SHIPPED	ITEM	DESCRIPTION		PALLET NUMBER	DATE CODE	HOLD CODE	GROSS WEIGHT	NET WEIGHT	CATCH WEIGHT
ORDERED QTY	LOT NUMBER	WAREHOUSE LOT							
\$F{prtnum}		\$F{ingdsc}		\$F{dte_code}					
\$F	\$F	\$F{lotnum}	\$F{revm}	\$F{lodnum}	\$F{expire_dte}	\$F	CEIL(\$F)	CEIL(\$F)	\$F
\$V	\$F						CEIL(\$V)	CEIL(\$V)	\$V
PALLET QTY: \$F{cont}				CEIL(\$V {sumgrswg	CEIL(\$V {sumnetwg	\$V {Sumtotalc			

Fuente: Los Autores (2025)

Con el diseño del reporte finalizado y compilado, se ubican los archivos de tipo. jrxml en las rutas del WMS destinadas al almacenamiento de reportes. Luego, se define el identificador de la operación en la interfaz web UI, donde será posible generar una vista previa e imprimirlo mediante una impresora previamente configurada.

Figura 33: Vista previa del reporte Packing List en la interfaz web UI

The screenshot shows a web interface for generating a Packing List report. The report header includes the Arcadia logo and client information: Arcadia Cold (13100 W. Joe R. Ramirez, El Mirage, AZ, 85335) and Client: TYSON FOODS INC (ATTN: WAREHOUSE, SPRINGDALE, AR, 72785, USA). It also lists ship from and to details, including Arcadia Cold Phoenix, LLC and TOLLESON DISTRIBUTION. A table of shipped items is displayed with the following data:

QTY SHIPPED	ITEM	DESCRIPTION	PALLET NUMBER	DATE CODE	HOLD CODE	GROSS WEIGHT	NET WEIGHT	CATCH WEIGHT
20	CS	LOT119	L000000048P	01-19-2025		652	576	
20	CS	LOT4	L0000000487	01-27-2024		652	576	
20	CS	LOT09	L0000000488	01-26-2024		652	576	
20	CS	LOT06	L0000000489	02-16-2024		652	576	

Fuente: Los Autores (2025)

Para completar la etapa de implementación, se previsualiza el reporte con distintos escenarios de salida donde se comprueban la concordancia y precisión de los datos.

Figura 34: Matriz de pruebas del reporte Packing List

	A	B	C	D	E	F	G	
1								
2	Client	Arcadia Cold	Testing Summary					
3	Date	12-Apr-24	Scenarios					
4			Review	Test In Review	0	0		
5			Pending	Test Pending Evaluation	0	0,00%		
6			Completed	Test Completed Successfully	9	100,00%		
7			Error	Test In Error	0	0,00%		
8			Quality	Test Pending Client Evaluation	0	0,00%		
9			Not Applicable	Test NOT Applies in the Process Model	0	0,00%		
10			Escalated	Cases Escalated for Blue Yonder	0	0		
11			Totals		9	100,00%		
13	No.	Process	Status	Activity	Details	Environment	Result	
14	1	Report Packing list	Completed	LOAD: LOADALD102 Order: ORD000000K ORD000000 ORDPATL10001186 ORDPATL10001187 Shipments: 4	Lines and Items: 0001-051835 0001-051835 0001-11911 0001-700885	NP1 WH: PHX01	Information is seen correctly	
15	2	Report Packing list	Completed	LOAD: LOAD_ALD1 Order: ORDALD10001 ORDALD1002 Shipments: 2	Lines and Items: 0001-700885 0001-11911	NP1 WH: PHX01	Information is seen correctly	
16	3	Report Packing list	Completed	Load: LD00000555 Order: ORDPATL10003008	Items: 47784 1CS=8.5 8092 1CS=13.2	PR2	Information is seen correctly	
17	Modifications							
18	4	Report Packing list	Completed	LOADBOL01	LPN's correction Item group	NP1	Information is seen correctly	
19	5	Report Packing list	Completed	LOPEZDL01	LPN's correction Item group	NP1	Information is seen correctly	
20	6	Report Packing list	Completed	Load: LD00000487 Shipments: SID0002078/SID0001699	Weights modification - no	NP1	Information is as expected	
21				Orders: OPHX10001065/OPHX10001142				
22	7	Report Packing list	Completed	Load: LOADCATCH Order: ORDCATCH	Landscape format / Weights modification - catch item	NP1	Information is as expected	

Fuente: Los Autores (2025)

5.3.3. Capacitación y pruebas de usuario

Se realizaron sesiones de capacitación con los usuarios finales para familiarizarlos con el nuevo reporte, generando una vista previa y verificando que los datos fueran precisos y completos. Principalmente, se mostró cómo se reflejaría la información para distintos órdenes salientes. Por ejemplo, como se vería el inventario que incluía cantidad de captura (catch quantity), como se determinaba la fecha del pallet según el código de fecha y como se agrupaban más de dos órdenes en el mismo reporte, entre otros aspectos. Además, se elaboró un manual de usuario que explica cómo generar y utilizar el reporte desde la interfaz web UI.

5.3.4. Puesta en marcha (Go-Live)

Para la instalación de la customización en los ambientes de producción, se estableció una fecha fuera del horario operativo. Una vez confirmada la instalación, se procede a validar en tiempo real que todos los componentes se hayan integrado correctamente al sistema. Asimismo, se realizó una prueba de visualización en cada uno de los seis ambientes productivos.

5.3.5. Optimización y soporte

Se identificaron posibles áreas de mejora en el apartado visual del reporte, como el número de caracteres permitido al mostrar LPNs o descripciones de ítems, puesto que en algunos casos la información aparecía incompleta. También se modificó el redondeo y la cantidad de decimales se mostraba en el peso neto, el peso bruto y la cantidad captura, con el fin de obtener una medición estandarizada.

Para actualizaciones posteriores y según necesidades identificadas, se incluyeron la inclusión del código de fecha, el código de retención y un cambio de formato en las fechas de cada pallet, así como la reestructuración del diseño visual para mostrar la nueva información.

5.3.6. Análisis de resultado

La implementación de la customización del **reporte Packing List** tuvo un impacto significativo en el monitoreo y la visibilidad de los envíos salientes, ya que no se contaba con una vista detallada que agrupara la información de cada orden con su respectivo inventario surtido. Como resultado, era común que la gestión se dificultara o que los tiempos de consolidación se alargaran para completar una carga. De igual modo, se encontraban inconsistencias en la trazabilidad con mayor frecuencia, debido a que la información relevante estaba dispersa en diferentes menús y pantallas del sistema.

Este reporte logró mejorar la precisión y transparencia en la planeación de envíos salientes al permitir el monitoreo en tiempo real del estado de una orden surtida y su carga asociada. Asimismo, la inclusión de una nueva documentación estructurada y confiable, fortaleció la trazabilidad que los clientes mantienen con sus compradores.

5.4. Bill Of Landing Report

El reporte Bill Of Landing es una solución implementada al WMS muy esencial ya que este es un documento legal y logístico que certifica el envío de mercancía de un remitente a un destinatario a través de un transportista. Su función principal es actuar como un comprobante de embarque, estableciendo los términos del transporte y asegurando que los productos sean entregados en la cantidad, peso y condiciones acordadas en este reporte. Además, facilita llevar la trazabilidad clara del pedido, ya que incluye detalles importantes como lo son el número de orden, descripción del producto, pesos y lotes.

Es por esto que el desarrollo de este reporte fue fundamental para satisfacer las necesidades del cliente, ya que este al manejar un flujo alto de envíos de mercancía, para el era fundamental que se lograra optimizar y automatizar la generación de estos reportes. La automatización aparte de agilizar el proceso logístico también evita que se cometan errores humanos por parte de los operarios, esto mejora la trazabilidad y garantiza que cada envío cuente con los datos correcto y toda su documentación. Con esto se buscó optimizar los tiempos y fortalecer los controles de calidad ya que esto facilita la gestión eficiente de los embarques, asegurando así una operación fluida y sin errores.

- Filtrado de información por número de orden y carga asignada, permitiendo visualizar todas las órdenes asociadas a una misma carga en caso de no especificar un pedido individual.
- Cálculo automático de totales, incluyendo el número total de unidades, peso bruto y peso neto de la carga, reduciendo errores manuales en el proceso.

En los requerimientos técnicos, se definió el uso de herramientas que permitan la generación y automatización del reporte de manera eficiente. Para esto, se utilizó el software Jaspersoft para establecer el diseño del reporte y generar el formato .jrxml asegurando compatibilidad con los sistemas existentes. La integración a la base de datos se realizará mediante consultas SQL, permitiendo extraer los datos de pedidos y cargas, además el reporte debe ser accesible desde la interfaz web del sistema, proporcionando una vista previa antes de su impresión o envío, garantizando así la validación de la información antes de su distribución.

Figura 36: Diagrama de flujo para el reporte Bill Of Landing

Fuente: Los Autores (2025)

5.4.2. Implementación

Con base en la información solicitada, la consulta se dividió en dos comandos MOCA, uno para obtener la información del encabezado del reporte y otro para obtener los detalles de la carga y los productos. Esta separación se realizó debido a que la estructura del reporte incluye un subreporte, lo que permite listar los productos de manera organizada por cada orden o carga.

Para la consulta del encabezado, se relacionaron 16 tablas, principalmente para recuperar información sobre el cliente (entidad que recibe la mercancía), el comprador (entidad que adquiere los bienes) y el almacén desde donde se despacha la orden. Como filtros opcionales, se permite consultar por número de orden o número de carga, asegurando flexibilidad en la generación del reporte.

En el caso de la consulta de detalles, se incluyeron todas las líneas de productos despachados, especificando cantidades, pesos, volumen y cualquier instrucción especial. Una vez validada la estructura de la información, ambos comandos fueron configurados como componentes de nivel "VAR" dentro del sistema, bajo los nombres "list var report bill of lading header" para el encabezado y "list var report bill of lading detail" para los detalles

Figura 37: Lógica del comando "list var report bill of lading"

```

40     ord_cponum,
41     to_char(ord_addtce, 'MM-DD-YYYY') as modtce,
42     car_move.carcod,
43     car_move.car_move_id,
44     trlr.trlr_num,
45     appt.appt_id,
46     appt.stoloc,
47     CASE WHEN ord_rmanum IS NULL
48           OR ord_rmanum = '' THEN trlr.trlr_seall
49           ELSE ord_rmanum
50     END AS seall,
51     ord.csr_phnum,
52     trlr.trlr_broker,
53     to_char(trlr.dispatch_dte, 'MM-DD-YYYY HH24:MI:SS') dispatch_dte,
54     ord.wh_id,
55     stop.stop_seal,
56     trlr.min_trlr_axle_pos temp_n,
57     trlr.max_trlr_axle_pos temp_m,
58     trlr.front_axle_pos temp_t,
59     trlr.mod_usr_id,
60     CASE WHEN ord_cponum IS NULL THEN SUBSTR(ord_note.nottxt, INSTR(ord_note.nottxt, 'SPE: PO # - ') + LENGTH('SPE: PO # - '))
61     ELSE NULL
62     END AS nottxt
63 from trlr
64 inner
65 join car_move
66   on trlr.trlr_id = car_move.trlr_id
67 inner
68 join stop
69   on car_move.car_move_id = stop.car_move_id
70 inner
71 join shipment
72   on stop.stop_id = shipment.stop_id
73 inner
74 join shipment_line
75   on shipment.ship_id = shipment_line.ship_id
76   and shipment.wh_id = shipment_line.wh_id
77 inner
78 join ord_line
79   on shipment_line.wh_id = ord_line.wh_id
80   and shipment_line.client_id = ord_line.client_id
81   and shipment_line.ordnum = ord_line.ordnum
82   and shipment_line.ordlin = ord_line.ordlin
83 inner

```

Fuente: Los Autores (2025)

Figura 38: Lógica del comando "list var detail report bill of lading"

```

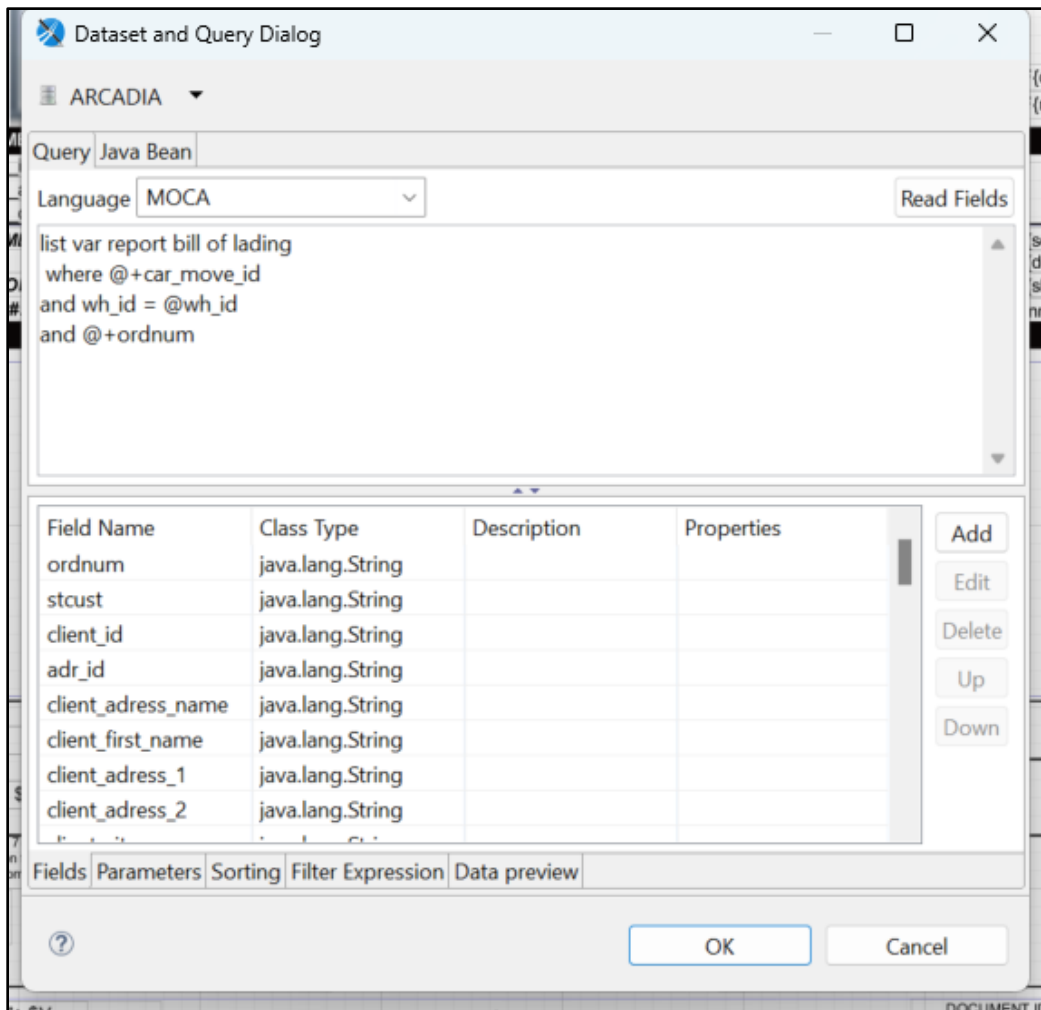
18 [select shipping_pckwrk_view.ftpcod,
19       shipping_pckwrk_view.pckqty,
20       shipping_pckwrk_view.car_move_id,
21       shipping_pckwrk_view.carcod,
22       shipping_pckwrk_view.stop_id,
23       shipping_pckwrk_view.wh_id,
24       prtmst.prtnum,
25       prtmst.stkuom,
26       ptd.lngdsc,
27       ord_line.ordnum,
28       ord_line.ordqty,
29       ord_line.ordlin,
30       prtftp_view.grswt,
31       prtftp_view.netwt,
32       all_inventory_view.lotnum,
33       all_inventory_view.untqty,
34       sum(all_inventory_view.untqty) untqty_l,
35       sum(shipping_pckwrk_view.pckqty) inloadqty,
36       to_char(appt.start_dte, 'MM-DD-YYYY HH24:MI:SS') start_dte,
37       to_char(trlr.arrdte, 'MM-DD-YYYY HH24:MI:SS') arrdte,
38       CASE WHEN (prtmst.catch_cod) is not null THEN '0'
39       ELSE sum(round(prtftp_view.untgrswgt *(shipping_pckwrk_view.pckqty), 3))
40     END AS gross_weight,
41       CASE WHEN (prtmst.catch_cod) is not null THEN '0'
42       ELSE sum(round(prtftp_view.untnetwt *(shipping_pckwrk_view.pckqty), 3))
43     END AS net_weight,
44       CASE WHEN (prtmst.catch_cod) is null THEN '0'
45       ELSE sum(all_inventory_view.catch_qty)
46     END AS total_catch_qty,
47       prtmst.catch_cod
48 from shipping_pckwrk_view
49 inner
50 join prtmst
51   on prtmst.prtnum = shipping_pckwrk_view.prtnum
52   and prtmst.prt_client_id = shipping_pckwrk_view.prt_client_id
53   and prtmst.wh_id_tmpl = shipping_pckwrk_view.wh_id
54 inner
55 join prtdsc ptd

```

Fuente: Los Autores (2025)

Una vez que se integran los comandos al sistema, el siguiente paso es realizar el diseño del reporte a través del software Jaspersoft el cual se enlaza al ambiente de pruebas por medio de un adaptador de datos personalizado. y mediante el comando se puede generar el diseño del reporte y realizar pruebas con datos reales.


Figura 39: Editor de consulta Jaspersoft para el reporte *Bill Of Landing*



Fuente: Los Autores (2025)

Es fundamental aclarar que por buenas prácticas se genera un comando para cada reporte diseñado y desarrollado para el cliente.

Figura 40: Diseño del reporte principal Bill Of Landing en Jaspersoft



STRAIGHT BILL OF LADING - SHORT FORM
ORIGINAL- NON-NEGOTIABLE

Page Header

LOAD: \$F{car_move_id}

CFA #: \$F{notbxt}

CUSTOMER:		DELIVER TO:	
\$F{client_id}	\$F{customer_address_name}	\$F{customer_address_1}	\$F{customer_address_1}
\$F{client_address_1}	\$F{customer_address_1}	\$F{customer_address_1}	\$F{customer_address_1}
\$F{client_city}+", "\$F{client_state}+", "+	\$F{customer_city}+", "+	\$F{customer_city}+", "+	\$F{customer_city}+", "+

CUSTOMER: \$F{client_id}	CARRIER: \$F{carcod}	SEAL NUMBER(S): \$F{seal}
PO #: \$F{cponum}	EQUIPMENT #: \$F{trlr_num}	TRAILER SET POINT: \$F{dep}
SALES ORDER #: \$F{dlvnum}	ORDER DATE: \$F{moddte}	FREIGHT TERM: \$F{short_dsc}
ORDER #: \$F{ordnum}	SHIP DATE: \$F{dispatch_dte}	TEMP REC SRL #: \$F{csr_phnum}

CUSTOMER ORDER INFORMATION

SUBREPORTE

Detail 1

<p>ACTING AS AGENT FOR CUSTOMER</p> <p>\$F{wh_address_name}</p> <p>\$F{wh_address_1}</p> <p>\$F{wh_city}+", "\$F{wh_adrstc}+' '+ \$F{wh_adrpsz}</p> <p>\$F{wh_phone}+", "\$F{wh_email}</p> <p><small>Subject to Section 7 of conditions of applicable bill of lading, if this shipment is to be delivered to the consignee without recourse on the consignor, the consignor shall sign the following statement: The carrier shall not make delivery of this shipment without payment of the freight and any lawful charges.</small></p> <p>SHIPPED BY:</p> <p style="text-align: center;">(Signature of Consignor)</p>	<p>THIS IS NOT AN INVOICE</p> <p>RECEIVED ABOVE MERCHANDISE IN GOOD CONDITION</p> <p>FIRM NAME: \$F{carcod}</p> <hr/> <p>RECEIVED BY:</p> <p style="text-align: center;">(Carrier Agent)</p>
--	---

"Page "+\$F{ }"+ \$V

Column Footer

DOCUMENT ID: OUTBR021.00

Fuente: Los Autores (2025)

Para la parte de las cantidades totales de ordenada, cantidad enviada, peso bruto, peso neto, así como el número de ítem, descripción del producto, lote y UOM, se calculan y se muestran en el subreporte.

Figura 41: Diseño del subreporte Bill Of Landing en Jaspersoft

ITEM	PRODUCT DESCRIPTION	QTY ORDERED	QTY SHIPPED	LOT	UOM	GROSS WEIGHT	NET WEIGHT	CATCH WEIGHT								
\$F{prtnum}	\$F{lngdsc}	\$F{ordqty}	\$F{shpqty}	\$F{lotnum}	\$F	\$F	\$F	CEIL(\$F								
ORDER REMARKS:																
Appointment Time: \$F{start_dte}						Summary <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>TOTAL UNITS</td> <td>TOTAL GROSS WEIGHT</td> <td>TOTAL NET WEIGHT</td> <td>TOTAL CATCH WEIGHT</td> </tr> <tr> <td>\$V</td> <td>CEIL(\$V)</td> <td>CEIL(\$V)</td> <td>CEIL(\$V)</td> </tr> </table>			TOTAL UNITS	TOTAL GROSS WEIGHT	TOTAL NET WEIGHT	TOTAL CATCH WEIGHT	\$V	CEIL(\$V)	CEIL(\$V)	CEIL(\$V)
TOTAL UNITS	TOTAL GROSS WEIGHT	TOTAL NET WEIGHT	TOTAL CATCH WEIGHT													
\$V	CEIL(\$V)	CEIL(\$V)	CEIL(\$V)													
Arrival Time: \$F{arrdte}																

Fuente: Los Autores (2025)

Para completar la etapa de implementación, se previsualiza el reporte con distintos escenarios de embarque donde se comprueban la concordancia, precisión y correcta visualización de los datos.

Figura 42: Matriz de pruebas del reporte Bill Of Landing

Client	Arcadia Cold		Testing Summary							
Date	5-Jul-24		Scenarios							
	Review	Test In Review	0	0						
	Pending	Test Pending Evaluation	0	0,00%						
	Completed	Test Completed Successfully	3	100,00%						
	Error	Test In Error	0	0,00%						
	Quality	Test Pending Client Evaluation	0	0,00%						
	Not Applicable	Test NOT Applies In the Process Model	0	0,00%						
	Escalated	Cases Escalated for Blue Yonder	0	0						
	Totals		3	100,00%						

No.	Process	Status	Activity	Details	Environment	Result	Date
1	BOL report generation	Completed	LOAD [CARTEST] DISPATCHED TO CARGILL - ALDI	BOL REPORT	NP1	The information is seen correctly	05/07/2024
2	BOL report generation	Completed	LOAD [MIXTEST] DISPATCHED TO CARGILL - ALDI AND ARMADA	BOL REPORT	NP1	The information is seen correctly	05/07/2024
3	BOL report generation	Completed	LOAD [NCTEST] DISPATCHED TO ARMADA	BOL REPORT	NP1	The information is seen correctly	05/07/2024
4	BOL report generation	Completed	LOAD [MIXTEST] DISPATCHED TO CARGILL - ALDI AND ARMADA	BOL REPORT	NP1	The information is seen correctly	05/07/2024
5	BOL report generation	Completed	LOAD [MULTI01] DISPATCHED TO ARMADA	BOL REPORT	NP1	The information is seen correctly	06/07/2024
6	BOL report generation	Completed	LOAD [NCTEST] DISPATCHED TO ARMADA	BOL REPORT	NP1	The information is seen correctly	06/07/2024
7	BOL report generation	Completed	INBOUND SHIPMENTS [RCVCARTEST] RECEIVED FROM CARGILL - ALDI	BOL REPORT	NP1	The information is seen correctly	06/07/2024
8	BOL report generation	Completed	INBOUND SHIPMENTS [RCVNCTEST] RECEIVED FROM ARMADA	BOL REPORT	NP1	The information is seen correctly	07/07/2024

Fuente: Los Autores (2025)

5.4.3. Capacitación y pruebas de usuario

En este aspecto de capacitación al usuario sobre como visualizar e imprimir el reporte se realizó un manual con las especificaciones técnicas y como usar dicho desarrollo. Adicionalmente se llevaron a cabo sesiones con el usuario en donde ellos pudieron interactuar y familiarizarse con el desarrollo llevando a cabo pruebas y realizando las validaciones correspondientes de la información.

5.4.4. Puesta en marcha (Go-Live)

Para la instalación de la customización en los ambientes de producción, se estableció junto con el cliente una fecha y hora cuando la operación estuviera detenida. Una vez confirmada la instalación, se procede a validar en tiempo real que todos los comandos y archivo .jrmxl se hayan integrado correctamente al sistema. Asimismo, se realizó una prueba de visualización en todos los ambientes productivos.

5.4.5. Optimización y soporte

Para garantizar un buen funcionamiento del desarrollo se tuvo durante tres semanas en monitoreo y atendiendo cualquier inconveniente o mal funcionamiento del mismo, este proceso se llevó a cabo mediante el sistema de tickets de atlassian 'Jira Service Management' en donde el cliente reportaba estos incidentes o solicitudes de mejora. Una vez finalizado este seguimiento de tres semanas se procedió con la transición del desarrollo al equipo de soporte, el cual queda como responsable de cualquier incidente que se presente en adelante.

5.4.6. Análisis de resultado

La Implementación del reporte Bill Of Landing permitió mejorar la precisión y disponibilidad de datos al integrarse directamente con el WMS, eliminando errores manuales y asegurando que la información de órdenes y cargas se estructure correctamente. La separación en dos consultas MOCA, una para el encabezado y otra para los detalles, optimizó la organización del reporte, facilitando su lectura a través de un subreporte.

Además, la generación automática permitió incluir cálculos automáticos de peso, volumen y cantidades, garantizando que el documento se muestre con exactitud las condiciones del estado de la mercancía. La consulta optimizada redujo significativamente el tiempo de generación del reporte, permitiendo obtener la información en segundos, incluso en escenarios con múltiples productos y cargas. También se implementó la opción de envío automatizado por correo (EMS) y

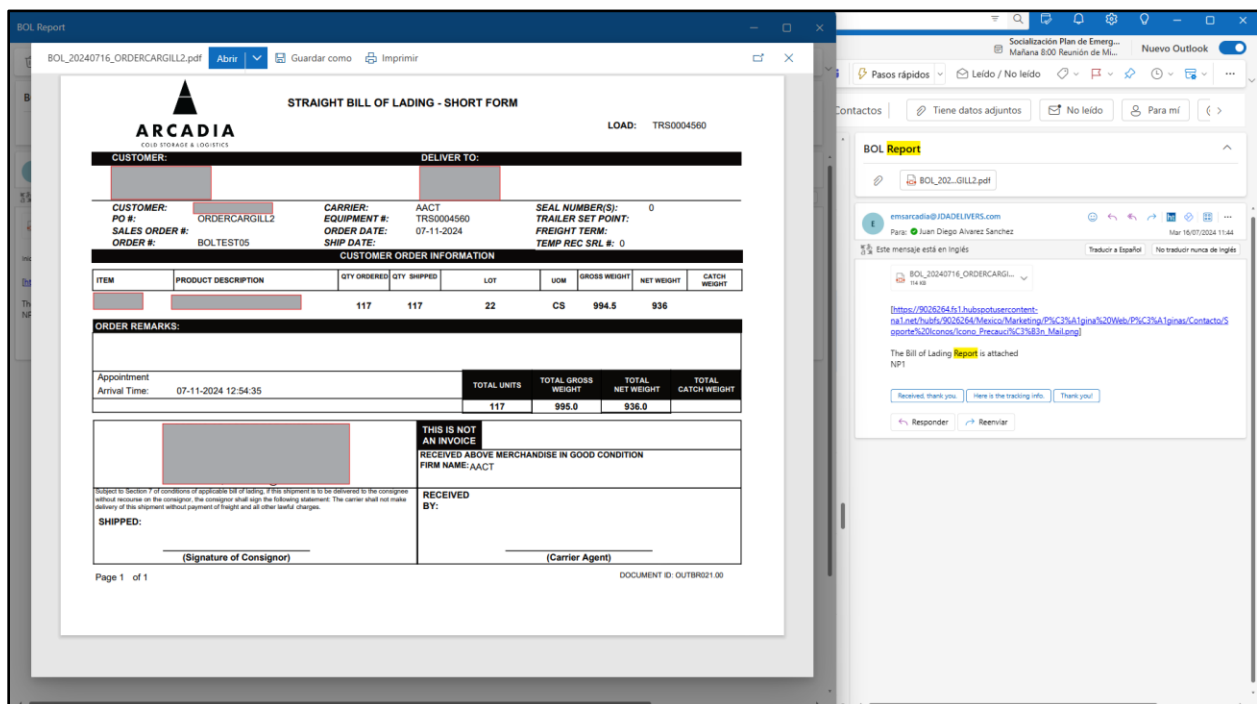
su integración con la interfaz web del WMS, lo que permitió a los usuarios visualizar el documento antes de descargarlo o enviarlo.

5.5. EMS Bill Of Landing

Tras finalizar el reporte anterior, el cliente identificó la necesidad de optimizar la generación y distribución de dicho reporte mediante un proceso automatizado, eliminando la intervención manual y reduciendo posibles errores operativos. Para abordar esta mejora, se propuso desarrollar un Express Mail Service (EMS), un mecanismo que permite la automatización de notificaciones y reportes en tiempo real.

Este sistema se integra con la infraestructura existente y opera mediante la configuración de eventos clave. En este caso, al cierre del camión, el EMS activa un workflow que genera el reporte correspondiente, lo convierte a formato PDF y lo distribuye automáticamente por correo electrónico a los destinatarios designados. Con esta solución, se garantiza eficiencia operativa, reducción de tiempos y mayor precisión en la entrega de información.

Figura 43: Reporte “Bill Of Landing” generado y enviado por el EMS



Fuente: Los Autores (2025)

5.5.1. Diseño

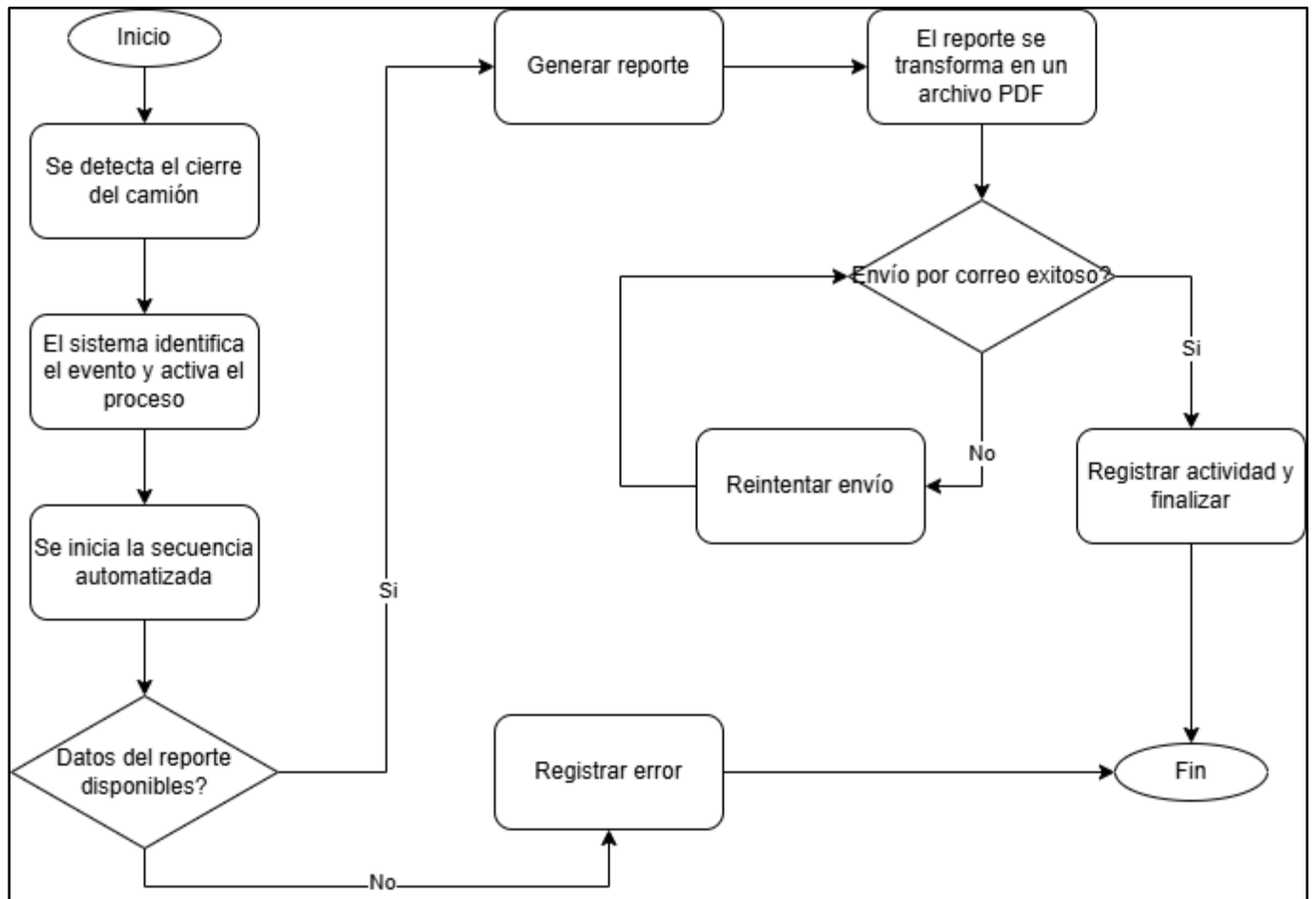
Para los requerimientos funcionales proporcionados por el cliente se identificaron los siguientes:

- El sistema debe generar reportes en función de eventos predefinidos, como el cierre de un camión.
- Los reportes generados deben transformarse automáticamente a PDF antes de su envío.
- El EMS debe enviar los reportes a los destinatarios configurados sin intervención manual.
- Permitir definir eventos que activen la generación del reporte, como el cierre de un camión
- Permitir la configuración de destinatarios de manera individual o grupal según reglas predefinidas.
- Guardar un historial detallado de reportes generados, correos enviados y posibles errores ocurridos.
- Implementar un mecanismo de reintento en caso de fallo en la generación o envío del reporte.

Para los requerimientos técnicos El job de liberación de restricciones de inventario debe estar desarrollado utilizando comandos MOCA/SQL, asegurando la correcta consulta, identificación y actualización de los registros en la base de datos. Debe ejecutarse de manera automática según una programación predefinida y permitir su ejecución manual en caso necesario. La solución debe garantizar un bajo impacto en el rendimiento del sistema, optimizando las consultas y evitando bloqueos en la base de datos.

A continuación, se presenta el diagrama de flujo donde se visualiza el proceso de creación y envío del reporte de forma automática

Figura 44: Diagrama de flujo para el EMS del reporte BOL



Fuente: Los Autores (2025)

5.5.2. Implementación

Para la creación de un EMS se debe realizar primero el comando generador del evento, en el cual se especifica el tipo de archivo y cuál es el reporte a crear.

Figura 45: Lógica del comando "Create var EMS costum bol report"

```

1 <command>
2   <name>create var ems custom bol report</name>
3   <description>Command created to generate custom bol report</description>
4   <type>Local Syntax</type>
5   <local-syntax>
6   <![CDATA[
7   /*JDALVAREZ 06/06/2024: Command created to generate custom bol report*/
8   [select rtstrl path
9     from poldat
10    where polcod = /*#nobind*/ 'VAR' /*#bind*/
11      and polvar = /*#nobind*/ 'PATH' /*#bind*/
12      and polval = /*#nobind*/ 'EMS-REPORTS' /*#bind*/]
13   |
14   generate moca report
15   where format_typ = 'pdf'
16     and locale_id = 'US_ENGLISH'
17     and gen_usr_id = @gen_usr_id
18     and rpt_id = 'Var-ReportBillLading'
19     and keep_days = 14
20     and dest = @path || @name || '.pdf'
21     and dest_typ = 'file'
22     and ordnum = @ordnum
23     and trlr_num = @trlr_num
24     and car_move_id = @car_move_id
25   ]]>
26 </local-syntax>
27 </command>

```

Fuente: Los Autores (2025)

Posterior mente se crea el comando “raise” el cual es el encargado de ejecutar el evento VAR-BOL-REPORT-CLIENT en el EMS, el cual envía un correo con el reporte Bill of Lading (BOL) adjunto. Primero, verifica si es la primera ejecución para inicializar el evento; si no, obtiene información del EMS y del servidor para validar si el evento está habilitado y localizar la ruta del archivo en la base de datos.

Figura 46: Lógica del comando "raise var EMS for bol report client"

```

7  /**JDALVAREZ 05/06/2024: Command created to execute VAR-BOL-REPORT-CLIENT event. Event that sends an email with the attached BOL report*/
8  get_ems_source system
9  |
10 if (@primer_flg and @primer_flg = 1)
11 {
12     /* We must prime the event definition */
13     publish data
14     where evt_msg = "The Bill of Lading Report is attached"
15     and evt_subj = "BOL Report"
16     and evt_nam = @evt_nam
17     and src_sys = @src_sys
18     and lngdsc = "The Bill of Lading Report is attached"
19 }
20 else
21 {
22     get_ems_event_information
23     where evt_nam = 'VAR-BOL-REPORT-CLIENT'
24 |
25 if (@enabled = 1)
26 {
27     get_server_information
28 |
29 [select rtstr1 path
30     from poldat
31     where polcod = /*#nobind*/ 'VAR' /*#bind*/
32     and polvar = /*#nobind*/ 'PATH' /*#bind*/
33     and polval = /*#nobind*/ 'EMS-REPORTS' /*#bind*/]
34 |
35 log_ems_event
36     where evt_nam = 'VAR-BOL-REPORT-CLIENT'
37     and key_val = 'VAR-BOL-REPORT-CLIENT' || to_char(sysdate)
38     and locale_id = 'US_ENGLISH'
39     and src_sys = @src_sys
40     and wh_id = @wh_id
41     and att_app_srv = @url
42     and att_path = @path || @name || '.pdf'
43     and att_file = @name || '.pdf'
44     and att_rm_file = '0'

```

Fuente: Los Autores (2025)

Continuando con el proceso de la customización se generó las políticas para configurar y parametrizar el comportamiento del evento VAR-BOL-REPORT-CLIENT, en este caso se realizó la configuración para que este EMS solo se generara para un cliente.

Figura 47: Políticas generadas para el EMS del reporte BOL

```

1  polcod,polvar,polval,wh_id,tmpl,srtseq,rtstr1,rtstr2,rtnum1,rtnum2,rtflt1,rtflt2,moddte,mod_usr_id,grp_nam,is_purged
2  VAR,EMS-CLIENT-BOL,EMS-CLIENT-BOL,---,0,<client_id>,,1,0,0,0,20240605175139,JDALVAREZ,,0
3  VAR,EMS-CLIENT-BOL,EMS-CLIENT-BOL,BUR01,1,CARGILL - ALDI,,1,0,0,0,20240605175051,JDALVAREZ,,0
4  VAR,EMS-CLIENT-BOL,EMS-CLIENT-BOL,PHX01,1,CARGILL - ALDI,,1,0,0,0,20240605175051,JDALVAREZ,,0
5  EMS,EVENTS,VAR-BOL-REPORT-CLIENT,---,0,raise var ems for bol report client,,1,0,0,0,20240606110343,JDALVAREZ,,0
6  EMS,EVENTS,VAR-BOL-REPORT-CLIENT,BUR01,1,raise var ems for bol report client,,1,0,0,0,20240606110343,JDALVAREZ,,0
7  EMS,EVENTS,VAR-BOL-REPORT-CLIENT,PHX01,1,raise var ems for bol report client,,1,0,0,0,20240606110343,JDALVAREZ,,0

```

Fuente: Los Autores (2025)

Para finalizar con el desarrollo se crea y configura el flujo de trabajo el cual es el encargado de ejecutar el comando moca que se encarga de crear y realizar las validaciones para enviar el reporte siempre que se cierre un equipo de transporte.

Figura 48: Lógica encargada de correr el workflow

```

/*JDALVAREZ: command to send EMS BOL depending on client*/
[select cnfrm_serv_id
  from cnfrm_noninv_serv
 where non_invtid = @non_invtid
       and serv_id = /*#nobind*/ 'VC-EMS-BOL-REPORT' /*#bind*/
       and serv_result = /*#nobind*/ 'PASS' /*#bind*/
 order by moddte desc] catch(-1403)
|
if (@? != 0)
{
  [select distinct shipment_line.ordnum,
    car_move.car_move_id,
    shipment.wh_id
  from trlr
  join car_move
    on trlr.trlr_id = car_move.trlr_id
  join stop
    on car_move.car_move_id = stop.car_move_id
  join shipment
    on stop.stop_id = shipment.stop_id
  join shipment_line
    on shipment.ship_id = shipment_line.ship_id
 where trlr.trlr_id = @trlr_id
       and shipment.wh_id = nvl(@wh_id, @@wh_id)] catch(-1403)
|
list var report bill of lading
 where wh_id = nvl(@wh_id, @@wh_id)
       and ordnum = @ordnum
       and car_move_id = @car_move_id catch(@?)
|
list policies
 where polcod = 'VAR'
       and polvar = 'EMS-CLIENT-BOL'
       and polval = 'EMS-CLIENT-BOL'
       and rtstrl = @client_id
       and wh_id = nvl(@wh_id, @@wh_id)
       and rtnuml = '1' catch(-1403)
|
if (@? = 0)
{
  publish data
  where name = substr(('BOL_' || @cponum), '', 25)
       and client_id = @client_id
       and evt_nam = 'VAR-BOL-REPORT-' || @rtstr2
  |
  create var ems custom bol report
  where trlr_num = @trlr_num
       and car_move_id = @car_move_id
       and name = @name
  |
  raise var ems for bol report
  where trlr_num = @trlr_num
       and car_move_id = @car_move_id
       and name = @name
       and evt_nam = @evt_nam
}
}

```

Fuente: Los Autores (2025)

Para completar la etapa de implementación, se realizaron pruebas unitarias para validar el correcto funcionamiento del EMS en el ambiente de pruebas del cliente.

Figura 49: Matriz de pruebas para el EMS del reporte Bill Of Landing

No.	Client	Date	Testing Summary					
2	Arcadia Cold	25-Jun-24	Scenarios					
4	Review	Test in Review	0	0				
5	Pending	Test Pending Evaluation	0	0,00%				
6	Completed	Test Completed Successfully	4	100,00%				
7	Error	Test In Error	0	0,00%				
8	Quality	Test Pending Client Evaluation	0	0,00%				
9	Not Applicable	Test NOT Applies in the Process Model	0	0,00%				
10	Escalated	Cases Escalated for Blue Yonder	0	0				
11	Totals		4	100,00%				
No.	Process	Status	Activity	Details	Environment	Result	Date	
1	BOL Mail	Completed	LOAD [TESTEMSMUL] DISPATCHED TO CARGILL - ALDI	BOL REPORT MAIL	NP1	E-MAIL SENT	25/06/2024	
2	BOL Mail	Completed	LOAD [LOADTESTLP] DISPATCHED TO CARGILL - ALDI	BOL REPORT MAIL	NP1	E-MAIL SENT	25/06/2024	
3	BOL Mail	Completed	LOAD [MULTI01] DISPATCHED TO ARMADA	BOL REPORT MAIL	NP1	E-MAIL NOT SENT	25/06/2024	
4	BOL Mail	Completed	LOAD [DGM22124] DISPATCHED TO CREATIVE PACKAGING	BOL REPORT MAIL	NP1	E-MAIL NOT SENT	25/06/2024	

Fuente: Los Autores (2025)

5.5.3. Capacitación y pruebas de usuario

Se elaboró un manual de usuario que explica cómo utilizar y configurar el evento para el reporte BOL, adicionalmente se generó un documento técnico donde se especificó todos los comandos creados y las políticas para este desarrollo. Se realizaron sesiones con los usuarios, que interactuaron con la configuración del evento y realizaron ejercicios de prueba para validar el correcto funcionamiento del envío automático del reporte.

5.5.4. Puesta en marcha (Go-Live)

Se estableció la fecha de instalación y por el horario operativo de los ambientes de producción se consideró prudente que la instalación se realizara en horas de la noche donde la operación estuviera detenida y no se vieran afectados los ambientes productivos. Una vez instalada la customización, se validó en tiempo real que los comandos se cargaron en el sistema y las funcionalidades funcionaran correctamente.

5.5.5. Optimización y soporte

Para garantizar un buen funcionamiento del desarrollo se tuvo durante tres semanas en monitoreo y atendiendo cualquier inconveniente o mal funcionamiento del mismo, este proceso se llevó a cabo mediante el sistema de tickets en donde el cliente reportaba estos incidentes o

solicitudes de mejora. Una vez finalizado este seguimiento de tres semanas se procedió con la transición del desarrollo al equipo de soporte, el cual queda como responsable de cualquier incidente que se presente en adelante.

5.5.6. Análisis de resultado

La implementación del Express Mail Service (EMS) en la generación y envío del reporte Bill of Lading (BOL) ha optimizado el proceso al automatizar su creación y distribución. Ahora, el reporte se genera de manera inmediata al cierre del camión, reduciendo tiempos de espera y eliminando la necesidad de intervención manual. Esto garantiza que la documentación esencial para el transporte de mercancías llegue oportunamente a los destinatarios correspondientes, mejorando la eficiencia operativa y reduciendo errores en la entrega de información.

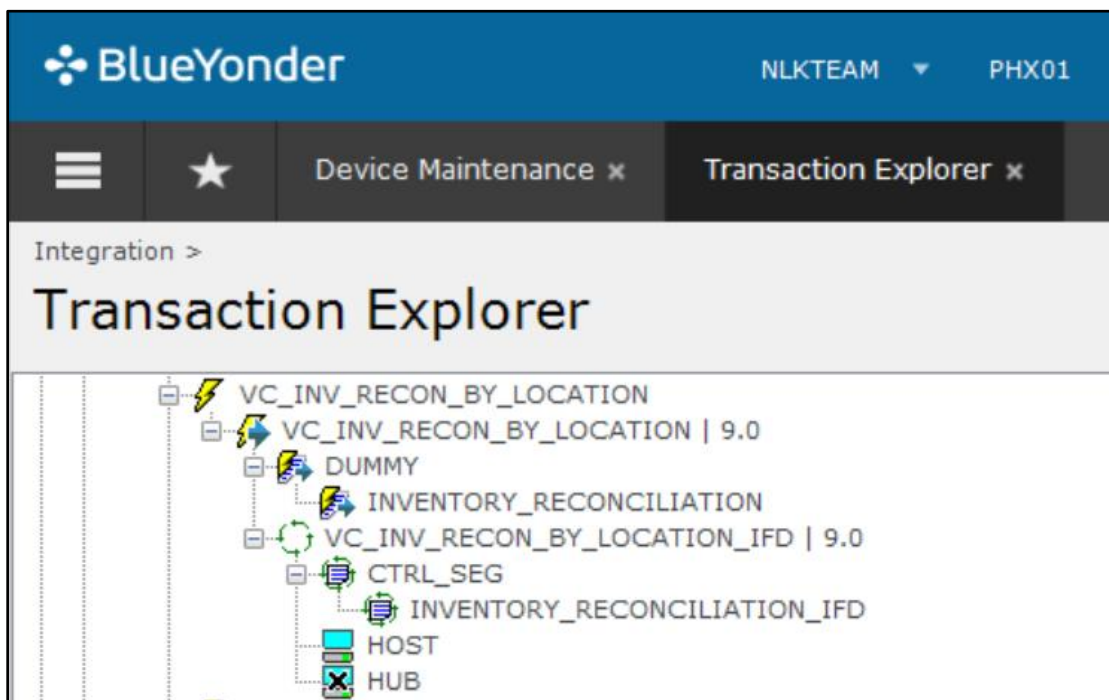
Además, el EMS ha fortalecido la trazabilidad del BOL, registrando cada evento relacionado con su generación y envío. Esto permite un monitoreo detallado del proceso y facilita la identificación de posibles fallos para su corrección inmediata. Gracias a la integración con la infraestructura existente, el sistema asegura una transición fluida y confiable, mejorando la precisión y disponibilidad del BOL en tiempo real, lo que contribuye a una mejor gestión logística y toma de decisiones dentro de la operación.

5.6. Interface Inventory Reconciliation by Location

Esta customización se basa en la modificación de una interfaz de salida de datos, que forma parte de los componentes estándar del WMS. Es una solución desarrollada para automatizar y optimizar la reconciliación de inventarios, consolidando información crítica por ubicación y número de pallet (LPN). Su implementación se realiza desde el módulo de integración del WMS, por medio del cual se enlazan sistemas externos para el intercambio de transacciones.

Entre los principales desafíos identificados en la operación del cliente se encuentran la falta de visibilidad en tiempo real del inventario dentro del almacén y la dificultad para rastrear atributos específicos como lotes, fechas de expiración o holds. Para resolverlo, se diseñó un proceso automatizado que, mediante comandos SQL y MOCA, consolida datos de más de 10 tablas del sistema como invdttl, locmst y prtmst, aplica validaciones dinámicas y genera un archivo CSV estructurado, listo para su análisis o integración con otras plataformas.

Figura 50: *Interface Inventory Reconciliation by Location*



Fuente: Los Autores (2025)

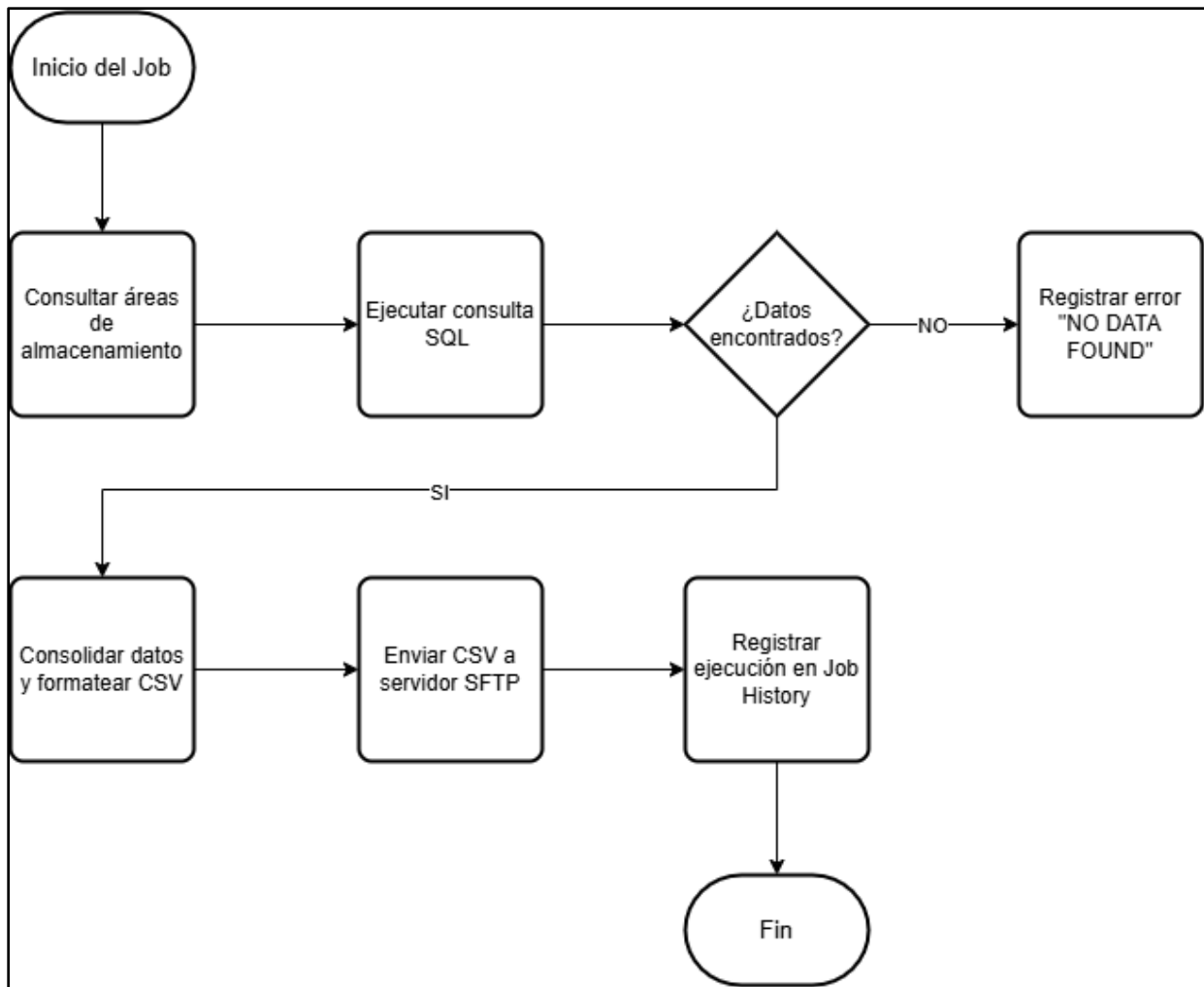
5.6.1. Diseño

Partiendo de la necesidad de llevar un registro periódico de la información, así como facilitar el tratamiento de estos datos para su posterior análisis, se plantearon los siguientes requerimientos funcionales:

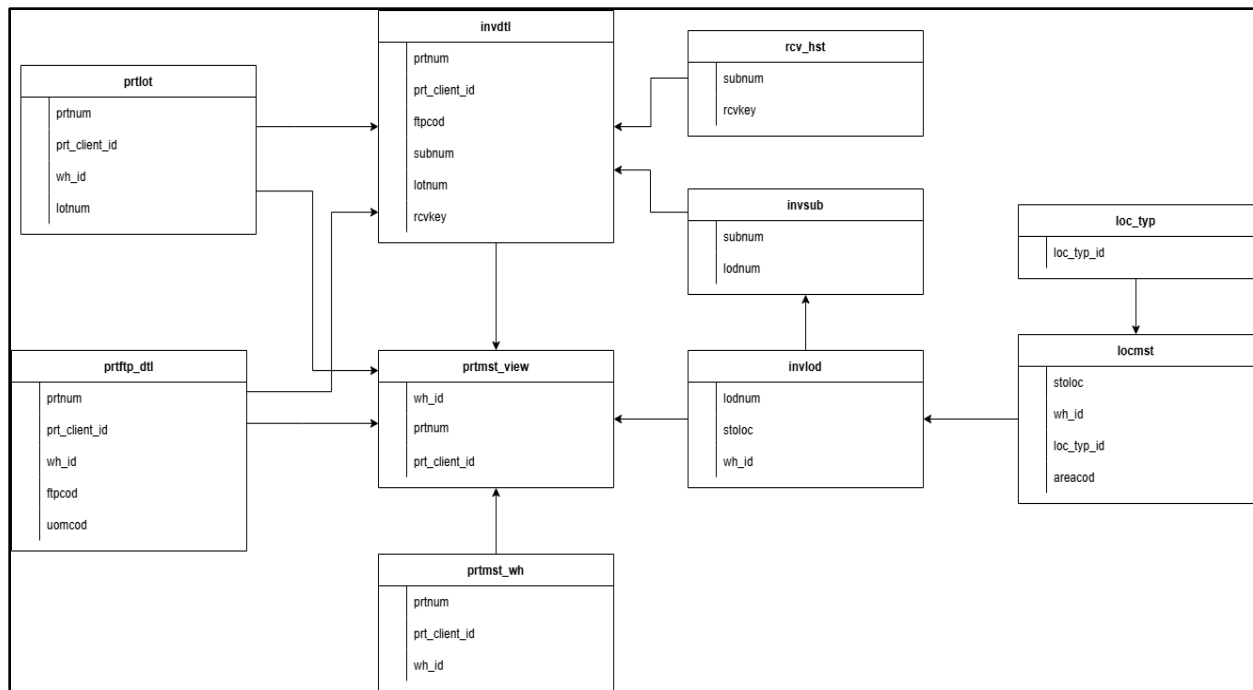
- Generar un archivo CSV con los datos consolidados del inventario por ubicación y LPN, asegurando que la captura automatizada de la información siempre se realice en un horario diferente de las actividades operativas.
- Campos del CSV: Área, Ubicación, LPN, Ítem, Familia de Ítem, Estado de Inventario, Cantidad, Unidad de Medida, Huella del Ítem, Número de Lote, País de Origen, Nivel de Revisión, Fecha de Producción, Fecha de Expiración, Cantidad de Captura, Cliente, Perfil de Envejecimiento, ID de Envío Entrante.
- Independientemente de que sea un proceso automatizado, debe ser posible generar el archivo CSV en cualquier momento que esa requerido.
- Permitir el filtrado del inventario por áreas específicas y la gestión de errores.

Como requerimientos técnicos, es imprescindible el uso de comandos MOCA/SQL para consultar y consolidar datos desde la base de datos, así como para ejecutar la activación de la interfaz personalizada, que se encuentra en el módulo de integración y permite procesar la información para generar el CSV correspondiente. Por último, se debe crear y configurar un JOB programado para automatizar todo el proceso.

A continuación, se presenta el diagrama de flujo donde se visualiza el proceso de generación del archivo CSV, así como el diagrama Entidad-Relación, que sirve como guía para el comando MOCA que realiza la consulta.

Figura 51: Diagrama de flujo para la interface Inventory Reconciliation

Fuente: Los Autores (2025)

Figura 52: Diagrama Entidad-Relación para la interface Inventory Reconciliation

Fuente: Los Autores (2025)

5.6.2. Implementación

Inicialmente, se modifica el comando MOCA “retrieve inventory reconciliation”, que es un componente de nivel JDA estándar y se encarga de ejecutar la consulta de datos para la interfaz estándar de Inventory Reconciliation. Se agregan los campos solicitados y se incorpora un filtro para que la búsqueda se acote según las áreas de almacenaje establecidas, la unión de estas tablas se elabora principalmente a partir de las tablas de detalle del inventario y la vista maestra de los productos, que brindan información más específica en función del identificador de cada pallet

El nuevo comando incluye un bloque catch (-1403) para evitar fallos por datos no encontrados. Además, ajusta los cálculos de la cantidad almacenada en relación con la cantidad configurada por pallet en la huella del ítem (no_pallet) y determina la proporción de unidades en un LPN que proviene de un registro específico (no_lpn), por ejemplo, cuando un LPN está

compuesto por diferentes sub-LPNs. Una vez se completa la consulta, se crea el comando como un componente de nivel “VAR” bajo el nombre de “retrieve var inventory reconciliation”.

Figura 53: Lógica del comando “retrieve var inventory reconciliation”

```

get warehouse id
where wh_id = nvl(@wh_id, @@wh_id)
|
[select loc. arecod areas
from invdtl dtl
join invsub sub
on dtl.subnum = sub.subnum
join invlod lod
on sub.lodnum = lod.lodnum
join locmst loc
on lod.stoloc = loc.stoloc
and lod.wh_id = loc.wh_id
join loc_typ lt
on loc.loc_typ_id = lt.loc_typ_id
where loc.wh_id = nvl(@wh_id, @@wh_id)
and lt.fwiflg = /*#nbind*/ '1' /*#bind*/
group by loc. arecod]
|
[select lm. arecod,
id.prtnum,
cast(id.prt_client_id as nvarchar) prt_client_id1,
pm.prtfam,
id.invsts,
ROUND(CAST(sum(id.untqty) AS DECIMAL) / NULLIF(CAST(pa.untqty AS
DECIMAL), 0), 2) AS no_pallet,
ROUND(CAST(id.untqty AS DECIMAL) / NULLIF(SUM(id.untqty) OVER (
PARTITION BY il.lodnum), 0), 2) as no_lpn,
id.untcas,
id.ftpcod,
id.lotnum,
id.rev1,
id.orgcod,
pl.lot_sts,
pm.stkuom,
sum(id.untqty) untqty,
sum(id.catch_qty) catch_qty,
pm.catch_unttyp,
lm.stoloc,
il.lodnum,
id.dtlnum,
lm.wh_id,
to_char(id.expire_dte, 'YYYY-MM-DD') expire_dte,
to_char(id.mandte, 'YYYY-MM-DD') mandte,
rc.trknum
from prtmst view pm

```

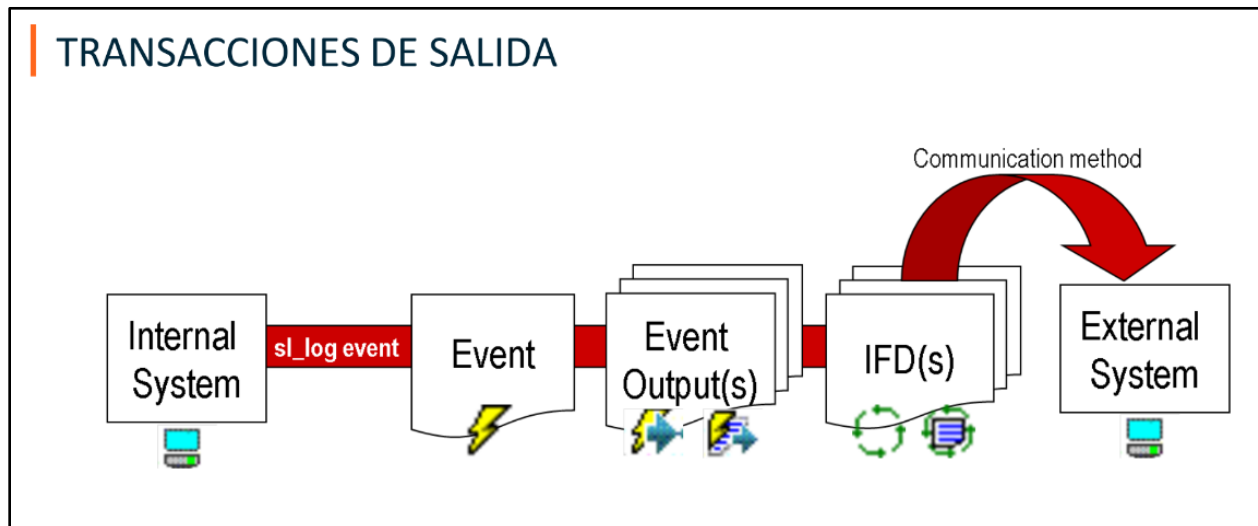
Fuente: Los Autores (2025)

Continuando con la modificación de la interfaz estándar, se debe entender el flujo de datos en las transacciones de la salida (*ver Figura 55*), donde el evento (Event) representa la acción definida que desencadena una transacción y permite que el módulo de integración recopile la información desde el WMS. Luego, el evento de salida (Event Output) representa el superconjunto de datos sin procesar que es comunicado en uno o más sistemas.

En el segmento IFD (Interface Document) se define la estructura de datos que el módulo de integración espera enviar a un sistema externo. Para recuperar los datos de una consulta, se

usa un método de recuperación (Retrieve Method), que se ejecuta desde el evento de salida y se encarga de obtener la información que el IFD debe interpretar.

Figura 54: *Flujo de datos en las transacciones de salida*



Fuente: BY Integrator

En esta ocasión fue necesario cambiar tanto el evento de salida como el segmento IFD. Para el primero se agregan los nuevos campos que tendrá el CSV, asegurando que el tipo de dato y el enlace con los valores del módulo de integración correspondan con las columnas de la consulta (**ver Figura 56**). Además, se actualizó el método de recuperación con el nuevo comando “retrieve var inventory reconciliation” y con el comando estándar “write csv file”, el cual genera un archivo CSV a partir de los resultados del comando anterior, asigna un nombre de archivo y lo envía a una ruta de destino en el sistema (**ver Figura 57**).

El segmento IFD se actualizó con los mismos campos. Al finalizar esta transacción, también se genera un archivo XML, el cual es un estándar para todas las transacciones de salida, almacenándose en sistemas externos para su posterior tratamiento. Esta nueva interfaz se crea con el nombre de VC_INV_RECON_BY_LOCATION.

Figura 55: Evento de salida de la interface Inventory Reconciliation

EO Field	Description	Order	Data Type	Maximum Chara...	Len...	SQL Order	Descendi...
PRTNUM	Item Number	1	Character Stri...	50	200		<input type="checkbox"/>
ORGCOD	Origin Code	2	Character Stri...	25	100		<input type="checkbox"/>
REVLVL	Revision Level	3	Character Stri...	25	100		<input type="checkbox"/>
LOTNUM	Lot Number	4	Character Stri...	25	100		<input type="checkbox"/>
INVSTS	Inventory Status (length changed to ...	5	Character Stri...	4	16		<input type="checkbox"/>
HSTACC	Host Account Code	6	Character Stri...	30	120		<input type="checkbox"/>
UNTQTY	Unit Quantity	7	Number		10		<input type="checkbox"/>
SUP_LOTNUM	Supplier Lot Number	8	Character Stri...	25	25		<input type="checkbox"/>
STKUOM	Stocking UOM	17	Character Stri...	2	8		<input type="checkbox"/>

Fuente: Los Autores (2025)

Figura 56: Método de recuperación de la interface Inventory Reconciliation

```

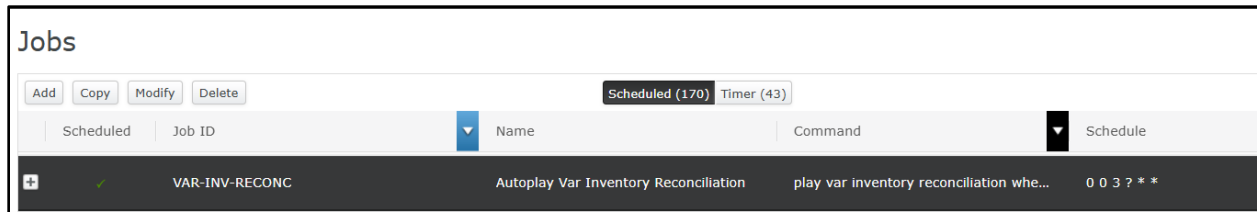
Algorithm Code:
hide stack variable
where name = 'prt_client_id'
retrieve var inventory reconciliation
get translated warehouse ID
{
  [[ String prt_client_id = prt_client_id.replace("-", "#").replace("#", "-");
  ]]
}
publish data
where siteId = @trans_warehouse_id
and area = @arecod
and hstacc = @hstacc
and itemId = @prtnum
and lgn = @lodnum
and itemClientId = @result
and itemFamily = @effam

```

Fuente: Los Autores (2025)

Por último, se configura un JOB automatizado que ejecuta el comando *custom* “play var inventory reconciliation” para desencadenar la activación de la interfaz modificada. Este JOB se ejecuta diariamente a las 3 am y cuenta con la opción de ejecutarse en cualquier momento para realizar pruebas.

Figura 57: Job automatizado para la interface Inventory Reconciliation



Fuente: Los Autores (2025)

Figura 58: Archivo CSV generado por la interface Inventory Reconciliation



Fuente: Los Autores (2025)

Para completar la etapa de implementación, se genera el archivo CSV en múltiples ocasiones y se comparan los datos del inventario listado con los que se visualizan en la interfaz web UI.

Figura 59: *Matriz de pruebas de la interface Inventory Reconciliation*

Client	Arcadia Cold	Testing Summary			
Date	24-Apr-24	Scenarios			
		Review	Test In Review	0	0
		Pending	Test Pending Evaluation	0	0,00%
		Completed	Test Completed Successfully	6	100,00%
		Error	Test In Error	0	0,00%
		Quality	Test Pending Client Evaluation	0	0,00%
		Not Applicable	Test NOT Applies in the Process Model	0	0,00%
		Escalated	Cases Escalated for Blue Yonder	0	0
			Totals	6	100,00%

No.	Process	Status	Activity	Details	Environment	Result
1	LPN:L000000004NL	Completed	LPN without hold flag matches both report and web services	N/A	Np1-PHX01	Successful
2	LPN:502491856	Completed	LPN without hold flag matches both report and web services	N/A	Np1-PHX01	Successful
3	LPN:L0000000025S	Completed	LPN with hold flag and hold number XOUT matches both report and web services	N/A	Np1-PHX01	Successful
4	LPN:100020340000772999	Completed	LPN with hold flag and hold number ANHC matches both report and web services	N/A	Np1-PHX01	Successful
5	LPN:00100796202621597816	Completed	LPN with hold flag and hold number XSFH matches both report and web services	N/A	Np1-PHX01	Successful
6	LPN:L00000000047A	Completed	LPN with hold flag and hold number OUIPOP-QA matches both report and web services	N/A	Np1-PHX01	Successful
7	LPN:120014492727371200	Completed	Item Family matches both report and web services	Added Item Family column	Np1-PHX01	Successful
8	LPN:400449790502705956	Completed	Item Family matches both report and web services	Added Item Family column	Np1-PHX01	Successful
9	LPN:L000000000409	Completed	Item Family matches both report and web services	Added Item Family column	Np1-PHX01	Successful

Fuente: Los Autores (2025)

5.6.3. Capacitación y pruebas de usuario

En las pruebas de usuario se validaron consultas con datos nulos, clientes restringidos y áreas inexistentes. Además, se verificó el formato de distintas columnas del CSV, como los decimales y fechas (ej: YYYY-MM-DD). En las sesiones de capacitación, se entregó el manual de usuario con pasos detallados y capturas de pantalla, y se realizó una demostración del funcionamiento del nuevo JOB programado.

5.6.4. Puesta en marcha (Go-Live)

La instalación de la customización en los ambientes de producción se llevó a cabo en horario no operativo (02:00 AM). Una vez confirmada la instalación, se habilitó el JOB y se realizó una ejecución manual en cada ambiente productivo para confirmar la generación de CSV. La validación finalizó con la revisión del log producido al concluir la transacción.

5.6.5. Optimización y soporte

Como mejoras post-implementación se incluyeron columnas adicionales como Shelf Life, Hold Flag, Hold Number. Asimismo, se optimizó la consulta del comando para reducir el tiempo de ejecución en un 30%. Con cada nueva modificación se monitorio el funcionamiento

automatizado de la interfaz por un periodo de una semana y se proporcionó soporte técnico continuo para resolver cualquier incidencia reportada por los usuarios.

5.6.6. Análisis de resultado

La implementación de este proceso automatizado repercutió directamente en el tiempo destinado a completar la reconciliación de inventario. Previamente, este proceso podía tomar horas o incluso días, debido a que implicaba considerar el tamaño del inventario, la recopilación manual de datos desde múltiples fuentes, su consolidación en hojas de cálculo y la verificación de inconsistencias. Con la inclusión de esta customización, el sistema es capaz de generar el archivo CSV en minutos, eliminando los pasos manuales y reduciendo el tiempo total en un 50% o más.

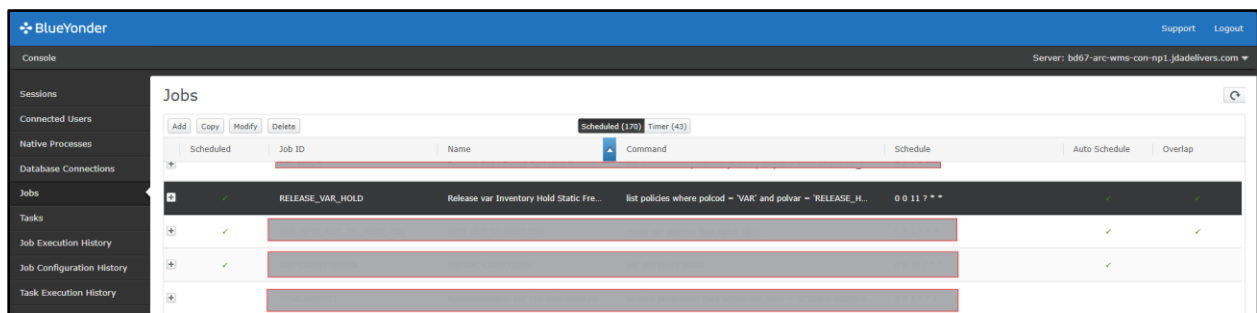
Por otro lado, los errores humanos en la recopilación y consolidación manual de datos, como errores de digitación u omisión de registros, eran comunes. Al automatizar el proceso con consultas SQL y MOCA, se minimizan estos sesgos y se garantiza que los datos sean consistentes y confiables. En términos generales, la valoración del cliente ha sido positiva debido a la claridad de los datos y la facilidad de exportación.

5.7. Job Release Hold

Este Job fue diseñado para automatizar la liberación de restricciones (Holds) aplicadas sobre el inventario. Su principal objetivo es identificar registros de inventario que han estado retenidos por un período superior a un número de días configurable y eliminarlos del estado de retención.

Las restricciones de inventario pueden aplicarse por diversas razones, como control de calidad, auditoría, errores en la recepción o congelación estática del inventario. Este job asegura que dichas restricciones sean eliminadas automáticamente cuando ya no sean necesarias, optimizando la disponibilidad del inventario.

Figura 60: Vista del Job Release Hold en la consola



Fuente: Los Autores (2025)

5.7.1. Diseño

Para el diseño del desarrollo del job "release var inventory hold" se requirió definir bien la estructura para garantizar la correcta identificación y liberación de las restricciones de inventario. Estos fueron los requerimientos funcionales que se tuvieron en cuenta:

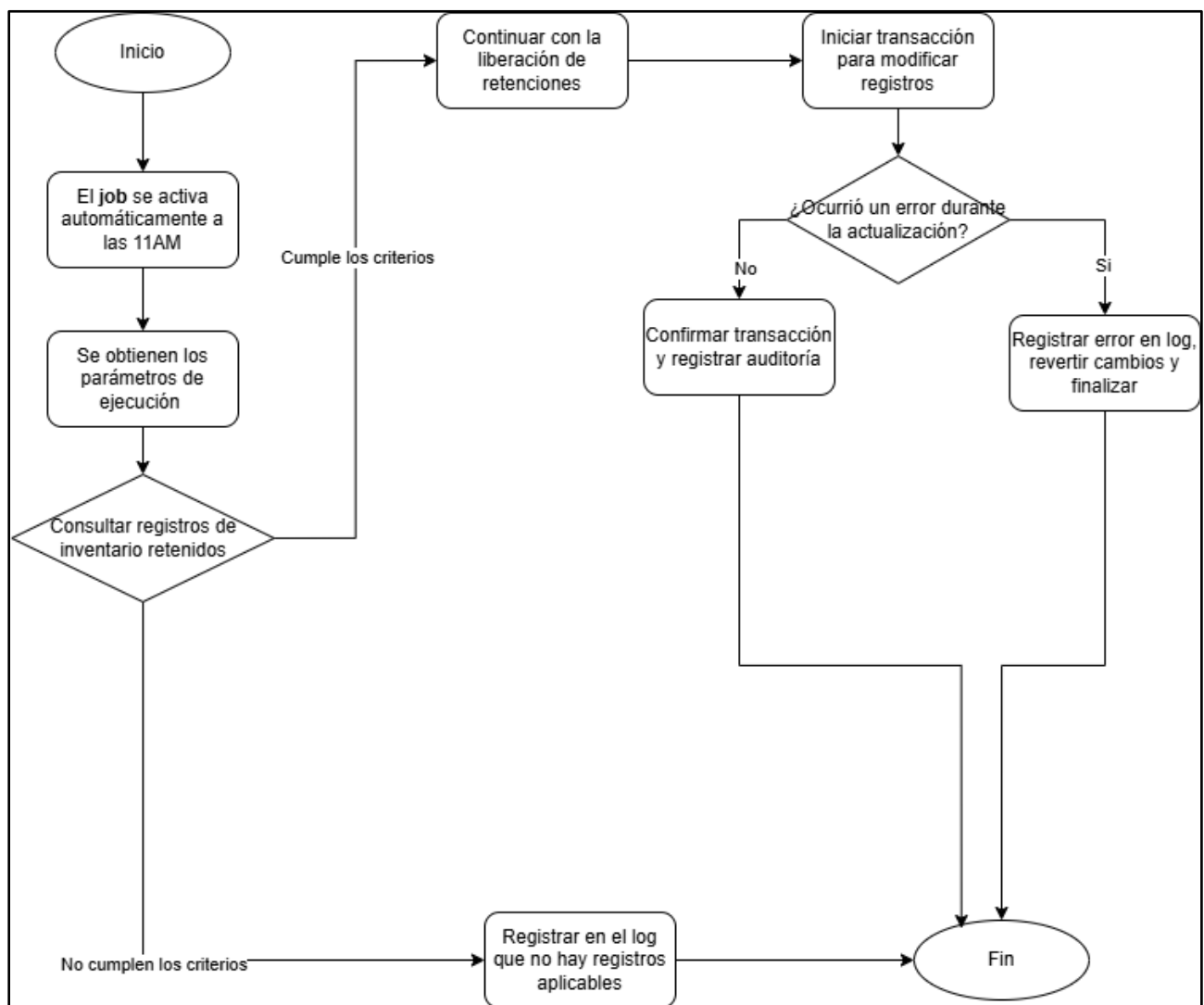
- Identificar y liberar restricciones de inventario en función de la política establecida, considerando únicamente aquellos registros cuya fecha de retención (hlddte) sea menor a sysdate - @numday.
- Permitir la ejecución automática del proceso sin intervención manual, asegurando que no interfiera con otras operaciones críticas del almacén.
- Consultar las tablas del sistema (invlod, invsub, invdttl, invhld, hldmst) para identificar registros de inventario que cumplen con los criterios de liberación de retención.
- Solo liberar registros de inventario que tengan el indicador de retención (hld_flg) activo ('1') y que correspondan al almacén configurado (wh_id).
- Ejecutar la operación de liberación mediante el comando process inventory hold change, asegurando que se actualicen correctamente los estados del inventario y las ubicaciones de los productos.
- Registrar todas las modificaciones en el inventario generadas por la ejecución del proceso, almacenando detalles como usuario, fecha y motivo de liberación.
- Permitir la ejecución manual del proceso en caso de ser necesario, asegurando que los usuarios autorizados puedan liberarlo bajo demanda.

En la parte de requerimientos técnicos si identifico que el El job de liberación de restricciones de inventario debe ejecutarse automáticamente en los horarios programados y permitir ejecución manual cuando sea necesario. Debe estar desarrollado con MOCA/SQL, asegurando consultas eficientes para no afectar el rendimiento del sistema ni generar bloqueos

en la base de datos. Además, debe manejar transacciones seguras para garantizar la integridad de los datos y registrar cada ejecución en un log de auditoría con los detalles del proceso. Por último, debe integrarse con los sistemas de notificación para informar a los usuarios sobre los resultados de cada ejecución.

A continuación, se presenta el diagrama de flujo donde se visualiza el proceso de ejecución del job según la configuración de ejecución

Figura 61: Diagrama de flujo para el job release hold



Fuente: Los Autores (2025)

5.7.2. Implementación

Para la implementación se desarrollo el comando “release var inventory hold” el cual realiza la liberación automática del inventario, evaluando registros cuya fecha de retención ha superado un umbral específico. Utiliza parámetros configurables como el número de días de retención y el almacén para identificar los registros aplicables y actualizar su estado, asegurando que el inventario vuelva a estar disponible para su uso. Además, incluye validaciones de seguridad, control de errores y auditoría de ejecución, garantizando un proceso eficiente y trazable dentro del sistema.

Figura 62: Lógica del comando release var inventory hold

```

7 /*Consultant: Juan Alvarez
8   Date: 02/05/2024
9   Command created to release the static freeze hold*/
10 [select *|
11   from invlod
12   inner
13     join invsub
14       on invlod.lodnum = invsub.lodnum
15   inner
16     join invdtl
17       on invdtl.subnum = invsub.subnum
18   inner
19     join invhld
20       on invdtl.dtlnum = invhld.dtlnum
21       and invdtl.is_purged = invhld.is_purged
22   join hldmst
23     on invhld.hldnum = hldmst.hldnum
24   where invhld.hlddte < sysdate - @numday
25     and invdtl.hld_flg = '1'
26     AND invhld.hldnum = @hldnum1
27     and invhld.wh_id = nvl(@wh_id, @@wh_id)] catch(-1403)
28 |
29 if (@? = 0)
30 {
31   process inventory hold change
32     WHERE fr_hld_flg = @hld_flg
33     AND hld_flg = @hld_flg
34     AND stoloc = @stoloc
35     AND prtnum = @prtnum
36     AND dtlnum = @dtlnum
37     AND mode = 'GUI'
38     AND prc_hldpfx_list = @hldpfx
39     AND bldg_id = @bldg_id
40     AND acttyp = 'R'
41     AND fr_invsts = @invsts
42     AND prc_hldnum_list = @hldnum1
43     AND subnum = @subnum
44     AND arecod = @arecod

```

Fuente: Los Autores (2025)

Una vez desarrollado el comando se procedió con la configuración del job en la consola moca de la siguiente para que corriera todos los días a las 11:00 AM en todos los ambientes productivos del cliente.

Figura 63: Lógica de configuración del Job Release Hold

Modify Job: Release var Inventory Hold Static Freeze

Job ID: *
RELEASE_VAR_HOLD

Name: *
Release var Inventory Hold Static Freeze

Group Name: Role ID:

Command: *
`list policies where polcod = 'VAR' and polvar = 'RELEASE_HOLD' and polval = 'RELEASE_HOLD' and wh_id = 'PHX01' and rtnum1 = '1' catch(-1403)|release var inventory hold where wh_id = 'PHX01' and hldnum1 = @rtstr1 and reacod1 = @rtstr2 and numday = @rtnum2`

General Environment

Schedule

Time: 11:00 AM AM PM Repeat: Sunday, Monday, Tuesday

Advanced Schedule

Tracing

Tracing Enabled: On

Trace File: \$LESDIR/log/releasevarhold.log

Behavior

Auto Schedule: On Overlap Executions: On

Save Cancel

Fuente: Los Autores (2025)

Una vez finalizada la configuración se finalizó con la realización de pruebas unitarias para verificar el correcto funcionamiento del desarrollo y se documentó en la siguiente matriz de pruebas antes de realizar la instalación en el ambiente productivo.

Figura 64: Matriz de pruebas para el job release var inventory hold

Client	Arcadia Cold		Testing Summary				
Date	6-May-24		Scenarios				
	Review	Test In Review		0		0	
	Pending	Test Pending Evaluation		0		0.00%	
	Completed	Test Completed Successfully		8		100.00%	
	Error	Test In Error		0		0.00%	
	Quality	Test Pending Client Evaluation		0		0.00%	
	Not Applicable	Test NOT Applies in the Process Model		0		0.00%	
	Escalated	Cases Escalated for Blue Yonder		0		0	
	Totals			8		100.00%	

No.	Process	Status	Activity	Details	Environment	Result	Date
1	Execute job RELEASE_VAR_HOLD	Completed	lodnum: 00012001440000123401 with USDA hold number	job correction, previously it did not register the release in the history	NP1 WH: PHX01	The job does what is expected, releases the holds associated with the given policies and records it in the hold history	06/05/2024
2	Test variable day in the policy	Completed	lodnum: 00012001440000123401	Modification of the retention date in the job so that it can be variable regardless of the policy	NP1 WH: PHX01	successfully	06/05/2024
3	Execute job RELEASE_VAR_HOLD	Completed	lodnum: 00012001440000970453 with STATICFREEZING hold number	job correction, previously it did not register the release in the history	NP1 WH: PHX01	The job does what is expected, releases the holds associated with the given policies and records it in the hold history	06/05/2024
4	Test variable day in the policy	Completed	lodnum: 00012001440000970453	Modification of the retention date in the job so that it can be variable regardless of the policy	NP1 WH: PHX01	successfully	06/05/2024
5	Execute job RELEASE_VAR_HOLD	Completed	lodnum: 00012001440000971115 with FPHOLD hold number	job correction, previously it did not register the release in the history	NP1 WH: PHX01	The job does what is expected, releases the holds associated with the given policies and records it in the hold history	06/05/2024
6	Test variable day in the policy	Completed	lodnum: 00012001440000971115	Modification of the retention date in the job so that it can be variable regardless of the policy	NP1 WH: PHX01	successfully	06/05/2024
7	Execute job RELEASE_VAR_HOLD	Completed	lodnum: 00012001440000972372 with USDA hold number	job correction, previously it did not register the release in the history	NP1 WH: PHX01	The job does what is expected, releases the holds associated with the given policies and records it in the hold history	06/05/2024
8	Test variable day in the policy	Completed	lodnum: 00012001440000972372	Modification of the retention date in the job so that it can be variable regardless of the policy	NP1 WH: PHX01	successfully	06/05/2024

Fuente: Los Autores (2025)

5.7.3. Capacitación y pruebas de usuario

Una vez finalizado el desarrollo en el ambiente de desarrollo continuamos con la capacitación con el usuario para el cual se desarrolló un manual de usuario en el cual se detalla cómo usar y configurar la implementación en caso de que requieran cambiar las condiciones de ejecución. Adicionalmente se tuvieron sesiones de capacitación en la cual el cliente pudo realizar pruebas y validar el correcto funcionamiento del job.

5.7.4. Puesta en marcha (Go-Live)

Se estableció la fecha de instalación y por el horario operativo de los ambientes de producción se consideró prudente que la instalación se realizara en horas de la de la madrugada en donde la operación estuviera detenida y no se vieran afectados los ambientes productivos. Una vez instalada la customización, se validó en tiempo real que los comandos se cargaron en el sistema y las funcionalidades funcionaran correctamente

5.7.5. Optimización y soporte

Para garantizar un buen funcionamiento del desarrollo se tuvo durante tres semanas en monitoreo y atendiendo cualquier inconveniente o mal funcionamiento del mismo, este proceso se llevó a cabo mediante el sistema de tickets en donde el cliente reportaba estos incidentes o solicitudes de mejora. Una vez finalizado este seguimiento de tres semanas se procedió con la transición del desarrollo al equipo de soporte, el cual queda como responsable de cualquier incidente que se presente en adelante.

5.7.6. Análisis de resultado

La implementación del job de liberación de restricciones de inventario ha optimizado significativamente la gestión del almacén al automatizar un proceso que anteriormente requería intervención manual. Gracias a su ejecución programada, se garantiza que los productos retenidos sean liberados en el momento adecuado, mejorando la disponibilidad del inventario y optimizando el uso del espacio de almacenamiento. Además, el tiempo de ejecución del proceso es eficiente y no interfiere con otras operaciones críticas, asegurando que la base de datos mantenga su rendimiento óptimo. El manejo de errores y la trazabilidad han sido puntos clave en el desarrollo, ya que se registra cada ejecución en un log de auditoría, permitiendo un monitoreo preciso y evitando modificaciones erróneas en los registros del sistema.

A pesar de los beneficios obtenidos, es recomendable continuar con el monitoreo del job para detectar posibles fallos y evaluar mejoras en la eficiencia de las consultas a medida que el volumen de datos aumente. También sería útil incorporar opciones de configuración más flexibles que permitan modificar dinámicamente los parámetros del proceso sin necesidad de ajustes técnicos. En general, el desarrollo ha cumplido con los objetivos propuestos, logrando una solución eficiente, segura y alineada con las necesidades operativas del almacén, reduciendo la carga administrativa y asegurando que los productos puedan ser utilizados oportunamente.

6. Conclusiones

El desarrollo del proyecto logró optimizar significativamente la eficiencia operativa de NetLogistik y sus clientes. Las necesidades críticas que se abordaron, como la reducción de errores manuales, la mejora en la visibilidad de datos y la automatización de procesos repetitivos, obtuvieron una solución satisfactoria gracias a cada una de las personalizaciones creadas para el WMS. Los apartados en donde se logró un cambio destacado son los siguientes:

1. **Trazabilidad y Precisión:** Soluciones como la *Trailer Release Page* y el *Packing List Report* permitieron un control riguroso del inventario, reduciendo discrepancias y facilitando auditorías.
2. **Automatización:** La integración de *EMS* y *Jobs* eliminó la dependencia de intervención humana en tareas como la generación de documentos (BOL) o la reconciliación de inventario, minimizando errores y acelerando procesos.
3. **Experiencia del Usuario:** Interfaces como *View Inbound Shipments* y *Trailer Release Page* mejoraron la usabilidad del sistema, permitiendo a los usuarios acceder a información clave de manera intuitiva y en tiempo real.
4. **Escalabilidad:** Las soluciones demostraron ser sostenibles y adaptables, sentando las bases para futuras integraciones tecnológicas (ej: IoT, análisis predictivo).

Tabla 1: Comparativa de resultados por customización

Customización	Objetivo Principal	Resultado Obtenido	Impacto Operativo
Trailer Release Page	Actualizar tiempos de liberación de trailers	Reducción de errores manuales.	Mayor control de tiempos de carga/descarga y optimización de estadías de trailers.
View Inbound Shipments	Visualizar y filtrar embarques entrantes	Menor tiempo en la consulta y búsqueda de datos.	Auditorías más rápidas y toma de decisiones basada en datos actualizados.
Packing List Report	Generar reporte detallado por pallet (LPN)	Reducción del tiempo de consolidación de cargas.	Trazabilidad precisa de productos y cumplimiento de normativas de envío.
Bill Of Landing Report (BOL)	Automatizar generación de documentos legales	Menos errores humanos en documentación legal.	Cumplimiento regulatorio y agilización de procesos aduaneros.
EMS BOL	Enviar BOL automáticamente vía correo	Entrega instantánea de documentos	Eliminación de retrasos en comunicación con transportistas y clientes.
Interface Inventory Reconciliation	Reconciliar inventario por ubicación	Reducción del tiempo de consulta de la información y generación de auditorías.	Detección proactiva de inconsistencias y optimización del espacio de almacenamiento.
Release Hold	Liberar retenciones de inventario	Menor tiempo en liberación de inventario retenido.	Agilidad en la gestión de inventario retenido y reducción de cuellos de botella.

Tabla elaborada por los autores. Creada para este documento

7. Recomendaciones

Mantener una buena documentación: A lo largo del desarrollo del proyecto, nos dimos cuenta de lo importante que es llevar un registro claro y bien estructurado de todas las personalizaciones realizadas en el WMS. Documentar cada cambio con diagramas, explicaciones y pruebas facilita no solo el mantenimiento futuro, sino que también ayuda a nuevos integrantes del equipo a comprender mejor el trabajo realizado.

Una posible mejora sería implementar un control de versiones para llevar un mejor seguimiento de los cambios en los comandos y desarrollos del WMS. Esto ayudaría a tener un registro claro de cada modificación y facilitaría volver a versiones anteriores si es necesario. Además, permitiría trabajar de forma más ordenada y evitar problemas como sobrescribir o perder información importante.

Referencias Bibliográficas

- Amazon Web Services. (s.f.). ¿Qué es una API? AWS. Recuperado de <https://aws.amazon.com/es/what-is/api/>
- Blue Yonder. (2023). *Configuration Guide: Integrator (Supply Chain Execution) (Release 2023.1.0.0)*. Recuperado de <https://www.blueyonder.com>
- Blue Yonder. (2023). *Warehouse management systems for the modern supply chain*. Recuperado de <https://www.blueyonder.com>
- Blue Yonder. (2024). *Communication Methods Guide: Integrator (Supply Chain Execution) (Release 2022.1.0.0)*. Recuperado de <https://www.blueyonder.com>
- Brown, R., Taylor, M., & Adams, L. (2021). Advances in IoT for cold chain logistics: Applications and challenges. *Journal of Supply Chain Innovation*, 12(3), 45-58.
- Control.com. (s.f.). *History of warehouse management systems (WMS)*. Recuperado de <https://control.com/technical-articles/history-of-warehouse-management-systems-wms/>
- Hollingsworth, D. (2004). *The Workflow Management Coalition Specification: The Workflow Reference Model*. Workflow Management Coalition.
- IBM. (2021). *SFTP server overview*. Recuperado de <https://www.ibm.com/docs/de/b2badv-communication/1.0.1?topic=concepts-sftp-server-overview>
- Jaspersoft. (s.f.). *Working with data*. Recuperado de <https://www.jaspersoft.com/getting-started/working-with-data>
- Jones, P. (2019). Modular architectures in logistics technology: The case of MOCA. *Logistics Today*, 8(4), 23-34.

Netlogistik. (s.f.). *Aumentó la exactitud del inventario al 97%*. Recuperado de <https://www.netlogistik.com/hubfs/MEXICO/MARKETING/Casos-de-exito/alkosto-Aumento-la-exactitud-del-inventario-al-97.pdf>

Netlogistik. (s.f.). *Servicios de implementación - Blue Yonder*. Recuperado de <https://www.netlogistik.com/es/blueyonder/servicios-implementacion>

Smith, J., & Johnson, K. (2020). Cold chain logistics: A technological perspective. *International Journal of Logistics Management*, 31(2), 150-169.

Van Der Aalst, W. M. P. (2016). *Process Mining: Data Science in Action* (2nd ed.). Springer.

Apéndice A: Descripción de la empresa

Netlogistik es una empresa especializada en soluciones integrales de logística y cadena de suministro que se destaca por su enfoque vanguardista en la gestión de operaciones comerciales. Ofrecen una amplia gama de servicios que abarcan desde la optimización de rutas y la gestión de almacenes hasta la implementación de sistemas de seguimiento y control de inventario. Netlogistik se caracteriza por su compromiso con la eficiencia y la tecnología, proporcionando a sus clientes herramientas avanzadas para mejorar la visibilidad y el rendimiento de sus operaciones logísticas. Con años de experiencia en el sector, la empresa se ha convertido en un referente en la industria de logística y cadena de suministro, brindando soluciones personalizadas que ayudan a optimizar procesos y reducir costos para organizaciones de diversos sectores, entre sus productos se encuentran sistemas logísticos informativos como los son el WMS (Warehouse Management System) y el TMS (Transport Manager System) quienes apoyan a grandes industrias dedicadas a la cadena de suministros a optimizar sus costes y facilitan la gestión de inventario y distribución de sus productos.

Apéndice B: Certificado de Finalización de Prácticas Empresariales Juan AlvarezTransforming **digital supply chains****NET LOGISTIK COLOMBIA S.A.S.**
Nit. 900.229.761-6**HACE CONSTAR QUE:**

El señor **JUAN DIEGO ALVAREZ SANCHEZ**, identificado con Cédula de Ciudadanía No. 1.004.864.057 expedida en Bucaramanga, realizo sus prácticas empresariales como modalidad de trabajo de grado en nuestra compañía, comprendidas desde el 30 de enero de 2024 hasta el 30 de julio de 2024, cumpliendo con los objetivos propuestos y las actividades asignadas para el desarrollo de la misma.

La presente certificación se expide a los treinta (30) días de julio del 2024 con destino a: **UNIVERSIDAD INDUSTRIAL DE SANTANDER**

Atentamente,

Johanny Navarro (30 jul. 2024 12:08 CDT)

JOHANNY CECILIA NAVARRO MOTTA
PROFESIONAL – TUTOR DEL ESUDIANTE www.netlogistik.com info@netlogistik.com**MÉXICO**Bivd. Adolfo López Mateos
#2235 Piso 5, Alvaro Obregón
C.P. 01710, CDMX
+(52) 55 5282 1321**COLOMBIA**Carrera 51 No. 96-23
Oficina 203 Bogotá D.C.
+57 (1) 656 3285**USA**12873 W. Harvard Avenue
Denver, CO 80228
+1 (262) 424 5745**ESPAÑA**Via Augusta No 29-31 6ª
Planta, 08006, Barcelona
+34 935 951 420

Apéndice C: Certificado de Finalización de Prácticas Empresariales Luis LinaresTransforming **digital supply chains**

NET LOGISTIK COLOMBIA S.A.S.
Nit. 900.229.761-6

HACE CONSTAR QUE:

El señor **LUIS ALEJANDRO LINARES TEJEDOR**, identificado con Cédula de Ciudadanía No. **1.002.458.255** expedida en Bucaramanga, realizo sus prácticas empresariales como modalidad de trabajo de grado en nuestra compañía, comprendidas desde el 30 de enero de 2024 hasta el 30 de julio de 2024, cumpliendo con los objetivos propuestos y las actividades asignadas para el desarrollo de la misma.

La presente certificación se expide a los treinta (30) días de julio del 2024 con destino a: **UNIVERSIDAD INDUSTRIAL DE SANTANDER**

Atentamente,

Johanny Navarro

Johanny Navarro (30 jul., 2024 12:08 CDT)

JOHANNY CECILIA NAVARRO MOTTA
PROFESIONAL – TUTOR DEL ESUDIANTE

 www.netlogistik.com

 info@netlogistik.com

MÉXICO

Blvd. Adolfo López Mateos
#2235 Piso 5, Álvaro Obregón
C.P. 01710, CDMX
+(52) 55 5282 1321

COLOMBIA

Camera 51 No. 96-23
Oficina 203 Bogotá D.C.
+57 (1) 656 3285

USA

12873 W. Harvard Avenue
Denver, CO 80228
+1 (262) 424 5745

ESPAÑA

Via Augusta No 29-31 6ª
Planta, 08006, Barcelona
+34 935 961 420