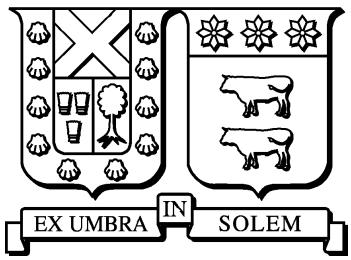
2019-03

REDISEÑO DE UN AMBIENTE OPERACIONAL, MEDIANTE SISTEMAS DE BASES DE DATOS NOSQL Y TÉCNICAS ANALÍTICAS PARA UNA DISTRIBUIDORA DE ALIMENTO DE MASCOTA

GONZÁLEZ IGLESIAS, ESTEBAN IGNACIO

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO – CHILE



REDISEÑO DE UN AMBIENTE OPERACIONAL, MEDIANTE SISTEMAS DE BASES DE DATOS NoSQL Y TÉCNICAS ANALÍTICAS PARA UNA DISTRIBUIDORA DE ALIMENTO DE MASCOTA

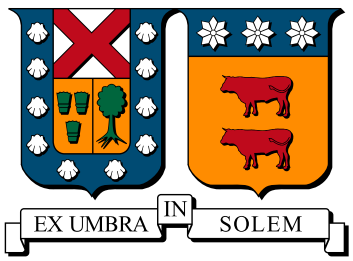
ESTEBAN IGNACIO GONZÁLEZ IGLESIAS

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INFORMÁTICO

PROFESOR GUÍA: JOSÉ LUIS MARTÍ LARA

MARZO 2019

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO – CHILE



REDISEÑO DE UN AMBIENTE OPERACIONAL, MEDIANTE SISTEMAS DE BASES DE DATOS NoSQL Y TÉCNICAS ANALÍTICAS PARA UNA DISTRIBUIDORA DE ALIMENTO DE MASCOTA

ESTEBAN IGNACIO GONZÁLEZ IGLESIAS

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INFORMÁTICO

PROFESOR GUÍA: JOSÉ LUIS MARTÍ LARA
PROFESOR CORREFERENTE: CECILIA REYES COVARRUBIAS

MARZO 2019

A mis padres Luis González y Virginia Iglesias,
Mis abuelos que desearon tanto este momento,
y a Camila Contreras.
Sin su apoyo, esto no hubiera sido posible.

Agradecimientos

Quiero expresar mi gratitud...

A mis profesores, por todo el apoyo y conocimiento que me entregaron durante mis años de universidad.

A José Luis Martí, por sus consejos y su gran labor de jefe de carrera que me ayudó a no perder el foco en un momento crítico de mi paso por la universidad.

A mis padres, Luis González y Virginia Iglesias, por siempre confiar en mi y también alentarme cuando más lo necesitaba.

A mis abuelos que, en algún lugar del cielo, estarán felices por saber que el ciclo ya está cerrando.

A Camila Contreras, por su amor incondicional que logró entregar calma y fortaleza cuando se requería.

A Jorge Villagrán, por las arduas horas utilizadas en su hogar para afinar y finalizar este documento.

A mi gran amigo Oscar Rencoret, por nunca dudar en brindarme ayuda cuando la necesitaba y que, gracias a el, logré superar obstáculos en mi carrera universitaria.

Resumen

En este trabajo, se entregan los resultados de un estudio sobre un ambiente operacional de una distribuidora de alimento de mascotas que presenta falencias en su arquitectura tecnológica. Se realiza un análisis comparativo de bases de datos y de tecnologías basadas en inteligencia de negocios, con el fin de entregar una propuesta de rediseño para este ambiente operacional usando una base de datos *NoSQL* para agilizar la toma de datos, y el uso de un software de inteligencia de negocios para generar reportes y lograr una toma de decisiones más efectiva y eficiente. La metodología utilizada consiste principalmente en replicar el ambiente operacional actual, pero con una base de datos *NoSQL* para evidenciar posibles mejoras en tiempos de respuesta, tiempos de generación de reportes, disminución de costos de licencia y mantención, entre otros.

Palabras Clave – Bases de Datos, Inteligencia de Negocios, *NoSQL*, Planeamiento de Recursos Empresariales.

Abstract

In this work, are delivered the results of an study about an operational environment of a pet food distributor that has flaws in its technological architecture. A comparative analysis of databases and technologies based on business intelligence is presented, in order to deliver a redesign proposal for this operational environment using a database *NoSQL* to streamline the data collection, and the use of business intelligence software to generate reports and achieve more effective and efficient decision making. The methodology to be used is based, mainly, on replicating the current environment used in the company and using a database *NoSQL* to show possible improvements in response times, generation of reports, decrease in license costs and maintenance, among others.

Keywords - Databases, Business Intelligence, *NoSQL*, Enterprise Resource Planning.

Índice de Contenidos

Índice de cuadros

Índice de figuras

Glosario

ACID (Atomicity, Consistency, Isolation and Durability; Atomicidad, Consistencia, Aislamiento y Durabilidad): características de los parámetros que permiten clasificar y garantizar las transacciones de los sistemas de gestión de bases de datos.

ASCII (American Standard Code for Information Interchange; Código Estándar Estadounidense para el Intercambio de Información): código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno. Utiliza 7 bits para representación.

BI (Business Intelligence; Inteligencia de Negocios): conjunto de estrategias, aplicaciones, datos, productos y tecnologías, los cuales están enfocados a la administración y creación de conocimiento sobre el medio, a través del análisis de los datos existentes en una organización o empresa.

Big Data (Gran volumen de datos): un conjunto de datos o combinaciones de estos cuyo tamaño (volumen), complejidad (variabilidad) y velocidad de crecimiento (velocidad) dificultan su captura, gestión, procesamiento o análisis mediante tecnologías y herramientas convencionales.

BSON (Binary JavaScript Objetc Notation; Notación de objeto binario de JavaScript): formato de intercambio de datos usado principalmente para su almacenamiento y transferencia en la base de datos *MongoDB*. Es una representación binaria de estructuras de datos y mapas.

CRP (Customer Relationship Management; Gestión de Relación con Clientes): sistema informático de apoyo a la gestión de las relaciones con los clientes, a la venta y al marketing. Utiliza la historia de los clientes para mejorar las relaciones comerciales, fidelizar clientes e impulsar el crecimiento de las ventas.

Data Mart: base de datos departamental, especializada en el almacenamiento de los datos de un área de negocio específica. Se caracteriza por disponer la estructura óptima de datos para analizar la información al detalle desde todas las perspectivas que afecten a los procesos de dicho departamento.

Data Warehouse (Almacén de datos): colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza. La diferencia con el *data mart* es que no se centra en un tema o área de negocio.

ERP (Enterprise Resource Planning; Planeamiento de recursos empresariales): sistema de información gerencial que integra y maneja muchos de los negocios asociados con las operaciones. Típicamente maneja la producción, logística, distribución, inventario, envíos, facturas y contabilidad de la compañía.

ETL (Extract, Transform and Load; Extraer, Transformar y Cargar): proceso que permite a las organizaciones mover datos desde múltiples fuentes, formatearlos, limpiarlos, y cargarlos en otra base de datos, *data mart* o *data warehouse* para su posterior análisis.

Excel: programa informático desarrollado y distribuido por Microsoft. Se trata de un software que permite realizar tareas contables y financieras gracias a sus funciones, desarrolladas específicamente para ayudar a crear y trabajar con hojas de cálculo.

Extranet: red privada que utiliza protocolos de Internet, protocolos de comunicación y probablemente infraestructura pública de comunicación para compartir de forma segura parte de la información u operación propia de una organización con proveedores, compradores, socios, clientes o cualquier otro negocio u organización

JSON (JavaScript Object Notation; Notación de objeto de JavaScript): formato de texto ligero para el intercambio de datos.

MongoDB: sistema de base de datos *NoSQL* orientado a documentos, desarrollado bajo el concepto de código abierto.

NoSQL (Not Only SQL; No solo SQL): amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico relacional. Permiten el uso de lenguaje *SQL* como también el propio.

ODBC (Open DataBase Connectivity; Conectividad abierta de base de datos): acceso estándar a las bases de datos. Hace posible acceder a cualquier dato desde cualquier aplicación, sin importar qué SGBD almacene los datos.

Open Source (fuente abierta): expresión que pertenece al ámbito de la informática. Programa que permite el acceso a su código de programación, lo que facilita modificaciones por parte de otros programadores. Generalmente, es gratis.

SGBD (Sistema de Gestión de Base de Datos): software dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan

Streaming (Transmisión o Retransmisión): distribución digital de contenido multimedia a través de una red de computadoras, de manera que el usuario utiliza el producto a la vez que se descarga. La palabra retransmisión se refiere a una corriente continua que fluye sin interrupción, y habitualmente a la difusión de audio o vídeo.

SQL (Structured Query Language; Lenguaje de consulta estructurado): lenguaje específico diseñado para administrar y recuperar información de SGBD relacionales.

UTF8 (8-bit Unicode Transformation Format; Formato de transformación de Unicode de 8 bits: formato de codificación de caracteres Unicode e ISO 10646 que utiliza símbolos de longitud variable. Es capaz de representar comillas, tildes de todo tipo y cualquier símbolo ASCII.

XML (eXtensible Markup Language; Lenguaje de Marcado Extensible): metalenguaje que permite definir lenguajes de marcas. Utilizado para almacenar datos en forma legible dado que da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.

Introducción

El manejo de información, para toda empresa, es fundamental para lograr eficiencia y eficacia en sus procesos. Es por esto, que es importante tener sistemas de bases de datos capaces de cumplir, en tiempo real, la entrega de información que se pueda requerir en ciertos momentos críticos de análisis. La importancia que adquieren los datos crece día a día cuando con estos es posible minimizar errores, costos y mejorar los tiempos de respuesta con los clientes. Cuando el manejo de la información se vuelve dificultoso, generalmente las empresas adquieren sistemas de información o un *software* capaz de dar valor a estos datos mediante el logro de objetivos estratégicos o del negocio o, simplemente, mejorar la planificación de los recursos que se disponen.

Dado que los datos en las empresas suelen acumularse a medida que el tiempo pasa, es necesario ir revisando los sistemas en los cuales estos se van conteniendo para así evitar los problemas asociados a tener grandes volúmenes de datos como son: lentitud en exportar datos, altos costos en mantención y respaldo de la información. Con lo anterior, buscar una base de datos adecuada comienza a ser un reto.

En este trabajo de investigación se hace un análisis comparativo de dos bases de datos presentes en el mercado, comparando tiempos de respuesta, arquitectura, almacenaje y costos, para así recomendar un ambiente operacional óptimo a la empresa en estudio. Para presentar el desarrollo de este trabajo, se tiene la siguiente estructura:

- Capítulo 1: se entrega una definición del problema presentando aspectos relevantes del trabajo actual que se tiene en la empresa en estudio y, por otra parte, se presentan el objetivo principal y los objetivos específicos asociados a este trabajo de investigación.

- Capítulo 2: se hace una recopilación y análisis exhaustivo de las tecnologías utilizadas en el contexto de este trabajo de investigación. Se entrega referencia a bases de datos *SQL*, bases de datos *NoSQL* e inteligencia de negocios.
- Capítulo 3: se entrega una propuesta de solución a los problemas mencionados en la definición del problema. Además, se indicará la metodología de comparación en las bases de datos a utilizar.
- Capítulo 4: se entrega una validación a la propuesta presentada mostrando resultados afines a la investigación, y se concluye con una propuesta de ambiente operacional que optimiza a la empresa en estudio.

Finalmente, se entregan las conclusiones de la investigación, incluyendo posibles trabajos a futuro relacionados con este tema.

Capítulo 1

Definición del Problema

En este capítulo se aborda, de forma general, el contexto del problema en el cual se basa la investigación realizada. En primera instancia se muestra cómo las empresas, al verse llenas de información, contratan *software* que les brindan apoyo para el manejo de datos o planificación de los recursos. Luego, se presenta como es el ambiente operacional de la empresa en estudio y los problemas asociados. Finalmente, se abordan los objetivos principales, específicos y el alcance que este trabajo tiene asociado.

1.1. Manejo de datos y contexto empresarial

Las bases de datos, hoy en día, son utilizadas por organizaciones de todo tipo dado que el acceso a estos servicios no tiene límites. El gran enfoque que estas organizaciones están teniendo para los datos que acumulan, hace referencia a sus clientes y la interacción que tienen con ellos. Para esto, existen aplicaciones llamadas *Customer Relationship Management* (CRM), que se definen como una estrategia de negocios dirigida o enfocada a entender, anticipar y responder a las necesidades de los clientes, actuales y potenciales, de una empresa para hacer que el valor de la relación entre ambas partes crezca. También, existen aplicaciones denominadas *Enterprise resource Planning* (ERP), que se hacen cargo de distintas operaciones internas de una empresa, desde producción a distribución e, incluso, recursos humanos. Los ERP también se alimentan de una base de datos que, generalmente, es tan grande como la de los CRM. Esta base de datos es del tipo incrustada

principalmente.

Contar con aplicaciones como los CRM y ERP suponen una gran inversión para las empresas. Según una encuesta de Panorama Consulting de 2013[1], un 40 % de las empresas que adquieren un ERP notan un aumento en la productividad. Además, ninguna empresa es ajena a adquirir este tipo de servicios. Para el caso de una distribuidora de alimento de mascota, rubro de la empresa en estudio de este trabajo, se debe estar en constante comunicación con los clientes, dado que proveer y mantener un stock de productos es vital para las relaciones y ganancias de ambas partes; por lo que la adquisición de un CRM o ERP es una acción más que necesaria. La cantidad de empresas que son del mismo tipo de la distribuidora, según datos del Servicio de Impuestos Internos de Chile [2], corresponden a 351.793 empresas, que equivalen al 34.4 % del total de las empresas que, actualmente, se encuentran activas [3].

Por otra parte, es necesario recalcar que estas empresas buscan eficiencia en la toma de decisiones ya que constantemente compiten por los clientes. Además, el número de empresas que se registran anualmente en Chile asciende a 109.974, según datos informados por el Banco Mundial[4] y que, de manera muy probable, requieren organizarse con alguna plataforma como los *CRM* o *ERP* o contar con un área de TI que deba ser la responsable de gestionar aplicaciones para la toma de decisiones o bienestar de sus clientes.

1.2. Ambiente actual

La empresa en estudio cuenta con un ERP llamado *Random* que opera bajo una base de datos *SQL* incrustada. El sistema tiene un servidor que registra, mediante consultas *SQL*, los pedidos de los clientes que son ingresados a través de una aplicación móvil. También, es posible ingresar estos pedidos desde la aplicación de escritorio. Todo el proceso de ventas es dirigido y mantenido por el sistema ERP mencionado.

El proceso de ventas actual sigue una doble metodología. Los nombres que la empresa

[1]Encuesta disponible en: https://bit.ly/2SckyVY

[2]Estadísticas de empresas por rubro económico, disponible en: https://goo.gl/CLq2WB

[3]Informe de resultados: Empresas en Chile, Cuarta Encuesta Longitudinal de Empresas, disponible en: https://goo.gl/iq2nWs

[4]Banco Mundial, nuevas empresas registradas, disponible en: https://bit.ly/2GF6AL8

en estudio asignó para estas dos modalidades son: Venta directa y Auto-Venta. El método Auto-Venta hace referencia a un mecanismo en el cual se tiene un camión cargado de productos que se ofrecen a varios clientes que quedan rezagados de las rutas que tienen asignadas los vendedores o también, cuando no existen clientes rezagados, el camión se dirige a comunas de las afueras de Santiago (Melipilla, Buin, etc) a realizar ventas, principalmente, a clientes que tienen una modalidad de venta "mayorista". Cabe destacar que Auto-Venta no sigue un patrón concreto de visitas y depende mucho del informe de salida que realizan los vendedores.

La venta directa es un mecanismo de venta tradicional que existe en muchas empresas de Chile. El modelo del proceso de venta, se puede observar en la figura 1.1.

En este formato, existen vendedores que deben seguir una ruta ya creada por la empresa para visitar clientes. En la visita, si logran la venta, el pedido del cliente debe ser ingresado a una aplicación móvil para así guardar los datos y facturar el pedido al final del día. Si dentro de la ruta, hay algunos clientes que no estén disponibles o tienen su negocio cerrado, los vendedores deben volver a visitarlos. Ahora bien, si el vendedor no generó una nueva venta con el cliente dado que este aún tiene mucho *stock* de productos, el vendedor debe documentar el *stock* restante de los productos que el cliente adquiere. Finalmente, al terminar la jornada laboral, el vendedor debe realizar un informe de salida donde se indica si hubo nuevos clientes (ya que estos deben ser agregados a la ruta del día de elaboración del informe) y entregar toda la documentación requerida para ingresar al sistema a este nuevo cliente (razón social, nombre, dirección, RUT, comuna, tipo y si será un cliente con descuento). Además, en este informe, se debe indicar qué clientes no fueron visitados.

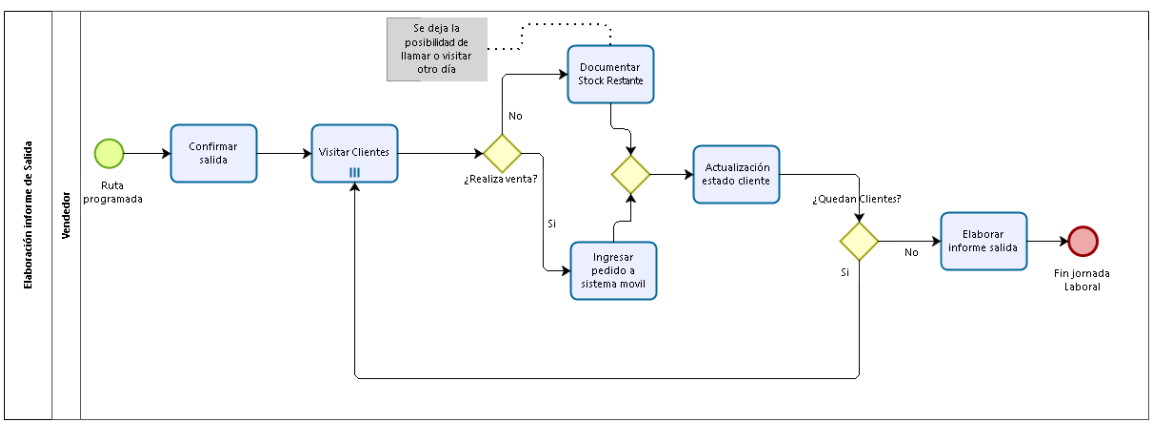


Figura 1.1: Modelo proceso de venta directa de la distribuidora en estudio

Fuente: Elaboración Propia.

Posterior a las ventas, comienza el proceso de facturación y creación de reportes. Diariamente, se generan facturas, como se ejemplifica en la figura 1.2, acorde a la cantidad de clientes que realizaron compras. Estas facturas son generadas con uno de los tantos módulos que tiene la herramienta *Random*, módulo que esté en conexión con el Servicio de Impuestos Internos de Chile y sigue todas las normas legales.

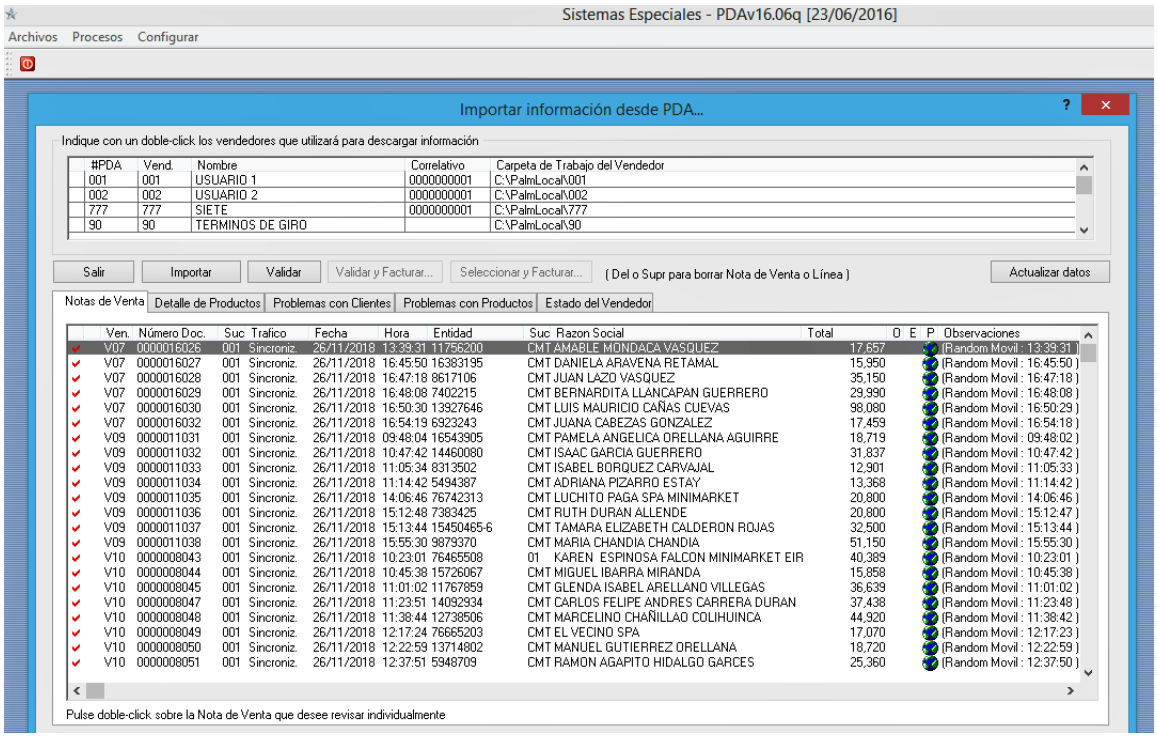
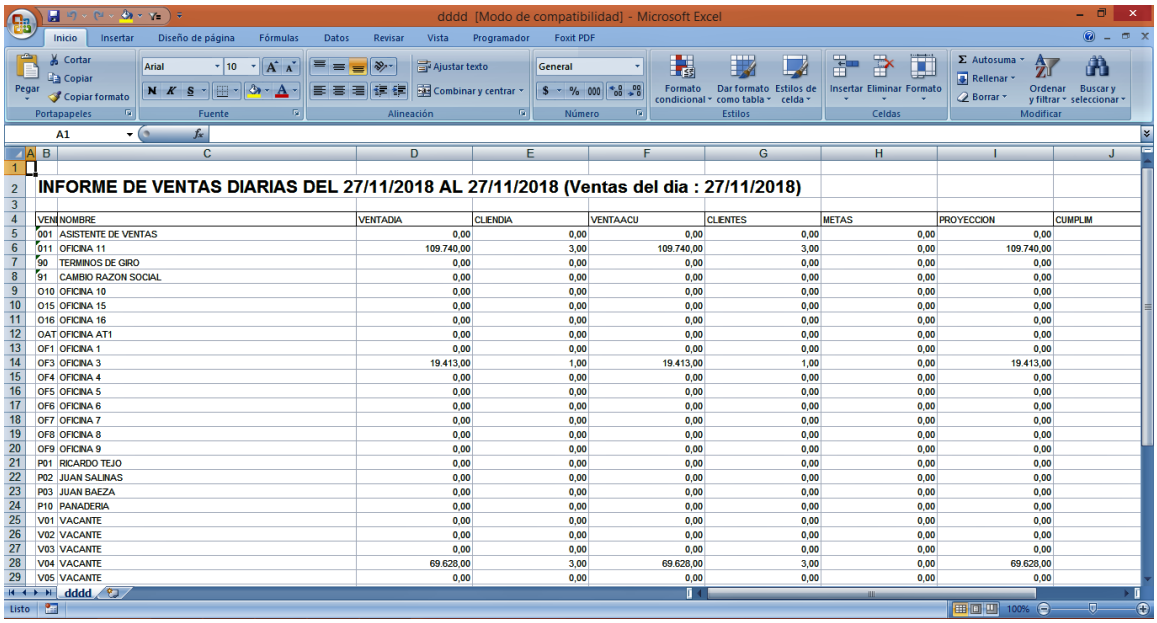


Figura 1.2: Módulo de facturación de *Random*.
Fuente: ERP *Random* - Sistemas especiales.

Finalmente, comienza el proceso de creación de los reportes. Este proceso inicia con el desarrollo de la cuadratura de las entregas realizadas de las ventas del día anterior, proceso que actualmente se realiza vía *Excel*. Luego, comienza el proceso de creación del informe de ventas diarias donde, por ruta, se obtienen la cantidad de productos vendidos y el monto total de las ventas. Este módulo de creación de informes es especial y no es accesible desde dentro del ERP *Random*, ya que se debe instalar un acceso especial. Además, se genera un consolidado de los pedidos que sirve de guía para la distribución de las ventas que se realizará el siguiente día hábil. El informe de consolidación se hace mediante *Excel* tras rescatar las facturas ya procesadas. Otros tipos de reportes son elaborados solo si se desea visualizar algún tipo de información más detallada. Estos reportes son creados, generalmente, para saber a quien entregar bonos por mayores ventas u otro tipo de regalías. La

generación de estos informes es mediante *Excel* y la edición se realiza manualmente. Un ejemplo de informe *Excel* a editar puede ser observado en la figura 1.3, en esto, se puede apreciar una gran cantidad de columnas que no tienen datos.



INFORME DE VENTAS DIARIAS DEL 27/11/2018 AL 27/11/2018 (Ventas del dia : 27/11/2018)

VEN	NOMBRE	VENTADIA	CLIENDIA	VENTAACU	CLIENTES	METAS	PROYECCION	CUMPLIM
001	ASISTENTE DE VENTAS	0,00	0,00	0,00	0,00	0,00	0,00	
011	OFICINA 11	109.740,00	3,00	109.740,00	3,00	0,00	109.740,00	
90	TERMINOS DE GIRO	0,00	0,00	0,00	0,00	0,00	0,00	
91	CAMBIO RAZON SOCIAL	0,00	0,00	0,00	0,00	0,00	0,00	
O10	OFICINA 10	0,00	0,00	0,00	0,00	0,00	0,00	
O15	OFICINA 15	0,00	0,00	0,00	0,00	0,00	0,00	
O16	OFICINA 16	0,00	0,00	0,00	0,00	0,00	0,00	
OAT	OFICINA AT1	0,00	0,00	0,00	0,00	0,00	0,00	
OF1	OFICINA 1	0,00	0,00	0,00	0,00	0,00	0,00	
OF3	OFICINA 3	19.413,00	1,00	19.413,00	1,00	0,00	19.413,00	
OF4	OFICINA 4	0,00	0,00	0,00	0,00	0,00	0,00	
OF5	OFICINA 5	0,00	0,00	0,00	0,00	0,00	0,00	
OF6	OFICINA 6	0,00	0,00	0,00	0,00	0,00	0,00	
OF7	OFICINA 7	0,00	0,00	0,00	0,00	0,00	0,00	
OF8	OFICINA 8	0,00	0,00	0,00	0,00	0,00	0,00	
OF9	OFICINA 9	0,00	0,00	0,00	0,00	0,00	0,00	
P01	RICARDO TEJO	0,00	0,00	0,00	0,00	0,00	0,00	
P02	JUAN SALINAS	0,00	0,00	0,00	0,00	0,00	0,00	
P03	JUAN BAEZA	0,00	0,00	0,00	0,00	0,00	0,00	
P10	PANADERIA	0,00	0,00	0,00	0,00	0,00	0,00	
V01	VACANTE	0,00	0,00	0,00	0,00	0,00	0,00	
V02	VACANTE	0,00	0,00	0,00	0,00	0,00	0,00	
V03	VACANTE	0,00	0,00	0,00	0,00	0,00	0,00	
V04	VACANTE	69.628,00	3,00	69.628,00	3,00	0,00	69.628,00	
V05	VACANTE	0,00	0,00	0,00	0,00	0,00	0,00	

Figura 1.3: Ejemplo de informe en *Excel* entregado por el ERP *Random*.
Fuente: Elaboración Propia.

1.3. Problema

El ambiente operacional, descrito anteriormente, está presentando ciertas falencias basadas, principalmente, en el uso de los datos. Como la información que maneja la distribuidora en estudio crece, dado que se generan ventas todos los días, la base de datos del sistema ERP *Random* crece en forma horizontal, dando cabida al problema de cuello de botella [Boncz *et al.*, 1999]. Con esto, toda solicitud o ingreso de datos que se hace a la aplicación ERP comienza a ser muy lenta. Lo anterior, afecta directamente a la generación de informes como también a la toma de decisiones que se realiza, ya que la información nunca está disponible en tiempo real y siempre se debe tener en cuenta que, para todo proceso operativo que necesite alguna información desde el ERP, se debe esperar un espacio considerable de tiempo. Como la aplicación ERP *Random* no es parametrizable por el usuario, es difícil tener información eficiente en tiempo real de los informes (en *Excel*) generados. Estos informes son de gran calibre y, por lo que se observa en la figura 1.3,

es necesario adicionar un tiempo a la limpieza de los datos nulos. La generación de este archivo bordea entre los 20 a 30 minutos y, su edición, los 15 minutos.

Por otra parte, para la empresa en estudio, la aplicación ERP *Random* supone un elevado costo de mantención. Actualmente, el servicio fue contratado con todos los módulos existentes para la versión del 15 de septiembre del 2014. La adquisición del producto ya es elevada, alcanzando los 4.980.000 pesos chilenos. Dentro de esta adquisición, se contemplan los módulos de: reportería, facturación, informes, bodegaje y captura de información del sistema móvil. Otros costos asociados a la aplicación ERP son:

- Licenciamiento mensual: 760.000 pesos chilenos.
- Servicio de Soporte (vía teléfono o Escritorio remoto): incluido en el licenciamiento mensual.
- Mantención técnico externo: 300.000 pesos chilenos.
- Membresía anual y servidor: 1.734.000 pesos chilenos

El licenciamiento mensual, corresponde al número de equipos que tienen habilitado el permiso para utilizar la aplicación. Con este licenciamiento, se opta al servicio de soporte que solo opera vía telefónica o vía internet haciendo uso de aplicaciones que permitan utilizar el computador de forma remota. En el caso de la empresa en estudio, la cantidad de licencias utilizadas es dos y tiene el servicio de actualización del sistema activo. Dado que se opera con una base de datos incrustada, toda actualización es de cuidado, ya que cualquier edición realizada a la base de datos, repercute directamente en los módulos presentados, de los cuales, generalmente, el módulo de facturación presenta fallas. Lo anterior, se debe a que al agregar o quitar parámetros de la base de datos, se pierde el formato de facturación impidiendo la impresión y envío al Servicio de Impuestos Internos.

La mantención técnica externa, corresponde a una persona que, dos veces al mes, realiza limpieza del servidor que utiliza el ERP *Random*. Además, respalda la información que se va teniendo ante cualquier eventualidad. También, en estas visitas, realiza mantención a los dos computadores que tienen el licenciamiento mensual.

La membresía anual corresponde al valor proporcional del monto de adquisición del servicio; también, abarca los gastos asociados a replicas del servidor. Todo esto con el

fin de tener respaldos útiles por si las actualizaciones del sistema producen algún error. Este valor aborda también la mantención del servidor y un respaldo de todas las versiones que ha tenido la empresa. Finalmente, se evidencian los altos costos asociados a tener una aplicación ERP, gran parte del problema radica en que no todos los módulos ofrecidos son utilizados pero, de igual forma, deben ser pagados.

1.4. Objetivos

1.4.1. Objetivo Principal

Mejorar el ambiente operacional de la distribuidora, mediante el diseño de una arquitectura de sistema, de manera tal que un gerente operacional pueda deliberar, apropiadamente, qué sistema de bases de datos utilizar para aumentar la eficiencia de su productividad.

1.4.2. Objetivos específicos

- Diseñar un ambiente operacional utilizando una nueva tecnología en base de datos, para disminuir los tiempos de respuesta y costos asociados.
- Evaluar el nuevo diseño propuesto respecto al ambiente operacional actual enfocándose en tiempos de respuesta, almacenamiento de datos y costos; para así verificar y recomendar el nuevo ambiente diseñado.
- Proponer la utilización de un software de inteligencia de negocios, para aumentar la eficiencia en la toma de decisiones y aminorar los costos asociados a estas decisiones.

1.5. Alcances

- Los datos a utilizar en este trabajo de investigación corresponden a ventas realizadas a clientes entre los años 2014 y 2017. Estos clientes tienen la característica de haber realizado compras, al menos, una vez por mes, que para la empresa corresponden a clientes “activos”. Además, se utilizan todos los productos de alimentos de mascotas existentes, incluyendo la publicidad.

- El nuevo ambiente se diseña en una base de datos *NoSQL* columnar y *open source*, debido a que se busca reducir costos y tiempo de ejecución. La estructura columnar de la base de datos es propicia para la creación de índices lo que beneficia a la reducción del tiempo.

- No se consideran otras bases de datos *NoSQL* en la comparación, debido a que utilizar bases de datos de grafos o documentales no agregan valor alguno en el proceso de generación de información relevante para la toma de decisiones. Además, la dificultad de uso asociada a estos tipos de bases de datos, haría inviable la propuesta de diseño.

- En la capa de análisis y presentación de la propuesta de arquitectura, no se realiza un enfoque al análisis de datos en sí; más bien, el camino a seguir en este trabajo es centrarse en la capa de presentación de la arquitectura propuesta con el fin de sentar bases para realizar un análisis de los datos en un futuro.

Capítulo 2

Estado del Arte

En este capítulo se realiza una revisión de los principales conceptos y fundamentos que se utilizan al trabajar con sistemas de almacenamiento de datos e inteligencia de negocios. Con el fin de contextualizar, se entrega una recopilación y análisis exhaustivo de las tecnologías utilizadas en esta investigación.

2.1. Bases de datos

Una base de datos es, de manera simple, una gran colección de datos organizados y relacionados entre si que pueden entregar información. Esta colección está usualmente organizada para modelar ciertos aspectos de la realidad, como por ejemplo un torneo de fútbol, distribuciones de las camillas en un hospital o simplemente un calendario de actividades. Estos datos se entregan de tal forma que puedan asistir a diferentes programas o procesos que necesiten esta información. Para esto, se utilizan los sistemas de gestión de bases de datos (SGBD) que son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta, tal como se indica en [Haigh, 2011]. Algunos de los SGBD más conocidos son MySQL, PostgreSQL y Oracle. Se destaca que una base de datos no es, usualmente, transferible entre diferentes SGBD dado que no siempre se utilizan los mismos esquemas de acceso a los datos; pero, los tres SGDB señalados como los más conocidos, pueden

operar entre ellos a través de lenguajes estándares como SQL u ODBC para permitir que una aplicación pueda trabajar con más de una base de datos a la vez.

El uso de SGDB presenta las siguientes ventajas:

- Control sobre la redundancia de datos: la redundancia de datos proporciona tolerancia a fallos, lo que permite que un sistema continúe la operación total o parcial, si una parte del sistema falla debido a la pérdida o corrupción de datos. Como no se almacenan varias copias de los datos en un mismo lugar, el SGBD controla la redundancia que se produce de estos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones existentes entre los datos.
- Consistencia de datos: dado el punto anterior, se reduce en gran medida el riesgo de inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez y estar disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantengan consistentes.
- Compartición de datos: la base de datos puede ser compartida por todos los usuarios dentro del dominio del SGBD.
- Accesibilidad a los datos: como un SGBD proporciona un lenguaje de consultas, es muy simple obtener información de ellos.
- Acceso concurrente: los SGBD gestionan los accesos múltiples de usuarios, por lo que pérdidas de información por actualización conjunta de datos no ocurre. Generalmente, se limita la cantidad de usuarios concurrentes.

Por otra parte, el uso de SGBD presenta las siguientes desventajas:

- Vulnerabilidad a fallos: dada la centralización de los datos, un SGDB es más vulnerable a fallos que puedan ocurrir. En necesario tener copias de seguridad como respaldo.
- Complejidad: los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad

- Altos costos: tanto el SGBD como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, puede que sea necesario adquirir una máquina más grande o una que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara.

2.1.1. Modelo de datos

Un modelo de datos es un tipo de modelo que determina la estructura lógica de una base de datos y de manera fundamental determina el modo de almacenar, organizar y manipular los datos.

Entre los modelos lógicos comunes para bases de datos, expuestos en [Conrick, 2006], se tienen:

- Modelo jerárquico: los datos están organizados en una estructura arbórea, lo que implica que cada registro sólo tiene un padre.
- Modelo en red: expande la estructura del modelo jerárquico, permitiendo relaciones "muchos a muchos" en una estructura tipo árbol que permite múltiples padres. Antes de la llegada del modelo relacional, el modelo en red era el más popular para las bases de datos. Definido con el enfoque CODASYL en [Michaels *et al.*, 1976].
- Modelo relacional: modelo matemático definido de [Codd, 1970]. Existen tres términos usados con profusión en el modelo relacional de bases de datos: relaciones, atributos y dominios. Una relación equivale a una tabla con filas y columnas. Las columnas de una relación se llaman con rigor atributos, y el dominio es el conjunto de valores que cada atributo puede tomar. La estructura básica de datos del modelo relacional es la relación (tabla), donde la información acerca de una determinada entidad se almacena en tuplas (filas), cada una con un conjunto de atributos (columnas).
- Modelo documental: encapsulamiento o codificación de la información siguiendo algún formato estándar. Los más conocidos son XML, JSON y BSON.
- Modelo en estrella: modelo que tiene una tabla de hechos (tabla central) que contiene los datos para el análisis, rodeada de las tablas de dimensiones. Los hechos contienen

datos medibles, cuantitativos, relacionados a la transacción del negocio, y las dimensiones son atributos que describen los datos indicados en los hechos. Esta forma de ordenar las tablas con una gran tabla al centro y varias rodeándola asemejan a una estrella, lo que concede el nombre a este tipo de construcciones.

Basado en los modelos anteriores, se presenta una descripción y conceptos de algunas clasificaciones de SGBD.

2.1.2. Bases de datos relacionales

Tipo de base de datos que cumple con el modelo relacional (el modelo más utilizado actualmente para implementar bases de datos) y sigue su misma especificación. Postulada por E.Codd [Codd, 1970], no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Básicamente, una relación representa un conjunto de entidades con las mismas propiedades. Cada relación se compone de una serie de filas o registros (tuplas), cuyos valores dependen de ciertos atributos (columnas). Asimismo, las bases de datos relacionales ofrecen varios niveles de refinamiento de sus tablas llamadas formas normales.

La tabla, concepto clásico de la organización de la información, es el formato que utiliza el modelo relacional para explicar de un modo visual la ordenación de los valores de una tupla en función de los atributos definidos en la relación. Una base de datos relacional no es otra cosa, entonces, que un conjunto de tablas interrelacionadas. Los valores que contiene cada tupla vienen determinados por los atributos definidos en el esquema relacional. Una representación de la tabla puede verse en la figura 2.1.

Con los datos ya estructurados, y como se mencionó anteriormente, el SGBD gestiona los accesos para que los usuarios interactúen con la base de datos mediante un lenguaje. Todo gestor de bases de datos relacionales soporta al menos un lenguaje formal que permite ejecutar las siguientes operaciones de definición:

- Estructura de datos: en la definición de los datos se guarda una descripción con metadatos de la estructura. Cuando un usuario crea una tabla nueva, se almacena su correspondiente esquema.

- Derechos: todos los lenguajes de bases de datos proporcionan una sintaxis que permite otorgar o retirar permisos.
- Condiciones de integridad: son los requisitos de estado que se exigen a un banco de datos. Si se definen condiciones para la integridad de datos, la base de datos garantiza que se cumplan en todo momento para estar en un estado de consistencia.
- Transacciones: cuando se lleva a una base de datos de un estado consistente a otro diferente. Las transacciones contienen una serie de instrucciones que deben ejecutarse siempre de forma íntegra.
- Vistas: representaciones virtuales de un subconjunto de los datos de una o más tablas.

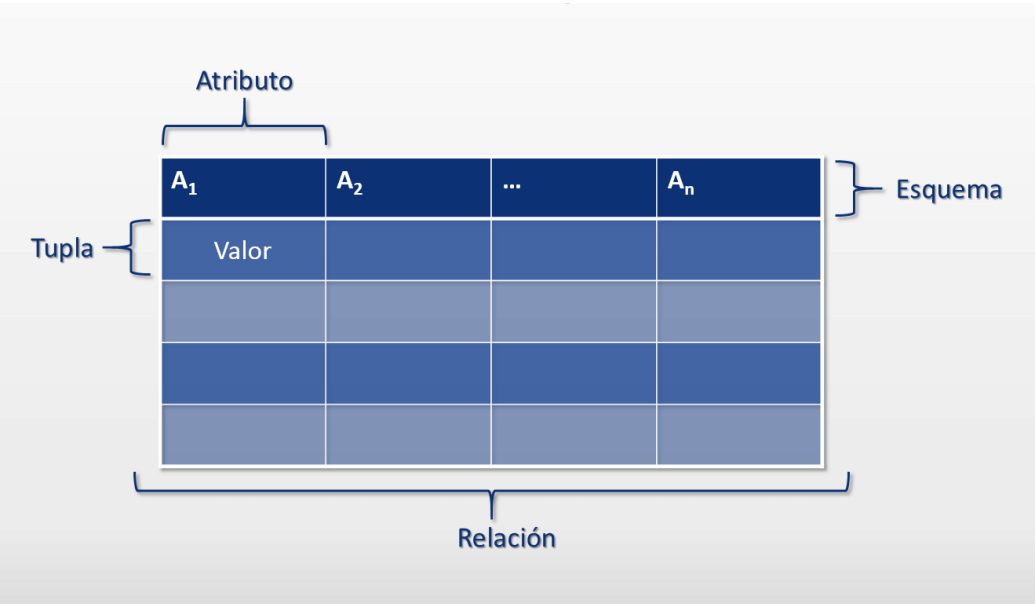


Figura 2.1: Representación del formato de una tabla en bases de datos relacionales. Fuente: Digital Guide - 1&1 IONOS.

Para crear una vista, el SGBD genera una tabla virtual sobre la base de las tablas físicas. En estas vistas pueden emplearse las mismas operaciones que se utilizarían en tablas físicas y, según la función de la vista de datos, pueden distinguirse distintos tipos de vista. Las más habituales son aquellas que filtran determinadas filas (consulta de selección) o columnas (vista de columnas) de una tabla, así como las que conectan diversas tablas entre sí (vista de conjunto).

En el modelo relacional se utiliza de forma estándar para las operaciones, el lenguaje de bases de datos *Structured Query Language* (*SQL*), basado en el álgebra relacional. Las operaciones típicas de las bases de datos como consultar, crear, actualizar o borrar datos se realizan por medio de las llamadas sentencias *SQL*, una combinación de órdenes *SQL*, semánticamente vinculadas al inglés y por este motivo bastante elocuentes.

Las tablas de las bases de datos relacionales se estructuran mediante el uso de claves. Se llama clave primaria (*primary key*) a un campo o a una combinación de campos que identifica de forma única a cada fila de una tabla. No pueden haber dos filas en una tabla que tengan la misma clave primaria.

Por su capacidad para identificar los registros en las bases de datos relacionales, las claves se ajustan a la perfección para interconectar las diferentes tablas que componen una base de datos. Para hacerlo, la clave primaria de una tabla se convierte en la clave externa (*foreign key*) de otra. Las operaciones de base de datos que abarcan varias tablas se realizan en el modelo relacional con ayuda de las llamadas sentencias *JOIN* que pueden traducirse, como la acción de unir o combinar, y en este contexto hacen referencia a una operación que permite consultar varias tablas de datos simultáneamente. Los datos que se extraen de las tablas seleccionadas se agrupan en un subconjunto de todos los posibles resultados y se entregan en función de las condiciones que se han definido.

Conceptos claves a tener en cuenta para la manipulación de bases de datos relacionales:

- Índices: en orden a facilitar las búsquedas de datos a través de elementos que no son su clave primaria, los SGBD implementan los llamados índices, que son estructuras de datos que almacenan la posición de los diferentes atributos seleccionados acelerando así las búsquedas al costo de inserciones más lentas.
- ACID: una serie de propiedades que garantizan que una transacción de alguna base de datos sea confiable. La gran mayoría, sino la totalidad de las bases de datos relacionales, cumplen con estos principios. Explicando aún más, se tiene:
    - Atomicidad se refiere a que toda transacción es indivisible. Todas las declaraciones se aplican o no se aplica ninguna.
    - Consistencia se refiere a que la base de datos permanece en un estado consistente antes y después de la ejecución de una transacción.

- Aislamiento se refiere a que una transacción no debe ver los efectos de otras transacción en curso.
- Durabilidad se refiere a que, ya guardada una transacción en la base de datos, se espera que sus cambios persistan incluso si hay una falla en el sistema operativo o hardware.

- Escalabilidad: las bases de datos relacionales se ven enfrentadas a un dilema cuando se trata de escalabilidad. Ante esto, la opción usualmente tomada es escalar verticalmente, mejorando el hardware del equipo. Como contraparte, cuando se trata de escalar horizontalmente, agregando más nodos que contengan la información de manera distribuida, se pierden las propiedades ACID dejando de lado su principal propuesta de valor. Por lo mismo, la gran mayoría de las bases de datos relacionales no soportan la escalabilidad horizontal.

2.1.3. Bases de datos NoSQL

Amplia clase de SGBD que difieren del modelo clásico relacional en aspectos importantes, siendo el más destacado que no usan *SQL* como lenguaje principal de consultas. Los datos almacenados no requieren estructuras fijas como tablas y no soportan operaciones *JOIN*; tampoco garantizan completamente ACID, pero sí trabajan una consistencia eventual, es decir, que tras un tiempo sin cambios, las diferencias se propagan a todos los nodos del sistema. Habitualmente, escalan bien horizontalmente. En los últimos años un boom de alternativas en este tipo han salido al mercado, cada una de las cuales trata de solucionar los problemas que conlleva *big data* a las bases de datos relacionales. Los sistemas *NoSQL* se denominan a veces "no sólo *SQL*" para subrayar el hecho de que también pueden soportar lenguajes de consulta de tipo *SQL*.

Típicamente las bases de datos relacionales modernas han mostrado poca eficiencia en determinadas aplicaciones que usan los datos de forma intensiva, incluyendo la indexación de un gran número de documentos, la presentación de páginas en sitios que tienen gran tráfico, y en sitios de *streaming* audiovisual, como se indica en [Ranjan, 2014]. En cambio, las bases de datos *NoSQL*, al no sufrir de cuellos de botella, permiten un alto uso de transacciones de escritura y lectura en forma simultánea lo que hace llamativo implementar una base de datos así. Con una base de datos *NoSQL* no es necesario afinar o acotar la

entrada y salida de datos.

Ventajas de trabajar con bases de datos *NoSQL* son:

- Responden a necesidades de escalabilidad horizontal.
- Permiten buen manejo de enormes cantidad de datos.
- No generan el problema de "cuello de botella".
- Permiten escalamiento sencillo y no requieren grandes CPU para ser utilizadas.
- Tienen menores costos asociados a la creación y levantamiento de un SGBD.
- Permiten procesamiento de datos estructurados y no estructurados.
- Tienen esquemas de modelos de datos flexibles, pues cualquier modelo de datos puede ser utilizado, pero no en todo motor *NoSQL*

Por otra parte, desventajas asociadas son:

- *Open source*: sin soporte claro cuando no hay una persona de confiabilidad. Sin embargo, este punto puede ser beneficioso si es posible amoldar y ajustar el código fuente a las necesidades concretas de cada empresa.
- Falta de experiencia: como las empresas no utilizan, generalmente, bases de datos *NoSQL*, es difícil encontrar personas con conocimientos técnicos apropiados.
- Compatibilidad: al no tener una norma común, las interfaces de consultas son únicas y tienen peculiaridades. Es imposible cambiar de un proveedor a otro.

Las principales clases de estas bases de datos, expuestas en [Popescu, 2010], son:

- Orientadas a Columnas: trabajan bajo la misma representación de datos que una relacional, es decir, una tabla. Su diferencia yace principalmente en la manera en que se guardan estos datos en el disco físico, y tal como su nombre lo indica, es a partir de las columnas. Por ejemplo, todos los nombres de pila se almacenan de manera consecutiva, por lo que puede compararse adecuadamente con algunos índices de

bases de datos relacionales. Su principal beneficio es que evita insertar datos repetidos para diferentes registros; de esta manera si se sigue la misma idea del ejemplo anterior, todas las personas con un mismo nombre de pila tendrían almacenado ese único nombre en un mismo lugar. Sus principales representantes son Cassandra [Carpenter y Hewitt, 2016] y HBase [George, 2011], ambos *open source* bajo licencia de Apache.

- Orientadas a Documentos: una base de datos orientada a documentos está diseñada para manejar información semi-estructurada. En vez de un registro con la misma cantidad de atributos que el resto, cada "documento" es un elemento abstracto el cual puede tener más o menos elementos que el documento anterior. La popularidad de esta orientación ha aumentado al unísono con la de *NoSQL*; tiende a ser la preferida para personas que aún no definen de manera estructurada su negocio, ya que pueden ir modificando el modelo de datos sin necesidad de refrescar todo el resto de los elementos. Algunas de las bases de datos populares que funcionan con esta orientación son CouchDB [Anderson *et al.*, 2010] y MongoDB [Chodorow, 2013], ambas *open source*.

- Clave-Valor: trabajan con arreglos asociativos, y al igual que en las bases de datos documentales no requieren de un modelo definido. Existen múltiples SGBD que trabajan bajo la concepción de clave-valor, pero de diferentes maneras; por ejemplo con foco en la consistencia como DynamoDB [Brunozzi, 2012] y Voldemort [Mateljan *et al.*, 2010], utilizadas en memoria principal como Redis [Carlson, 2013] o con una perspectiva en la persistencia de datos como MemcacheDB [Chu, 2008].

- de Grafos: una base de datos orientada a grafos es aquella que permite almacenar la información como nodos de un grafo y sus respectivas relaciones con otros nodos, permitiendo así aplicar la teoría de grafos para recorrer la base de datos; son muy útiles para guardar información en modelos con muchas relaciones como redes y conexiones sociales. Cada nodo consta de un grado que indica el número de aristas que tiene; a su vez un grafo puede ser dirigido o no dirigido, dependiendo de si las aristas tienen nodos origen y nodos destino. El uso de este tipo de bases de datos depende altamente de la lógica de negocio donde se encuentre involucrada la información a almacenar, ya que no puede aplicar en todos los escenarios, o tal vez no

se podría aprovechar su potencial en unos u otros contextos. Algunos de los principales representantes en los SGBD son Neo4j [Huang y Dong, 2013], InfiniteGraph [Kumar Kaliyar, 2015] e IBM-DB2 [Karlsson *et al.*, 2001].

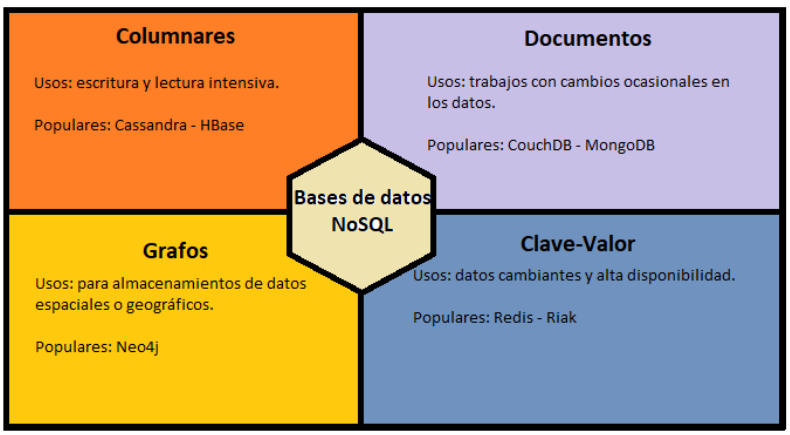


Figura 2.2: Esquema de los principales usos para bases de datos *NoSQL*.
Fuente: Relational Database & Knowledge.

Una tabla comparativa de las clases presentadas sobre bases de datos *NoSQL* es entregada en el cuadro 2.1. El fin, es conocer la capacidad de estas diferentes bases de datos con respecto a su rendimiento, escalabilidad, flexibilidad a cambios, complejidad de uso y funcionamiento.

Cuadro 2.1: Categorización y comparación de bases de datos *NoSQL*.

Fuente: [Popescu, 2010].

	Rendimiento	***Escalabilidad***	***Flexibilidad***	***Complejidad***	***Funcionamiento***
Clave-Valor	Alto	Alto	Alto	N/A	Variable
Columnar	Alto	Alto	Moderado	Bajo	Mínimo
Documental	Alto	Variable	Alto	Bajo	Variable
Grafo	Variable	Variable	Alto	Alto	Teoría grafos

Enfocándose en las bases de datos columnares, es necesario comentar que mientras una base de datos relacional está optimizada para almacenar filas de datos, normalmente para aplicaciones transaccionales, una base de datos columnar está optimizada para lograr una recuperación rápida de columnas de datos, normalmente en aplicaciones analíticas, ya que reduce notablemente los requisitos globales de entrada/salida del disco, y disminuye el volumen de datos que hay que cargar desde él, como se indica en [AmazonWS, 2017].

De la misma forma que otras bases de datos *NoSQL*, las bases de datos columnares están diseñadas para reducir los tiempos de carga utilizando *clústeres* distribuidos de hardware de bajo coste para aumentar el desempeño, de manera que resultan ideales para el almacenamiento y procesamiento de altas cantidades de datos.

Conceptos claves a tener en cuenta para la manipulación de bases de datos columnares:

- Las tablas son indexadas por una única *rowkey*, semejante a clave primaria en bases de datos relacionales.
- La "columna" de datos es un mapa multidimensional ordenado, cuyos valores son identificados por nombre de columna y tiempo.
- Tienen alta disponibilidad y buen escalamiento.
- Soportan datos semiestructurados y aplicaciones de procesamiento analítico en línea (*OLAP*).
- Su modelo de datos se basa en columnas y familias de columnas.

Algunas técnicas de modelado de datos, que son utilizadas en bases de datos columnares y mencionadas en [Katsov, 2012], son:

- Desnormalización (técnica conceptual): puede ser definida como la copia de los mismos datos en múltiples documentos o tablas para simplificar u optimizar el procesamiento de consultas. También, puede utilizarse para ajustar los datos del usuario en un modelo de datos particular. Esta técnica es muy beneficiosa cuando se tiene un alto volumen de datos y se requiere procesamiento complejo de la información, o las consultas a realizar en la base de datos son de muy alto volumen.
- Reducción de la dimensionalidad (técnica general): permite mapear datos multidimensionales a un modelo clave-valor o a otros no multidimensionales.
- Tablas de índices (técnica general): permite aprovechar los índices en almacenamientos que no admiten índices internamente. La idea es crear y mantener una tabla especial con claves que sigan el patrón de acceso. Se considera como un análogo a la utilización de vistas materializadas en las bases de datos relacionales. El formato a cambiar puede apreciarse en la figura 2.3.

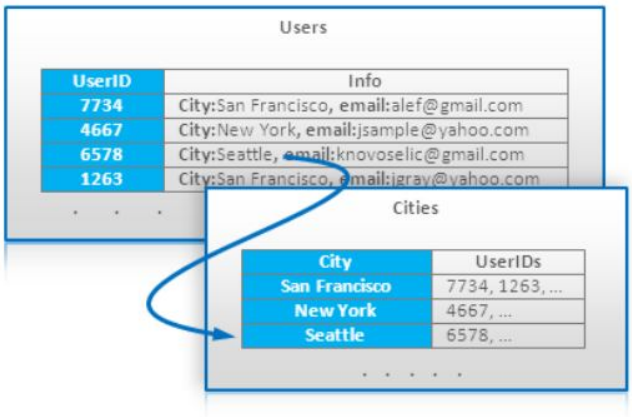


Figura 2.3: Ejemplo de estructura de una tabla índice.
Fuente: *NoSQL Data Modeling Techniques Post*. Highly Scalable Blog, Katsov, Ilya, 2012.

2.2. Inteligencia de negocios

El término inteligencia de negocios (BI) hace referencia al uso de estrategias y herramientas que sirven para transformar información en conocimiento, con el objetivo de mejorar el proceso de toma de decisiones en una empresa. Como se indica en [Gartner, 2013], la inteligencia de negocios es un término que abarca aplicaciones, infraestructura, herramientas y buenas prácticas que permiten el acceso y análisis de la información para optimizar el desempeño y las decisiones. Es importante recordar que el uso de *BI* se realiza con la finalidad de aplicar valor a la empresa.

Desde un punto de vista más pragmático, y asociándolo directamente con las tecnologías de la información, se define BI como el conjunto de metodologías, aplicaciones y tecnologías que permiten reunir, depurar y transformar datos de los sistemas transaccionales e información desestructurada en información estructurada, para su explotación directa (reportería, análisis *OLAP*, alertas, etc.) o para su análisis y conversión en conocimiento [Vitt *et al.*, 2002].

La inteligencia de negocios actúa como un factor estratégico para una empresa u organización, generando una potencial ventaja competitiva, que no es otra cosa que proporcionar información privilegiada para responder a los problemas de negocio: entrada a nuevos mercados, promociones u ofertas de productos, eliminación de islas de información, control

financiero, optimización de costos, planificación de la producción, análisis de perfiles de clientes y muchos más ámbitos.

Entre los posibles usos que se le da a *BI*, se tiene:

- Analítica: procesos cuantitativos para obtener decisiones óptimas y lograr conocimientos del negocio.
- Reportería: creación de infraestructura para reportes estratégicos que sirven al manejo no operaciones de una empresa.
- Plataformas de colaboración: áreas dentro y fuera de la empresa trabajan en conjunto compartiendo datos para fortalecer la toma de decisiones.

Los sistemas y componentes del BI se diferencian de los sistemas operacionales en que están optimizados para preguntar y entregar información a partir de los datos. Esto significa típicamente que en un almacén de datos (*data warehouse*), estos están desnormalizados para apoyar consultas de alto rendimiento, mientras que en los sistemas operacionales suelen encontrarse normalizados para apoyar operaciones continuas de inserción, modificación y eliminación de datos. En este sentido, los procesos de extracción, transformación y carga (ETL) que nutren los sistemas BI, tienen que traducir de uno o varios sistemas operacionales normalizados e independientes a un único sistema desnormalizado, cuyos datos estén completamente integrados. Es importante tener claro, entonces, la arquitectura BI a utilizar.

A pesar que existen variadas formas de armar una arquitectura BI, la mayor parte de ellas incluye una serie de componentes principales que se debe conocer. Uno de los formatos más utilizados se muestra en la figura 2.4 y se detalla a continuación:

- ETL: corresponde a mover datos desde su origen (*Excel*, servicio *web* o una base de datos), para formatear, limpiar y cargar dichos datos en otra base de datos o *data warehouse* para ser analizados y apoyar algún proceso de negocio.
- *Data warehouse*: base de datos con información ya elaborada a partir de los datos de las fuentes citadas en el punto anterior. Para alimentarlos con información se ejecutarán procesos ETL periódicamente.

- Análisis y presentación de datos: se hace un enriquecimiento de esta información mediante sistemas analíticos (cubos OLAP, Minería de Datos, etc.). Además, siempre habrá un último componente que es la capa de presentación, en la que el usuario visualizará y analizará la información, para interactuar con ella y utilizarla como apoyo a la toma de decisiones.

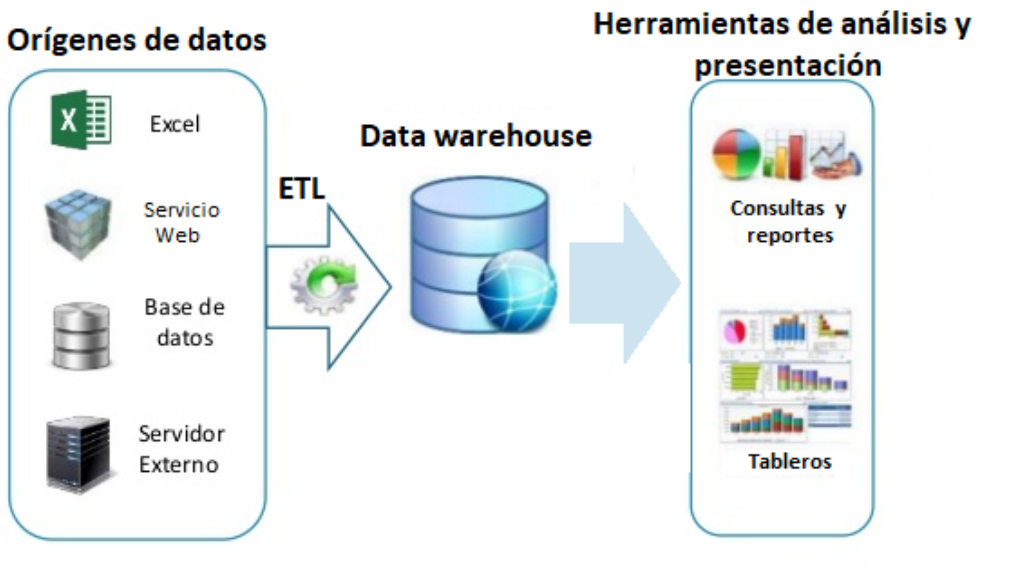


Figura 2.4: Representación de una arquitectura BI.
Fuente: Elaboración Propia inspirado de Profesora Brenda Lopez - Universidad Central de Venezuela.

Se debe tener en cuenta que cualquier arquitectura que se implemente para una solución de BI deben existir todos estos componentes. Ahora bien, la forma en que se dispongan será de cada quien lo implemente, en base a las necesidades que presente.

Como se aprecia en la figura 2.4, el componente análisis y presentación, hace referencia a herramientas de BI para apoyar la labor de visualización de la información. Las principales ventajas que se tiene al utilizar herramientas BI, evidenciadas en [Calzada y Abreu, 2009], son:

- Alta profundidad en análisis de datos.
- Amplia capacidad de reportería.

- Análisis en retrospectiva, gracias a la posibilidad de guardado de registros históricos.
- Capacidad de realizar proyecciones y pronósticos de futuro en base a toda la información.

Si se habla de los tipos de herramientas BI existentes, es posible categorizar de la siguiente manera:

1. Gestión de datos: permiten la depuración y estandarización de datos de procedencia diversa hasta la extracción, transformación y traslado de la información a un determinado sistema.
2. Descubrimiento de nuevos datos: permiten recopilar y evaluar nueva información (minería de datos), y aplicar sobre esa información técnicas de análisis predictivo para realizar proyecciones de futuro.
3. Reportería: con la información ya recopilada, permiten a las empresas visualizar los datos de manera gráfica e intuitiva. También sirven para integrarla en cuadros de mando que midan si se cumplen o no determinados *KPI's*.

Existen muchas herramientas BI en el mercado, y consultores e investigadores de tecnologías que dedican su tiempo a buscar cual de todas estas herramientas existentes logran potenciar la visión de una empresa. Gartner Inc, anualmente, publica información relevante a muchos ámbitos de la gestión de procesos e información, con el fin de mostrar aquellas empresas que logran perseguir y ejecutar la visión estratégica que tiene una empresa. Como es posible observar en la figura 2.5, Microsoft, Tableau y Qlik, son las 3 empresas líderes en este mercado y que, por supuesto, cuentan con sus herramientas de BI. Esta imagen, es conocida como el "Cuadrante Mágico de Gartner" que, para este caso en particular, consiste en una Matriz de Competitividad que representa, a nivel global, la posición estratégica en el mercado de los actores de plataformas de BI. La clasificación se realiza según los siguientes criterios:

- Eje X: "Integridad de visión". Se tienen en cuenta: comprensión del mercado, estrategia de marketing y de ventas, enfoque para el desarrollo y la entrega de productos, modelo de negocios e innovación, entre otras cuestiones.

- Eje Y: “Habilidad para ejecutar”. Se tienen en cuenta: productos principales y servicios ofrecidos de acuerdo a su calidad y características, evaluación de la salud financiera de la organización en general, ejecución de ventas y fijación de precios, capacidad de respuesta o de registro del mercado y experiencia del cliente, entre otras.

Figura 2.5: Cuadrante mágico de Gartner para análisis y plataformas de inteligencia de negocios publicado en febrero del 2018.

Fuente: Gartner - Febrero 2018.

Las plataformas de análisis y de BI son analizadas de acuerdo a la infraestructura, manejo de datos, análisis y creación de contenido. A su vez se toman 5 casos de uso: Suministro ágil y centralizado de BI, Análisis descentralizado, Descubrimiento de datos gobernados, BI incorporado e Implementación de “Extranet”.

Según [Innovación-Necesaria, 2018], lo que se ha valorado para incluir a Microsoft en la parte superior de su Cuadrante Mágico son las siguientes fortalezas:

- Facilidad de uso e interfaz atractiva: según el informe de Gartner, para un 14 % de los clientes de Microsoft la usabilidad es el principal criterio de compra. La importancia

de ambas cualidades en *MS Power BI* forma parte de la estrategia "cinco segundos para registrarse y cinco minutos para sorprender al cliente".

- Costes muy competitivos de las licencias en sus diferentes versiones: incluye una versión gratuita para uso de aplicación en Escritorio. El estudio de Gartner considera decisivas ambas cuestiones a la hora de elegir una plataforma de BI u otra.
- Visión global de producto: la estrategia diferenciadora de Microsoft con otros de sus productos como Cortana, Surface Hub, Dynamics CRM, etc. son notorias y, además, su fácil integración con *MS Power BI* es otra de las fortalezas que Gartner ha extraído en su informe para situar esta herramienta de BI en la parte superior del cuadrante
- Experiencia de uso de los clientes: los clientes de Microsoft han valorado muy positivamente su experiencia con la marca, especialmente, a la hora de compartir información con el resto de usuarios. La multinacional tecnológica cuenta con una gran comunidad de compañeros (*partners*), revendedores (*resellers*) y particulares que disponen de una gran variedad de recursos para compartir dicha información (experiencias, casos prácticos, analíticas, etc.) con clientes, entre empleados o con otras compañías.

La herramienta BI *MS Power BI*, ofrece también:

- Mejor integración con sistemas de bases de datos *NoSQL*.
- Paneles de información con mejor edición.
- Paneles de navegación fáciles de usar.
- Mejor visibilidad al incorporar múltiples fuentes de datos.
- Integraciones nativas con exportaciones a documentos como Adobe Reader (PDF) y MS Excel.
- Posibilidad de ver los datos en algún sistema operativo Android.

Al igual que para Microsoft, las principales fortalezas para los productos de Qlik que observó Gartner, según [DataIQ, 2018], son las siguientes:

- Producto escalable para aplicaciones robustas: QlikView y Qlik Sense se pueden usar como un tipo de mercado de datos.
- Marketing diferenciado desde el año pasado (2017): Qlik hizo mucho más para proporcionar ejemplos claros de negocios y mejorar la alfabetización de datos como parte de un programa general de análisis.
- Visión del producto: Qlik desarrolló aún más la capacidad de analizar datos en reposo y análisis de transmisión, multi-nubes, *Big Data* y su motor de recomendación.
- Red de socios: se estima que el 70 % de las implementaciones de Qlik están dirigidas por socios que a menudo tienen relaciones a largo plazo con sus clientes y entienden sus requisitos particulares.

2.3. Heurísticas de Usabilidad

Hoy en día existen muchos tipos de pruebas para evaluar la usabilidad y asegurar que los diseños ofrecen una buena experiencia al usuario. Uno de los marcos de referencia más conocidos y utilizados en el sector son las 10 heurísticas de Jakob Nielsen [Nielsen, 1995].

Se trata de diez normas que toda interfaz interactiva debería cumplir para los usuarios. Se debe estar alerta con este grupo de reglas generales antes, durante y después tanto de crear proyectos como a la hora de realizar cambios. Las heurísticas son las siguientes:

- Visibilidad: explicar al usuario cuál es el estado del sistema en cada momento, y mantenerlo informado de lo que está pasando.
- Relación con la realidad: utilizar un lenguaje familiar y apropiado para los usuarios. Se organiza la información con un orden natural y lógico.
- Control y libertad: ofrecer funciones de rehacer y deshacer que permitan al usuario tener el control de sus interacciones con libertad.
- Consistencia y estándares: establecer convenciones lógicas y mantenerlas siempre (mismo lenguaje, mismo flujo de navegación, etc).

- Prevención de errores: se debe ayudar a los usuarios a evitar equivocarse antes de que cometan el error.
- Reconocimiento: se debe hacer visible todo lo que sea posible. Evitar que los usuarios recuerden o memoricen información.
- Flexibilidad: permitir que el sistema pueda adaptarse a los usuarios frecuentes. Se debe diseñar la realización de tareas avanzadas de manera fluida y eficiente.
- Estética y minimalismo: mostrar sólo lo necesario y relevante en cada situación.
- Recuperarse de los errores: ayudar a los usuarios a reconocer y corregir los errores que sufran. Se deben sugerir soluciones constructivas.
- Ayuda y documentación: la información de ayuda debe ser breve, concisa, fácil de buscar y enfocada a las tareas del usuario.

Capítulo 3

Propuesta de Solución

En este capítulo se presenta el diseño del ambiente operacional propuesto en la empresa en estudio. Además, se indica el modelo de datos y los procesos ETL que se utilizan en las replicas de las dos bases de datos: una relacional (*MySQL*) y otra columnar (*Cassandra*). También, se evidencian los criterios de comparación que se utilizan para realizar las validaciones pertinentes.

Finalmente, se exhibe una propuesta para la visualización de datos a considerar en la herramienta BI, con un diseño basado en las necesidades de la empresa y heurísticas de interfaces usuarias.

3.1. Arquitectura del sistema

La arquitectura a utilizar se basa en la figura 2.4, presentada en el capítulo anterior. Para este caso en particular, se tienen las siguientes características:

- Orígenes de datos: corresponden a archivos Excel extraídos desde el ERP *Random* que, en la empresa en estudio, son llamados “documentos maestros”. Estos archivos son consolidados de datos mantenidos como respaldo a la información que maneja el ERP.

- ETL: el proceso de extracción se realiza en el mismo ERP *Random* dado que es necesario generar archivos Excel correspondiente a los cuatro años operativos de ventas que aborda este trabajo. También, para la consistencia de los datos extraídos, se realiza un proceso de transformación basado, principalmente, en mantener la codificación. Además, los archivos deben ser limpiados de sus datos nulos para, posteriormente, modificar la extensión de estos (de *.xls* a *.csv*). Finalmente, para realizar las cargas de estos datos, se utilizan características propias de los *software* empleados en este trabajo (*MySQL* Workbench y base de datos columnar). Estas características permiten realizar una importación y exportación, de ser necesario, a los datos de manera masiva lo que permite ahorrar tiempo en este proceso.

- *Data warehouse*: se tendrán modelos característicos de la base de datos *MySQL* y de la base de datos *Cassandra*. Para el caso relacional, se utilizan varias tablas de dimensiones y una tabla de hechos que recopilan los datos tratados. Para el caso no relacional, se usa una gran tabla de datos.

- Herramientas de análisis y presentación: se utiliza *MS Power BI* como herramienta de análisis para los datos. La elección y uso de esta herramienta se basa en su alta competitividad en el mercado y su licenciamiento gratuito. La presentación de esta información, se realiza mediante paneles confeccionados en la misma herramienta.

Una imagen explicativa de la arquitectura, puede ser observada en la figura 3.1.

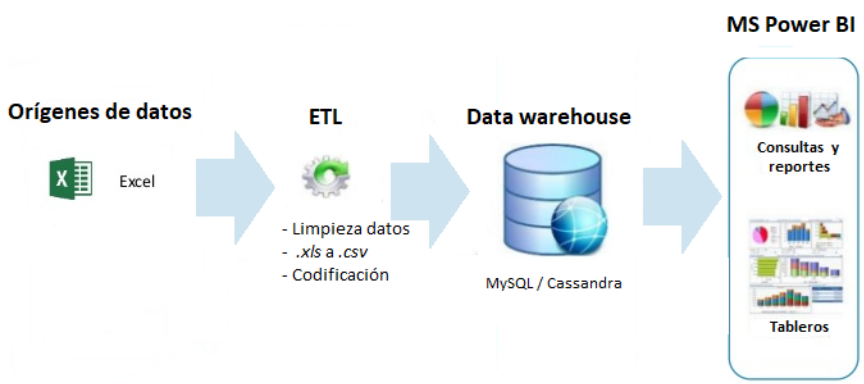


Figura 3.1: Arquitectura que tiene el sistema a usar.
Fuente: Elaboración propia.

La arquitectura presentada tiene que satisfacer las principales necesidades de la empresa al momento de buscar información. Para esto, debe ser capaz de entregar información correspondiente a:

- *Ranking* de vendedor con más ventas por trimestre.
- *Ranking* de comunas con más ventas.
- *Ranking* de tipo de cliente con más ventas
- *Ranking* de tipo de producto más comprado entre los clientes.
- Productos retirados de la bodega.

La necesidad de esta información se basa, principalmente, en conocer el trabajo realizado por los vendedores, ya que ellos no son controlados cuando hacen sus rutas. Por ejemplo, conocer el vendedor con más ventas por trimestre permite a esta persona adjudicarse un bono por su buen desempeño. La información referente a la comuna con más ventas, permite al gerente operacional entender cuáles son los sectores o zonas que deben ser atendidas con mucho más énfasis por los vendedores. También, esta información muestra si el vendedor visita o no a los clientes.

Por otra parte, conocer los productos más comprados y los clientes con más ventas permite considerar nuevas promociones o descuentos asociados a las compras para fidelizar clientes con respecto a la competencia. Finalmente, conocer los productos retirados de la bodega permite un control del *stock* que, actualmente, no es muy bien manejado dado que se realiza mediante cuadraturas manuales por parte de los repartidores o un estimado "visual" de la cantidad de productos en la bodega. No se lleva un control como tal de estos números.

3.2. Diseño

Se replica el sistema del ERP *Random* en dos bases de datos: una relacional (*MySQL*) y otra columnar (*Cassandra*). Cada una maneja distintos aspectos en el diseño y que se basan en las necesidades de la empresa en estudio. Se debe considerar lo siguiente.

3.2.1. Modelo de datos

Para el caso de la base de datos *MySQL*, el modelo a utilizar se presenta en la figura 3.2. Cada tabla contiene su clave primaria (PK) y, de ser necesario, se utilizan claves foráneas (FK), como es el caso de la tabla central.

Se puede apreciar que el modelado corresponde a un esquema estrella, ya que se busca enfocar el análisis a realizar en la tabla central. Para este caso, las tablas de dimensiones, corresponden a las tablas clientes, vendedor, tiempo, bodega y producto. En cambio, la tabla central, corresponde a la tabla de hechos (tabla venta). Dentro de las dimensiones, los datos son, generalmente, textos y, en la tabla de hechos, se encuentran los datos numéricos relevantes para la realización de análisis.

Este esquema es una visión general y abstracta del proceso más crítico que tiene la empresa en estudio, proceso que fue ilustrado en la figura 1.1. Se estructuró de forma que se relacionaran los datos que se disponen. Por otra parte, cada par de tablas relacionadas tiene una cardinalidad "uno a muchos" debido a que la relación entre los datos se da de esta forma.

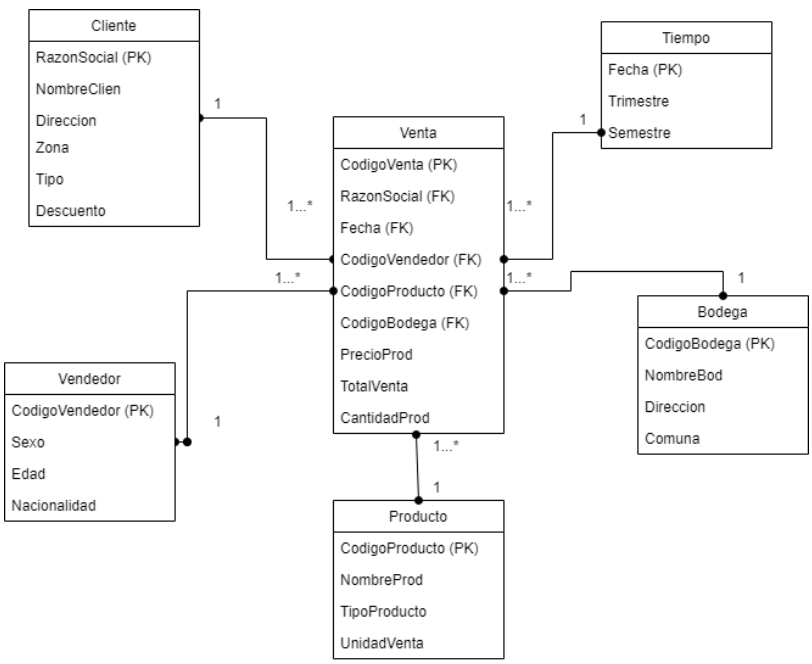


Figura 3.2: Representación UML basada en el modelado de la operación de la empresa en estudio. *PK* indica que atributo es clave principal y *FK* indica que atributo es clave foránea. Fuente: Elaboración Propia.

Para el caso de *Cassandra*, se debe "aplanar" el modelo estrella del caso anterior. El tabajo de aplanado hace referencia a traspasar el modelo completo a una sola gran tabla de datos. El modelo UML puede observarse en la figura 3.3.

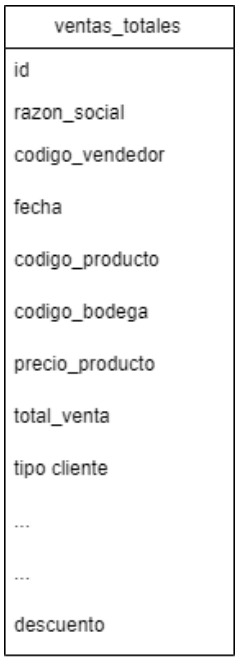


Figura 3.3: Modelo aplanado utilizado en base de datos Cassandra. Dada su extensión, se minimiza para indicar que el formato que presenta es producto de todos los datos del modelo estrella en una sola gran tabla.

Fuente: Elaboración Propia.

3.2.2. ETL

Para el caso de la base de datos *MySQL*, se realiza una selección de los datos más relevantes mediante eliminación de registros duplicados y registros nulos. Esta preparación genera datos de calidad, los cuales pueden conducir a encontrar patrones o anomalías. Posteriormente, se deben traspasar estos archivos a un formato *.csv*. Finalmente, el proceso de carga se realiza con *MySQL* Workbench. Este *software* solo permite archivos que tienen la extensión *.csv* por lo que, el paso anterior, es un punto necesario.

Para el caso de *Cassandra*, los archivos son obtenidos de *MySQL* Workbench en formato *.csv*. Se debe modificar este archivo para incorporar el formato de codificación "UTF8" con el fin de que los nombres colocados dentro del archivo no pierdan su formato. Para esto,

se utiliza un comando del sistema operativo Linux que viene de forma nativa. El comando tiene la siguiente estructura:

```
iconv -f ISO8859-1 -t UTF8
carpeta_contenedora/archivo_original.csv >
carpeta_contenedora/archivo_original_utf8.csv
```

Luego, para la inserción de los datos, se utiliza la consulta *COPY* que importa, de manera completa, el archivo *.csv* realizando las inserciones en los lugares apropiados y sin errores. La consulta tiene el siguiente formato:

```
COPY ventas_totales (id,razon_social,codigo_vendedor,fecha,
codigo_producto,codigo_bodega,precio_producto,
total_venta,cantidad_producto,sexo,edad,
nacionalidad,nombre_bodega,direccion,
comuna_bodega,trimestre,semestre,
nombre_producto,tipo_producto,unidad_venta,
zona_venta,tipo_cliente,descuento)
FROM '/src/ventas_all_utf8.csv'
WITH HEADER = TRUE and DELIMITER = ';';
```

COPY permite, también, exportar los datos a un archivo *.csv*. La consulta para realizar esta acción se encuentra en anexo A.3.

3.2.3. Criterios de Comparación

Basado en el diseño y las necesidades que la empresa en estudio tiene, se comparan los casos propuestos (relacional y *NoSQL*) según los siguientes aspectos:

- Tiempos: de ejecución en la creación de la base de datos, de respuesta a consultas, de limpieza, de almacenamiento y exportación de los datos.
- Costos: asociados a licenciamiento, mantención y uso de del servicio. También, costos asociados a la instalación y utilización de las bases de datos presentadas.

- Arquitectura: adopción de la arquitectura propuesta.

El método para evaluar estos criterios será una comparación. Para el caso del tiempo, se buscará tener tiempos razonables en las ejecuciones que se realizan. Para el caso de los costos, se compara por cantidad de dinero utilizado o necesario para la realización de los procesos. La importancia de hacer estas comparaciones es apoyar la gestión operacional disminuyendo el tiempo necesario para tomar decisiones. También, mostrar que es posible disminuir los costos operacionales asociados, actualmente, a los procesos operativos.

3.2.4. Paneles BI

La presentación de la información se realiza con ayuda de la herramienta *MS Power BI*. Esta herramienta permite generar paneles con la información que se requiera y, para este caso, se satisfacen las necesidades de la empresa que fueron presentadas anteriormente. El uso de la herramienta BI y los paneles creados, son evaluados siguiendo las heurísticas de Nielsen para los siguientes casos:

- Visibilidad: se chequea que la herramienta sea capaz de informar al usuario sobre lo que está aconteciendo en los paneles de información y, asegurar, que esta retroalimentación se realice en un tiempo razonable.
- Control y libertad: se evalúa que la herramienta sea capaz de salir de estados indeseados de manera rápida y sin dificultad alguna.
- Consistencia y estándares: se evalúa que la herramienta tenga una consistencia clara tanto en el uso como en su visión.
- Flexibilidad: se chequea que la herramienta permita acelerar algunos procesos de creación, por ejemplo, la utilización de atajos. Además, se evalúa que la herramienta permita personalizar acciones frecuentes.
- Ayuda y documentación: se chequea que la herramienta tenga disponible para el usuario ayuda o documentación si este la requiere.

La evaluación de las heurísticas debe indicar que la herramienta utilizada no presenta dificultades en su uso. Además, no se utilizan todas las heurísticas de Nielsen, ya que

algunas no son afines a este trabajo de investigación y la propuesta de presentación de información debe ser lo más simple posible. Por otra parte, se debe contemplar que el uso de estas heurísticas, y su evaluación, deben enfocarse a la finalidad que tiene esta herramienta, que es la de apoyar la toma rápida y eficiente de decisiones. Con esto, la validación de las heurísticas, se realiza en la propuesta de interfaz para la reportería.

Como parte del diseño de los paneles, se presentan requisitos entregados por los usuarios y que se ven reflejados en los números de los bosquejos presentados en las figuras 3.4 y 3.5:

- (1): la muestra de la información debe ser concisa. El uso de gráficos de torta o barra se permite, siempre y cuando se usen colores claros para diferencias las comunas o zonas involucradas. Con respecto al tamaño, no debe superar a las demás muestras pero si debe hacerse notar.
- (2): se debe utilizar un formato de tabla dinámica. Es necesario observar todos los vendedores asociados a la distribuidora por su código y por tipo de cliente que visita. El dato referente al total de ventas, debe ser un número.
- (3): los datos debe ser entregados en un formato de gráfico de barras para que la vista sea clara y de fácil entendimiento. No profundizar en la entrega de datos, sino más bien, realizar una entrega de barra por colores para conocer los productos que se han vendido. De ser posible incluir a los vendedores en la muestra. El tamaño del gráfico debe ser consistente a la cantidad de datos mostrados.
- (4): utilizar un gráfico de barras que presente los datos claramente. No utilizar un gran tamaño, ya que estos datos no son utilizados recurrentemente. Incluir el total de ventas como número.
- (5): utilizar un formato de gráfico de fácil lectura donde la interpretación de los datos sea clara. Solo entregar números.
- (6): presentar los datos en un *ranking* de los vendedores que tienen más ventas totales. Incluir las ventas en semestres y trimestres de las comunas que ellos atendieron. La información necesaria del vendedor corresponde, solamente, al código que tiene asociado. La presentación debe ser en un formato de tabla dinámica.

- (7): presentar los datos en formato de gráfico de barra para lograr evidenciar, visualmente, cuales son los tipos de productos más vendidos. Incluir la publicidad en estos datos. Esta información debe destacar con respecto a las otras, por lo que se recomienda utilizar un tamaño considerable.

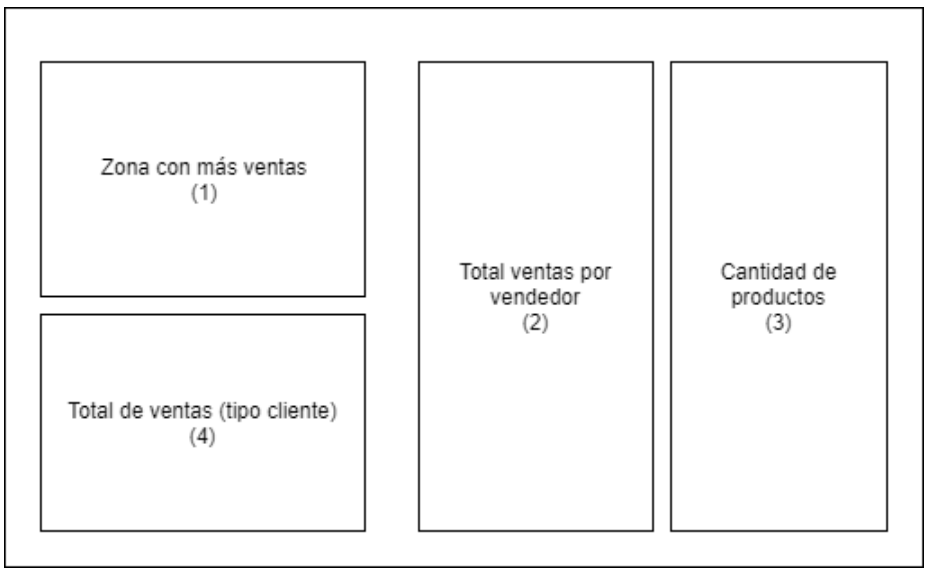


Figura 3.4: Bosquejo del primer panel de presentación de datos en la herramienta BI
Fuente: Elaboración Propia.

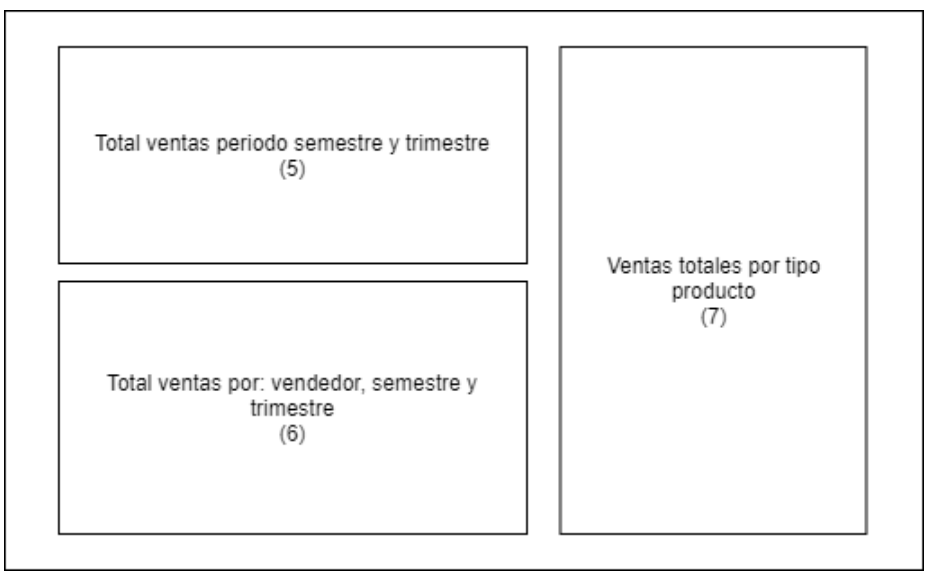


Figura 3.5: Bosquejo del segundo panel de presentación de datos en la herramienta BI
Fuente: Elaboración Propia.

Capítulo 4

Implementación y Validación

En este capítulo se aborda la implementación de la propuesta de solución diseñada en el capítulo anterior. Se presenta el hardware utilizado, los tiempos, costos y adopción de la arquitectura para las bases de datos señaladas. Además, se realizan las validaciones pertinentes en los dos ambientes implantados (actual y propuesta) con la herramienta BI para el análisis de datos.

Finalmente, se presenta cómo las heurísticas de Nielsen actúan en la visualización de los datos y resultados obtenidos con la herramienta BI.

4.1. Hardware utilizado

Se consideran las mismas características técnicas del equipo donde está montado, actualmente, el sistema ERP *Random*. Se recuerda que los datos utilizados corresponden a ventas realizadas desde el año 2014, fecha de inicio de operación de los servicios, hasta Diciembre del año 2017. Los clientes son quienes presentaron compras durante el periodo señalado de manera "recurrente". es decir, al menos una compra al mes. Los datos de clientes sin compras en periodos de un trimestre son manejados en una lista "negra" por la empresa en estudio y es información que no es incluida en esta investigación. El servidor está montado sobre un sistema operativo Windows Server 2008, tiene una conexión de internet de Fibra Óptica con una velocidad de 50 *Mbps*. Se ha evitado la instalación de

software externo como, por ejemplo, impresoras, ya que no serán utilizadas en esta investigación y no influye en el ambiente operacional. El manejo de este tipo de periféricos es de exclusiva responsabilidad de la empresa y cada instalación necesaria es realizada por el soporte técnico que ellos tienen.

La base de datos relacional *SQL*, se prepara en un equipo que presenta las características técnicas que se detallan en el cuadro 4.1. El método de creación de la base de datos *MySQL* puede observarse en el anexo A.1. Se utiliza el ambiente "DBmemoriaV1".

Cuadro 4.1: Información del equipo replica que contiene Windows.

Fuente: Elaboración Propia.

Sistema Operativo	Windows Server 2008 SP2
Procesador	Intel Xeon E312xx 2,19 Ghz (4 Procesadores)
Ram	8,00 GB
Tipo de Sistema	64 bits

La base de datos columnar *Cassandra*, se prepara en un equipo que presenta las características técnicas que se detallan en el cuadro 4.2.

Cuadro 4.2: Información del equipo replica que contiene Linux.

Fuente: Elaboración Propia.

Sistema Operativo	Ubuntu 18.04.1 LTS
Núcleo	Linux (Monolítico)
Procesador	Intel Xeon E312xx 2,19 Ghz (4 Procesadores)
Ram	8,00 GB
Tipo de Sistema	x86-64 bits

El método de creación de la base de datos *Cassandra* puede observarse en el anexo A.2. Se utiliza el ambiente "dbmemov2". Se destaca que al momento de crear, es necesario colocar un dato de la columna como índice. Para esta propuesta de ambiente operacional, se realizan tres pruebas para conocer cuál de estos atributos es mejor para el acceso a los datos de manera independiente a la resolución de las consultas requeridas. Las pruebas se hacen en:

- ID transacción como índice (número de transacción realizada).

- Razón Social del cliente como índice.
- Código de vendedor como índice.

4.2. Base de datos relacional (*MySQL*)

La obtención de los datos se realiza directamente con el ERP *Random*. Se generan dos grandes informes que contienen la información de cuatro años de operación. Este proceso demora bastante tiempo debido a que el ERP no realiza ningún tipo de validación para la generación de los archivos y, generalmente, ocurre que se excede la capacidad máxima de datos que pueden ser ingresados en *Excel* (la extensión del archivo es *.xls*). Es necesario destacar que el proveedor de archivos *Excel* indica que el máximo número de filas para este tipo de archivos es de 1.048.576 [Microsoft, 2017]. Por este motivo, los archivos fueron particionados por semestre para así reducir la cantidad de filas obtenidas en los *Excel*. Por otra parte, cada archivo contiene 81 columnas de datos.

Aproximadamente, el tiempo para generar informes, es de 10 horas y, como los usuarios concurrentes permitidos por el ERP son dos, la obtención de la información debe hacerse en días no hábiles para así no afectar el proceso de ventas. Por otra parte, para los datos referentes a los clientes, se dispone de un documento ya consolidado con los datos necesarios y que, por necesidades de respaldo, es mantenido fuera del ERP *Random*. Por último, los datos referentes a productos, son obtenidos en la creación de otro informe que demora mucho menos tiempo (aproximadamente 15 minutos).

Los datos necesarios de vendedores y bodega, son adquiridos directamente en la empresa, realizando consultas a los trabajadores. La obtención de cada código de vendedor se extrae desde el ERP *Random*.

Con el proceso de extracción ya completo, es necesario limpiar los archivos generados para eliminar información inútil. Como el ERP *Random* tiene todos los módulos activos, pero no operativos, se tiene bastante información nula dentro de los archivos *Excel* creados. El tiempo utilizado en realizar la limpieza es de, aproximadamente, una hora y media, debido a que por cada archivo semestral generado (para cada año), se deben realizar diversas tareas de limpieza. La integración de datos es un proceso *JOIN* de los seis archivos semestrales generados; además, es necesario cambiar la extensión del archivo a *.csv*.

La carga de datos, para dar fin al proceso de ETL, se realiza mediante el apoyo de la herramienta *MySQL* WorkBench. Se utiliza el servicio de importación de datos mediante los archivos *.csv*. También, con *MySQL* Workbench es posible exportar los datos de cada tabla asociada a la base de datos. Este proceso es un poco tedioso dado que se debe realizar la exportación de datos tabla por tabla con ayuda del asistente, esperando un tiempo considerable. Para mejorar este proceso de exportación, se utiliza una operación *JOIN* que integra toda la información para que el asistente genere un archivo de salida consolidado. La consulta *JOIN* mencionada es la siguiente:

```
1 -- Join tabla venta
2 SELECT * FROM venta join vendedor ON venta.vendedor_CodigoVendedor =
3 vendedor.CodigoVendedor
4 join bodega ON venta.bodega_CodigoBodega = bodega.CodigoBodega
5 join tiempo ON venta.tiempo_Fecha = tiempo.Fecha
6 join producto ON venta.producto_CodigoProducto =
7 producto.CodigoProducto
8 join cliente ON venta.cliente_RazonSocial = cliente.RazonSocial;
9 -- fin consulta
```

Se contabiliza un total de 5 horas y media de operación para insertar, dentro de la base de datos *MySQL*, todos los datos que se manejan. Estos tiempos se alcanzan debido a que el sistema deja de responder en ciertos momentos en que se efectúa la carga de datos. Los periodos "muertos" de tiempo son desde el comienzo de importación hasta que es desplegada la información en *MySQL* WorkBench.

Otro tipo de instrucciones *SQL* (como *SELECT*, *UPDATE* o *DELETE*) que son realizadas a la base de datos, no demoran más de dos minutos en ser ejecutadas satisfactoriamente.

4.3. Base de datos columnar (*Cassandra*)

Los datos para este ambiente operacional vienen desde la consulta *JOIN* realizada en la base de datos relacional. Continuando con el proceso ETL, se debe realizar el ajuste de formato UTF8 al archivo *.csv*. Este proceso no toma más de un minuto dado que no es

necesario instalar ningún paquete extra en Linux para su realización.

Los tiempos asociados a la carga de datos, que se realiza con el comando *COPY* presentado en el anexo A.3, se reducen considerablemente. El proceso documentado, se puede observar en las figuras 4.1, 4.2 y 4.3. Se destaca que al momento de utilizar el código de vendedor como índice de la tabla de datos, esta inserción demora menos tiempo. Con este antecedente, se decide operar el acceso a los datos con código vendedor como índice de entrada. No se observaron mayores problemas con el proceso ETL.

Las consultas realizadas a la base de datos se ejecutan bastante rápido, y están en el orden de las centésimas de segundo. Se evidencia mayor rapidez en este ambiente operacional.

```
DBMEMOV1

-Creación Base de datos con Script

0.001 seconds.

-Proceso de importación de los datos usando COPY:

Using 3 child processesStarting

copy of dbmemov1.ventas_totales with columns [id, razon_social, codigo_vendedor, fecha, codigo_producto,
codigo_bodega, precio_producto, total_venta, cantidad_producto, sexo, edad, nacionalidad, nombre_bodega,
direccion, comuna, trimestre, semestre, nombre_producto, tipo_producto, unidad_venta].

Processed: 141173 rows; Rate:
4086 rows/s; Avg. rate:
5460 rows/s

141173 rows imported from 1 files in 25.858 seconds (0 skipped).
```

Figura 4.1: Tiempos de ejecución en Cassandra, con ID Transacción como índice. Fuente: Elaboración Propia.

```
DBMEMOV1

-Creación Base de datos con Script

0.001 seconds.

-Proceso de importación de los datos usando COPY:

Using 3 child processesStarting

copy of dbmemov1.ventas_totales with columns [id, razon_social, codigo_vendedor, fecha, codigo_producto,
codigo_bodega, precio_producto, total_venta, cantidad_producto, sexo, edad, nacionalidad, nombre_bodega,
direccion, comuna, trimestre, semestre, nombre_producto, tipo_producto, unidad_venta].

Processed: 141173 rows; Rate:
4286 rows/s; Avg. rate:
5160 rows/s

141173 rows imported from 1 files in 32.858 seconds (0 skipped).
```

Figura 4.2: Tiempos de ejecución en Cassandra, con Razón Social como índice. Fuente: Elaboración Propia.

```
DBMEMOV1

-Creación Base de datos con Script

0.001 seconds.

-Proceso de importación de los datos usando COPY:

Using 3 child processesStarting

copy of dbmemov1.ventas_totales with columns [id, razon_social, codigo_vendedor, fecha, codigo_producto,
codigo_bodega, precio_producto, total_venta, cantidad_producto, sexo, edad, nacionalidad, nombre_bodega,
direccion, comuna, trimestre, semestre, nombre_producto, tipo_producto, unidad_venta].

Processed: 141173 rows; Rate:
3952 rows/s; Avg. rate:
4267 rows/s

141173 rows imported from 1 files in 18.251 seconds (0 skipped).
```

Figura 4.3: Tiempos de ejecución en Cassandra, con Código Vendedor como índice.
Fuente: Elaboración Propia.

4.4. Comparación y análisis

Dados los aspectos presentados en ambos ambientes operacionales, los **tiempos** asociados a creación de la base de datos, limpieza, almacenamiento y exportación de datos, así como los de respuesta ante consultas son los presentados en el cuadro 4.3. Para esta evaluación comparativa se dejan de lado los tiempos de adquisición de datos para ambas bases de datos presentadas, debido a que hay una gran diferencia porque la base de datos *Cassandra*, utiliza la información proveniente del *JOIN* de la base de datos relacional.

Cuadro 4.3: Mejores tiempos de ejecución encontrados las réplicas del ambiente operacional.

Fuente: Elaboración Propia.

	Base de datos	
	SQL	NoSQL
Creación	0,525 segundos	0,001 segundos
Limpieza de datos	1 hora 17 minutos	7,392 segundos
Almacenamiento de datos	5 horas 30 minutos	18,251 segundos
Exportación de datos	54,234 segundos	32,213 segundos
Tiempo de respuesta a consultas	1 minuto 12 segundos	0,010 segundos

Los tiempos expuestos en la tarea de almacenamiento, se refieren a un tiempo en almacenaje masivo de datos y no por secciones. Se destaca que al realizar un trabajo parcial en la inserción de datos los tiempos se reducen en el orden de minutos, pero fue imposible disminuir las 5 horas. Por otra parte, la cantidad de datos ingresados, y que se puede apreciar en la figura 4.3, es de 141.173 filas de datos, que para una base de datos de empresa no es un número grande. Con esto, se evidencia que el volumen de datos es un aspecto importante a considerar dado que si los clientes siguen creciendo, que es una tendencia normal y esperada en una distribuidora, los tiempos de ejecución podrían aumentar. El principal problema observado en *MySQL* es el cuello de botella que producen la lectura y escritura de datos, que se hacen de manera simultánea y, considerando que existen 2 cuentas concurrentes en el ERP *Random*, este problema seguirá existiendo por lo que se pronostica que los tiempos expuestos se elevarán.

Por otra parte, la base de datos Cassandra no tiene este problema con el volumen de datos actual; tampoco sufre del cuello de botella presentado anteriormente. Además, es necesario destacar el gran rendimiento que tiene para generar informes de ventas anuales. El índice finalmente utilizado por la base de datos *Cassandra* fue sobre el dato "código vendedor" que fue el que mejor tiempo de respuesta tuvo con las consultas presentadas. Lo anterior, se debe a que la mayor parte de la información relevante es entregada directamente por el vendedor y, por ende, la agrupación de estos datos se da casi de manera natural a pesar que los otros dos índices (ID transacción y razón social) podían responder las mismas consultas de prueba utilizadas (generación de un informe anual de ventas) solo que que no de forma tan rápida como lo hizo el índice sobre "código vendedor". Los tiempos de cada índice, que se observan en las figuras 4.1, 4.2 y 4.3, son:

- ID Transacción: 25.858 segundos.
- Razón social: 32.858 segundos.
- Código Vendedor: 18.251 segundos.

La utilización de un sistema *NoSQL* es factible en la empresa en estudio, donde cada uno de los módulos que se cobran en el ERP *Random* son totalmente replicables a excepción del de facturación. Para esto, y así no realizar un cambio drástico de paradigma de trabajo, es posible conservar el uso del servidor que recepciona los datos enviados mediante el sistema

móvil. Luego, mediante el uso de ODBC, conectar la base de datos de Cassandra para que esta pueda recepcionar los datos de ingreso y opere de la manera que fue presentada en este trabajo. Con lo anterior, se rescatan las ventas realizadas por todos los vendedores y se generan las facturas a todos los clientes visitados, usando el módulo de facturación de *Random*.

Los **costos** asociados a ambos ambientes operativos presentados se evidencian en las horas hombres utilizadas para la creación del sistema (instalación y poblamiento). Para el caso de la base de datos relacional, el programa *MySQL* WorkBench es gratuito al igual que el sistema operativo donde opera, por lo que costos en licencias no existen. Ahora bien, estimando una hora hombre en el mercado de ingenieros en informática, el costo asociado a creación del sistema bordea los 480.000 pesos chilenos. Sin embargo, para Cassandra, el costo asociado a horas hombre utilizadas es más elevado dado que se debe instruir al usuario en su nuevo sistema operativo, por lo que el monto anterior sube a 1.200.000 pesos chilenos. Al igual que el caso relacional, Linux y Cassandra son *open source* por lo que no hay valor extra en el uso de sus capacidades.

La propuesta de este ambiente operacional afecta directamente a los costos del ERP *Random* dado que el licenciamiento, para el uso de los módulos de gestión, ya no serían necesarios y, más aún, tampoco es necesario pagar la membresía anual del uso de servicio (donde se incluye el guardado de versiones) . Solo se pagaría el uso del servicio de respaldo del servidor que aborda los 47.000 pesos chilenos al mes.

Finalmente, la **adopción de la arquitectura** presentada en la base de datos *Cassandra*, hace que las labores de gestión realizadas en la empresa en estudio sean más fructíferas en el tiempo. Considerando que recientemente el volumen de datos aumentó debido a que se abrieron nuevos mercados en la empresa, el cuello de botella que se produce en *MySQL* es aún peor ya que todos estos datos se almacenan en el mismo servidor. Cassandra, y sus servicios asociados, no presentarían ningún problema dado que la conexión ODBC al servidor donde llegan los datos desde los móviles, no se afecta por el incremento. El problema real radica en la forma en que opera la base de datos incrustada del ERP *Random* y su crecimiento horizontal. Además, Cassandra logra entregar en tiempos muy pequeños toda la información que necesita el gerente de operaciones. El “nuevo” volumen de datos no es menor ya que los nuevos mercados a los que la empresa en estudio se expandió son: artículos escolares, bisutería, costurería, artículos para la limpieza del hogar, galletas y

chocolates. Considerando que en esta memoria se tiene un 65 % de los datos que se maneja actualmente en la empresa, el problema que se presenta seguirá creciendo a tal punto que será necesario aumentar la capacidad de los servidores actuales para reducir los problemas de tiempo asociados a los cuellos de botella de la base de datos relacional.

4.5. Herramienta de análisis y presentación

Tal como se mencionó en el capítulo anterior, se utiliza *MS Power BI* como herramienta de BI para el proceso de análisis de la información. Para conectar la herramienta con el servidor, se utiliza la opción disponible en la herramienta llamada "Conexión por ODBC" para así evitar realizar actualizaciones manuales de la información mediante archivos *.csv*.

Con las conexiones realizadas sin problemas, siguiendo el asistente de instalación, basta con utilizar las características que tiene *MS Power BI* para formular paneles de información ad-hoc a los diseños propuestos en el capítulo anterior. Dichos paneles pueden observarse en las figuras 4.4 y 4.6.

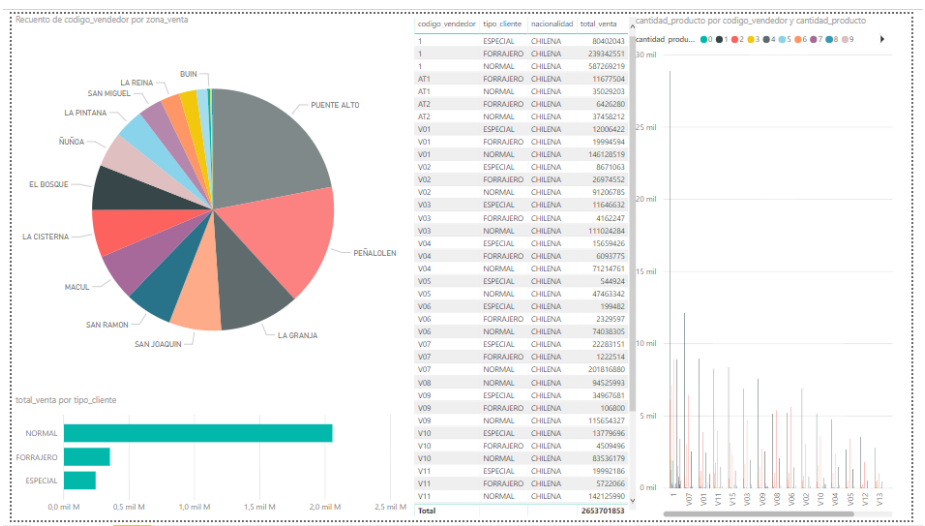


Figura 4.4: Panel de información creado a partir del diseño en figura 3.4. Se destaca que este panel es *clickeable* y va mostrando cómo cambia la información en tiempo real.
Fuente: Elaboración Propia.

Como se indica en la glosa de la figura 4.4, este panel puede mostrar información adicional o centralizada si se hace "click" en algún dato que está expuestos en él. La información

se actualiza en tiempo real con los datos obtenidos a ese momento. En la figura 4.5, a modo de ejemplo, se presenta información relevante de la comuna de Puente Alto.

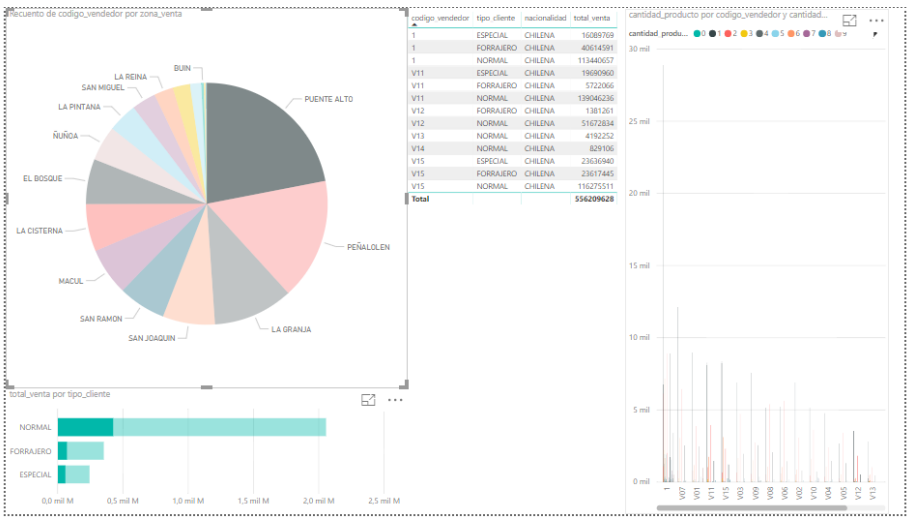


Figura 4.5: Información centralizada de la comuna de Puente Alto.
Fuente: Elaboración Propia.

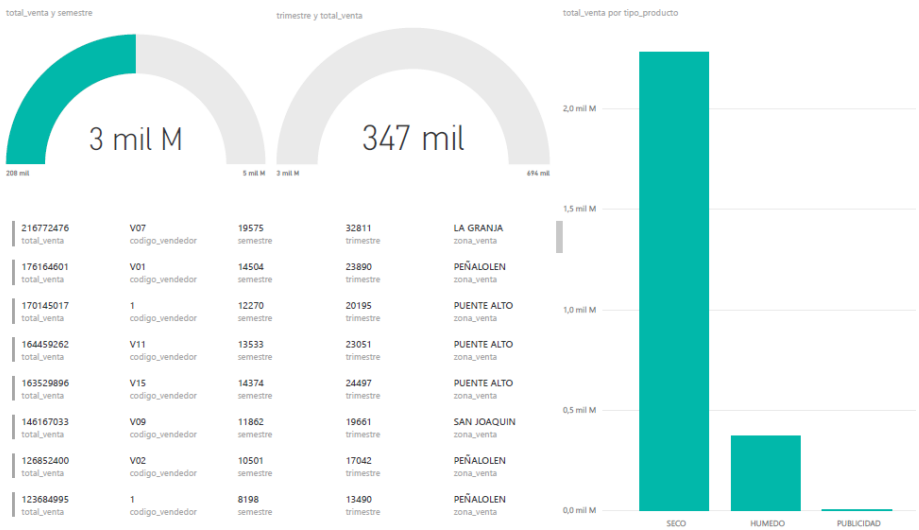


Figura 4.6: Panel de información creado a partir del diseño en figura 3.5.
Fuente: Elaboración Propia.

Por lo tanto, *MS Power BI* es totalmente operacional y no incurre en gasto alguno su adquisición. Cambiar el tipo de licenciamiento de esta herramienta BI no está previsto para el año 2019 (en su nueva versión), por lo que los datos presentados aquí tienen 3 años, como mínimo, de fidelidad dado que ese es el tiempo que demora esta herramienta en

cambiar de versión.

Una de los grandes potenciales que tiene esta herramienta es que permite el acceso de múltiples fuentes de información. Para agregar más valor aún a esta herramienta, se incorpora, a los datos que ya se reflejan actualmente, información que se obtiene del proveedor de *GPS* que utiliza el área de distribución. El fin de esto es observar la posición en la cual se encuentran los camiones de reparto. Esta funcionalidad se propone como extra para el nuevo ambiente operacional ya que, al ser una forma novedosa de ir controlando a los camiones, se debe adquirir la lógica de negocio necesaria para acoplar esta visión.

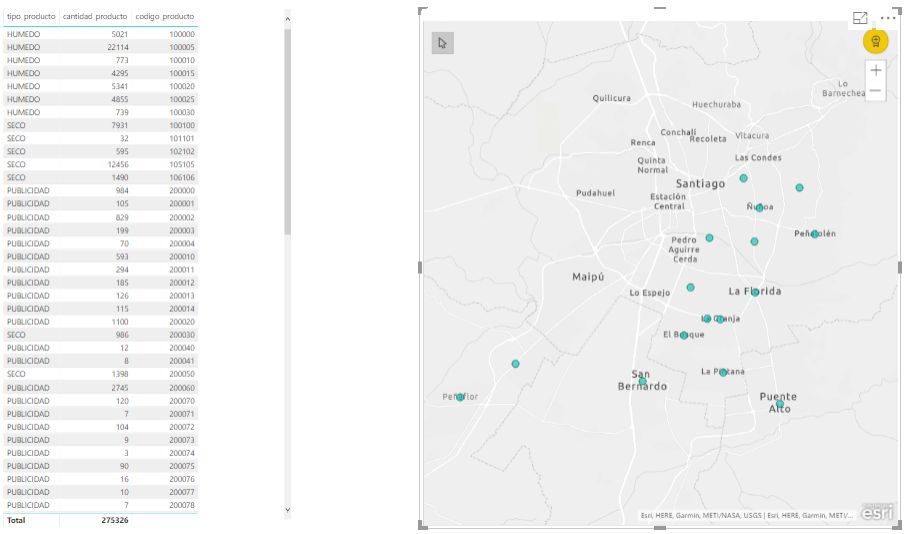


Figura 4.7: Panel con la posición de los camiones de distribución y productos que llevan. Este panel apoya al diseño propuesto en la figura 3.4.

Fuente: Elaboración Propia.

Al considerar **costos** en esta implementación, se debe tener presente que es necesaria una capacitación ad-hoc a la creación de nuevos paneles si fueran requeridos. Ahora bien, el uso de la herramienta BI es totalmente gratuito y no se pierde nada con su versión de pago, ya que solo se agrega la opción de tener los datos de manera remota en la nube y como no se implementa una solución de ese estilo, la pérdida es nula.

Se realiza la evaluación de las heurísticas de Nielsen en la herramienta y la confección de los paneles BI. Se busca que la usabilidad de la capa de presentación de los datos sea fácil. Para cada heurística seleccionada, se evalúa su diseño, donde se tiene:

- Visibilidad: se evidencia que basta con realizar “clicks” en la información de los

paneles para evidenciar el estado de ellos. La información se va desplegando en tiempo real para cada panel dado que las conexiones están realizadas directamente a la fuente de información. Por otra parte, utilizando la heurística en la herramienta, el usuario puede observar, gracias a los paneles que están en el costado derecho, en que lugar se encuentra.

- Control y libertad: se evidencia que, dentro de los paneles creados, el usuario puede seleccionar la información que desea desplegar haciendo uso de un formato "soltar y arrastrar". También, como es una herramienta de Microsoft, está habilitada la opción "deshacer" que se encuentra visible en su formato de flecha oblicua en la parte superior izquierda de la pantalla.

- Consistencia y estándares: se evidencia que la barra lateral no cambia en formato (tamaño, texto, imágenes, etc) al modificar los paneles o mover la vista al abrir la barra de tareas. Se tiene una alta consistencia en botones. Ahora bien, la creación de paneles es una tarea en la cual pueden existir falencias si no se tiene cuidado en la etapa de creación.

- Flexibilidad: se observa que, en general, esta heurística no se cumple dado que la realización de paneles de información no permite que estos sean duplicados. Además, los paneles deben ser confeccionados a medida de las necesidades del usuario. Sí existen atajos en el sistema para la creación información en los paneles.

- Ayuda y documentación: se observa que, al ser una herramienta de Microsoft, la utilización de ayuda se basa en resolución a preguntas frecuentes, por lo que hay que buscar un tema semejante a algún problema para saber cómo actuar. Documentación referente a la herramienta, puede encontrarse en la página del proveedor sin costo alguno.

Entonces, se puede indicar que el objetivo de la herramienta es cumplido a cabalidad. La evaluación de las heurísticas muestra que, en gran medida, la herramienta no presenta dificultades en el uso. Ahora bien, si asociamos esta evaluación heurística a objetivos del negocio, se observa que estos se pueden lograr totalmente ya que las actividades que pueden realizarse dentro de la herramienta siguen el mismo hilo conductor del negocio debido a que los paneles son creados con información propia de la empresa. Por otra parte, se

busca que la herramienta apoye la toma de decisiones de manera rápida y eficiente, y bajo el ambiente propuesto, es totalmente aceptable.

4.6. Impacto de la propuesta

Se realiza una pequeña evaluación del impacto que tiene el uso de la propuesta de ambiente operacional. Para lograr determinar en qué medida se produce este impacto (positiva o negativamente), se abordan los siguientes puntos de interés con los usuarios finales:

- Relación de causalidad: conocimiento de los cambios que se producen tras el uso de las nuevas herramientas.
- Intervención del ambiente operacional propuesto: previsto o no en relación a las necesidades de la empresa.
- Tipos de impacto afectados: en relación a la sociedad, a la empresa y a la persona.

Tras pequeñas pruebas de uso realizadas a 3 personas en ambas bases de datos y catalogadas como los principales beneficiados de este nuevo ambiente operacional, se observa que para la relación de causalidad entienden el nuevo uso de las herramientas (base de datos y *MS Power BI*). Los cambios comentados hacen mayor referencia a nuevas interfaces más que a la productividad en la operación de la empresa. Con esto, se logra impactar fuertemente en cambios visuales, sin afectar o agregar dificultad al trabajo cotidiano realizado.

Con respecto a la intervención del ambiente operacional propuesto, los comentarios hacen referencia a que se logra mayor velocidad en el trabajo para la generación de informes y reportes dado el uso de la herramienta BI. Se indica que, el nuevo ambiente operacional, cumple con atender las necesidades planteadas y, por ende, era un resultado esperado.

Los tipos de impacto asociados a este nuevo ambiente operacional, mencionados por los usuarios, se detallan a continuación:

- Social: el trabajo que se realiza es más veloz, por lo que los tiempos de espera son decentes y no exagerados. Con esta nueva tecnología se espera elevar la competitividad en el mercado laboral de la distribuidora.

- Empresa: la productividad aumenta y el clima laboral mejora dado que se tiene más tiempo para interactuar con los compañeros de trabajo o atender otras responsabilidades, sin tener preocupaciones por demoras en los requerimientos que realizan los jefes. Se indica que estos cambios son muy positivos. La innovación en la forma de entregar información es muy llamativa.
- Personal: no se conocían herramientas BI, por lo que el desafío de aprender a usar nuevas tecnologías genera un aumento en el desarrollo profesional ya que hay nuevas competencias adquiridas. Se espera que las condiciones laborales mejoren y se cumplan los tiempos legales de trabajo sin tener que incurrir en horas extras por demoras en la información.

Finalmente, el impacto de esta propuesta de ambiente operacional puede ser catalogado como exitoso debido a los comentarios positivos tras el uso de, principalmente, la herramienta BI. Los puntos de interés planteados se resuelven completamente con los comentarios emitidos por los usuarios.

Conclusiones

Con el desarrollo de esta investigación, es posible observar cómo los ambientes operacionales de una empresa no pueden estancarse. Es necesario estar atento a los volúmenes de información que se van manejando y no simplemente dejar que un proveedor de aplicaciones ERP o CRM tome el mando del sistema de trabajo de manera exclusiva. También, gracias a la investigación, se observa el potencial que tiene una herramienta de BI en la gestión de información de una empresa.

En el experimento de diseñar un nuevo ambiente operacional, se evidencia por los resultados entregados que los tiempos de respuesta son mucho menores a los que actualmente maneja la empresa en estudio. Esto se debe a que el ambiente operacional actual no permite un esquema flexible en la información, provocando que la generación de informes anuales no sea posible dado que el sistema entrega toda la información sin ser parametrizada con anterioridad. Existen muchos módulos inactivos en el ERP *Random* que, además, están siendo pagados. Esta inactividad de servicios produce información nula en gran parte de las columnas que se generan en los archivos de *Excel*. Imaginar que de las 81 columnas, solo son necesarias las postuladas en este nuevo ambiente, deja mucho que pensar en cómo las empresas guían al buen uso de sus herramientas. Si hablamos de costos asociados a este nuevo ambiente operacional, son totalmente disminuidos dado que solo se necesita el módulo de facturación y un par de capacitaciones.

El nuevo ambiente operacional que se propone viene de la mano con una gestión del cambio que será necesaria, ya que estas nuevas tecnologías no tienen nada en común con lo que se usa actualmente. Por otra parte, se destaca que el paradigma del trabajo no se modifica en su núcleo, sino que solamente en su actuar.

Tras la evaluación del ambiente operacional propuesto, los tiempos para respuesta de

consultas, almacenamiento de datos y adquisición de información se reducen en un 97 %. Por otra parte, los costos se reducen en gran medida, ya que se estaría pagando una octava parte del servicio que actualmente se tiene al año.

Un apartado importante en esta investigación es el uso de la inteligencia de negocios. Una herramienta, gratuita, como *MS Power BI* aumenta considerablemente el análisis de los datos. Las decisiones ya no se basan en un recuerdo de lo que se hizo, sino que es posible revisar un historial o comportamiento que tienen los clientes. La información es entregada a tiempo real, por lo que la toma de decisiones será mucho más rápida y eficiente. *MS Power BI* tiene la capacidad de generar mucha información de manera rápida y fácil para ser una herramienta de licenciamiento gratuito, por lo que consideraciones para un alineamiento estratégico o el cumplimiento de objetivos de negocio, será más eficaz. Otras herramientas BI lográn también generar este tipo de información pero no con la misma facilidad. De manera personal se utilizó Qlikview para una prueba y fue un poco más complicado utilizar el *software* de manera operativa.

Con todo lo anterior, se espera que la empresa en estudio se beneficie con el uso de este nuevo ambiente operacional propuesto. Se ha generado una pauta completa del funcionamiento de las bases de datos involucradas y cómo se afecta, positivamente, la productividad.

Al hablar sobre la aplicación y adopción de la metodología, se evidencian ciertas falencias presentes en *SQL*. Teniendo en cuenta que el *hardware* utilizado es el mismo para ambas bases de datos, a excepción del sistema operativo, la arquitectura propuesta es más efectiva en la base de datos *Cassandra*. Las dos réplicas realizadas en este trabajo son capaces de tener orígenes "simples" de datos, como lo es un archivo de tipo *Excel*; luego, el proceso ETL necesario para crear una inserción efectiva a la base de datos actúa de mejor manera, con respecto al tiempo, en la base de datos columnar. Esto se logra dado que la información está desnormalizada y puede ser utilizada como más se estime conveniente. Por otro lado, basado en las opiniones de los usuarios, la rapidez de un sistema es algo que se valora cuando no es la única tarea a realizar; más aún, se valora que sea fácil de usar. Con esto, al mantener el mismo formato de trabajo pero con una nueva tecnología, se evidencia que los usuarios no muestran desagrado por utilizar herramientas desconocidas por ellos, como es el caso de *MS Power BI*. Además, utilizar heurísticas de usabilidad para el diseño

es una práctica que favorece los resultados y el impacto esperado. Los comentarios obtenidos son, en gran medida, positivos. Por último, como la herramienta BI y la base de datos columnar permiten conexiones del tipo ODBC, la información que provee la herramienta BI es en tiempo real y, con un proceso ETL periódico, esta información es confiable y sin anomalías.

Se recomienda como extensión a esta investigación:

- Creación de herramienta que utilice una base de datos *NoSQL* y que integre el análisis de datos en ella: se podría facilitar una herramienta capaz de actuar a la par con un CRM o ERP, que fuera *open source*, y permitiera acoplar o desacoplar módulos de trabajo. Además, que tolere almacenamiento para datos y utilización de estos mismo para la toma de decisiones bajo una base de datos *NoSQL* operable en una sola interfaz. En Chile se ve una falencia de herramientas que utilicen nuevas tecnologías, como bases de datos *NoSQL*, y aprovechen al máximo los recursos de hardware de equipos dado que no se incurre tiempo productivo en gestión del cambio.
- Acoplar un sistema que afecte al bodegaje apenas un producto sea retirado de un camión de distribución: integrar un sistema móvil a la sección de distribución de una empresa (camiones de transporte) que, en tiempo real, lleve la carga del camión y retire del stock de la bodega el producto inmediatamente cuando este es retirado del camión. Esto debe pasar por validación del conductor del camión. Con esto, tanto el camión como los productos, pueden ser rastreados mediante una aplicación y se lleva el stock de productos en tiempo real sin necesidad de cuadraturas.
- Aplicación móvil de fácil uso que permita a los clientes de una distribuidora solicitar, vía remota, productos para sus propios negocios: integrar un sistema móvil o programa capaz de interactuar directamente con un módulo de ventas para generar compras remotas (sin vendedor físico o teléfono) y proporcionar, vía correo, la factura de los productos solicitados asegurando que dichos productos lleguen el día solicitado. Más bien, un asistente móvil capaz de interactuar con el módulo de ventas de una empresa y generar una compra segura con el uso de ciertas credenciales.

Finalmente, en relación a lo aprendido en la carrera de Ingeniería Civil Informática, se destacan las herramientas entregadas para enfrentar y solucionar problemas. Información relevante para la realización de esta investigación viene de las siguientes asignaturas:

- Bases de datos y Bases de datos Avanzada. Estas asignaturas entregaron los conocimientos fundamentales y claves para entender el núcleo de esta investigación (funcionamiento, operaciones y modos de uso de los sistemas involucrados).
- Bases tecnológicas para la inteligencia de negocios. Esta asignatura enseña a cómo entender y usar los datos para beneficiar negocios.
- Arquitectura empresarial. Asignatura que entrega los conocimientos necesarios para entender la estructura interna de un negocio.

Las asignaturas que desenvolvían un carácter matemático en la carrera (computación científica, estadística computacional, etc), permiten abstraerse para tener un pensamiento más lógico y centrado en la búsqueda de soluciones.

Anexos

A.1. Código de creación de la base de datos *MySQL*

· El script de creación de la base de datos, es el siguiente:

```
1  -- Inicio creacion BD
2  -- nombre_ambiente: DBmemoriaV1
3
4  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
5  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
6  FOREIGN_KEY_CHECKS=0;
7
8  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE=
9   'TRADITIONAL, ALLOW_INVALID_DATES';
10
11 CREATE SCHEMA IF NOT EXISTS `DBmemoriav1` DEFAULT CHARACTER
12 SET utf8 COLLATE utf8_general_ci ;
13 USE `DBmemoriav1` ;
14
15 -- -----------------------------------------------------
16 -- Table `DBmemoriav1`.`cliente`
17 -- -----------------------------------------------------
18 CREATE  TABLE IF NOT EXISTS `DBmemoriav1`.`cliente` (
19    `RazonSocial` VARCHAR(200) NOT NULL ,
20    `NombreClien` VARCHAR(45) NOT NULL ,
21    `Direccion` VARCHAR(200) NOT NULL ,
22    `Zona` VARCHAR(45) NOT NULL ,
23    `Tipo` VARCHAR(45) NOT NULL ,
```

```
24    `Descuento` VARCHAR(45) NOT NULL ,
25    PRIMARY KEY (`RazonSocial`) )
26  ENGINE = InnoDB;
27  -- -----------------------------------------------------
28  -- Table `DBmemoriav1`.`vendedor`
29  -- -----------------------------------------------------
30  CREATE TABLE IF NOT EXISTS `DBmemoriav1`.`vendedor` (
31    `CodigoVendedor` VARCHAR(10) NOT NULL ,
32    `Sexo` VARCHAR(45) NOT NULL ,
33    `Edad` INT NOT NULL ,
34    `Nacionalidad` VARCHAR(45) NOT NULL ,
35    PRIMARY KEY (`CodigoVendedor`) )
36  ENGINE = InnoDB;
37  -- -----------------------------------------------------
38  -- Table `DBmemoriav1`.`tiempo`
39  -- -----------------------------------------------------
40  CREATE TABLE IF NOT EXISTS `DBmemoriav1`.`tiempo` (
41    `Fecha` DATE NOT NULL ,
42    `Trimistre` INT NOT NULL ,
43    `Semestre` INT NOT NULL ,
44    PRIMARY KEY (`Fecha`) )
45  ENGINE = InnoDB;
46  -- -----------------------------------------------------
47  -- Table `DBmemoriav1`.`producto`
48  -- -----------------------------------------------------
49  CREATE TABLE IF NOT EXISTS `DBmemoriav1`.`producto` (
50    `CodigoProducto` INT NOT NULL ,
51    `NombreProd` VARCHAR(200) NOT NULL ,
52    `TipoProducto` VARCHAR(45) NOT NULL ,
53    `UnidadVenta` VARCHAR(45) NOT NULL ,
54    PRIMARY KEY (`CodigoProducto`) )
55  ENGINE = InnoDB;
56  -- -----------------------------------------------------
57  -- Table `DBmemoriav1`.`bodega`
58  -- -----------------------------------------------------
59  CREATE TABLE IF NOT EXISTS `DBmemoriav1`.`bodega` (
```

```
60    `CodigoBodega` INT NOT NULL ,
61    `NombreBod` VARCHAR(45) NOT NULL ,
62    `Direccion` VARCHAR(45) NOT NULL ,
63    `Comuna` VARCHAR(45) NOT NULL ,
64    PRIMARY KEY (`CodigoBodega`) )
65  ENGINE = InnoDB;
66  -- -----------------------------------------------------
67  -- Table `DBmemoriav1`.`venta`
68  -- -----------------------------------------------------
69  CREATE TABLE IF NOT EXISTS `DBmemoriav1`.`venta` (
70    `CodigoVenta` INT NOT NULL ,
71    `cliente_RazonSocial` VARCHAR(200) NOT NULL ,
72    `vendedor_CodigoVendedor` VARCHAR(10) NOT NULL ,
73    `tiempo_Fecha` DATE NOT NULL ,
74    `producto_CodigoProducto` INT NOT NULL ,
75    `bodega_CodigoBodega` INT NOT NULL ,
76    `PrecioProd` INT NOT NULL ,
77    `TotalVenta` INT NOT NULL ,
78    `CantidadProd` INT NOT NULL ,
79    PRIMARY KEY (`CodigoVenta`) ,
80    INDEX `fk_venta_cliente_idx`
81    (`cliente_RazonSocial` ASC) ,
82    INDEX `fk_venta_vendedor1_idx`
83    (`vendedor_CodigoVendedor` ASC) ,
84    INDEX `fk_venta_tiempo1_idx`
85    (`tiempo_Fecha` ASC) ,
86    INDEX `fk_venta_producto1_idx`
87    (`producto_CodigoProducto` ASC) ,
88    INDEX `fk_venta_bodega1_idx`
89    (`bodega_CodigoBodega` ASC) ,
90    CONSTRAINT `fk_venta_cliente`
91      FOREIGN KEY (`cliente_RazonSocial`)
92      REFERENCES `DBmemoriav1`.`cliente` (`RazonSocial`)
93      ON DELETE NO ACTION
94      ON UPDATE NO ACTION,
95    CONSTRAINT `fk_venta_vendedor1`
```

```
96      FOREIGN KEY (`vendedor_CodigoVendedor`)
97      REFERENCES `DBmemoriav1`.`vendedor` (`CodigoVendedor`)
98      ON DELETE NO ACTION
99      ON UPDATE NO ACTION,
100   CONSTRAINT `fk_venta_tiempo1`
101     FOREIGN KEY (`tiempo_Fecha`)
102     REFERENCES `DBmemoriav1`.`tiempo` (`Fecha`)
103     ON DELETE NO ACTION
104     ON UPDATE NO ACTION,
105   CONSTRAINT `fk_venta_producto1`
106     FOREIGN KEY (`producto_CodigoProducto`)
107     REFERENCES `DBmemoriav1`.`producto` (`CodigoProducto`)
108     ON DELETE NO ACTION
109     ON UPDATE NO ACTION,
110   CONSTRAINT `fk_venta_bodega1`
111     FOREIGN KEY (`bodega_CodigoBodega`)
112     REFERENCES `DBmemoriav1`.`bodega` (`CodigoBodega`)
113     ON DELETE NO ACTION
114     ON UPDATE NO ACTION)
115 ENGINE = InnoDB;
116
117 USE `DBmemoriav1`;
118
119 SET SQL_MODE=@OLD_SQL_MODE;
120 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
121 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
122 -- fin script
```

A.2. Código de creación de la base de datos *Cassandra*

· El script de creación de base de datos, es el siguiente:

```
1
2 DROP KEYSPACE dbmemov2;
3
```

```
4  CREATE KEYSPACE dbmemov2 WITH replication =
5  {'class': 'SimpleStrategy','replication_factor' :1};
6
7  USE dbmemov2;
8
9  CREATE TABLE ventas_totales
10 (id int,razon_social text,codigo_vendedor text,
11 fecha date,codigo_producto text,codigo_bodega int,
12 precio_producto int,total_venta int,
13 cantidad_producto int,sexo text,edad int,
14 nacionalidad text,nombre_bodega text,
15 direccion text,comuna_bodega text,
16 trimestre int,semestre int,
17 nombre_producto text,tipo_producto text,
18 unidad_venta text,zona_venta text,
19 tipo_cliente text,descuento text,
20 PRIMARY KEY (id));
21 -- PRIMARY KEY (razon_social)) %para test 2
22 -- PRIMARY KEY (codigo_vendedor)) %para test 3
23 -- fin script
```

A.3. Consulta COPY para exportar datos a un archivo

```
1  COPY ventas_totales
2  (id,razon_social,codigo_vendedor,fecha,
3  codigo_producto,codigo_bodega,precio_producto,
4  total_venta,cantidad_producto,sexo,
5  edad,nacionalidad,nombre_bodega,direccion,
6  comuna_bodega,trimestre,semestre,
7  nombre_producto,tipo_producto,
8  unidad_venta,zona_venta,
9  tipo_cliente,descuento)
10 TO '/carpeta_contenedora/nombre_archivo.csv'
11 WITH HEADER = TRUE and DELIMITER = ';';
```

Bibliografía

[AmazonWS, 2017] AmazonWS (2017). ¿qué es una base de datos columnar? url https://aws.amazon.com/es/nosql/columnar/. Accedido 19-11-2018.

[Anderson *et al.*, 2010] Anderson, J. C., Lehnardt, J., y Slater, N. (2010). *CouchDB: The Definitive Guide: Time to Relax*. .ºReilly Media, Inc.".

[Boncz *et al.*, 1999] Boncz, Peter A and Manegold, Stefan and Kersten, Martin L and others (1999). Database architecture optimized for the new bottleneck: Memory access. En *VLDB*, volumen 99, pp. 54–65.

[Brunozzi, 2012] Brunozzi, S. (2012). Big data and nosql with amazon dynamodb. En *Proceedings of the 2012 workshop on Management of big data systems*, pp. 41–42. ACM.

[Calzada y Abreu, 2009] Calzada, L. y Abreu, J. L. (2009). El impacto de las herramientas de inteligencia de negocios en la toma de decisiones de los ejecutivos. *Revista Daena (International Journal of Good Conscience)*, 4(2).

[Carlson, 2013] Carlson, J. L. (2013). *Redis in action*. Manning Publications Co.

[Carpenter y Hewitt, 2016] Carpenter, J. y Hewitt, E. (2016). *Cassandra: The Definitive Guide: Distributed Data at Web Scale*. .ºReilly Media, Inc.".

[Chodorow, 2013] Chodorow, K. (2013). *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. .ºReilly Media, Inc.".

[Chu, 2008] Chu, S. (2008). Memcachedb: The complete guide.

[Codd, 1970] Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.

[Conrick, 2006] Conrick, M. (2006). *Health Informatics*. Cengage Learning Australia.

[DataIQ, 2018] DataIQ (2018). ¡qlik sense es líder del "cuadrante mágico" de gartner por octavo año consecutivo! url https://bit.ly/2HusQCG. Accedido 28-11-2018.

[Gartner, 2013] Gartner, I. (2013). Gartner it glossary. *Technology Research*.

[George, 2011] George, L. (2011). *HBase: the definitive guide: random access to your planet-size data*. .º'Reilly Media, Inc.".

[Haigh, 2011] Haigh, T. (2011). Charles w. bachman: Database software pioneer. *IEEE Annals of the History of Computing*, 33(4):70–80.

[Huang y Dong, 2013] Huang, H. y Dong, Z. (2013). Research on architecture and query performance based on distributed graph database neo4j. En *Consumer Electronics, Communications and Networks (CECNet), 2013 3rd International Conference on*, pp. 533–536. IEEE.

[Innovación-Necesaria, 2018] Innovación-Necesaria (2018). Ms power bi lidera el cuadrante mágico de gartner 2018. url https://bit.ly/2zOOPTE. Accedido 28-11-2018.

[Karlsson *et al.*, 2001] Karlsson, J. S., Lal, A., Leung, C., y Pham, T. (2001). Ibm db2 everyplace: A small footprint relational database system. En *Data Engineering, 2001. Proceedings. 17th International Conference on*, pp. 230–232. IEEE.

[Katsov, 2012] Katsov, I. (2012). Nosql data modeling techniques. *Highly Scalable Blog*.

[Kumar Kaliyar, 2015] Kumar Kaliyar, R. (2015). Graph databases: A survey. En *Computing, Communication & Automation (ICCCA), 2015 International Conference on*, pp. 785–790. IEEE.

[Mateljan *et al.*, 2010] Mateljan, V., Cisic, D., y Ogrizovic, D. (2010). Cloud database-as-a-service (daas)-roi. En *MIPRO, 2010 proceedings of the 33rd International convention*, pp. 1185–1188. IEEE.

[Michaels *et al.*, 1976] Michaels, A. S., Mittman, B., y Carlson, C. R. (1976). A comparison of the relational and codasyl approaches to data-base management. *ACM Computing Surveys (CSUR)*, 8(1):125–151.

[Microsoft, 2017] Microsoft (2017). Especificaciones y limites de excel. url https://bit.ly/2yiZ6sX. Accedido 23-11-2018.

[Nielsen, 1995] Nielsen, J. (1995). 10 usability heuristics for user interface design. *Nielsen Norman Group*, 1(1).

[Popescu, 2010] Popescu, A. (2010). Nosql at codemash—an interesting nosql categorization.

[Ranjan, 2014] Ranjan, R. (2014). Streaming big data processing in datacenter clouds. *IEEE Cloud Computing*, 1(1):78–83.

[Vitt *et al.*, 2002] Vitt, Elizabeth and Luckevich, Michael and Misner, Stacia and Rosas Gallardo, Oscar and others (2002). *Business intelligence: técnicas de análisis para la toma de decisiones estratégicas*. McGraw-Hill,.