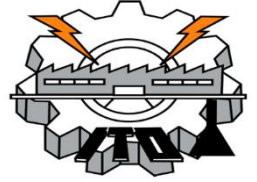




**TECNM**  
TECNOLOGICO NACIONAL DE  
**MEXICO**



# INSTITUTO TECNOLÓGICO DE OAXACA

**ASIGNATURA:**  
INGENIERIA DE SOFTWARE

**GRUPO:** ISB

**DOCENTE:**  
HERNANDEZ ABREGO ANAYANSI CRISTINA

**SEMESTRE:** SEXTO

**DOCUMENTACION TECNICA:**  
SISTEMA DE COMPRA-VENTA PARA LA  
MISCELANEA “EL PORTILLO”

**NOMBRE DEL EQUIPO:**  
SOFTWARE INC. CORPORATION

**INTEGRANTES:**  
FRANCO PEREZ DIANA  
GARCIA MENDOZA MARIA FERNANDA  
MARTINEZ RESENDIZ DIANA ARODI  
SANTIAGO GARCIA MARIA ALEJANDRA  
VASQUEZ SALINAS VICTOR MANUEL

**FECHA DE ENTREGA:** 03/06/2019

# Contenido

a) Definición general del proyecto .....	4
b) Especificación de requerimientos.....	5
c) Procedimientos de desarrollo .....	7
d) Procedimientos de instalación y prueba .....	12
2.- Arquitectura del sistema .....	14
a) Descripción jerárquica y de módulos .....	14
b) Dependencias externas.....	17
3.- Diseño del modelo de datos.....	21
4.- Descripción de procesos y servicios ofrecidos por el sistema.	23
Primera Iteración .....	23
Segunda Iteración .....	26
Tercera Iteración .....	29
5.- Documentación Técnica – Especificación API .....	33
6.- Conclusiones .....	33
7.- Bibliografía o referencias.....	34

## **Anexo 1. Lista de imágenes ..... 33**

<b>ILUSTRACIÓN 1: LISTA MAESTRA DE HISTORIAS DE USUARIO .....</b>	<b>6</b>
<b>ILUSTRACIÓN 2: DIAGRAMA DE DESPLIEGUE.....</b>	<b>14</b>
<b>ILUSTRACIÓN 3: DIAGRAMA DE MÓDULOS .....</b>	<b>15</b>
<b>ILUSTRACIÓN 4: CUADRO COMPARATIVO HERRAMIENTAS CASE .....</b>	<b>18</b>
<b>ILUSTRACIÓN 5: CUADRO COMPARATIVO DE SGBD .....</b>	<b>19</b>
<b>ILUSTRACIÓN 6: CUADRO COMPARATIVO DE SGBD .....</b>	<b>19</b>
<b>ILUSTRACIÓN 7: CUADRO COMPARATIVO DE SGBD .....</b>	<b>20</b>
<b>ILUSTRACIÓN 8: DIAGRAMA ENTIDAD-RELACIÓN .....</b>	<b>21</b>
<b>ILUSTRACIÓN 9: DIAGRAMA RELACIONAL .....</b>	<b>22</b>
<b>ILUSTRACIÓN 10: BPMN DE REGISTRO DE PRODUCTOS.....</b>	<b>23</b>
<b>ILUSTRACIÓN 11: BPMN DE INGRESO AL SISTEMA.....</b>	<b>25</b>
<b>ILUSTRACIÓN 12: BPMN DE REGISTRO DE NUEVO EMPLEADO .....</b>	<b>26</b>
<b>ILUSTRACIÓN 13: BPMN DEL REGISTRO DE PROVEEDORES .....</b>	<b>27</b>
<b>ILUSTRACIÓN 14: BPMN DE BÚSQUEDA Y EDICIÓN DE PROVEEDORES .....</b>	<b>28</b>
<b>ILUSTRACIÓN 15: BPMN PARA LA GESTIÓN DE COMPRAS .....</b>	<b>29</b>
<b>ILUSTRACIÓN 16: BPMN DEL REGISTRO DE VENTAS .....</b>	<b>30</b>
<b>ILUSTRACIÓN 17: BPMN DE LOS CORTES DE CAJA .....</b>	<b>31</b>
<b>ILUSTRACIÓN 18: BPMN PARA SEGURIDAD DE VENTAS .....</b>	<b>31</b>
<b>ILUSTRACIÓN 19: BPMN PARA CONSULTA DE GANANCIAS .....</b>	<b>32</b>

# 1.- Definiciones y especificación de requerimientos

## a) Definición general del proyecto

### ***Antecedentes***

**La Miscelánea Portillo** es una pequeña empresa que ha tenido buena aceptación en el mercado desde hace aproximadamente cuatro años. Esta pequeña empresa, al igual que varias en la región de San Antonio de Cal, Oaxaca donde manejan sus registros, ya sea de compras, ventas, inventario y demás de forma manual, lo que la mayoría de veces causa demasiados errores o redundancias a la hora de guardar la información, provocando en muchas ocasiones pérdidas para la empresa.

Gracias a la buena aceptación de los clientes, la Miscelánea Portillo ha buscado modernizar sus formas de registrar sus procesos, esto mediante la implantación de un sistema que ayude a la empresa a gestionar todas sus operaciones. Con la implantación de éste sistema, la empresa espera tener un mejor registro, principalmente de su inventario y el registro de las cuentas de sus clientes, esto con el fin de reducir las pérdidas que se tienen mediante el registro manual de estos procesos que se tienen actualmente.

### ***¿Por qué se desarrolla el proyecto?***

El desarrollo de este proyecto para la empresa "El Portillo" es principalmente para obtener un control sobre los clientes, productos, proveedores, determinar el monto de dinero a invertir en la siguiente compra a los proveedores, conocer los productos que se venden más, o los que se venden menos, esto con el fin de poder hacer una inversión extra para contar con más de éstos y tener la seguridad de que se venderán, conocer con certeza la cantidad de productos que se va a adquirir con los diferentes proveedores, conocer con exactitud cuánto es el saldo total disponible en recargas con el que se cuenta, buscar las deudas totales de un cliente y tener el control de compra y venta de productos, debido a los avances de la tecnología y la creciente competencia que ha provocado que la empresa pierda utilidades correspondientes, ya que la empresa ha generado ingresos muy bajos y ha adquirido pérdidas de ganancias.

### ***Nuestro objetivo general***

Generar un sistema de calidad como base a las necesidades del cliente para lograr una administración eficiente y ágil.

### ***Propósitos y Objetivos***

- Definir los principales procesos del cliente para el desarrollo del software.
- Desarrollar interfaces que obtengan las características fundamentales de los principales procesos del cliente.
- Crear interfaces gráficas amigables al contexto del cliente.
- Realizar diversas pruebas en el sistema para un mejor rendimiento.

### ***Usuarios del sistema***

Nuestro sistema de control de ventas está concebido como herramienta de trabajo y referencia, y su utilización, por tanto, tiene un marcado perfil técnico. Sus actores previstos son:

- **Administrador o Gerente:** Como una herramienta de seguimiento (edición y consulta) de las actividades principales de la empresa.
- **Empleado:** Como herramienta y guía en sus actividades (consulta y seguimiento de ventas).
- **Desarrolladores:** Para verificar la parte operativa del sistema (realizar las pruebas, correcciones y asesorías necesarias) y mantener su correcto funcionamiento.

## **b) Especificación de requerimientos**

### ***Normas y pautas establecidas para el sistema***

**La norma IEEE Std 610** define software como "programas, procedimientos y documentación y datos asociados, relacionados con la operación de un sistema informático". "Sistema de Control de Ventas" utiliza tal definición como referencia y, por tanto, considera la existencia de tres componentes en el conjunto de desarrollo.

- **Programas:** Conjuntos de instrucciones que proporcionan la funcionalidad deseada cuando son ejecutadas por el ordenador. Están escritos usando lenguajes específicos que los ordenadores pueden leer y ejecutar, tales como lenguaje ensamblador, Basic, Java, Ruby, PHP, Python.... Los programas también pueden ser generados usando generadores de programas.
- **Datos:** Elementos significativos portadores de la información que se pretende tratar mediante el empleo de programas.

- Documentos. Imprescindibles para que los actores dispongan de una explicación de cómo usar los programas (manuales de usuario y de operatoria) y de cómo intervenir sobre ellos para ajustar su funcionalidad.

Según la clasificación recogida en el documento **“Ingeniería de Software: metodologías y ciclos de vida” (INTECO 2009)**, la aplicación de este sistema puede considerarse como integrante de las siguientes categorías:

- Software de aplicaciones: se usan para proveer servicios a clientes y ejecutar negocios de forma más eficiente. El software de aplicaciones puede ser un sistema pequeño o uno grande integrado.
- Software de gestión: el proceso de la información comercial constituye la mayor de las áreas de aplicación del software. Los sistemas discretos (por ejemplo: nóminas, cuentas de haberes-débitos, inventarios, etc.) han evolucionado hacia el software de sistemas de información de gestión (SIG) que accede a una o más bases de datos que contienen información comercial. Las aplicaciones en esta área reestructuran los datos existentes para facilitar las operaciones comerciales o gestionar la toma de decisiones.

### ***Requisitos funcionales del sistema***

El desarrollo del sistema (mediante Programación Extrema o Metodología XP) está planeado en 3 iteraciones en donde se deben cumplir con 9 historias de usuario basadas en las necesidades básicas que fueron descritas por la empresa y que están enlistadas a continuación:

No de historia	Historias de usuarios	Puntos estimados	Prioridad	Iteración
1	Gestión de producto	2	2	1
2	Gestión de empleado	5	2	1
3	Login	3	1	1
4	Gestión de usuarios	2	1	1
5	Gestión de Proveedor	3	2	2
6	Gestión de compra	6	2	2
7	Gestión de venta	4	3	3
8	Seguridad de venta	3	3	3
9	Consultas de ganancia	4	2	3

*Ilustración 1: Lista maestra de historias de usuario*

### ***Información de Autoría***

El proyecto aquí documentado no forma parte de proyectos previos o pre-existentes, siendo así la primera vez que se crea un proyecto dentro del tipo de giro de la empresa. Todo su desarrollo es originalmente propiedad de la empresa **Software Inc. Corporation.**

### ***Alcances***

El proyecto contempla abarcar el registro de la cantidad de productos y su descripción de un pedido a proveedores, el registro y control del inventario y las ventas del día, además de la generación de estadísticas (inversión, ganancias netas, productos más y menos vendidos e inventario) al término de cada mes.

### ***Limitaciones***

En el área de productos, no se pueden hacer pedidos directamente a los proveedores, tan solo se podrán visualizar los datos de éstos tales como su número de teléfono, dependerá del usuario si es que se requiere hacer un pedido. No se tiene un enlace directo con el sistema de recargas, éstas se deben registrar manualmente.

## **c) Procedimientos de desarrollo**

### ***Herramientas***

Las herramientas que se utilizarán para el desarrollo de nuestro software, en base a las necesidades de la Miscelánea "El Portillo" son las siguientes:

- **JDeveloper (Programa para el diseño de interfaces y de el diagrama E-R)**

JDeveloper es un entorno de desarrollo integrado desarrollado por Oracle Corporation para los lenguajes Java, SQL, UML y otros.

Esta herramienta muestra la ventaja de facilidad de diseño de interfaces para el software, además de que posee diagramas UML para la implementación de la lógica del programa, así como la unificación del desarrollo de DDBB con el de la aplicación.

Con este programa se utiliza un ejemplo de las **Herramientas CASE**, puesto que se utiliza para representar los elementos clave del proceso de software, de modo que sea posible entenderlo mejor. Y proporcionar vínculos con descripciones de procesos que

ayuden a quienes estén implicados en el proceso de comprender las tareas que se requieren para llevar a cabo ese proceso.

- **Balsamiq MockUps (Programa para el en maquetado de las interfaces del usuario)**

Balsamiq MockUps es una herramienta rápida de baja fidelidad para la interfaz de usuario que reproduce la experiencia de dibujar en un bloc de notas o pizarra, pero utilizando una computadora. Pues te centra en la estructura y el contenido, evitando largas discusiones sobre colores y detalles que deberían aparecer más adelante en el proceso.

- **Postgresql (Gestor para la base de datos)**

Es un potente sistema de base de datos relacional de objetos de código abierto con desarrollo activo que le ha ganado una sólida reputación de confiabilidad, solidez de características y rendimiento.

El cual nos ayudara en cuanto a las tareas administrativas como la instalación, la gestión de usuarios, las copias de seguridad, restauraciones y el uso de prestaciones internas avanzadas, donde se aprecian las diferencias entre gestores de bases de datos.

### ***Lenguajes***

Los lenguajes que se implementaran para el desarrollo del software, tanto para la creación de interfaces, como para la gestión de la base de datos que estará implícita en el programa son:

- Java: Es un lenguaje de programación de propósito general, concurrente, orientado a objetos. En el cual utilizamos librerías o paquetes que nos ayudaran a la generación de las interfaces, pues el diseño de Java, su robustez, el respaldo de la industria y su fácil portabilidad han hecho este lenguaje un mayor crecimiento y amplitud de uso en distintos ámbitos.
- Sql: Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Este nos permite el manejo del álgebra y el cálculo relacional para efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella, este explota la flexibilidad y potencia de los sistemas relacionales y nos permitirá una gran variedad de operaciones.



### ***Metodología de desarrollo que se utiliza en el proyecto (Programación Extrema o XP)***

Metodología XP (Programación extrema o Extreme Programming) es una metodología de desarrollo de la ingeniería de software y fue elegida por nuestro grupo de trabajo debido a la fácil aplicación de este y que cada una de las fases son descritas de manera sencilla y entendible. Esta consta de cuatro fases que se desarrollan de la siguiente manera:

#### **1. Planeación**

La Metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y los coordinadores. El proyecto comienza recopilando las historias de usuarios (necesidades que el cliente expone a los desarrolladores para que estas sean resueltas) , las que constituyen a los tradicionales casos de uso. Una vez obtenidas estas historias de usuarios, los programadores evalúan rápidamente el tiempo de desarrollo de cada una.

#### **2. Diseño**

La Metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes:

Simplicidad: Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione.

Soluciones "Spike": Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba (llamados "Spike"), para explorar diferentes soluciones.

Recodificación ("Refactoring"): Consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de crearlo más simple, conciso y entendible. Las metodologías de XP sugieren recodificar cada vez que sea necesario.

Metáforas: XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto, así como guiar la estructura del mismo. Una buena metáfora debe ser fácil de comprender para el cliente y a su vez debe tener suficiente contenido como para que sirva de guía a la arquitectura del proyecto.

### 3. Codificación

**Disponibilidad del Cliente:** Uno de los requerimientos de XP es tener al cliente disponible durante todo el proyecto. No solamente como apoyo a los desarrolladores, sino formando parte del grupo. El Involucramiento del cliente es fundamental para que pueda desarrollarse un proyecto con la metodología XP. Al comienzo del proyecto, el este debe proporcionar las historias de usuarios. Pero, dado que estas historias son expresamente cortas y de "alto nivel", no contienen los detalles necesarios para realizar el desarrollo del código. Estos detalles deben ser proporcionados por el cliente, y discutidos con los desarrolladores, durante la etapa de desarrollo.

**Uso de Estándares:** XP promueve la programación basada en estándares, de manera que sea fácilmente entendible por todo el equipo, y que facilite la recodificación.

**Programación Dirigida por las Pruebas ("Test-Driven Programming"):** En las metodologías tradicionales, la fase de pruebas, incluyendo la definición de las pruebas, es usualmente realizada sobre el final del proyecto, o el final del desarrollo de cada módulo. La metodología XP propone un modelo inverso, primero se escribe las pruebas que el sistema debe pasar. Luego, el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas. Las pruebas a los que se refiere esta práctica son las pruebas unitarias, realizados por los desarrolladores. La definición de estas pruebas al comienzo condiciona o "dirige" el desarrollo.

**Programación en Pares:** XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador. Si bien parece que ésta práctica duplica el tiempo asignado al proyecto (y por ende, los costos en recursos humanos), al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas. El producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales.

**Integraciones Permanentes:** Todos los desarrolladores necesitan trabajar siempre con la "última versión". Realizar cambios o mejoras sobre versiones antiguas causan graves problemas, y retrasan al proyecto. Es por eso que XP promueve publicar lo antes posible las nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores. Idealmente, todos los días deben existir nuevas versiones

publicadas. Para evitar errores, solo una pareja de desarrolladores puede integrar su código a la vez.

**Propiedad Colectiva del Código:** En un proyecto XP, todo el equipo puede contribuir con nuevas ideas que apliquen a cualquier parte del proyecto. Asimismo, una pareja de programadores puede cambiar el código que sea necesario para corregir problemas, agregar funciones o re-codificar.

**Ritmo Sostenido:** La Metodología XP indica que debe llevarse un ritmo sostenido de trabajo. El concepto que se desea establecer con esta práctica es planificar el trabajo de forma a mantener un ritmo constante y razonable, sin sobrecargar al equipo.

**Detección y Corrección de Errores:** Cuando se encuentra un error ("Bug"), éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto.

**Pruebas de Aceptación:** Son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El Cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta que pase correctamente todas las pruebas de aceptación. (Joskowicz, 2008).

#### 4. Pruebas

**Pruebas Unitarias,** Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código ("Test-Driven Programmimg"). Que todo código liberado pase correctamente las pruebas unitarias, es lo que habilita que funcione la propiedad colectiva del código.

Para poder establecer un plan de trabajo para el proyecto a través de la **Metodología XP** se desarrolló un cronograma de trabajo el cual se presenta a continuación junto a una descripción del mismo:

## Cronograma

Es una herramienta muy importante en la gestión de proyectos. Puede tratarse de un documento impreso o de una aplicación digital; en cualquier caso, el cronograma incluye una lista de actividades o tareas con las fechas previstas de su comienzo y final.

### d) Procedimientos de instalación y prueba

#### *Requisitos generales*

**Verificación de la compatibilidad:** Se debe comprobar si se cumplen los requisitos para la instalación en cuanto a hardware y software.

**Verificación de la integridad:** Se verifica que el paquete de software es el original, esto se hace para evitar la instalación de programas maliciosos.

**Creación de los directorios requeridos:** Para mantener el orden en el directorio cada sistema puede tener un estándar para la instalación de ciertos archivos en ciertos directorios.

**Creación de los usuarios requeridos:** Para deslindar responsabilidades y tareas se pueden o deben usar diferentes usuarios para diferentes paquetes de software.

**Concesión de los derechos requeridos:** Para ordenar el sistema y limitar daños en caso necesario, se le conceden a los usuarios solo el mínimo necesario de derechos.

**Copia, desempaque y descompresión de los archivos desde el paquete de software:** Para ahorrar Ancho de banda y tiempo en la transmisión por internet o espacio de Disco duro, los paquetes vienen empacados y comprimidos.

- Archivos principales, sean de fuente o binarios.
- Archivos de datos, por ejemplo, datos, imágenes, modelos, documentos XML, etc.
- Documentación
- Archivos de configuración
- Bibliotecas
- Enlaces duros o enlaces simbólicos a otros archivos

**Compilación y enlace con las bibliotecas requeridas:** En algunos casos no se puede evitar el complicado paso de la compilación y enlace que a su vez tiene severos requerimientos de software al sistema. El enlace con bibliotecas requeridas puede ser un problema si en su instalación no se acataron los estándares establecidos.

**Configuración:** Por medio de archivos de configuración se le da a conocer al software con que parámetros debe trabajar. Por ejemplo, los nombres de las personas que pueden usar el software, como verificar su clave de ingreso, la ruta donde se encuentran los archivos con datos o la dirección de nuestro proveedor de correo electrónico. Para sistemas complejos se debe desarrollar el Software Configuration Management.

**Definir las variables de entorno requeridas:** Algunos comportamientos del software solo pueden ser determinados por medio de estas variables. Esto es parte de la configuración, aunque es más dinámica.

**Registro ante el dueño de la marca:** Para el Software comercial a veces el desarrollador de software exige el registro de la instalación si se desea su servicio.

### ***Obtención e instalación***

El equipo de trabajo desarrollador del software para la Miscelánea "El Portillo" entrego de manera exitosa el software con todas las iteraciones cumplidas y en el tiempo acordado por la empresa cliente y la empresa desarrolladora.

Anteriormente al programa se le implementaron una diversidad de pruebas para su correcta ejecución al momento de que la empresa cliente "Miscelánea El Portillo" ejecutara el mismo y comenzara a trabajar con él.

La instalación de este software se llevó a cabo, al día siguiente de la entrega y aprobación total del cliente hacia el software, por lo que solo una parte del equipo de trabajo se encargó de instalar el software en la computadora de la empresa cliente, siguiendo las especificaciones acordadas, y después de la correcta instalación, el cliente junto con su equipo de trabajo observo que el software funcionara de manera correcta y de acuerdo a los requerimientos que se habían determinado desde un principio.

### ***Especificación de prueba y ejecución***

Las especificaciones de las pruebas se establecieron cuando el programa se completó con todas las validaciones y todas las iteraciones estaban terminadas, cada una de las tres iteraciones contiene historias de usuarios, las cuales establecen, que parte del software se va a generar, así como las observaciones que existen en esas partes del programa.

Las pruebas especifican que el software a probar no se debe quebrar al momento de ejecutarlas por lo cual se generaron escenarios de prueba por parte de la empresa desarrolladora para cada historia de usuario, es por eso, que, al término del programa, el

equipo de trabajo, comenzó con la codificación de cada caso de prueba, lo cual le permite revisar a detalle si el software funciona de acuerdo a las especificaciones del cliente o no.

### *Ejecución*

La ejecución de pruebas se llevó a cabo, al término de su codificación por parte del equipo de trabajo, y al momento de implementarlas en el software, determinando en que caso y en que parte se podría encontrar un error en el mismo, para así realizar una revisión y que el programa no se detenga al momento de su instalación en la empresa cliente.

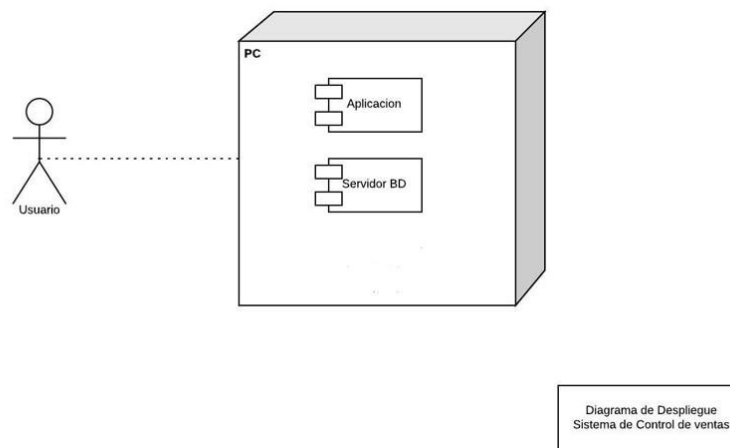
Todo esto se realiza con la finalidad de que el software que la empresa desarrollo, se ejecute de manera exitosa y sin obstrucciones en la empresa cliente.

## **2.- Arquitectura del sistema**

### **a) Descripción jerárquica y de módulos**

A través de los diagramas siguientes, podemos observar la composición de nuestro sistema:

#### ***Diagrama de despliegue***



*Ilustración 2: Diagrama de despliegue*

## Diagrama de módulos o componentes

(Partes que componen el sistema y las relaciones que existen entre ellas)

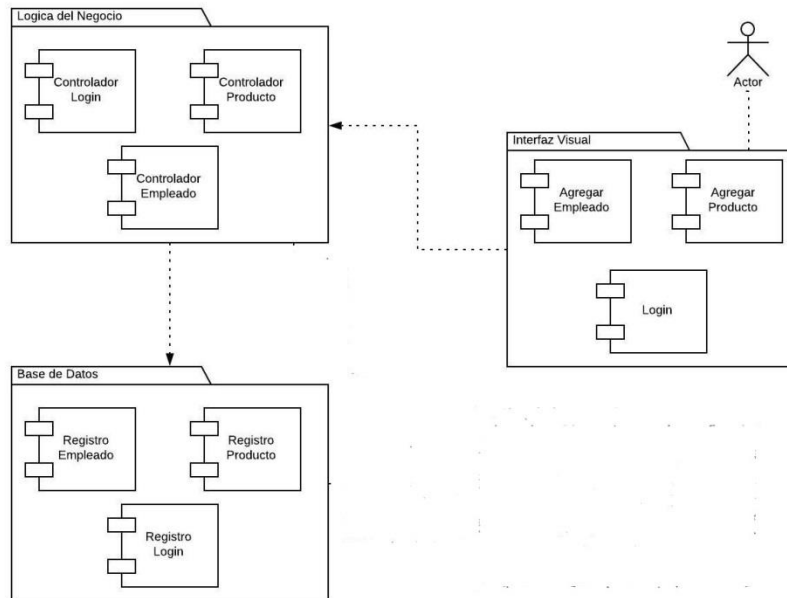


Diagrama de Componentes  
Sistema de Control de Ventas

Ilustración 3: Diagrama de módulos

## Descripción individual de los módulos

*Descripción general y propósito, responsabilidad y restricciones, dependencias e implementación (únicamente una idea general del porque existe en el sistema).*

Nuestro sistema está compuesto por los tres módulos que se muestran en el diagrama anterior y que son descritos a continuación:

### Lógica del negocio

La principal del sistema que es la que se encarga del control de uso de los datos a través de las órdenes del usuario, quien se encarga de solicitar dicha información a la parte del **Modelo** (base de datos) y comunicárselos a las **Vistas** (interfaces) y se compone de:

**-Controlador del login:** Encargado de operar la entrada al sistema, en la cual se pide ingresar usuarios y contraseñas siempre y cuando se cumpla con las condiciones que el sistema requiera.

**-Controlador de productos:** Es aquí donde se tiene el control de todas las funciones que tengan que ver con los productos de la tienda, ya sea sus compras, sus ventas, el inventario, así como el control de ganancias.

**-Controlador de empleados:** La cual se encarga del control de los empleados, ya sea para añadir o eliminar nuevos empleados, modificar sus datos personales o para saber cuales son las operaciones dentro del sistema que el puede llevar a cabo (como consultas o venta de productos).

### **Interfaz de Usuario**

Es la representación visual de los datos de nuestro sistema, donde se conecta la parte del modelo y el controlador, compuesto por las partes siguientes:

**-Login:** Esta interfaz será visible para los dos usuarios principales del sistema, pues es la parte del acceso.

**-Agregar producto:** La interfaz encargada de visualizar los productos que se encuentran en el inventario y los nuevos que van llegando a la tienda.

**-Agregar empleado:** Interfaz donde visualizamos a cada nuevo empleado que se va integrando a la empresa junto a sus datos correspondientes.

### **Base de Datos**

También llamada Modelo es la que se encarga de los datos al consultar la base de datos y en donde se encuentra la parte de actualizaciones, consultas, búsquedas.

La información contenida en este módulo tiene que ver con todo lo relacionado a los **usuarios que acceden al sistema** (que contarán desde su registro al sistema con una contraseña única e intransferible), **sus datos personales** (entre los que destacara la edad que tiene que ser mayor a los dieciocho años para poder tener acceso al mismo) así como el tipo de usuario ya sean **Administradores o Empleados**.

**Los productos que se venden en la tienda** (Tipo, Presentaciones, Cantidades de Adquisición, Precios de Compra, Precios de Venta, Fechas de Caducidad, Códigos de Producto entre otros).



**Los proveedores** que se encargan del abastecimiento de productos para nuestra tienda y de los cuales el sistema ofrece información como: Nombre, Dirección y Teléfono.

## **b) Dependencias externas**

*Si el software utiliza librerías o servicios externos deben listarse junto con una breve descripción de ellas.*

*También vamos a incluir una descripción de las decisiones de diseño asociadas que nos llevaron a elegir la o las tecnologías empleadas en el proyecto.*

*(Incluir Cuadros comparativos de Herramientas CASE, Lenguajes de programación, Gestores de Bases de Datos y si hubiera algún otro).*

Para llegar a una decisión del porque elegir cada una de las herramientas que se utilizaron en el proyecto ya antes descritas realizamos en la clase distintos cuadros comparativos y sinópticos que nos ayudaron a comparar todas las herramientas disponibles. Mas que nada nuestras elecciones se basaron en aspectos como: Facilidad de uso, modo de obtenerlo (gratis o con pago).

- a. Para elegir Herramienta CASE

NOMBRE	DEFINICION	CARACTERISTICAS
Platinum Erwin	<p>Herramienta de Diseño de BD. Brinda productividad en Diseño, generación y mantenimiento de aplicaciones. Desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las Características específicas de la BD diseñada.</p> <p>Establece una conexión entre una base de datos diseñada y una base de datos, permitiendo transferencia entre ambas. Usando esta conexión, genera automáticamente tablas, vistas, índices, reglas de integridad referencial (llaves primarias y foráneas), valores por defecto y restricciones de campos y dominios.</p>	<ul style="list-style-type: none"> <li>• Permite visualizar la estructura, los elementos importantes, y optimizar el Diseño de la BD.</li> <li>• Genera automáticamente las tablas y miles de líneas de stored procedure y triggers para los principales tipos de BD.</li> <li>• Facilidad en el Diseño de una BD.</li> <li>• Automatiza el proceso de Diseño de una manera inteligente. Por ejemplo, habilita la creación de un diccionario de atributos reutilizable, asegurando la consistencia de nombres y definiciones para su BD.</li> <li>• Mantienen las vistas de la BD como componentes integrados al modelo, permitiendo que los cambios en las tablas sean reflejados automáticamente en las vistas definidas.</li> </ul>
JDeveloper	<p>Entorno de desarrollo integrado, desarrollado por Oracle Corporation. Se trata de un entorno que cubre todo el ciclo de vida del desarrollo de Software proporcionando funcionalidad para las fases de Diseño, codificación, depuración y despliegue, mediante una aproximación visual y declarativa.</p>	<ul style="list-style-type: none"> <li>• Es un entorno gratis, aunque previamente se debe suscribir para descargarlo.</li> <li>• Ofrece funcionalidades para el desarrollo en Java, HTML, XML, SQL, PL/SQL, JavaScript, PHP, UML, Oracle ADF y otros.</li> <li>• Funciona en los siguientes OS: Windows, Linux, Mac OSX.</li> <li>• Posee diagramas de clases (UML).</li> </ul>
Oracle Designer	<p>Herramienta de software para analizar los requerimientos de negocios y para diseñar y generar sistemas cliente/servidor que satisfagan tales requerimientos. Ofrece una interfaz intuitiva basada en el explorador, que es capaz de administrar las bases de datos, crear tablas, vistas y otros objetos de base de datos, importar, exportar y visualizar datos de tablas, ejecutar Scripts SQL y generar informes.</p>	<ul style="list-style-type: none"> <li>• Incorpora soporte para el proceso de modelado de negocio, para el análisis de sistemas, para el Diseño de SW y para la generación de sistemas.</li> <li>• Abarca el ciclo de vida del desarrollo de sistemas completo, generando formularios, informes y menus totalmente operativos y libres de error, además del código SQL necesario para crear los objetos de la BD</li> <li>• Posee un repositorio multiusuario y está fuertemente relacionado con Oracle Developer</li> </ul>
Power Designer	<p>Herramienta para el análisis, Diseño inteligente y construcción sólida de una base de datos, un desarrollo orientado a modelos de datos a nivel físico y conceptual que da a los desarrolladores cliente/servidor la mas firme base para aplicaciones de alto rendimiento.</p> <p>Es un juego de herramientas para guardar las definiciones que necesitas el usuario y automatiza la construcción rápida de aplicaciones cliente/servidor flexibles y graficas.</p>	<ul style="list-style-type: none"> <li>• Permite a las empresas, de manera más fácil, visualizar, analizar y manipular metadatos, logrando una efectiva arquitectura empresarial de información.</li> <li>• Brinda un enfoque basado en modelos, el cual permite alinear al negocio con la tecnología de información, facilitando la implementación de arquitecturas efectivas.</li> <li>• Mejora la productividad en grupo</li> <li>• Es altamente personalizable</li> <li>• Alinea el negocio con el area de tecnología</li> <li>• Brinda facilidad de uso grafica.</li> </ul>

Ilustración 4: Cuadro Comparativo Herramientas CASE

## b. Para elegir Sistema Gestor de Base de Datos

SGBD	Descripción General	Plataforma	Lenguaje de Programación	Herramienta CASE	Memoria Almacenamiento	o Transacciones	Ventajas	Licencia	Versión
<b>Oracle</b>	Oracle es un sistema de gestión de base de datos objeto-relacional.	Microsoft Windows, Linux, Unix.	PL/SQL, PHP, Java, .NET, XML	Oracle Designer	Almacenamiento: 11 GB Memoria: 1 GB	x	-Puede ejecutarse en todas las plataformas, desde una PC hasta un supercomputador.  -El software del servidor puede ejecutarse en multitud de sistemas operativos.	Libre-Gratis.  Es la única versión gratuita.	Oracle Database 11g Express Edition.
<b>SQL Server</b>	Es un sistema para la gestión de base de datos producido por Microsoft basado en el modelo relacional.	Microsoft Windows.	T-SQL	Management Studio.	Almacenamiento: 2 GB	x	-Proporciona agilidad sus operaciones de análisis y administración de datos.  -Es útil para manejar y obtener datos de la red de redes	Libre-Gratis.	SQL Express Edition.
<b>MySQL</b>	Es un sistema de gestión de base de datos relacional, multiusuario, desarrolla MySQL como	Microsoft Windows, Linux, Unix.	C, C++, Pascal, PHP ...	MySQL Workbench.	Almacenamiento: 200 MB Memoria: 512 MB	x	-Es la base de código fuente más usada.  -Velocidad al realizar las operaciones, lo que le hace uno	Libre-Gratis.	MySQL 5.5.30

Ilustración 5: Cuadro comparativo de SGBD

software libre.							de los gestores con mayor rendimiento		
<b>PostgreSQL</b>	No es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada.	Microsoft Windows, Mac OS, Linux, Unix.	PL/ Pascal, C, C++, Java, PL/Java Web ...	Power Designer.	Almacenamiento: 1.5 GB Memoria: 1 GB	x	-Es una base de datos orientada a objetos.  -Instalación ilimitada y gratuita  -Estabilidad y confiabilidad	Libre-Gratis.	PostgreSQL 1.14.1
<b>Informix</b>	El DBMS fue concebido y diseñado por Roger Sijpola finales de los años 1970.	Multiplataforma	C++	ISQL, ISQL Forms, Reports	Almacenamiento: 8 GB Memoria: 1 GB	x	-Proporciona funciones de desarrollo rápido y depuración interactiva.  -Incluye un sistema de administración de bases de datos relacionales basado en SQL.  -Herramientas para la inclusión de SQL en programas de aplicación y un lenguaje de cuarta generación.	Libre-Gratis	Informix Developer Edition.

Ilustración 6: Cuadro comparativo de SGBD

<b>Freebird</b>	El sistema de administración de base de datos relacional de código abierto <b>Freebird</b> cuenta con un rendimiento excelente y se escala de manera impresionante	Multiplataforma	C, C++	<b>Freebird Database Designer</b>	Almacenamiento: 1 GB Memoria: 1 GB	x	-Nueva arquitectura <b>SuperClassic</b> , Auditoria, Consultas entre base de datos.  -No consume grandes recursos en el servidor.  -Se pueden usar librerías externas que mejoran la operativa de <b>Freebird</b> .	Libre-Gratis	<b>Freebird 2.5</b>
-----------------	--	-----------------	--------	-----------------------------------	---	---	---	--------------	---------------------

*Ilustración 7: Cuadro comparativo de SGBD*

### 3.- Diseño del modelo de datos

Entidades involucradas en el sistema (Incluir el diagrama de clases, entidad-relación, relacional, etc).

--- Un programa, aplicación o librería puede a su vez trabajar con varios tipos de datos ya sean de entrada, internos o de salida (distinguir cada uno de ellos y describir su modelo).

Se identificaron las siguientes entidades que formaran parte del sistema, a través de ellas se construyeron los diagramas correspondientes para el diseño de modelo de datos:

#### Diagrama Entidad-Relación

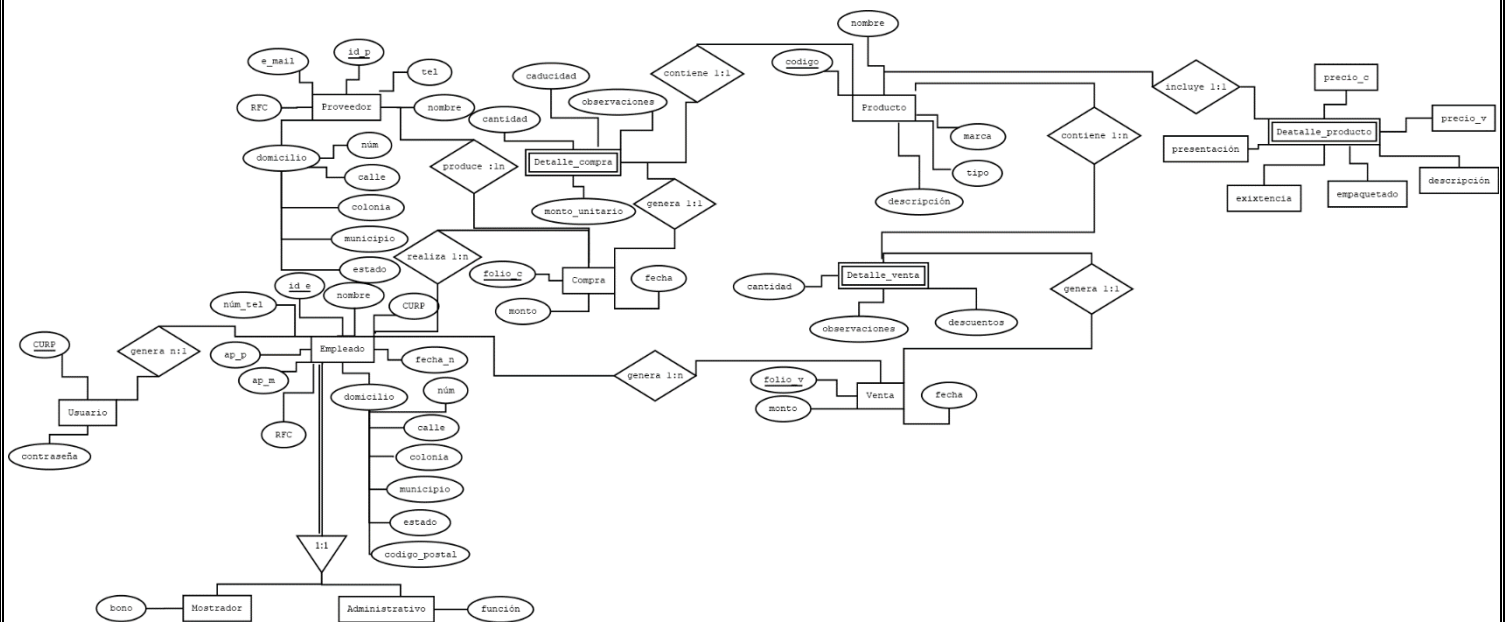


Ilustración 8: Diagrama Entidad-Relación

Las entidades principales para el sistema las identificamos a través de las entrevistas realizadas al Administrador y Empleados de la tienda para conocer las necesidades que se buscan cubrir con el presente sistema.

## Diagrama Relacional

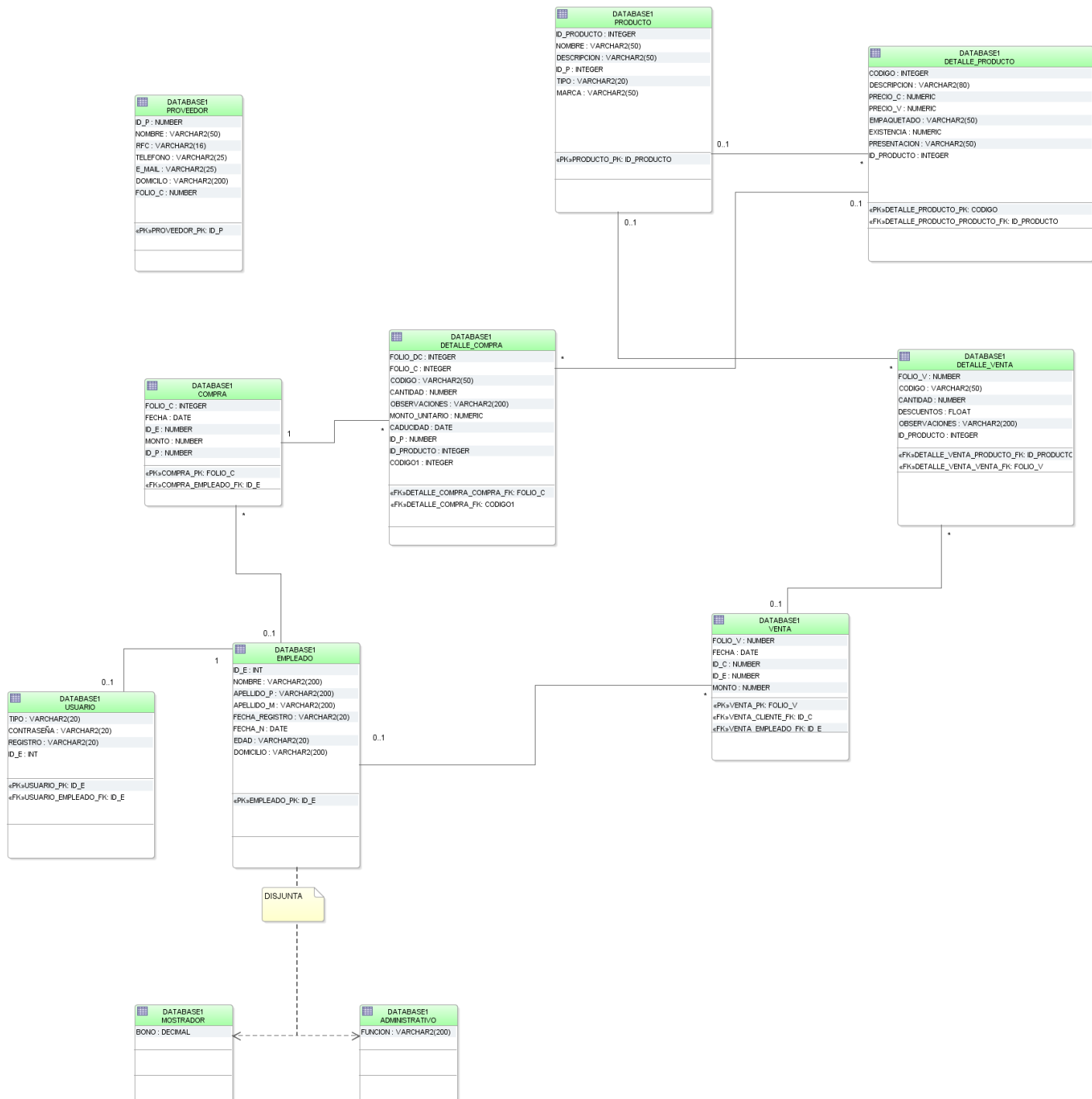


Ilustración 9: Diagrama Relacional

## 4.- Descripción de procesos y servicios ofrecidos por el sistema

Mencionar cuales son los servicios o tareas que el sistema ofrece/implementa y describir los procesos que realizan, para entender cómo funcionan y como se puede invocar/utilizar.

(Incluir diagramas de flujo para cada iteración, BPMN o cualquier otro)

1 Describir brevemente que hace o cuál es su propósito

2 Describir los datos de entrada o salida

A través de las tres iteraciones establecidas pudimos ubicar diez historias de usuario que abarcan las necesidades presentadas por la empresa. A continuación, se explica de manera concreta las tareas o funciones que ofrece nuestro sistema a través de Diagramas BPMN y Diagramas de Flujo que ayudan a visualizar de mejor manera dicha explicación.

### Primera Iteración

En ella haremos **la gestión de los productos** (agregación, eliminación y búsqueda), el **ingreso al sistema a través del login** y **la gestión de los empleados** (agregación, eliminación, busque y creación de cuentas).

-Para el registro de productos

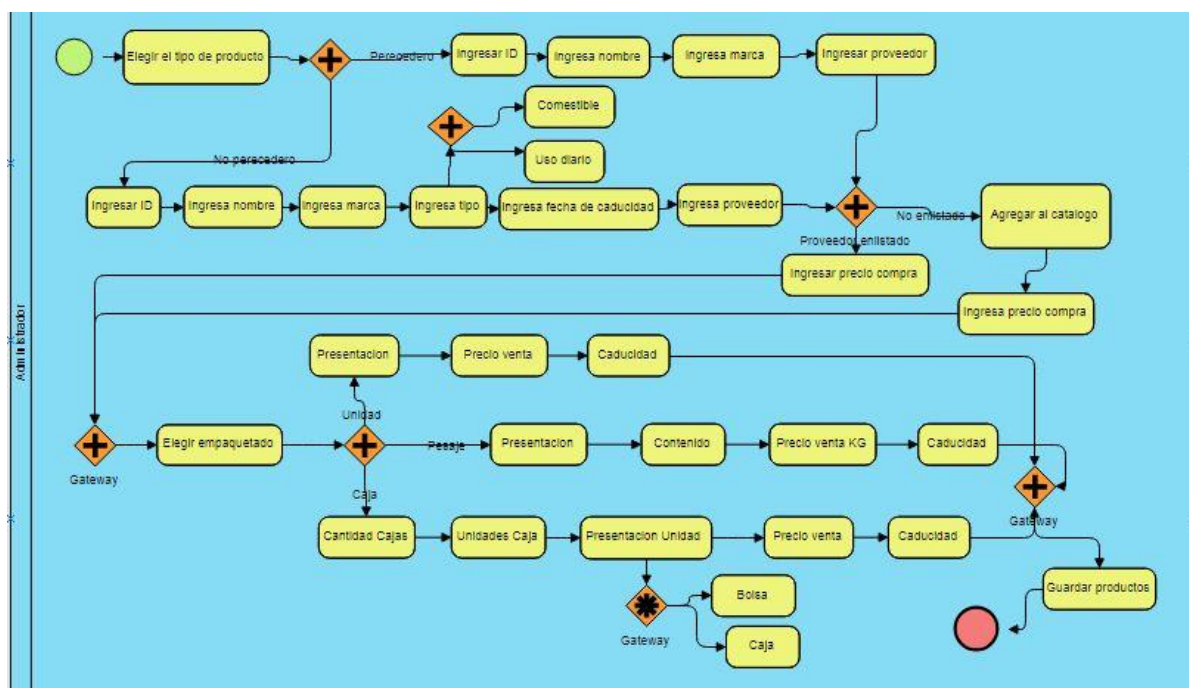


Ilustración 10: BPMN de registro de productos

Una vez dentro del sistema el administrador ingresa a la opción de registro de producto donde podrá ingresar nuevos productos que se podrán comprar para la tienda, teniendo diversas opciones de datos de entrada:

### **Producto perecedero**

1. Elige el tipo de producto en este caso, perecedero
2. Asignarle un ID que será único para cada producto.
3. Ingresar el nombre del producto.
4. Ingresar su marca.
5. Ingresar el proveedor que nos surte dicho producto.
6. Verificar que dicho proveedor este enlistado ya en nuestro catálogo, si es así continuar con los siguientes datos y en caso contrario agregarlo.
7. Ingresar el precio con el que estamos adquiriendo el producto.
8. Elegir el tipo de empaquetado con el que ingresa a nuestra tienda.

### **Producto no perecedero**

1. Elige el tipo de producto en este caso, no perecedero.
2. Asignarle un ID que será único para cada producto.
3. Ingresar el nombre del producto.
4. Ingresar su marca.
5. Ingresar el tipo (alimento, bebida o algún otro).
6. Ingresar su fecha de caducidad
7. Ingresar el proveedor que nos surte dicho producto.
8. Verificar que dicho proveedor este enlistado ya en nuestro catálogo, si es así continuar con los siguientes datos y en caso contrario agregarlo.
9. Elegir el tipo de empaquetado con el que ingresa a nuestra tienda.

### **Para elegir tipo de empaquetado**

#### **En caja**

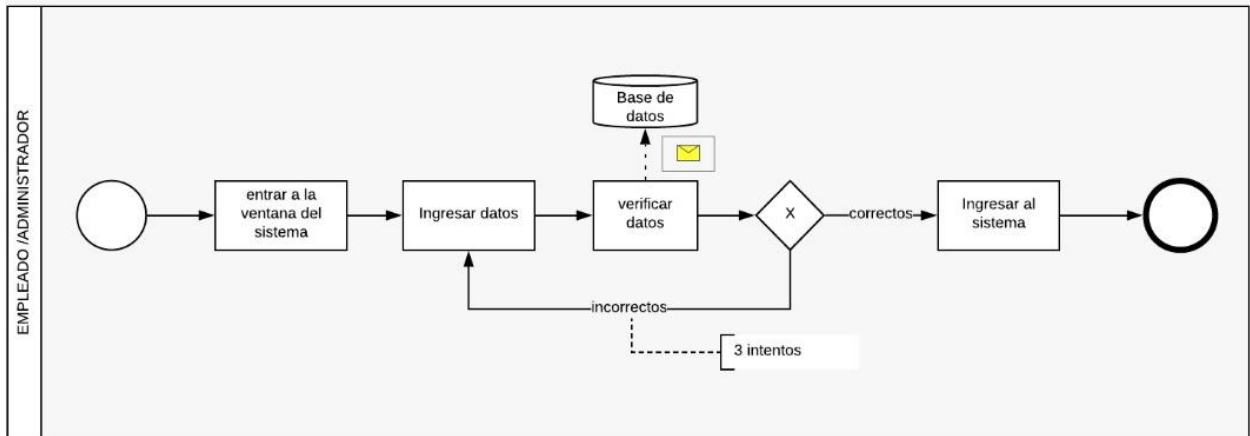
1. Ingresar la cantidad de las cajas compradas.
2. Ingresar las unidades de caja.
3. La presentación en la que viene el producto (bolsas o cajas).
4. Ingresar el precio de venta.
5. Ingresar su caducidad.
6. Guardar los datos del producto.



### **Por unidad**

1. Ingresar su presentación.
2. Ingresar el precio de venta.
3. Ingresar la caducidad del producto
4. Guardar los datos del producto.

*Para el ingreso al sistema (Login)*



*Ilustración 11: BPMN de ingreso al sistema*

La primera interfaz que se mostrará a los usuarios será la del ingreso al sistema donde los dos tipos de usuarios tienen que pasar dado que es la puerta principal de acceso, y cuenta además con los siguientes pasos y datos de entrada para dicha acción:

1. Entrar a la ventana de acceso al sistema.
2. Ingresar los datos que se piden (usuario y contraseña), si fuese la primera vez que se ingresa de debe llenar el formulario de registro correspondiente.
3. Dar la opción de ingresar para que la base de datos pueda realizar la verificación de estos.
4. Si son correctos se ingresará satisfactoriamente a la ventana principal.
5. En caso contrario se deberán ingresar los datos nuevamente, se tendrán tres nuevos intentos para esto.

### Para el registro de un nuevo empleado al sistema

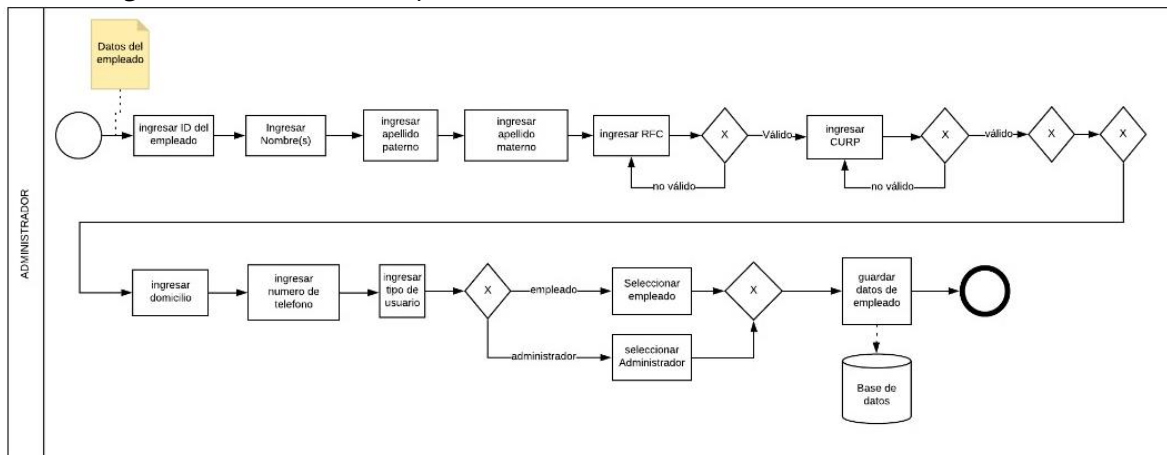


Ilustración 12: BPMN de registro de nuevo empleado

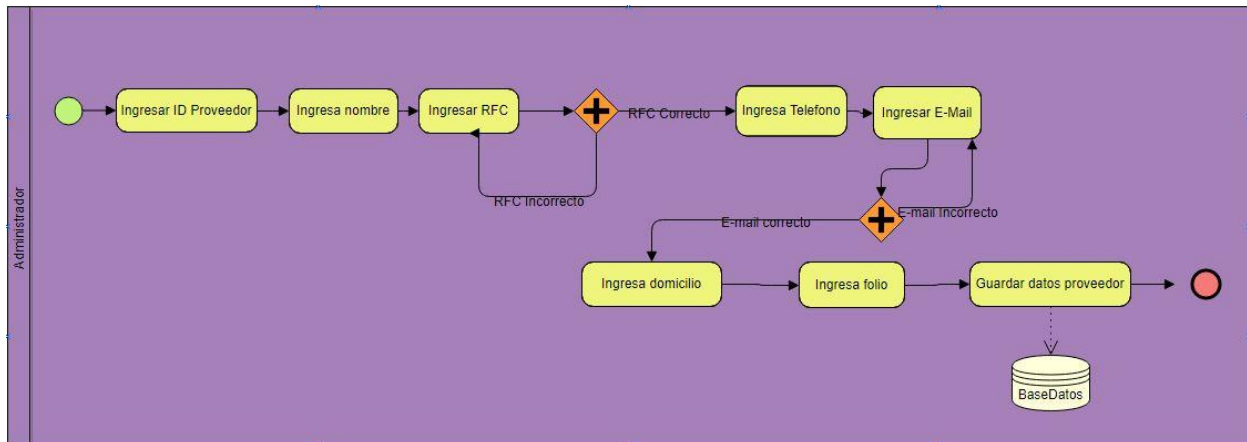
Para los usuarios que se registraran por primera vez al sistema se tendrán como base los siguientes puntos que requieren diversos datos de entrada para el sistema:

1. Ingresa ID del empleado.
2. Ingresar el nombre.
3. Ingresar apellido paterno.
4. Ingresar apellido materno.
5. Ingresar el RFC del empleado y a continuación en este campo se validará que dicho dato sea correcto y en caso de no ser así deberá ingresar nuevamente este dato.
6. Ingresar la CURP del empleado y después de rellenar dicho campo se repite la acción del punto numero 5 (validar el dato).
7. Ingresar el domicilio.

## Segunda Iteración

En esta segunda iteración se trabaja la **gestión de los proveedores** la cual se divide en la agregación, eliminación y búsqueda de clientes y **gestiones de compra** que incluyen la realización, cancelamiento, estatus y mal venta de productos.

### *Para el registro de proveedores*



*Ilustración 13: BPMN del registro de proveedores*

Cuando el administrador necesite registrar un nuevo proveedor para su compra de productos, necesitara proporcionar al sistema los siguientes datos:

1. Un ID para el proveedor
2. Su nombre completo
3. El RFC del proveedor (en caso de ser correcto podrá seguir ingresando los datos y contrariamente si no es aceptado por el sistema tendrá que realizarse este paso nuevamente).
4. El teléfono del proveedor
5. Su correo electrónico (en caso de ser correcto podrá seguir ingresando los datos y contrariamente si no es aceptado por el sistema tendrá que realizarse este paso nuevamente).
6. Ingresar su domicilio
7. Generar su folio (dato de salida).
8. Confirmar que los datos sean guardados en la base de datos.

### *Para buscar y editar datos de un proveedor*

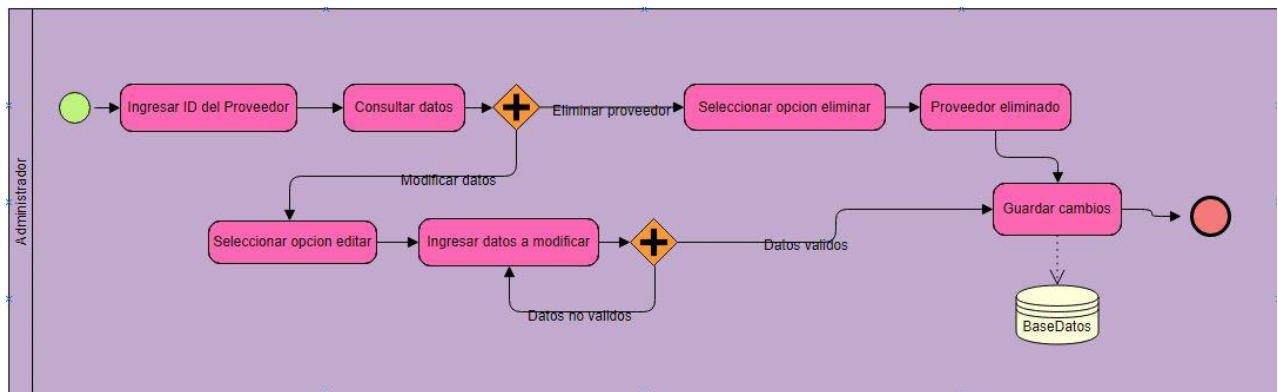


Ilustración 14: BPMN de búsqueda y edición de proveedores

El proveedor puede acceder al sistema para buscar y editar información de sus proveedores en caso necesario, así pues, para esto necesita ingresar datos y confirmar las siguientes opciones:

1. Ingresar en el buscador el id del proveedor que desea consultar (en caso de ser correcto podrá seguir adelante y contrariamente si no es aceptado por el sistema tendrá que realizarse este paso nuevamente)
2. Se mostrarán los datos del proveedor
3. En caso de querer eliminar la información, seleccionar la opción "Eliminar proveedor" y a continuación se mostrará un mensaje donde se confirma la eliminación.
4. Para modificar los datos se selecciona primeramente la opción de "Editar".
5. Modificamos los datos deseados (en caso de ser correctos podrá seguir adelante y contrariamente si no es aceptado por el sistema tendrá que realizarse este paso nuevamente)
6. Se muestra la confirmación de que dicho proveedor ha sido eliminado y el sistema guarda los cambios en la base de datos.

### Para la gestión de compra

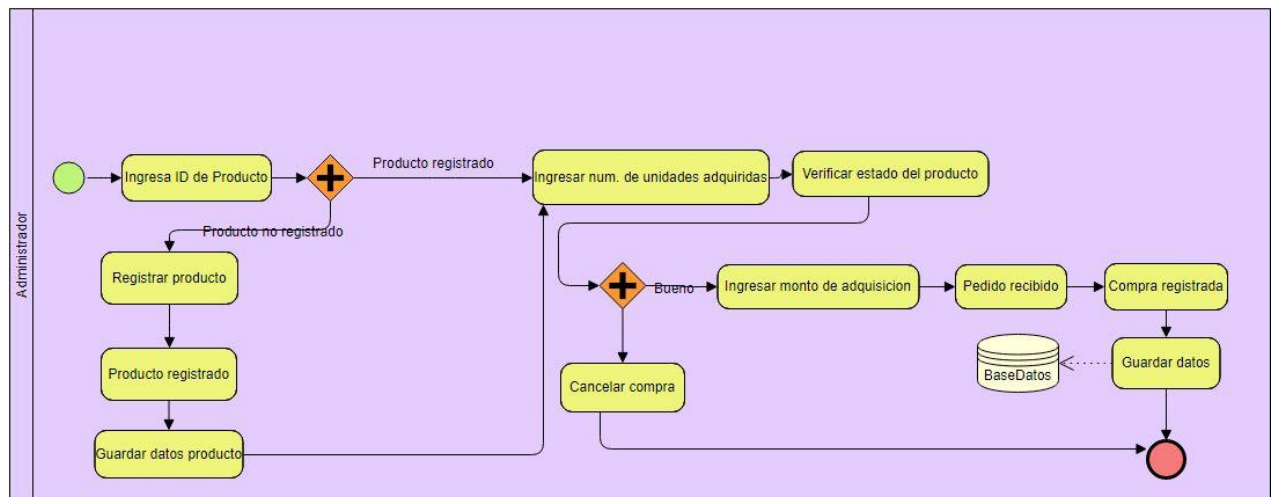


Ilustración 15: BPMN para la gestión de compras

Para registrar la compra de un producto ya existente en el sistema o uno que está agregándose en el momento, así como también para poder eliminarlo en su proceso debido a su mal estado el administrador debe seguir con los pasos siguientes:

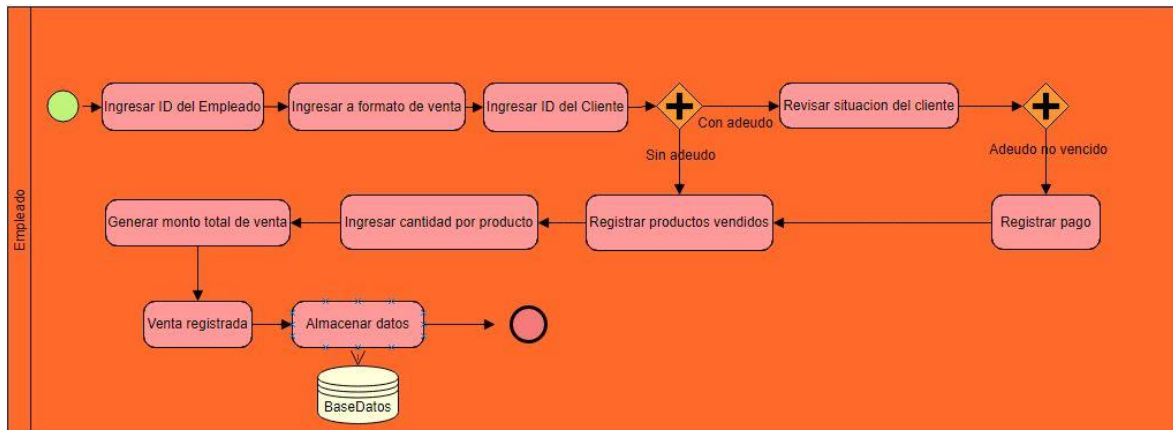
1. Ingresar el ID del producto a encontrar
2. Si es uno no registrado aún se podrá dar de alta en el sistema al momento.
3. Si ya se encuentra dado de alta se ingresará a continuación el número de unidades que se están adquiriendo de dicho producto.
4. Es aquí donde el administrador puede verificar el estado del producto.
5. Si el estado del producto no es el adecuado se procede a seleccionar el cancelar la compra y terminar la operación.
6. Si el producto está en condiciones de poner a la venta continuamos con los siguientes requerimientos como ingresar el monto por el cual estamos adquiriendo el producto.
7. Confirmamos que el pedido ha sido recibido.
8. Registramos la compra.
9. Se guardan los datos en la base de datos y terminamos la operación.

### Tercera Iteración

La más importante para el sistema la cual se encarga de la **gestión de ventas** (donde se registran dichas ventas y se dan a conocer los cortes de caja), **la seguridad de la venta** (donde se puede eliminar una de ellas dado un mal registro o se puede eliminar un producto que este a la venta) cuyas acciones o tareas estarán a cargo ya sea por el

administrador únicamente o compartida con el empleado y **las consultas de ganancias** (en donde se podrá tener acceso a los montos de compra de productos así como los montos por la venta de los mismos).

*Para la realización de las ventas*



*Ilustración 16: BPMN del registro de ventas*

El empleado se ocupa en realizar las ventas a los clientes de nuestra tienda, para realizar esta actividad debe ingresar datos y realizar las siguientes acciones:

1. Ingresar su ID (en caso de ser correcto podrá seguir ingresando los datos y contrariamente si no es aceptado por el sistema tendrá que realizarse este paso nuevamente).
2. Seleccionar nuevo formato de venta
3. Ingresar el ID del cliente (en caso de ser uno nuevo se genera en ese momento).
4. Si es un cliente ya registrado se verificará si tiene adeudos (se revisa su historial y en caso de tener adeudos no vencidos el cliente puede saldarlos en ese momento para que así pueda realizar una nueva compra) y en caso contrario de no contar con ellos se pasa directamente a recabar los siguientes datos.
5. Registramos los productos que estamos vendiendo a nuestro cliente.
6. Ingresamos la cantidad de cada producto que se está vendiendo.
7. El sistema genera el monto total de venta que se cobrara al cliente.
8. El sistema genera un mensaje de confirmación de registro y esta información de almacena en la base de datos concluyendo la operación.

### De los cortes de caja

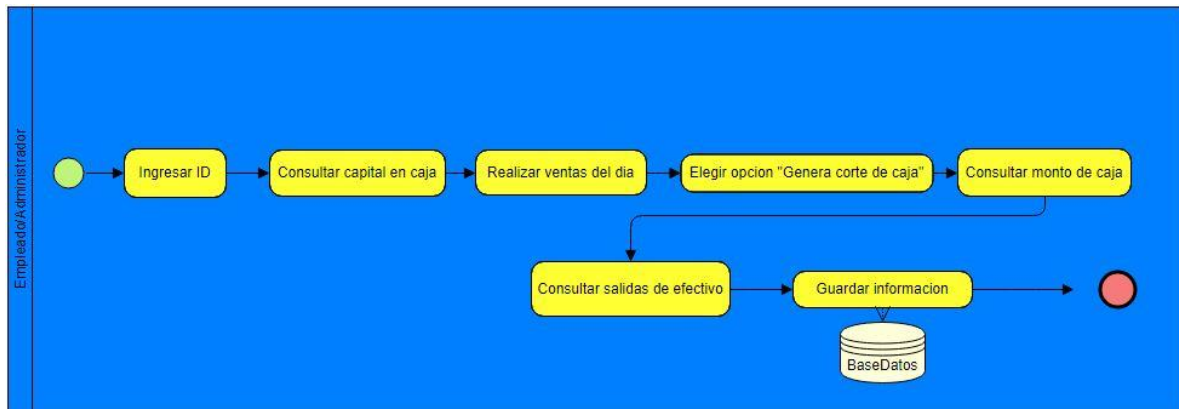


Ilustración 17: BPMN de los cortes de caja

El administrador de la tienda o los empleados pueden generar un corte de caja diario cumpliendo con los siguientes requisitos:

1. Ingresar el ID (siendo administrador o empleado).
2. Seleccionar la opción "Consultar capital de caja".
3. Realizar las ventas del día.
4. Elegir la opción "Generar corte de caja".
5. Consultar el monto de caja
6. Consultar las salidas de efectivo
7. El sistema confirma y guarda la información en la base de datos.

### De la seguridad de ventas

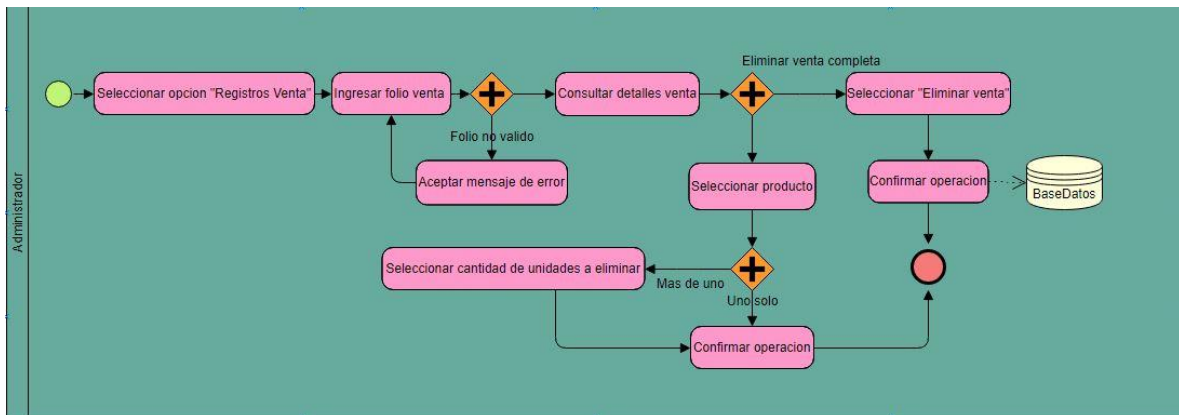
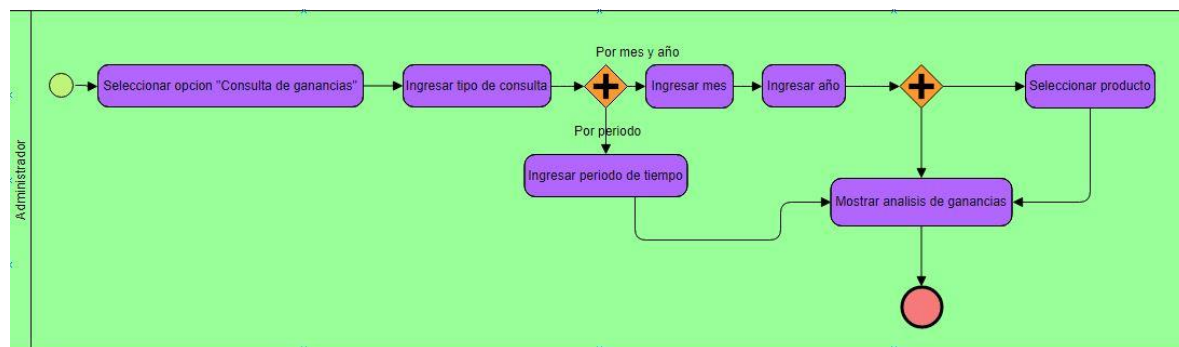


Ilustración 18: BPMN para seguridad de ventas

En este apartado únicamente el administrador puede tener acceso a la eliminación de productos en una venta o de una en su totalidad para mantener la integración y seguridad de los datos que se manejan, cumpliendo con los requerimientos siguientes:

1. Seleccionar la opción de "Registros de Venta"
2. Ingresar el folio de la venta que se está buscando (en caso de ser correcto podrá seguir ingresando los datos y contrariamente si no es aceptado por el sistema tendrá que realizarse este paso nuevamente).
3. Si el folio es válido podemos visualizar la venta encontrada.
4. Si se decide eliminar la venta entera, se da clic en la opción "Eliminar venta" y confirmamos dicha operación (los cambios se verán reflejados en la base de datos).
5. Si se desea eliminar únicamente un producto de la lista lo seleccionamos (en caso de ser más de una unidad marcamos el número total) y confirmamos su eliminación.

#### *Para la consulta de ganancias*



*Ilustración 19: BPMN para consulta de ganancias*

Para consultar la rentabilidad y las ganancias de la empresa únicamente el administrador puede tener acceso a la eliminación de productos en una venta o de una en su totalidad para mantener la integración y seguridad de los datos que se manejan, requiriendo para ello seguir los pasos siguientes:

1. Seleccionar la opción de "Consulta de ganancias"
2. Ingresar el tipo de consulta que se desea hacer.
3. Si la consulta es por mes y año se ingresan estos datos en el sistema.
4. Si la consulta es por un periodo de tiempo largo se ingresa este dato en el sistema
5. Si queremos un análisis en general se generan el análisis de ganancias.
6. Si el análisis lo queremos por producto especificamos este en el sistema y se genera el análisis de ganancias.



## 5.- Documentación Técnica – Especificación API

*Utilizar una herramienta de generación de documentación automática.*

## 6.- Conclusiones

### a) Complicaciones que se encontraron

Al inicio del proyecto tuvimos algunas dudas respecto a la metodología que utilizaría para el desarrollo de este, después de un análisis profundo elegimos la Metodología XP puesto que al investigar las fases y los procesos que seguía nos llamó la atención la forma de trabajo que dicha metodología maneja y se llegó a la conclusión de que nos adaptaríamos rápidamente al modo de esta.

En la primera fase que es la **planificación** del proyecto tuvimos que planear un primer encuentro con los clientes para conocer las necesidades básicas que necesitarían ser resueltas a través del sistema para poder partir de dicha entrevista a la creación de las historias de usuario, donde la parte más difícil fue decidir que tareas eran más importantes, decidir el grado de complicación que tendrían, su importancia en el sistema y en que iteración tendría que ser resuelta (ya que a nuestro criterio ninguna tendría más importancia que otra).

En la segunda fase de **diseño** una complicación en general fue decidir el diseño que tendrían las interfaces que conformarían el sistema, como se plasmarían las funciones (en cada historia de usuario e iteración) tanto de botones, menús, como de otras tareas.

En la tercera fase (las más importante y tediosa de la metodología) de **codificación** un factor clave fueron los constantes cambios que los clientes dieron al resultado de las interfaces, la modificación de entidades y características en la base de datos que fueron de manera constante, las diversas opiniones de los miembros del equipo en cuanto al desarrollo de estas así como de otras correcciones hechas por nuestra asesora de proyecto que tal vez nos dieron lugar a constantes tiempos de trabajo en exceso.

La cuarta y última fase que es la de **pruebas** fue la clave de comprobación de éxito para el proyecto ya que es aquí donde se desarrollaron diversos tipos de estas como:

Pruebas internas del sistema: Donde se comprobó el perfecto funcionamiento de cada una de las interfaces (que se cumplieran con los criterios establecidos al inicio de este).

Pruebas de integración: Al unir el sistema en uno solo las funciones del proyecto debían seguir su curso normal, sin sufrir cambios ni alteraciones en el manejo de los datos.

### **b) ¿Cuál fue la experiencia que obtuvimos al realizar este proyecto?**

El proyecto que se desarrolló nos hizo comprender muchas de las preguntas que teníamos sobre cómo se realizaban sistemas de este tipo, y porque dan a conocer la esencia del mundo ágil, así como su impacto en las empresas evitando situaciones de riesgo en los proyectos gracias a su adaptación a los cambios.

Es importante tener en cuenta cómo la elección de una mala metodología de gestión de proyecto puede provocar un impacto negativo en el proyecto y a su vez incomodidad en el cliente que ha solicitado el proyecto, ya que la mejor forma que el cliente vea un avance del producto es a través de los entregables que ofrecen las metodologías ágiles.

Por otro lado, evidenciamos la importancia en la definición de los requerimientos, pues de ellos depende la conformidad y satisfacción de los usuarios del sistema y además de cómo el trabajo en equipo debe ser productivo, comprensivo y con tolerancia (promoviendo la buena convivencia del grupo de trabajo).

## **7.- Bibliografía o referencias**

- Universidad mayor de San Marcos: Desarrollo e implementación de un sistema web para generar valor en una pyme aplicando una metodología ágil. Caso de estudio: Manufibras Perez SRL  
[http://cybertesis.unmsm.edu.pe/bitstream/handle/cybertesis/4668/Castillo\\_ap.pdf?sequence=1&isAllowed=y](http://cybertesis.unmsm.edu.pe/bitstream/handle/cybertesis/4668/Castillo_ap.pdf?sequence=1&isAllowed=y)
- Núñez Mori, J. G. (19 de Noviembre de 2010). usabilidad en metodologías ágiles (Tesis de Master, Universidad Politécnica de Madrid - Facultad de Informática - España). Recuperado el 17 de Junio de 2013, de Repositorio de la Universidad Politécnica de Madrid:  
[http://www.fi.upm.es/catedraibmrational/sites/www.fi.upm.es.catedraibmrational/files/Tesis\\_MarcoAgilTrabajo.pdf](http://www.fi.upm.es/catedraibmrational/sites/www.fi.upm.es.catedraibmrational/files/Tesis_MarcoAgilTrabajo.pdf)