**1.- Definiciones y especificación de requerimientos**

1. **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.

1. **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.

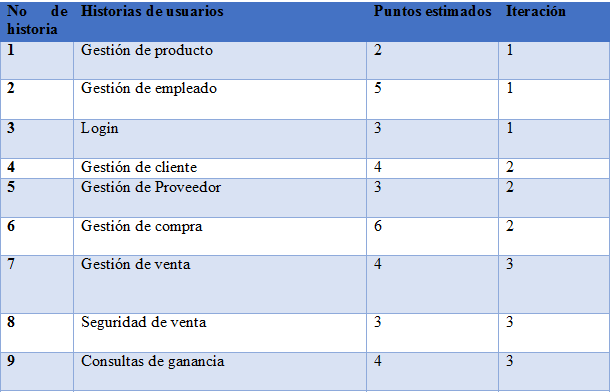
* Programas: Conjuntos de instrucciones que proporcionan la funcionalidad deseada cuando son ejecutadas por el ordenador. Están escritos usando lenguajes específicos que los ordenadores pueden leer y ejecutar, tales como lenguaje ensamblador, Basic, Java, Ruby, PHP, Python…. Los programas también pueden 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:



***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 él 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.

1. **Procedimientos de desarrollo**

***Herramientas***

Las herramientas que se utilizaran 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.

1. 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.

1. 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).

1. 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 Programmming”). 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.

1. **Procedimientos de instalación y prueba**

***Requisitos no funcionales***

***Obtención e instalación***

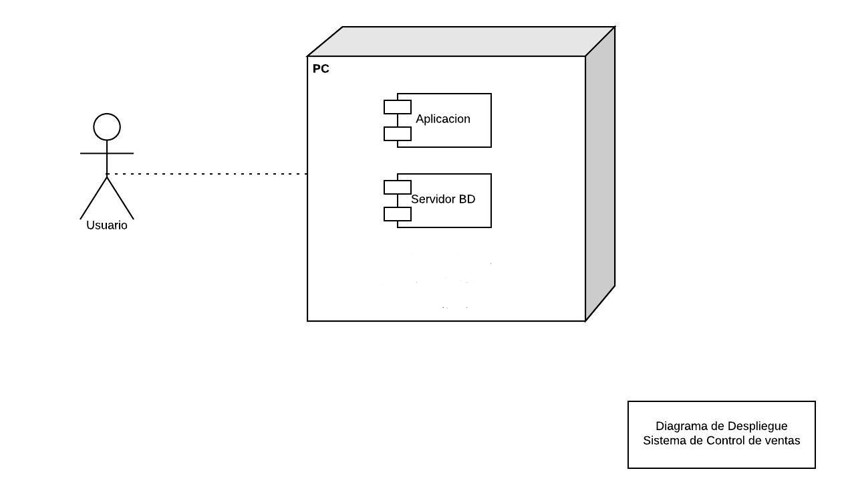
***Especificación de prueba y ejecución***

**2.- Arquitectura del sistema**

**a) Descripción jerárquica y de módulos**

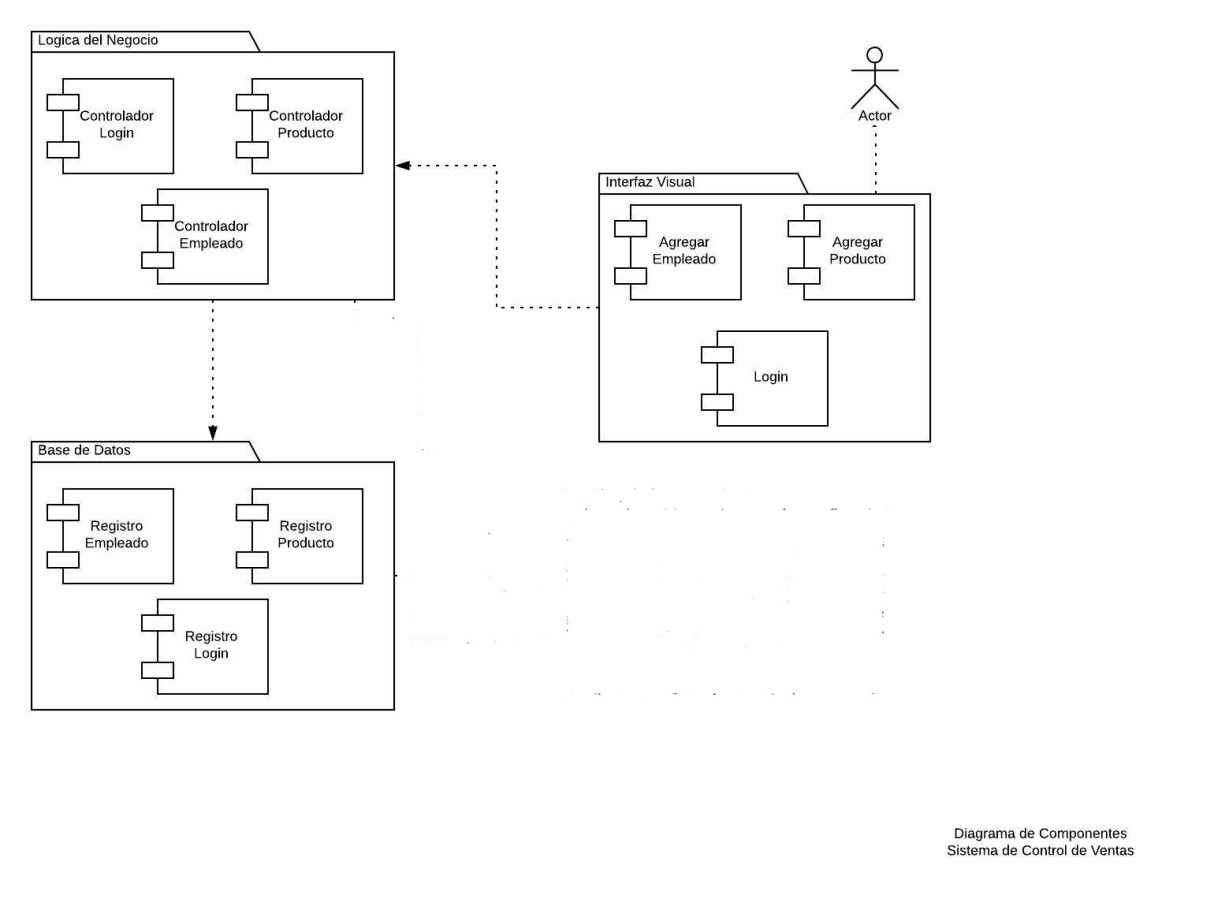
A través de los diagramas siguientes, podemos observar la composición de nuestro sistema:

***Diagrama de despliegue***



***Diagrama de módulos o componentes***

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



***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 modulo tiene que ver con todo lo relacionado a los **usuarios que acceden al sistema** (que contaran 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.

***Dependencias externas***

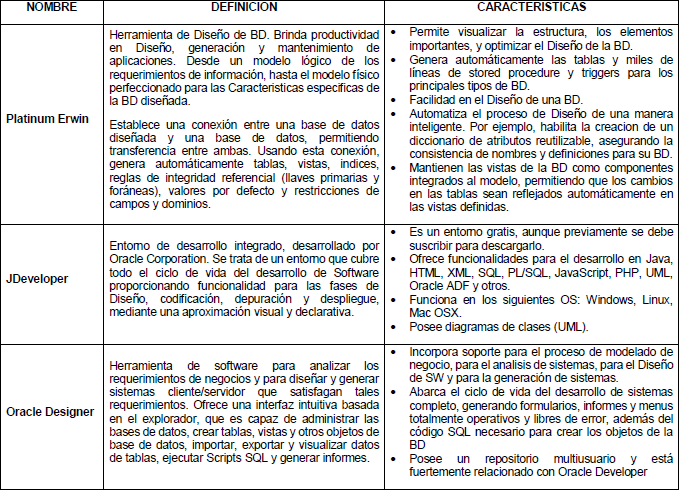
*Si el software utiliza librerías o servicios externos deben listarse junto con una breve descripción de ellas.*

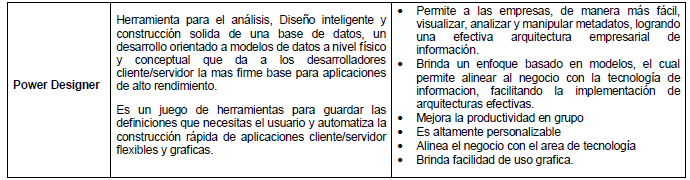
*También vamos a incluir una descripción de las decisiones de diseño asociadas que nos llevaron a elegir la o las tecnologías empleadas en el proyecto.*

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

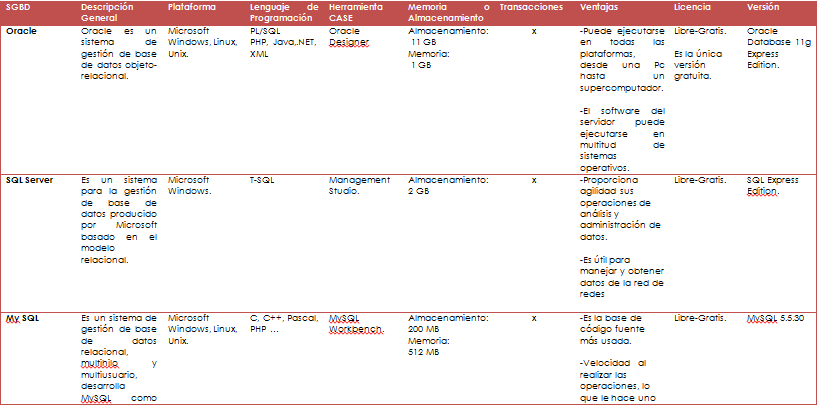
Para llegar a una decisión del porque elegir cada una de las herramientas que se utilizaron en el proyecto ya antes descritas realizamos en la clase distintos cuadros comparativos y sinópticos que nos ayudaron a comparar todas las herramientas disponibles. Mas que nada nuestras elecciones se basaron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

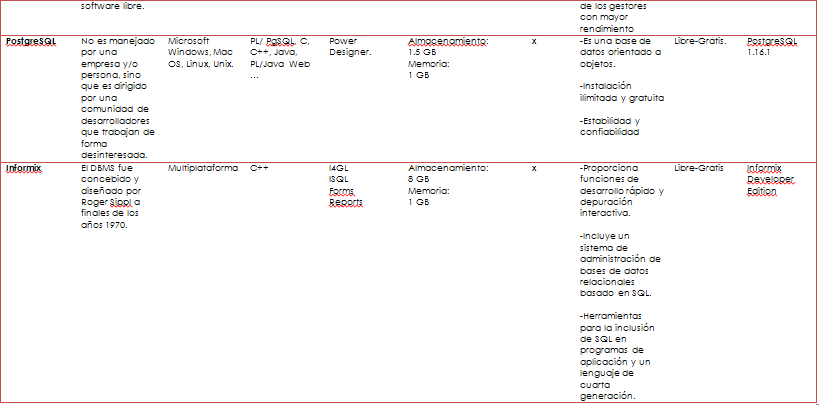
1. Para elegir Herramienta CASE

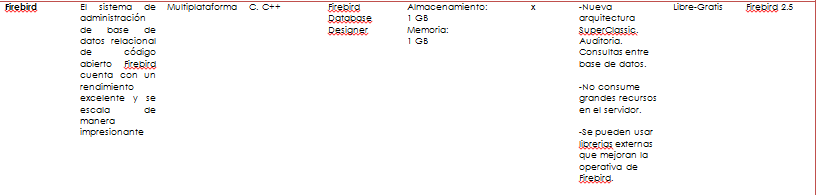




1. Para elegir Sistema Gestor de Base de Datos







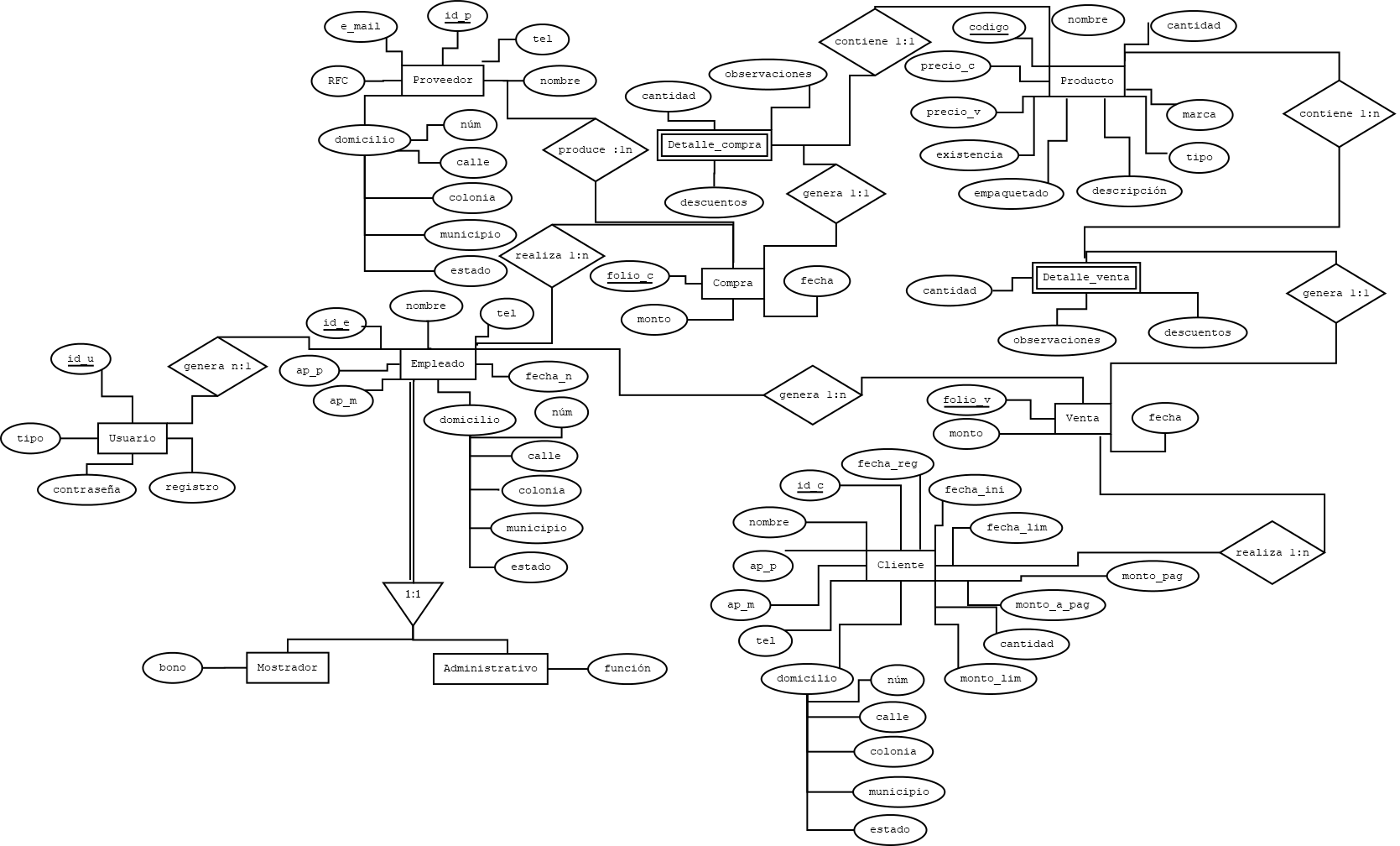
**3.- Diseño del modelo de datos**

*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).*

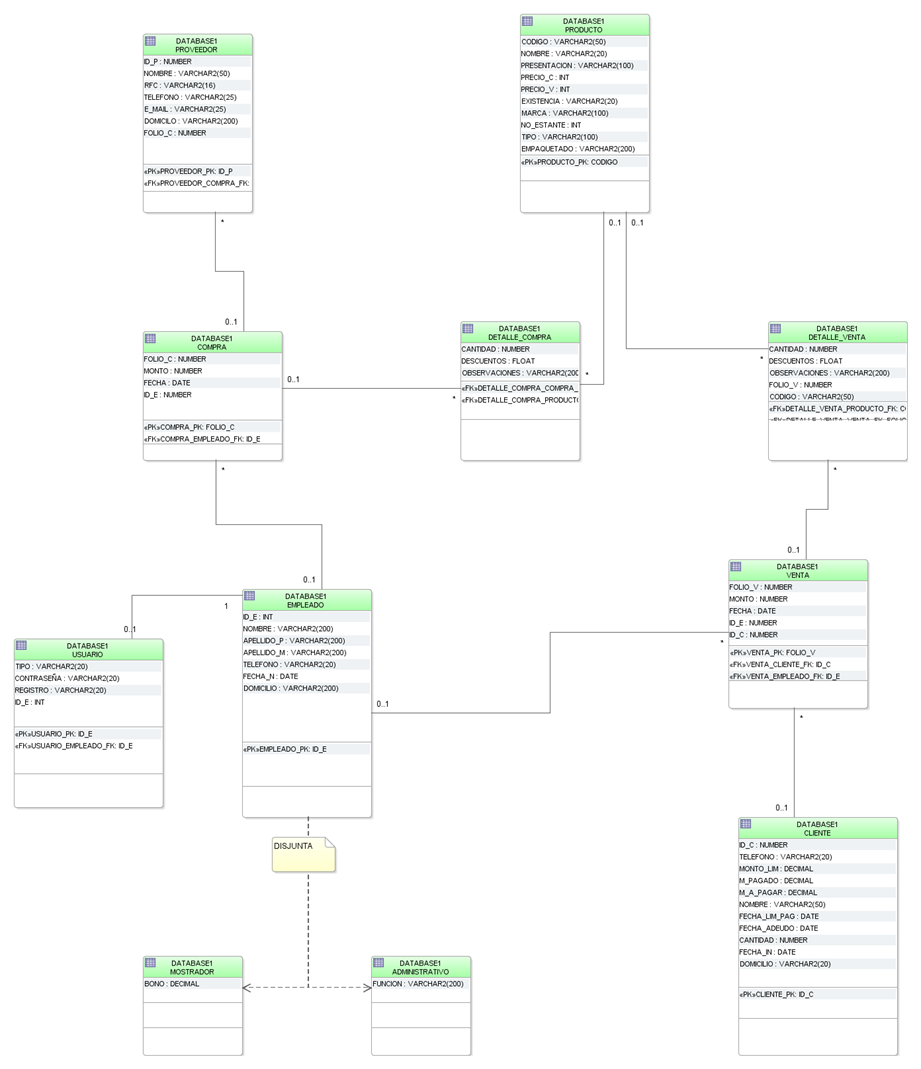
Se identificaron las siguientes entidades que formaran parte del sistema, a través de ellas se construyeron los diagramas correspondientes para el diseño de modelo de datos:

***Diagrama Entidad-Relación***



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**

****

**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 como 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*

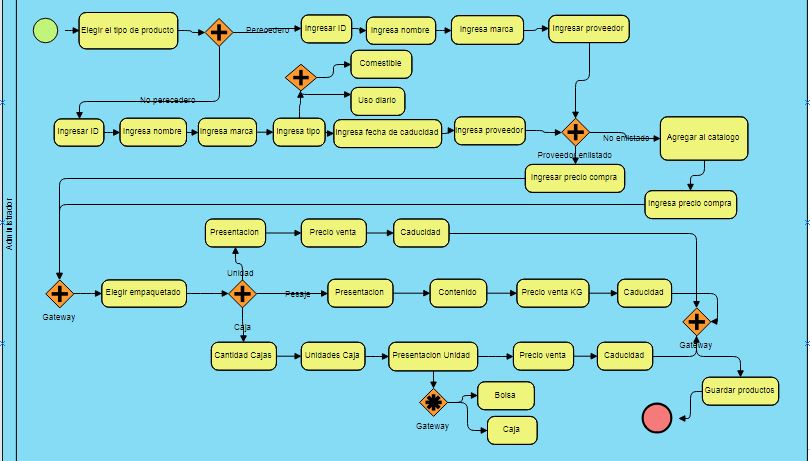
*2 Describir los datos de entrada o salida*

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*



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**

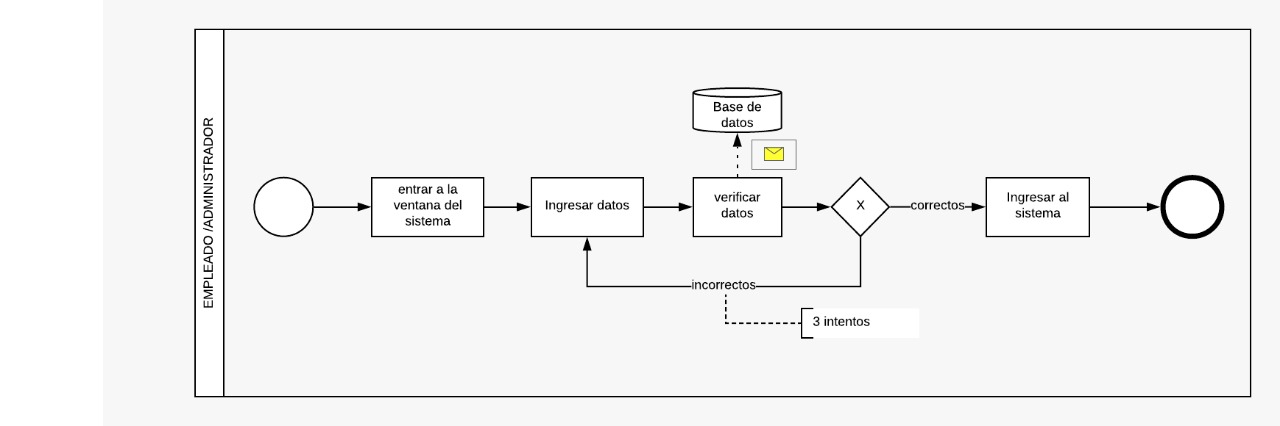
**En caja**

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**

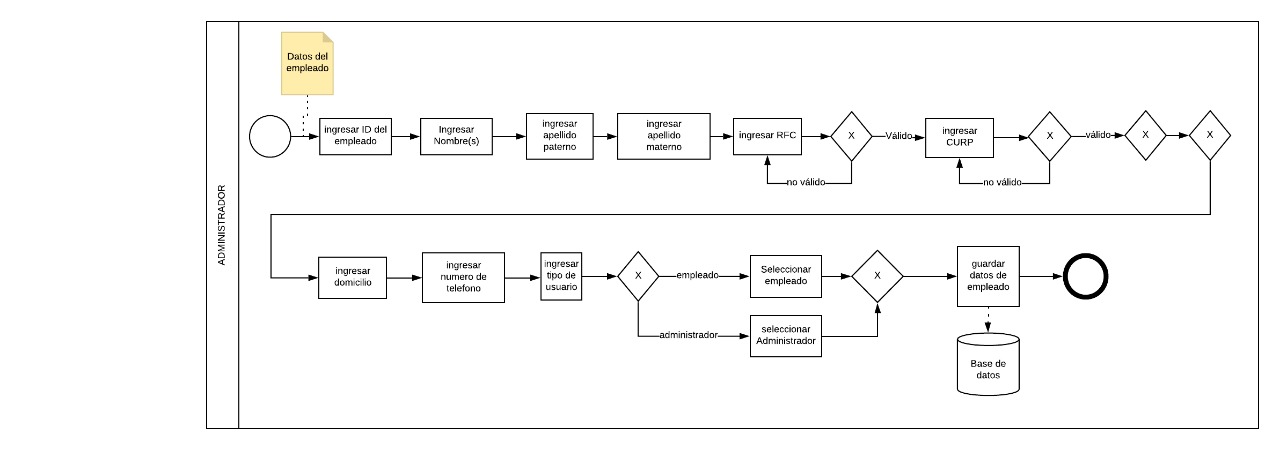
1. Ingresar su presentación.
2. Ingresar el precio de venta.
3. Ingresar la caducidad del producto
4. Guardar los datos del producto.

*Para el ingreso al sistema (login)*



La primera interfaz que se mostrara 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 ingresara 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*

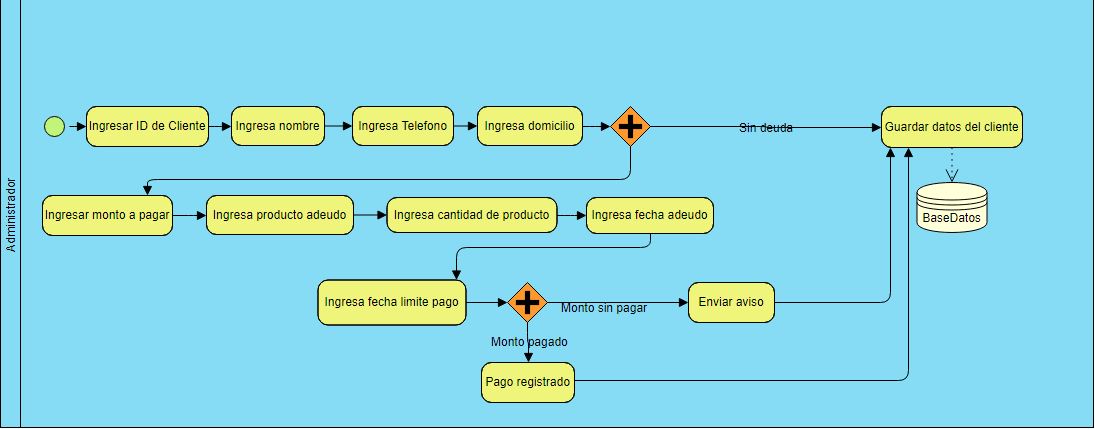
Para los usuarios que se registraran por primera vez al sistema se tendrán como base los siguientes puntos que requieren diversos datos de entrada para el sistema:

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.

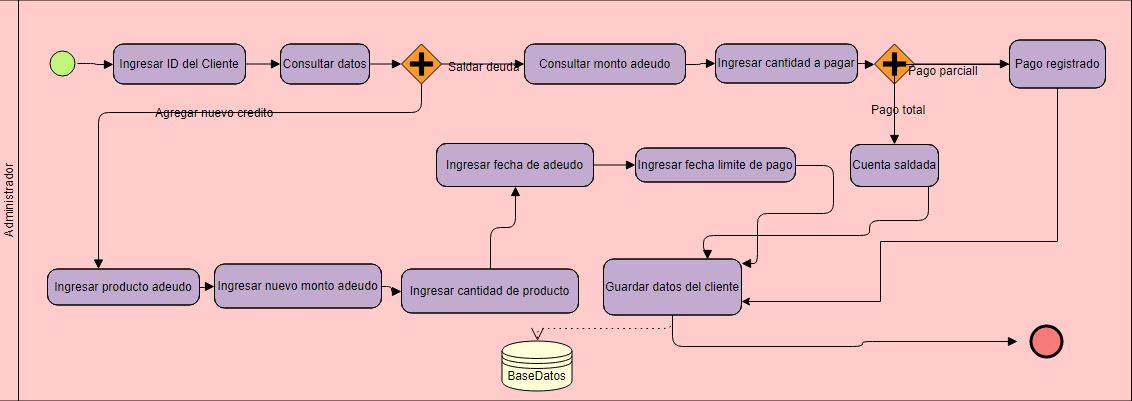
* **Segunda Iteración**

En esta segunda iteración se trabaja la **gestión de los clientes** la cual se divide en dos partes (la parte que será manejada por el administrador que incluye la agregación eliminación y búsqueda de clientes, así como la manejada únicamente por los empleados que se basará en la modificación de datos). La **gestión de los proveedores** (con las mismas funciones que tareas ya antes mencionadas) y **gestiones de compra** (que incluyen la realización, cancelamiento, estatus y mal venta de productos).

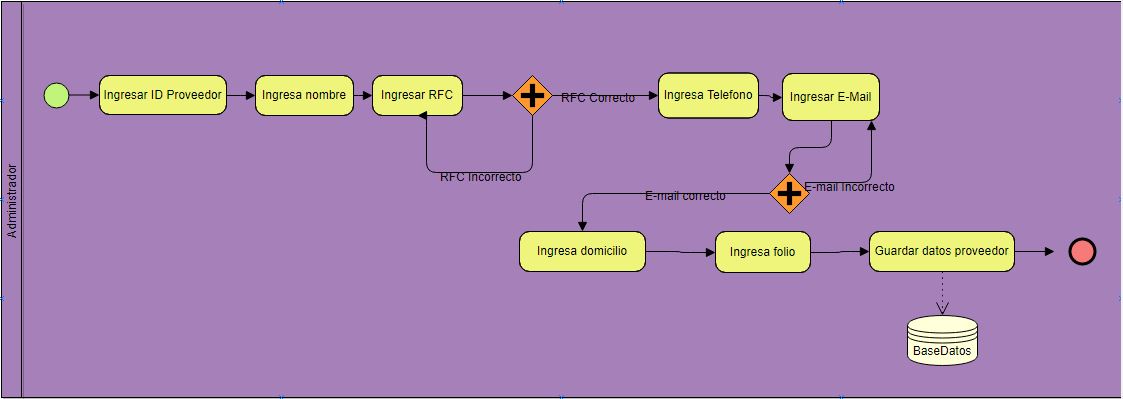
*Para el registro de un nuevo cliente*

**

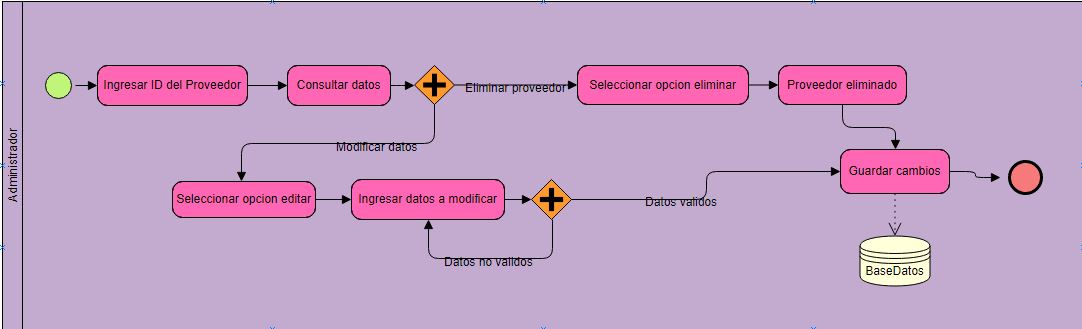
*Buscar clientes*

**

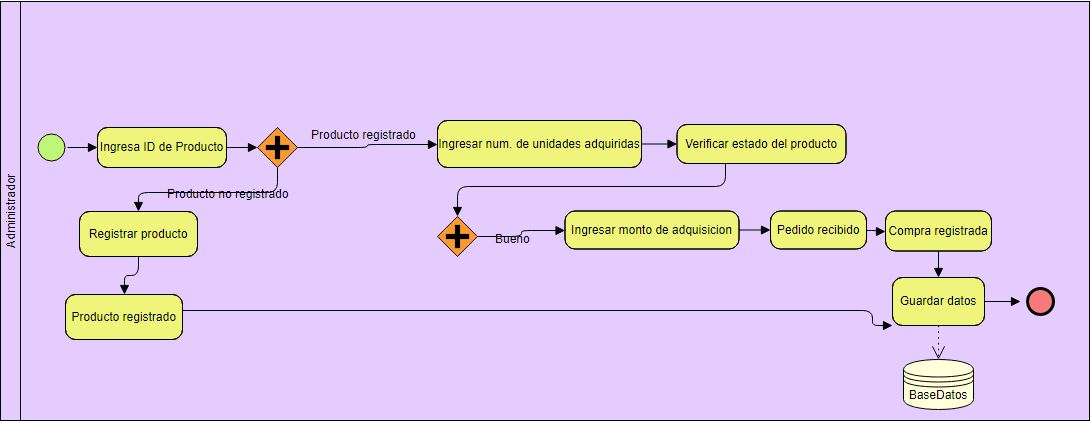
*Para el registro de proveedores*

**

*Para buscar y editar datos de un proveedor*

**

*Para la gestión de compra*

**

* **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).

**5.- Documentación Técnica – Especificación API**

*Utilizar una herramienta de generación de documentación automática.*

**6.- Conclusiones**

*Complicaciones encontradas durante el desarrollo del proyecto, políticas adaptadas para su resolución, restricciones al problema original, casos particulares y finalmente aspectos relacionados a la experiencia obtenida en base a la temática del proyecto.*