Contenido

[1 Introducción 2](#_Toc430677737)

[2 Descripción del producto y entregables 3](#_Toc430677738)

[3 Objetivos 4](#_Toc430677739)

[4 Justificación 4](#_Toc430677740)

[5 Alcances y limitaciones 5](#_Toc430677741)

[6 Herramientas 5](#_Toc430677742)

[7 Marco teórico 7](#_Toc430677743)

# 1 Introducción

**1.1 Antecedentes de la empresa**

Esta empresa fue fundada en 1990 por el Sr. Cuan Chio Hong de origen Chino en la colonia San José situada en Fortín Veracruz.

El empezó reciclando plástico de la Industria del Café con el tiempo se fue haciendo de prestigio y empezó a recibir materia prima de algunos proveedores con el tiempo la empresa fue creciendo en el año 2000 fue adquirida por el Sr Edmundo Lagunés Ordoñez, el cual le dio un giro a la empresa con nuevos clientes proveedores y maquinaria, la cual llevo a la empresa a progresar y llegar hasta donde está ahora.

En la actualidad es una empresa que recicla y fabrica bolsas de plástico de distintos tipos y tamaños.

**1.2 Descripción de la problemática**

La problemática que presenta la empresa Recicladoray Fábrica de Plásticos REFORMA, es que no lleva un control de sus compras y ventas de plásticos, tomando en cuenta que ellos compran la materia prima, la procesan y venden el producto secundario.

Algunos problemas son:

* Necesitan llevar un inventario de la materia prima que adquieren.
* Necesitan llevar un inventario de los productos que venden.
* Necesitan llevar un conteo semanal de ingresos y egresos.

Como propuesta de solución hemos pensado en crear una aplicación de escritorio para llevar el control de compras y ventas de dicha empresa, así como generar reportes de sus compras de materia prima y ventas.  
El conteo de gastos emitidos en cuanto a nómina y las ganancias.

# 2 Descripción del producto y entregables

El producto final que se desarrollará para la empresa Recicladoray Fábrica de Plásticos REFORMA será una aplicación de escritorio que contará con los siguientes módulos:

* **Módulo de Compra:** En este módulo la empresa podrá llevar el control de cada uno de los proveedores de los que adquieren materia prima, así como el egreso de dinero que cada uno de estos significa, la fecha en la que se adquiere la materia prima así como la cantidad que ingresa al almacén de la empresa.
* **Módulo de Venta:** En este módulo la empresa podrá llevar el control de cada una de las ventas que realizan en el transcurso del día, así como el ingreso de dinero que supone cada venta, la fecha en la que esta es realizada y el stock del producto que resta en el almacén de la empresa.
* **Módulo de Registro:** En este módulo la empresa podrá registrar, actualizar y/o eliminar clientes o proveedores con el cual se podrá realizar una factura por la compra o venta del producto y si el cliente o proveedor así lo desea con el fin de llevar un control más preciso de ingresos y egresos.
* **Módulo de Reportes:** En este módulo la empresa podrá realizar reportes ya sea de compras, ventas o realizar el inventario del almacén de la empresa se podrá realizar reportes por fechas ya sea por día, por semana o como el cliente lo prefiera.
* Este producto tiene como finalidad el poder llevar un control de compras y ventas que realiza la empresa así como también llevar un control de productos y de materia prima que cuenta.

# 3 Objetivos

**3.1 General**

* Desarrollar una aplicación de escritorio para la empresa Recicladoray Fábrica de Plásticos REFORMA que permita llevar el control de compras, ventas, llevar el inventario de los productos así como generar reportes de ingresos y egresos de dicha empresa.

**3.2 Específicos**

* Elegir las herramientas que se ocuparán para el desarrollo de la aplicación.
* Elegir el lenguaje en el cual se desarrollará la aplicación.
* Realizar la base de datos para la aplicación.
* Realizar los módulos necesarios con los que contará la aplicación.

# 4 Justificación

Debido a los problemas de control de venta y compra que presenta la empresa Recicladoray Fábrica de Plásticos REFORMA, se nos pidió desarrollar un sistema que facilite la administración adecuada y lleve la gestión de ventas, compras y productos de la empresa, el sistema será manejado por el personal que labora en dicha empresa.

Los beneficios que se obtendrán con el sistema serán:

* Llevar control absoluto de las ventas y compras que realice la empresa.
* Realizar reportes con la finalidad de llevar un control preciso de ingresos y egresos.
* Control y manejo de información más grande.

# 5 Alcances y limitaciones

**5.1 Alcances**

* La aplicación llevara el control de compras de la materia prima.
* La aplicación llevara el control de ventas de los productos.
* La aplicación llevara el control de horas que laboran los empleados.
* La aplicación llevara el control de entradas y salidas de los empleados.
* La aplicación llevara el control semanal de ingresos y egresos.

**5.2 Limitaciones**

* La aplicación solo llevara el control del departamento que se le indico inicialmente.
* Tiempo de elaboración para a la aplicación de escritorio y ejecutable.
* La aplicación no se conectara a internet.
* La aplicación no podrá enviar correos.

# 6 Herramientas

Lenguaje de programación (JAVA)

El lenguaje para la programación en Java, es un lenguaje orientado a objeto, de una plataforma independiente.

El lenguaje para la programación en Java, fue desarrollado por la compañía Sun Microsystems, con la idea original de usarlo para la creación de páginas WEB.

Otra de las utilidades de la programación en Java es el desarrollo de aplicaciones, que son programas que se ejecutan en forma independiente, es decir con la programación Java, se pueden realizar aplicaciones como un procesador de palabras, una hoja que sirva para cálculos, una aplicación gráfica, etc. en resumen cualquier tipo de aplicación se puede realizar con ella. Java permite la modularidad por lo que se pueden hacer rutinas individuales que sean usadas por más de una aplicación, por ejemplo tenemos una rutina de impresión que puede servir para el procesador de palabras, como para la hoja de cálculo.

La programación en Java, permite el desarrollo de aplicaciones bajo el esquema de Cliente Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar.

Base de Datos (BD)

La base de datos estará alojada en PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos*para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

**Especificaciones del equipo**

Sony VAIO

**Especificaciones Técnicas:**

Procesador: Procesador Intel® Core™ i5-3210M, 2.5GHz1 con Turbo Boost hasta 3.10GHz,

Intel® HM76 Express Chipset

Sistema Operativo: Windows 8.1 64 bits

Pantalla: Pantalla VAIO de 14" (1366 x 768), Tecnología backlight LED

Tarjeta de Video: Externa: AMD Radeon™ HD 7550M, 1GB

Memoria: 8GB (velocidad de 1600MT/s)

Disco Duro: HDD 500GB (velocidad de 5400 rpm)

Unidad Óptica: Reproductor y Grabador DVD

Ranuras de Expansión: Memory Stick Duo™4, SD memory card, entrada DC, USB 2.0 x3, Cargador USB 3.0 x1, salida HDMI, entrada para audífonos y micrófono

Audio: Intel® High Definition Audio, xLOUD, Clear Phase

Ethernet: 1000Base-T/100Base-TX/10Base-T

Red Inalámbrica: Bluetooth® estándar versión 4.0 + HS, IEEE 802.11b/g/n, Wi-Fi certificado.

**Herramientas para el desarrollo**

Eclipse: Se utilizará como entorno de desarrollo para la codificación de la aplicación en lenguaje JAVA

Scene Builder 2.0: Se utilizará como herramienta para realizar la interfaz gráfica de la aplicación así como los bosquejos y/o prototipos.

PgAdmin III: Se utilizará para diseñar, mantener y administrar las bases de datos de Postgres en lenguaje SQL.

Visual Paradigm: Se utilizará para realizar los diagramas de casos de uso así como el modelado en general que conlleva la parte de Análisis y Diseño.

# 7 Marco teórico

**Software:**

Se conoce como *software* *al equipo lógico o*[*soporte lógico*](https://es.wikipedia.org/wiki/Sistema_l%C3%B3gico) de un [sistema informático](https://es.wikipedia.org/wiki/Sistema_inform%C3%A1tico), que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados [hardware](https://es.wikipedia.org/wiki/Hardware).

Los componentes lógicos incluyen, entre muchos otros, las [aplicaciones informáticas](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica); tales como el [procesador de texto](https://es.wikipedia.org/wiki/Procesador_de_texto), que permite al usuario realizar todas las tareas concernientes a la edición de textos; el llamado [software de sistema](https://es.wikipedia.org/wiki/Software_de_sistema), tal como el [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo), que básicamente permite al resto de los programas funcionar adecuadamente, facilitando también la interacción entre los componentes físicos y el resto de las aplicaciones, y proporcionando una [interfaz](https://es.wikipedia.org/wiki/Interfaz_de_usuario) con el usuario.

**Hardware:**

El término *hardware* se refiere a todas las partes físicas de un [sistema informático](https://es.wikipedia.org/wiki/Sistema_inform%C3%A1tico); sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos. Son cables, gabinetes o cajas, [periféricos](https://es.wikipedia.org/wiki/Perif%C3%A9rico_(inform%C3%A1tica)) de todo tipo y cualquier otro elemento físico involucrado; contrariamente, el soporte lógico es intangible y es llamado [*software*](https://es.wikipedia.org/wiki/Software). El término, aunque sea lo más común, no solamente se aplica a las computadoras; del mismo modo, también un [robot](https://es.wikipedia.org/wiki/Robot), un [teléfono móvil](https://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil), una [cámara fotográfica](https://es.wikipedia.org/wiki/C%C3%A1mara_fotogr%C3%A1fica), un [reproductor multimedia](https://es.wikipedia.org/wiki/Reproductor_multimedia_digital) o cualquier otro electrónico que procese datos poseen *hardware* (y *software*).

**Requisito Funcional:**

Un [requisito](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_requisitos) funcional define una función del sistema de [software](https://es.wikipedia.org/wiki/Software) o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requisitos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, un sistema debe cumplir. Los requerimientos de comportamiento para cada requerimiento funcional se muestran en los [casos de uso](https://es.wikipedia.org/wiki/Casos_de_uso). Son complementados por los [requisitos no funcionales](https://es.wikipedia.org/wiki/Requisito_no_funcional), que se enfocan en cambio en el diseño o la implementación.

**Requisito no Funcional:**

Un requisito no funcional o atributo de calidad es, en la [ingeniería de sistemas](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_sistemas) y la [ingeniería de software](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software), un [requisito](https://es.wikipedia.org/wiki/Requisito_(sistemas)) que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los [requisitos funcionales](https://es.wikipedia.org/wiki/Requisito_funcional). Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar.

**Java:**

Java es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) de [propósito general](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_de_prop%C3%B3sito_general), [concurrente](https://es.wikipedia.org/wiki/Computaci%C3%B3n_concurrente), [orientado a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos) que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los [desarrolladores](https://es.wikipedia.org/wiki/Desarrollador_de_software) de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el [código](https://es.wikipedia.org/wiki/C%C3%B3digo_fuente) que es ejecutado en una plataforma no tiene que ser [recompilado](https://es.wikipedia.org/wiki/Compilaci%C3%B3n_en_tiempo_de_ejecuci%C3%B3n) para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de [cliente-servidor](https://es.wikipedia.org/wiki/Cliente-servidor) de web, con unos 10 millones de usuarios reportados.

**Modelo Vista Controlador:**

El modelo–vista–controlador (MVC) es un patrón de [arquitectura de software](https://es.wikipedia.org/wiki/Arquitectura_de_software) que separa los [datos](https://es.wikipedia.org/wiki/Datos) y la [lógica de negocio](https://es.wikipedia.org/wiki/L%C3%B3gica_de_negocio) de una aplicación de la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres [componentes](https://es.wikipedia.org/wiki/Componentes) distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de [arquitectura de software](https://es.wikipedia.org/wiki/Arquitectura_de_software) se basa en las ideas de [reutilización de código](https://es.wikipedia.org/wiki/Reutilizaci%C3%B3n_de_c%C3%B3digo) y la [separación de conceptos](https://en.wikipedia.org/wiki/Separation_of_concerns), características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

**Scene Builder:**

JavaFX Scene Builder es una herramienta de diseño visual que permite a los usuarios diseñar rápidamente interfaces de usuario de aplicaciones JavaFX, sin necesidad de programación. Los usuarios pueden arrastrar y soltar los componentes de interfaz de usuario a un área de trabajo, modificar sus propiedades, solicitar las hojas de estilo, y el código FXML para la disposición de que están creando se genera automáticamente en segundo plano. El resultado es un archivo FXML que a continuación se puede combinar con un proyecto Java mediante la unión de la interfaz de usuario a la lógica de la aplicación.

**Eclipse:**

Eclipse es un [programa informático](https://es.wikipedia.org/wiki/Programa_inform%C3%A1tico) compuesto por un conjunto de herramientas de programación de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto) multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar [entornos de desarrollo integrados](https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) (del inglés IDE), como el IDE de [Java](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java) llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

**PostgreSQL:**

PostgreSQL es un [Sistema de gestión de bases de datos](https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos) [relacional](https://es.wikipedia.org/wiki/Modelo_relacional) [orientado a objetos](https://es.wikipedia.org/wiki/Base_de_datos_orientada_a_objetos) y [libre](https://es.wikipedia.org/wiki/Software_libre), publicado bajo la [licencia PosgreSQL](https://es.wikipedia.org/w/index.php?title=Licencia_PosgreSQL&action=edit&redlink=1) , similar a la [BSD](https://es.wikipedia.org/wiki/Licencia_BSD) o la [MIT](https://es.wikipedia.org/wiki/Licencia_MIT).

Como muchos otros proyectos de [código abierto](https://es.wikipedia.org/wiki/Software_de_c%C3%B3digo_abierto), el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, [altruista](https://es.wikipedia.org/wiki/Altruismo), libre y/o apoyados por [organizaciones comerciales](https://es.wikipedia.org/wiki/Empresas). Dicha comunidad es denominada el [PGDG](https://es.wikipedia.org/w/index.php?title=PostgreSQL_Global_Development_Group&action=edit&redlink=1) (PostgreSQL Global Development Group).

**SQL:**

El lenguaje de consulta estructurado o SQL (por sus siglas en [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s) Structured Query Language) es un [lenguaje declarativo](https://es.wikipedia.org/wiki/Lenguajes_declarativos) de acceso a [bases de datos](https://es.wikipedia.org/wiki/Base_de_datos) relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del [álgebra](https://es.wikipedia.org/wiki/%C3%81lgebra_relacional) y el [cálculo relacional](https://es.wikipedia.org/wiki/C%C3%A1lculo_relacional) que permiten efectuar [consultas](https://es.wikipedia.org/w/index.php?title=Consulta_(base_de_datos)&action=edit&redlink=1) con el fin de recuperar, de forma sencilla, [información](https://es.wikipedia.org/wiki/Informaci%C3%B3n) de bases de datos, así como hacer cambios en ellas.

**Modelo Entidad-Relación:**

Un diagrama o modelo entidad-relación (a veces denominado por sus siglas en inglés, *E-R* "Entity relationship", o del español *DER* "Diagrama de Entidad Relación") es una herramienta para el [modelado de datos](https://es.wikipedia.org/wiki/Modelo_de_datos) que permite representar las entidades relevantes de un [sistema de información](https://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n) así como sus interrelaciones y propiedades.

**Modelo Relacional:**

El modelo relacional, para el modelado y la gestión de [bases de datos](https://es.wikipedia.org/wiki/Bases_de_datos), es un [modelo de datos](https://es.wikipedia.org/wiki/Modelo_de_datos) basado en la [lógica de predicados](https://es.wikipedia.org/wiki/L%C3%B3gica_de_primer_orden) y en la [teoría de conjuntos](https://es.wikipedia.org/wiki/Teor%C3%ADa_de_conjuntos).

Su idea fundamental es el uso de relaciones. Estas relaciones podrían considerarse en forma lógica como [conjuntos de datos](https://es.wikipedia.org/wiki/Conjuntos_de_datos) llamados [tuplas](https://es.wikipedia.org/wiki/Tupla" \o "Tupla). Pese a que esta es la teoría de las bases de datos relacionales creadas por Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar, pensando en cada relación como si fuese una [tabla](https://es.wikipedia.org/wiki/Tabla_(base_de_datos)) que está compuesta por [registros](https://es.wikipedia.org/wiki/Registro_(base_de_datos)) (cada fila de la tabla sería un registro o "tupla") y [columnas](https://es.wikipedia.org/wiki/Columna_(base_de_datos)) (también llamadas "[campos](https://es.wikipedia.org/wiki/Campo_(inform%C3%A1tica))").

Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

**Diccionario de datos:**

Un diccionario de datos es un conjunto de [metadatos](https://es.wikipedia.org/wiki/Metadato) que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

Es un catálogo, un depósito, de los elementos en un sistema. Como su nombre lo sugiere, estos elementos se centran alrededor de los datos y la forma en que están estructurados para satisfacer los requerimientos de los usuarios y las necesidades de la organización. En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos en todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario guarda los detalles y descripciones de todos estos elementos.

**Visual Paradigm**

Visual Paradigm para UML (VP-UML) es un [UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language)[CASE](https://en.wikipedia.org/wiki/Computer-aided_software_engineering) herramienta de apoyo UML 2, SysML y [notación de modelado de procesos de negocio (BPMN)](https://en.wikipedia.org/wiki/Business_Process_Modeling_Notation) desde el Object Management Group (OMG). Además del apoyo de modelado, que ofrece generación de informes y capacidades de ingeniería de código, incluyendo [la generación de](https://en.wikipedia.org/wiki/Automatic_programming)código. Se puede [invertir ingeniero](https://en.wikipedia.org/wiki/Reverse_engineer) diagramas de código, y proporcionar [la ingeniería de ida y vuelta](https://en.wikipedia.org/wiki/Round-trip_engineering) para los diversos [lenguajes de](https://en.wikipedia.org/wiki/Programming_language)programación.

Cortés Meza Cynthia Yuridiana, Lagunes Mellado Dylan Israel & Reyes Juarez Jesús Aldahir. (2015). Descripción del proyecto. Paso del Macho; Ver: UTCV.