



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**CREACIÓN E IMPLANTACIÓN DE UN SISTEMA DE PUNTO DE VENTA EN LA  
NUBE COMO AYUDA AL DESARROLLO DE LAS PYMES EN GUATEMALA**

**Joel Estuardo Juárez Back**

Asesorado por el Ing. Helmerson David Oliva

Guatemala, agosto de 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**CREACIÓN E IMPLANTACIÓN DE UN SISTEMA DE PUNTO DE VENTA EN LA  
NUBE COMO AYUDA AL DESARROLLO DE LAS PYMES EN GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**JOEL ESTUARDO JUÁREZ BACK**

ASESORADO POR EL ING. HELMERSON DAVID OLIVA

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, AGOSTO DE 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Ing. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**


DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Ricardo Morales Santos
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. Óscar Alejandro Paz Campos
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **CREACIÓN E IMPLANTACIÓN DE UN SISTEMA DE PUNTO DE VENTA EN LA NUBE COMO AYUDA AL DESARROLLO DE LAS PYMES EN GUATEMALA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 28 de marzo de 2012.

  
**Joel Estuardo Juárez Back**

Guatemala, 03 de marzo de 2014

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Ing. Carlos Azurdia

Estimado Ingeniero Azurdia

El motivo de la presente carta es para hacer constar que, bajo mi asesoría profesional, el alumno Joel Estuardo Juárez Back, que se identifica con carné universitario 200714182, realizó el desarrollo de su proyecto de tesis, habiendo cumplido con el trabajo requerido para finalizar con el mismo.

El proyecto que realizó el alumno es un sistema de punto de venta como servicio en la nube, esto orientado a aportar para el desarrollo de las pequeñas y medianas empresas en Guatemala.

Sin más que hacer constar, se despide atentamente,

Helmerson David Oliva

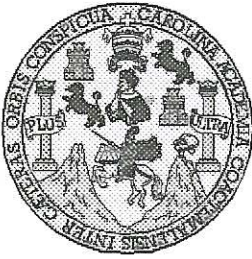


*Helmerson David Oliva*  
Ing. en Ciencias y Sistemas  
Colegiado 9490

---

Ing. Helmerson David Oliva

No. Colegiado 9490



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 11 de Marzo de 2014

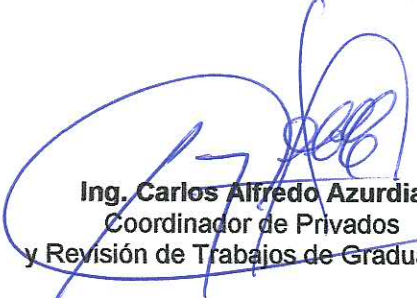
Ingeniero  
**Marlon Antonio Pérez Turk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **JOEL ESTUARDO JUÁREZ BACK** con carné **2007-14182**, titulado: **“CREACIÓN E IMPLANTACIÓN DE UN SISTEMA DE PUNTO DE VENTA EN LA NUBE COMO AYUDA AL DESARROLLO DE LAS PYMES EN GUATEMALA”**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación



E  
S  
C  
U  
E  
L  
A  
  
D  
E  
  
C  
I  
E  
N  
C  
I  
A  
S  
  
Y  
  
S  
I  
S  
T  
E  
M  
A  
S

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“CREACIÓN E IMPLANTACIÓN DE UN SISTEMA DE PUNTO DE VENTA EN LA NUBE COMO AYUDA AL DESARROLLO DE LA PYMES EN GUATEMALA”**, realizado por el estudiante JOEL ESTUARDO JUÁREZ BACK, aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑAD A TODOS”**

*Ing. **Martín Antonio Pérez Türk***

*Director, Escuela de Ingeniería en Ciencias y Sistemas*

*Guatemala, 18 de agosto 2014*





El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **CREACIÓN E IMPLEMENTACIÓN DE UN SISTEMA DE PUNTO DE VENTA EN LA NUBE COMO AYUDA AL DESARROLLO DE LAS PYMES EN GUATEMALA**, presentado por el estudiante universitario: **Joel Estuardo Juárez Back** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Olympo Paiz Recinos  
Decano



Guatemala, agosto de 2014

/cc



## **AGRADECMIENTOS A:**

- Dios** Por su infinita misericordia para conmigo. Solamente por su soberana voluntad y sus bendiciones en mi vida integral es que puedo celebrar este éxito. Para Él sea la honra y la gloria por los siglos de los siglos.
- Mis padres** Luis Armenio Juárez Vega y Lesbia Amparo Back Charuc, por darme la vida, todo su amor y por sus esfuerzos invaluableles al brindarme todas las facilidades para que pudiera concluir esta etapa de mi vida. Ellos son ejemplo de vida para mí.
- Mis hermanos** Luis David y Daniel Alberto, por ser mis mejores amigos y por su apoyo incondicional en cada aspecto de mi vida.
- Mi prometida y futura esposa** Ana Luisa Solórzano Reyes, por su apoyo incondicional y su motivación constante.
- Mi familia** Por todo el apoyo recibido, muchas veces unas simples palabras servían de motor para continuar.

**Mis colegas y amigos**

Por todas las experiencias y toda la convivencia que hizo de esta etapa de estudios algo único e inolvidable.

**Universidad de San Carlos de Guatemala**

Mi casa de estudios, por todo el conocimiento adquirido dentro y fuera de las aulas de clase. Por brindarme la oportunidad de completar mis estudios universitarios.

**Mi país**

Tierra que me vio nacer y me da la oportunidad de estudiar y ahora desenvolverme como profesional.

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO .....	IX
RESUMEN.....	XIII
OBJETIVOS.....	XV
INTRODUCCIÓN .....	XVII
1. MARCO TEÓRICO INICIAL. CONTEXTO DE LAS PYMES EN GUATEMALA .....	1
1.1. ¿Qué es un punto de venta? .....	2
1.2. Contexto de las pymes en Guatemala.....	3
1.2.1. Adopción de las Tic en las pymes en Guatemala .....	4
1.3. Punto de venta como una etapa en la cadena de valor de las pymes .....	9
1.4. ¿Qué es un servicio en la nube? .....	12
1.4.1. Ventajas de computación en la nube.....	14
1.4.2. Desventajas de computación en la nube .....	15
2. ENFOQUE, ESTRATEGIA Y REQUERIMIENTOS GENERALES DEL PRODUCTO.....	17
2.1. ¿Hacia quiénes está dirigido? .....	18
2.2. Alcance del producto .....	19
2.3. Requerimientos generales del producto .....	20
2.4. Estrategia de distribución del punto de venta en la nube .....	21

2.4.1.	Afiliación o suscripción .....	23
2.4.2.	Ventajas y desventajas de la afiliación.....	24
2.5.	Tecnologías para el desarrollo del producto .....	26
2.5.1.	Infraestructura .....	27
2.5.2.	Persistencia de datos .....	28
2.5.3.	Interfaz de usuario.....	30
2.5.4.	Base de datos .....	32
2.5.5.	Servidor de servlets.....	33
2.5.6.	Construcción del proyecto .....	34
2.6.	Procesos y actividades.....	35
2.7.	Esquema general del producto .....	41
3.	IMPLEMENTACIÓN DE UN PUNTO DE VENTA EN LA NUBE .....	47
3.1.	Análisis del producto .....	48
3.1.1.	Toma de requerimientos.....	48
3.1.2.	Casos de uso .....	54
3.1.3.	Dinámica del sistema .....	66
3.2.	Infraestructura y arquitectura del software .....	80
3.3.	Diseño del producto .....	85
3.3.1.	Componentes .....	86
3.3.2.	Clases y modelo de datos .....	89
3.4.	Codificación.....	95
3.5.	Pruebas.....	107

4.	IMPLANTACIÓN Y ANÁLISIS DE RESULTADOS.....	113
4.1.	Requerimientos de hardware e infraestructura para la implantación .....	113
4.2.	Implantación del producto.....	115
4.3.	Análisis del rendimiento del POS como una etapa en la cadena de valor .....	117
	CONCLUSIONES .....	121
	RECOMENDACIONES.....	123
	BIBLIOGRAFÍA.....	125
	APÉNDICES .....	127



## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Acceso a las Tic por parte de las pymes en Centroamérica .....	5
2.	Uso de la banca electrónica por parte de las pymes en Centroamérica.....	8
3.	Esquema de la cadena de valor .....	10
4.	Esquema de computación en la nube .....	13
5.	Spring <i>Framework Runtime</i> .....	28
6.	Macroproceso del punto de venta .....	35
7.	Proceso de registro de ventas.....	36
8.	Proceso de ingreso de mercadería .....	37
9.	Proceso de corte de caja .....	39
10.	Diagrama de caso de uso general .....	56
11.	Diagrama de caso de uso administrar POS .....	57
12.	Inclusiones caso de uso administrar catálogos del sistema.....	58
13.	Herencia caso de uso administrar catálogos del sistema .....	59
14.	Diagrama de caso de uso monitorear POS.....	64
15.	Diagrama de caso de uso operar POS .....	65
16.	Dinámica general del caso de uso operar POS .....	67
17.	Diagrama de secuencia iniciar día de operaciones.....	70
18.	Diagrama de secuencia ingreso de mercadería.....	72
19.	Diagrama de secuencia registrar ventas .....	75
20.	Diagrama de secuencia cerrar día de operaciones.....	79
21.	Infraestructura del sistema de punto de venta en la nube.....	81
22.	Arquitectura general del sistema de punto de venta en la nube.....	83

23.	Detalle de la arquitectura del sistema de punto de venta en la nube.....	85
24.	Diagrama de componentes del Sistema de Punto de Venta en la nube.....	88
25.	Diagrama de clases del componente configuración POS.....	90
26.	Diagrama de clases del componente catálogo de productos.....	91
27.	Diagrama de clases del componente facturación .....	92
28.	Modelo de datos del sistema de punto de venta en la nube .....	94
29.	Estructura de directorios de un proyecto java gestionado con maven .....	96
30.	Estructura de paquetes y carpetas del proyecto de punto de venta en la nube .....	97
31.	Mapeo JPA de la entidad Tienda.....	98
32.	Repositorio JPA con Spring-Data para la entidad Tienda.....	100
33.	Interfaz e implementación del servicio de catálogo de productos....	101
34.	Estructura interna de la implementación del componente de catálogo de productos .....	102
35.	ManagedBean para el catálogo de categorías de productos.....	104
36.	Vista JSF para el listado de productos utilizando el ManagedBean de categorías de productos .....	106
37.	Configuración de una clase de pruebas con JUnit y Spring .....	108
38.	Cantidad de errores reportados durante los cuatro meses de pruebas beta.....	111

## TABLAS

I.	Facturación por publicidad en medios de comunicación en Estados Unidos – 2010.....	6
II.	Requerimientos funcionales sistema de punto de venta.....	50



III.	Requerimientos no funcionales sistema de punto de venta .....	53
IV.	Caso de uso administrar personal.....	60
V.	Caso de uso configurar precios de productos.....	61
VI.	Caso de uso configurar descuentos.....	62
VII.	Costo mensual de operación del sistema de punto de venta en la nube .....	118
VIII.	Promedio mensual de reducción de costo al utilizar el sistema de punto de venta en la nube.....	119



## GLOSARIO

<b>AJAX</b>	Técnica de desarrollo web para el intercambio de información con un servidor, y la actualización de una página, sin necesidad de recargar toda la página web.
<b>API</b>	Conjunto de funciones y procedimientos que ofrece un <i>framework</i> , para ser utilizado por otro software como una capa de abstracción.
<b>Aplicación web</b>	Herramienta, software o aplicación que los usuarios pueden acceder a través de internet, o bien una red interna privada, mediante un navegador. Conjunto de páginas web que proveen un determinado servicio al usuario.
<b>Bean</b>	Componente de software, cuya finalidad es el encapsulamiento de comportamiento y la reutilización.
<b><i>Cloud computing</i></b>	Concepto que involucra un gran número de computadores conectadas en tiempo real, a través de internet, que permite ofrecer servicio de computación.

<b>DBMS</b>	Conjunto de programas o herramientas que permiten el almacenamiento, manipulación y extracción de la información de una base de datos.
<b><i>Framework</i></b>	Marco de trabajo que define una estructura conceptual y tecnológica. Utilizado como base para el desarrollo de un programa de software.
<b>Https</b>	Protocolo de aplicación destinado a la transferencia segura de datos de Hypertexto.
<b>J2EE</b>	Plataforma de programación para desarrollar y ejecutar software en lenguaje Java.
<b>JSF</b>	Java Server Faces, <i>framework</i> , para aplicaciones Java web que provee un conjunto de componentes enriquecidos para la creación de interfaces de usuario.
<b>JSP</b>	Java Server Page, tecnología utilizada para crear páginas web dinámicas basadas en HTML, XML, etc.
<b>Logs</b>	Anglicismo utilizado en la lengua de habla hispana, en el campo de la informática. Registro de eventos durante un periodo de tiempo.
<b>ManagedBean</b>	JavaBean controlado por el <i>framework</i> JSF. Utilizado como modelo de un componente de la interfaz de usuario.

<b>Mapeo Objeto-Relacional</b>	Técnica de programación para convertir datos contenidos en objetos a bases de datos relacionales, y viceversa. Se utiliza un motor de persistencia.
<b>Nube</b>	Conjunto de aplicaciones, plataformas e infraestructuras, distribuidas en cualquier parte del mundo, disponibles a través de internet.
<b>PIB</b>	Producto interno bruto. Medida macroeconómica para expresar el valor monetario de la producción de bienes y servicios de un país.
<b>Pyme</b>	Pequeña y mediana empresa. Empresa con ciertos límites físicos y financieros, definidos por el Estado o gobierno al que está sujeta.
<b>Query</b>	Cadena de texto con una sintaxis bien definida, que expresa una consulta de información a una base de datos.
<b>SaaS</b>	Software como servicio. Modelo de distribución de software, en donde las aplicaciones de software y los datos manejados por ellas se almacenan en un servidor, y se accede a través de internet, desde un navegador web.
<b>Servlet</b>	Clase programada en lenguaje Java, utilizada para extender las capacidades de un servidor.

**Tic's**

Tecnologías de la información y comunicación.

**XML**

Lenguaje semiestructurado, basado en etiquetas, utilizado para almacenar información de forma legible y estándar.

## RESUMEN

Un punto de venta es el lugar en donde se paga por lo que se ha consumido en un establecimiento, ya sea un producto o un servicio. Forma parte de la cadena de valor de una empresa; es por esto que si se logra mejorar el rendimiento de este eslabón, se puede tener un impacto positivo en la productividad de toda la cadena de la empresa.

Hasta el 2009, las pequeñas y medianas empresas en Guatemala representan un 85 por ciento de los empleos activos. Al lograr impulsar la productividad de las pymes en Guatemala, se puede obtener un gran impulso en la economía general del país. Este impulso que se busca, puede lograrse a través de la adopción de tecnología por parte de las pymes, específicamente, el uso de internet para automatizar algunos procesos dentro de la empresa.

En informática, la nube es un conjunto de aplicaciones, plataformas e infraestructuras, distribuidas en cualquier parte del mundo, y que se comunican a través de internet. Cuando se habla de una aplicación en la nube, es referirse precisamente, a colocar una aplicación de software en cualquier lugar, pero que esté disponible para su uso solamente con poseer una conexión a internet.

Un servicio en la nube ofrece muchas ventajas, al enfocarse a un POS en la nube, se puede obtener beneficios como la centralización de la información, información actualizada en tiempo real, rapidez en la generación de reportes y/o estadísticas para la toma de decisión, entre otros.

En el presente trabajo se definen las bases para la construcción y distribución de un POS en la nube, desde la toma de requerimientos y definición de las características del sistema; hasta el despliegue de la aplicación como un servicio en la nube, y su forma de distribución. Asimismo, se definen las tecnologías utilizadas en cada capa del sistema.



## **OBJETIVOS**

### **General**

Desarrollar e implantar un sistema de punto de venta en la nube, utilizando la estrategia de afiliación para la distribución del mismo, contribuyendo así con el desarrollo y el crecimiento de las pequeñas y medianas empresas en Guatemala, disminuyendo directamente los costos en que se incurre al contratar un sistema de punto de venta cliente-servidor con licenciamiento.

### **Específicos**

1. Utilizar la tecnología de la nube para brindar un servicio como una parte de la cadena de valor de las pequeñas y medianas empresas en Guatemala.
2. Aprovechar al máximo los beneficios que un servicio en la nube puede proveer a una empresa en su productividad.
3. Implementar la estrategia de afiliación como distribución de un servicio de software, tomando las ventajas que esta presenta frente a los sistemas de pago de licencia.
4. Definir un esquema general de un punto de venta orientado a su implementación como un servicio en la nube.

5. Contribuir con el desarrollo y crecimiento de las pequeñas y medianas empresas en Guatemala, por medio de la optimización de la inversión en tecnologías de información y comunicación, específicamente en un sistema de punto de venta.

## INTRODUCCIÓN

Actualmente, las pequeñas y medianas empresas en Guatemala contribuyen aproximadamente con el 40 por ciento del producto interno bruto (PIB) del país, y el 85 por ciento del empleo; además se estima que el 80 por ciento de los empleos pertenecen a empresas con menos de 20 empleados. Las estadísticas anteriores develan la importancia del crecimiento y el fortalecimiento de las pequeñas y medianas empresas en Guatemala, pues un desarrollo de las mismas tendría una incidencia enorme en la economía general del país.

Desde la segunda mitad de la década de los noventa, el uso intensivo del internet y de otras tecnologías de información y comunicación, produjeron y siguen produciendo una revolución en la economía de las empresas a nivel mundial. El incremento de las actividades internacionales de las empresas también está ligado al uso de las tecnologías de la información, y la comunicación. Al final, esto desemboca en lo que se conoce como globalización.

El conocimiento ha llegado a constituirse en el factor principal de la producción en las empresas, esto debido a la adopción de aplicaciones de cómputo que automatizan los procesos productivos, principalmente las aplicaciones basadas en el trabajo en internet.

Esto genera una dependencia total de la productividad de las empresas en el nivel de inversión en la tecnología en internet y del beneficio que se obtiene de esta inversión.

Las aplicaciones de cómputo que se utilizan para automatizar procesos de producción, así como otras muchas actividades en las empresas, han sufrido una revolución increíblemente grande debido a los servicios y beneficios que se pueden obtener con tan solo estar conectados a internet.

Este trabajo de graduación se enfoca en cómo un servicio primordial en la cadena de valor de las empresas (punto de venta), puede ser desarrollado como un servicio en la nube, y así ayudar en el rendimiento productivo de las pequeñas y medianas empresas en Guatemala.

En el primer capítulo se proporciona un marco teórico inicial, el cual ayudará al lector a entender cada uno de los conceptos abordados en este trabajo. También se contextualizan cada uno de los conceptos a la actualidad en Guatemala.

En el segundo capítulo se describe el enfoque del producto. También se definen los requerimientos para el desarrollo del punto de venta en la nube y la estrategia utilizada para la distribución del mismo.

En el tercer capítulo se encuentra el proceso de desarrollo del punto de venta en la nube, los principales procesos y actividades necesarios para las etapas del desarrollo de software: análisis, diseño y codificación.

En el cuarto capítulo se describe el proceso de implantación del software y los requerimientos mínimos para su funcionalidad. También se incluye un análisis del rendimiento del punto de venta en la nube perteneciente a una etapa de la cadena de valor en las empresas.

# **1. MARCO TEÓRICO INICIAL. CONTEXTO DE LAS PYMES EN GUATEMALA**

En Guatemala, las pequeñas y medianas empresas conforman la mayor parte de generación de empleo en el país y contribuyen con casi la mitad del producto interno bruto. Es por esto que atender las necesidades de las pequeñas y medianas empresas en Guatemala es una forma efectiva de contribuir con el desarrollo y crecimiento de la economía guatemalteca. A partir de este momento, para referirse a las pequeñas y medianas empresas se utilizará pymes.

En toda empresa se necesita de una cadena de valor para poder desechar las actividades que no aportan nada para darle un valor agregado al producto o servicio y poder integrar eficientemente todos los aspectos que generan un valor agregado para el cliente. En las pymes, de igual forma se necesita de una cantidad de actividades que le den un valor agregado a los productos y servicios. Entre esta gran cantidad de actividades se puede encontrar el punto de venta, que es una parte del proceso de ventas de una empresa.

Un punto de venta bien organizado y apoyado por las tecnologías más recientes de aplicaciones web, puede generar muchos beneficios a una empresa y cumplir con su propósito como un eslabón de la cadena de valor de la misma.

## 1.1. ¿Qué es un punto de venta?

Una definición acertada es que un punto de venta es el lugar en donde los clientes pueden conocer y adquirir los productos que una persona o empresa en particular pueden ofrecerle. Esta expresión: punto de venta, proviene de la traducción, del idioma inglés, del término *point of sale* (POS), aunque en este idioma se refiere específicamente, al lugar en donde se realiza un transacción económica, es decir, la caja o línea de cajas.

Para los propósitos de este trabajo el enfoque será el punto de venta como el lugar en donde se realizan las transacciones económicas. Asimismo, como el lugar en donde se paga por lo que se ha consumido en un establecimiento, ya sea un producto o un servicio. También denominado Terminal de Punto de Venta.

Un Sistema de Punto de Venta está compuesto de tecnología y dispositivos o, hardware y software, que ayuda en la gestión de ventas de un establecimiento que ofrece sus productos o servicios al público; esto lo realiza mediante una interfaz accesible al vendedor. Los lugares, negocios, tiendas, etc. En donde se utilizan sistemas de Punto de Venta son muy diversos. Se puede encontrar desde pequeñas abarroterías, hasta grandes cadenas de supermercados.

Durante mucho tiempo, las cajas registradoras eran el sistema de punto de venta más utilizado por los negocios. Actualmente se continúa con este tipo de sistemas, aunque en algunos casos han evolucionado conectando dichas cajas registradoras a la computadora, para tener un sistema computarizado que les ayude a tener el control de su negocio. Estos sistemas computarizados están fiscalizados por la Superintendencia de Administración Tributaria (SAT), y

deben cumplir con una lista de requisitos para poder ser auto impresores de facturas.

La necesidad de tener mayor información, y sobre todo de calidad, ha desencadenado la evolución total de los sistemas de punto de venta. Cada vez son más sofisticados y se han agregado más periféricos que permiten una operación eficiente, por ejemplo: los lectores de códigos de barras, impresoras de tickets, caja con el dinero, entre otros.

## **1.2. Contexto de las pymes en Guatemala**

Con una población multiétnica alrededor de 13 millones y un PIB per cápita de US\$2 610,00 muestra que Guatemala es la economía más grande de América Central.

En Guatemala, las pequeñas y medianas empresas (pymes) dan cuenta de la mayor parte del empleo. La Federación de la Pequeña y Mediana Empresa Guatemalteca calcula que las pymes contribuyen con el 40 por ciento del PIB y el 85 por ciento del empleo (2009). “El informe sobre el estado de la región (2008) calcula que el 80 por ciento de los puestos de trabajo responden a empresas con menos de 20 empleados”<sup>1</sup>.

Mejorar el desempeño de las pymes e incrementar su participación en los mercados locales y mundiales, puede causar un impacto enorme y positivo para toda la economía del país. Por ejemplo: medidas que aumenten la productividad, bajen los costos logísticos, estimulen el acceso a nuevas

---

<sup>1</sup>

URL([http://siteresources.worldbank.org/GUATEMALAINSPANISHEXT/Resources/Desarrollo\\_de\\_PyMEs\\_en\\_Guatemala\\_\\_Banco\\_Mundial.pdf](http://siteresources.worldbank.org/GUATEMALAINSPANISHEXT/Resources/Desarrollo_de_PyMEs_en_Guatemala__Banco_Mundial.pdf)). [Consulta: mayo de 2012]

tecnologías, e integren las pymes a las cadenas de valor; pueden todas ellas generar empleo, elevar el nivel de ingreso y reducir la pobreza.

Uno de los principales retos que enfrentan hoy las pymes de cualquier país, en especial aquellos en vías de desarrollo, consisten en saber cómo tomar ventaja de los nuevos recursos tecnológicos, principalmente las aplicaciones de cómputo e internet, y al mismo tiempo enfrentar una intensa y creciente competencia, tanto en el interior del país como en las exportaciones. En este contexto llamado globalización, y en una sociedad que cada vez está más basada en el conocimiento, las pequeñas y medianas empresas de Guatemala enfrentan un reto, ya que actualmente se cuenta con un nivel demasiado bajo de adopción de aplicaciones de cómputo o internet, y en general, de tecnologías de información y comunicación.

### **1.2.1. Adopción de las Tic en las pymes en Guatemala**

Las tecnologías de información y comunicación juegan un papel muy importante en la productividad y competitividad de las empresas en el mercado, tanto en países desarrollados como en vías de desarrollo. En un análisis realizado por Ricardo Monge-González, basado en un estudio que se realizó a 788 micros, pequeñas y medianas empresas de los cinco países de la región centroamericana, se pudo obtener información muy importante de la situación en ese momento de las pymes y las Tic, no solo de Guatemala sino de Centro América.

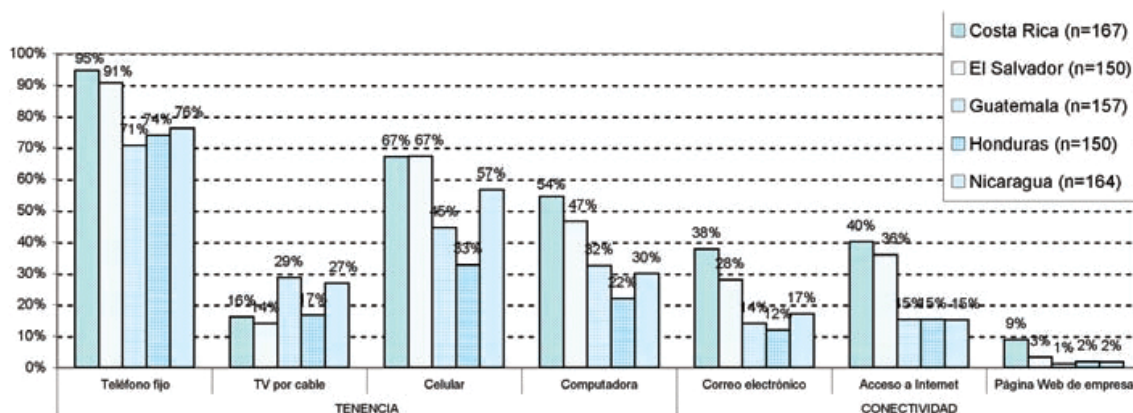
En la figura 1 se puede observar cómo se encuentra el uso de las tecnologías de información y comunicación más importantes en los países de la región centroamericana. Los países incluidos son: Costa Rica, El Salvador, Guatemala, Honduras, Nicaragua, y se evalúan las siguientes tecnologías:



telefonía fija, telefonía celular, internet, computadora, página web de empresa, entre otras.

Con los datos recolectados se puede observar que casi tres cuartas partes de las pymes en Guatemala poseen teléfonos fijos (71 por ciento), pero en el caso de los teléfonos celulares solo el 45 por ciento cuentan con él. Cerca de un 32 por ciento de las pymes utilizan computadoras en sus procesos de negocio, cayendo a un 15 por ciento en relación a las que cuentan con acceso a internet. El dato más notorio y más alarmante es el 1 por ciento que representa a las pymes que poseen páginas web propias, en las cuales pueden dar a conocer los productos o servicios que ofrecen<sup>2</sup>.

Figura 1. Acceso a las Tic por parte de las pymes en Centroamérica



Fuente: MONGE, Ricardo. Tic's en las Pymes de Centroamérica. International Development Research Center y la editorial tecnológica de Costa Rica. p 85.

La importancia de una empresa que tenga acceso a internet y que utilice este medio para su productividad es elevada, esta radica en dar a conocer qué

<sup>2</sup> URL(<http://web.idrc.ca/openebooks/214-7/>) [Consulta: mayo de 2012]

es lo que ofrecen al mercado. Actualmente, la publicidad por internet ha crecido al punto de desplazarla en diarios en Estados Unidos, colocándose únicamente por debajo de la televisión<sup>3</sup> (ver tabla I), cifras que fácilmente pueden extrapolarse a toda América, por ser Estados Unidos la potencia económica mundial. Esta es solo una ventaja que las pymes pueden obtener al utilizar las Tic en sus procesos de negocio. De igual forma que en la publicidad, las Tic pueden ser de mucha utilidad al apoyar en la automatización de procesos dentro de una organización, en este caso, dentro de las pymes.

Tabla I. **Facturación por publicidad en medios de comunicación en Estados Unidos – 2010**

<b>Medio de comunicación</b>	<b>Facturación por publicidad (en billones)</b>
<b>Distribución por TV</b>	<b>28,6</b>
<b>Internet</b>	<b>26,0</b>
<b>Periódicos</b>	<b>22,8</b>
<b>Radio</b>	<b>15,3</b>
<b>Directorios</b>	<b>11,5</b>
<b>Revistas</b>	<b>10,2</b>

Fuente: elaboración propia, basado en el artículo: Juan Varela. Internet supera la publicidad de diarios y cable en EE UU. Periodistas 21 <http://periodistas21.blogspot.com/2011/04/internet-supera-la-publicidad-de.html>. Consulta: mayo de 2012

Los beneficios que una empresa puede obtener al automatizar sus procesos están relacionados con reducción de costos y optimización del uso del tiempo. Es por esto que al generar una cultura de adopción de tecnología en las

<sup>3</sup> URL(<http://periodistas21.blogspot.com/2011/04/internet-supera-la-publicidad-de.html>).

[Consulta: mayo de 2012]

empresas se puede ayudar al desarrollo de las mismas. El tema de estudio de este trabajo de investigación está orientado a promover el uso de estas tecnologías en una parte de la cadena de valor de las empresas, el punto de venta.

El hecho de no adoptar las tecnologías como parte de su proceso de negocio, en las pymes de Guatemala, se debe principalmente a tres razones:

- Pensar que las herramientas tecnológicas no son necesarias para su actividad productiva.
- Considerar que el costo de implementar nuevas tecnologías es demasiado elevado.
- Desconocer cómo usar estas herramientas.

Las primeras dos razones están relacionadas con el desconocimiento que las pymes tienen acerca de las tecnologías que están a su alcance y de la importancia de estas para mejorar el desempeño de su proceso de producción. La tercera razón habla de la importancia de proveer programas de capacitación para que las pymes puedan aprovechar al máximo los beneficios de las Tic, principalmente del uso de computadoras y de internet.

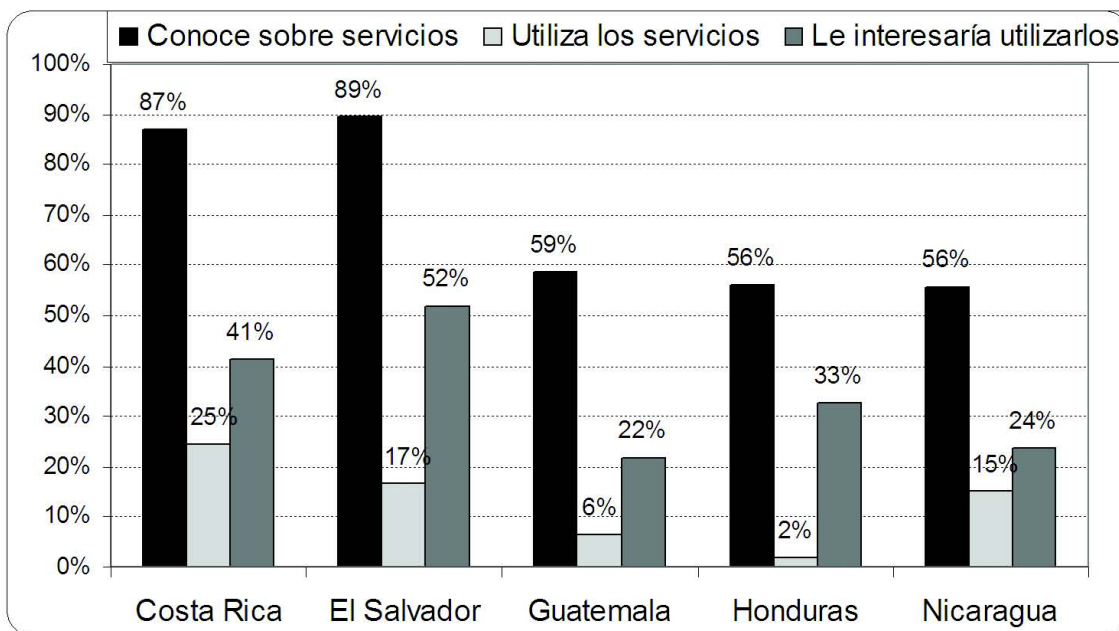
Otra forma de ver cómo se encuentra la actualidad de las pymes en Guatemala en el uso de Tic, es el empleo de banca electrónica en sus negocios. En relación a este tema, en Guatemala solamente el 6 por ciento de las pymes utilizan este servicio, y además solo a un 22 por ciento le interesaría utilizarlas y conocerlas<sup>4</sup> (ver figura 2). Esta cifra es alarmante por la importancia y los beneficios que las Tic brindan a las empresas; estas demuestran que la gran mayoría de las pymes en Guatemala se encuentran en desventaja

---

<sup>4</sup> URL(<http://web.idrc.ca/openebooks/214-7/>). [Consulta: mayo de 2012]

competitiva con otros países de Centroamérica y con las grandes empresas del medio, por el simple hecho de no aprovechar los beneficios y ventajas de las Tic.

Figura 2. **Uso de la banca electrónica por parte de las pymes en Centroamérica**



Fuente: elaboración propia, basado la fuente: Monge, Ricardo. Tic's en las Pymes de Centroamérica. International Development Research Center y la editorial tecnológica de Costa Rica.

Las pymes en Guatemala tienen un atraso en la adopción de las Tic como parte de sus procesos de producción y negocio, lo cual las deja en clara desventaja con competidores que si las han adquirido. La automatización de procesos dentro de la empresa es un aspecto que puede dejar grandes beneficios a la misma. De la misma manera, el uso de computadoras e internet es de vital importancia para el crecimiento y desarrollo de la empresa, algo que

en Guatemala está muy atrasado cuando a pymes se refiere. La incorporación de estas tecnologías puede crear una competencia sana entre las mismas pymes y, además puede brindarles una oportunidad para que, en algún momento, estar compitiendo con empresas grandes en su medio.

### **1.3. Punto de venta como una etapa en la cadena de valor de las pymes**

La cadena de valor es un modelo teórico que permite identificar y describir cada una de las actividades de una organización, las cuales se ven involucradas directa o indirectamente en el proceso de producción y en toda la actividad económica de la empresa. Esta herramienta fue descrita y popularizada por Michael Porter en su libro: *Competitive Advantage: Creating and Sustaining Superior Performance*.

La cadena de valor permite hacer un análisis a fondo de cada actividad de la empresa, califica su importancia y cuantifica una estimación del impacto que esta tiene en el proceso de producción o actividad económica de la empresa. También evalúa los costos que implica cada actividad. Este análisis genera una ventaja competitiva en las empresas, pues permite identificar las actividades críticas en la organización, así como las más importantes y que generan más valor agregado al producto o servicio final, es decir, generan valor agregado al cliente. Si una actividad es diferenciada de las demás empresas que conforman la competencia, también se puede explotar y tomar ventaja ante estas empresas.

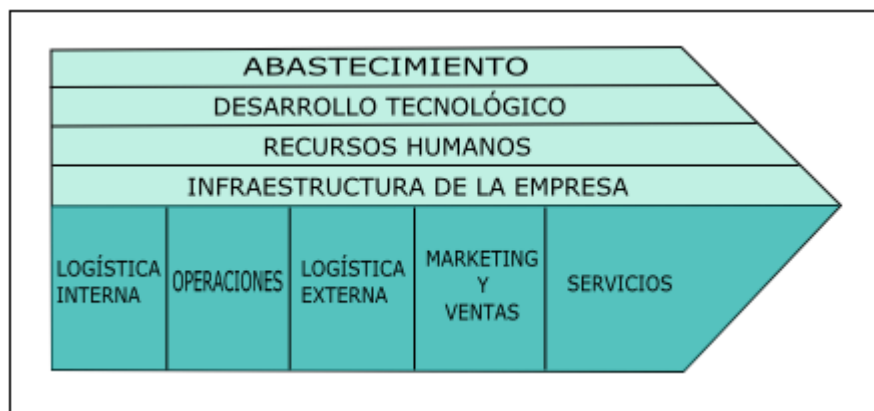
Este análisis puede variar en cada organización, dependiendo de la actividad económica, a que se dedique y el enfoque que tenga. Con base en ese análisis se pueden definir actividades primarias, que son las que le dan

mucho valor al proceso de producción, y las secundarias, las cuales auxilian a las primarias para generar el producto o servicio final.

Las actividades primarias se refieren a la creación física del producto, su venta y el servicio posventa. El modelo de la cadena de valor define cinco actividades primarias (ver figura 3), las cuales se listan a continuación:

- Logística interna
- Operaciones de producción
- Logística externa
- *Marketing* y ventas
- Servicio posventa

Figura 3. Esquema de la cadena de valor



Fuente: Cadena de valor. Wikipedia. [http://es.wikipedia.org/wiki/Cadena\\_de\\_valor](http://es.wikipedia.org/wiki/Cadena_de_valor). Consulta: junio de 2012.

Las actividades secundarias son las que apoyan a las primarias. Estas no son fundamentales, pero si ayudan al desempeño eficiente de las demás

actividades. Las secundarias pueden variar mucho en cada empresa, algunos ejemplos de ellas son:

- Infraestructura de la organización
- Dirección de recursos humanos
- Investigación y desarrollo de tecnología
- Abastecimiento

El proceso de venta de los productos es considerado como una actividad primaria en la cadena de valor. Debido a esto, las pymes deben fortalecer este proceso para poder tener beneficios y obtener una ventaja competitiva. El punto de venta, al ser parte del proceso de venta de una empresa, forma parte importante de la cadena de valor en una organización. Optimizando las actividades en los puntos de venta se puede agregar un valor final al cliente.

Lo importante es saber cómo puede optimizarse el proceso de punto de venta. A continuación se enuncian algunas de las formas y ventajas de optimizar un punto de venta:

- Automatizar las actividades en un punto de venta: beneficia en reducción de tiempo; esto da un valor agregado al cliente, pues se reduce su tiempo de espera para finalizar el proceso de compra/venta. También beneficia a los empleados, pues se invierte menos tiempo en esta actividad y agiliza la atención al cliente.
- Digitalizar este proceso: utilizar una computadora para registrar los detalles de la venta y la impresión del ticket. Si se implementa esta opción, la empresa se puede beneficiar al tener la información actualizada en tiempo real, como su inventario, dinero en caja, entre otros. Al realizar una

transacción se modifican los registros necesarios para actualizar el estado del negocio.

- La generación de reportes de venta y de inventario se vuelve instantánea con resultados en tiempo real, lo que permite a los gerentes o encargados, una toma de decisión más acertada con base en el estado real de la empresa.
- Si se implementan tecnologías de informática avanzadas, como la minería de datos, se pueden generar estadísticas y tendencias en el consumo de los clientes. Esta información es de utilidad para establecer estrategias organizacionales que generen ventajas competitivas ante otras empresas en el mercado.

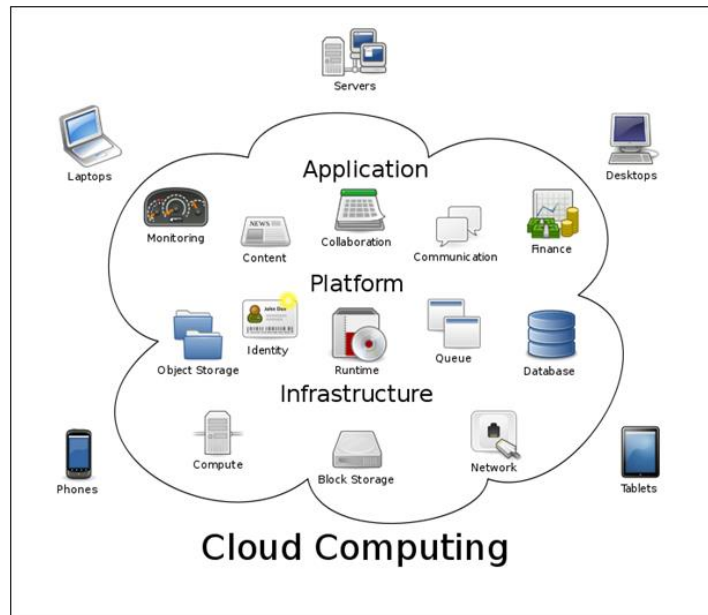
#### **1.4. ¿Qué es un servicio en la nube?**

Para poder definir el concepto de servicio en la nube, primero hay que aclarar qué es la nube en relación a la informática. Término relativamente nuevo, pero que está marcando la tendencia de las tecnologías de informática. Muchas veces se ha oído cosas como: mantener los archivos en la nube, pero ¿qué significa verdaderamente la nube?

Es un paradigma, una forma de utilizar los recursos existentes, más no un objeto físico. Sin embargo, la nube está conformada por un conjunto de aplicaciones, plataformas e infraestructuras distribuidas en cualquier parte del mundo. Estos componentes se relacionan a través de internet y se mantienen en constante transferencia de información entre sí (ver figura 4). Como ya se dijo, es solamente un paradigma, una forma de utilizar los recursos, porque en realidad todos estos componentes siempre han existido.



Figura 4. **Esquema de computación en la nube**



Fuente: Computación en la nube. Wikipedia. [http://es.wikipedia.org/wiki/Computaci3n\\_en\\_nube](http://es.wikipedia.org/wiki/Computaci3n_en_nube).  
Consulta: junio de 2012.

El objetivo de la nube es proveer de servicios a los usuarios a través de internet, de modo que puedan acceder a estos servicios sin tener conocimiento de la gestión de los recursos que se utilizan. El usuario tiene a su disposición un catálogo de servicios estandarizados que le ayudan a responder a las necesidades, de forma flexible y adaptativa.

Un concepto importante para la computación en la nube es la abstracción. Esta consiste en aislar un elemento de su contexto o de otros elementos que lo acompañen. Para decirlo en palabras más sencillas, cuando se abstrae la funcionalidad de un software, se refiere a que el usuario solamente sabrá que hace un servicio y lo utilizará, mas no le interesa cómo lo hace.

El usuario final, únicamente utilizará los servicios provistos por los proveedores, como transacciones de compra/venta, almacenamiento de archivos, gestores de agendas y calendarios, entre muchos más; pero solamente los utilizará y no le interesara la forma en que los proveedores gestionan la información.

#### **1.4.1. Ventajas de computación en la nube**

Si se observa desde el punto de vista del usuario, y este es una empresa que utilizará servicios en la nube, los beneficios son muchos. Primero se puede decir que, al utilizar la computación en la nube, los usuarios tienen acceso a su información desde cualquier punto en el mundo, solamente se precisa de una conexión a internet y listo, la información está al alcance en cualquier parte y en todo momento; también se rompen las fronteras y las limitaciones de tiempo y espacio.

Los servicios en la nube, para el usuario final, requieren de una implementación fácil y rápida, se puede empezar a utilizar una solución de software de forma casi instantánea; ya no se debe esperar semanas o meses para comenzar a utilizar las aplicaciones, solamente se precisa de horas o unos pocos días para favorecerse de los beneficios de la solución.

Con los servicios en la nube se transfieren los riesgos y el costo de los riesgos a terceros, esto quiere decir que la responsabilidad de la persistencia de los datos, los planes de contingencia, la planificación de riesgos, entre otros; se transfiere a la empresa que está prestando el servicio; mientras que el usuario final se dedica únicamente a utilizar los servicios que le interesen. De cualquier forma los proveedores de servicios deben generar copias de respaldo a los clientes, para que ellos puedan contar con un respaldo ante cualquier incidente.

La centralización de la información tiene sus criterios de ventaja y de desventaja. En este caso, las ventajas que provee la centralización de la información es que se puede obtener información actualizada en tiempo real y sin variaciones entre un usuario y otro. Todos los usuarios que tengan privilegios de obtener cierta información, tendrán el mismo resultado en un instante específico, sin importar su ubicación geográfica. También se puede obtener una respuesta a requerimientos con información precisa y así hacer una toma de decisión basada en información real y actualizada en cualquier lugar y en cualquier momento.

#### **1.4.2. Desventajas de computación en la nube**

Al igual que todo, los servicios en la nube tienen algunos puntos en contra, los cuales gestionados de una forma correcta pueden mitigarse y reducir su impacto si en algún momento se produjeran.

La centralización de las aplicaciones y el almacenamiento de los datos en puntos determinados y por empresas específicas, crean interdependencia entre los proveedores de servicios. De igual forma, el uso de los servicios deseados está ligado y depende de la disponibilidad que se tiene para acceder a internet. Este último punto es una desventaja, pero con la tendencia de acceso a internet no se considera un problema primario, debido al crecimiento acelerado en el índice de empresas que tienen acceso a internet.

Cuando se trabaja con información confidencial, puede crearse cierta vulnerabilidad ante el robo de esta información; al no tener los datos almacenados físicamente en un ámbito local, todos estos datos deben salir a navegar por la red y podrían volverse vulnerables. Para mitigar este riesgo se pueden utilizar protocolos de comunicación seguros, por ejemplo: HTTPS, con

la única salvedad que al utilizar estos protocolos, la velocidad y el rendimiento se ven afectados por la sobrecarga de transferencia de datos que estos requieren.

El paradigma de computación en la nube ha sido criticado desde su creación, porque algunas personas piensan que se está privando al usuario de la libertad y el derecho que tiene sobre su propia información, al no tenerla almacenada físicamente en sus servidores. Según estas personas se está obligando a los usuarios a ser dependientes de los proveedores de servicios.

## **2. ENFOQUE, ESTRATEGIA Y REQUERIMIENTOS GENERALES DEL PRODUCTO**

Después de haber establecido la situación actual de las pymes en Guatemala, conocer un poco más de su contexto, sus oportunidades de desarrollo; así como de conocer la importancia de un punto de venta en la cadena de valor de las empresas; ahora se plantea una forma de ayudar al desarrollo de las pymes en Guatemala, crear una oportunidad para que empresas con pocos recursos de inversión adquieran una ventaja competitiva a través de las Tic.

Antes de continuar con el planteamiento de este tema, debe quedar claro que una herramienta de software o una tecnología de información y comunicación no es diseñada para reemplazar al 100 por ciento los procesos o actividades económicas de una empresa. Un software se desarrolla con la finalidad de apoyar en estos procesos y facilitar su gestión, pero nunca para reemplazarlos por completo.

El propósito de este trabajo es crear un sistema de punto de venta como un servicio en la nube, explotar los beneficios de la automatización de procesos dentro de una empresa y aprovechar las ventajas que un servicio en la nube provee a las organizaciones. Generar valor agregado al cliente por medio de este eslabón en la cadena de valor (punto de venta) puede contribuir con el desarrollo de las pymes.

Como ya se mencionó, la adopción de tecnologías de información y comunicación por parte de las pymes en Guatemala es bastante bajo, esto

debido a diversos factores. Una de las principales causas que provoca que las empresas no adopten las Tic como parte de su actividad productiva y económica es el costo elevado que implica la implementación de las mismas tecnologías, una pequeña o mediana empresa muchas veces no cuenta con los recursos necesarios para poder invertir en una solución de software que le ayude en la gestión de su negocio, o en alguna actividad específica.

## **2.1. ¿Hacia quiénes está dirigido?**

El sistema de punto de venta como un servicio en la nube es desarrollado con el firme propósito de contribuir con la transición y adopción de las Tic por parte de las pequeñas y medianas empresas en Guatemala. Cualquier empresa que cuente con un lugar en donde vende sus productos o servicios al público y desea automatizar el proceso de cobro puede hacer uso de esta solución de software.

Empresas que están en crecimiento y desean expandirse geográficamente se encuentran con la dificultad de tener aislados cada una de sus sucursales. El tiempo para poder tomar la información de ventas de una sucursal, y luego centralizar esa información junto con la de las demás sucursales suele ser tardado y complicado para empresas pequeñas, pues no cuentan con una herramienta que les gestione este proceso.

De igual forma, el tiempo de respuesta ante solicitudes de los clientes en cuanto a disponibilidad de mercadería, o el tiempo de respuesta ante toma de decisiones acerca de abastecimiento para las bodegas, también se ve afectada por el hecho de no contar con información verídica en tiempo real acerca de cada uno de los puntos de venta y sucursales.

A todas estas empresas que, por falta de capital para invertir en alguna de las soluciones ya existentes en el mercado, que tienen un costo elevado, además de la gran inversión que se necesita hacer en infraestructura para echar a andar un sistemas de estos; a estas empresas se dirige el presente trabajo y solución de software, que contribuye con la expansión y el desarrollo, organizacional y comercial de dichas empresas.

Al ser un producto enfocado a pequeñas y medianas empresas, por consiguiente se habla de un producto que representará para las empresas una inversión bastante pequeña, que esté al alcance de las posibilidades de pago de la mayoría de las pymes según el contexto guatemalteco.

## **2.2. Alcance del producto**

En este trabajo, el producto se refiere a un punto de venta implementado como un servicio en la nube, aprovechando todos los beneficios de la computación en la nube y optimizando al máximo las actividades propias de un punto de venta.

El alcance del sistema se enfoca en un punto de venta genérico, esto quiere decir que no se toman requerimientos de una empresa en específico, sino que se utiliza la abstracción para obtener un proceso y un modelo genérico de un punto de venta, que pueda ser aplicado en cualquier empresa que así lo necesite.

Orientado a un alcance geográfico de un país completo, este producto cumple con todas las normas de ley necesarias para que sea funcional en cualquier ubicación geográfica. Se realizan cualquier tipo de cálculos numéricos basados estrictamente en las leyes regulatorias de Guatemala, tomando en

cuenta cualquier tasa de impuesto, interés o cualquier otro tipo, que esté regido por alguna ley en el país.

### **2.3. Requerimientos generales del producto**

Este punto de venta implementado en la nube, como ya se mencionó, está basado en un modelo genérico de un sistema de punto de venta tradicional. Partiendo desde lo anterior se puede definir los objetivos que se deben alcanzar al implementar este sistema en un cualquier pyme en Guatemala:

- Automatizar el proceso en donde se consuma el pago de parte del cliente por un servicio o producto que ha consumido. Esta automatización generará agilidad y eficiencia en este proceso y significa un ahorro en el recurso tiempo para la empresa.
- Digitalizar la información de los clientes y de las ventas, proveyendo así una herramienta para análisis de información más eficiente.
- Centralizar la información anteriormente descrita para poder tener toma de decisión fundamentada en datos reales y actualizados en cualquier momento y lugar.
- Eliminar las barreras de espacio y localización, entonces se podrá tener información de cualquier sucursal en algún lugar del país en tiempo real.
- Reducir los costos de infraestructura que representa implantar un sistema de punto de venta convencional, los cuales normalmente trabajan con una tecnología cliente-servidor. La inversión en estos sistemas va desde la compra del mismo sistema, hasta la infraestructura de red necesaria para que cada punto de venta este sincronizado con el servidor. Para distancias grandes esta solución no es factible.
- Tener una funcionalidad intuitiva y fácil de adoptar por parte de los clientes, es decir, que los procesos de negocio puedan ser entendidos y



adoptados fácilmente en cualquier empresa que traslade su punto de venta a esta plataforma.

- El sistema debe ser capaz de configurar y personalizar un documento legal que representa la transacción de compra venta, en este caso es la factura. Al ser un documento legalmente válido, el sistema permitirá que cada una de las empresas puedan personalizar una factura con todos los aspectos legales que esta requiere.
- El afiliado, administrador de la empresa, puede gestionar cada uno de sus puntos de venta registrados, así como organizarlos por sucursales si así lo requiere. Esta organización permitirá tener reportes de ventas ordenados y detallados, útiles para la toma de decisión de la Gerencia.
- Generar reportes de ventas detallados y generalizados, ya sea por punto de venta, sucursal o la empresa en general. Estos reportes son actualizados en tiempo real, pues todas las transacciones de cada punto de venta están centralizadas en un solo núcleo de información disponible en todo lugar y en cualquier momento.

#### **2.4. Estrategia de distribución del punto de venta en la nube**

Hasta los últimos años, la forma más utilizada para adquirir un sistema o un software era la compra del mismo. Una empresa desarrolla un sistema y lo vende bajo alguna licencia a sus clientes. Para las empresas que se dedican al desarrollo de software es un buen negocio, pues invierten en la creación de los sistemas al inicio del ciclo del negocio y luego solamente obtienen beneficios de venderlo a diferentes clientes.

Este modelo de distribución de software y la venta del mismo, presenta algunas ventajas a los consumidores, por ejemplo, que ellos adquieren físicamente un producto y tienen derechos sobre el mismo hasta el punto en

donde el licenciamiento lo permita. Cuando un empresa vende un software puede estipular si las fuentes de este están incluidas en el precio, es decir, si la empresa desarrolladora entregará o no, al final del proyecto, los archivos fuente que componen el sistema.

Pero si se colocan en una balanza las ventajas y desventajas de comprar un software, con la tecnología que se tiene actualmente, el peso se inclinaría en un gran porcentaje hacia las desventajas.

Al comprar un software, lo primero que se debe tener en cuenta es el precio. Un proveedor de software coloca precios altos por entregar un sistema a un cliente. Esto debido a que el cliente realiza una compra y el sistema le pertenece, a diferencia del licenciamiento, que el cliente tiene derechos sobre el sistema, pero en un periodo de tiempo determinado. Si el sistema se desarrolla a la medida el costo del mismo se eleva en una gran cantidad con respecto a sistemas estándar que ya se encuentran desarrollados.

El mantenimiento de un software comprado e instalado en una o varias computadoras es muy complicado, pues se tiene que estar físicamente en el lugar de distribución para poder realizar un cambio. En el mejor de los casos, se puede distribuir un instalador o se pueden realizar cambios remotamente desde otra computadora. Pero de cualquier forma esto implica un gasto para el cliente y para el mismo proveedor.

La forma tradicional de distribución de software se basa en instalar un conjunto de herramientas (paquetes de software) para satisfacer las necesidades del cliente. Se utiliza la clonación de una estación de trabajo, la cual ya tiene todos los paquetes necesarios instalados, y se replica en las computadoras que se quiere configurar con el sistema. También se están

utilizando mucho los sistemas portables, estos son los que a través de un conjunto de archivos, sin necesidad de instalación, proveen las soluciones a las necesidades del cliente.

Como ya se había mencionado, esta forma de distribución, no solo incurre en desventaja para el cliente, sino también para el proveedor. Cuando se quiere realizar un cambio en un sistema que está distribuido a una gran cantidad de clientes, se deben actualizar los paquetes en las computadoras de cada cliente. Esto es un costo de operación demasiado alto para las empresas desarrolladores de software.

Con la tecnología de la nube se está incursionando una nueva forma de distribuir soluciones de software a los clientes. Esta es la afiliación, membresía o alquiler de los sistemas. Esta modalidad se puede aprovechar al máximo para generarle un producto de valor al cliente, y para reducir muchos costos de operación y mantenimiento a los proveedores de software, entre otras muchas ventajas.

#### **2.4.1. Afiliación o suscripción**

La distribución de software, al igual que cualquier otra actividad económica, está inmersa en el contexto capitalista de la sociedad. El alquiler, o mejor definido como pago por uso, cada día está cobrando más valor en el capitalismo. Se debe pensar que como sociedad hay que compartir los gastos de estructura, para poder aprovechar la economía a escala y ser más eficientes en las actividades productivas.

Cada día se paga más por usar las cosas. Se paga por el uso de la electricidad, una línea telefónica, y otros servicios básicos. En algunos países,

incluso, ya se maneja el alquiler de carros u otros medios de transporte por hora, por día o por algún otro intervalo de tiempo. El software y el hardware no son la excepción. Si se quiere tener una red de servidores disponible, ya no se necesitaría comprar físicamente estos equipos, simplemente se debe alquilar servidores en la nube por el tiempo que mejor convenga.

Los conceptos de *cloud computing*, *software as a service*, entre otros responden a un cambio social de pago por uso de los recursos y de economías de escala. Esto significa para las empresas: eficiencia.

#### **2.4.2. Ventajas y desventajas de la afiliación**

Cuando se habla de afiliación como una forma de vender software, se debe saber exactamente cuáles serán los beneficios al utilizar este modelo de distribución.

Desde el punto de vista del cliente, una ventaja muy clara es la reducción del precio del software. El cliente puede pagar por un software una cantidad de dinero que no afecte el balance financiero de su empresa y, además lo puede hacer por el tiempo que mejor le convenga. Para una pyme en Guatemala es mucho más accesible invertir una cantidad pequeña de dinero cada mes que invertir una gran cantidad que pueda afectar su estabilidad como negocio.

El software como servicio, que se basa en el alquiler, les da a los usuarios o clientes una ventaja de facilidad de acceso. Esto significa que el cliente puede hacer uso de su servicio, en este caso el software, en cualquier lugar y en cualquier momento, le basta con una conexión a internet y estará listo para usarlo. No se necesita más que un navegador de internet, sin necesidad de instalar algún paquete de software a la computadora.

Desde el punto de vista del proveedor de servicios, una ventaja principal es la distribución de sus productos o servicios. Como es un software en la nube, solamente se necesita desplegar una sola vez este en un servidor de aplicaciones propio, instalar una sola vez los paquetes de software necesarios para el funcionamiento del producto en ese mismo servidor, y el sistema estará listo para que todos los usuarios puedan acceder a él desde cualquier punto o terminal.

Si se le quiere dar mantenimiento a un software en la nube que se alquila a una cantidad elevada de usuarios, solamente se debe dar mantenimiento al servidor que está en poder mismo del proveedor y los cambios estarán de forma instantánea en todos los usuarios que tengan acceso al sistema. Todo esto de forma eficiente e instantánea. La reducción en costos por mantenimiento de un sistema alquilado a N usuarios, ya no es proporcional a la cantidad de los mismos usuarios, sino que se reduce al costo de un solo mantenimiento.

Al igual que cualquier modelo propuesto para cualquier situación, este no es perfecto y se le puede encontrar algunas desventajas. Pero comparado con lo que se maneja tradicionalmente en software, las ventajas son mucho mayores que las desventajas.

Una desventaja puede ser que el cliente no posee físicamente un producto, y que su información no la tiene físicamente en su computadora. Esto se convierte en desventaja porque los usuarios no están acostumbrados a este paradigma, pero con una buena seguridad en los servidores y una buena capacitación a los usuarios, esto se puede mitigar.

Por todo lo que se ha descrito en los puntos anteriores es que el modelo propuesto para este proyecto de tesis, es la estrategia de afiliación o alquiler como distribución del software. Esto para proveer de un producto o servicio de valor real a una pequeña o mediana empresa en Guatemala, cuidando principalmente su economía.

El propósito de distribuir con el modelo de alquiler el punto de venta es:

- Primero: contribuir con el desarrollo tecnológico de las pymes en Guatemala por medio de la adopción de las nuevas tecnologías por parte de las mismas.
- Segundo: que esta transición hacia el desarrollo tecnológico no se vea truncada por una situación económica, provocando el estancamiento en el desarrollo de las pymes en Guatemala.

## **2.5. Tecnologías para el desarrollo del producto**

A lo largo de este documento se ha planteado que el punto de venta a desarrollar, será implementado como un servicio en la nube. Esto implica la utilización de herramientas para desarrollo de aplicaciones web.

Las aplicaciones web tienen una gran cantidad de opciones para ser desarrolladas. Las tecnologías más importantes en el contexto de programación tienen *frameworks* dedicados a aplicaciones web. Por ejemplo, las tecnologías desarrolladas por Microsoft, tiene el *framework* ASP.NET para aplicaciones web, el cual es usado para construir sitios web dinámicos. Para el lenguaje de programación Python, también existe un gran número de *frameworks* web, por ejemplo: Django, el cual se basa en el desarrollo rápido y diseño limpio.

Para este trabajo se ha seleccionado el lenguaje de programación Java, por lo que a continuación se describen todas las tecnologías utilizadas en el desarrollo del punto de venta.

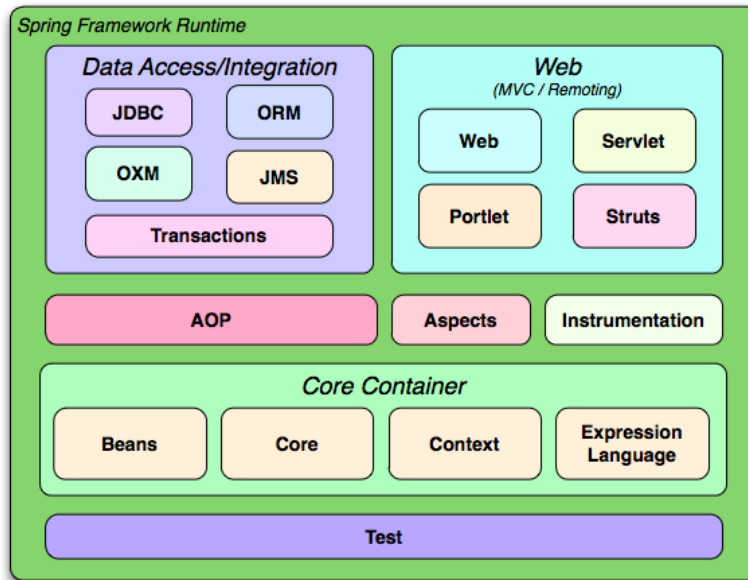
### **2.5.1. Infraestructura**

Spring es un *framework* que proporciona una infraestructura que actúa como soporte para el desarrollo de aplicaciones Java. Spring maneja toda la infraestructura lo cual permite dedicarse a lo que realmente importa, el desarrollo de la aplicación o sistema. Para entenderlo de una mejor manera, Spring funciona como el pegamento que une todos los componentes de una aplicación, manejando su propio ciclo de vida y la interacción entre todos los componentes.

Spring es conocido como un contenedor ligero de aplicaciones, al contrario de un servidor de aplicaciones J2EE. Para poder utilizar Spring basta con tener un contenedor de servlets como Tomcat o Jetty.

El núcleo de este *framework* realiza la inversión de control, o como se conoce actualmente, inyección de dependencias. Esto significa que la creación de los objetos en una aplicación es realizada externamente a la lógica de programación del negocio, a cargo de un contenedor externo, en este caso Spring. Además de crear los objetos necesarios para la aplicación, los inyecta en otros objetos, según se haya definido. Esto deja limpio el código que contiene la lógica del negocio, o la lógica de la vista, es decir que desaparece toda la parte de instanciar objetos y colocar las dependencias de unos con otros.

Figura 5. **Spring Framework Runtime**



Fuente: References Documentation. Spring Framework  
<http://static.springsource.org/spring/docs/3.1.2.RELEASE/spring-framework-reference/html/index.html>. Consulta: junio de 2012.

Otra característica fundamental de Spring es que orienta al programador a hacer uso de interfaces como base de la programación, con lo que se logra un máximo desacoplamiento entre clases y funcionalidades. Spring puede integrar una gran cantidad de módulos (ver figura 5) y otros *frameworks* por lo que sirve para armar la infraestructura de la aplicación, y también maneja una comunicación, transparente para el programador, entre los distintos módulos.

### 2.5.2. Persistencia de datos

Cuando se habla de la persistencia de datos, significa la parte de la programación que manejará la conexión con la base de datos. Para Java se



pueden utilizar muchas formas de lograr estas conexiones. Por ejemplo: realizarlas implementando directamente las librerías JDBC que existen para cada DBMS, lo cual hace muy tardado y poco eficiente el desarrollo de las conexiones con la base de datos.

El *framework* a utilizar se llama Java Persistence API (JPA). Es un *framework* Java que permite realizar un mapeo objeto-relacional, esto significa que se puede manipular la información de una base de datos, almacenada de forma relacional, con objetos del paradigma orientado a objetos.

Este *framework* no es más que una abstracción sobre JDBC que permite realizar el mapeo objeto-relacional de forma rápida y sencilla. También aprovecha todas las ventajas que la programación orientada a objetos provee, al utilizar estos, los cuales tienen estado y comportamiento para manipular información relacional almacenada en el formato de tablas, columnas, filas, etc. que una base de datos tiene comúnmente.

Estos objetos que se utilizan para el mapeo de la información almacenada de forma relacional, se conocen como POJO's (Plain Old Java Object) y sirven para la definición de los datos. JPA trabaja por medio de una interface implementada por un proveedor de persistencia de datos, entre los cuales se pueden encontrar: Hibernate, Elcipse Link, etc.

La persistencia en JPA involucra tres áreas:

- El Api en sí mismo.
- Un lenguaje para consultas, Java Persistence Query Language (JPQL).
- Los metadatos que realizan el mapeo objeto-relacional.

Además de las ventajas y bondades que JPA puede ofrecer, se puede disponer de su conexión directa con Spring, quien ha desarrollado un proyecto llamado Spring Data, en el cual se encuentra Spring JPA. Spring JPA agrega al Api descrito anteriormente características que harán que el desarrollo de conexiones con bases de datos y la persistencia de información sean realizadas fácil y rápidamente. Entre las principales características de Spring JPA están:

- Soporte de paginación para los resultados de las consultas.
- Ejecución de query's dinámicas.
- Creación de query's basado en el nombrado de métodos en los objetos.
- Implementación de repositorios.

### **2.5.3. Interfaz de usuario**

La creación de las interfaces gráficas para los usuarios es uno de los trabajos más costosos en el recurso de tiempo. Esta tarea conlleva muchas consideraciones, desde que la interfaz sea 100 por ciento intuitiva para el usuario, hasta validaciones de todo tipo, para evitar la inconsistencia en la información.

Java Server Faces (JSF) es la tecnología elegida en este proyecto. JSF es un *framework* para aplicaciones web Java, que simplifica el desarrollo de las interfaces de usuario. JSF utiliza las Java Server Page (JSP) para hacer el despliegue de las páginas web, agregando un conjunto de librerías de etiquetas propias para crear los elementos que componen los formularios en una página web.

Una gran ventaja que brinda JSF es el poder administrar el estado de los componentes web, manejar eventos enlazados a funciones del lado del

servidor, validar entradas, y todo esto de una forma sencilla y que ahorra mucho trabajo al desarrollador. También predefinir en archivos de configuración xml, reglas de navegación en las páginas.

El estado de los componentes, los eventos que se ejecutan, los componentes en sí mismos, pueden estar enlazados a un objeto en el servidor, a estos objetos se le llaman ManagedBeans, el cual es una instancia de una clase que se utiliza para administrar un componente de interfaz de usuario del lado del cliente. Los ManagedBeans sirven de respaldo para la vista de una aplicación y almacenan los datos a mostrar o los que se recuperaron de la interacción del usuario con el sistema.

Los ManagedBeans pueden tener distintos niveles de alcance en su ciclo de vida. Los alcances más comunes son:

- Aplicación: objetos disponibles para todas las peticiones de cualquier usuario en todas las vistas de la aplicación.
- Sesión: objetos disponibles para todas las peticiones que formen parte de una sola sesión de un usuario.
- Request: objeto disponible desde el momento que se envía una petición al servidor hasta el momento en que el servidor envía la respuesta al cliente.

JSF es un *framework* que facilita mucho el trabajo, pero si se quiere decorar aún más la vista de la aplicación se puede utilizar otras herramientas para hacerlo. Para este proyecto se utilizará PrimeFaces, que no es más que una abstracción de las JSF, es decir, que es un componente que utiliza JSF como base para proveer elementos ricos en diseño y funcionalidad al usuario final y al desarrollador.

Entre las principales características que brinda PrimeFaces está el soporte de Ajax con despliegue parcial, es decir, que se puede controlar cuáles componentes de la página se actualizarán y cuáles no. Además provee un conjunto de componentes especializados en aplicaciones web para móviles.

Hasta este momento se han visto las tecnologías en infraestructura, persistencia de datos e interfaces de usuario. Como ya se había mencionado, Spring es un *framework* que permite integrar muchos módulos especializados en las distintas áreas de una aplicación web, y las tecnologías que se han escogido, JPA, JSF, PrimeFaces, no son la excepción. Todo el proyecto del punto de venta estará gestionado en su infraestructura por Spring, incluyendo la seguridad e identificación de usuarios, a través de Spring Security.

Además de lo que se ha descrito en cada aspecto de la aplicación web, hay un común denominador en las versiones de los *frameworks* que se utilizarán. Esto es que tanto la interfaz de usuario, como la infraestructura y la persistencia de datos se van a manejar con anotaciones. Esto quiere decir que, los archivos xml de configuración en cada *framework* no serán necesarios para definir cada componente dentro de la arquitectura del software. Bastará con colocar las anotaciones correctas en las mismas clases y la inyección de dependencias de Spring funcionará de forma correcta.

#### **2.5.4. Base de datos**

Para el manejo de la base de datos se ha escogido MySQL, el cual es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Entre las principales características de MySQL están:

- Soporta los estándares de SQL e incorpora otras funcionalidades de lenguaje útiles para los usuarios.
- Se puede seleccionar el tipo de motor de almacenamiento en tiempo de ejecución. Esto a partir de la versión 5.1 en adelante.
- Transacciones y llaves foráneas: esto en algunos motores de almacenamiento.
- Replicación.
- Búsqueda e indexación de campos de texto.

Para la implementación del Punto de Venta se utilizará el motor de base de datos InnoDB. Este motor de almacenamiento de MySQL cumple con los requisitos mínimos para el sistema, como la ejecución de transacciones y las llaves foráneas en las relaciones de las tablas.

### **2.5.5. Servidor de servlets**

Como ya se había mencionado, con la implementación de Spring en el punto de venta, como un contenedor ligero de la aplicación, no es necesario tener un servidor de aplicaciones java robusto. Basta con tener un servidor de servlets en el cual se pueda incluir todas las librerías necesarias para que cada uno de los *frameworks* funcione correctamente.

Para el punto de venta se ha seleccionado Apache Tomcat. Este funciona como un contenedor de servlets, el cual implementa las especificaciones de los mismos servlets y de las Java Server Pages (JSP). Tomcat no es un servidor de aplicaciones como JBoss o JOnAS, solamente incluye el compilador Jasper que compila los JSP's convirtiéndolos en servlets.

Normalmente, Tomcat funciona en conjunto con el servidor web Apache. Como Tomcat fue escrito en código Java, para su funcionamiento no se necesita más que el sistema operativo en donde se está ejecutando disponga de la Java Virtual Machine.

### **2.5.6. Construcción del proyecto**

Al momento de construir un proyecto de software se debe tener en cuenta que muchos de los módulos que se desarrollarán, necesitan librerías, paquetes, *frameworks*, etc. para funcionar. Todas estas librerías se conocen como dependencias y es importante saber exactamente cuáles incluir en el proyecto y qué versiones se utilizarán.

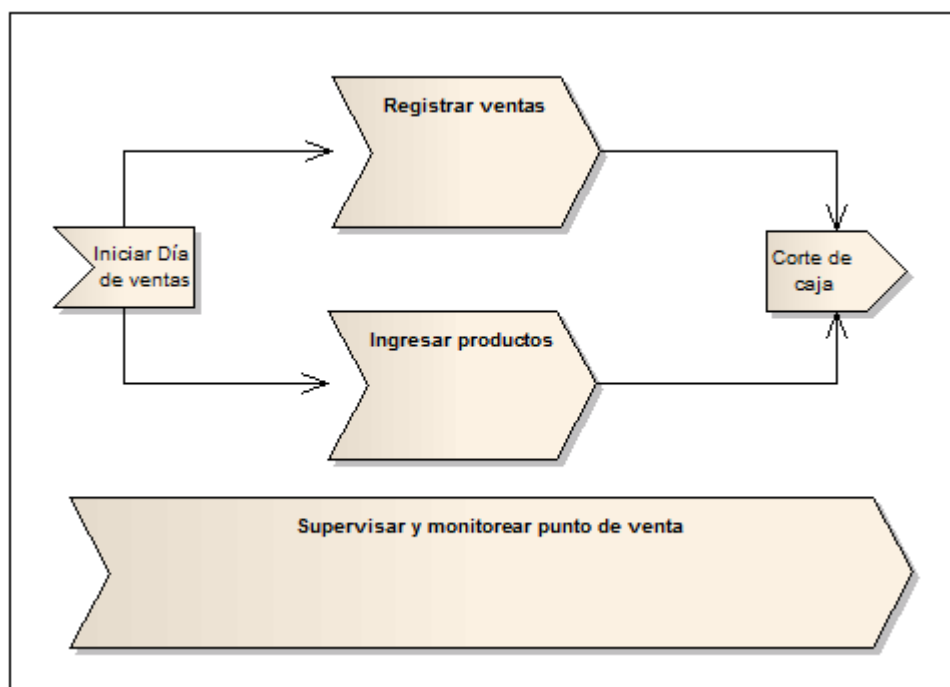
Para la construcción del proyecto de punto de venta se utilizará Apache Maven que es una herramienta para la gestión y la construcción del proyecto basada en configuraciones con archivos XML. Maven utiliza un Project Object Model (POM), para describir el proyecto que se está construyendo, las dependencias hacia otros módulos y componentes externos, y el orden de construcción de los elementos.

Maven incluye los paquetes de dependencias del proyecto a través de repositorios, lo cuales contienen las librerías que se están solicitando. La configuración de las dependencias y los repositorios respectivos en donde se encuentran almacenadas estas dependencias, son configurables a través del archivo de configuración XML de Maven.

## 2.6. Procesos y actividades

A partir de esta sección se comenzará a tratar puntos más específicos con la implementación del sistema. Lo primero que se debe definir es el proceso general del punto de venta. Como se puede observar en la figura 6, el inicio del día de ventas es el punto de partida, esto implica una o más actividades humanas o del sistema que se detallarán más adelante.

Figura 6. **Macroproceso del punto de venta**



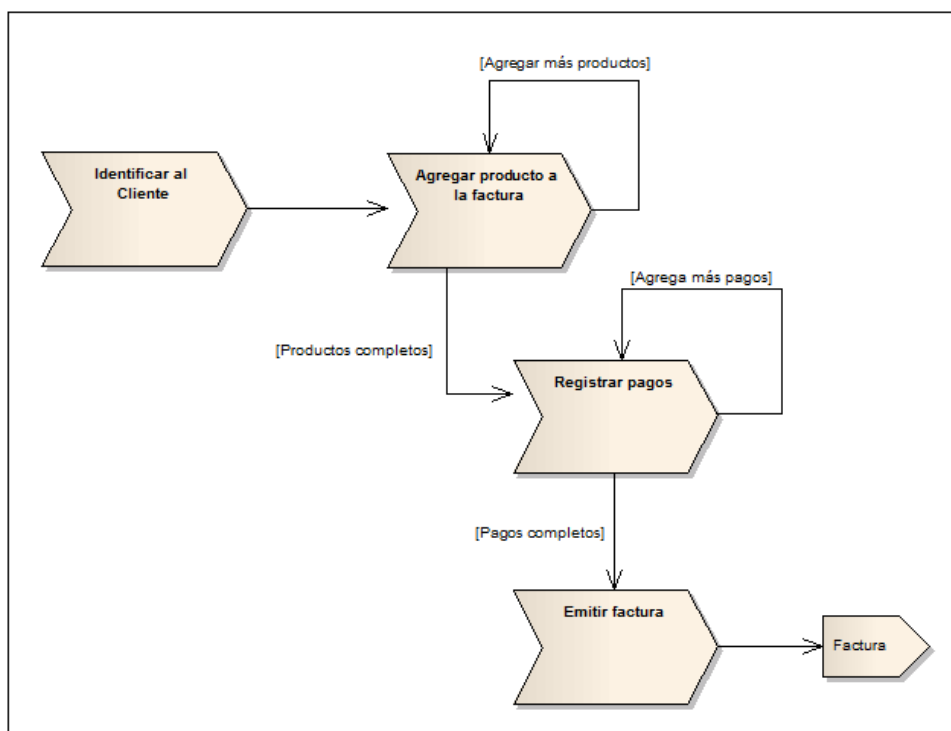
Fuente: elaboración propia.

El proceso tiene tres partes generales que lo conforman: el registro de las ventas, el ingreso de productos a inventarios, y la supervisión y monitoreo de las actividades registradas en el punto de venta. Las dos primeras partes del proceso que se mencionaron son realizadas por el usuario normal del punto de

venta, el cual se conoce comúnmente como dependiente o cajero. La tercera parte es realizada por un usuario administrador.

Al momento que un cajero inicia el día de operaciones en la caja, este es responsable de las transacciones que se registren. Esto hasta el momento que se genera el corte de caja del día, con lo cual se cierran operaciones y se deja el sistema preparado para el siguiente día.

Figura 7. **Proceso de registro de ventas**



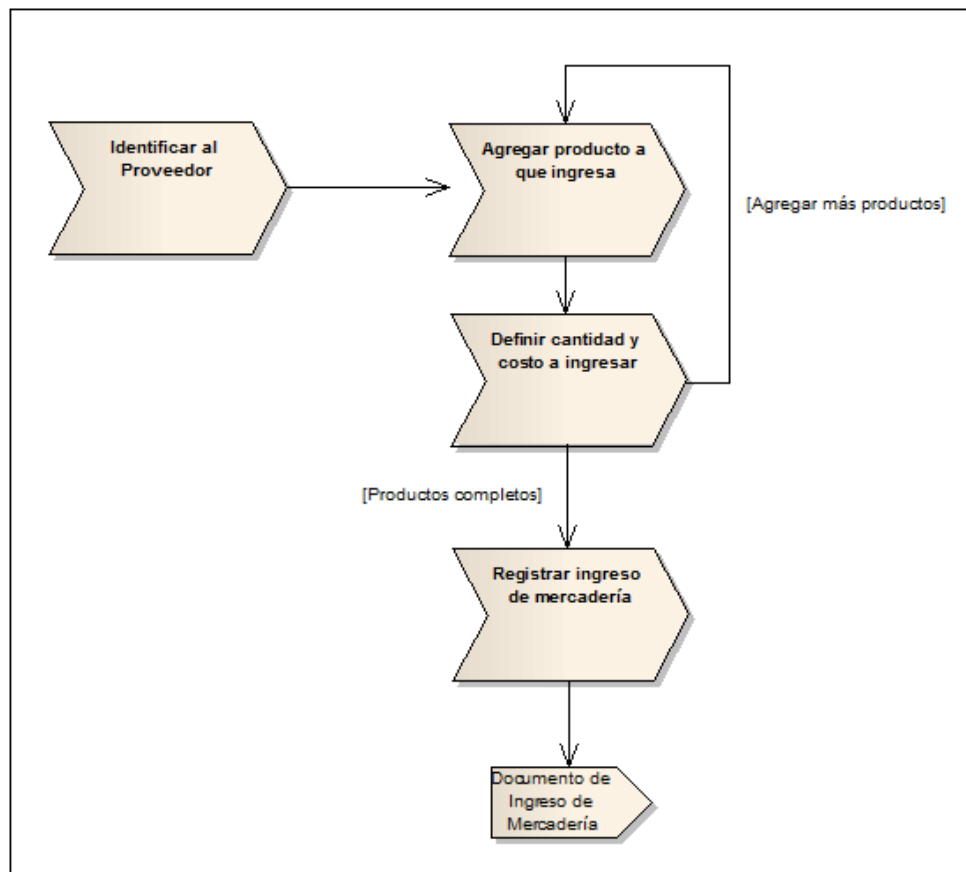
Fuente: elaboración propia.

El registro de ventas se compone de un conjunto de actividades específicas. Primero debe ingresar los datos del cliente para poderlo identificar. Cuando ya se ha identificado al cliente, se puede comenzar a agregar



productos a la compra que está realizando. La actividad de agregar productos a la compra es repetitiva indefinidamente, es decir que, el cajero puede registrar cuántos productos sean necesarios.

Figura 8. **Proceso de ingreso de mercadería**



Fuente: elaboración propia.

Esta actividad, la de registrar productos en la compra del cliente, está limitada por la cantidad de cada producto disponible en el inventario. El inventario debe estar actualizado en tiempo real. La actualización se realizará

en cada ingreso de mercadería al punto de venta, y en el registro de venta respectivamente.

Al haber terminado de ingresar todos los productos, el cajero continúa y comenzará a registrar la forma en que el cliente pagará por lo que ha consumido. Genéricamente las formas de pago que se utilizarán en este contexto son, efectivo y tarjeta de crédito/débito. Para finalizar se debe emitir la factura que se entregará al cliente, y el registro de la venta estará finalizado.

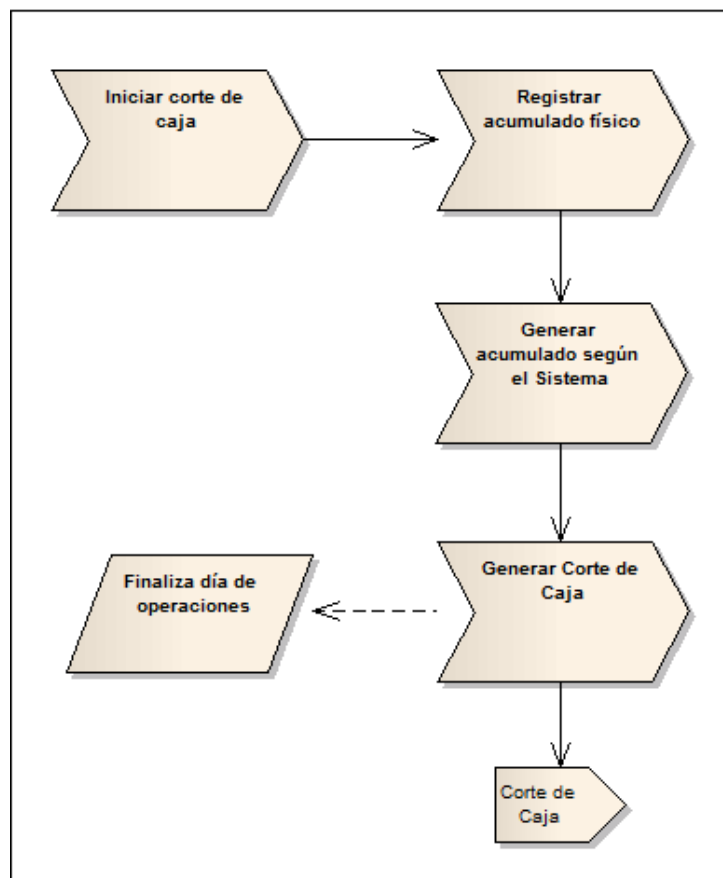
El proceso de ingreso de productos describe las actividades con las cuales el inventario del punto de venta será alimentado, y por consiguiente, los productos estarán disponibles para la venta. Cuando se registra el ingreso de productos, el inventario de cada uno será incrementado en la cantidad que se está ingresando, respectivamente.

Al igual que en el proceso de registro de ventas, se deben agregar los productos que ingresarán al inventario. Este proceso es repetitivo por cada producto que se desee ingresar. Para cada registro de producto se define la cantidad que ingresará al inventario del punto de venta. Al tener completa la lista de productos, se procede a confirmar el ingreso.

Este proceso genera un documento como constancia del ingreso de los productos al inventario del punto de venta. En este documento se hace constar la fecha y hora del ingreso, el punto de venta a donde ingresaron, el cajero/dependiente responsable del registro de ingreso de productos. Además incluye el listado detallado de cada producto y las cantidades que ingresaron al inventario.

El proceso de corte de caja es el que finalizará el día de operaciones para el punto de venta. El cajero deberá registrar los montos de dinero que ingresaron en el día para cada forma de pago que el punto de venta acepta. Para este caso se tomará efectivo y tarjeta de crédito/débito. El sistema generará el acumulado de efectivo y de tarjetas de crédito/débito para poder generar el reporte de corte de caja, que incluye lo reportado físicamente por el cajero, lo calculado por el sistema y la diferencia entre ambos montos, tanto para efectivo como tarjeta de crédito/débito.

Figura 9. **Proceso de corte de caja**



Fuente: elaboración propia.

El corte de caja, en un punto de venta, puede ser parcial o total. En este sistema de punto de venta se manejará únicamente el corte de caja total. Esto quiere decir que para un día de operación habrá solamente un corte de caja, el cual marcará el fin del día. Es muy importante resaltar que si se registra el corte de caja del día, ya no se pueden realizar operaciones de facturación en ese día, sino que se debe esperar hasta el siguiente para que el sistema habilite nuevamente esta opción.

Uno de los propósitos esenciales del desarrollo de este sistema, como ya se ha mencionado, es proveer una herramienta de toma de decisión a los puestos de Gerencia, en relación al comportamiento de las ventas, del movimiento de mercadería, entre otros. Es por esto que el proceso de supervisión y monitoreo es fundamental.

En este proceso no se tiene una secuencia de actividades establecidas, sino más bien un conjunto de reportes que reflejan la información generada por los procesos anteriores, ordenada de la mejor forma para brindarle al usuario conocimiento acerca del punto de venta implementado. En estos reportes se busca proveer una interpretación cuantitativa básica de todos los datos generados por el proceso de venta e inventarios.

Como se puede observar en el macroproceso, la supervisión y el monitoreo es un proceso que se extiende en todo el ciclo del punto de venta, es decir, que son actividades continuas que se pueden realizar en cualquier momento, sin importar en qué actividad se encuentren los demás procesos. El conjunto de reportes que pertenecen a este proceso se dividen en dos grandes grupos: los reportes de monitoreo y los de estadísticas.

Tanto los reportes de monitoreo y los estadísticos pueden ser consultados en cualquier momento y desde cualquier lugar, obteniendo resultados en tiempo real, es decir, completamente actualizados hasta el mismo instante de consulta de los reportes. Los de monitoreo se enfocan en el detalle de cuál es el movimiento diario en cada uno de los puntos de venta. Los reportes de estadística son resúmenes de la información agrupada en intervalos de tiempo desde semanas hasta años, mostrando datos importantes para la toma de decisión dentro de una empresa, así como representaciones gráficas de dicha información.

## **2.7. Esquema general del producto**

El Sistema de Punto de Venta que se desarrollará, estará conformado por tres secciones principales:

- Administración y configuración
- Operación
- Reportes y estadísticas

La administración y configuración, como su nombre lo indica, contiene todas las opciones para que el administrador del sistema pueda gestionar los diferentes catálogos involucrados en el sistema de punto de venta. En esta sección, el término catálogo no se refiere exactamente al de productos que tiene el punto de venta, sino más bien, a los generales que se involucran en todo el proceso del punto de venta. Estos son los siguientes:

- Clientes
- Tiendas
- Categorías de productos

- Productos
- Resoluciones de facturas
- Tipo de tarjeta de crédito/débito
- Descuentos

Otras opciones que se encuentran en la sección de administración son:

- Definición de precios
- Asignación de empleados a tiendas
- Asignación de descuentos a productos

En la sección de operación, que es la base fundamental del punto de venta se tienen las siguientes funciones:

- Iniciar día de operaciones
- Consulta de inventario
- Facturar
- Ingresar mercadería
- Generar corte de caja

Estas funciones de operación, es la sección que manejará el cajero directamente en el punto de venta. Todas estas pueden ser monitoreadas por parte del gerente en cualquier momento.

Los reportes del sistema de punto de venta se basarán tanto en dar información detallada de cada operación, como en mostrar resúmenes y estadísticas a lo largo del tiempo. Los reportes que se desarrollarán serán los siguientes:

- Historial de ingresos de mercadería
- Historial de movimientos de mercadería
- Historial de facturación
- Historial de cortes de caja
- Estadística de ventas por tienda
- Comportamiento de ventas por tienda
- Reporte de ventas por categoría/producto
- Reporte de utilidades

Una vez listadas todas las opciones que tendrá el sistema, se ordenan estas opciones, como un menú de aplicación basado en los distintos roles que tendrá el sistema:

- Rol del administrador
  - Clientes
    - Creación
    - Modificación
  - Tiendas
    - Creación
    - Modificación
  - Resoluciones de facturas
    - Gestión de resoluciones

- Productos
  - Categorías
    - ✓ Creación
    - ✓ Modificación
  - Productos
    - ✓ Creación
    - ✓ Modificación
    - ✓ Asignación de precios
- Descuentos
  - Creación
  - Modificación (activación/desactivación)
  - Asignación de descuentos a productos
- Tipo de tarjeta de crédito/débito
  - Creación
  - Modificación
- Rol del cajero
  - Iniciar día de operaciones
  - Mercadería



- Ingreso de mercadería
  - Consultar inventario
- Facturación
  - Facturar
- Cierre del día
  - Generar corte de caja
- Rol del gerente
  - Mercadería
    - Inventario
      - ✓ Inventario general
      - ✓ Inventario por tienda
    - Historial de Ingresos de mercadería
    - Historial de movimientos de mercadería
  - Operaciones de caja
    - Historial de facturación
    - Historial de cortes de caja
  - Reportes y estadísticas

- Ventas por tienda
  - ✓ Estadística de ventas por tienda
  - ✓ Comportamiento de ventas por tienda
  
- Ventas por producto
  - ✓ Reporte de ventas por categoría/producto.
  - ✓ Reporte de utilidades

### **3. IMPLEMENTACIÓN DE UN PUNTO DE VENTA EN LA NUBE**

En el primer capítulo se describieron los conceptos fundamentales de un punto de venta y lo que significa, en términos de tecnologías de la información, la nube. También se explicó lo que implica que un sistema funcione en la nube. En el segundo capítulo se describen las tecnologías que se utilizarán en este desarrollo, y también la lógica de negocio del punto de venta a través de procesos y actividades.

En este capítulo se inicia el desarrollo del sistema de punto de venta en la nube, desde el análisis del sistema, hasta el despliegue funcional del punto de venta para las pruebas de rendimiento. Existen muchas metodologías de desarrollo de software, cada una con ventajas y desventajas. Estas metodologías se clasifican, principalmente en dos grandes categorías: metodologías tradicionales (pesadas), y metodologías ágiles.

Las metodologías tradicionales se basan en la definición de un proceso robusto y la ejecución del mismo al pie de la letra. Invierten mucho esfuerzo en las primeras fases del desarrollo para definir en su totalidad el sistema. Generar una gran cantidad de documentación, la cual se usa como base para la implementación y codificación. Estas metodologías dan mucha importancia a las fases de análisis y diseño, y no se puede avanzar a otra fase del desarrollo hasta que la anterior esté completamente terminada.

Las metodologías ágiles se fundamentan en cuatro pilares: promueven al individuo y a sus interacciones antes que los procesos y herramientas, priorizan entregas de software funcional sobre la documentación, la confianza y

colaboración con el cliente son más importantes que las negociaciones contractuales y se prefiere tener una respuesta al cambio antes que seguir al pie de la letra un plan.

En las metodologías ágiles, la documentación es más escasa y menos elaborada, pero se promueve el desarrollo y entrega de software en iteraciones, brindando una respuesta rápida al cambio por parte del cliente.

La decisión de utilizar una u otra metodología depende de las preferencias de quién toma esta decisión, también del entorno empresarial en el que se desarrolle el proyecto. Para el desarrollo de este sistema se tomará ventajas de ambos tipos de metodologías, generando una metodología híbrida. El enfoque será un análisis y diseño documentado no tan extenso, pero que brinde las guías para la implementación del punto de venta como un sistema en la nube.

### **3.1. Análisis del producto**

El análisis de sistema de información es un conjunto de procedimientos y/o programas en los cuales se ejecutan determinadas acciones con objetivos específicos, entre los cuales se pueden mencionar: identificar las necesidades del cliente, definir el dominio de la información, definir las funciones que debe realizar el sistema, representar el comportamiento del sistema, entre otros.

#### **3.1.1. Toma de requerimientos**

Un requerimiento es una definición o una declaración abstracta de alto nivel que representa un servicio que debe ser proporcionado por un sistema (software). Se dice que es de alto nivel porque se plantea en función del negocio y no en función a aspectos técnicos para desarrollar el sistema. El

proceso de toma de requerimientos es de suma importancia, pues es aquí en donde el cliente plasma todas sus necesidades y la forma en la que él espera que sean resultas.

Los requerimientos son la base de la cual parte el desarrollo de un sistema de software; con ellos se pueden realizar estimaciones de tiempo, esfuerzo y recursos necesarios, se pueden elaborar cronogramas, planes de trabajo, cartas de compromiso, entre otros muchos documentos. Es el punto de partida y de acuerdo entre las dos partes involucradas, cliente y desarrollador.

Otro punto por lo que los requerimientos son tan importantes es que se convierten en los parámetros para comprobar y verificar si se obtuvieron los resultados esperados y si se alcanzaron los objetivos planteados para el desarrollo del sistema.

Los requerimientos se pueden dividir en tres clasificaciones: funcionales, no funcionales y de implementación. Un requerimiento funcional describe las interacciones entre el sistema y su entorno, usuarios, entre otros. Un requerimiento no funcional es el que describe detalles o restricciones del sistema, pero que no están involucrados con el funcionamiento del mismo. Un requerimiento de implementación está involucrado directamente con la operación del sistema.

Cuando se habla del proceso de toma de requerimientos para el desarrollo de un sistema, necesariamente hay que hablar de entrevistas con el cliente. Solamente de una forma personal se puede llegar a comprender mejor las necesidades que el cliente tiene y cuáles son sus expectativas respecto de las soluciones que el sistema le brindará. Para el sistema de punto de venta en la nube no fue la excepción, por lo que la toma de requerimientos se basó en

entrevistar a dos expertos en la administración de puntos de venta, tres entrevistas a cada uno.

Por el hecho de que en este proyecto se quiera proveer de una herramienta para el desarrollo de las pymes en Guatemala, lo que se hizo fue tomar los requerimientos y las ideas planteadas por estos dos expertos, y se realizó una abstracción para poder obtener los requerimientos de un punto de venta genérico. Luego de este proceso se obtuvo la siguiente lista de requerimientos funcionales descritos en la tabla II, y requerimientos no funcionales en la tabla III.

Tabla II. **Requerimientos funcionales sistema de punto de venta**

No.	Área	Requerimiento
1	Caja	El cajero debe iniciar el día con una cantidad de efectivo en la caja disponible para vueltos.
2	Caja	El cajero debe registrar la mercadería que entra a su tienda. Al registrar la mercadería entrante automáticamente se debe actualizar el inventario de productos disponible para la venta.
3	Caja	Cuando ingresa un producto al inventario, se debe calcular su costo con la regla de promedio ponderado.
4	Caja	El cajero solamente puede vender el producto que tiene disponible en su inventario.
5	Caja	El cajero puede, en cualquier momento, consultar el inventario que tiene disponible en la tienda.
6	Caja	En el registro de la venta, se debe especificar la forma en la que se paga la factura, efectivo o tarjeta de crédito/débito.

Continuación de la tabla II.

7	Caja	El monto de lo que paga debe coincidir automáticamente con el total de la factura.
8	Caja	Los productos que son registrados en una factura, deben ser descontados automáticamente del inventario de la tienda.
9	Caja	El cajero puede aplicar un descuento por cada producto agregado a la factura. Estos descuentos deberán ser activados por el administrador del sistema.
10	Caja	El cajero no puede consultar en el momento, el historial de todas las facturas que ha emitido.
11	Caja	Al finalizar el día el cajero debe registrar el total de efectivo que ha recaudado, y la suma de todos los comprobantes de transacciones con tarjetas de crédito/débito.
12	Caja	El cajero no puede ver el total de ventas, ni el total de efectivo y de tarjetas de crédito/débito recaudado según el sistema, sino hasta después de haber reportado lo que él ha contado físicamente.
13	Caja	El sistema debe hacer una sumatoria de las formas de pago y generar un Corte de Caja, mostrando lo que el sistema calculó y lo que el cajero ingresó.
14	Caja	Al generar el corte de caja, el sistema debe notificar al gerente vía correo electrónico, y colocar el corte de caja como pendiente de revisión/aprobación.
15	Caja	En el sistema se manejarán solamente cortes de caja totales, más no se manejarán cortes de caja parciales. Esto implica que al generar el corte de caja el sistema ya no debe permitir registrar operaciones de venta, hasta el siguiente día calendario.
16	Gerencia	El gerente será el encargado de revisar el historial de facturación de cada tienda.

Continuación de la tabla II.

17	Gerencia	El sistema debe tener un registro de los cortes de caja que no han sido revisados. La revisión debe ser efectuada por el gerente, quien podrá agregar una nota de observaciones en cada corte de caja, además de marcarlos como ya revisados.
18	Gerencia	Se debe contar con un historial de los cortes de caja de todas las tiendas.
19	Gerencia	En el sistema se debe generar el libro de ventas, en el cuál se detalle el total facturado por día en cada tienda, así como las sumatorias totales de ventas. También debe expresar la cantidad de facturas emitidas. Este reporte debe poder generarse por mes.
20	Gerencia	Las estadísticas que debe presentar el sistema son: ventas diarias por cada tienda. Cantidad vendida de cada producto en cada tienda, mensualmente. Monto total de las compras efectuadas por cada cliente.
21	Gerencia	El sistema debe proveer un histórico del movimiento de cada producto en todas las tiendas, entradas y salidas de productos.
22	Gerencia	El gerente debe tener acceso al inventario general y a los inventarios individuales de todas las tiendas, además de poder exportar este inventario a un archivo .xls
23	Administración	El sistema debe ser multi-sucursal. Se manejará una caja por tienda.
24	Administración	Se deben configurar las resoluciones de facturas para que los cajeros puedan emitir las facturas. Estas resoluciones son para referencia, pues el sistema no emitirá las facturas, sino que solamente imprimirá el contenido de las mismas en formatos preimpresos.



Continuación de la tabla II.

25	Administración	Se debe poder dar de alta y baja a cajeros, así como asignar a un cajero específicamente a una y solo una tienda. Esto es necesario para que cuando un cajero entre al sistema, automáticamente esté ubicado en la tienda que le corresponde.
26	Administración	El sistema debe proveer un mecanismo de clasificación de productos por medio de categorías. Se pueden crear, modificar y dar de baja a los productos.
27	Administración	El sistema debe manejar descuentos sobre productos. Estos descuentos deben tener una fecha de inicio y una fecha de fin, las cuales determinarán el rango de tiempo en el cuál estarán disponibles los mismos.
28	Administración	Los descuentos deben ser activados directamente sobre un producto y una tienda específicos.
29	Administración	En el sistema se debe poder modificar el precio de venta de los productos. El precio de venta será un atributo del producto.

Fuente: elaboración propia.

Tabla III. **Requerimientos no funcionales sistema de punto de venta**

No.	Requerimiento
1	El sistema debe contar con mecanismos de seguridad para la identificación de los usuarios. Los usuarios se deben autenticar por medio de un nombre de usuario y un password.
2	Los passwords de los usuarios deben estar encriptados.

Continuación de la tabla III.

3	Toda acción realizada en el sistema debe contar con un registro en la bitácora de ejecución. Se debe poder trazar y rastrear las acciones y transacciones realizadas en el sistema.
4	En cualquier momento que el usuario ejecute una acción sobre el sistema, se debe hacer notar al usuario que el sistema está procesando la información. Mientras se ejecuta el procesamiento de la información, se debe mantener bloqueada la pantalla para no realizar alguna otra acción.
5	Se debe manejar los errores y excepciones de forma segura y entendible para el usuario.
6	El sistema tiene que contar con notificaciones intuitivas, pues esta será la forma en que el sistema se comunice con el usuario. Toda acción debe contar con notificaciones de error, precaución o éxito en el procesamiento de la información.
7	El sistema tiene que cuidar la integridad de la información. Debe ser transaccional.
8	El sistema tiene que manejar acceso concurrente, pues cada cajero estará utilizando la misma aplicación del servidor, pero cada uno debe contar con su propia sesión.

Fuente: elaboración propia.

Ahora que ya se tienen los requerimientos del punto de venta, se procede con la siguiente etapa del análisis: la construcción de los diagramas de casos de uso.

### 3.1.2. Casos de uso

Un diagrama de caso de uso servirá para representar de una manera gráfica la interacción que tendrá el usuario con el sistema en cuestión. Dicho de

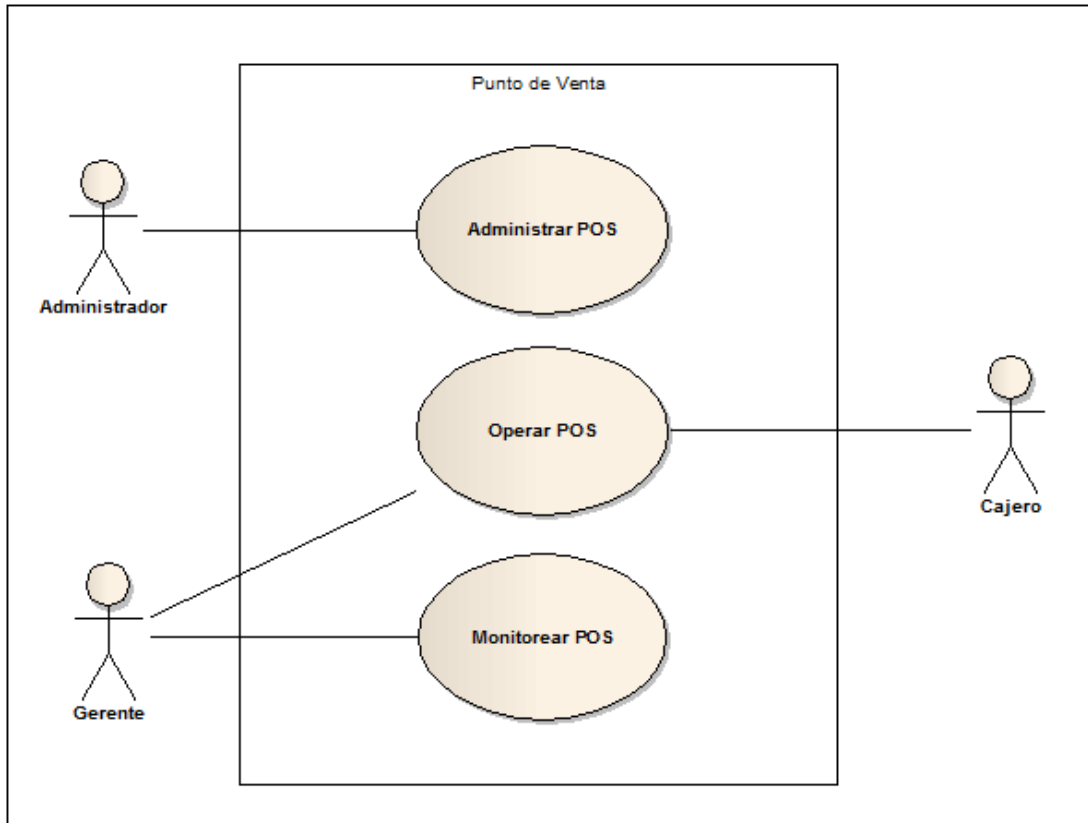
otra forma los diagramas de casos de uso representan el comportamiento de un sistema desde el punto de vista del usuario. Con los diagramas de casos de uso se pueden representar cuáles serán las características del sistema y cómo se va a interactuar con el mismo, para que los interesados puedan validarlo con el proceso del negocio.

Un caso de uso determina los requerimientos funcionales de un sistema, especifica un comportamiento deseado, más no como se llevará a cabo ese comportamiento. Los casos de uso deben responder el qué pero no el cómo del sistema. Está más orientado al negocio que a los aspectos técnicos de la implementación del sistema, es por esto que son especialmente útiles para la comunicación con el cliente.

Como primer paso se va a identificar a los actores que utilizarán el sistema de punto de venta. En la toma de requerimientos se pueden obtener tres actores: cajero, gerente y administrador, cada uno de ellos con tareas bien específicas, generalizadas en el diagrama de caso de uso general, ver la figura 10.

El sistema de punto de venta se puede resumir en tres comportamientos fundamentales: administrar el POS, operar el POS y monitorear el POS. En estos tres comportamientos se puede resumir todos los requerimientos descritos anteriormente. Este diagrama es bastante generalizado y abstracto, por lo que, lo siguiente será profundizar en cada uno de esos comportamientos.

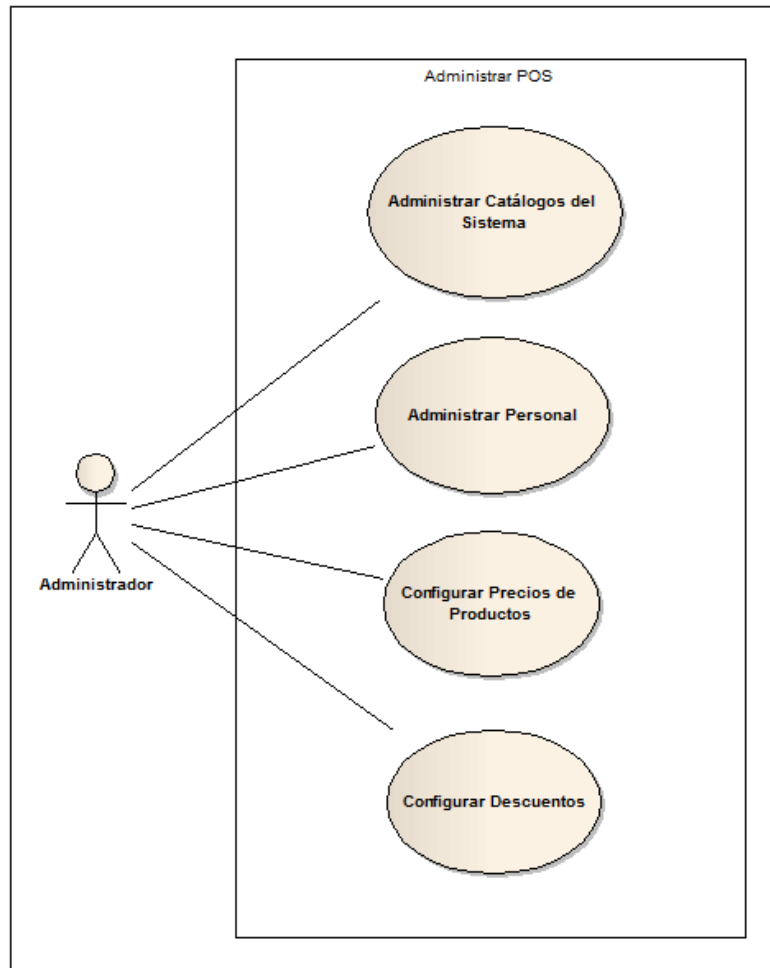
Figura 10. Diagrama de caso de uso general



Fuente: elaboración propia.

En el proceso de administrar el POS se tienen cuatro funcionalidades específicas: administrar los catálogos del sistema, administrar el personal, configurar precios de productos y configurar descuentos. En la administración de los catálogos del sistema, al ser una funcionalidad abstracta, se derivan dos diagramas más: uno en donde se define lo que incluye la administración de los catálogos y otro en donde se define qué catálogos se extienden de este comportamiento abstracto.

Figura 11. Diagrama de caso de uso administrar POS

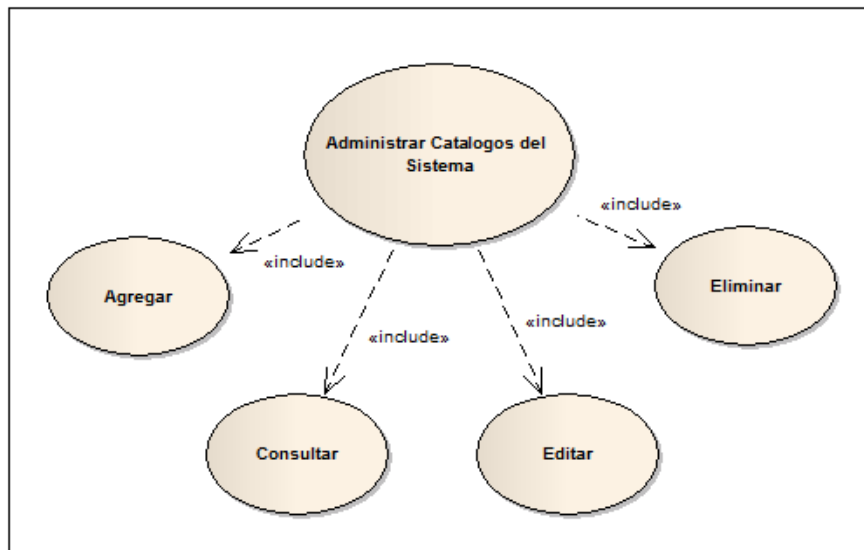


Fuente: elaboración propia.

El comportamiento administrar catálogos del sistema incluye: agregar, consultar, editar, eliminar (ver figura 12), esto quiere decir que todos los comportamientos que extiendan de administrar catálogos del sistema, deberán incluir estos cuatro comportamientos. Los comportamientos que extienden de administrar catálogos del sistema son los que administraran los siguientes

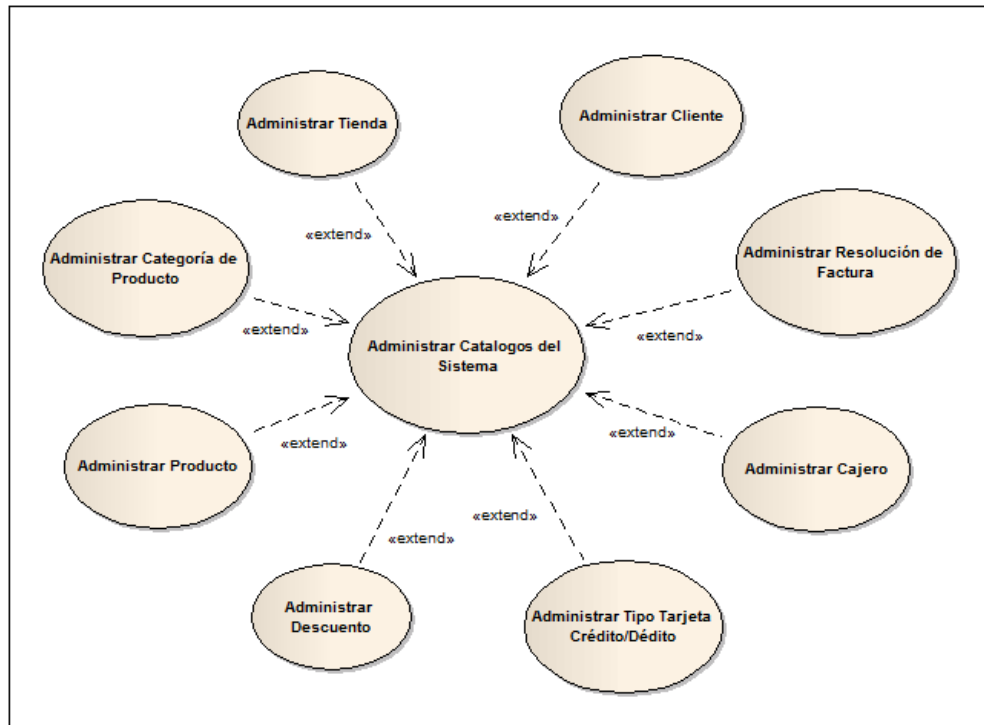
recursos: categoría de productos, productos, descuento, tipo de tarjeta débito/crédito, cajero, tienda, resolución de factura y cliente (ver figura 13).

Figura 12. **Inclusiones caso de uso administrar catálogos del sistema**



Fuente: elaboración propia.

Figura 13. Herencia caso de uso administrar catálogos del sistema



Fuente: elaboración propia.

El caso de uso administración de personal define el comportamiento para asignar los usuarios cajeros a las tiendas previamente creadas. En la administración del POS también se encuentra la asignación de precios a los productos, y la asignación de descuentos a cada uno de los productos. Esto se puede ver especificado en la descripción de los casos de uso con ID 1, 2 y 3 (ver tablas IV, V y VI).

Tabla IV. **Caso de uso administrar personal**

<b>ID</b>	<b>1</b>
<b>Nombre</b>	Administrar personal
<b>Pre- condiciones</b>	Se debe haber creado las tiendas y los cajeros en sus respectivos catálogos.
<b>Post- condiciones</b>	El cajero estará asignado a una y solo una tienda para que pueda operar.
<b>Actores</b>	Administrador
<b>Descripción</b>	Asignación de cajeros a las respectivas tiendas.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema le despliega el listado de tiendas disponibles.</li> <li>2. El administrador selecciona una tienda del listado provisto.</li> <li>3. El sistema muestra un resumen con los cajeros ya asignados a la tienda seleccionada, y una lista con los cajeros no asignados a la tienda seleccionada para que puedan ser asignados.</li> <li>4. El administrador selecciona uno o más cajeros y los asigna.</li> <li>5. El sistema almacena las asignaciones.</li> </ol>
<b>Flujos alternativos</b>	
<b>Reglas de negocio</b>	<p>Un cajero solo puede estar asignado a una tienda. Si se asigna a una tienda diferente, se debe desasignar de la tienda en la que se encuentra asignado actualmente.</p> <p>Si un cajero es dado de baja, se debe eliminar la asignación automáticamente.</p>
<b>Excepciones</b>	No se puede asignar a un cajero que haya sido de baja en el sistema.

Fuente: elaboración propia.



Tabla V. **Caso de uso configurar precios de productos**

<b>ID</b>	2
<b>Nombre</b>	Configurar precios de productos
<b>Pre- condiciones</b>	Se debe haber creado los productos en su respectivo catálogo.
<b>Post- condiciones</b>	Un producto no estará disponible para la venta, hasta que se le haya definido un precio de venta.
<b>Actores</b>	Administrador
<b>Descripción</b>	Definición de precios de venta a cada producto.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra un panel de búsqueda para los productos.</li> <li>2. El administrador completa los datos y realiza el filtrado</li> <li>3. El sistema muestra el listado de los productos que coinciden con la búsqueda.</li> <li>4. El administrador selecciona el producto que desea configurar.</li> <li>5. El sistema muestra los detalles del producto. Es importante que se muestre el costo, y que en el campo en donde se coloque el precio cargue el precio actual del producto.</li> <li>6. El administrador completa el campo de precio y envía la información.</li> <li>7. El sistema almacena el precio del producto seleccionado.</li> </ol>
<b>Flujos alternativos</b>	<p>6.1 El valor del precio no es mayor a 0.00</p> <p>6.2 El sistema muestra un mensaje informado el error. Se regresa al punto 5.</p>
<b>Reglas de negocio</b>	<p>En el filtro de búsqueda no deben aparecer los productos que han sido dados de baja.</p> <p>El precio de los productos debe ser mayor a 0.00</p>
<b>Excepciones</b>	

Fuente: elaboración propia.

Tabla VI. **Caso de uso configurar descuentos**

<b>ID</b>	3
<b>Nombre</b>	Configurar descuentos
<b>Pre- condiciones</b>	Se debe haber creado los productos y los descuentos en sus respectivo catálogos.
<b>Post- condiciones</b>	El cajero puede aplicar un descuento específico a un producto específico si y solo si el descuento ha sido asignado al producto y se encuentra en el rango de fecha definido en la asignación.
<b>Actores</b>	Administrador
<b>Descripción</b>	Asignar descuentos a productos en un rango de fecha específico.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El sistema muestra un panel de búsqueda para los productos.</li> <li>2. El administrador completa los datos y realiza el filtrado</li> <li>3. El sistema muestra el listado de los productos que coinciden con la búsqueda.</li> <li>4. El administrador selecciona el producto al que desea asignarle un descuento.</li> <li>5. El sistema muestra un resumen de los descuentos asignados al producto seleccionado, y un listado de todos los descuentos disponibles para la asignación.</li> <li>6. El administrador selecciona un descuento.</li> <li>7. El sistema muestra dos campos de fecha. Uno para el inicio y otro para el final de la disponibilidad del descuento sobre el producto.</li> <li>8. El administrador selecciona las fechas y realiza la asignación.</li> <li>9. El sistema valida los datos para la asignación.</li> <li>10. El sistema almacena la información del descuento y el producto.</li> </ol>

Continuación de la tabla VI.

<b>Flujos alternativos</b>	9.1 Las fechas no representan un rango de fecha válido. 9.2 El sistema muestra un mensaje informando del error. Se regresa al punto 7.
<b>Reglas de negocio</b>	En el filtro de búsqueda no deben aparecer los productos que han sido dados de baja.
<b>Excepciones</b>	

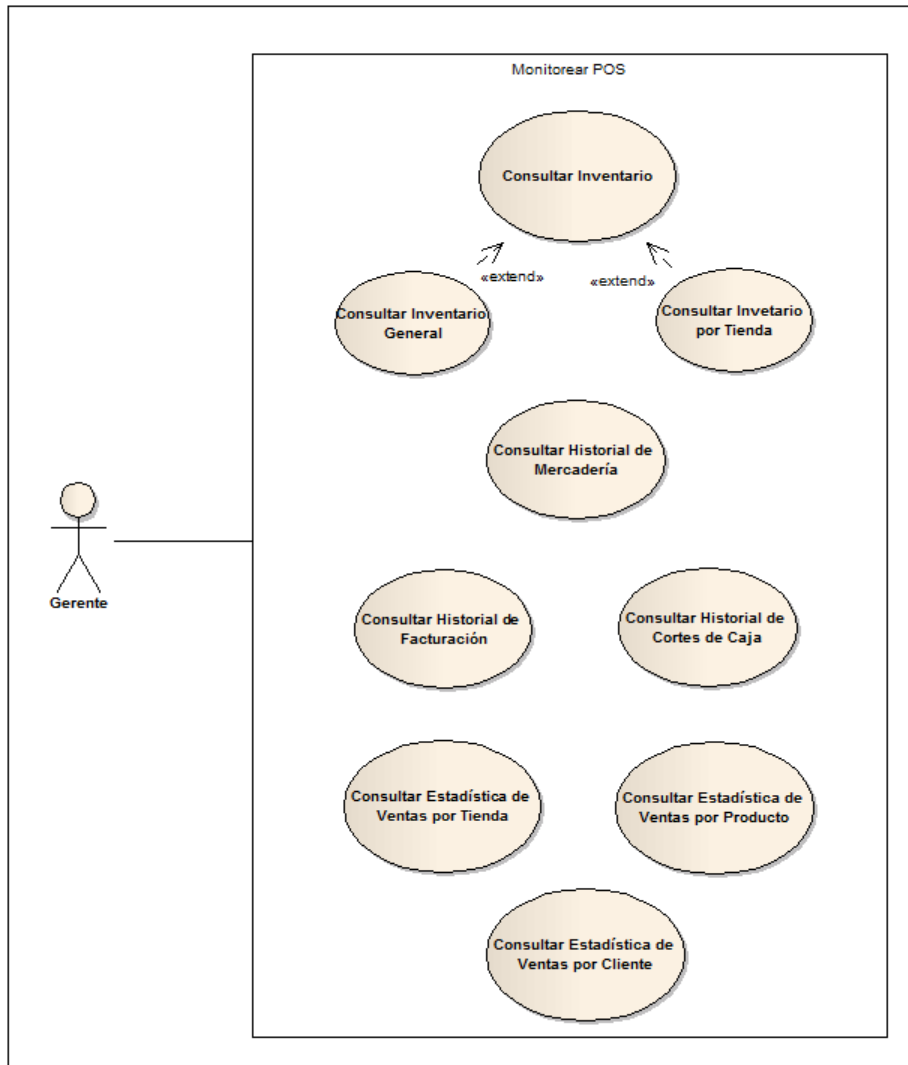
Fuente: elaboración propia.

El monitoreo del POS engloba un conjunto de reportes que darán información totalmente actualizada de las operaciones que se realicen en el POS. Estos reportes se pueden dividir en dos grandes grupos: uno para el seguimiento diario de cada una de las operaciones, y otro para los datos estadísticos y sumatorias de diferentes criterios de evaluación.

Entre los reportes de seguimiento definidos están: inventario general y por tienda, historial de movimiento de mercadería, historial de facturación e historial de cortes de caja. Entre los reportes de estadísticas definidos están: estadística de ventas por tienda, estadística de ventas por producto, estadística de ventas por cliente.

Como se puede observar, el caso de uso monitorear POS está conformado por comportamientos de consulta, por lo que no se profundizará en cada uno de ellos. En la figura 14 se pueden observar los comportamientos que conforman el caso de uso monitorear POS.

Figura 14. Diagrama de caso de uso monitorear POS

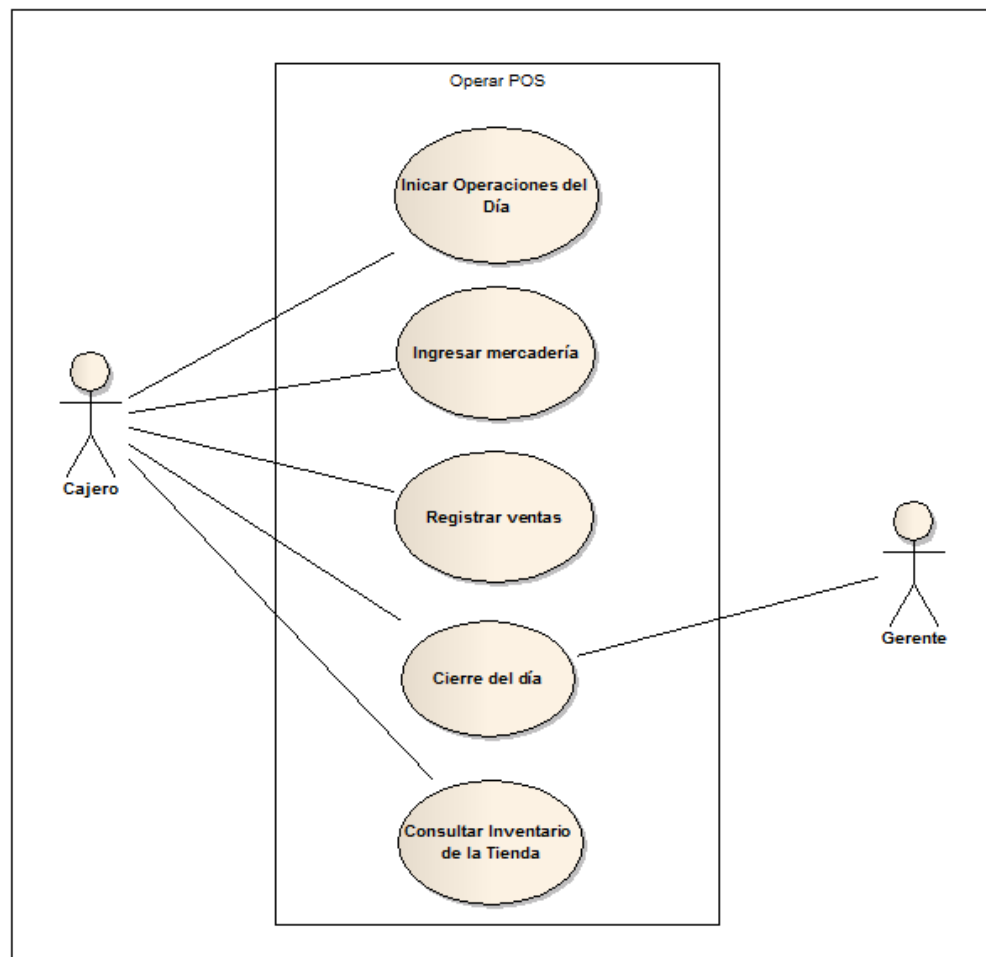


Fuente: elaboración propia.

La operación del POS es el caso de uso más crítico de todo el sistema, pues con base a lo que acá se defina, así será el comportamiento del mismo. En este caso de uso participa el cajero, pues es quién generará todas las operaciones que alimentarán al sistema. Hasta el momento se ha visto la

administración, la configuración, los reportes, pero no se ha especificado la operación fundamental del POS.

Figura 15. Diagrama de caso de uso operar POS



Fuente: elaboración propia.

El proceso de operaciones se basa en cuatro comportamientos principales: iniciar operaciones del día, ingresar mercadería, registrar ventas y cerrar el día de operaciones. Además de estos cuatro comportamientos, el cajero también puede consultar el inventario actual de la tienda a la que está

asignado. En la operación del POS, el gerente está involucrado en el cierre del día de operaciones, quien revisa y valida el corte de caja de cada tienda.

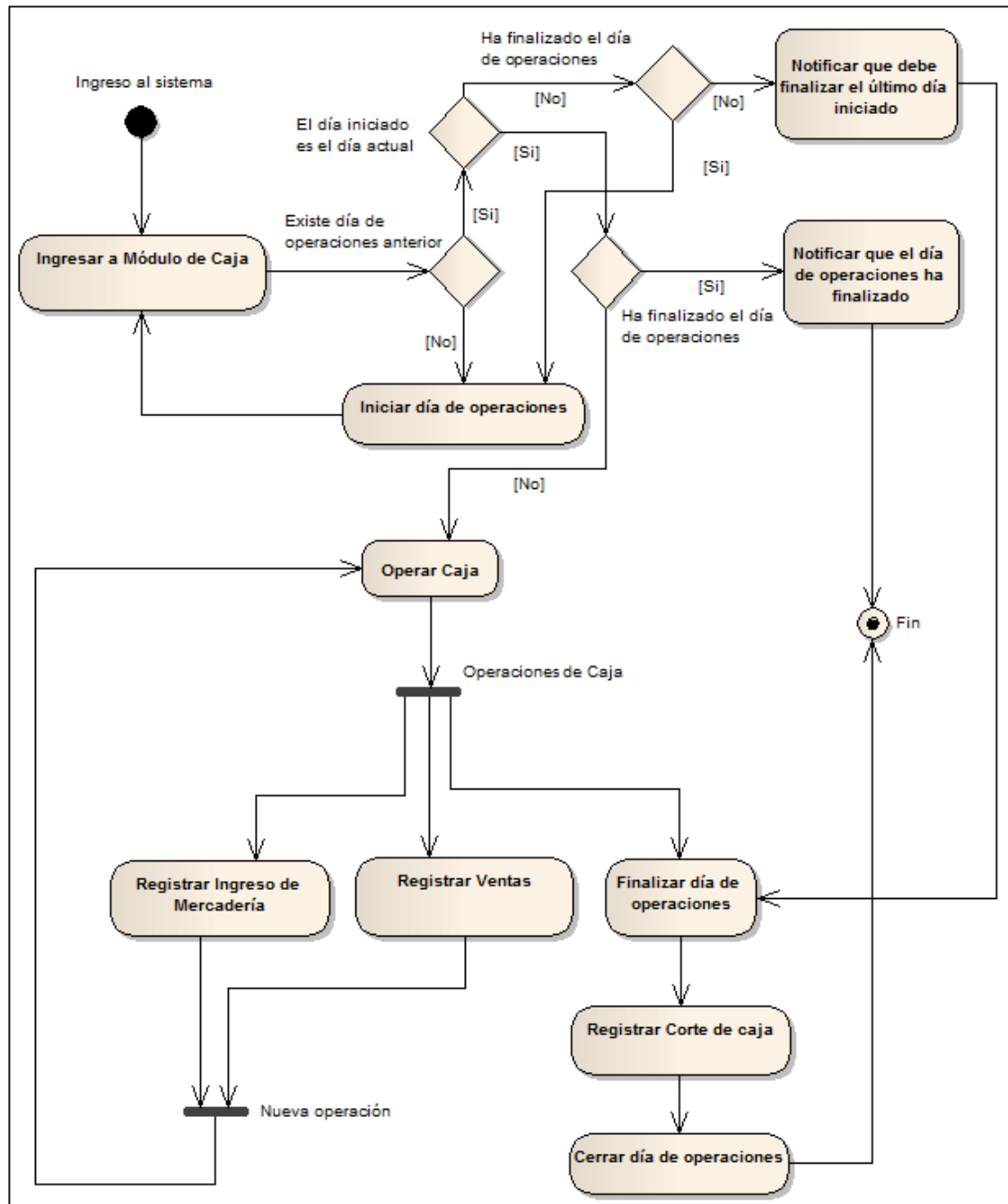
En la figura 15 se puede observar el caso de uso operar POS, en cual se especifica cada uno de los comportamientos que lo componen. En esta sección no se profundizará en los casos de uso detallados, pero como ya se ha dicho, este caso de uso es el más importante del sistema, a este se le dedicará la siguiente sección por completo: dinámica del producto.

### **3.1.3. Dinámica del sistema**

Ya se han analizado los requerimientos y planteado los casos de uso que determinan el comportamiento del sistema. Ahora se profundizará en el caso de uso operar POS, describiendo paso a paso la dinámica de cada uno de los comportamientos que lo conforman. Este trabajo busca definir las bases de un sistema de punto de venta como servicio en la nube, por lo que en esta sección se hará énfasis específicamente en los casos de uso de la operación del POS.

Cuando un cajero ingresa al sistema, automáticamente se debe verificar a que tienda está asociado, y almacenar los datos del cajero y de la tienda en la sesión activa porque se utilizarán en todo el proceso de operación del POS. Esta primera verificación debe ser transparente para el usuario, el cual únicamente visualizará un mensaje de error, notificándole que no puede operar porque no está asignado a ninguna tienda, o simplemente tendrá acceso a todas las opciones asignadas para ese rol.

Figura 16. Dinámica general del caso de uso operar POS



Fuente: elaboración propia.

Ya con los datos del cajero y la tienda, se puede comenzar a operar. Lo primero que el sistema debe verificar es el último día de operaciones, ordenado cronológicamente, qué fue registrado. Si el sistema no encuentra ningún día de operaciones, entonces le solicita al cajero que inicie el día de operaciones para poder comenzar a operar la caja.

Si el sistema encuentra el último día de operaciones, entonces procede con otras verificaciones. Si el último día de operaciones corresponde al día actual, el sistema debe verificar si ese día de operaciones ya ha sido finalizado. Si el día de operaciones ha finalizado entonces el sistema notifica al cajero que no puede realizar ninguna otra acción hasta el siguiente día, pues el día actual ha finalizado. Si el último día de operaciones correspondiente al día actual, no ha sido finalizado, entonces el sistema presenta todas las opciones de operación que el cajero tiene asignadas para que proceda a utilizarlas.

Si el último día de operaciones encontrado no corresponde al día actual, de igual forma el sistema verifica si ese día ya ha sido finalizado. Si el día ya ha sido finalizado, simplemente el sistema dirige al cajero a la opción de iniciar un nuevo día de operaciones. Si el día encontrado no ha sido finalizado, entonces el sistema notifica al cajero que es necesario finalizar el último día de operaciones registrado, y generar su corte de caja. El sistema dirige al cajero a la opción de finalizar el día de operaciones.

Con todas estas verificaciones, cuando el cajero llega al punto de poder operar en la caja, tendrá las opciones correspondientes a los casos de uso específicos de la operación de la caja: Ingresar mercadería y registrar ventas. En la figura 16 se puede observar todo lo que se ha descrito anteriormente, correspondiente a la dinámica general de la operación del POS.

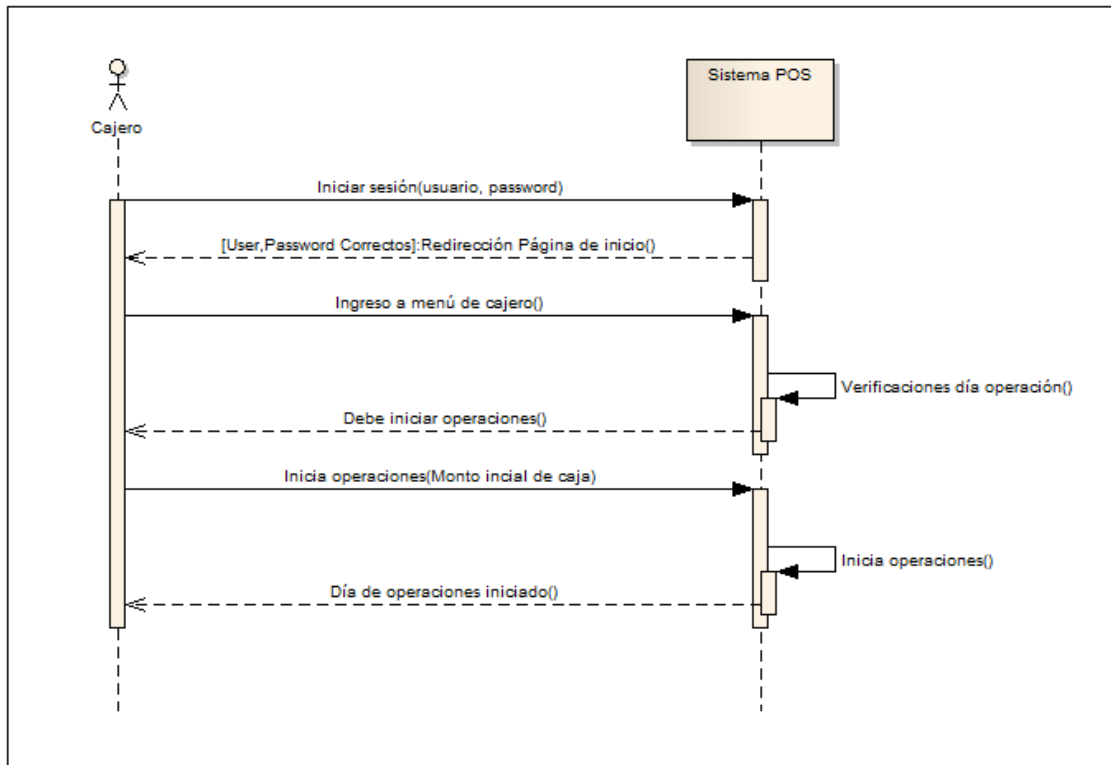


En la dinámica general de operación del POS se muestran cuatro casos de uso: iniciar día de operaciones, ingresar mercadería, registrar ventas y cerrar día de operaciones. El quinto caso de uso es el de consultar inventario, pero no se incluye en este diagrama pues esta acción es de consulta, por lo que no tiene mayor interacción del usuario. La consulta del inventario de la tienda asignado, que realiza el cajero, no está sujeta a todas las verificaciones anteriormente descritas, por consiguiente, en cualquier momento se puede hacer la consulta, aunque no se haya iniciado ningún día de operaciones. La única condicionante para esta consulta es que el cajero esté asignado a una tienda activa.

Para profundizar aún más en los comportamientos esenciales del sistema de punto de venta, ahora se analizarán los cuatro casos de uso principales de la operación. Para cada caso de uso se tiene un diagrama de secuencia; esto para poder entender toda la interacción que tendrá el sistema con los usuarios. Como ya se ha descrito, estos cuatro casos de uso están condicionados por una serie de verificaciones; estas verificaciones ya se han detallado, por lo que el enfoque será directamente en cada caso de uso, y no en las verificaciones mencionadas.

El inicio del día de operaciones es un comportamiento bastante sencillo. El sistema mostrará un mensaje al cajero en donde le notifica que es necesario que inicie el día de operaciones. El cajero le indicará al sistema con cuánto dinero en efectivo está iniciando el día, luego confirma la operación y el sistema registrará el nuevo día de operaciones. En la figura 17 se puede observar cómo es la interacción del sistema con el cajero.

Figura 17. Diagrama de secuencia iniciar día de operaciones



Fuente: elaboración propia.

El ingreso de mercadería es el proceso que alimentará el inventario de la tienda. Lo primero que el cajero debe realizar es ingresar los datos del proveedor al sistema, este verifica que estén correctos y le muestra la pantalla para agregar productos. Los pasos para agregar productos están dentro de un ciclo que se repite cuantas veces sea necesario, según la cantidad de productos que se agreguen.

El cajero ingresa el código del producto, y el sistema verifica que el producto exista en el catálogo de productos. Al encontrar el producto, el sistema verifica si en esa orden de ingreso ya se ha agregado anteriormente ese

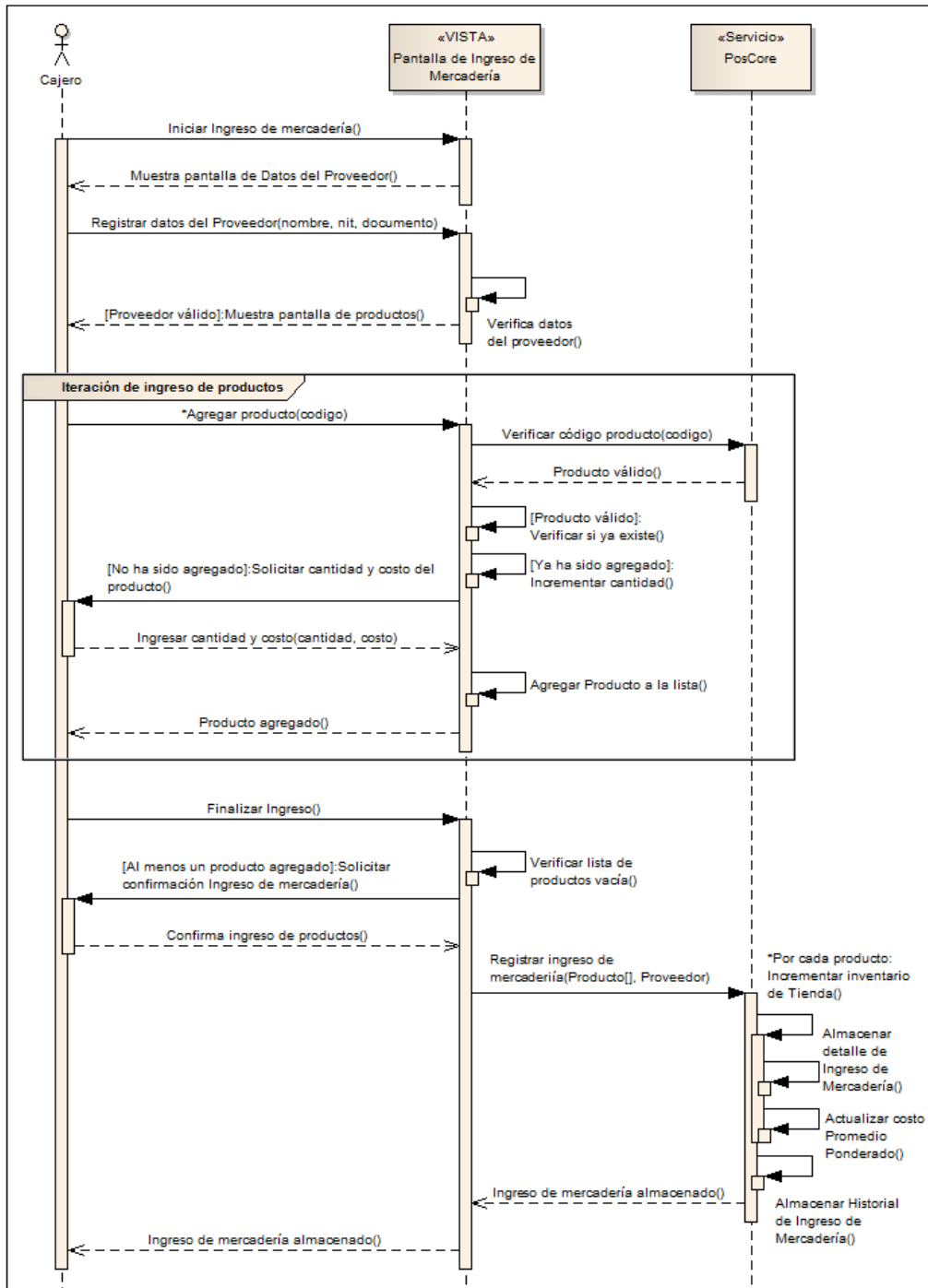
producto; si ya se ha agregado, el sistema solamente incrementará en uno la cantidad a ingresar de dicho producto. Si el producto no ha sido agregado, el sistema solicitará al cajero la cantidad a ingresar y el costo del producto.

El cajero deberá ingresar los datos solicitados, y al momento de confirmar la operación, el sistema validará que la cantidad sea mayor a cero y que el costo del producto sea mayor o igual que 0,00. Con estos datos el sistema agregará a la lista de productos a ingresar, el producto que el cajero ha indicado. Como ya se mencionó, este proceso es iterativo cuantas veces sea necesario.

Cuando el cajero notifique que ha terminado de ingresar los productos y finalice el proceso, el sistema realiza la verificación de que se haya agregado al menos un producto con cantidad mayor a cero y costo mayor o igual a 0,00 al listado de productos a ingresar. Si no cumple con la verificación, el sistema notificará el error al cajero, y este podrá continuar con el proceso de agregar productos.

Una vez que cumpla con la verificación, el sistema solicitará una confirmación al cajero, para poder procesar el ingreso de productos. Cuando el cajero confirme la ejecución, el sistema procederá con las siguientes acciones internas: por cada producto agregado, se debe incrementar la existencia de ese producto en la tienda asignada al cajero, en la cantidad que se ha definido en el ingreso de productos. Luego el sistema debe calcular el nuevo costo que tendrá el producto con el método de promedio ponderado, tomando en cuenta todas las existencias en todas las tiendas de ese producto.

Figura 18. Diagrama de secuencia ingreso de mercadería



Fuente: elaboración propia.

El sistema debe almacenar el registro de la orden de ingreso de productos a la tienda, así como actualizar el historial de movimientos de productos. Después de realizar toda la transacción, transparente al cajero, el sistema debe notificar al usuario que todo el proceso se ejecutó con éxito, dejando la pantalla lista para ingresar una nueva orden de ingreso de productos a la tienda.

El proceso de registro de ventas es, sin lugar a duda, el comportamiento principal en todo el sistema. Pues todo lo que se ha detallado hasta el momento, y lo que se descubra después de este comportamiento, es todo el entorno del POS. Se puede decir que todos los demás casos de uso complementan el contexto de un POS pero el registro de ventas es la razón de ser del sistema.

En el registro de ventas, lo primero que el cajero debe hacer es ingresar los datos del cliente, datos básicos: nit, nombre y dirección obligatorios, y teléfono, email opcionales. El sistema debe validar que los datos tengan el formato correcto, y luego de pasar esa validación, muestra al cajero la pantalla para agregar productos a la venta.

Al igual que en el ingreso de mercadería, en el registro de ventas, el agregar productos es un proceso iterativo, y se repite cuantas veces sea necesario. El cajero ingresa el código del producto y el sistema verifica que exista en catálogo de productos. Cuando encuentra el producto, el sistema verifica si ya se ha agregado el producto a la lista de la venta, si ya se ha agregado solamente incrementa en uno la cantidad del producto, y si no, entonces agrega el producto a la lista.

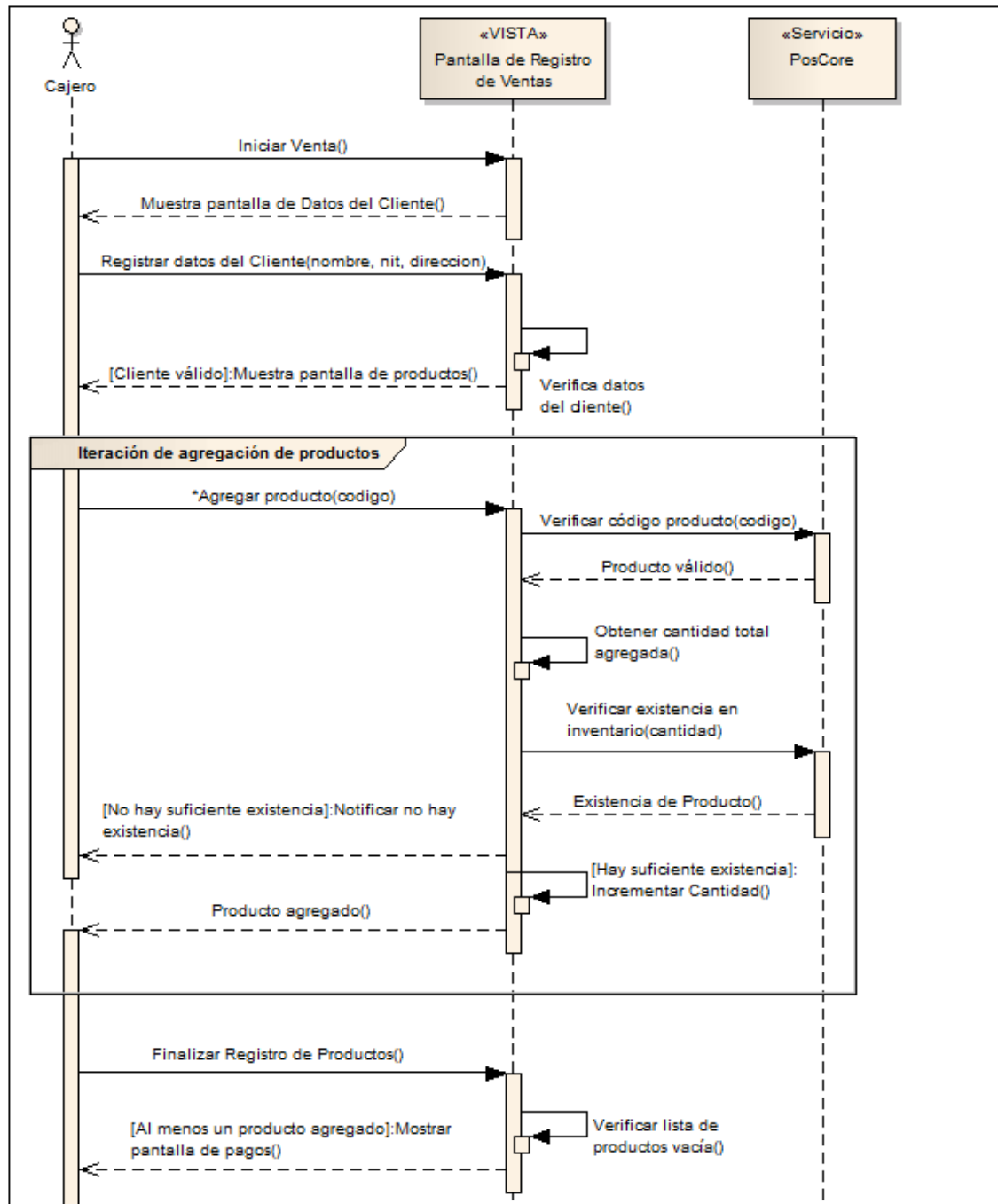
En cualquiera de los dos casos, el sistema realiza una validación previa; que la existencia del producto en el inventario de la tienda sea mayor o igual a

la cantidad total agregada en la lista de la venta. Si no cumple con esta validación, entonces muestra un mensaje al cajero notificando que no hay suficiente existencia del producto en el inventario, y no incrementa la cantidad o agrega un producto a la lista de la venta.

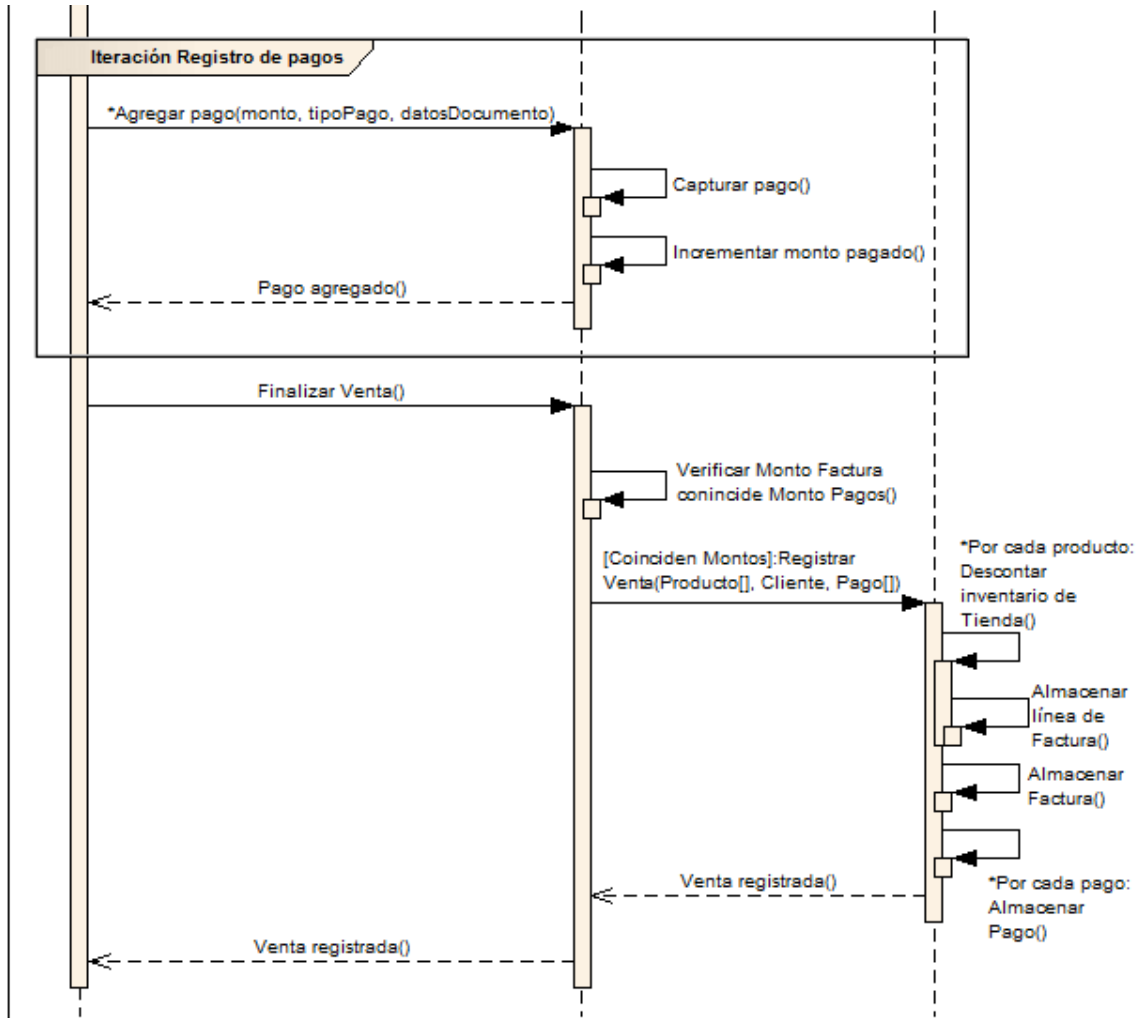
Después de haber agregado los productos que se desee, el cajero continúa a registrar los pagos. La validación que se realiza es que al menos se haya agregado un producto a la lista de venta. El precio de venta de cada producto es tomado del catálogo, según haya sido definido por el administrador. El proceso de registro de pagos es, al igual que agregar productos, iterativo cuantas veces sea requerido.

Existen dos formas de pago: efectivo y tarjeta de crédito/débito. Cuando se agrega un pago en efectivo, el sistema verifica si ya se ha agregado esta forma de pago. Si ya se encuentra el pago en efectivo, simplemente se incrementa el monto del efectivo, y si no ha sido agregado entonces se agrega la forma de pago y se define el monto con la cantidad especificada. Al agregar un pago con tarjeta de crédito/débito, se agrega el pago a la lista de pagos de la venta y se almacenan los datos del comprobante de la transacción de la tarjeta. Este proceso se repite cuantas veces sea requerido.

Figura 19. Diagrama de secuencia registrar ventas



Continuación de la figura 19.



Fuente: elaboración propia.

El siguiente paso para el cajero será finalizar la venta. En este momento el sistema verifica que el total de la venta coincida con el total de pagos registrados. Al cumplir con este requisito, el sistema realiza un conjunto de acciones, transparentes al usuario, para registrar la venta. Este proceso incluye lo siguiente: almacenar el encabezado de la factura, por cada producto se descuenta la cantidad vendida de la existencia en el inventario de la tienda y se



almacena el detalle de la venta como una línea de la factura, por cada pago se almacena la forma de pago que se realizó en la factura, y por último se registra la venta en el historial de facturación, historial de movimientos de productos y en las estadísticas de venta de la tienda, de los productos y del cliente.

Una vez finalizado este proceso, el sistema muestra el documento que respalda la venta para que pueda ser impreso por el cajero, y notifica que el registro de la venta se realizó con éxito. Luego al cajero solamente le queda iniciar una nueva venta y deberá comenzar el proceso de registro de ventas.

Por último, en el caso de uso cerrar día de operaciones. Cuando el cajero solicite al sistema el cierre del día de operaciones, el sistema le mostrará una interfaz en donde el deberá ingresar el total recaudado en efectivo y la suma de todos los comprobantes de transacciones con tarjeta de crédito/débito. El cajero envía la información del acumulado físico que él tiene. El sistema no realizará ninguna verificación, sino que solicitará al cajero que confirme el cierre del día.

Cuando el cajero confirma el cierre del día, el sistema muestra una nueva interfaz con el resumen de lo que el cajero ha ingresado, además de solicitar una última confirmación por parte del cajero. El cajero puede cancelar y regresar a la interfaz en donde ingresa el acumulado físico, o confirmar el cierre del día. Si el cajero confirma nuevamente el cierre del día, el sistema procederá con el cierre del día.

El sistema almacenará el acumulado físico que ha registrado el cajero, tanto del efectivo como de la tarjeta de crédito/débito. Después de esto el sistema generará, con base en las ventas y los pagos registrados en el día, el acumulado que debería tener en la caja, tanto de efectivo como en los

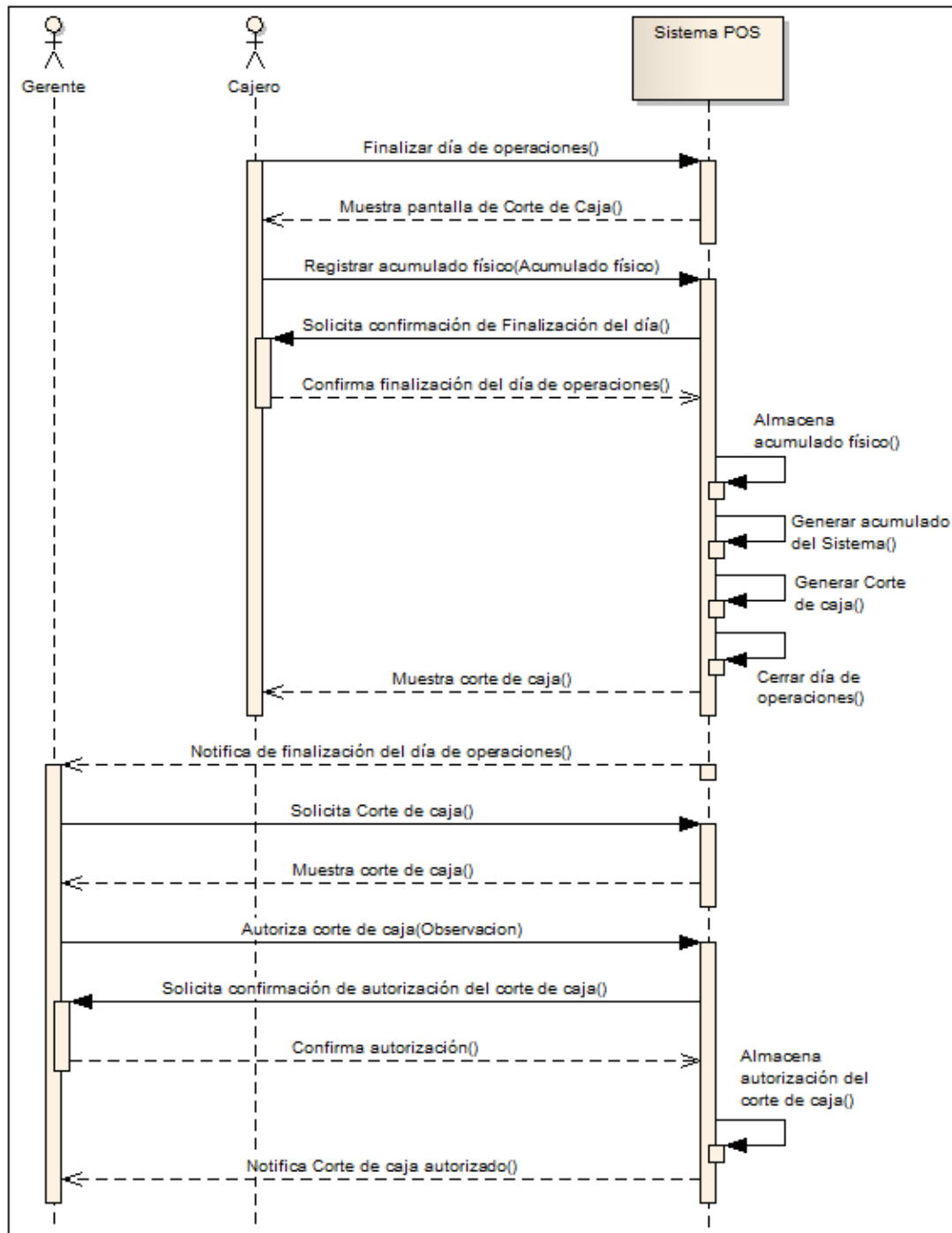
comprobantes de transacciones. Con esta información, el sistema debe generar el corte de caja, que es el documento que respalda esta información.

Con el corte de caja generado, se marca el día de operaciones como finalizado, y muestra al cajero el corte de caja para que pueda imprimirlo, y termina la interacción con el cajero. Al mismo tiempo que termina este proceso, envía una notificación por correo electrónico al gerente de que el día de operaciones ha finalizado.

Cuando el gerente ingrese al sistema, podrá ver el historial de los cortes de caja de cada tienda. El gerente solicita el detalle de un corte de caja en específico, y el sistema le muestra el documento. El gerente tiene la opción de confirmar el corte de caja, incluyéndole o no, observaciones que luego le generen conocimiento del mismo corte de caja. El sistema solicita la confirmación de la aprobación del corte de caja, y cuando el gerente confirma, almacena las observaciones y marca el corte de caja como confirmado.

Con el proceso de cierre del día de operaciones se completa el análisis del sistema de punto de venta. Ya se ha definido exactamente qué es lo que se espera del sistema, las expectativas funcionales que se tienen, ahora entraremos de lleno a la implementación, es decir, el cómo para lograr cumplir con los requerimientos y las expectativas.

Figura 20. Diagrama de secuencia cerrar día de operaciones



Fuente: elaboración propia.

### **3.2. Infraestructura y arquitectura del software**

Según el Diccionario de la Lengua Española, de la Real Academia Española, infraestructura significa: “conjunto de elementos o servicios que se consideran necesarios para la creación y funcionamiento de una organización cualquiera”. En el contexto de software no cambia la acepción de infraestructura, pues se refiere a todos los componentes, tanto de software y hardware, necesarios para que un sistema de informática pueda funcionar.

Hasta el momento se ha definido el proceso del punto de venta, los requerimientos funcionales y no funcionales, se ha realizado el análisis del sistema y ya están definidas las bases de qué es lo que el sistema debe hacer. A partir de este momento se comienza a tratar temas mucho más técnicos, que establecerán el cómo vamos a realizar lo que se ha requerido.

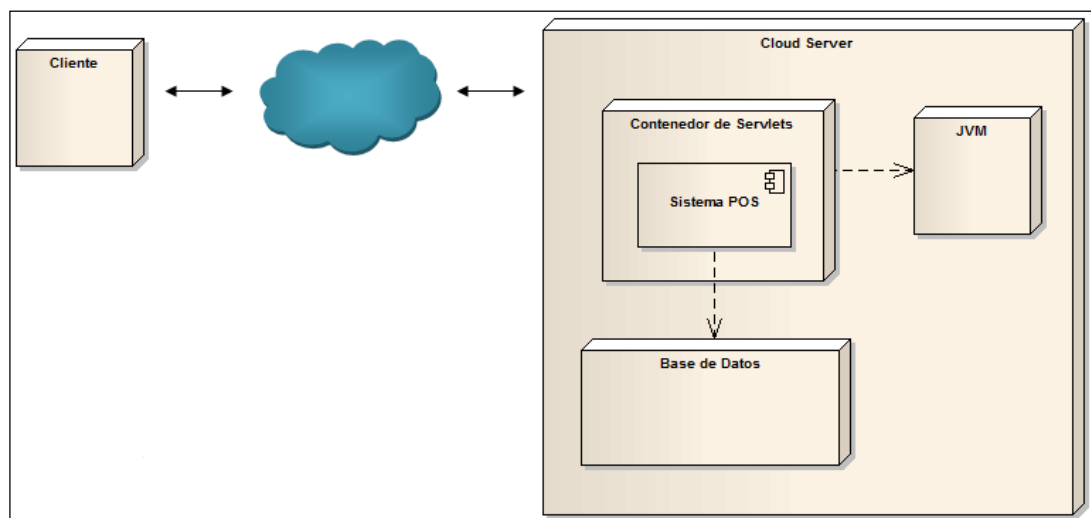
Si se habla de un sistema en la nube, que es el tema principal de este trabajo, una de las cosas que deben tener mayor importancia es la infraestructura necesaria para que el sistema funcione, tanto para el proveedor del sistema como para el cliente que lo utilizará. El objetivo principal de este desarrollo es ayudar al desarrollo de las pymes promoviendo la utilización de herramientas de tecnología de información; y en la consecución de este objetivo, el factor costo es de vital importancia.

Es por esto que la infraestructura propuesta, para que el sistema de punto de venta en la nube funcione, es sumamente sencilla y orientada en la reducción de costos. La infraestructura está dividida en dos partes: la del lado cliente y del lado servidor. Ambas infraestructuras están conectadas a través de lo que se conoce como nube, es decir, a través de una conexión a internet.

La infraestructura necesaria del lado del cliente es una computadora con un navegador de internet y una conexión a internet, eso es todo. Al ser un servicio en la nube, absolutamente toda la carga de trabajo está del lado del servidor. La infraestructura del lado del servidor es sumamente sencilla también: un servidor virtualizado adquirido con alguno de los muchos proveedores que existen. Si se observa detenidamente, un servidor como nube no es más que un servicio que prestan algunos proveedores, y que permite tener el servidor sin necesitar alguna infraestructura.

Todo esto deriva en la reducción del costo, pues, del lado del proveedor del sistema de punto de venta, el costo en infraestructura se reduce al alquiler de un servidor en la nube, y al tiempo que se invierte en monitorearlo. Este impacto en la reducción de costos se ve reflejado en el precio que se defina para el alquiler del sistema de punto de venta; y del lado del cliente porque prácticamente no necesita hardware especializado, ni otro equipo.

Figura 21. **Infraestructura del sistema de punto de venta en la nube**



Fuente: elaboración propia.

La arquitectura de un software es el diseño de más alto nivel de abstracción de la estructura de un sistema, e indica las partes más generales del software y la interacción entre ellas. En el libro: *An introduction to Software Architecture*, los autores definen que la arquitectura de un software se enfoca en aspectos más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema.

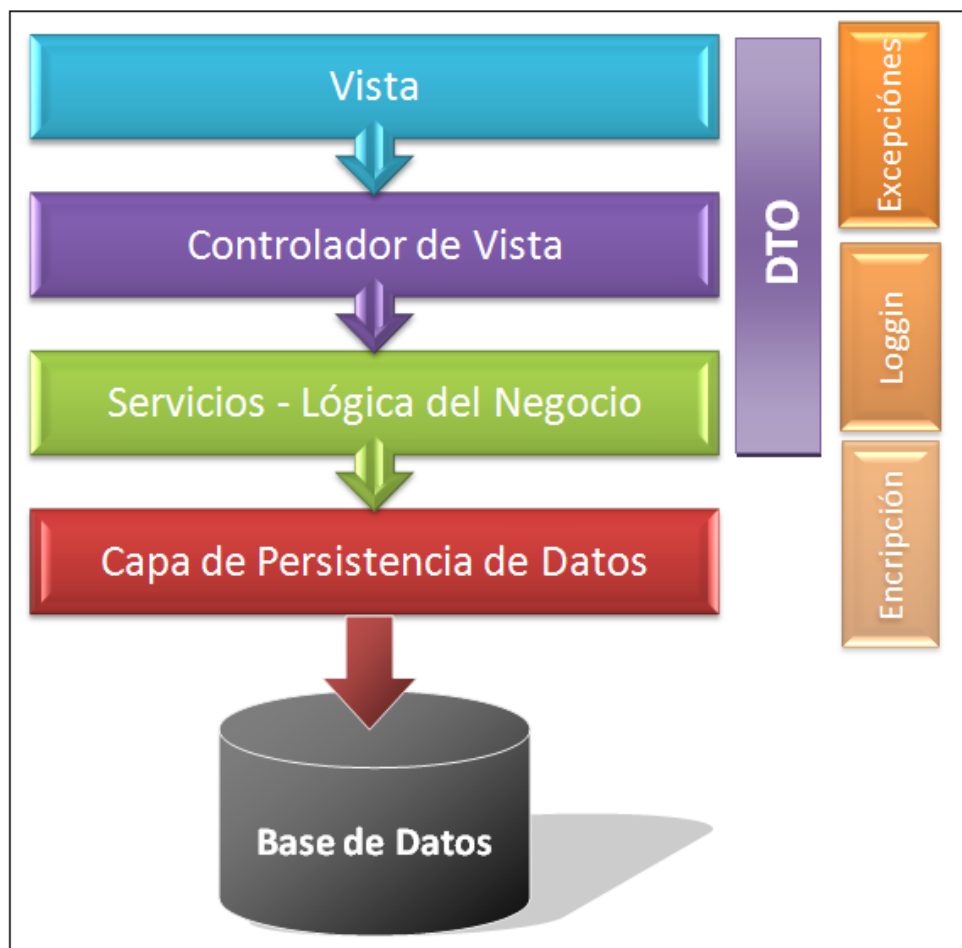
Existen diferentes arquitecturas de software conocidas, las cuales se pueden adoptar al momento de desarrollar un sistema. Estas arquitecturas están enmarcadas por sus propias fortalezas y debilidades, por lo que no es correcto decir que una arquitectura es mejor que otra, sino que una arquitectura se adapta mejor a un contexto específico de un sistema. Algunas de las más comunes son: arquitectura monolítica, cliente-servidor, arquitectura de tres capas.

La arquitectura diseñada para el sistema objeto de este trabajo, está muy orientada a una arquitectura en tres capas, aunque en realidad se manejarán cuatro capas principales, y otras capas de apoyo. Las cuatro capas principales son: vista, controlador de la vista, servicios y persistencia de datos. Cada capa se comunica de forma lineal, como se puede observar en la figura 22. Además de estas cuatro capas, existe una capa más que servirá para transferencia de datos entre las capas de la vista, controlador de la vista y los servicios.

Agregadas a las capas que ya se mencionaron, existen tres capas más que apoyan el trabajo de las principales. Estas capas son: encriptación para proteger la información almacenada en la base de datos y la transferencia de la misma, la capa de logging para manejar todos los logs de las transacciones, acciones y ejecuciones que se realicen en el sistema, y la capa de excepciones

para controlar los posibles errores que puedan ocurrir en el procesamiento de la información, y así poder tener un manejo seguro de errores y prevenir la divulgación de información no deseada.

Figura 22. **Arquitectura general del sistema de punto de venta en la nube**



Fuente: elaboración propia.

La capa de persistencia de datos debe asegurar la integridad de la información almacenada en la base de datos. Para esto se utilizarán las

tecnologías Hibernate, JPA y Spring-Data, creando un contexto completamente transaccional en cada una de las operaciones realizadas por el usuario, que involucren acceso y/o modificación de la información en la base de datos. En esta capa el esfuerzo está en la correcta implementación de los diferentes *frameworks* involucrados.

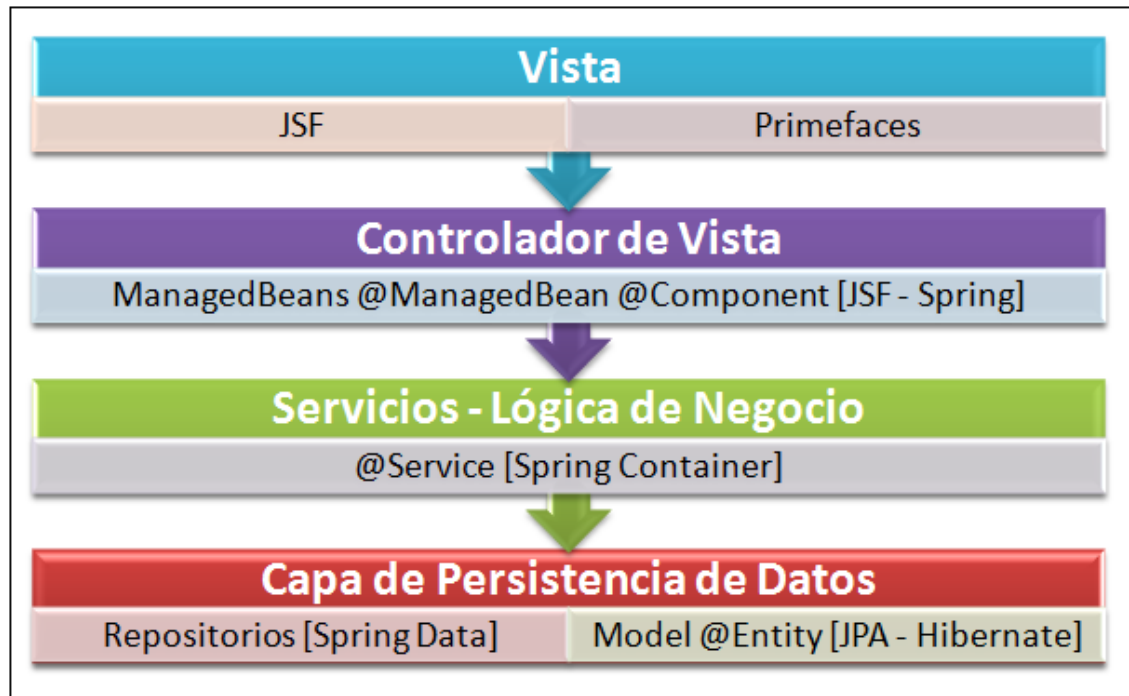
La capa que tendrá la mayor carga de desarrollo y programación es la de servicios, pues en esta capa se implementará toda la lógica del negocio, es decir, todo lo que se ha capturado en la fase de análisis. En la capa de servicios se aprovecharán todos los beneficios que el contenedor ligero de Spring ofrece; por medio de este se comunicarán las cuatro capas.

Las capas de vista y controlador de la vista, están íntimamente relacionadas. Por cada vista existirá un controlador (@ManagedBean) que gestionará toda la lógica de la vista, y será el encargado de la comunicación con los servicios. Las vistas serán implementadas bajo los estándares de JSF, utilizando el *framework* Primefaces para todos los componentes web. Primefaces es una abstracción de JSF por lo que cumple con todos los estándares.

Una de las capas auxiliares, pero que es de suma importancia, es la capa de transferencia de información. Desde la capa de vista hasta la capa de servicios, se comunicarán a través de la capa de transferencia para poder proveer información adecuada; una vez la información se encuentra en los servicios, estos se encargarán de manipularla y convertirla en información de la base de datos, a través de la capa de persistencia de datos.



Figura 23. **Detalle de la arquitectura del Sistema de Punto de Venta en la nube**



Fuente: elaboración propia.

### 3.3. Diseño del producto

El diseño de sistemas es la definición de la arquitectura de software y hardware, componentes, módulos y datos de un sistema, con el objetivo de cumplir los requerimientos capturados en la fase de análisis. En la sección anterior se describió la infraestructura y la arquitectura del sistema. Esta sección se enfocará en el diseño de los componentes que conformarán el sistema de punto de venta, así como las clases y objetos de cada componente.

### **3.3.1. Componentes**

En un sistema de software, un componente es un elemento que ofrece un conjunto de servicios o funcionalidades a través de interfaces definidas. Otra definición de componente de software dice que es una unidad modular con interfaces bien definidas y que define su comportamiento a través de estas interfaces y de interfaces requeridas de otros componentes. Un componente de software debe poseer 4 características esenciales: reutilizable, intercambiable, poseer interfaces bien definidas y ser cohesivo.

Una característica fundamental que debe poseer un sistema de software es el bajo acoplamiento entre las diferentes unidades o componentes, es decir, que mientras menos dependientes sean las partes que conforman un sistema, mejor será el resultado obtenido. El bajo acoplamiento ofrece muchas ventajas, por ejemplo: mejora la mantenibilidad de los componentes, aumenta la reutilización de código, evita el riesgo de tener que modificar varios componentes cuando se desea alterar uno solo.

Por todo lo anterior es que en este proyecto, la implementación de un sistema de punto de venta en la nube, todo el análisis está basado en componentes desacoplados, para crear un producto de mayor valor y calidad.

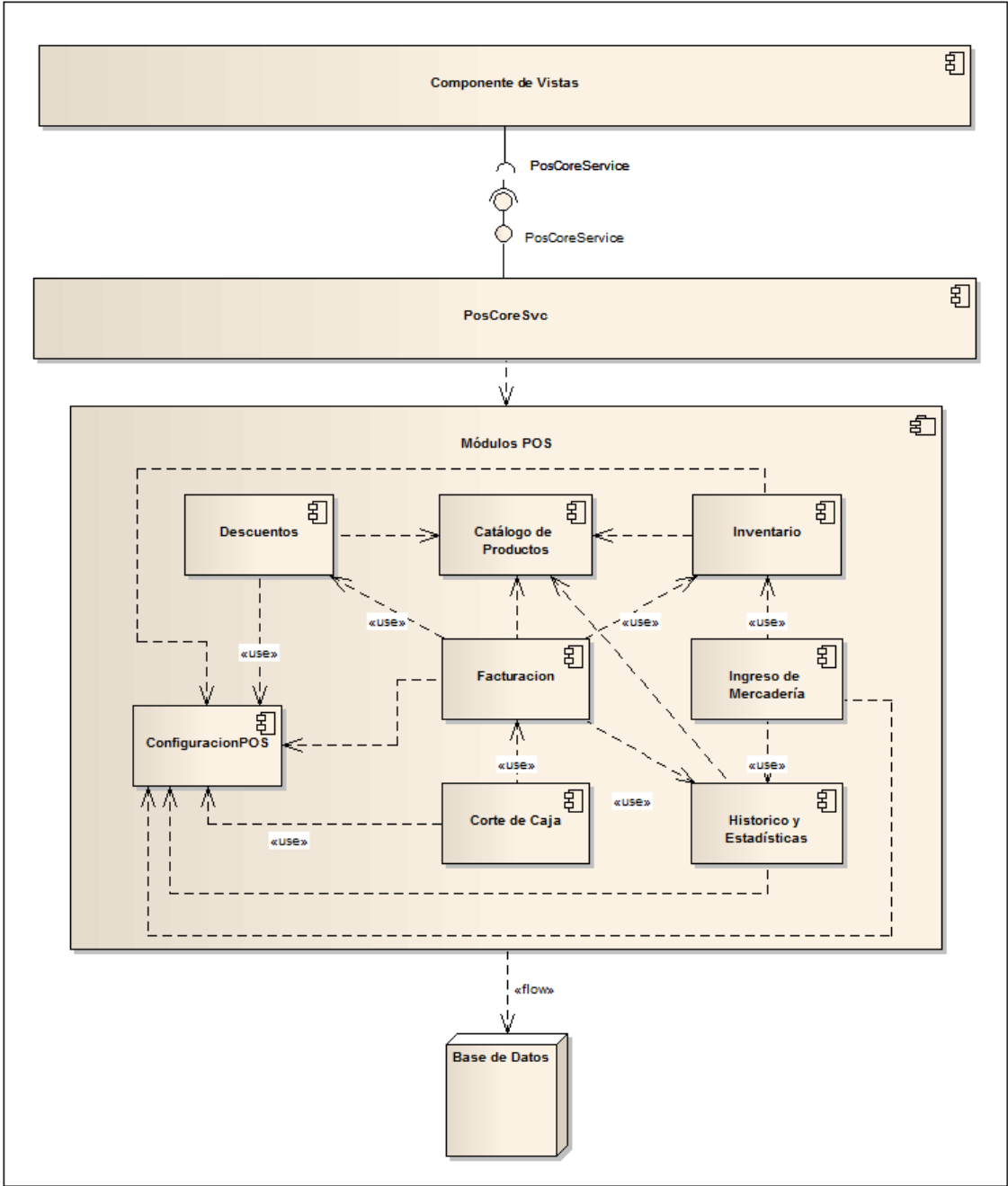
En el diseño del sistema se han definido 9 componentes, implementado en cada uno un concepto de negocio independiente. Cada uno de estos componentes expone una interfaz para que cualquiera de los otros componentes pueda utilizar los servicios expuestos en la interfaz. Internamente cada componente implementará la lógica de negocio, definida en los requerimientos, para su correspondiente concepto de negocio. Los componentes que se definieron son los siguientes:

- Configuración POS
- Día de operaciones
- Catálogo de productos
- Inventario
- Descuentos
- Facturación
- Corte de caja
- Ingreso de mercadería
- Histórico y estadísticas

Estos 9 componentes se comunican entre sí a través de la interfaz expuesta por cada uno (ver figura 24). A su vez, serán utilizados por el servicio principal del sistema, llamado CoreService. El CoreService expone una interfaz para que la capa de controlador de la vista tenga acceso a las funcionalidades de los diferentes componentes, como ya se detalló en la arquitectura del sistema.

Los componentes, internamente, poseen una estructura de clases que definen sus atributos y funcionalidades específicos. Cada uno de estos componentes posee la arquitectura del sistema de forma individual, es decir que cumplen con las capas de servicios y de persistencia de datos. A continuación se detallará la estructura de clases utilizada en los componentes.

Figura 24. Diagrama de componentes del sistema de punto de venta en la nube



Fuente: elaboración propia.

### 3.3.2. Clases y modelo de datos

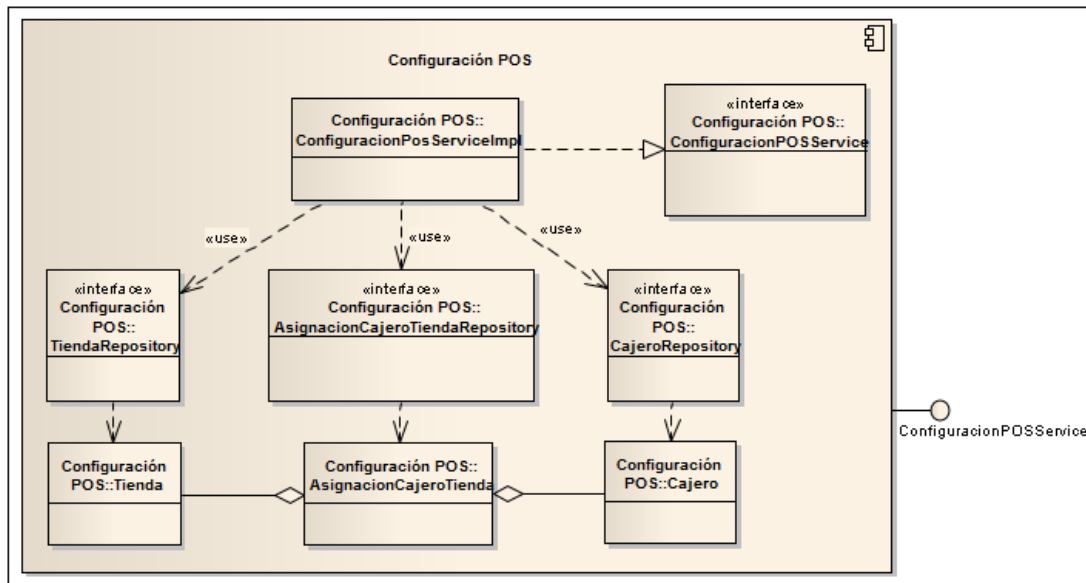
Una clase es la definición de las propiedades y comportamientos de un tipo de objeto específico. Un objeto es la instancia de una clase; en un objeto los atributos de la clase contienen los datos y la información de este. Los comportamientos definidos en una clase serán los que se puedan ejecutar en un objeto. Una clase puede tener muchos objetos creados a partir de ella.

Como ya se ha descrito, cada uno de los componentes está compuesto por una estructura de clases relacionadas y que interactúan entre sí, por lo que el análisis de los diagramas de clases será independiente para cada componente, siguiendo el principio de desacoplamiento. Además al analizar el diagrama de clases de cada componente se puede observar cómo se cumple con la arquitectura definida para el sistema de punto de venta en la nube.

En el componente Configuración POS se implementará la lógica para la administración de las tiendas, los cajeros y la asignación de cada uno de ellos. El componente cuenta con una interfaz que expone su propio comportamiento a los demás componentes. La clase principal de comportamiento implementa todos los métodos definidos en la interfaz del componente, en la cual se programará toda la lógica de negocio. Esta clase utiliza la capa de persistencia de datos a través de las interfaces de los repositorios (*framework* Spring-Data).

Además de las interfaces de los repositorios, también se definieron las clases que representan las entidades de la base de datos. En el desarrollo del sistema se utiliza el mapeo de entidad-relación a objetos, a través de las tecnologías Hibernate y JPA. Los repositorios y las clases que mapean las entidades de la base de datos conforman la capa de persistencia de datos, que se comunica directa y únicamente con la capa de servicios.

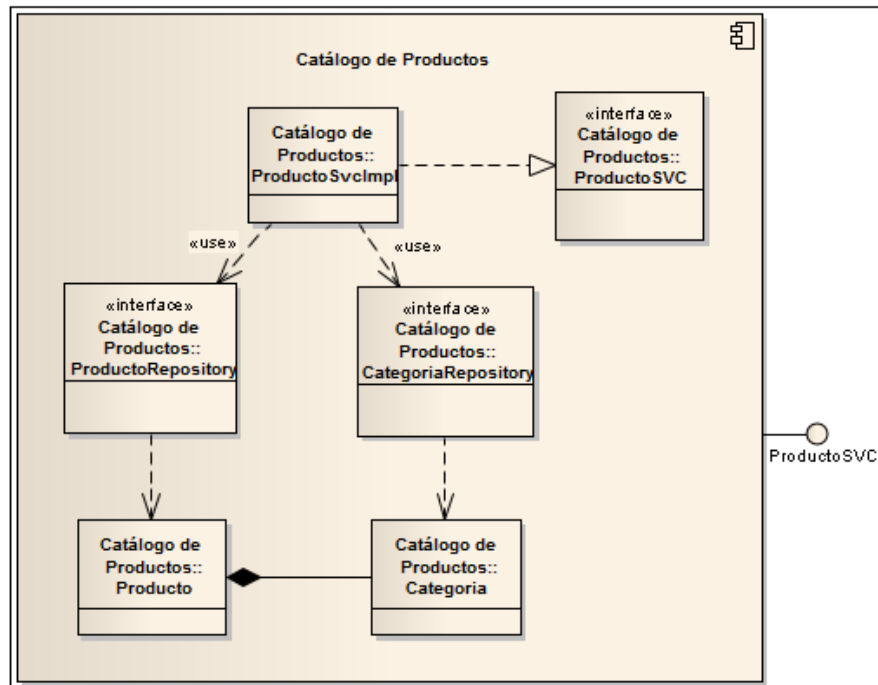
Figura 25. Diagrama de clases del componente configuración POS



Fuente: elaboración propia.

El componente catálogo de productos será el que implemente la lógica de administración de categorías de productos y productos, con el fin de proveer la información a los demás componentes que lo requieran como por ejemplo: el componente de facturación, el de inventario y otros. Internamente, sus clases están estructuradas bajo la misma arquitectura, servicios y persistencia de datos, y expone sus servicios a través de una interfaz (ver figura 26).

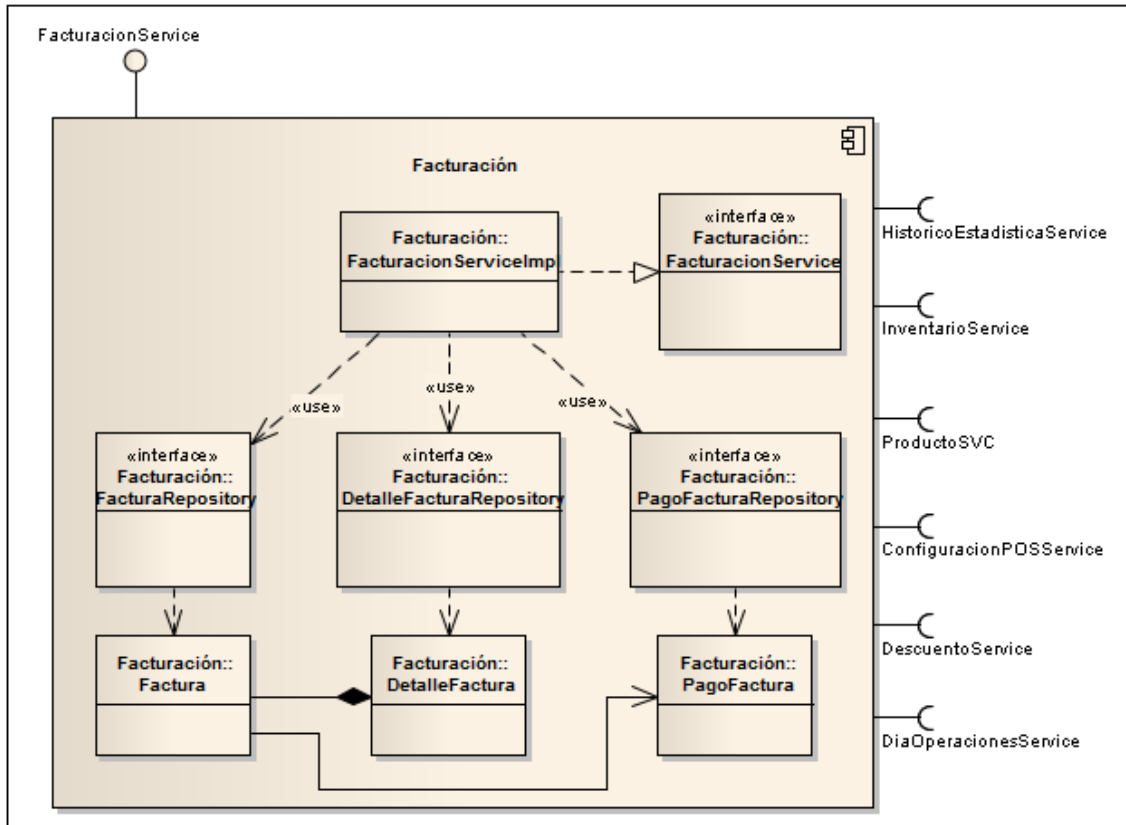
Figura 26. Diagrama de clases del componente catálogo de productos



Fuente: elaboración propia.

El siguiente componente que se analizará es facturación. Este es el encargado de todo lo concerniente a las facturas y al detalle de cada una de ellas, y es el principal para el caso de uso del registro de ventas. Internamente se encontrará una estructura que cumple con la arquitectura y que, al igual que los dos anteriores, expone sus servicios a través de una interfaz, por lo que no se profundizará más en este tema.

Figura 27. Diagrama de clases del componente facturación



Fuente: elaboración propia.

Se debe analizar que este componente ya no solo expone sus servicios, sino que también requiere de los servicios de otros componentes. La facturación utiliza los servicios de los componentes configuración POS, catálogo de productos, día de operaciones, inventario, descuento, histórico y estadística. Aunque el de facturación utilice otros componentes, se puede ver que su comportamiento es independiente al de los demás, esto refuerza el concepto de bajo acoplamiento.

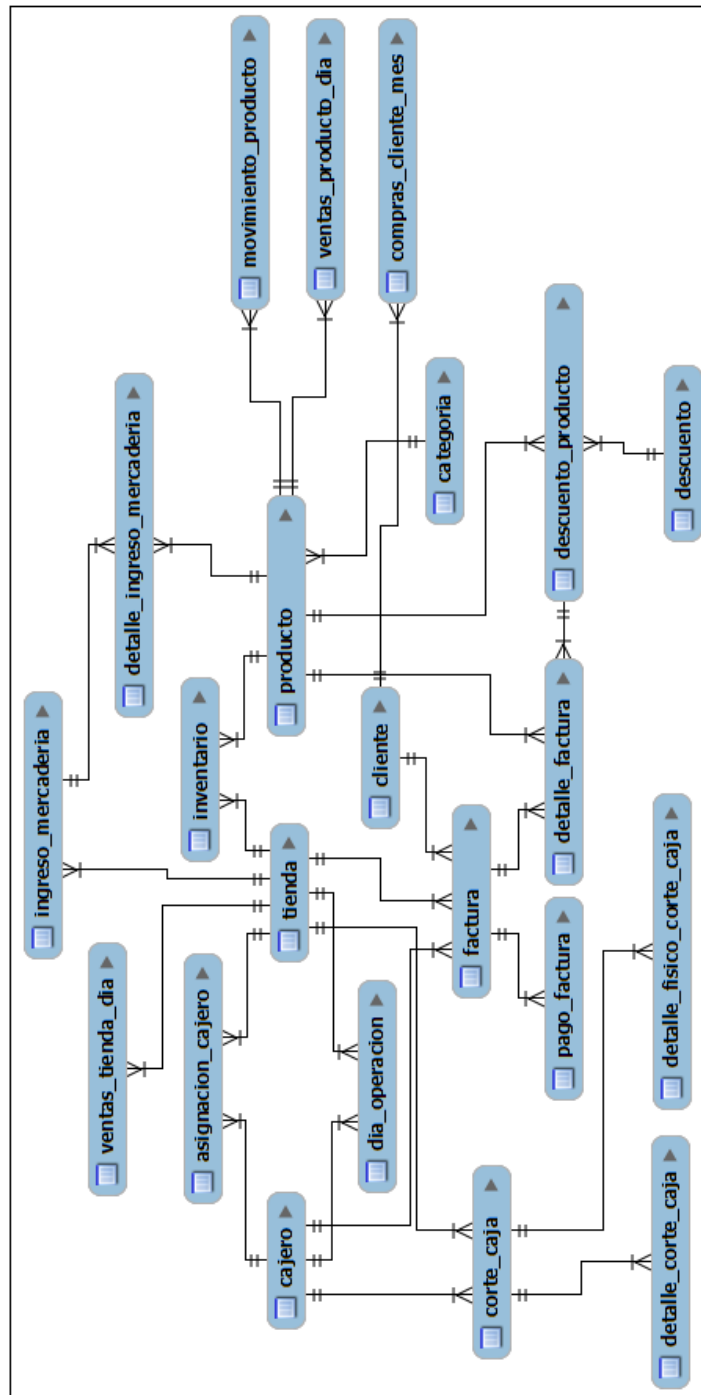


Para entender este concepto de mejor forma se analizará un ejemplo. El componente facturación utiliza el histórico y estadística, independientemente de la lógica de negocio que utilice este. Para el componente de facturación, lo que realiza el componente histórico y estadística es ajeno, por lo que este segundo componente podría ser reemplazado por distintas versiones o implementaciones que el primer componente no sufriría ninguna variante; por ejemplo, si las estadísticas se almacenan en una base de datos o en archivos del sistema operativo.

Si se continúa analizando cada uno de los componentes se puede observar que cada uno de ellos cumple con la arquitectura definida para el sistema y que cada uno expone sus servicios por medio de una interfaz. Es por esto que por el momento solamente se profundizará en los tres componentes que ya se han explicado. El detalle de cada uno de estos se puede encontrar en el apéndice de este documento.

En los componentes del sistema se pueden observar las clases que mapean las entidades de la base de datos; estas clases están distribuidas en sus respectivos componentes. Para completar el diseño del sistema, se visualizará un diagrama completo del modelo de datos utilizado en el sistema de punto de venta. Este modelo representa la estructura de la base de datos en donde se almacenará toda la información generada por las transacciones del sistema de punto de venta (ver figura 28).

Figura 28. Modelo de datos del sistema de punto de venta en la nube



Fuente: elaboración propia.

### 3.4. Codificación

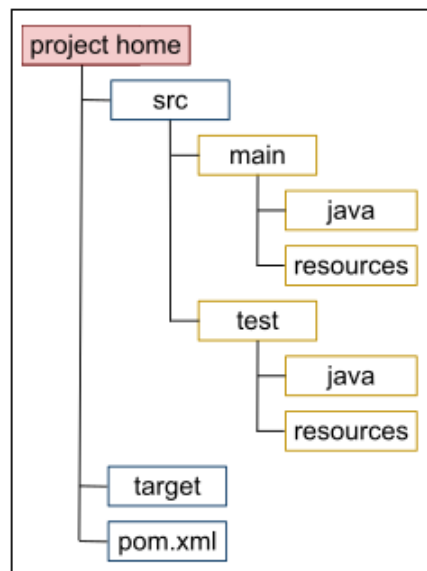
La etapa de codificación consiste en las tareas comúnmente llamadas programación, tareas en las que se toma todo el diseño realizado en la etapa previa y se genera el código fuente del sistema. Se puede pensar que esta es la que más tiempo consume en todo el proceso de desarrollo del sistema; pero al tener un análisis y diseño correctamente realizado y validado con el cliente, esta se convierte en no más que trasladar los requerimientos y los lineamientos del diseño en código de programación.

Para este proyecto, se definió que el lenguaje de programación a utilizar es Java, y la interfaz que se utilizará para la codificación será Eclipse. Es importante mencionar también que para la gestión y la construcción del proyecto se utiliza la herramienta maven, la que gestionará también la dependencia del proyecto en los *frameworks* definidos.

La estructura de un proyecto maven consiste en un archivo de configuración pom.xml, de una carpeta src en donde se encuentran absolutamente todos los archivos fuentes que componen el proyecto y que servirán para la construcción del mismo. Dentro de la carpeta src se tienen dos carpetas: main que tiene las fuentes del proyecto y test para los archivos utilizados en las pruebas del sistema. En la carpeta main se tiene la estructura de tres carpetas: La carpeta java contiene todos los archivos de código fuente java utilizados en el proyecto y que se encuentran ordenados en la configuración de paquetes que mejor nos convenga, la carpeta resources que contiene cualquier otro archivo que necesitemos para el proyecto, y la carpeta webapp en donde se encuentran todos los archivos que conforman el proyecto web, es decir, las páginas web, los scripts, las hojas de estilo, entre otros. Esta

última carpeta, webapp es específicamente para un proyecto java web, que es la que se utilizará para este trabajo.

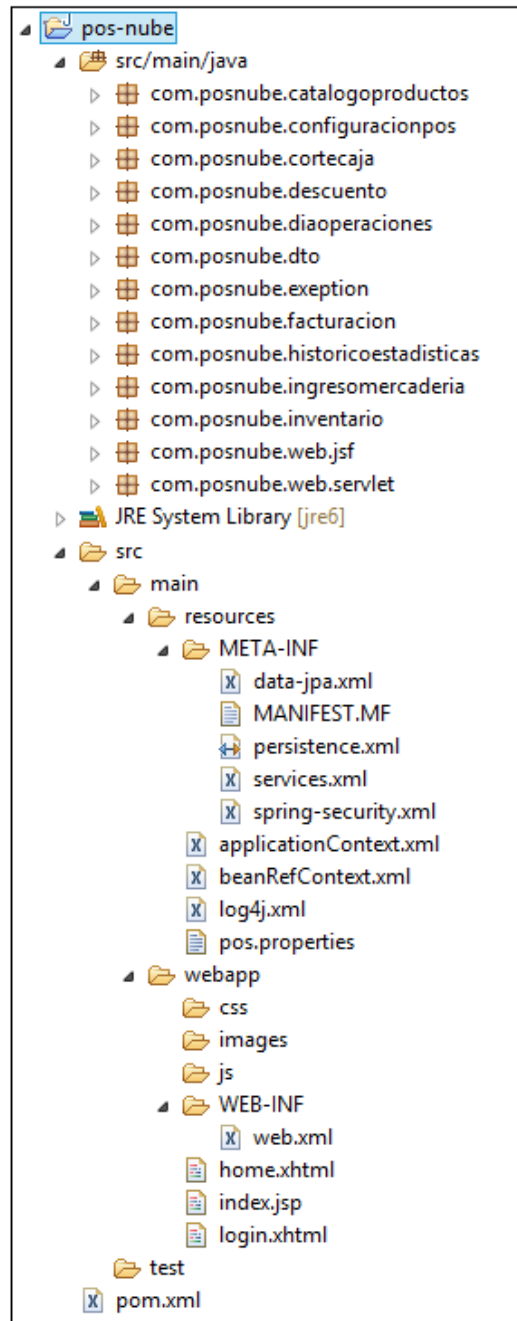
Figura 29. **Estructura de directorios de un proyecto java gestionado con maven**



Fuente: Apache Maven. Wikipedia. [http://en.wikipedia.org/wiki/Apache\\_Maven](http://en.wikipedia.org/wiki/Apache_Maven). Consulta junio de 2012.

En este proyecto se tiene una estructura de paquetes que sigue los lineamientos definidos en el diseño de componentes y en la arquitectura. Por cada componente se tiene un paquete, y se tiene un paquete para la capa de controlador de la vista. Además de los archivos de código fuente java, se tiene en la carpeta de resources los archivos de configuración de spring, hibernate, jpa y spring-security. En la carpeta webapp se incluyen todos los archivos de la capa de la vista y también el archivo principal de configuración de una aplicación java web, dentro de la carpeta WEB-INF (ver figura 30). También se encuentran los paquetes de transferencia de datos y de las excepciones.

Figura 30. Estructura de paquetes y carpetas del proyecto de punto de venta en la nube



Fuente: elaboración propia.

Para que se pueda entender de una forma generalizada lo que se trabajó en la codificación, se analizará un ejemplo por cada capa en la arquitectura, y como es que están desarrollados los componentes; con lo cual también se analizará el contenido de cada paquete del proyecto. El objetivo de este trabajo es mostrar los fundamentos del sistema, por lo que con los ejemplos se abarcará cada uno de los puntos importantes en la codificación y desarrollo.

Se iniciará por la capa de persistencia de datos. Esta capa está conformada por dos paquetes: model y repository. En el paquete model se tiene cada clase que mapea una entidad de la base de datos. Este mapeo se realiza con los *frameworks* Hibernate y JPA, y es por medio de anotaciones, lo que reduce el uso de configuraciones a través de archivos xml. En la figura 31 se encuentra un ejemplo del mapeo de la entidad Tienda de la base de datos:

Figura 31. **Mapeo JPA de la entidad Tienda**

```
@Entity
@Table(name="tienda")
public class Tienda implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name="id_tienda")
    private Long idTienda;

    private String codigo;

    private String direccion;

    private boolean eliminado;

    private String nombre;

    private String telefono;
}
```

Fuente: elaboración propia.

Como se puede observar con el uso de anotaciones se indica a qué entidad en la base de datos hace referencia la clase. También están las anotaciones para el identificador, es decir la llave primaria, y el método de autogenerado que se quiera utilizar. Si no se le indica explícitamente el nombre de la columna que mapea cada atributo de la clase, el *framework* intentará mapear a través del nombre del atributo, la columna de la entidad.

El hecho de utilizar mapeo objeto-relacional es una gran ventaja, pues en la codificación de todo el sistema, el acceso y tratamiento de toda la información de la base de datos se hará a través de objetos y atributos de estos objetos. Si a la ventaja antes mencionada se le agrega el hecho de que se utiliza el *framework* Spring-Data, el uso de repositorios, que son una abstracción del uso del manejador de transacciones de Hibernate, esto permitirá un desarrollo mucho más ágil y se puede enfocar esfuerzo en donde realmente se debe aplicar, en el desarrollo de toda la lógica de negocio y en la interfaz de usuario.

Los repositorios se definen como interfaces. En estas interfaces se definen métodos con un estándar de nombrado, para que el *framework* pueda ejecutar consultas a la base de datos. Los repositorios de Spring-Data ya tienen implementados los métodos básicos como por ejemplo: save, count, delete, exists, findAll, findOne. Las consultas pueden definirse solamente con el de los métodos de la interfaz, y el *framework* interpretará correctamente si se siguen los lineamientos de nombrado; pero también se puede utilizar *querys* anotados para consultas más complejas.

Figura 32. Repositorio JPA con Spring-Data para la entidad Tienda

```
@Transactional
public interface TiendaRepository extends JpaRepository<Tienda, Long>{
    Tienda findByNombreAndEliminado(String nombre, boolean eliminado);

    @Query("SELECT T from tienda T WHERE T.nombre = :nombre AND T.eliminado =
false")
    Tienda findByNombreAndNotEliminado(@Param("nombre") String nombre);

    List<Tienda >findByEliminado(boolean eliminado);
}
```

Fuente: elaboración propia.

En el ejemplo del repositorio se puede observar que para ejecutar una consulta a la base de datos, no se necesita implementar nada de código ni de lógica de acceso a la base de datos. También se observa que el primero y el segundo método ejecutan exactamente la misma consulta, pero de dos formas diferentes. Esto demuestra que el uso del *framework* Spring-Data hará generar código de forma más eficiente.

En el repositorio también se debe resaltar el uso de la anotación *@Transactional*; el uso de esta anotación permite que sea el *framework* de JPA quien se encargue de las transacciones, y que se pueda tener un contexto que asegure la integridad en la información almacenada en la base de datos. Con estos dos paquetes, *model* y *repository*, se completa la capa de persistencia de datos.

La siguiente capa que se analizará es la de servicios. Estos son los que contienen toda la programación de la lógica de negocio. Estos servicios utilizan los repositorios para tener acceso a la base de datos y a la información almacenada en ella. La capa de servicio estará siempre compuesta por una



interfaz que se encarga de exponer las funcionalidades que brinda el servicio, y por una clase que implementa dicha interfaz, en donde se programa toda la lógica de ejecución de cada método.

Figura 33. **Interfaz e implementación del servicio de catálogo de productos**

```
public interface CatalogoProductoService {
    public void agregarCategoria(Categoria categoria);

    public void agregarProducto(Producto producto);

    public List<Producto> buscarProductosPorCategoria(Categoria categoria);

    public Producto buscarProductoPorCodigo(String codigo);

    public Producto buscarProductoPorId(Long id);

    public Categoria buscarCategoriaPorId(Long id);

    public List<Categoria> buscarTodasCategorias();
}

@Service("catalogoProductoSvc")
public class CatalogoProductoServiceImpl implements CatalogoProductoService{

    @Autowired
    private ProductoRepository productoRepository;

    @Transactional(readOnly = true, propagation = Propagation.REQUIRED)
    public Producto buscarProductoPorCodigo(String codigo){
        return productoRepository.findByCodigo(codigo);
    }

    @Transactional(readOnly = true, propagation = Propagation.REQUIRED)
    public Producto buscarProductoPorId(Long id){
        Producto producto = productoRepository.findOne(id);
        if(producto != null){
            if(producto.getCategoriaBean() != null)
                producto.getCategoriaBean().getIdCategoria();
        }
        Return producto;
    }

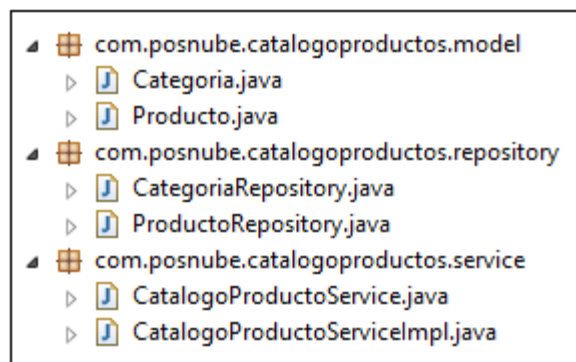
    @Transactional(readOnly = false, propagation = Propagation.REQUIRED)
    public void guardarProducto(Producto producto){
        productoRepository.saveAndFlush(producto);
    }
}
```

Fuente: elaboración propia.

El *framework* de Spring-Data crea una instancia de cada uno de los repositorios y los aloja en el contenedor de Spring. Es por esto que en el servicio basta con inyectar el repositorio por medio de la anotación `@Autowired` y ya se podrá hacer uso del repositorio. En el servicio también se utilizan las transacciones, siguiendo con el manejo integro de los datos. Para que el servicio esté disponible para otros componentes del sistema, se utiliza la anotación `@Service` del *framework* Spring, lo que hará que en el contendor se cree una instancia de este servicio.

Cada uno de los componentes del sistema contiene estas dos capas: servicios y persistencia de datos, es decir los tres paquetes: *model*, *repository* y *service*, además de un paquete para la capa de transferencia de datos. Esta estructura de paquetes se replica en cada componente, por lo que cada uno es independiente y desacoplado del resto del sistema.

Figura 34. **Estructura interna de la implementación del componente de catálogo de productos**



Fuente: elaboración propia.

La capa de controlador de la vista está implementada en el paquete `web.jsf` del proyecto. Este paquete contiene todas las clases que se encargan de controlar la información de entrada y salida de la vista. Siguiendo el estándar de JSF, se utilizan `ManagedBeans` en las clases del paquete, y además se cargan al contenedor de Spring por medio de la anotación `@Component`. En estas clases se integra un nuevo concepto, es el alcance de los beans en el contenedor de Spring; para el caso de los controladores de las vistas se utiliza alcance de sesión.

Como el controlador de la vista se aloja en el contenedor de Spring, es muy sencillo inyectar el servicio principal del sistema. El controlador de la vista tiene acceso al servicio, y es a través del servicio y solo a través de él que puede ejecutar alguna acción de la lógica del negocio. En la figura 35 se muestra un ejemplo de un controlador para mostrar el listado de productos pertenecientes a una categoría específica.

En el controlador de la vista se puede observar que el servicio que se inyectó es el principal, tal y como se definió en la arquitectura y en el diagrama de componentes del sistema. El `CoreService` es el encargado de gestionar todas las transacciones que se realicen en el sistema, y es quien utilizará los servicios expuestos por todos los componentes que conforman el sistema.

Figura 35. ManagedBean para el catálogo de categorías de productos

```
@Component("consultaProductoCategoriaBean")
@Scope(value = "session")
@ManagedBean
public class ConsultaProductoCategoriaBean {

    @Autowired
    private PosCoreService posCoreService;

    private List<Categoria> categorias;

    private String idCategoria;

    private Categoria categoria;

    private List<Producto> productos;

    @PostConstruct
    public void init(){
        categorias = posCoreService.buscarTodasCategorias();
        categoria = null;
        productos = new ArrayList<Producto>();
    }

    public void handleCategoriaChange(){
        if(idCategoria != null && !idCategoria.trim().equals("")){
            categoria = posCoreService.buscarCategoriaPorId(
                Long.parseLong(idCategoria)
            );
            if(categoria != null)
                productos =
                    posCoreService.buscarProductosPorCategoria(
                        categoria
                    );
        }
    }

    /***** Incluir Getters y Setters para cada atributo de la clase *****/
}
```

Fuente: elaboración propia.

Para completar el análisis de la codificación de cada una de las capas de la arquitectura, solamente hace falta la capa de la vista. En esta capa se construirá toda la interfaz del sistema con el usuario, utilizando el estándar JSF

y el *framework* Primefaces que es una abstracción de JSF, es decir, está desarrollado con base en este estándar.

Un ManagedBean es un Java Bean administrado por el *framework* JSF, y es accedido desde una página JSF. En el ManagedBean se definen los *getters* y *setters* de los atributos que se utilizarán en la vista. Este concepto es muy importante, pues a través de estos Beans la vista, una página JSF, se comunicará desde el lado cliente con todo el entorno del lado del servidor. Se puede tener acceso tanto a propiedades como a métodos del Bean.

Primefaces es un *framework* que provee componentes de interfaz de usuario enriquecidos. Todos estos componentes extienden de los componentes básicos de JSF por lo que siguen el estándar. Estos buscan crear mejores interfaces de usuario con mayor eficiencia. En la figura 36 se puede observar una página JSF utilizando tanto los componentes del mismo *framework* JSF como los componentes de Primefaces.

En una pequeña porción de código se puede crear una interfaz de usuario con componentes enriquecidos, lo cual permite optimizar el tiempo de desarrollo. A la larga todas estas tecnologías ayudan a ser más eficientes en la creación del sistema, por lo que todo esto tiene un impacto positivo en el costo de un sistema de software. Si se une este punto positivo con la reducción de costos en la infraestructura tanto del servidor como del cliente, se obtiene un producto de calidad y sobre todo accesible para las pymes en cuanto a costo, contribuyendo en el desarrollo de su cadena de valor.

Figura 36. Vista JSF para el listado de productos utilizando el ManagedBean de categorías de productos

```

<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:p="http://primefaces.org/ui" template="template.xhtml">
    <ui:define name="content">

        <p:ajaxStatus onstart="statusDialog.show();"
            onSuccess="statusDialog.hide();"/>

        <p:dialog modal="true" widgetVar="statusDialog"
            header="Procesando..." draggable="false" closable="false">
            <p:graphicImage value="/design/ajaxloadingbar.gif"/>
        </p:dialog>

        <h:form id="formMain">

<p:outputPanel id="panelMain">
<br/>
<p:panel style="width:850px; height:650px;" >
    <f:facet name="header">
        <h:outputText value="ConsultarProductos"/>
    </f:facet>
<br/>
<p:panelGrid columns="2">
    <h:outputText value="Categoría"/>
    <p:selectOneMenu
        value="#{consultaProductoCategoriaBean.idCategoria}"
        <f:selectItem
            itemLabel="Seleccione una categoría" itemValue=""/>
        <f:selectItems
            value="#{consultaProductoCategoriaBean.categorias}"
            var="categoria" itemLabel="#{categoria.nombre}"
            itemValue="#{categoria.idCategoria.toString()}" />
        <p:ajax
            listener="#{consultaProductoCategoriaBean.handleCategoriaChange}"
            update="tablaProductos"/>
    </p:selectOneMenu>
</p:panelGrid>
<br/>
<p:dataTable value="#{consultaProductoCategoriaBean.productos}"
    var="producto" id="tablaProductos"
    paginator="true" rows="5" rowsPerPageTemplate="5,10,15"
    paginatorTemplate="{FirstPageLink} {PreviousPageLink}
    {CurrentPageReport} {NextPageLink} {LastPageLink}">
    <f:facet name="header">
        <h:outputText value="Productos"/>
    </f:facet>
    <p:column>
        <f:facet name="header">
            <h:outputText value="Código"/>
        </f:facet>
        <h:outputText value="#{producto.codigo}"/>
    </p:column>

```

Continuación de la figura 36.

```
        <p:column>
            <f:facet name="header">
                <h:outputText value="Nombre"/>
            </f:facet>
            <h:outputText value="#{producto.nombre}"/>
        </p:column>
    </p:dataTable>
</p:panel>
</p:outputPanel>
</h:form>
<br/><br/>
</ui:define>
</ui:composition>
```

Fuente: elaboración propia.

### 3.5. Pruebas

Las pruebas de software se encargan de validar el correcto funcionamiento de un sistema y la calidad con la que fueron codificados todos los requerimientos. Existen diferentes tipos de pruebas que se le pueden realizar a un sistema de software, y estos tipos dependen de lo que se quiera validar. Uno de los métodos más utilizados es el V; este válida la funcionalidad y la calidad del sistema a distintos niveles de abstracción.

Para cada etapa en el ciclo de desarrollo del software existe un tipo de pruebas para la validación del mismo. Desde el nivel más abstracto, el análisis de los requerimientos se valida con pruebas de funcionamiento y usabilidad con los usuarios finales. El diseño de alto nivel y la arquitectura se validan con las pruebas de integración. El diseño detallado y la codificación se validan con las pruebas unitarias.

JUnit es un *framework* que permite ejecutar clases Java de una forma controlada, y permite evaluar el funcionamiento de cada uno de los métodos de la clase. Este *framework* permite realizar las pruebas unitarias del sistema. Además de utilizar JUnit, se utiliza Spring JUnit que permite ejecutar las pruebas utilizando todo el contexto del contenedor. Esta combinación permite ejecutar también, pruebas de integración, pues se pueden ejecutar los servicios que consumen la capa de persistencia de datos.

Como las pruebas se ejecutan dentro del contexto del contenedor de Spring, se puede inyectar los servicios para comprobar su funcionamiento. También se pueden hacer las pruebas pertinentes para verificar la transaccionabilidad del sistema, y asegurar la integridad de los datos almacenados. En la figura 37 se muestra la configuración de una clase para poder ejecutar pruebas unitarias y de integración.

Figura 37. Configuración de una clase de pruebas con JUnit y Spring

```
@RunWith(SpringJUnit4ClassRunner.class)
@Transactional(transactionManager="txManager", defaultRollback
= false)
@ContextConfiguration(locations={"/applicationContext.xml"})
public class TestPos {

    @Autowired
    private PosCoreService posCoreService;

    @Test
    public void testBuscarCategorias(){
        List<Categoria> categorias =
            posCoreService.buscarTodasCategorias();
        for(Categoria categoria : categorias){
            System.out.println(categoria.getNombre());
        }
    }
}
```

Fuente: elaboración propia.



Para las pruebas de funcionalidad, que verifican el correcto funcionamiento y la correcta implementación de la lógica del negocio, se ejecutaron pruebas alfa, utilizando el método de caja negra. Las pruebas alfa se ejecutan por el usuario y el desarrollador como observador, estas pruebas son controladas y simulan un ambiente de producción. El método de caja negra establece que las pruebas se realizan con base en las entradas y las salidas del sistema, ignorando el funcionamiento interno. Estas pruebas se realizaron durante una semana, lo cual generó cambios y ajustes tanto en el funcionamiento como en la interfaz de usuario.

Entre las correcciones y observaciones más destacadas y generalizadas que se obtuvieron en la fase de pruebas alfa están:

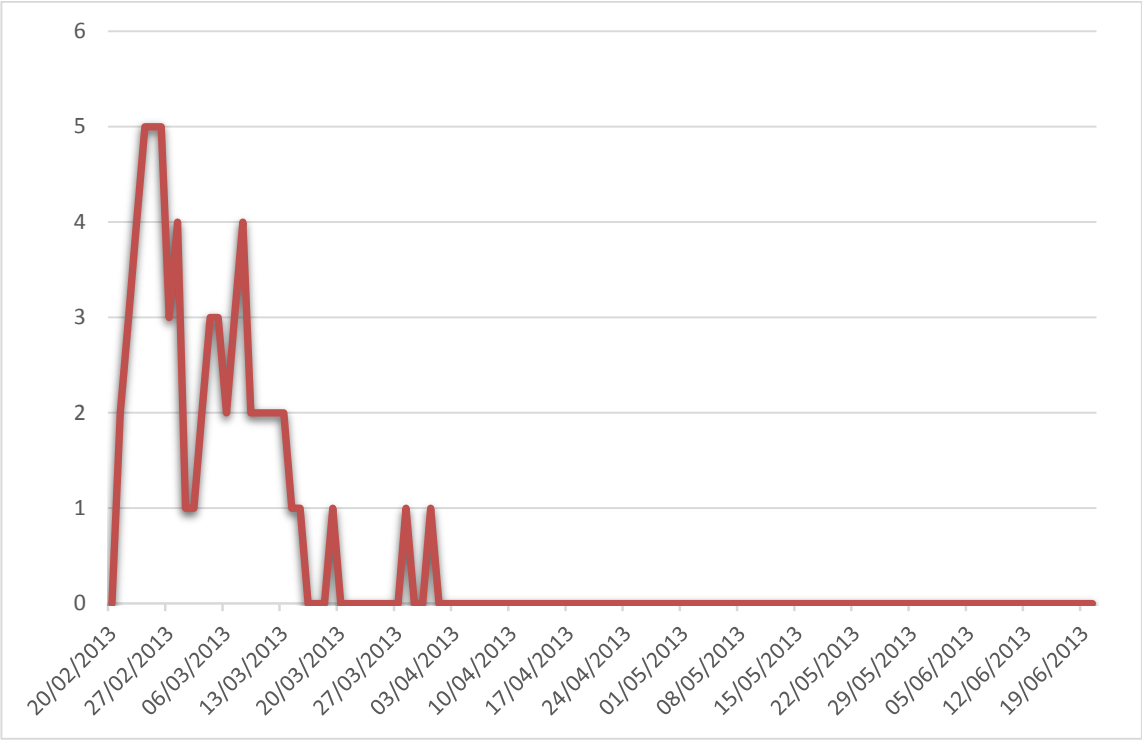
- Filtros de búsqueda rápida para los catálogos
- Búsquedas avanzadas para la mayoría de catálogos
- Ajuste del método de promedio ponderado para el costo de los productos
- Mayor utilización del teclado, reduciendo el del mouse
- Tipografía de la aplicación. Utilizar arial para mayor legibilidad
- El proceso de facturación debe ser más intuitivo y simplificado
- Mejorar aspecto visual de los reportes

Luego de atender a los cambios y correcciones obtenidas en las pruebas alfa, se procedió a colocar la aplicación en un ambiente de producción. A lo largo de cuatro meses se ejecutaron pruebas beta, las cuales son realizadas por el usuario, pero ya en su entorno de trabajo y sin observadores de ningún tipo. En este momento el usuario ya está utilizando el sistema de forma cotidiana, pero se mantiene cierto control a distancia por los posibles errores que pueda presentar el sistema.

Durante estos cuatro meses se obtuvieron algunas mejoras y correcciones como por ejemplo: algunos problemas de base de datos, mejoras en la usabilidad del sistema, algunas validaciones importantes para el correcto funcionamiento. En la figura 38 se puede observar la evolución, a lo largo de los cuatro meses, de la cantidad de reportes de errores, funcionamiento incorrecto o mejoras al sistema. Entre los reportes más concurrentes están:

- Datos de mayor longitud de lo que se estableció en la base de datos.
- El rollback del servicio de emisión de factura no se ejecutaba correctamente.
- Validación del Nit del cliente.
- Agregar combinaciones de teclas para ejecutar acciones específicas.
- Manejar toda operación monetaria con dos decimales para evitar desajustes por aproximaciones.
- Aumentar el tiempo de expiración de la sesión.

Figura 38. Cantidad de errores reportados durante los cuatro meses de pruebas beta



Fuente: elaboración propia.



## **4. IMPLANTACIÓN Y ANÁLISIS DE RESULTADOS**

La implantación es una etapa del ciclo de vida del software, la cual se realiza luego de haber concluido con la etapa de pruebas. En el capítulo tres de este documento se abarcaron todas las etapas previas a la implantación: toma de requerimientos, análisis, diseño, codificación y pruebas. Para completar el ciclo del software hacen faltan dos etapas: una es la implantación y otra el mantenimiento. En este trabajo se llegará hasta la etapa de implantación, pues el mantenimiento es un proceso continuo y cíclico, por lo que aún terminando este trabajo, no se finalizaría de documentar este proceso.

La etapa de implantación está compuesta por varias tareas, como la entrega e instalación del sistema en la infraestructura definida, capacitación y formación de los usuarios, instalación de hardware si fuera necesario y todas las acciones que permitirán que el sistema de software pueda ser utilizado y esté disponible a los interesados.

En este capítulo se definirán todos los recursos necesarios para la implantación del sistema de punto de venta en la nube, los procesos para realizar dicha implantación y se analizan algunos resultados obtenidos luego de haber puesto el sistema en marcha, y después de algunos meses que el sistema ha sido utilizado en un punto de venta.

### **4.1. Requerimientos de hardware e infraestructura para la implantación**

En la infraestructura definida para el sistema de punto de venta en la nube se analizaron cada uno de los requerimientos de software y hardware para el

correcto funcionamiento de dicho sistema. Existen dos lados en la infraestructura, el lado del cliente y el lado del servidor. En la definición de cada uno se buscó la eficiencia tanto en rendimiento como en costo para beneficiar a las pymes que utilicen este sistema.

Para el lado del servidor, solamente se necesita un servidor en el cuál se alojará la aplicación, y desde el cual estará disponible el sistema. Siguiendo con la línea de desarrollo en la nube, el servidor que se utiliza en este trabajo es un servidor virtualizado y que se adquiere como un servicio en la nube. Existen muchos proveedores de este tipo de servicios, y el costo de alquiler de dichos servidores es variado.

Para este trabajo se adquirió un servidor con las siguientes características: Procesador de 4 núcleos, 256 MB de RAM y disco duro de 10 GB. Este servidor tiene un costo de alquiler de \$ 10,95 mensuales, más la cantidad de datos de salida transferidos en el mes, que su costo es de \$ 0,12 por GB transferido. Esto genera un costo promedio de \$ 12,00 mensuales por el alquiler del servidor en la nube. El servidor se adquirió con el sistema operativo Debian 6.0 Squeeze, que es gratuito y no requiere la compra de ningún tipo de licenciamiento.

En este servidor se necesita instalar JDK 1.6, que es la herramienta necesaria para que se puedan ejecutar aplicaciones Java. También se instaló el motor de la base de datos, MySQL 5.5. Para poder desplegar la aplicación Java Web, se instaló el servidor de servlets Apache Tomcat 7.0.16, el cual escucha peticiones en un puerto específico, y las procesa según la lógica de la aplicación correspondiente. Uno de los puntos en común de todas las herramientas mencionadas anteriormente, es que todas ellas son Open Source, lo que significa que no implica ningún costo el utilizarlas en el servidor.

El hecho de que el sistema de punto de venta es un servicio en la nube, reduce los requerimientos de hardware y software del lado del cliente. Lo único que se necesita del lado del cliente es una computadora con un navegador de internet instalado, y conexión a internet disponible. De los dos requerimientos, la computadora es una inversión inicial y el servicio de internet es un costo mensual.

Actualmente, en Guatemala las pymes pueden tener un fácil acceso a los proveedores de servicios de telecomunicaciones. En esta área, el país está en un buen punto de desarrollo. Para una conexión a internet de 512 Kbps, el costo actual en Guatemala es de aproximadamente Q. 250,00 mensuales. Tomando en cuenta que las conexiones a internet cada día son más frecuentes y necesarias para el desarrollo de las pymes, este costo puede ser diluido en otras áreas de la empresa, pues más allá del punto de venta, también se verían beneficiados con el acceso a internet.

Como se puede observar, los servicios en la nube contribuyen con el desarrollo de sistemas de bajo costo; y si a esto se une la estrategia de afiliación para la distribución del software, se puede generar sistemas de valor para las pymes y que no causen un impacto tan fuerte en la inversión inicial, ni en el mantenimiento del sistema. Este escenario, también beneficia al proveedor del servicio, pues le permite distribuir eficientemente su producto o servicio, aumentando el margen de ganancia al no sufrir un impacto tan grande en el costo del sistema.

#### **4.2. Implantación del producto**

Una vez se tengan todos los requerimientos de infraestructura, es decir, el servidor y todas las herramientas necesarias instaladas en el mismo, se

procede a instalar la aplicación. Para la ejecución del servidor Apache Tomcat se necesita haber realizado algunas configuraciones en el sistema operativo y en la base de datos:

- Definir la variable de entorno `$JAVA_HOME` apuntando a la dirección en donde se ha instalado el JDK. Esto es necesario para el funcionamiento del servidor Apache Tomcat.
- Crear un usuario en la base de datos con el cual la aplicación se conectará a la misma. Este usuario debe tener los permisos adecuados sobre la base de datos del punto de venta.
- Por seguridad se implementará un firewall en el servidor. Se debe asegurar que el puerto en el que el servidor Apache Tomcat escuchará las peticiones de los clientes, esté abierto y permita la entrada y salida de datos.

Con el *framework* Maven, el cual gestiona la construcción de la aplicación, se obtiene el instalador del sistema, un archivo con extensión `.war`. Este archivo será el que utilizará el servidor Apache Tomcat para desplegar la aplicación y hacerla disponible para los usuarios en una URL específica. El archivo `.war` se debe colocar en la carpeta `webapps` del servidor Apache Tomcat, y luego ejecutar el archivo `bin/startup.sh` para arrancar los servicios.

Hasta este momento, un usuario fue quien ejecutó las pruebas beta, por lo que es necesario realizar capacitaciones para todo el personal de la empresa del cliente, para que puedan utilizar el sistema y obtener el mayor beneficio de él. Las capacitaciones se dividieron en dos sesiones, cada sesión de 3 horas. La primera sesión se realizó con tres personas que utilizan el rol de cajero. La segunda sesión fue con dos personas, una de ellas con rol de gerente, y la otra persona con los roles de gerente y administrador del sistema.



Después de impartir las capacitaciones el sistema comenzó a ser utilizado por todos los usuarios y el entorno natural de trabajo de cada uno de ellos y dentro de sus actividades cotidianas. Durante las siguientes dos semanas los usuarios siguieron generando algunas dudas en cuanto al funcionamiento del sistema, las cuales fueron solventadas, hasta estabilizar a prácticamente 0 incidencias por día.

Como ya se ha mencionado, en este trabajo, se llega hasta la etapa de implantación, pues el mantenimiento es un proceso que se sigue dando a lo largo de toda la vida útil del sistema. Es por esto que con esta sección se llega al final del ciclo de desarrollo del sistema de punto de venta en la nube, el cual es el objeto de estudio de este trabajo. Solamente hace falta analizar algunos resultados obtenidos en el tiempo que lleva en uso el sistema.

#### **4.3. Análisis del rendimiento del POS como una etapa en la cadena de valor**

Después de haber finalizado con el proceso de desarrollo del sistema de punto de venta en la nube, haber implantado el sistema, haber ejecutado las capacitaciones correspondientes, se puede obtener información del comportamiento del mismo y, también del impacto que ha tenido en la cadena de valor del negocio. La información que se muestra a continuación corresponde a un período de 6 meses, entre los cuales están los 4 meses de pruebas beta que se ejecutaron.

El costo mensual del sistema, desde el punto de vista del proveedor del servicio en la nube, está compuesto por el costo de alquiler mensual del servidor y la ganancia marginal que pueda obtener. Por la naturaleza de este trabajo, en la empresa que se implantó el sistema no se tomará en cuenta la

ganancia marginal, pues fue de mutuo acuerdo que la empresa colaboraría en todo el proceso de desarrollo de software, a cambio de obtener el servicio a un precio basado únicamente en el costo de operación.

Basado en lo anterior, se analiza cuál es el costo que le representa el funcionamiento del sistema al negocio que lo ha implantado como parte de su proceso y su cadena de valor. Mensualmente el negocio debe invertir en la conexión a internet y en la afiliación al sistema, entre otras cosas. En la tabla VII se observa el detalle de la inversión mensual del negocio en el sistema de punto de venta en la nube.

Tabla VII. **Costo mensual de operación del sistema de punto de venta en la nube**

<b>Concepto</b>	<b>Cantidad</b>	<b>Costo unitario</b>	<b>Costo total</b>
Afiliación al sistema de punto de venta	1	Q. 100,00	Q. 100,00
Servicio de internet	1	Q. 250,00	Q. 250,00
Depreciación del equipo de computo	2	Q. 150,00	Q. 300,00
<b>TOTAL</b>			<b>Q. 650,00</b>

Fuente: elaboración propia.

Con el uso cotidiano del sistema, el negocio comenzó a ver resultados muy satisfactorios. Los beneficios y utilidades que el negocio ha obtenido pueden analizarse desde dos puntos de vista: El costo directo que ha podido reducir, y el tiempo en que obtiene resultados e información para toma de decisiones.

El costo directo que se ha podido reducir se tiene en función del tiempo de trabajo invertido por diferentes personas, con diferentes perfiles, dentro de la organización. Por ejemplo: una persona se encarga de tomar todo el registro de ventas y trasladarlo a una hoja de cálculo, este proceso lo realiza en los primeros días del mes, con la información del mes que recién termina. Luego de este proceso, un gerente consolida la información y genera los reportes de ventas del mes. También se tiene el consumo de tiempo en el control de inventarios, y la actualización de esta información en las hojas de cálculo.

Estas actividades representan un costo para el negocio, que el sistema de punto de venta en la nube ha ayudado a reducir. En la tabla VIII se presenta el reporte que el negocio ha provisto, con cálculos promedio del costo mensual que han logrado reducir, o bien enfocarlo en otras actividades que les generen mayores utilidades.

Tabla VIII. **Promedio mensual de reducción de costo al utilizar el sistema de punto de venta en la nube**

<b>Concepto</b>	<b>Cantidad</b>	<b>Costo unitario</b>	<b>Costo total</b>
Trabajo de digitador para registrar los movimientos de ventas	1	Q. 800,00	Q. 800,00
Trabajo de gerencia para integrar los resultados y cuadrar inventarios	1	Q. 700,00	Q. 700,00
Generación de reportes de ventas mensuales	1	Q. 600,00	Q. 600,00
<b>TOTAL</b>			<b>Q. 2 100,00</b>

Fuente: elaboración propia

El segundo gran beneficio que el negocio ha podido obtener del uso del sistema de punto de venta en la nube, es el tiempo en el que se puede obtener resultados e información, con lo cual se puede generar conocimiento. Antes de utilizar el sistema, el reporte de ventas del mes vencido lo completaban entre el 15 y el 20 del mes en curso. Con el uso del sistema, este tiempo se ha podido reducir a menos de la mitad, pues actualmente tiene la capacidad de completar este reporte entre el 5 y el 10 del mes en curso.

Para la toma de decisiones que realiza la Gerencia, el hecho de que puedan tener los reportes más rápido, representa una ventaja competitiva para reaccionar ante los comportamientos cambiantes del mercado. También en el aspecto de supervisión y control, se pueden detectar a tiempo posibles anomalías en la operación de sus puntos de venta.

En general, el cliente ha encontrado en el sistema de punto de venta en la nube, una herramienta altamente eficiente y que le ha generado un gran valor agregado a su cadena de valor; teniendo un impacto positivo, no solo en el punto de venta directamente, sino en otras áreas del negocio, es decir, impactando en todo el proceso del negocio.

## CONCLUSIONES

1. En Guatemala, las pequeñas y medianas empresas contribuyen con un gran porcentaje del PIB y aportan la mayoría de las oportunidades de empleo.
2. Las tecnologías de información y comunicación son vitales para el crecimiento y desarrollo de las empresas en un mercado completamente globalizado.
3. Las empresas, por muy pequeñas que sean, tienen una cadena de valor, sencilla o compleja, definida por el giro de su negocio, y los puntos de venta son una parte fundamental en esta cadena de valor.
4. Los sistemas computarizados son herramientas que colaboran directamente con el desarrollo, crecimiento, automatización y mejora continua de las empresas.
5. Un sistema de software provisto como un servicio en la nube, agrega dinamismo a las operaciones de un negocio, y puede aportar muchísimas ventajas competitivas a la empresa.

6. Las tecnologías que pueden ser utilizadas para el desarrollo de un sistema de software son muy variadas y existen muchos *frameworks* que facilitan el trabajo, o mejor dicho, ayudan para concentrarse en el desarrollo, específicamente de la lógica de negocio, dejando en segundo plano la implementación de todos los requerimientos no funcionales.
7. Al desarrollar un sistema de software, el diseño de la arquitectura y de los componentes que formarán parte del sistema, es una parte de vital importancia. Esto ayudará a crear un producto de calidad que sea funcional, usable, orientado al usuario, mantenible y escalable.
8. Un servicio en la nube debido al bajo monto de la inversión que un negocio debe realizar al adoptarlo, es una herramienta que puede ayudar al desarrollo de las pequeñas y medianas empresas en Guatemala; sobre todo porque este desarrollo está al alcance de las pymes.

## RECOMENDACIONES

1. Apoyar y fomentar la adopción de las tecnologías de información y comunicación en las pequeñas y medianas empresas en Guatemala, pues esto va a generar un desarrollo y una impulsión de los negocios no solo a nivel nacional, sino a nivel internacional en mercados globalizados.
2. Romper las barreras, que por falta de conocimiento, generan resistencia a adoptar cualquier tipo de tecnología en los procesos productivos y administrativos de las empresas. Capacitaciones constantes a los trabajadores pueden ayudar a esta causa.
3. Por muy pequeño que sea el negocio, es importante contar con un proceso de negocio, y con una cadena de valor que permita monitorear y controlar la productividad. Es necesario invertir recursos para la mejora continua en la ejecución de cada proceso.
4. Contar con una herramienta que permita automatizar el control de inventarios, ventas e ingresos de dinero en el negocio, aporta una ventaja competitiva. La adopción de un sistema de punto de venta en la nube es un paso, pero es importante mantenerse actualizados y en ciclo de mejora continua por parte de las empresas.





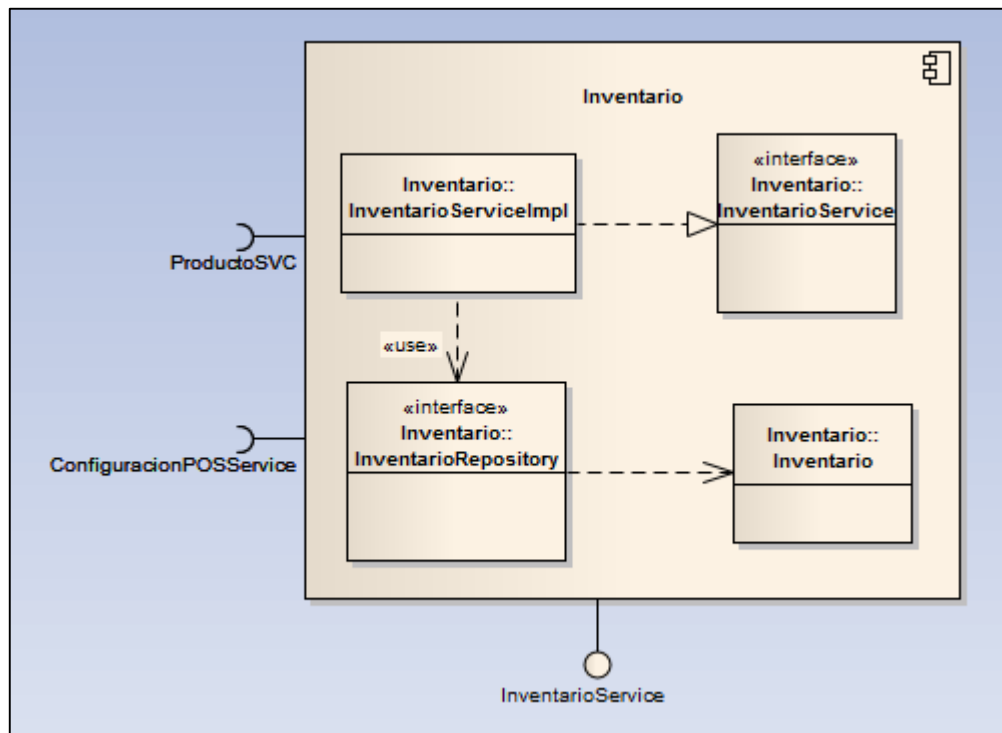
## BIBLIOGRAFÍA

1. MAR CORNELIO, Omar. *Tendencias en la distribución de software*. [en línea]. <<http://www.monografias.com/trabajos-pdf2/tendencias-distribucion-software/tendencias-distribucion-software.pdf>>. [Consulta: septiembre de 2012].
2. MONGE GONZÁLEZ, Ricardo. *TIC en las Pymes de Centroamérica: impacto de la adopción de las tecnologías de la información y la comunicación en el desempeño de las empresas*. [en línea]. <[http://web.idrc.ca/openebooks/214-7/#page\\_132](http://web.idrc.ca/openebooks/214-7/#page_132)>. [Consulta: mayo de 2012].
3. VARELA, Juan. *Internet supera la publicidad de diarios y cable en EEUU*. [en línea]. <<http://www.periodistas21.com/2011/04/internet-supera-la-publicidad-de.html>>. [Consulta: mayo de 2012].



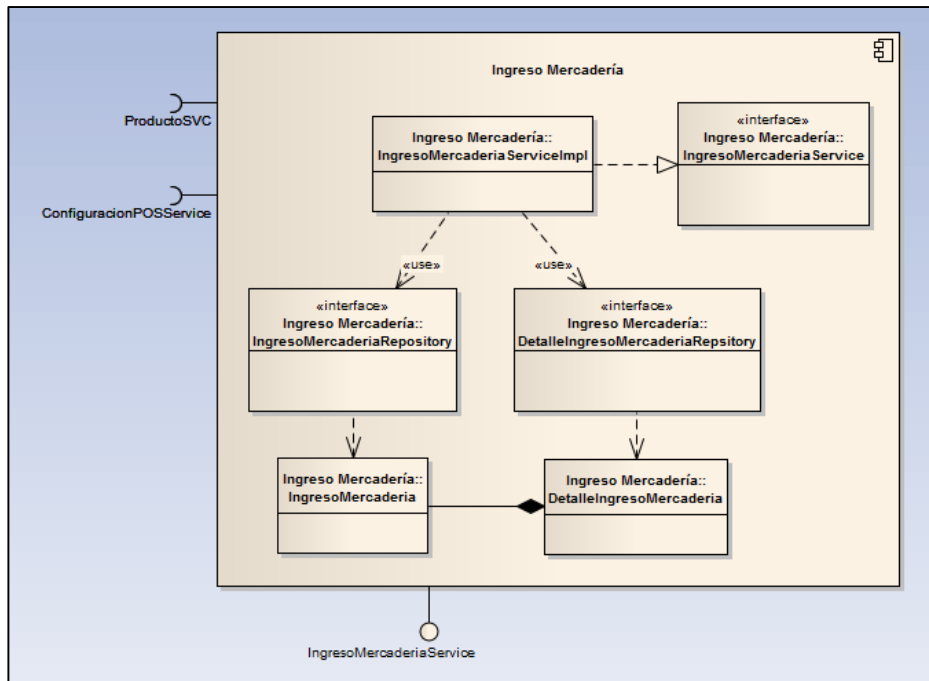
# APÉNDICES

Apéndice 1. Diagrama de clases del componente de inventario



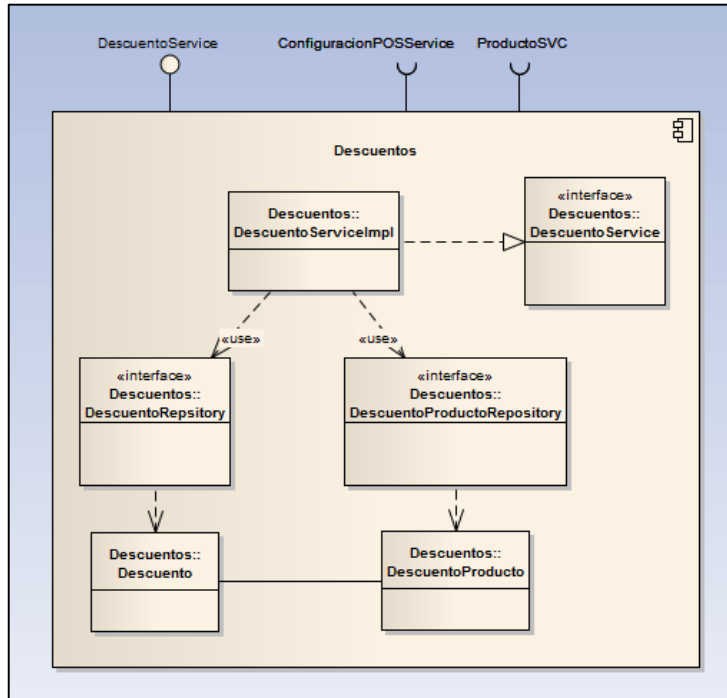
Fuente: elaboración propia.

Apéndice 2. **Diagrama de clases del componente de ingreso de mercadería**



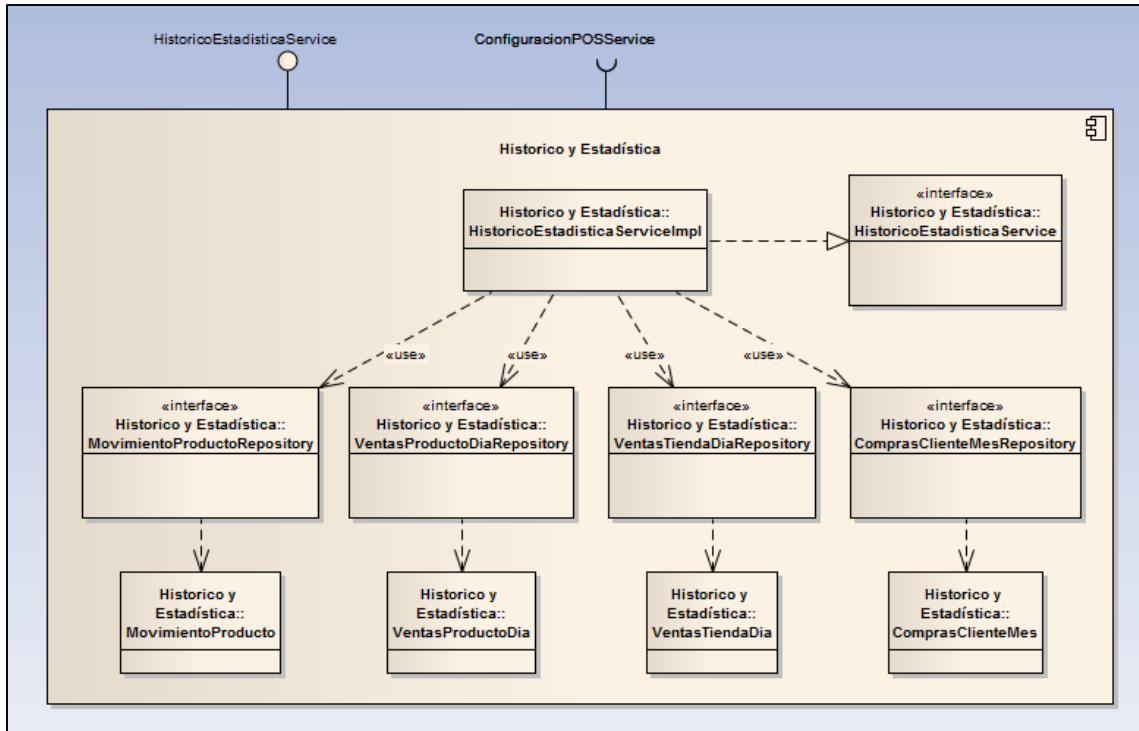
Fuente: elaboración propia.

Apéndice 3. **Diagrama de clases del componente de descuento**



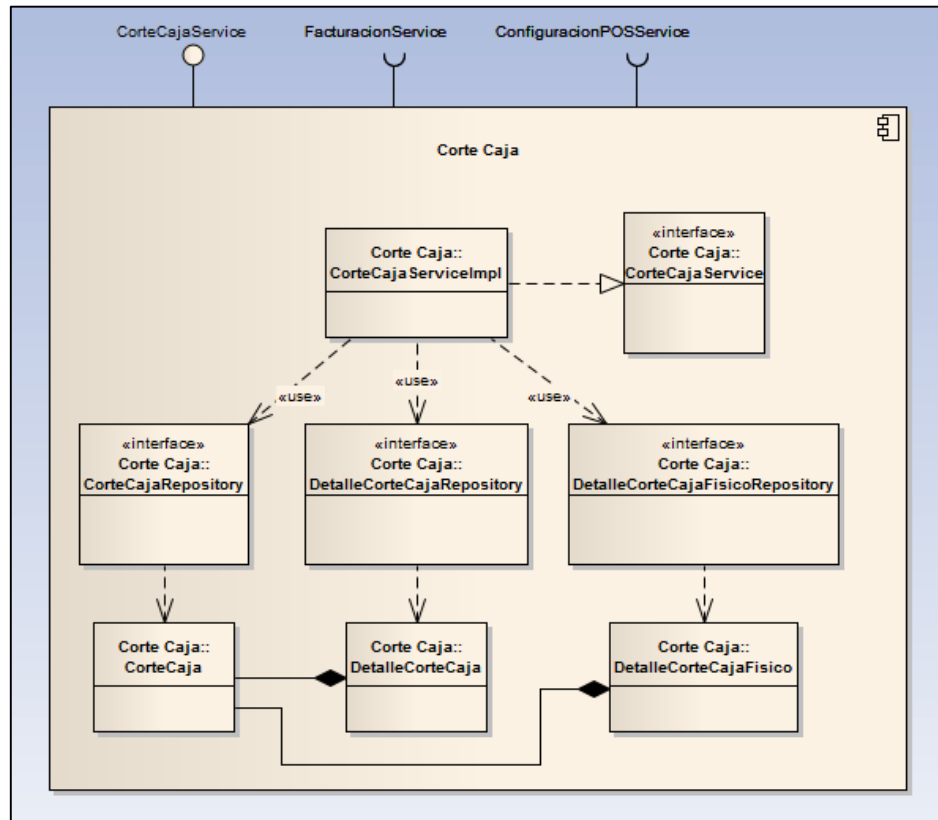
Fuente: elaboración propia.

Apéndice 4. Diagrama de clases del componente de histórico y estadística



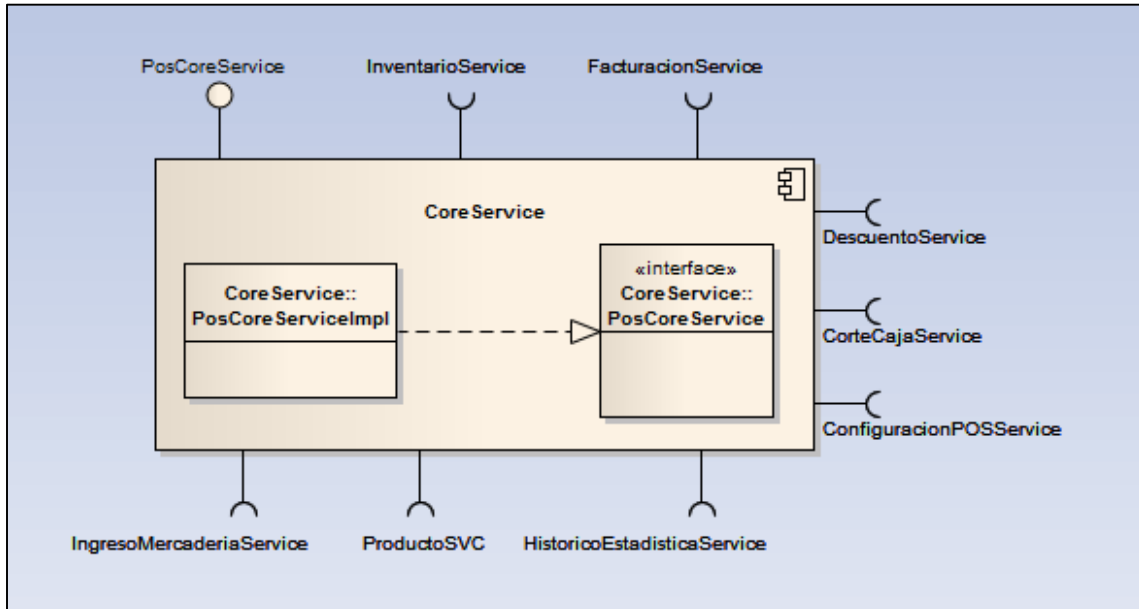
Fuente: elaboración propia.

Apéndice 5. Diagrama de clases del componente de corte de caja



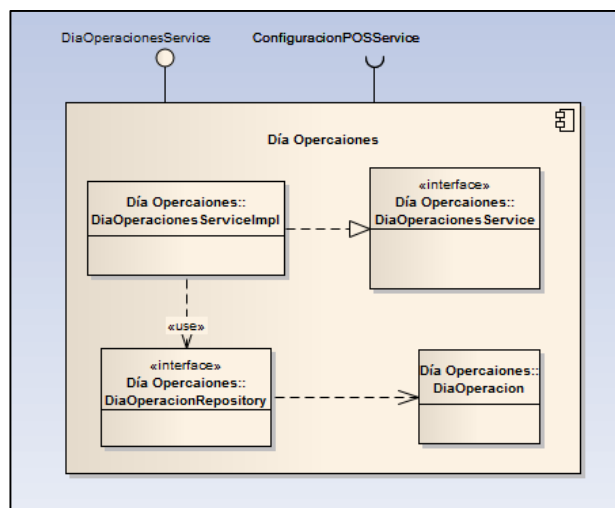
Fuente: elaboración propia.

Apéndice 6. **Diagrama de clases del componente principal del sistema**



Fuente: elaboración propia.

Apéndice 7. **Diagrama de clases del componente de día de operaciones**



Fuente: elaboración propia.