

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS

DISERTACION PREVIA A LA OBTENCION DEL TÍTULO DE
INGENIERO EN SISTEMAS Y COMPUTACIÓN

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
FACTURACIÓN PARA EL RESTAURANTE LA FOGATA”

NOMBRE:
OSVALDO RAÚL TAPIA TATÉS

DIRECTORA:
MGTR. MASAPANTA SUSANA

QUITO, 2015

Tabla de contenido

CAPITULO 1. INTRODUCCIÓN	1
1.1 Metodología De Programación Rational Unified Process (RUP)	1
1.1.1 Introducción.-	1
1.1.2 Concepto.-	1
1.1.3 Características.-	2
1.1.4 Fases y ciclo de vida.-	3
1.2 Lenguaje de Programación JAVA	4
1.2.1 Introducción.-	4
1.2.2 Historia del lenguaje de programación Java.-	5
1.2.3 Características Fundamentales.-	6
1.2.4 Ventajas y Desventajas.-	8
1.2.5 Aplicaciones Java.-	9
1.3 Base de datos PostgreSQL	13
1.3.1 Introducción.-	13
1.3.2 Historia de PostgreSQL.-	14
1.3.3 Características Fundamentales.-	14
1.3.4 Ventajas y Desventajas.-	16
1.3.5 Aplicaciones de PostgreSQL.-	17
CAPITULO 2. ANALISIS DE REQUERIMIENTOS	20
2.1 Modelado Del Negocio	20
2.1.1 Situación actual del negocio.-	20
2.1.2 Descripción del sistema propuesto.-	20
2.1.3 Descripción de las necesidades del negocio en el sistema.-	21
2.2 Requerimientos	21
2.2.1 Requerimientos Funcionales	21
2.2.2 Diagrama De Casos de Uso	22
2.2.3 Reportes necesarios.-	57
2.2.4 Requerimientos de hardware y software.-	58
2.2.5 Requerimientos de facturación electrónica.-	58
CAPITULO 3. Diseño	61
3.1 Estructura del producto.-	61
3.1.1 Estructura del producto.-	61
3.2 Diseño de interfaz	62
3.2.1 Prototipo de interfaz.-	62
3.3 Diagrama de Clases	63

3.3.1 Diagramas de Clases.....	63
3.4 Diagrama de Secuencia	68
3.4.1 Diagrama de Secuencias de Ingreso al Sistema.....	68
3.5 Diagrama Entidad Relación	78
3.5.1 Modelo conceptual.-	78
3.5.1 Modelo físico.-	79
3.6 Diccionario de datos.....	80
3.6.1 Lista de dominios.-	80
3.6.2 Diccionario de datos.-	80
CAPITULO 4. Desarrollo del sistema	83
4.1 Desarrollo	83
4.1.1 Definición de clases.-	83
4.1.2 Codificación de clases.-	84
4.1.3 Conexión con la base de datos.-	87
4.2 Pruebas	89
4.2.1 Plan de pruebas del sistema	89
CAPITULO 5. IMPLEMENTACIÓN	96
5.1 Despliegue.....	96
5.1.1 Software empaquetado.-	96
5.1.2 Instalación de Software.-	102
CAPITULO 6. CONCLUSIONES Y RECOMENDACIONES	105
6.1 Conclusiones.-	105
6.2 Recomendaciones.-	106
TABLA DE DIAGRAMAS, TABLAS Y GRAFICAS.....	108
GLOSARIO DE TÉRMINOS.....	112
BIBLIOGRAFÍA.....	114
ANEXOS.....	116

CAPITULO 1. INTRODUCCIÓN

1.1 Metodología De Programación Rational Unified Process (RUP)

1.1.1 Introducción.-

Los procesos de desarrollo de software empezaron a complicarse a medida que las empresas dependían aún más de sus sistemas informáticos, inicialmente no existían metodologías de desarrollo de software, lo que complicaba la elaboración de sistemas, aumentaba sus costos, errores, tiempos de entrega, dificultad de mantenimiento, abandono de proyectos. Como respuesta a los problemas mencionados aparecieron las metodologías de desarrollo de software.

“Existen varios tipos de metodologías que surgieron por la necesidad de poner orden al proceso de construcción del software. Resulta importante que el desarrollo de sistemas informático sea tratado bajo una disciplina ingenieril, con el fin de desarrollar e implementar sistemas optimizando tiempo, costo y que se ajusten a las necesidades del usuario”¹ (Almeida & Fernández, 2009, pág. 9)

El proceso racional unificado (RUP) surge como una alternativa sólida, nos brinda un acercamiento disciplinado a la asignación de responsabilidades y tareas en una organización de desarrollo, el propósito del proceso racional unificado (RUP) es la aseguración de producción de software de alta calidad, que se ajuste a las necesidades de los usuarios finales con costos y tiempos de entrega predecibles.

1.1.2 Concepto.-

Es una metodología de desarrollo de software formal que abarca todos los aspectos a tener en cuenta durante el ciclo de vida del software, utiliza el lenguaje unificado de modelado UML para el análisis, implementación y documentación de sistemas orientados a objetos que pueden abarcar tanto a pequeños como grandes proyectos de software.

¹ Almeida, S., & Fernández, E. (2009). *Aplicación de la metodología RUP en el desarrollo del sistema de manejo de inventario y facturación para la empresa SISCONTAV S.A.* Quito, Ecuador.: Pontificia Universidad Católica del Ecuador.

1.1.3 Características.-

- **Proceso dirigido por casos de uso.-** los casos de uso proporcionan una guía que integra los requisitos, análisis y diseño, implementación y pruebas.
- **Proceso centrado en la arquitectura.-** nos permite estructurar las partes del sistema y que su interacción este muy clara para usuarios y desarrolladores.
- **Proceso interactivo e incremental.-** al estar separado en estructuras se puede ir completando por partes o mini proyectos además de compactarlos para notar el avance del proyecto.
- **Proceso controlado.-** El control se hace mucho más fácil ya que cada uno de los mini procesos pueden ser monitorizados.
- **Soportado por herramientas.-** existen varias herramientas para el desarrollo mediante RUP, aunque la mayoría son pagas como: IBM rational method composer.
- **Fomenta el control de calidad.-** podemos identificar las partes críticas del sistema tempranamente y no al final en la integración del sistema.
- **Distribuye la carga de trabajo.-** cuando se trabaja en equipo, se puede repartir con mayor eficiencia las tareas, lo que reduce costos.
- **Acelera el ritmo de desarrollo.-** debido a que se trabaja con objetivos específicos y tiempos de entrega, se puede reducir tiempo.

1.1.4 Fases y ciclo de vida.-

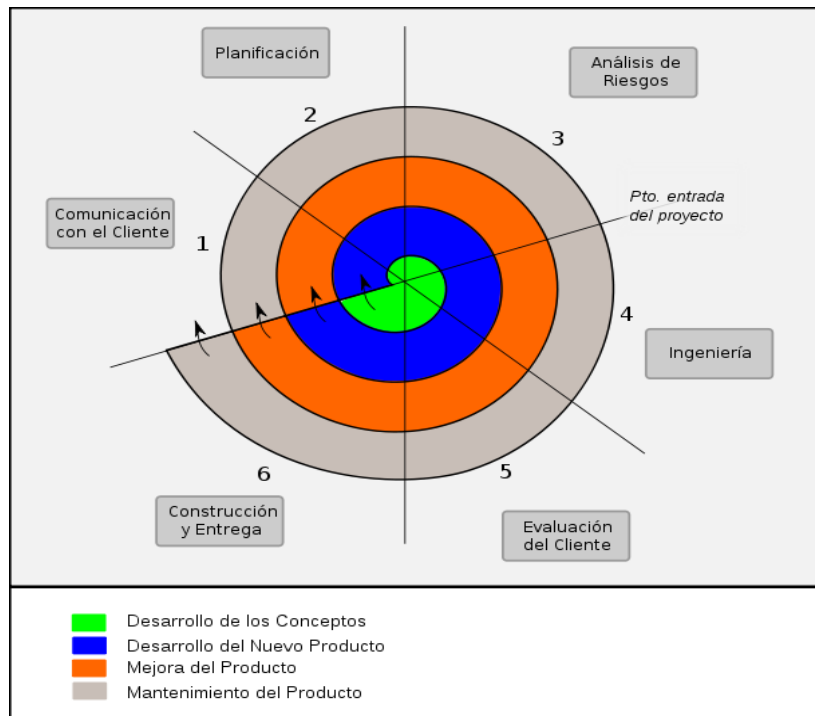


Diagrama 1: Ciclo de vida en espiral.

Fuente: Ebzn (2008) En Wikipedia Commons. Recuperado de https://commons.m.wikimedia.org/wiki/File:Modelo_Espiral_Boehm.svg

El RUP se divide en cuatro fases: Iniciación, elaboración, construcción y transición.

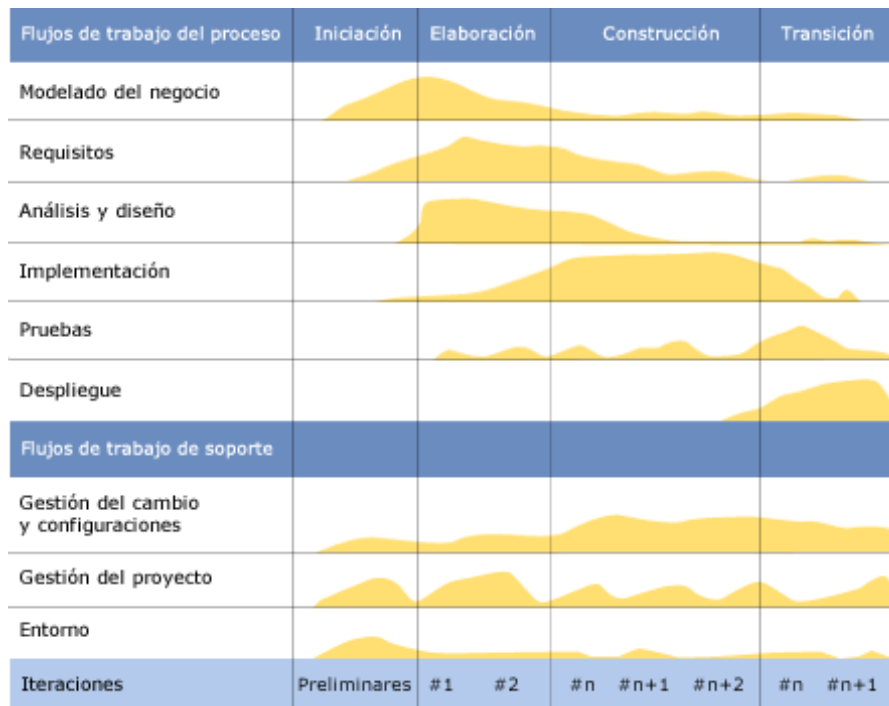


Diagrama 2: Fases del Proceso Racional Unificado (RUP).

Fuente: Booch, G., Rumbaugh J., Jacoboson, I., El proceso Unificado de Desarrollo de Software.

Fase de Iniciación.- en esta fase se define el alcance del proyecto, esto se hace en conjunto con el dueño del proyecto, además se identifican los posibles riesgos y complicaciones que podría acarrear la creación del proyecto, también nos sirve para crear una visión general de la arquitectura de software además de la elaboración de un plan de fases e iteraciones.

Fase de Elaboración.- esta fase nos permite definir la arquitectura con mayor detalle mediante los casos de uso y se desarrolla la solución preliminar del sistema.

Fase de Construcción.- el objetivo de esta fase es terminar la funcionalidad del sistema, además de realizar los cambios necesarios de acuerdo a las evaluaciones realizadas por los usuarios.

Fase de Transición.- en esta etapa completamos el software, es decir que esté listo para el usuario final, en este momento corregimos los errores encontrados en las pruebas de aceptación, capacitamos a los usuarios y verificamos que el producto cumpla con especificaciones definidas inicialmente.

1.2 Lenguaje de Programación JAVA

1.2.1 Introducción.-

Java, es un lenguaje de programación orientado a objetos, diseñado para ser multiplataforma, es decir que el mismo programa puede ser empleado en diversos sistemas operativos. Todo programa en Java debe compilarse y este programa compilado es interpretado por una máquina virtual.

Java es distinto de los demás lenguajes de programación. Está diseñado para facilitar el trabajo en la red mundial de computadores, utilizando programas de uso común llamados navegadores, esto significa que se pueden revisar documentos distribuidos por todo el mundo y activar un programa Java que llegara a nosotros a través de la red y se ejecutara en nuestro computador, el usuario ni siquiera debería enterarse que está usando Java.² (Bishop, 1999)

² Bishop, J. (1999). *JAVA Fundamentos de programación, 2a Ed.* Madrid: Addison Wesley.

Es un lenguaje robusto ya que cuenta con el garbage collector que nos sirve para la recuperación de memoria con componentes que no están siendo utilizados, para minimizar los errores el lenguaje java elimina la creación de punteros, y los reemplaza con objetos, otra de las mejores que son implementadas por java es que nos obliga a corregir o prever las posibles excepciones, finalmente Java posee un gestor de seguridad para restringir el acceso al sistema.

1.2.2 Historia del lenguaje de programación Java.-

Durante la década de 1960 se creó un lenguaje de programación llamado Algol 60, que proviene de algoritmo. Este lenguaje fue popular entre los círculos académicos. En el Reino Unido se creó una versión extendida de Algol 60 el BCPL (Lenguaje de programación combinado básico).

En la década de los 70 en los Laboratorios Bell transformaron el BCPL en un lenguaje llamado B, que posteriormente se mejoró para convertirse en el C que llegó a ser muy popular e inclusive fue utilizado para escribir una gran parte de Unix. El siguiente avance fue cuando Bjarne Stroustrup creó C en el transcurso de los años 80 también en laboratorios Bell, este lenguaje hizo la posible creación y reutilización de secciones independientes de código, lo que dio paso a la programación orientada a objetos³ (Bell & Parr, 2011, pág. 1)

En los primeros años de la década de los noventa, Sun Microsystems decidió intentar introducirse en el mercado de la electrónica de consumo y desarrollar programas para pequeños dispositivos electrónicos. Tras unos comienzos dudosos, Sun decidió crear una filial, denominada FirstPerson Inc., para dar margen de maniobra al equipo responsable del proyecto.

La reducida potencia de cálculo de memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar código de tamaño muy reducido, finalmente Java fue diseñado por James Gosling y originalmente se llamó Oak (roble).

Debido a la existencia de distintos tipos de CPUs y a los continuos cambios, era importante conseguir una herramienta independiente del tipo de CPU utilizada.

³ Bell, D., & Parr, M. (2011). *Java para estudiantes 6ta Ed.* Naucalpan de Juárez, Mexico: Pearson Educación.

Desarrollaron un código “neutro” que no dependía del tipo de electrodoméstico, el cual se ejecutaba sobre una “maquina hipotética o virtual” denominada Java Virtual Machine (JVM). Era la JVM quien interpretaba el código neutro convirtiéndolo a código particular de la CPU utilizada. Esto permitía lo que luego se ha convertido en el principal lema del lenguaje: “White Once, Run everywhere”. A pesar de los esfuerzos realizados por sus creadores, ninguna empresa de electrodomésticos se interesó por el nuevo lenguaje.

Como lenguaje de programación para computadores, Java se introdujo a finales de 1995. La clave fue la incorporación de un intérprete Java en la versión 2.0 del programa Netscape Navigator, produciendo una verdadera revolución en internet. Java 1.0 oficialmente es presentado en Enero de 1996. Java 1.1 apareció a principios de 1997, mejorando sustancialmente la primera versión del lenguaje. Java 1.2, más tarde se rebautizado como Java 2, nació a finales de 1998.⁴ (Coronel, 2010, pág. 25)

En la actualidad se encuentra en completo funcionamiento la versión número 8 de Java que fue lanzada el 18 de marzo de 2014, además que se prevén actualizaciones de parches críticos que se realizarían: el 14 de julio de 2015, 20 de octubre de 2015 y 19 de enero de 2016, según la página web oficial de Oracle.

1.2.3 Características Fundamentales.-

- **Lenguaje Simple.-** está basado en C++ pero elimina algunas características que complican la programación como:
 - No soporta Struct, union o pointer.
 - No ofrece typedef ni #define.
 - No soporta herencia múltiple.
 - Tiene una clase String que permite un mejor manejo de arrays de terminación nula.
 - Tiene un sistema automático de asignación y liberación de memoria.

⁴ Coronel, E. (2010). *Lenguaje de programación java*. Lima, Perú: Empresa Editora Macro.

- **Orientado a objetos.-** da buen soporte a las técnicas de desarrollo de orientación a objetos, agrupa las estructuras y las encapsula facilitando su manipulación.
- **Distribuido.-** fue diseñado para trabajar en conexión de red o clientes remotos, para ello contiene un biblioteca de clases para la utilización de protocolo TCP/IP, HTTP, FTP, etc. Facilitando la programación de aplicaciones en red.
- **Interpretado.-** El compilador de Java traduce los ficheros fuente a un código de bytes que puede ser interpretado por todas las maquinas que den soporte a un visualizado que función con java, es decir que es independiente de su sistema operativo.
- **Robusto.-** Es muy fiable en comparación con C, ya que se han eliminado punteros además que proporciona varias comprobaciones en compilación y en tiempo de ejecución.
- **Seguro.-** Ya que java fue creado para funcionar en red, se han creado barreras de seguridad en el lenguaje y en el sistema, ejecutados en tiempo real, además se está trabajando en la encriptación del código.
- **Arquitectura Neutral.-** “El compilador crea los códigos de byte que envía al intérprete de Java o a un visualizador que funcione con java, es decir que no importa en que se esté corriendo el programa de java.” ⁵ (Valladolid, 2015)
- **Portable.-** Al ser arquitectura neutral lo convierte automáticamente en portable haciendo que sus programas funcionen igual en cualquier plataforma.
- **Alto Desempeño.-** “Al ser compilado e interpretado en tiempo real, el código se transforma en una especie de código de máquina.” ⁶ (Valladolid, 2015)
- **Multihilos.-** Java puede aplicarse a la realización de aplicaciones en las que ocurra más de una cosa a la vez, apoyándose en un sistema de gestión de eventos permite apoyar la conducta en tiempo real e interactivo en programas.

⁵ Valladolid, U. d. (02 de 12 de 2015). <http://www.infor.uva.es>. Obtenido de <http://www.infor.uva.es: http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html>

⁶ Valladolid, U. d. (02 de 12 de 2015). <http://www.infor.uva.es>. Obtenido de <http://www.infor.uva.es: http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html>

- **Dinámico.-** Utiliza un sistema de interfaces que permite aligerar la utilización de clases, es decir que las crea según vayan siendo usadas y puede eliminarlas de memoria sin que sea necesario destruirlas en código.
- **Produce Applets.-** estos son programas independientes y se comportan como cualquier aplicación y pueden ser tan complejos como otros programas escritos en diferentes lenguajes.

1.2.4 Ventajas y Desventajas.-

Puede haber varias formas de comparar las ventajas y desventajas de un lenguaje de programación, la primera será en general, es decir ventajas como lenguaje de programación y luego realizaremos una comparación con respecto a C y C++.

1.2.4.1 Ventajas.-

1. **Portabilidad.-** no será necesario reescribir el código si lo queremos ocupar en otra máquina.
2. **POO.-** al ser orientado a objetos ofrece todos los beneficios de esta metodología: abstracción, encapsulamiento, polimorfismo, herencia.
3. **Lenguaje de programación.-** posee todas las características de cualquier lenguaje de programación por lo que soporta: graficas, animaciones, sonidos, bases de datos, etc.
4. **Plugins.-** ahorra la instalación de plugins a los usuarios, porque cualquier navegador de internet, soporta java.
5. **JDK.-** el software de desarrollo es una herramienta libre, es decir sin costo.
6. **Base de datos.-** el manejo de base de datos es uniforme.
7. **Conocimiento.-** el conocimiento de programación del lenguaje java es latamente cotizado.

1.2.4.2 Desventajas.-

1. **Velocidad.-** al realizar aplicaciones grandes puede disminuir la velocidad de ejecución.
2. **Soporte.-** no ofrece mucha ayuda con respecto al soporte técnico.
3. **Conexión datos.-** puede que no existan clases para la conexión a base de datos poco conocidas.

4. Diseño.- para el diseño web y java no existen mucha ventaja con respecto a los entornos gráficos de diseño.

1.2.4.3 Ventajas y desventajas en comparación con C y C++.-

Característica	C	C++	Java
Expresividad	Regular	Muy Buena	Muy Buena
Bien Definido	Regular	Muy Buena	Muy Buena
Tipos y Estructuras de datos	Deficiente	Muy Buena	Muy Buena
Modularidad	Regular	Muy Buena	Muy Buena
Facilidades de entrada/salida	Buena	Buena	Buena
Portabilidad	Buena	Buena	Excelente
Eficiencia / Performance	Excelente	Excelente	Buena
Pedagogía	Regular	Regular	Buena
Generalidad	Buena	Muy Buena	Muy Buena
Estandarización	Buena	Buena	Excelente
Evolución	Estable	Estable	Acelerada
Soporte de Librerías	Bueno	Muy Bueno	Excelente

Tabla 1: Tabla comparativa C, C++ y Java

Fuente: Americati, Recuperado el 12/05/2015 de https://www.americati.com/doc/ventajas_c/ventajas_c.html#_referencias

1.2.5 Aplicaciones Java.-

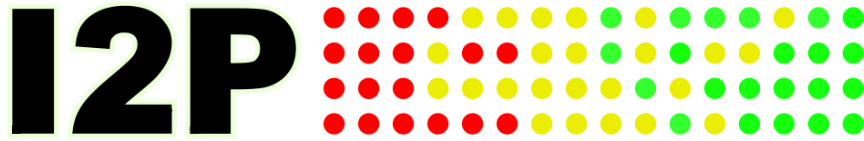
- FreeMind.- es una herramienta de software libre que permite la creación de mapas mentales o conceptuales.



Grafica 1: Logo FreeMind

Fuente: Dyvim (2007) En Wikipedia Commons. Recuperado de http://freemind.sourceforge.net/wiki/images/6/64/Freemind_logo_by_dyvim.png

- I2P.- es una capa de red que nos permite navegar de forma segura y totalmente anónima.



Grafica 2: Logo I2P

Fuente: Lichtspiel (2011) En Freemind.sourceforge.net. Recuperado de http://upload.wikimedia.org/wikipedia/commons/a/ae/I2P_logo.svg

- HeliosPaint.- es un clon de Paint pero agrega opciones para hacer retoque fotográfico además de poder exportar a fichero PDF.



Grafica 3: Logo HeliosPaint

Fuente: Heliospaint. Recuperado el 12/05/2015 de <http://www.heliospaint.com/HeliosPaint.png>

- Jajuk.- es un programa libre que, precisamente se dedica a catalogar y organizar tu colección musical. El manejo es sencillo y la potencia es enorme



Grafica 4: Logo Jajuk

Fuente: Bflorat (2007).En Wikipedia.org. Recuperado de <http://upload.wikimedia.org/wikipedia/en/d/de/Jajuk-icon.png>

- Ants P2P.- es un sistema abierto de cifrado para envío de archivos P2P.



Grafica 5: Logo ANTS P2P

Fuente: Sourceforge.net. Recuperado el 12/05/2015 de <http://sourceforge.net/projects/antsp2p>

- BLUEJ.- es un entorno de desarrollo integrado para el lenguaje de programación Java, desarrollado principalmente con propósitos educacionales.



Grafica 6: Logo BlueJ

Fuente: Bluej.org. Recuperado el 12/05/2015 de <http://www.bluej.org/bluej-icon-256.png>

- Buho21.- es una aplicación que se basa en juegos humano, esta pequeña pero poderosa aplicación, es capaz de proporcionar las herramientas para poder jugar con personas de otros países.



Grafica 7: Logo Buho21

Fuente: Buho21. Recuperado el 12/05/2015 de <http://www.buho21.com/img/click.gif>

- JDownloader.- es un gestor de descargas de código abierto, que permite la descarga automática de archivos de sitios de alojamiento inmediato como Mediafire, Rapidshare, MEGA, entre otros.



Grafica 8: Logo JDownloader

Fuente: Jdownloader.org. Recuperado el 12/05/2015 de <http://jdownloader.org/lib/tpl/arctic/images/logo.png>

- LimeWire.- fue un cliente peer-to-peer (P2P) libre y gratuito para la red Gnutella, diseñado para la compartición de archivos.



Grafica 9: Logo limewire

Fuente: Enwiki (2009).En Wikipedia.org. Recuperado de http://en.wikipedia.org/wiki/File:Limewire_logo.gif

- Open Office.- es un paquete de programas muy parecido al paquete Office de Microsoft que contiene el Word, Excel, Power Point, etc.



Grafica 10: Logo OpenOffice

Fuente: Openoffice.org. Recuperado el 12/05/2015 de https://www.openoffice.org/images/AOO_logos/AOO4_website_logo.png

1.3 Base de datos PostgreSQL

1.3.1 Introducción.-

En los actuales momentos en los que la única constante es el cambio, aún más en el tema que nos concierne, el aspecto tecnológico. El software libre se esta ha sufrido un gran crecimiento, entre estudiantes, docentes, empresas y gobierno, cada día aumentan aplicaciones basadas en tecnologías libres, y no pueden quedar de lado los sistemas de gestión de base de datos en donde existe una gran alternativa, confiable, robusta, segura y que además cuenta con soporte adecuado, PostgreSQL.

PostgreSQL.- es un sistema de gestión de base de datos objeto-relacional, con su código fuente disponible libremente para la descarga y distribuido bajo licencia BSD (Berkeley Software Distribution), utiliza un modelo cliente/servidor además de mutihilos y multiprocesos que garantizan la estabilidad del sistema, además que posee un control en los procesos, es decir si existe algún fallo en un proceso el sistema continua funcionando.

“PostgreSQL es el más avanzado servidor de bases de datos de código abierto” ⁷ (Momjian, 2001, pág. 15).

“PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.” ⁸ (Cáliz, 2007, pág. 7)

La administración de este proyecto no pertenece a alguna empresa sino a una comunidad de desarrolladores muy organizada, ello se demuestra en los lanzamientos de versiones ya que cada versión demuestra notoriamente las mejoras en rendimiento, capacidad, escalabilidad y sobre todo puedes pasar de una versión a otra sin mayores complicaciones, posee un lenguaje fácil de aprender además de ser multiplataforma.

⁷ Momjian, B. (2001). *POSTGRESQL, Introduction and Concepts. 1ra Ed.* Minneapolis: Pearson Educación.

⁸ Cáliz, D. (2007). *Generador de reportes para postgres.* Quito, Ecuador: Escuela Politécnica nacional.

1.3.2 Historia de PostgreSQL.-

IBM había estado trabajando desde 1973 con los primeros conceptos, ideas y teorías sobre bases de datos relacionales. Su proyecto "System R" fue entre otras cosas la primera implementación del lenguaje SQL (Structured Query Language). Por aquel entonces un profesor de la Universidad de Berkeley, Michael Stonebraker, utilizando el dinero de otro proyecto que ya tenía asignado, Ingres (INteractive Graphics REtrieval System), Stonebraker empezó a desarrollar sus ideas sobre bases de datos relacionales. Durante estos años Ingres mantuvo su código fuente abierto y permaneció en gran medida similar en conceptos a "System R".

A principio de los 80, Ingres estuvo compitiendo con Oracle por el liderazgo en el mundo de bases de datos relacionales y su código e implementación evolucionaron y fueron el origen de otras bases de datos relacionales (Martinez, 2009)

PostgreSQL comenzó su vida en 1986 como POSTGRES, un proyecto de investigación de la universidad de Berkeley en California, el equipo de Michael Stonebraker busco mejorar INGRES activamente por ocho años. Luego de ello Postgres se comercializó como ilustra, que fue adquirido por Informix, este a su vez fue comprado por IBM.

Postgres usaba su propio lenguaje de consultas, Postquel, aunque teóricamente superior al dominante SQL, con una mayor profundidad de expresión debido a sus fundamentos teóricos más avanzados, en la práctica Postquel no se alineaba con las necesidades de la industria así que en el 95 nació Postgres95. En 1996 el nombre cambio a PostgreSQL e iniciaron con la versión 6.⁹ (2ndquadrant, 2001)

Actualmente Postgresql se encuentra en la versión 9.4, su lanzamiento oficial fue el 18 de diciembre del 2014.

1.3.3 Características Fundamentales.-

⁹ 2ndquadrant. (2001). *Second Quadrant*. Obtenido de <http://2ndquadrant.com/>: <http://2ndquadrant.com/es/postgresql/postgresql-la-historia-hasta-ahora/>

- Contiene las normas de sintaxis de SQL92 y SQL99, en caso de que exista diferencias de sintaxis están son relacionados a rasgos únicos de PostgreSQL.
- PostgreSQL protege los datos con las transacciones que procesa y coordina a varios usuarios al mismo tiempo ya que el modelo de transacción está basado en el control de concurrencia de multiversión (MVCC) que da a PostgreSQL un mejor funcionamiento,
- Soporta diferentes tipos de datos como los de fecha, monetarios, elementos gráficos, datos para la comunicación de redes, cadenas de bits, datos booleanos, y datos explícitamente creados para conexiones de red.
- Permite que el usuario cree sus propios tipos de datos.
- Su estructura soporta arreglos de datos.
- Posee funciones para el manejo de fechas, geométricas como el punto, la recta, etc. Y también para ayudar en la operación con redes.
- Gestiona diferentes usuarios, y los permisos asignados a cada uno de ellos.
- Permite la declaración de funciones propias del usuario, así como la definición de disparadores o triggers o a su vez múltiples triggers, o también se pueden incluir nuevas funciones y operadores.
- Posee una fuerte integridad referencial, controlada con claves foráneas, primarias y también con triggers.
- Ayuda al mejor control de datos con el uso de índices, vistas y reglas.
- La herencia es una característica primordial de esta base de datos, se lo hace entre tablas pero no es posible establecer herencia entre objetos, ya que no existen, por lo que es incluido entre los gestores objeto – relacionales.
- Una de sus principales ventajas es su acceso gratuito.
- Es una herramienta construida en base a la arquitectura cliente/servidor.¹⁰ (Cáliz, 2007, págs. 11-12)

¹⁰ Cáliz, D. (2007). *Generador de reportes para postgres*. Quito, Ecuador: Escuela Politécnica nacional.

1.3.4 Ventajas y Desventajas.-

1.3.4.1 Ventajas.

1. **Instalación Ilimitada.-** Vale destacar que con PostgreSQL, no se viola ningún acuerdo de licencia, ya que no existe un costo asociado a la licencia de software.
2. **Soporte.-** Aunque la versión de licencia de software es gratuita, existe un sinnúmero de empresas y profesionales dispuestos a colaborar con el soporte de las bases de datos, aunque esto ya no es gratuito.
3. **Ahorros considerables en costos de operación.-** PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.
4. **Estabilidad y confiabilidad.-** es muy común que se no se reporten caídas en varios años de operación de las bases de datos.
5. **Extensible.-** El código fuente está disponible para todos sin costo, si se necesita extender o personalizar PostgreSQL de alguna manera, puede hacerlo con un mínimo esfuerzo, y sin costos adicionales.
6. **Diseñado para ambientes de algo volumen.-** PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes.
7. **Herramientas graficas de diseño y administración de base de datos.-** Existen varias herramientas graficas de alta calidad para administrar las bases de datos como: pgAdmin, pgAccess. Y para diseño como: Tora, Data Architect.
8. **Integridad referencial.-** soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
9. **Bloqueos de tabla y filas.-** ofrece varios modos de bloqueo para controlar el acceso concurrente a los datos en tablas, algunos de

ellos PostgreSQL los realiza automáticamente antes de la ejecución de una declaración.¹¹ (Humberto, 2005)

1.3.4.2 Desventajas.

1. En comparación con MySQL es más lento en inserciones y actualizaciones, ya que cuenta con cabeceras de intersección que no tiene MySQL.
2. El soporte en línea es escaso ya que no cuenta con ayuda obligatoria, y la ayuda que podemos conseguir no es económica.
3. Consume mayor cantidad de recursos que otros motores de base de datos como MySQL.
4. La sintaxis de algunos de sus comandos o sentencias no es intuitiva.¹² (Vázquez, 2012)

1.3.5 Aplicaciones de PostgreSQL.-

- **Afilias.-** “es el segundo mayor registro de nombres de dominio de Internet del mundo, con más de 20 millones de nombres bajo gestión”¹³ (Afilias, s.f.).



Grafica 11: Logo Afilias

Fuente: Afilias.info. Recuperado el 12/05/2015 de <http://www.afilias.info/sites/afilias.info/themes/afilias.info.2014/images/logo.png>

- **BASF.-** es una empresa química. Fue fundada a mediados de 1865 en la ciudad de Ludwigshafen, Renania-Palatinado, por Friedrich Engelhorn con el propósito de producir tintes.

¹¹ Humberto, E. (2005). *Link Global Services*. Obtenido de Link Global Services: http://www.lgs.com.ve/pres/PresentacionES_PSQL.pdf

¹² Vázquez, L. (06 de Enero de 2012). *Blogestudio red de medios*. Obtenido de Empresa y economía: <http://www.empresayeconomia.es/aplicaciones-para-empresas/ventajas-y-desventajas-de-postgresql.html>

¹³ Afilias. (s.f.). *Afilias página oficial*. Obtenido de Afilias: <http://www.afilias.info/about-us>



Grafica 12: Logo BASF

Fuente: Basf.com.ec. Recuperado el 12/05/2015 de http://www.basf.com.ec/img/navegacion/BASF_logo.jpg

- **Mohawk Software.-** crea aplicaciones y soluciones para distintos negocios y en el proceso, desarrolla tecnología y servicios públicos que tienen aplicaciones más amplias que para las que fueron diseñadas.



Grafica 13: Logo Mohawk

Fuente: Mohawk college. Recuperado el 12/05/2015 de <http://www.mohawkcollege.ca/Sites/3/templates/images/futureready-template-logo.gif>

- **Proximity Group.-** es una empresa líder en el mercado de gestión y evaluación de medios de comunicación.



Grafica 14: Logo Proximity Group

Fuente: Proximitygroup.com. Recuperado el 12/05/2015 de <http://www.proximitygroup.com/images/ProximityGroup-Final-Logo.gif>

- **Radio Paradise.-** es una estación de radio por internet dedicada a ofrecer música alternativa.



Grafica 15: Logo Radio Paradise

Fuente: Radioparadise.com. Recuperado el 12/05/2015 de <http://www.radioparadise.com/clear.gif>

- **DealBase.-** es un buscador de hoteles, vuelos, viajes, etc.



Grafica 16: Logo DealBase

Fuente: Dealbase.com. Recuperado el 12/05/2015 de http://www.dealbase.com/assets/web_logo_trans-3d3d0d6d9191bae10f76e34e0b0317e0.png

- **Valhalla Data Systems.-** es una empresa de productos y servicios para ayudar a las organizaciones en sus problemas de tecnologías de la información.



Grafica 17: Logo Valhalla Data Systems

Fuente: Valhallads.com. Recuperado el 12/05/2015 de Valhallads.com

- **Vanten Inc.-** desarrolla soluciones de tecnología de código abierto para el mercado japonés.



Grafica 18: Logo Vanten Inc.

Fuente: Vanten.com. Recuperado el 12/05/2015 de <http://www.vanten.com/E/images/header-E.jpg>

CAPITULO 2. ANALISIS DE REQUERIMIENTOS

2.1 Modelado Del Negocio

2.1.1 Situación actual del negocio.-

En la actualidad el restaurante “La Fogata” carece de fuentes fidedignas que informen cual es la utilidad real del negocio, tampoco se poseen datos de los gastos realizados diariamente, adicionalmente, la naturaleza del ser humano es resistirse a cambios, en el caso de los propietarios de "La Fogata" no es la excepción, ellos realizan su trabajo de manera autónoma, en base a su destreza y experiencia, temen incursionar en nuevos métodos de control, consideran que los cambios que se ocasionen en el negocio no va a permitir obtener los resultados deseados.

Existen varios problemas debido a la falta de control, uno de ellos es el cobro de los consumos en el local ya que ha habido muchas veces en las cuales los clientes se han ido sin pagar. El control de emisión de comprobantes de venta es nulo, el mayor inconveniente es cuando el cliente solicita se le aumente algunos dólares al total de la factura, se lo hace sin ninguna objeción. Por otro lado en el caso de las compras no existe un registro diario de anotaciones, la cancelación a proveedores se la realiza en relación al efectivo del momento, es decir si existe efectivo se cancela, si no existe efectivo se posterga el pago.

2.1.2 Descripción del sistema propuesto.-

El sistema de administración para la facturación mantendrá un registro de compras, ventas con un registro del respectivo usuario que realizó la creación o modificación de dichos documentos, almacenará la lista de proveedores y clientes. Vale recalcar que para que el sistema de facturación funcione adecuadamente se debe realizar una modificación en la forma actual de cobros de los pedidos, se los debe cancelar por anticipado. Los reportes del sistema ayudaran a obtener una información clara de las utilidades mensuales del negocio, conjuntamente se emitirá estadísticas mensuales de ventas y compras para conocer cuáles son los meses de mayor afluencia de público, en los reportes que ameriten se incluirá un componente con el cual podremos escoger la fecha de inicio y final del reporte, por último el sistema emitirá un informe que facilitará las declaraciones de impuestos mensuales y anuales.

2.1.3 Descripción de las necesidades del negocio en el sistema.-

“Siempre intentamos satisfacer las que creemos que son las necesidades de la empresa y de los que trabajan en ella, cuando en realidad no tenemos ni idea de lo que de verdad necesitan. Todos hacemos las cosas por razones que no valen solo a nosotros.”¹⁴ (Vivo Emprendiendo, 2015)

Cuando llega el momento de adaptar los servicios de TI, surgen varias necesidades apremiantes que las empresas necesitan satisfacer. Suelen centrarse en las siguientes cinco áreas: expansión, elasticidad, velocidad y entrega, niveles de servicio y detalles comerciales.

Para nuestro propósito nos concentraremos en la velocidad de entrega y los detalles comerciales, el negocio entorno a pedidos funciona de manera caótica, y es lo que necesitamos optimizar, mejorando con ello la velocidad en reparto de pedidos además de controlar de una manera óptima los ingresos y gastos que es el primordial deseo de la administración del restaurante “La Fogata”.

2.2 Requerimientos

2.2.1 Requerimientos Funcionales

2.2.1.1 Administraciones

- Usuario
- Cliente
- Proveedor
- Pago
- Menú

2.2.1.2 Procesos

- Ingreso al sistema
- Facturación
- Compras

2.2.1.3 Perfiles de Actores

- Administrador
- Cajero
- Súper

¹⁴ Vivo Emprendiendo. (30 de 11 de 2015). <http://vivoemprendiendo.com/>. Obtenido de <http://vivoemprendiendo.com/>: <http://vivoemprendiendo.com/>

2.2.2 Diagrama De Casos de Uso

2.2.2.1 Diagrama General

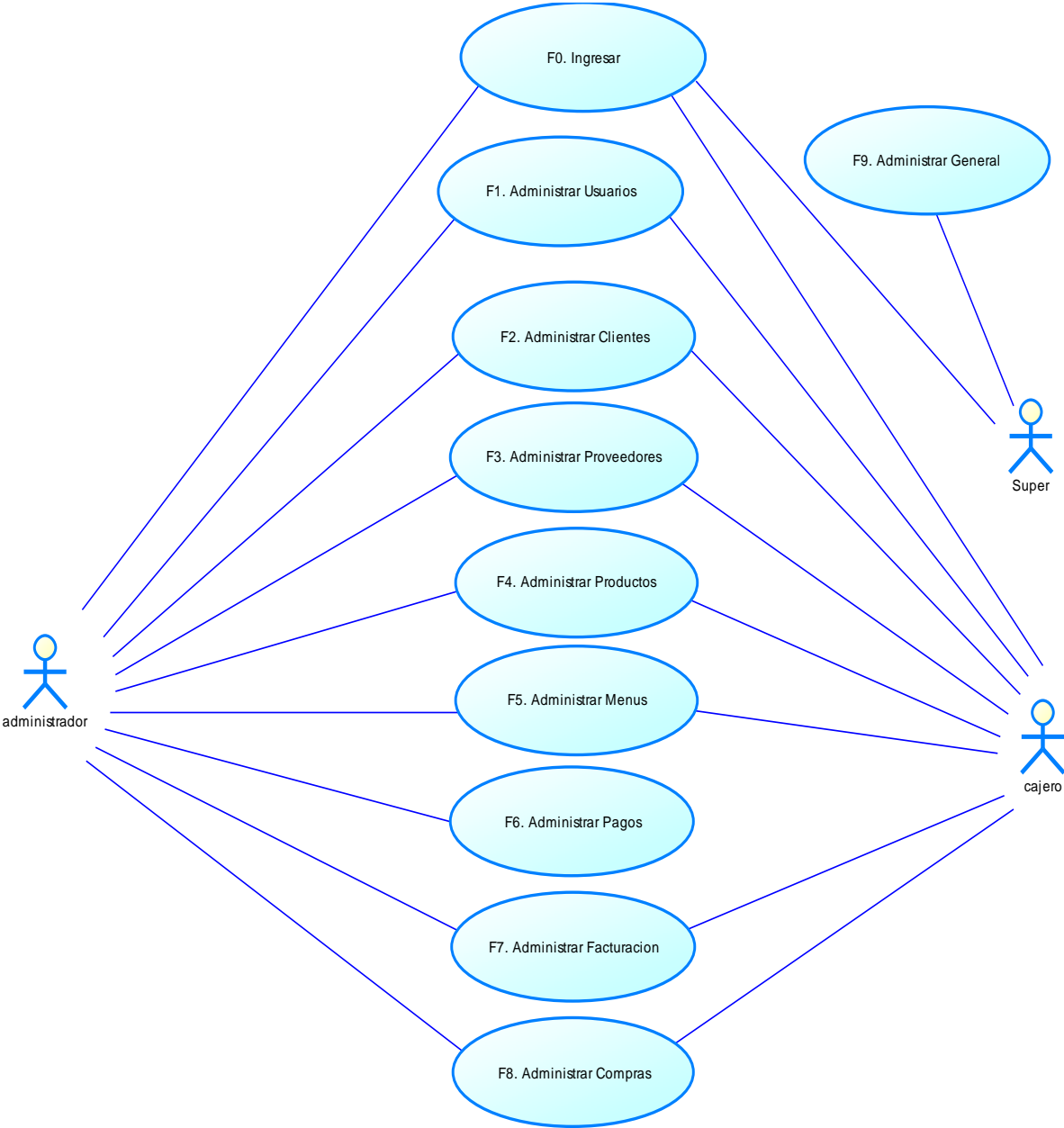


Diagrama 3: Diagrama General

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.2 Ingreso al Sistema

Caso de Uso: Ingreso al Sistema

Identificador: F0

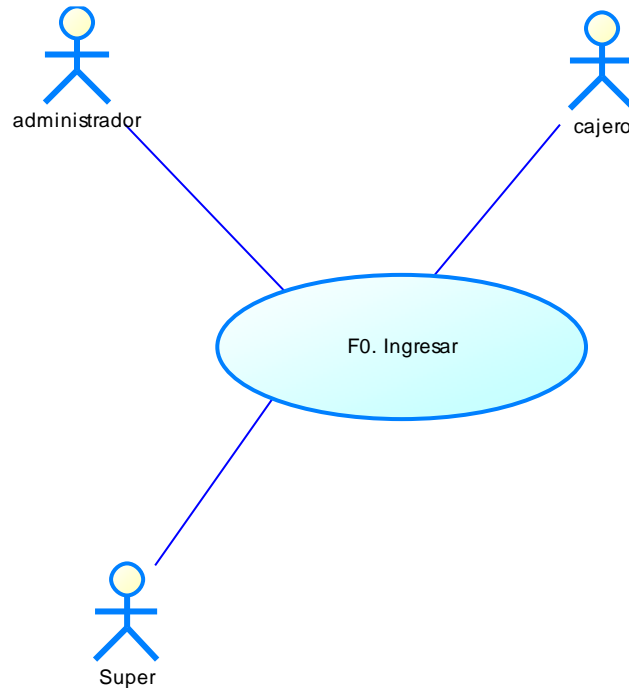


Diagrama 4: Ingreso al Sistema

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Es solicitado al abrir al programa, todo usuario debe pasar por este proceso para su respectiva validación de usuario y clave.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	El actor abre el sistema			
2	El actor Ingresar la clave y contraseña	3	El sistema verifica los datos	E1, E2, E3
		4	El sistema permite el acceso, mostrando la pantalla principal	

Tabla 2: Flujo de Ingreso al Sistema

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Usuario no encontrado	Mensaje de Error
E2	Contraseña incorrecta	Mensaje de Error
E3	Error de conexión con la base de datos	Mensaje de Error

Tabla 3: Excepciones de Ingreso al Sistema

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.3 Administrar Usuarios

Caso de Uso: Administración de Usuarios

Identificador: F1

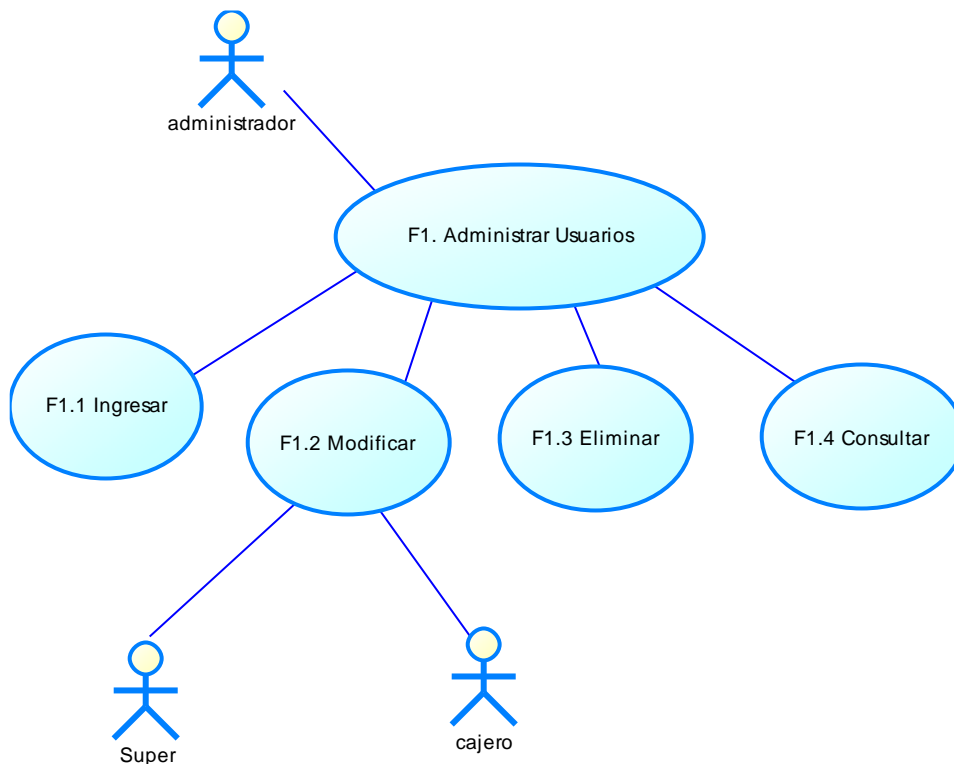


Diagrama 5: Administrar Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Esta administración nos sirve para el manejo de usuarios, pero solo el Administrador tendrán acceso total de la administración, el cajero y el súper usuario solo podrán modificar la contraseña de su usuario.

2.2.2.3.1 Ingresar Usuarios

Caso de Uso: Ingresar Usuarios

Identificador: F1.1

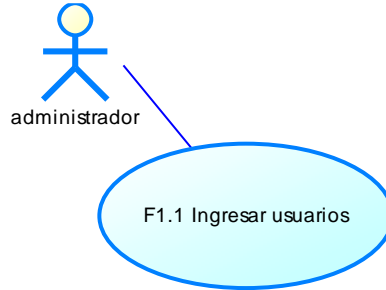


Diagrama 6: Ingresar Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la creación de usuarios y la otorgación de niveles, es decir un administrador podrá crear otro usuario como administrador o tipo usuario.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de usuarios	2	El sistema presenta el administrador de usuarios	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2

Tabla 4: Flujo de Ingreso de Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Usuario ya Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 5: Excepciones de Ingreso de Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.3.2 *Modificar Usuarios*

Caso de Uso: Modificar Usuarios

Identificador: F1.2

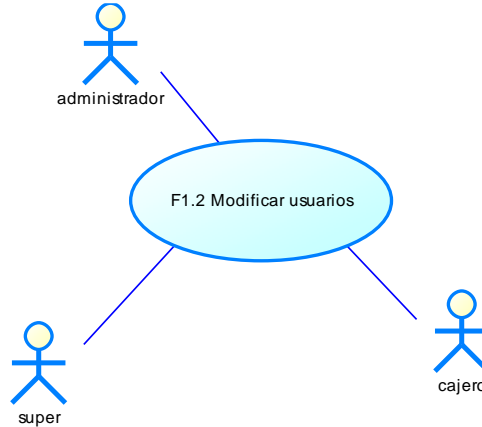


Diagrama 7: Modificar Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la modificación de los usuarios por parte del administrador, pero solo se puede modificar el nivel del usuario, no se tiene acceso a los otros datos como nombre o contraseña ya que se consideran datos personales. Los perfiles súper y cajero solo podrán modificar sus respectivas contraseñas.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de usuarios	2	El sistema presenta el administrador de usuarios	
3	Se abre la pestaña de modificación			
4	Se elige al usuario y se modifica el nivel	6	Se verifican los datos	E1, E2
5	Se presiona el botón guardar	7	Se guardan los datos	E2

Tabla 6: Flujo de Modificación de Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Usuario no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 7: Excepciones de Modificación de Usuarios

Elaborado por: Osvaldo Tapia
 2.2.2.3.3 *Eliminar Usuarios*
Caso de Uso: Eliminar Usuarios

Fecha: 13/05/2015

Identificador: F1.3

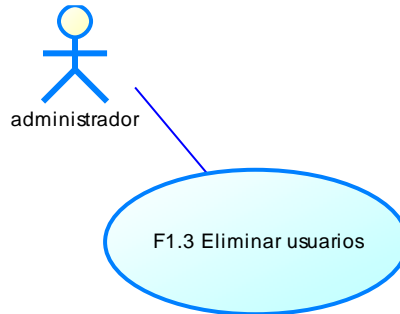


Diagrama 8: Eliminar Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite el borrado de los usuarios.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de usuarios	2	El sistema presenta el administrador de usuarios	
3	Se abre la pestaña de Eliminar			
4	Se elige un usuario a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el usuario	E2

Tabla 8: Flujo de Eliminación de Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Usuario no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 9: Excepciones de Eliminación de Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.3.4 Consultar Usuarios

Caso de Uso: Consultar Usuarios

Identificador: F1.4

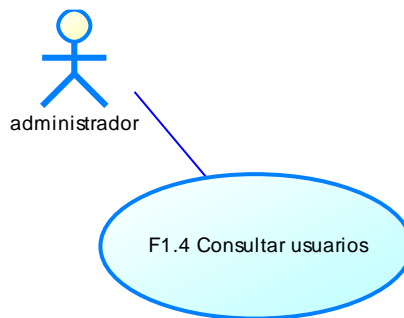


Diagrama 9: Consultar Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la consulta de los usuarios.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar a consulta de usuarios	2	El sistema presenta los datos de usuarios	E1, E2

Tabla 10: Flujo de Consulta de Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	No existen datos	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 11: Excepciones de Consulta de Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.4 Administrar Clientes

Caso de Uso: Administración de Clientes

Identificador: F2

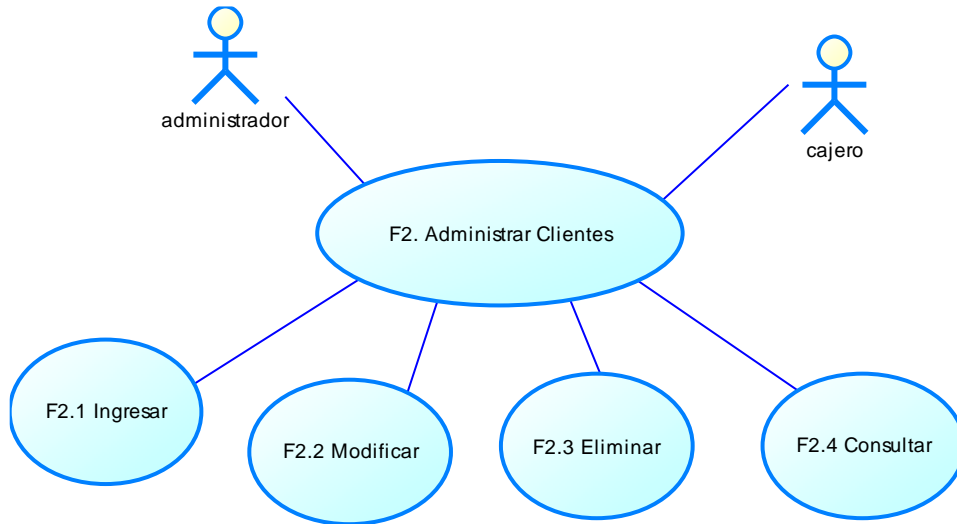


Diagrama 10: Administrar Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Esta administración nos sirve para el manejo de clientes.

2.2.2.4.1 Ingresar Clientes

Caso de Uso: Ingresar Clientes

Identificador: F2.1

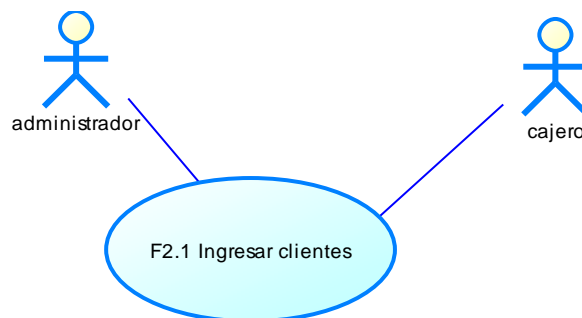


Diagrama 11: Ingresar Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la creación de clientes.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de clientes	2	El sistema presenta el administrador de clientes	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2

Tabla 12: Flujo de Ingreso de Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Cliente ya Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 13: Excepciones de Ingreso de Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.4.2 Modificar Clientes

Caso de Uso: Modificar Clientes

Identificador: F2.2

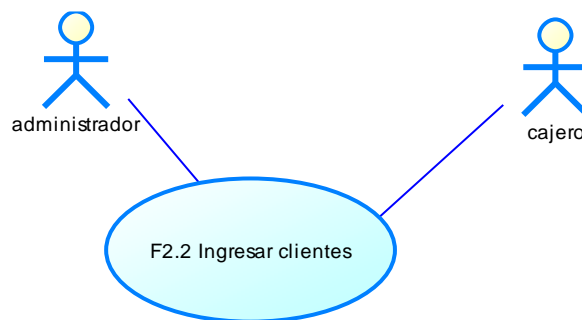


Diagrama 12: Modificar Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la modificación de los clientes.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de clientes	2	El sistema presenta el administrador de clientes	
3	Se abre la pestaña de modificación			
4	Se elige al Cliente y se modifica los datos	6	Se verifican los datos	E1, E2
5	Se presiona el botón guardar	7	Se guardan los datos	E2

Tabla 14: Flujo de Modificación de Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Cliente no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 15: Excepciones de Modificación de Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.4.3 Eliminar Clientes

Caso de Uso: Eliminar Clientes

Identificador: F2.3

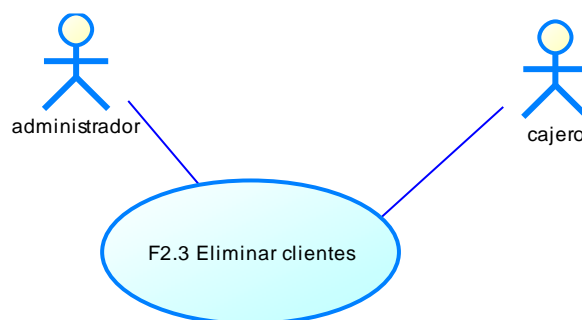


Diagrama 13: Eliminar Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite el borrado de los clientes.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de clientes	2	El sistema presenta el administrador de clientes	
3	Se abre la pestaña de Eliminar			
4	Se elige un Cliente a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el Cliente	E2

Tabla 16: Flujo de Eliminación de Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Cliente no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 17: Excepciones de Eliminación de Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.4.4 Consultar Clientes

Caso de Uso: Consultar Clientes

Identificador: F2.4

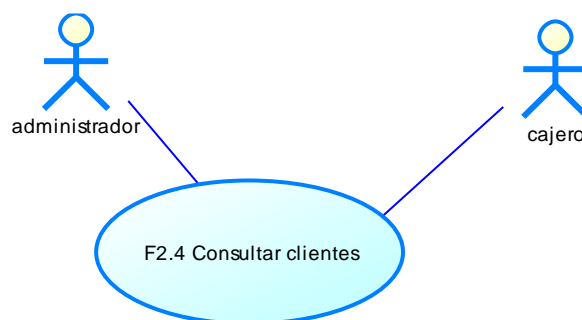


Diagrama 14: Consultar Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la consulta de los Clientes.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar a consulta de clientes	2	El sistema presenta los datos de clientes	E1, E2

Tabla 18: Flujo de Consulta de Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	No existen datos	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 19: Excepciones de Consulta de Clientes

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.5 Administrar Proveedores

Caso de Uso: Administración de Proveedores

Identificador: F3

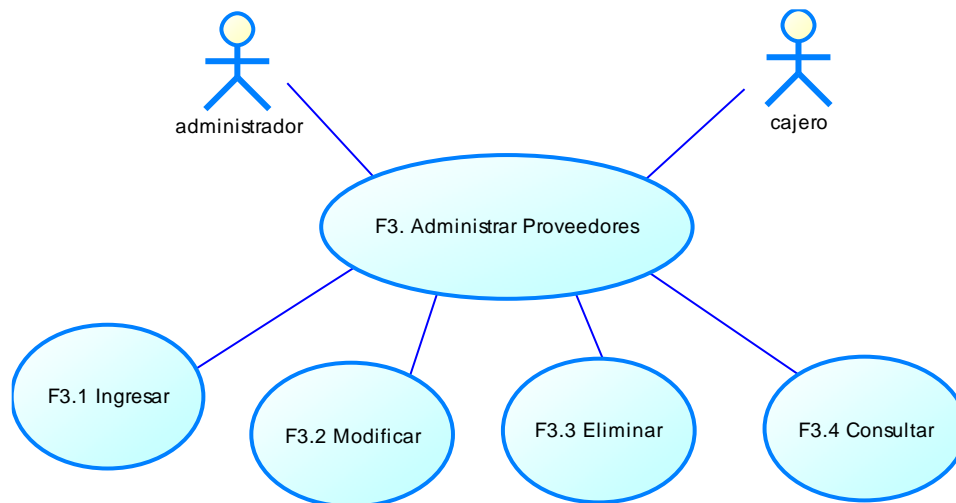


Diagrama 15: Administrar Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Esta administración nos sirve para el manejo de proveedores.

2.2.2.5.1 Ingresar Proveedores

Caso de Uso: Ingresar Proveedores

Identificador: F3.1

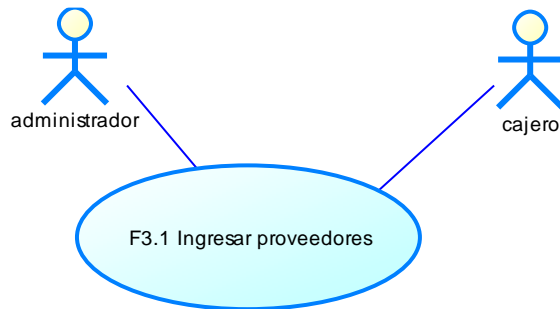


Diagrama 16: Ingresar Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la creación de proveedores.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Proveedores	2	El sistema presenta el administrador de Proveedores	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2

Tabla 20: Flujo de Ingreso de Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Proveedor ya Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 21: Excepciones de Ingreso de Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.5.2 *Modificar Proveedores*

Caso de Uso: Modificar Proveedores

Identificador: F3.2

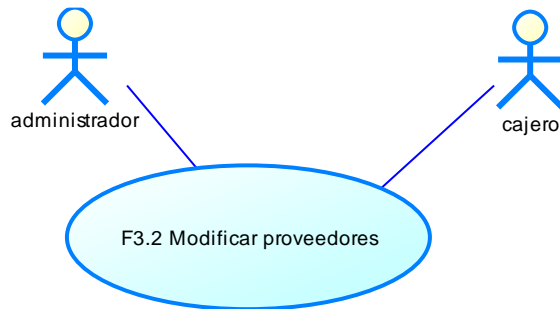


Diagrama 17: Modificar Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la modificación de los Proveedores.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Proveedores	2	El sistema presenta el administrador de Proveedores	
3	Se abre la pestaña de modificación			
4	Se elige al Proveedor y se modifica los datos	6	Se verifican los datos	E1, E2
5	Se presiona el botón guardar	7	Se guardan los datos	E2

Tabla 22: Flujo de Modificación de Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Proveedor no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 23: Excepciones de Modificación de Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.5.3 Eliminar Proveedores

Caso de Uso: Eliminar Proveedores

Identificador: F3.3

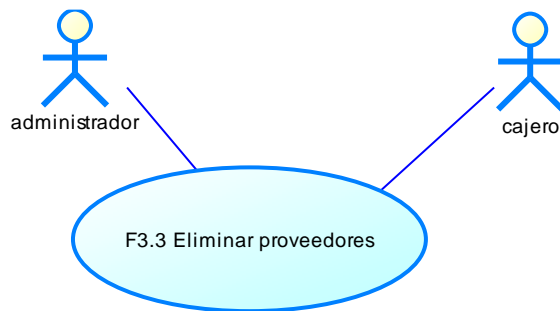


Diagrama 18: Eliminar Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite el borrado de los Proveedores.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Proveedores	2	El sistema presenta el administrador de Proveedores	
3	Se abre la pestaña de Eliminar			
4	Se elige un Proveedor a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el Proveedor	E2

Tabla 24: Flujo de Eliminación de Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Proveedor no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 25: Excepciones de Eliminación de Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.5.4 Consultar Proveedores

Caso de Uso: Consultar Proveedores

Identificador: F3.4

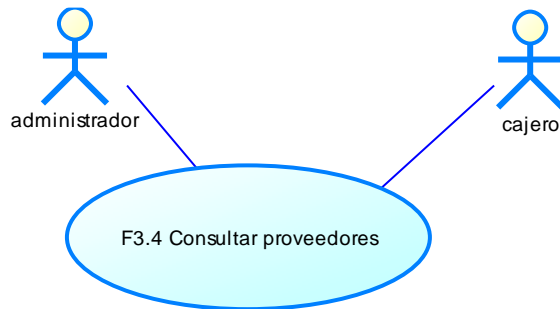


Diagrama 19: Consultar Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la consulta de los Proveedores.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar a consulta de Proveedores	2	El sistema presenta los datos de Proveedores	E1, E2

Tabla 26: Flujo de Consulta de Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	No existen datos	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 27: Excepciones de Consulta de Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.6 Administrar Productos

Caso de Uso: Administración de Productos

Identificador: F4

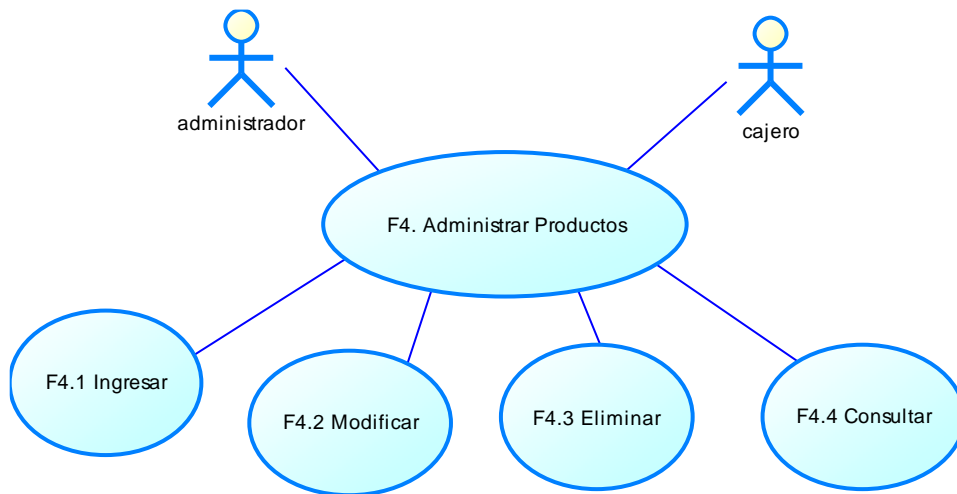


Diagrama 20: Administrar Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Esta administración nos sirve para el manejo de Productos.

2.2.2.6.1 Ingresar Productos

Caso de Uso: Ingresar Productos

Identificador: F4.1

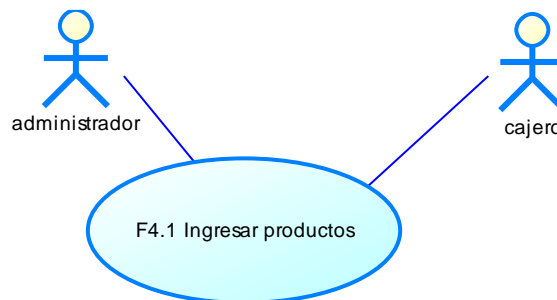


Diagrama 21: Ingresar Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la creación de Productos.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Productos	2	El sistema presenta el administrador de Productos	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2

Tabla 28: Flujo de Ingreso de Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Producto ya Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 29: Excepciones de Ingreso de Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.6.2 Modificar Productos

Caso de Uso: Modificar Productos

Identificador: F4.2

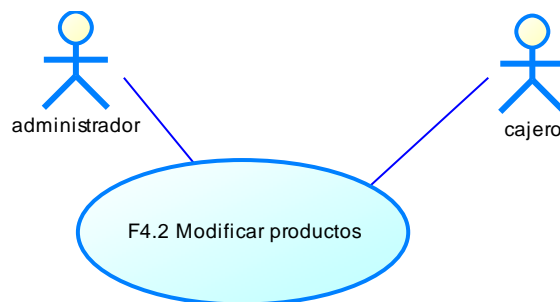


Diagrama 22: Modificar Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la modificación de los Productos.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Productos	2	El sistema presenta el administrador de Productos	
3	Se abre la pestaña de modificación			
4	Se elige al Producto y se modifica los datos	6	Se verifican los datos	E1, E2
5	Se presiona el botón guardar	7	Se guardan los datos	E2

Tabla 30: Flujo de Modificación de Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Producto no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 31: Excepciones de Modificación de Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.6.3 Eliminar Productos

Caso de Uso: Eliminar Productos

Identificador: F4.3

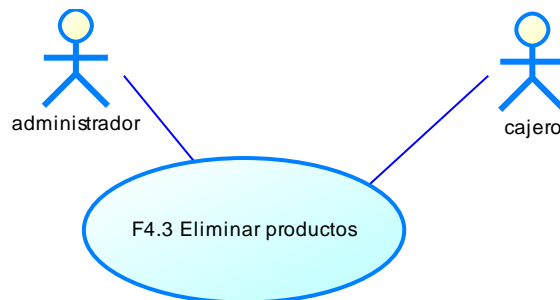


Diagrama 23: Eliminar Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite el borrado de los Productos.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Productos	2	El sistema presenta el administrador de Productos	
3	Se abre la pestaña de Eliminar			
4	Se elige un Producto a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el Producto	E2

Tabla 32: Flujo de Eliminación de Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Producto no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 33: Excepciones de Eliminación de Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.6.4 Consultar Productos

Caso de Uso: Consultar Productos

Identificador: F4.4

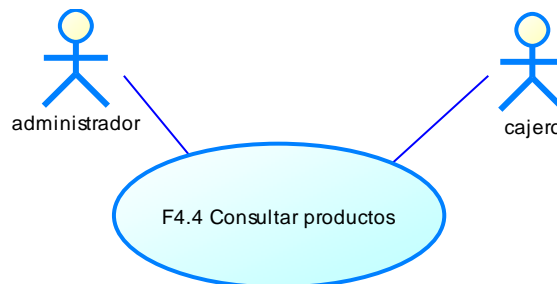


Diagrama 24: Consultar Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la consulta de los Productos.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar a consulta de Productos	2	El sistema presenta los datos de Productos	E1, E2

Tabla 34: Flujo de Consulta de Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	No existen datos	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 35: Excepciones de Consulta de Productos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.7 Administrar Menús

Caso de Uso: Administración de Menús

Identificador: F5

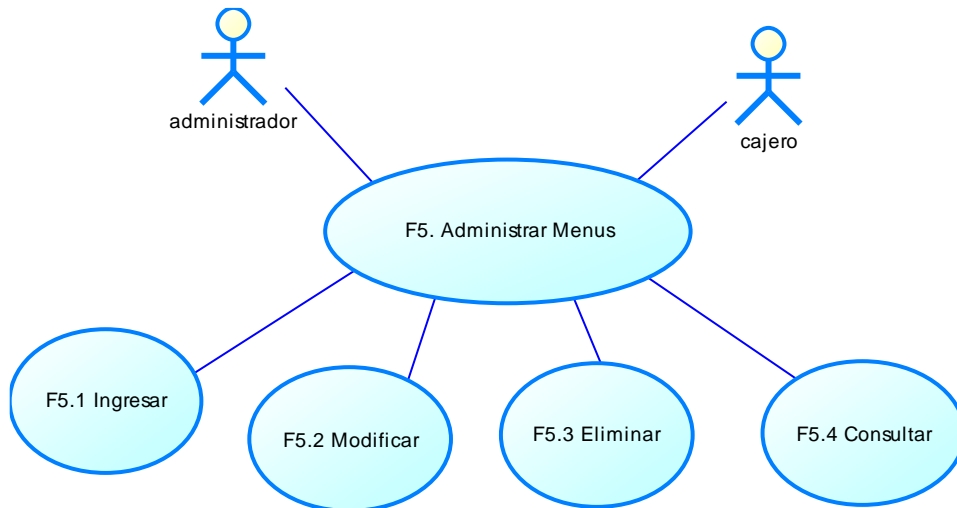


Diagrama 25: Administrar Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Esta administración nos sirve para el manejo de Menús.

2.2.2.7.1 Ingresar Menús

Caso de Uso: Ingresar Menús

Identificador: F5.1

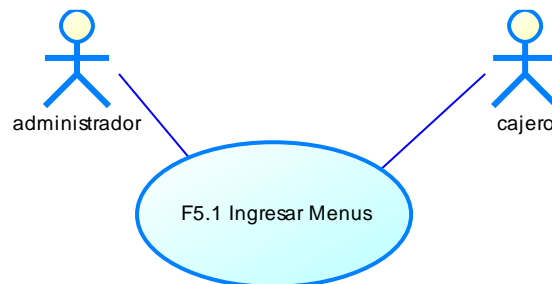


Diagrama 26: Ingresar Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la creación de Menús.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Menús	2	El sistema presenta el administrador de Menús	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2

Tabla 36: Flujo de Ingreso de Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Menú ya Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 37: Excepciones de Ingreso de Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.7.2 Modificar Menús

Caso de Uso: Modificar Menús

Identificador: F5.2

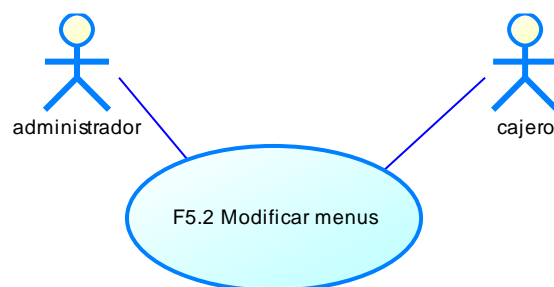


Diagrama 27: Modificar Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la modificación de los Menús.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Menús	2	El sistema presenta el administrador de Menús	
3	Se abre la pestaña de modificación			
4	Se elige al Menú y se modifica los datos	6	Se verifican los datos	E1, E2
5	Se presiona el botón guardar	7	Se guardan los datos	E2

Tabla 38: Flujo de Modificación de Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Menú no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 39: Excepciones de Modificación de Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.7.3 Eliminar Menús

Caso de Uso: Eliminar Menús

Identificador: F5.3

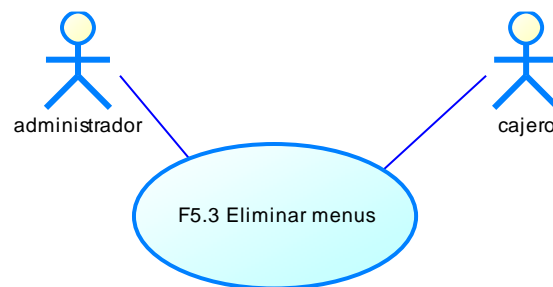


Diagrama 28: Eliminar Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite el borrado de los Menús.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Menús	2	El sistema presenta el administrador de Menús	
3	Se abre la pestaña de Eliminar			
4	Se elige un Menú a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el Menú	E2

Tabla 40: Flujo de Eliminación de Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Menú no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 41: Excepciones de Eliminación de Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.7.4 Consultar Menús

Caso de Uso: Consultar Menús

Identificador: F5.4

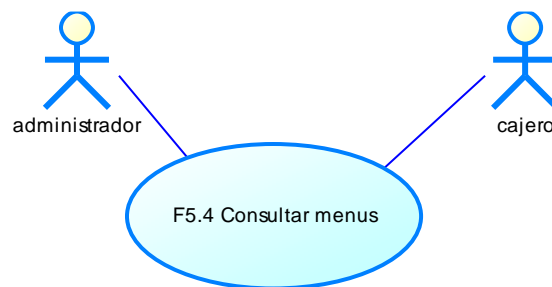


Diagrama 29: Consultar Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la consulta de los Menús.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar a consulta de Menús	2	El sistema presenta los datos de Menús	E1, E2

Tabla 42: Flujo de Consulta de Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	No existen datos	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 43: Excepciones de Consulta de Menús

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.8 Administrar Pagos

Caso de Uso: Administración de Pagos

Identificador: F6

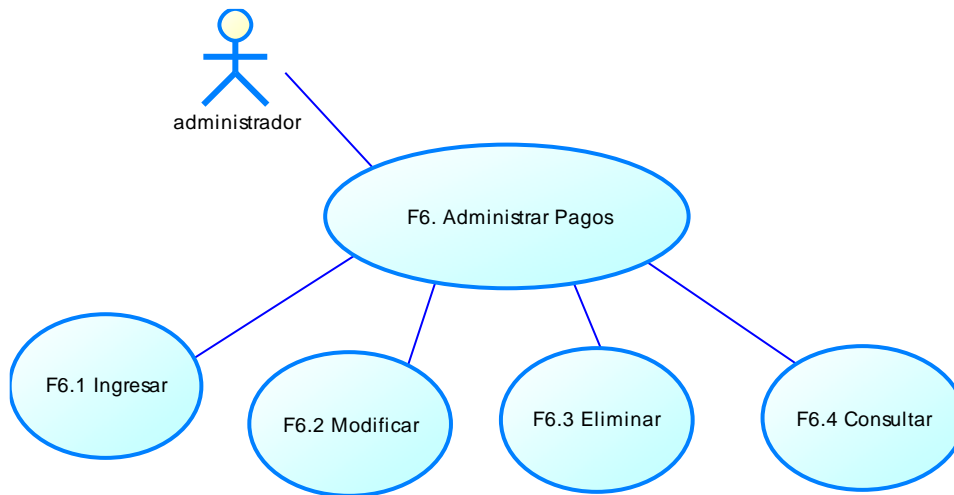


Diagrama 30: Administrar Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Esta administración nos sirve para el manejo de la forma de pago tanto en compras como en facturación.

2.2.2.8.1 Ingresar Pago
Caso de Uso: Ingresar Pagos

Identificador: F6.1

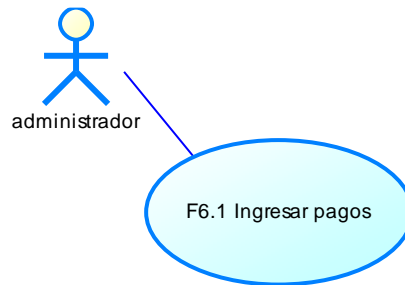


Diagrama 31: Ingresar Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la creación de Pagos para compras y para ventas.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Pagos	2	El sistema presenta el administrador de Pagos	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2

Tabla 44: Flujo de Ingreso de Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Pago ya Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 45: Excepciones de Ingreso de Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.8.2 *Modificar Pagos*

Caso de Uso: Modificar Pagos

Identificador: F6.2

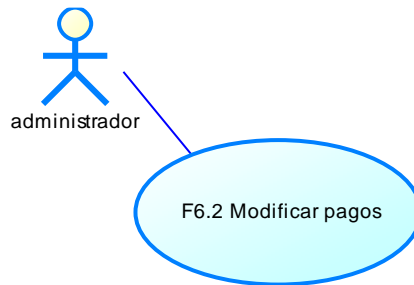


Diagrama 32: Modificar Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la modificación de los Pagos.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Pagos	2	El sistema presenta el administrador de Pagos	
3	Se abre la pestaña de modificación			
4	Se elige al Pago y se modifica los datos	6	Se verifican los datos	E1, E2
5	Se presiona el botón guardar	7	Se guardan los datos	E2

Tabla 46: Flujo de Modificación de Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Pago no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 47: Excepciones de Modificación de Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.8.3 Eliminar Pago

Caso de Uso: Eliminar Pagos

Identificador: F6.3

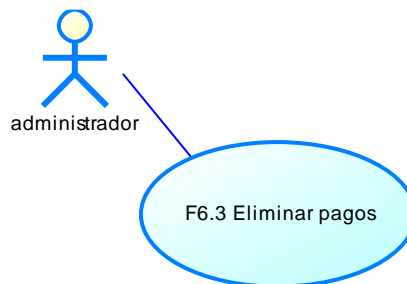


Diagrama 33: Eliminar Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite el borrado de los Pagos.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Pagos	2	El sistema presenta el administrador de Pagos	
3	Se abre la pestaña de Eliminar			
4	Se elige un Pago a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el Pago	E2

Tabla 48: Flujo de Eliminación de Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Pago no Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 49: Excepciones de Eliminación de Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.8.4 Consultar Pagos

Caso de Uso: Consultar Pagos

Identificador: F6.4

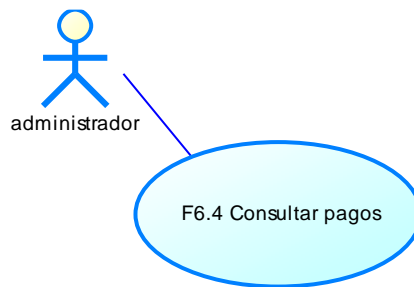


Diagrama 34: Consultar Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la consulta de los Pagos.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar a consulta de Pagos	2	El sistema presenta los datos de Pagos	E1, E2

Tabla 50:

Flujo de Consulta de Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	No existen datos	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 51: Excepciones de Consulta de Pagos

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.9 Administrar Factura

Caso de Uso: Administración de Facturas

Identificador: F7

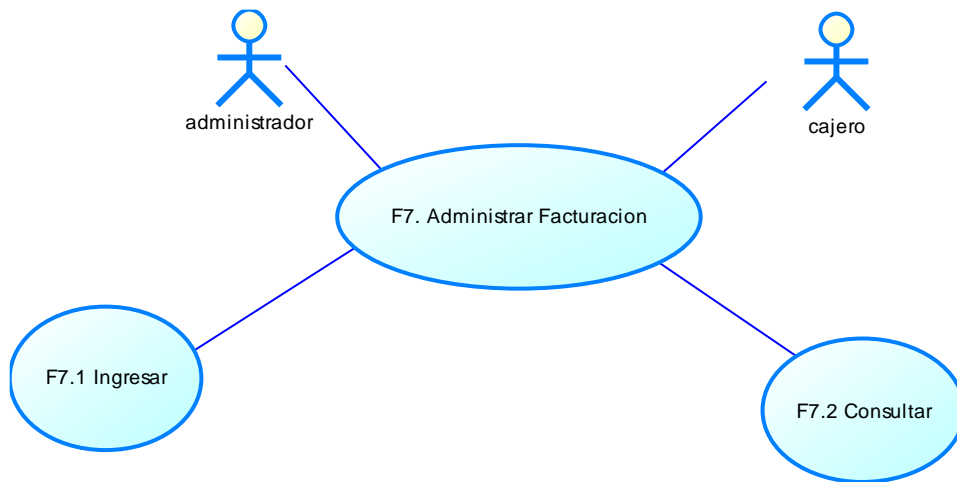


Diagrama 35: Administrar Facturas

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Esta administración nos sirve para el manejo de Facturas.

2.2.2.9.1 Ingresar Facturas

Caso de Uso: Ingresar Facturas

Identificador: F7.1

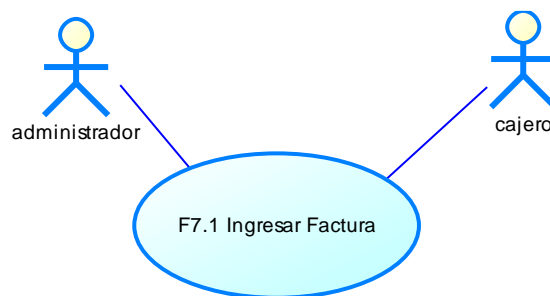


Diagrama 36: Ingresar Facturas

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la creación de Facturas.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Facturas	2	El sistema presenta el administrador de Facturas	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2

Tabla 52: Flujo de Ingreso de Facturas

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Factura ya Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 53: Excepciones de Ingreso de Facturas

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.9.2 Consultar Facturas

Caso de Uso: Consultar Facturas

Identificador: F7.2

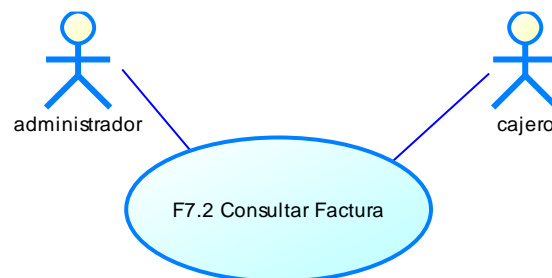


Diagrama 37: Consultar Facturas

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la consulta de las Facturas.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar a consulta de Facturas	2	El sistema presenta los datos de Facturas	E1, E2

Tabla 54: Flujo de Consulta de Facturas

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	No existen datos	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 55: Excepciones de Consulta de Facturas

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.10 Administrar Compras

Caso de Uso: Administración de Compras

Identificador: F8

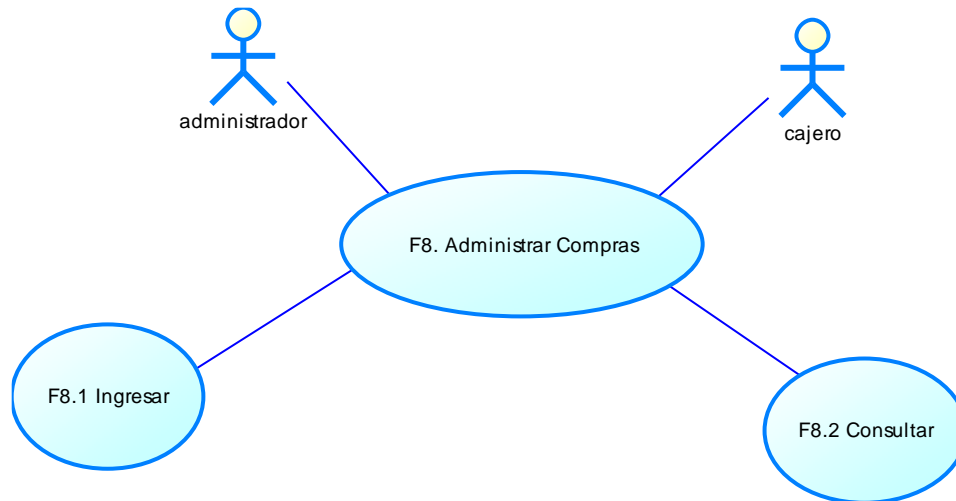


Diagrama 38: Administrar Compras

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Esta administración nos sirve para el manejo de Compras.

2.2.2.10.1 Ingresar Compras

Caso de Uso: Ingresar Compras

Identificador: F8.1

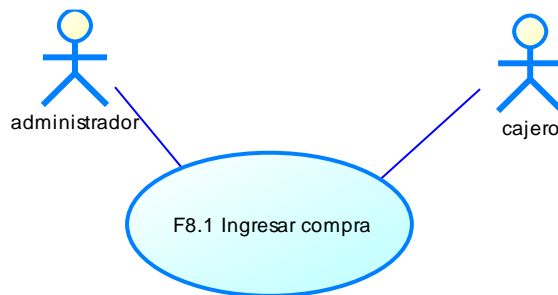


Diagrama 39: Ingresar Compras

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la creación de Compras.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador de Compras	2	El sistema presenta el administrador de Compras	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2

Tabla 56: Flujo de Ingreso de Compras

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Compra ya Existe	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 57: Excepciones de Ingreso de Compras

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.10.4 Consultar Compras

Caso de Uso: Consultar Compras

Identificador: F8.2

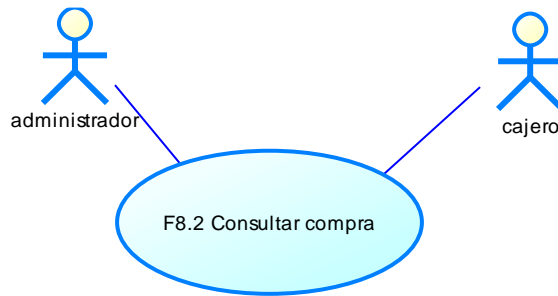


Diagrama 40: Consultar Compras

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la consulta de las Compras.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar a consulta de Compras	2	El sistema presenta los datos de Compras	E1, E2

Tabla 58: Flujo de Consulta de Compras

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	No existen datos	Mensaje de Error
E2	Error de conexión con la base de datos	Mensaje de Error

Tabla 59: Excepciones de Consulta de Compras

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.2.11 Administrar General

Caso de Uso: Administración General

Identificador: F9

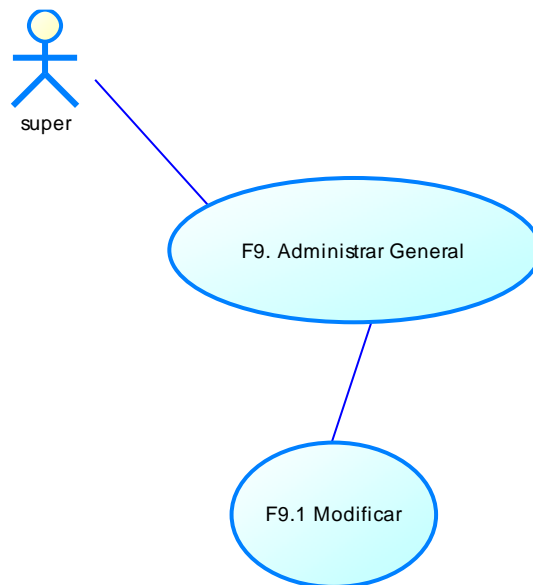


Diagrama 41: Administrar General

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Esta administración nos sirve para el manejo de los porcentajes de impuestos. Se maneja los impuestos de IVA y el impuesto anual de renta.

2.2.2.11.1 Modificar General.

Caso de Uso: Modificar General

Identificador: F9.1

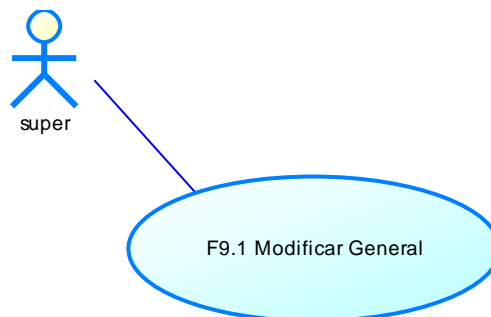


Diagrama 42: Modificar General

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Descripción: Permite la modificación de la tabla General.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador General	2	El sistema presenta el administrador General	
3	Se abre la pestaña de modificación			
4	Se elige el impuesto a modificar, y se lo modifica	6	Se verifican los datos	E1
5	Se presiona el botón guardar	7	Se guardan los datos	E1

Tabla 60: Flujo de Modificación General

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

Excepciones		
Id	Nombre	Acción
E1	Error de conexión con la base de datos	Mensaje de Error

Tabla 61: Excepciones de Modificación General

Elaborado por: Osvaldo Tapia

Fecha: 13/05/2015

2.2.3 Reportes necesarios.-

Los reportes del sistema nos sirven para proporcionar información acerca de las labores de nuestra empresa, pero su principal objeto es la toma de decisiones, para nuestro cometido incluiremos los siguientes reportes:

- Reporte de compras.
- Reporte de ventas.
- Reporte con datos para realizar las declaraciones al SRI.
- Reporte de clientes.
- Reporte de proveedores.

Vale indicar que en los reportes de compras, ventas, con y sin gráficos además del reporte para declaraciones existe un componente para elegir las fechas inicio y fin de los reportes.

2.2.4 Requerimientos de hardware y software.-

2.2.4.1 Requerimientos de hardware mínimos.-

- Pentium 4 o equivalente.
- 512 MB en RAM.
- 256 MB de espacio libre en disco duro.

2.2.4.1 Requerimientos de software.-

- Windows, desde XP hasta 8, Mac OS X.
- Java 7 (JRE)

2.2.5 Requerimientos de facturación electrónica.-

Este requerimiento se tomó en cuenta para ser analizado, por proposición de la contadora externa del restaurante “La Fogata”. Las siguientes tablas fueron tomadas de la página web del servicio de rentas internas del Ecuador. Con el título de “Quienes están obligados a emitir comprobantes electrónicos”

GRUPO	FECHA DE INICIO	SUJETOS PASIVOS
1	A partir de 01/08/2014	<ul style="list-style-type: none"> • Sociedades emisoras y administradoras de tarjetas de crédito
2	A partir de 01/10/2014	<ul style="list-style-type: none"> • Instituciones financieras bajo el control de la Superintendencia de Bancos y Seguros excepto mutualistas de ahorro y crédito para la vivienda y sociedades emisoras y administradoras de tarjetas de crédito. • Contribuyentes Especiales que realicen, según su inscripción en el RUC actividades económicas correspondientes al sector telecomunicaciones y subsector: televisión pagada. • Exportadores calificados por el SRI como contribuyentes especiales.
3	A partir de 01/01/2015	<ul style="list-style-type: none"> • Los demás contribuyentes especiales no señalados en los grupos anteriores. • Contribuyentes que posean autorización de impresión de comprobantes de venta, retención y documentos complementarios, a través de sistemas computarizados (autoimpresores). • Contribuyentes que realicen ventas a través de internet. • Los sujetos pasivos que realicen actividades económicas de exportación.

Diagrama 43: Sector privado, obligados a emitir comprobantes electrónicos.

Fuente: Servicio de rentas internas En SRI. Recuperado de <http://www.sri.gob.ec/de/10109>

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

GRUPO	FECHA DE INICIO	SUJETOS PASIVOS
1	01/01/2015	<ul style="list-style-type: none"> • Empresas públicas y empresas de servicios públicos. • Entidades del Sector Público Financiero. • Empresas de economía mixta.
2	01/04/2015	<ul style="list-style-type: none"> • Organismos y entidades de la Función Ejecutiva. • La Asamblea Nacional • Organismos y entidades de la Función Judicial, con excepción de sus organismos auxiliares mencionados en el Art. 178 de la Constitución de la República del Ecuador. • Los organismos y entidades de la Función de Transparencia y Control Social. • Los organismos y entidades de la Función Electoral. • Universidades y Escuelas Politécnicas públicas.
3	01/07/2015	<ul style="list-style-type: none"> • Los organismos y entidades de los Gobiernos Autónomos Descentralizados, incluidas las mancomunidades conformadas por los mismos. • Los organismos y entidades públicas no descritas en ninguno de los grupos señalados anteriormente.

Diagrama 44: Sector público, obligados a emitir comprobantes electrónicos.

Fuente: Servicio de rentas internas En SRI. Recuperado de <http://www.sri.gob.ec/de/10109>

Adicionalmente se incluye la tabla de obligatoriedad y cronograma obtenido del registro oficial.

Artículo 1.- Los sujetos pasivos enumerados a continuación, excepto las entidades públicas, deberán emitir los documentos mencionados en el artículo 2 de la Resolución No. NAC-DGERCGC12-00105, únicamente a través de mensajes de datos y firmados electrónicamente, de acuerdo con el siguiente calendario:

Grupo	Fecha de inicio	Sujetos pasivos
1	A partir del 01 de junio de 2014	<ul style="list-style-type: none">• Instituciones financieras bajo el control de la Superintendencia de Bancos y Seguros, excepto mutualistas de ahorro y crédito para la vivienda y sociedades emisoras y administradoras de tarjetas de crédito.• Contribuyentes especiales que realicen, según su inscripción en el RUC, actividades económicas correspondientes al sector y subsector: telecomunicaciones y televisión pagada, respectivamente.
2	A partir del 01 de agosto de 2014	<ul style="list-style-type: none">• Sociedades emisoras y administradoras de tarjetas de crédito.
3	A partir del 01 de octubre de 2014	<ul style="list-style-type: none">• Exportadores calificados por el SRI como contribuyentes especiales.
4	A partir del 01 de enero de 2015	<ul style="list-style-type: none">• Los demás contribuyentes especiales no señalados en los tres grupos anteriores.• Contribuyentes que posean autorización de impresión de comprobantes de venta, retención y documentos complementarios, a través de sistemas computarizados (autoimpresores).• Contribuyentes que realicen ventas a través de internet.• Los sujetos pasivos que realicen actividades económicas de exportación.

Diagrama 45: registro oficial de obligados a emitir comprobantes electrónicos.

Fuente: Resolución NAC-DGERCGC13-00236, Registro Oficial 956 de 17 de Mayo de 2013

Con lo anteriormente expuesto se resuelve que el restaurante “La Fogata” no es sujeto obligado a emitir comprobantes electrónicos.

CAPITULO 3. Diseño

3.1 Estructura del producto.-

3.1.1 Estructura del producto.-

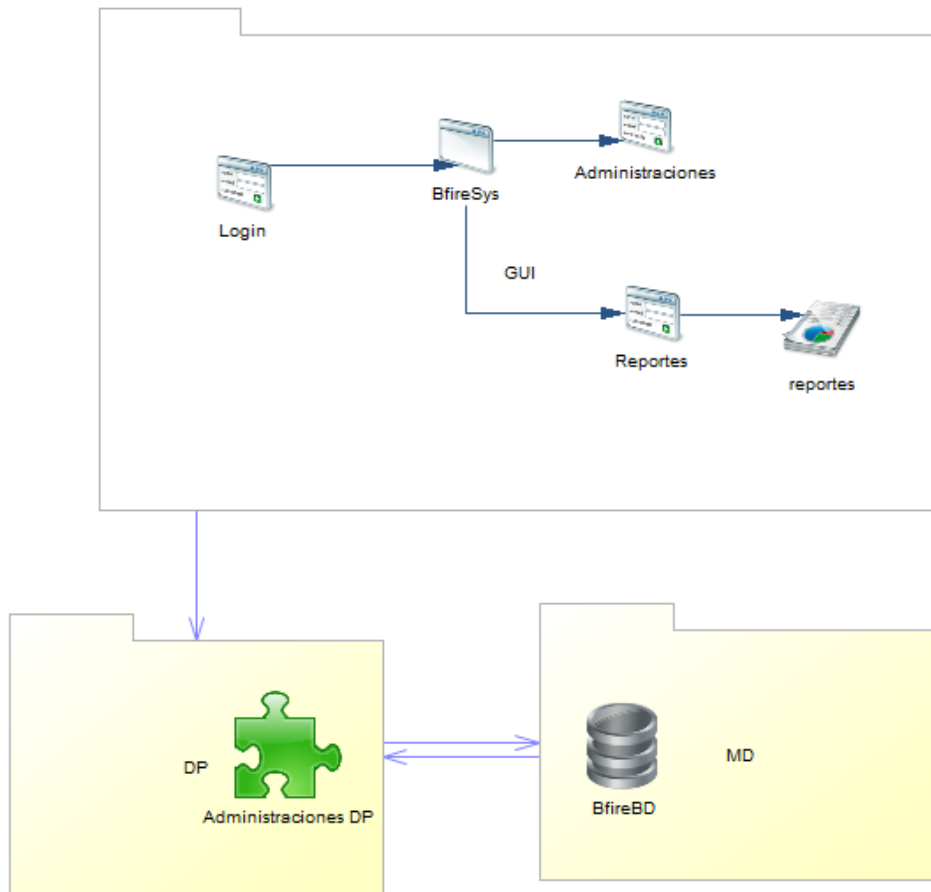


Diagrama 46: Estructura del producto

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.2 Diseño de interfaz

3.2.1 Prototipo de interfaz.-

The image shows a screenshot of a software application window titled "BonfireSys". The window has a menu bar with "Archivo", "Administrar", "Reportes", and "Ayuda". The main content area is titled "Compras" and contains several input fields: "Cedula/RUC", "Nombre", "Dirección", "Teléfono", "Fecha" (set to "Wed 12/02/2015"), and "Compra" (set to "1"). Below these fields is a table with columns "Can...", "Nombre", and "Precio". Under the table, there are two sections: "Pagos" and "Descripción de la factura". To the right of these sections are fields for "Iva 0%", "Iva 12.0 %", "Subtotal", "IVA", and "Total", along with a checked checkbox for "Efectivo". At the bottom of the window is a toolbar with icons for a green checkmark, a circular arrow, a person icon, a printer, an eraser, and a red X.

Diagrama 47: Prototipo de Interfaz

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.3 Diagrama de Clases

3.3.1 Diagramas de Clases

3.3.1.1 Diagrama de Clases de Interfaz

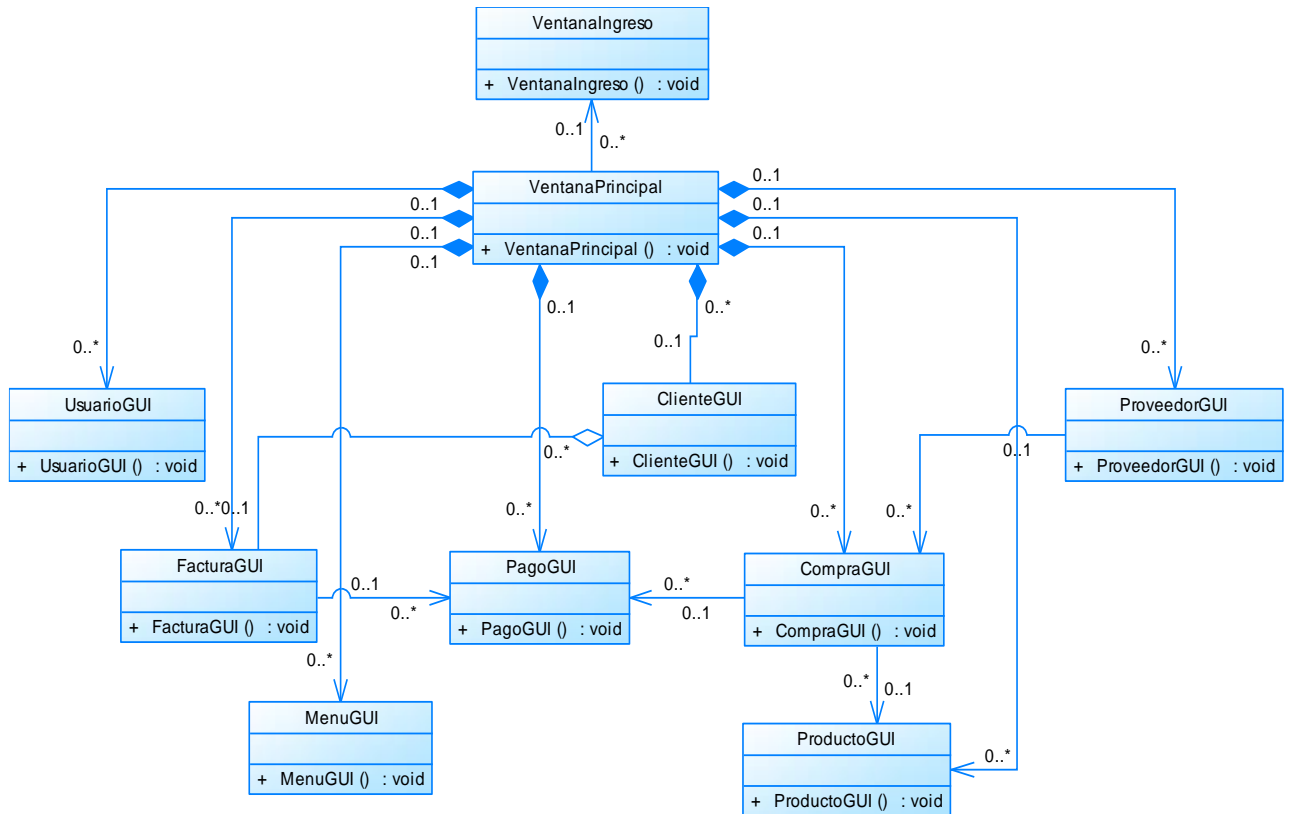


Diagrama 48: Diagrama de Clases de Interfaz

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.3.1.2 Diagrama de Clases de Dominio del Problema

UsuarioDP		
- nombre	: String	
- contraseña	: String	
- nivel	: int	
<hr/>		
+	verificar ()	: boolean
+	ingresar ()	: void
+	modificar ()	: void
+	eliminar ()	: void
+	usuarioMD ()	
+	<<Constructor>> UsuarioDP ()	
+	<<Getter>> getnombre ()	: String
+	<<Setter>> setnombre (String newNombre)	: void
+	<<Getter>> getcontrasena ()	: String
+	<<Setter>> setcontrasena (String newContrasena)	: void
+	<<Getter>> getnivel ()	: int
+	<<Setter>> setnivel (int newNivel)	: void

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

MenuDP		
- nombre	: String	
- precio	: double	
- descripcion	: String	
- iva	: boolean	
+ verificar ()		: boolean
+ ingresar ()		: void
+ modificar ()		: void
+ eliminar ()		: void
+ menuMD ()		
+ <<Getter>> getnombre ()		: String
+ <<Setter>> setnombre (String newName)		: void
+ <<Getter>> getprecio ()		: double
+ <<Setter>> setprecio (double newPrecio)		: void
+ <<Getter>> getdescripcion ()		: String
+ <<Setter>> setdescripcion (String newDescripcion)		: void
+ <<Getter>> isiva ()		: boolean
+ <<Setter>> setiva (boolean newIva)		: void
+ <<Constructor>> MenuDP ()		

ClienteDP		
- codigo	: int	
- nombre	: String	
- ruc	: String	
- direccion	: String	
- telefono	: String	
- mail	: String	
+ verificar ()		: boolean
+ ingresar ()		: void
+ modificar ()		: void
+ eliminar ()		: void
+ clienteMD ()		
+ <<Constructor>> ClienteDP ()		
+ <<Getter>> getcodigo ()		: int
+ <<Setter>> setcodigo (int newCodigo)		: void
+ <<Getter>> getnombre ()		: String
+ <<Setter>> setnombre (String newName)		: void
+ <<Getter>> getruc ()		: String
+ <<Setter>> setruc (String newRuc)		: void
+ <<Getter>> getdireccion ()		: String
+ <<Setter>> setdireccion (String newDireccion)		: void
+ <<Getter>> gettelefono ()		: String
+ <<Setter>> settelefono (String newTelefono)		: void
+ <<Getter>> getmail ()		: String
+ <<Setter>> setmail (String newMail)		: void

ProductoDP		
- nombre	: String	
- precio	: double	
- descripcion	: String	
- iva	: boolean	
+ verificar ()		: boolean
+ ingresar ()		: void
+ modificar ()		: void
+ eliminar ()		: void
+ productoMD ()		
+ <<Getter>> getnombre ()		: String
+ <<Setter>> setnombre (String newName)		: void
+ <<Getter>> getprecio ()		: double
+ <<Setter>> setprecio (double newPrecio)		: void
+ <<Getter>> getdescripcion ()		: String
+ <<Setter>> setdescripcion (String newDescripcion)		: void
+ <<Getter>> isiva ()		: boolean
+ <<Setter>> setiva (boolean newIva)		: void
+ <<Constructor>> ProductoDP ()		

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

FacturaDP		
-	codigo	: int
-	fecha	: Date
-	iva0	: double
-	iva12	: double
-	iva	: double
-	descuento	: double
-	total	: double
+	verificar ()	: boolean
+	ingresar ()	: void
+	modificar ()	: void
+	eliminar ()	: void
+	facturaMD ()	
+	<<Getter>> getcodigo ()	: int
+	<<Setter>> setcodigo (int newCodigo)	: void
+	<<Getter>> getiva0 ()	: double
+	<<Setter>> setiva0 (double newIva0)	: void
+	<<Getter>> getiva12 ()	: double
+	<<Setter>> setiva12 (double newIva12)	: void
+	<<Getter>> getiva ()	: double
+	<<Setter>> setiva (double newIva)	: void
+	<<Getter>> getdescuento ()	: double
+	<<Setter>> setdescuento (double newDescuento)	: void
+	<<Getter>> gettotal ()	: double
+	<<Setter>> settotal (double newTotal)	: void
+	<<Constructor>> FacturaDP ()	
+	<<Getter>> getfecha ()	: Date
+	<<Setter>> setfecha (Date newFecha)	: void

CompraDP		
-	codigo	: int
-	fecha	: Date
-	iva0	: double
-	iva12	: double
-	iva	: double
-	total	: double
+	verificar ()	: boolean
+	ingresar ()	: void
+	modificar ()	: void
+	eliminar ()	: void
+	compraMD ()	
+	<<Getter>> getcodigo ()	: int
+	<<Setter>> setcodigo (int newCodigo)	: void
+	<<Getter>> getfecha ()	: Date
+	<<Setter>> setfecha (Date newFecha)	: void
+	<<Getter>> getiva0 ()	: double
+	<<Setter>> setiva0 (double newIva0)	: void
+	<<Getter>> getiva12 ()	: double
+	<<Setter>> setiva12 (double newIva12)	: void
+	<<Getter>> getiva ()	: double
+	<<Setter>> setiva (double newIva)	: void
+	<<Getter>> gettotal ()	: double
+	<<Setter>> settotal (double newTotal)	: void
+	<<Constructor>> CompraDP ()	

ProveedorDP		
-	codigo	: int
-	nombre	: String
-	ruc	: String
-	direccion	: String
-	telefono	: String
-	mail	: String
-	descripcion	: String
+	verificar ()	: boolean
+	ingresar ()	: void
+	modificar ()	: void
+	proveedorMD ()	
+	eliminar ()	: void
+	<<Getter>> getcodigo ()	: int
+	<<Setter>> setcodigo (int newCodigo)	: void
+	<<Getter>> getnombre ()	: String
+	<<Setter>> setnombre (String newNombre)	: void
+	<<Getter>> getruc ()	: String
+	<<Setter>> setruc (String newRuc)	: void
+	<<Getter>> getdireccion ()	: String
+	<<Setter>> setdireccion (String newDireccion)	: void
+	<<Getter>> gettelefono ()	: String
+	<<Setter>> settelefono (String newTelefono)	: void
+	<<Getter>> getmail ()	: String
+	<<Setter>> setmail (String newMail)	: void
+	<<Getter>> getdescripcion ()	: String
+	<<Setter>> setdescripcion (String newDescripcion)	: void
+	<<Constructor>> ProveedorDP ()	

Diagrama 49: Diagrama de Clases de Dominio del Problema

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.3.1.3 Diagrama de Clases de Manejo de Datos

UsuarioMD		
+	usuarioDP ()	
+	verificar ()	: boolean
+	ingresar ()	: void
+	modificar ()	: void
+	eliminar ()	: void
+	<<Constructor>> UsuarioMD ()	

FacturaMD		
+	facturaDP ()	
+	verificar ()	: boolean
+	ingresar ()	: void
+	modificar ()	: void
+	eliminar ()	: void
+	<<Constructor>> FacturaMD ()	

CompraMD		
+	compraDP ()	
+	verificar ()	: boolean
+	ingresar ()	: void
+	modificar ()	: void
+	eliminar ()	: void
+	<<Constructor>> CompraMD ()	

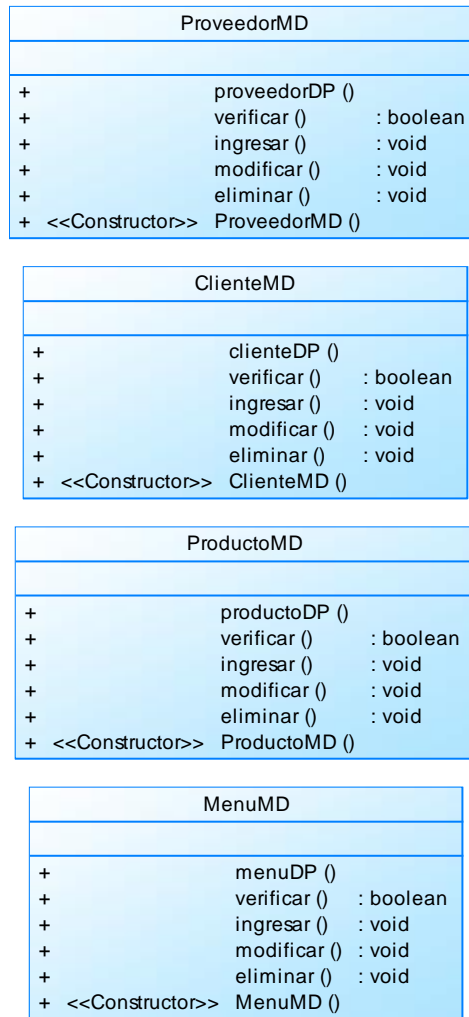


Diagrama 50: Diagrama de Clases de Manejo de Datos

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.4 Diagrama de Secuencia

3.4.1 Diagrama de Secuencias de Ingreso al Sistema

Caso de Uso: Ingreso al Sistema

Identificador: F0

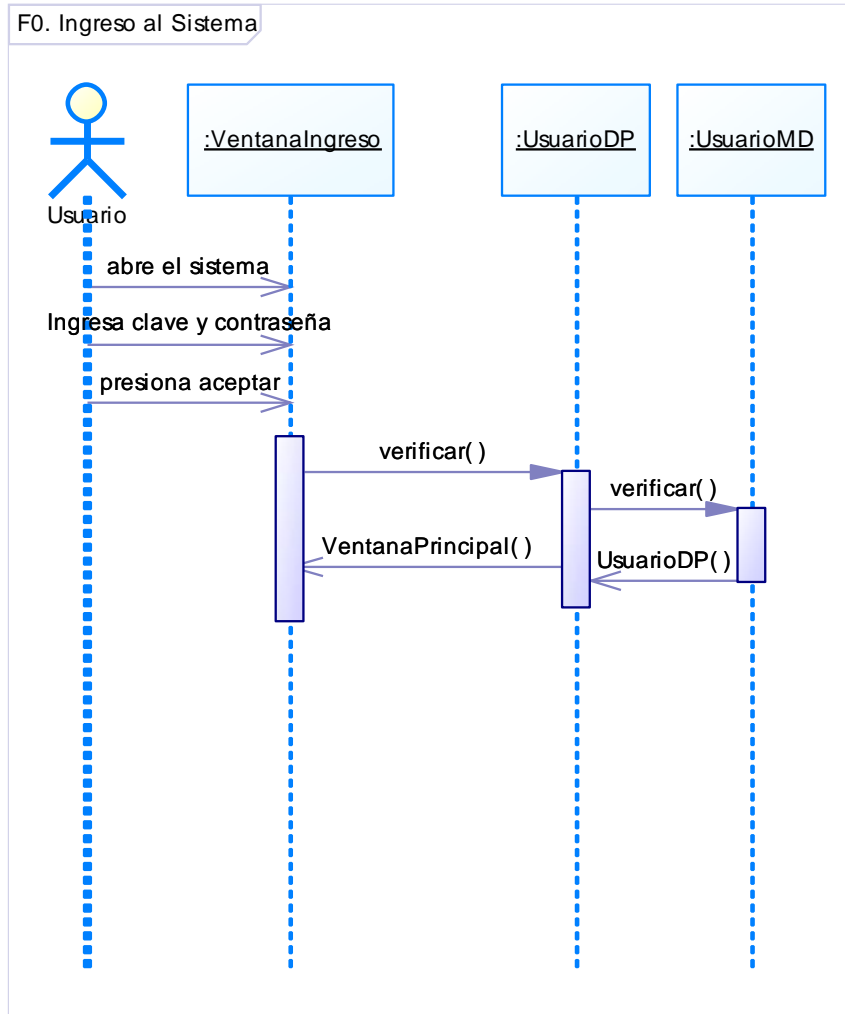


Diagrama 51: Diagrama de Secuencias de Ingreso al Sistema

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.4.2 Diagrama de Secuencias de Administración de Usuarios

Caso de Uso: Administración de Usuarios

Identificador: F1

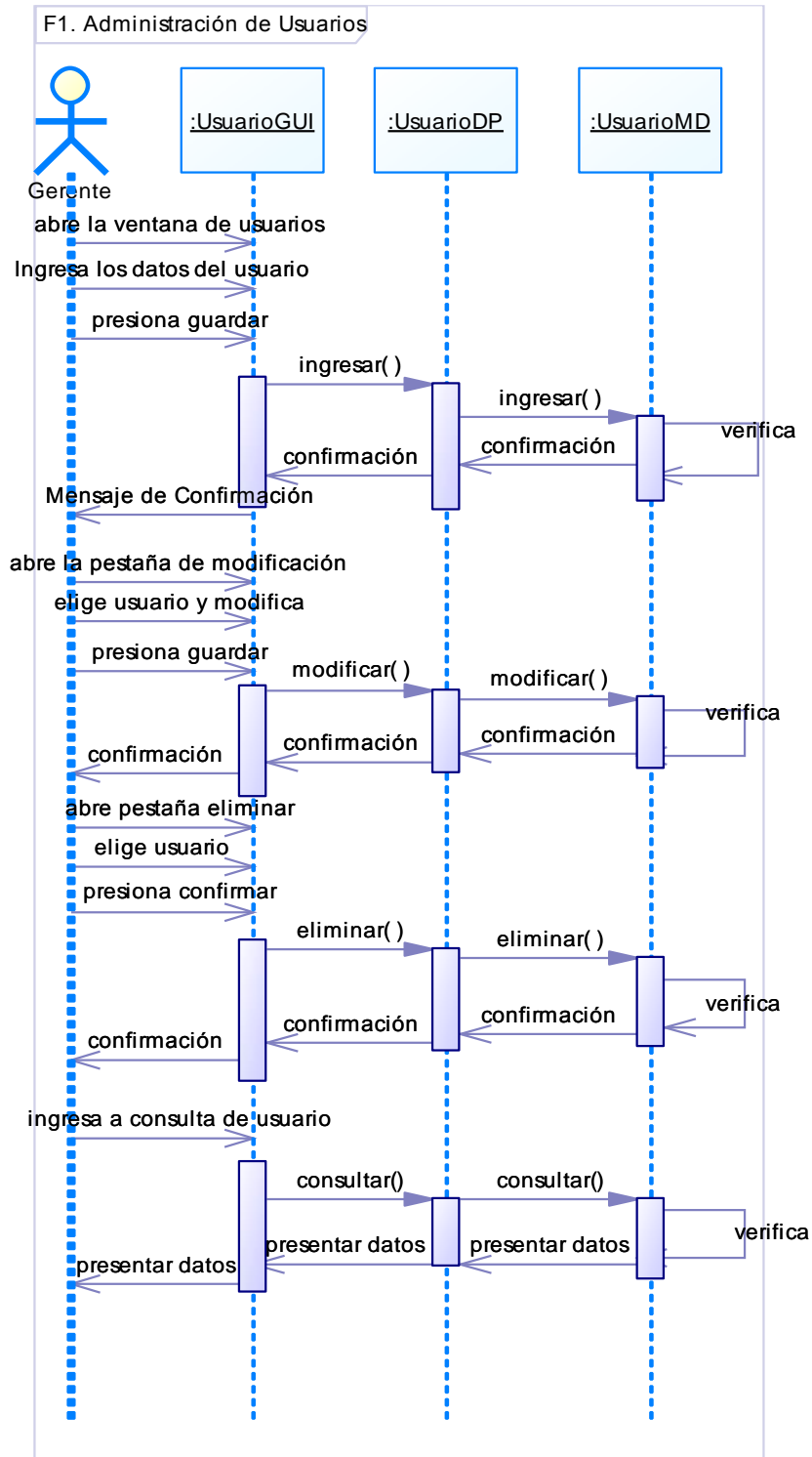


Diagrama 52: Diagrama de Secuencias de Administración de Usuarios

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.4.3 Diagrama de Secuencias de Administración de Clientes

Caso de Uso: Administración de Clientes

Identificador: F2

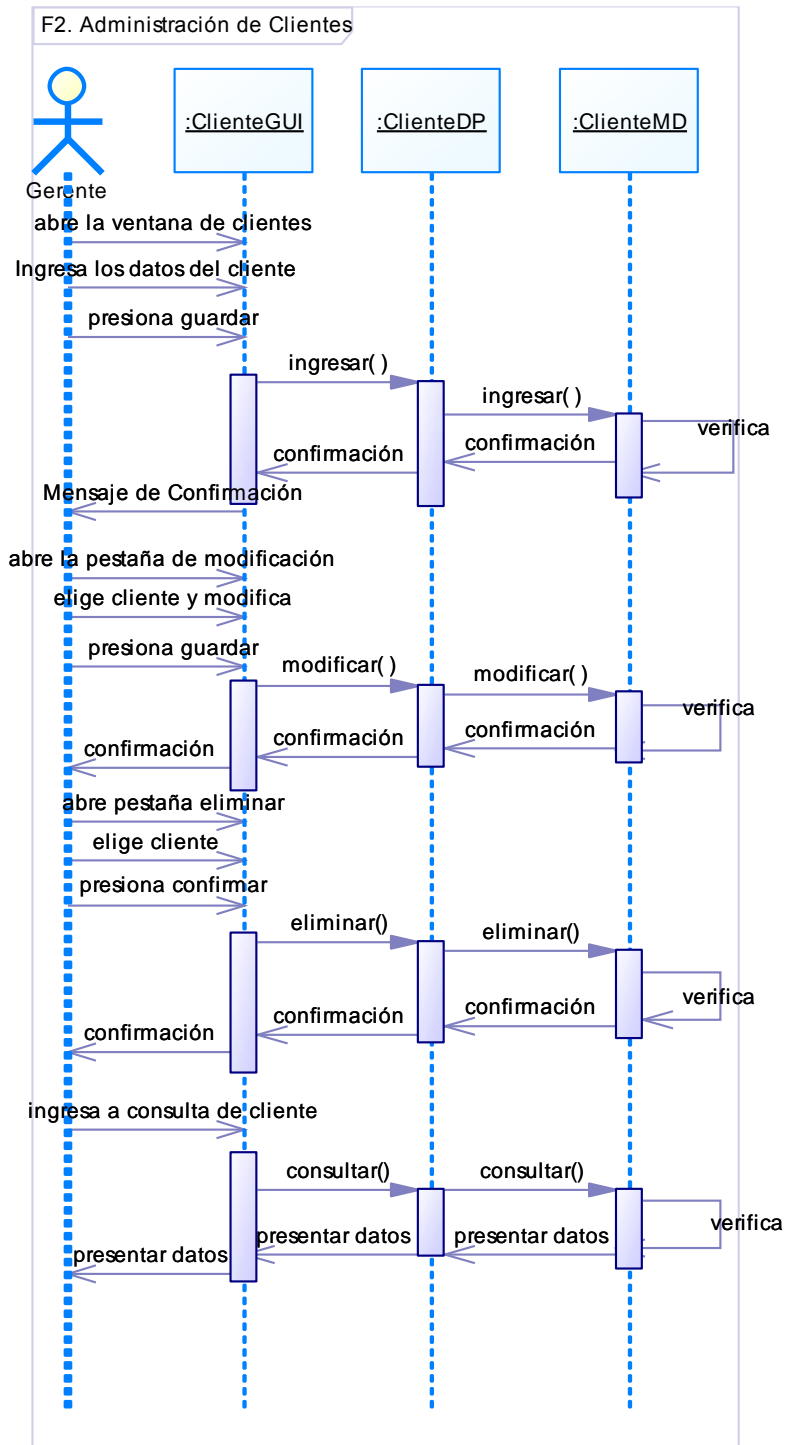


Diagrama 53: Diagrama de Secuencias de Administración de Clientes

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.4.4 Diagrama de Secuencias de Administración de Proveedores

Caso de Uso: Administración de Proveedores

Identificador: F3

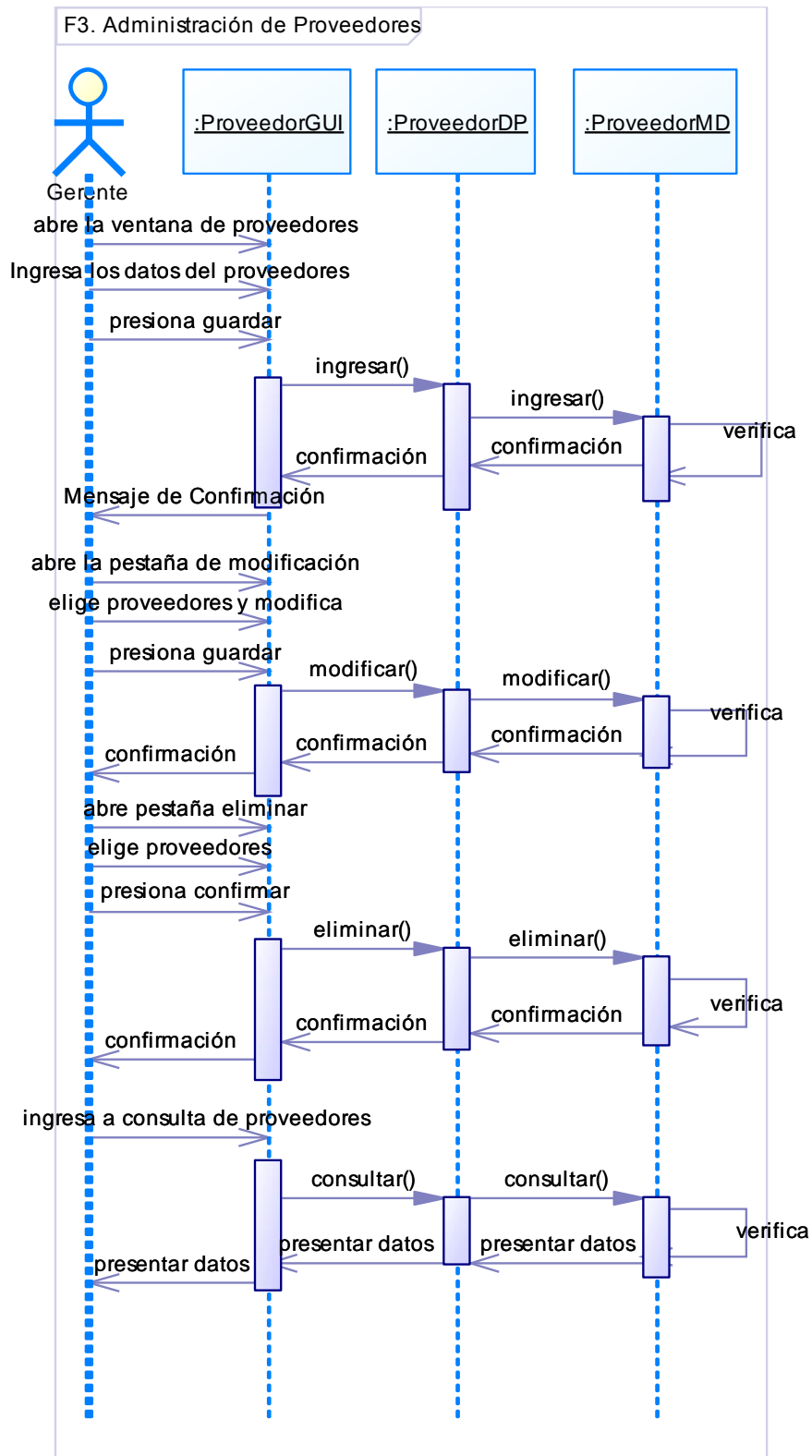


Diagrama 54: Diagrama de Secuencias de Administración de Proveedores

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.4.5 Diagrama de Secuencias de Administración de Productos

Caso de Uso: Administración de Productos

Identificador: F4

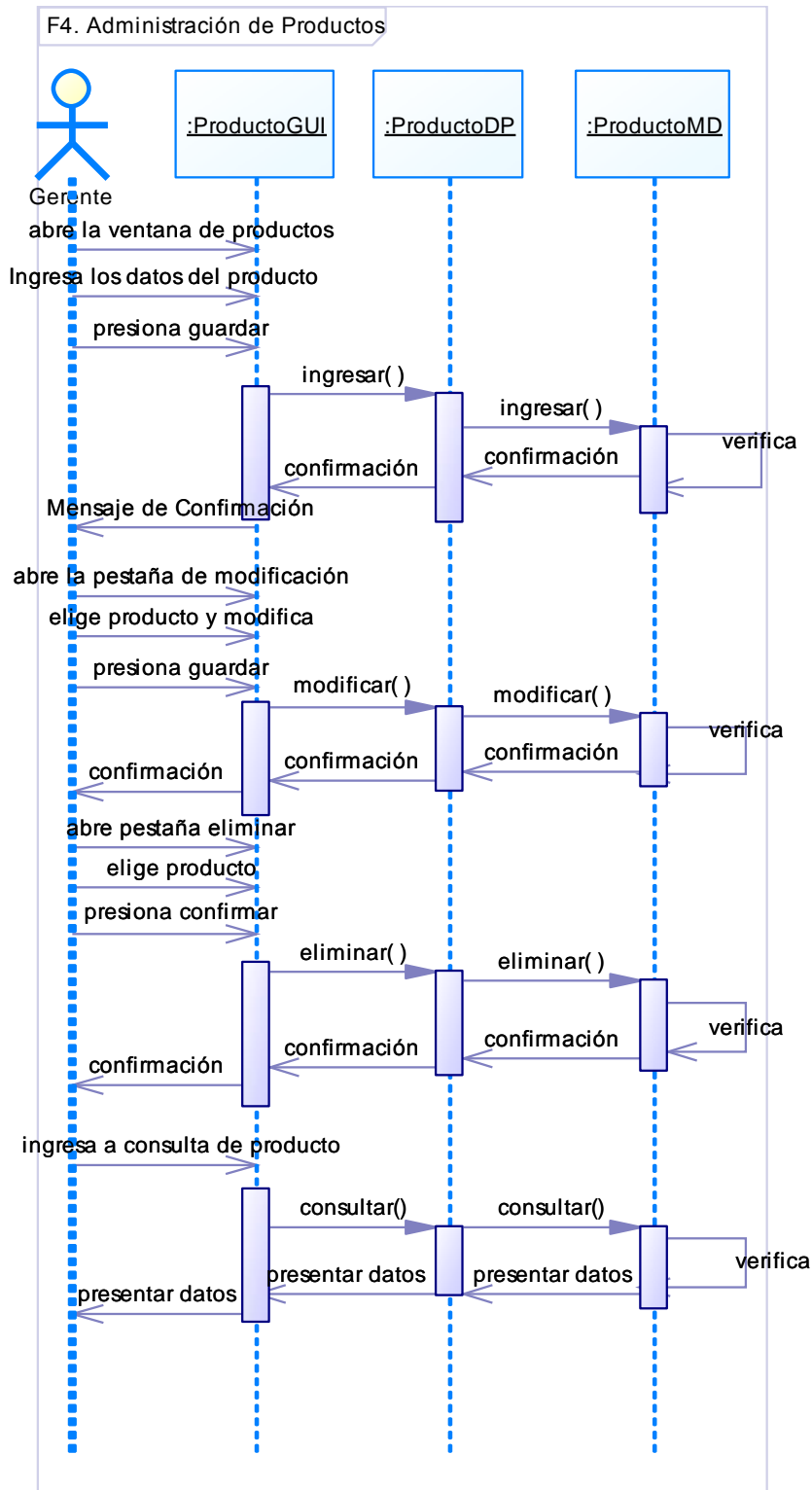


Diagrama 55: Diagrama de Secuencias de Administración de Productos

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.4.6 Diagrama de Secuencias de Administración de Menús

Caso de Uso: Administración de Menús

Identificador: F5

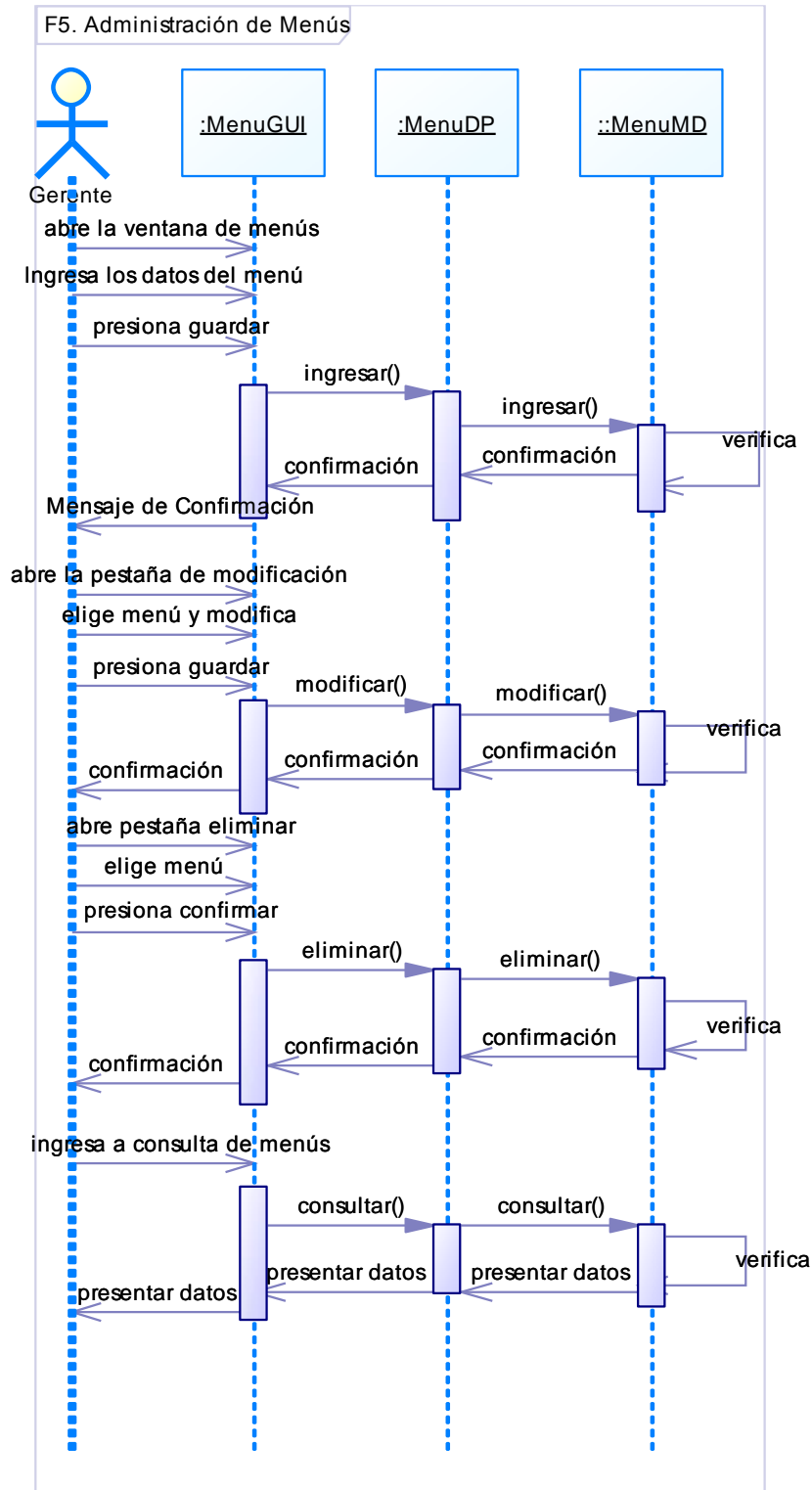


Diagrama 56: Diagrama de Secuencias de Administración de Menús

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.4.7 Diagrama de Secuencias de Administración de Pagos

Caso de Uso: Administración de Pagos

Identificador: F6

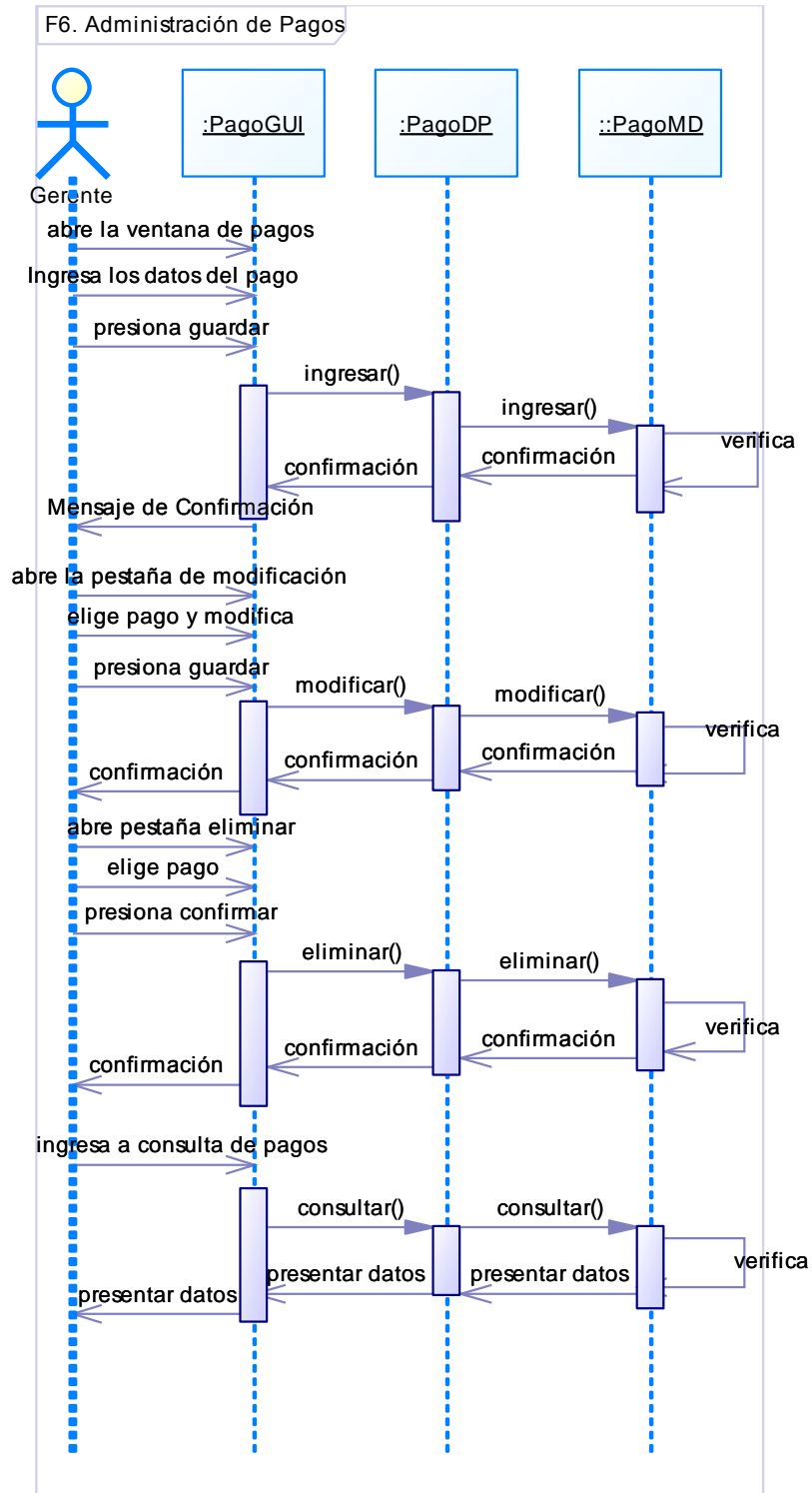


Diagrama 57: Diagrama de Secuencias de Administración de Pagos

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.4.8 Diagrama de Secuencias de Administración de Facturación

Caso de Uso: Administración de Facturación

Identificador: F7

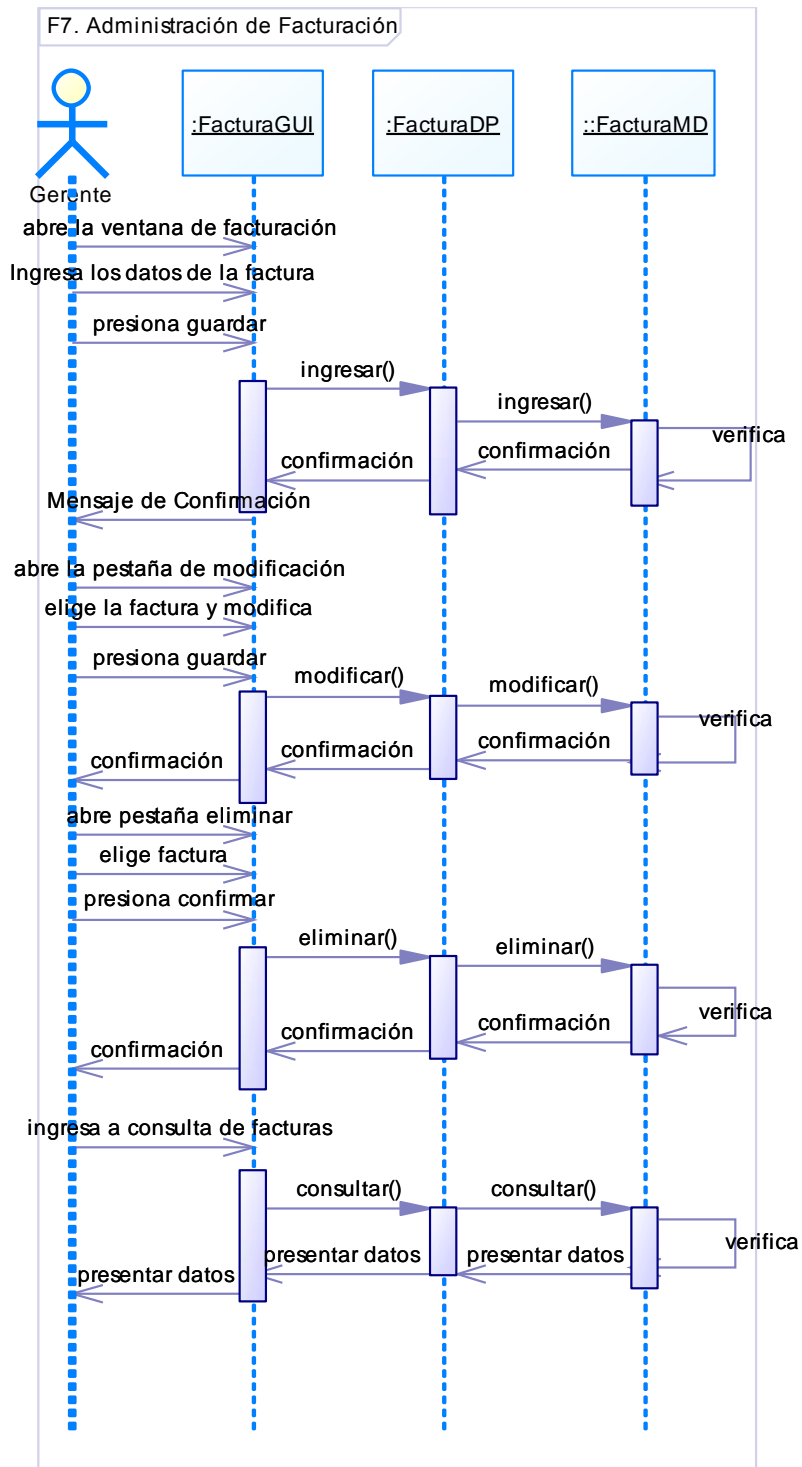


Diagrama 58: Diagrama de Secuencias de Administración de Facturación

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.4.9 Diagrama de Secuencias de Administración de Compras

Caso de Uso: Administración de Compras

Identificador: F8

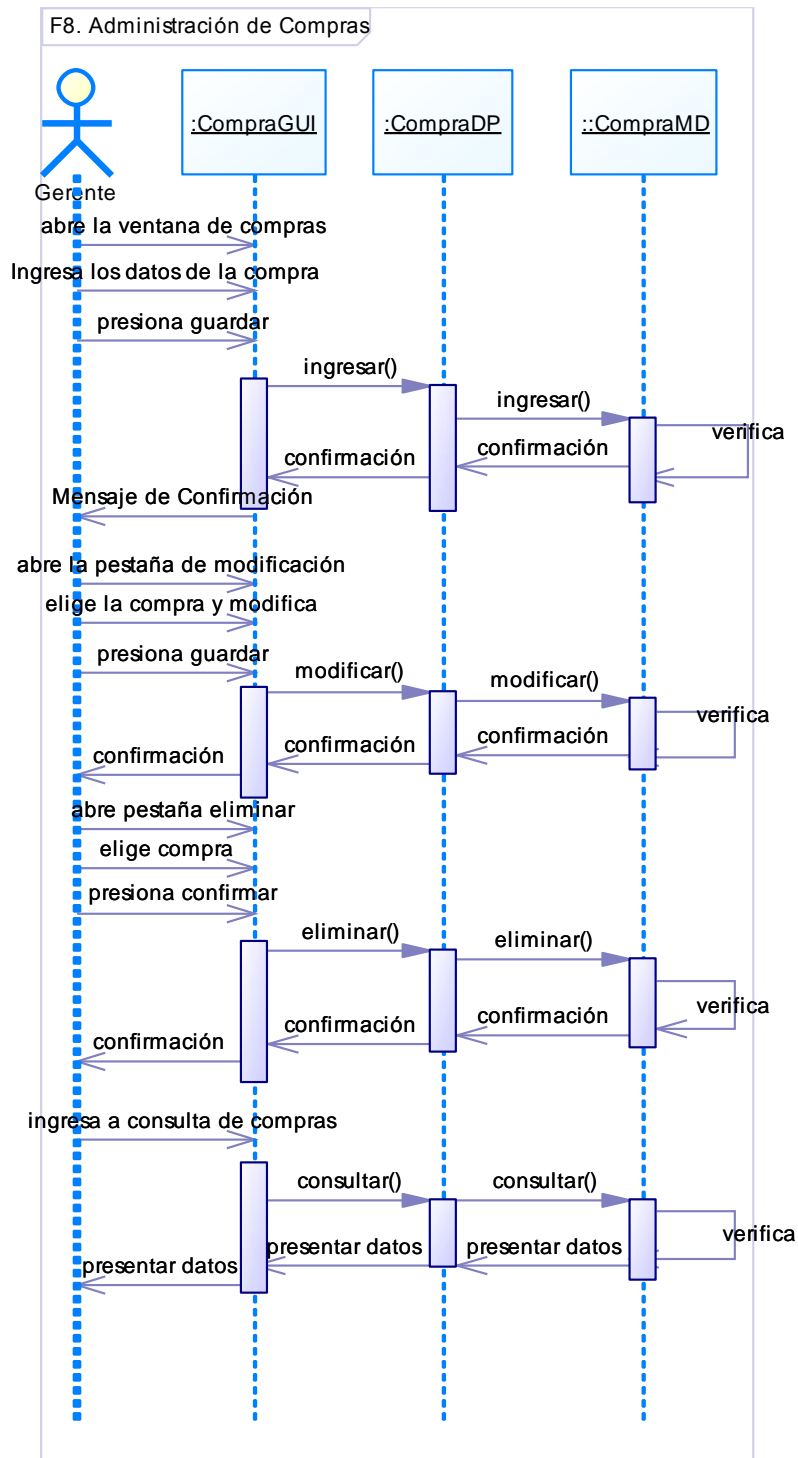


Diagrama 59: Diagrama de Secuencias de Administración de Compras

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.4.10 Diagrama de Secuencias de Administración de General

Caso de Uso: Administración General

Identificador: F9

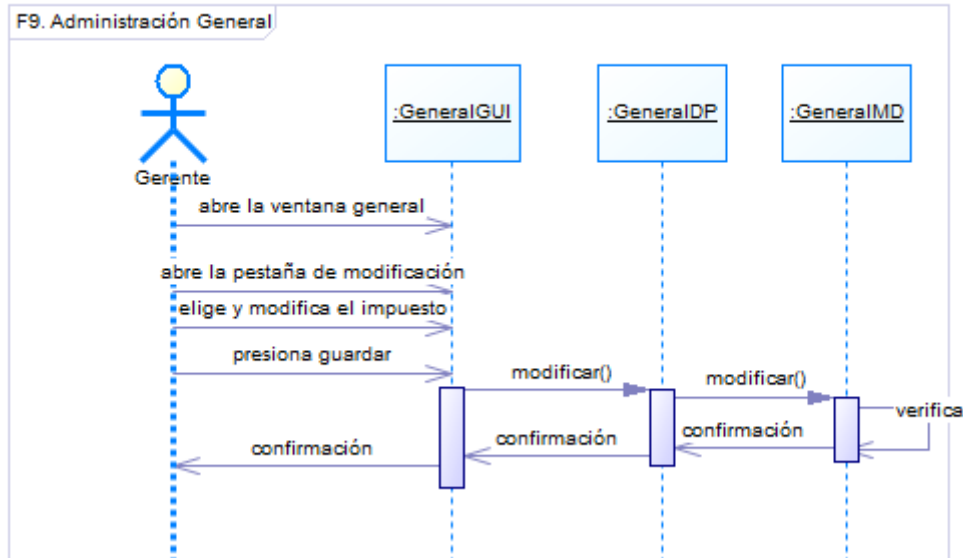


Diagrama 60: Diagrama de Secuencias de Administración General

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.5 Diagrama Entidad Relación

3.5.1 Modelo conceptual.-

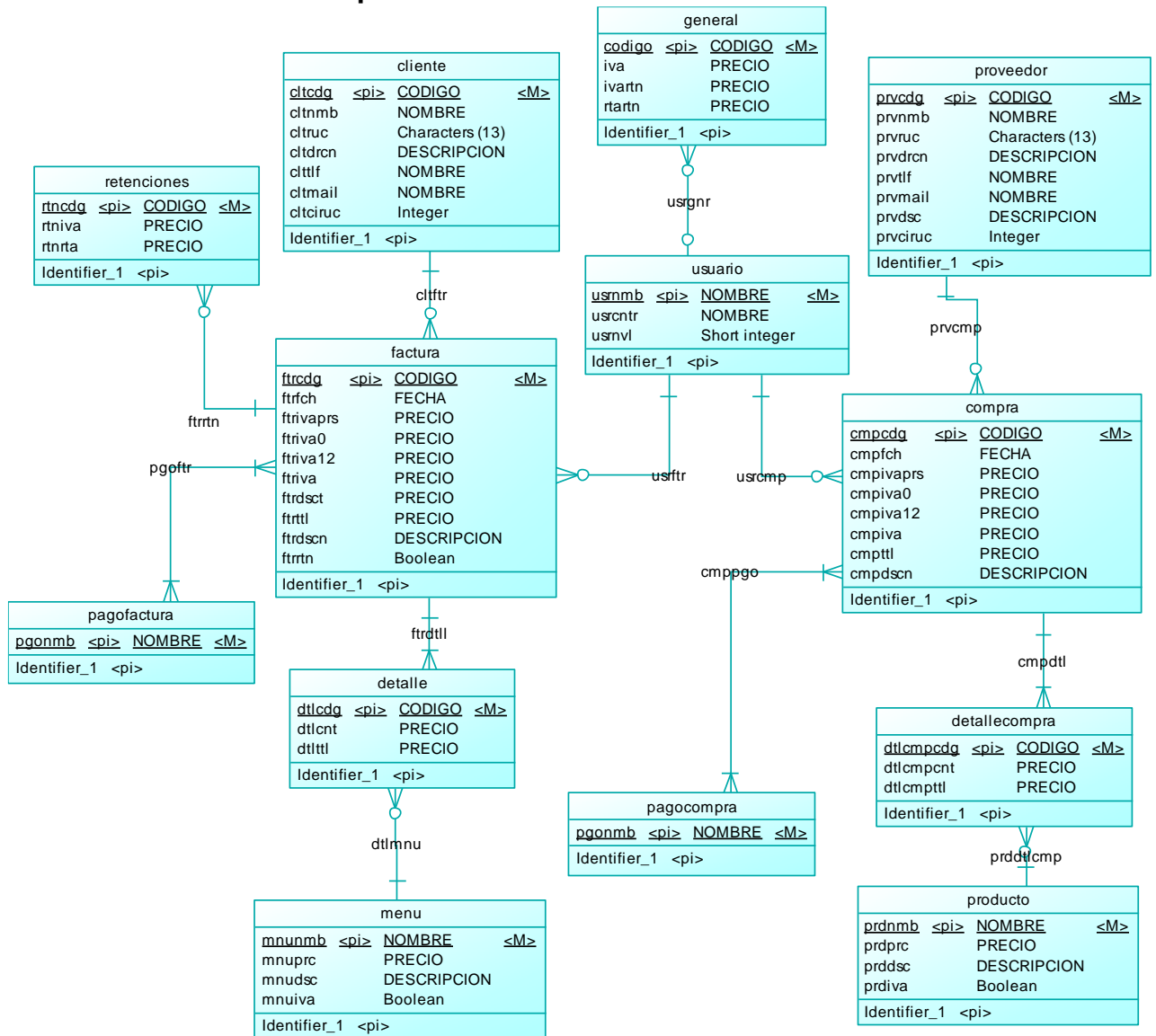


Diagrama 61: Diagrama de modelo conceptual

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.5.1 Modelo físico.-

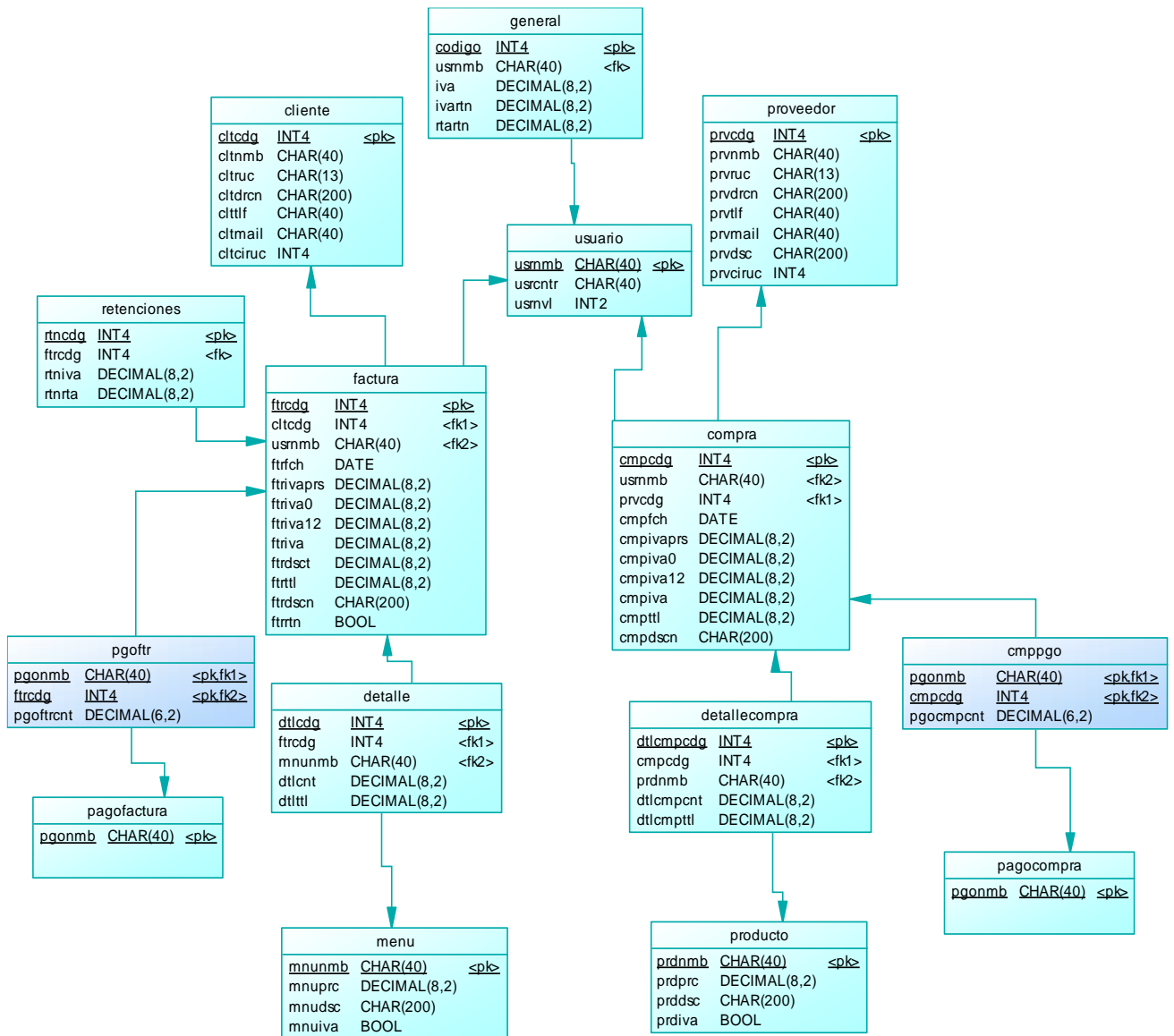


Diagrama 62: Diagrama de modelo físico

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.6 Diccionario de datos

3.6.1 Lista de dominios.-

	Name	Code	Com	Data Type	Length	Preci	M	Test
1	codigo	CODIGO		INT4	4		<input type="checkbox"/>	<None
2	descripcion	DESCRIPCION		CHAR(200)	200		<input type="checkbox"/>	<None
3	fecha	FECHA		DATE			<input type="checkbox"/>	<None
4	nombre	NOMBRE		CHAR(40)	40		<input type="checkbox"/>	<None
5	precio	PRECIO		DECIMAL(8,2)	8	2	<input type="checkbox"/>	<None

Diagrama 63: Diagrama de lista de dominios

Elaborado por: Osvaldo Tapia

Fecha: 15/05/2015

3.6.2 Diccionario de datos.-

Nombre	Código	Tipo de Dato	Longitud de caracteres	Número de decimales	Descripción
cltcdg	CLTCDG	Integer			Código del cliente
cltdrcn	CLTDRCN	Characters (200)	200		Dirección del cliente
cltmail	CLTMAIL	Characters (40)	40		Correo electrónico del cliente
cltnmb	CLTNMB	Characters (40)	40		Nombre del cliente
cltruc	CLTRUC	Characters (13)	13		Cedula o Ruc del cliente
cltllf	CLTTLF	Characters (40)	40		Teléfono del cliente
cmpcdg	CMPCDG	Integer			Código de compra
cmpdscn	CMPDSCN	Characters (200)	200		Descripción de la compra
cmpfch	CMPFCH	Date			Fecha de compra
cmpiva	CMPIVA	Decimal (8,2)	8	2	IVA en la compra
cmpiva0	CMPIVA0	Decimal (8,2)	8	2	IVA tarifa 0 en la compra
cmpiva12	CMPIVA12	Decimal (8,2)	8	2	IVA gravado en la compra
cmpivaprs	CMPIVAPRS	Decimal (8,2)	8	2	Porcentaje de IVA en la compra

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

cmpttl	CMPTTL	Decimal (8,2)	8	2	Total de la compra
dtlcdg	DTLCDG	Integer			Código del detalle de la factura
dtlcmpcdg	DTLCMPCDG	Integer			Código del detalle de la compra
dtlcmpcnt	DTLCMPCNT	Decimal (8,2)	8	2	Cantidad del detalle de la compra
dtlcmpttl	DTLCMPTTL	Decimal (8,2)	8	2	Total del detalle de la compra
dtlcnt	DTLCNT	Decimal (8,2)	8	2	Cantidad del detalle de la factura
dtlttl	DTLTTL	Decimal (8,2)	8	2	Total del detalle de la factura
ftrcdg	FTRCDG	Integer			Código de la factura
ftrdscn	FTRDSCN	Characters (200)	200		Descripción de la factura
ftdsct	FTRDST	Decimal (8,2)	8	2	Descuento de la factura
ftfch	FTRFCH	Date			Fecha de la factura
ftriva	FTRIVA	Decimal (8,2)	8	2	IVA en la factura
ftriva0	FTRIVA0	Decimal (8,2)	8	2	IVA tarifa 0 en la factura
ftriva12	FTRIVA12	Decimal (8,2)	8	2	IVA tarifa 12 en la factura
ftrivaps	FTRIVAPRS	Decimal (8,2)	8	2	Porcentaje de IVA en la factura
ftrrtn	FTRRTN	Boolean			Indicador si lleva o no retención la factura
ftrttl	FTRTTL	Decimal (8,2)	8	2	Total de la factura
mnudsc	MNUDSC	Characters (200)	200		Descripción del menú o plato
mnuiva	MNUIVA	Boolean			Indicador si lleva o no IVA el menú o plato
mnunmb	MNUNMB	Characters (40)	40		Nombre del menú o plato
mnuprc	MNUPRC	Decimal (8,2)	8		Precio del menú
pgonmb	PGONMB	Characters (40)	40		Nombre del pago en las compras
pgonmb	PGONMB2	Characters (40)	40	2	Nombre del pago en las facturas
pgoval	PGOVAL	Decimal (8,2)	8		Precio del pago
prddsc	PRDDSC	Characters (200)	200	2	Descripción del producto
prdiva	PRDIVA	Boolean			Indicador si lleva o no IVA el producto
prdnmb	PRDNMB	Characters (40)	40		Nombre del producto
prdprc	PRDPRC	Decimal (8,2)	8		Precio del producto
prvcdg	PRVCDG	Integer			Código del proveedor
prvrcn	PRVRCN	Characters (200)	200		Dirección del proveedor

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

prvdsc	PRVDSC	Characters (200)	200		Descripción del proveedor
prvmail	PRVMAIL	Characters (40)	40		Correo electrónico del proveedor
prvnmb	PRVNMB	Characters (40)	40		Nombre del proveedor
prvruc	PRVRUC	Characters (13)	13		RUC o Cedula del proveedor
prvtf	PRVTLF	Characters (40)	40		Teléfono del proveedor
rtncdg	RTNCDG	Integer			Retención en la renta
rtniva	RTNIVA	Decimal (8,2)	8	2	Código de la retención
rtnrta	RTNRTA	Decimal (8,2)	8		Retención del IVA
usrctr	USRCNTR	Characters (40)	40	2	Contraseña del usuario
usrnmb	USRNMB	Characters (40)	40	2	Nombre del usuario
usrnvl	USRNVL	integer			Nivel del usuario

Tabla 62: Diccionario de datos

Elaborado por: Osvaldo Tapia

Fecha: 01/12/2015

CAPITULO 4. Desarrollo del sistema

4.1 Desarrollo

4.1.1 Definición de clases.-

Clase: abstracción que define un tipo de objeto especificando que propiedades (atributos) y operaciones disponibles va a tener

Ejemplo:

```
private int codigo;  
private String nombre;  
private String ruc;  
private String direccion;  
private String telefono;  
private String mail;
```

 cltMD cliente = new cltMD(this); //definimos un atributo cliente con el que podremos interactuar con el manejo de datos de la clase cliente.

```
public cltDP() { //utilizamos el constructor de la clase clienteDP para inicializar sus atributos
```

```
    this.codigo =0;  
    this.nombre ="";  
    this.ruc ="";  
    this.direccion ="";  
    this.telefono ="";  
    this.mail ="";  
}
```

// Esta sección es para getter y setter es decir para establecer y tomar valores de los atributos de la clase

```
public int getCodigo() {  
    return codigo; }  
public void setCodigo(int codigo) {  
    this.codigo = codigo;  
}  
public String getNombre() {  
    return nombre; }  
}
```

```
public void setNombre(String nombre) {
    this.nombre = nombre; }
public String getRuc() {
    return ruc; }
public void setRuc(String ruc) {
    this.ruc = ruc; }
public String getDireccion() {
    return direccion; }
public void setDireccion(String direccion) {
    this.direccion = direccion; }
public String getTelefono() {
    return telefono; }
public void setTelefono(String telefono) {
    this.telefono = telefono; }
public String getMail() {
    return mail; }
public void setMail(String mail) {
    this.mail = mail; }
```

4.1.2 Codificación de clases.-

Para nuestro cometido explicaremos la codificación de la clase cliente del paquete DP (dominio del problema).

```
package dp;//nombre del paquete que contiene las clases de manejo de dominio del problema.
```

```
//apartado para los import necesarios para que funcione la clase.
```

```
import md.cltMD;
```

```
/**
```

```
*
```

```
* @author osvaldo
```

```
*/
```

```
public class cltDP { //inicio de la clase clienteDP (DP por dominio del problema)
```

//Definimos sus atributos en forma privada para que se pueda acceder a ellos a través de sus funciones.

```
private int codigo;  
private String nombre;  
private String ruc;  
private String direccion;  
private String telefono;  
private String mail;
```

 cltMD cliente = new cltMD(this); //definimos un atributo cliente con el que podremos interactuar con el manejo de datos de la clase cliente.

public cltDP() { //utilizamos el constructor de la clase clienteDP para inicializar sus atributos

```
    this.codigo =0;  
    this.nombre ="";  
    this.ruc ="";  
    this.direccion ="";  
    this.telefono ="";  
    this.mail ="";  
}
```

// Esta sección es para getter y setter es decir para establecer y tomar valores de los atributos de la clase

```
public int getCodigo() {  
    return codigo; }  
public void setCodigo(int codigo) {  
    this.codigo = codigo;  
}  
public String getNombre() {  
    return nombre; }  
public void setNombre(String nombre) {  
    this.nombre = nombre; }  
public String getRuc() {  
    return ruc; }  
public void setRuc(String ruc) {  
    this.ruc = ruc; }
```

```
public String getDireccion() {  
    return direccion; }  
public void setDireccion(String direccion) {  
    this.direccion = direccion; }  
public String getTelefono() {  
    return telefono; }  
public void setTelefono(String telefono) {  
    this.telefono = telefono; }  
public String getMail() {  
    return mail; }  
public void setMail(String mail) {  
    this.mail = mail; }
```

//En esta sección de código encontraremos las funciones adecuadas para interactuara con el paquete MD de la clase cliente. Cada clase DP tiene una clase MD.

```
public boolean verificarDP(){  
    if (cliente.verificarMD())  
        return true;  
    else  
        return false; }  
public int MaxDP(){  
return cliente.maxMD(); }  
public void insertarDP(){  
    cliente.insertarMD(); }  
public void modificarDP()  
{  
    cliente.modificarMD();  
}  
  
public void eliminarDP()  
{  
    cliente.eliminarMD();  
}  
public Object[][] datosCliente()
```

```
{  
    return cliente.obtenerCedula();  
}  
  
}
```

Extracto de código 1: Clase cliente DP

Elaborado por: Osvaldo Tapia

Fecha: 16/05/2015

4.1.3 Conexión con la base de datos.-

En el programa bonfiresys tenemos la conexión de datos en un paquete llamado utilitarios, al cual podemos acceder desde el paquete MD a través de cada una de las clases MD.

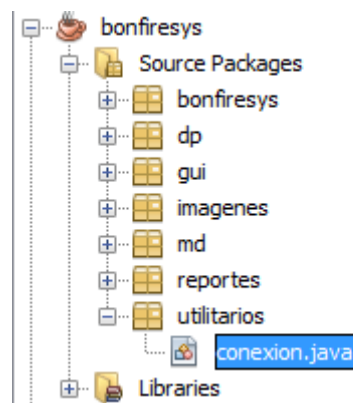


Diagrama 64: Cuadro de paquetes de bonfiresys

Elaborado por: Osvaldo Tapia

Fecha: 16/05/2015

```
package utilitarios;  
import java.sql.*;  
  
/**  
 *  
 * @author Osvaldo  
 */  
  
public class conexion {  
    Connection con;  
    Statement stmt;  
    private String driver="org.postgresql.Driver";  
    private String conex="jdbc:postgresql://localhost:5432/bfirebd";  
    //localhost:'pueto de acceso';bfirebd:nombre de la base de datos
```

```
private String user="postgres";
private String pass="postgres";
public conexion(){
    con=null;
}
public Connection conectar()
{
    try
    {

        Class.forName(driver);
    }
    catch(ClassNotFoundException e)
    {
        System.out.println(e.getMessage());
    }
    try
    {
        con=DriverManager.getConnection(conex, user, pass);
        stmt=con.createStatement();
    }
    catch(SQLException e)
    {
    }
    return con;
}
}
```

Extracto de código 2: Conexión a base de dato

Elaborado por: Osvaldo Tapia

Fecha: 16/05/2015

4.2 Pruebas

4.2.1 Plan de pruebas del sistema

4.2.1.1 Pruebas de ingreso al sistema

Casos de Prueba: F0 ingreso al sistema.

Precondiciones: Tabla usuario con datos.

Entrada	Resultado Esperado	Caso de uso	Cumple
Presenta pantalla de ingreso al sistema	Despliega plantilla de ingreso	F0	✓
Ingreso de los datos			
Correcto	Despliega plantilla principal	F0	✓
Incorrecto	Despliega mensaje de error	F0	✓

Tabla 63: Pruebas de ingreso al sistema

Elaborado por: Osvaldo Tapia

Fecha: 17/05/2015

4.2.1.2 Pruebas de administración de usuarios

Casos de Prueba: F1 administración de usuarios.

Precondiciones: pantalla principal.

Entrada	Resultado Esperado	Caso de uso	Cumple
Genera evento para ingresar al administrador de usuarios	Presenta pantalla de administración de usuarios	F1	✓
Ingreso de usuarios			
Correcto	Datos almacenados en la tabla usuarios de la base de datos	F1.1	✓
Incorrecto	Despliega mensaje de error	F1.1	✓
Modificar usuarios			
Correcto	Datos actualizados en la tabla usuarios de la base de datos	F1.2	✓
Incorrecto	Despliega mensaje de error	F1.2	✓
Eliminar usuario			
Correcto	Datos eliminados en la tabla usuarios de la base de datos	F1.3	✓
Incorrecto	Despliega mensaje de error	F1.3	✓
Consulta de usuarios			

Correcto	Se visualiza los datos deseados	F1.4	✓
Incorrecto	Despliega mensaje de error	F1.4	✓

Tabla 64: Pruebas de administración de usuarios

Elaborado por: Osvaldo Tapia

Fecha: 17/05/2015

4.2.1.3 Pruebas de administración de clientes

Casos de Prueba: F2 administración de clientes.

Precondiciones: pantalla principal.

Entrada	Resultado Esperado	Caso de uso	Cumple
Genera evento para ingresar al administrador de clientes	Presenta pantalla de administración de clientes	F2	✓
Ingreso de clientes			
Correcto	Datos almacenados en la tabla clientes de la base de datos	F2.1	✓
Incorrecto	Despliega mensaje de error	F2.1	✓
Modificar clientes			
Correcto	Datos actualizados en la tabla clientes de la base de datos	F2.2	✓
Incorrecto	Despliega mensaje de error	F2.2	✓
Eliminar cliente			
Correcto	Datos eliminados en la tabla clientes de la base de datos	F2.3	✓
Incorrecto	Despliega mensaje de error	F2.3	✓
Consulta de clientes			
Correcto	Se visualiza los datos deseados	F2.4	✓
Incorrecto	Despliega mensaje de error	F2.4	✓

Tabla 65: Pruebas de administración de clientes

Elaborado por: Osvaldo Tapia

Fecha: 17/05/2015

4.2.1.4 Pruebas de administración de proveedores

Casos de Prueba: F3 administración de proveedores.

Precondiciones: pantalla principal.

Entrada	Resultado Esperado	Caso de uso	Cumple
Genera evento para ingresar al administrador de proveedores	Presenta pantalla de administración de proveedores	F3	✓
Ingreso de proveedores			
Correcto	Datos almacenados en la tabla proveedores de la base de datos	F3.1	✓
Incorrecto	Despliega mensaje de error	F3.1	✓
Modificar proveedores			
Correcto	Datos actualizados en la tabla proveedores de la base de datos	F3.2	✓
Incorrecto	Despliega mensaje de error	F3.2	✓
Eliminar proveedores			
Correcto	Datos eliminados en la tabla proveedores de la base de datos	F3.3	✓
Incorrecto	Despliega mensaje de error	F3.3	✓
Consulta de proveedores			
Correcto	Se visualiza los datos deseados	F3.4	✓
Incorrecto	Despliega mensaje de error	F3.4	✓

Tabla 66: Pruebas de administración de proveedores

Elaborado por: Osvaldo Tapia

Fecha: 17/05/2015

4.2.1.5 Pruebas de administración de productos

Casos de Prueba: F4 administración de productos.

Precondiciones: pantalla principal.

Entrada	Resultado Esperado	Caso de uso	Cumple
Genera evento para ingresar al administrador de productos	Presenta pantalla de administración de productos	F4	✓
Ingreso de productos			

Correcto	Datos almacenados en la tabla productos de la base de datos	F4.1	✓
Incorrecto	Despliega mensaje de error	F4.1	✓
Modificar productos			
Correcto	Datos actualizados en la tabla productos de la base de datos	F4.2	✓
Incorrecto	Despliega mensaje de error	F4.2	✓
Eliminar productos			
Correcto	Datos eliminados en la tabla productos de la base de datos	F4.3	✓
Incorrecto	Despliega mensaje de error	F4.3	✓
Consulta de productos			
Correcto	Se visualiza los datos deseados	F4.4	✓
Incorrecto	Despliega mensaje de error	F4.4	✓

Tabla 67: Pruebas de administración de productos

Elaborado por: Osvaldo Tapia

Fecha: 17/05/2015

4.2.1.6 Pruebas de administración de menús

Casos de Prueba: F5 administración de menús.

Precondiciones: pantalla principal.

Entrada	Resultado Esperado	Caso de uso	Cumple
Genera evento para ingresar al administrador de menús	Presenta pantalla de administración de menús	F5	✓
Ingreso de menú			
Correcto	Datos almacenados en la tabla menús de la base de datos	F5.1	✓
Incorrecto	Despliega mensaje de error	F5.1	✓
Modificar menú			
Correcto	Datos actualizados en la tabla menús de la base de datos	F5.2	✓
Incorrecto	Despliega mensaje de error	F5.2	✓
Eliminar menú			
Correcto	Datos eliminados en la tabla menús de la base de datos	F5.3	✓

Incorrecto	Despliega mensaje de error	F5.3	✓
Consulta de menú			
Correcto	Se visualiza los datos deseados	F5.4	✓
Incorrecto	Despliega mensaje de error	F5.4	✓

Tabla 68: Pruebas de administración de menú

Elaborado por: Osvaldo Tapia

Fecha: 17/05/2015

4.2.1.7 Pruebas de administración de pagos

Casos de Prueba: F6 administración de pagos.

Precondiciones: pantalla principal.

Entrada	Resultado Esperado	Caso de uso	Cumple
Genera evento para ingresar al administrador de pagos	Presenta pantalla de administración de pagos	F6	✓
Ingreso de pagos			
Correcto	Datos almacenados en la tabla pagos de la base de datos	F6.1	✓
Incorrecto	Despliega mensaje de error	F6.1	✓
Modificar pagos			
Correcto	Datos actualizados en la tabla pagos de la base de datos	F6.2	✓
Incorrecto	Despliega mensaje de error	F6.2	✓
Eliminar pagos			
Correcto	Datos eliminados en la tabla pagos de la base de datos	F6.3	✓
Incorrecto	Despliega mensaje de error	F6.3	✓
Consulta de pagos			
Correcto	Se visualiza los datos deseados	F6.4	✓
Incorrecto	Despliega mensaje de error	F6.4	✓

Tabla 69: Pruebas de administración de pagos

Elaborado por: Osvaldo Tapia

Fecha: 17/05/2015

4.2.1.8 Pruebas de administración de factura

Casos de Prueba: F7 administración de facturación.

Precondiciones: pantalla principal, las siguientes tablas deben contener datos: cliente, usuario, menú, pago.

Entrada	Resultado Esperado	Caso de uso	Cumple
Genera evento para ingresar al administrador de facturación	Presenta pantalla de administración de facturación	F7	✓
Ingreso de factura			
Correcto	Datos almacenados en la tabla factura de la base de datos	F7.1	✓
Incorrecto	Despliega mensaje de error	F7.1	✓
Consulta de factura			
Correcto	Se visualiza los datos deseados	F7.4	✓
Incorrecto	Despliega mensaje de error	F7.4	✓

Tabla 70: Pruebas de administración de facturación

Elaborado por: Osvaldo Tapia

Fecha: 17/05/2015

4.2.1.9 Pruebas de administración de compras

Casos de Prueba: F8 administración de compras.

Precondiciones: pantalla principal, las siguientes tablas deben contener datos: proveedor, usuario, menú, pago.

Entrada	Resultado Esperado	Caso de uso	Cumple
Genera evento para ingresar al administrador de compra	Presenta pantalla de administración de compras	F8	✓
Ingreso de compra			
Correcto	Datos almacenados en la tabla compra de la base de datos	F8.1	✓
Incorrecto	Despliega mensaje de error	F8.1	✓
Consulta de compra			
Correcto	Se visualiza los datos deseados	F8.4	✓
Incorrecto	Despliega mensaje de error	F8.4	✓

Tabla 71: Pruebas de administración de compra

Elaborado por: Osvaldo Tapia

Fecha: 17/05/2015

4.2.1.10 Pruebas de administración general

Casos de Prueba: F9 administración general.

Precondiciones: pantalla principal, las siguientes tablas deben contener datos: usuario, general.

Entrada	Resultado Esperado	Caso de uso	Cumple
Genera evento para ingresar al administrador general	Presenta pantalla de administración general	F9	✓
Modificar general			
Correcto	Datos almacenados en la tabla general de la base de datos	F9.1	✓
Incorrecto	Despliega mensaje de error	F9.1	✓

Tabla 72: Pruebas de administración general

Elaborado por: Osvaldo Tapia

Fecha: 17/05/2015

CAPITULO 5. IMPLEMENTACIÓN

5.1 Despliegue

5.1.1 Software empaquetado.-

El empaquetado de software consiste en proporcionar las aplicaciones en forma de paquetes, estos paquetes están formados por los programas ejecutables de la aplicación así como por todas las bibliotecas de las que depende y otros tipos de ficheros de forma que se proporcionan como un conjunto. (Wikipedia, 2015)

Para realizar el empaquetado de nuestro programa vamos a necesitar dos programas que son de libre uso.

Inicialmente creamos el archivo .jar con el programa Netbeans 8.0.2

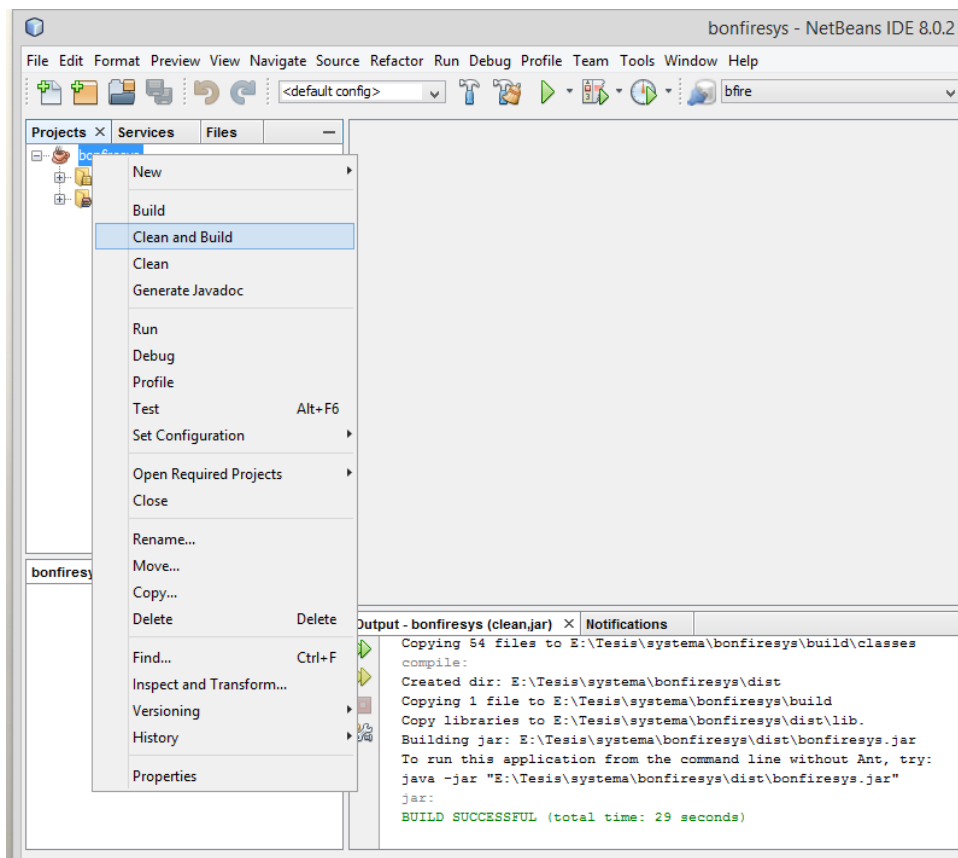


Diagrama 65: Creación del .jar

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

Ya que tenemos el .jar lo convertimos en un archivo ejecutable para Windows, lo realizaremos con **Launch4j**.- “es una herramienta multiplataforma para envolver aplicaciones Java en ejecutables nativos de Windows” (Kowal, 2015)

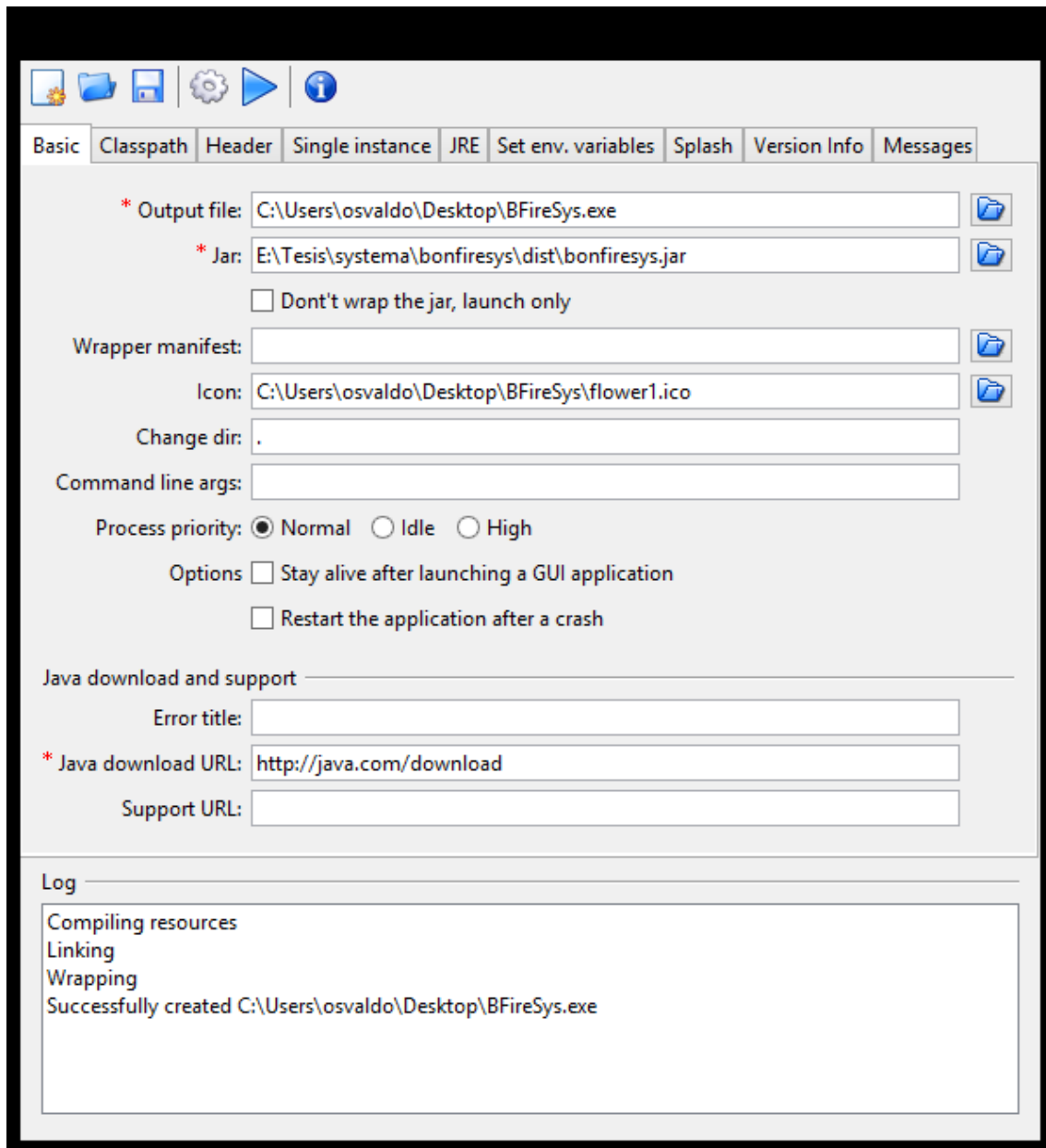


Diagrama 66: Creación de BFireSys.exe

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

Ahora que tenemos nuestro archivo BFireSys.exe creamos el instalador a partir de este archivo, para este proceso vamos a necesitar el siguiente programa Inno Setup “es un instalador libre para los programas de Windows. Presentado por primera vez en 1997, Inno Setup hoy es rival e incluso supera muchos instaladores comerciales en conjunto de características y estabilidad” (Ruseell, 2015)

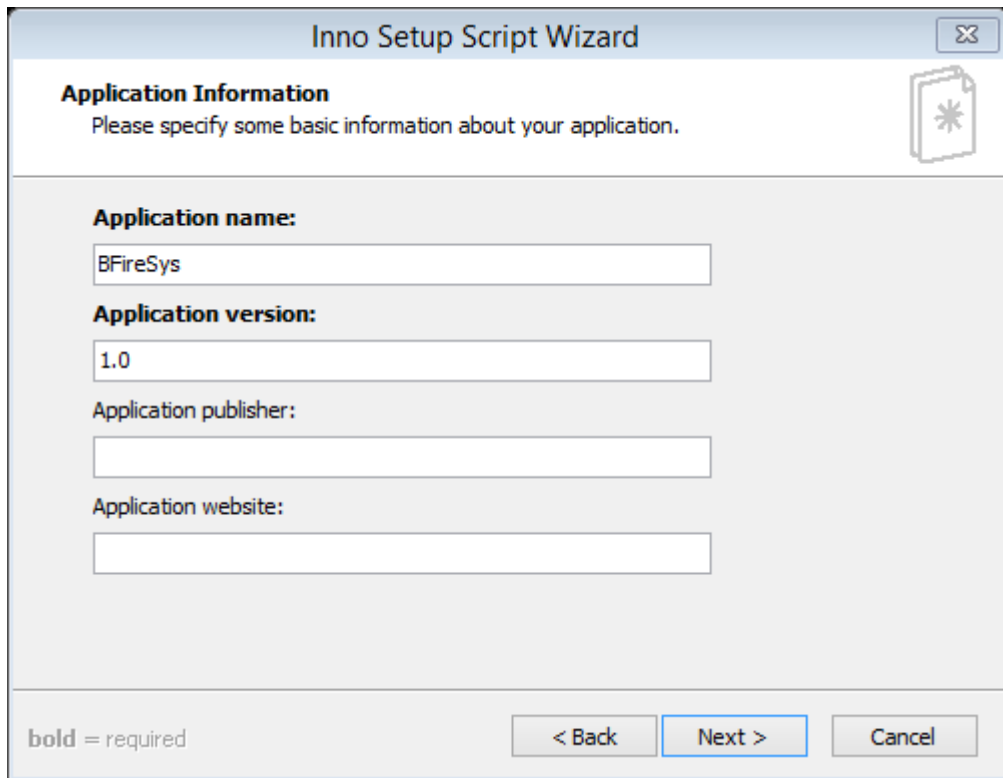


Diagrama 67: Información básica de BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

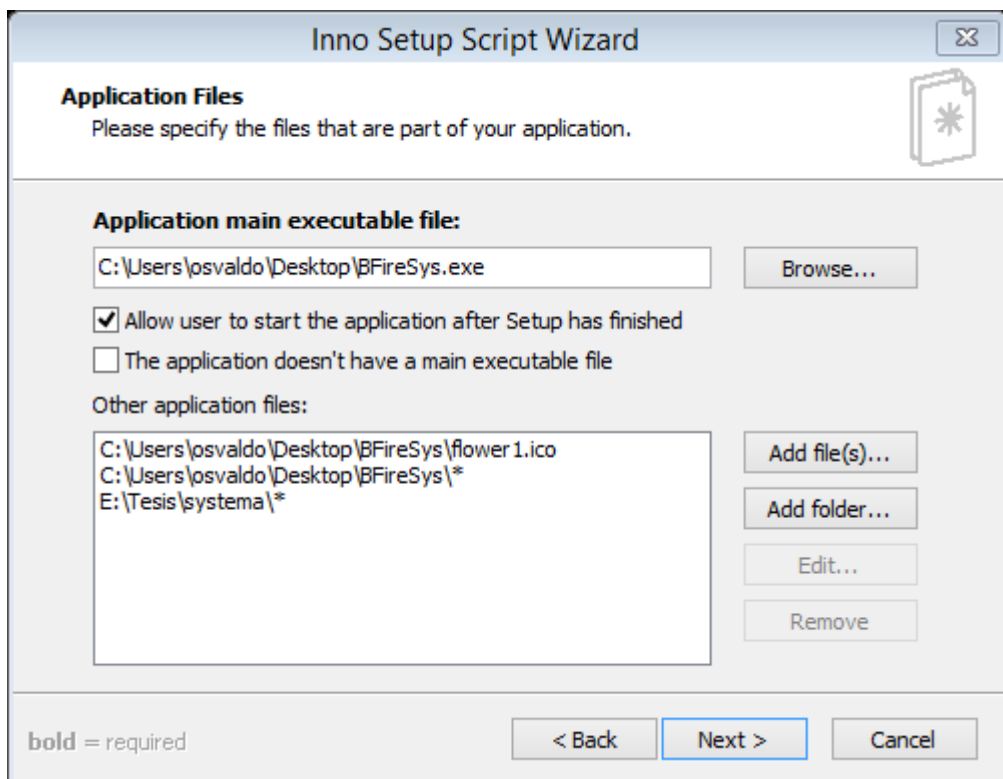


Diagrama 68: Selección del ejecutable de BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

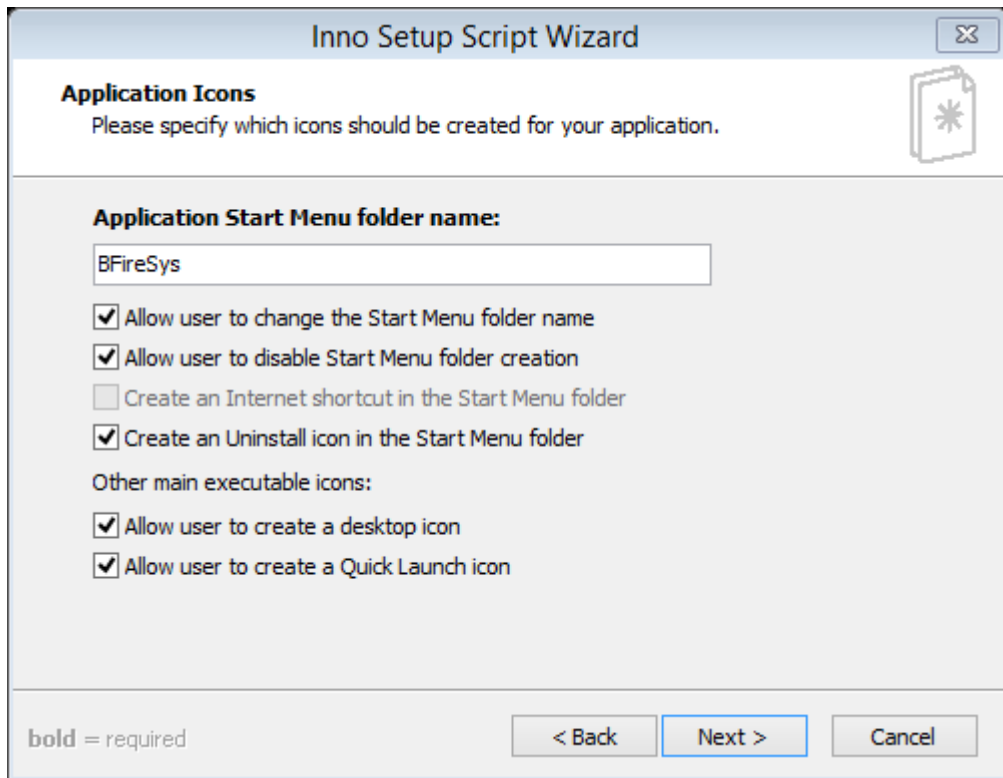


Diagrama 69: Selección de los iconos de BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

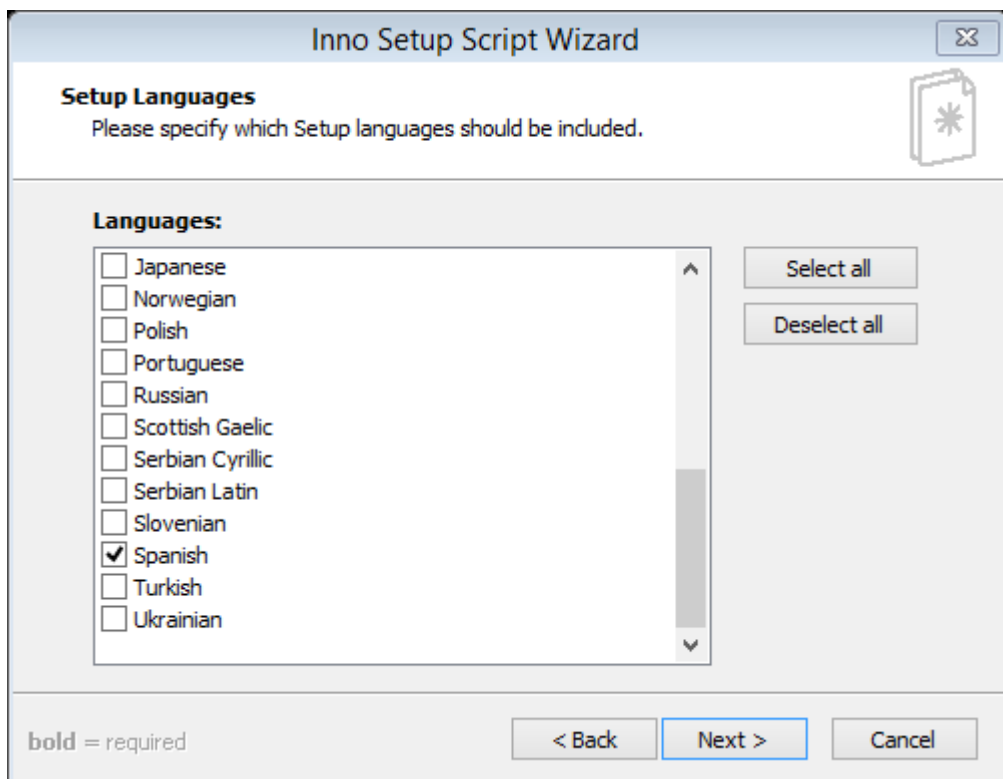


Diagrama 70: Selección de los idiomas para el instalador de BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

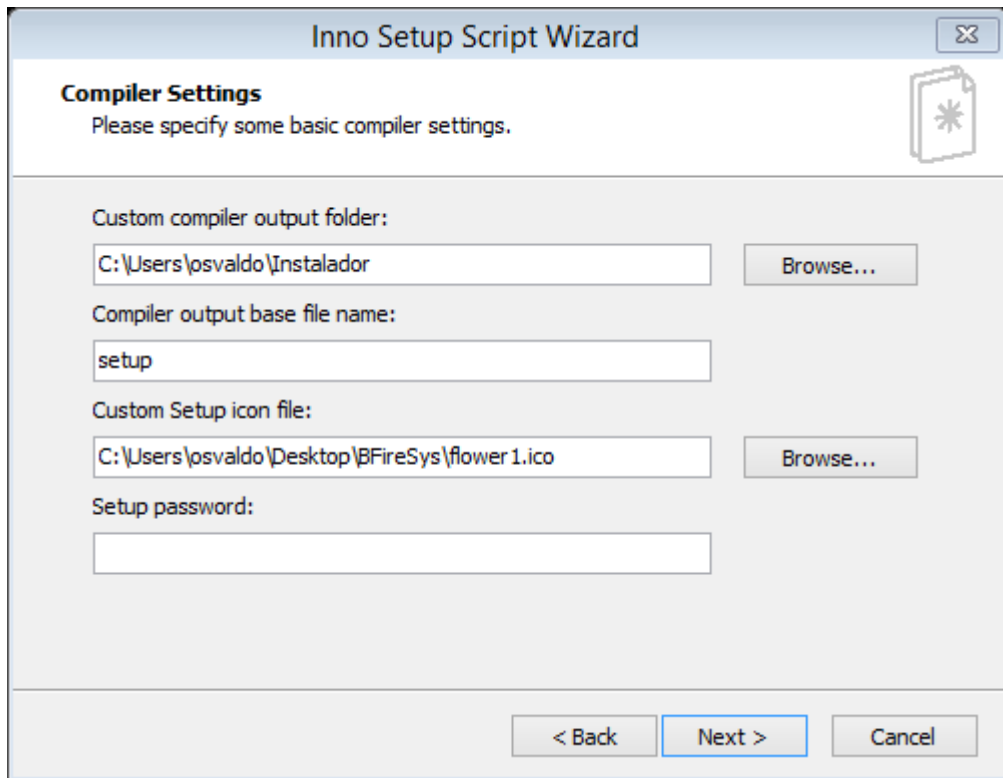


Diagrama 71: Selección de la carpeta para crear el instalador de BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

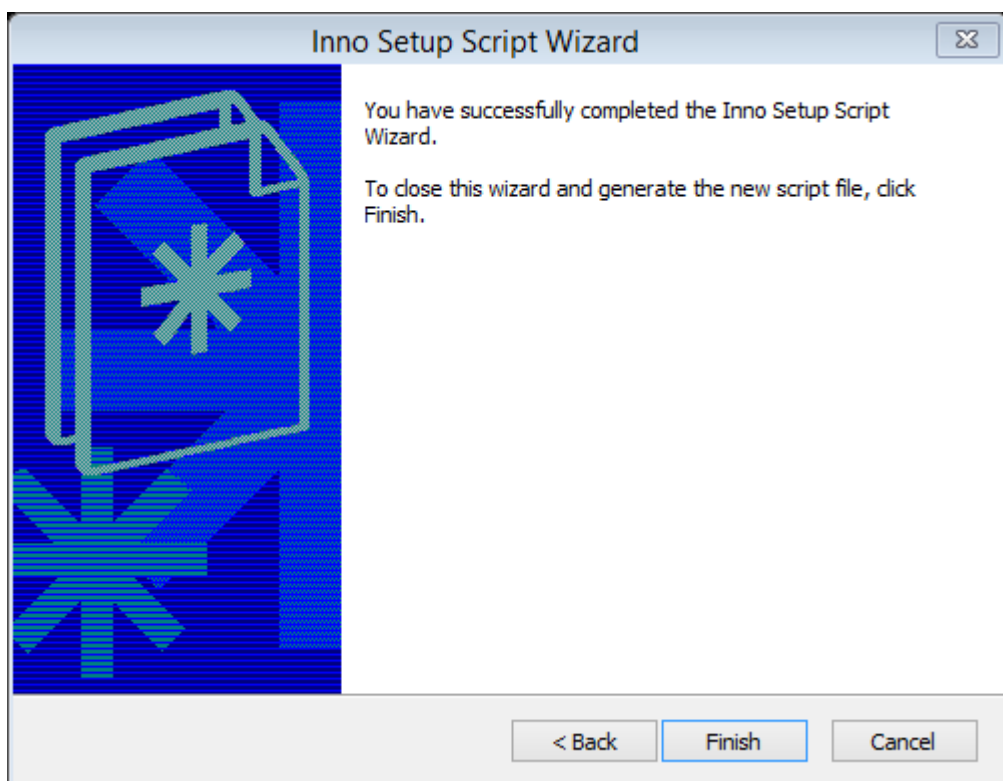
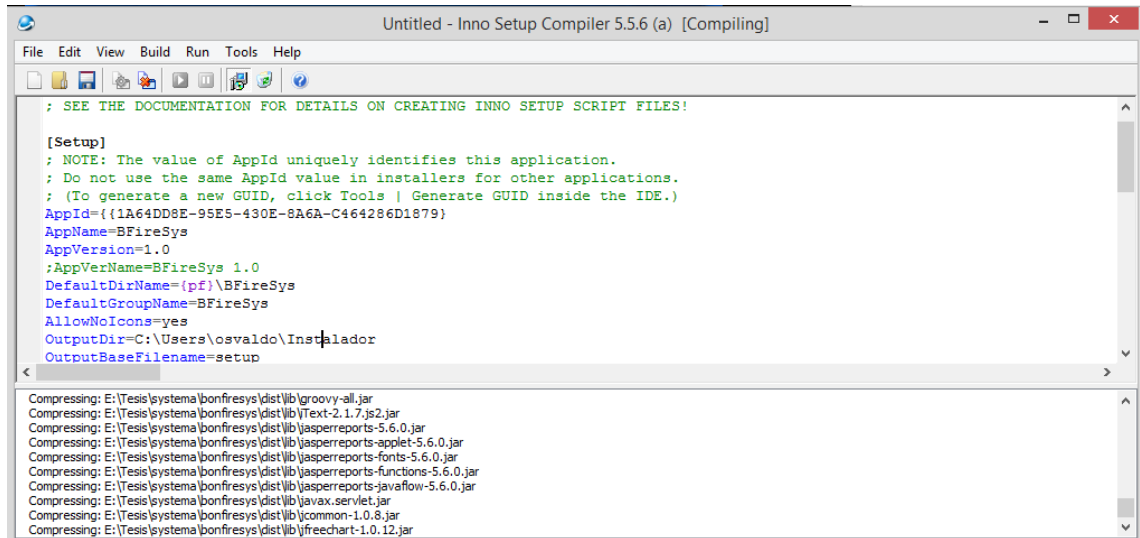


Diagrama 72: finalización de las opciones para la creación del instalador de BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

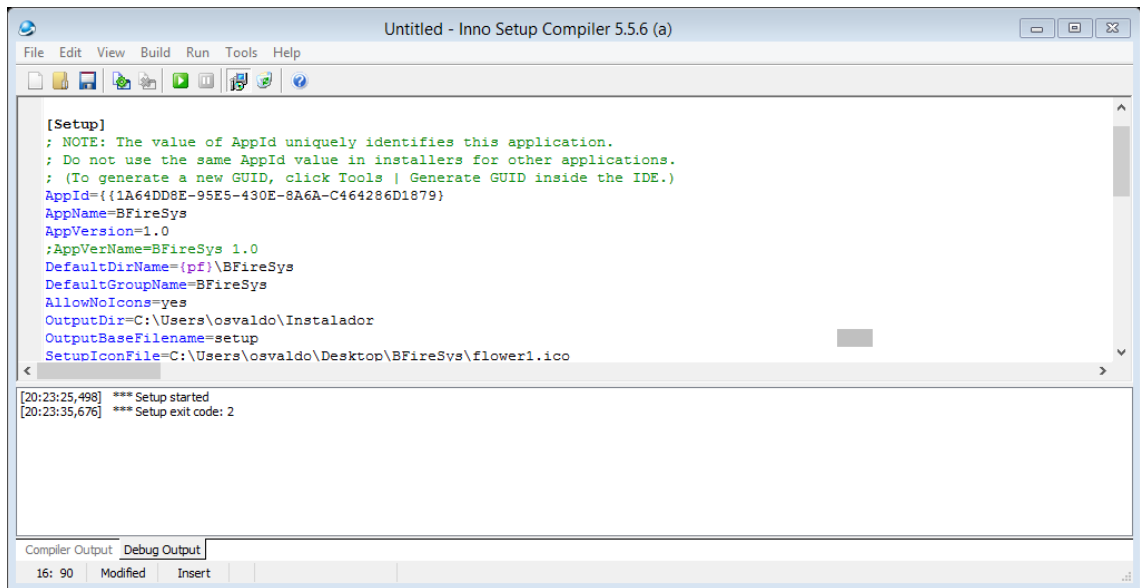


```
Untitled - Inno Setup Compiler 5.5.6 (a) [Compiling]
File Edit View Build Run Tools Help
; SEE THE DOCUMENTATION FOR DETAILS ON CREATING INNO SETUP SCRIPT FILES!
[Setup]
; NOTE: The value of AppId uniquely identifies this application.
; Do not use the same AppId value in installers for other applications.
; (To generate a new GUID, click Tools | Generate GUID inside the IDE.)
AppId={{1A64DD8E-95E5-430E-8A6A-C464286D1879}}
AppName=BFireSys
AppVersion=1.0
;AppVerName=BFireSys 1.0
DefaultDirName={pf}\BFireSys
DefaultGroupName=BFireSys
AllowNoIcons=yes
OutputDir=C:\Users\osvaldo\Instalador
OutputBaseFilename=setup
Compressing: E:\Tesis\systema\bonfiresys\dist\lib\groovy-all.jar
Compressing: E:\Tesis\systema\bonfiresys\dist\lib\Text-2.1.7.js2.jar
Compressing: E:\Tesis\systema\bonfiresys\dist\lib\jasperreports-5.6.0.jar
Compressing: E:\Tesis\systema\bonfiresys\dist\lib\jasperreports-applet-5.6.0.jar
Compressing: E:\Tesis\systema\bonfiresys\dist\lib\jasperreports-fonts-5.6.0.jar
Compressing: E:\Tesis\systema\bonfiresys\dist\lib\jasperreports-functions-5.6.0.jar
Compressing: E:\Tesis\systema\bonfiresys\dist\lib\jasperreports-javaflow-5.6.0.jar
Compressing: E:\Tesis\systema\bonfiresys\dist\lib\javax.servlet.jar
Compressing: E:\Tesis\systema\bonfiresys\dist\lib\common-1.0.8.jar
Compressing: E:\Tesis\systema\bonfiresys\dist\lib\frechart-1.0.12.jar
```

Diagrama 73: Proceso de creación del instalador BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015



```
Untitled - Inno Setup Compiler 5.5.6 (a)
File Edit View Build Run Tools Help
[Setup]
; NOTE: The value of AppId uniquely identifies this application.
; Do not use the same AppId value in installers for other applications.
; (To generate a new GUID, click Tools | Generate GUID inside the IDE.)
AppId={{1A64DD8E-95E5-430E-8A6A-C464286D1879}}
AppName=BFireSys
AppVersion=1.0
;AppVerName=BFireSys 1.0
DefaultDirName={pf}\BFireSys
DefaultGroupName=BFireSys
AllowNoIcons=yes
OutputDir=C:\Users\osvaldo\Instalador
OutputBaseFilename=setup
SetupIconFile=C:\Users\osvaldo\Desktop\BFireSys\flower1.ico
[20:23:25,498] *** Setup started
[20:23:35,676] *** Setup exit code: 2
Compiler Output Debug Output
16: 90 Modified Insert
```

Diagrama 74: Creación del instalador de BFireSys.exe

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

5.1.2 Instalación de Software.-

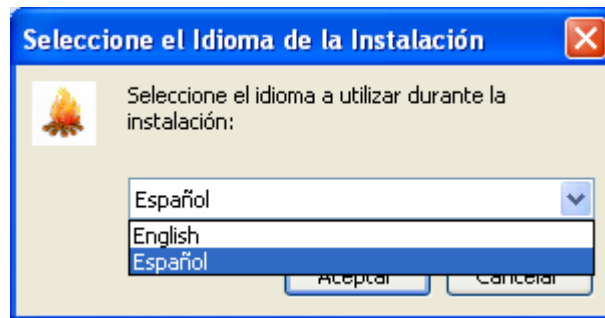


Diagrama 75: Selección del idioma de BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

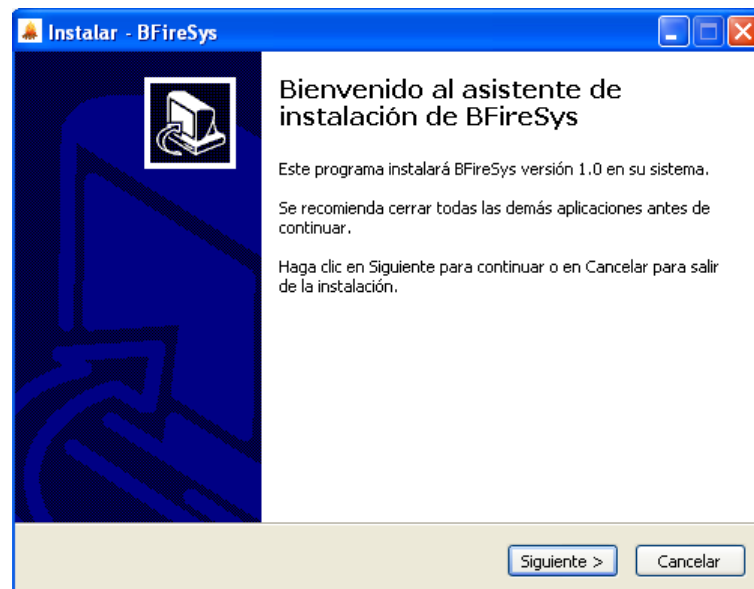


Diagrama 76: Bienvenida del instalador de BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

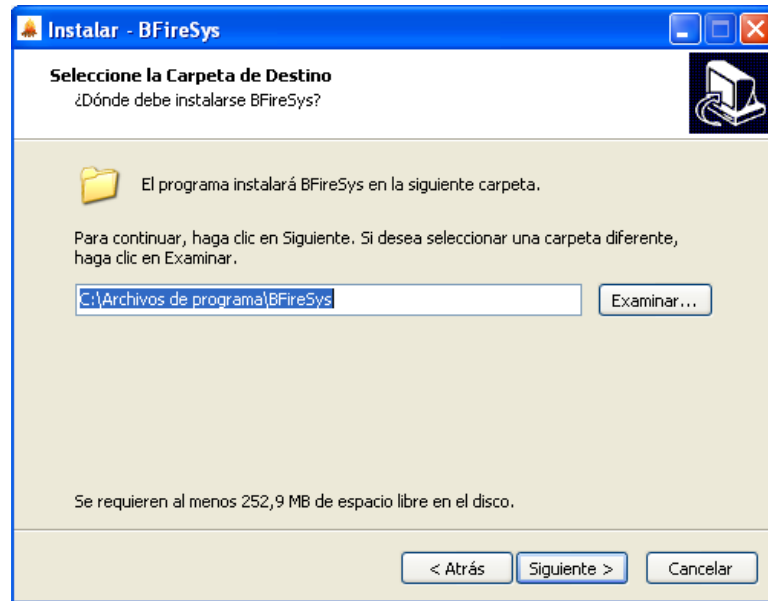


Diagrama 77: Selección de la carpeta para instalar BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

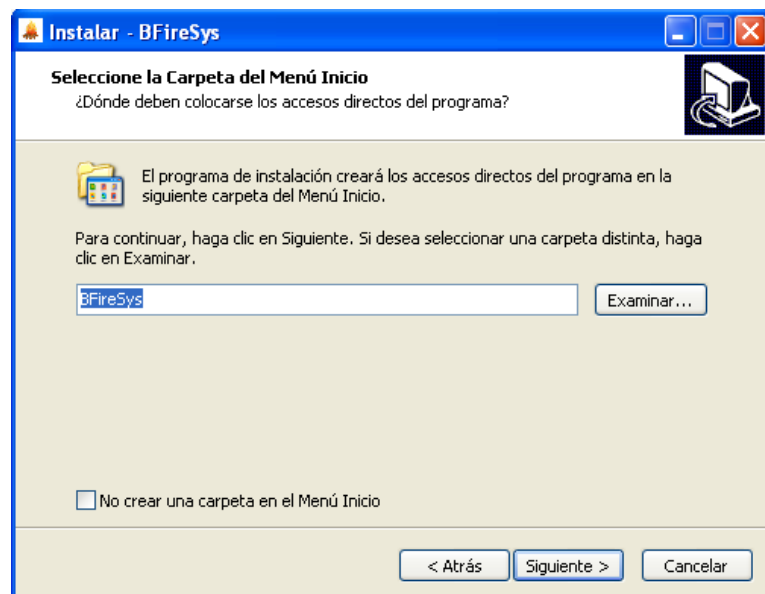


Diagrama 78: Nombre del menú donde se crea el acceso directo de BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

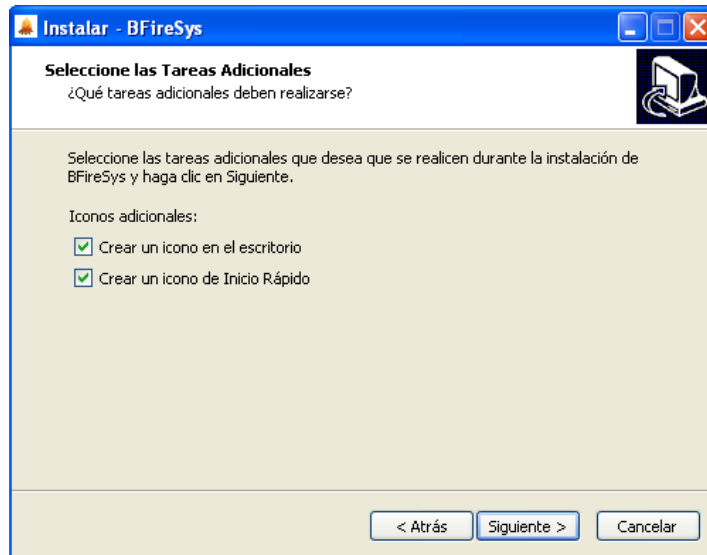


Diagrama 79: Selección de las opciones finales para instalar BFireSys

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

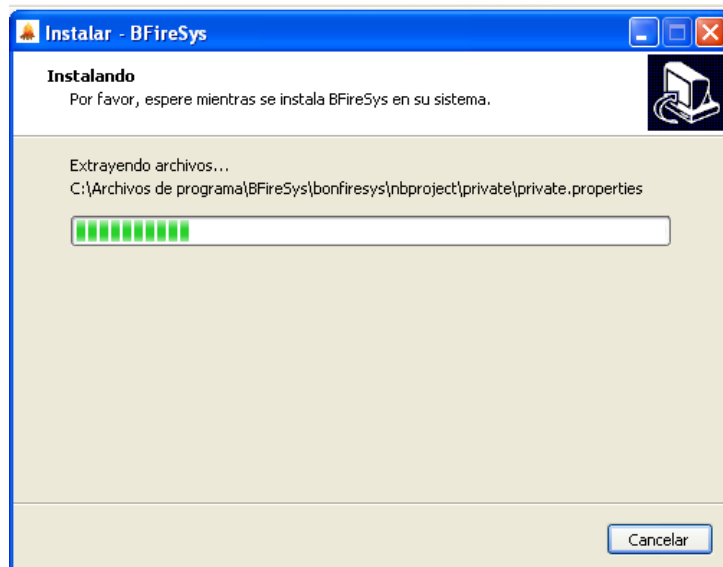


Diagrama 80: Instalación del sistema BFireSys.exe

Elaborado por: Osvaldo Tapia

Fecha: 01/11/2015

CAPITULO 6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones.-

- ✓ Se logró la creación del sistema para la administración de la facturación del restaurante “La Fogata”, el mismo que permite extraer y almacenar registros de: clientes, proveedores, compras y facturación discriminando los usuarios del programa, el cual provee un conjunto eficiente de reportes que reflejan la información requerida por el usuario y dueños del negocio.
- ✓ Se optimizó el conocimiento adquirido en cuanto a la creación del diseño del modelo de la base de datos, logrando con ello la obtención de un esquema amigable y a la vez seguro y eficiente de todos y cada uno de los procesos requeridos en la implementación del sistema de facturación del restaurante “La Fogata”.
- ✓ En cuanto a la facturación electrónica fue necesaria la absorción de nuevo conocimiento, pues en la actualidad el gobierno intenta imponer en los negocios la implementación de la facturación electrónica, sin embargo en el caso particular del lugar de funcionamiento del negocio que nos compete, será factible en un tiempo aproximado de cinco años porque actualmente los clientes están acostumbrados a recibir facturas físicas, el programa de comprobantes electrónicos del SRI no sirve para control de las operaciones de la empresa sino que, está diseñado para que el SRI obtenga información y control para el cobro de impuestos.
- ✓ Durante la elaboración del sistema para la administración de facturación del restaurante “La Fogata” se determinó que es imprescindible la realización de un doble casting en el ingreso de datos, el primero lo realizamos en el momento de la digitación y el segundo a la hora de enviar los registros a guardar en la base de datos, con ello el sistema controla el registro inadecuado de información y por ende proporciona seguridad de que la información sea la idónea y esperada por el usuario.
- ✓ Los conocimientos adquiridos en la Universidad me dieron las bases en cuanto al manejo de las herramientas: Netbeans 8.0 y PostgreSQL que

fueron elegidas para la producción del sistema de facturación del restaurante “La Fogata”. La documentación relacionada con estos programas es extensa lo que facilitó la consecución del objetivo planteado en el presente trabajo de disertación de grado.

- ✓ Se concluyó que es imprescindible la implementación de un sistema de control de compras, ventas, proveedores y clientes en el restaurante “La Fogata” el cual provee de herramientas gráficas que facilitan la toma de decisiones, como es el caso de: ahorro en el desperdicio de insumos, búsqueda de mejores costos, mejoramiento en la elaboración de productos, enfoque en productos que generen mayor rentabilidad y la creación de un producto estrella.
- ✓ El sistema fue diseñado para un restaurante en específico, sin embargo al concluir el trabajo de desarrollo se determinó que puede ser fácilmente utilizado en restaurantes de la misma categoría, por lo tanto se ha creado una herramienta la cual se pondrá a disposición de la comunidad, contribuyendo de esta manera a la sociedad con un producto informático elaborado gracias a los conocimientos adquiridos en la escuela de Ingeniería en Sistemas y Computación de la Pontificia Universidad Católica del Ecuador.

6.2 Recomendaciones.-

- ✓ Es indispensable tomarse el tiempo suficiente a la hora de realizar el diseño del modelo de la base de datos, solicitar ayuda a colegas en caso de tener alguna duda y luego de una exhaustiva revisión iniciar con el proceso de programación. Con la finalidad de evitar la pérdida de tiempo que ocasionaría la repetición de procesos erróneos.
- ✓ Es imprescindible que las herramientas de elaboración de sistemas sean probadas previamente por el programador, y con ello determinar si las mismas pueden interactuar libremente y sin errores de compatibilidad.
- ✓ La implementación de la facturación electrónica al no ser obligatoria para todas personas o sociedades, se recomienda un análisis concienzudo y particularizado de cada uno de las empresa y personas naturales

obligadas a llevar contabilidad que deseen incluir la facturación electrónica.

- ✓ Es necesario que el sistema sea aprobado por los usuarios finales antes de la implementación ya que ellos saben más del problema que nosotros como programadores, ello nos ayudará a identificar las posibles debilidades con mayor agilidad.
- ✓ La inclusión de herramientas informáticas para el manejo administrativo de empresas actualmente es de obligatoriedad, en el sentido de que permite el ahorro de tiempo, mejora la eficiencia en la obtención de resultados de las transacciones realizadas, y a su vez incrementa la agilidad en la toma de decisiones gerenciales.
- ✓ El sistema debe ser diseñado de tal manera que permita corregir los errores en los que podría incurrir el usuario en el momento de ingresar los datos, los cuales contiene caracteres que por su naturaleza son susceptibles de ser digitados de forma inexacta.
- ✓ Se aconseja analizar profundamente la lista de requerimientos en conjunto con la persona al mando de la empresa y firmar un contrato, inclusive si el mismo no va ser cancelado en dinero en efectivo, con ello se evitará cambios pequeños o grandes a la hora de entregar el sistema.

TABLA DE DIAGRAMAS, TABLAS Y GRAFICAS

Diagramas

Diagrama 1: Ciclo de vida en espiral.....	3
Diagrama 2: Fases del Proceso Racional Unificado (RUP).....	3
Diagrama 3: Diagrama General	22
Diagrama 4: Ingreso al Sistema.....	23
Diagrama 5: Administrar Usuarios	24
Diagrama 6: Ingresar Usuarios	25
Diagrama 7: Modificar Usuarios.....	26
Diagrama 8: Eliminar Usuarios	27
Diagrama 9: Consultar Usuarios	28
Diagrama 10: Administrar Clientes.....	29
Diagrama 11: Ingresar Clientes	29
Diagrama 12: Modificar Clientes	30
Diagrama 13: Eliminar Clientes.....	31
Diagrama 14: Consultar Clientes	32
Diagrama 15: Administrar Proveedores	33
Diagrama 16: Ingresar Proveedores	34
Diagrama 17: Modificar Proveedores.....	35
Diagrama 18: Eliminar Proveedores	36
Diagrama 19: Consultar Proveedores.....	37
Diagrama 20: Administrar Productos	38
Diagrama 21: Ingresar Productos	38
Diagrama 22: Modificar Productos.....	39
Diagrama 23: Eliminar Productos	40
Diagrama 24: Consultar Productos	41
Diagrama 25: Administrar Menús.....	42
Diagrama 26: Ingresar Menús.....	42
Diagrama 27: Modificar Menús	43
Diagrama 28: Eliminar Menús.....	44
Diagrama 29: Consultar Menús	45
Diagrama 30: Administrar Pagos	46
Diagrama 31: Ingresar Pagos	47
Diagrama 32: Modificar Pagos.....	48
Diagrama 33: Eliminar Pagos	49
Diagrama 34: Consultar Pagos	50
Diagrama 35: Administrar Facturas.....	51
Diagrama 36: Ingresar Facturas	51
Diagrama 37: Consultar Facturas	52
Diagrama 38: Administrar Compras	53
Diagrama 39: Ingresar Compras.....	54
Diagrama 40: Consultar Compras.....	55
Diagrama 41: Administrar General.....	56
Diagrama 42: Modificar General	56
Diagrama 43: Sector privado, obligados a emitir comprobantes electrónicos.....	58
Diagrama 44: Sector público, obligados a emitir comprobantes electrónicos.	59
Diagrama 45: registro oficial de obligados a emitir comprobantes electrónicos.....	60
Diagrama 46: Estructura del producto.....	61

Diagrama 47: Prototipo de Interfaz	62
Diagrama 48: Diagrama de Clases de Interfaz.....	63
Diagrama 49: Diagrama de Clases de Dominio del Problema.....	66
Diagrama 50: Diagrama de Clases de Manejo de Datos.....	67
Diagrama 51: Diagrama de Secuencias de Ingreso al Sistema.....	68
Diagrama 52: Diagrama de Secuencias de Administración de Usuarios	69
Diagrama 53: Diagrama de Secuencias de Administración de Clientes	70
Diagrama 54: Diagrama de Secuencias de Administración de Proveedores.....	71
Diagrama 55: Diagrama de Secuencias de Administración de Productos	72
Diagrama 56: Diagrama de Secuencias de Administración de Menús	73
Diagrama 57: Diagrama de Secuencias de Administración de Pagos.....	74
Diagrama 58: Diagrama de Secuencias de Administración de Facturación	75
Diagrama 59: Diagrama de Secuencias de Administración de Compras.....	76
Diagrama 60: Diagrama de Secuencias de Administración General	77
Diagrama 61: Diagrama de modelo conceptual	78
Diagrama 62: Diagrama de modelo físico	79
Diagrama 63: Diagrama de lista de dominios.....	80
Diagrama 64: Cuadro de paquetes de bonfiresys	87
Diagrama 65: Creación del .jar	96
Diagrama 66: Creación de BFireSys.exe	97
Diagrama 67: Información básica de BFireSys	98
Diagrama 68: Selección del ejecutable de BFireSys	98
Diagrama 69: Selección de los iconos de BFireSys	99
Diagrama 70: Selección de los idiomas para el instalador de BFireSys	99
Diagrama 71: Selección de la carpeta para crear el instalador de BFireSys	100
Diagrama 72: finalización de las opciones para la creación del instalador de BFireSys	100
Diagrama 73: Proceso de creación del instalador BFireSys.....	101
Diagrama 74: Creación del instalador de BFireSys.exe	101
Diagrama 75: Selección del idioma de BFireSys.....	102
Diagrama 76: Bienvenida del instalador de BFireSys.....	102
Diagrama 77: Selección de la carpeta para instalar BFireSys.....	103
Diagrama 78: Nombre del menú donde se creara el acceso directo de BFireSys	103
Diagrama 79: Selección de las opciones finales para instalar BFireSys.....	104
Diagrama 80: Instalación del sistema BFireSys.exe	104

Tablas

Tabla 1: Tabla comparativa C, C++ y Java	9
Tabla 2: Flujo de Ingreso al Sistema.....	23
Tabla 3: Excepciones de Ingreso al Sistema	24
Tabla 4: Flujo de Ingreso de Usuarios	25
Tabla 5: Excepciones de Ingreso de Usuarios	25
Tabla 6: Flujo de Modificación de Usuarios.....	26
Tabla 7: Excepciones de Modificación de Usuarios	26
Tabla 8: Flujo de Eliminación de Usuarios	27
Tabla 9: Excepciones de Eliminación de Usuarios.....	27
Tabla 10: Flujo de Consulta de Usuarios	28
Tabla 11: Excepciones de Consulta de Usuarios.....	28
Tabla 12: Flujo de Ingreso de Clientes.....	30
Tabla 13: Excepciones de Ingreso de Clientes	30

Tabla 14: Flujo de Modificación de Clientes.....	31
Tabla 15: Excepciones de Modificación de Clientes	31
Tabla 16: Flujo de Eliminación de Clientes	32
Tabla 17: Excepciones de Eliminación de Clientes	32
Tabla 18: Flujo de Consulta de Clientes.....	33
Tabla 19: Excepciones de Consulta de Clientes	33
Tabla 20: Flujo de Ingreso de Proveedores	34
Tabla 21: Excepciones de Ingreso de Proveedores	34
Tabla 22: Flujo de Modificación de Proveedores.....	35
Tabla 23: Excepciones de Modificación de Proveedores	35
Tabla 24: Flujo de Eliminación de Proveedores	36
Tabla 25: Excepciones de Eliminación de Proveedores.....	36
Tabla 26: Flujo de Consulta de Proveedores	37
Tabla 27: Excepciones de Consulta de Proveedores	37
Tabla 28: Flujo de Ingreso de Productos.....	39
Tabla 29: Excepciones de Ingreso de Productos	39
Tabla 30: Flujo de Modificación de Productos.....	40
Tabla 31: Excepciones de Modificación de Productos	40
Tabla 32: Flujo de Eliminación de Productos	41
Tabla 33: Excepciones de Eliminación de Productos	41
Tabla 34: Flujo de Consulta de Productos	41
Tabla 35: Excepciones de Consulta de Productos	42
Tabla 36: Flujo de Ingreso de Menús.....	43
Tabla 37: Excepciones de Ingreso de Menús	43
Tabla 38: Flujo de Modificación de Menús	44
Tabla 39: Excepciones de Modificación de Menús.....	44
Tabla 40: Flujo de Eliminación de Menús.....	45
Tabla 41: Excepciones de Eliminación de Menús	45
Tabla 42: Flujo de Consulta de Menús.....	46
Tabla 43: Excepciones de Consulta de Menús	46
Tabla 44: Flujo de Ingreso de Pagos	47
Tabla 45: Excepciones de Ingreso de Pagos	47
Tabla 46: Flujo de Modificación de Pagos.....	48
Tabla 47: Excepciones de Modificación de Pagos	48
Tabla 48: Flujo de Eliminación de Pagos	49
Tabla 49: Excepciones de Eliminación de Pagos.....	49
Tabla 50: Flujo de Consulta de Pagos	50
Tabla 51: Excepciones de Consulta de Pagos	50
Tabla 52: Flujo de Ingreso de Facturas.....	52
Tabla 53: Excepciones de Ingreso de Facturas	52
Tabla 54: Flujo de Consulta de Facturas.....	53
Tabla 55: Excepciones de Consulta de Facturas	53
Tabla 56: Flujo de Ingreso de Compras	54
Tabla 57: Excepciones de Ingreso de Compras.....	54
Tabla 58: Flujo de Consulta de Compras	55
Tabla 59: Excepciones de Consulta de Compras.....	55
Tabla 60: Flujo de Modificación General.....	57
Tabla 61: Excepciones de Modificación General.....	57
Tabla 62: Diccionario de datos.....	82
Tabla 63: Pruebas de ingreso al sistema	89
Tabla 64: Pruebas de administración de usuarios.....	90

Tabla 65: Pruebas de administración de clientes	90
Tabla 66: Pruebas de administración de proveedores	91
Tabla 67: Pruebas de administración de productos	92
Tabla 68: Pruebas de administración de menús	93
Tabla 69: Pruebas de administración de pagos	93
Tabla 70: Pruebas de administración de facturación.....	94
Tabla 71: Pruebas de administración de compra	94
Tabla 72: Pruebas de administración general	95

Graficas

Grafica 1: Logo FreeMind	9
Grafica 2: Logo I2P	10
Grafica 3: Logo HeliosPaint	10
Grafica 4: Logo Jajuk	10
Grafica 5: Logo ANTS P2P	11
Grafica 6: Logo BlueJ	11
Grafica 7: Logo Buho21	11
Grafica 8: Logo JDownloader.....	12
Grafica 9: Logo limewire	12
Grafica 10: Logo OpenOffice	12
Grafica 11: Logo Afiliat	17
Grafica 12: Logo BASF	18
Grafica 13: Logo Mohawk	18
Grafica 14: Logo Proximity Group.....	18
Grafica 15: Logo Radio Paradise	19
Grafica 16: Logo DealBase	19
Grafica 17: Logo Valhalla Data Systems.....	19
Grafica 18: Logo Vanten Inc.	19

Extracto de código

Extracto de código 1: Clase cliente DP	87
Extracto de código 2: Conexión a base de dato	88

GLOSARIO DE TÉRMINOS

Algoritmo, 8.- "Un algoritmo consiste en una lista ordenada de operaciones que tienen el propósito de buscar la solución a un problema en matemática, informática y disciplinas afines" (Definición ABC, 2015)

Applets, 10.- "Son pequeños programas escritos en lenguaje Java, diseñados para ser ejecutados desde internet, que podemos colocar en nuestro servidor, junto con el resto de ficheros que componen un sitio Web (documentos HTML, ficheros de imagen, sonido, etc.) para lograr múltiples efectos con texto, imágenes, sonidos, etc." (Vaquero, 2015)

Base de datos, 11.- "Es el conjunto de datos informativos organizados en un mismo contexto para su uso y vinculación" (Definición ABC, 2015)

Bell, 8.- "Bell Telephone Company era una empresa estadounidense de telecomunicaciones creada el 9 de julio de 1877¹ por Alexander Graham Bell y por Gardiner Greene Hubbard" (Wikipedia, 2015)

C++, 9.- "Es un lenguaje imperativo orientado a objetos derivado del C [1]. En realidad un supe conjunto de C, que nació para añadirle cualidades y características de las que carecía. El resultado es que como su ancestro, sigue muy ligado al hardware subyacente, manteniendo una considerable potencia para programación a bajo nivel, pero se la han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción." (Zator Systmes, 2015)

Factura, 23, 84, 98.- "Es la relación de los artículos comprendidos en una operación comercial. Se trata de la cuenta detallada de estas operaciones, incluyendo factores como la cantidad, el peso, la medida y el precio" (Definicion.de, 2015)

FTP, 9.- "File Transfer Protocol, 'Protocolo de Transferencia de Archivos') en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red" (Definición ABC, 2015)

HTTP, 9.- "HTTP es un protocolo de transferencia de hipertexto que se usa en la Web. HTTP es una sigla que significa HyperText Transfer Protocol, o Protocolo de Transferencia de Hipertexto" (Definición ABC, 2015)

IBM, 5, 16, 17.- "Es una empresa multinacional estadounidense de tecnología y consultoría, que fabrica y comercializa hardware y software para computadoras y ofrece servicios de infraestructura, alojamiento de internet y consultoría en una amplia gama de secciones relacionadas con la informática (Wikipedia, 2015)

IVA, 59, 83, 84, 85.- "Hace referencia a un tributo o impuesto que deben pagar los consumidores al Estado por el uso de un determinado servicio o la adquisición de un bien." (Definicion.de, 2015)

Multihilos, 10.-"Se trata en la cual los procesos tienen diversos hilos de ejecución. Si uno de ellos deja de correr no se cae toda la aplicación" (GlosarioIT, 2015)

MySQL, 19.-"El sistema de base de datos operacional MySQL es hoy en día uno de los más importantes en lo que hace al diseño y programación de base de datos de tipo relacional" (Definición ABC, 2015)

Plugins, 11.- "Es una noción que no forma parte del diccionario de la Real Academia Española (RAE). Se trata de un concepto de la lengua inglesa que puede entenderse como “inserción” y que se emplea en el campo de la informática" (Definicion.de, 2015)

Programación orientada a objetos, 8.-"Es una técnica para desarrollar soluciones computacionales utilizando componentes de software (objetos de software)." (Microsoft, 2015)

RAM, 61.-"Una memoria RAM o de acceso aleatorio se utiliza frecuentemente en informática para el almacenamiento de programas y datos informativos" (Definición ABC, 2015)

Sistemas, 4, 7, 16.-"Conjunto de las distintas partes relacionadas entre sí, considerándose por un lado a las partes físicas, definidas bajo el concepto de Hardware, mientras que por otro lado tenemos las otras partes lógicas interactuando con ellas (Software) que a su vez interactúan entre sí en todo momento" (Definicion.de, 2015)

Software, 4, 7, 11, 12, 16, 18, 61.-"Software se refiere al conjunto de instrucciones preparadas para ser interpretadas por una computadora que realiza procesamiento electrónico de datos." (Definición ABC, 2015)

SRI, 60.-" El Servicio de Rentas Internas (SRI) es un organismo autónomo de la República del Ecuador que nació el 2 de diciembre de 1997 basándose en los principios de justicia y equidad, como respuesta a la alta evasión tributaria, alimentada por la ausencia casi total de cultura tributaria" (Wikipedia, 2015)

TCP/IP, 9.-"Es el Protocolo de Control de Transmisión, mientras que el segundo, IP, es el Protocolo de Internet" (Definición ABC, 2015)

Triggers, 18.-" son objetos que se asocian con tablas y se almacenan en la base de datos. Su nombre se deriva por el comportamiento que presentan en su funcionamiento, ya que se ejecutan cuando sucede algún evento sobre las tablas a las que se encuentra asociado" (Definicion.de, 2015)

BIBLIOGRAFÍA

- ✓ 2ndquadrant. (2001). *Second Quadrant*. Obtenido de <http://2ndquadrant.com/>: <http://2ndquadrant.com/es/postgresql/postgresql-la-historia-hasta-ahora/>
- ✓ Afilias. (s.f.). *Afilias página oficial*. Obtenido de Afilias: <http://www.afilias.info/about-us>
- ✓ Almeida, S., & Fernández, E. (2009). *Aplicación de la metodología RUP en el desarrollo del sistema de manejo de inventario y facturación para la empresa SISCONTAV S.A.* Quito, Ecuador.: Pontificia Universidad Católica del Ecuador.
- ✓ Bell, D., & Parr, M. (2011). *Java para estudiantes 6ta Ed.* Naucalpan de Juarez, Mexico: Pearson Educacion.
- ✓ Bishop, J. (1999). *JAVA Fundamentos de programación, 2a Ed.* Madrid: Addison Wesley.
- ✓ Cáliz, D. (2007). *Generador de reportes para postgres.* Quito, Ecuador: Escuela Politécnica nacional.
- ✓ Coronel, E. (2010). *Lenguaje de programación java.* Lima, Perú: Empresa Editora Macro.
- ✓ Definición ABC. (2 de 12 de 2015). *Definicionabc*. Obtenido de <http://www.definicionabc.com/ciencia/algoritmo.php>
- ✓ Definicion.de. (01 de 12 de 2015). *Definicion.de*. Obtenido de Definicion.de: <http://definicion.de/factura/>
- ✓ GlosarioIT. (01 de 12 de 2015). *GlosarioIT*. Obtenido de GlosarioIT: <http://www.glosarioit.com/#!Multihilo>
- ✓ Humberto, E. (2005). *Link Global Services*. Obtenido de Link Global Services: http://www.lgs.com.ve/pres/PresentacionES_PSQL.pdf
- ✓ Kowal, G. (02 de 12 de 2015). *http://launch4j.sourceforge.net/*. Obtenido de <http://launch4j.sourceforge.net/>: <http://launch4j.sourceforge.net/>
- ✓ Martinez, R. (2009). *Sobre postgres*. Obtenido de <http://www.postgresql.org.es>: http://www.postgresql.org.es/sobre_postgresql
- ✓ Microsoft. (01 de 12 de 2015). *Microsoft ayuda*. Obtenido de Microsoft: <https://msdn.microsoft.com/es-es/library/bb972232.aspx>
- ✓ Momjian, B. (2001). *POSTGRESQL, Introduction and Concepts. 1ra Ed.* Minneapolis: Pearson Educación.

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

- ✓ Ruseell, J. (02 de 12 de 2015). *http://www.jrsoftware.org/isinfo.php*. Obtenido de <http://www.jrsoftware.org/isinfo.php>: <http://www.jrsoftware.org/isinfo.php>
- ✓ Valladolid, U. d. (02 de 12 de 2015). *http://www.infor.uva.es*. Obtenido de <http://www.infor.uva.es>: <http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html>
- ✓ Vaquero, M. (01 de 12 de 2015). *www.deciencias.net*. Obtenido de <http://www.deciencias.net/simulaciones/paginas/appletjava.htm>: <http://www.deciencias.net/simulaciones/paginas/appletjava.htm>
- ✓ Vázquez, L. (06 de Enero de 2012). *Blogestudio red de medios*. Obtenido de Empresa y economía: <http://www.empresayeconomia.es/aplicaciones-para-empresas/ventajas-y-desventajas-de-postgresql.html>
- ✓ Vivo Emprendiendo. (30 de 11 de 2015). *http://vivoemprendiendo.com/*. Obtenido de <http://vivoemprendiendo.com/>: <http://vivoemprendiendo.com/>
- ✓ Wikipedia. (02 de 12 de 2015). *Wikipedia*. Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Empaquetado_de_aplicaciones
- ✓ Zator Systemes. (01 de 12 de 2015). *Zator*. Obtenido de [www.Zator.com](http://www.zator.com): http://www.zator.com/Cpp/E1_2.htm

ANEXOS



Manual Técnico

BonFireSys.

Sistema de facturación del restaurante “La Fogata”

Versión 1.0

Oswaldo Tapia

15/12/2015

Índice

1. Introducción	118
2. Objetivo general.....	118
3. Objetivos específicos.....	118
4.1 Modificación de porcentajes del IVA y retenciones..	118
4.2 Cargar la base de datos.....	119
5. Normas, políticas y procedimientos.	119
6. Fundamentación de la tecnología utilizada.....	119
6 Descripción de los actores del sistema	121
7 Especificación de requisitos	121
8 Vista Funcional	121
9 Modelo Lógico de datos.....	131
10 Diccionario de datos	131
11 Codificación	134

1. Introducción

El siguiente sistema fue creado para cubrir las necesidades de automatización de facturación del restaurante “La Fogata” de acuerdo a ello, antes de la utilización de este medio no se contaba con ningún control de los recursos consumidos y/o adquiridos.

Debido a la poca información y/o capacitación, la facturación y controles de ingresos y gastos están totalmente descuidados ya que no se lleva ningún control de estas funciones tan necesarias, debido a que se facturaba mediante el Régimen Impositivo Simplificado Ecuatoriano “RISE” no era necesario almacenar un registro de ventas.

Entre sus principales funciones constan la facturación, registro de compras, además de la creación, modificación y eliminación de usuarios, productos (lo que compra el restaurante), menús (lo que vende el restaurante), proveedores y clientes incluyendo sus respectivos reportes.

Tomando en cuenta lo anteriormente mencionado presento ante ustedes el siguiente sistema informático BonFireSys.

2. Objetivo general

Diseñar e implementar un sistema para la administración de la facturación del restaurante “La fogata”

3. Objetivos específicos

- Enlistar los requerimientos necesarios para el desarrollo del sistema de facturación para el restaurante “La Fogata”.
- Realizar el diseño del modelo de base de datos de los procesos a automatizar.
- Desarrollar la aplicación utilizando las herramientas adecuadas.
- Identificar las debilidades del sistema y corregirlas.

4. Para el administrador del sistema.

Existen algunos cambios que se pueden realizar como super usuario del sistema.

4.1 Modificación de porcentajes del IVA y retenciones

1. Ingresamos al sistema con el usuario y contraseña “super”.
2. Solo tenemos 2 opciones **modificar contraseña** y **modificar impuestos**
3. En la ventana modificar impuestos. Tenemos 3 pestañas: **Modificar IVA**, **modificar Retención IVA**, **Modificar retención RTA** Figura 1.

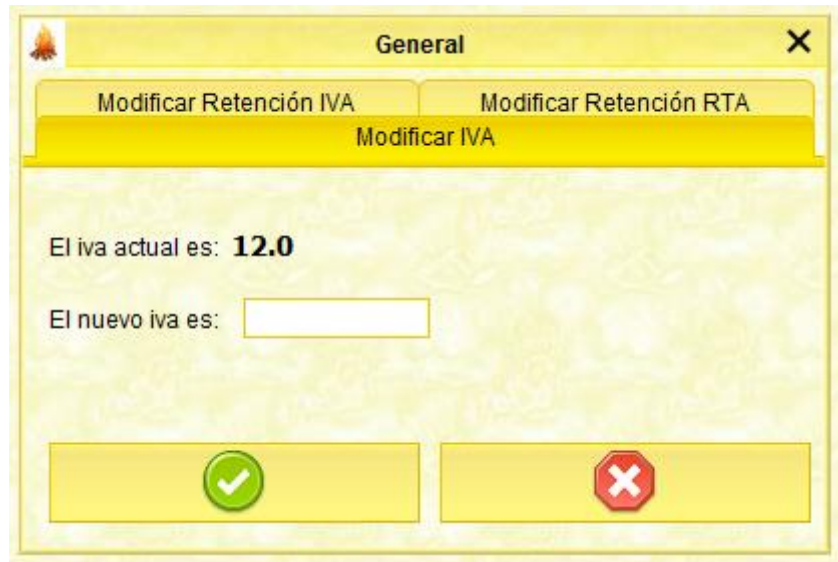


Figura 1: Pantalla General.

4. Cada una de las pestaña tiene su botón para guardar los respectivos cambios. Los cambios realizados a los porcentajes se verán reflejados en los futuros registros, es decir no afectara a las compras o facturas realizadas con anterioridad.

4.2 Cargar la base de datos.

Nuestro respaldo inicial de base de datos se encuentra en la siguiente dirección.

C:\Archivos de programa\BFireSys\bonfiresys\base

Sin embargo si deseamos realizar respaldo o restauraciones de la base de datos necesariamente debe ser realizado con el programa PGadminIII (administrador de base de datos de postgresQL)

5. Normas, políticas y procedimientos.

El restaurante “La Fogata” en el momento de realizar el sistema no constaba con muchas restricciones a la hora de realizar los procedimientos, lo que se tomó en cuenta es los impuestos actuales, y por parte del dueño del negocio se consideró un énfasis en el la siguiente norma:

- Se cobra a la llegada del cliente.
- El valor actual del impuesto al valor agregado es del 12%.
- El valor actual de la retención del impuesto al valor agregado es del 70%.
- El valor actual de la retención en las ventas correspondiente al impuesto a la renta anual, considerando que se vende alimentos procesados y/o revendidos es del 2%

6. Fundamentación de la tecnología utilizada

Para la elección del lenguaje de programación y su respectiva base de datos se tomó en cuenta la gratuidad en el uso y la concatenación con su respectiva

base de datos. Se eligió el lenguaje de programación Java y la base de datos es Postgresql.

➤ **Java.-**

Ventajas.-

8. **Portabilidad.-** no será necesario reescribir el código si lo queremos ocupar en otra máquina.
9. **POO.-** al ser orientado a objetos ofrece todos los beneficios de esta metodología: abstracción, encapsulamiento, polimorfismo, herencia.
10. **Lenguaje de programación.-** posee todas las características de cualquier lenguaje de programación por lo que soporta: graficas, animaciones, sonidos, bases de datos, etc.
11. **Plugins.-** ahorra la instalación de plugins a los usuarios, porque cualquier navegador de internet, soporta java.
12. **JDK.-** el software de desarrollo es una herramienta libre, es decir sin costo.
13. **Base de datos.-** el manejo de base de datos es uniforme.
14. **Conocimiento.-** el conocimiento de programación del lenguaje java es latamente cotizado.

➤ **Postgresql.-**

Ventajas.

1. **Instalación Ilimitada.-** Vale destacar que con PostgreSQL, no se viola ningún acuerdo de licencia, ya que no existe un costo asociado a la licencia de software.
2. **Soporte.-** Aunque la versión de licencia de software es gratuita, existe un sinnúmero de empresas y profesionales dispuestos a colaborar con el soporte de las bases de datos, aunque esto ya no es gratuito.
3. **Ahorros considerables en costos de operación.-** PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.
4. **Estabilidad y confiabilidad.-** es muy común que se no se reporten caídas en varios años de operación de las bases de datos.
5. **Extensible.-** El código fuente está disponible para todos sin costo, si se necesita extender o personalizar PostgreSQL de alguna manera, puede hacerlo con un mínimo esfuerzo, y sin costos adicionales.
6. **Diseñado para ambientes de algo volumen.-** PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes.
7. **Herramientas graficas de diseño y administración de base de datos.-** Existen varias herramientas graficas de alta calidad para administrar las bases de datos como: pgAdmin, pgAccess. Y para diseño como: Tora, Data Architect.

8. **Integridad referencial.-** soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
9. **Bloqueos de tabla y filas.-** ofrece varios modos de bloqueo para controla el acceso concurrente a los datos en tablas, algunos de ellos PostgreSQL los realiza automáticamente antes de la ejecución de una declaración. (Humberto, 2005)

6 Descripción de los actores del sistema

Cajero.- es la persona que con mayor frecuencia usara el sistema, se encargara de la creación, eliminación y modificación de los productos, menús, proveedores, clientes además de la creación de facturas y compras, y podrá modificar su contraseña de usuario.

Gerente.- es la persona que será encargada de la creación de los pagos de las compras y facturas además de su respectiva modificación, adicionalmente podrá realizar las tareas del usuario. Y podrá modificar su contraseña de usuario, crear y eliminar nuevos usuarios.

Súper.- este usuario tiene pocas opciones sin embargo es muy necesario diferenciarlo, dicho usuario será el encargado de la modificación de los porcentajes de IVA y retenciones.

7 Especificación de requisitos

Requisitos funcionales.-

- Usuario
- Cliente
- Proveedor
- Pago
- Menú
- Facturación
- Compras
- Ingreso al sistema

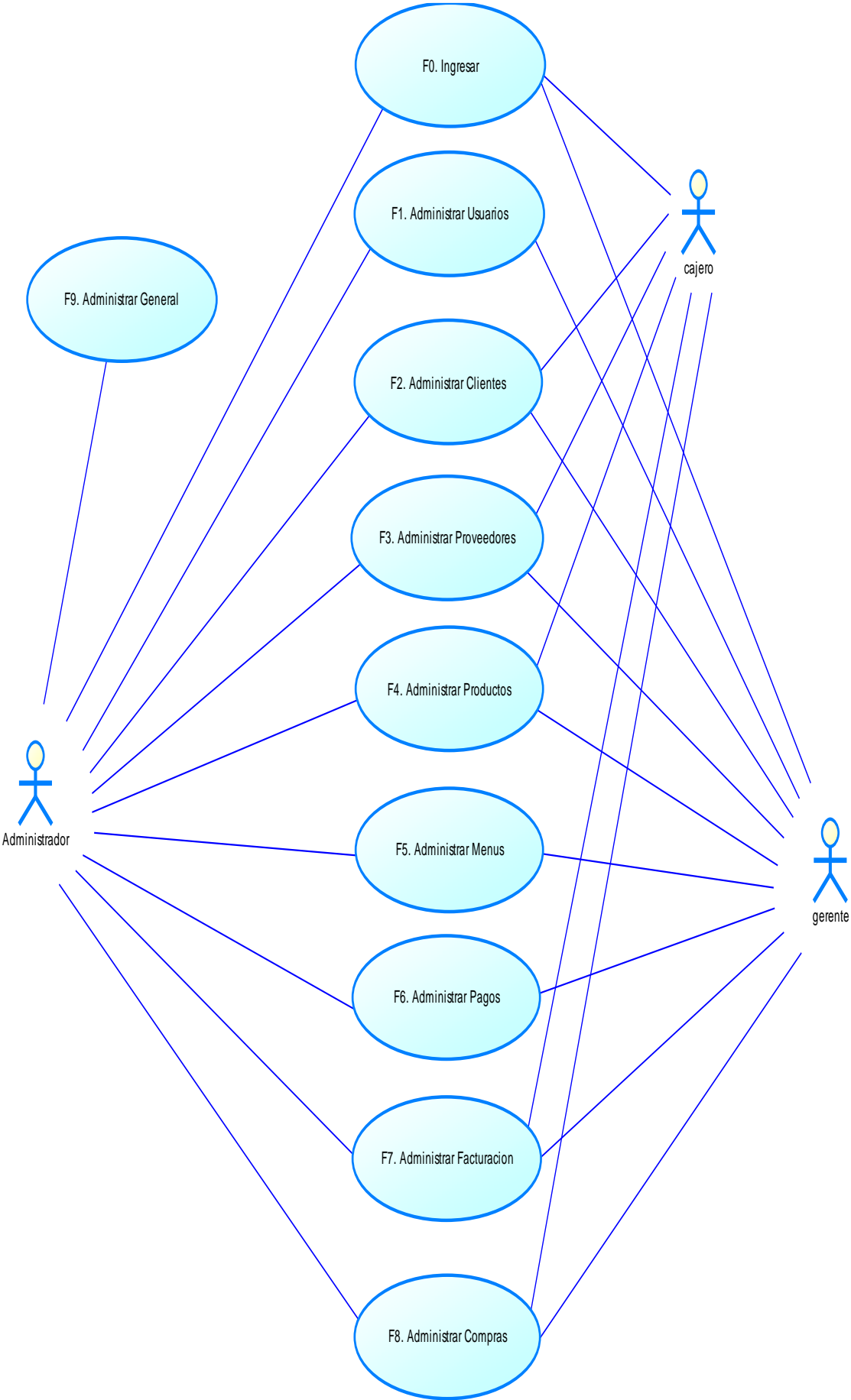
Requisitos no funcionales.-

- No debe depender de conexión a internet.
- Las facturas deben poder imprimirse al instante.
- **Requisitos de hardware**
 - Pentium 4 o equivalente
 - 512 MB en RAM
 - 256 MB de espacio libre en disco duro.
- **Requisitos de software**
 - Windows, desde XP hasta 8
 - Java 7

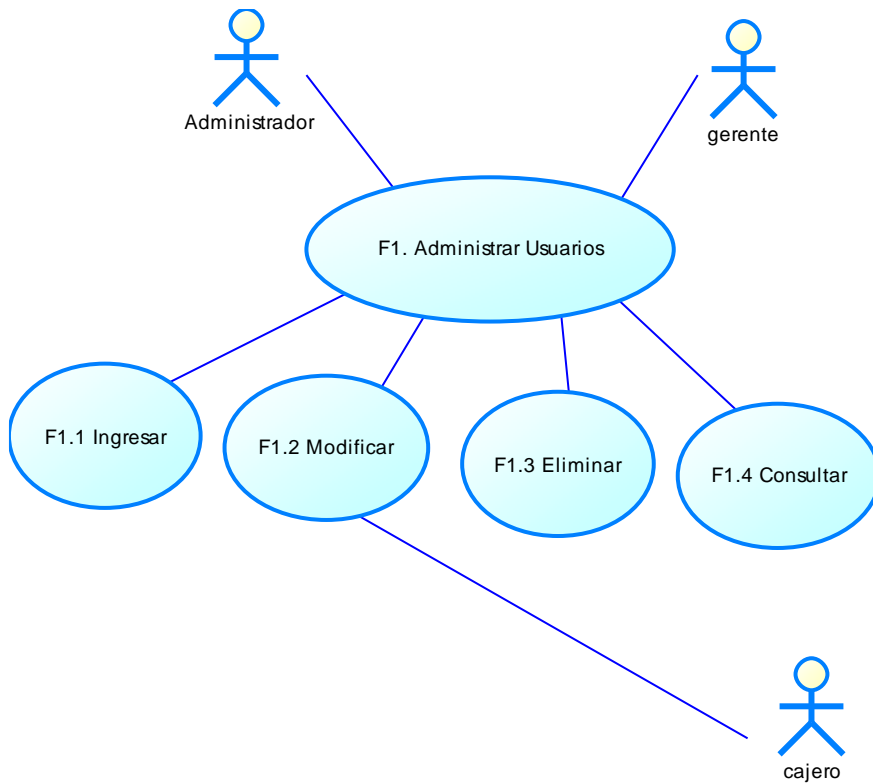
8 Vista Funcional

Diagrama general

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”



Caso de Uso: Administración de Usuarios



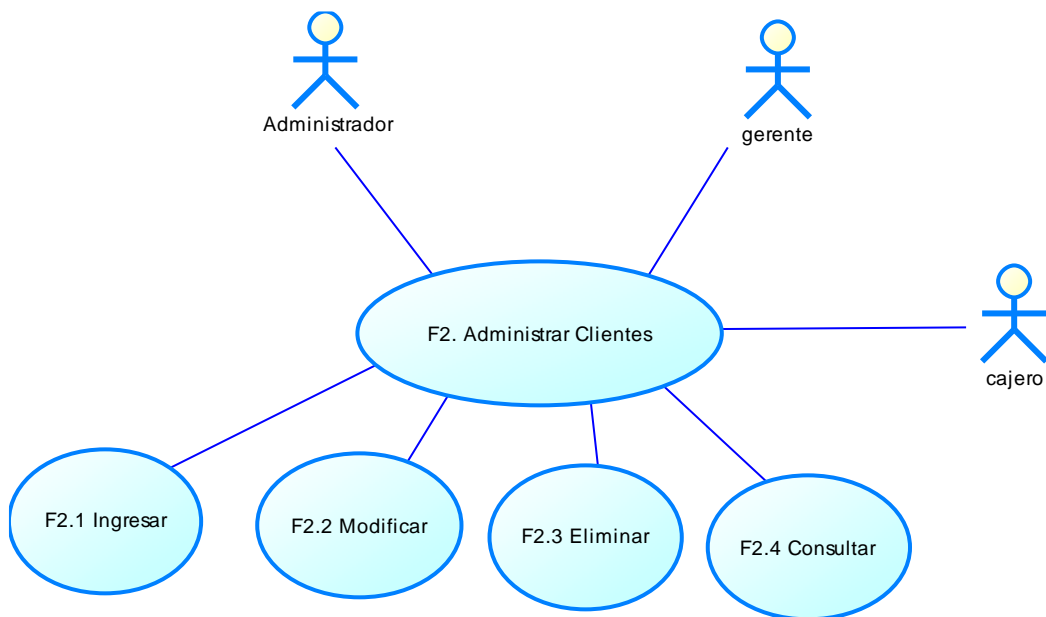
Descripción: Esta administración nos sirve para el manejo de usuarios, pero solo los actores Administrador y Gerente tendrán acceso total de la administración, el cajeros solo podrá modificar la contraseña de su usuario.

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
Ingresar usuario				
1	Genera el evento para ingresar al administrador de usuarios	2	El sistema presenta el administrador de usuarios	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2
Modificar usuario				
1	Genera el evento para ingresar al administrador de usuarios	2	El sistema presenta el administrador de usuarios	
3	Se abre la pestaña de modificación			
4	Se elige al usuario y se modifica el nivel	6	Se verifican los datos	E1, E2
5	Se presiona el botón guardar	7	Se guardan los datos	E2
Eliminar usuario				
1	Genera el evento para ingresar al administrador de usuarios	2	El sistema presenta el administrador de usuarios	

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

3	Se abre la pestaña de Eliminar			
4	Se elige un usuario a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el usuario	E2

Caso de Uso: Administración de Clientes

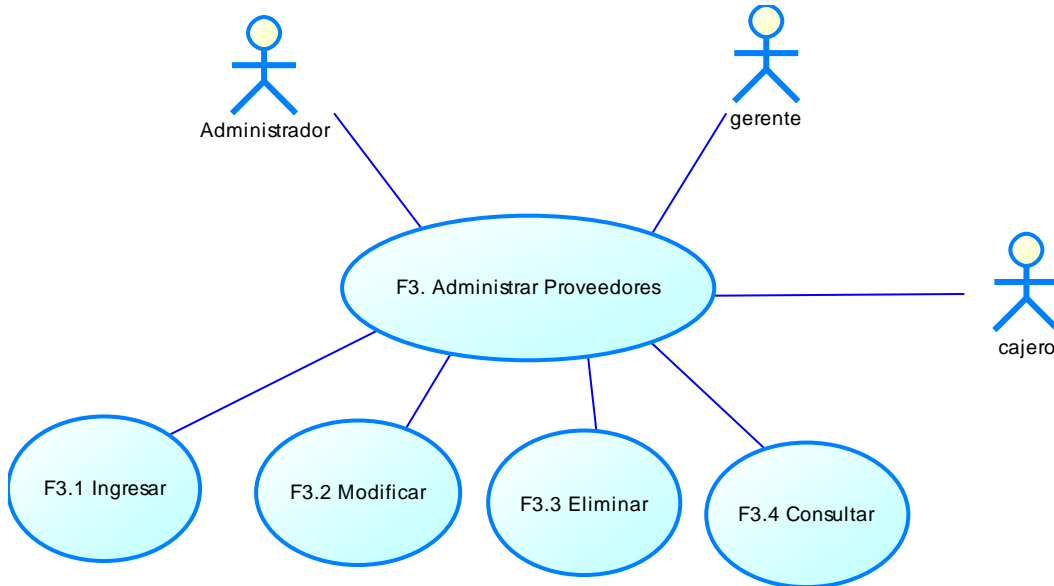


Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
	Ingresar Cliente			
1	Genera el evento para ingresar al administrador de clientes	2	El sistema presenta el administrador de clientes	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2
	Modificar cliente			
1	Genera el evento para ingresar al administrador de clientes	2	El sistema presenta el administrador de clientes	
3	Se abre la pestaña de modificación			
4	Se elige al Cliente y se modifica los datos	6	Se verifican los datos	E1, E2
5	Se presiona el botón guardar	7	Se guardan los datos	E2
	Eliminar Cliente			

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

1	Genera el evento para ingresar al administrador de clientes	2	El sistema presenta el administrador de clientes	
3	Se abre la pestaña de Eliminar			
4	Se elige un Cliente a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el Cliente	E2
Consultar Cliente				
1	Genera el evento para ingresar a consulta de clientes	2	El sistema presenta los datos de clientes	E1, E2

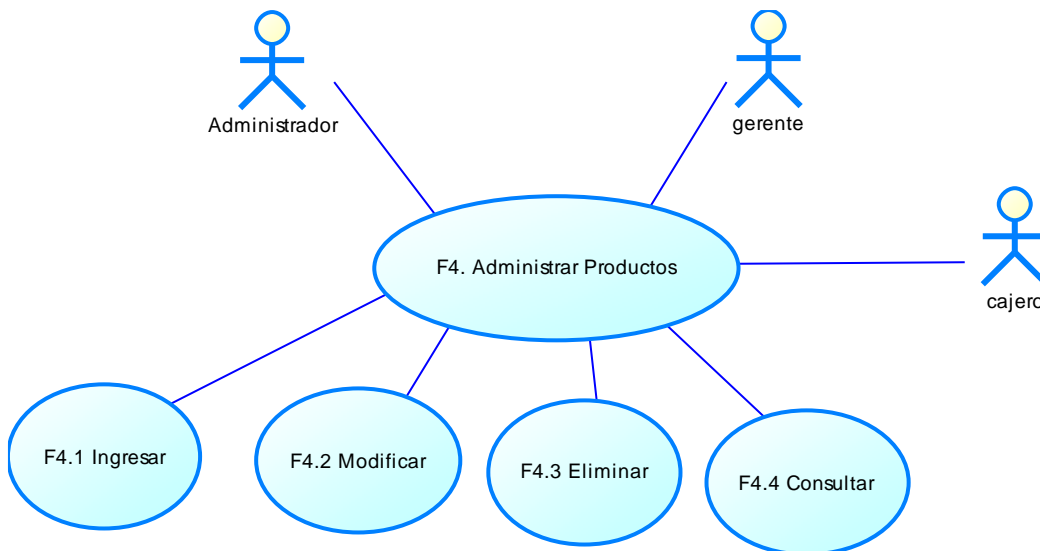
Caso de Uso: Administración de Proveedores



Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
Ingresar Proveedor				
1	Genera el evento para ingresar al administrador de Proveedores	2	El sistema presenta el administrador de Proveedores	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
Modificar Proveedor				
1	Genera el evento para ingresar al administrador de Proveedores	2	El sistema presenta el administrador de Proveedores	
3	Se abre la pestaña de modificación			
4	Se elige al Proveedor y se modifica los datos	6	Se verifican los datos	E1, E2
5	Se presiona el botón guardar	7	Se guardan los datos	E2

Eliminar Proveedor				
1	Genera el evento para ingresar al administrador de Proveedores	2	El sistema presenta el administrador de Proveedores	
3	Se abre la pestaña de Eliminar			
4	Se elige un Proveedor a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el Proveedor	E2
Consultar Proveedor				
1	Genera el evento para ingresar a consulta de Proveedores	2	El sistema presenta los datos de Proveedores	E1, E2

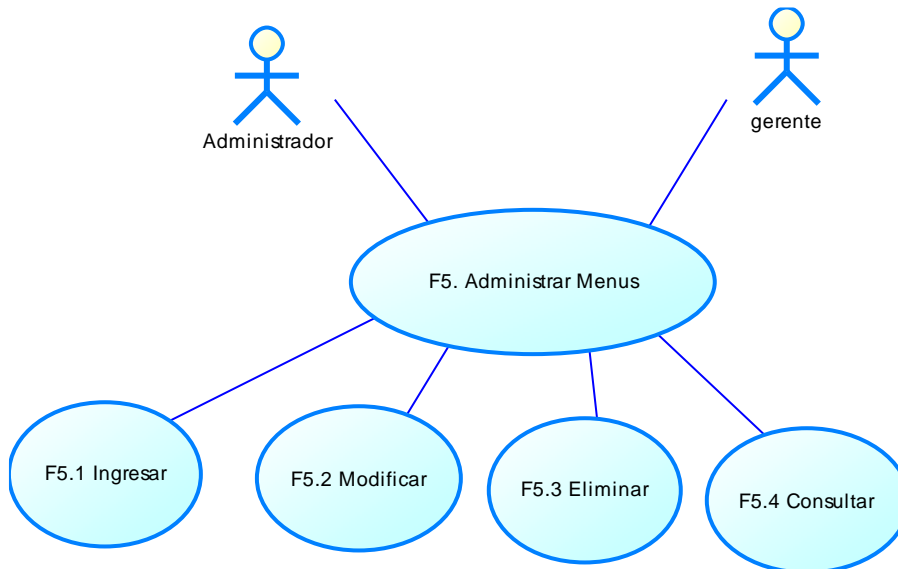
Caso de Uso: Administración de Productos



Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
Ingresar Producto				
1	Genera el evento para ingresar al administrador de Productos	2	El sistema presenta el administrador de Productos	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2
Modificar Producto				
1	Genera el evento para ingresar al administrador de Productos	2	El sistema presenta el administrador de Productos	

3	Se abre la pestaña de modificación			
4	Se elige al Producto y se modifica los datos	6	Se verifican los datos	E1, E2
5	Se presiona el botón guardar	7	Se guardan los datos	E2
Eliminar Producto				
1	Genera el evento para ingresar al administrador de Productos	2	El sistema presenta el administrador de Productos	
3	Se abre la pestaña de Eliminar			
4	Se elige un Producto a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el Producto	E2
Consultar Producto				
1	Genera el evento para ingresar a consulta de Productos	2	El sistema presenta los datos de Productos	E1, E2

Caso de Uso: Administración de Menús

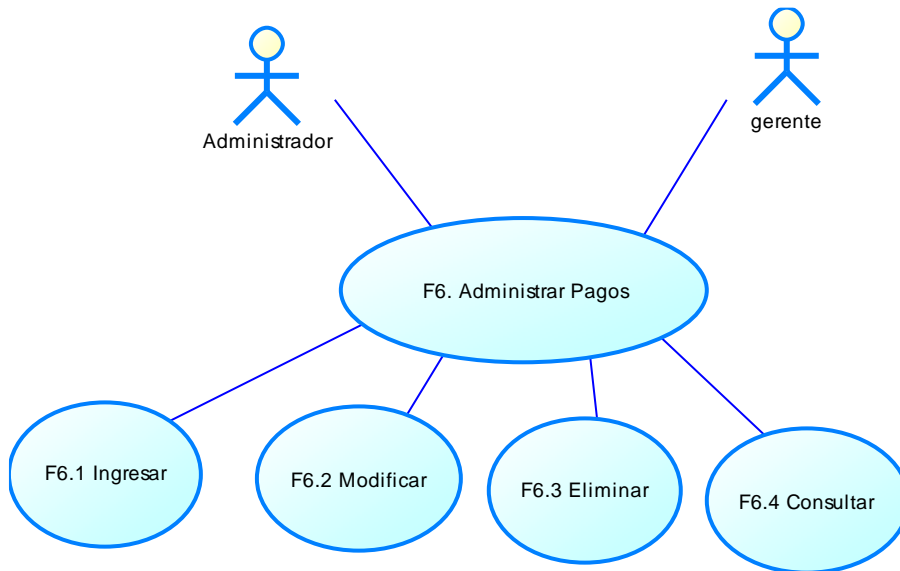


Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
Ingresar Menú				
1	Genera el evento para ingresar al administrador de Menús	2	El sistema presenta el administrador de Menús	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2
Modificar Menú				
1	Genera el evento para ingresar al administrador de Menús	2	El sistema presenta el administrador de Menús	
3	Se abre la pestaña de modificación			

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

4	Se elige al Menú y se modifica los datos	6	Se verifican los datos	E1, E2
5	Se presiona el botón guardar	7	Se guardan los datos	E2
Eliminar Menú				
1	Genera el evento para ingresar al administrador de Menús	2	El sistema presenta el administrador de Menús	
3	Se abre la pestaña de Eliminar			
4	Se elige un Menú a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el Menú	E2
Consultar Menú				
1	Genera el evento para ingresar a consulta de Menús	2	El sistema presenta los datos de Menús	E1, E2

Caso de Uso: Administración de Pagos

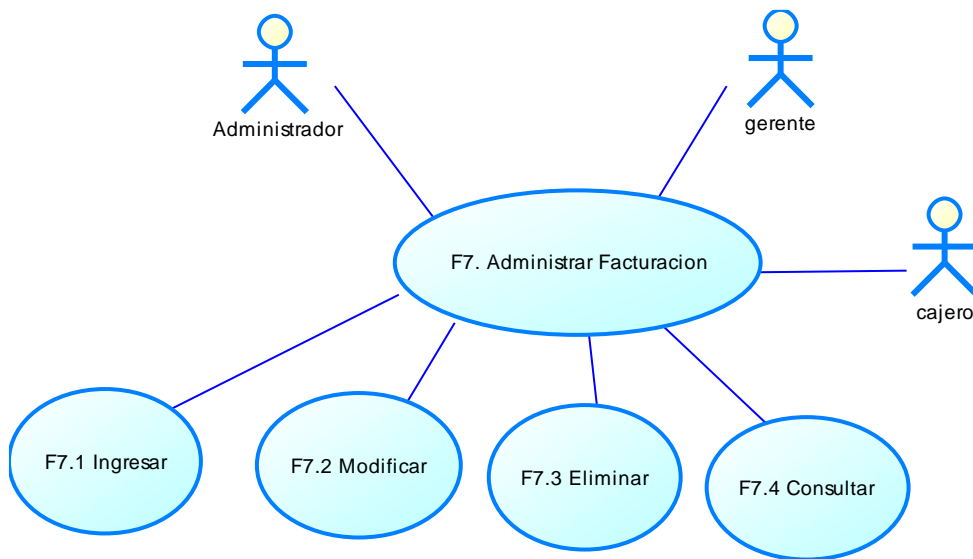


Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
Ingresar Pago				
1	Genera el evento para ingresar al administrador de Pagos	2	El sistema presenta el administrador de Pagos	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2
Modificar Pago				
1	Genera el evento para ingresar al administrador de Pagos	2	El sistema presenta el administrador de Pagos	
3	Se abre la pestaña de modificación			
4	Se elige al Pago y se modifica los datos	6	Se verifican los datos	E1, E2

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

5	Se presiona el botón guardar	7	Se guardan los datos	E2
Eliminar Pago				
1	Genera el evento para ingresar al administrador de Pagos	2	El sistema presenta el administrador de Pagos	
3	Se abre la pestaña de Eliminar			
4	Se elige un Pago a Eliminar	6	Se verifican los datos	E1, E2
5	Se presiona el botón confirmar	7	Se elimina el Pago	E2
Consultar Pago				
1	Genera el evento para ingresar a consulta de Pagos	2	El sistema presenta los datos de Pagos	E1, E2

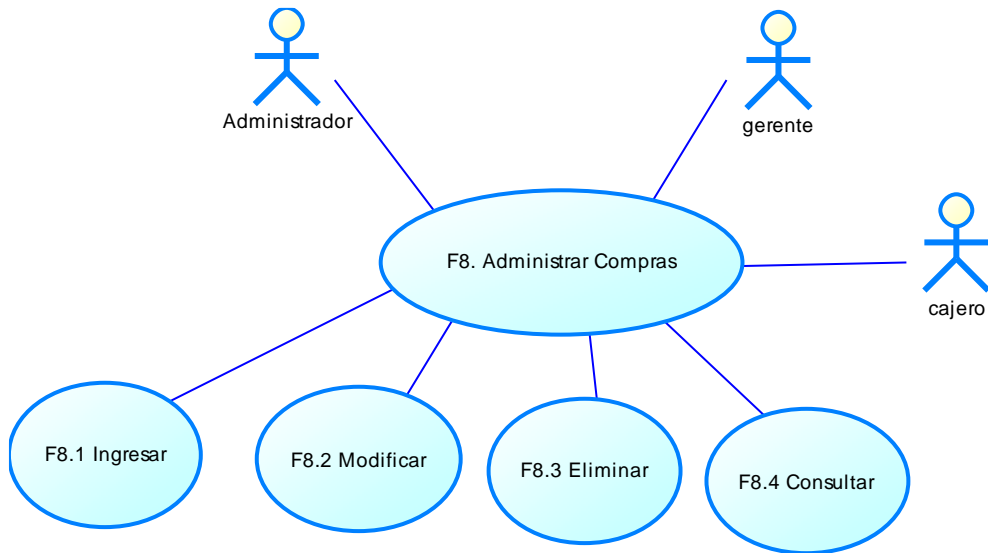
Caso de Uso: Administración de Facturas



Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
Ingresar Factura				
1	Genera el evento para ingresar al administrador de Facturas	2	El sistema presenta el administrador de Facturas	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2
Consultar Factura				
1	Genera el evento para ingresar a consulta de Facturas	2	El sistema presenta los datos de Facturas	E1, E2

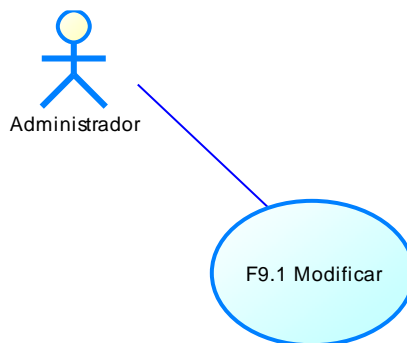
Caso de Uso: Administración de Compras

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”



Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
	Ingresar compra			
1	Genera el evento para ingresar al administrador de Compras	2	El sistema presenta el administrador de Compras	
3	Se ingresan los datos			
4	Se presiona el botón guardar	5	Se verifican los datos	E1, E2
		6	Se guardan los datos	E2
	Consultar compra			
1	Genera el evento para ingresar a consulta de Compras	2	El sistema presenta los datos de Compras	E1, E2

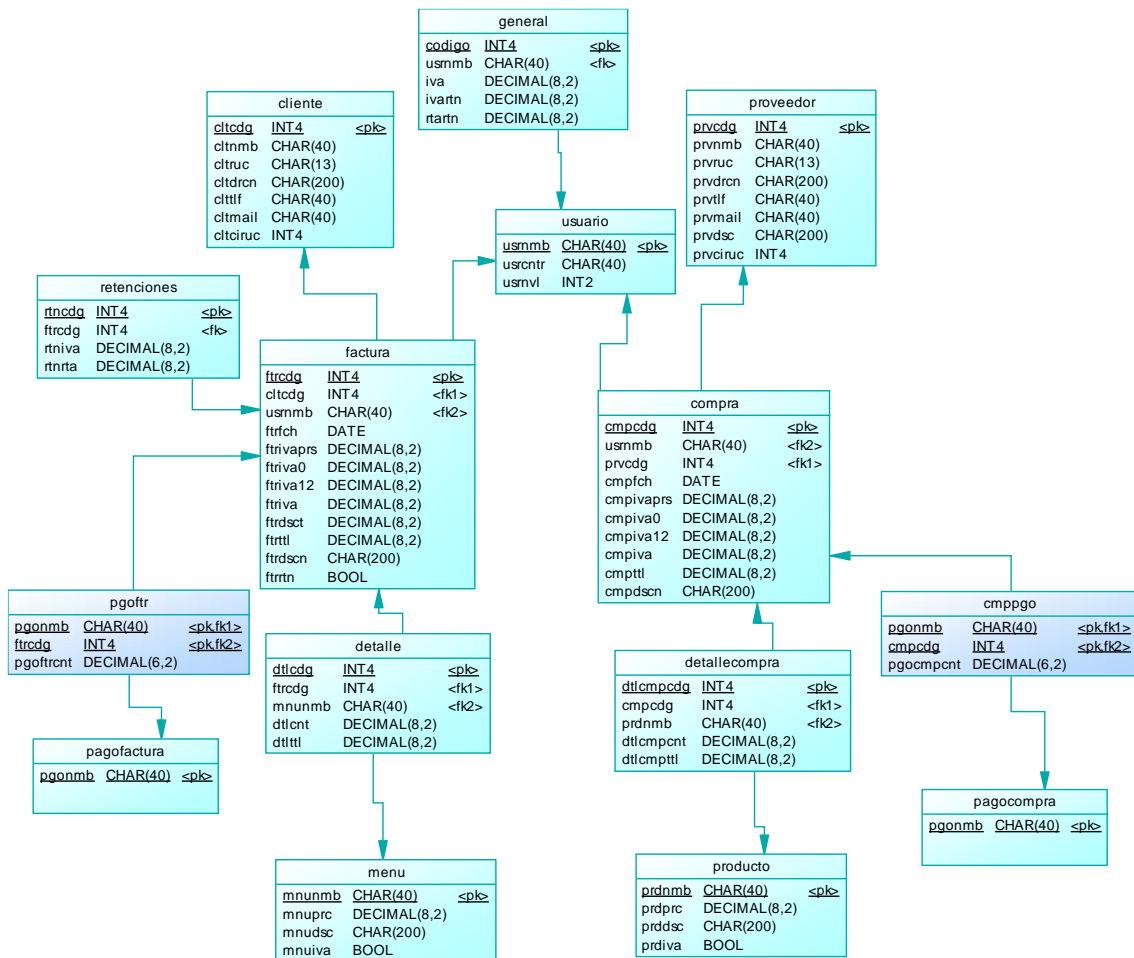
Caso de Uso: Modificar General



Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Genera el evento para ingresar al administrador General	2	El sistema presenta el administrador General	
3	Se abre la pestaña de modificación			

4	Se elige el impuesto a modificar, y se lo modifica	6	Se verifican los datos	E1
5	Se presiona el botón guardar	7	Se guardan los datos	E1

9 Modelo Lógico de datos



10 Diccionario de datos

Dominios

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

	Name	Code	Com	Data Type	Length	Preci	M	Test
→	codigo	CODIGO		INT4	4		<input type="checkbox"/>	<None
2	descripcion	DESCRIPCION		CHAR(200)	200		<input type="checkbox"/>	<None
3	fecha	FECHA		DATE			<input type="checkbox"/>	<None
4	nombre	NOMBRE		CHAR(40)	40		<input type="checkbox"/>	<None
5	precio	PRECIO		DECIMAL(8,2)	8	2	<input type="checkbox"/>	<None

Diccionario de datos

Nombre	Código	Tipo de Dato	Longitud de caracteres	Número de decimales	Descripción
cltcdg	CLTCDG	Integer			Código del cliente
cltdrcn	CLTDRCN	Characters (200)	200		Dirección del cliente
cltmail	CLTMAIL	Characters (40)	40		Correo electrónico del cliente
cltnmb	CLTNMB	Characters (40)	40		Nombre del cliente
cltruc	CLTRUC	Characters (13)	13		Cedula o Ruc del cliente
cltllf	CLTTLF	Characters (40)	40		Teléfono del cliente
cmpcdg	CMPCDG	Integer			Código de compra
cmpdscn	CMPDSCN	Characters (200)	200		Descripción de la compra
cmpfch	CMPFCH	Date			Fecha de compra
cmpiva	CMPIVA	Decimal (8,2)	8	2	IVA en la compra
cmpiva0	CMPIVA0	Decimal (8,2)	8	2	IVA tarifa 0 en la compra
cmpiva12	CMPIVA12	Decimal (8,2)	8	2	IVA gravado en la compra
cmpivaprs	CMPIVAPRS	Decimal (8,2)	8	2	Porcentaje de IVA en la compra
cmpttl	CMPTTL	Decimal (8,2)	8	2	Total de la compra
dtlcdg	DTLCDG	Integer			Código del detalle de la factura
dtlcmpcdg	DTLCMPCDG	Integer			Código del detalle de la compra
dtlcmpcnt	DTLCMPCNT	Decimal (8,2)	8	2	Cantidad del detalle de la compra
dtlcmpttl	DTLCMPTTL	Decimal (8,2)	8	2	Total del detalle de la compra
dtlcnt	DTLCNT	Decimal (8,2)	8	2	Cantidad del detalle de la factura

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

dtlttl	DTLTTL	Decimal (8,2)	8	2	Total del detalle de la factura
ftrcdg	FTRCDG	Integer			Código de la factura
ftrdscn	FTRDSCN	Characters (200)	200		Descripción de la factura
ftrdsct	FTRDSCCT	Decimal (8,2)	8	2	Descuento de la factura
ftrfch	FTRFCH	Date			Fecha de la factura
ftriva	FTRIVA	Decimal (8,2)	8	2	IVA en la factura
ftriva0	FTRIVA0	Decimal (8,2)	8	2	IVA tarifa 0 en la factura
ftriva12	FTRIVA12	Decimal (8,2)	8	2	IVA tarifa 12 en la factura
ftrivaprs	FTRIVAPRS	Decimal (8,2)	8	2	Porcentaje de iva en la factura
ftrrtn	FTRRTN	Boolean			Indicador si lleva o no retención la factura
ftrttl	FTRTTL	Decimal (8,2)	8	2	Total de la factura
mnudsc	MNUDSC	Characters (200)	200		Descripción del menú o plato
mnuiva	MNUIVA	Boolean			Indicador si lleva o no IVA el menú o plato
mnunmb	MNUNMB	Characters (40)	40		Nombre del menú o plato
mnuprc	MNUPRC	Decimal (8,2)	8		Precio del menú
pgonmb	PGONMB	Characters (40)	40		Nombre del pago en las compras
pgonmb	PGONMB2	Characters (40)	40	2	Nombre del pago en las facturas
pgoval	PGOVAL	Decimal (8,2)	8		Precio del pago
prddsc	PRDDSC	Characters (200)	200	2	Descripción del producto
prdiva	PRDIVA	Boolean			Indicador si lleva o no IVA el producto
prdnmb	PRDNMB	Characters (40)	40		Nombre del producto
prdprc	PRDPRC	Decimal (8,2)	8		Precio del producto
prvcdg	PRVCDG	Integer			Código del proveedor
prvdrcn	PRVDRCN	Characters (200)	200		Dirección del proveedor
prvdsc	PRVDSC	Characters (200)	200		Descripción del proveedor
prvmail	PRVMAIL	Characters (40)	40		Correo electrónico del proveedor
prvnmb	PRVNMB	Characters (40)	40		Nombre del proveedor
prvruc	PRVRUC	Characters (13)	13		RUC o Cedula del proveedor
prvtlf	PRVTLF	Characters (40)	40		Teléfono del proveedor
rtncdg	RTNCDG	Integer			Retención en la renta
rtniva	RTNIVA	Decimal (8,2)	8	2	Código de la retención

rtnrta	RTNRTA	Decimal (8,2)	8		Retención del IVA
usrcntr	USRCNTR	Characters (40)	40	2	Contraseña del usuario
usrnmb	USRNMB	Characters (40)	40	2	Nombre del usuario
usrnvl	USRNVL	integer			Nivel del usuario

11 Codificación

Conexión base de datos

```

package utilitarios;
import java.sql.*;
/**
 *
 * @author Osvaldo
 */
public class conexion {
    Connection con;
    Statement stmt;
    private String driver="org.postgresql.Driver";
    private String conex="jdbc:postgresql://localhost:5432/bfirebd";
    //localhost:'pueto de acceso';bfirebd:nombre de la base de datos
    private String user="postgres";
    private String pass="postgres";
    public conexion(){
        con=null;
    }
    public Connection conectar()
    {
        try
        {

            Class.forName(driver);
        }
        catch(ClassNotFoundException e)
    
```

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

```
{
    System.out.println(e.getMessage());
}
try
{
    con=DriverManager.getConnection(conex, user, pass);
    stmt=con.createStatement();
}
catch(SQLException e)
{
}
return con;
}
}
```

Codificación de clases.-

Cliente

```
import md.cltMD;

/**
 *
 * @author osvaldo
 */

public class cltDP { //inicio de la clase clienteDP (DP por dominio del problema)
//Definimos sus atributos en forma privada para que se pueda acceder a ellos a
través de sus funciones.

    private int codigo;
    private String nombre;
    private String ruc;
    private String direccion;
    private String telefono;
    private String mail;
```

cltMD cliente = new cltMD(this); //definimos un atributo cliente con el que podremos interactuar con el manejo de datos de la clase cliente.

```
public cltDP() { //utilizamos el constructor de la clase clienteDP para inicializar sus atributos
```

```
    this.codigo =0;
```

```
    this.nombre ="";
```

```
    this.ruc ="";
```

```
    this.direccion ="";
```

```
    this.telefono ="";
```

```
    this.mail ="";
```

```
}
```

// Esta sección es para getter y setter es decir para establecer y tomar valores de los atributos de la clase

```
public int getCodigo() {
```

```
    return codigo; }
```

```
public void setCodigo(int codigo) {
```

```
    this.codigo = codigo;
```

```
}
```

```
public String getNombre() {
```

```
    return nombre; }
```

```
public void setNombre(String nombre) {
```

```
    this.nombre = nombre; }
```

```
public String getRuc() {
```

```
    return ruc; }
```

```
public void setRuc(String ruc) {
```

```
    this.ruc = ruc; }
```

```
public String getDireccion() {
```

```
    return direccion; }
```

```
public void setDireccion(String direccion) {
```

```
    this.direccion = direccion; }
```

```
public String getTelefono() {
```

```
    return telefono; }
```

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

```
public void setTelefono(String telefono) {
    this.telefono = telefono; }
public String getMail() {
    return mail; }
public void setMail(String mail) {
    this.mail = mail; }
public boolean verificarDP(){
    if (cliente.verificarMD())
        return true;
    else
        return false; }
public int MaxDP(){
return cliente.maxMD(); }
public void insertarDP(){
    cliente.insertarMD(); }
public void modificarDP()
{
    cliente.modificarMD();
}

public void eliminarDP()
{
    cliente.eliminarMD();
}

public Object[][] datosCliente()
{
    return cliente.obtenerCedula();
}

}
```

MD

```
package md;

import dp.cltDP;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import utilitarios.conexion;

/**
 *
 * @author osvaldo
 */
public class cltMD {
    cltDP cliente;
    private Connection con;
    private Statement stmt;
    private ResultSet result;

    public cltMD(cltDP para)
    {
        this.cliente=para;
        conexion a =new conexion();
        con=a.conectar();
    }
    public boolean verificarMD ()
    {
        boolean band=false;
        try
        {
```

```
        stmt=con.createStatement();

        result=stmt.executeQuery("SELECT * FROM cliente WHERE
cltruc='"+cliente.getRuc()+"'");
        while(result.next())
        {
            cliente.setCodigo(result.getInt(1));
            cliente.setNombre(result.getString(2).trim());
            cliente.setRuc(result.getString(3).trim());
            cliente.setDireccion(result.getString(4).trim());
            cliente.setTelefono(result.getString(5).trim());
            cliente.setMail(result.getString(6).trim());
            cliente.setCiruc(result.getInt(7));
            cliente.setRuc(cliente.getRuc().trim());

            band=true;

        }

    }
    catch(Exception EX)
    {
        System.out.println(EX.getMessage());
    }
    return band;
}

public int maxMD(){
    int codigo=0;
    try
    {
        stmt=con.createStatement();
```

Diseño e implementación de un sistema de facturación para el restaurante "La Fogata"

```
String mayor="SELECT MAX(cltcdg)FROM cliente";
result=stmt.executeQuery(mayor);
while(result.next())
{
    codigo=result.getInt(1);

}
codigo+=1;
}catch(Exception EX)
{
    System.out.println(EX.getMessage());
}
return codigo;
}
public void insertarMD ()
{
    try
    {
        stmt=con.createStatement();
        stmt.executeUpdate("INSERT INTO cliente
VALUES("+cliente.getCodigo()
        +",""+cliente.getNombre()
        +",""+cliente.getRuc()
        +",""+cliente.getDireccion()
        +",""+cliente.getTelefono()
        +",""+cliente.getMail()
        +",""+cliente.getCiruc()+");");

    }
    catch(Exception EX)
```

Diseño e implementación de un sistema de facturación para el restaurante "La Fogata"

```
{
    System.out.println(EX.getMessage());
}

}

public void eliminarMD()
{
    try
    {
        stmt=con.createStatement();
        stmt.executeUpdate("DELETE FROM cliente WHERE
cltruc='"+cliente.getRuc()+"");

    }
    catch(Exception EX)
    {
        System.out.println(EX.getMessage());
    }
}

public void modificarMD()
{
    try
    {
        stmt=con.createStatement();
        stmt.executeUpdate("UPDATE cliente SET
cltnmb='"+cliente.getNombre()+

        "','cltruc='"+cliente.getRuc()+
```

Diseño e implementación de un sistema de facturación para el restaurante "La Fogata"

```
        ",cltdrcn="+cliente.getDireccion()+
        ",clttlf="+cliente.getTelefono()+
        ",cltmail="+cliente.getMail()+
        ",cltciruc="+cliente.getCiruc()+
        " WHERE cltcdg="+cliente.getCodigo()+"";");
    }
    catch(Exception EX)
    {
        System.out.println(EX.getMessage());
    }
}

public Object [][] DatosUsuarios(){
    int reg=0;
    try
    {
        stmt=con.createStatement();
        result=stmt.executeQuery("SELECT count(1) as total FROM cliente");
        result.next();

        reg=result.getInt("total");
        result.close();
    }
    catch(Exception EX)
    {
        System.out.println(EX.getMessage());
    }

    Object[][] datos=new String[reg][6];
    try{
```

Diseño e implementación de un sistema de facturación para el restaurante "La Fogata"

```
        stmt=con.createStatement();
        result=stmt.executeQuery("SELECT * FROM cliente ORDER BY
cltnmb");
        int i=0;
        while(result.next())
        {
            datos[i][0]= result.getInt("cltcdg")+"";
            datos[i][1]=result.getString("cltnmb");
            datos[i][2]=result.getString("cltruc");
            datos[i][3]=result.getString("cltdrcn");
            datos[i][4]=result.getString("clttlf");
            datos[i][5]=result.getString("cltmail");
            datos[i][6]=result.getString("cltciruc");

            i++;
        }
        result.close();stmt.close();

    }catch(Exception EX)
    {
        System.out.println(EX.getMessage());
    }

return datos;
}
```

```
public String [][] obtenerCedula()
{
    int reg=0;
    int i=0;

    try
    {
        stmt=con.createStatement();
        result=stmt.executeQuery("SELECT count(1) as total FROM cliente");
        result.next();
        reg=result.getInt("total");

        result.close();
    }
    catch(Exception EX)
    {
        System.out.println(EX.getMessage());
    }

    String [][] cedula=new String[reg][3];

    try{
        stmt=con.createStatement();
        result=stmt.executeQuery("select cltnmb, cltruc,cltcdg from cliente");
        while(result.next())
        {
            cedula[i][0]=result.getString("cltnmb");
            cedula[i][1]=result.getString("cltruc");
            cedula[i][2]=result.getInt("cltcdg")+"";
        }
    }
```

```
        i++;  
    }  
}  
catch(Exception EX)  
{  
    System.out.println(EX.getMessage());  
}  
return cedula;  
  
}  
}
```

Botón Guardar

```
control=true;  
  
    controles();//función implementada para comprobar si los datos  
ingresados están correctos  
  
    if(!control){  
        JOptionPane.showMessageDialog(this, "Los campos coloreados son  
obligatorios", "Atención", //mensaje y titulo  
        JOptionPane.PLAIN_MESSAGE);  
  
    }else{  
        if(!modificar)  
  
        {  
            int res=JOptionPane.showConfirmDialog(this, "Seguro que desea  
ingresar el nuevo Cliente?", "Confirmación", JOptionPane.YES_NO_OPTION);  
            if(res==JOptionPane.YES_OPTION){  
//guardo los datos en la clase clienteDP  
                cliente.setCodigo(Integer.parseInt(tfcodigo.getText()));  
                cliente.setDireccion(tadir.getText().trim().toLowerCase());  
                cliente.setTelefono(tftelefono.getText().trim());
```

Diseño e implementación de un sistema de facturación para el restaurante "La Fogata"

```
cliente.setNombre(tfnombre.getText().trim().toUpperCase());
```

```
cliente.setRuc(tfruc.getText().trim());
```

```
cliente.setMail(tfmail.getText().trim().toLowerCase());
```

```
//verifico si es ruc, cedula o pasaporte
```

```
if(rbCedula.isSelected())&&10==tfruc.getText().trim().length()
```

```
cliente.setCiruc(1);
```

```
if(rbRuc.isSelected())&&13==tfruc.getText().trim().length()
```

```
cliente.setCiruc(2);
```

```
if(rbPass.isSelected())
```

```
cliente.setCiruc(3);
```

```
///inserto cliente
```

```
cliente.insertarDP();
```

```
borrar();
```

```
eliminar=false;
```

```
JOptionPane.showMessageDialog(this, "El Cliente fue ingresado exitosamente", "Insertar Cliente",
```

```
JOptionPane.PLAIN_MESSAGE);
```

```
tfruc.requestFocus();
```

```
}
```

```
}else{
```

```
int res=JOptionPane.showConfirmDialog(this, "Seguro que desea modificar el Cliente?", "Confirmación", JOptionPane.YES_NO_OPTION);
```

```
if(res==JOptionPane.YES_OPTION){
```

```
cliente.setDireccion(tadir.getText().trim().toLowerCase());
```

```
cliente.setTelefono(tftelefono.getText().trim());
```

```
cliente.setNombre(tfnombre.getText().trim().toUpperCase());
```

```
cliente.setMail(tfmail.getText().trim().toLowerCase());
```

```
if(rbCedula.isSelected())&&10==tfruc.getText().trim().length()
```

```
cliente.setCiruc(1);
```

Diseño e implementación de un sistema de facturación para el restaurante "La Fogata"

```
        if(rbRuc.isSelected()&&13==tfruc.getText().trim().length())
            cliente.setCiruc(2);
        if(rbPass.isSelected())
            cliente.setCiruc(3);

        cliente.modificarDP();
        eliminar=true;

        modificar=false;
        JOptionPane.showMessageDialog(this, "El Cliente fue modificado
        exitosamente", "Modificar Cliente",
            JOptionPane.PLAIN_MESSAGE);
tfruc.requestFocus();
        }}
    }
}
}else
{ JOptionPane.showMessageDialog(this, "El Ruc o Numero de Cedula no es
valido", "Atención", //mensaje y titulo
        JOptionPane.PLAIN_MESSAGE);
tfruc.requestFocus();}

//función creada para verificar si los datos a ingresar son correctos, si no
se pintan de color los jtextfield respectivos
private void controles() {

        Color color=new Color(153,202,0);
        //tfnombre
        //tfruc
        //tftelefono
        //tfmail

        if(tfnombre.getText().length()<5||tfnombre.getText()==null){control=false;
            tfnombre.setBackground(color);}else
```

```
tfnombre.setBackground(Color.white);

if(tfruc.getText().length()<5||tfruc.getText()==null){control=false;
    tfruc.setBackground(color);}else
    tfruc.setBackground(Color.white);

if(tftelefono.getText().length()<5||tftelefono.getText()==null){control=false;
    tftelefono.setBackground(color);}else
    tftelefono.setBackground(Color.white);

if(tfmail.getText().length()<5||tfmail.getText()==null){control=false;
    tfmail.setBackground(color);}else
    tfmail.setBackground(Color.white);
    if(!util.validarEmail(tfmail.getText().toLowerCase().trim())){control=false;
    tfmail.setBackground(color);
        JOptionPane.showMessageDialog(this, "El correo electronico no es valido",
        "Atención", //mensaje y titulo
            JOptionPane.PLAIN_MESSAGE);
    }else
        tfmail.setBackground(Color.white);

    }
```

Algoritmo para comprobar cedula

```
/**
 *
 * @author osvaldo
 */
public class utilitarios {
    private int provincias=24;
```

```
public Boolean validacionCedula(String cedula) {
    boolean isValid = false;
    if (cedula == null || cedula.length() != 10) {
        isValid = false;
    }else{
        int prov = Integer.parseInt(cedula.substring(0, 2));

        if (!((prov > 0) && (prov <= provincias))) {
            isValid = false;
        }else{

            int[] d = new int[10];
            for (int i = 0; i < d.length; i++) //convierto todos los digitos y los
inserto en el arreglo
            {
                d[i] = Integer.parseInt(cedula.charAt(i) + "");
            }

            int imp = 0;
            int par = 0;
            ///
            for(int i=1;i<d.length-1;i+=2)//suma de pares *1

            par+=d[i];

            for(int i=0;i<d.length;i+=2)//suma de impares*2

            {
                if(d[i]*2>9)
                    imp=imp+(d[i]*2-9);
            }
        }
    }
}
```

Diseño e implementación de un sistema de facturación para el restaurante "La Fogata"

```
else
    imp=imp+d[i]*2;

}
```

```
int suma = imp + par;
```

```
int d10 = Integer.parseInt(String.valueOf(suma + 10).substring(0,
1) + "0") - suma;// subo a la siguiente decena y le resto la sumatoria de pares e
impares
```

```
if(d10==10)
d10=0;
```

```
if (d10 == d[9]) {
    isValid = true;
} else {
    isValid = false;
}
}}
```

```
return isValid;
```

```
}
```

```
public Boolean validacionRUC(String cedula) {
    boolean isValid = false;
    if (cedula == null || cedula.length() != 13) {
        isValid = false;
    }else{
        int prov = Integer.parseInt(cedula.substring(0, 2));
```

```
        if (!(prov > 0) && (prov <= provincias)) {
            isValid = false;
        }else{
int ruc = Integer.parseInt(cedula.substring(10));

        if (!(ruc == 001)) {
            isValid = false;
        }else{
            int[] d = new int[10];
            for (int i = 0; i < d.length; i++) //convierto todos los digitos y los
inserto en el arreglo
            {
                d[i] = Integer.parseInt(cedula.charAt(i) + "");
            }

            int imp = 0;
            int par = 0;
            ///
            for(int i=1;i<d.length-1;i+=2)//suma de pares *1

            par+=d[i];

            for(int i=0;i<d.length;i+=2)//suma de impares*2

            {
                if(d[i]*2>9)
                    imp=imp+(d[i]*2-9);
                else
                    imp=imp+d[i]*2;
```

```
}
```

```
int suma = imp + par;
```

```
int d10 = Integer.parseInt(String.valueOf(suma + 10).substring(0, 1) + "0") - suma;// subo a la siguiente decena y le resto la sumatoria de pares e impares
```

```
if(d10==10)
```

```
d10=0;
```

```
if (d10 == d[9]) {
```

```
    isValid = true;
```

```
} else {
```

```
    isValid = false;
```

```
}
```

```
}
```

```
}}
```

```
return isValid;
```

```
}
```

```
public Boolean validacionRUCJUR(String cedula) {
```

```
    boolean isValid = false;
```

```
int[] coeficientes={4,3,2,7,6,5,4,3,2};
```

```
if (cedula == null || cedula.length() != 13) {
```

```
    isValid = false;
```

```
}else{
```

```
int prov = Integer.parseInt(cedula.substring(0, 2));

    if (!((prov > 0) && (prov <= provincias))) {
        isValid = false;
    }else{
int ruc = Integer.parseInt(cedula.substring(10));

        if (!(ruc == 001)) {
            isValid = false;
        }else{
            int[] d = new int[10];
            for (int i = 0; i < d.length; i++) //convierto todos los digitos y los
inserto en el arreglo
            {
                d[i] = Integer.parseInt(cedula.charAt(i) + "");
            }

            int aux,prueba;
            int suma=0;
            for(int i=0;i<d.length-1;i++)
            {d[i]=d[i]*coeficientes[i];
                suma+=d[i];
            }

            aux=suma%11;
            prueba=11-aux;

            if(aux==0)
                prueba=0;
```

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

```
        if (prueba == d[9]) {
            isValid = true;
        } else {
            isValid = false;
        }
    }
}

return isValid;
}

public Boolean validacionRUCCEP(String cedula) {
    boolean isValid = false;
    int[] coeficientes={3,2,7,6,5,4,3,2,7};
    if (cedula == null || cedula.length() != 13) {
        isValid = false;
    }else{
        int prov = Integer.parseInt(cedula.substring(0, 2));

        if (!(prov > 0) && (prov <= provincias)) {
            isValid = false;
        }else{
            int ruc = Integer.parseInt(cedula.substring(10));

            if (!(ruc == 001)) {
                isValid = false;
            }else{
                int[] d = new int[10];
                for (int i = 0; i < d.length; i++) //convierto todos los digitos y los
                inserto en el arreglo
            {
```

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

```
        d[i] = Integer.parseInt(cedula.charAt(i) + "");
    }

    int aux,prueba;
    int suma=0;
    for(int i=0;i<d.length-1;i++)
    {d[i]=d[i]*coeficientes[i];
    suma+=d[i];

    }

    aux=suma%11;
    if(aux==0)
    return true;

    prueba=11-aux;

    if (prueba == d[9]) {
        isValid = true;
    } else {
        isValid = false;
    }
}
}}

return isValid;
}
```

```
public boolean validarRUC(String cedula){
```

```
if(validacionCedula(cedula))
return true;
else
  if(validacionRUC(cedula))
    return true;
  else
    if(validacionRUCJUR(cedula))
      return true;
    else
      if(validacionRUCEP(cedula))
        return true;
else
  return false;

}
```

Reporte de declaración

El código SQL para el reporte de declaración.

```
SELECT
  extract (year from cmp.cmpfch) as ano,
  extract (month from cmp.cmpfch)as mes,
  sum(cmp.cmpiva0) as iva0,
  sum(cmp.cmpiva12) as iva12,
  sum(cmp.cmpiva) as iva,
  sum(cmp.cmpttl) as total
FROM
  public.compra cmp,
  public.detallecompra del
where
```

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

```
cmp.cmpcdg = del.cmpcdg  
and cmp.cmpfch>= ${P{fechaini}}  
and cmp.cmpfch<= ${P{fechafin}}  
group by ano,mes  
order by ano,mes
```



Manual de Usuario del sistema:

BonFireSys.

Sistema de facturación del restaurante “La Fogata”

Versión 1.0

Oswaldo Tapia

15/12/2015

Índice

1.	A tomar en cuenta	160
2.	Ingreso al sistema	161
3.	Cerrar sesión	161
4.	Administrar usuarios	161
4.1	Crear usuario.....	162
4.2	Modificar usuario	163
4.3	Eliminar usuario.....	163
5.	Administrar proveedores y clientes	164
5.1	Crear proveedor o cliente	166
5.2	Modificar proveedor o cliente	166
5.3	Eliminar proveedor o cliente.....	168
6.	Administrar productos y menús	169
6.1	Crear producto o menú	170
6.2	Modificar producto o menú	170
6.3	Eliminar producto o menú.....	172
7.	Compras	173
7.1	Ingresar compra	173
8.	Facturación	176
8.1	Ingresar Factura	177

7. A tomar en cuenta





- ✓ Tenemos dos usuarios:

usuarios	1	2
nombre	user	admin
contraseña	user	admin
nivel	cajero	administrador

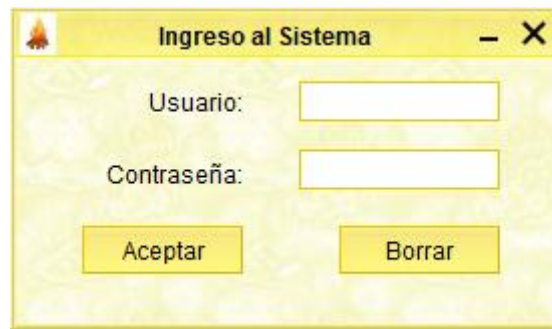
- ✓ Para todos los números decimales se usa el punto “.” No la coma “,”

correcto	incorrecto
2.5	2,5
3.5	3,5
4.26	4,26

- ✓ Los siguientes botones serán encontrados en las ventanas del sistema

Botón	Nombre	Función
	Guardar	Almacena información en la base de datos
	Limpiar	Borra los datos de la pantalla
	Eliminar	Borra la información de la base de datos
	Cerrar	Cierra la ventana

8. Ingreso al sistema



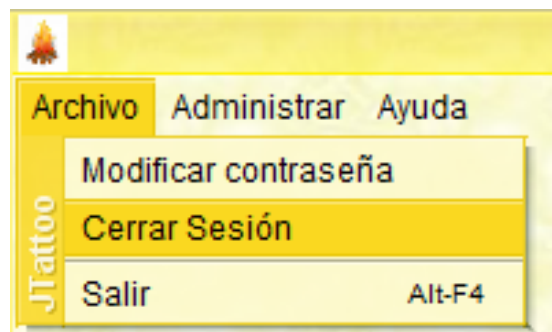
Para el ingreso al sistema, colocamos el nombre de usuario y la contraseña vale recordar que el nombre de usuario no distingue entre mayúsculas y minúsculas, **pero la contraseña si lo hace.**

Tenemos 3 intentos para ingresar la contraseña correctamente, si lo hacemos mal por una cuarta ocasión el programa se cerrara automáticamente.

9. Cerrar sesión

una cuarta ocasión el programa

1. Abra el menú **Archivo**.
2. Elija la opción **Cerrar sesión**.



Y podemos ingresar al sistema con otro usuario.

10. Administrar usuarios

Condiciones de administrar usuarios:

N	Condición	recomendación	ejemplo
1	Nombre de usuario	Debe tener mínimo 4 caracteres	prueba
2	Nombre de usuario	Al guardar las letras se transforman en minúsculas	"PrueBa" se convertira en "prueba"
3	Contraseña	Debe tener mínimo 6 caracteres entre números y letras	prba01
4	Contraseña	Tiene que ser fácil de recordar	

Tabla 1: Condiciones para usuarios.

3. Abra el menú **Administrar**.
4. Elija el menú **Usuarios**.
5. Nos aparece la ventana de la Figura 1

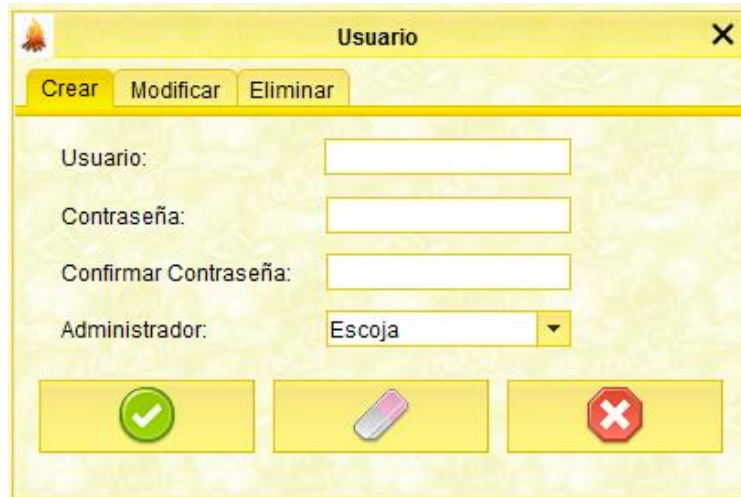



Figura 1: Ventana administrar usuarios.

Esta ventana consta de tres pestañas crear, modificar y eliminar. Figura 2



Figura 2: Pestañas de la ventana administrar usuarios.

10.1 Crear usuario

1. Elegimos la pestaña **Crear** de la ventana **Usuario**
2. Ingresamos el nombre del usuario, ver tabla 1
3. Ingresamos la contraseña.
4. Repetimos la contraseña.
5. Seleccionamos el nivel de usuario.
6. Presionamos el botón guardar 
7. Emergen las siguientes ventanas Figura 3, Figura 4.

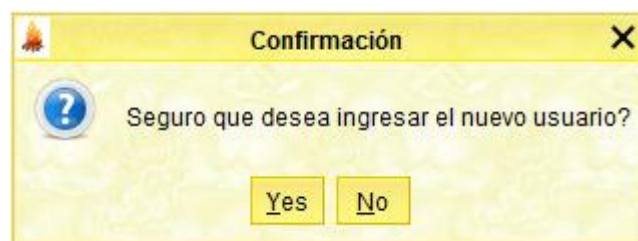


Figura 3: Confirmación de ingreso de usuario.

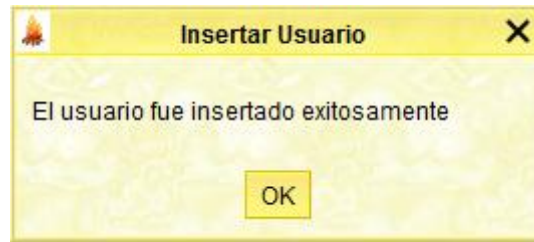



Figura 4: Mensaje de confirmación de ingreso de usuario.

10.2 Modificar usuario

Nuestro nombre de usuario aparece automáticamente.

1. Elegimos la pestaña **Modificar** de la ventana **Usuario**
2. El nombre de nuestro usuario aparece automáticamente.
3. Ingresamos la contraseña actual.
4. Ingresamos la nueva contraseña, ver tabla 1
5. Repetimos la nueva contraseña.

6. Presionamos el botón guardar 
7. Emergen las siguientes ventanas Figura 5, Figura 6.

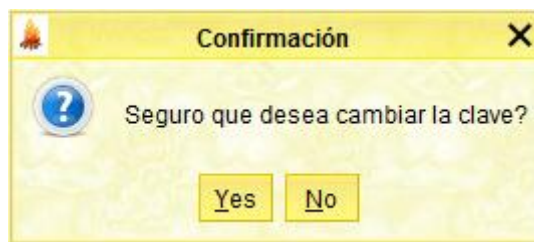


Figura 5: Confirmación de modificación de clave.

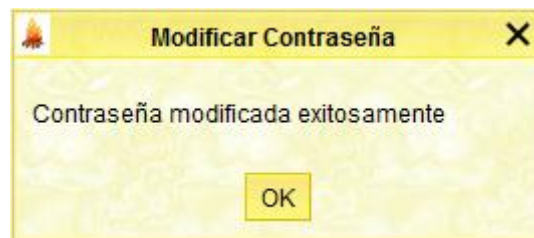



Figura 6: Mensaje de confirmación de modificación de clave.

10.3 Eliminar usuario

Existe una lista con todos los usuarios que podríamos eliminar.

1. Elegimos la pestaña **Eliminar** de la ventana **Usuario**
2. Seleccionamos el usuario a eliminar.
3. Presionamos el botón eliminar 
4. Aparecen las siguientes ventanas Figura 7, Figura 8.

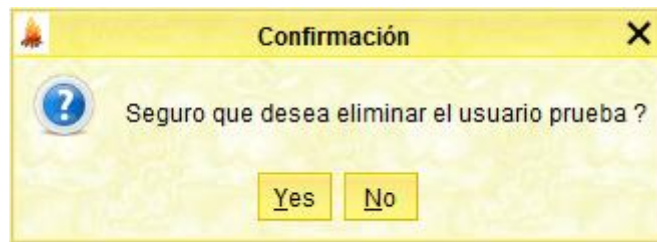


Figura 7: Confirmación de eliminación de usuario.

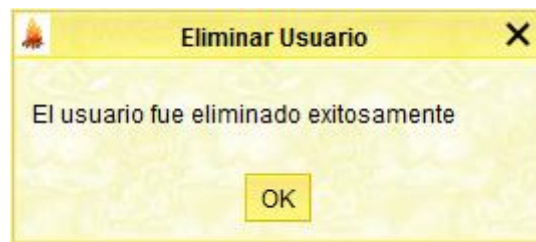


Figura 8: Mensaje de confirmación de eliminación de usuario.

11. Administrar proveedores y clientes

Se las agrupa entre sí porque solo hay una diferencia entre sus maneras de ser administrados.

Condiciones de proveedores y clientes.

N	Condición	recomendación	ejemplo
1	Nombre de cliente o proveedor	Debe tener mínimo 5 caracteres	PRUEBA
2	Nombre de cliente o proveedor	Al guardar las letras se transforman en mayúsculas	"PrueBa" se convertirá en "PRUEBA"
3	Dirección de cliente o proveedor	Al guardar las letras se transforman en minúsculas	"QuiTo" se convertirá en "quito"
4	Dirección de cliente o proveedor	No es obligatorio	
5	Teléfono de cliente o proveedor	Solo se aceptan números además de "-" y "/"	
6	Teléfono de cliente o proveedor	Debe tener mínimo 7 caracteres	
7	Correo electrónico del cliente o proveedor	Debe ser un correo electrónico real	prba@mail.ru
8	Correo electrónico del cliente o proveedor	Al guardar las letras se transforman en minúsculas	"prBA@mail.ru" se convertirá en "prba@mail.ru"
9	Descripción de proveedor	Al guardar las letras se transforman en minúsculas	"PlasTicos" se convertirá en "plasticos"
10	Descripción de proveedor	No es obligatorio	

Tabla 2: Condiciones para proveedores y clientes.

1. Abra el menú **Administrar**.
2. Elija el menú **Cientes**, Figura 9 o **Proveedores** Figura 10



Figura 9: Abrir administrar **Cientes**.



Figura 10: Abrir administrar **Proveedores**.

3. Aparencen las siguientes ventanas **Cientes** Figura 11 o **Proveedores** Figura 12.

El código de proveedor o cliente se genera automáticamente.


A screenshot of a software window titled 'Clientes'. The window contains several input fields: 'Código:' with the value '7', 'RUC:', 'Nombre:', 'Dirección:', 'Teléfono:', and 'E-mail:'. Below the 'Código:' field are three radio buttons labeled 'Cédula', 'RUC', and 'Pasaporte', with 'Cédula' selected. At the bottom of the window are four buttons: a green checkmark, a red minus sign, a pink pill icon, and a red X.

Figura 11: Ventana administrar **Cientes**.



Figura 12: Ventana administrar **Proveedores**.

11.1 Crear proveedor o cliente

1. Elegimos si nuestro cliente o proveedor tiene cedula, RUC o pasaporte.
2. Ingresamos el número de cedula, RUC o pasaporte.
3. Ingresamos el nombre del cliente o proveedor.
4. Ingresamos la dirección.
5. Ingresamos el teléfono.
6. Ingresamos el correo electrónico.
7. Ingresamos la descripción en el proveedor.
8. Presionamos el botón guardar 
9. Surgen las ventanas de confirmación.

11.2 Modificar proveedor o cliente

1. Posicionamos el cursor en el campo RUC y presionamos F3. *Figura 13*



Figura 13: Posicionar el cursor en el campo RUC.

2. Surgen las siguientes pantallas **Elegir proveedor** *Figura 14* o **Elegir Cliente** *Figura 15*



Figura 14: Ventana **Elegir Proveedor**.



Figura 15: Ventana **Elegir Cliente**.

3. Elegimos nuestro cliente o proveedor.

- **Con doble clic**

- **O con el botón de confirmación**



4. Automáticamente los datos son presentados en la ventana **Cientes** *Figura 16* o **Proveedores** *Figura 17*

Clientes

Código: 4

Cédula RUC Pasaporte

RUC: 1500815756

Nombre: OSVALDO RAUL TAPIA TATES

Dirección: quito

Teléfono: 3200594

E-mail: ort_t@hotmail.com

Figura 16: Ventana **Clientes**.

Proveedores

Código: 7

Cédula RUC Pasaporte

RUC: 1001864436001

Nombre: TATES PEREIRA JENNY


Dirección: quito

Teléfono: 3200594

E-mail: liz_t@gmail.com

Descripción: plasticos

Figura 17: Ventana **Proveedores**.


5. Modificamos el campo deseado ver Tabla 2
6. Presionamos guardar  y se presentan las ventanas de confirmación.

11.3 Eliminar proveedor o cliente

1. Posicionamos el cursor en el campo RUC y presionamos F3. *Figura 13*
2. Surgen las siguientes pantallas **Elegir proveedor** *Figura 14* o **Elegir Cliente** *Figura 15*
3. Elegimos nuestro cliente o proveedor.
 - **Con doble clic**

➤ **O con el botón de confirmación**



4. Automáticamente los datos son presentados en la ventana **Cientes** *Figura 16* o **Proveedores** *Figura 17*
5. Presionamos el botón eliminar  y surgen las ventanas de confirmación.

12. Administrar productos y menús

Se las agrupa entre sí porque no hay diferencia entre sus maneras de ser administrados.

N	Condición	recomendación	ejemplo
1	Nombre de producto o menú	Debe tener mínimo 3 caracteres	
2	Nombre de producto o menú	Al guardar las letras se transforman en mayúsculas	"ArroZ" se convertirá en "ARROZ"
3	Precio	Para los números decimales se usa el punto "."	2.54 3.55 2.40
4	Descripción de proveedor	Al guardar las letras se transforman en minúsculas	"AGua MiNeral" se convertirá en "agua mineral"

Tabla 3: Condiciones para productos y menús.

1. Abra el menú **Administrar**.
2. Elija el menú **Productos**. *Figura 18* o **Menús** *Figura 19*



Figura 18: Abrir administrar **Productos**.



Figura 19: Abrir administrar **Menús**.

3. Aparecen las siguientes ventanas **Productos** *Figura 20* o **Menús** *Figura 21*.




Figura 20: Ventana administrar **Producto**.



Figura 21: Ventana administrar **Menú**.

12.1 Crear producto o menú

1. Ingresamos el nombre del producto o menú.
2. Ingresamos el precio.
3. Ingresamos la descripción.
4. Marcamos la casilla de gravar IVA según sea el caso.
5. Presionamos el botón guardar 
6. Nos aparecen las ventanas de confirmación.

12.2 Modificar producto o menú

1. Posicionamos el cursor en el campo Nombre y presionamos F3. *Figura 22*



Figura 22: Posicionar el cursor en el campo Nombre.

2. Emergen las siguientes pantallas **Elegir Menú** *Figura 23* o **Elegir Producto** *Figura 24*

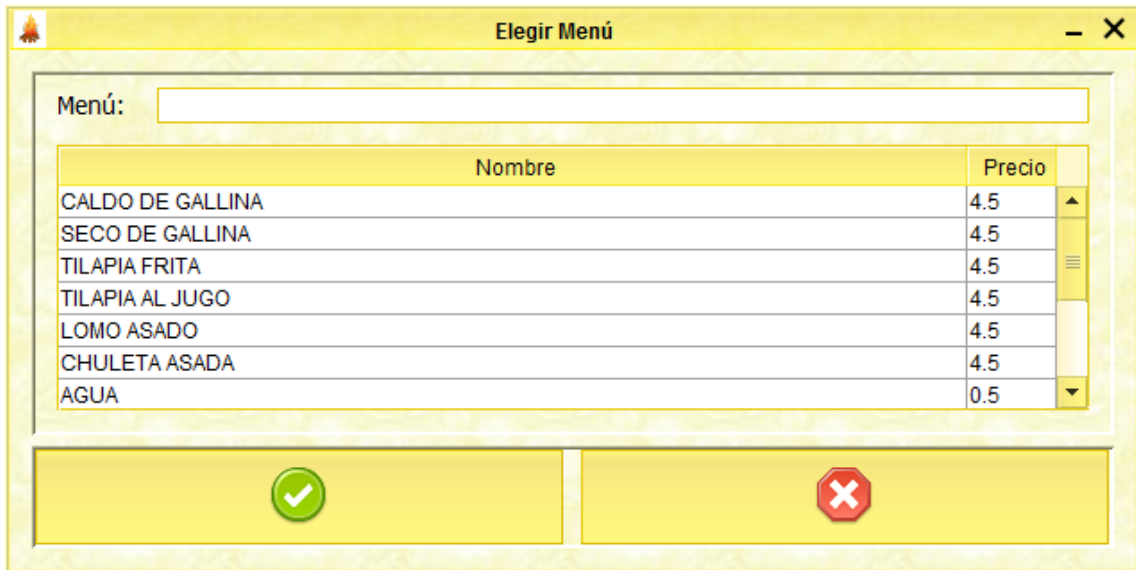


Figura 23: Ventana **Elegir Menú**.



Figura 24: Ventana **Elegir Producto**.

3. Elegimos nuestro menú o producto.

➤ **Con doble clic**

➤ **O con el botón de confirmación**



4. Automáticamente los datos son presentados en la ventana **Menú** *Figura 25* o **Productos** *Figura 26*



Figura 25: Ventana **Menú**.







Figura 26: Ventana **Productos**.

5. Modificamos el campo deseado ver [Tabla 3](#).
6. presionamos guardar  y aparecen las ventanas de confirmación.




12.3 Eliminar producto o menú

1. Posicionamos el cursor en el campo Nombre y presionamos F3. [Figura 22](#)
2. Emergen las siguientes pantallas **Elegir Menú** [Figura 23](#) o **Elegir Producto** [Figura 24](#)
3. Elegimos nuestro menú o producto.
 - **Con doble clic**
 - **O con el botón de confirmación** 
4. Automáticamente los datos se muestran en la ventana **Menú** [Figura 25](#) o **Productos** [Figura 26](#)

- Presionamos el botón eliminar  y nos aparecerán las ventanas de confirmación.

13. Compras

Encontraremos los siguientes botones en la ventana de compras.

Botón	Nombre	Función
	Procesar	Comprueba si se ingresaron correctamente los datos proveedor y cantidades, las procesa realiza los cálculos para que la tabla detalle se llene automáticamente.
	Agregar Proveedor	En caso de necesitar agregar un nuevo proveedor también lo podemos hacer mediante este botón
	Imprimir	Luego de ingresar la factura, con este icono podremos imprimir la última factura ingresada

- Abra el menú **Administrar**.
- Elija el menú **Compras**.
- Nos aparece la ventana de la [Figura 27](#)



Figura 27: Elegir menú **compras**.

13.1 Ingresar compra

- Posicionamos el cursor en el campo RUC y presionamos F3. [Figura 28](#)

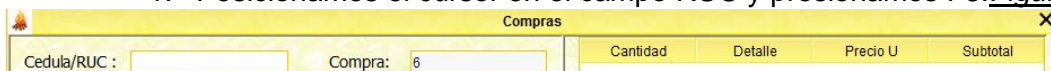


Figura 28: Posicionar el cursor en el campo RUC.

- Aparece la siguiente pantalla [Figura 29](#)



Figura 29: Ventana **Elegir Proveedor**.

3. Elegimos nuestro proveedor.

➤ **Con doble clic**

➤ **O con el botón de confirmación**



4. Automáticamente los datos aparecen en la ventana de compra *Figura 30*

The screenshot shows a form with the following fields:

- Cedula/RUC : 1001864436001
- Compra: 6
- Nombre: TATES PEREIRA JENNY
- Dirección: quito
- Teléfono: 3200594
- Fecha: Sun 12/06/2015

Figura 30: Datos del proveedor en la compra.

5. Ingresamos la cantidad de acuerdo a los productos que vamos a comprar.

6. Presionamos el botón procesar



y se cargarán los datos en

la tabla detalle *Figura 31*

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

Figura 31: Datos procesados compra.


7. En caso de que la factura no sea cancelada en efectivo. Caso contrario seguimos con el paso 8.

7.1 Ingresar los pagos haciendo clic en Efectivo y nos aparecerá la tabla de los pagos. *Figura 32*

Pagos	
Pagos	Cantidad
cheque	
efectivo	
otros	
transferencia	

Figura 32: Tabla para pagos en compras.

8. Presionamos el botón guardar  y emergen mensajes de confirmación.

9. En este punto se puede imprimir la compra con  y nos aparece la siguiente pantalla *Figura 33*

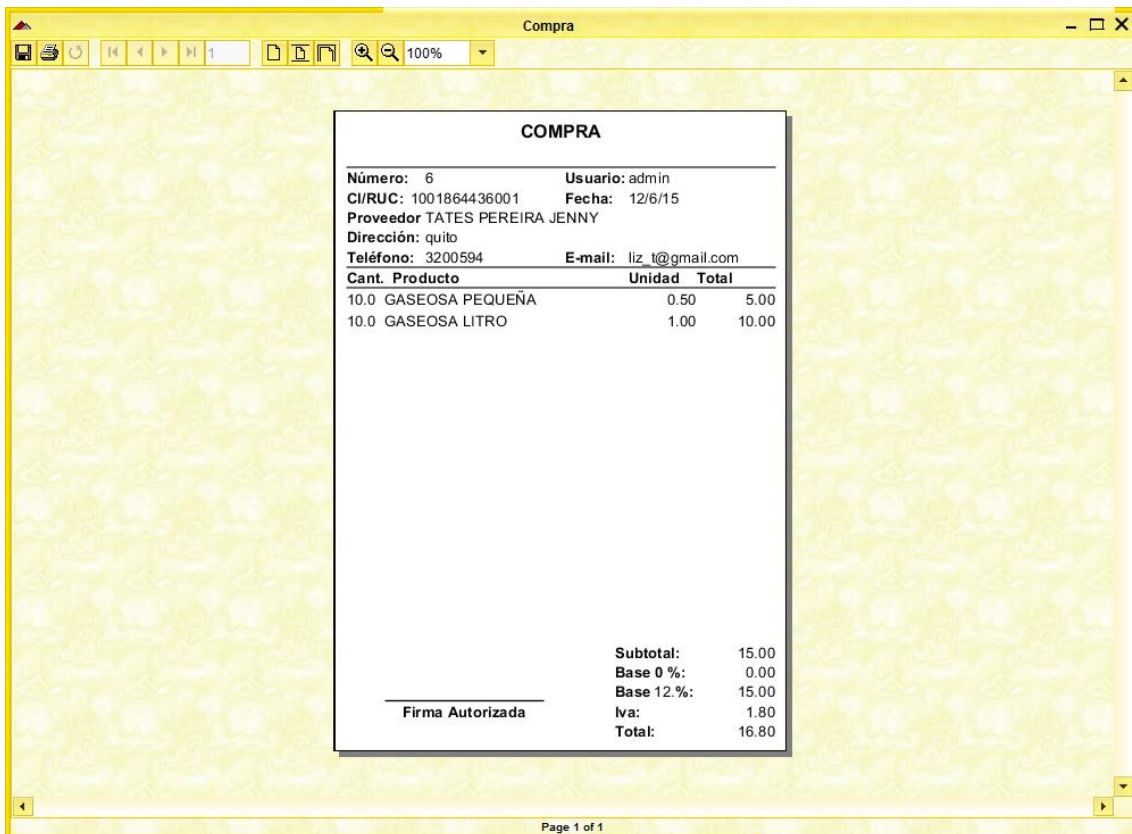


Figura 33: Reporte de compra.

14. Facturación

Encontraremos los siguientes botones en la ventana facturación.

Botón	Nombre	Función
	Procesar	Comprueba si se ingresaron correctamente los datos proveedor y cantidades, las procesa realiza los cálculos para que la tabla detalle se llene automáticamente.
	Agregar Proveedor	En caso de necesitar agregar un nuevo proveedor también lo podemos hacer mediante este botón
	Imprimir	Luego de ingresar la factura, con este icono podremos imprimir la última factura ingresada
	Anular	Nos sirve para eliminar una o un rango de facturas.

1. Abra el menú **Administrar**.
2. Elija el menú **Facturación**.
3. Nos aparece la ventana de la *Figura 34*



Figura 34: Elegir menú **Facturación**.

14.1 Ingresar Factura

1. Elegimos nuestro cliente. Para ello nos posicionamos en el RUC y presionamos F3 *Figura 35*

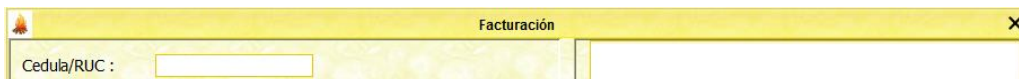


Figura 35: Posicionar el cursor en el campo RUC.

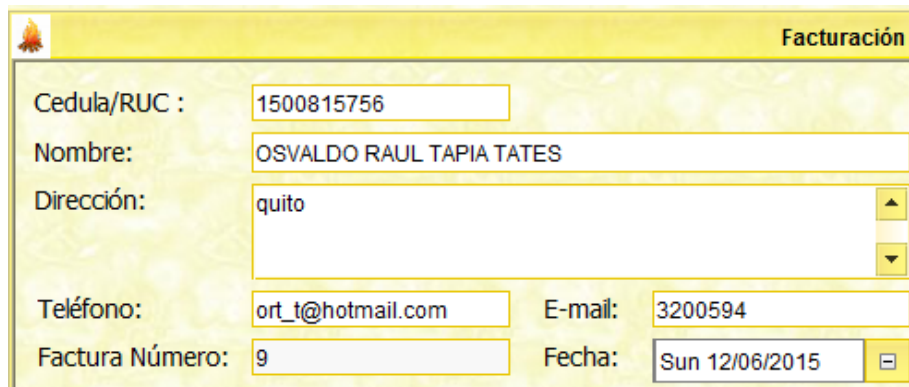
2. Nos aparecerá la siguiente pantalla *Figura 36*



Figura 36: Ventana **Elegir Cliente**.

3. Elegimos nuestro cliente.
 - **Con doble clic**
 - **O con el botón de confirmación** 
4. Automáticamente los datos emergen en la ventana de compra *Figura 37*

Diseño e implementación de un sistema de facturación para el restaurante “La Fogata”

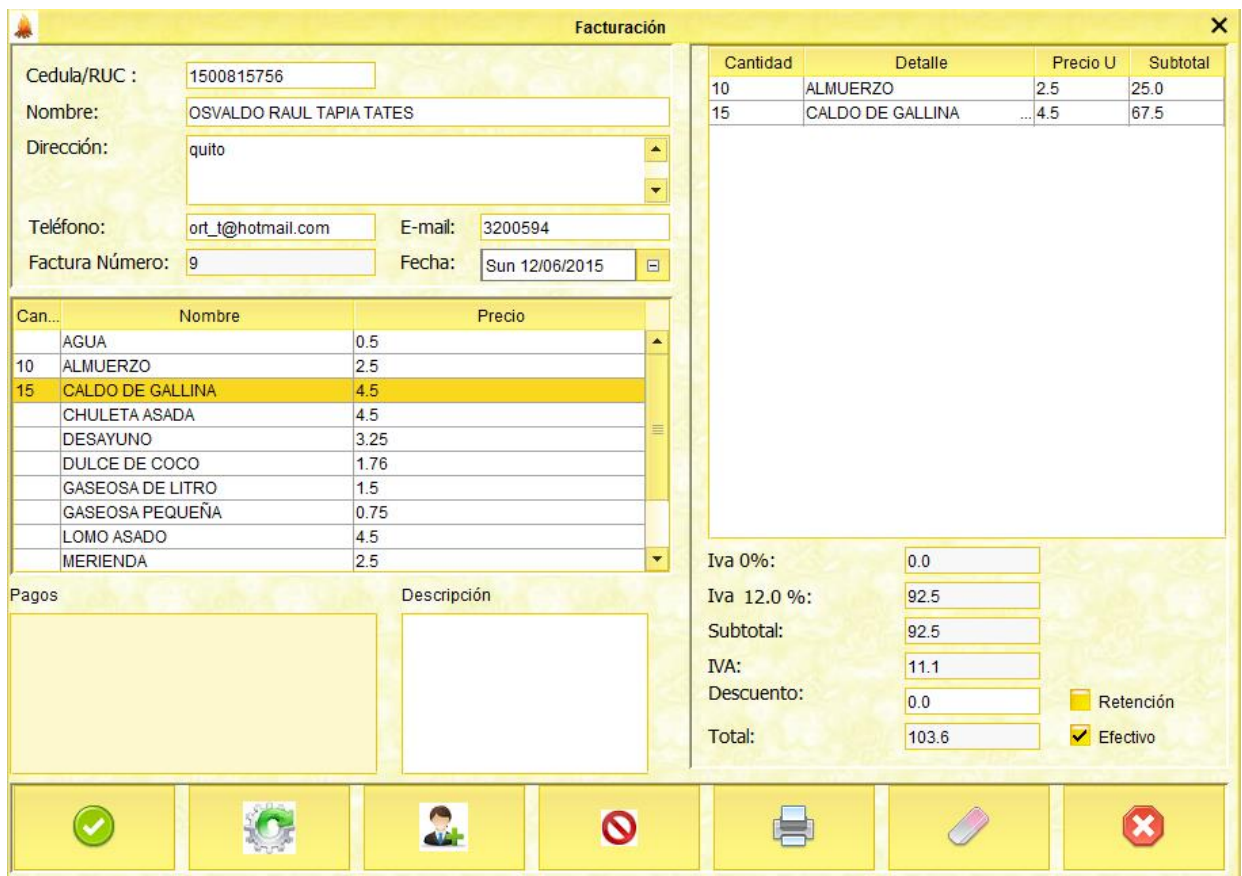


The screenshot shows a software window titled "Facturación" with a yellow background. It contains several input fields for customer information: "Cedula/RUC" (1500815756), "Nombre" (OSVALDO RAUL TAPIA TATES), "Dirección" (quito), "Teléfono" (ort_t@hotmail.com), "E-mail" (3200594), "Factura Número" (9), and "Fecha" (Sun 12/06/2015).

Figura 37: Datos del cliente en la factura.

5. Ingresamos la cantidad de acuerdo a los ítems que vamos a facturar.

6. Presionamos el botón procesar  y se cargarán los datos en la tabla detalle *Figura 38*



The screenshot shows the "Facturación" window with the same customer data as Figure 37. It also displays a table of items, a summary of taxes, and a total amount.

Can...	Nombre	Precio
	AGUA	0.5
10	ALMUERZO	2.5
15	CALDO DE GALLINA	4.5
	CHULETA ASADA	4.5
	DESAYUNO	3.25
	DULCE DE COCO	1.76
	GASEOSA DE LITRO	1.5
	GASEOSA PEQUEÑA	0.75
	LOMO ASADO	4.5
	MERIENDA	2.5

Cantidad	Detalle	Precio U	Subtotal
10	ALMUERZO	2.5	25.0
15	CALDO DE GALLINA	4.5	67.5

Iva 0%:	0.0
Iva 12.0 %:	92.5
Subtotal:	92.5
IVA:	11.1
Descuento:	0.0
Total:	103.6

Retención
Efectivo

Figura 38: Datos procesados factura.

7. En caso de que la factura no sea cancelada en efectivo, caso contrario seguimos con el paso 8.


7.1 Ingresar los pagos haciendo clic en Efectivo y nos aparecerá la tabla de los pagos. *Figura 39*

Pagos	
Pagos	Cantidad
cheque	
efectivo	
otros	
transferencia	

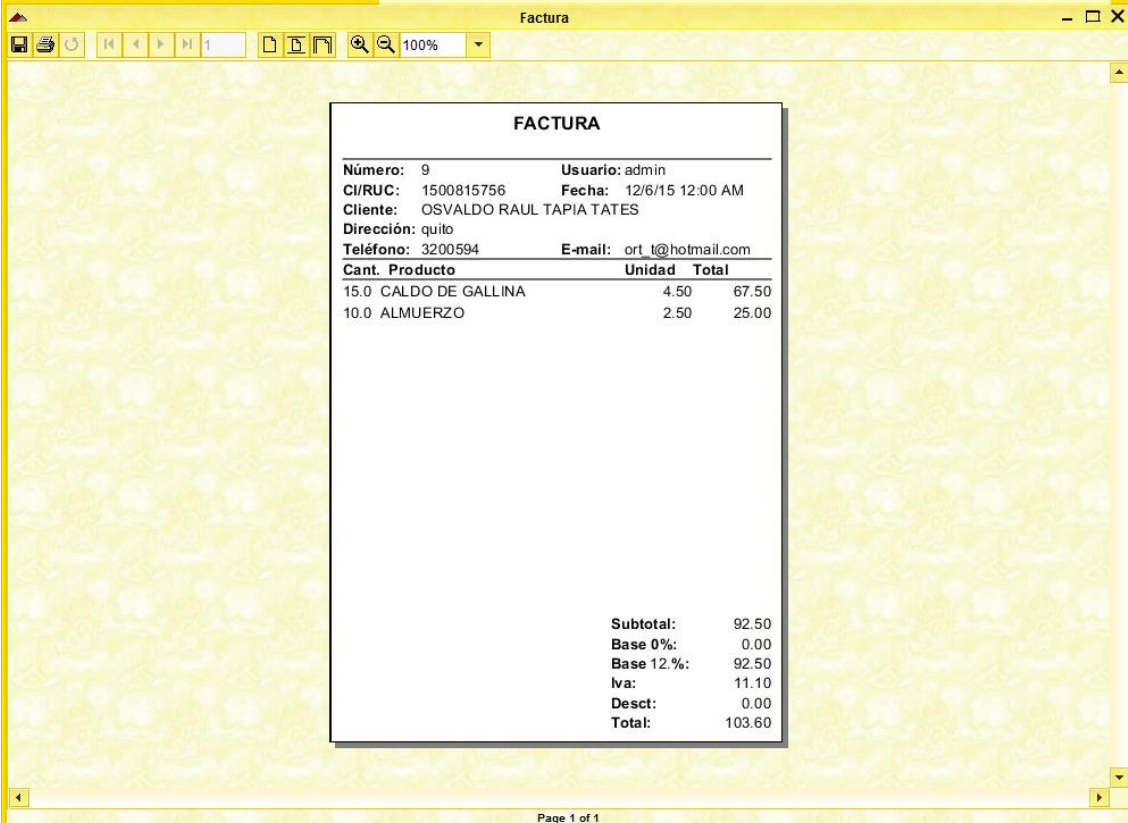
Figura 39: Tabla para pagos en facturas.

- De ser necesario insertamos el descuento respectivo.
- Además podemos marcar la casilla de retención, es solo para fines de declaraciones. (se lo puede visualizar en el reporte de declaraciones)

- Presionamos el botón guardar  y nos aparecerán los siguientes mensajes.

- En este punto se puede imprimir la factura con  y nos aparece la siguiente pantalla *Figura 40*

f



FACTURA			
Número:	9	Usuario:	admin
CI/RUC:	1500815756	Fecha:	12/6/15 12:00 AM
Cliente:	OSVALDO RAUL TAPIA TATES		
Dirección:	quito		
Teléfono:	3200594	E-mail:	ort_t@hotmail.com
Cant.	Producto	Unidad	Total
15.0	CALDO DE GALLINA	4.50	67.50
10.0	ALMUERZO	2.50	25.00
Subtotal:			92.50
Base 0%:			0.00
Base 12.0%:			92.50
Iva:			11.10
Desct:			0.00
Total:			103.60

Figura 40: Reporte de factura.