

2017

DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE INFORMACIÓN COMO APOYO A LA GESTIÓN DE DISPONIBILIDAD DE PRODUCTOS DE CONSUMO MASIVO EN GÓNDOLAS DE SUPERMERCADO

CARRASCO CARRASCO, ROBERTO LEONARDO

<http://hdl.handle.net/11673/14080>

Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

Universidad Técnica Federico Santa María
Departamento de Informática
Santiago - Chile



**Diseño e Implementación de Sistema de Información
como Apoyo a la Gestión de Disponibilidad de Productos
de Consumo Masivo en Góndolas de Supermercado**

Roberto Carrasco Carrasco

Memoria para optar al título de Ingeniero Civil Informático

Profesor Guía: José Luis Martí Lara

Abril 2017

DEDICATORIA

A mis papás, Dorothy y Roberto; y a mis hermanos, Engelbert y Ariel; por brindarme apoyo, motivación y nunca perder la ilusión de verme convertido en ingeniero. Decirles que son el pilar de mi vida y gracias a ustedes es posible que esto suceda.

Resumen

El presente trabajo desarrolla la creación de un sistema informático como apoyo a la gestión de disponibilidad de productos de consumo masivo en góndolas de supermercado. La necesidad surge debido a que la organización no cuenta con un sistema que procese la gran cantidad de datos de venta e inventario de sus productos, en cada una de las salas de las cadenas de supermercado en donde está presente, datos puestos a su disposición periódicamente por cada una de ellas, bajo el marco de alianzas estratégicas establecidas entre supermercados y proveedores denominado trabajo colaborativo o modelo CPFR.

El resultado obtenido con este trabajo es un sistema que, con la implementación de un nuevo indicador de desempeño (KPI), permitió a la organización modificar su forma de trabajo (enfocándose mayormente en el proceso), poder hacer seguimiento evolutivo de sus equipos y mejorar la optimización de los espacios destinados a sus productos en las góndolas de los distintos supermercados en donde está presente.

Palabras Claves— Gestión, Trabajo colaborativo, KPI.

Abstract

This work develops the creation of a computer system to support the management of the availability of mass consumer products in supermarket shelves. The need arises because the organization does not have a system that processes the large amount of data of sales and inventory of its products, in each one of the stores of the supermarket chains where it is present, data made available to it periodically by Each under the framework of strategic alliances established between supermarkets and suppliers called collaborative work or model CPFR.

The result obtained from this work is a system that, with the implementation of a new performance indicator (KPI), allowed the organization to modify its way of work (focusing mainly on the process), to be able to follow up evolution of its equipment and to improve the optimization of the spaces destined to its products in the shelves of the different supermarkets where it is present.

Keywords— Management, Collaborative work, KPI

INTRODUCCIÓN

El mundo empresarial se ha visto enfrentado, sin apenas percibirlo, a una revolución profunda, similar a la iniciada en el siglo XIX con el desplazamiento del valor de la fuerza del músculo por el rendimiento de la máquina de vapor. Esa revolución es la entrada a una sociedad basada en la información [Zimmermann00].

Es en este contexto donde una importante compañía de Retail decide modificar su modelo de gestión de datos que dispone bajo el Modelo de Trabajo Colaborativo que establece con sus distintos socios estratégicos, conformado por grandes cadenas de supermercados, con el fin de contar con información de mejor calidad a la hora de la toma de decisiones en el ámbito de gestión de reabastecimiento de sus productos a las distintas salas.

El problema particular a abordar es el manejo de la cantidad de datos que cada uno de sus clientes le provee para el análisis de sus propios productos en las distintas salas de supermercado; dado el gran volumen de éstos, es que se dedica gran parte de la jornada al análisis por no contar con una herramienta que lo haga de forma automática.

Por lo anterior es que la se decide diseñar e implementar un sistema de información que realice la labor de análisis y generación de reportes para cuatro cadenas de supermercados que son parte de los principales clientes y socios estratégicos.

Para el diseño de éste es que se realiza un levantamiento del proceso manual que se realiza previo trabajo de esta memoria, y en función de esto se desarrolla e implementa un sistema que apoya y da soporte a la toma de decisiones para las distintas áreas involucradas en el proceso de gestión de la correcta exhibición de productos en una sala de supermercado.

La estructura del documento consta de 4 capítulos, comenzando por el capítulo 1 denominado "Definición del problema", se podrá consultar conceptos claves para entender el contexto en donde se desarrolla, así como también los objetivos y alcances de la solución. Luego en el capítulo 2, denominado "Marco Conceptual", se establecen distintas metodologías, técnicas y conceptos a fin para entender de mejor forma el desarrollo de este trabajo. En el capítulo 3, denominado "Solución Propuesta", se presenta el desarrollo mismo de la solución y las etapas que lo contemplan. En el capítulo 4, denominado "Validación de la solución propuesta", se muestra la nueva forma de gestión de la información y cifras de desempeño de la empresa previo y posterior a la implementación del desarrollo. Finalmente, se brindan conclusiones del trabajo realizado.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

En este capítulo se procede a explicar la razón de este trabajo, detallando el contexto en donde se realiza, se presenta el problema que se propone resolver, así como las circunstancias previas al desarrollo de la solución. Para finalizar, se muestran tanto los objetivos como los alcances de la solución implementada.

1.1. Contexto

El desarrollo de este proyecto es en una empresa multinacional de Retail dedicada a productos alimenticios de consumo masivo, presente en todas las cadenas de supermercados. En cada una de ellas puede tener incorporados para venta 400 ítemes en promedio, lo que llevado la cantidad de salas disponibles de una cadena entrega un resultado aproximado de 35.000 combinaciones posibles de productos habilitados en las góndolas (estantes) de los supermercados.

El trabajo colaborativo bajo el modelo CPFR (por sus siglas en inglés *Collaborative Planning, Forecasting and Replenishment*) desarrollado a mediados de los años 90, comprende los procesos de planeamiento, proyecciones y reabastecimiento entre socios comerciales para cooperar sobre sus estrategias, tácticas y ejecución de planes productivos, y de este modo reducir o eliminar la incerteza de abastecimiento/demanda a través de mejoras en la colaboración/comunicación [Hollmann15].

Bajo dicho modelo es que cada una de estas grandes cadenas de supermercados pone a disposición datos de inventario y venta (entre otros especificados más adelante) para que su socio estratégico, de ahora en adelante la Compañía, pueda realizar análisis de disponibilidad, movimientos de ventas y proyecciones tanto de compra como de venta.

A continuación, se procederá a detallar conceptos técnicos para facilitar el entendimiento de la solución propuesta dado que impacta diferentes áreas de la Compañía, no sólo del área en donde se desarrolla el proyecto, sino que también a su cliente, tanto interno como externo.

- **Cadenas de Supermercado:** corresponden a un segmento empresarial complejo y diverso, pero con característica similares. Son los distintos clientes externos a los que La Compañía le provee sus productos. Ellos son la fuente que va a disponer sus datos de diferente forma al correspondiente analista asignado a cargo del trabajo colaborativo.
- **Salas de supermercado:** cada cadena cuenta con al menos 300 salas de supermercados distribuidas en todas las regiones del país, éstas se pueden abastecer de productos de distinta forma. Para el contexto de la solución no es necesario indagar tanto en este tema, sin embargo, resulta importante entender los siguientes conceptos:

- Abastecimiento directo: sala o local de supermercado que se abastece de productos mediante el despacho de productos que La Compañía realiza de forma directa; explicado de forma simple, camión que va desde el Centro de distribución de la Compañía hasta la sala de supermercado.
 - Abastecimiento centralizado: en este formato de salas los productos llegan a través del denominado “camión de la casa”, esto implica que son los Centros de Distribución de la misma cadena de supermercados los que envían camiones con los productos que se van a vender, previa compra a La Compañía quien ya ha realizado el despacho a dicho centro. Dado que son almacenes de gran tamaño, pueden almacenar productos para cubrir varios días de venta.
- **Analista de abastecimiento:** corresponde al analista por parte de cada cliente (cadena de supermercado). Es quien administra, modifica y gestiona los datos que tendrá disponible la Compañía. Es el contacto del analista de la compañía con la cadena, con quien sostiene reuniones semanales para revisar indicadores; es en esta instancia en donde se presentan análisis que permiten realizar correcciones en el sistema del cliente.
 - **CPFR Analyst:** corresponde al analista por parte de La Compañía encargado de la extracción y proceso de los datos, además es quien lidera las acciones que apoyan al equipo de Gestión Comercial y Fuerza de Ventas en indicadores de gestión. Es quien debe escalar las necesidades que la cuenta requiera y quien entrega la información que apoya la toma de decisiones a las áreas mencionadas anteriormente.
 - **Gestor de venta:** son el cliente interno del área en donde se desarrolla el sistema. La Compañía, dentro de su estructura de gestión, ha determinado un equipo de encargados de revisar y realizar distintas gestiones en terreno de un grupo de salas de supermercado, esta división puede ser tanto por cadena como por región, es decir, puede que el gestor de venta tenga un grupo de salas de supermercados de la misma cadena bajo su gestión o puede tener una zona geográfica asignada sin importar a que cadena corresponda la sala. Es quien gestiona en terreno (directo con los jefes de las salas de supermercado) la información que el CPFR Analyst le provee.

La periodicidad establecida para la extracción y análisis de los datos desde los clientes es semanal, pues permite tener una mirada del movimiento de los productos de un periodo que, por un lado, es lo suficientemente actualizado para realizar el análisis, pero que también contempla la estacionalidad que se produce en una semana (venta en días hábiles versus la que se produce los fines de semana). Adicionalmente cada uno de los clientes pone a disposición los datos con distintos formatos y por distintos medios (portales B2B, descarga desde sus sistemas en forma presencial y envíos por correo

electrónico). Este modelo de trabajo se mantendrá a través de todo el desarrollo e implementación del nuevo sistema.

1.2. Forma de trabajo previo al desarrollo

La forma en que los datos son procesados es por separado según cada CPFR Analyst a cargo de cada cuenta usando la herramienta de hojas de cálculo Excel. Lo que esencialmente se realiza es, independiente de la forma de obtención de éstos, depurar los datos para llegar a una cantidad mínima y más manejable de información. Finalmente, se generan reportes en esta misma herramienta.

Para mantener un lenguaje común entre ambas organizaciones (Compañía y Cadena de Supermercado) es que se controlan KPI's o indicadores de rendimiento (por sus siglas en inglés *Key Performance Indicator*), los cuales se detallan a continuación:

- **DOH (*Days on hand*) días de inventario:** mide la rotación de un producto en góndola, según su variación es cuan rotador llega a ser un producto. Además, indica cuántos días de venta se pueden cubrir con el stock disponible según la venta estimada de un día.
- **Instock:** mide la disponibilidad de producto en unidades para un día de venta; es una variable de tipo binaria cuyo valor es 1 si la cantidad disponible en sala (góndola y bodega) es superior a la venta calculada para cubrir un día, de lo contrario toma valor 0.

Los indicadores mencionados anteriormente son calculados en base a datos de venta y disponibilidad de productos a nivel de sala por cadena; éste es el nivel más detallado de información que se puede tener de un producto.

Previo al desarrollo propuesto, la Compañía no posee un sistema que gestione esos datos. La forma en que se trabaja y transmite la información, dentro de la organización y hacia el cliente, es a través de reportes que muestran los KPI's anteriormente mencionados, calculados para el total de las combinaciones producto-sala, en planillas de cálculo donde se muestran tablas dinámicas que permiten realizar distintos filtros a dicha información.

Resulta importante también entender la forma en que los equipos trabajan previo al desarrollo del presente trabajo, esto debido que la solución propuesta producirá un impacto en la forma en que gestionan la información. Como se ha mencionado, la información se transmite a través de reportes que no llevan un análisis minucioso de los datos, solo cálculo de KPI; por ende, cada una de los actores antes mencionados, toma estos reportes y realiza sus propios análisis para luego gestionar lo que cada uno considera importante.

La **Imagen 1.1** corresponde al diagrama de actividades que muestra como el *CPFR Analyst* genera el mencionado reporte, y como a su vez gestiona la información que obtiene de éste con el analista de abastecimiento por el lado del cliente.

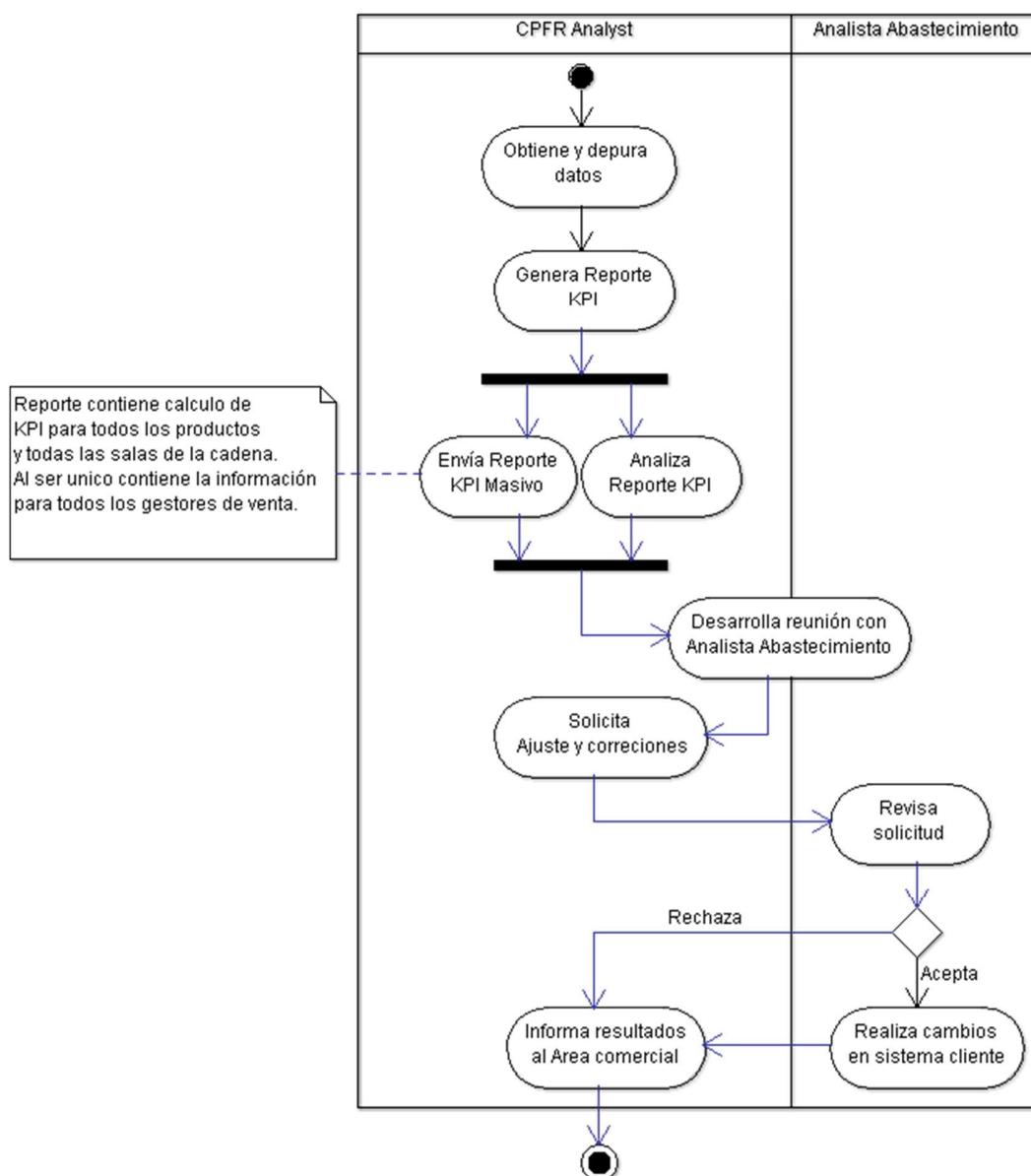


Imagen 1.1: Diagrama de actividades para la Gestión del CPFR Analyst en donde se genera el reporte y se gestiona la información con el analista de abastecimiento de la cadena.

Fuente: Creación Propia

La **Imagen 1.2** muestra la forma en que el Gestor de venta ejecuta su gestión con la sala de supermercado. Al compararla con la **Imagen 1.1** se ve un proceso similar de gestión, sin

embargo, entre ambos no existe comunicación efectiva, ni claridad de qué va a gestionar cada uno.

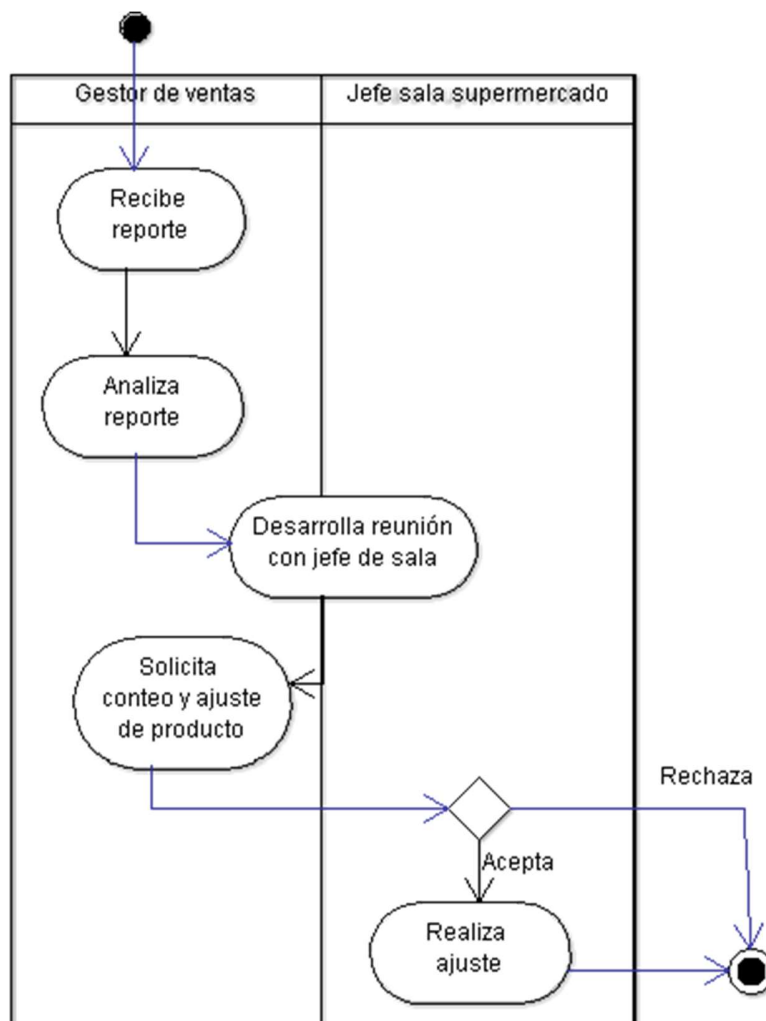


Imagen 1.2: Diagrama de actividades que realiza el gestor de venta posterior a que recibe el reporte de KPI

Fuente: Creación Propia

1.3. Objetivo de la Solución

Considerando lo expuesto anteriormente es que el presente trabajo tiene como objetivo “Diseñar e Implementar un sistema de información como apoyo a la gestión de disponibilidad de productos de consumo masivo en góndolas de supermercado”. Esto implicará tener mejor análisis del aprovisionamiento de las góndolas de supermercado soportado por tecnologías de la información, y ajustar los procesos de gestión que se tienen actualmente.

Para el cumplimiento de esto se debe lograr:

1. Analizar los datos disponibles para realizar un análisis transversal del estado actual de los stocks, permitiendo que los mismos indicadores sean usados en distintas Cadenas de supermercado, manteniendo la homogeneidad y capacidad de comparación del sistema.
2. Implementar un indicador global que permita evaluar el cumplimiento y desempeño real de un producto en góndola.
3. Entregar, periódica y sistémicamente, reportes en base a las mediciones obtenidas para el apoyo a las decisiones que serán tomadas por el área encargada de la gestión de productos en los distintos puntos de venta (salas de supermercado).

A nivel general lo primero que se busca es un mayor análisis de los datos disponibles, no sólo a nivel de cada Cadena sino a nivel Compañía. Esto implica unir toda la información que cada Cadena provee, generar vistas que permitan hacer una comparación del estado de un producto entre una Cadena y otra; o simplemente poder tener una vista global del indicador y desde ahí realizar la apertura del dato al nivel que se desee llegar; partir con el indicador a nivel Compañía, poder mirarlo a nivel Cadena, luego a nivel zona geográfica hasta finalmente llegar a nivel de la sala.

Es por lo anterior que resulta importante implementar un nuevo KPI que permita relacionar varias variables y mejorar la calidad del análisis, esto aportaría en una mejor gestión y por ende más incidencia en la toma de buenas decisiones. Como se menciona en el punto anterior, hasta el momento no se hace una relación entre indicadores para llegar a un análisis más depurado, tampoco se tiene un foco de productos ni de salas a los que se quiere dar mayor relevancia.

Por el lado de la entrega de dichos análisis es que se busca un cambio en la forma en que los reportes actualmente se presentan, esto implica reducción en la cantidad de información; el objetivo está en entregar la información más depurada y con enfoque en los puntos críticos a los cuales atacar. Por último, se debe implementar una vista evolutiva que permita controlar la gestión que se realiza dado que los cambios/ajustes solicitados, tanto en las salas como en los sistemas de los clientes, no siempre se hacen efectivos.

1.4. Alcances de la Solución

Inicialmente el enfoque será disminuir los tiempos de análisis para que cada analista pueda empeñar el mayor tiempo de su día en la gestión de los resultados de éstos. Por esta razón es que el desarrollo no considerará mejora en la forma de la obtención de la información.

Los clientes que serán considerados para la implementación de la aplicación son los que estén dentro del trabajo colaborativo, y que cumplan con la condición de disponibilidad de los datos mínimos requeridos para el correcto análisis (esta condición será explicada en

mayor detalle en el Capítulo 3). Como resultado de este filtro es que se determina trabajar con las 4 principales cadenas de supermercado con los que se tiene una relación estratégica.

CAPÍTULO 2: MARCO CONCEPTUAL

En este capítulo se revisarán los conceptos y técnicas asociadas a las metodologías de desarrollo de software, lenguaje UML y arquitectura de software. El objetivo es poder definir y delimitar el contenido más adecuado utilizado como referencia en el desarrollo de este trabajo.

2.1. Metodologías de desarrollo de software

Una metodología se define como un enfoque para el desarrollo de software que incluye modelos de sistemas, notaciones, reglas, sugerencias de desarrollos y guías de proceso. La **Tabla 3.1** muestra las diferencias entre métodos estructurados (también llamados métodos tradicionales) y metodologías ágiles de desarrollo de software; ambos serán detallados posteriormente.

Métodos estructurados	Metodologías ágiles
Más artefactos. El modelo es esencial, mantenimiento de modelos.	Pocos artefactos. El modelo es prescindible, modelos desechables.
Más roles y más específicos.	Pocos roles, más genéricos y flexibles.
Existe un contrato prefijado.	No existe un contrato tradicional, debe ser bastante flexible.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes.	Orientada a proyectos pequeños, y en el mismo lugar.
Se promueve que la arquitectura se defina tempranamente en el proyecto.	La arquitectura se va definiendo y mejorando a lo largo del proyecto.
Énfasis en la definición del proceso: roles, actividades y artefactos.	Énfasis en los aspectos humanos: el individuo y el trabajo en equipo.
Basadas en la definición del proceso: roles, actividades y artefactos.	Basadas en heurísticas provenientes de prácticas de producción de código.
Se espera que no ocurran cambios de gran impacto durante el desarrollo.	Se espera cambios durante el desarrollo del proyecto

Tabla 3.1: principales diferencias entre Metodologías Ágiles y estructuradas para el desarrollo del sistema.

Fuente: [Cendejas14]

2.1.1. Métodos estructurados

Acorde a lo expuesto por Ian Sommerville [Sommerville05], un modelo de procesos de software es una descripción simplificada de un proceso de software que presenta una

visión de ese proceso. Estos modelos pueden incluir actividades que son parte de los procesos y productos del software; y el papel de las personas involucradas en la ingeniería de software.

La mayor parte de los modelos de procesos del software se basan en modelos de procesos genéricos que se utilizan ampliamente en la práctica actual de la ingeniería de software. No son excluyentes entre si y a menudo se suele utilizar juntos. Estos modelos son:

a. Modelo en cascada

Es el primer modelo de proceso de desarrollo de software que se publicó y derivó de procesos de ingeniería de sistemas más generales. Se conoce con este nombre debido a que el modelo va de una fase a otra. Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo:

1. Análisis y definición de requerimientos. Los servicios, restricciones y metas del sistema se definen a partir de consultas con los usuarios. Entonces, se definen en detalle y sirven como una especificación del sistema.
2. Diseño del sistema y del software. El proceso de diseño del sistema divide los requerimientos en sistemas de hardware y software. Establece una arquitectura completa del sistema. El diseño del software identifica y describe las abstracciones fundamentales del sistema de software y sus relaciones.
3. Implementación y prueba de unidades. Durante esta etapa, el diseño del software se lleva a cabo como un conjunto de unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación.
4. Funcionamiento y mantenimiento. Por lo general (aunque no necesariamente), ésta es la fase más larga del ciclo de vida. El sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en etapas previas al ciclo de vida, mejorar la implementación de las unidades del sistema y evaluar modificaciones si es que se descubren nuevos requerimientos.

En principio, el resultado de cada fase es uno o más documentos aprobados. La siguiente fase no debe empezar hasta que la fase previa haya finalizado. En la práctica, estas etapas se superponen y proporcionan información a las otras. Durante el diseño se identifican problemas con los requerimientos; durante el diseño del código se encuentran problemas y así sucesivamente.

La ventaja principal del modelo en cascada es que la documentación se produce en cada fase. Su principal problema es su inflexibilidad al dividir el proyecto en distintas etapas; se deben hacer compromisos en etapas iniciales, lo que hace difícil responder a cambios en los requerimientos del cliente. Por lo tanto, el modelo en cascada es recomendado usarlo

cuando los requerimientos se comprendan bien y sea improbable que cambien radicalmente durante el desarrollo del sistema.

b. Desarrollo evolutivo

Se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios de los usuarios y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado. Las actividades de especificación, desarrollo y validación se entrelazan en vez de separarse, con una rápida retroalimentación entre éstas.

La ventaja de un proceso de desarrollo de software que se basa en un enfoque evolutivo es que la especificación se puede desarrollar de forma creciente. Tan pronto como los usuarios desarrollen un mejor entendimiento del problema, éste se puede reflejar en el sistema de software. Las desventajas de este modelo tienen que ver con una perspectiva de ingeniería y gestión, dado que este modelo se enfoca en entregas regulares para medir el progreso hace que el proceso como tal no sea visible. Además, si los sistemas se desarrollan rápidamente, no es rentable producir documentación que refleje cada versión del sistema.

c. Ingeniería de software basada en componentes

En la mayoría de los proyectos de software existe algo de reutilización de software. En general esto se da de forma informal dado que las personas que trabajan en los proyectos tienen experiencia en diseños o códigos similares. Sin embargo, ha surgido un enfoque de desarrollo de software denominado ingeniería del software basada en componentes (CBSE). Esta técnica supone que las partes del sistema ya existen y el proceso de desarrollo se enfoca en la integración de las mismas más que en crear el código desde el principio.

La ventaja más obvia que tiene este modelo es que reduce la cantidad de software a desarrollarse, así reduce costos y riesgos. Además, permite entrega temprana del software. La desventaja que tiene es que puede resultar en un sistema que no cumpla con las necesidades reales de los usuarios al no apegarse completamente a los compromisos hechos en los requerimientos.

2.1.2. Metodologías Ágiles

Nacidas para dar respuesta al entorno siempre cambiante y en rápida evolución en que se han de desarrollar los sistemas informáticos. En lugar de hacer proyectos pensados como una sola pieza y perfectamente estructurados en fases (una detrás de otra), se plantea trabajar en ciclos cortos (como mini proyectos) que implementan una parte de las funcionalidades, pero sin perder el rumbo general del proyecto global.

Su objetivo es permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que podrían surgir a lo largo de los proyectos, basados en valores y principios, ofreciendo una alternativa a los modelos estructurados o tradicionales, mayormente enfocados en el proceso y la documentación.

Los principios de las metodologías ágiles son:

- Participación del cliente: los clientes deben estar implicados en todo proceso de desarrollo. Su papel principal es proporcionar e interiorizar nuevos requerimientos del sistema, además de evaluar el resultado de las iteraciones.
- Entrega incremental: el software se desarrolla en incrementos, donde el cliente especifica los requerimientos a incluir en cada uno de éstos.
- Personas, no procesos: se deben reconocer y explotar las habilidades del equipo de desarrollo. Se les debe dejar desarrollar sus propias formas de trabajar, sin procesos formales.
- Aceptar el cambio: se debe contar con que los procesos del sistema cambian, por lo que el sistema se diseña para dar cabida a estos cambios.

2.2. Lenguaje de modelado unificado UML

Independiente de la metodología que se utilice para el desarrollo de software, la notación estandarizada para modelado de sistemas de software; lenguaje útil y universal, que ayuda con la especificación y documentación, es la denominada notación UML (sigla que proviene del nombre en inglés *Unified Model Language*), por medio del uso de diagramas, asegura que las funcionalidades del sistema estén completas y sean correctas.

Esta notación presenta varios tipos, los cuales se dividen en dos categorías:

1. Diagramas de estructuras: pone énfasis en la estructura estática que debe estar presente en el sistema que está siendo modelado. Por ejemplo, el diagrama de componentes describe como un sistema de software se divide en componentes y muestra las dependencias entre estos componentes.
2. Diagramas de comportamiento: pone énfasis en lo que debe ocurrir en el sistema que está siendo modelado, es usado para describir funcionalmente el sistema. Por ejemplo, el diagrama de actividades describe el flujo de trabajo (paso a paso) de las actividades de negocio u operacionales de los componentes del sistema.

La **Imagen 2.1** muestra los diferentes diagramas que esta notación ofrece. Notar que los diagramas de comportamiento incluyen un subgrupo denominado Diagrama de Interacción cuya finalidad es el flujo de control y datos de los componentes del sistema que está siendo modelado.

2.3. Arquitectura de software

De acuerdo a lo expuesto por Ian Sommerville [Sommerville05], los grandes sistemas siempre se descomponen en subsistemas que proporcionan algún conjunto de servicios relacionados. El proceso de diseño inicial que identifica estos subsistemas y establece un marco para el control y comunicación de los subsistemas se llama diseño arquitectónico. El resultado de este proceso de diseño es una descripción de la arquitectura del software.

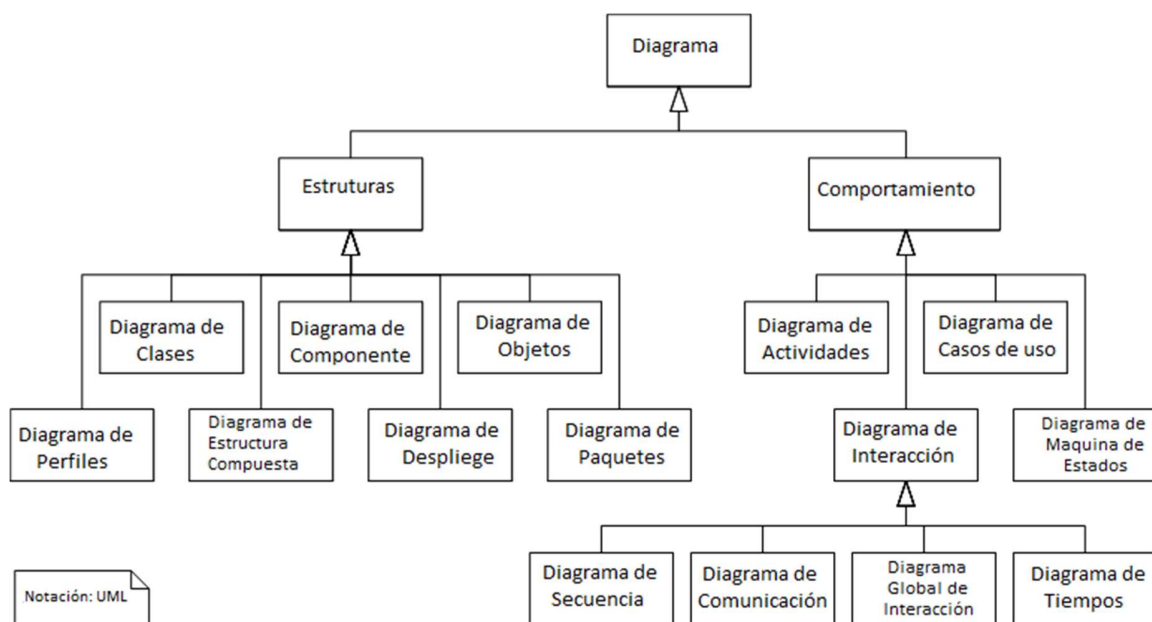


Imagen 2.1: Jerarquía de diagramas en UML 2.2, mostrado como diagrama de clases.

Fuente: Unified Modeling Language [UMLs.f]

El diseño arquitectónico es la primera etapa del diseño y representa un enlace crítico entre los procesos de ingeniería de diseño y de requerimientos. La arquitectura de software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de estos componentes según se le percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o aplazamiento del detalle inherente a la mayor parte de las abstracciones.

Existen diferentes estilos y patrones de arquitecturas de software. Sin embargo, para el propósito de este capítulo solo se hará referencia en particular al estilo Modelo-Vista-Controlador (MVC, que proviene del inglés *Model-View-Controller*), el cual forma parte de

los estilos denominados de “llamada y retorno”, cuya familia enfatiza la modificabilidad y escalabilidad [Reynoso04].

Este tipo de arquitectura separa la lógica de la aplicación de la lógica de la vista de la aplicación usando los siguientes componentes:

- **Modelo:** se encarga de los datos; su almacenamiento y recuperación. Generalmente (pero no de forma obligatoria) consultando la base de datos.
- **Vista:** es el componente encargado de la visualización, responsable de mostrar toda o una porción de los datos de la aplicación. Son los elementos que el usuario recibe y que puede interactuar.
- **Controlador:** recibe las peticiones del usuario y se encarga de solicitar los datos al modelo y comunicarlos a la vista.

La **Imagen 2.2** representa la interacción entre los componentes de un sistema basado en arquitectura MVC y el usuario.

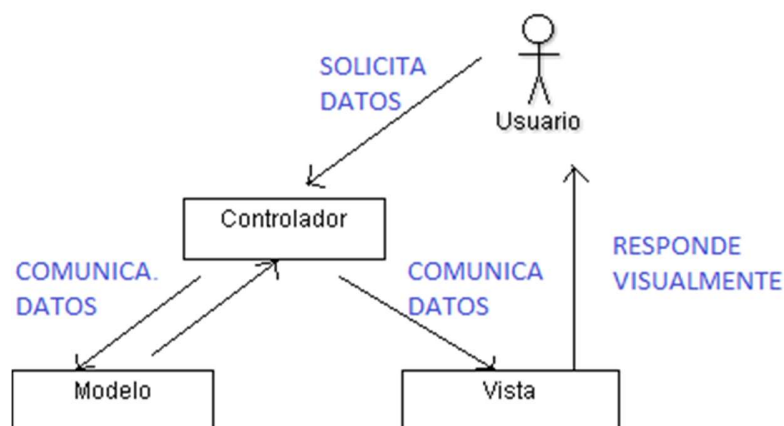


Imagen 2.2: Representación de Estilo de Arquitectura MVC.

Fuente: Creación Propia

CAPÍTULO 3: SOLUCION PROPUESTA

Este capítulo muestra la forma en que fue abordada la solución al problema presentado en el Capítulo 1. Primero se presenta la metodología definida para el desarrollo de la aplicación; acá se muestra el modelo del flujo de trabajo. A su vez se detalla la nueva forma de Gestión de la información y el cambio operacional que se produce posterior a la puesta en marcha del sistema, dado que esto también produjo un cambio en el proceso de trabajo para el equipo y es parte del aporte de esta memoria.

3.1. Metodología a usar

La metodología operante está basada en los métodos estructurados expuestos en el marco conceptual. El modelo de proceso de software definido es de tipo cascada, dado que los requerimientos del sistema fueron definidos de forma temprana, estableciendo que no iban a ser modificados ya que se trata de un sistema pequeño. El proceso de desarrollo se trabajó en 4 etapas bien definidas las cuales se detallan en la **Imagen 3.1**. Cada una de estas etapas son sucesivas, pero no secuenciales, esto significa que no se esperó el término de una etapa para comenzar la siguiente.

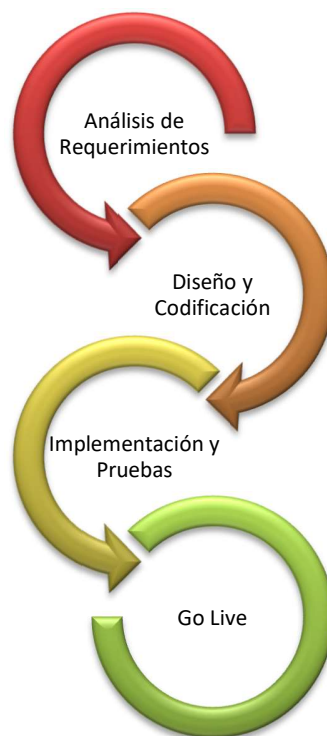


Imagen 3.1: Metodología para el desarrollo del sistema.
Fuente: Creación Propia

3.2. Desarrollo de la propuesta

Análisis de Requerimientos

Para determinar las funcionalidades requeridas por el sistema se desarrolló un *workshop* que incluyó al equipo de analistas, gerente y jefatura del área, además de un analista de sistemas; todos pertenecientes a la Compañía. Esto tuvo dos objetivos, levantar los requerimientos para la funcionalidad del sistema; determinar el nuevo KPI. Para esto, se realiza una “inspección del proceso manual” de la forma en que se revisaban los datos para llegar al resultado.

Es importante mencionar que cada analista realiza un análisis “artesanal” a la información. El trabajo tiene más que ver con la iniciativa del analista, no existe un método establecido, por lo que semana a semana se cae en redefiniciones de qué se va a analizar. Finalmente, al no tener establecido el método, se pierde conocimiento de lo previamente aplicado.

A continuación, se estableció que el sistema debía tener las siguientes funcionalidades:

- La aplicación controlará el acceso y lo permitirá solamente a usuarios autorizados. Los usuarios ingresarán con un nombre de usuario y contraseña.
- La aplicación debe ser de escritorio y podrá ser utilizada en el sistema operativo Windows.
- La aplicación deberá permitir carga semanal de datos de *Instock* diferenciado por cadena en formato único.
- La aplicación deberá proveer módulo de mantención (incorporación, modificación y eliminación) de los datos maestros como asociados a salas de supermercado, gestores de venta y productos.
- La aplicación deberá entregar los siguientes reportes:
 - Gráfico ADP *Instock*: gráfico de torta del *Instock* semanal.
 - Gráfico Evolución de *Instock*: Gráfico de líneas con la evolución del *Instock* en el tiempo (Desde-Hasta).
 - Gráfico Comparativo de *Instock*: gráfico de columnas con la evolución del *Instock* en el tiempo (Desde-Hasta), comparativo por clientes.
 - Gráfico para la medición de indicador de *sell out* fruto de la gestión del equipo.
- La aplicación generará un Informe semanal de Gestores de venta. Además, permitirá enviarlo por correo Outlook a cada uno de ellos.

Diseño y Codificación

Para cumplir con los requerimientos de la etapa previa, el sistema tendrá 3 módulos cuyas funcionalidades se detallan a continuación:

1. Carga y mantención: será el cual permitirá el ingreso de datos al sistema y la mantención de los datos maestros. Ambas funciones deben cumplir los requerimientos previamente definidos.
2. Consultas: tendrá la opción de realizar distintas consultas a medida y permitirá tener información desde el macro (indicadores a nivel compañía) al micro (información de un producto en una sala), pudiendo seleccionar periodos de tiempo (desde – hasta). A su vez generará los gráficos establecidos en los requerimientos
3. Escritor de reportes: generará los reportes y permitirá el envío de éstos mediante correo electrónico.

La arquitectura del sistema se asemeja a las de tipo MVC (modelo-vista-controlador). El componente controlador serán los módulos de Carga y mantención y de Consultas que van a recibir solicitudes del usuario; la lógica del sistema será el componente Modelo dado que interactuará directamente con la base de datos; la Vista será el módulo Escritor de reportes. Esta arquitectura se muestra en la **Imagen 3.2**.

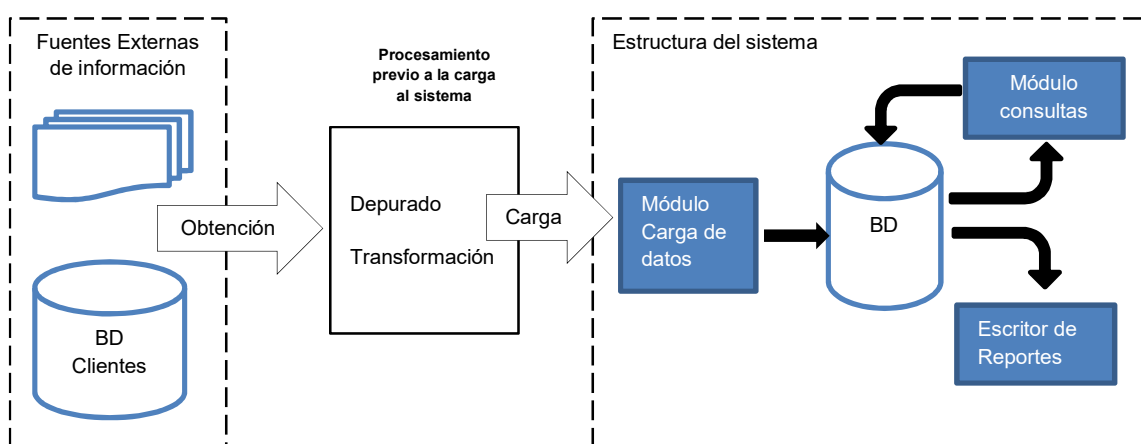


Imagen 3.2: Modelo para el diseño del sistema.
Fuente: Creación propia

La **Tabla 3.1** muestra el formato único de datos que tendrá el módulo de carga. Dada la importancia de tener bien definidos los datos de entrada al sistema, se establecen los campos para el archivo que deberá ser cargado semanalmente.

Nombre columna	Descripción
Store_ID	Número de sala asignado por la cadena
DESC SALA	Nombre de la sala; en general tiene relación con la dirección donde se encuentra ubicada
Abast. Local	Fuente por la cual se abastece la sala; esta puede ser a través de despacho directo o centralizado.
PLU Cliente	Código interno del producto asociado por la cadena de supermercado

SAP Cía.	Código SAP del producto asignado por la compañía
DESC SKU	Descripción del producto; Indicando nombre y gramaje
OH	del inglés <i>On Hand</i> ,; indica las unidades totales disponibles en la sala, esto es góndola más bodega.
Order	Unidades solicitadas automática o manualmente a través de los sistemas de reabastecimiento.
SO 4 SEMANAS	del inglés <i>Sell Out</i> , indica la venta total del producto durante las últimas 4 semanas.
Status ITEM	El estado de habilitación del producto en la cadena. Puede ser Activo (A), Inactivo (I) o Descontinuado(D).
Status ITEM-Local	Indica si el producto es o no parte del surtido habilitado en la sala; valor binario, donde toma el valor 1 si lo es y 0 en el caso contrario.

Tabla 3.1: Datos depurados disponibles desde los cubos de información de cada cliente transversal a todas las cuentas.

Fuente: Creación Propia.

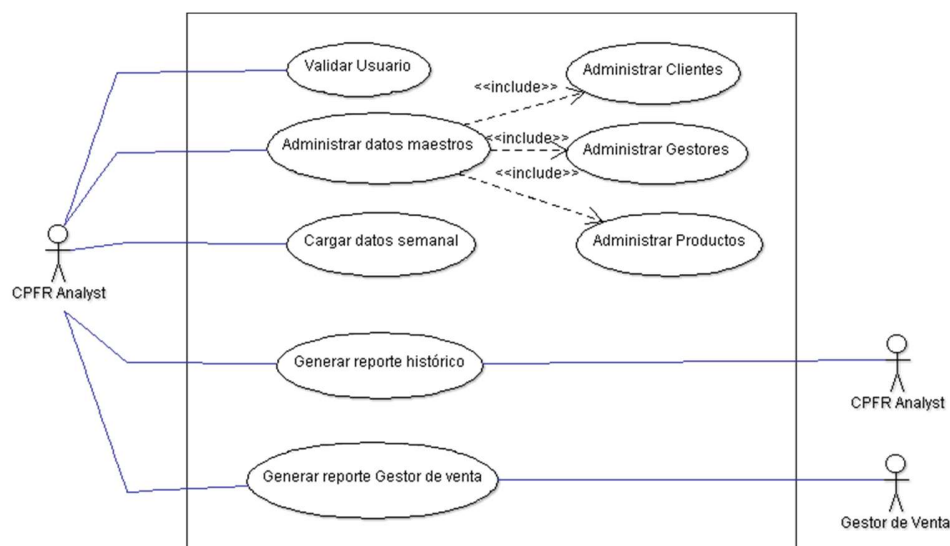


Imagen 3.3: Diagrama de Casos de Uso definidos para el desarrollo.

Fuente: Creación Propia.

Para el cálculo del nuevo indicador, denominado **Instock Ajustado**, los datos van a pasar por dos estados:

1. Cálculo regular de los KPI mencionados, sólo aplicando la lógica matemática.
2. Cálculo nuevo KPI, pasando por dos etapas: Revisión relación entre KPI (una vez hecho el cálculo previo, se hace cruce con las variables *stock* y *venta*) y

Categorización (de cada combinación Ítem-Sala en función de la relación del cálculo de los dos KPI antes mencionados), las que se muestran en la **Imagen 3.4**.

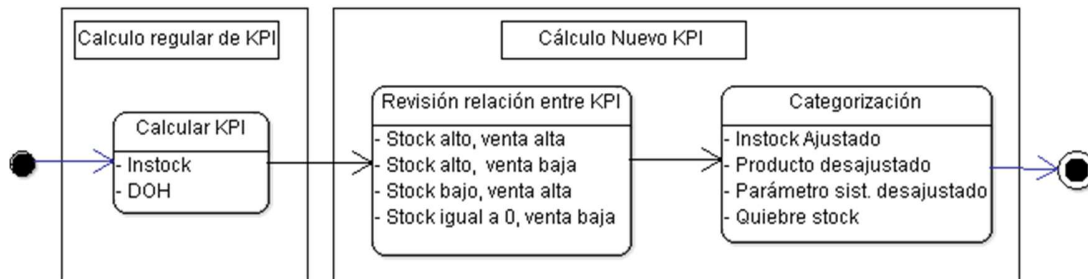


Imagen 3.4: Estados por los que pasa la información para el cálculo del nuevo indicador.

Fuente: Creación propia

Para la lógica que genera el análisis del nuevo KPI, se hizo el diseño de un árbol con las relaciones que hay entre los distintos indicadores y el significado que tenía la no existencia o inconsistencia de esta relación (ver **Imagen 3.5**). Estas relaciones son el resultado de la información levantada durante la etapa de análisis de requerimientos.

Implementación y Pruebas

Para esta etapa lo que se realizó fue cargar los datos de periodos anteriores en la base de datos, específicamente, de dos años. En paralelo, se programó en la herramienta Excel un prototipo de prueba de la lógica que se iba a aplicar a los datos para poder realizar comparaciones/validaciones con datos actuales y con otros periodos aleatorios disponibles.

Go Live

Es la etapa final, constituye la realización y puesta en marcha del sistema. Posterior al término de las pruebas es que se procede al periodo de implantación del proyecto en donde se capacita a los analistas en el uso de los módulos ya mencionados, para que comiencen con la carga y extracción de información ya procesada del sistema.

3.3. Implementación del modelo de apoyo de gestión

Además de entender el proceso de desarrollo del sistema, es importante también comprender como su desarrollo se relaciona con el modelo de apoyo a la gestión, que es lo que este proyecto indirectamente busca. El entorno en el que este sistema se implanta se define a nivel macro como las áreas relevantes de la organización en donde los resultados entregados por el sistema impactan, y a nivel micro del ámbito del propio equipo de trabajo que gestiona dicha información.

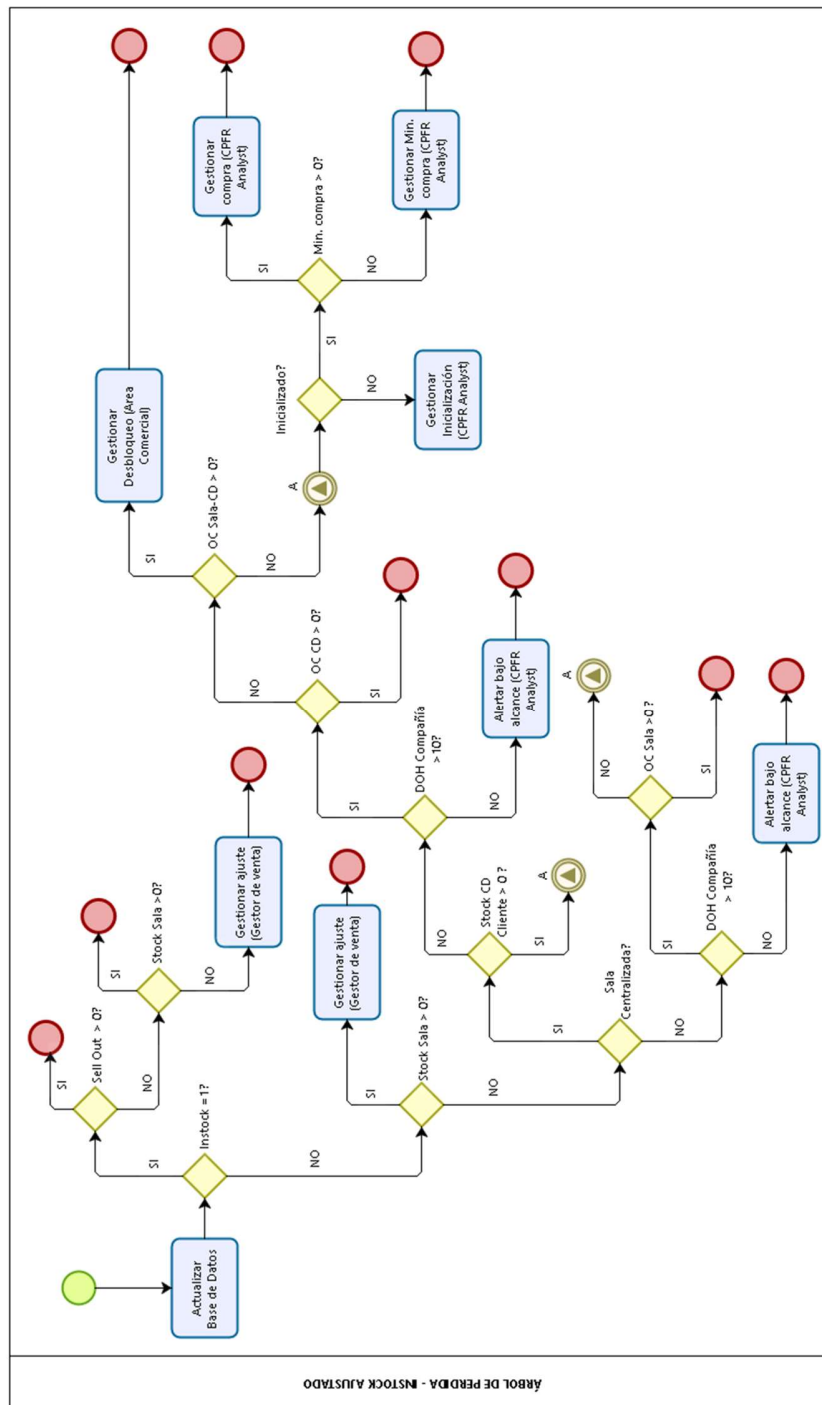


Imagen 3.5: Árbol de decisión para nuevo análisis de Indicador.
Fuente: Creación propia

Al haber implementado una nueva herramienta tecnológica se genera una reorganización en el modelo de trabajo del equipo y de las relaciones que éste tiene con las otras áreas. Con la entrega de información oportuna se debió establecer el “cómo” se va a gestionar. Para entender el modelo en particular se usan los siguientes descriptores:

- **Fundamentación:** el modelo se basa en la cultura ya existente en La Compañía, por ende, no altera la forma en que ésta se organiza, sino en cómo se transmite la información.
- **Estrategia:** los resultados de los análisis realizados por el sistema se consideran libre de compartir a la organización, que da poder de decisión a los miembros de los equipos y fomenta la libre comunicación en todos los niveles organizativos. Para lo anterior es que se motivó a los distintos participantes a que la compartan y usen de forma habitual.
- **Participantes:** es importante tener claro la responsabilidad y campo de acción que el nuevo modelo de trabajo plantea. Para facilitar el entendimiento es que se hace la diferenciación según el actor involucrado de la siguiente forma:
 - Gestores de venta: se encargan de la corrección de las desviaciones que el sistema indica que son por inconsistencia del valor de stock disponible en sala; esto porque bajo el modelo de los supermercados este ajuste lo debe validar y ejecutar la sala de supermercado con la autorización del jefe de ésta.
 - CPFR Analyst de cada cuenta: se encarga de la gestión de los patrones de compra cuando el sistema determina que el valor de reorden de los productos está desajustado. En general, este tipo de caso se presenta para el producto en toda la cadena y se requiere una modificación al parámetro, la cual realiza el analista por el lado de la cadena, previa validación de esto.
 - Equipo comercial: para poder medir la evolución o el alcance de las gestiones realizadas, se determina una revisión general con el equipo comercial y con el de Gestores de venta. El objetivo es revisar el valor del indicador que muestra el producto ajustado en sala; en caso de no presentar mejoras, el equipo comercial debe gestionar directamente con sus pares en las cadenas y solicitar la modificación que los dos actores antes mencionados no lograron.

La **Imagen 3.6** muestra cómo van a interactuar los tres actores antes mencionados. Se forma un trabajo por niveles de profundidad, en donde el nivel siguiente va dando soporte al anterior en caso de no existir aprobación en la solicitud de ajuste o corrección. Cada una de estas áreas continúa trabajando de forma independiente, por ejemplo el CPFR Analyst no está a la espera de la entrega de información por parte de los gestores de venta, pues tiene sus propias incidencias que levantar (que pueden o no tener relación con las del gestor). Sin embargo, la información que recibe de él le da claridad respecto de cuáles debe atacar primero.

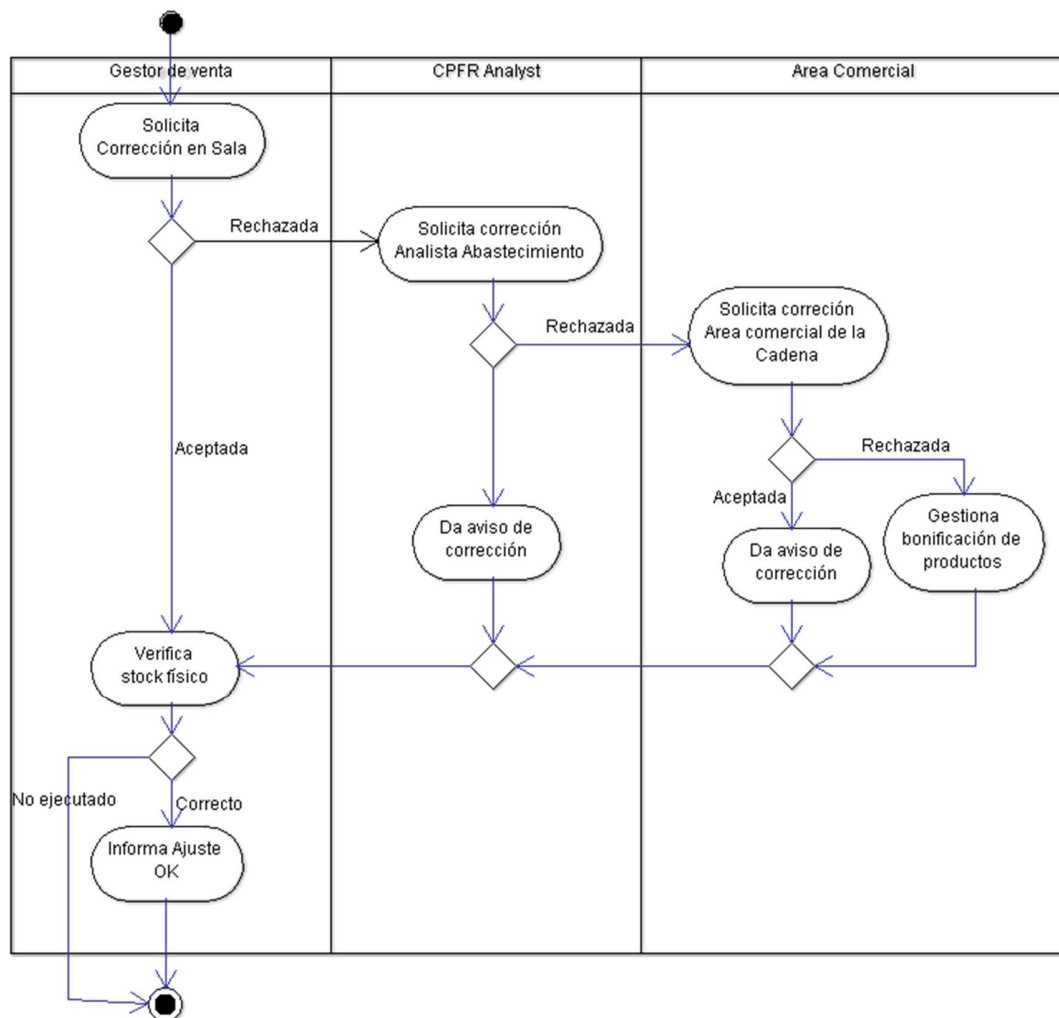


Imagen 3.6: Diagrama de Actividades de la nueva forma de gestión basada en información entregada por sistema.

Fuente: Creación propia

Al comparar la Imagen 3.6 con la Imagen 1.1 y la Imagen 1.2, se aprecia que la nueva forma de gestión de la información reduce el retrabajo, pues ya no hay distintos análisis sobre los mismos datos; produce mejor sincronía en el equipo y permite un mejor flujo de la información entre las tres áreas.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Para validar los resultados de la implementación del sistema, así como también del cambio estratégico que se produce en los equipos, es que en este capítulo se hace un análisis de un antes y un después de la forma de trabajar la información, en los aspectos más importantes de la gestión de ésta. Para finalizar, se mostrarán datos reales que se obtuvieron posterior al desarrollo e implementación del sistema en función del nuevo indicador.

Todo cambio organizacional empieza con una apertura mental de las personas que trabajan en la organización. Sólo las personas cambian sus habilidades, conductas y relaciones, y sólo pueden hacerlo en el trabajo real que desempeñan. Nadie -ni un jefe, ni un subordinado, ni un proveedor, ni un cliente- puede tomar la responsabilidad del cambio de otra persona. Por otro lado, la experiencia enseña que casi siempre la mayoría de las personas cambia más fácil y rápidamente las habilidades, conductas y relaciones de trabajo cuando de ellos dependen los resultados concretos y específicos del desempeño [Zimmermann00].

La importancia de que el sistema cumpliera las expectativas de la gerencia no sólo depende de que el diseño y la implementación estén correctamente ejecutados, sino que también se incorpore a las personas que están involucradas en la gestión de la información proveniente de éste. Como se detalla en el capítulo anterior, donde se detalla la implementación del nuevo modelo de gestión, lo que se buscó producir con el cambio fueron los siguientes tres aspectos:

- **Estabilidad:** el sistema permitió establecer un cambio en la gestión de la información lo que hizo que se integrara a las distintas áreas, asegurando el funcionamiento de estas en función de rutinas y reglas de trabajo.
- **Eficiencia:** sin entrar aún en detalle de los resultados en términos de cifras, el cambio en la forma de trabajo de los equipos en función de la información entregada por el sistema produjo que ellos logaran sus objetivos con menor esfuerzo; desde los analistas que ya no debían realizar el análisis por ellos mismos, al equipo de gestores de venta a quienes les llegaba la información precisa de cuáles eran las gestiones que se esperaba de ellos para realizar mejoras sustanciales en los indicadores.
- **Sinergia:** el desarrollo y la implementación del sistema no solo tuvo impacto en la gestión de los analistas de cada cuenta. Debido al cambio en la forma de trabajo en función de la información obtenida, es que se logró que todas las áreas involucradas en el proceso comenzaran a trabajar como un sistema global e integrado.

La **Tabla 4.1** muestra un comparativo de la forma de trabajo (antes y después de la implementación), en función de tres aspectos importantes que se buscó abarcar para el cumplimiento de los objetivos de este trabajo, los cuales son:

- **Análisis de los datos:** se buscó contar con una herramienta que entregara un análisis transversal del estado actual de los stocks, permitiendo que los mismos indicadores sean usados en distintas cadenas de supermercado, manteniendo la homogeneidad y capacidad de comparación del sistema.
- **Entendimiento de la información:** éste aspecto resulta de absoluta relevancia dado que, a mayor entendimiento y precisión de la información, más rápida y ágil resulta la gestión y la medición del desempeño real de un producto en góndola.
- **Gestión de la información:** dado que es un sistema de apoyo a la gestión de productos en sala, lo que se busca es dar apoyo para la toma de decisiones a distintas áreas.

Aspecto	Antes	Después
Análisis de los datos	Se realiza de forma manual y varía según analista y cuenta. La calidad del análisis depende de cada analista, y semana a semana se pierde noción del trabajo previo.	Se realiza de formas automática y transversal a todas las cuentas, además se tiene registro de resultado de periodos anteriores y de problemas recurrentes.
Entendimiento de la información	Sólo se tiene reportes que muestran mediciones aplicadas al total de los productos y salas para la semana en que fue extraído el dato; por ende los puntos críticos tienden a perderse debido a la gran cantidad de información entregada en estos reportes.	Se tienen dos vistas de la información: una global, para controlar la evolución de los indicadores; y una específica para gestión, en donde se detallan los productos críticos que requieren revisión y gestión. Adicionalmente, sólo se entrega la información necesaria a cada involucrado.
Gestión de la información	Depende directamente de lo que se haya rescatado del reporte semanal; si no se identificó un problema, no se gestiona nada.	A cada involucrado le llega los puntos a revisar y gestionar en función del campo de acción que posea. Adicional a esto, se tiene registro de las recurrencias, por ende hay una medición indirecta del desempeño del involucrado.

Tabla 4.1: Comparación de la forma de trabajo de la información antes y después de la implementación del sistema en los tres aspectos señalados.

Fuente: Creación Propia.

Para finalizar este capítulo, se presentarán cifras entregadas por las mediciones efectuadas desde que se realizó la implementación. El sistema comenzó a ser utilizado en agosto del año 2014; durante todo este año se trabaja a modo de marcha blanca; esto involucró que tanto el sistema como la nueva estructura de trabajo estuvieron siendo monitoreados durante este periodo. Es importante destacar que el mes de agosto representa la foto inicial, momento en el cual no se realizaron gestiones en particular, sólo carga y extracción de información. Ya en enero del año 2015 se consideró aprobados tanto el sistema como la estructura de trabajo.

Las Tablas 4.2 y 4.3 muestran los resultados del KPI *Instock Ajustado* de ambos periodos.

	Objetivo	2014					Resultado	Dif. versus
Cliente		ago-2014	sep-2014	oct-2014	nov-2014	dic-2014	2014	objetivo
Cliente 1	97%	88.6%	94.2%	92.0%	90.1%	90.6%	90.5%	-6.5
Cliente 2	95%	77.4%	74.8%	77.5%	79.2%	82.8%	78.7%	-16.3
Cliente 3	95%	86.3%	88.0%	90.9%	92.5%	93.3%	91.3%	-3.7
Cliente 4	95%	89.8%	93.1%	95.6%	92.8%	92.1%	93.0%	-2.0
Total General		85.1%	86.2%	87.4%	86.8%	89.2%	87.2%	

Tabla 4.2: Resultados obtenidos para el periodo agosto a diciembre de 2014.

Fuente: Datos obtenidos de la medición realizada por el sistema desde su implementación a diciembre de 2014.

	Objetivo	2015							Resultado	Dif. versus
Cliente		ene-2015	feb-2015	mar-2015	abr-2015	may-2015	jun-2015	jul-2015	2015	objetivo
Cliente 1	97%	95.3%	94.8%	91.9%	90.8%	96.1%	96.0%	95.3%	94.0%	-3.0
Cliente 2	95%	83.3%	86.4%	85.7%	87.9%	87.9%	88.3%	90.8%	86.8%	-8.2
Cliente 3	95%	89.2%	94.6%	89.9%	91.6%	94.5%	94.8%	94.8%	92.5%	-2.5
Cliente 4	95%	94.0%	95.7%	92.8%	94.9%	96.0%	97.0%	95.0%	95.1%	0.1
Total General		89.2%	92.2%	89.4%	90.3%	93.2%	93.2%	93.8%	91.4%	

Tabla 4.3: Resultados obtenidos para el periodo enero a julio de 2015.

Fuente: Datos obtenidos de la medición realizada por el sistema para el periodo mencionado.

De éstas, se puede rescatar que de forma simple ya se puede tener un análisis evolutivo y comparativo entre cadenas. Desde el punto de vista de las cifras, destacar que hay un aumento entre el primer y el último mes de medición de 8 puntos porcentuales promedio. Sin embargo, lo más rescatable de este dato es la mejora que presenta el Cliente 2, dado que obtiene una mejora de 13,4 puntos porcentuales desde que se implementa el sistema hasta la última medición.

Adicionalmente, es importante mencionar que, si bien las cuentas aún no alcanzan el objetivo fijado para cada una, se ve que en cada semestre la diferencia se reduce a la mitad, alcanzándose inclusive ya al final del periodo mostrado (julio 2015) por parte del Cliente 4.

CONCLUSIONES

Primero, destacar la importancia del buen manejo de la información que las empresas disponen del comportamiento de sus productos en las salas, pues aumenta la “usabilidad” e integración de la información con el diario funcionar de la empresa; esto quiere decir que estos desarrollos transforman datos en información de calidad.

En este mismo contexto, cabe recalcar la contribución que este tipo de desarrollo entrega a las organizaciones como en la que se ejecutó el proyecto. Contar con una herramienta que procesara de forma automática el volumen de datos disponible generó una ventaja competitiva; adicional a la disminución de carga laboral dentro de la organización misma integrando de mejor forma los procesos entre las distintas áreas, permitió tener otra postura frente su socio estratégico generando una mayor fidelización por parte de éste. Un buen análisis de la información – por supuesto acompañado de una buena gestión – permite cumplir uno de los objetivos principales del trabajo colaborativo entre empresas de Retail y cadenas de supermercado: la maximización de la rentabilidad de los espacios en góndola, garantizando la disponibilidad de productos de forma continua para el consumidor. Esto aumenta el posicionamiento de la marca y como consecuencia, se logra el punto más importante del comercio que es la fidelización del consumidor final y los beneficios económicos que esto conlleva.

Por otro lado, la validez de los objetivos planteados se ve plasmada en los resultados mostrados en el capítulo anterior. Sin embargo, se considera importante destacar que el efecto no lo produce el desarrollo por sí solo, si no que también el orden que le entrega a la organización. Tomar decisiones de administración de recursos en empresas de tamaño grande no resulta fácil y es éste el aspecto que se refuerza con estos desarrollos, dado que permitió a la organización tener una vista evolutiva de la gestión de sus equipos: estableció una medición. Esto se traduce en un mayor control de la operación.

Mencionar también que el uso de indicadores en las empresas no asegura el correcto control de la operación o de la gestión. El trabajo desarrollado durante esta memoria demuestra que las organizaciones deben ser creativas y buscar indicadores que se adapten a sus objetivos y estén en línea con sus procesos. Implementar un nuevo indicador permitió a la organización conocer y relacionar de forma óptima las distintas etapas que conlleva tener sus productos en cantidad y surtido correcto; conjugar de mejor forma sus equipos, minimizando esfuerzos y eliminando el retrabajo.

Desde el punto de vista de uso de metodologías de desarrollo de software por parte de la industria (refiriéndose a la que no se dedica al desarrollo de software), aún no se aprovechan las mejores prácticas en el uso de desarrollo de software. Aunque no existe un proceso de software ideal, en las organizaciones se tiende a incurrir en técnicas más anticuadas o no se aprovechan las mejores prácticas de la industria de software. La

mejora en este aspecto también ayuda a la automatización de procesos cuando está enfocada en la reducción de la diversidad de estos mismos, en donde el siguiente paso natural debiese ser su estandarización, lo que además ayuda a introducir nuevos métodos, técnicas y buenas prácticas.

REFERENCIAS BIBLIOGRÁFICAS

[Cendejas14]. JOSE LUIS CENDEJAS VALDÉZ. **Implementación del modelo integral colaborativo (MDSIC) como fuente de innovación para el desarrollo ágil de software en las empresas de la zona centro –occidente en México**. Sitio revisado última vez en abril de 2017.
<<http://www.eumed.net/tesis-doctorales/2014/jlcv/index.htm>>

[Hollmann15] ROBERTO LUIS HOLLMANN, LUIS FELIPE SCAVARDA, ANTONIO MARCIO THOMÉ. **Collaborative planning, forecasting and replenishment: a literature review**. International Journal of Productivity and Performance Management. Vol 64, No 7, pp 971 – 993. 2015.
<<http://www.emeraldinsight.com/doi/abs/10.1108/IJPPM-03-2014-0039>>

[Reynoso04]. CARLOS BILLY REYNOSO. **Introducción a la Arquitectura de Software**. V1.0 2004.
Sitio revisado última vez en abril de 2017.
<carlosreynoso.com.ar/archivos/arquitectura/Introduccion.PDF>

[Sommerville05]. IAN SOMMERVILLE. **Ingeniería del Software**. Séptima edición 2005.

[UMLs.f] En Wikipedia. **Unified Modeling Language**. Sitio revisado última vez en abril de 2017.
https://en.wikipedia.org/wiki/Unified_Modeling_Language

[Zimmermann00]. ARTHUR ZIMMERMANN. **Gestión del cambio organizacional. Caminos y herramientas**. Edición año 2000.