Universidad de Oriente.

Nucleó Anzoátegui.

Escuela de Ingeniería y Ciencias Aplicadas.

Departamento de Ingeniería.

Desarrollo de Software.



Desarrollo de una aplicación para el control de compra y venta de un

supermercado ubicado en la zona norte del Estado Anzoátegui

Tutor Profesor Ing. Víctor Mujica

Grupo N° 6. Integrantes:

Manuel Dun C.I: 19.738.854

Frank Rondón 19.738.854

Luis Correa C.I: 19.840.230

Copyright © 2015 por Luis Correa & Manuel Dun & Frank Rondón. Todos los derechos reservados.

Barcelona, 10 de Junio de 2015

**Tabla de Contenidos**

**1.** **Modelar la situación actual (Diagrama de actividad)** 3

**2.** **Requisitos funcionales y no funcionales** 4

2.1 Requisitos funcionales 4

2.2 Requisitos no funcionales 5

**3.** **Requisitos para el desarrollo del sistema (Diagrama de Topología de Red)** 5

**4.** **Modelar el sistema con los requisitos (Diagrama de Casos de Uso)** 6

**5.** **Detallar el Modelado del Sistema (Diagrama de Secuencia)** 8

**6.** **Explicar la Integración del Sistema (Diagrama de MVC)** 9

**7.** **Modelar la BD y explicar (Diagrama de E/R)** 10

**8.** **Prototipo de la Interfaz de Usuario (Las ventanas y como se utilizan que se utilizó)** 10

**9.** **Lista de referencias** 11

# **Modelar la situación actual (Diagrama de actividad)**

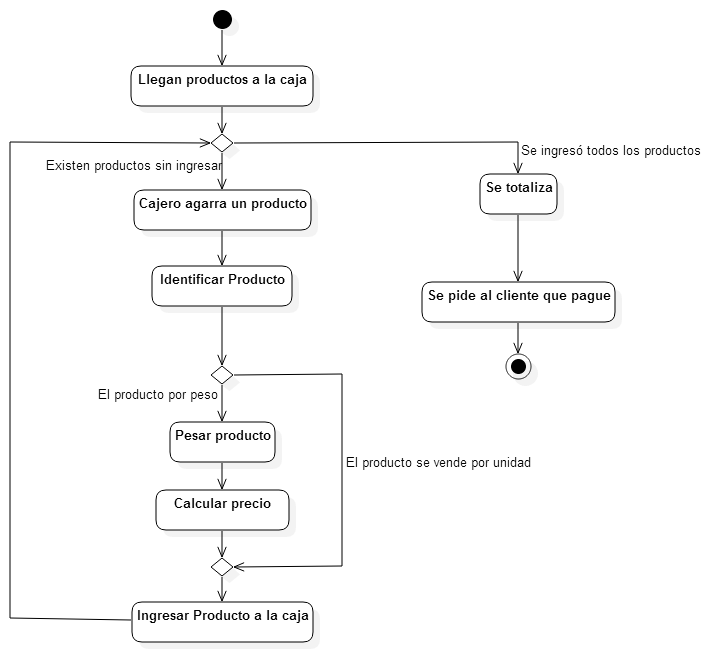


Figura N° 1 Diagrama Actividades de la aplicación del supermercado.

La deficiencia que se puede observar se encuentra en el paso de identificar producto, éste se puede optimizar para que ocurra con facilidad para el cajero y para que ocurra con mayor rapidez. Nuestro programa logrará este objetivo con la utilización de bases de datos relacionales e interfaz gráfica que permita una mejor experiencia de usuario para el cajero y los clientes del establecimiento comercial.

# **Requisitos funcionales y no funcionales**

### Requisitos funcionales

El sistema de software que se desarrollara para el supermercado es un sistema de software netamente administrativo que contendrá las siguientes características y funciones:

* Fácil búsqueda de productos.
* Avisos oportunos de falta de productos.
* Ingreso a la base de datos los productos
* En el módulo de inventario, se agregará una lista donde esté contenido los productos, detallando el tipo de producto que se venderá (hortalizas, cereales, detergentes, entre otros), su código y se especificará el nombre del producto, su precio costo unitario, el porcentaje que se le ganará al producto para que el sistema con sus algoritmos, calculara el precio de venta al público, incluyendo el impuesto al valor agregado (IVA), que las leyes venezolanas es de un 12%.
* En el módulo compras, se agregará una lista de proveedores donde se cargará la información del mismo que está comprendida por el nombre del proveedor (sea una persona jurídica o una persona natural), la dirección y los números de contacto. Luego de cargada esta información se procederá a cargar la factura de compra, que comprende el nombre del proveedor, el número de factura, la descripción detallada de cada producto que se le compran con las cantidades de cada producto. Para que luego el sistema cargue las cantidades existentes de cada ítem o producto disponible para la venta. Y así de la información que se tenga registrada en el módulo inventario, se actualizara. Únicamente se actualizan los inventarios si se carga una factura de compra. De lo contrario el producto se puede generar, pero en inventario aparecerá que el producto no posea ninguna existencia del mismo, o en el argot comercial, no halla stock disponible.
* En el módulo de facturación tendrá una opción de agregar clientes, donde se agregaran los siguientes datos: Código, Rif (o cédula), nombre, dirección y teléfono. Luego de agregado el cliente se procede a crear la factura, que está compuesto por los siguientes atributos: Nombre del cliente y su rif o cédula, nombre del vendedor y los productos que compra el cliente. Se agrega al cliente y se cargan los productos que compra el cliente incluyendo las cantidades, estas cantidades se van restando del inventario, luego se totaliza el monto de la venta, que muestra el subtotal de la venta que es el valor del producto sin los impuestos, luego se muestra el total de IVA a pagar por el cliente, se muestra el total a pagar por el cliente, se registra el monto que cancelará y se genera la factura.
* En el módulo de reportes, permitirá generar el total de ventas y compras que se han realizado por un determinado período de tiempo (diario, semanal, mensual o especificando una fecha). Generará cierres fiscales como lo son el reporte X, que es el total de las ventas que se hicieron en un día mas no cierra la caja, y está un reporte Z, que a diferencia del reporte X, éste da el total de las ventas con información detallada de cada factura, cierra la caja y el reporte generado es válido para guardar los registros fiscales de ventas, que se le entregan a los contadores, que luego declara dichos reportes al SENIAT.

### Requisitos no funcionales

* Rendimiento: La aplicación contara con un alto rendimiento ya que utilizaremos el sistema gestor de base de datos Mysql y el lenguaje de programación Java ambos utilizados en múltiples proyectos empresariales por su alto rendimiento.
* Disponibilidad: Estará disponible para los empleados y el administrador.
* Seguridad: El administrador tendrá una ventana diferente a la del empleado con lo cual el empleado no tendrá acceso a funciones que si podrá acceder el administrador, se utilizaran procedimientos almacenados para evitar ataques de inyección de sql.
* Accesibilidad: Este es un programa, que solo estará diseñado para escritorio y tendrá dos tipos de usuarios: Cajeros y administrador. Administrador tiene habilitada todas las funciones, compra, venta, etc, mientras que la sesión cajero solo podrá acceder al módulo de ventas y reportes.
* Usabilidad: En proceso de facturación, se implementaron un esquema para optimizar el rápido ingreso de datos a la hora de compra.
* portabilidad: Portable debido a que se utiliza la Orientación a Objetos en el código Java.
* Bajo costo: La aplicación se realizó con herramientas de software libre y no tendremos que pagar ninguna licencia.
* Escalabilidad: Esta aplicación se puede seguir realizando en un futuro más modificaciones, dependiendo de las necesidades que tenga el cliente o desarrollar una versión más actualizada.
* Concurrencia: La BD Mysql maneja internamente la concurrencia.
* Mantenibilidad: El mantenimiento del software se realizará cuando el cliente requiera q tenga una función adicional al programa.
* interfaz gráfica: Garantiza la fácil utilización y alta velocidad de procesamiento de datos.
* Estabilidad: Se utilizara un manejador de base de datos Mysql que además de rápido, es estable y totalmente gratis.

# **Requisitos para el desarrollo del sistema (Diagrama de Topología de Red)**

El supermercado contara con una topología de árbol, la cual es llamada así por su apariencia estética, por la cual puede comenzar con la inserción del servicio de internet desde el proveedor pasando por el router luego por un switch y este deriva a otro switch u otro router o sencillamente a los [hosts](http://es.wikipedia.org/wiki/Host) (estaciones de trabajo), el resultado de esto es una red con apariencia de árbol porque desde el primer router que se tiene se ramifica la distribución de [Internet](http://es.wikipedia.org/wiki/Internet).

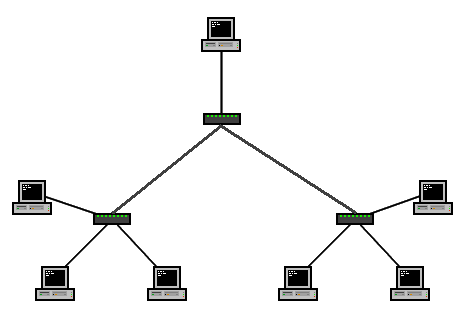


Figura N° 2 Diagrama de Topología de Red del supermercado.

# **Modelar el sistema con los requisitos (Diagrama de Casos de Uso)**

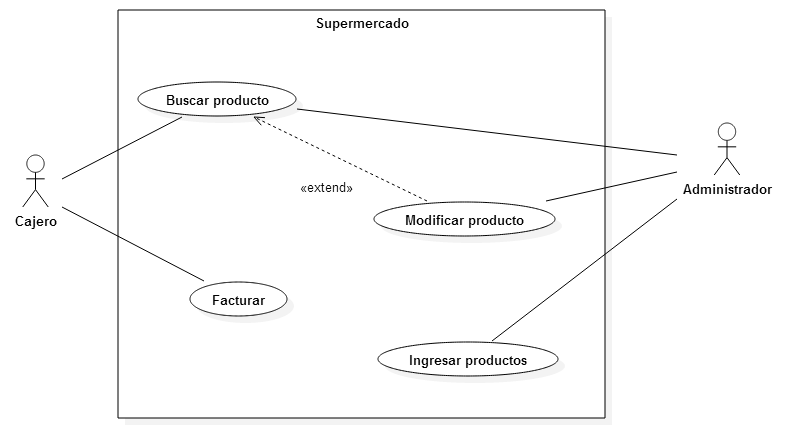


Figura N° 3 Diagrama de Casos de Uso de la aplicación del supermercado.

* **Facturar:**

Descripción:

Realiza transacción de compra.

Actores:

Cajero

Precondición:

El cajero debe estar en la caja con el programa de facturación corriendo y el cliente debe llevar los productos a la caja.

Flujo normal:

El cliente lleva los productos que va a comprar a la caja. El cajero procede a ingresar uno por uno todos los productos en el sistema de facturación. El programa da el total a pagar. El cliente paga. El cajero genera la factura.

Flujo alternativo:

Si el cliente no posee suficiente dinero se cancela la transacción.

Poscondiciones:

Los productos son sacados de la base de datos y el dinero obtendido del cliente.

**Buscar Producto:**

Descripción:

Buscar todas las características del producto incluyendo disponibilidad, lugar y precio.

Actores:

Administrador y cajero.

Precondición:

Tener acceso al software.

Flujo normal:

El usuario abre la opción de buscar producto. Ingresa texto descriptivo. El programa muestra resultados según el texto descriptivo mostrando también información referente al producto. El usuario elige un producto para obtener mas información del producto.

Flujo alternativo:

Si el usuario es administrador el software mostrará información que puede considerarse confidencial.

Poscondiciones:

El usuario debería haber obtenido la información requerida.

**Modificar Producto:**

Descripción:

Proceso en el que se modifica los datos de los productos del establecimiento.

Actores:

Administrador.

Precondición:

Haber buscado y seleccionado el producto a modificar.

Flujo normal

El administrador busca el producto. Lo selecciona. Modifica todos los atributos del producto que quiere modificar a travez de una interfaz gráfica ingresando la información necesaria.

Flujo alternativo: Si la información no está validada no permitir la modificación de la base de datos.

Poscondicion:

La base de datos debe quedar modificada como corresponde.

# **Detallar el Modelado del Sistema (Diagrama de Secuencia)**

public class AppMercado1 {

public static void main(String[] args) {

MVentana mve=new MVentana();

VVentana vve= new VVentana(mve);

CVentana cve = new CVentana(vve,mve);

}

}

Código N° 1 Código de la clase principal main.

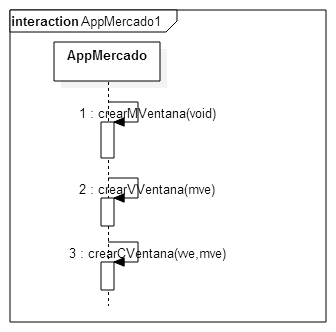


Figura N° 4 Diagrama de Secuencia del inicio de sesión de la aplicación del supermercado.

# **Explicar la Integración del Sistema (Diagrama de MVC)**

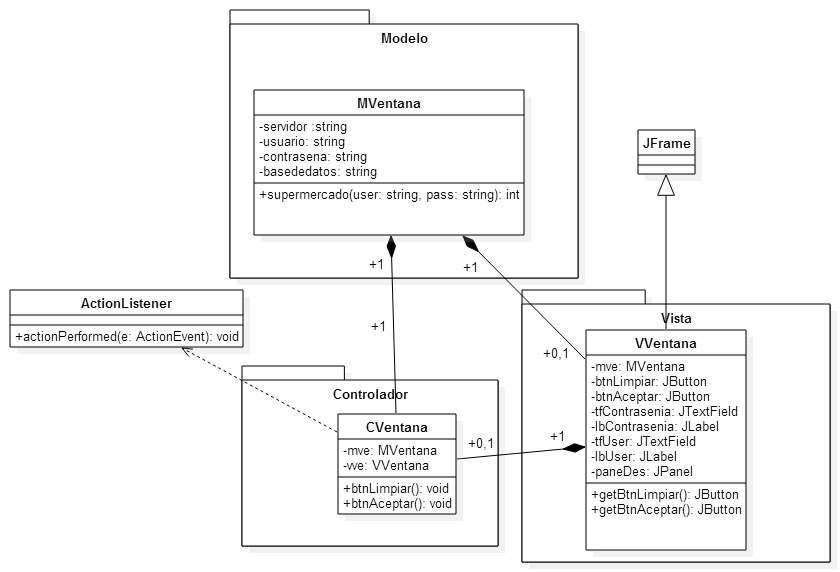


Figura N° 5 Diagrama de Paquetes del Modelo-Vista-Controlador de la ventana

de inicio de sesión de la aplicación del supermercado.

Clase MVentana: En la cual se maneja la lógica de la ventana principal la cual con el método supermercado que recibe usuario y contraseña se conectara con la base de datos para verificar si el usuario y la contraseña coinciden con lo que se tiene en la tabla empleados luego retornara un valor numérico que será uno para administrador y 2 para administrador si no existe será veinte.

Clase VVentana: En la cual se maneja todo lo relacionado a la interfaz gráfica para permitir al vendedor y al administrador trabajar en el mercado.

Clase CVentana: En la cual se controlaran los eventos de cada elemento de la interfaz gráfica que se requiera.

# **Modelar la BD y explicar (Diagrama de E/R)**

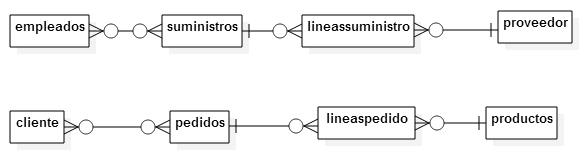


Figura N° 6 Diagrama Entidad Relación de la base de datos del supermercado.

* Relación empleados-suministros: En la tabla empleados solo tendremos una tupla de un empleado administrador que será la única relacionada con todas las tuplas suministros.
* Relación suministros-lineassuministros: Cada tupla de suministro tendrá múltiples tuplas en la tabla líneassuministro y lineassuministro tendrán un id que lo relacionara con algún suministro.
* Relación lineassuministros-proveedor: Las tuplas de lineassuministros tienen una sola tupla proveedor, mientras que cada tupla de la tabla proveedor estará relacionada con múltiples tuplas de lineassuministro.
* Relación cliente-pedidos: Cada tupla de la tabla cliente tendrá múltiples tuplas en la tabla pedidos.
* Relación pedidos-lineaspedidos: Cada tupla de la tabla pedidos tendrá múltiples tuplas en la tabla lineaspedidos.
* Relación lineaspedido-productos: Cada tupla de la tabla lineaspedidos tendrá una sola tupla en productos.

# **Prototipo de la Interfaz de Usuario (Las ventanas y como se utilizan que se utilizó)**



Figura N° 7 Ventana de inicio de sesión.

Tenemos JFrame que tiene cuadros de texto uno para el usuario y otro para la contraseña, dos botones uno para iniciar sesión y poder visualizar la ventana administrador o vendedor dependiendo del usuario, y otra botón limpiar para borrar lo escrito en los cuadros de texto usuario y contraseña.

# **Lista de referencias**

* Elmasri, N. (2000). Sistemas De Bases De Datos. Conceptos Fundamentales (2da ed.), Ciudad de México, México: Addison Wesley.
* James Rumbaugh, Ivar Jacobson, Grady Booch 2007. El Lenguaje de Modelado Unificado Ed 2. Madrid. Addison Wesley.