

**Sistema de información  
para Restaurantes en línea**

Carolina V Ching  
Universidad de Manizales  
[carolina@princetonasa.com](mailto:carolina@princetonasa.com)

## CONTENIDO

	<u>Pág.</u>
INTRODUCCIÓN.....	1
1. DESCRIPCIÓN DEL PROBLEMA .....	3
1.1 ANÁLISIS DE LA SITUACIÓN ACTUAL.....	4
1.2 FORMULACIÓN DEL PROBLEMA.....	4
2. OBJETIVOS.....	5
2.1 OBJETIVO GENERAL.....	5
2.2 OBJETIVOS ESPECÍFICOS.....	5
3. JUSTIFICACIÓN.....	6
4. MARCO TEÓRICO.....	7
4.1 CLASIFICACIONES BÁSICAS DE SISTEMAS GENERALES.....	7
4.2 UML.....	7
4.3 HERRAMIENTAS DE DESARROLLO.....	9
4.4 LENGUAJES DE PROGRAMACIÓN.....	11
4.5 SISTEMAS DE BASES DE DATOS.....	13
4.6 SISTEMAS PARA MODELAR SOFTWARE.....	15
4.7 ANTECEDENTES.....	16
5 METODOLOGÍA.....	18
5.1 TIPO DE TRABAJO.....	18
5.2 PROCEDIMIENTO.....	18
6. RESULTADOS.....	20
7. CONCLUSIONES.....	21
8. RECOMENDACIONES.....	22
BIBLIOGRAFIA.....	23
ANEXOS.....	25

## LISTA DE ANEXOS

	<u>Pág.</u>
<b>ANEXO A. Modelo de Requisitos.....</b>	<b>266</b>
<b>ANEXO B. Modelo de Análisis .....</b>	<b>300</b>
<b>ANEXO C. Modelo de Diseño .....</b>	<b>39</b>
<b>ANEXO D. Modelo de Pruebas.....</b>	<b>90</b>
<b>ANEXO E. Estudio y Selección de Alternativas .....</b>	<b>94</b>
<b>ANEXO F. Medición y Estimación .....</b>	<b>96</b>
<b>ANEXO G. SCRIPT para la creación de la base de datos .....</b>	<b>102</b>
<b>ANEXO H. Definición de la base de datos (aspectos técnicos).....</b>	<b>112</b>
<b>ANEXO I. Metodología de Desarrollo .....</b>	<b>118</b>
<b>ANEXO J. Manual Técnico .....</b>	<b>123</b>
<b>ANEXO K. Manual de UsuarioTécnico .....</b>	<b>146</b>

## **RESUMEN**

El sistema de información para restaurantes en línea es una herramienta que aumenta la productividad de los restaurantes tradicionales, permitiendo ampliar su margen de participación en el mercado, debido a su incursión en Internet.

Se convierte en una herramienta de uso general en la Web, donde los clientes pueden consultar el plato del día o los diferentes productos; permitiendo mediante un gestor comercial construido con base a los requerimientos, realizar pedidos.

El sistema está basado en una plataforma de desarrollo orientada a la Web, tanto en la fase de desarrollo como en el motor para la base de datos. Dicha característica permite la gestión de la información en forma distribuida.

El proyecto se desarrolla con los lineamientos de la ingeniería de software para dar calidad y oportunidad al producto, para tales efectos se utiliza el Proceso Unificado, el cual fue inicialmente concebido por modelos, es decir, para separar los requisitos, el análisis, el diseño y las pruebas.

## **ABSTRACT**

The system of information for restaurants in line is a tool that increases the productivity of the traditional restaurants, allowing to extend your margin of market share, due to his incursion in Internet.

It turns into a tool of general use into the Web, where the clients can consult the plate of the day or the different products; allowing by means of a commercial manager constructed with base to the requirements, to realize orders.

The system is based on a platform of development orientated to the Web, both on the phase of development and on the engine for the database. The above mentioned characteristic allows the management of the information in distributed form.

The project develops with the limits of the engineering software to give quality and opportunity to the product, for such effects there is in use the Unified Process, which was initially conceived by models, that is to say, to separate the requirements, the analysis, the design and the tests.

## INTRODUCCIÓN

Con los avances en los desarrollos tecnológicos, la masificación de los mismos crean los escenarios propicios para abordar procesos de cambio en cuanto a los aspectos de base en las TIC en Colombia. Dichos procesos responden al ingreso de las telecomunicaciones a la cotidianidad de las empresas.

Las sociedades económicas actuales observan la necesidad de contar y ofrecer su información en medios distribuidos, por esto una de las grandes posibilidades para el ingeniero de sistemas y telecomunicaciones es realizar una solución que integre estas áreas (los sistemas y las telecomunicaciones) como ejes principales para la resolución de un problema sentido de la región.

La presente propuesta surge como apoyo al servicio que prestan los restaurantes, permitiendo o facilitando la organización de la información, y especialmente ofrecer un valor agregado a sus clientes en cuanto a: rapidez en la obtención del servicio, el cual consiste en brindarles la posibilidad de realizar sus pedidos vía Web. Para esto se cuenta con el apoyo de un grupo de restaurantes y de la Casa de la Cultura de Cartago-Valle, entidad del gobierno que agrupa dichos negocios.

La propuesta se fundamenta en el concepto de montar un servidor en la sede principal de la Casa de la Cultura de Cartago-Valle, donde se instalará la aplicación de software que administrará la información de los restaurantes, cada restaurante tendrá en su local un computador con acceso a Internet para consultar los pedidos hechos por los clientes y estos tendrán disponible una página Web y por intermedio de ella realizarán sus pedidos; lo anterior corresponde a aspectos logísticos puntualmente, mas adelante se profundiza en el análisis de requisitos.

La base fundamental está regida en la centralización de los recursos, lo cual evitará costos para los restaurantes la compra de servidores, accesos a Internet con anchos de banda costosos y de administración, y permitirá a la Casa de la Cultura además de brindarle un servicio a los integrantes del gremio que agrupa, obtener estadísticas de consumo, gustos y de más información que se pueda obtener de la base de datos lograda en el desarrollo del proyecto.

Aunque es la misma aplicación, cada restaurante solo conocerá la información propia, ya que los accesos se manejan de manera restringida; lo anterior se enmarca como un requerimiento esencial de seguridad.

El beneficio para el cliente\* es que desde su oficina o desde cualquier otro sitio puede realizar un pedido a domicilio, y puede programar su hora y lugar de entrega.

La información que se va a manejar de cada restaurante es la referente a pedidos, procesos de administración e inventarios y distribución de sus productos (comidas); pretendiendo con lo anterior implementar un sistema de control que le apoye en la toma de decisiones.

El sistema de ventas vía Web contará con toda la infraestructura física y tecnológica necesaria para realizar y recibir pedidos vía Internet y poder realizar una adecuada y oportuna comercialización de fácil acceso donde el actor cliente interactúa con el sistema comercial utilizando las telecomunicaciones (TIC's) mediante la plataforma que nos brinda Internet y no como es tradicional visitando las instalaciones para realizar dicha compra.

La base fundamental de la propuesta es la comercialización de productos vía Internet (caso: restaurantes), permitiendo que los usuarios hagan los pedidos por este medio.

Como resultado final el proyecto ofrecerá un sistema comercial innovador, donde los clientes adquieren los productos del restaurante sin salir de la oficina u hogar y a domicilio, guardando un historial de su cuenta y ofreciendo la posibilidad de contar con perfiles prediseñados para obtener mayor agilidad.

---

\* Actor principal para el desarrollo del proyecto. Según la definición conceptual del proyecto, el cliente es toda persona interesada en recibir los beneficios de un restaurante Virtual.

## 1. DESCRIPCIÓN DEL PROBLEMA

En la actualidad, los restaurantes de la región, no cuentan con un sistema organizado ni sistematizado para realizar el manejo de su información, esto debido tanto a problemas culturales como a los altos costos en materia de hardware, software, acceso a Internet y gastos administrativos.

Los negocios de restaurantes se deben a sus clientes, desafortunadamente como en términos generales cuentan con suficiente clientela, no han sentido la necesidad de organizar sus procesos, en especial los que tienen que ver con el manejo de la información y atención a sus clientes. Con la situación económica que vive el país, la competencia existente en todos los campos, se abre camino la filosofía del buen servicio como elemento de valor agregado para los negocios.

Los pedidos a domicilio se realizan vía telefónica, con las habituales demoras en la atención debido a la congestión telefónica y al mal manejo en la toma de pedidos, no se lleva la información organizada, ni mucho menos estadística sobre pedidos, clientes, tipo de clientes, gusto de los clientes, frecuencia de solicitudes, tiempos de atención entre otros.

El comercio electrónico no ha entrado aún en este mercado. Además no se cuenta con la característica de poder brindar a sus clientes la posibilidad de hacer solicitudes de pedidos vía Web.

Se cuenta con la actitud de un empresario emprendedor que ha planteado la iniciativa de negocio sin tener conocimiento de las virtudes que brindan las TIC's. Inicialmente se estructura un sistema comercial utilizando un inventario manual y la gestión de despachos mediante llamados telefónicos, incurriendo en el riesgo de congestión, hecho que impide el normal crecimiento del negocio, tal como lo indica sus proyecciones de crecimiento a mediano y corto plazo.

Uno de los aspectos primordiales y que se convierte en factor clave del éxito es el hecho de tener perfiles\* de usuario para los clientes, debido a que permite conocer mejor sus preferencias, brindando la posibilidad de crear sistemas promocionales cumpliendo rápidamente sus metas. Este hecho también le permite agilizar el proceso de compra ya que el sistema mantendrá guardada las últimas compras.

La posibilidad de iniciar un negocio bajo dicha modalidad comercial sin los beneficios que brindan las TIC's corre muchos riesgos de no tener éxito.

---

\* Los perfiles están ligados a un análisis previo a los clientes, con el fin de detectar sus gustos preferenciales en cuanto a productos.



Los empresarios deben moverse en la realidad del mercado, facilitando la satisfacción del cliente, teniendo en cuenta, que es menos costoso mantener a un cliente que conseguir uno nuevo. Por tanto, es necesario conocer a tiempo y tener disponible la información del cliente, sus gustos y preferencias; para cumplir con esta fundamentalidad, se requiere contar con sistemas de información que permitan almacenar y seleccionar la información que sean fáciles de usar y que no afecten el ambiente laboral de tal manera que la administración de la información sea más eficaz, precisa, útil, oportuna y con calidad y se refleje en resultados positivos que permitan la permanencia de los empresarios en el mercado conociendo y satisfaciendo las necesidades y los deseos presentes y previsibles de los clientes.

### **1.1 ANÁLISIS DE LA SITUACIÓN ACTUAL**

Según un estudio preliminar de requerimientos, en la actualidad no se cuenta con ningún sistema que tenga la naturaleza del sistema propuesto por parte de los beneficiados\*, la situación actual está basada en el esquema tradicional, el cual no corresponde con lo trazado como objetivos.

### **1.2 FORMULACIÓN DEL PROBLEMA**

En forma general la situación problemática está especificada en la necesidad de implementar un gestor Web que permita el procedimiento general en cuanto a comercialización de productos en un restaurante.

---

\* Los propietarios del restaurante, el cual funciona en forma tradicional

## **2. OBJETIVOS**

### **2.1 OBJETIVO GENERAL**

Analizar, diseñar e implementar una aplicación bajo ambiente Web que permita controlar los procesos de pedidos y ventas para restaurantes. Caso de estudio con el apoyo de la Secretaría de Cultura y Turismo de Cartago.

### **2.2 OBJETIVOS ESPECÍFICOS**

- Analizar el funcionamiento del sistema de información actual de los procesos de toma de pedidos y distribución.
- Diseñar un solución informatizada para dar solución al manejo de la información que incluya:
  - Diseñar una Base de Datos que permita implementar ambientes Web.
  - Diseñar un módulo que permita hacer pedidos mediante página Web.
  - Diseñar un módulo que permita al vendedor obtener estadísticas de pedidos y ventas
  - Diseñar un módulo que permita al comprador obtener información de ofertas y estadísticas de sus compras.
  - Diseñar un módulo que permita que permita administrar perfiles del cliente.
  - Diseñar un módulo para el control de despachos a domicilio.
- Desarrollar un módulo de pedidos de tal forma que cuente con un subsistema que se adapte a una Call Center.
- Diseñar e implementar un módulo que permita hacer reservaciones

### 3. JUSTIFICACIÓN

El proyecto se justifica en la necesidad de realizar un manejo eficiente de la información por parte de los restaurantes, y la de valorar y utilizar el apoyo tan importante de la Casa de la Cultura, ya que este esquema se podría llevar en un futuro a otro tipo de agrupaciones de negocios de la región como Ferretería, Tiendas entre otros. Además de la innovación al desarrollar una aplicación centralizada que beneficie a todos los integrantes de un gremio, optimizando recursos y economizando costos.

No importa que tipo de negocio o compañía se tenga ya que solo en la medida en que se aprenda a manejar y administrar la información, los resultados serán a favor al igual que la productividad obtenida.

En los tiempos actuales, donde se exigen respuestas rápidas y atención inmediata, las empresas de cualquier talla no pueden quedarse estancadas en la telefonía. No deben subestimar las alternativas de mayor rapidez para gestionar, operar y concluir una transacción comercial, por esto las telecomunicaciones constituyen uno de los caminos más rápidos y seguros para aumentar su rentabilidad; no se trata de ahorrar mano de obra, sino de dar un servicio rápido y sin fallos, haciendo uso de los beneficios de una herramienta como Internet por medio del cual se puede tener un sitio Web bien estructurado con forma y funcionalidad para darse a conocer a nuevos clientes, abrirse aún más a los existentes y mejorar los servicios que se ofrecen.

Así pues para el caso en mención objeto de la propuesta, se tiene la necesidad de potenciar mediante las nuevas tecnologías las posibilidades de comercialización de los pequeños comerciantes y la posibilidad que los clientes puedan de una manera ágil realizar sus pedidos vía Web y que además este sistema tenga una recordación o historial de sus pedidos pasados es una de las motivaciones principales del presente proyecto.

## 4. MARCO TEÓRICO

### 4.1 CLASIFICACIONES BÁSICAS DE SISTEMAS GENERALES

**4.1.1 Sistema de gestión de contenido.** Un Sistema de gestión de contenido (CMS\*) permite la creación y administración de contenidos principalmente en páginas Web.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo público.

**4.1.2 Sistemas Distribuidos.** "Sistemas cuyos componentes hardware y software, que están en ordenadores conectados en red, se comunican y coordinan sus acciones mediante el paso de mensajes, para el logro de un objetivo. Se establece la comunicación mediante un protocolo prefijado por un esquema cliente-servidor"<sup>1</sup>

**4.1.3 Sitio Web.** Es un conjunto de archivos electrónicos y páginas Web referentes a un tema en particular, que incluye una página inicial de bienvenida, generalmente denominada HOME PAGE, con un nombre de dominio y dirección en Internet específicos.

**4.1.4 Portal.** Portal es un término, sinónimo de puente, para referirse a un Sitio Web que sirve o pretende servir como un sitio principal de partida para las personas que se conectan al World Wide Web. Son sitios que los usuarios tienden a visitar como sitios ancla. Los portales tienen gran reconocimiento en Internet por el poder de influencia que tienen sobre grandes comunidades.

### 4.2 UML

Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG. "Es un lenguaje

---

\* Content Management System, en inglés, abreviado CMS

<sup>1</sup> HURTADO JARÁN, Omar. Nuevos Paradigmas de los Sistemas de Información. [en línea]. Madrid : Universidad Carlos III > Sistemas distribuidos>Definición . Disponible en : <http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>

gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables"<sup>2</sup>.

El desarrollo de UML comenzó a finales de 1994 cuando Grady Booch y Jim Rumbaugh de Rational Software Corporation empezaron a unificar sus métodos. A finales de 1995, Ivar Jacobson y su compañía Objectory se incorporaron a Rational en su unificación, aportando el método OOSE\*.

De las tres metodologías de partida, las de Booch y Rumbaugh pueden ser descritas como centradas en objetos, ya que sus aproximaciones se enfocan hacia el modelado de los objetos que componen el sistema, su relación y colaboración. Por otro lado, la metodología de Jacobson es más centrada a usuario, ya que todo en su método se deriva de los escenarios de uso. UML se ha ido fomentando y aceptando como estándar desde el OMG\*\*, que es también el origen de CORBA\*\*\* el estándar líder en la industria para la programación de objetos distribuidos. En 1997 UML 1.1 fue aprobada por la OMG convirtiéndose en la notación estándar de facto para el análisis y el diseño orientado a objetos.

UML es el primer método en publicar un meta-modelo en su propia notación, incluyendo la notación para la mayoría de la información de requisitos, análisis y diseño. Se trata pues de un meta-modelo auto-referencial (cualquier lenguaje de modelado de propósito general debería ser capaz de modelarse a sí mismo).

En un relativamente corto período (4 años) de tiempo UML ha emergido como el lenguaje de modelado más dominante en la industria. Existe actualmente una Revision Task Force (RTF) responsable de la generación de revisiones menores de la especificación UML 1.1. La primera RTF de UML terminó su revisión en Junio de 1999 con el draft final UML 1.3. Los miembros de la segunda RTF están trabajando en lo que sería la siguiente revisión (UML 1.4). Además de esta

---

<sup>2</sup> WIKIPEDIA. Lenguaje unificado de modelado. [en línea]. <http://es.wikipedia.org/wiki/Uml> [citado en 1 de junio de 2007]

\* Object-Oriented Software Engineering, en ingles, abreviado OOSE.

\*\* El OMG (Object Management Group) se formó en 1989 con el propósito de crear una arquitectura estándar para objetos distribuidos en redes (componentes). En Octubre de 1989 empezó a funcionar como una corporación independiente y sin ánimo de lucro. El compromiso asumido por el OMG busca el desarrollo de especificaciones para la industria del software que sean técnicamente excelentes.

\*\*\* Common Object Request Broker Architecture (CORBA) es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

revisión menor también se está trabajando en una revisión mayor en lo que sería el UML 2.0.

### 4.3 HERRAMIENTAS DE DESARROLLO

Las herramientas de desarrollo proporcionan un alto grado de personalización a los proyectos al igual que la optimización de las tareas involucradas en el proceso de desarrollo. De una adecuada selección dependerá la rapidez de la creación de las aplicaciones.

**4.3.1 Netbeans 5.5.** NetBeans se refiere a una plataforma para el desarrollo de aplicaciones de escritorio usando Java y a un Entorno integrado de desarrollo (IDE) desarrollado usando la Plataforma NetBeans.

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

El IDE NetBeans es un entorno de desarrollo - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso.

También disponible está la plataforma de NetBeans; una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

Ambos productos son de código abierto y gratuitos para el uso tanto comercial y como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL).

**4.3.2 Dreamweaver 8.** Macromedia Dreamweaver es un editor WYSIWYG de páginas Web, creado por Macromedia\*. Es el programa de este tipo más utilizado en el sector del diseño y la programación Web, por sus funcionalidades, su

---

\* Actualmente adquirida por Adobe Systems

integración con otras herramientas como Macromedia Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Su principal competidor es Microsoft Frontpage. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas

Existe un programa libre competencia de Dreamweaver que es el D4L.

Hasta la versión MX, fue duramente criticado por su escaso soporte de los estándares de la Web, ya que el código que generaba era con frecuencia sólo válido para Internet Explorer, y no validaba como HTML estándar. Esto se ha ido corrigiendo en las versiones recientes.

Dreamweaver 8 es la herramienta de desarrollo Web líder del mercado y permite a sus usuarios diseñar, desarrollar y mantener de forma eficaz sitios y aplicaciones Web basados en normas. Con Dreamweaver 8, los desarrolladores Web lo abarcan todo, desde la creación y mantenimiento de sitios Web básicos hasta aplicaciones avanzadas compatibles con las mejores prácticas y las tecnologías más recientes.

¿Por qué Dreamweaver?. Porque es: - Utilizar un editor de diseño y código de primera calidad en una sola herramienta, - Aprender a usar una herramienta dejando que sea ella misma quien enseñe; - Dreamweaver ayuda y guía a los usuarios conforme éstos van aumentando sus conocimientos y a medida que las tecnologías Web van evolucionando, facilitando una adopción fácil y rápida de las nuevas tecnologías y metodologías.

- **Agregar vídeo fácilmente.** Con Dreamweaver 8 y Flash Video se podrá agregar rápidamente contenidos de vídeo en la Web, con sólo arrastrar y colocar vídeos de Flash en Dreamweaver 8 para agregar de forma rápida vídeos a sitios y aplicaciones Web.
- **Simplificar el trabajo con los estilos CSS.** El panel unificado de CSS ofrece una representación sencilla y directa de la cascada de estilos aplicados al contenido y ofrece acceso rápido para realizar cambios sin necesidad de realizar búsquedas en el código probando por ensayo y error.
- **Empezar a utilizar XML.** Simplemente con señalar una página Web en un archivo XML o una URL de una entrada XML y Dreamweaver lo introspeccionará para luego poder arrastrar y colocar los campos apropiados en la página.
- **Sumarse a la amplia comunidad Dreamweaver.** Con más de tres millones de usuarios, la comunidad Dreamweaver ofrece numerosas ventajas, entre las que cabe mencionar el Centro para Desarrolladores de Macromedia, los programas de certificación para desarrolladores, los cursos de formación y seminarios, los foros de usuarios y los sitios

independientes que ofrecen soporte a la comunidad. Hay disponibles más de 800 extensiones que pueden descargarse de forma gratuita a través de Macromedia Dreamweaver Exchange.

## 4.4 LENGUAJES DE PROGRAMACIÓN

Los lenguajes de programación permiten especificar sobre que datos se debe operar y la forma de transmisión y almacenamiento de los mismos para cumplir con los requerimientos definidos para un proyecto dado.

**4.4.1 Java.** “Java es un lenguaje de programación orientado a objetos desarrollado por James Gosling y sus compañeros de Sun Microsystems al inicio de la década de 1990. A diferencia de los lenguajes de programación convencionales, que generalmente están diseñados para ser compilados a código nativo, Java es compilado en un bytecode que es ejecutado (usando normalmente un compilador JIT), por una máquina virtual Java”.<sup>3</sup>

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos mucho más simple y elimina herramientas de bajo nivel como punteros.

Java está sólo lejanamente emparentado con JavaScript, aunque tengan nombres similares y compartan una sintaxis al estilo de C algo parecida.

**4.4.2 HTML.** World Wide Web (WWW) simplemente, que es un sistema de información, el sistema de información propio de Internet. Sus características son:

- Información por hipertexto: Diversos elementos (texto o imágenes) de la información que se nos muestra en la pantalla están vinculados con otras informaciones que pueden ser de otras fuentes. Para mostrar en pantalla esta otra información bastará con hacer clic sobre ellos.
- Gráfico: En la pantalla aparece simultáneamente texto, imágenes e incluso sonidos.
- Global: Se puede acceder a él desde cualquier tipo de plataforma, usando cualquier navegador y desde cualquier parte del mundo.
- Pública: Toda su información está distribuida en miles de ordenadores que ofrecen su espacio para almacenarla. Toda esta información es pública y toda puede ser obtenida por el usuario.

---

<sup>3</sup> WIKIPEDIA. Java. [en línea]. <http://es.wikipedia.org/wiki/Java//es.wikipedia.org> [citado en 3 de junio de 2007]



- Dinámica: La información, aunque esta almacenada, puede ser actualizada por el que la publico sin que el usuario deba actualizar su soporte técnico.
- Independiente: Dada la inmensa cantidad de fuentes, es independiente y libre.

HTML, estas siglas se corresponden con la definición "Lenguaje para marcado de hipertexto". Más claro aún, se trata de un lenguaje para estructurar documentos a partir de texto en World Wide Web. Este lenguaje se basa en tags (instrucciones que le dicen al texto como deben mostrarse) y atributos (parámetros que dan valor al tag). HTML es una aplicación de SGML conforme al estándar internacional ISO 8879. XHTML es una reformulación de HTML 4 como aplicación XML 1.0, y que supone la base para la evolución estable de este lenguaje. Además XHTML permite la compatibilidad con los agentes de usuario que ya admitían HTML 4 siguiendo un conjunto de reglas.

**4.4.3 JSP.** Es una tecnología Java que permite a los programadores generar contenido dinámico para Web, en forma de documentos HTML, XML, o de otro tipo. Las JSP's permite al código Java y a algunas acciones predefinidas ser incrustadas en el contenido estático del documento Web.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones Web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento html en el archivo jsp. La idea fundamental detrás de este criterio es el de separar la lógica del negocio de la presentación de la información.

Sin embargo, JSP no se puede considerar un script al 100% ya que antes de ejecutarse el servidor Web compila el script y genera un servlet, por lo tanto, se puede decir que aunque este proceso sea transparente para el programador no deja de ser una aplicación compilada. La ventaja de esto es algo más de rapidez y disponer del API de Java en su totalidad.

Debido a esto la tecnología JSP, así como Java está teniendo mucho peso en el desarrollo Web profesional (sobre todo en intranets).

**4.4.4 JavaScript.** JavaScript es un lenguaje interpretado, al igual que VisualBasic, Perl, TCL... sin embargo, posee una característica que lo hace especialmente idóneo para trabajar en Web, ya que son los navegadores que utilizamos para viajar por ella los que interpretan (y por tanto ejecutan) los programas escritos en JavaScript. De esta forma, podemos enviar documentos a través de la Web que incorporan el código fuente de un programa, convirtiéndose de esta forma en documentos dinámicos, y dejando de ser simples fuentes de información estáticas.

Los programas en JavaScript no son la primera forma que conoce la Web para transformar información, dado que el uso de CGIs está ampliamente

difundido. La diferencia básica que existe entre un programa CGI y uno escrito en JavaScript es que el CGI se ejecuta en el servidor de páginas Web mientras que el programa en Javascript se ejecuta en el cliente (es decir, en el navegador). Por regla general, el CGI necesita unos datos de entrada (que normalmente se proporcionan mediante un formulario), los procesa y emite un resultado en forma de documento HTML. Esto implica tres transacciones en la red. La primera carga la página del formulario, la segunda envía los datos al servidor, y la tercera recibe la nueva página que ha generado el CGI. Por el contrario, los programas escritos en JavaScript se ejecutan en el navegador del cliente, sin necesidad de que intervenga el servidor. De esta forma, una sola transacción basta para cargar la página en la que se encuentra tanto el formulario, para los datos de entrada, como el programa en JavaScript que proporciona los resultados. Sin embargo, esto no significa (como veremos después) que los CGI vayan a ser substituidos por JavaScript<sup>4</sup>.

## 4.5 SISTEMAS DE BASES DE DATOS

**4.5.1 POSTGRES.** PostgreSQL es el último resultado de una larga evolución comenzada con el proyecto Ingres en la Universidad de Berkeley. El líder del proyecto, Michael Stonebraker abandonó Berkeley para comercializar Ingres en 1982, pero finalmente regresó a la academia.

Mediante un sistema denominado MVCC\* PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

---

<sup>4</sup> RIVAS SANTOS, Victor. Curso Java Script. [en línea]. Madrid: Geneura > Tutoriales on - line > Javascript. Disponible en: [http://geneura.ugr.es/~victor/cursillos/javascript/js\\_intro.html](http://geneura.ugr.es/~victor/cursillos/javascript/js_intro.html)

\* Acceso concurrente multiversión

En 1986 otro equipo dirigido por Michael Stonebraker de Berkeley continuó el desarrollo del código de Ingres para crear un sistema de bases de datos objeto-relacionales llamado Postgres. En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, Postgres fue renombrado a PostgreSQL, tras un breve periplo como Postgres95. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto<sup>5</sup>.

- **Plataformas que soporta el Postgres.** En general, cualquier plataforma moderna compatible con Unix debería ser capaz de ejecutar PostgreSQL. Las plataformas que han sido probadas al momento de la distribución son listadas en las instrucciones de instalación. PostgreSQL también corre nativamente sobre sistemas operativos basados en Microsoft Windows NT tales como Win2000, WinXP y Win2003. Existe incluso un port para Novell Netware 6 y una versión para OS/2 (eComStation).
- **Cómo se compara PostgreSQL con otros DBMS.** Existen varias maneras de medir el software: características, desempeño, fiabilidad, soporte y precio.
  - Características. PostgreSQL tiene la mayoría de las características presentes en grandes DBMS comerciales, tales como transacciones, subconsultas, gatillos (triggers), vistas, integridad referencial con llaves externas, y bloqueo sofisticado. “También tiene algunas características que no tienen las otras, como tipos definidos por el usuario, herencia, reglas, y control de concurrencia multi-versión para reducir el bloqueo de controversias”<sup>6</sup>.
  - Desempeño. El desempeño de PostgreSQL es comparable con el de otras bases de datos comerciales y de código abierto. Es más rápida para algunas cosas, más lenta para otras. El desempeño es usualmente +/-10% comparado con otras bases de datos.
  - Fiabilidad. Se cree que un DBMS debe ser fiable, o es inútil. Se procura liberar código estable y bien probado que tenga mínimos errores. Cada versión tiene al menos un mes de pruebas beta, y la historia de liberaciones muestra que se puede proveer versiones estables y sólidas que se encuentran listas para su uso en producción. Sé esta seguro que en esta área se compara favorablemente con otras bases de datos.

---

<sup>5</sup> POSTGRESQL. Qué es postgresql. [en línea]. <http://www.postgresql.org.mx/?q=node/6> [citado en 10 de febrero de 2007]

<sup>6</sup>TLDP. Tutorial de Postgresql. [en línea]. <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.htm> [citado en 10 de febrero de 2007]

- Soporte. Las listas de correo, proporcionan contacto con un gran grupo de desarrolladores y usuarios que le ofrecen ayuda para resolver cualquier problema encontrado. No se puede garantizar una corrección, los DBMSs comerciales no siempre suministran correcciones. El acceso directo a desarrolladores, la comunidad de usuarios, manuales y el código fuente suelen hacer que el soporte de PostgreSQL sea superior al de otras DBMSs. También hay soporte comercial disponible por incidente para aquellos que lo necesiten.
- Precio. Está disponible para cualquier uso, ya sea comercial o no, puede agregar el código a su producto sin limitación alguna.

## 4.6 SISTEMAS PARA MODELAR SOFTWARE

Rational es actualmente conocida como una familia de software de IBM para el levantamiento de requerimientos, diseño, construcción, pruebas y administración de proyectos en el proceso desarrollo de software.

Las herramientas de Rational dan grandes beneficios sin limitar la libertad creativa de los equipos de desarrollo. Permite seleccionar el nivel de abstracción correcto para cada tarea. Utiliza el ambiente de desarrollo y el IDE\* preferido y adecuado para el equipo. Adapta el ambiente del proyecto a las necesidades específicas. No importa cuál sea la plataforma de desarrollo o las funciones dentro del equipo (rol), hay alguna solución de Rational que puede ayudar.

Las herramientas de Rational fortalecen el desarrollo con Java con la principal plataforma de desarrollo para la comunidad Java. Nuestras herramientas galardonadas, te permitirán trabajar intuitivamente, con mayor control, y con el conjunto más amplio de recursos Java. A partir de la adquisición de Rational por IBM, la integración de las herramientas ha sido sin precedente, ofreciendo una gran facilidad de uso. Otras alianzas estratégicas con Sun Microsystems, HP, y otros asegura una compatibilidad total (end-to-end) en tu ambiente de desarrollo Java seleccionado.

**4.6.1 Racional Software Development Platform.** IDE que es fácil de aprender y utilizar. Optimizado para el software de IBM WebSphere y capaz de soportar entornos de ejecución de múltiples proveedores, IBM Rational Web Developer for WebSphere Software está basado en la plataforma de código abierto de Eclipse para que pueda adaptar y ampliar su entorno de desarrollo para satisfacer sus necesidades y aumentar la productividad. Si se utiliza con IBM Software Development Platform, se podrá acceder a una amplia gama de requisitos y cambiar funciones de gestión directamente desde Web Developer.

---

\* Ambiente Integrado de Desarrollo

## 4.7 ANTECEDENTES

Luego de realizar una revisión sobre los sistemas de información que manejan en los restaurantes de Cartago-Valle, se encontró:

- Todos los negocios realizan un manejo independiente y centralizado de la información con los sabidos costos de infraestructura y administración.
- No existe un esquema en el manejo de la información que permita consolidados y estadística.
- Ningún gremio público o privado que agrupe por tipos de negocio, ofrece el servicio de manejo de la información vía Web de manera cliente servidor.

Ya en materia de software, las empresas que ofrecen opción de compras vía Web, en la actualidad se pueden clasificar dentro de la clasificación:

- **Software Genérico.** Las casas de software que ofrecen aplicaciones rígidas con módulos como Facturación, Inventarios, Compras, Pedidos, pero no ofrecen las posibilidades Web. En la ciudad los más difundidos son el Apolo, COMODIN, SIIGO entre otros.

### 4.7.1 Los Sistemas de Información

- **Actividades Básicas de un Sistema de Información.** Un sistema agregado de cosas organizado y autocontenido. El concepto tiene dos usos muy diferenciados, que se refieren respectivamente a los sistemas de conceptos y a los objetos reales más o menos complejos y dotados de organización. La palabra sistema ha adquirido muchos usos especializados, algunos muy anteriores a la Teoría de sistemas, pero siempre referentes a conjuntos estructurados y organizados, casi siempre en el campo de los sistemas ideales, conceptuales o formales.

“La información la componen datos que se han colocado en un contexto significativo y útil y se ha comunicado a un receptor, quien la utiliza para tomar decisiones”<sup>7</sup>. Un sistema de información se puede definir como un conjunto de funciones o componentes interrelacionados que forman un todo, es decir, obtiene, procesa, almacena y distribuye información (datos manipulados) para apoyar la toma de decisiones y el control en una organización. Igualmente apoya la coordinación, análisis de problemas, visualización de aspectos complejos, entre otros aspectos.

---

<sup>7</sup> GRUDNITSKI, Burch. Introducción al desarrollo de los sistemas de información. [en línea]. Venezuela : Introducción al desarrollo de los sistemas de información > Diseño de sistemas de información. Disponible en: <http://www.geocities.com/siliconvalley/pines/7894/sistemas/introduccion.html>

“Sistema de Información: es el conjunto de procesos que, operando sobre una colección de datos estructurada de acuerdo a una empresa, recopila, elabora y distribuye (parte de) la información necesaria para la información de dicha empresa y para las actividades de dirección y control correspondientes, apoyando al menos en parte, la toma de decisiones necesarias para desempeñar las funciones y procesos de negocios de la empresa de acuerdo a su estrategia”<sup>8</sup>.

#### 4.7.2 Teorías Refenciales Explicativas

- **Antecedentes y Conceptualización del tema.** Históricamente las empresas han buscado una solución a la sistematización de sus procesos, normalmente instalan software que les permita lograr seguridad, agilidad y control sobre sus procesos; en el caso de las intermediarías en seguros se tienen una situación particular: sus procesos son muy diversos y complejos, basados en la gama de productos que pueden ofrecer, hecho que las convierte en un foco problemático en el mundo de la informática.

Gran parte de las empresas dedicadas a intermediar en los negocios de lo seguros usan software no especializado, adaptando, en muchas ocasiones, sus procesos para poder sistematizar algunos de sus procesos; o utilizado algunas aplicaciones que les suministran las aseguradoras par que se ayuden en el momento de cotizar.

- **Alternativas de solución propuesta.** Basados en la naturaleza de los procesos a sistematizar, la compañía tiene las siguientes opciones para buscar solución a toda la problemática:
  - **Contratación:** mediante la modalidad de OUTSOURCING, la compañía puede contratar los servicios de una persona natural o empresas para que se encargue de dar solución a la problemática existente.
  - **Compra de software:** consiste en la compra de un producto ya hecho, en el mercado es difícil de encontrar un software que cumpla con las características actuales respecto a los procedimientos llevados a cabo en el momento de generar una cotización de seguros vehiculares.

---

<sup>8</sup> WIKIPEDIA. Sistema de información. [en línea].  
[http://es.wikipedia.org/wiki/Sistema\\_de\\_informaci%C3%B3n](http://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n) [citado en 20 de febrero de 2007]

## 5. METODOLOGÍA

### 5.1 TIPO DE TRABAJO

Desarrollo tecnológico que permite la aplicación de TICs a PYMES (restaurantes) mediante la metodología de Proceso Unificado (UP). Permite que la tecnología esté al servicio del negocio y del recurso humano, proporcionando dinamismo, método y mejorando la eficiencia.

Se trata de un trabajo de tipo tecnológico ya que después de establecido este tipo de conocimiento, es aplicado para dar solución a un problema específico; la necesidad de implementar un sistema que permita administrar, agilizar, controlar y mejorar el procedimiento general en cuanto a comercialización de productos en un restaurante.

### 5.2 PROCEDIMIENTO

#### 5.2.1 Fase I: Modelo de requisitos

- **Etapa 1:** Captura de los requisitos de usuario
  - **Actividad a:** Modelo de requisitos
- **Etapa 2:** Establecer requisitos no funcionales
  - **Actividad a:** Modelo del negocio
  - **Actividad b:** Análisis de los objetivos de la organización
- **Etapa 3:** Establecer funciones de la aplicación
  - **Actividad a:** Modelo de casos de uso simplificado

#### 5.2.2 Fase II: Modelo de análisis

- **Etapa 1:** Profundizar en cada elemento definido en el modelo de requisitos
  - **Actividad a:** Estudio requisitos funcionales y no funcionales
  - **Actividad b:** Determinar casos de uso de análisis
- **Etapa 2:** Entender el uso del sistema
  - **Actividad a:** Diagramas de caso de uso detallados

#### 5.2.3 Fase III: Modelo de diseño

- **Etapa 1:** Construcción modelo de diseño
  - **Actividad a:** Diseñar casos de uso
  - **Actividad b:** Diseñar clases
  - **Actividad c:** Diseñar subsistemas

- **Actividad d:** Diagramas de despliegue
- **Actividad e:** Diagramas de componentes
- **Actividad f:** Diagramas de colaboración
- **Actividad g:** Diagramas estéticos



## 6. RESULTADOS

El sistema de información desarrollado para la gestión comercial de productos alimenticios vía Internet resulto una herramienta muy útil que proyecta la unidad de negocio.

La fase inicial de requerimientos ayudo a la comprensión de las necesidades del cliente, hecho que fue proyectado en sus inicios, obteniendo el modelo de requisitos; éste se convirtió en factor clave de éxito para la construcción del modelo de análisis, debido a que su contexto está definido en el lenguaje del cliente.

Las características arquitectónicas obtenidas en el modelo de diseño resultaron, y después varios procesos de depuración, en lo que el cliente estaba necesitando, debido a las planimetrías logradas en su esquema jerárquico, tal y como se indica en el marco teórico.

Se logró desarrollar un software en ambiente Web, que permite el ingreso de personas con bajos conocimientos, permitiendo la elaboración de pedidos en forma ágil y fácil para los clientes, teniendo un control de registro, todo con el fin de auditar los pedidos y sus compradores.

Se convirtió en una herramienta que ayuda a los administradores del negocio a controlar la calidad del servicio, puesto que los diferentes reportes evidencian los estados de los pedidos, permitiendo el control en los tiempos de espera de los clientes.

El sistema de información desarrollado se adapta a un Call Center tan sólo con ingresar bajo un usuario configurado para tal fin, con este módulo una persona tiene dedicación completa a contestar el teléfono y tomar pedidos en forma tradicional.

En términos generales, la aplicación provee una serie de ayudas a los administradores para conocer mejor su negocio y a su vez, contribuye con el crecimiento económico, debido a que se abre un nicho del mercado, el cual está enmarcado en Internet.

## **7. CONCLUSIONES**

- En el momento de analizar el funcionamiento del sistema de información actual, en cuanto a procesos de toma de pedidos y distribución se logró construir una herramienta ágil de fácil de manejar, debido a la construcción efectiva del modelo de datos, y el uso de POSTGRES como plataforma procedimental para la gestión de la información en línea.
- El diseño de una solución informatizada para dar solución al manejo de la información en los procesos resultó ser eficaz y apropiada para la necesidad planteada al inicio del proyecto, logrando un módulo para los procesos y procedimientos en el momento de hacer pedidos.
- El control para despachos se fortaleció en gran medida, debido a la inclusión de nuevos estados en los pedidos, permitiendo así al seguimiento de los mismos.

## 8. RECOMENDACIONES

- En el desarrollo de un proyecto de información para sistemas distribuidos, en el caso particular la publicación Web, es necesario estar actualizados en todos los aspectos en seguridad informática y las plataformas de desarrollo.
- Es necesario tener en cuenta la utilización de navegadores compatibles con la tecnología mas común, es decir, en el momento de desarrollar un proyecto de ésta índole, es apropiado usar plataformas que se adapten a los navegadores más usados por los usuarios (clientes) a los que el sistema esta orientado.
- En el momento de la instalación del sistema se debe tener en cuenta los requisitos y los pasos a seguir los cuales se encuentran en el manual técnico, para evitar posibles problemas o complicaciones tanto en el proceso de instalación como en el manejo del software.
- En términos generales es apropiado contar con los recursos apropiados para en desarrollo de un proyecto de éstas características, debido a la apropiada aplicación de los mismos en todas las etapas del mismo.
- Es necesario fortalecer la cultura de Internet en la región, la comunidad habilitada para realizar compras en línea aun es muy reducida, los accesos a la aplicación que se esperan y están proyectados se encuentran estrechamente relacionado con esta problemática.
- Teniendo presente que uno de los objetivos principales de este desarrollo busca promover el turismo en el municipio a todos los niveles, se hace necesario tener en cuenta para un futuro, desarrollar versiones en otros idiomas.

## BIBLIOGRAFÍA

FIGUEROA, Pablo. Metodología de Software Orientada por Objetos. [en línea]. Canadá : Versión 1.1. Metodología de Software Orientada por Objetos > Una metodología de desarrollo de software O.O basada en UML. Disponible en: <http://www.cs.ualberta.ca/~pfiguero/soo/metod/>. 2004

GILL, Harjinder y RAO, Prakash. Data Warehousing. (2000). La Integración de la Información para la Toma de Decisiones. México : Prentice Hall, 2000. 382 p.

GRACIA, Joaquín. Diagramas UML. [en línea]. España, Fecha actualización 2004. ingenierosoftware.com > análisis y diseño > casos de uso. Disponible en: <http://www.ingenierosoftware.com/analisisydiseno/casosdeuso.php>

GRUDNITSKI, Burch. Introducción al desarrollo de los sistemas de información. [en línea]. Venezuela : Introducción al desarrollo de los sistemas de información > Diseño de sistemas de información. Disponible en : <http://www.geocities.com/siliconvalley/pines/7894/sistemas/introduccion.html>

HURTADO JARÁN, Omar. Nuevos Paradigmas de los Sistemas de Información. [en línea]. Madrid : Universidad Carlos III > Sistemas distribuídos > Definición . Disponible en : [http://www.monografias.com/trabajos16/sistemas - distribuidos/sistemas-distribuidos.shtml](http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml)

JACOBSON, Ivar; BOOCH, Grady y RAMBAUGH, James. El Proceso Unificado de Desarrollo de Software. Madrid : Addison Wesley, 2000. 438 p.

LARMAN, Graig. UML y Patrones: Introducción al Análisis y Diseño Orientado a Objetos. México : Prentice Hall, 2000. 507 p.

McCONNEL, Steve. Desarrollo y Gestión de Proyectos Informáticos. Madrid : McGraw-Hil,. 1999. 691 p.

MEYER, Bertrand. Construcción de Software Orientado a Objetos, 2 ed. Madrid: Prentice Hall, 1999. 1198 p.

PIATTINI, Mario G. Análisis y Diseño de Aplicaciones Informáticas de Gestión – Una Perspectiva de Ingeniería del Software. México : Alfaomega RA-MA, 2004. 750 p.

PRESSMAN, Roger S. Ingeniería del Software – Un Enfoque Práctico. 6 ed. México : Mc Graw Hill, 2006. 958 p.

RIVAS SANTOS, Victor. Curso Java Script. [en línea]. Madrid : Geneura > Tutoriales on - line > Javascript. Disponible en : [http://geneura.ugr.es/~victor/cursillos/javascript/js\\_intro.html](http://geneura.ugr.es/~victor/cursillos/javascript/js_intro.html)

TAMAYO A., Alonso. Sistemas de información. Manizales : Centro de publicaciones Universidad Nacional de Colombia sede Manizales, 1998. 104 p.

WHITTEN L, Jeffrey y BENTLEY, Lonnie y Ba. Análisis y Diseño de Sistemas Informáticos. México: McGraw-Hill – Hill, 2000. 907 p.

## **ANEXOS**

## ANEXO A

### MODELO DE REQUISITOS

#### 1. ARTEFACTO MODELO DE REQUISITOS

##### 1.1 Casos de Uso (nivel requisitos)

###### Casos de Uso: Necesidades del usuario

###### Descripción general

Uno de los actores principales del sistema es el cliente, dicho componente es el que más utiliza el sistema, en este modelo se indican los requerimientos de uso del sistema.

El usuario cuando ingresa al sistema necesita tener a mano los productos (platos) que el restaurante ofrece, para poder tomar la decisión respecto a su preferencia, también el sistema debe permitir al cliente tome el pedido, para procesarlo y enviárselo en el menor tiempo posible. Dichos elementos procedimentales pueden ser vislumbrados en la Figura 1 como casos de uso.

A continuación se hace referencia a los pasos que cubren el proceso.

1. El Cliente ingresa al sitio Web
2. El Cliente debe consultar las diferentes opciones en cuanto a platos
3. El Cliente debe registrar pedidos, que más tarde, le llegara a domicilio

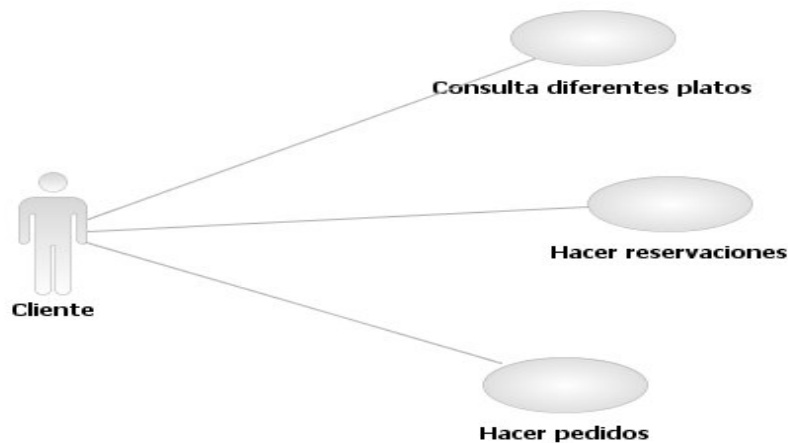


Figura 1 Caso de Uso Requisitos: Descripción general

## Casos de Uso: Necesidades del Administrador del Restaurante

### Descripción general

El sistema debe permitir al administrador del sistema ingresar y controlar toda la información que procesa el restaurante, este administrador debe tener una clave especial, para permitir aumentar la seguridad en sus procesos.

Para el administrador es de vital importancia que pueda ingresar, modificar, eliminar y consultar cualquier tipo de información (como lo indica la Figura 5), incluido los datos del cliente, como preferencias, clave y otros datos de interés.

A continuación se hace referencia a los pasos que cubren el proceso.

1. El administrador ingresa al sistema
2. El administrador se ve posibilitado para ingresar, consultar. Modificar y eliminar la información de la empresa
3. La información que el administrador manipulará hace referencia a usuarios, productos, pedidos y reservaciones.

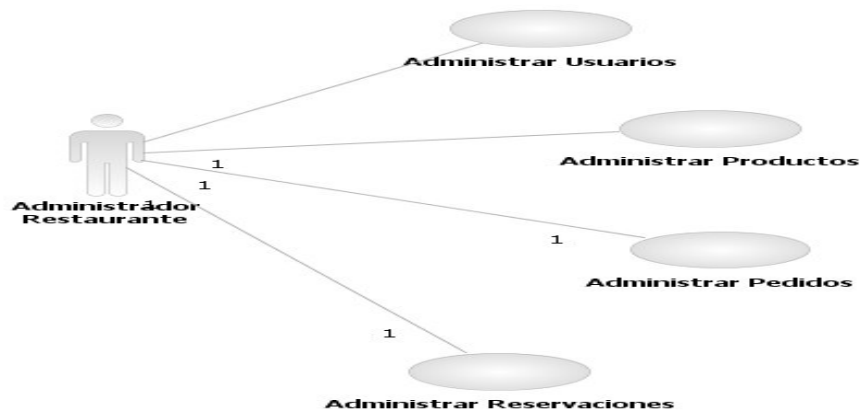


Figura 2 Diagrama de Caso de Uso Requisitos: Administrador del Restaurante

## Casos de Uso: Necesidades del Administrador del Sistema

### Descripción general

El actor Administrador del Sistema está encargado de hacer funciones mas administrativas en el orden técnico, es un futuro es posible que el administrador del restaurante y administrador del sistemas sean las mismas personas, pero, en la toma de requisitos iniciales se vislumbran como entes separados.

Al administrador del sistema se le está permitido la creación de restaurantes, hecho que toma un poco más de tiempo, debido a las consideraciones técnicas que esto representa, en el orden de creación de logos, estilos, y demás



características técnicas que deben ser operadas por un usuario un poco más experto. Lo anterior puede verse evidenciado mediante la **Figura 6**.

A continuación se hace referencia a los pasos que cubren el proceso.

1. El Administrador ingresa al sistema
2. El Administrador configura, conforma y estructura Restaurantes nuevos
3. El Administrador puede modificar información referente a los usuarios

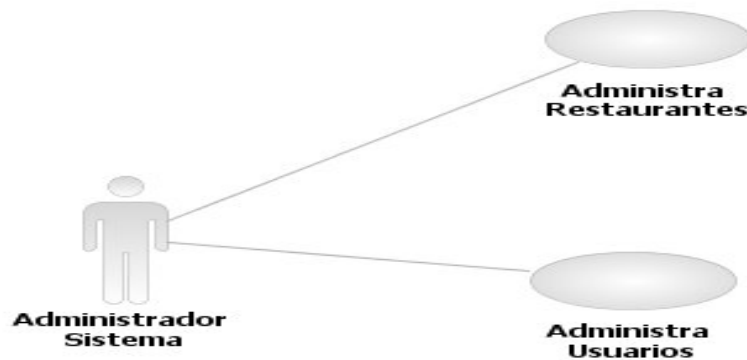


Figura 3 Caso de Uso Requisitos: Administrador del Restaurante

## **Casos de Uso: Necesidades del Empleado CALL CENTER**

### **Descripción general**

El actor Empleado CALL CENTER corresponde a la necesidad de crear un departamento de ventas para los usuarios que no tiene acceso o nivel cultural hacia Internet. Esto consiste en crear una central de llamadas, donde los usuarios llaman a hacer sus pedidos, y un empleado le ofrece las posibilidades del día, el cliente toma la decisión y el empleado de la central de llamadas le toma el pedido por el módulo que se requiere para tales efectos (Ver Figura 7).

A continuación se hace referencia a los pasos que cubren el proceso.

1. El Empleado CALL CENTER ingresa al sistema
2. El Empleado CALL CENTER recibe una llamada de un cliente
3. El Empleado CALL CENTER le informa de los platos disponibles
4. El cliente le indica su preferencia
5. El Empleado CALL CENTER toma el pedido

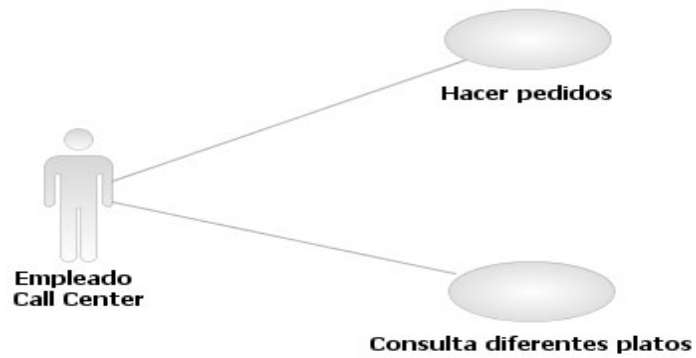


Figura 4 Caso de Uso Requisitos: Empleado CALL CENTER

## 1.2 Diagramas de Actividad (nivel requisitos).

### Diagrama de Actividad: Toma de Pedidos

#### Descripción general

Como complemento al caso de uso referenciado anteriormente, se cita el proceso de la toma de pedidos mediante un diagrama de actividad.

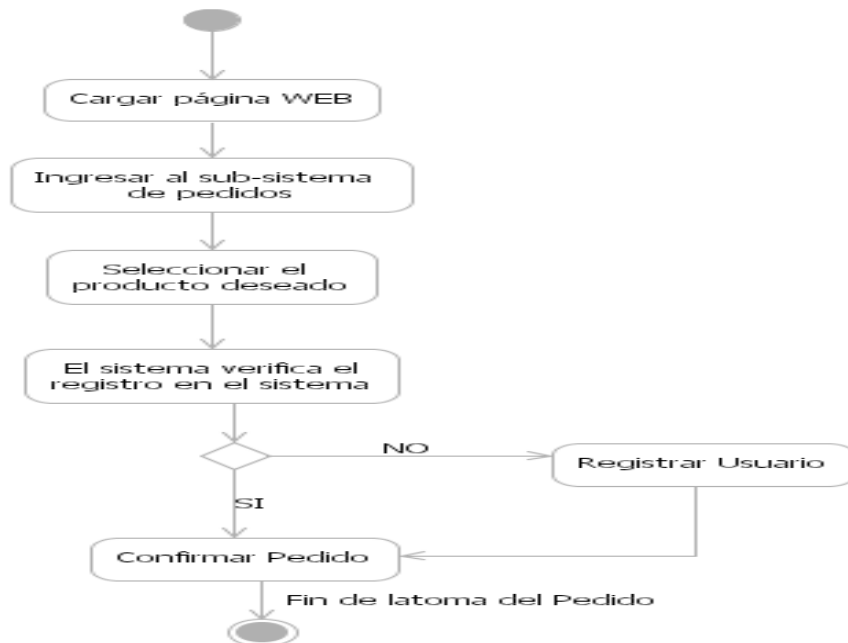


Figura 5 Diagrama de Actividad: Toma de Pedidos

## ANEXO B

### MODELO DE ANÁLISIS

#### 1. ARTEFACTO MODELO DE ANÁLISIS

##### 1.1 Descripción de la situación Actual

El sistema procedimental que opera actualmente corresponde al sistema tradicional de gestión en pedidos, el cual el cliente hace una llamada telefónica al restaurante seleccionado en la publicidad y el cliente escucha todas las posibilidades que ofrece el restaurante; éste toma una decisión y el administrador toma su pedido, que será procesado administrativamente para ser enviado.

El proceso actual de cotización se representa mediante el siguiente diagrama de Actividad

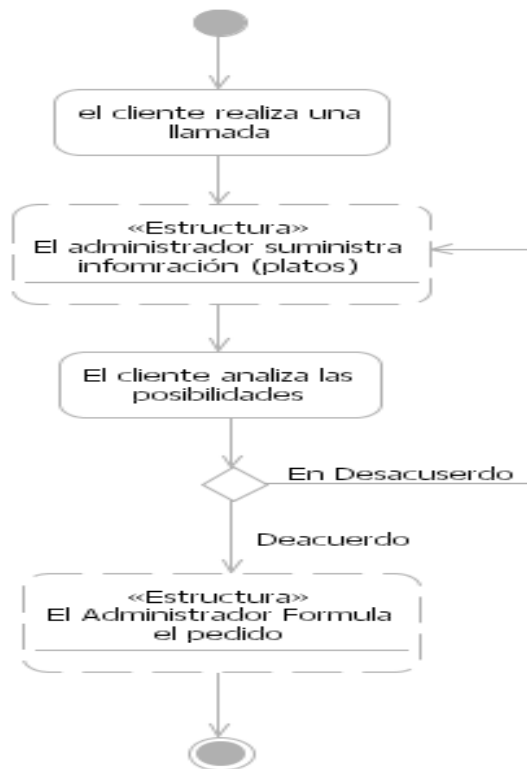


Figura 1 Diagrama de Actividad: Sistema Actual

## 1.2. Descripción del sistema

El sistema propuesto para la sistematización de los procesos en el restaurante en línea, tal y como aparece en la siguiente figura:



Figura 2 Diagrama de Paquetes: General

Como se observa en la figura 2, el sistema general está modelado bajo los lineamientos de dos paquetes generales; uno corresponde al Sub-sistema de gestión en clientes, el cual alberga todo el modelo de análisis para el proceso comercial con los clientes, el otro paquete corresponde al Sub-sistema de gestión administrativa, elemento que contiene la gestión de la administración de la información.

### 1.2.1 Sub-sistema Gestión Clientes

#### Diagrama de casos de Uso: gestión del pedido

Para la generación del pedido vía Internet, el cliente ingresa al sub-sistema para gestión clientes, dichos elementos procedimentales están dados por medio de la gestión del pedido, la consulta de productos y el propio registro del sistema.

La toma de pedido deberá estar validada en el orden de registro en el sistema, es decir, todo usuario debe estar debidamente registrado para poder realizar dicha transacción.

Los productos (platos) serán publicados por el actor Administrador del restaurante.

Los casos de uso están basados en la información que administran los actores: administrador del sistema y administrador del restaurante, dichos actores están modelados al interior del sub-sistema: **Gestión Administrativa**.

A continuación se hace referencia a los pasos que cubren el proceso.

1. El cliente ingresa al sistema
2. El sistema ofrece la posibilidad de consultar productos
3. El sistema ofrece la posibilidad de registrarse al sistema
4. El cliente puede solicitar un pedido
5. El sistema valida los pedidos para usuarios registrados

6. El sistema registra la información del cliente y el pedido para ser procesada administrativamente.



Figura 3 Caso de Uso: Gestión del Pedido

**Diagrama de actividad: gestión del pedido**

Como de indicó en párrafos anteriores, el proceso del pedido es diferente al que se lleva a cabo por la empresa actualmente, ya que se elabora mediante un proceso en línea, tomando pocos segundos para que el cliente y el restaurante den cuentas de cómo será la negociación del pedido, en la figura 3 se indica dicho procedimiento.

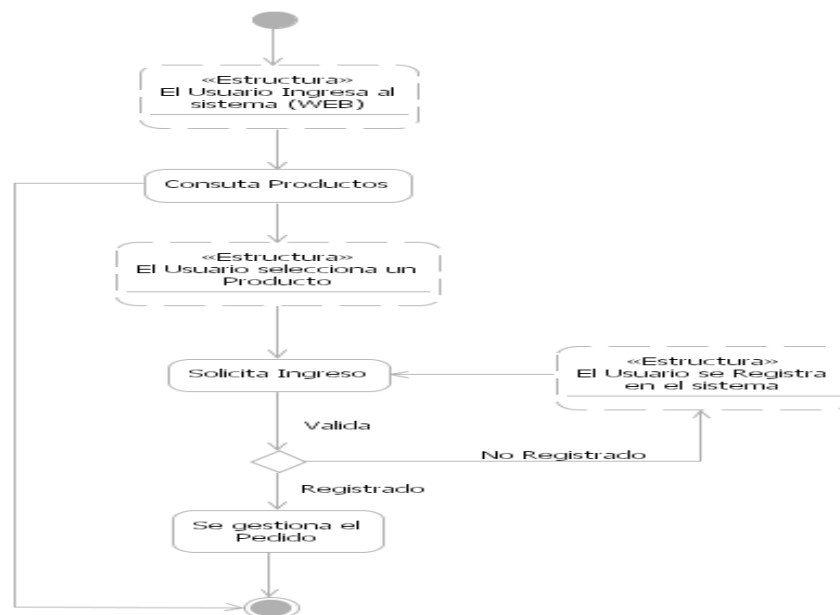


Figura 4 Diagrama de actividad: gestión del pedido

## Diagrama de Estados: Estados

Como de indicó en párrafos anteriores, el proceso del pedido está atado a una serie de estados sobre los cuales se les puede hacer seguimiento, comprende desde el estado de preparación hasta la entrega final y a satisfacción; tal y como se observa en el diagrama de estados para la clase Estados.

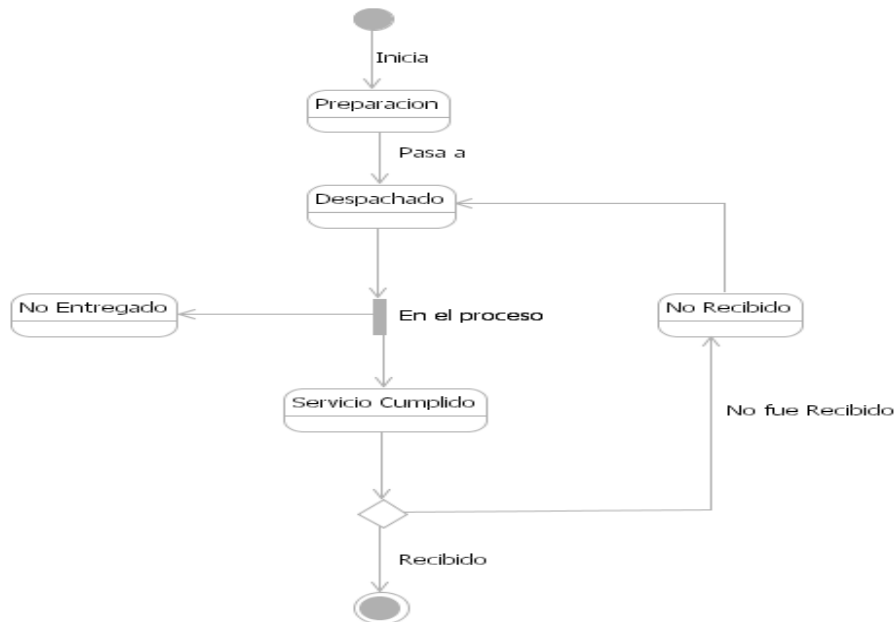


Figura 5 Diagrama de Estados: Estados

## Diagrama de clases

Los diagramas de clases en el modelo de software presentado como herramienta para comprensión esta regido desde dos puntos de vista. Los diagramas de clase para análisis de procesos y los diagramas de clase para indicar las clases núcleo del sistema, que más adelante se convertirá en el la base de datos.

Todo en el modelo esta contextualizado mediante dos paquetes, los cuales se indican en la figura 5, abajo relacionada.

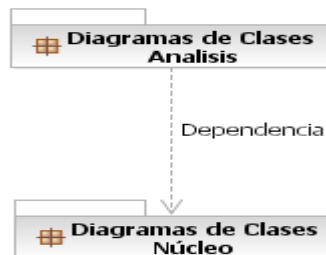


Figura 6 Diagrama de Paquetes: Clases

## Diagrama de clases (Análisis): gestión del pedido

En el diagrama de clases (Análisis) para la gestión del pedido se define una clase interfaz para controlar el ingreso de los clientes al sistema, se tienen tres clases control para administrar la gestión de la consulta, el pedido y la validación, y por último se vislumbra tres clases entidad que corresponden a los productos, los pedidos y los clientes, estas últimas se validarán y convertirán en las clase núcleo del sistema.

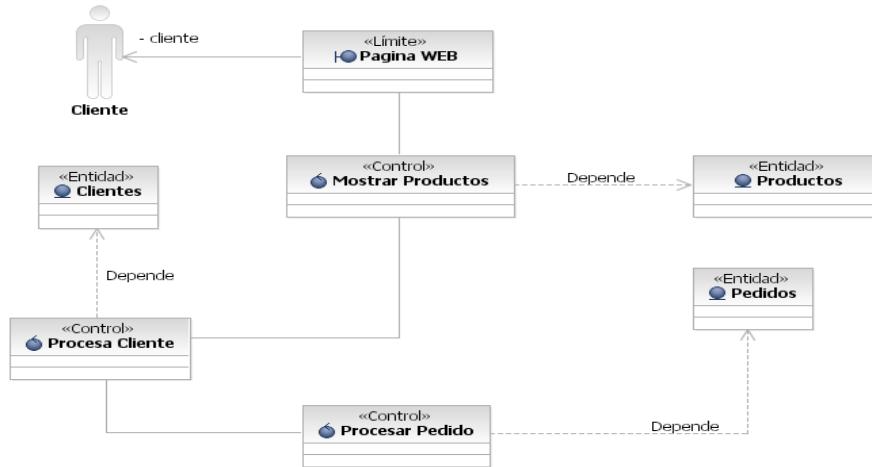


Figura 7 Diagrama de clases (Análisis): gestión del pedido

## Modelo Entidad Relación (UML 2.0)

En la fase el análisis de determinó el siguiente diagrama de clases núcleo, para ser tenido en cuenta en la fase de la creación de la base de datos inicial; su multiplicidad esta orientada a las relaciones que existen entre las entidades actuales.

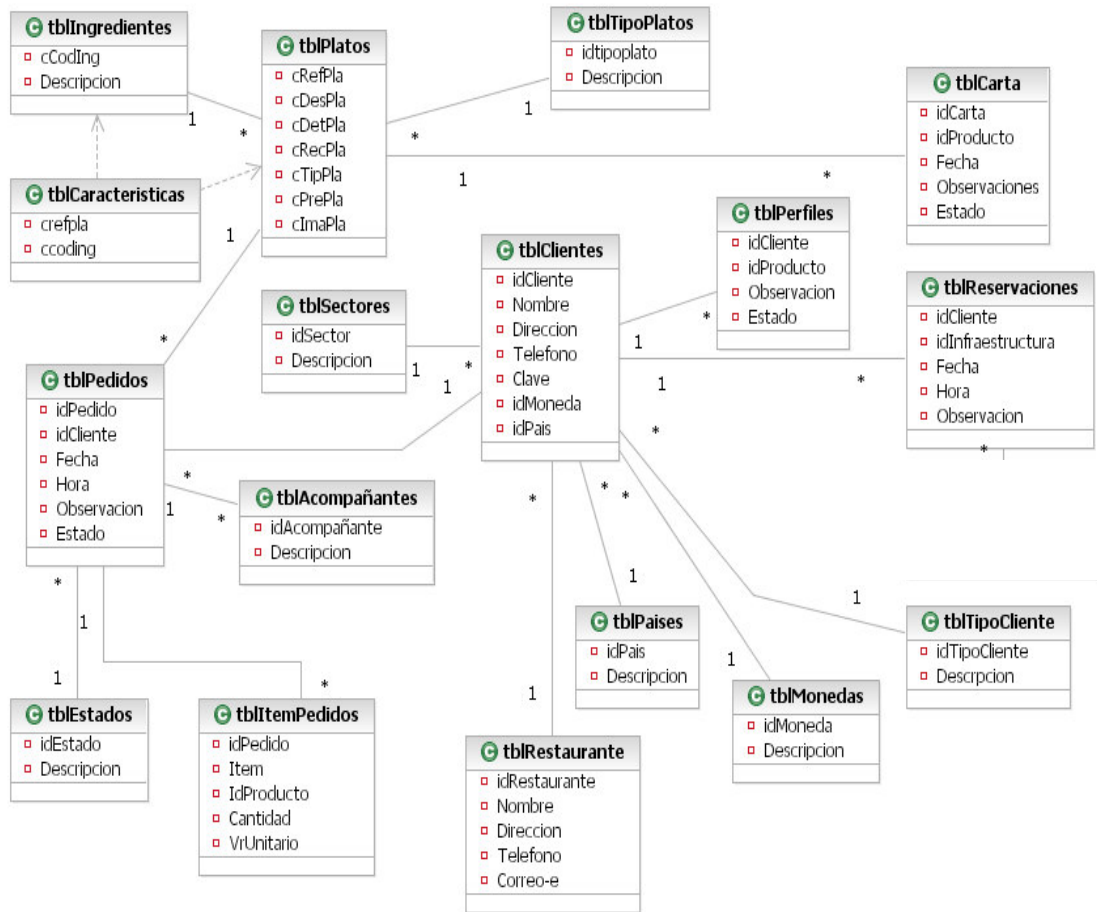


Figura 8 Modelo Entidad Relación



## Modelo Entidad Relación (Diagrama de clases núcleo)

Tomando como base la última versión de los libros de LARMAN y PRESSMAN, se define la relación de las clases como entidades, todo con el fin de aplicar conceptos más novedosos en el proyecto; abajo se ilustra un diagrama generado con RATIONAL MODELER 2006, el cual fue diagramado y compilado bajo los lineamientos de UML 2.0.

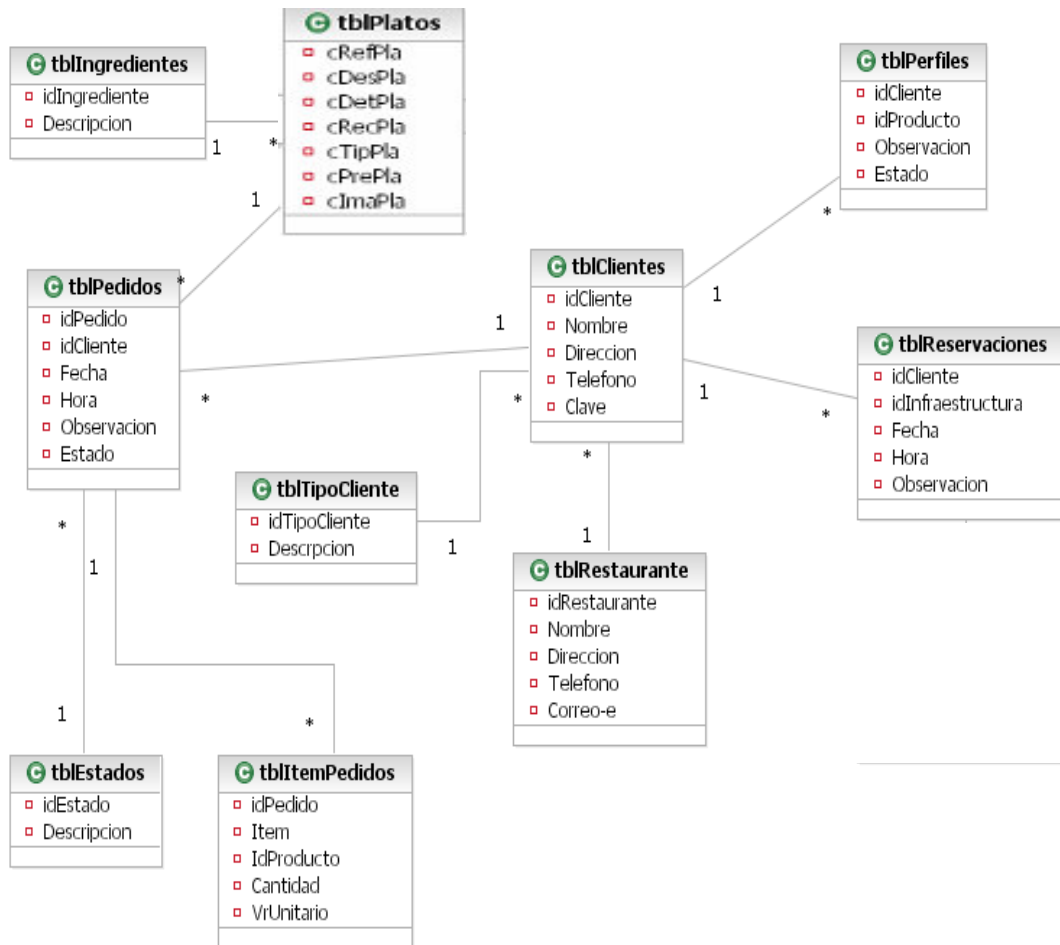


Figura 9 Diagrama de clases Núcleo (Relacional)

## 1.2.2 Sub-sistema Gestión Administrativa

### Diagrama de casos de Uso: gestión administrativa

El caso de uso para la gestión administrativa esta ligado al caso de uso el cual esta representado mediante la **figura 5**, en este caso se detalla la forma de interactuar las administrativas con los electos del modelo.

La forma de interactuar con la información del sistema esta ligada al ingreso, consulta, modificación y eliminación; todo está orientado a lo indicado en el modelo como clases núcleo, el cual se representa en UML 2.0 como un subsistema que apunta a dichos elementos representativos.

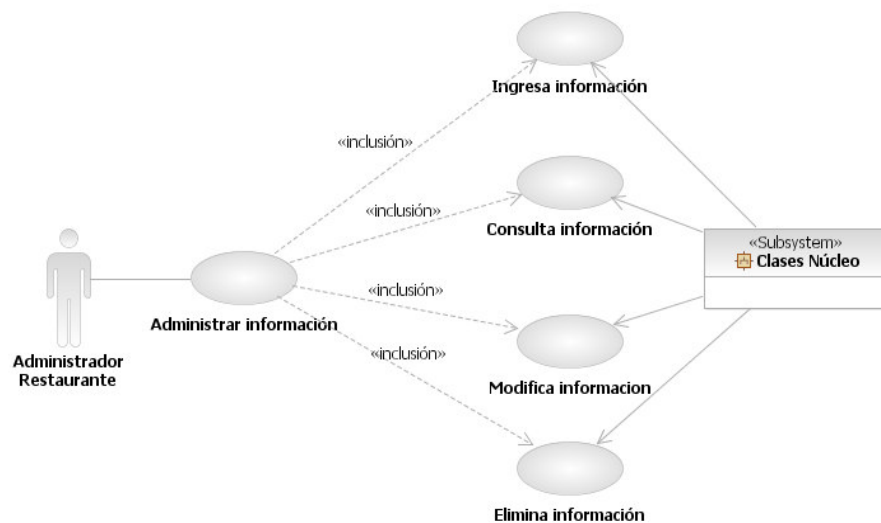


Figura 10 Diagrama de casos de Uso: gestión administrativa

### Diagrama de clases (Análisis): gestión administrativa (valida)

La gestión administrativa se logra mediante un acceso WEB, al cual cuenta con una clase de interfaz, esta a su vez, es controlada mediante una clase de interfaz (<<límite>>), la cual es la que permite la comunicación con la clase control validación clave, todo con el fin de poder hacer contacto con la clase de entidad Usuarios.

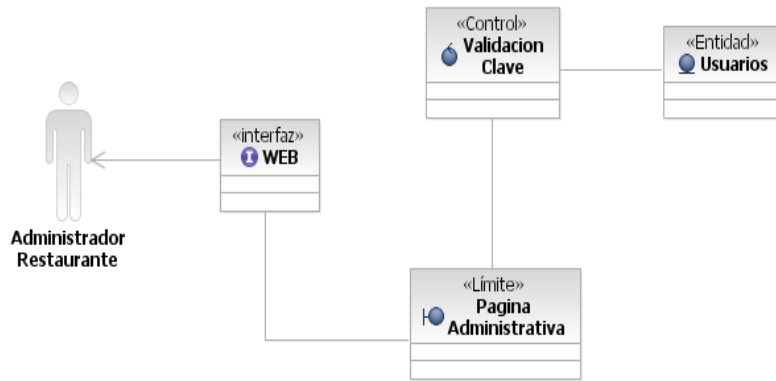


Figura 11 Diagrama de clases (Análisis): gestión Administrativa

## ANEXO C

### MODELO DE DISEÑO

#### 1. DISEÑO

##### 1.1. MODELO DE DESPLIEGUE

##### 1.1.1 Nodos y Configuraciones en la red

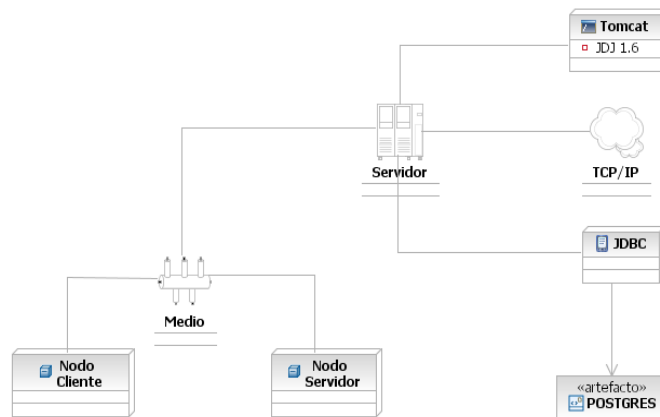


Figura 1 Configuraciones en la red

##### 1.2. MODELO DE COMPONENTES

##### 1.2.1 Diagrama de Componentes: Página principal

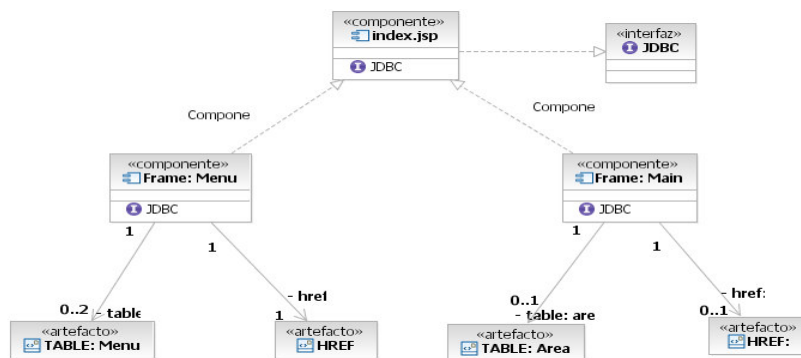


Figura 2 Diagrama de Componentes: Página Principal

## 1.2.2 Diseño estético: Página principal

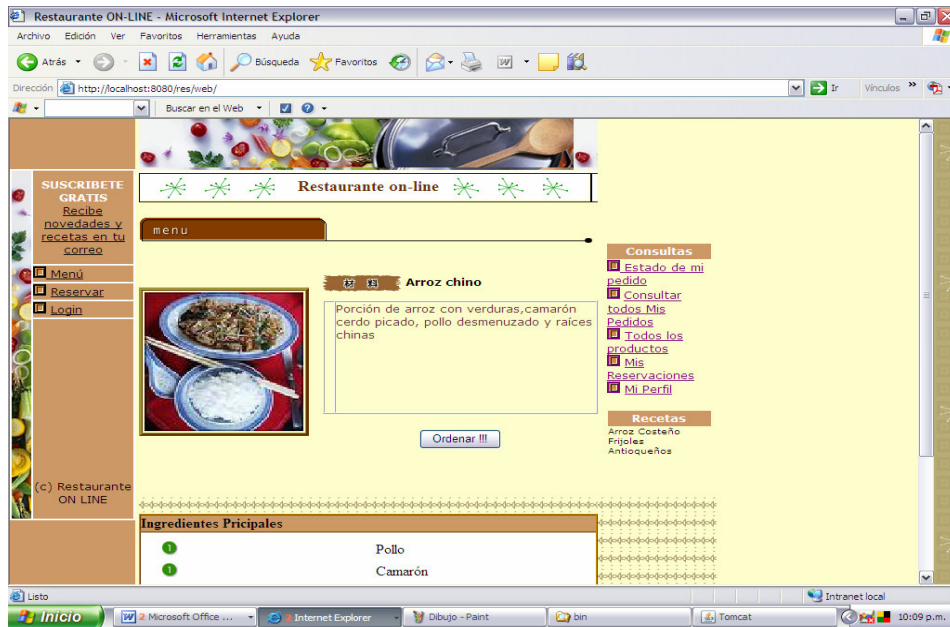


Figura 3 Diseño estético: Página principal

## 1.3. SUB-SISTEMA: GESTION DEL CLIENTE

### 1.3.1 Diagrama de Componentes: Registrarse en el Sistema

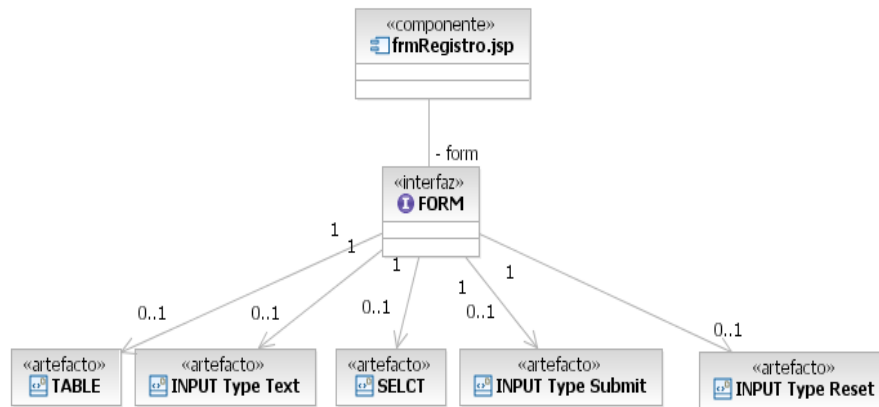


Figura 4 Diagrama de Componentes: Registrarse en el Sistema

### 1.3.2 Diagrama Estético: Registrarse en el Sistema

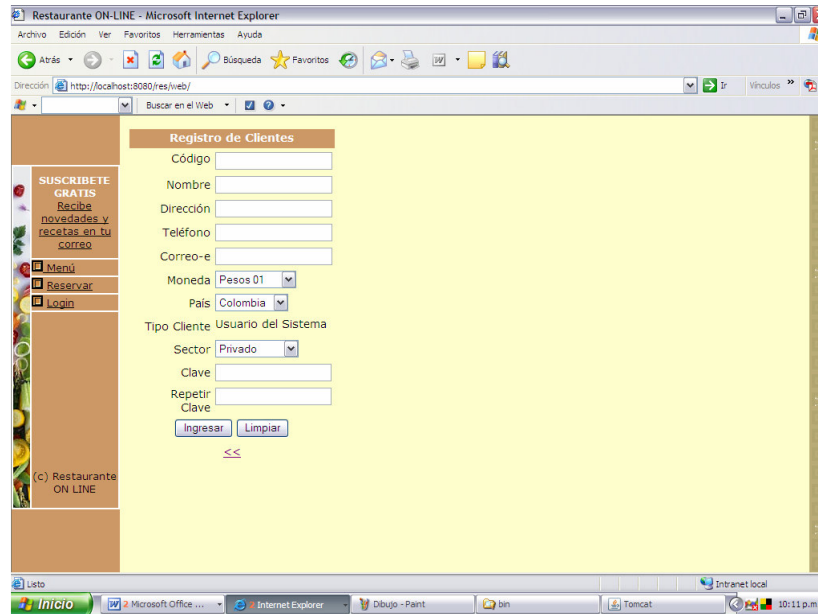


Figura 5 Diagrama Estético: Registrarse en el Sistema

### 1.3.3 Diagrama de estructura compuesta: Registrarse en el Sistema

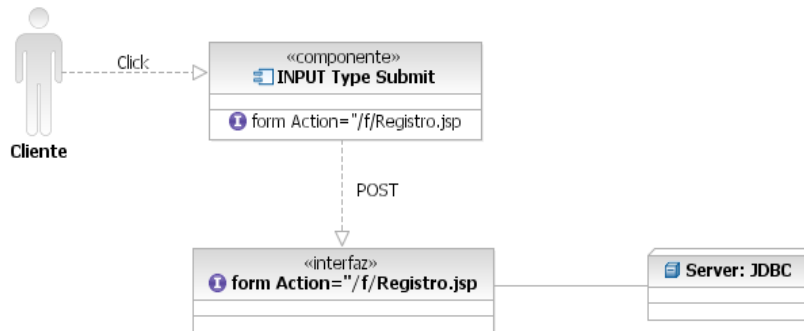


Figura 6 Diagrama de estructura compuesta: Registrarse en el Sistema

## 1.4. SUB-SISTEMA: GESTION ADMINISTRATIVA

### 1.4.1 Diagrama de Componentes: Menú Administrar Restaurantes

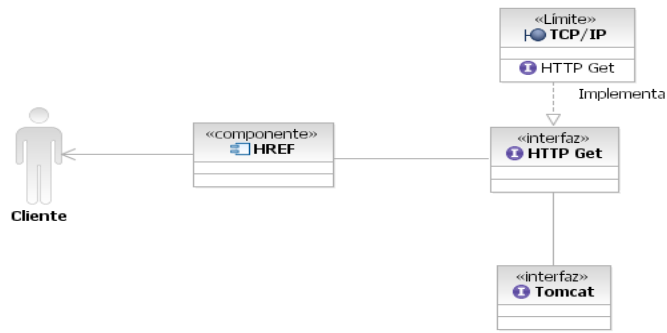


Figura 7 Diagrama de Componentes: Menú Administrar Restaurantes

### 1.4.2 Diagrama Estético: Menú Administrar Restaurantes

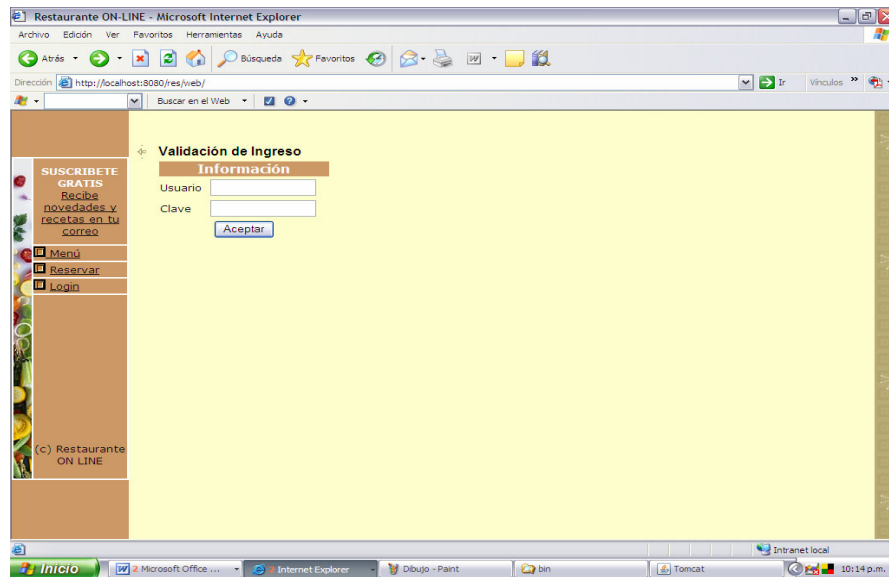


Figura 8 Diagrama Estético: Menú Administrar Restaurantes

### 1.4.4 Diagrama de Componentes: Administrar Restaurantes

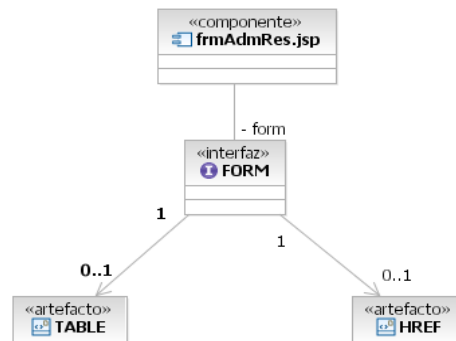


Figura 9 Diagrama de Componentes: Administrar Restaurantes

### 1.4.5 Diagrama Estético: Menú Administrar Restaurantes

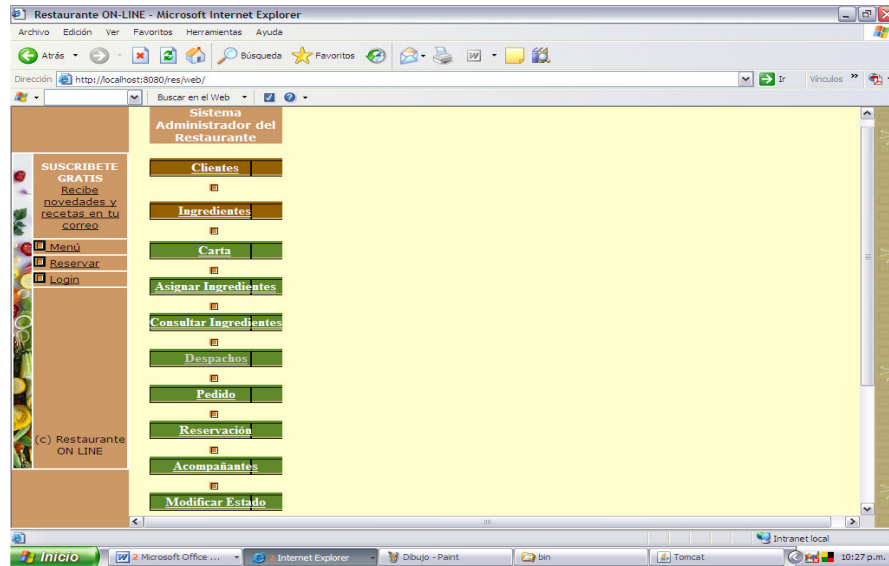


Figura 10 Diagrama Estético: Menú Administrar Restaurantes

### 1.4.6 Diagrama de estructura compuesta: Administrar Restaurantes

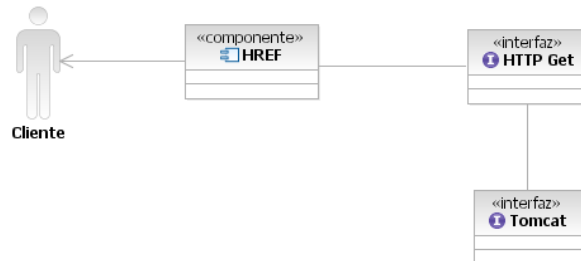


Figura 11 Diagrama de estructura compuesta: Administrar Restaurantes

### 1.4.7 Diagrama de Componentes: Menú Ingredientes

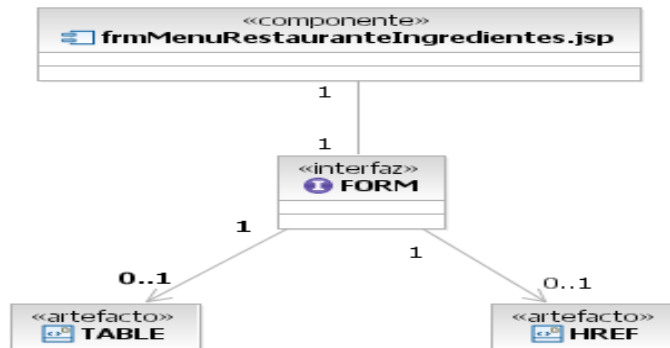


Figura 12 Diagrama de Componentes: Ingredientes



## 1.4.8 Diagrama Estético: Menú Ingredientes

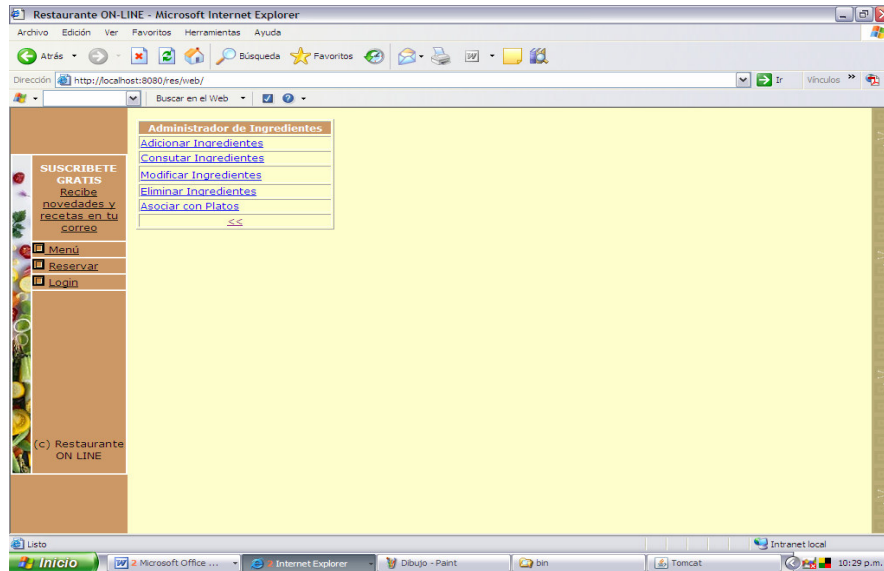


Figura 13 Diagrama Estético: Administrar Restaurantes

## 1.4.9 Diagrama de Componentes: Ingredientes

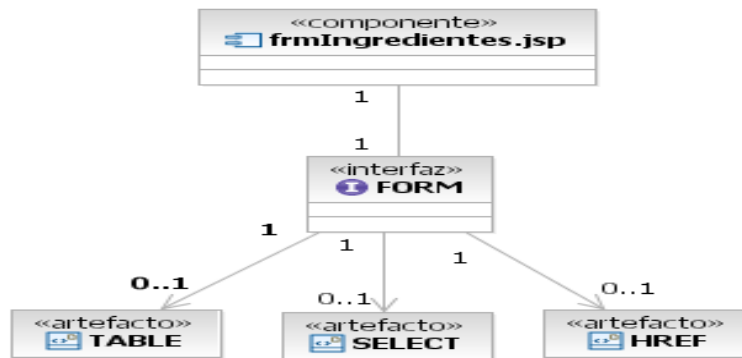


Figura 14 Diagrama de Componentes: Ingredientes

### 1.4.10 Diagrama Estético: Ingresar Ingredientes

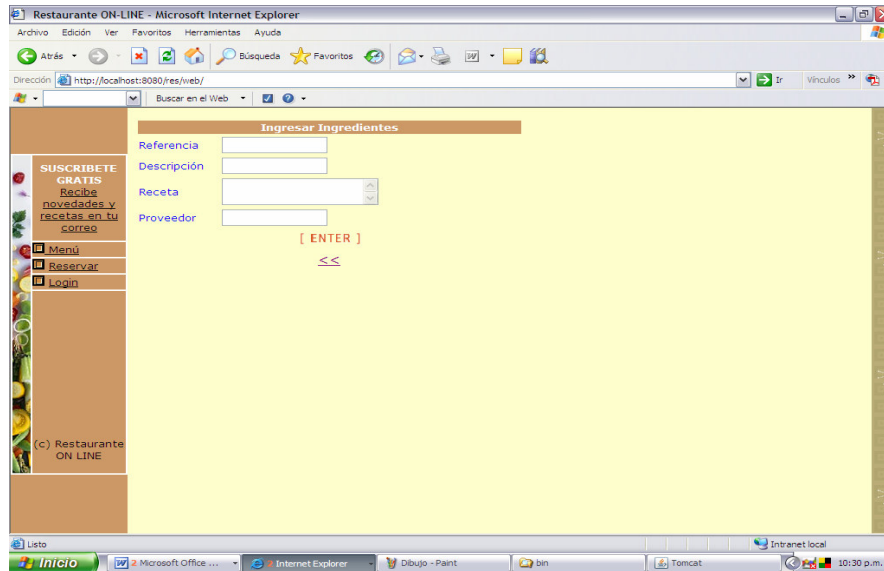


Figura 15 Diagrama Estético: Ingresar Ingredientes

### 1.4.11 Diagrama de estructura compuesta: Ingredientes

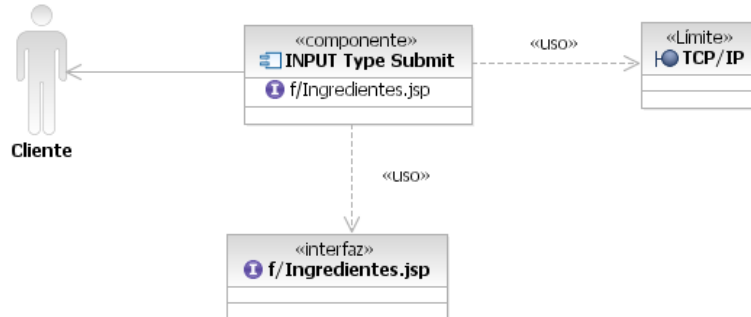


Figura 16 Diagrama de estructura compuesta: Ingredientes

### 1.4.12 Diagrama de estructura compuesta:

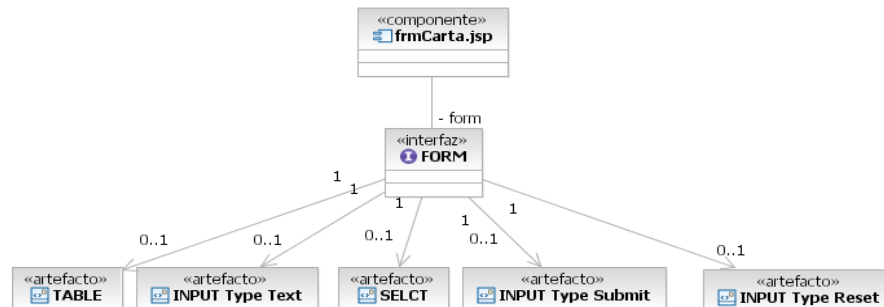


Figura 17 Diagrama de Componentes: Crear Platos Primera Ventana

### 1.4.13. Diagrama Estético: Menú Crear Platos

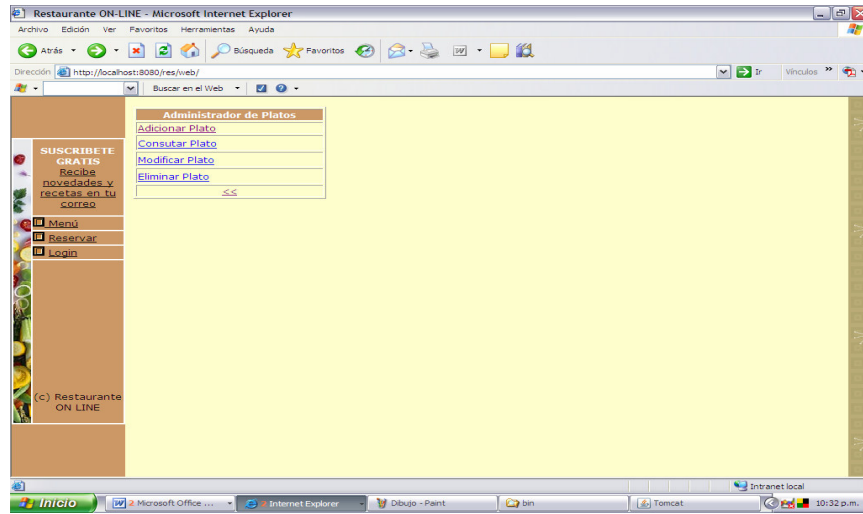


Figura 78 Diagrama Estético: Menú Crear Platos

## 1.5 SUB-SISTEMA: GESTIÓN CALLCENTER

Para el ingreso al subsistema de CALL CENTER es necesario identificarse en el sistema como un usuario pertinente para tales efectos, tal y como se diseña a continuación

### 1.5.1 Diagrama de estructura de comunicación: Ingreso al CALL CENTER

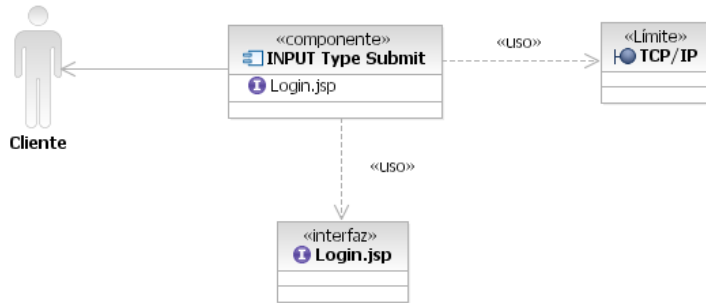


Figura 89 Diagrama de estructura de comunicación: Ingreso

### 1.5.2 Diagrama de estructura compuesta: Ingreso al CALL CENTER

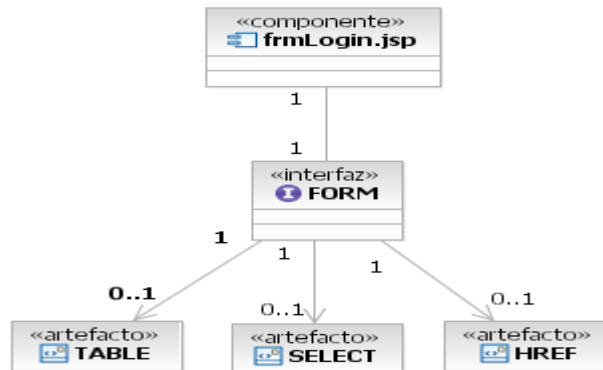


Figura 20 Diagrama de estructura comunicación: Ingreso

### 1.5.3 Diagrama Estético: Ingreso al CALL CENTER

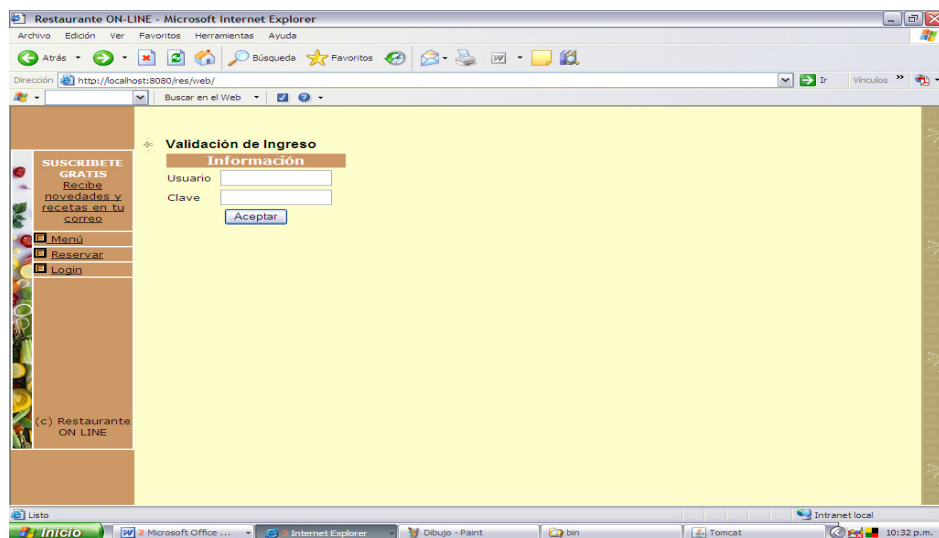


Figura 29 Diagrama Estético: Ingreso al CALL CENTER

### 1.5.4 Diagrama de estructura de comunicación: Menú general CALL CENTER

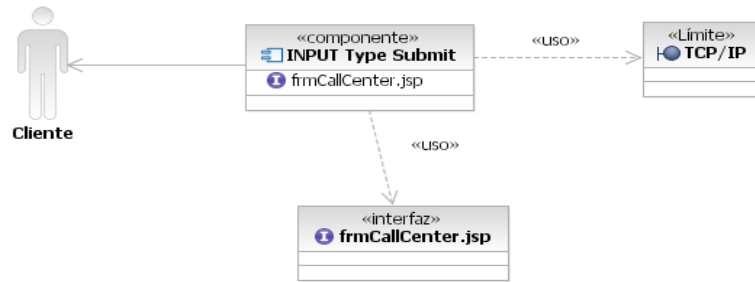


Figura 22 Menú general CALL CENTER

### 1.5.5 Diagrama de estructura compuesta: Menú general CALL CENTER

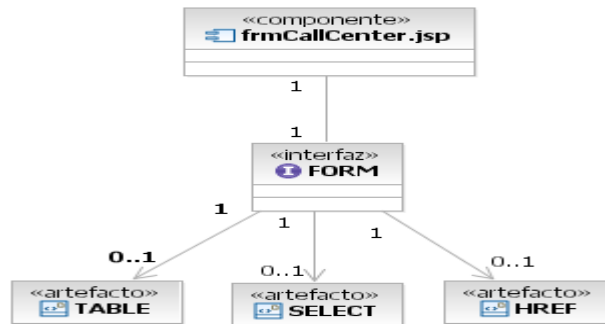


Figura 23 Menú general CALL CENTER

### 1.5.6 Diagrama Estético: Menú general CALL CENTER

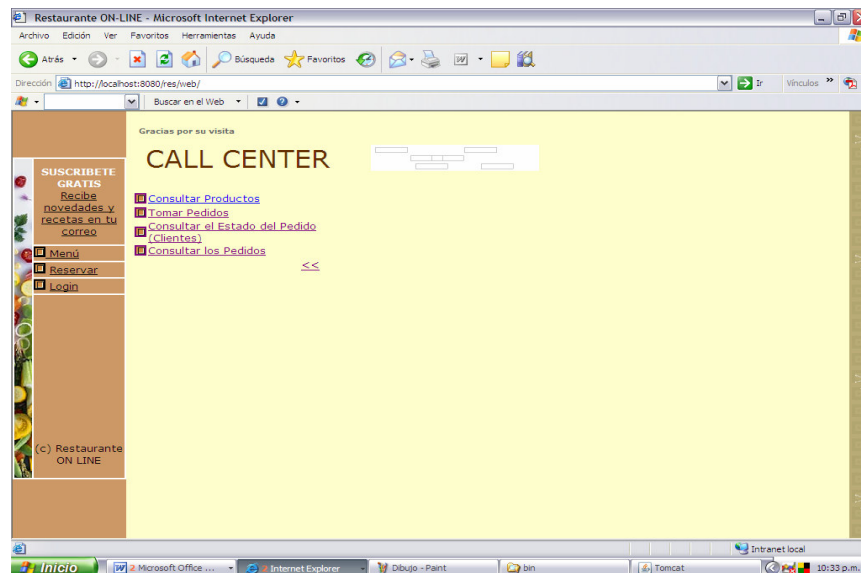


Figura 24 Menú general CALL CENTER

### 1.5.6 Diagrama de estructura de comunicación: Consultar Productos CALL CENTER

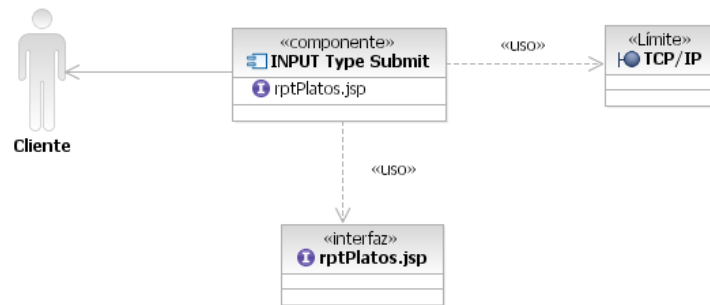


Figura 25 Diagrama de estructura de comunicación: Consultar Productos CALL CENTER

### 1.5.7 Diagrama de estructura compuesta: Consultar Productos CALL CENTER

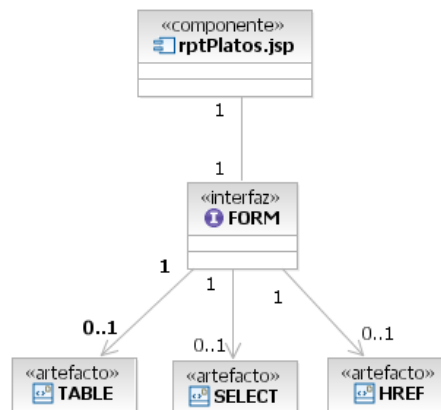


Figura 26 Diagrama de estructura compuesta: Consultar Productos CALL CENTER

### 1.5.8 Diagrama Estético: Consultar Productos CALL CENTER

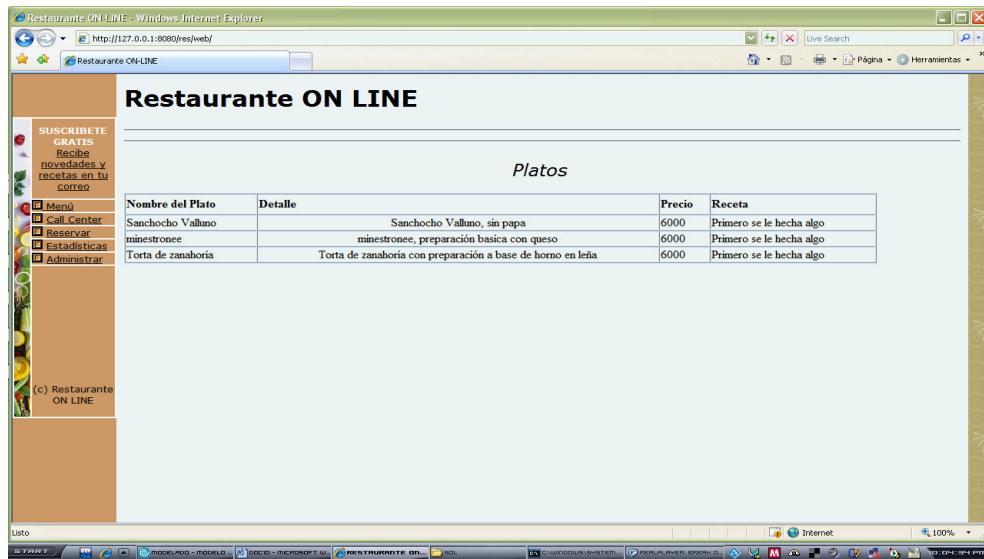


Figura 27 Diagrama Estético: Consultar Productos CALL CENTER

### 1.5.9 Diagrama de estructura de comunicación: Tomar Pedidos CALL CENTER

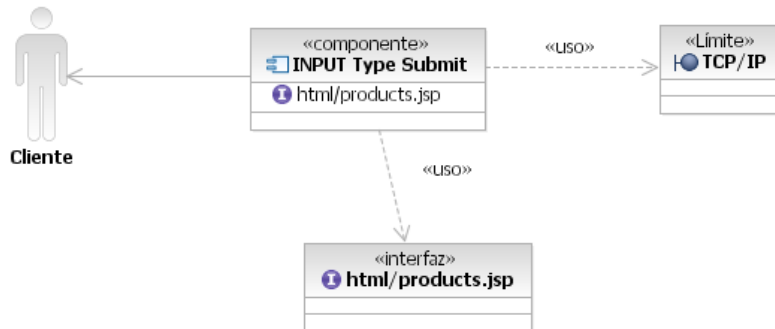


Figura 28 Diagrama de estructura de comunicación: Tomar Pedidos CALL CENTER

### 1.5.10 Diagrama de estructura compuesta: Tomar Pedidos CALL CENTER

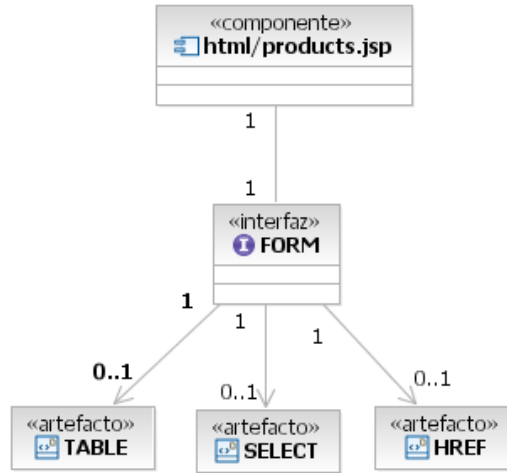


Figura 29 Diagrama de estructura compuesta: Tomar Pedidos CALL CENTER

### 1.5.11 Diagrama Estético: Tomar Pedidos CALL CENTER

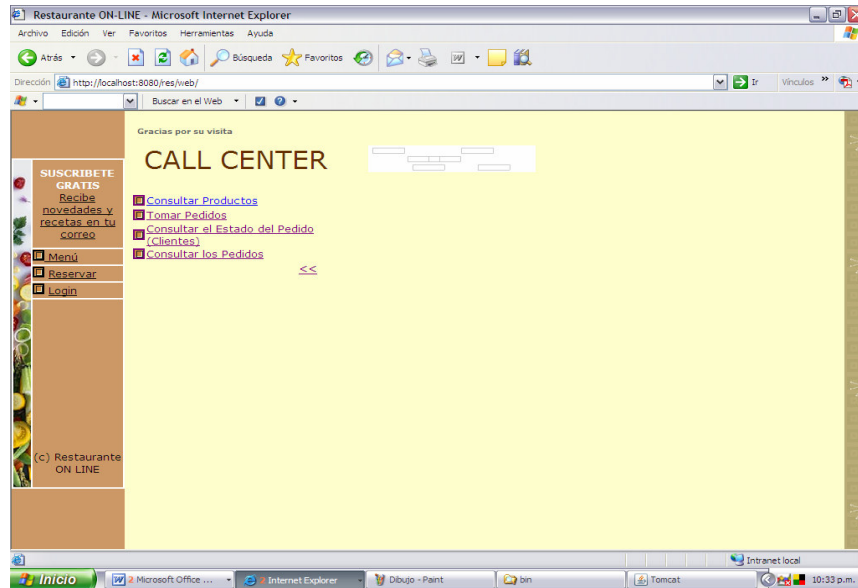


Figura 30 Diagrama Estético: Tomar Pedidos CALL CENTER



### 1.5.12 Diagrama de estructura de comunicación: Consulta Estado Pedido CALL CENTER

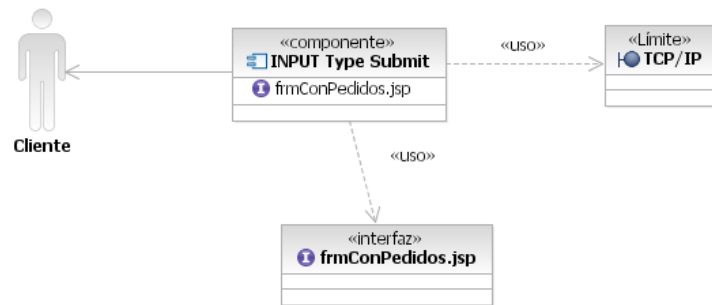


Figura 31 Diagrama de estructura de comunicación: Consulta Estado Pedido CALL CENTER

### 1.5.13 Diagrama de estructura compuesta: Consulta Estado Pedido CALL CENTER

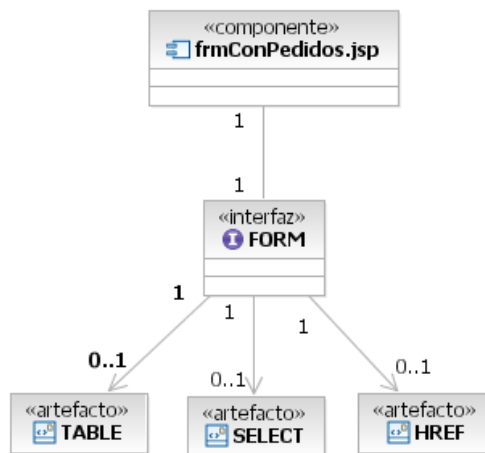


Figura 32 Diagrama de estructura compuesta: Consulta Estado Pedido CALL CENTER

### 1.5.14 Diagrama Estético: Consulta Estado Pedido CALL CENTER

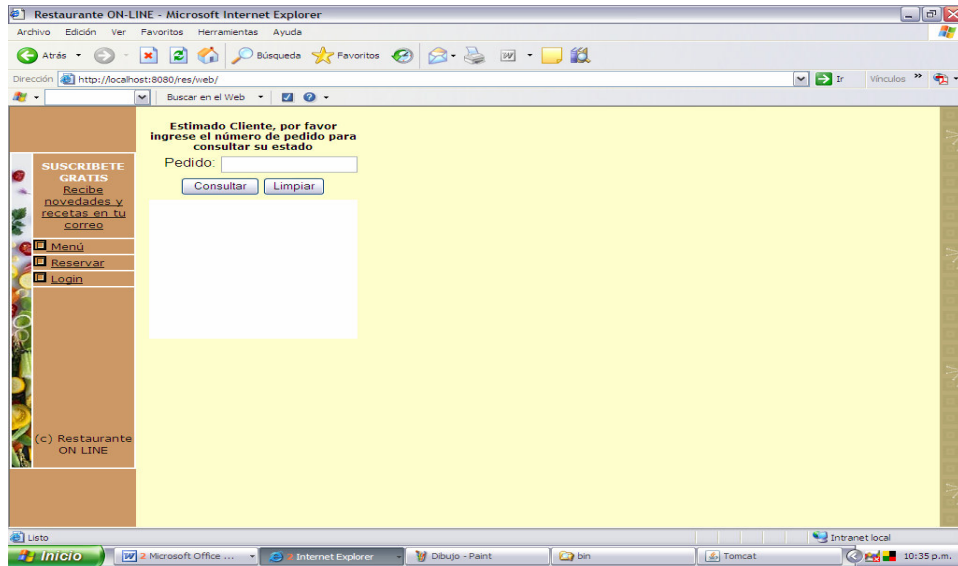


Figura 33 Diagrama Estético: Consulta Estado Pedido CALL CENTER

### 1.5.15 Diagrama de estructura de comunicación: Consulta los Pedido CALL CENTER

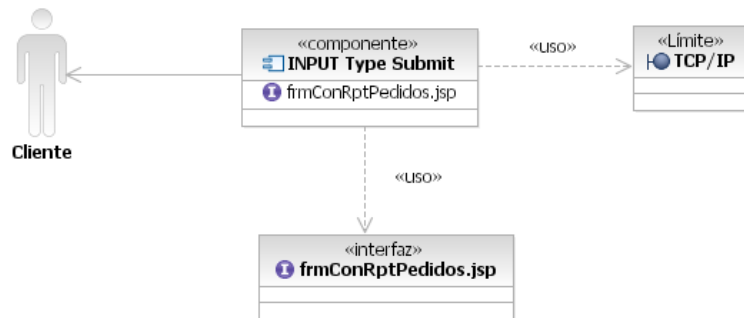


Figura 34 Diagrama de estructura de comunicación: Consulta los Pedido CALL CENTER

### 1.5.16 Diagrama de estructura compuesta: Consulta los Pedido CALL CENTER

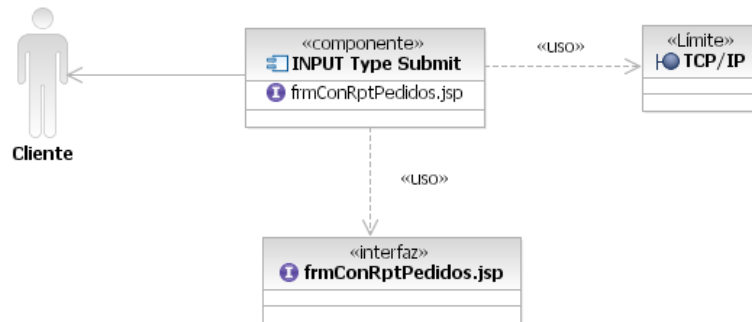


Figura 35 Diagrama de estructura compuesta: Consulta los Pedido CALL CENTER

### 1.5.17 Diagrama Estético: Consulta Estado los CALL CENTER (Parte A)

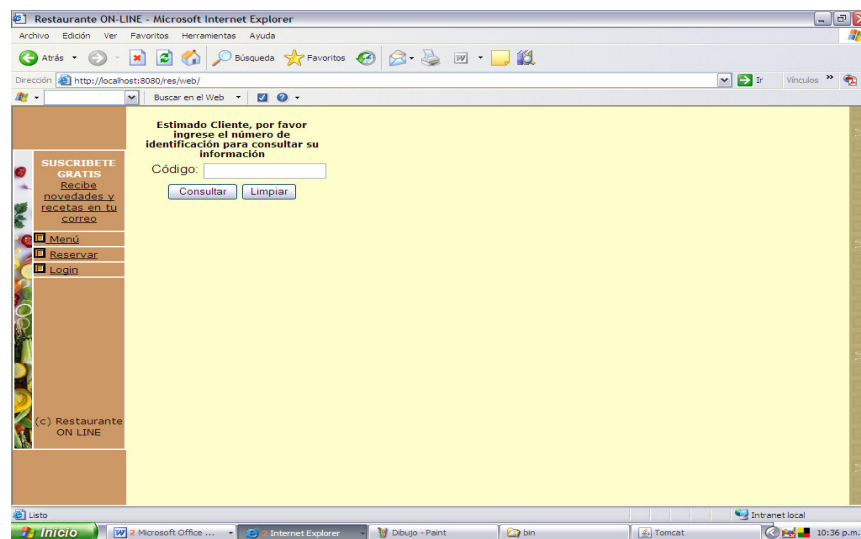


Figura 36 Diagrama Estético: Consulta estado los CALL CENTER (Parte A)

### 1.5.18 Diagrama Estético: Consulta Estado los CALL CENTER (Parte B)

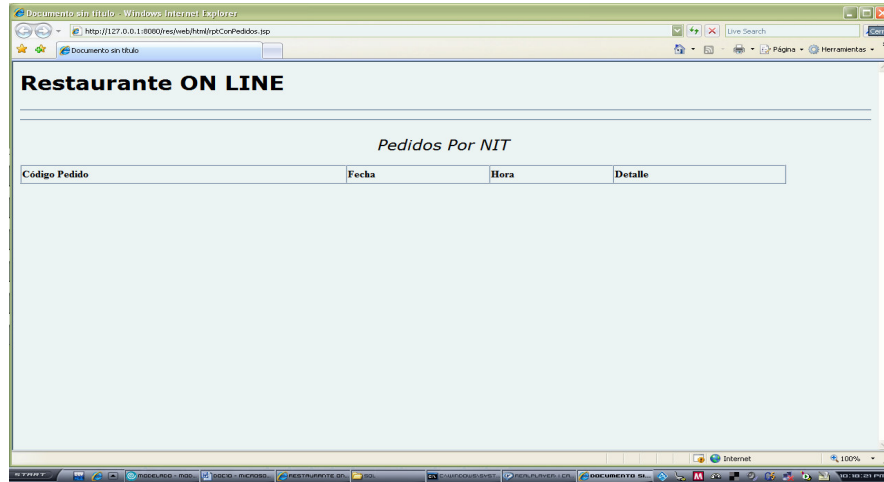


Figura 37 Diagrama Estético: Consulta estado los CALL CENTER (Parte B)

### 1.6 SUB-SISTEMA: RESERVACIONES

En el momento de realizar una reservación es necesario tener en cuenta aspectos tales como la la fecha, la mesa, y otros datos de importancia que se muestran a continuación en los diagramas de diseño utilizados para cada caso

#### 1.6.. Diagrama de estructura de comunicación: Reservaciones

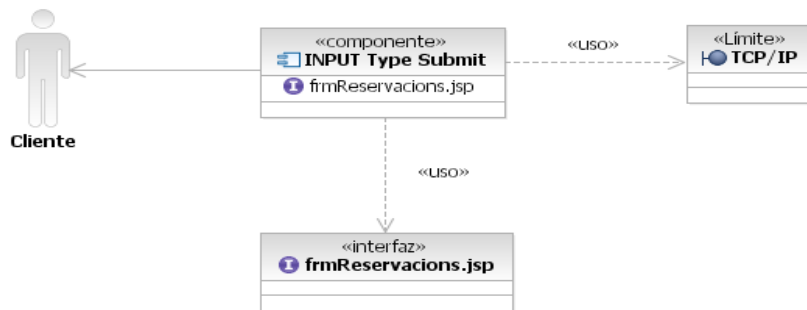


Figura 38 Diagrama de estructura de comunicación: Reservaciones

## 1.6.2 Diagrama de estructura compuesta: Reservasiones

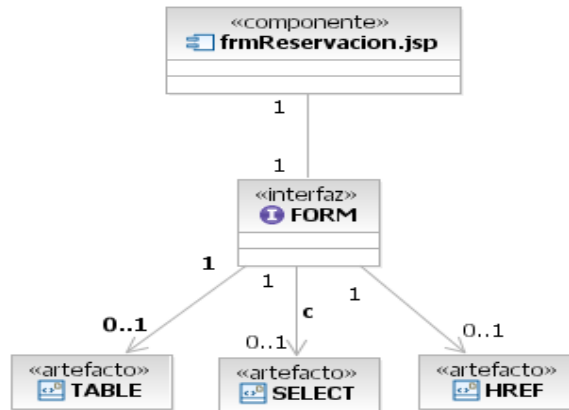


Figura 39 Diagrama de estructura de comunicaci3n: Reservasiones

## 1.6.3 Diagrama Est3tico: Reservasiones

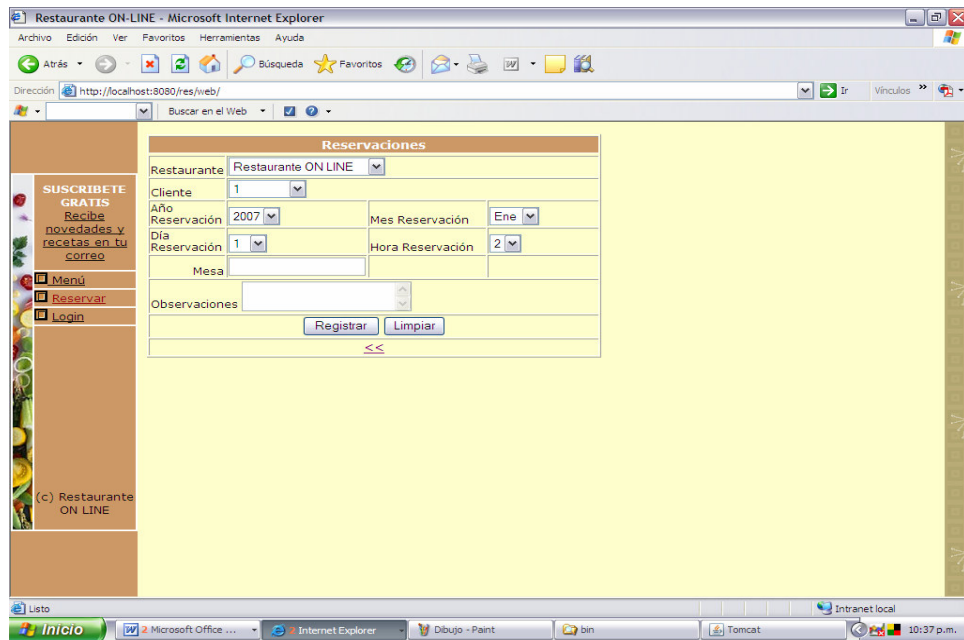


Figura 40 Diagrama Est3tico: Reservasiones

## 1.7. SUB-SISTEMA: ESTADISTICAS

El diseño de las estadísticas están estructurada en forma estricta tal y como lo requiere el beneficiario del proyecto, el cual, definen dos tipos de estadísticas: reportes administrativos y reportes estadísticos; éstos se estructuran en el diseño del sistema con base a dichos lineamientos en los siguientes diagramas de UML 2.0 y UP.

### 1.7.1 Diagrama de estructura de comunicación: Menú estadísticas

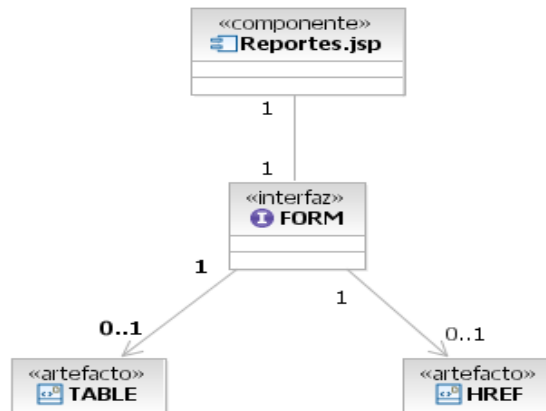


Figura 41 Diagrama de estructura de comunicación: Menú estadísticas (Clientes)

### 1.7.2 Diagrama de estructura compuesta: Menú estadísticas

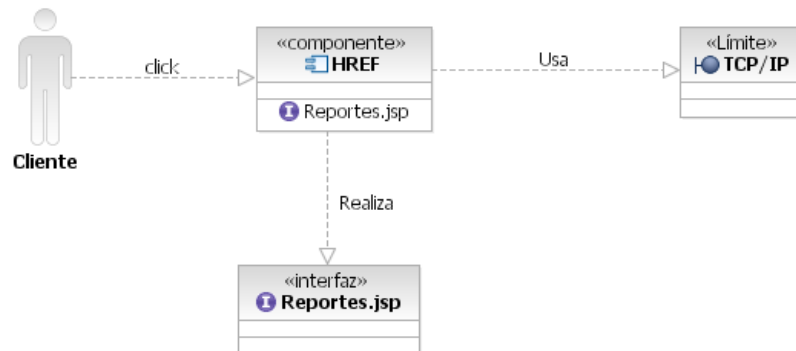


Figura 42 Diagrama de estructura compuesta: Menú estadísticas

### 1.7.3 Diagrama Estético: Menú estadísticas

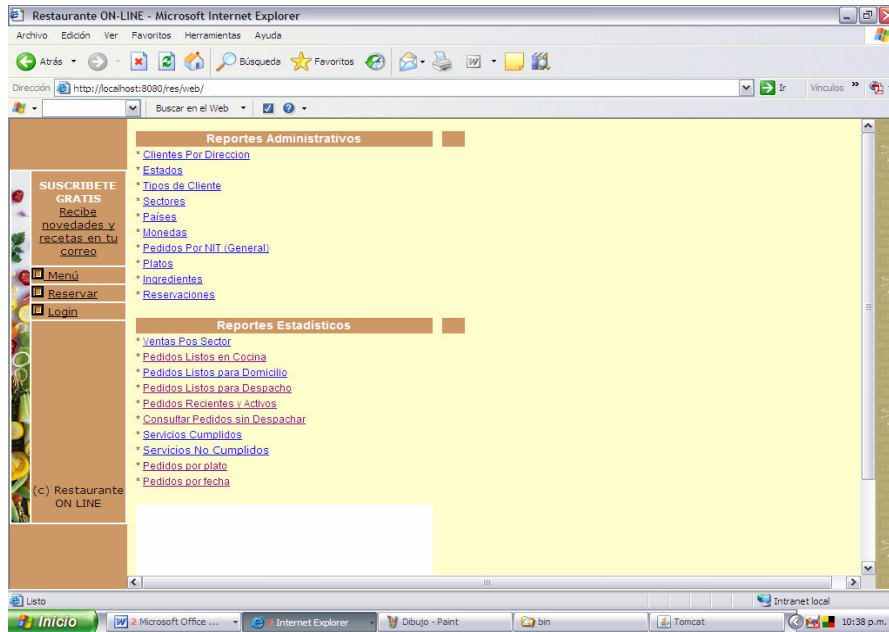


Figura 43 Diagrama Estético: Menú estadísticas

### 1.7.4 Diagrama de estructura de comunicación: Reportes Administrativos (Clientes)

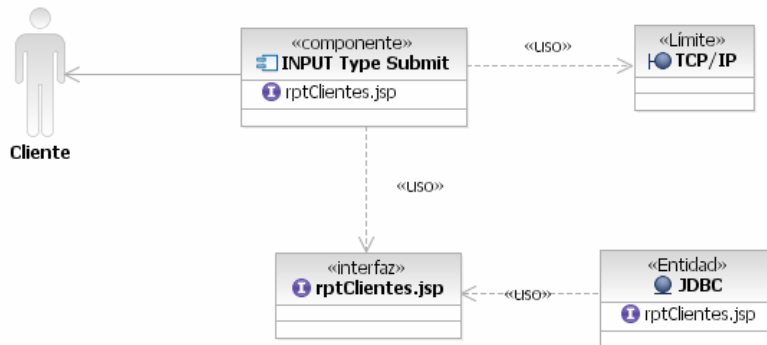


Figura 44 Diagrama de estructura de comunicación: Menú estadísticas (Clientes)

### 1.7.5 Diagrama de estructura compuesta: Reportes Administrativos (Clientes)

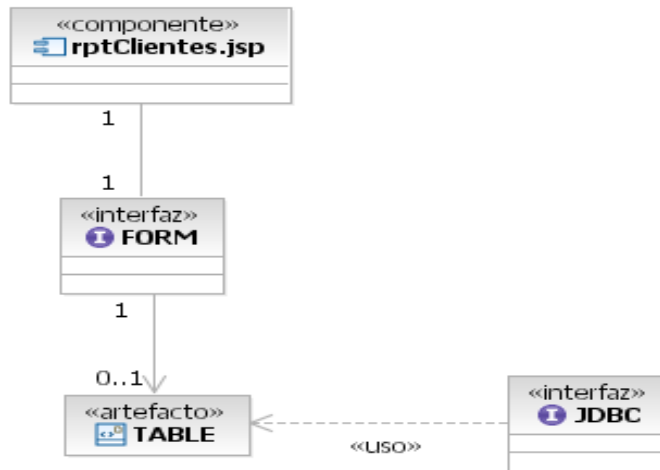


Figura 45 Diagrama de estructura compuesta: Menú estadísticas

### 1.7.6 Diagrama Estético: Reportes Administrativos (Clientes)

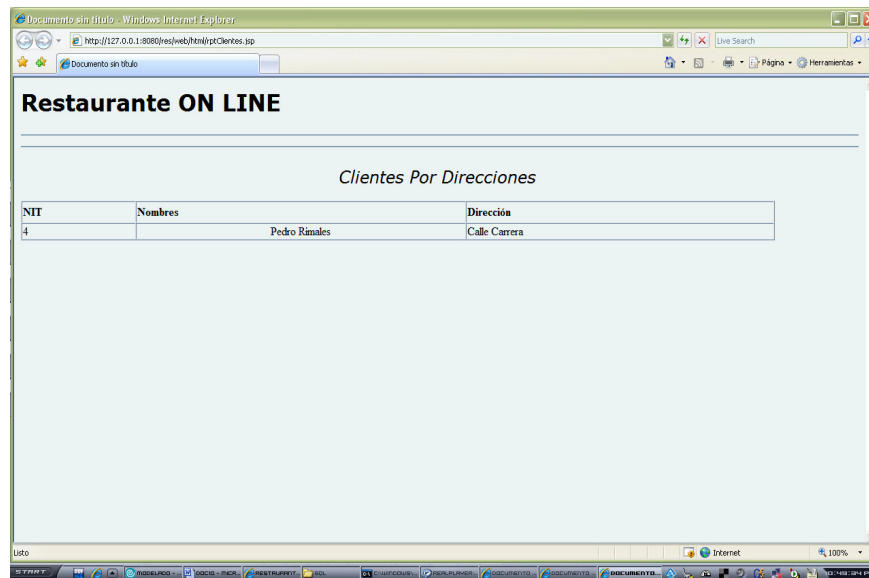


Figura 46 Diagrama Estético: Menú estadísticas



### 1.7.7 Diagrama de estructura de comunicación: Reportes Administrativos (Estados)

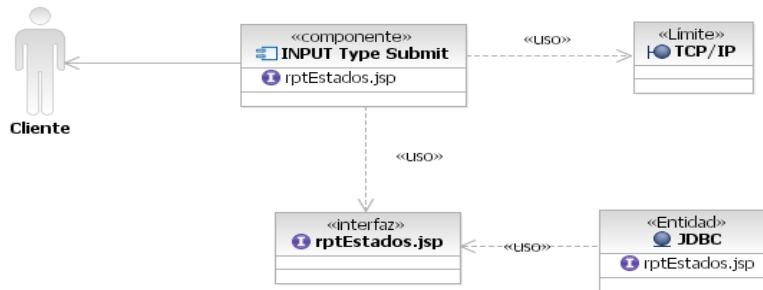


Figura 47 Diagrama de estructura de comunicación: Reportes Administrativos (Estados)

### 1.7.8 Diagrama de estructura compuesta: Reportes Administrativos (Estados)

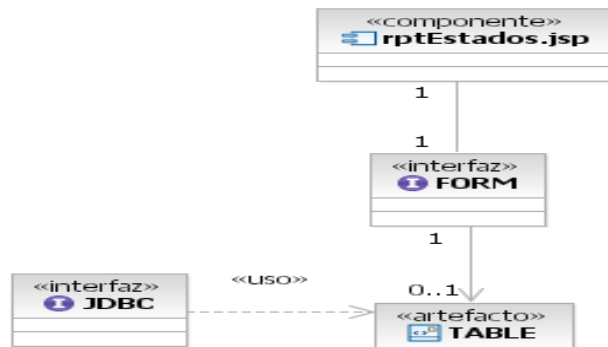


Figura 48 Diagrama de estructura compuesta: Reportes Administrativos (Estados)

### 1.7.9 Diagrama Estético: Reportes Administrativos (Estados)

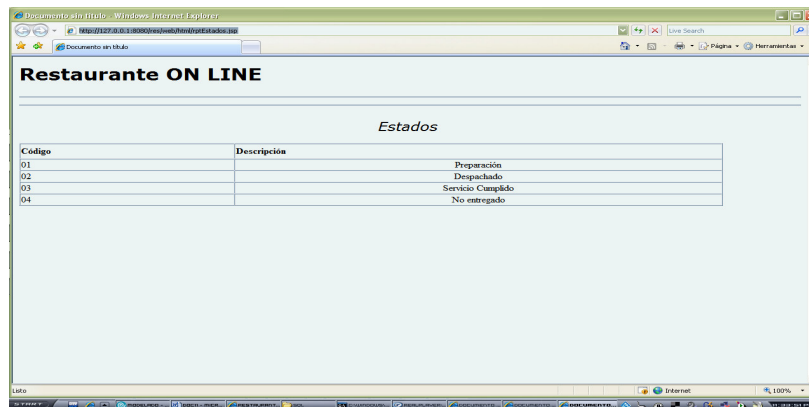


Figura 49 Diagrama Estético: Reportes Administrativos (Estados)

### 1.7.10 Diagrama de estructura de comunicación: Reportes Administrativos (Tipo de Clientes)

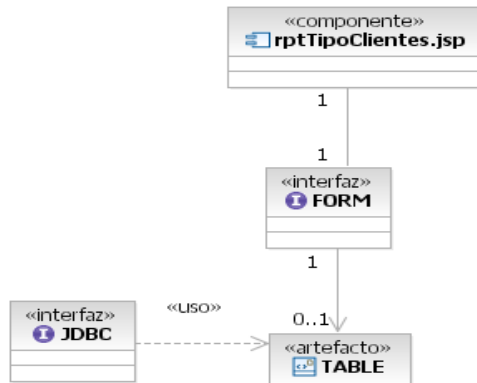


Figura 50 Diagrama de estructura de comunicación: Reportes Administrativos (Tipo de Clientes)

### 1.7.11 Diagrama de estructura compuesta: Reportes Administrativos (Tipo de Clientes)

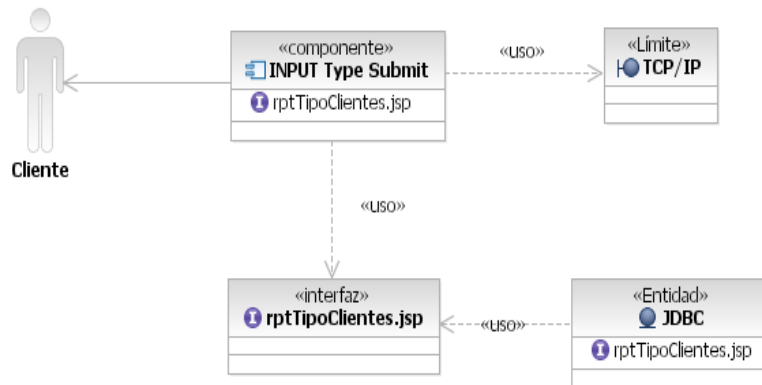


Figura 51 Diagrama de estructura compuesta: Reportes Administrativos (Tipo de Clientes)

### 1.7.12 Diagrama Estético: Reportes Administrativos (Tipo de Clientes)

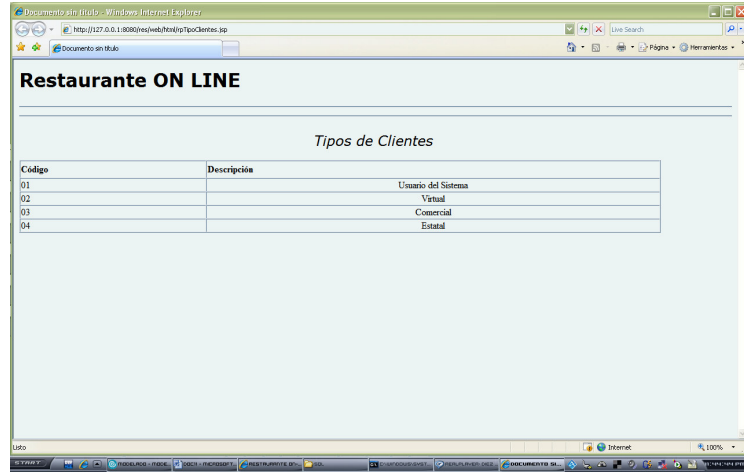


Figura 52 Diagrama Estético: Reportes Administrativos (Tipo de Clientes)

### 1.7.13 Diagrama de estructura de comunicación: Reportes Administrativos (Sectores)

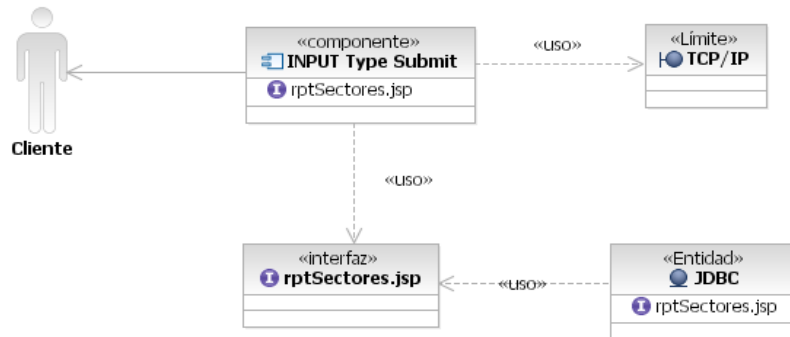


Figura 53 Diagrama de estructura de comunicación: Reportes Administrativos (Sectores)

### 1.7.14 Diagrama de estructura compuesta: Reportes Administrativos (Sectores)

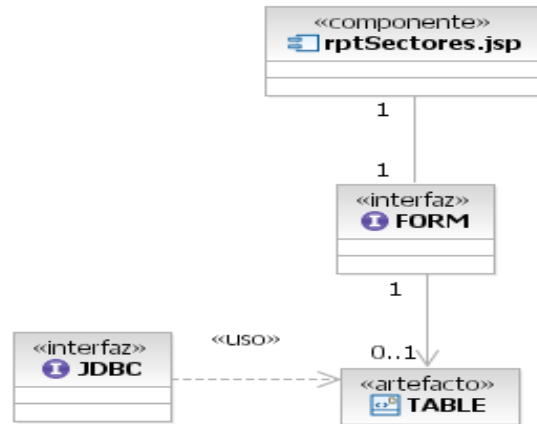


Figura 54 Diagrama de estructura compuesta: Reportes Administrativos (Sectores)

### 1.7.15 Diagrama Estético: Reportes Administrativos (Sectores)

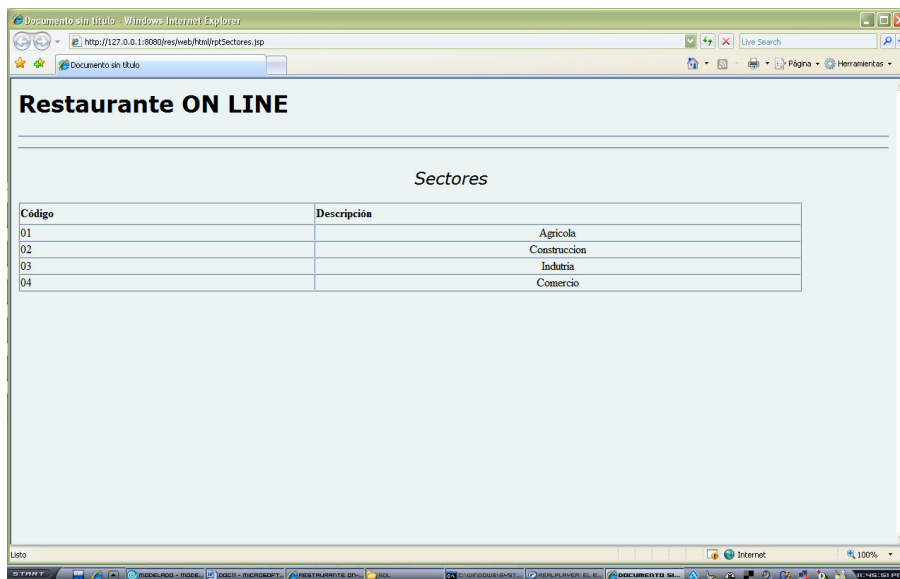


Figura 55 Diagrama Estético: Reportes Administrativos (Sectores)

### 1.7.16 Diagrama de estructura de comunicación: Reportes Administrativos (Países)

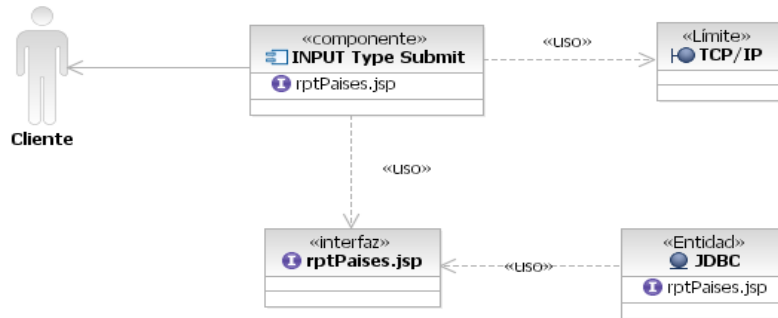


Figura 56 Diagrama de estructura de comunicación: Reportes Administrativos (Países)

### 1.7.17 Diagrama de estructura compuesta: Reportes Administrativos (Países)

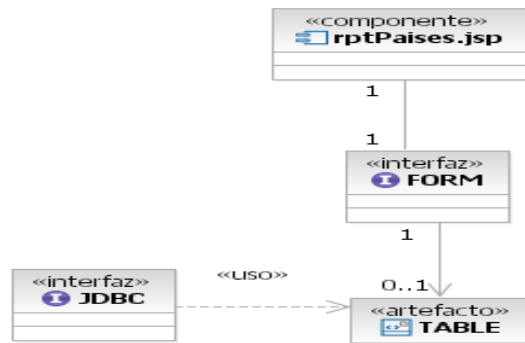


Figura 57 Diagrama de estructura compuesta: Reportes Administrativos (Países)

### 1.7.18 Diagrama Estético: Reportes Administrativos (Países)

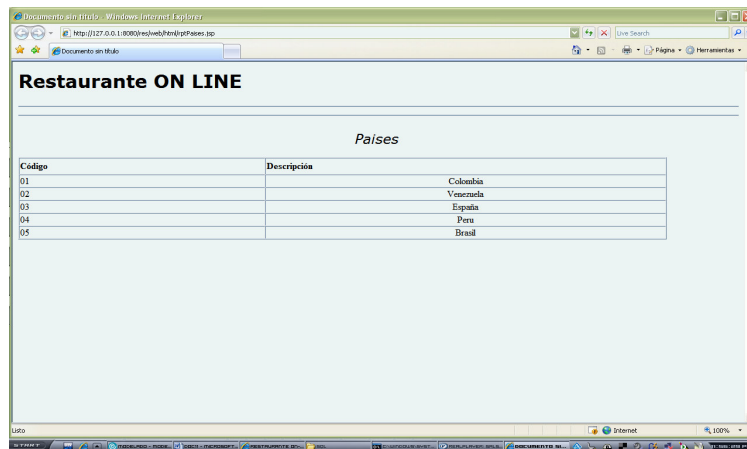


Figura 58 Diagrama Estético: Reportes Administrativos (Países)

### 1.7.16 Diagrama de estructura de comunicación: Reportes Administrativos (Monedas)

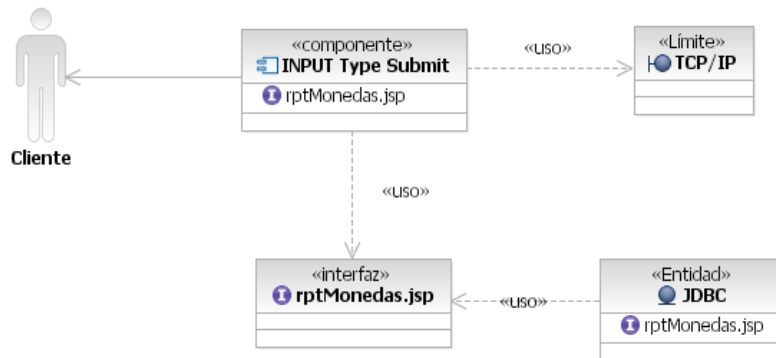


Figura 59 Diagrama de estructura de comunicación: Reportes Administrativos (Monedas)

### 1.7.17 Diagrama de estructura compuesta: Reportes Administrativos (Monedas)

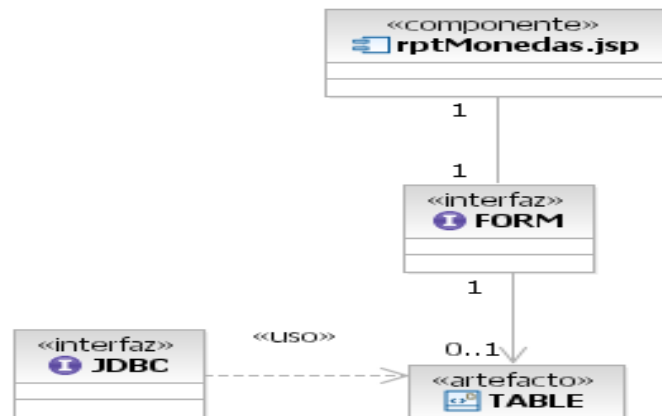


Figura 60 Diagrama de estructura compuesta: Reportes Administrativos (Monedas)

### 1.7.18 Diagrama Estético: Reportes Administrativos (Monedas)

Restaurante ON LINE

*Monedas*

Código	Descripción
01	Pesos 01
02	Dolares 02
03	Bolívares 03

Figura 61 Diagrama Estético: Reportes Administrativos (Monedas)

### 1.7.19 Diagrama de estructura de comunicación: Reportes Administrativos (Pedidos por NIT)

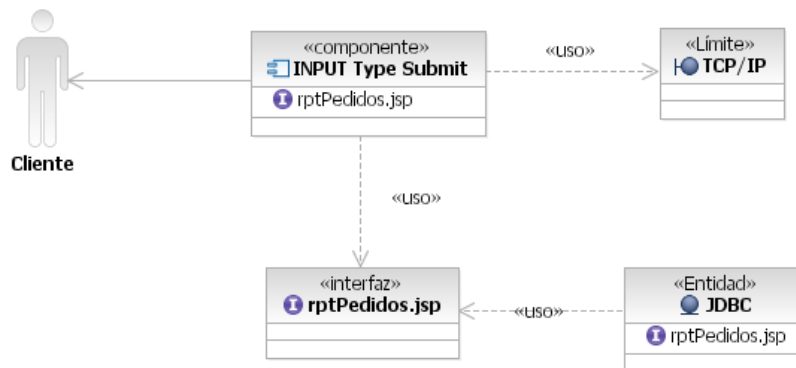


Figura 62 Diagrama de estructura de comunicación: Reportes Administrativos (Pedidos por NIT)

### 1.7.20 Diagrama de estructura compuesta: Reportes Administrativos (Pedidos por NIT)

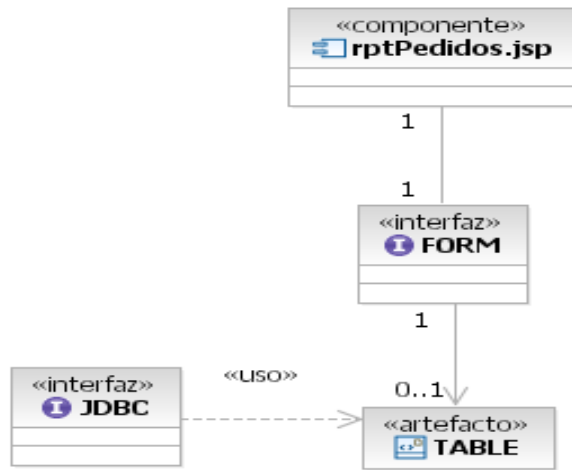


Figura 63 Diagrama de estructura compuesta: Reportes Administrativos (Pedidos por NIT)

### 1.7.21 Diagrama Estético: Reportes Administrativos (Pedidos por NIT)

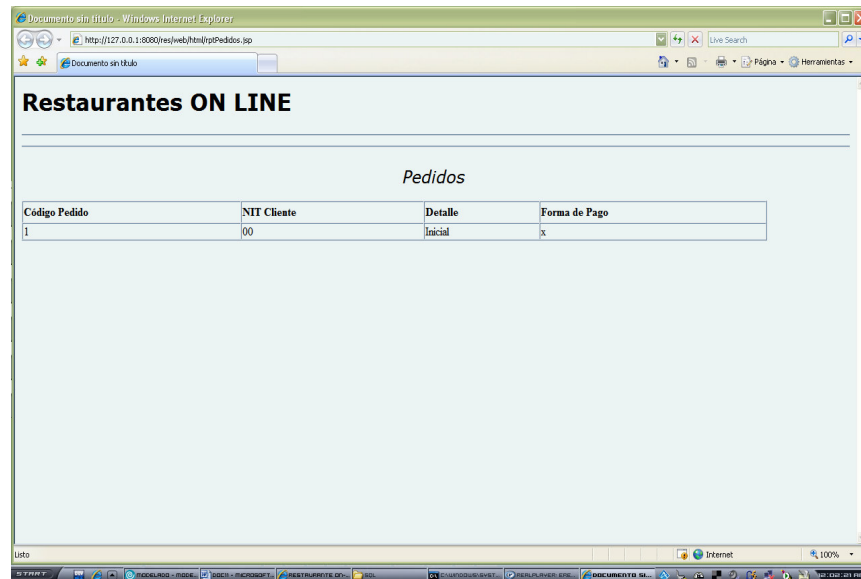


Figura 64 Diagrama Estético: Reportes Administrativos (Pedidos por NIT)



### 1.7.25 Diagrama de estructura de comunicación: Reportes Administrativos (Platos)

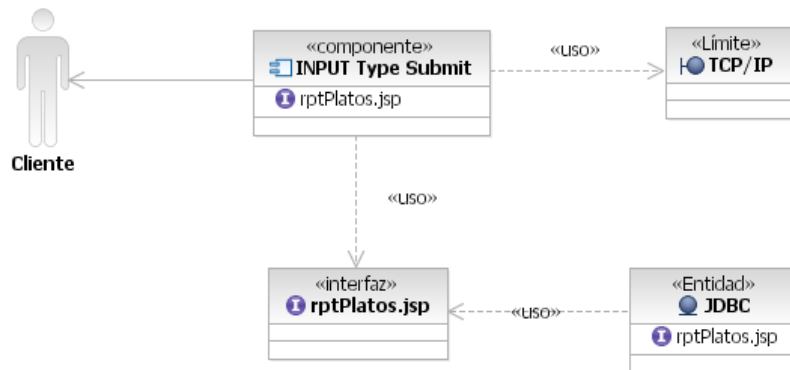


Figura 65 Diagrama de estructura de comunicación: Reportes Administrativos (Platos)

### 1.7.26 Diagrama de estructura compuesta: Reportes Administrativos (Platos)

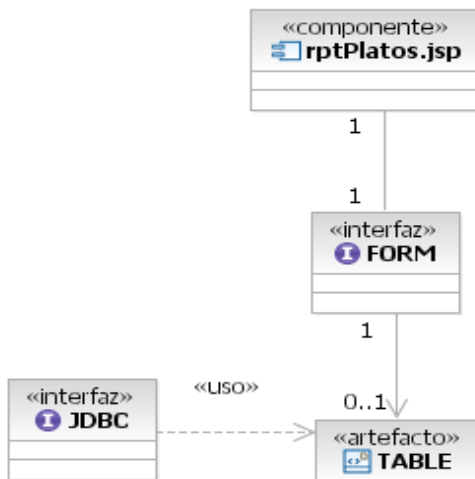


Figura 66 Diagrama de estructura compuesta: Reportes Administrativos (Platos)

### 1.7.27 Diagrama Estético: Reportes Administrativos (Platos)

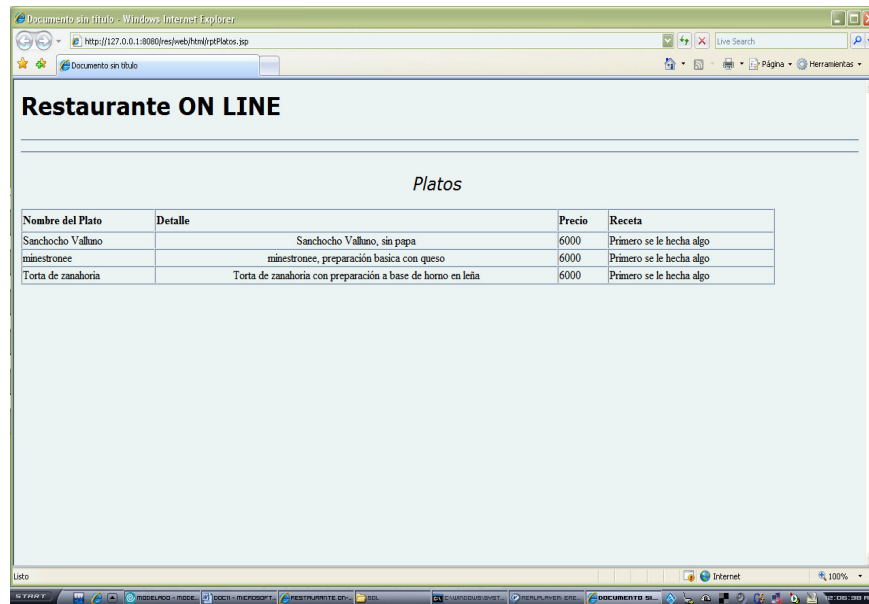


Figura 67 Diagrama Estético: Reportes Administrativos (Platos)

### 1.7.28 Diagrama de estructura de comunicación: Reportes Administrativos (Ingredientes)

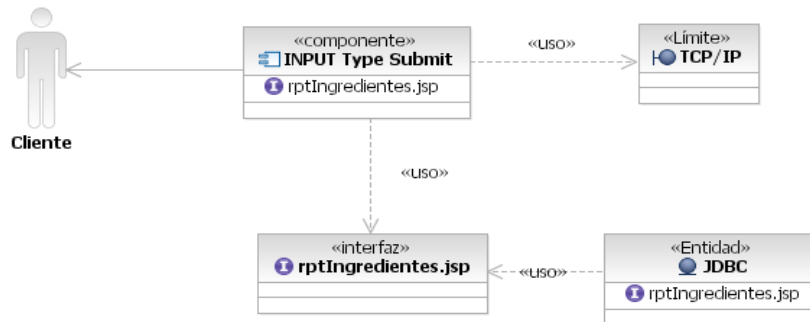


Figura 68 Diagrama de estructura de comunicación: Reportes Administrativos (Ingredientes)

### 1.7.29 Diagrama de estructura compuesta: Reportes Administrativos (Ingredientes)

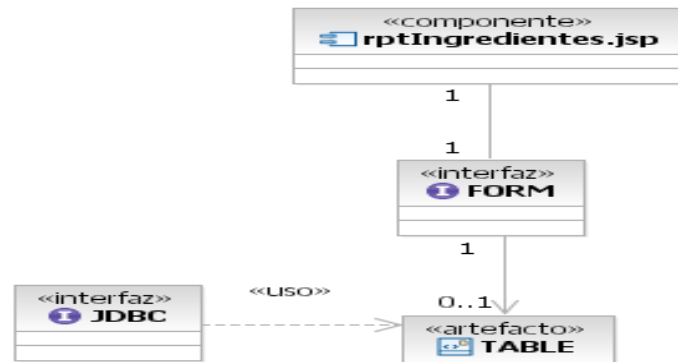


Figura 69 Diagrama de estructura compuesta: Reportes Administrativos (Ingredientes)

### 1.7.30 Diagrama Estético: Reportes Administrativos (Ingredientes)

The screenshot shows a web browser window displaying a page titled 'Restaurante ON LINE'. The page content includes a table of ingredients. The table has four columns: 'Codigo', 'Nombre del Ingrediente', 'Receta', and 'Proveedor'. The data rows are as follows:

Codigo	Nombre del Ingrediente	Receta	Proveedor
01	Platano Picado	Platano Picado medio asado	La 14
02	Yuca Costeña	Yuca asada	La 14
03	Papa	Papa	La 14
04	Ñame	Ñame	La 14
05	Queso parmesano	Queso parmesano	La 14

Figura 70 Diagrama Estético: Reportes Administrativos (Ingredientes)

### 1.7.31 Diagrama de estructura de comunicación: Reportes Administrativos (Reservaciones)

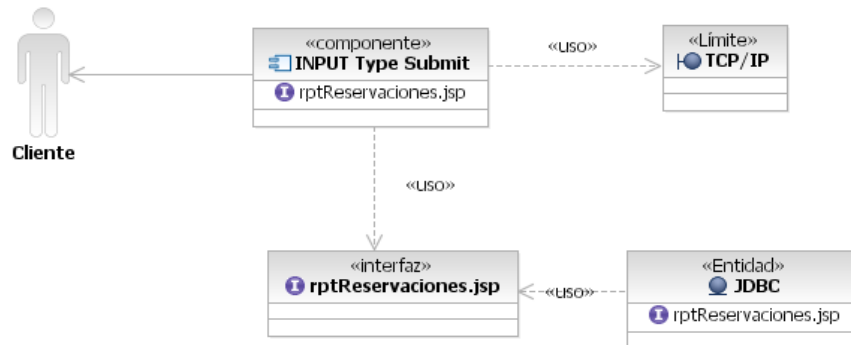


Figura 71 Diagrama de estructura de comunicación: Reportes Administrativos (Reservaciones)

### 1.7.32 Diagrama de estructura compuesta: Reportes Administrativos (Reservaciones)

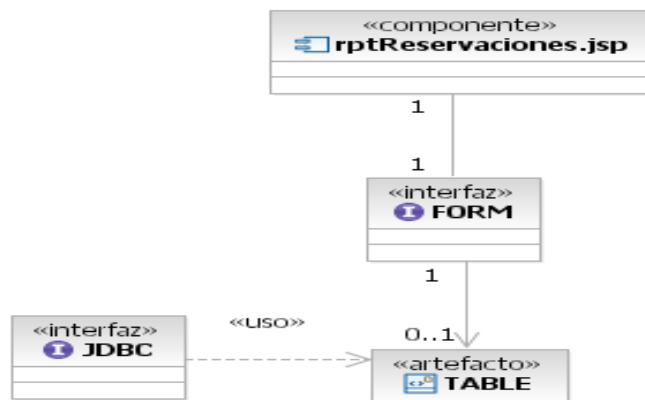


Figura 72 Diagrama de estructura compuesta: Reportes Administrativos (Reservaciones)

### 3.7.33. Diagrama Estético: Reportes Administrativos (Reservaciones)

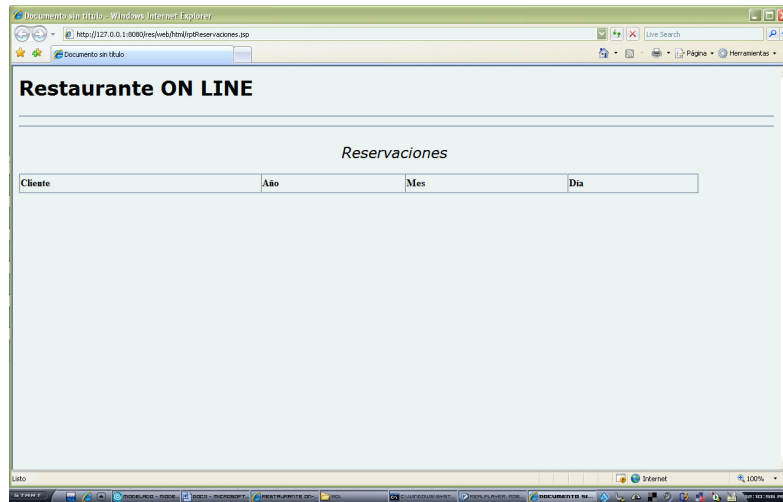


Figura 73 Diagrama Estético: Reportes Administrativos (Reservaciones)

### 1.7.34 Diagrama de estructura de comunicación: Reportes Estadísticos (Ventas por Sector)

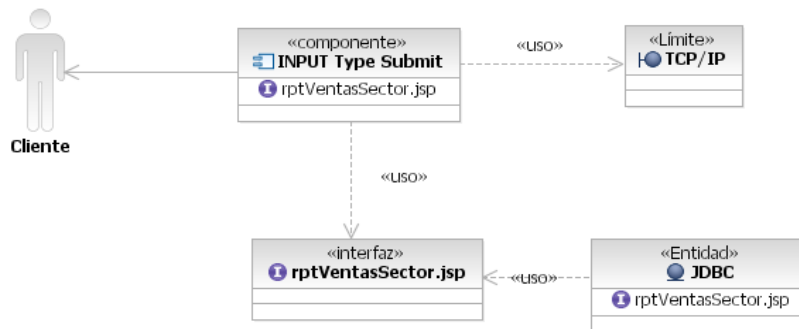


Figura 74 Diagrama de estructura de comunicación: Reportes Estadísticos (Ventas por Sector)

### 1.7.35 Diagrama de estructura compuesta: Reportes Estadísticos (Ventas por Sector)

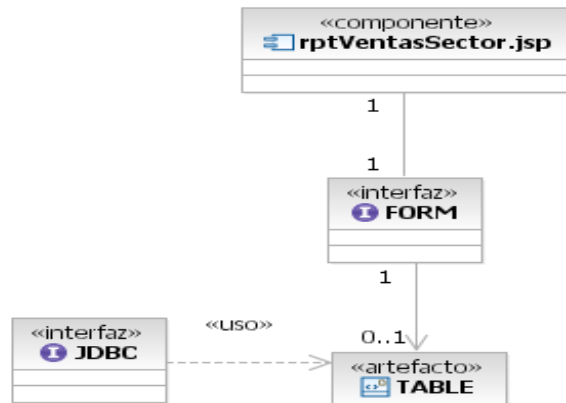


Figura 75 Diagrama de estructura compuesta: Reportes Estadísticos (Ventas por Sector)

### 1.7.36 Diagrama Estético: Reportes Estadísticos (Ventas por Sector) Parte A

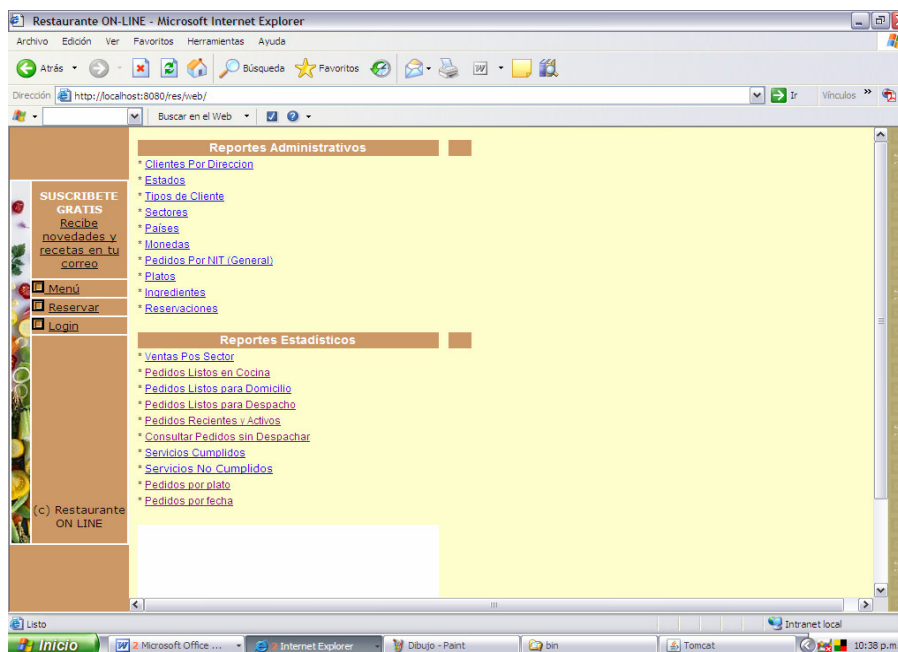


Figura 76 Diagrama Estético: Reportes Estadísticos (Ventas por Sector) Parte A

### 1.7.37 Diagrama Estético: Reportes Estadísticos (Ventas por Sector) Parte B

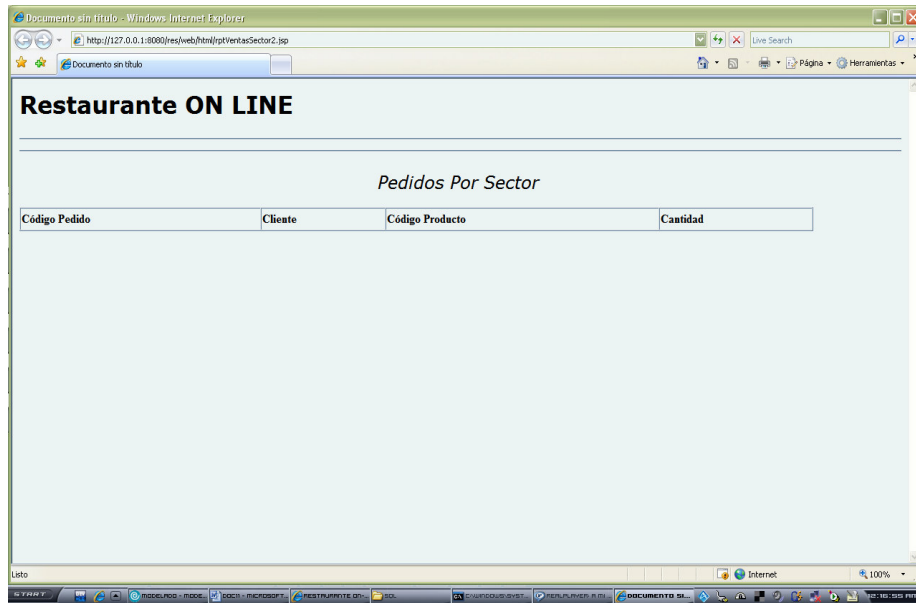


Figura 77 Diagrama Estético: Reportes Estadísticos (Ventas por Sector) Parte B

### 1.7.38 Diagrama de estructura de comunicación: Reportes Estadísticos (Pedidos Para Cocina)

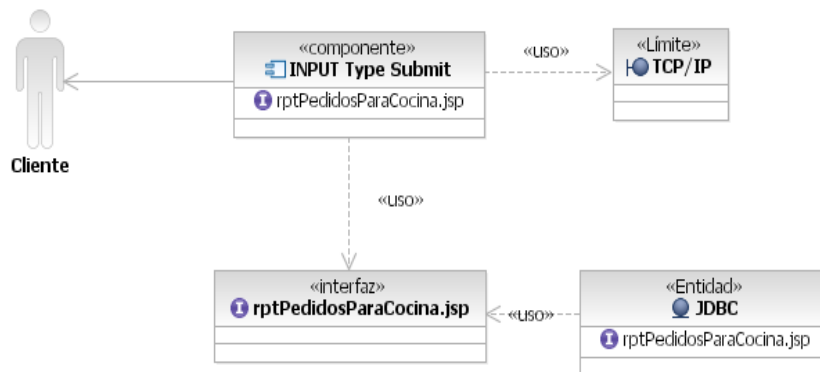


Figura 78 Diagrama de estructura de comunicación: Reportes Estadísticos (Pedidos Para Cocina)

### 1.7.39 Diagrama de estructura compuesta: Reportes Estadísticos (Pedidos Para Cocina)

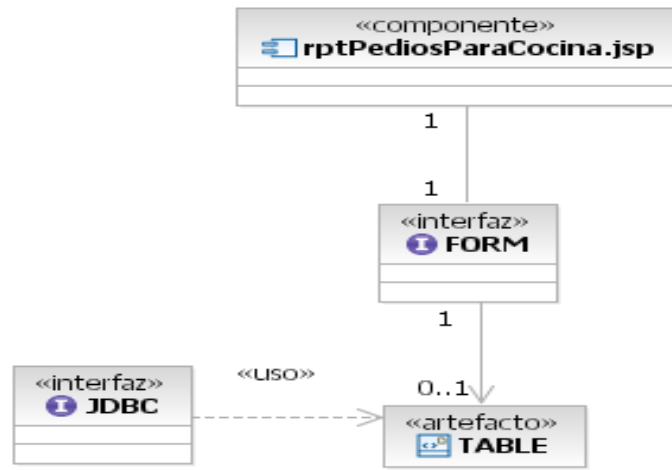


Figura 79 Diagrama de estructura compuesta: Reportes Estadísticos (Pedidos Para Cocina)

### 1.7.40 Diagrama Estético: Reportes Estadísticos (Pedidos Para Cocina)

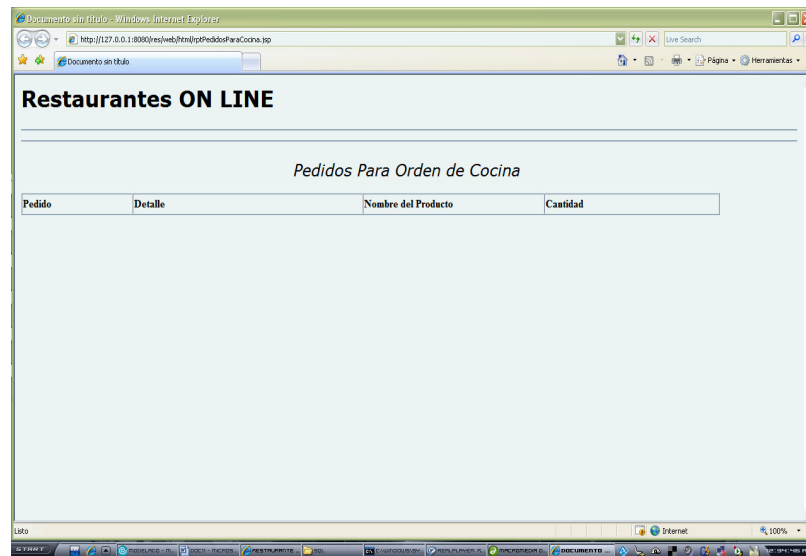


Figura 80 Diagrama Estético: Reportes Estadísticos (Pedidos Para Cocina)



### 1.7.41 Diagrama de estructura de comunicación: Reportes Estadísticos (Pedidos Para Domicilio)

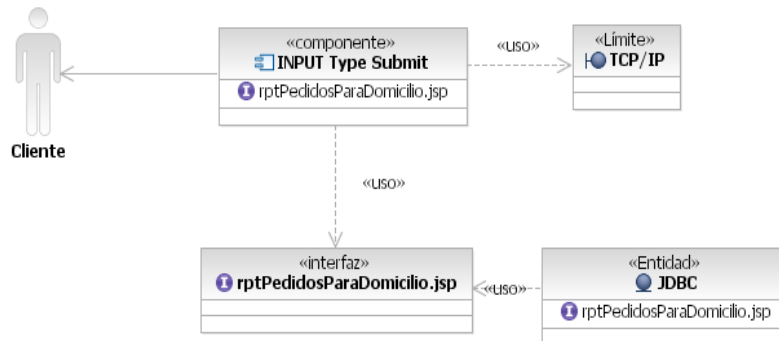


Figura 81 Diagrama de estructura de comunicación: Reportes Estadísticos (Pedidos Para Cocina)

### 1.7.42 Diagrama de estructura compuesta: Reportes Estadísticos (Pedidos Para Domicilio)

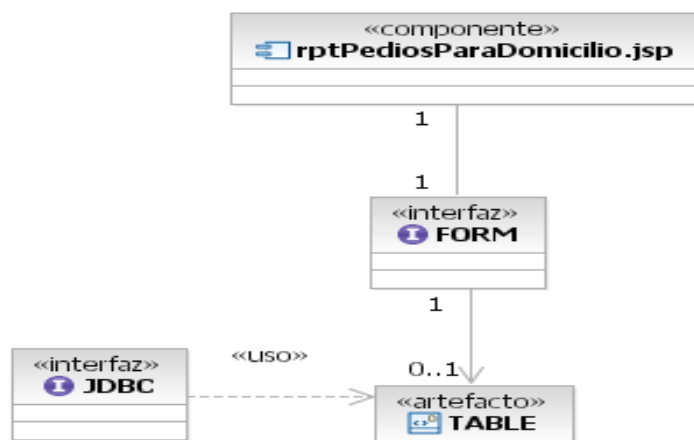


Figura 82 Diagrama de estructura compuesta: Reportes Estadísticos (Pedidos Para Cocina)

### 1.7.43 Diagrama Estético: Reportes Estadísticos (Pedidos Para Domicilio)



Figura 83 Diagrama Estético: Reportes Estadísticos (Pedidos Para Domicilio)

### 1.7.44 Diagrama de estructura de comunicación: Reportes Estadísticos (Pedidos Para Despacho)

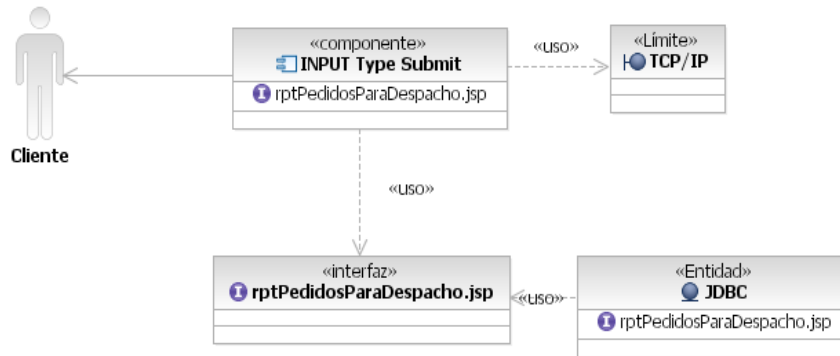


Figura 84 Diagrama de estructura de comunicación: Reportes Estadísticos (Pedidos Para Despacho)

### 1.7.45 Diagrama de estructura compuesta: Reportes Estadísticos (Pedidos Para Despacho)

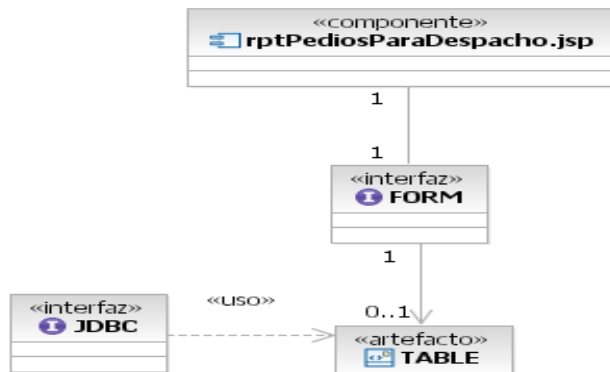


Figura 85 Diagrama de estructura compuesta: Reportes Estadísticos (Pedidos Para Despacho)

### 1.7.46 Diagrama Estético: Reportes Estadísticos (Pedidos Para Despacho)

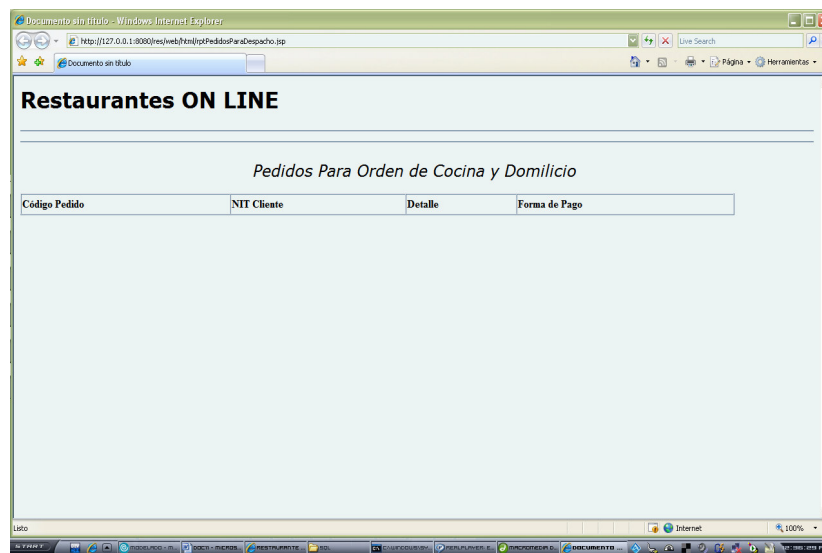


Figura 86 Diagrama Estético: Reportes Estadísticos (Pedidos Para Despacho)

### 1.7.47 Diagrama de estructura de comunicación: Reportes Estadísticos (Pedidos Activos)

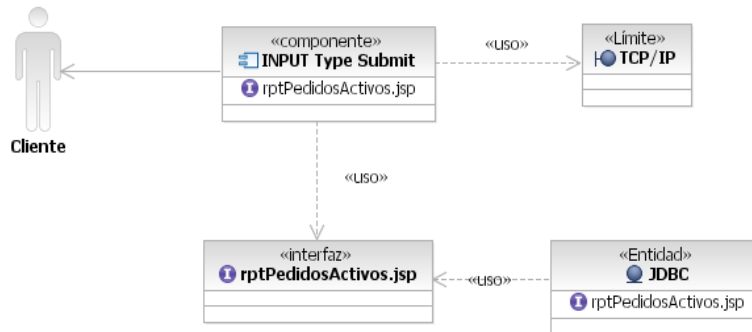


Figura 87 Diagrama de estructura de comunicación: Reportes Estadísticos (Pedidos Activos)

### 1.7.48 Diagrama de estructura compuesta: Reportes Estadísticos (Pedidos Activos)

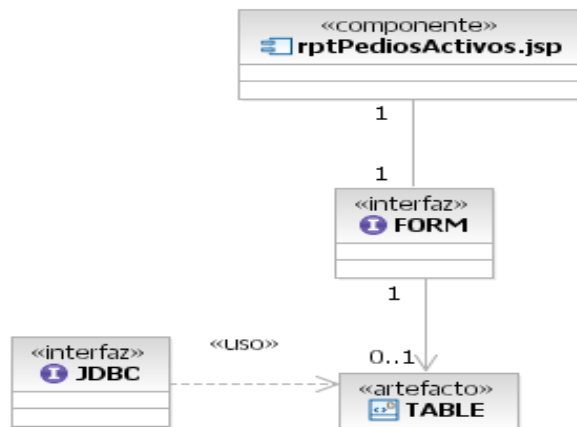


Figura 88 Diagrama de estructura compuesta: Reportes Estadísticos (Pedidos Activos)

### 3.7.49. Diagrama Estético: Reportes Estadísticos (Pedidos Activos)

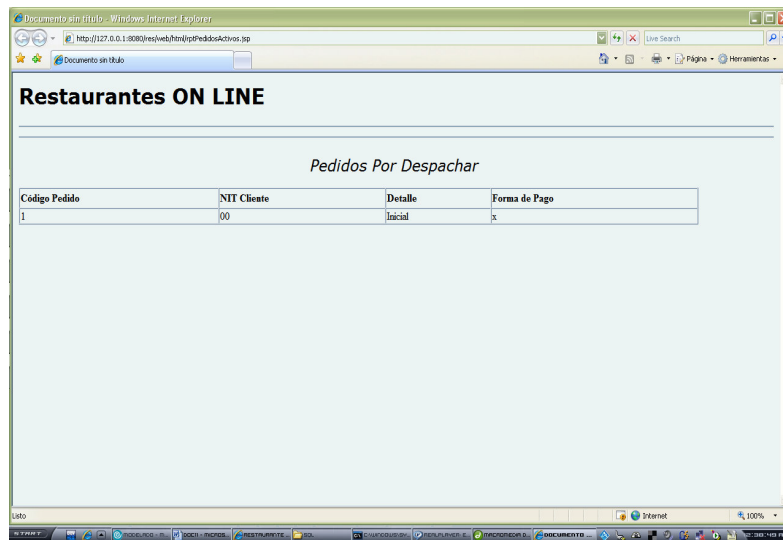


Figura 89 Diagrama Estético: Reportes Estadísticos (Pedidos Activos)

## 1.8 SUB-SISTEMA: PEDIDOS

El diseño de los pedidos está organizado de tal forma que atiendan los requerimientos funcionales del beneficiario. Donde se debe adaptar a la capacidad de producción del negocio; por tal motivo se diseña una interfaz que permita hacer hasta 5 pedidos por cliente en determinada transacción.

### 1.8.1 Diagrama de estructura de comunicación: Toma de pedidos

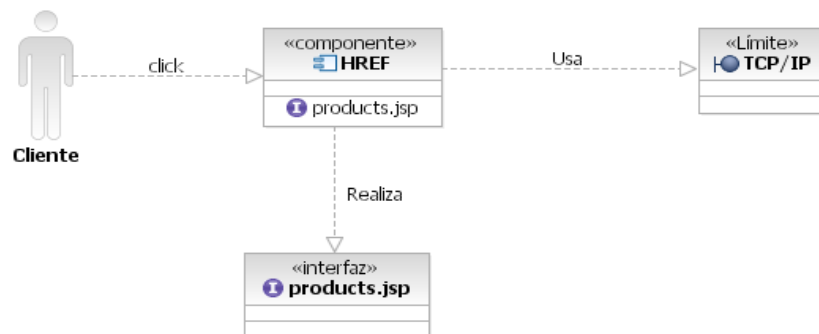


Figura 90 Diagrama de estructura de comunicación: Toma de pedidos

## 1.8.2 Diagrama de estructura compuesta Toma de pedidos

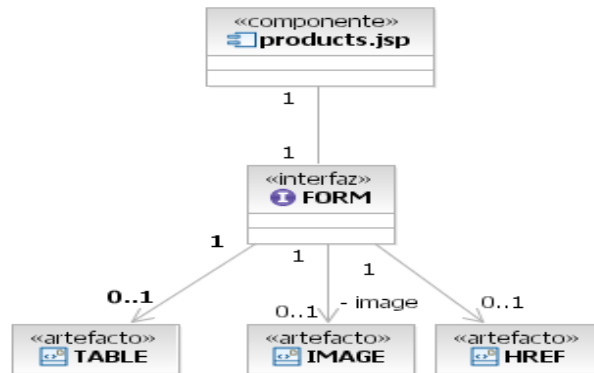


Figura 91 Diagrama de estructura compuesta Toma de pedidos

## 1.8.3 Diagrama Estético: Toma de pedidos

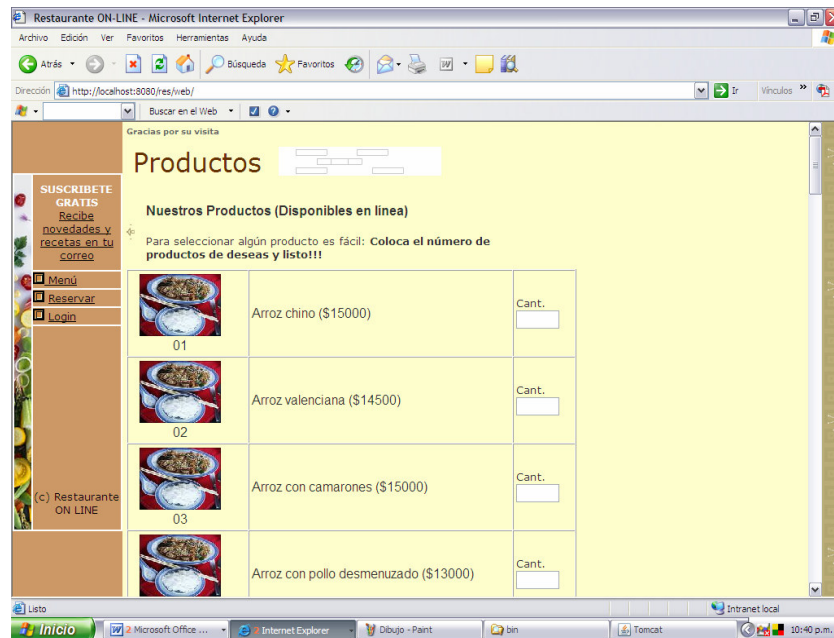


Figura 92 Diagrama Estético: Toma de pedidos

### 1.8.4 Diagrama de estructura de comunicación: Toma de pedidos (LOGIN)

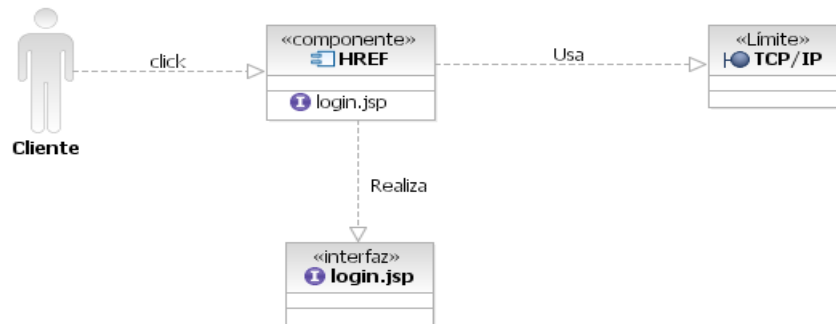


Figura 93 Diagrama de estructura de comunicación: Toma de pedidos (LOGIN)

### 1.8.5 Diagrama de estructura compuesta Toma de pedidos (LOGIN)

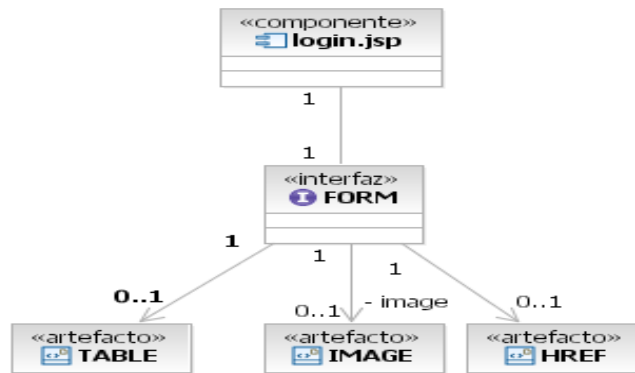


Figura 94 Diagrama de estructura compuesta Toma de pedidos (LOGIN)

### 1.8.6 Diagrama Estético: Toma de pedidos (LOGIN)

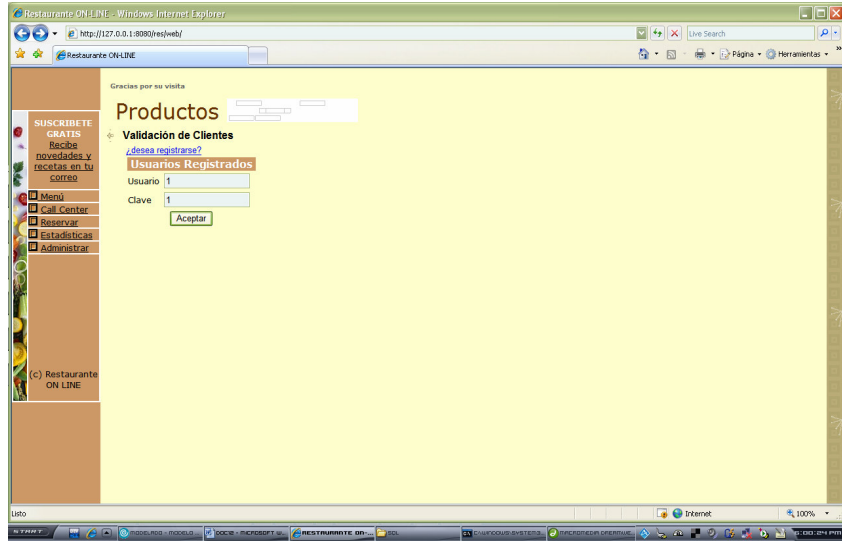


Figura 95 Diagrama Estético: Toma de pedios(LOGIN)

### 1.8.7 Diagrama de estructura de comunicación: Toma de pedidos (registro)

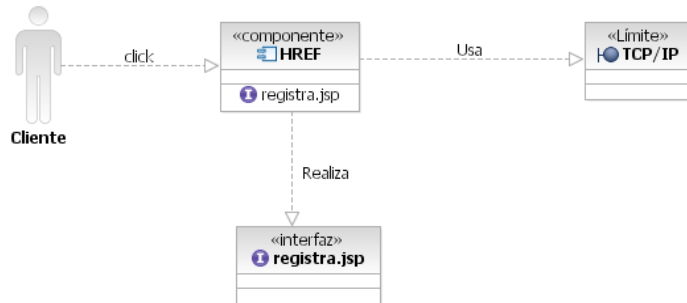


Figura 96 Diagrama de estructura de comunicación: Toma de pedidos (registro)



### 1.8.8 Diagrama de estructura compuesta Toma de pedidos (registro)

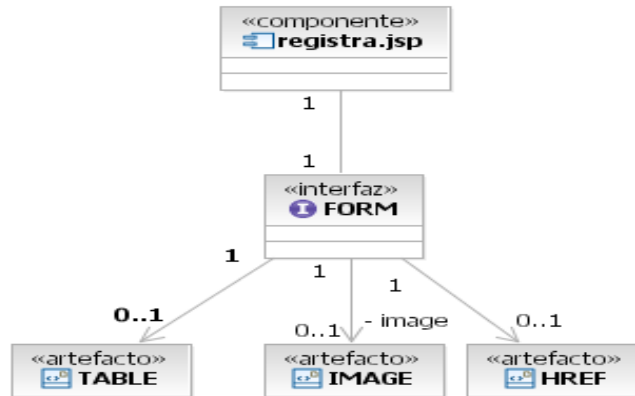


Figura 97 Diagrama de estructura compuesta Toma de pedidos (registro)

### 1.8.9 Diagrama Estético: Toma de pedidos (registro)

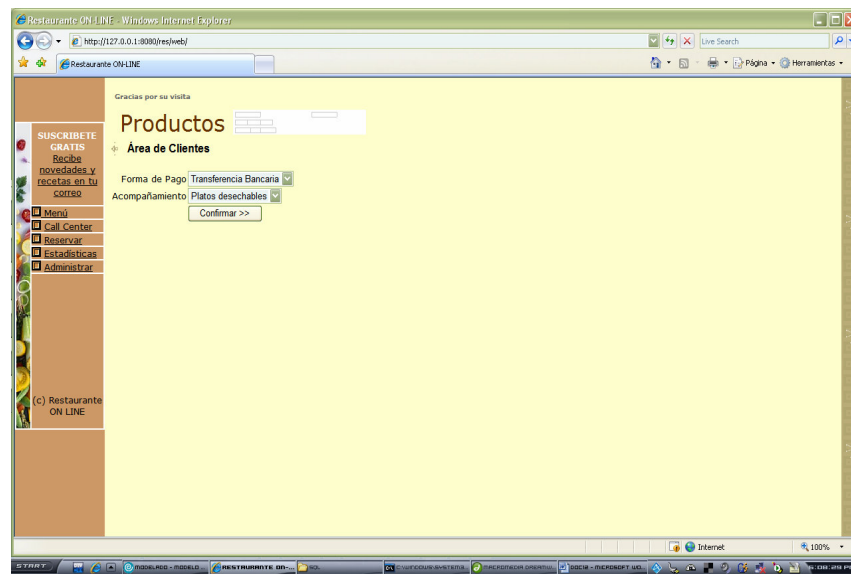


Figura 98 Diagrama Estético: Reportes Estadísticos Toma de pedidos (registro)

## 1.9 SUB-SISTEMA: PERFILES

El diseño de los perfiles están estipulados de tal forma que se pueda reconfigurar en determinado momento, en caso de estar desactualizados o en caso de atender a la necesidad de cambio el en los registros automáticos.

### 1.9.1 Diagrama de estructura de comunicación: Perfiles (LOGIN)

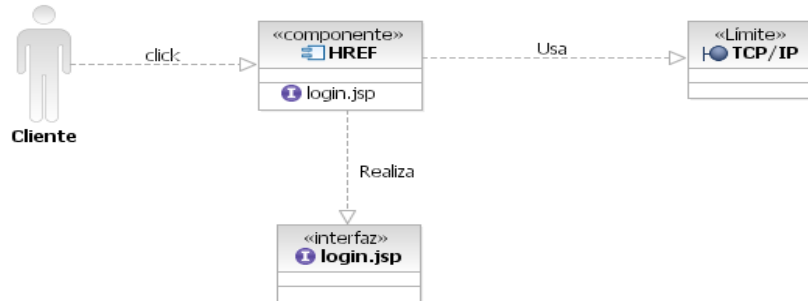


Figura 99 Diagrama de estructura de comunicación: Perfiles (LOGIN)

### 1.9.1 Diagrama de estructura compuesta Perfiles (LOGIN)

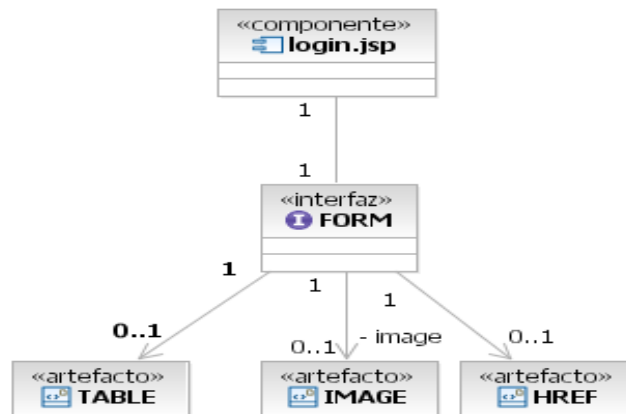


Figura 100 Diagrama de estructura compuesta Perfiles (LOGIN)

### 1.9.3 Diagrama Estético: Perfiles (LOGIN)

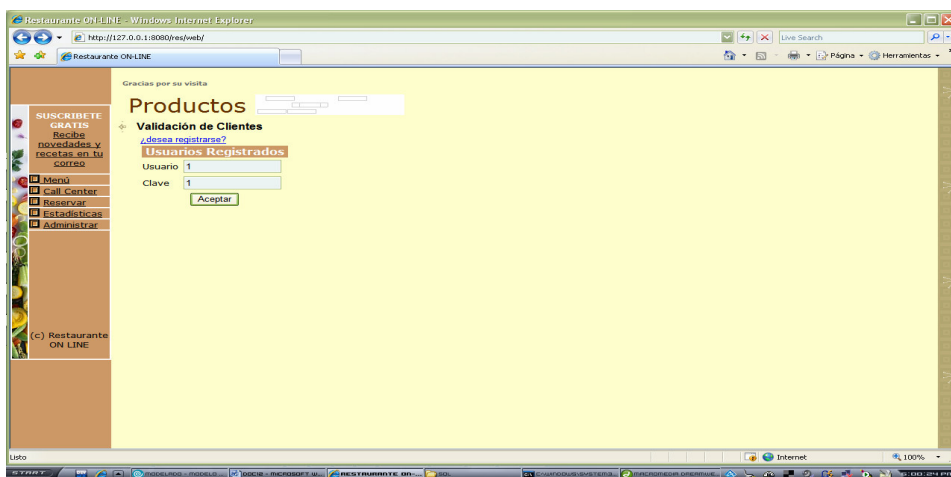


Figura 101 Diagrama Estético: Perfiles (LOGIN)

### 1.9.4 Diagrama de estructura de comunicación: Perfiles (MENU)

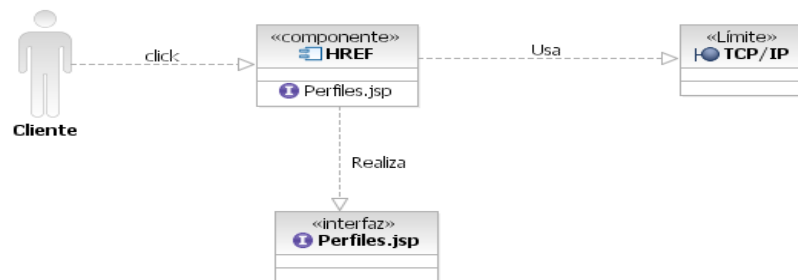


Figura 1002 Diagrama de estructura de comunicación: Perfiles (MENU)

### 1.9.5 Diagrama de estructura compuesta Perfiles (MENU)

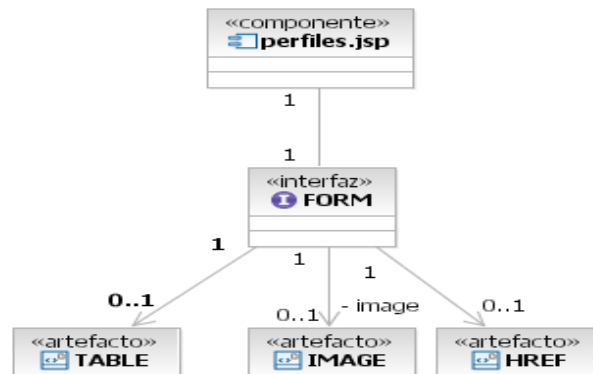


Figura 103 Diagrama de estructura compuesta Perfiles (MENU)

### 3.9.6. Diagrama Estético: Perfiles (MENU)

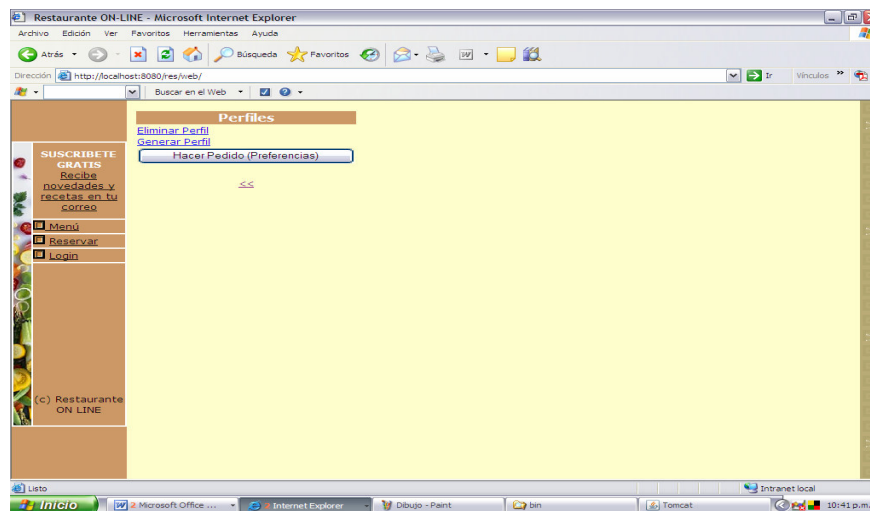


Figura 104 Diagrama Estético : Perfiles (MENU)

### 1.9.7 Diagrama de estructura de comunicación: Generar Perfil

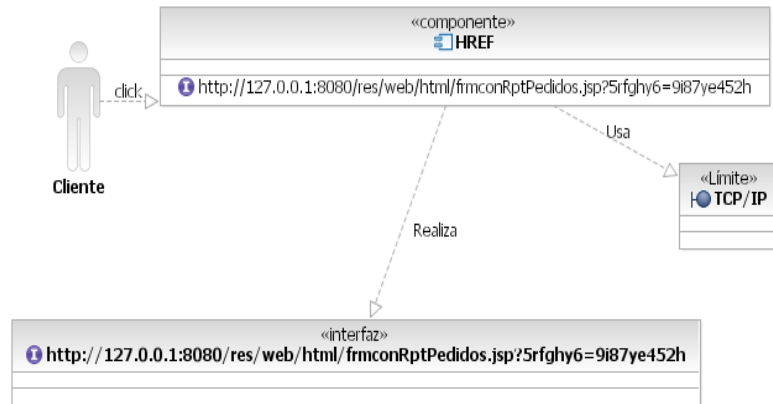


Figura 105 Diagrama de estructura de comunicación: Generar Perfil

### 1.9.8 Diagrama de estructura compuesta Generar Perfil

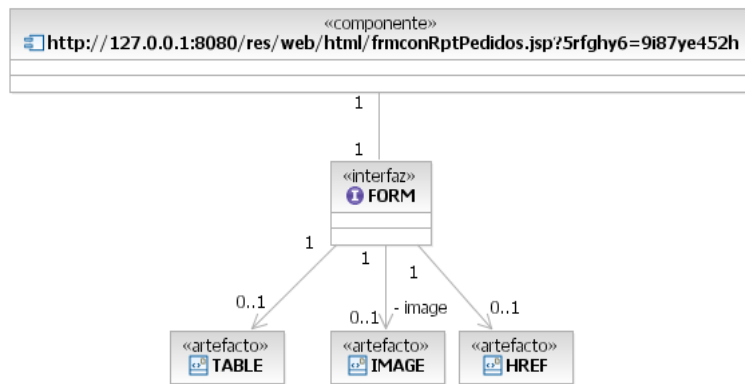


Figura 106 Diagrama de estructura compuesta Generar Perfil

### 1.9.9 Diagrama Estético: Generar Perfil

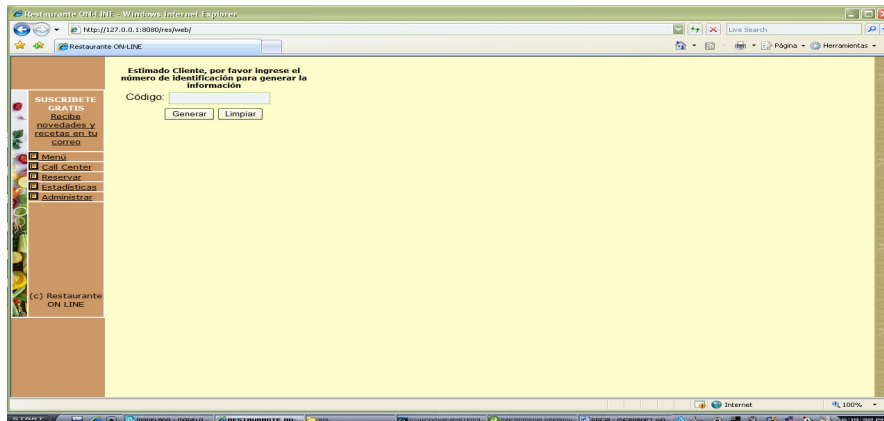


Figura 107 Diagrama Estético: Reportes Estadísticos Generar Perfil

### 1.10 SUB-SISTEMA: CONSULTAS EN PEDIDOS Y ESTADO DE PEDIDOS

El diseño de los perfiles están estipulados de tal forma que se pueda reconfigurar en determinado momento, en caso de estar desactualizados o en caso de atender a la necesidad de cambio en los registros automáticos.

#### 1.10.1 Diagrama de estructura de comunicación: de pedidos

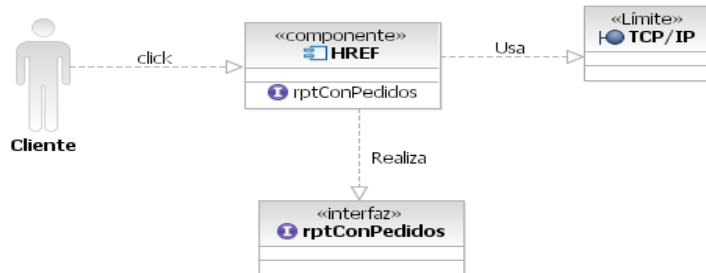


Figura 108 Diagrama de estructura de comunicación: de pedidos

#### 1.10.2 Diagrama de estructura compuesta: de pedidos

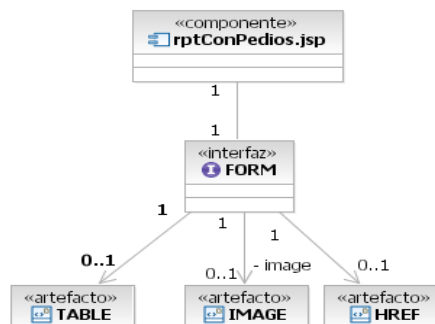


Figura 109 Diagrama de estructura compuesta: de pedidos

### 1.10.3 Diagrama Estético: Reportes Estadísticos: de pedidos (Parte A)

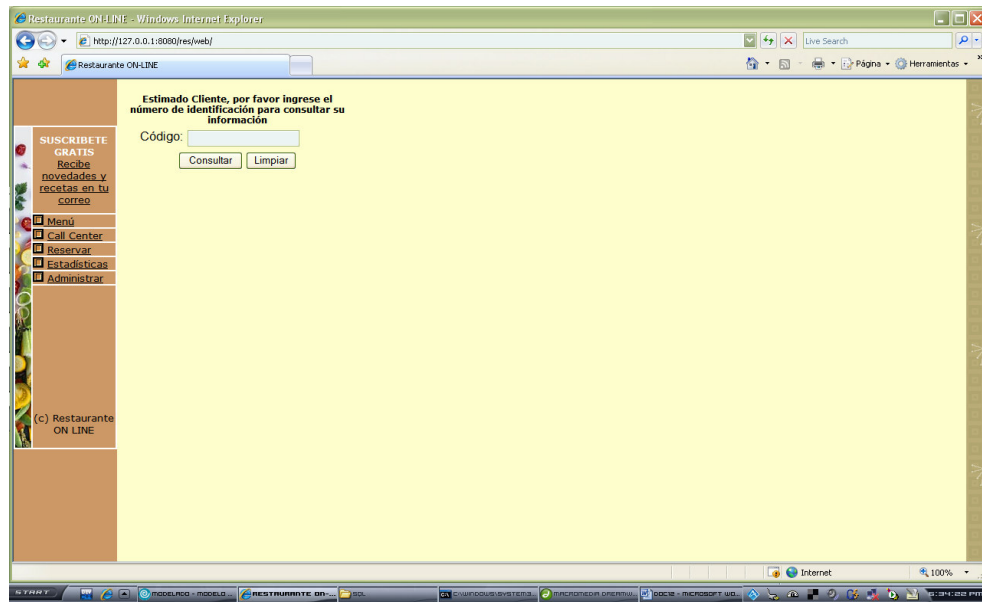


Figura 110 Diagrama Estético: Reportes Estadísticos: de pedidos (Parte A)

### 1.10.4 Diagrama Estético: Reportes Estadísticos: de pedidos (Parte B)



Figura 111 Diagrama Estético: Reportes Estadísticos: de pedidos (Parte B)

## ANEXO D

### MODELO DE PRUEBAS

#### 1. ARTEFACTO MODELO DE PRUEBAS

En software, el término prueba describe el proceso de validar cambios de código antes de que los cambios se registren en el árbol de origen del producto. Después de las revisiones de código, las pruebas son el método más rentable para identificar y corregir defectos en el software. Las pruebas están diseñadas para confirmar que los cambios en el código funcionan como se espera y no desestabilizan una versión completa. Entre los objetivos principales se encuentran:

- Encontrar defectos en el software
- Una prueba tiene éxito si descubre un defecto
- Una prueba fracasa si hay defectos pero no los descubre

Para lograr tales objetivos, las instrucciones siguientes describen los procedimientos recomendados para las pruebas. Los efectos de seguir estas directrices varían considerablemente, desde una mejora de la comunicación entre miembros del equipo hasta el desarrollo de formas específicas de usar las herramientas de prueba y depuración.

#### **Trabajar con el desarrollador**

Dado que las pruebas se centran en el código modificado, se debe trabajar con el desarrollador que escribió el código. Es necesario comprender:

- Qué ha cambiado en el código. Para comprender el cambio, también tendrá que comprender la tecnología utilizada; el desarrollador puede ayudar a explicarlo.
- Cómo afecta el cambio a la funcionalidad.
- Cómo afecta el cambio a las dependencias entre diversos componentes.

#### **Realizar una revisión del código antes de la prueba.**

Antes de ejecutar una prueba, se realiza una revisión del código centrada en todos los cambios en el código. Las revisiones de código son el método más efectivo y eficaz para validar la calidad del código y asegurarse contra defectos de código y

errores en la ejecución. Las pruebas garantizan que se valide en primer lugar el área débil o crítica principal identificada mediante revisión del código o evaluación de riesgos, puesto que si falla, la prueba no puede continuar.

### **Instalar binarios privados en una versión de depuración limpia**

Dado que una prueba se debe centrar en validar únicamente los cambios funcionales en binarios actualizados, la prueba se debe ejecutar en un entorno de prueba limpio, utilizando los binarios de depuración para los archivos que se están probando.

### **Pruebas Web y de carga**

Cuando se generan pruebas Web y pruebas de carga, es recomendable ejecutar una prueba general antes de ejecutar otra prueba larga y pesada. En las pruebas Web y de carga, las pruebas generales son breves y ligeras. Dichas pruebas se utilizan para validar que todo está configurado correctamente y que funciona como se espera antes de ejecutar las pruebas de rendimiento o exceso de carga.

Una prueba Web, también llamada prueba Web declarativa, está compuesta por una serie de solicitudes HTTP. Las pruebas Web funcionan en la capa de protocolo emitiendo solicitudes HTTP. Las pruebas Web no ejecutan JavaScript. Sin embargo, puede simular acciones de JavaScript en tiempo de ejecución utilizando complementos de prueba Web, complementos de solicitud de prueba Web, reglas de extracción, o pruebas Web codificadas\*.

#### **1.1. Documentación de pruebas**

Hay que tener en cuenta dos puntos importantes, en primer lugar las pruebas de caja blanca no reemplazan, solo complementan a las de caja negra y de aceptación, y en segundo lugar, las pruebas de cobertura deben ser realizadas una vez terminado el software y no deben ser confundidas con las pruebas informales que realiza el programador en momentos de desarrollo, dado que si bien estas van cubriendo distintos fragmentos de cada módulo, nunca son eficaces por no tener un diseño apropiado.

El valor porcentual de pruebas de cobertura de un sistema terminado nunca deberá ser inferior al 51%, y elevándose en función al coste que podría ocasionar las fallas posibles, ascendiendo a un 99% cuando estén involucradas vidas humanas o cuando la falla no da una segunda oportunidad.

---

\* Basado en la metodología trabajada con **Visual Studio Team System**



Según los párrafos anteriores documentamos en términos generales los siguientes hallazgos:

<b>Modulo</b>	<b>Descripción</b>	<b>Hallazgo</b>
<b>Platos</b>	Módulo para la administración de platos	Los platos deberían estar clasificados: Arroces, Verduras, Carnes, Platos especiales, Sopas, Tortillas, Pollos
<b>Pedido</b>	Módulo para el ingreso de pedidos por Internet o desde el CALL CENTER	No hay opción para pedir acompañamientos y otros: salsas, ensaladas, porciones de arroz o papa, gaseosa, platos desechables, entre otros
<b>Pedido</b>	Módulo para el ingreso de pedidos por Internet o desde el CALL CENTER	ERROR en la página para hacer pedido de un producto, permite al usuario modificar / borrar los campos. El único campo habilitado para el usuario debería ser la cantidad a pedir
<b>Clientes</b>	Módulo para la administración de clientes	Para el registro de clientes no todos tienen NIT o no a todos les gusta dejar su número sin saber para que se va a usar.
<b>Clientes</b>	Módulo para la administración de clientes	ERROR en el momento de registro
<b>Perfiles</b>	Módulo para la administración de perfiles	No se encuentra activo
<b>Pedido</b>	Módulo para el ingreso de pedidos por Internet	No hay forma de seleccionar el

	o desde el CALL CENTER	restaurante al que el cliente desea hacer los pedidos
<b>Reservaciones</b>	Módulo para administración reservaciones	la En el módulo de reservaciones, el campo Cliente hace referencia a información errada
<b>Restaurantes</b>	Módulo para administración restaurantes	la La tabla restaurantes debería tener otros campos como dirección y al menos un teléfono. Un usuario puede hacer el pedido por Internet pero puede pasar a recogerlo
<b>General</b>	Observación general	No se validan los ingresos pueden ingresar caracteres inválidos o números en campos de texto.
<b>General</b>	Observación general	En algunos archivos <b>jsp</b> cuenta con ingresos inválidos en los títulos

Figura 1 Diagrama Estético: Tabla de Pruebas

## ANEXO E

### ESTUDIO Y SELECCIÓN DE ALTERNATIVAS

A fin de determinar cual iba a ser la mejor alternativa se hizo un estudio de tres posibilidades que cumplieran en más o menos medida las necesidades de la empresa. Entre ellas estaba la de desarrollar este proyecto.

Los posibles nuevos sistemas de información podían ser desde aplicaciones estándar hasta la realización de software a medida.

Un desarrollo a medida con el tiempo y los recursos económicos necesarios, puede llegar a cubrir todos los requisitos demandados por el cliente, mientras que una solución estándar normalmente dejará algún requisito sin cubrir, o no cubierto de la manera deseada.

A continuación mostramos una tabla comparativa de las tres principales opciones para un sistema de información adecuado para la comercialización de productos alimenticios en Internet:

"S" - Si implementa.

"P" - Implementa parcialmente.

"N" - No implementa.

1.- Sistema de Información desarrollado a medida por los autores de este proyecto.

2.- Conta SQL 2000.

3.- E-Comerce.

<b>Requisito</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>Gestión de Pedidos de Platos</b>	S	P	S
Altas, bajas y modificaciones	S	N	S
Búsqueda por campos	S	N	S
Gestión de pedidos (albaranes, facturación)	S	P	S
Consulta de platos	S	N	S
Consulta de mis pedidos	S	N	S
<b>Gestión de envíos</b>	S	N	S
<b>Informes</b>	S	P	S

<b>Gestión de interconexión</b> (Red Area Local)	S	N	N
<b>Niveles de acceso según usuario</b>	S	S	N
<b>Adaptación a usuarios</b> (es lo que han pedido)	S	N	P
<b>Facilidad de uso</b>	S	S	S
<b>Modificabilidad</b>	S	N	N
<b>Ayuda en pantalla</b>	S	S	P
<b>Comerciales</b> (comisiones)	S	N	S
<b>Banca electrónica</b> (Internet)	N	N	S

Notas: Las opciones 2 y 3 cubren áreas de la empresa que no se desean contemplar en este proyecto, como los asientos de los pedidos. Por tanto, no son tan interesantes como un proyecto a medida. La opción 3 necesita una parametrización inicial muy complicada.

En conclusión, es más conveniente un desarrollo a medida para la empresa. Las demás opciones son más complicadas y voluminosas. Además, excepto el desarrollo a medida, el resto de opciones no podrían ser modificadas ni ampliadas en un futuro.

## ANEXO F

### MEDICIÓN Y ESTIMACIÓN

#### Introducción de los Puntos de Función

Para la medición del proyecto que se pretende desarrollar utilizaremos la técnica de los Puntos de Función, debido a que es la más extendida en la actualidad y tiende a convertirse en un estándar de facto. Además, es la más conocida por los desarrolladores de entre las distintas técnicas que existen, y consideramos que de esta manera la medición será más fiable.

Esta técnica de medición y estimación trata de evaluar una aplicación informática con base a sus características externas. Estas características se descomponen en dos grupos:

- Los Elementos de Función: elementos como formularios de entrada, salida, consultas y archivos a los que debe dar soporte la aplicación.
- Los Factores de Complejidad: indicadores del entorno en el que se ha de desarrollar y explotar la aplicación informática.

Este método de estimación contempla la aplicación a desarrollar como una caja negra, es decir, no se interesa por los factores internos de la aplicación, sino que se centra en lo que puede ver el usuario.

Por otra parte, evaluamos de forma explícita los factores de desarrollo que influyen sobre la productividad, como lenguajes de desarrollo, entornos de trabajo, etc.

Seguiremos manteniendo la idea de caja negra, ya que los gestores de desarrollo no están interesados en como se desarrolla en cada lenguaje, sino que se centran en los diferentes niveles de productividad que se tiene con cada uno de estos.

#### 1. Identificación de los Elementos de Función

Se parte de la especificación de la aplicación a desarrollar y el Diagrama General de Casos de Uso de dicha aplicación s que utilizará.

Entendemos por elementos de función los siguientes componentes de la aplicación:

- Entradas desde el exterior del sistema: pantallas de entrada de datos y otros tipos de entradas a través de periféricos

- Salidas al exterior: pantallas, listados, publicación WEB etc.
- Consultas: entrada seguida de una salida
- Archivos lógicos internos: grupos de datos que se mantienen internamente.
- Archivos de Interfaz: grupos de datos que se mantienen externamente

## 2. Cálculo de los Puntos de Función sin Ajustar

Sumando los elementos de función y teniendo en cuenta su peso como primera parte del cálculo del esfuerzo, se obtienen los Puntos de Función sin Ajustar que aparecen en la tabla siguiente:

Tipo De Elemento	Dificultad	Peso	Cantidad	Total Puntos	Total Elemento
Entradas	Simple	3	9	27	
	Media	4	2	8	
	Compleja	6			
	<b>Total Puntos de Función Entradas:</b>				<b>35</b>
Salidas	Simple	4	4	16	
	Media	5	1	5	
	Compleja	7			
	<b>Total Puntos de Función Salidas:</b>				<b>21</b>
Consultas	Sal Simple	4			
	Ent Media	4	9	36	
	Sal Media	5			
	Ent Compleja	6			
	Sal Compleja	7			
	<b>Total Puntos de Función Consultas:</b>				<b>36</b>
Arch. Internos*	Simple	7	5	35	
	Media	10	2	20	
	Compleja	15			
	<b>Total Puntos de Función Archivos Internos:</b>				<b>55</b>
Arch.	Simple	5			

\* Número de archivos lógicos internos (ALI). Cada *archivo lógico Interno* es un agrupamiento lógico de datos que reside dentro de los límites de las aplicaciones y que se mantiene mediante entradas externas.

Interfaces*	Media	7			
	Compleja	10			
	<b>Total Puntos de Función Archivos Interfaces:</b>				<b>0</b>
<b>Conteo Total (PFSA)</b>		<b>Total Puntos de Función sin ajustar</b>			<b>147</b>

### 3 Cálculo del Factor de Complejidad Total

En esta fase mediremos las características externas de la aplicación basándonos en el entorno en el que la aplicación se implantará. Valoraremos una serie de factores de complejidad evaluándolos en una escala de 0 al 5.

	<b>Factor de Complejidad</b>	<b>Descripción del Valor</b>	<b>Valor</b>
1	Comunicación de Datos	Sistema Aislado	0
2	Proceso Distribuido	Sistema Totalmente Centralizado	0
3	Rendimiento	Rendimiento estudiado durante el diseño, pero sin especiales requerimientos	1
4	Configuración Operacional Compartida	Existen las restricciones usuales	1
5	Ratio de Transacciones	Se prevén horas puntas diarias	3
6	Entrada de Datos EN-LINEA	Las entradas de datos interactivas superan el 30 %	5
7	Eficiencia con el Usuario Final	Seis o más factores	3
8	Actualizaciones en Línea	Actualización en línea de archivos lógicos importantes	3
9	Complejidad del Proceso Interno	La aplicación llevará incorporados sistemas de seguridad y control	1
10	Reusabilidad del Código	Se pretende reutilizar entre 40% y el 50 %	4
11	Contempla la Conversión de dólares a pesos. Instalación	Se ha solicitado Facilidad de instalación	1
12	Facilidad de Operación	Se proveerá de procesos de arranque,	1

\* Ibid. Número de archivos de interfaz externos (AIE). Cada archivo de *Interfaz externo* es un agrupamiento lógico de datos externo a la aplicación pero que proporciona datos que podrían usarse en ésta.

	<b>Factor de Complejidad</b>	<b>Descripción del Valor</b>	<b>Valor</b>
		back-up y recuperación, pero con operador	
<b>13</b>	Instalaciones Múltiples	Un solo lugar	0
<b>14</b>	Facilidad de Cambios	El Sistema puede tener cambios medios	3
<b>Factor De Complejidad Total (FCT) = <math>[\sum(F_i)]</math></b>			<b>26</b>

#### **4 Obtención de los Puntos de Función Ajustados (PFA)**

$$PFA = PFSA * (0.65 + (0.01 * FCT))$$

Para su cálculo aplicamos la siguiente fórmula:

$$PF = \text{Conteo total} * [0.65 + 0.01 * \sum(F_i)]$$

En otros documentos la misma fórmula aparece expresada así:

Esta fórmula nos indicará que en principio cada factor de complejidad puede actuar sobre los PFSA incrementando o decrementando en un 2.5 % la cantidad de Puntos de Función Ajustados.

$$PFA = 147 * (0.65 + (0.01 * 26)) = 133.77$$

#### **5 Estimación del Esfuerzo Requerido por la Aplicación**

El objetivo ahora es estimar la cantidad de esfuerzo necesario para desarrollar la aplicación. Este esfuerzo se mide en horas/personas, meses/personas o años/personas.

Los Puntos de Función en cierto modo son una medida subjetiva, ya que se pueden realizar valoraciones diferentes por personas con diferentes puntos de vista. Además, en principio, este número no nos dice nada. Nos hace falta conocer la cantidad de meses/persona que costará cada Punto de Función.

La herramienta utilizada para el desarrollo de nuestra aplicación será Java - JSP.

En esta planificación realizaremos una aproximación de 5 Horas/Punto Función por ser Java - JSP una herramienta extendida en el mercado, de uso sencillo y conocida por los desarrolladores.

Utilizaremos la siguiente fórmula para calcular el esfuerzo:



$$\text{Esfuerzo} = \text{PFA} * \text{Promedio Organización (Lenguaje)}$$

Debemos tener en cuenta que en una jornada de 8 horas de trabajo sólo 5 son productivas, también que un mes tiene 20 días de trabajo y un año once meses. Con todo esto obtenemos que al año se trabaja alrededor de 220 días reales.

$$\begin{aligned} \text{Esfuerzo} &= (\text{PFA} * \text{Horas día}) = (133.77 * 5) = 668.85 \text{ Horas} \\ \text{Esfuerzo} &= (\text{Horas} / \text{Horas Día}) = (668.85 / 5) = 134 \text{ Días} \\ \text{Esfuerzo} &= (\text{Días} / \text{Días Mes}) = (134 / 20) = 6 \text{ Meses y 21 Días} \end{aligned}$$

Teniendo en cuenta el número de trabajadores, en nuestro caso dos, el proyecto podrá realizarse en algo menos de 3 meses y medio (unos 67 días aproximadamente).

De todas formas, pensando en los posibles retrasos y teniendo en cuenta que el cliente no tiene excesiva prisa en recibir el programa, realizaremos el proyecto en cuatro meses para así tener un pequeño margen de maniobra.

## 6. Presupuesto de la Aplicación

El costo de la aplicación a desarrollar se hace en función de las horas dedicadas en su implementación obtenidas de la planificación estimada, la aplicación se instala incluyendo como hardware un servidor central multiusuario.

Calcularemos el costo de la mano de obra en función de los diferentes salarios para analista, programadores, etc.

Actividad	Costo Hora
Análisis y Diseño	50.000
Programación	35.000
Formación	40.000
Documentación y Pruebas	35.000

### Presupuesto Software (en pesos)

- Análisis y especificación de requisitos.*

Planificación elemental de sistemas	10	h	*	50.000
500.000				

- Desarrollo de la aplicación.*

Análisis y especificación de requisitos	60 h * 50.000	3'000.000
	40 h * 35.000	1'400.000

Adaptación a nuevos entornos y aprendizaje	20 h * 50.000	1'000.000
	70 h * 50.000	3'500.000
Medición y estimación	65 h * 35.000	2'275.000
Diseño	25 h * 35.000	875.000
Implementaron parcial	20 h * 35.000	700.000
Pruebas	25 h * 40.000	1'000.000
Documentación		

Cursillo usuarios

**Total presupuesto software: 13'750.000**

**Nota: Este costo difiere del calculado en el estudio de pre-viabilidad (9'299.600 pesos) porque aquel era solamente una estimación aproximada.**

### Presupuesto Hardware

<b>Procesador</b>	Intel Pentium IV 3000 MHz
<b>Memoria</b>	1000 MB SDRAM PC 133
<b>Disco duro</b>	240 GB Ultra DMA
<b>Tarjeta de sonido</b>	SoundBlaster 128
<b>Tarjeta gráfica</b>	Aceleradora 3D, 32 MB/SDRAM
<b>Disquetera</b>	1,44 MB
<b>DVD R/W</b>	52x/32x
<b>Módem interno</b>	Fax-Módem interno 56K
<b>Altavoces</b>	Dolby Surround
<b>Monitor</b>	17"
<b>Micrófono</b>	De sobremesa
<b>Teclado</b>	Botones adicionales navegar por Internet
<b>Ratón</b>	Intellimouse de Microsoft
<b>Software</b>	Microsoft Windows XP

**Total presupuesto hardware: \$ 2'155.000**

## ANEXO G

### SCRIPT PARA LA CREACIÓN DE LA BASE DE DATOS

```
DROP DATABASE res;  
CREATE DATABASE res  
  WITH OWNER = postgres  
      ENCODING = 'LATIN1'  
      TABLESPACE = pg_default;
```

```
\c res;
```

```
CREATE TABLE tblPlatos
```

```
(  
  cRefPla varchar NOT NULL,  
  cDesPla varchar,  
  cDetPla varchar,  
  cRecPla varchar,  
  cTipPla varchar,  
  cPrePla varchar,  
  clmaPla varchar,  
  cCodRes varchar NOT NULL,  
  cMenPla varchar,  
  cEstPla varchar,  
  CONSTRAINT cPlatos PRIMARY KEY (cRefPla) USING INDEX TABLESPACE pg_default  
) WITHOUT OIDS  
TABLESPACE pg_default;  
ALTER TABLE tblPlatos OWNER TO postgres;  
COMMENT ON COLUMN tblPlatos.cRefPla IS 'Código del Plato';  
COMMENT ON COLUMN tblPlatos.cDesPla IS 'Descripcion del Plato';
```

```
CREATE TABLE tblIngredientes
```

```
(  
  cCodIng varchar NOT NULL,  
  cDesIng varchar,  
  cReclIng varchar,  
  cProIng varchar,  
  CONSTRAINT Ingredientes PRIMARY KEY (cCodIng) USING INDEX TABLESPACE pg_default  
) WITHOUT OIDS  
TABLESPACE pg_default;  
ALTER TABLE tblIngredientes OWNER TO postgres;  
COMMENT ON COLUMN tblIngredientes.cCodIng IS 'Código del Ingrediente';  
COMMENT ON COLUMN tblIngredientes.cDesIng IS 'De la Característica Ingrediente';
```

```
CREATE TABLE tblCaracteristicas
```

```
(  
  cRefPla varchar NOT NULL,  
  cCodIng varchar NOT NULL,
```

```

        CONSTRAINT Caracteristicas PRIMARY KEY (cRefPla, cCodIng) USING INDEX TABLESPACE
pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblCaracteristicas OWNER TO postgres;
COMMENT ON COLUMN tblCaracteristicas.cRefPla IS 'Código del Plato';
COMMENT ON COLUMN tblCaracteristicas.cCodIng IS 'Código del Ingrediente';

```

```

CREATE TABLE tblClientes
(
    cod serial NOT NULL,
    idCliente varchar NOT NULL,
    Nombre varchar NOT NULL,
    Direccion varchar NOT NULL,
    Telefono varchar NOT NULL,
    Correo varchar NOT NULL,
    Moneda varchar NOT NULL,
    Pais varchar NOT NULL,
    idTipoCliente varchar NOT NULL,
    idSector varchar NOT NULL,
    Clave varchar NOT NULL,
    Privilegio varchar NOT NULL,
    CONSTRAINT IDClientes PRIMARY KEY (cod, idCliente) USING INDEX TABLESPACE
pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblClientes OWNER TO postgres;

```

```

CREATE TABLE tblMonedas
(
    idMoneda varchar NOT NULL,
    Descripcion varchar NOT NULL,
    CONSTRAINT IDMonedas PRIMARY KEY (idMoneda) USING INDEX TABLESPACE pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblMonedas OWNER TO postgres;

```

```

CREATE TABLE tblPaises
(
    idPais varchar NOT NULL,
    Descripcion varchar NOT NULL,
    CONSTRAINT IDPaises PRIMARY KEY (idPais) USING INDEX TABLESPACE pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblPaises OWNER TO postgres;

```

```

CREATE TABLE tblSectores
(
    idSector varchar NOT NULL,
    Descripcion varchar NOT NULL,
    CONSTRAINT IDSector PRIMARY KEY (idSector) USING INDEX TABLESPACE pg_default
) WITHOUT OIDS
TABLESPACE pg_default;

```

```

ALTER TABLE tblSectores OWNER TO postgres;

CREATE TABLE tblEstados
(
  idEstado varchar NOT NULL,
  Descripcion varchar NOT NULL,
  CONSTRAINT IDEstados PRIMARY KEY (idEstado) USING INDEX TABLESPACE pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblEstados OWNER TO postgres;

CREATE TABLE tblTipoClientes
(
  idTipoCliente varchar NOT NULL,
  Descripcion varchar NOT NULL,
  CONSTRAINT IDTipoClientes PRIMARY KEY (idTipoCliente) USING INDEX TABLESPACE
pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblTipoClientes OWNER TO postgres;

CREATE TABLE tblTipoPlatos
(
  idTipoPlato varchar NOT NULL,
  Descripcion varchar NOT NULL,
  CONSTRAINT IDTipoPlatos PRIMARY KEY (idTipoPlato) USING INDEX TABLESPACE
pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblTipoPlatos OWNER TO postgres;

CREATE TABLE tblReservaciones
(
  idReservacion serial NOT NULL,
  idRestaurante varchar NOT NULL,
  idCliente varchar NOT NULL,
  FechaRegistro date NOT NULL,
  Año varchar NOT NULL,
  Mes varchar NOT NULL,
  Dia varchar NOT NULL,
  Hora varchar NOT NULL,
  Mesa varchar,
  Observacion varchar,
  Estado varchar,
  CONSTRAINT IDReservaciones PRIMARY KEY (idReservacion) USING INDEX TABLESPACE
pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblReservaciones OWNER TO postgres;

CREATE TABLE tblRestaurantes
(
  idRestaurante varchar NOT NULL,

```

```

    Descripcion varchar NOT NULL,
    Direccion varchar NOT NULL,
    Telefono varchar NOT NULL,
    Fax varchar NOT NULL,
    Correeo varchar NOT NULL,

    CONSTRAINT IDRestaurante PRIMARY KEY (idRestaurante) USING INDEX TABLESPACE
pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblRestaurantes OWNER TO postgres;

CREATE TABLE tblAcompañantes
(
    idAcompañante varchar NOT NULL,
    Descripcion varchar NOT NULL,
    CONSTRAINT Acompañante PRIMARY KEY (idAcompañante) USING INDEX TABLESPACE
pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblAcompañantes OWNER TO postgres;

CREATE TABLE tblPerfiles
(
    idPerfil varchar NOT NULL,
    idCliente varchar NOT NULL,
    idItem varchar NOT NULL,
    idProducto varchar NOT NULL,

    CONSTRAINT IDPerfiles PRIMARY KEY (idPerfil, idItem) USING INDEX TABLESPACE pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblPerfiles OWNER TO postgres;

CREATE TABLE tblPedidos
(
    idPedido int NOT NULL,
    idCliente varchar NOT NULL,
    Detalle varchar NOT NULL,
    FormaPago varchar NOT NULL,
    Fecha date NOT NULL,
    Hora time NOT NULL,
    idEstado varchar NOT NULL,
    CONSTRAINT IDPedido PRIMARY KEY (idPedido) USING INDEX TABLESPACE pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblPedidos OWNER TO postgres;

CREATE TABLE tblItemPedidos
(
    idPedido int NOT NULL,
    idCliente varchar NOT NULL,
    idItem varchar NOT NULL,

```

```

    idPro varchar NOT NULL,
    VrUnitario int NOT NULL,
    Cantidad int NOT NULL,
    idAcompañante varchar NOT NULL,
    CONSTRAINT IDItemPedidos PRIMARY KEY (idPedido, idItem) USING INDEX TABLESPACE
pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblItemPedidos OWNER TO postgres;

```

```

CREATE TABLE tblTmplItemPedidos
(
    idPedido int NOT NULL,
    idCliente varchar NOT NULL,
    idItem varchar NOT NULL,
    idPro varchar NOT NULL,
    VrUnitario int NOT NULL,
    Cantidad int NOT NULL,
    idAcompañante varchar NOT NULL,
    CONSTRAINT IDTmplItemPedidos PRIMARY KEY (idPedido, idItem) USING INDEX
TABLESPACE pg_default
) WITHOUT OIDS
TABLESPACE pg_default;
ALTER TABLE tblTmplItemPedidos OWNER TO postgres;

```

```

insert into tblplatos values ('01','Arroz chino','Porción de arroz con verduras, camarón cerdo picado,
pollo desmenuzado y raíces chinas','Se frita el arroz adicionándole el color, condimentos y las
verduras, por separado se fritan las carnes (camarón, cerdo, pollo) previamente cocinadas y
adobadas, por último se incorporan todos los ingredientes en una cocción
final', 'null', '15000', './image/arroces.jpg', 'Toy San', 'SI', 'A');

```

```

insert into tblplatos values ('02','Arroz valenciana','1/2 pollo más porción de arroz con verduras y
raíces chinas','Se frita el arroz adicionándole el color, condimentos y las verduras, adicionalmente
se frita el pollo previamente cocinado y adobado', 'null', '14500', './image/arroces.jpg', 'Toy
San', 'SI', 'A');

```

```

insert into tblplatos values ('03','Arroz con camarones','Porción de arroz con verduras, camarones y
raíces chinas','Se frita el arroz adicionándole el color, condimentos y las verduras, por separado se
fritan los camarones previamente cocinados y adobados, por último se incorporan todos los
ingredientes en una cocción final', 'null', '15000', './image/arroces.jpg', 'Toy San', 'SI', 'A');

```

```

insert into tblplatos values ('04','Arroz con pollo desmenuzado','Porción de arroz con verduras,
pollo desmenuzado y raíces chinas','Se frita el arroz adicionándole el color, condimentos y las
verduras, por separado se frita el pollo previamente cocinado y adobado, por último se incorporan
todos los ingredientes en una cocción final', 'null', '13000', './image/arroces.jpg', 'Toy San', 'SI', 'A');

```

```

insert into tblplatos values ('05','Arroz con mariscos','Porción de arroz con verduras,
pulpo, calamar, camarones y langostinos','Se frita el arroz adicionándole el color, condimentos y las
verduras, por separado se tienen previamente cocinados y adobados los mariscos, por último se
incorporan todos los ingredientes en una cocción final', 'null', '17000', './image/arroces.jpg', 'Toy
San', 'SI', 'A');

```

```

insert into tblplatos values ('06','Arroz frito','Porción de arroz con verduras, cerdo picado y raíces
chinas','Se frita el arroz adicionándole el color, condimentos y las verduras, por separado se frita el

```

cerdo previamente cocinado y adobado, por último se incorporan todos los ingredientes en una cocción final',null,'4000','../image/arroces.jpg','Toy San','SI','A');

insert into tblplatos values ('07','Arroz con presa','1/2 pollo más porción de arroz con verduras y raíces chinas','Se frita el arroz adicionándole el color, condimentos y las verduras, adicionalmente se frita el pollo previamente cocinado y adobado',null,'12000','../image/arroces.jpg','Toy San','SI','A');

insert into tblplatos values ('08','Arroz pintado','Porción de arroz con verduras y raíces chinas','Se frita el arroz adicionándole el color, condimentos y las verduras',null,'3500','../image/arroces.jpg','Toy San','SI','A');

insert into tblplatos values ('09','Pollo chino','1/2 pollo más porciones de arroz con verduras,papa frita,ensalada y salsa','Se frita el arroz adicionándole el color, condimentos y las verduras, adicionalmente se fritan las papas y el pollo previamente cocinado y adobado, finalmente se incorpora la salsa.',null,'15000','../image/pollos.jpg','Toy San','SI','A');

insert into tblplatos values ('10','Pollo a la friscasé','1/2 pollo con verduras en salsa y papa más porción de arroz frito','Se frita el arroz adicionándole el color, condimentos y las verduras y el cerdo picado, adicionalmente se cocinan las verduras con los condimentos y las papas previamente fritas y el pollo previamente cocinado y adobado',null,'14500','../image/pollos.jpg','Toy San','SI','A');

insert into tblIngredientes values ('01','Pollo','Pollos, Arroz chino,Cremas,Sopa china,Chuleta,Verduras,Espaguetis','Pollos Gar');

insert into tblIngredientes values ('02','Camarón','Pollo chino, Arroces,Verduras,Coctel,Cremas, Sopa china, espaguetis chinos','Mar Azul');

insert into tblIngredientes values ('03','Cerdo','Arroces, Pollo chino,Verduras, Chuleta,Sopa china, Pernil','La Rigoreña');

insert into tblIngredientes values ('04','Carne de res','Carne en bistek, Chuleta','La Rigoreña');

insert into tblIngredientes values ('05','Pescado','Trucha, Chuleta, Cremas','Mar Azul');

insert into tblIngredientes values ('06','Langostinos','Arroz con mariscos, Langostinos a la mariposa','Mar azul');

insert into tblIngredientes values ('07','Pulpo','Arroz con mariscos','Mar Azul');

insert into tblIngredientes values ('08','Calamar','Arroz con mariscos','Mar Azul');

insert into tblIngredientes values ('09','Champiñones','Pollo en salsa de champiñones, Cremas','Particular');

insert into tblIngredientes values ('10','Arroz','Todo','Surtigranos');

insert into tblIngredientes values ('11','Papa','Pollos, Arroces, TOrtillas, Chuletas','Disfruver');

insert into tblIngredientes values ('12','Zanahoria','Arroces, Pollo a la friscasé, Verduras,Tortillas','Disfruver');

insert into tblIngredientes values ('13','Brócoli','Queso parmesano','Disfruver');

insert into tblIngredientes values ('14','Cebolla','Arroces, verduras, Espaguetis','Disfruver');



```
insert into tblIngredientes values ('15','Apio','Pollo a la friscasé, Verduras','Disfruver');
```

```
insert into tblIngredientes values ('16','Cidra','Pollo a la friscasé, Verduras','Disfruver');
```

```
insert into tblIngredientes values ('17','Raices chinas','Arroces','Particular');
```

```
insert into tblIngredientes values ('18','Huevos','Sopa china, Tortillas','Particular');
```

```
insert into tblIngredientes values ('19','Ajinomoto','Queso parmesano','Particular');
```

```
insert into tblIngredientes values ('20','Aceite','Queso parmesano','Surtigranos');
```

```
insert into tblIngredientes values ('21','Caramelo','Arroces','Particular');
```

```
insert into tblIngredientes values ('22','Leche','Cremas, salsas','Ninguno');
```

```
insert into tblIngredientes values ('23','Espaguetis','Espaguetis, Sopa china','Ninguno');
```

```
insert into tblCaracteristicas values ('01','01');
```

```
insert into tblCaracteristicas values ('01','02');
```

```
insert into tblCaracteristicas values ('01','03');
```

```
insert into tblCaracteristicas values ('01','10');
```

```
insert into tblCaracteristicas values ('01','12');
```

```
insert into tblCaracteristicas values ('01','14');
```

```
insert into tblCaracteristicas values ('01','17');
```

```
insert into tblCaracteristicas values ('02','01');
```

```
insert into tblCaracteristicas values ('02','03');
```

```
insert into tblCaracteristicas values ('02','10');
```

```
insert into tblCaracteristicas values ('02','12');
```

```
insert into tblCaracteristicas values ('02','14');
```

```
insert into tblCaracteristicas values ('02','17');
```

```
insert into tblCaracteristicas values ('03','02');
```

```
insert into tblCaracteristicas values ('03','10');
```

```
insert into tblCaracteristicas values ('03','12');
```

```
insert into tblCaracteristicas values ('03','14');
```

```
insert into tblCaracteristicas values ('03','17');
```

```
insert into tblCaracteristicas values ('04','01');
```

```
insert into tblCaracteristicas values ('04','10');
```

```
insert into tblCaracteristicas values ('04','12');
```

```
insert into tblCaracteristicas values ('04','14');
```

```
insert into tblCaracteristicas values ('04','17');
```

```
insert into tblCaracteristicas values ('05','02');
```

```
insert into tblCaracteristicas values ('05','06');
```

```
insert into tblCaracteristicas values ('05','07');
```

```
insert into tblCaracteristicas values ('05','08');
```

```
insert into tblCaracteristicas values ('05','10');
```

```
insert into tblCaracteristicas values ('05','12');
```

```
insert into tblCaracteristicas values ('05','14');
```

```
insert into tblCaracteristicas values ('05','17');
```

```
insert into tblCaracteristicas values ('06','03');
```

```
insert into tblCaracteristicas values ('06','10');
```

```
insert into tblCaracteristicas values ('06','12');
```

```
insert into tblCaracteristicas values ('06','14');
```

```
insert into tblCaracteristicas values ('06','17');
```

```
insert into tblCaracteristicas values ('07','01');
```



```
insert into tblCaracteristicas values ('22','12');
insert into tblCaracteristicas values ('22','13');
insert into tblCaracteristicas values ('22','14');
insert into tblCaracteristicas values ('22','15');
insert into tblCaracteristicas values ('22','16');
```

```
insert into tblSectores values ('01', 'Privado');
insert into tblSectores values ('02', 'Construccion');
insert into tblSectores values ('03', 'Indutria');
insert into tblSectores values ('04', 'Comercio');
```

```
insert into tblmonedas values ('01', 'pesos 01');
insert into tblmonedas values ('02', 'dolares 02');
insert into tblmonedas values ('03', 'bolivares 03');
```

```
insert into tblpaises values ('01', 'colombia');
insert into tblpaises values ('02', 'venezuela');
insert into tblpaises values ('03', 'españa');
insert into tblpaises values ('04', 'peru');
insert into tblpaises values ('05', 'brasil');
```

```
insert into tblrestaurantes values ('01', 'restaurante on line','calle 12 # 12-12','0000000','0000000','restauranteuno@hotmail.com');
insert into tblrestaurantes values ('02', 'restaurante toy san','carrera 5 # 6-49','2123017','0000000','restaurantetoysan@ert.com');
insert into tblrestaurantes values ('03', 'restaurante ching li','carrera 4 # 13-94','2137500','2113443','restaurantechingli@ert.com');
insert into tblrestaurantes values ('04', 'retaurante chino chino','avenida circunvalar','2105855','0000000','restaurantechinochino@ert.com');
```

```
insert into tblclientes values (1,'1', 'administrador sitio','calle carrera','550-23232 3222','administradorsitio@hotmail.com','01','01','01','01','1','01');
insert into tblclientes values (2,'2', 'administrador restaurante','calle carrera','550-23232 3222','administradorrestaurante@hotmail.com','01','01','01','01','2','02');
insert into tblclientes values (3,'3', 'despachos','calle carrera','550-23232 3222','calcenter@hotmail.com','01','01','01','01','3','03');
insert into tblclientes values (4,'4', 'domicilios','calle carrera','550-23232 3222','calcenter@hotmail.com','01','01','01','01','4','04');
insert into tblclientes values (5,'5', 'call center','calle carrera','550-23232 3222','calcenter@hotmail.com','01','01','01','01','5','05');
insert into tblclientes values (6,'6', 'carolina ching','calle 12 # 13-55','550-23232 3222','caro@hotmail.com','01','01','01','01','6','00');
```

```
insert into tbltipoclientes values ('01', 'usuario del sistema');
insert into tbltipoclientes values ('02', 'virtual');
insert into tbltipoclientes values ('03', 'comercial');
insert into tbltipoclientes values ('04', 'persona jurídica');
```

```
insert into tbltipoplatos values ('01', 'arrocés');
insert into tbltipoplatos values ('02', 'pollos');
insert into tbltipoplatos values ('03', 'carnes');
insert into tbltipoplatos values ('04', 'platos especiales');
```

```
insert into tbltipoplatos values ('05', 'comidas de mar');
insert into tbltipoplatos values ('06', 'cremas');
insert into tbltipoplatos values ('07', 'sopas');
insert into tbltipoplatos values ('08', 'spaguetis');
insert into tbltipoplatos values ('09', 'tortillas');

insert into tblpedidos values (1,'00','inicial','x','01/01/2007','11:00','00');

insert into tblestados values ('01', 'preparación');
insert into tblestados values ('02', 'despachado');
insert into tblestados values ('03', 'servicio cumplido');
insert into tblestados values ('04', 'no entregado');

insert into tblacompañantes values ('01', 'platos desechables');
insert into tblacompañantes values ('02', 'gaseosa');
insert into tblacompañantes values ('03', 'cubiertos desechables');
insert into tblacompañantes values ('04', 'agua');
insert into tblacompañantes values ('05', 'jugos');
insert into tblacompañantes values ('06', 'no aplica');
```

## ANEXO H

### DEFINICIÓN DE LA BASE DE DATOS (Aspectos técnicos)

#### Tabla: public\_tblacompañantes

##### Propiedades

DateCreated:	22/03/2007 11:18:23 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {FA3B1997-729E-4511- 11:18:25 p.m.		LastUpdated: 22/03/2007
	86DA-79C307AC49D7}}		
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	2
Updatable:	Verdadero		

##### Columnas

Nombre	Tipo	Tamaño
idacompañante	Texto	254
descripcion	Texto	254

#### Tabla: public\_tblcaracteristicas

##### Propiedades

DateCreated:	22/03/2007 11:18:23 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {3FB1D30A-28BD-44FF- 11:18:25 p.m.		LastUpdated: 22/03/2007
	8CCD-2AD536E49335}}		
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	2
Updatable:	Verdadero		

##### Columnas

Nombre	Tipo	Tamaño
crefpla	Texto	254
ccoding	Texto	254

#### Tabla: public\_tblclientes

##### Propiedades

DateCreated:	22/03/2007 11:18:23 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {557925A9-7A1B-4BC7- 11:18:25 p.m.		LastUpdated: 22/03/2007
	886D-C779B1BD29C6}}		
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	4
Updatable:	Verdadero		

##### Columnas

Nombre	Tipo	Tamaño
idcliente	Texto	254
nombre	Texto	254

direccion	Texto	254
telefono	Texto	254
correoe	Texto	254
moneda	Texto	254
pais	Texto	254
idtipocliente	Texto	254
idsector	Texto	254
clave	Texto	254
privilegio	Texto	254

**Tabla: public\_tblestados**

**Propiedades**

DateCreated:	22/03/2007 11:18:23 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {4D3A0FA0-E630-42B5-8D9B-8F8C7135E2AD}}	LastUpdated:	22/03/2007
11:18:25 p.m.			
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	4
Updatable:	Verdadero		

**Columnas**

Nombre	Tipo	Tamaño
idestado	Texto	254
descripcion	Texto	254

**Tabla: public\_tblingredientes**

**Propiedades**

DateCreated:	22/03/2007 11:18:23 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {60B6CEA5-EFAF-451D-8AA9-FB7FDE236876}}	LastUpdated:	22/03/2007
11:18:25 p.m.			
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	5
Updatable:	Verdadero		

**Columnas**

Nombre	Tipo	Tamaño
ccoding	Texto	254
cdesing	Texto	254
crecing	Texto	254
cproing	Texto	254

**Tabla: public\_tblitempedidos**

**Propiedades**

DateCreated:	22/03/2007 11:18:24 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {B6028027-E9A3-4E6C-AA26-CAF0DE75A19C}}	LastUpdated:	22/03/2007
11:18:25 p.m.			
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	2
Updatable:	Verdadero		

**Columnas**

Nombre	Tipo	Tamaño
--------	------	--------

idpedido	Entero largo	4
idcliente	Texto	254
iditem	Texto	254
idpro	Texto	254
vrunitario	Entero largo	4
cantidad	Entero largo	4
idacompañante	Texto	254

**Tabla: public\_tblmonedas**

**Propiedades**

DateCreated:	22/03/2007 11:18:24 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {CB50D998-C23D-40CE-995A-22DD41B30A2C}}	LastUpdated:	22/03/2007
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	3
Updatable:	Verdadero		

**Columnas**

Nombre	Tipo	Tamaño
idmoneda	Texto	254
descripcion	Texto	254

**Tabla: public\_tblpaises**

**Propiedades**

DateCreated:	22/03/2007 11:18:24 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {67FE6662-E427-4060-BFCC-EABF136EA62A}}	LastUpdated:	22/03/2007
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	5
Updatable:	Verdadero		

**Columnas**

Nombre	Tipo	Tamaño
idpais	Texto	254
descripcion	Texto	254

**Tabla: public\_tblpedidos**

**Propiedades**

DateCreated:	22/03/2007 11:18:24 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {F74BA485-C503-4439-B498-4CF71188D52C}}	LastUpdated:	22/03/2007
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	2
Updatable:	Verdadero		

**Columnas**

Nombre	Tipo	Tamaño
idpedido	Entero largo	4
idcliente	Texto	254
detalle	Texto	254
formapago	Texto	254
fecha	Fecha/Hora	8

hora	Fecha/Hora	8
idestado	Texto	254

**Tabla: public\_tblperfiles**

**Propiedades**

DateCreated:	22/03/2007 11:18:24 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {98EB67CC-4AC6-4D5B- 11:18:25 p.m.	AA32-9B32D3B4AFE7}}	LastUpdated: 22/03/2007
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	0
Updatable:	Verdadero		

**Columnas**

Nombre	Tipo	Tamaño
idperfil	Texto	254
idcliente	Texto	254
iditem	Texto	254
idproducto	Texto	254

**Tabla: public\_tblplatos**

**Propiedades**

DateCreated:	22/03/2007 11:18:24 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {36C5D5AA-8521-4339- 11:18:25 p.m.	BA2B-9E43EFE41DE2}}	LastUpdated: 22/03/2007
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	3
Updatable:	Verdadero		

**Columnas**

Nombre	Tipo	Tamaño
crefpla	Texto	254
cdespla	Texto	254
cdetpla	Texto	254
crecpla	Texto	254
ctipla	Texto	254
cprepla	Texto	254
cimapla	Texto	254
ccodres	Texto	254
cmenpla	Texto	254
cestpla	Texto	254

**Tabla: public\_tblreservaciones**

**Propiedades**

DateCreated:	22/03/2007 11:18:24 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {2E2A8952-A049-4F45- 11:18:25 p.m.	9233-A3C1D396B5F6}}	LastUpdated: 22/03/2007
NameMap:	Datos binarios largos	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordCount:	0
Updatable:	Verdadero		

**Columnas**



Nombre	Tipo	Tamaño
idreservacion	Entero largo	4
idrestaurante	Texto	254
idcliente	Texto	254
fecharegistro	Fecha/Hora	8
año	Texto	254
mes	Texto	254
dia	Texto	254
hora	Texto	254
mesa	Texto	254
observacion	Texto	254

**Tabla: public\_tblrestaurantes**

**Propiedades**

DateCreated:	22/03/2007 11:18:24 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {A824F7DA-1E5E-4205-11:18:25 p.m.	LastUpdated:	22/03/2007
NameMap:	AFCD-0DCF63D4A564}}	OrderByOn:	Falso
Orientation:	Datos binarios largos	RecordCount:	4
Updatable:	De izquierda a derecha		
	Verdadero		

**Columnas**

Nombre	Tipo	Tamaño
idrestaurante	Texto	254
descripcion	Texto	254
direccion	Texto	254
telefono	Texto	254
fax	Texto	254
correoe	Texto	254

**Tabla: public\_tblsectores**

**Propiedades**

DateCreated:	22/03/2007 11:18:24 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {EA3A8678-6384-420D-11:18:25 p.m.	LastUpdated:	22/03/2007
NameMap:	BE11-F4988468B1CF}}	OrderByOn:	Falso
Orientation:	Datos binarios largos	RecordCount:	4
Updatable:	De izquierda a derecha		
	Verdadero		

**Columnas**

Nombre	Tipo	Tamaño
idsector	Texto	254
descripcion	Texto	254

**Tabla: public\_tbltipoclientes**

**Propiedades**

DateCreated:	22/03/2007 11:18:24 p.m.	DefaultView:	Hoja de datos
GUID:	{guid {440BF20C-8410-451A-11:18:25 p.m.	LastUpdated:	22/03/2007
NameMap:	B69B-AD1B2CD369B8}}	OrderByOn:	Falso
Orientation:	Datos binarios largos	RecordCount:	4
	De izquierda a derecha		

Updatable: Verdadero

**Columnas**

Nombre	Tipo	Tamaño
idtipocliente	Texto	254
descripcion	Texto	254

**Tabla: public\_tbltipoplatos**

**Propiedades**

DateCreated: 22/03/2007 11:18:24 p.m. DefaultView: Hoja de datos  
GUID: {guid {19637138-D0E6-45CC-11:18:25 p.m. B491-E5ADA0684F22}} LastUpdated: 22/03/2007  
NameMap: Datos binarios largos OrderByOn: Falso  
Orientation: De izquierda a derecha RecordCount: 7  
Updatable: Verdadero

**Columnas**

Nombre	Tipo	Tamaño
idtipoplato	Texto	254
descripcion	Texto	254

**Tabla: public\_tbltmpitempedidos**

**Propiedades**

DateCreated: 22/03/2007 11:18:25 p.m. DefaultView: Hoja de datos  
GUID: {guid {1505E574-DF49-4362-11:18:25 p.m. B09A-FC57C37CEE47}} LastUpdated: 22/03/2007  
NameMap: Datos binarios largos OrderByOn: Falso  
Orientation: De izquierda a derecha RecordCount: 2  
Updatable: Verdadero

**Columnas**

Nombre	Tipo	Tamaño
idpedido	Entero largo	4
idcliente	Texto	254
iditem	Texto	254
vrunitario	Entero largo	4
cantidad	Entero largo	4
idacompañante	Texto	254

## ANEXO I

### METODOLOGÍA DE DESARROLLO

#### Conceptos de Ingeniería

La Metodología a utilizar se basa en UP (Proceso Unificado). La selección de dicha metodología esta basada en un estudio no forma, donde se indago varias fuentes y después de realizar un análisis al respecto se llevo a dicha conclusión. El Lenguaje para el proceso de Ingeniería del Software es el Lenguaje Unificado de Modelado UML.

La programación se hará en: Lenguaje JSP, HTML. La Base de Datos a utilizar es POTGRES.

Las fases que forman parte de este proyecto están directamente ligadas al Proceso Unificado de Desarrollo, entre las que tenemos:

1. Modelo de Requisitos
2. Modelo de Análisis
3. Modelo de Diseño
4. Pruebas
5. Implementación
6. Gestión de la configuración.

Se utilizarán los siguientes artefactos:

- Modelo de casos de uso: Incluye los casos de uso, el diagrama de los mismos y los diagramas de diseño del sistema.
- Modelo de Dominio.
- Modelo de clases
- Diagrama de estados del sistema
- Diagramas de arquitectura
- Diagramas de Actividad y colaboración

#### Modelo de Requisitos

El primer problema que se presenta es la captura de los requisitos del usuario. Para empezar, necesitamos recoger los requisitos de los usuarios o clientes de una manera sistemática y organizada. Para ello precisamos de unas directrices o líneas guía, ya que en general los usuarios expresan los requerimientos de la aplicación de forma muy variable, tanto en la forma como en el contenido. Nos

interesa pues sistematizar la captura, con el fin de hacer los requisitos manejables y analizables.

Para completar esta fase debemos establecer los requisitos no funcionales, relacionados por ejemplo con el rendimiento, la disponibilidad o la seguridad. Cuando estén asociados a un caso de uso, podrán especificarse en la plantilla de caso de uso propuesta. Los requisitos no funcionales que afecten a varios casos de uso o sean globales a toda la organización, serán recogidos en el apartado correspondiente de la plantilla de ERS elegida. El modelo del negocio puede ayudar a encontrar requisitos no funcionales, pues por ejemplo los campos tiempo de ejecución y coste de ejecución de la plantilla de descripción de casos de uso del negocio expresan necesidades no funcionales que deben ser trasladadas a la plantilla de ERS y asociadas a los correspondientes casos de uso del sistema. Por otro lado, el que los procesos del negocio sean el resultado del análisis de los objetivos de la organización, permite que los requisitos (funcionales y no funcionales) del sistema puedan ser validados y verificados contra los objetivos

Hay varios modelos de proceso, y normalmente las organizaciones tienen "su proceso" y lo utilizan siempre, para seguir siempre unos pasos sistemáticos en la construcción de su software. En el modelo de requisitos, en una adaptación del llamado Proceso Unificado (UP), que es el más utilizado en los últimos tiempos y que propone una serie de pasos que seguiremos hasta lograr nuestro programa que sea capaz de sumar dos números; se inicia un trabajo partiendo de las necesidades del sistema o de los usuarios.

Para establecer las funciones que debe proporcionar la aplicación debemos elaborar generalmente un modelo de casos de uso, que es un resumen de "lo que se podrá hacer" con nuestro programa. Así, un cajero automático tendría estos casos de uso "sacar dinero", "ingresar dinero", "consultar saldo", y los que se nos ocurran.

Los casos de uso representan posibles escenarios de uso del programa, y contribuyen a concentrar el esfuerzo en torno a las funciones exactas del programa, al tiempo que facilitan establecer los requisitos que debe reunir nuestra aplicación. Para nuestro caso, sólo hay un caso de uso (que traducción más horrible, por cierto) que sería "sumar dos números".

A continuación, en la siguiente gráfica, se muestra un ejemplo de: cómo puede culminar un proceso de requisitos basados en UP

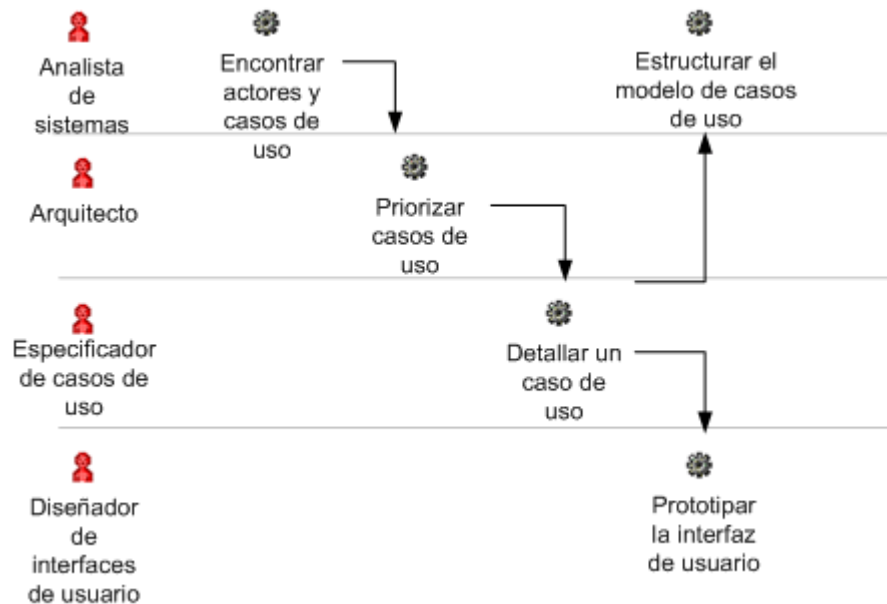


Figura 1 Técnicas captura de Requisitos

### Modelo de Análisis

La siguiente fase es la de construcción del modelo de análisis. Es éste un modelo de objetos del sistema, cuya operación debe cumplir los requisitos especificados por el usuario, pero a nivel conceptual, es decir, sin incluir consideraciones de implementación. Debe ser un modelo completo, es decir, detallado, que permita verificar que su operación cumple los requisitos de usuario, sin dejar lugar a dudas, es decir, debe estar especificado lo que hace, cuando y porqué, si bien no es necesario bajar al detalle de cómo lo hace. Debe ser un modelo completo en el plano estático y en el dinámico.

Este modelo es necesario para profundizar en cada elemento definido en el modelo de requisitos, se inicia un estudio previo a los requisitos funcionales y no funcionales del sistema; todo con el fin de determinar los casos de uso análisis.

Como lo indica el apartado anterior, unos de los objetivos del modelo de análisis es entender el Uso del Sistema: En la reunión con los usuarios potenciales, los desarrolladores se reunieron con los usuarios para descubrir a los actores que inician cada Caso de Uso y a los actores que se beneficiaran de ellos (Un actor puede ser tanto un sistema como una persona). (Diagrama de Casos de Uso)

Un ejemplo de trabajo a realizar, y basados en METOCUBE lo indica la siguiente ilustración:

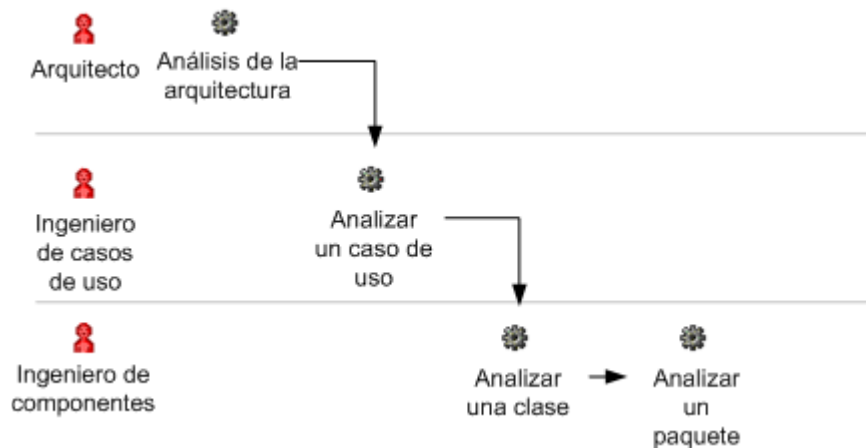


Figura 11 Técnicas de Análisis

### Modelo de Diseño

El modelo de diseño es la fase más interesante del modelado de software, ya que nos posibilita la concepción del producto, para lograr una visualización correcta del diseño del sistema y que este acorde al ciclo de vida que plantea UP\*, es necesario constituir los siguientes diagramas:

Cuando el modelo de análisis está en una etapa madura, se inicia con el proceso de construcción del producto, éste se logra mediante la construcción de un modelo de diseño, que nos permite ilustrar todas las características correspondientes al software; para lograr un buen modelo es necesario atender a las técnicas de ingeniería de software vigentes, si observamos, el siguiente diagrama nos permite organizar dicha labor.

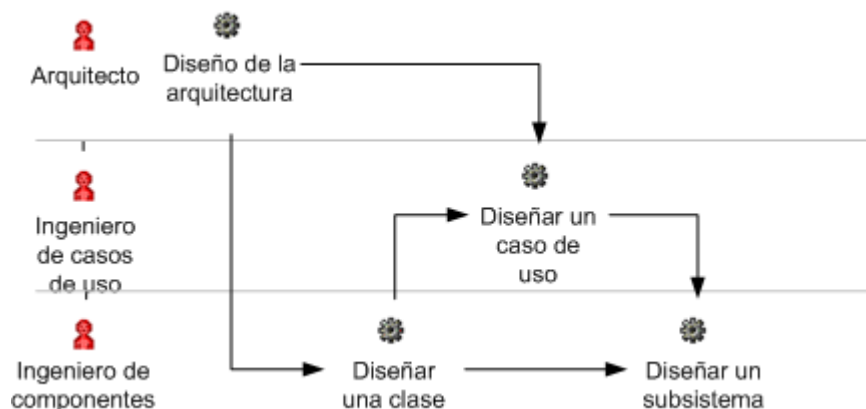


Figura 3 Técnicas de Diseño

\* Proceso Unificado

## **Los Diagramas de Despliegue**

Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

## **Diagrama de componentes**

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

## **Diagrama de Colaboración (Diseño)**

El Diagrama de Colaboración presenta una alternativa al diagrama de secuencia para modelar interacciones entre objetos en el sistema. Mientras que el diagrama de secuencia se centra en la secuencia cronológica del escenario que estamos modelando, el diagrama de colaboración se centra en estudiar todos los efectos de un objeto dado durante un escenario.

## **Diagrama Estético**

Se indica en forma estética como quedará la interfaz, para que el ingeniero implementador adopte una idea general en el momento de la construcción del software.

## ANEXO J

### MANUAL TÉCNICO

#### Aspectos técnicos

##### Instalación de la aplicación

Para la instalación de la aplicación es necesario configurar un servidor WEB de tal forma que permita la interacción de los clientes con el restaurante en línea desde cualquier punto de acceso.

El servidor es un elemento esencial de distribución de recursos para el proyecto y para tales efectos se usa el software TOMCAT, el cual permite la interacción de la base de datos con los clientes, los cuales cuentan con un BROWSER (navegador) que procese las transacciones.

Los siguientes pasos son requeridos para la instalación y configuración de la aplicación:

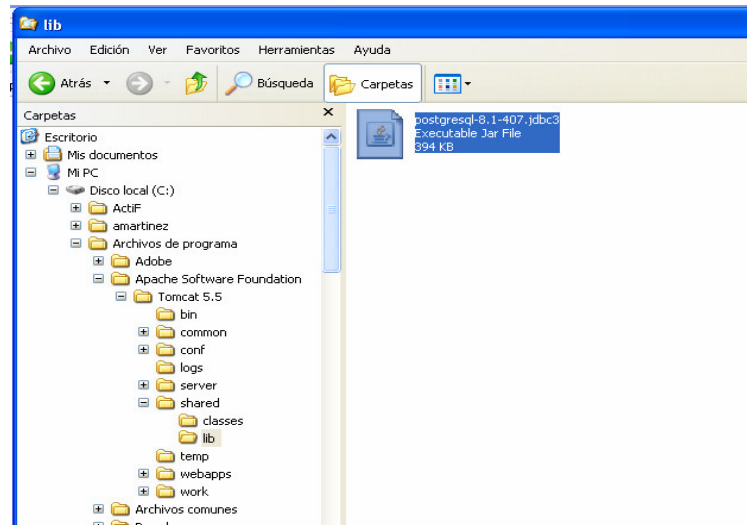
1. Instalar JAVA
2. Instalar TOMCAT
3. Tomar del CD el archivo res.rar que se encuentra adjunto y copiarlo en la carpeta C:\Archivos de programa\Apache Software Foundation\Tomcat 5.5\WEBapps\ROOT, y luego de descomprimirla allí mismo se deberá crear una carpeta llamada res.
4. Instalación de DRIVERS
5. Ejecución de SCRIPT SQL
6. Compilación clase base
7. Configuración de los DRIVERS para los reportes

#### Pasos

##### Instalación de DRIVERS

Para la puesta en marcha de la aplicación es necesario el uso de DRIVERS que permitan la conexión con las bases de datos. Para tales efectos es necesario copiar el archivo `postgresql-8.1-407.jdbc3.jar` en la carpeta C:\Archivos de programa\Apache Software Foundation\Tomcat 5.5\shared\lib, tal y como lo indica el siguiente gráfico:



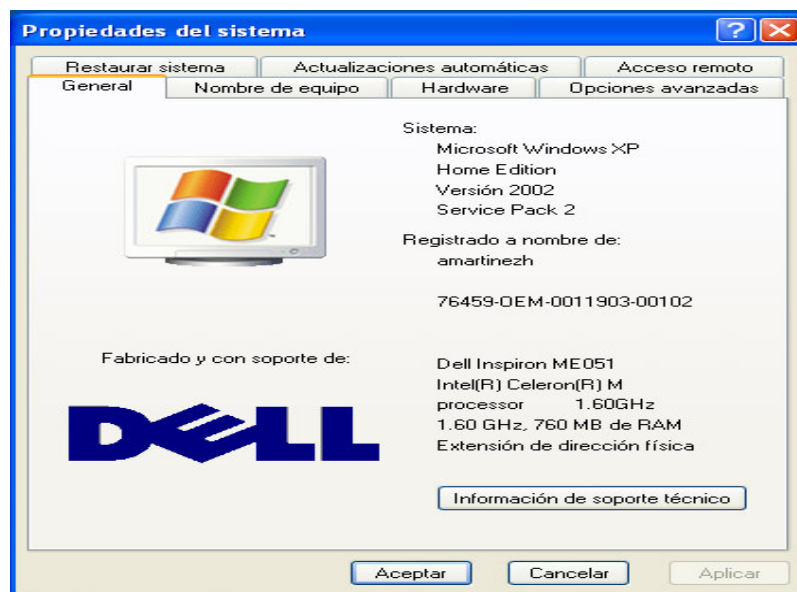


## Ejecución de SCRIP SQL

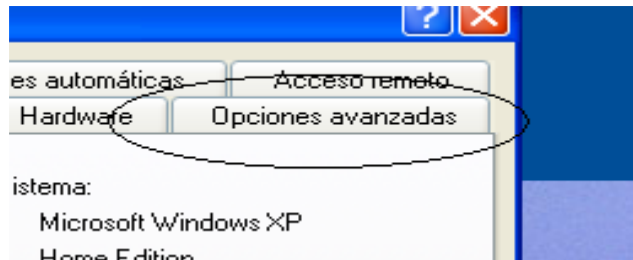
Para la instalación de las bases de datos es necesario ejecutar el SCRIP SQL que permita la creación de las bases de datos del proyecto.

Primero que todo es necesario ingresar una variable de entorno para que se pueda ejecutar el comando *psql*, y se logra mediante los siguientes pasos

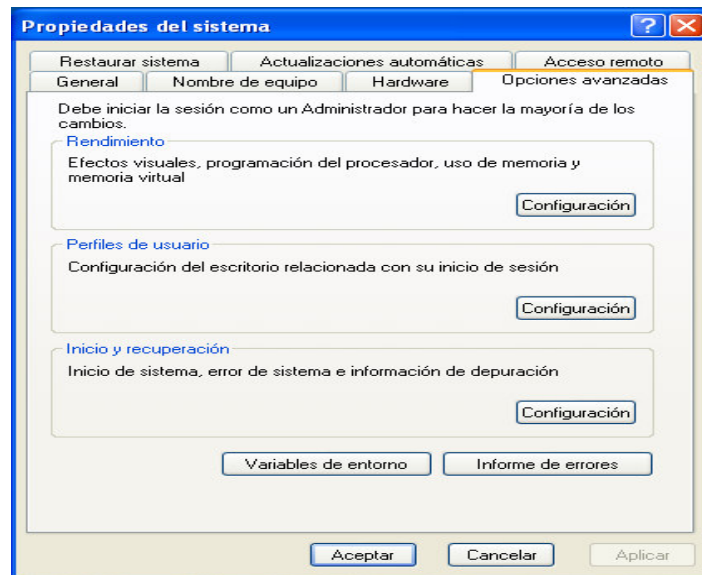
### a. Abrir la ventana de propiedades de MI PC



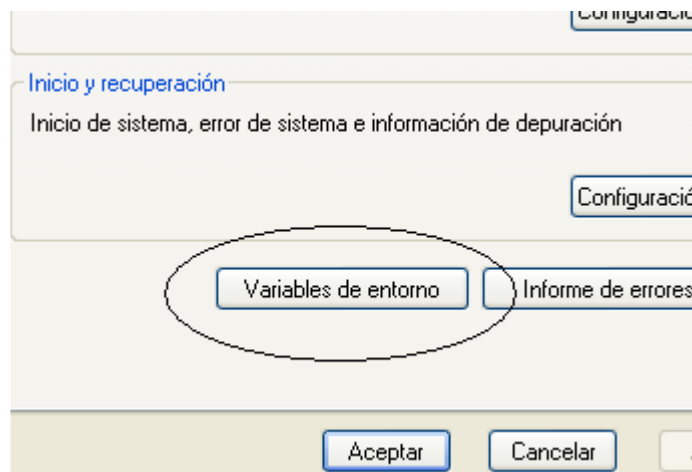
Luego de lograr abrir la ventana anterior es necesario hacer clic en la pestaña de opciones avanzadas, tal y como aparece en el siguiente gráfico:



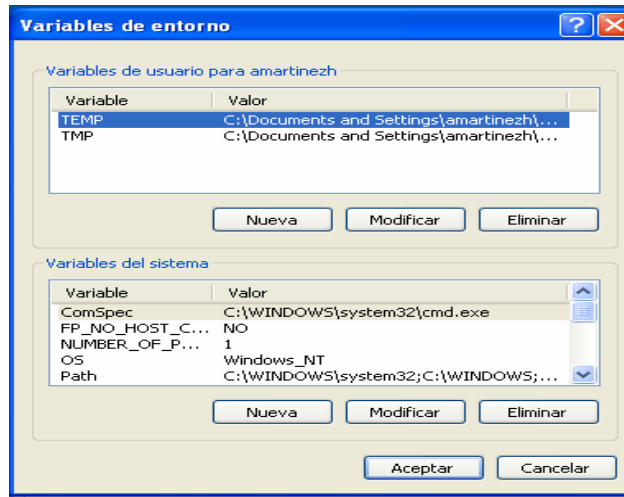
Lo anterior mostrará la siguiente ventana



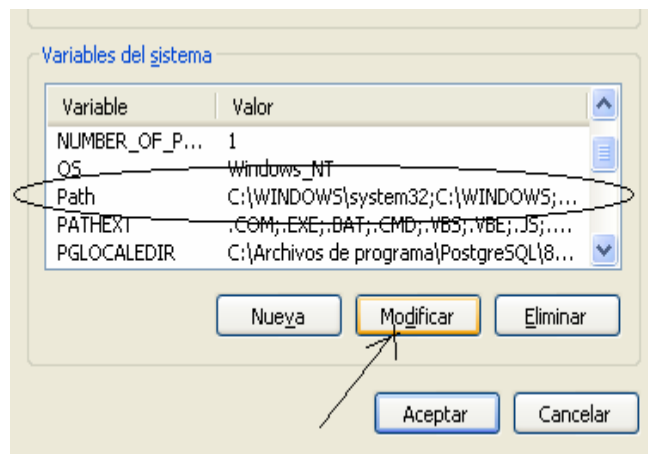
b. Para luego hacer clic en el botón variables de entorno, basarse en el siguiente gráfico:



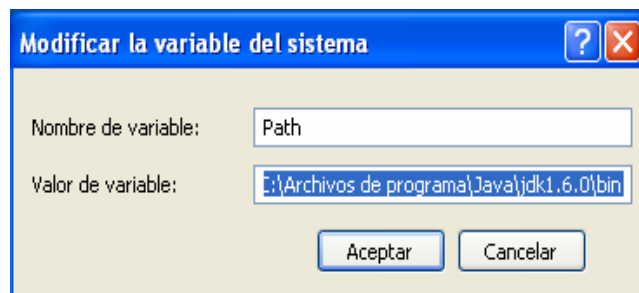
Lo anterior nos mostrará la siguiente interfaz:



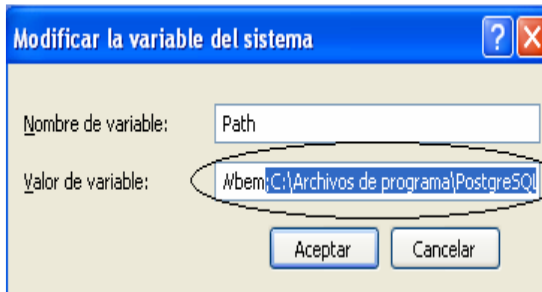
c. El paso a seguir es buscar en el área de Variables del sistema la variable Path, todo con el fin de incluir el PATH del POSTGRES, observe el siguiente gráfico:



Y como lo indica el gráfico será necesario hacer clic en el botón modificar para hacer el nuevo ingreso, lo anterior se logra mediante la siguiente ventana:

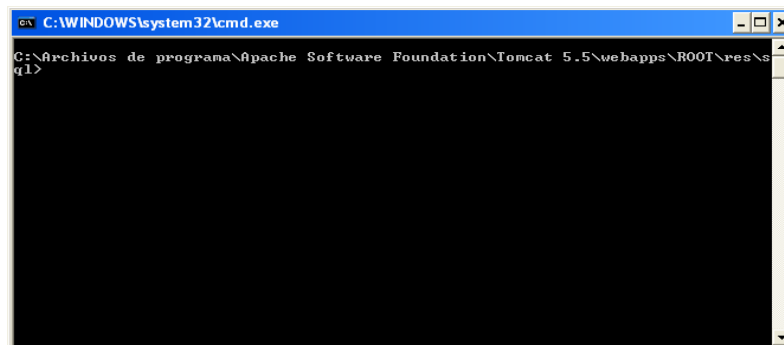


d. Para lograr dicho cometido es necesario agregar la cadena “ ;C:\Archivos de programa\PostgreSQL\8.1\bin;” tal y como aparece en el resalto sobre el siguiente gráfico:



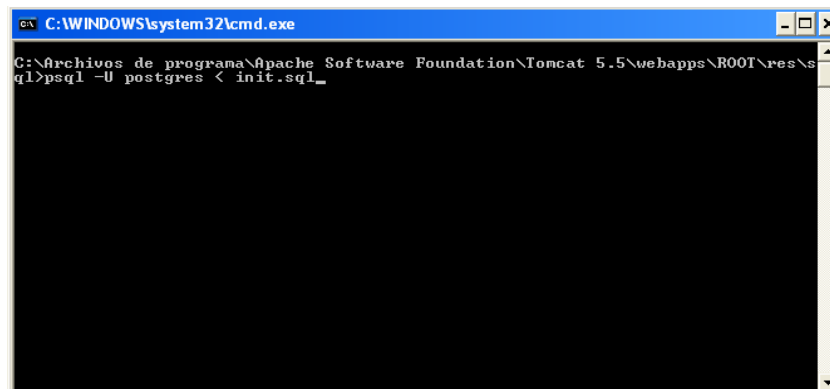
Para terminar el primer paso es necesario aceptar en todas las ventanas.

Segundo que todo que todo es necesario ingresar en el símbolo de sistema para la ejecución del\_programa\_PSQL. En el momento que se ingrese al símbolo de sistema es necesario entrar en la carpeta C:\Archivos de programa\Apache Software Foundation\Tomcat 5.5\WEBapps\ROOT\res\sql por medio de los comandos correspondientes. Lo cual deberá aparecer de la siguiente forma:



Luego se deberá ingresar el siguiente comando:

*Psql -U postgres < init.sql*



Donde la U deberá estar dispuesta en mayúsculas, y al terminar la operación aparecerá la siguiente ventana.



```
C:\WINDOWS\system32\cmd.exe
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
C:\Archivos de programa\Apache Software Foundation\Tomcat 5.5\webapps\ROOT\res\sg1>
```

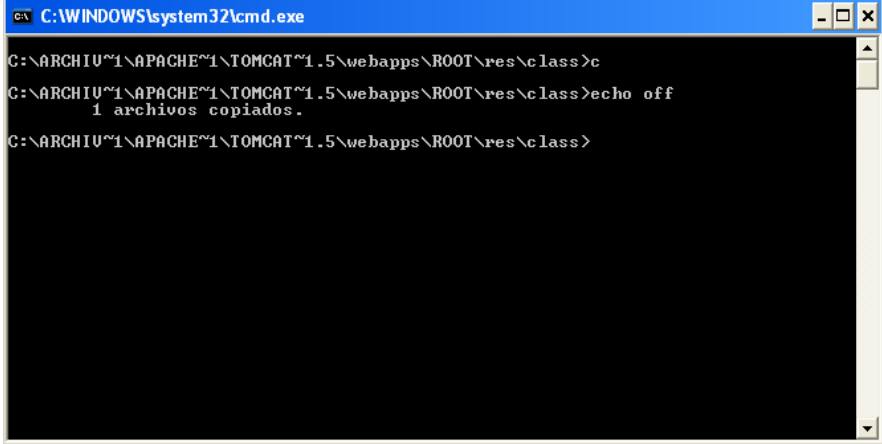
### Compilación clase base

El software que se está instalando requiere de la configuración de la clase que permite la conexión con la base de datos, ésta deberá ser compilada y copia en la carpeta C:\Archivos de programa\Apache Software Foundation\Tomcat 5.5\WEBapps\ROOT\WEB-INF\classes\res, es importante anotar que es necesario crearla, ya que en el sistema no se encuentra.

Luego, y por medio del símbolo de sistema se compilará utilizando un BAT que contiene las siguientes instrucciones:

```
echo off
javac co.java
copy co.class ..\..\WEB-INF\classes\res
```

La ejecución se logra mediante el programa C, tal y como lo indica la siguiente ventana:



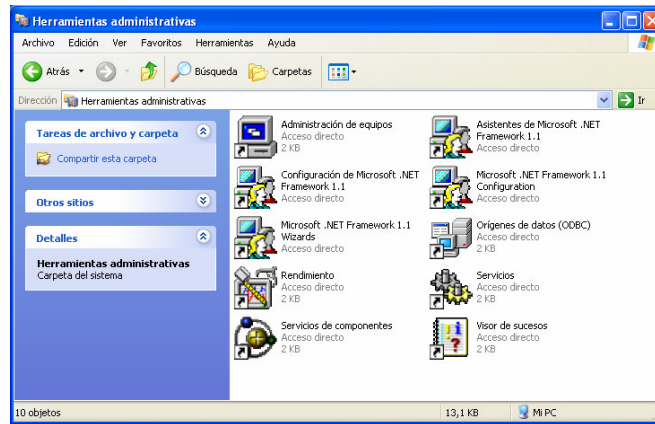
```
C:\WINDOWS\system32\cmd.exe
C:\ARCHIVO~1\APACHE~1\TOMCAT~1.5\webapps\ROOT\res\class>c
C:\ARCHIVO~1\APACHE~1\TOMCAT~1.5\webapps\ROOT\res\class>echo off
1 archivos copiados.
C:\ARCHIVO~1\APACHE~1\TOMCAT~1.5\webapps\ROOT\res\class>
```

## Configuración de los DRIVES para los REPORTES

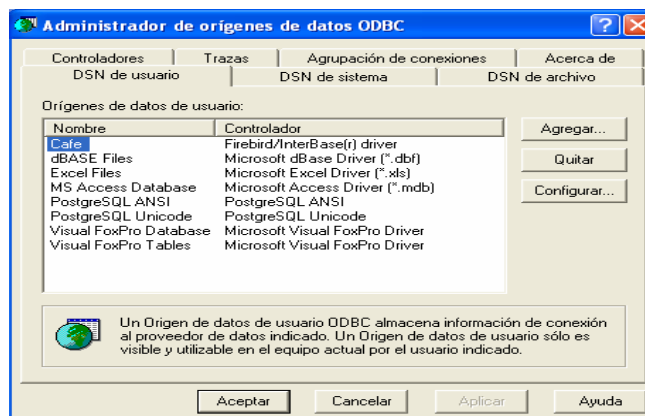
Los reportes hacen conexión con la base de datos mediante ODBC, y para tales efectos es necesario ingresar al panel de control y buscar el icono de Herramientas administrativas, abajo se ilustra un ejemplo de cómo identificarlo:



Al momento de hacer clic en dicho icono se desplegará la siguiente ventana:



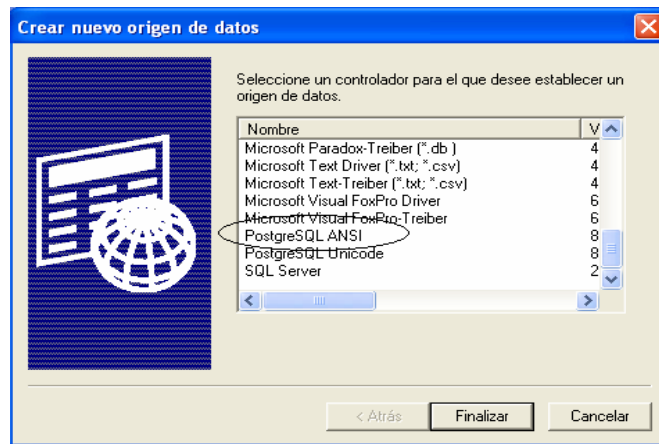
Y luego se deberá hacer clic en el icono Orígenes de datos (ODBC), para luego ingresar un nuevo elemento mediante la siguiente ventana:



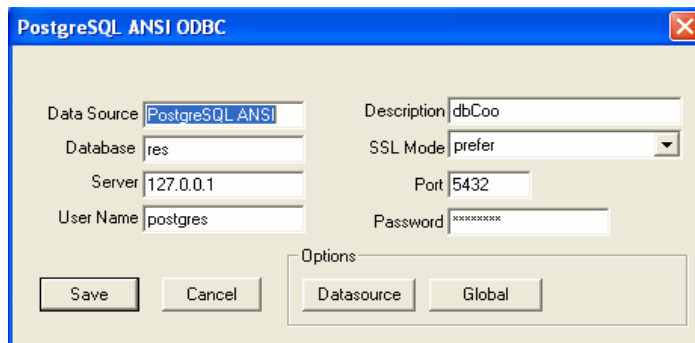
Luego de debe hacer clic en la pestaña DNS de sistema para que aparezca la siguiente ventana:



Luego hacer clic en el botón agregar, esto nos dará como resultado la siguiente ventana:



El paso a seguir buscar el nombre: PostgreSQL ANSO y presionar el botón finalizar, para que nos dé como resultado la siguiente interfaz:



Observar todos los campos y que información contiene, debido a que se deberá ingresar la misma información para lograr un buen efecto en la configuración.

## **Instalación Y Configuración De TOMCAT**

Ya vimos que si queremos hacer uso de las características avanzadas que se han desarrollado sobre la plataforma Java 2 como son el uso de los SERVLETs y de las páginas JSP (Java Server Pages) necesitamos de un motor que al incorporarlo a nuestro servidor WEB añada dicha funcionalidad. Dentro de los distintos motores que existen en el mercado para extender las características adicionales de Java 2 probablemente uno de los proyectos más interesantes liderado por Apache.org es el proyecto JAKARTA en el cual se ha desarrollado la aplicación TOMCAT, la cual no es más que un servidor de aplicaciones con las características de servirnos como motor de SERVLETs y JSP y con la ventaja adicional de ser gratuito, actualmente la última versión estable disponible es la 3.2 aunque ya se encuentra en desarrollo la beta 4.0.

En este tutorial vamos a ver los pasos que tenemos que seguir si queremos instalar correctamente esta aplicación. Como siempre lo primero que debemos hacer es bajarnos los archivos que contienen los distintos ejecutables del programa, esto puede hacerlo desde el WEB de Sun desde la página de desarrolladores en la siguiente dirección:

<http://java.sun.com/products/jsp/tomcat>

o bien directamente desde el servidor ftp de Apache en la siguiente dirección:

<http://jakarta.apache.org/builds/tomcat/release/v3.2.1/bin/>

El archivo que nos vamos a bajar recibe el nombre de jakarta-tomcat-3.2.1.zip y ocupa aproximadamente unos 3 megas de tamaño. Una vez que tenemos el programa en nuestro disco la instalación no es muy complicada y podemos resumirla en los siguientes pasos básicos:

### **Pasos**

1.- Descomprimir el archivo JAKARTA-TOMCAT-3.2.1.zip, por defecto esta descompresión se realiza sobre el directorio JAKARTA-TOMCAT-3.2.1.zip pero podéis hacerla en cualquier otro directorio aunque después tiene que tener en cuenta que esta nueva ruta será la que debe especificar en la configuración de TOMCAT.

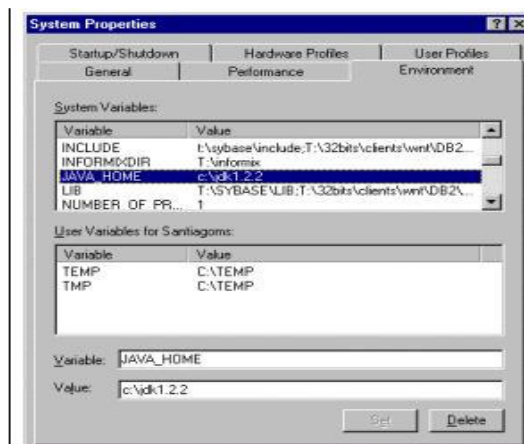
2.- TOMCAT presupone que tiene instalada en vuestra máquina alguna versión del JDK, siempre versión 1.1 o superior junto con la máquina virtual correspondiente (JRE), por lo que si no tiene instalado el JDK este es el momento de hacerlo. Y puede descargarlo también desde el WEB de Sun. Obviamente cuando finalicemos la instalación del JDK no olvide establecer los valores de la variable de sistema CLASSPATH ni olvide añadir la ruta de la carpeta BIN del JDK a vuestra variable PATH a fin de poder encontrar los ejecutables del mismo desde cualquier ubicación.



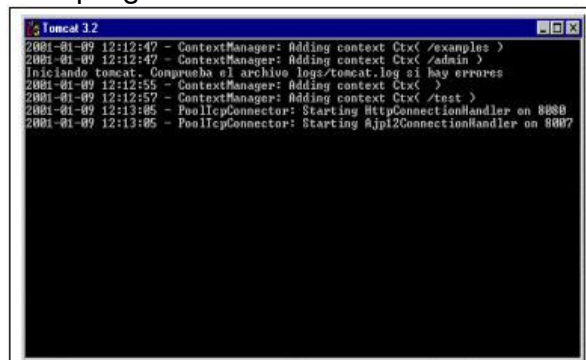
3.- Suponiendo que tenemos instalado correctamente tanto el JDK como el TOMCAT en las rutas adecuadas y que tenemos fijadas las variables de entorno típicas, el siguiente paso que tenemos que hacer es fijar una nueva variable de entorno dentro de nuestra máquina. Esta nueva variable va a recibir el nombre de JAVA\_HOME y debe de contener el valor de la ruta donde hemos instalado el JDK. Por ejemplo, si el JDK versión 1.2.2 lo hemos instalado en la siguiente ruta: C:\JDK1.2.2 la variable JAVA\_HOME debería de contener como valor el valor de esta ruta, es decir: JAVA\_HOME=C:\JDK1.2.2, desde NT la introducción de nuevas variables de entorno puede hacerlo con el comando SET del modo siguiente:

*SET JAVA\_HOME=C:\JDK1.2.2*

O bien usando las opciones de propiedades del icono de "Mi PC" y seleccionando "Entorno" como muestra la figura 1.



4.- El último paso que queda para finalizar la configuración de TOMCAT es la ejecución del archivo STARTUP.BAT que se encuentra en el directorio BIN de la carpeta TOMCAT, el cual iniciara la ejecución del programa sobre el puerto 8080. En las figuras siguientes tenéis el aspecto que presentará TOMCAT al usuario después de la llamada al programa STARTUP.BAT.



**Figura 2. Tomcat 3.2 en ejecución**

Para ver que Tomcat funciona perfectamente os recomiendo que trate de visualizar alguna página JSP en vuestro navegador WEB estandar y posteriormente retocarla para que el contenido de la misma sea WML, para ello os remito al articulo sobre programación en JSP para que vea como se crean páginas WML sobre JSP. No obstante puede comprobar que la instalación y configuración es correcta porque si abre un explorador y os conecte al puerto en cuestión (8080) aparecerá la página de bienvenida de TOMCAT tal y como podéis ver en la figura 3, desde esta pantalla tenemos acceso por un lado a la documentación tanto de JSP como de SERVLETS y a distintos ejemplos también relacionados sobre JSP y sobre SERVLETS que nos serán de utilidad cuando estemos haciendo nuestras pruebas.

Hay un aspecto importante que no debemos olvidar cuando trabajemos con TOMCAT y es que igual que tenemos un comando para iniciar el programa (STARTUP.BAT) cuando deseemos finalizar la ejecución del mismo debemos de hacerlo a través de otro programa también ubicado en el directorio bin de la aplicación y que recibe el nombre de SHUTDOWN.BAT.

Existen además dos archivos .XML que son los que almacenan la configuración de Tomcat y que reciben el nombre de SERVER.XML Y WEB.XML, el primero se encarga de especificar la configuración global de Tomcat mientras que el segundo se encarga de recoger aquellos parámetros que son opcionales. Seguramente abra muchos de vosotros que estarás pensando, si abriendo un navegador sobre un puerto y sin tener ningún servidor WEB instalado a priori TOMCAT es capaz de servirme páginas, entonces, ¿estoy ante un servidor WEB?, ¿para que quiero entonces instalar otro servidor WEB?, la respuesta a estas cuestiones es que si bien TOMCAT puede actuar como un servidor WEB, realmente esta pensado para enlazarse contra Apache, ya que TOMCAT carece de algunas características funcionales importantes como son:

Velocidad, TOMCAT es mucho más lento que Apache

Configuración, TOMCAT no es configurable en muchos de sus aspectos

ROBUSTED, TOMCAT no está pensado como servidor WEB sino como motor de JSP y SERVLET por tanto no debemos de sustituir a nuestro servidor WEB por TOMCAT.



Figura 3. Pantalla de inicio de Tomcat

## Introducción al Servidor de Aplicaciones Tomcat de Apache

## Introducción

TOMCAT es un contenedor de SERVLETS con un entorno JSP. Un contenedor de SERVLETS es un SHELL de ejecución que maneja e invoca SERVLETS por cuenta del usuario.

Podemos dividir los contenedores de SERVLETs en:

Contenedores de SERVLETS STAND-ALONE (Independientes)

Estos son una parte integral del servidor WEB. Este es el caso cuando usando un servidor WEB basado en Java, por ejemplo, el contenedor de SERVLETS es parte de JAWAWEBSERVER (actualmente sustituido por IPLANET). Este el modo por defecto usado por TOMCAT.

Sin embargo, la mayoría de los servidores, no están basados en Java, los que nos lleva los dos siguientes tipos de contenedores:

### **Contenedores de SERVLETs dentro-de-Proceso**

El contenedor SERVLET es una combinación de un plugin para el servidor WEB y una implementación de contenedor Java. El plugin del servidor WEB abre una JVM (Máquina Virtual Java) dentro del espacio de direcciones del servidor WEB y permite que el contenedor Java se ejecute en él. Si una cierta petición debería ejecutar un SERVLET, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando JNI). Un contenedor de este tipo es adecuado para servidores multi-thread de un sólo proceso y proporciona un buen rendimiento pero está limitado en escalabilidad

Contenedores de SERVLETs fuera-de-proceso

El contenedor SERVLET es una combinación de un plugin para el servidor WEB y una implementación de contenedor Java que se ejecuta en una JVM fuera del servidor WEB. El plugin del servidor WEB y el JVM del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets TCP/IP). Si una cierta petición debería ejecutar un SERVLET, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando IPCs). El tiempo de respuesta en este tipo de contenedores no es tan bueno como el anterior, pero obtiene mejores rendimientos en otras cosas (escalabilidad, estabilidad, etc.).

Tomcat puede utilizarse como un contenedor solitario (principalmente para desarrollo y depuración) o como plugin para un servidor WEB existente (actualmente se soporan los servidores Apache, IIS y Netscape). Esto significa que siempre que desplaguemos Tomcat tendremos que decidir cómo usarlo, y, si seleccionamos las opciones 2 o 3, también necesitaremos instalar un adaptador de servidor WEB

### **¿Cuál es la Diferencia entre Tomcat y Jserv?**

¿No es Tomcat == Jserv?

Es una confusión común, Jserv es un contenedor compatible con el API SERVLET 2.0 que fue creado para usarse con Apache. Tomcat es una re-escritura completa y es un contenedor compatible con los APIs SERVLET 2.2 y JSP 1.1.

Tomcat utiliza algún código escrito para Jserv, especialmente el adaptador de servidor para Apache, pero aquí se terminan todas las similitudes.

- ¿Cómo Instalar la Versión Binaria de Tomcat?

Muy sencillo. Deberíamos:

Descargar el archivo zip/tar.gz/ desde <http://jakarta.apache.org/downloads/binindex.html>.

Desempaquetamos el archivo en algún directorio (digamos foo). Esto debería crear un subdirectorio llamado jakarta-tomcat-3.2.1. Si no es el lugar que queremos, movemos este directorio a la localización deseada.

Cambiamos al directorio jakarta-tomcat-3.2.1 y configuramos una nueva variable de entorno (TOMCAT\_HOME) que apunte a la raíz de nuestro directorio Tomcat.

En Win32 deberíamos teclear:

```
set TOMCAT_HOME=foo\jakarta-tomcat-3.2.1
```

Sobre UNIX deberíamos teclear:

para bash/sh:

```
TOMCAT_HOME=foo/jakarta-tomcat-3.2.1 ;
```

```
export TOMCAT_HOME
```

para tcsh

```
setenv TOMCAT_HOME foo/jakarta-tomcat-3.2.1
```

Configuramos la variable de entorno JAVA\_HOME para que apunte al directorio raíz de nuestra instalación del JDK, luego añadimos el intérprete Java a nuestra variable de entorno PATH.

¡Esto es todo! Ahora podemos ejecutar Tomcat y se ejecutará como un contenedor de SERVLETs independiente (del tipo 1).

- Arrancar y Parar Tomcat

>

Arrancamos y paramos Tomcat usando los scripts que hay en el directorio bin:

Para arrancar Tomcat ejecutamos:

Sobre UNIX:

```
bin/startup.sh
```

Sobre Win32:

```
bin\startup
```

Para parar Tomcat ejecutamos:

Sobre UNIX:

```
bin/shutdown.sh
```

Sobre Win32:

```
bin\shutdown
```

- Como podemos ver, la versión Win32 de tomcat.bat no es tan robusta como la de UNIX. Especialmente, no se averigua los valores de JAVA\_HOME y sólo intenta "." como averiguación de TOMCAT\_HOME. Puede construir el CLASSPATH dinámicamente, pero no en todos los casos. No puede construir el CLASSPATH

dinámicamente si TOMCAT\_HOME contiene espacios, o sobre Win9x, si TOMCAT\_HOME contiene nombres de directorios que no son 8.3 caracteres.

## Archivos de Configuración de Tomcat

La configuración de Tomcat se basa en dos archivos:

server.xml - El archivo de configuración global de Tomcat.

WEB.xml - Configura los distintos contextos en Tomcat.

Esta sección trata la forma de utilizar estos archivos. No vamos a cubrir la interioridad de WEB.xml, esto se cubre en profundidad en la especificación del API SERVLET. En su lugar cubriremos el contenido de server.xml y discutiremos el uso de WEB.xml en el contexto de Tomcat.

### • server.xml

server.xml es el archivo de configuración principal de Tomcat. Sirve para dos objetivos:

Proporcionar configuración inicial para los componentes de Tomcat.

Especifica la estructura de Tomcat, lo que significa, permitir que Tomcat arranque y se construya a sí mismo ejemplarizando los componentes especificados en server.xml.

Se puede encontrar información adicional dentro del archivo server.xml.

Arrancar Tomcat desde Otros Directorio

Por defecto tomcat usará TOMCAT\_HOME/conf/server.xml para su configuración. La configuración por defecto usará TOMCAT\_HOME como la base para sus contextos.

Podemos cambiar esto usando la opción -f /path/to/server.xml, con un archivo de configuración diferente y configurando la propiedad home del controlador de contexto. Necesitamos configurar los archivos requeridos dentro del directorio home:

Un directorio WEBapps/ (si hemos creado uno) - todos los archivos war se expandirán y todos sus subdirectorios se añadirán como contextos.

Directorio conf/ - podemos almacenar tomcat-users.xml y otros archivos de configuración.

logs/ - todos los logs irán a este directorio en lugar de al principal

TOMCAT\_HOME/logs/.

work/ - directorio de trabajo para los contextos.

Si la propiedad ContextManager.home de server.xml es relativa, será relativa al directorio de trabajo actual.

### • WEB.xml

Podemos encontrar una detallada descripción de WEB.xml y la estructura de la aplicación WEB (incluyendo la estructura de directorios y su configuración) en los capítulos 9, 10 y 14 de la SERVLET API Spec en la site de Sun Microsystems.

Hay una pequeña característica de Tomcat que está relacionada con WEB.xml. Tomcat permite al usuario definir los valores por defecto de WEB.xml para todos los contextos poniendo un archivo WEB.xml por defecto en el directorio conf. Cuando construimos un nuevo contexto, Tomcat usa el archivo WEB.xml por

defecto como la configuración base y el archivo WEB.xml específico de la aplicación (el localizado en el WEB-INF/WEB.xml de la aplicación), sólo sobrescribe estos valores por defecto.

- Configurar Tomcat para Cooperar con Apache WEB Server

Hasta ahora no hemos explicado Tomcat como un plugin, en su lugar lo hemos considerado como un contenedor independiente y hemos explicado como usarlo. Sin embargo, hay algunos problemas:

Tomcat no es tan rápido como Apache cuando sirve páginas estáticas.

Tomcat no es tan configurable como Apache.

Tomcat no es tan robusto como Apache.

Hay mucho sites que llavan mucho tiempo de investigación sobre ciertos servidores WEB, por ejemplo, sites que usan scripts CGI o módulos perl o php... No podemos asumir que todos ellos quieran abandonar dichas tecnologías.

Por todas estas razones es recomendable que las sites del mundo real usen un servidor WEB, como Apache, para servir el contenido estático de la site, y usen Tomcat como un plugin para SERVLETs/JSP.

No vamos a cubrir las diferentes configuraciones en profundidad, en su lugar:

Cubriremos el comportamiento fundamental de un servidor WEB.

Explicaremos la configuración que necesitamos.

Demonstraremos esto sobre Apache.

- Operación del Servidor WEB

En resumidas cuentas un servidor WEB está esperando peticiones de un cliente HTTP. Cuando estas peticiones llegan el servidor hace lo que sea necesario para servir las peticiones proporcionando el contenido necesario. Añadirle un contenedor de SERVLETs podría cambiar de alguna forma este comportamiento. Ahora el servidor WEB también necesita realizar lo siguiente:

Cargar la librería del adaptador del contenedor de SERVLETs e inicializarlo (antes de servir peticiones).

Cuando llega una petición, necesita chequear para ver si una cierta petición pertenece a un SERVLET, si es así necesita permitir que el adaptador tome el control y lo maneje.

Por otro lado el adaptador necesita saber qué peticiones va a servir, usualmente basándose en algún patrón de la URL requerida, y dónde dirigir estas peticiones.

Las cosas son incluso más complejas cuando el usuario quiere seleccionar una configuración que use hosts virtuales, o cuando quieren que múltiples desarrolladores trabajen en el mismo servidor WEB pero en distintos contenedores de SERVLETs. Cubriremos estos dos casos en las secciones avanzadas.

### ¿Cuál es la Configuración Necesaria

La configuración más óbvia en la que uno puede pensar es en la identidad de las URLs SERVLET que están bajo la responsabilidad del contenedor de SERVLETs. Esto está claro, alguien debe conocer qué peticiones transmitir al cotenedor de **SERVLETs...**

Todavía hay algunos ítems de configuración adicionales que deberíamos proporcionar a la combinación WEB-server/SERVLET-container:

Necesitamos proporcionar la configuración sobre los procesos Tomcat disponibles y sobre los puertos/host TCP/IP sobre los que éstos están escuchando.

Necesitamos decirle al servidor WEB la localización de la librería adaptador (para que pueda cargarla en la arrancada).

Necesitamos seleccionar la información interna del adaptador sobre cuando log guardar, etc.

Toda esta información debe aparecer en el archivo de configuración del servidor WEB, o en un archivo de configuración privado usado por el adaptador. La siguiente sección explicará cómo se puede implementar esta configuración en Apache.

#### Haciéndolo en Apache

Esta sección nos enseña como configurar Apache para que trabaje con Tomcat; intenta proporcionar explicaciones sobre las directivas de configuración que deberíamos usar. Podemos encontrar información adicional en la página <http://java.apache.org/jserv/install/index.html>.

Cuando Tomcat arranque generará automáticamente un archivo de configuración para apache en TOMCAT\_HOME/conf/tomcat-apache.conf. La mayoría de las veces no necesitaremos hacer nada más que incluir es archivo (añadir Include TOMCAT\_HOME/conf/tomcat-apache.conf) en nuestro archivo httpd.conf. Si tenemos necesidades especiales, por ejemplo un puerto AJP distinto de 8007, podemos usar este archivo como base para nuestra configuración personalizada y grabar los resultados en otro archivo. Si manejamos nosotros mismos la configuración de Apache necesitaremos actualizarlo siempre que añadamos un nuevo contexto.

Tomcat: debemos re-arrancar tomcat y apache después de añadir un nuevo contexto; Apache no soporta cambios en su configuración sin re-arrancar. Cuando tomcat arranca, también se genera el archivo TOMCAT\_HOME/conf/tomcat-apache.conf cuando arrancar tomcat, por eso necesitamos arrancar Tomcat antes que Apache. Tomcat sobrescribirá TOMCAT\_HOME/conf/tomcat-apache.conf cada arrancada para que se mantenga la configuración peronalizada.

La configuración Apache-Tomcat usa las directivas de configuración principales de Apache así como directivas únicas de Jserv por eso podría ser confuso al principio, sin embargo hay dos cosas que lo simplifican:

En general podemos distinguir dos familias de directivas porque las directivas únicas de jserv empiezan con un prefijo ApJServ.

Toda la configuración relacionada con Tomcat está concentrada en un sólo archivo de configuración llamado tomcat.conf, el automáticamente generado tomcat-apache.conf, por eso podemos mirar en un sólo archivo.

Veamos ahora un simple archivo tomcat.conf:

```
#####  
#   A minimalistic Apache-Tomcat Configuration File   #  
#####
```

# Note: this file should be appended or included into your httpd.conf

# (1) Loading the jserv module that serves as Tomcat's apache adapter.  
LoadModule jserv\_module libexec/mod\_jserv.so

# (1a) Module dependent configuration.  
<IfModule mod\_jserv.c>

# (2) Meaning, Apache will not try to start Tomcat.  
ApJServManual on

# (2a) Meaning, secure communication is off  
ApJServSecretKey DISABLED

# (2b) Meaning, when virtual hosts are used, copy the mount  
# points from the base server  
ApJServMountCopy on

# (2c) Log level for the jserv module.  
ApJServLogLevel notice

# (3) Meaning, the default communication protocol is ajpv12  
ApJServDefaultProtocol ajpv12

# (3a) Default location for the Tomcat connectors.  
# Located on the same host and on port 8007

ApJServDefaultHost localhost  
ApJServDefaultPort 8007

# (4)  
ApJServMount /examples /root  
# Full URL mount  
# ApJServMount /examples ajpv12://hostname:port/root  
</IfModule>

Como podemos ver el proceso de configuración está dividido en 4 pasos que explicaremos ahora:

En este paso instruimos a Apache para que cargue el objeto compartido jserv (o la librería dll en NT). Esta es una directiva familiar de Apache. Si la carga fue bien y el módulo vino de un archivo llamado mod\_jserv.c (1a) podemos arrancar con el resto de la configuración Jserv-Tomcat.

Este paso configura varios parámetros internos de Jserv, estos parámetros son:  
Instruye a jserv para que no arranque el proceso Tomcat. Esto no está implementado todavía.

Desactiva la clave secreta challenge/response entre Apache y Tomcat. De nuevo, esto tampoco está implementado aún.

Instruye a jserv para que copie el punto de montaje del servidor base (ver siguiente sección) en caso de hosting virtual

Instruye a jserv para usar el nivel de log de noticia. Otros niveles de log incluidos son: emerg, alert, crit, error, warn, info y debug.

Este paso configura los parámetros de comunicación por defecto. Básicamente dice que el protocolo por defecto utilizado para la comunicación es ajpv12 que el



proceso Tomcat se ejecuta sobre la misma máquina y que escucha en el puerto 8807. Si ejecutamos Tomcat en una máquina distinta a la usada por Apache deberíamos actualizar `ApJServDefaultHost` o usar una URL completa cuando montemos los contextos. También, si configuramos los conectores Tomcat para usar un puerto distinto al 8007, deberíamos actualizar `ApJServDefaultPort` o usar una URL completa cuando montemos los contextos.

Este paso monta un contexto para Tomcat. Básicamente dice que todos los paths del servidor WEB que empiecen con `/example` irán a Tomcat. Este ejemplo `ApJServMount` es uno muy simple, de hecho, `ApJServMount` también puede proporcionar información sobre el protocolo de comunicación usado y la localización donde está escuchando el proceso Tomcat, por ejemplo:

```
ApJServMount /examples ajpv12://hostname:port/root
```

monta el contexto `/examples` en un proceso tomcat que se está ejecutando en el host `hostname` y que escucha en el puerto número `port`.

Ahora que hemos entendido las diferentes instrucciones de configuración en el archivo de ejemplo, ¿cómo podríamos añadirla a la configuración de Apache? Un método sencillo es escribir su contenido en `httpd.conf` (el archivo de configuración de Apache), sin embargo, esto puede ser muy desordenado. En su lugar deberíamos usar la directiva `include` de apache. Al final de archivo de configuración de Apache (`httpd.conf`) añadimos la siguiente directiva:

```
include <full path to the Tomcat configuration file>
```

Por ejemplo:

```
include /tome/tomcat/conf/tomcat.conf
```

Esto añadirá nuestra configuración de Tomcat a apache, después de haber copiado el módulo `jserv` al directorio `libexec` de Apache (o `modules` en caso de Win32) y re-arrancar (parar+arrancar) Apache, deberíamos poder conectar con Tomcat.

- Obtener el Módulo `Jserv` (`mod_jserv`)

Como vimos anteriormente, necesitamos un adaptador de servidor WEB para situarlo en Apache y redirigir las peticiones a Tomcat. Para Apache, este adaptador es una versión ligeramente modificada de `mod_jserv`.

Podríamos intentar buscarlo en <http://jakarta.apache.org/downloads/binindex.html> para ver si hay una versión pre-construida de `mod_jserv` que corresponda con nuestro sistema operativo (Normalmente hay uno para NT), sin embargo, siendo una librería nativa, no deberíamos esperar que esté ya (demasiados sistemas operativos, pocos desarrolladores, la vida es muy corta...) Además, pequeñas variaciones en la forma de construir la variante UNIX de Apache podrían resultar en errores de enlaces dinámicos. Realmente deberíamos intentar construir `mod_jserv` para nuestro sistema (no te asustes, no es tan difícil!).

## **CARACTERÍSTICAS DE JAVA**

### **Introducción**

No es arriesgado afirmar que Java supone un significativo avance en el mundo de los entornos software, y esto viene avalado por tres elementos claves que diferencian a este lenguaje desde un punto de vista tecnológico:

Es un lenguaje de programación que ofrece la potencia del diseño orientado a objetos con una sintaxis fácilmente accesible y un entorno robusto y agradable.

Proporciona un conjunto de clases potente y flexible.

Pone al alcance de cualquiera la utilización de aplicaciones que se pueden incluir directamente en páginas WEB (aplicaciones denominadas applets).

Java aporta a la WEB una interactividad que se había buscado durante mucho tiempo entre usuario y aplicación.

A lo largo de este apartado se estudian en detalle las principales características de Java.

### **Potente**

#### **a.) Orientación a objetos**

En este aspecto Java fue diseñado partiendo de cero, no siendo derivado de otro lenguaje anterior y no tiene compatibilidad con ninguno de ellos.

En Java el concepto de objeto resulta sencillo y fácil de ampliar. Además se conservan elementos "no objetos", como números, caracteres y otros tipos de datos simples.

#### **b.) Riqueza semántica**

Pese a su simpleza se ha conseguido un considerable potencial, y aunque cada tarea se puede realizar de un número reducido de formas, se ha conseguido un gran potencial de expresión e innovación desde el punto de vista del programador.

#### **c.) Robusto**

Java verifica su código al mismo tiempo que lo escribe, y una vez más antes de ejecutarse, de manera que se consigue un alto margen de codificación sin errores. Se realiza un descubrimiento de la mayor parte de los errores durante el tiempo de compilación, ya que Java es estricto en cuanto a tipos y declaraciones, y así lo que es rigidez y falta de flexibilidad se convierte en eficacia. Respecto a la gestión de memoria, Java libera al programador del compromiso de tener que controlar especialmente la asignación que de ésta hace a sus necesidades específicas. Este lenguaje posee una gestión avanzada de memoria llamada gestión de basura, y un manejo de excepciones orientado a objetos integrados. Estos elementos realizarán muchas tareas antes tediosas a la vez que obligadas para el programador.

#### **d.) Modelo de objeto rico**

Existen varias clases que contienen las abstracciones básicas para facilitar a los programas una gran capacidad de representación. Para ello se contará con un conjunto de clases comunes que pueden crecer para admitir todas las necesidades del programador.

Además la biblioteca de clases de Java proporciona un conjunto único de protocolos de Internet.

El conjunto de clases más complicado de Java son sus paquetes gráficos AWT (Abstract Window Toolkit) y Swing. Estos paquetes implementan componentes de una interfaz de usuario gráfica básica común a todos los ordenadores personales modernos.

### **Simple**

#### a.) Fácil aprendizaje

El único requerimiento para aprender Java es tener una comprensión de los conceptos básicos de la programación orientada a objetos. Así se ha creado un lenguaje simple (aunque eficaz y expresivo) pudiendo mostrarse cualquier planteamiento por parte del programador sin que las interioridades del sistema subyacente sean desveladas.

Java es más complejo que un lenguaje simple, pero más sencillo que cualquier otro entorno de programación. El único obstáculo que se puede presentar es conseguir comprender la programación orientada a objetos, aspecto que, al ser independiente del lenguaje, se presenta como insalvable.

#### b.) Completado con utilidades

El paquete de utilidades de Java viene con un conjunto completo de estructuras de datos complejas y sus métodos asociados, que serán de inestimable ayuda para implementar applets y otras aplicaciones más complejas. Se dispone también de estructuras de datos habituales, como pilas y tablas hash, como clases ya implementadas.

Existirá una interfaz Observer/Observable que permitirá la implementación simple de objetos dinámicos cuyo estado se visualiza en pantalla.

El JDK (Java Development Kit) suministrado por Sun Microsystems incluye un compilador, un intérprete de aplicaciones, un depurador en línea de comandos, y un visualizador de applets entre otros elementos.

### **Interactivo y orientado a red**

#### a.) Interactivo y animado

Uno de los requisitos de Java desde sus inicios fue la posibilidad de crear programas en red interactivos, por lo que es capaz de hacer varias cosas a la vez sin perder rastro de lo que debería suceder y cuándo. Para se da soporte a la utilización de múltiples hilos de programación (multithread).

Las aplicaciones de Java permiten situar figuras animadas en las páginas WEB, y éstas pueden concebirse con logotipos animados o con texto que se desplace por la pantalla. También pueden tratarse gráficos generados por algún proceso. Estas animaciones pueden ser interactivas, permitiendo al usuario un control sobre su apariencia.

#### b.) Arquitectura neutral

Java está diseñado para que un programa escrito en este lenguaje sea ejecutado correctamente independientemente de la plataforma en la que se esté actuando (Macintosh, PC, UNIX...). Para conseguir esto utiliza una compilación en una

representación intermedia que recibe el nombre de códigos de byte, que pueden interpretarse en cualquier sistema operativo con un intérprete de Java. La desventaja de un sistema de este tipo es el rendimiento; sin embargo, el hecho de que Java fuese diseñado para funcionar razonablemente bien en microprocesadores de escasa potencia, unido a la sencillez de traducción a código máquina hacen que Java supere esa desventaja sin problemas.

#### c.) Trabajo en red

Java anima las páginas WEB y hace posible la incorporación de aplicaciones interactivas y especializadas. Aporta la posibilidad de distribuir contenidos ejecutables, de manera que los suministradores de información de la WEB pueden crear una página de hipertexto (página WEB) con una interacción continuada y compleja en tiempo real; el contenido ejecutable es transferido literalmente al ordenador del usuario.

Los protocolos básicos para trabajar en Internet están encapsulados en unas cuantas clases simples. Se incluyen implementaciones ampliables de los protocolos FTP, HTTP, NNTP y SMTP junto con conectores de red de bajo nivel e interfaces de nombrado. Esto le permite interactuar con esos servicios de red poderosos sin tener que comprender realmente los detalles de bajo nivel de esos protocolos. Este lenguaje está diseñado para cumplir los requisitos de entrega de contenidos interactivos mediante el uso de applets insertados en sus páginas HTML. Además, las clases de Java admiten muy bien estos protocolos y formatos. El envío de las clases de Java a través de Internet se realiza con gran facilidad, ya que existe una interfaz unificada, resolviendo así los típicos problemas de diferencia de versiones.

Java proporciona un conjunto de clases para tratar con una abstracción de los conectores de red (sockets) originales de la versión UNIX de Berkeley, encapsular la noción de una dirección de Internet o conectar sockets con flujos de datos de Entrada/Salida.

Con todas estas posibilidades aumenta el dinamismo y competitividad de la WEB, puesto que es capaz de captar el interés del usuario durante largo tiempo y permite a los programadores convertir la WEB en un sistema de entrega de software.

#### d.) Applets

Una applet (miniaplicación) es un pequeño programa en Java transferido dinámicamente a través de Internet. Presentan un comportamiento inteligente, pudiendo reaccionar a la entrada de un usuario y cambiar de forma dinámica. Sin embargo, la verdadera novedad es el gran potencial que Java proporciona en este aspecto, haciendo posible que los programadores ejerzan un control sobre los programas ejecutables de Java que no es posible encontrar en otros lenguajes.

### **Y mucho más**

#### a.) Seguridad

Existe una preocupación lógica en Internet por el tema de la seguridad: virus, caballos de Troya, y programas similares navegan de forma usual por la red, constituyendo una amenaza palpable. Java ha sido diseñado poniendo un énfasis

especial en el tema de la seguridad, y se ha conseguido lograr cierta inmunidad en el aspecto de que un programa realizado en Java no puede realizar llamadas a funciones globales ni acceder a recursos arbitrarios del sistema, por lo que el control sobre los programas ejecutables no es equiparable a otros lenguajes.

Los niveles de seguridad que presenta son:

Fuertes restricciones al acceso a memoria, como son la eliminación de punteros aritméticos y de operadores ilegales de transmisión.

Rutina de verificación de los códigos de byte que asegura que no se viole ninguna construcción del lenguaje.

Verificación del nombre de clase y de restricciones de acceso durante la carga.

Sistema de seguridad de la interfaz que refuerza las medidas de seguridad en muchos niveles.

En futuras versiones se prevé contar también con encriptación y técnicas similares.

b.) Lenguaje basado en C++

Java fue desarrollado basándose en C++, pero eliminando rasgos del mismo poco empleados, optándose por una codificación comprensible. Básicamente, encontramos las siguientes diferencias con C++:

Java no soporta los tipos struct, union ni punteros.

No soporta typedef ni #define.

Se distingue por su forma de manejar ciertos operadores y no permite una sobrecarga de operadores.

No soporta herencia múltiple.

Java maneja argumentos en la línea de comandos de forma diversa a como lo hacen C o C++.

Tiene una clase String que es parte del paquete java.lang y se diferencia de la matriz de caracteres terminada con un nulo que usan C y C++.

Java cuenta con un sistema automático para asignar y liberar memoria, con lo que no es necesario utilizar las funciones previstas con este fin en C y C++.

c.) Gestión de la Entrada/Salida

En lugar de utilizar primitivas como las de C para trabajar con archivos, se utilizan primitivas similares a las de C++, mucho más elegantes, que permiten tratar los archivos, sockets, teclado y monitor como flujos de datos.

De este modo se pueden utilizar dichas primitivas para cualquier operación de Entrada/Salida.

d.) Diferentes tipos de aplicaciones

En Java podemos crear los siguientes tipos de aplicaciones:

Aplicaciones: Se ejecutan sin necesidad de un navegador.

Applets: Se pueden descargar de Internet y se observan en un navegador.

JavaBeans: Componentes software Java, que se puedan incorporar gráficamente a otros componentes.

JavaScript: Conjunto del lenguaje Java que puede codificarse directamente sobre cualquier documento HTML

SERVLETs: Módulos que permiten sustituir o utilizar el lenguaje Java en lugar de programas CGI (Common Gateway Interface) a la hora de dotar de interactividad a las páginas WEB.

## ANEXO K

### MANUAL DEL USUARIO

#### **Aspectos generales**

Existen muchas aplicaciones que soportan los requerimientos de la empresa en cuando a su incursión en Internet. Pero el común denominador en la mayoría de los casos, son desarrollos hechos a la medida por las mismas empresas, y según se indagó, esto corresponde básicamente a que los sistemas de información de cada empresa son bastante particulares, además que, en su mayoría cuentan con información confidencial, como lo son: los costos, las formulas de producción, entre otros.

Un Sistema de gestión de contenido (CMS\*) permite la creación y administración de contenidos principalmente en páginas WEB.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo público.

Portal es un término, sinónimo de puente, para referirse a un Sitio Web que sirve o pretende servir como un sitio principal de partida para las personas que se conectan al World Wide Web. Son sitios que los usuarios tienden a visitar como sitios ancla. Los portales tienen gran reconocimiento en Internet por el poder de influencia que tienen sobre grandes comunidades.

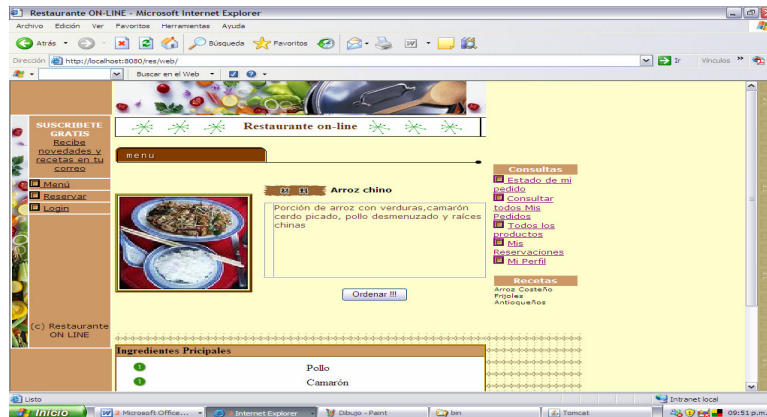
---

\* Content Management System, en inglés, abreviado CMS

## CAPÍTULO: PERFIL GENERAL

### Página principal

La página principal es la interfaz que tanto los clientes como los usuarios consultan

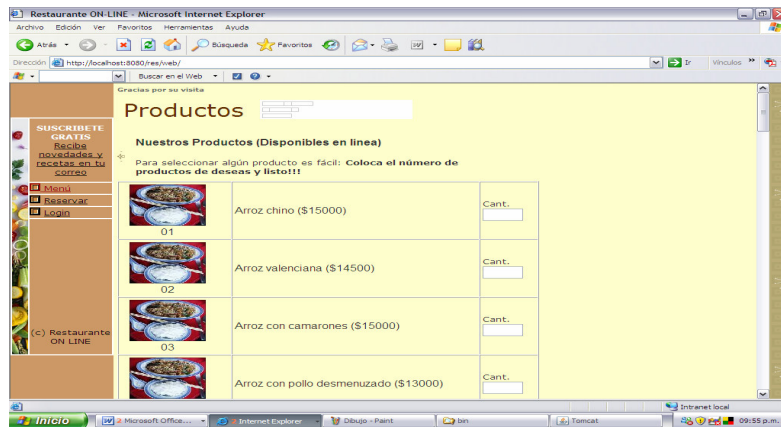


Dicha pantalla esta estructurada de tal forma que los usuarios puedan observar el menú de productos, realizar reservaciones, consultar el estado de pedidos y administrar el sitio (si es el usuario adecuado; por ejemplo en el caso observar los productos que están disponibles en línea se hace clic en el enlace correspondiente, tal y como lo indica el siguiente gráfico:



En el momento de hacer clic en dicho enlace aparecerá la siguiente figura:

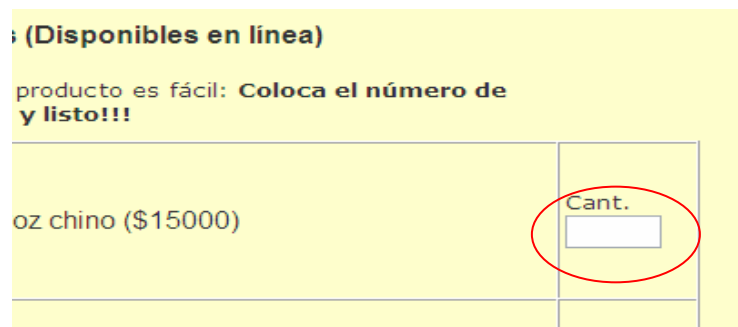




Donde cada uno de los productos se pueden consultar en forma individual en otro aparte del software.

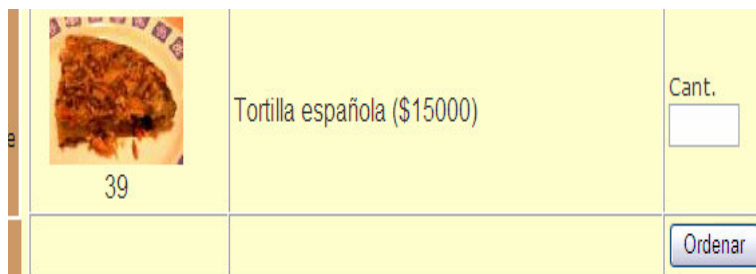
### Toma de Pedidos

Para la toma de pedidos es necesario indicar la cantidad de productos a solicitar, para tales efectos se encuentra una casilla habilitada para indicar dicho dato, observemos la siguiente figura:



La anterior figura ilustra el campo habilitado para indicar la cantidad de productos para cada ítem

En el momento de hacer el pedido es necesario presionar el botón adecuado, observar la siguiente figura:



Ésta nos indica con rojo en donde el usuario debe hacer clic para iniciar el pedido.

Después de realizar dicha operación aparecerá el registro en el sistema, debido a que cualquier persona no puede hacer pedidos (sólo clientes registrados)

La pantalla que aparecerá para iniciar el proceso es la siguiente:



Gracias por su visita

# Productos

Validación de Clientes

[¿desea registrarse?](#)

**Usuarios Registrados**

Usuario

Clave

Aceptar

Como se indicó en párrafos anteriores, al sistema se le ingresa la información adecuada para tales efectos, es entonces como se valida el ingreso, después de ingresar la información correcta se muestra la siguiente pantalla:



Gracias por su visita

# Productos

Área de Clientes

Forma de Pago

Acompañamiento

Confirmar >>

El sistema permite seleccionar la forma de pago (informativo), el acompañamiento es una forma de darle la posibilidad al cliente de ordenar algo adicional.

Para confirmar el pedido se hace clic en el botón, observar la siguiente figura:



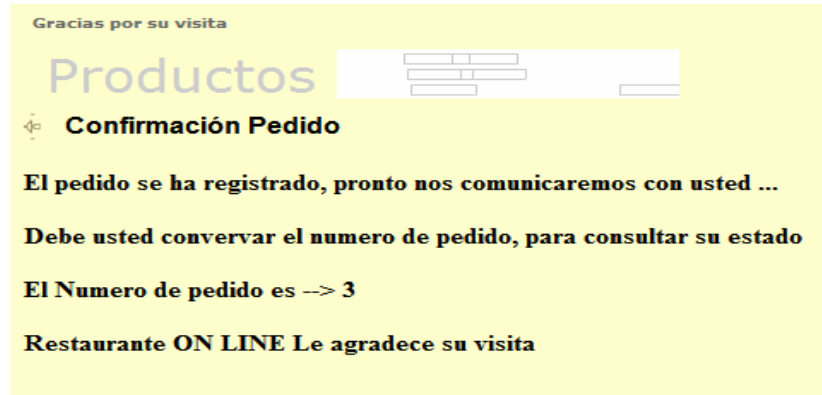
Área de Clientes

Forma de Pago

Acompañamiento

Confirmar >>

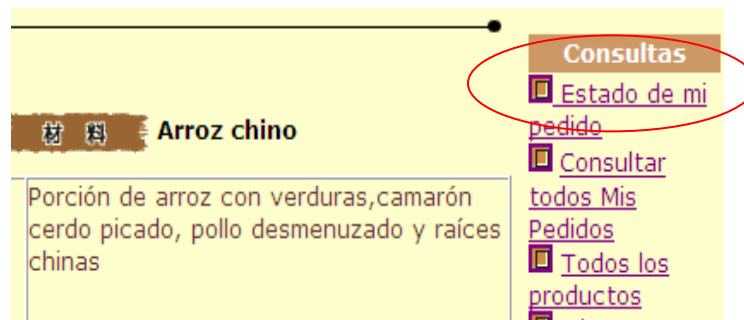
Después de hacer clic aparecerá la orden, cabe anotar que es de vital importancia que sea anotado el número de pedido, para más adelante poder consultar el estado del pedido; la información que aparece esta indicada en la siguiente figura:



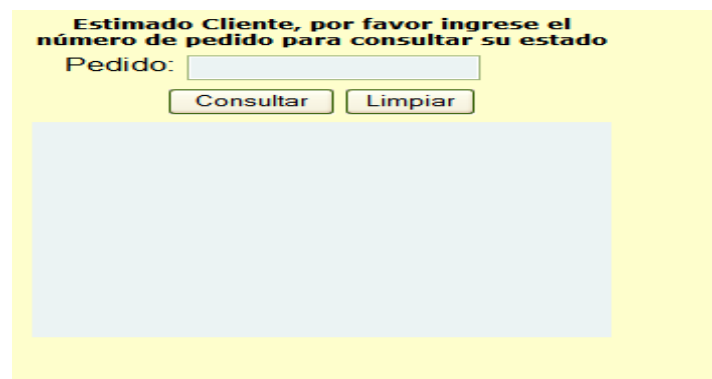
No olvidar tener en cuenta el número de pedido asignado.

### Consultar el estado de mi pedido

Una de las formas requeridas para la consulta de los pedidos, por parte de los clientes está en la página principal, observe el siguiente gráfico:



Al momento de hacer clic en dicha pantalla aparecerá la siguiente ventana:



Donde se debe ingresar el número del pedido para consultar su estado, observe el siguiente gráfico que ilustra un ejemplo:

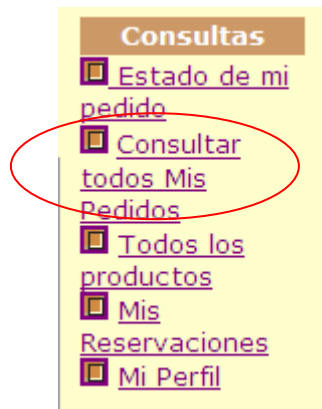
Estimado Cliente, por favor ingrese el número de pedido para consultar su estado

Pedido:

Estimado Cliente, el estado de su pedido se encuentra actualmente en

### Consultar todos mis pedidos

En caso de querer conocer todos mis pedidos se debe hacer clic en el enlace que indica el siguiente gráfico:

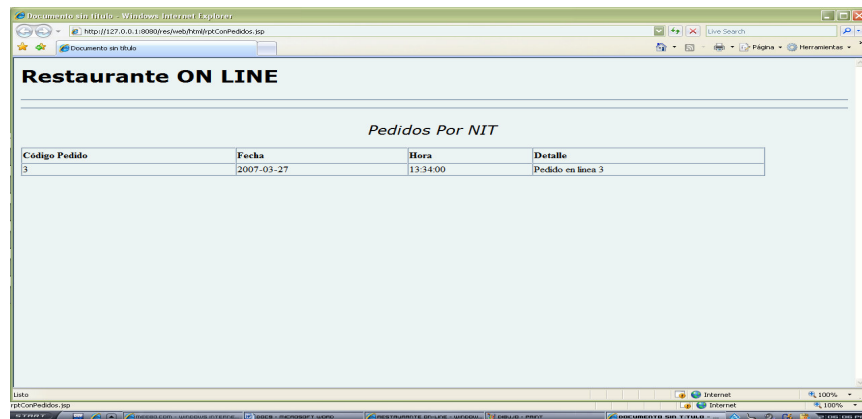


Siguiente a ello aparecerá la siguiente figura:

Estimado Cliente, por favor ingrese el número de identificación para consultar su información

Código:

Donde es necesario suministra el código de cliente con el cual se registro. Cuando es procedimiento se realiza, el sistema le mostrará los pedidos realizados, observe la siguiente pantalla:

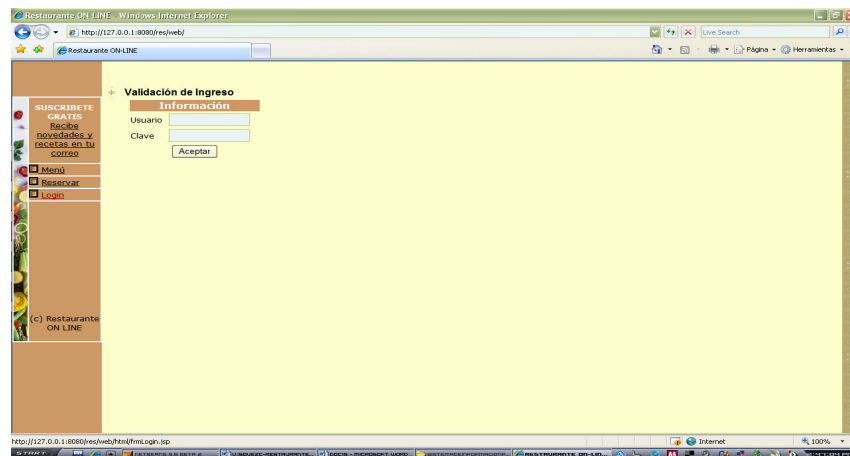


The screenshot shows a web browser window titled "Restaurante ON LINE". The page displays a table of orders under the heading "Pedidos Por NIT". The table has four columns: "Código Pedido", "Fecha", "Hora", and "Detalle". There is one row of data.

Código Pedido	Fecha	Hora	Detalle
3	2007-03-27	13:34:00	Pedido en línea 3

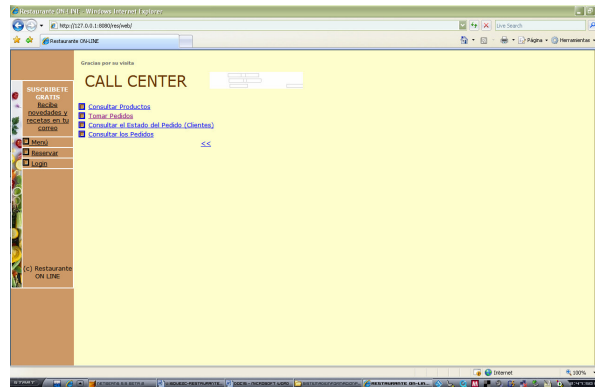
## CAPÍTULO: PERFIL CALLCENTER

El usuario del CALL CENTER esta orientado a la utilización del sistema como intermediario del cliente cuando éste llama a realizar los pedidos, todo inicia cuando se ingresa al sistema por medio de la siguiente interfaz:



The screenshot shows a web browser window titled "Restaurante ON LINE". The page displays a login form with the heading "Validación de Ingreso". The form includes a "Usuario" field, a "Clave" field, and an "Aceptar" button. On the left side, there is a sidebar with a navigation menu containing links for "SUSCRIBETE GRATIS", "Recibe novedades y recetas en tu correo", "Menú", "Reservar", and "Login".

Cuando se ingresa la el login y la clave adecuadas, aparece la siguiente pantalla:

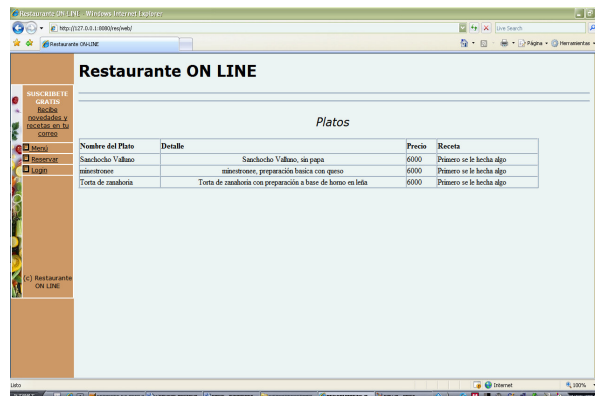


La anterior pantalla cuenta con cuatro opciones, las cuales se muestran a continuación:



## Consultar Productos

La primera opción es Consultar Productos, la cual muestra la siguiente pantalla:



## Toma de Pedidos

Para la toma de pedidos es necesario indicar la cantidad de productos a solicitar, para tales efectos se encuentra una casilla habilitada para indicar dicho dato, observemos la siguiente figura:


**(Disponibles en línea)**

producto es fácil: **Coloca el número de y listo!!!**

oz chino (\$15000)	Cant. <input type="text"/>
--------------------	-------------------------------

La anterior figura ilustra el campo habilitado para indicar la cantidad de productos para cada ítem

En el momento de hacer el pedido en necesario presionar el botón adecuado, observar la siguiente figura:

 39	Tortilla española (\$15000)	Cant. <input type="text"/>
		<input type="button" value="Ordenar"/>

Ésta nos indica con rojo en donde el usuario debe hacer clic para iniciar el pedido.

Después de realizar dicha operación aparecerá el registro en el sistema, debido a que cualquier persona no puede hacer pedidos (sólo clientes registrados)

La pantalla que aparecerá para iniciar el proceso es la siguiente:

Gracias por su visita

## Productos

[¿desea registrarse?](#)

**Usuarios Registrados**

Usuario

Clave

Como se indicó en párrafos anteriores, al sistema se le ingresa la información adecuada para tales efectos, es entonces como se valida el ingreso, después de ingresar la información correcta se muestra la siguiente pantalla:

Gracias por su visita

# Productos

Área de Clientes

Forma de Pago: Transferencia Bancaria

Acompañamiento: Platos desechables

Confirmar >>

El sistema permite seleccionar la forma de pago (informativo), el acompañamiento es una forma de darle la posibilidad al cliente de ordenar algo adicional.

Para confirmar el pedido se hace clic en el botón, observar la siguiente figura:

Área de Clientes

Forma de Pago: Transferencia Bancaria

Acompañamiento: Platos desechables

Confirmar >>

Después de hacer clic aparecerá la orden, cabe anotar que es de vital importancia que sea anotado el número de pedido, para más adelante poder consultar el estado del pedido; la información que aparece esta indicada en la siguiente figura:

Gracias por su visita

## Confirmación Pedido

El pedido se ha registrado, pronto nos comunicaremos con usted ...

Debe usted conservar el numero de pedido, para consultar su estado

El Numero de pedido es --> 3

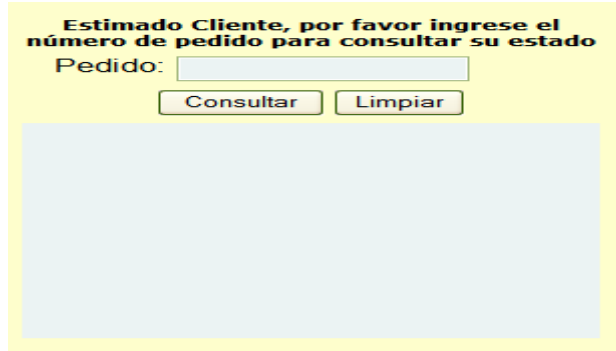
Restaurante ON LINE Le agradece su visita

No olvidar tener en cuenta el número de pedido asignado.



## Consultar estados de pedidos (Clientes)

Para consultar el estados de los pedidos para los clientes, en el momento en que el cliente se comunica con el CALL CENTER, es el siguiente:



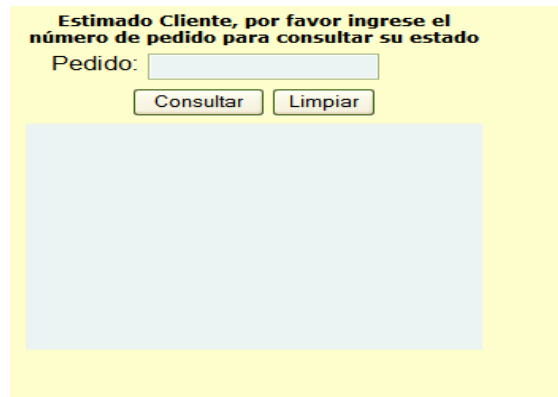
Estimado Cliente, por favor ingrese el número de pedido para consultar su estado

Pedido:

A large light blue rectangular area is present below the buttons, which is currently empty.

## Consultar los pedidos

Al momento de hacer clic en dicha pantalla aparecerá la siguiente ventana:

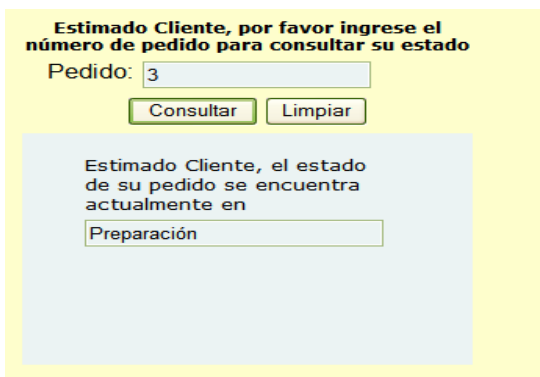


Estimado Cliente, por favor ingrese el número de pedido para consultar su estado

Pedido:

A large light blue rectangular area is present below the buttons, which is currently empty.

Donde se debe ingresar el número del pedido para consultar su estado, observe el siguiente gráfico que ilustra un ejemplo:



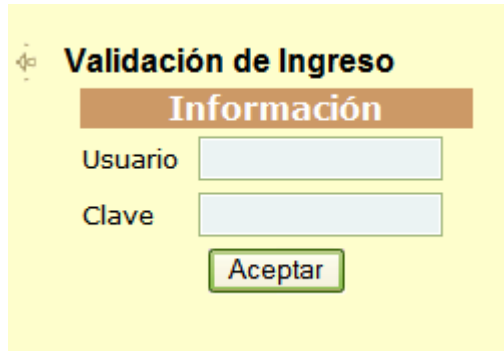
Estimado Cliente, por favor ingrese el número de pedido para consultar su estado

Pedido:

Estimado Cliente, el estado de su pedido se encuentra actualmente en

## CAPÍTULO: PERFIL ADMINISTRATIVO DE SITIO

Para la administración es necesario contar una clave apropiada para tales efectos, el ingreso se hace por medio de la siguiente interfaz:



The image shows a login validation form titled "Validación de Ingreso". It features a header "Información" in a brown box. Below this, there are two input fields: "Usuario" and "Clave", each with a light blue rectangular box. At the bottom of the form is a green button labeled "Aceptar".

Luego de suministrar dicha información aparecerá el siguiente menú:



The image shows a vertical menu titled "Sistema Administrador del Sitio" in a brown box. Below the title, there are eight menu items, each in a green box with a small square icon to its right: "Restaurantes", "Clientes", "Paises", "Sectores", "Tipos de Cliente", "Tipos de Platos", "Estados", and "Nivel de Acceso".

Donde cada una de éstas opciones son utilizadas para ingresar, buscar, eliminar y modificar las opciones de cada elemento. Para efectos explicativos se mostrará un ejemplo que debe ser tomado para todas las opciones.

### Ingresar clientes.

Al momento de ingresar clientes e hace clic sobre la opción adecuada para tales efectos, el menú que se muestra para lograr el objetivo es el siguiente:



Y en el momento de hacer clic sobre Crear Clientes aparecerá la siguiente opción.

A screenshot of a web application form titled "Registro de Clientes". The form is displayed on a light yellow background with a brown header bar containing the title. Below the header, there are several input fields and dropdown menus: "Código" (text input), "Nombre" (text input), "Dirección" (text input), "Teléfono" (text input), "Correo-e" (text input), "Moneda" (dropdown menu with "Pesos 01" selected), "País" (dropdown menu with "Colombia" selected), "Tipo Cliente" (dropdown menu with "Usuario del Sistema" selected), "Sector" (dropdown menu with "Agricola" selected), "Clave" (text input), and "Repetir Clave" (text input). At the bottom of the form, there are two buttons: "Ingresar" and "Limpiar". Below the buttons, there is a blue double-left arrow icon.

El botón ingresar será el encargado de llevar la información a la base de datos.

Buscar clientes.

La opción buscar clientes nos lleva a la siguiente pantalla:

A screenshot of a web application form titled "Consulta de Clientes". The form is displayed on a light yellow background with a brown header bar containing the title. Below the header, there is a text input field labeled "NIT". Below the input field, there are two buttons: "Consultar" and "Limpiar". Below the buttons, there is a blue double-left arrow icon. The bottom half of the form is a large, empty light blue rectangular area.

La cual nos indica que debemos suministrar el cliente a consultar para lograr éxito en dicha operación.

### **Modificar un cliente.**

En el momento de modificar un cliente aparecerá una interfaz similar a la de consulta, esto debido a que para modificar un cliente es necesario consultarlo, ésta figura se muestra a continuación.

Modificación de Clientes

NIT

Modificar Limpiar

<<

Luego de suministrar el NIT sobre el cual se requiere hacer la modificación aparecerá un botón para tales efectos.

### **Eliminar clientes.**

En el momento de eliminar clientes, es necesario ingresar a la opción pertinente en el menú, ésta desplegará la siguiente pantalla:

Eliminar Clientes

NIT

Eliminar Limpiar

<<

En el momento de suministrar el NIT, éste validará si existencia, en caso tal, en el recuadro aparecerá si fue eliminado con éxito o el informe de su inexistencia.

## **CAPÍTULO: PERFIL ADMINISTRATIVO DE RESTAURANTE**

Para la administración es necesario contar una clave apropiada para tales efectos, el ingreso se hace por medio de la siguiente interfaz:

Validación de Ingreso

Información

Usuario

Clave

Aceptar

Luego de suministrar dicha información aparecerá el siguiente menú:



### CAPÍTULO: PERFIL CONTROL A DESPACHOS A DOMICILIO

Para efectos administrativos, el control de despachos es usado con el fin de adelantar la gestión comercial, el menú se muestra a continuación:



### Generar Orden de Despacho

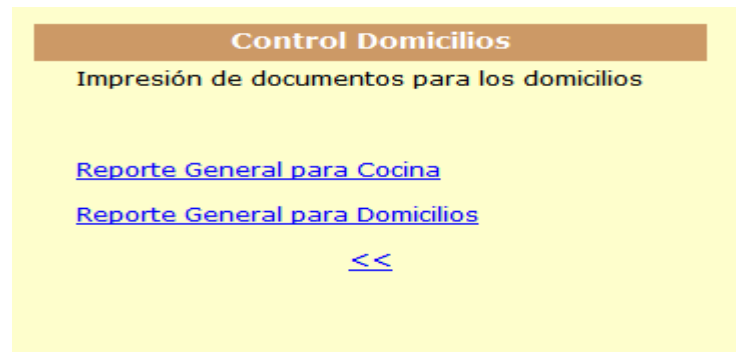
Es una opción para diligenciar los procesos en las ordenes de despacho, obsérvese el siguiente gráfico:



Como se observa, es por ésta opción se logran todos los procesos de administración respecto a despachos.

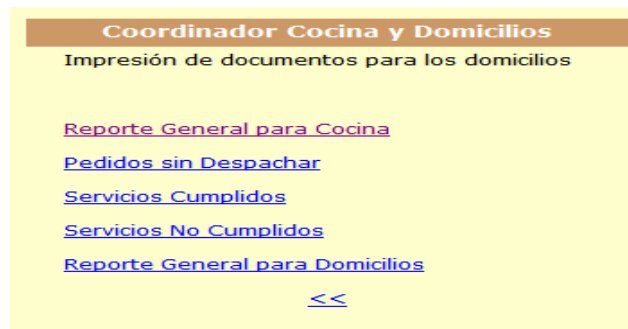
### **Imprimir reportes para Domicilio**

Es la herramienta que se usa como medio físico para que los empleados logren a cabalidad el oficio de producción y comercialización productos alimenticios. Las opciones para éste proceso aparecerán en la siguiente pantalla:



### **CAPÍTULO: PERFIL COORDINADOR COCINA Y DOMICILIOS**

Dicho perfil mostrará el siguiente menú:



Donde se tiene todas las opciones necesarias y requeridas para coordinar las opciones en la gestión de los pedidos, en éste menú se registran los pedidos que se envían al cliente, tanto para el momento de salida de los productos, como para el registro de efectividad del servicio, ya sea cumplido o no cumplido.