



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
La Universidad Católica de Loja

ESCUELA DE CIENCIAS DE LA COMPUTACIÓN

PROGRAMACIÓN DE ALGORITMOS



Profesional en formación:

Carchi Ludeña Katherine Mishell

Docente:

Ing. Quezada Sarmiento Pablo Alejandro

Loja – Ecuador

TEMA:

Proceso de facturación comercial

OBJETIVOS

General:

Realizar un sistema de facturación que permita al usuario realizar este proceso de manera ágil y rápida. Que puedan tener un informe sobre sus clientes y la mercadería registrada.

Específicos:

- Brindar un programa capaz de mantener la información actualizada
- Crear un programa flexible a modificaciones y mejoras
- Poner en práctica los conocimientos que día a día vamos adquiriendo dentro de la materia.

JUSTIFICACIÓN:

Actualmente el proceso de facturación es muy importante dentro de una actividad comercial ya que ayuda tanto al vendedor a mantener un registro de sus actividades económicas y verificar si obtuvo pérdidas o ganancias; al comprador a tener una constancia sobre la adquisición que realizó y a la entidad tributaria para mantener un control sobre los ingresos y gastos que obtiene tanto el vendedor como el comprador.

Un sistema de facturación dentro de una empresa es muy beneficioso ya que agiliza el proceso de generar una factura. Permite al vendedor emitir el documento de forma automática y continuar atendiendo a sus clientes. De esta manera el vendedor ya no tendrá que realizar la factura de forma manual pues si la compra se realiza en grandes cantidades se evitará la tarea de escribir dentro del documento producto por producto y hacer que el cliente espere demasiado tiempo mientras realiza la factura. Con el sistema de facturación el vendedor simplemente deberá seleccionar el producto y registrar la cantidad evitándose así escribir.

Un sistema de facturación obtiene rápidamente el costo total de la compra pues ya tiene registrado el producto y su costo.

Como una factura debe contener todos los datos del cliente el sistema de facturación permite al vendedor obtener un registro de sus clientes llenando así los campos de la factura de forma automática buscándolo dentro de la base de datos con el número de cédula que es un código único que diferencia a una persona de otra. De igual manera si el cliente no se encuentra registrado se lo puede agregar a la lista.

Crece es el objetivo de la empresa y empezar con su servicio es la clave. Si se brinda un servicio de calidad, automatizado, eficiente y rápido hará que la empresa mejore sus resultados económicos y cumpla con servir a sus clientes.

El presente proyecto está destinado a crear un sistema de facturación que cumpla con estos beneficios y así satisfacer y mejorar la actividad comercial dentro de la empresa.

ALCANCE

El presente proyecto está diseñado en un sistema de facturación que permite a una empresa mantener un registro de sus clientes y mercaderías para hacer el proceso de facturación ágil, rápido y sencillo.

Debido a que el programa servirá para la administración tanto de cliente como de productos que tiene una empresa vamos a considerar las siguientes características:

- El programa generará una factura con los datos del cliente y los artículos correspondientes a la compra
- Permitirá al usuario el ingreso de datos
- Permitirá la búsqueda de datos de clientes mediante el uso de una base de datos

Estas son las características que poseerá la aplicación. A más de ello podemos decir que esta aplicación constará de 10 interfaces en donde el usuario podrá interactuar con la aplicación. A continuación se realizará una breve descripción de las interfaces que componen la aplicación:

La primera ventana es una presentación del sistema de facturación KAFACT 3000 donde el usuario podrá dar inicio al programa con tan solo dar clic en el botón comenzar.

Seguidamente se le presentará al usuario una ventana en la que en la barra de menú constarán los módulos con los que el usuario puede interactuar ya sea a la opción de clientes, mercaderías, factura o ayuda.

En el menú clientes se presentarán 3 opciones entre las cuales tendrá Agregar Cliente, Buscar Cliente.

Dentro de la opción Agregar Cliente el usuario tendrá que llenar todos los campos con la información del cliente como es: nombres, apellidos, número de cédula, dirección y teléfono. Y por último guardar la información. En la opción buscar cliente el usuario mediante el número de cédula, podrá buscar al cliente. Y en la opción ver lista se presentará el registro de todos los clientes.

La opción de mercaderías constará igualmente de 3 opciones que describimos en la opción anterior con la diferencia que con esta interfaz se administrara la mercadería más no los clientes.

En la interfaz de factura el usuario podrá visualizar todos los campos que tiene una factura pero de forma digital. Estos son: nombre del Cliente, cedula de identidad, dirección, los datos de todos los artículos que el cliente va a adquirir así como el subtotal, IVA, y el total de la factura.

LIMITACIONES

Entre las limitaciones que tendrá nuestra aplicación tendremos las siguientes:

- Excepciones debido a la utilización de base de datos e interfaces graficas
- No se podrá mostrar informes de ningún tipo referentes a la información que posee la base de datos con respecto a clientes, mercaderías y facturas.
- No se validaran los datos registrados del cliente por el usuario
- No se aplicaran descuentos ni tampoco se podrá imprimir puesto que es un sistema de facturación a prueba
- No se llevará un control de stock
- No se podrá eliminar los clientes desde el programa puesto que es información confidencial se lo podrá realizar solo desde la base de datos

MARCO TEORICO

LA EMPRESA:

Una empresa es una organización, institución o industria dedicada a actividades o persecución de fines económicos o comerciales para satisfacer las necesidades de bienes y servicios de los demandantes.[1] Es una unidad de producción, de decisión, financiera y organizada. Cumplen una función económica, proporcionan una fuente de empleo y una función social. [2]

La empresa comercial es aquella que sus actividades están destinadas a la compra y venta de productos terminados [2], busca sustentabilidad, ganancia y crecimiento. Es muy importante para esta empresa administrar sus actividades con la finalidad de conocer si se ha obtenido una ganancia o una pérdida durante sus actividades económicas. De igual manera uno de sus requisitos como empresa está el emitir una factura.

FACTURA :

La factura comercial es un documento mercantil que refleja toda la información de una operación de compraventa. Se considera como el justificante fiscal de la entrega de un producto o de la provisión de un servicio, que afecta al vendedor y al comprador. [3] La emisión de la misma es una obligación para el vendedor por parte del Servicio de Rentas Internas con la finalidad de controlar los ingresos que el empresario manifiesta que obtiene.

CLIENTE:

El usuario es muy importante dentro de una empresa ya que sin usuario no se produce una actividad comercial pues se necesita una persona que compre los productos que se están ofreciendo. Así mismo se busca satisfacer la necesidad de las personas y para la empresa es indispensable tener una estrategia de venta, haciendo que el usuario encuentre el producto que está buscando, obtenga un buen servicio y transforme al usuario en CLIENTE.

Para poder desarrollar su actividad la empresa necesita disponer de una tecnología que especifique que tipo de factores productivos precisa y como se combinan [4].

Es por ello que este proyecto está destinado a ser una herramienta que aporte al desarrollo de las actividades de la empresa. En base de que el registro y la contabilidad es la actividad más importante dentro de la misma mi propósito es brindar un sistema de facturación simple y eficiente que facilite al usuario la emisión de facturas, registro de su mercadería y de sus clientes. También se desea entender cómo funciona éste dentro de una empresa y dejar un sistema que pueda ir creciendo junto con ella.

LENGUAJE DE PROGRAMACION JAVA

2

Dentro de nuestro programa se utilizará el lenguaje Java.

Java fue diseñado por James Gosling, de Sun Microsystems, en 1990, como software para dispositivos electrónicos de consumo, como calculadoras y microondas. Inicialmente se llamó Oak (roble en inglés), aunque tuvo que cambiar debido a que dicho nombre ya estaba registrado por otra empresa. [5]

Java es un lenguaje moderno, de alto nivel, que recoge los elementos de programación que típicamente se encuentran en todos los lenguajes de programación, permitiendo la realización de programas profesionales.

Java se dice orienta a objetos pues, los objetos son los elementos básicos para modelar los datos sobre los que trabaja un programa. es un lenguaje sencillo , o todo lo sencillo que puede ser un lenguaje orientado a objetos, eliminando la mayor parte de los problemas de C++. Es un lenguaje independiente de plataforma, por lo que un programa hecho en Java se ejecutará igual en un PC con Windows que en una estación de trabajo basada en Unix. También hay que destacar su seguridad, desarrollar programas que accedan ilegalmente a la memoria o realizar caballos de troya es una tarea propia de titanes.

NETBEANS

Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

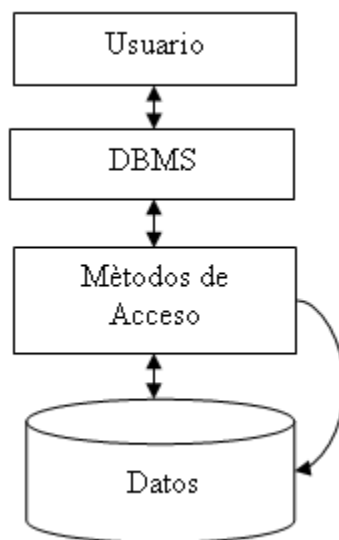
La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software [6].

BASE DE DATOS

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso [7].

Componentes de una Base de Datos:

- Hardware: constituido por dispositivo de almacenamiento como discos, tambores, cintas, etc.
- Software: que es el DBMS o Sistema Administrador de Base de Datos.
- Datos: los cuales están almacenados de acuerdo a la estructura externa y van a ser procesados para convertirse en información.



Conceptos Básicos de Base de datos

- Archivo: son conjuntos de registros.
- Registros: son conjuntos de campos.
- Campos: es la mínima unidad de referencia.

Archivo Clientes

Juan Pérez	20	\$500	← registro
Ana Alban	28	\$1000	
José Mora	30	\$1500	

↑
campo

XAMPP

Es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. es de plataforma independiente y tiene licencia GNU GPL.

Una de las ventajas de XAMPP es que de una forma muy sencilla y rápida te puedes montar en tu máquina un entorno de desarrollo de cualquier aplicación web que use PHP y base de datos. La configuración por defecto de XAMPP tiene algunas deficiencias de seguridad por lo que no es recomendable usarla como una herramienta para producción, sin embargo con algunas

modificaciones es lo suficientemente seguro para ser usada como servidor de sitios web en internet.

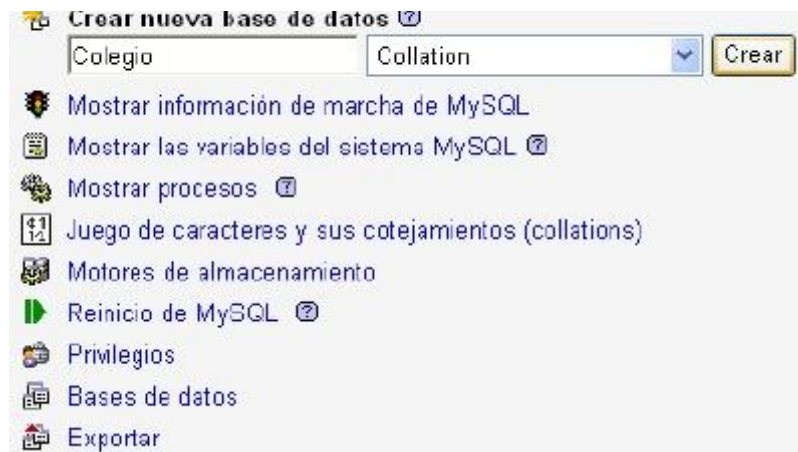
COMUNICACIÓN CON LA BASE DE DATOS

Para comunicarnos con la base de datos, primero nos conectamos con la base de datos (como hemos explicado antes) y después crearemos un objeto Statement con el que realizaremos las consultas a la base de datos. Para realizar una consulta podemos usar, alguno de los siguientes métodos:

- 1.ExecuteUpdate(sentenciaSQL); : Sera usada para los casos en los que usemos instrucciones SQL de manipulación de datos tales como INSERT, DELETE o UPDATE.
- 2.ExecuteQuery(sentenciaSQL); : Sera usada para los casos en los que usemos instrucciones SQL de tipo SELECT. Para este último caso deberemos usar una variable de tipo ResultSet o ResultSetMetaData para almacenar² los valores obtenidos con la consulta.

CREAR UNA BASE DE DATOS

En el apartado Crear nueva....
Ingrese en nombre de su BDD y presiones
Crear



Crear tablas

Servidor: localhost ► Base de datos: colegio ► Tabla: Profesores

Campo	Tipo	Longitud/Valores	Collation	Atributos	Null	Predeterminado
idProfesor	VARCHAR	4			not null	
	VARCHAR				not null	
	VARCHAR				not null	
	VARCHAR				not null	

Comentarios de la tabla:

Tipo de tabla: MyISAM Collation:

Añada campo(s)

* Si el tipo de campo es "enum" o "set", por favor ingrese los valores usando este formato: 'a','b','c'...

** Si alguna vez necesita poner una barra invertida("\") o una comilla simple (") entre esos valores, siempre ponga una barra invertida. (Por ejemplo \'xyz\')

*** Para valores predeterminados, por favor ingrese solamente un valor, sin "escaping backslash" ni comillas, usando este formato: a

**** Por favor ingrese los valores para las opciones de transformación usando este formato: 'a','b','c'...

Si alguna vez requiere insertar un Backslash ("\") o comilla sencilla (") entre esos valores, use backslash (por ejemplo \'xyz\' o 'a\b').

Para una lista de opciones de transformación disponibles y sus transformaciones MIME-type transformations, de clic en [transformation descriptions](#)

Puede hacerlo inmediatamente creada la BDD o encualquiera desde Estructura o del SQL[8]

Ingresa el nombre de la tabla y la cantidad decampos que ud. cree necesitará.

Crear Tabla

2

phpMyAdmin

Base de datos:

Seleccione una base de datos

Servidor: localhost ► Base de datos: colegio

Base de datos Colegio se creó.

consulta SQL:
CREATE DATABASE 'Colegio';

[Editar] [Crear código PHP]

No se han encontrado tablas en la base de datos.

Nombre:

Campos:

Pantalla para la creación de campos y definición de tipos.

ENFOQUES DEL CONTROL DE EXCEPCIONES

Siempre se deben crear controladores de excepción para responder a cualquier circunstancia excepcional, es decir, no prevista en la ejecución de un programa.

El enfoque más práctico para detectar excepciones de base de datos es probar la aplicación en los diferentes escenarios de excepción. Determinar las

excepciones posibles que pueden producirse para una operación del código pertinente y provocar su aparición. Después, se debe examinar los resultados del seguimiento para ver qué excepción se inicia, o examinar la información del error devuelto en el depurador. Esto permite saber qué códigos devueltos se obtendrán para los escenarios de excepción utilizados.

EXCEPCIONES EN LA BASE DE DATOS

EN el paquete **java.sql** Se encuentra la clase **SQLException** Que captura las excepciones ocurridas en el manejo de la base de datos. Su uso no difiere del resto de excepciones, pero incorpora nuevos métodos interesantes:

GetSqlState:. Describe el error según las convenciones xopen.

Getmessage. El método típico de excepciones, salvo que éste recoge el texto que envía el controlador jdbc según lo informado por el gestor de bases de datos, lo que le hace muy efectivo

getErrorCode. Devuelve el código de error ocurrido (según lo informado por el gestor de la base de datos)

getNextException. Que permite ver la siguiente excepción ocurrida, ya que a veces ocurren varias a la vez (especialmente en transacciones y operaciones complejas). Ejemplo:

```
try{  
    //instrucciones de manejo de la base de datos  
}catch(  
    SQLException  
    sqle  
){while(e!=null){  
    System.err.println("Estado:"+e.getSQLState());System.err.println("Código  
: "+e.getErrorCode());System.err.println("Mensaje:  
"+e.getMessage());e.getNextException();}}
```

SQLWarning

Otras veces, ocurre que la base de datos provoca excepciones, sino advertencias(

warnings

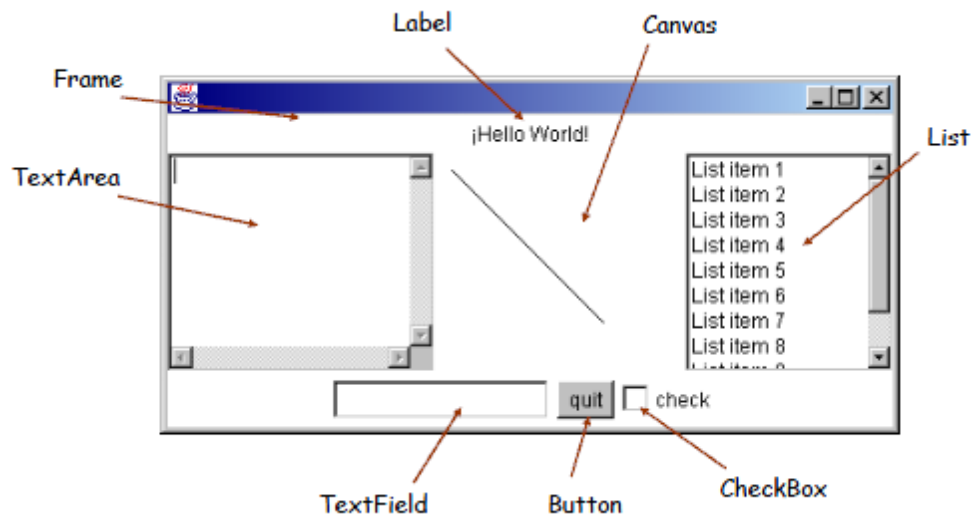
)

Los objetos que las provocan (sobre todo los ResultSet) las van almacenando en objetos de tipo SQLWarning. Para ver los errores basta con llamar repetidamente al método getSQLWarning. En cada llamada obtendremos un nuevo objeto SQLWarning hasta que, finalmente no haya más (devolverá entonces el valor null). Los métodos que permiten observar el contenido de la advertencia son los mismos que los de la clase SQLException, ya que esta clase es heredera suya. Añade el método getNextWarning que funciona de forma similar a getNextException², pero que en esta clase no se suele utilizar ya que las sucesivas llamadas al método getSQLWarnings provocan el mismo resultado. El método clearWarnings de la clase ResultSet permite borrar las advertencias almacenadas hasta ese momento.

INTERFAZ GRAFICA EN JAVA

La interfaz de usuario es la parte del programa que permite al usuario interactuar con él. La API de Java proporciona una biblioteca de clases para el desarrollo de Interfaces gráficas de usuario (en realidad son dos). La biblioteca proporciona un conjunto de herramientas para la construcción de interfaces gráficas que tienen una apariencia y se comportan de forma semejante en todas las plataformas en las que se ejecuten. La estructura básica de la biblioteca gira en torno a componentes y contenedores. Los contenedores contienen componentes y son componentes a su vez, de forma que los eventos pueden tratarse tanto en contenedores como en componentes. La API está constituida por clases, interfaces y derivaciones. AWT y Swing[9].

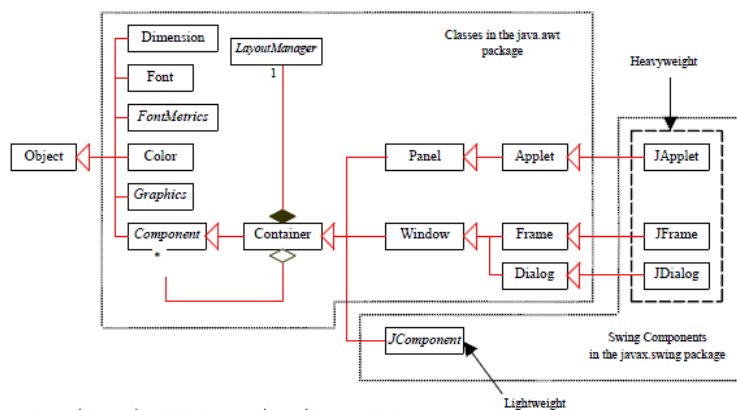
COMPONENTES DE AWT



SWING 2

Paquete de Java para la generación del GUI en aplicaciones reales de gran tamaño. Disponible como paquete externo en Java 1.1 e integrado desde Java 1.2. Es una de las API de JFC (Java Foundation Classes): AWT, Java 2D, Accessibility, Drag and Drop, Swing, ETC. Escrito totalmente en Java. No reemplaza a AWT. Se apoya sobre AWT y añade JComponents. Utiliza el modelo de eventos de Java 1.1. Elección entre diferentes aspectos (look & feel). Arquitectura Model-View-Controller (MVC). Nuevos componentes (árboles, tablas, frames internos, iconos, bordes, tooltips, beans, etcétera).

JERARQUIA DE CLASES



Las clases de AWT reemplazadas por Swing se llaman igual pero con una J delante.

CONTENEDORES

Contenedores de alto nivel:

JFrame: Habitualmente la clase JFrame se emplea para crear la ventana principal de una aplicación en Swing.

JDialog: Ventanas de interacción con el usuario.

Contenedores intermedios:

JPanel: Agrupa a otros componentes.

JScrollPane: Incluye barras de desplazamiento.

ESQUEMA DE APLICACIÓN EN SWING

```
import javax.swing.*;

public class Gui00 extends JFrame {
    // Constantes y componentes (objetos)

    public Gui00(){
        super("Ejemplo 00");

        // Configurar Componentes ;
        // Configurar Manejadores Eventos ;

        setVisible(true);

        setDefaultCloseOperation(EXIT_ON_CLOSE);

    } // Terminar la aplicación al cerrar la ventana.

    public static void main(String args[]){
        Gui00 aplicacion = new Gui00();
    }
}
```

INTERACCIÓN CON EL USUARIO

Al interactuar con la aplicación, el usuario:

- ✓ Acciona componentes (ActionEvent).
- ✓ El usuario pulsa un botón.
- ✓ El usuario termina de introducir un texto en un campo y presiona Intro.

- ✓ El usuario selecciona un elemento de una lista pulsando el preferido (o de un menú).
- ✓ Pulsa o suelta botones del ratón (MouseEvent).
- ✓ Minimiza, cierra o manipula una ventana (WindowEvent).
- ✓ Escribe con el teclado (KeyEvent).
- ✓ Descubre porciones de ventanas (PaintEvent).

Cuando el usuario de un programa o applet mueve el ratón, presiona un pulsador o pulsa una tecla, genera un evento (actionEvent).

Los eventos son objetos de ciertas clases. Normalmente un objeto de alguna subclase de EventObject que indica:

- ✓ El elemento que accionó el usuario.
- ✓ La identificación del evento que indica la naturaleza del evento.
- ✓ La posición del ratón en el momento de la interacción.
- ✓ Teclas adicionales pulsadas por el usuario, como la tecla. Control, la tecla de Cambio a mayúsculas, etcétera [10] .

METODOLOGIA DE LA PROGRAMACIÓN

La metodología de programación a usarse es la XP, ya que la misma se basa en los deseos del cliente, permite una programación organizada y como se va a desarrollar en un proyecto a corto plazo es muy recomendable este tipo de metodología. Además como este tipo de programación está sujeta a las prioridades del cliente es flexible a cambios y pruebas para la finalización de la aplicación.

Las fases que define XP son 6:

- Exploración
- Planificación de la entrega
- Iteraciones
- Producción
- Mantenimiento
- Muerte del proyecto

Dentro de nuestra aplicación estas fases cumplirán las siguientes acciones:

EXPLORACION

- Historias de Usuario: Dentro de esta fase se encontrara las historias de usuario en donde el cliente será parte importante del equipo de desarrollo de esta aplicación. Entre las historias de usuario que se han tomado en cuenta esta las siguientes según las recomendaciones que se hace acerca de ellas.
 1. Funcionamiento de la aplicación
 2. Registro de clientes
 3. Registro de mercaderías
 4. Generación de factura

Estas son en sí 4 historias de usuario que se usaran para el desarrollo del presente proyecto.

Historia de Usuario	
Número:	Nombre:
Usuario:	
Modificación de Historia Número:	Iteración Asignada:
Prioridad en Negocio: (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alto / Medio / Bajo)	Puntos Reales:
Descripción:	
Observaciones:	

- **Herramientas:** Entre las herramientas que se usaron para el desarrollo del presente proyecto están las siguientes:

Java.- Se uso java como lenguaje de programación, esto es debido a que en el caso de este proyecto y un poco contradictorio a lo que dicta la metodolgia XP, la aplicación va a ser desarrollada por una persona con conocimientos en este lenguaje de programación, esto podrá satisfacer de la mejor manera los requisitos del cliente ya que si se realizara el programa en otro lenguaje esto tendría repercusiones en el tiempo de entrega y desarrollo del presente proyecto.

Netbeans.- Netbeans es un IDE de licenciamiento libre que sirve para el desarrollo de una variedad de aplicaciones Java ya sea estas para celular, páginas web y aplicaciones de escritorio.

Se opto por elegir este IDE debido a que cuenta con un asistente para la aplicación de interfaces graficas el cual es de muy fácil manejo y ayudará a reducir y optimizar el desarrollo del proyecto.

MySQL.- Es un sistema de gestión de base de datos relacional, multihilo y multiusuario, además de poseer versiones gratuitas las cuales no ocasionaran gastos extras al proyecto.

PLANIFICACION DE LA ENTREGA

A partir de la planificación se encontrara la estructura básica del proyecto según algunos de los aspectos de la metodología empleada, esto es debido a que en algunas ocasiones no se podrán cumplir todas debido a que la aplicación será desarrollada por un solo programador y el cliente.

jueves 5 abril	PARTE TEORICA: Primera entrega introducción – limitaciones
Jueves 12 abril	Segunda entrega: metodología - prototipos
Jueves 19 – Jueves 3 de mayo	Programación y elaboración de interfaz
Viernes 4 de mayo – Viernes 19 de mayo	Programación base de datos y codificación
Lunes 21 de mayo – Miércoles 23 de mayo	Pruebas y finalización del proyecto

ITERACIONES

Las iteraciones pueden dividirse según las historias de usuario o los módulos que se van a implementar para la misma. Para tener una mejor idea de lo que se trata este proceso dentro de la metodología XP entonces se hará un repaso de lo que se va a cumplir en cuanto a las historias de usuario y los modulo que se van a implementar para cumplir con las mismas.

Historias de usuario

- Funcionamiento de la aplicación
- Registro de clientes
- Registro de mercaderías
- Generación de factura

Historia de Usuario	
Número: 1	Nombre: Funcionamiento de la aplicación
Usuario: Usuario final	
Modificación de Historia Número:	Iteración Asignada:
Prioridad en Negocio: (Alta / Media / Baja) Alta	Puntos Estimados: 5/5
Riesgo en Desarrollo: (Alto / Medio / Bajo) Medio	Puntos Reales: 5
Descripción: El funcionamiento de la aplicación debe ser sencillo y fácil de manejar que brinde las opciones principales para generar la factura.	
Observaciones:	

Historia de Usuario	
Número: 2	Nombre: Registro de clientes
Usuario: Usuario final	
Modificación de Historia Número:	Iteración Asignada:
Prioridad en Negocio: (Alta / Media / Baja) Media	Puntos Estimados: 3/5
Riesgo en Desarrollo: (Alto / Medio / Bajo) Bajo	Puntos Reales: 4
Descripción: En el registro de clientes debe almacenar todos los datos dentro de la base presentarlos al momento de facturar, y buscarlos mediante el número de cedula.	
Observaciones:	

Historia de Usuario	
Número: 3	Nombre: Registro de mercaderías
Usuario: Usuario final	
Modificación de Historia Número:	Iteración Asignada:
Prioridad en Negocio: (Alta / Media / Baja) Media	Puntos Estimados: 3/5
Riesgo en Desarrollo: (Alto / Medio / Bajo) Media	Puntos Reales: 4
Descripción: Es necesario dentro del registro de la mercadería clasificarla mediante un código, que se pueda ingresar la cantidad el producto que se compra y asignar el código manualmente.	
Observaciones:	

Historia de Usuario	
Número: 4	Nombre: Generación de factura
Usuario: Usuario final	
Modificación de Historia Número:	Iteración Asignada:
Prioridad en Negocio: (Alta / Media / Baja) Alta	Puntos Estimados: 5/5
Riesgo en Desarrollo: (Alto / Medio / Bajo) Media	Puntos Reales: 4
Descripción: En el proceso de facturación se requiere llenar los campos necesarios con los datos del cliente y los productos adquiridos. Realizar los cálculos correspondientes y obtener resultados.	
Observaciones:	

Interfaces.

- Interfaces de Opciones de la aplicación
- Interfaz de Cliente
 - Interfaz de Ingreso de nuevo Cliente
 - Interfaz Buscar Cliente
 - Interfaz Ver lista
- Interfaz de Mercadería
 - Interfaz Agregar
 - Interfaz Buscar
 - Interfaz Ver lista
- Interfaz Factura
- Interfaz Ayuda

Debido a que son más interfaces a implementar que historias de usuario entonces cada iteración que se haga se la realizará en base a las interfaces más no a las historias de usuario.

PRODUCCIÓN

Debido a que la mayor parte de la aplicación ya fue desarrollada en la fase anterior en esta fase nos dedicaremos a la realización de pruebas las cuales serán de mucha importancia antes de que la aplicación vaya al medio en donde el cliente la va a utilizar, la realización de pruebas en esta fase nos ayuda a ver si es que existiera que hacer alguna mejora en la versión que se le está presentando al cliente; en esta fase también se notara que el tiempo de cada una de las iteraciones será más corto que antes.

MANTENIMIENTO

Durante esta fase se debe tener en cuenta que el sistema ya tiene que estar en funcionamiento; aunque el sistema se encuentre en funcionamiento no significa que el desarrollo del mismo haya terminado, entonces se siguen desarrollando las historias de usuario restantes a manera de soporte técnico de la aplicación, en esta fase es posible que se necesite ayuda adicional para poderla llevar a cabo.

ARQUITECTURA DE LA APLICACIÓN



USUARIO



APLICACIÓN



BASE DE DATOS

PROTOTIPO DE LA APLICACIÓN



Fig 1. Pantalla de inicio



Fig 2. Pantalla de opciones

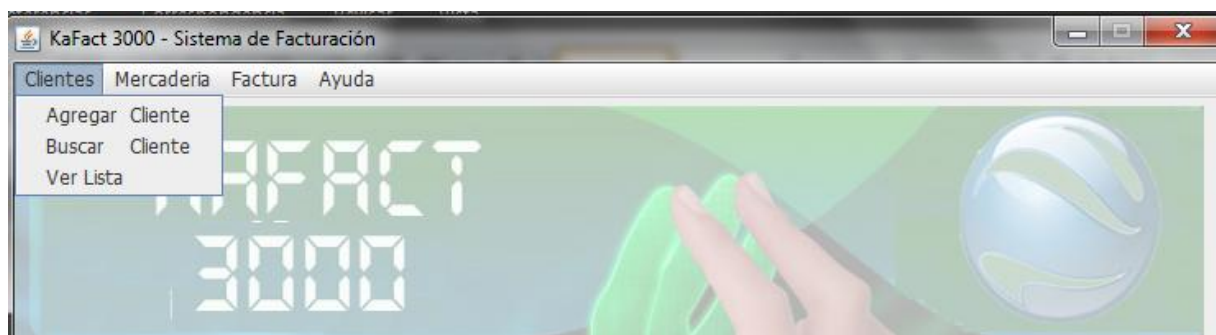


Fig 3. Opciones menú clientes

Fig 3.1 Opción agregar cliente

ID	NOMBRES	APELLIDOS	CEDULA	DIRECCION	TELEFONO
1	luis	mena	1100000000	loja	25757911
2	paul	jose	1102020202	loja	2596565
3	Maria	Lopez	1104811300	quito	2575911
4	cesar	carchi	1102131917	atamer	2577470
5	Katherine	Carchi	1104811301	atamer	0699859304
6	katty	perez	1102154879	loja	2575954
7	Nelson	Pupiales	1003075624	Mercadillo y Su...	090234997

Fig 3.2 Opción agregar cliente

ID	NOMBRES	APELLIDOS	CEDULA	DIRECCION	TELEFONO
1	luis	mena	1100000000	loja	25757911
2	paul	jose	1102020202	loja	2596565
3	Maria	Lopez	1104811300	quito	2575911
4	cesar	carchi	1102131917	atamer	2577470
5	Katherine	Carchi	1104811301	atamer	0699859304
6	katty	perez	1102154879	loja	2575954
7	Nelson	Pupiales	1003075624	Mercadillo y ...	090234997

Fig 3.3 Opción ver lista clientes



Fig 4. Opciones menú mercaderías

codigo	producto	cantidad	costo
001	Galletas	20	0.4
002	colombinas	50	0.25
003	Caramelos	100	0.05
004	leche	1	10
005	chitos	10	0.5
006	gomitas	100	0.25

Fig 4.1 Opción agregar producto

CODIGO	PRODUCTO	CANTIDAD	COSTO
001	Galletas	20	0.4
002	colombinas	50	0.25
003	Caramelos	100	0.05
004	leche	1	10
005	chitos	10	0.5
006	gomitas	100	0.25

Fig 4.2 Buscar producto

CODIGO	PRODUCTO	CANTIDAD	COSTO
001	Galletas	20	0.4
002	colombinas	50	0.25
003	Caramelos	100	0.05
004	leche	1	10
005	chitos	10	0.5
006	gomitas	100	0.25

Fig 4.3 Ver lista de productos



Fig 5. Opción generar factura

FACTURA

Datos de la Factura

FACTURA Nº : 0001 FECHA EMISION: 23/05/2012

CEDULA / RUC :

NOMBRES : TELÉFONO:

DIRECCIÓN :

Detalles de la Compra

Codigo	Producto	Cantidad	P. Unitario	P. Total

Subtotal :

Impuesto IVA 12%:

Total a Pagar :

Fig 5.1 Proceso de creación de factura



Fig 6 Opción de ayuda

DESCRIPCION DE LA APLICACION

KAFAC 3000 es un facturador que se compone de 10 ventanas y una base de datos para el manejo de los datos; en las 10 ventanas que comprende la aplicación se resolverá los requerimientos que fueron expresados por el usuario en las historias de usuario.

Ahora veremos el funcionamiento de cada una de las ventanas o interfaces de usuario.

Primeramente se presentara una pantalla de inicio con un botón con la leyenda **Comenzar** para poder entrar a la aplicación. Seguidamente de eso se tendrá la ventana principal de la aplicación en donde se tendrá una barra de menús con las opciones para cada uno de los elementos que comprende una factura es decir **Cliente, Producto, Factura, Ayuda**; cada uno de los menús tendrá a su vez submenús con otras opciones para el manejo de los datos.

Menú Cliente

En el menú cliente se tendrán las opciones de **Agregar Cliente, Buscar Cliente y ver Lista**, cada una de ellas mostrara una nueva ventana para el manejo de los datos.

Menú Mercadería

En el menú producto se encontrara las opciones para **Agregar Producto, Buscar Producto, ver Lista**; los cuales permitirán manejar los datos contenidos en la base de datos pero en esta ocasión de los productos.

Menú Factura

En el menú factura se encontrara únicamente la opción de **Generar**, la cual nos permitirá mostrar la ventana para la realización de la factura.

Menú Ayuda

En el menú ayuda se encontrara únicamente la opción en donde se mostrara las opciones de ayuda de la aplicación.

Hasta aquí el análisis de los que comprende los menús de la ventana principal, ahora se analizara las ventanas que presenta el primer menú es decir el menú de Clientes que son los siguientes:

- **Agregar Clientes**

En la ventana de Agregar Clientes se mostrara diferentes campos en los cuales el usuario ingresara la información del cliente según los identificativos de cada uno de los campos, se tendrá un botón para guardar los datos y otro para cancelar el ingreso del Cliente.

- **Buscar Cliente**

En la ventana de Agregar Cliente se encontrara un campo en donde se ingresara el numero de cedula del cliente que se desea buscar, además de un botón de buscar que realizara la búsqueda del cliente en la base de datos, los datos se presentaran en una tabla que se encuentra debajo de los botones de buscar, cancelar y del campo de ingreso de datos.

- **Lista de Clientes**

En esta ventana se presentara la lista de todos los clientes que se encuentran en la base de datos; es decir de todos los clientes a los cuales se les ha emitido una factura por lo menos una vez.

En el menú de Mercadería en cambio se tendrá las siguientes opciones o submenús.

- **Agregar Producto**

En esta ventana se podrá observar 2 secciones en la sección de la parte superior tenemos los datos del producto a ingresar como son: código, nombre del producto, costo y cantidad; en la sección de la parte inferior se mostrara en cambio los productos que ya se encuentran ingresados.

- **Buscar Producto**

En esta ventana se encontrara 2 secciones también en la parte superior se encontrara un campo para el ingreso del código del producto y en la otra en

cambio se podrá ingresar el nombre del producto y en la sección inferior se mostrara el resultado de la búsqueda; una vez que se haya pulsado el botón de Buscar, también se tendrá en la sección superior una botón de cancelar para salir de esta ventana.

- **Lista Productos**

En esta ventana se mostrara una tabla con toda la información de los productos que se han registrado, además tendrá un botón de Salir para poder regresar a la ventana anterior.

En el menú de factura solo se tendrá solo la opción de generar factura la cual lanzara las siguientes ventanas:

- **Factura**

En esta ventana también se tendrá 2 secciones en donde en la primera sección se encontrara los datos de la factura, también se encontrara un botón para Agregar el cual buscara el numero de cedula dentro de la base de datos y mostrara los datos obtenidos si el usuario ya se encuentra registrado, también se encontrara el botón de Nuevo para registrar un nuevo cliente, otro de los botones que se encontrara en esta primera sección es el botón de cancelar para regresar a la pantalla principal.

En la segunda sección se encontrara en cambio los comandos necesarios para el ingreso de los producto que van a ser vendidos, se tendrá una tabla para mostrar los productos a vender, junto a ella se encontraran 3 botones de los cuales el primero servirá para ingresar un producto a la tabla mediante el ingreso del código del producto y la cantidad, el segundo nos permitirá buscar el producto mediante una consulta a la base de datos y el tercero nos permitirá grabar los datos principales de la factura en la base de datos. Se mostrara en los campos correspondientes los valores de la venta incluido el valor del impuesto según el porcentaje programado.

CONCLUSIONES

Un sistema de facturación dentro de una empresa juega un papel muy importante ya que agiliza el proceso pero según lo desarrollado es necesario tener un contacto más relacionado con el cliente para especificar lo que se desea obtener y así mejorar nuestro producto.

Debido a la falta de conocimiento con respecto a la metodología de la programación el desarrollo de la aplicación tardó más de lo esperado con respecto a ese tema.

El uso de una base de datos facilita la forma de trabajo agrega los componentes que necesitamos y en donde los necesitamos. Se puede obtener la información de manera directa y concreta.

RECOMENDACIONES

Es muy importante tener un conocimiento profundo sobre lo que se va a llevar a cabo, conocer sobre las herramientas que se van a aplicar y la manera a ejecutarse para así mejorar el desarrollo e implementación del programa.

Presentar de forma clara, concreta y precisa toda la información para así optimizar la implementación en el desarrollo del proyecto.

Utilizar nuestros conocimientos y realizar investigación referente a la aplicación para así crecer en conocimientos y poder impartirlos hacia los demás.

Buscar herramientas que favorezcan en el desarrollo de la metodología para así facilitar la comprensión y agilizar el desarrollo.

BIBLIOGRAFIA

- [1] http://es.wikipedia.org/wiki/Empresa_comercial
- [2] <http://www.monografias.com/trabajos25/gestion-administrativa/gestion-administrativa.shtml>
- [3] <http://es.wikipedia.org/wiki/Factura>
- [4] <http://www.monografias.com/trabajos11/empre/empre.shtml>
- [5] http://java.ciberaula.com/articulo/que_es_java
- [6] <http://es.wikipedia.org/wiki/NetBeans>
- [7] <http://www.monografias.com/trabajos34/base-de-datos/base-de-datos.shtml#base>
- [8] <http://es.scribd.com/doc/17040093/XAMPP-Trabajo-Con-MySQL>
- [9] DEITEL, Paul J. y Harvey M. Cómo programar en Java. Séptima Edición (2008)
- [10] <http://www.fdi.ucm.es/profesor/jpavon/poo/Tema6resumido.pdf>

Tabla de contenido

TEMA:	i
OBJETIVOS	ii
General:	ii
Específicos:.....	ii
JUSTIFICACIÓN:	iii
ALCANCE	iv
LIMITACIONES	vi
MARCO TEORICO	1
METODOLOGIA DE LA PROGRAMACIÓN	12
Exploracion	12
planificacion de la entrega	14
iteraciones.....	14
producción	17
mantenimiento	18
ARQUITECTURA DE LA APLICACIÓN	19
PROTOTIPO DE LA APLICACIÓN	20
DESCRIPCION DE LA APLICACION	25
CONCLUSIONES	28
RECOMENDACIONES	29