

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERIA DE SISTEMAS**

**Desarrollo de una aplicación Web para Servicio de entrega de  
comida a domicilio.**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**AUTORES:**

**SALAZAR PAREDES ESTEBAN FRANCISCO  
esteban\_francisco8@hotmail.com**

**SANCHEZ MACAS FREDY FABIAN  
fredy88\_san@hotmail.com**

**DIRECTOR: FRANCISCO VILLAVICENCIO  
fvillavi@yahoo.com**

**Quito, Enero 2013**

## **DECLARACIÓN**

Nosotros, Esteban Francisco Salazar Paredes y Fredy Fabian Sánchez Macas, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

**Esteban Salazar**

---

**Fredy Sánchez**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Esteban Francisco Salazar Paredes y Fredy Fabian Sánchez Macas, bajo mi supervisión.

---

**Francisco Villavicencio**

**DIRECTOR DE PROYECTO**

## AGRADECIMIENTOS

A Dios por la vida y bendiciones que me ha concedido ya que por su gracia he conseguido lo que tengo.

A mi madrecita linda y mi padre por su respaldo incondicional. A mí enamorada Jimena por estar junto a mí durante el transcurso de mis estudios. A mis tíos y ñaños gracias por su apoyo en todo momento. A mis hermanos y hermanas por ser la fuerza que me impulsa a salir adelante. A mis abuelitos por sus enseñanzas de vida y ganas que le ponen en su trabajo. A Fredy por el esfuerzo para la culminación de este proyecto.

A toda mi familia gracias las enseñanzas de vida. Por los momentos que he compartido con mis compañeros de clase, del futbol, de mi vida estudiantil.

Agradezco por todos los consejos que me han brindado ya que me han servido para proponerme metas que paso a paso se espero ir alcanzando.

Esteban

## **AGRADECIMIENTOS**

Quiero agradecer primero a Dios por brindarme fuerzas a cumplir las metas propuestas.

A mis padres por darme el apoyo necesario para desenvolverme en la vida y pueda dar termino a mi carrera profesional.

A mi tutor por darnos las guías para desarrollar la tesis de la mejor manera, y por brindarnos su tiempo con sus consejos y pautas.

A mis amigos que durante la carrera supimos apoyarnos para comprender los conceptos y por brindarme su sincera amistad.

A mi compañero de tesis por haber puesto dedicación y empeño para la consecución de este proyecto.

Fredy

## ÍNDICE

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA .....	1
1.1 DESCRIPCIÓN DEL PROBLEMA.....	1
1.1.1 MODELO DE NEGOCIO.....	1
1.1.2 PROCESO DE VENTAS A DOMICILIO.....	9
1.1.3 PROCESO DE DISTRIBUCIÓN DEL PEDIDO A DOMICILIO.....	11
1.1.4 PROCESO DE MARKETING.....	11
1.1.5 PROPUESTA DE AUTOMATIZACIÓN.....	13
1.2 JUSTIFICACIÓN DE LA METODOLOGÍA DE DESARROLLO.....	13
1.2.1 ICONIX.....	14
1.2.2 JUSTIFICACIÓN DE LA METODOLOGÍA SELECCIONADA.....	17
1.3 SELECCIÓN DE LA PLATAFORMA TECNOLÓGICA.....	17
1.3.1 PLATAFORMA DE DESARROLLO .....	18
1.3.2 BASE DE DATOS.....	20
1.3.3 SERVIDOR WEB DE APLICACIONES.....	23
1.3.4 ENTORNO DE DESARROLLO.....	25
CAPÍTULO II: DESARROLLO DEL SISTEMA .....	26
2.1 ANÁLISIS DEL SISTEMA .....	26
2.1.1 REQUERIMIENTOS FUNCIONALES DEL SISTEMA.....	26
2.1.2 MODELO DE DOMINIO.....	29
2.1.3 CASOS DE USO.....	32
2.1.4 DESCRIPCIÓN DE CASOS DE USO.....	35
2.1.5 PROTOTIPO DE INTERFACES .....	42
2.1.6 REVISIÓN DE REQUERIMIENTOS .....	50
2.2 DISEÑO DEL SISTEMA.....	51
2.2.1 DISEÑO PRELIMINAR .....	51
2.2.2 DISEÑO DETALLADO.....	58
2.3 IMPLEMENTACIÓN DEL SISTEMA .....	63
2.3.1 CÓDIGO DEL SISTEMA.....	63
2.4 PRUEBAS .....	80

2.4.1	PLAN DE PRUEBAS .....	80
CAPÍTULO III: EVALUACIÓN DEL SISTEMA CON UN CASO DE ESTUDIO.....		88
3.1	RECOLECCIÓN DE INFORMACIÓN SOBRE EL CASO DE ESTUDIO .	88
3.1.1	APRESTAMIENTO EMPRESARIAL .....	88
3.1.2	PROCESO DE ATENCIÓN AL CLIENTE .....	91
3.1.3	PROCESO DE DISTRIBUCIÓN DEL PRODUCTO .....	92
3.1.4	PROCESO DE MARKETING.....	92
3.1.5	PRECIOS.....	93
3.2	PROPUESTA DE IMPLANTACIÓN DE LA APLICACIÓN. ....	95
3.2.1	INSTALACIÓN DEL SISTEMA .....	95
3.2.2	CARGA DE DATOS.....	96
3.3	PRUEBAS DE CARGA Y STRESS.....	105
3.3.1	PRUEBAS DE CARGA .....	105
3.3.2	PRUEBAS DE STRESS.....	106
3.3.3	PROCESO DE PRUEBAS .....	106
3.3.4	EJECUCIÓN DE LAS PRUEBAS DE CARGA.....	109
3.3.5	EJECUCIÓN DE LAS PRUEBAS DE STRESS .....	109
3.4	ANÁLISIS DE RESULTADOS.....	110
3.4.1	ANÁLISIS DE LAS PRUEBAS DE CARGA .....	110
3.4.2	ANÁLISIS DE LAS PRUEBAS DE STRESS.....	124
CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES. ....		128
4.1	CONCLUSIONES.....	128
4.2	RECOMENDACIONES .....	130
BIBLIOGRAFÍA .....		131
ANEXOS .....		135

## ÍNDICE DE TABLAS

Tabla 1. 1 Formato de Descripción de un Proceso .....	9
Tabla 1. 2 Descripción del Proceso de Ventas a Domicilio .....	10
Tabla 1. 3 Descripción del Proceso de Distribución del Pedido a Domicilio .....	11
Tabla 1. 4 Descripción del Proceso de Publicidad .....	12
Tabla 2. 1 Caso de uso RegistrarUsuarioAdministradorProveedor .....	36
Tabla 2. 2 Caso de uso RealizarPedido .....	39
Tabla 2. 3 Caso de uso VerCarroCompras .....	40
Tabla 2. 4 Caso de uso EditarParametrosGeneralesProveedor.....	41
Tabla 2. 5 Caso de uso IngresarProducto .....	42
Tabla 2. 6 Definición de estilos para las páginas del sistema .....	66
Tabla 2. 7 Pruebas unitarias de los métodos de negocio .....	84
Tabla 2. 8 Pruebas de Validación de páginas web.....	87
Tabla 3. 1 Menú Restaurante “Piguari” .....	94
Tabla 3. 2 Requerimientos de HW .....	95
Tabla 3. 3 Requerimientos de SW.....	95
Tabla 3. 4 Objetos, contadores y valores umbrales del servidor [38] .....	110



## ÍNDICE DE FIGURAS

Figura 1. 1 Organigrama para una empresa de comida a domicilio. ....	6
Figura 1. 2 Mapa de Procesos para empresa de comida a domicilio. ....	8
Figura 1. 3 Arquitectura JEE .....	19
Figura 2. 1 Modelo de dominio .....	31
Figura 2. 2 Diagrama de casos de uso para el administrador del sistema .....	32
Figura 2. 3 Diagrama de casos de uso para el administrador del sistema .....	33
Figura 2. 4 Diagrama de casos de uso para el cajero del proveedor .....	34
Figura 2. 5 Diagrama de casos de uso para el cajero del proveedor .....	35
Figura 2. 6 Pantalla de Ingreso al sistema .....	42
Figura 2. 7 Prototipo página principal.....	43
Figura 2. 8 Pantalla para Registrarse en el sistema.....	43
Figura 2. 9 Prototipo Pantalla para Editar Cuenta Usuario.....	44
Figura 2. 10 Prototipo vista carrito de compras.....	44
Figura 2. 11 Prototipo pantalla para realizar el pedido.....	45
Figura 2. 12 Prototipo pantalla de listado de pedidos pendientes. ....	45
Figura 2. 13 Prototipo pantalla de confirmación del pedido. Asignar repartidor y hora de entrega.....	46
Figura 2. 14 Prototipo pantalla Consultar Pedidos .....	46

Figura 2. 15 Prototipo de pantalla para ingresar un producto .....	47
Figura 2. 16 Prototipo de pantalla para ingresar una nueva categoría.....	47
Figura 2. 17 Prototipo de pantalla para ingresar empleados.....	48
Figura 2. 18 Prototipo de pantalla para Modificar estado de disponibilidad y publicidad de productos .....	48
Figura 2. 19 Prototipo de pantalla para asignar horario de trabajo a repartidores.	49
Figura 2. 20 Prototipo de pantalla para Ingresar un nuevo proveedor. ....	49
Figura 2. 21 Prototipo de pantalla para crear una cuenta de administrador proveedor.....	50
Figura 2. 22 Prototipo de pantalla para editar la vigencia para un proveedor en el sistema.....	50
Figura 2. 23 Análisis de Robustez para el Caso de Uso registrarUsuarioAdministradorProveedor .....	51
Figura 2. 24 Análisis de Robustez para el Caso de Uso RealizarPedido .....	52
Figura 2. 25 Análisis de robustez para el caso de uso VerCarroCompras .....	53
Figura 2. 26 Análisis de Robustez para el Caso de Uso EditarParametroGeneralesProveedor .....	53
Figura 2. 27 Análisis de Robustez para el Caso de Uso IngresarProducto.....	54
Figura 2. 28 Arquitectura de la aplicación shopping food.....	55
Figura 2. 29 Revisión del diseño preliminar del modelo de dominio.....	57
Figura 2. 30 Diagrama de secuencia para el caso de uso RegistrarUsuarioAdministradorProveedor .....	58

Figura 2. 31 Diagrama de secuencia para el caso de uso RealizarPedido .....	59
Figura 2. 32 Diagramas de secuencia para el caso de uso VerCarroCompras....	59
Figura 2. 33 Diagrama de secuencia para el caso de uso Editar Parámetros Generales Proveedor .....	60
Figura 2. 34 Diagrama de secuencia para el caso de uso Ingresar Producto .....	60
Figura 2. 35 Diseño del diagrama de clases .....	62
Figura 2. 36 Entidad-Relación de la Base de datos .....	64
Figura 2. 37 Sección urn:jboss:domain:web:1.1 del archivo Standalone.xml.....	72
Figura 2. 38 Configuración del Conector https en Jboss .....	72
Figura 2. 39 Definición del Puerto https para Jboss .....	72
Figura 2. 40 Autorización de Spring Security .....	76
Figura 2. 41 Pruebas Unitarias de Pedido con JUnit.....	85
Figura 2. 42 Pruebas Unitarias de Producto con JUnit.....	85
Figura 3. 1 Organigrama funcional Restaurante “Piguari” .....	89
Figura 3. 2 Publicidad de promociones restaurante “Piguari” .....	92
Figura 3. 3 Publicidad de productos de mayor acogida restaurante “Piguari” .....	93
Figura 3. 4 Login en la aplicación como administrador del sitio .....	96
Figura 3. 5 Advertencia del browser del certificado para HTTPS .....	97
Figura 3. 6 Agregación del certificado de la aplicación al browser. ....	97

Figura 3. 7 Creación del proveedor en el administrador del sitio.....	98
Figura 3. 8 Creación del usuario administrador proveedor del restaurante “Piguari”.....	98
Figura 3. 9 Login como usuario administrador proveedor .....	99
Figura 3. 10 Pantalla principal del administrador del restaurante “Piguari”. .....	99
Figura 3. 11 Actualización de los parámetros generales del restaurante “Piguari”. .....	100
Figura 3. 12 Creación de las categorías del restaurante “Piguari”. .....	100
Figura 3. 13 Listado de categorías del proveedor “Piguari”.....	101
Figura 3. 14 Formulario de creación de Productos.....	101
Figura 3. 15 Menú de creación de empleados del restaurante “Piguari”. .....	102
Figura 3. 16 Formulario de creación de empleado Cajero. ....	102
Figura 3. 17 Formulario de creación de empleado Repartidor. ....	103
Figura 3. 18 Formulario de asignación de horario a empleado. ....	103
Figura 3. 19 Pantalla principal del módulo de atención al cliente de shopping food. .....	104
Figura 3. 20 Pantalla principal de atención al cliente para el restaurante Piguari. .....	104
Figura 3. 21 Primer Paso para grabación en el Proxy escenario 3 .....	107
Figura 3. 22 Segundo Paso para grabación en el Proxy .....	108
Figura 3. 23 Resultados de grabación del escenario 3. ....	108

Figura 3. 24 Tiempo de Respuesta escenario Realizar Pedido para 25 Usuarios .....	111
Figura 3. 25 Carga procesada por segundo escenario Realizar Pedido para 25 Usuarios .....	111
Figura 3. 26 Uso de memoria escenario Realizar Pedido para 25 Usuarios .....	112
Figura 3. 27 Uso del CPU escenario Realizar Pedido para 25 Usuarios.....	112
Figura 3. 28 Tiempo de Respuesta escenario Realizar Pedido para 50 Usuarios .....	113
Figura 3. 29 Carga procesada por segundo escenario Realizar Pedido para 50 Usuarios .....	113
Figura 3. 30 Uso de memoria escenario Realizar Pedido para 50 Usuarios .....	114
Figura 3. 31 Uso del CPU escenario Realizar Pedido para 50 Usuarios.....	114
Figura 3. 32 Tiempo de respuesta escenario consultar pedidos pendientes 25 usuarios.....	115
Figura 3. 33 Carga procesada escenario consultar pedidos pendientes 25 usuarios.....	116
Figura 3. 34 Uso de memoria escenario consultar pedidos pendientes 25 usuarios .....	116
Figura 3. 35 Uso de CPU escenario consultar pedidos pendientes 25 usuarios	117
Figura 3. 36 Tiempo de respuesta escenario consultar pedidos pendientes 50 usuarios.....	117
Figura 3. 37 Carga procesada escenario consultar pedidos pendientes 50 usuarios.....	118

Figura 3. 38 Uso de memoria escenario consultar pedidos pendientes 50 usuarios .....	118
Figura 3. 39 Uso de CPU escenario consultar pedidos pendientes 50 usuarios	119
Figura 3. 40 Tiempo de respuesta escenario realizar carro de compras visitante 25 usuarios.....	120
Figura 3. 41 Carga procesada escenario realizar carro de compras visitante 25 usuarios.....	121
Figura 3. 42 Uso de Memoria escenario realizar carro de compras visitante 25 usuarios.....	121
Figura 3. 43 Uso de CPU escenario realizar carro de compras visitante 25 usuarios.....	122
Figura 3. 44 Tiempo de respuesta escenario realizar carro de compras visitante 50 usuarios.....	122
Figura 3. 45 Carga procesada escenario realizar carro de compras visitante 50 usuarios.....	123
Figura 3. 46 Uso de Memoria escenario realizar carro de compras visitante 50 usuarios.....	123
Figura 3. 47 Uso de CPU escenario realizar carro de compras visitante 50 usuarios.....	124
Figura 3. 48 Configuración de tamaño de memoria asignado a JBoss .....	125
Figura 3. 49 Uso de Memoria prueba de Stress escenario realizar pedido .....	125
Figura 3. 50 Uso de CPU prueba de Stress escenario realizar pedido .....	125
Figura 3. 51 Uso de Memoria prueba de Stress consultar pedidos pendientes.	126

Figura 3. 52 Uso de CPU prueba de Stress consultar pedidos pendientes.....	126
Figura 3. 53 Uso de Memoria prueba de Stress escenario Realizar Carro de Compras.....	127
Figura 3. 54 Uso de CPU prueba de Stress escenario Realizar Carro de Compras. .....	127

## PRESENTACIÓN

El negocio de la alimentación en los últimos años ha presentado un incremento considerable ya que satisface una necesidad primordial de los seres humanos. Las empresas de comida han implementado varias estrategias empresariales que permiten mejorar la atención al cliente y con ello mantenerse en vigencia en el mercado.

Una estrategia de mercado de las empresas alimenticias consiste en proveer a los clientes el servicio de entrega a domicilio. En la actualidad algunas empresas proveen este servicio mediante el uso de teléfonos convencionales. En este proyecto se ha propuesto automatizar los procesos de venta, realización y entrega de pedidos con la creación de un sistema web de pedidos para entrega a domicilio que se encuentre disponible en internet.

Para el desarrollo de la aplicación se ha seleccionado la metodología ágil de desarrollo ICONIX. Durante las etapas de esta metodología se capturó los requerimientos, elaboró diseños que permiten entender las necesidades de negocio, se seleccionó las plataformas para la implementación del sistema y se realizó pruebas unitarias que permitieron validar el funcionamiento del sistema mediante el cumplimiento de los requerimientos del cliente.

Para proteger la información sensible del sistema desarrollado se implementaron algunos controles de seguridad como la utilización de un canal de transmisión seguro (HTTPS), control de autenticación y autorización, manejo de claves seguras, y uso de certificados digitales para el pago con tarjeta de crédito.

Para probar el rendimiento del servidor de aplicaciones web y de la aplicación se realizaron pruebas de Carga y Stress. Con los resultados de estas pruebas se pudo proyectar la escalabilidad de la aplicación.



## RESUMEN

En el capítulo I se realizó una descripción de las empresas de comida a domicilio. Con la información recopilada se ha propuesto la creación de una aplicación para la automatización de los procesos de venta, realización y entrega de pedidos de estas empresas. Una vez conocida la problemática, se ha descrito la metodología de desarrollo ICONIX que fue seleccionada para este proyecto. Así mismo en este capítulo se han descrito las herramientas para la implementación de la aplicación web.

El capítulo II contiene información sobre la metodología de desarrollo seleccionada para la aplicación. Utilizando las fases de la metodología, se han elaborado los casos de uso, el diseño de diagramas de robustez y diagramas de secuencia. Luego con los diseños realizados se ha elaborado un diagrama de clases que permitió continuar con la implementación de la aplicación.

Después de haber escrito el código, se realizó la etapa de pruebas unitarias y pruebas de las páginas web que permitieron validar el sistema. Las pruebas unitarias permitieron verificar que la lógica del negocio realiza las funciones necesarias para cumplir con los requerimientos del negocio. Las pruebas de las páginas web permitieron certificar que el código XHTML generado cumple con los estándares internacionales.

En el capítulo III se procedió a la evaluación de la aplicación en el restaurante "Piguari" como caso de estudio. Con la información recopilada de la empresa se ha personalizado la aplicación para validar la funcionalidad completa. Se procedió con la carga de datos de la empresa para luego efectuar las pruebas de carga y stress. Estas pruebas permitieron conocer la carga que la aplicación será capaz de soportar manteniendo un rendimiento aceptable.

En el capítulo IV se han escrito las conclusiones y recomendaciones obtenidas en el presente proyecto.

# **CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA**

## **1.1 DESCRIPCIÓN DEL PROBLEMA**

### **1.1.1 MODELO DE NEGOCIO**

En la actualidad el negocio de comida se ha convertido en una gran oportunidad de mercado, debido al crecimiento en la demanda de este servicio. Con el fin de acercarse al cliente, las empresas han establecido una estrategia de entrega a domicilio que les permite aumentar sus ventas.

Se puede ordenar comida para una ocasión especial como un cumpleaños, aniversario, graduación, etc., o se puede enviar comida como un regalo para un amigo o un colega de trabajo.

La entrega de comida a domicilio ha crecido en popularidad ya que con ello la gente puede ahorrar tiempo y dinero.

La comida a domicilio puede definirse como:

La actividad económica consistente en la elaboración y transporte de comida desde un establecimiento industrial alimentario u hotelero hasta el domicilio del cliente. Puede referirse tanto a la comida rápida (pizza, hamburguesa, bocadillo, tapas, etc.); como al denominado catering, de mayor sofisticación. [1]

Existen diferentes empresas que proveen el servicio de entrega de comida a domicilio, y el tipo de producto que se entrega puede variar de acuerdo a cada empresa, estos pueden ser:

- Comida vegetariana
- Comida rápida

---

[1] Pagina: [http://es.wikipedia.org/wiki/Comida\\_a\\_domicilio](http://es.wikipedia.org/wiki/Comida_a_domicilio)

- Comida tradicional
- Catering

A continuación se presenta una breve reseña sobre el negocio de comida a domicilio.

El primer servicio de entrega a domicilio se dio durante la segunda guerra mundial, particularmente durante los bombardeos, donde muchas viviendas fueron destruidas y gran cantidad de personas se quedaron sin un lugar donde pudieran cocinar. La idea vino gracias al trabajo del servicio de mujeres voluntarias que proveían comida a los militares, transfiriendo el proceso a las familias que habían perdido sus casas debido a los bombardeos. [2]

Después de la guerra, los programas de entrega de comida elaborados por el servicio de mujeres voluntarias continuaron en operación. Se cree que el primer uso de este servicio se produjo en Hemel Hempstead Inglaterra. En esta etapa, en lugar del transporte que se usa hoy en día, la comida era entregada en viejos coches. Además, la comida que se entregaba era en su mayoría pre-cocida y caliente. En la época moderna, la mayoría de los locales han desarrollado servicio similares, aunque en la mayoría de los locales ahora se utiliza platos preparados congelados en lugar de la entrega de alimentos calientes.

Si bien fue en Reino Unido donde se creó la primera red de entrega de comida a domicilio, la idea pronto se esparció alrededor del mundo. En los Estados Unidos el primer sitio de entrega de comida a domicilio se creó en Filadelfia. A partir de 1954 este servicio se enfocó en alimentar a aquellas personas que estaban fuera de sus casas, asegurando que se cubran sus necesidades alimentarias y no

---

[2] Página: <http://www.squidoo.com/food-delivery-fast-food>

pasar hambre. El modelo de Filadelfia fue usado alrededor del país, primero en Columbus y luego en Nueva York. [3]

Los australianos rápidamente se dieron cuenta de los beneficios de los servicios de entrega de comida para los ancianos. El primer servicio en este país inició en 1952, aparentemente entregando la comida por una mujer en su triciclo. Otros servicios se iniciaron en Australia, incluyendo uno en Port Adelaide y Sídney.

#### **1.1.1.1 Visión**

Visión es:

Es la definición de lo que la empresa quiere ser en un futuro (hacia dónde quiere llegar). Recoge las metas y logros planteados por las organizaciones. Una declaración de visión es una breve representación filosófica de cómo quiere ser la empresa y como desea ser percibida en un futuro sus clientes, empleados, propietarios y otras personas importantes. [4]

Para la empresa Mc Donald, que brinda el servicio de entrega comida rápida a domicilio, su misión es:

“Brindar cien por ciento de satisfacción a nuestros clientes, manteniendo el más alto estándar de calidad, limpieza y servicio en la venta de comida de rápida ayudando al crecimiento integral de nuestros empleados”. [5]

Para una empresa que ofrece el servicio de entrega de comida a domicilio su visión suele incluir:

- Posicionarse como líder en el mercado de comidas

---

[3] Página: <http://ezinearticles.com/?Food-Delivery-Services---A-History&id=4767478>

[4] Administración Estratégica Conceptos y Casos de Thompson Arthur y Strickland A. J.

- Expandir su alcance

### 1.1.1.2 Misión

Para una empresa, la misión se define como:

La misión es el motivo, propósito, fin o razón de ser de la existencia de una empresa u organización porque define: 1) lo que pretende cumplir en su entorno o sistema social en el que actúa, 2) lo que pretende hacer, y 3) el para quién lo va a hacer; y es influenciada en momentos concretos por algunos elementos como: la historia de la organización, las preferencias de la gerencia y/o de los propietarios, los factores externos o del entorno, los recursos disponibles, y sus capacidades distintivas. [6]

La Misión de Domino's Ecuador que se especializa en la elaboración de pizzas y la entrega a domicilio es:

“Somos una empresa dedicada a la elaboración y producción de pizzas y productos complementarios, cuidando la satisfacción alimenticia de nuestros clientes, creando momentos de felicidad entre familia y amigos, basándose siempre en nuestros criterios de calidad y servicio excepcional.” [7]

Para una empresa que ofrece el servicio de entrega de comida a domicilio su misión contempla:

- Ofrecer los mejores productos
- Ofrecer la mejor atención al cliente

---

[5]<http://mcdonalds.com.ec/>

[6]Página: <http://www.promonegocios.net/mercadotecnia/mision-definicion.html>

### 1.1.1.3 Objetivos estratégicos

Los objetivos estratégicos son metas que una empresa tiene a corto plazo. Para una empresa de comida sus principales objetivos estratégicos son:

- Las empresas desean recuperar de la mejor forma las inversiones realizadas
- Tener una mayor participación en el mercado en el corto y mediano plazo.
- Tener mayor calidad de los productos
- Tener un alto nivel de satisfacción de los clientes
- Brindar el mejor servicio a los clientes

### *Estrategias empresariales*

Para Fred Nichols, la estrategia empresarial es “la compleja red de pensamientos, ideas, experiencias, objetivos, experticia, memorias, percepciones y expectativas que proveen una guía general para tomar acciones específicas en la búsqueda de fines particulares”. [8]

Algunas de las estrategias para una empresa de comida son:

- Enfocar la empresa hacia la mercadotecnia, promoción y distribución
- Mejorar las relaciones laborales y el ambiente de trabajo
- Movimientos para establecer el alcance competitivo de la compañía dentro de la industria
- Movimientos para adaptarse rápidamente a los cambios del entorno, a las exigencias de los clientes

---

[7] [www.dominos.com.ec](http://www.dominos.com.ec)

[8] Nichols, Fred. Three Forms of Strategy: Corporate, Competition and Strategy in general. 2000

El servicio de entrega de comida a domicilio es una nueva oportunidad para lograr un negocio rentable en la industria alimentaria. Esta estrategia es relativamente nueva y cada vez más empresas se interesan en conocer más acerca de lo que involucra esta oportunidad.

#### 1.1.1.4 Organigrama funcional

Tomando como referencia la estructura organizacional de algunas empresas que brindan el servicio de entrega de comida a domicilio, se ha establecido en la Figura 1.1, el organigrama funcional básico para una empresa de este tipo.

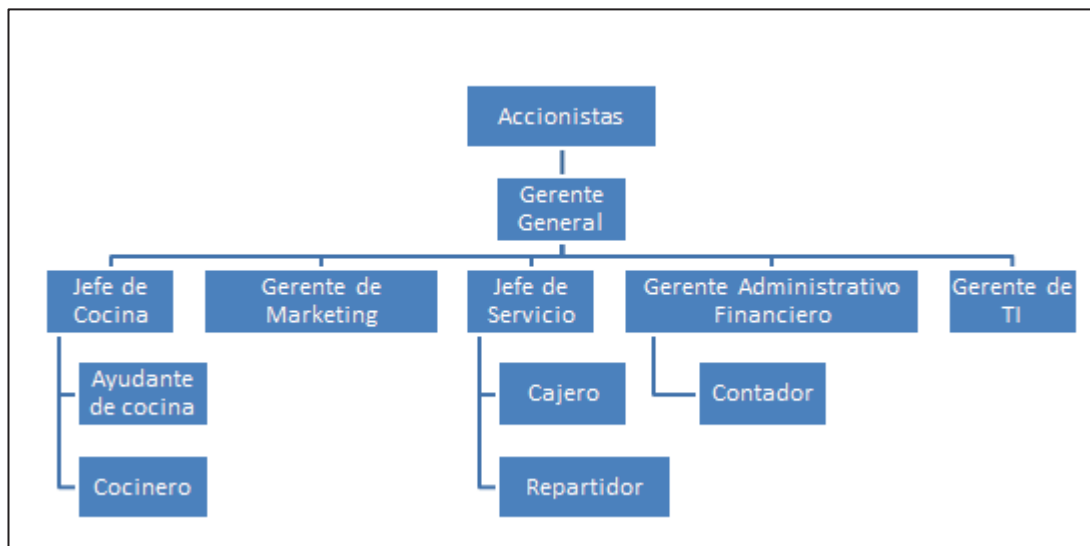


Figura 1. 1 Organigrama para una empresa de comida a domicilio.

#### *Descripción de los roles*

**Accionistas.-** Ejercen la representación legal de la empresa judicial.

**Gerente General.-** Cumplirá con las funciones de administradores del restaurante, serán los encargados de negociar con proveedores y con empresas relacionadas.

**Gerente de marketing.-** Se encargará de desarrollar planes de marketing y ventas para la empresa, así como su publicidad para llegar al segmento determinado.

**Gerente Administrativo Financiero.-** Es el responsable de controlar las finanzas de la empresa

En la parte inferior del organigrama se encuentra el personal relacionado con el proceso productivo y de venta de los productos, sus funciones fueron detalladas en el título que hace referencia a la mano de obra necesaria.

**Cocineros.-** Deben ser personas del sector que conozcan la sazón y la forma de hacer los productos ofrecidos por la empresa, capaces de satisfacer cualquier necesidad y gusto de los clientes.

**Meseros.-** Es la persona encargada de la atención de los clientes del restaurante y ayudarlos en todo lo posible, se encarga de tomar el pedido al cliente, realizar comandas, servir en la mesa el pedido del cliente, y despedir al cliente con amabilidad y cortesía.

**Cajero de restaurante.-** Es el encargado de la facturación y cobro de toda el área de restaurante; envía los vales de consumo a recepción para el ingreso en el sistema del mismo.

**Repartidor.-** Persona encargada de repartir los productos a los clientes.

**Gerente de TI.-** Encargado de manejar el área informática.

#### **1.1.1.5 MAPA DE PROCESOS**

Para el entendimiento del negocio de los restaurantes se requiere un análisis por procesos de la organización, el cual el mapa de procesos permitirá tener una visión objetiva sobre la organización.

Un mapa de procesos es una representación gráfica de los procesos realizados en la empresa y sus interrelaciones. Para la industria en general los procesos operativos inician desde el suministro de materia prima, pasando por la elaboración, comercialización y finalmente entrega del producto a los clientes.



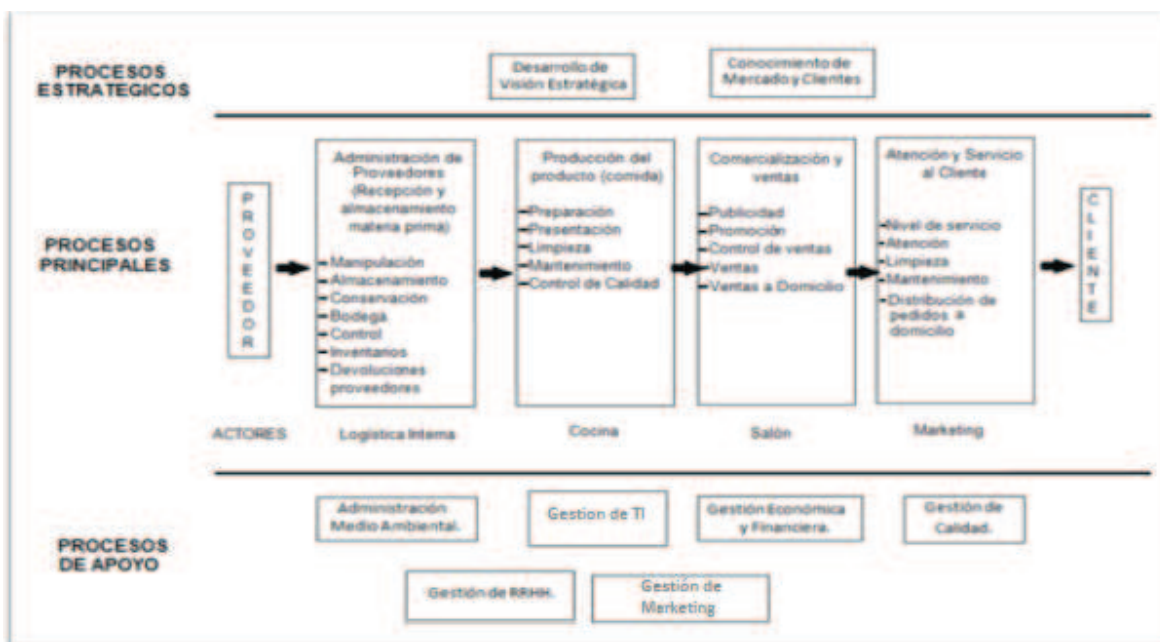


Figura 1. 2 Mapa de Procesos para empresa de comida a domicilio.

El mapa de procesos de la Figura 1.2 para un restaurante tiene inmerso entre sus procesos principales las ventas a Domicilio y la Distribución de pedidos a domicilio que son los procesos de interés de este trabajo [9]. Además todo restaurante maneja la publicidad para la comercialización y venta de sus productos por lo tanto como en este trabajo se desarrollará la aplicación web se manejará también el proceso de marketing porque es un portal al cual ingresarán los clientes.

### *Descripción de Procesos*

Un proceso es un conjunto de actividades que transforman una entrada en una salida, dándole un mayor valor a la entrada recibida.

Para describir un proceso es necesario identificar los requisitos para realizar ese proceso, los recursos (humanos y materiales), la salida (resultados) y a quien va dirigido (cliente).

---

[9]Página:[http://www.gestionrestaurantes.com/llegir\\_article.php?article=262](http://www.gestionrestaurantes.com/llegir_article.php?article=262)

La tabla 1.1 muestra el formato básico para describir un proceso:

<b>Objetivo</b>	<b>Descripción breve y concisa del objetivo del proceso.</b>
<b>Propietario</b>	Responsable del proceso.
<b>Requisitos</b>	Qué se requiere para iniciar el proceso.
<b>Recursos</b>	Recursos humanos y materiales para llevar a cabo el proceso
<b>Salida</b>	Producto o servicio creado por el proceso.
<b>Cliente</b>	A quién va dirigido el proceso.
<b>Actividades</b>	Acciones realizadas durante el proceso.

**Tabla 1. 1 Formato de Descripción de un Proceso**

Al tratarse de una aplicación web que permitirá la comercialización de la comida a domicilio los procesos que serán gestionados mediante la aplicación son: Ventas a Domicilio, la Distribución del pedido a domicilio y marketing. Estos procesos serán tratados a continuación en la forma en la que se realizan actualmente.

### **1.1.2 PROCESO DE VENTAS A DOMICILIO**

El proceso de ventas a domicilio como se describe en la Tabla 1.2 se trata de las actividades que lleva a cabo el cliente con la persona que recibe la llamada en el restaurante para obtener el producto en su domicilio.

<b>Objetivo</b>	<b>Realizar un pedido de venta para que el cocinero pueda preparar el pedido y el repartidor (motociclista) pueda empacar y distribuir la comida a domicilio.</b>
<b>Propietario</b>	Vendedor, cajero
<b>Requisitos</b>	Cliente efectúe una llamada para pedir comida a domicilio.  Red telefónica activa.

<b>Recursos</b>	Humano: cliente, cajero, vendedor, repartidor. Material: Teléfono.
<b>Salida</b>	Pedido de venta a domicilio
<b>Cliente</b>	Cliente final.

**Tabla 1. 2 Descripción del Proceso de Ventas a Domicilio**

### **1.1.2.1 Actividades en el proceso**

1. Cliente llama por teléfono.
2. Vendedor recibe llamada y verifica registro del cliente en el sistema.
3. Si no existe registra nuevo cliente en el sistema.
4. Cliente realiza pedido.
5. Vendedor registra pedido y da al cliente el valor a pagar.
6. Cliente confirma pedido.
7. Vendedor proporciona el tiempo de entrega del pedido a domicilio al cliente.
8. Cliente confirma.
9. Vendedor da opciones de pago (tarjeta de crédito o efectivo).
10. Cliente elige opción de pago.
11. Vendedor verifica opción de pago y finaliza llamada.
12. Vendedor envía pedido a Cocinero para la preparación del pedido.
13. PROCESO DE DISTRIBUCIÓN DEL PRODUCTO.
14. Repartidor entrega el dinero al vendedor.
15. Finaliza Venta a domicilio.

### 1.1.3 PROCESO DE DISTRIBUCIÓN DEL PEDIDO A DOMICILIO

El proceso de distribución del pedido a domicilio se trata de las actividades que realiza principalmente el repartidor para que el pedido sea el correcto y llegue al lugar indicado en el menor tiempo posible, tal como se describe en la Tabla 1.3.

<b>Objetivo</b>	Cliente obtenga el producto que solicitó.
<b>Propietario</b>	Repartidor (motociclista).
<b>Requisitos</b>	Cocinero tiene listo los productos pedidos por el cliente.
<b>Recursos</b>	Humano: cliente, cajero, vendedor, repartidor, cocinero. Material: Una motocicleta, dinero.
<b>Salida</b>	Pedido de venta a domicilio
<b>Cliente</b>	Cliente final.

Tabla 1. 3 Descripción del Proceso de Distribución del Pedido a Domicilio

#### 1.1.3.1 Actividades en el proceso

1. Vendedor asigna pedido a domicilio a un repartidor disponible.
2. Repartidor y cocinero verifican que el pedido este completo.
3. Repartidor entrega pedido al domicilio del cliente.
4. Cliente cancela (efectivo, tarjeta) el valor del pedido.
5. Finaliza distribución.

### 1.1.4 PROCESO DE MARKETING

La tabla 1.4 muestra la descripción del proceso de marketing, que está relacionado con los letreros, carta, hojas volantes que anuncian los productos del restaurante, donde el objetivo primordial es atraer la atención del cliente para que satisfaga una necesidad consumiendo los productos ofrecidos por el comedor.

<b>Objetivo</b>	<p>Informar al consumidor sobre los productos o servicios del restaurante.</p> <p>Probabilidad de que los productos anunciados sean adquiridos por el consumidor final.</p>
<b>Propietario</b>	Gerente de marketing,
<b>Requisitos</b>	Aprobación del gerente financiero para contratación de servicios de publicidad.
<b>Recursos</b>	<p>Humano: Gerente financiero, Gerente de marketing, empresa de Publicidad.</p> <p>Material: Dinero.</p>
<b>Salida</b>	Cliente informado sobre los productos y servicios del restaurante.
<b>Cliente</b>	Cliente final.

Tabla 1. 4 Descripción del Proceso de Publicidad

#### 1.1.4.1 Actividades en el proceso

1. El gerente de marketing elige los productos y promociones más relevantes que quiere que se muestren en la publicidad.
2. El gerente de marketing junto con el gerente financiero contratan una empresa de publicidad con los productos, fotografías y precios que quiere se muestren en la publicidad.
3. La empresa de publicidad realiza el diseño del letrero y lo muestra al gerente de marketing.
4. El gerente de marketing aprueba.
5. La empresa de publicidad realiza los distintos productos para la publicidad.
6. La empresa de publicidad entrega o instala los productos de publicidad.
7. Finaliza el proceso.

### **1.1.5 PROPUESTA DE AUTOMATIZACIÓN**

La aplicación web será un sitio que será administrado por el personal del restaurante para administrar los pedidos a domicilio, la entrega de los pedidos y la publicidad.

En el proceso de ventas a domicilio ya no será necesaria la participación del vendedor para receptor el pedido mediante llamadas telefónicas, sino que el cliente actuará directamente sobre el sitio y realizará el pedido mediante el sistema web.

El sistema tendrá el catálogo de los productos y el cliente podrá escoger los productos y las cantidades que desea para su pedido a domicilio. Antes de que el cliente envíe el pedido al restaurante podrá verificar las cantidades a pagar y la forma de pago que serán en efectivo o mediante tarjeta de crédito.

Una vez que el cliente ha realizado el pedido en el sistema, el cajero confirmará su entrega indicando la hora y el repartidor asignado según la disponibilidad de repartidores.

Los pedidos pueden ser cancelados mediante tarjeta de crédito, o en efectivo, en cualquiera de los casos el repartidor al momento de la entrega del pedido también entregará la factura correspondiente. En caso del pago en efectivo además llevará el cambio.

En cuanto a la publicidad el que se encargará del proceso será el administrador del restaurante quien publicará en el sitio los productos, promociones que desee simplemente cambiando fotografías o banners destinados para el sitio web.

## **1.2 JUSTIFICACIÓN DE LA METODOLOGÍA DE DESARROLLO.**

Para la realización del presente proyecto existen algunas metodologías de desarrollo como son: RUP, ICONIX, UWE y EORM, las mismas que se encuentran descritas en casi todos los proyectos de desarrollo de aplicaciones web. Para el caso de estudio se ha elegido ICONIX.

## 1.2.1 ICONIX

Es una metodología de desarrollo de software simple y práctico [10], ya que se encuentra entre la complejidad de RUP (Rational Unified Process) porque está dirigido por casos de uso y la simplicidad de eXtreme Programming (XP) donde se captura requerimientos, diseña y repite estos pasos de forma iterativa.

ICONIX es una metodología compuesta por 4 fases:

- Análisis y requerimientos
- Diseño preliminar
- Diseño detallado
- Implementación

### 1.2.1.1 Análisis y Requerimientos

Esta fase contiene 4 subfases, que cumplen una función específica dentro de la etapa de requerimientos.

*Obtener/Elaborar requerimientos funcionales:* Consiste en capturar y definir cuáles son los requerimientos del sistema según las necesidades del cliente, negocio.

*Realizar el modelo del dominio:* Se enfoca en elaborar el modelo de dominio que se refiere a identificar los objetos del mundo real que interactúan y se relacionan con el problema a resolver. También sirve de guía para la definición del diagrama de clases.

*Elaborar los requerimientos de comportamiento:* Describe la interacción entre el sistema y los usuarios del negocio. Elaboración del diagrama de casos de uso y escenarios. Luego de esto se obtiene los primeros prototipos de las interfaces de usuario.

---

[10] Página <http://www.iconixweb.com/>

*Revisión de requerimientos:* Verificar que los casos de uso cumplan con los requerimientos del negocio y expectativas de los usuarios del negocio.

Cabe mencionar que los diagramas de casos de uso y de clases no deben ser realizados con exhaustividad porque es la primera revisión y luego se afinarán.

#### **1.2.1.2 Diseño Preliminar**

Compuesto principalmente por la descripción de los casos de uso y el diagrama de robustez con el fin de identificar los objetos y las interacciones entre tales objetos.

Los pasos para completar esta fase de Iconix son las siguientes:

*Análisis de robustez:* Consiste en describir los casos de uso mediante los diagramas de colaboración y comunicación en la medida que se avanza.

*Actualizar modelo de dominio:* Se centra en actualizar el diagrama de clases, se lo hace paralelamente con el análisis de robustez de forma que se identifican nuevas clases, se eliminan o corrigen clases y se completan funciones que aparecen durante el proceso.

*Depurar los casos de uso:* Actualizar los casos de uso que se realizó en la primera fase.

*Revisión del diseño preliminar:* Verificar congruencia entre los diagramas de robustez, casos de uso y modelo de dominio.

Cuando esté completa esta subfase se podrá pasar a la fase de Diseño Detallado.

#### **1.2.1.3 Diseño Detallado**



La Fase del Diseño Detallado de Iconix presenta como principal objetivo la realización de los diagramas de secuencia, donde se podrá identificar los mensajes entre los diferentes objetos [11]. Además esta fase trabaja de forma iterativa sobre las etapas de la fase anterior.

*Elaborar diagramas de secuencia:* Elaborar los diagramas de secuencia que son aquellos que asignan las funciones a cada clase. Este proceso permitirá visualizar en detalle la implementación.

*Actualizar el modelo del dominio:* Esta parte se enfoca en depurar y agregar funciones a las clases del modelo de dominio. El resultado obtenido será el Diagrama de Clases del sistema.

*Depurar el modelo estático:* Radica en mejorar el diagrama de clases del sistema.

*Revisión crítica del diseño detallado:* Asegurar congruencia y consistencia entre diagramas de secuencia y diagrama de clases.

#### **1.2.1.4 Implementación**

Es la fase final donde se pasa a la parte técnica del proyecto, aplicando el diseño previo del sistema, además de probar la solución con la realización de pruebas.

*Codificación y pruebas:* Escribir código y realizar pruebas.

*Integración y escenario de pruebas:* Realizar pruebas en base a los escenarios descritos en los casos de uso.

*Revisión de codificación:* Realizar una revisión del código fuente.

---

[11]<http://www.portalhuarpe.com.ar/Seminario09/archivos/MetodologiaCONIX.pdf>.

### **1.2.2 JUSTIFICACIÓN DE LA METODOLOGÍA SELECCIONADA**

Para el desarrollo de este proyecto se ha elegido ICONIX porque es una metodología de desarrollo ágil que favorece la participación de los usuarios finales y documentación suficientemente necesaria durante todas sus fases. Al interactuar constantemente con el usuario final y mediante el uso de los prototipos, se logra de manera rápida el cumplimiento de los requisitos del negocio para el sistema. Además esta metodología utiliza modelos y diagramas definidos por UML, pero sin la necesidad de enfocarse demasiado en la documentación.

También ICONIX favorece a los proyectos de pequeña y mediana escala como el que se desea implementar. Primero se debe describir los casos de uso de la misma forma que en RUP, luego se pasa al diseño preliminar donde se elaboran diagramas de robustez, y también donde se elabora los prototipos de las interfaces. Esta información es trascendente para el cliente en un proyecto web ya que percibe la presentación del producto final. Finalmente se realizan los diagramas de secuencia, que permiten identificar el comportamiento de los objetos del modelo de dominio.

ICONIX es una metodología iterativa e incremental por lo que paulatinamente el equipo de desarrollo valida modificaciones del modelo de dominio para irlo mejorando en base a los diseños, y con ello realizar una mejor implementación ya que se tienen claros los requerimientos del negocio.

La curva de aprendizaje de la metodología ICONIX es corta además de que la documentación existente es clara y suficiente por lo cual se puede empezar a utilizarla de manera rápida.

### **1.3 SELECCIÓN DE LA PLATAFORMA TECNOLÓGICA**

Para el desarrollo de la aplicación se necesita definir algunas herramientas como el lenguaje de programación, el servidor de aplicaciones y el motor de base de datos. A continuación se describirán las herramientas seleccionadas para el presente proyecto.

### 1.3.1 PLATAFORMA DE DESARROLLO

En la actualidad existe un amplio número de alternativas para el desarrollo de aplicaciones web. Para el presente proyecto se ha elegido el lenguaje de programación Java con la plataforma de desarrollo JEE.

#### 1.3.1.1 Java EE

JEE (Java Enterprise Edition), es una plataforma para el desarrollo de aplicaciones empresariales distribuidas que se ejecutan sobre un servidor de aplicaciones. JEE es una extensión de Java que provee una API (Application Programming Interface) para tolerancia a fallas, mapeo objeto-relacional, arquitecturas distribuidas y multicapas, y web services para la construcción de aplicaciones empresariales. Esto permite a los desarrolladores crear aplicaciones que son portables y escalables, y que se puede integrar con tecnologías existentes. [12]

#### **Características:**

- Provee una plataforma para el desarrollo de aplicaciones distribuidas multicapas basadas en Web.
- La plataforma JEE está basada en componentes que residen en un servidor de aplicaciones y son gestionados por él.
- Es un conjunto de especificaciones de API's Java (no es un producto o aplicación en sí mismo).
- Su objetivo es simplificar el modelo de desarrollo.
- Uso de anotaciones Java para añadir meta-información al código y que serán explotadas por el entorno de ejecución Java EE.
- Uso de inyección de dependencia para simplificar el desarrollo de las aplicaciones JEE, facilitar su instalación y reducir el acoplamiento.

---

[12] CORE J2EE Best practices and Design Strategies

## Componentes de la arquitectura Java EE

JEE tiene una arquitectura basada en capas, contenedor, componentes y servicios. Las aplicaciones JEE son divididas en 4 capas: la capa cliente, la capa web, la capa de negocio y la capa de datos. [13]

En la Figura 1.3 se puede apreciar la arquitectura JEE y los componentes que la conforman.

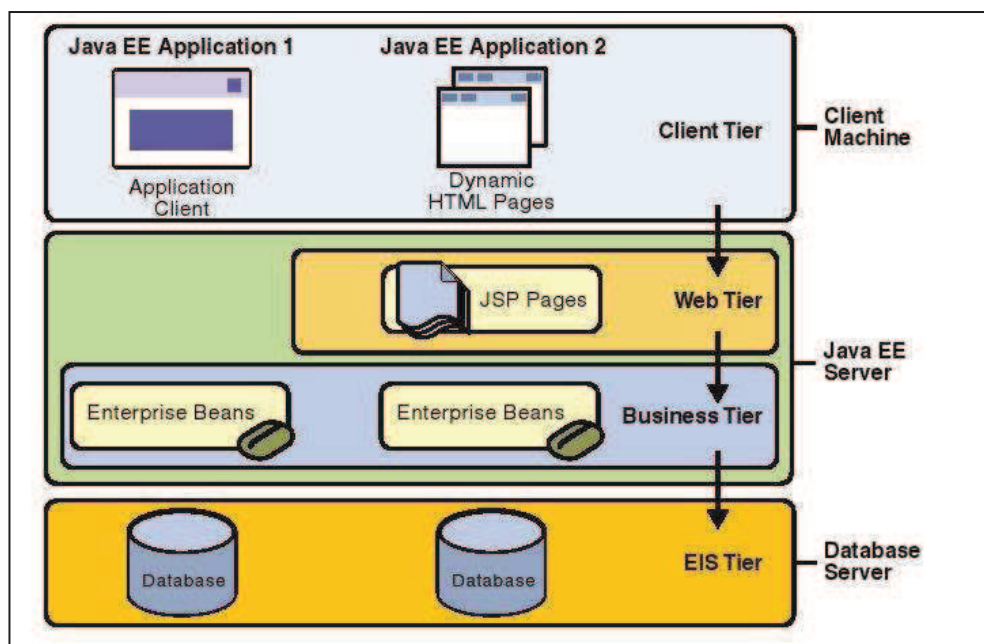


Figura 1. 3 Arquitectura JEE [13]

**Capa cliente.-** corresponde a los clientes de la aplicación, tales como: navegadores o clientes de escritorio.

**Componentes Web.-** se encuentra en el contenedor web y contiene la lógica de presentación que es la que mira el cliente.

---

[13]<http://ccia.ei.uvigo.es/docencia/SCS/Tema5-1.pdf>.

**Capa Negocio.-** se encuentra en contenedor de EJBs (Enterprise Java Beans) y contiene la lógica del negocio de la aplicación.

**Capa de datos.-** es el sistema almacenamiento de información de la empresa, tales como bases de datos y aplicaciones empresariales existentes.

**Ventajas:**

- Establece estándares para áreas de la empresa tales como conectividad con la base de datos, componentes de negocio, capa orientada a mensajes, componentes Web, protocolos de comunicación e interoperabilidad.
- Promueve una plataforma estándar para construir componentes de software portables entre implementaciones de terceros, evitando atarse a un solo proveedor. (Independencia de fabricante).
- Soporte de múltiples sistemas operativos ya que es basada en el lenguaje Java que provee portabilidad.
- Tiene alta madurez tecnológica gracias a la cantidad de proyectos existentes y su existencia en el mercado.

### 1.3.2 BASE DE DATOS

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a definir, construir, manipular la BD.

- Definición de la BD es definir el tipo, la estructura y restricciones de los datos
- Construir una BD es almacenar los datos en algún medio controlado por el SGBD

- Manipular la BD es consultar, insertar, actualizar, eliminar los datos y generar información [14]

Para el presente proyecto se ha elegido el motor de Base de Datos PostgreSQL.

### 1.3.2.1 PostgreSQL

PostgreSQL es un sistema de base de datos objeto-relacional (ORDBMS), basado en POSTGRES, Versión 4.2. Desarrollado en la Universidad Berkeley de California en el Departamento de Informática. POSTGRES fue pionero en muchos conceptos que sólo estuvieron disponibles en algunos sistemas de bases de datos comerciales mucho más tarde. [15]

#### Características:

- Cumple completamente con ACID
- Cumple con ANSI SQL
- Integridad referencial
- Replicación (soluciones comerciales y no comerciales) que permiten la duplicación de bases de datos maestras en múltiples sitios de replica
- Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby
- Reglas
- Vistas
- Triggers
- Unicode
- Secuencias
- Herencia

---

[14]A. Cob, Diseño y programación de bases de datos.

[15] <http://www.postgresql.org/docs/9.0/interactive/intro-what-is.html>.

- OuterJoins
- Sub-selects
- Procedimientos almacenados
- Soporte nativo SSL
- Lenguajes procedurales
- Respaldo en caliente
- Bloqueo a nivel mejor-que-fila
- Índices parciales y funcionales
- Autenticación Kerberos nativa
- Soporte para consultas con UNION, UNION ALL y EXCEPT
- Extensiones para SHA1, MD5, XML y otras funcionalidades
- Herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL
- Sistema de tipos de datos extensible para proveer tipos de datos definidos por el usuario, y rápido desarrollo de nuevos tipos
- Funciones de compatibilidad para ayudar en la transición desde otros sistemas menos compatibles con SQL [16]

### Ventajas:

- **Instalación ilimitada.**- Para proveedores comerciales la principal fuente de incumplimiento de licencia es la utilización del producto en más de una unidad, mientras que PostgreSQL no tiene costo asociado a la licencia del software y tiene como ventaja principal la escalabilidad.
- **Extensible.**- El código fuente está disponible sin costo. Es posible extender la funcionalidad o personalizar PostgreSQL.

---

[16] [http://soporte.tiendalinux.com/portal/Portfolio/postgresql\\_ventajas\\_html](http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html).

- **Multiplataforma.-** PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y en Windows está actualmente en estado beta de pruebas.
- **Diseñado para ambientes de alto volumen.-** PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC (Multi-version concurrency control) para conseguir una mejor respuesta en ambientes de grandes volúmenes.
- **Herramientas gráficas de diseño y administración de bases de datos.-** Existen herramientas gráficas para administrar las bases de datos (pgAdmin , pgAccess ) y para hacer diseño de bases de datos (Tora ,Data Architect). [16]

### 1.3.3 SERVIDOR WEB DE APLICACIONES

Un Servidor Web es aquel que se encarga atender los requerimientos de los usuarios mediante la entrega de páginas Web a los navegadores usando el protocolo de transferencia de hipertexto (HTTP). El Servidor Web incluye el hardware, sistema operativo, un software de Servidor Web, soporte al conjunto de protocolos TCP/IP.

A menudo el término servidor Web se refiere solo al software que procesa los requerimientos HTTP en el servidor que provee las páginas Web.

Para el presente proyectos se ha elegido el software servidor de aplicaciones Web JBoss.

#### 1.3.3.1 JBoss Application Server

JBoss AS es un servidor de aplicaciones Open Source que implementa el estándar Java EE 6. Debido a que es basado en Java, JBoss Application Server

---

[16] [http://soporte.tiendalinux.com/portal/Portfolio/postgresql\\_ventajas\\_html](http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html).



es multiplataforma, fácil de instalar y usar en cualquier sistema operativo que soporte Java.

JBoss AS es un componente de la fundación JBoss Enterprise Middleware System (JEMS) luego adquirida por Red Hat en abril del 2006, empresa que provee una variedad de productos completamente integrados y probados, incluyendo: JBoss AS, Hibernate, Apache Tomcat, JBoss Cache, Jboss jBPM, y JBoss Eclipse IDE. [17]

Con millones de descargar al día, JBoss AS es el servidor de aplicaciones JEE Open Source con alta popularidad en el mercado actual. JBoss ha logrado este nivel de popularidad debido a su madurez, excelencia técnica, soporte robusto y soporte multiplataforma.

#### **Características:**

- **Open Source.-** El grupo Gartner sugiere que la infraestructura JEE está siendo comercializada y que la respuesta de los mayores proveedores será incrementar las características propietarias de sus Suites de plataformas de aplicaciones. Por ello Gartner sugiere tener cuidado al momento de seleccionar un proveedor, debido a que el proveedor puede decidir convertir su suite en privativa, lo que aumentaría los costos debido a la necesidad de investigación. [18]
- **Simple.-** JBoss AS es preferido por los desarrolladores por su habilidad de brindar poderosas características que requiere una aplicación empresarial, sin sacrificar la simplicidad de los objetos Java. JBoss AS se ajusta a un completo sistema probado e integrado que provee una arquitectura modular.

---

[17]JBoss Application Server STANDARDS BASED INFRASTRUCTURE FOR THE ETP

[18]Y. Natis, CIO Update: Enterprise Java Lock-in Will Increase Through 2006 IGG-11122003-02, 2003

JBoss AS está certificado para JEE 6, habiendo pasado cerca de 23000 pruebas de compatibilidad con la suite de pruebas de Sun Micro Systems (empresa adquirida por Oracle en 2011). Esto representa una medida clara para asegurar que las aplicaciones JEE de JBoss podrán correr en otros servidores JEE certificados sin requerir de modificaciones costosas.

**Otras características:**

- Soporte para Programación Orientada a Aspectos (AOP)
- Soporte de Clustering
- Cache distribuido (usando JBoss Cache)
- Despliegue distribuido (Granja de servidores)
- Implementa el estándar EJB 3
- Integración con Hibernate (para implementar persistencia con Java Persistence API o JPA)
- Soporte del Servicio de Autenticación y Autorización (JAAS)
- Integración con Servicio de mensajería de Java (JMS)
- Soporte Java Server Faces 2 (Mojarra 2)
- Soporte Java Server Pages (JSP) / Java Servlet 2.1/2.5 (Tomcat)
- Balanceo de Carga
- Tolerancia a fallas
- Framework OSGi

#### **1.3.4 ENTORNO DE DESARROLLO**

Para la elaboración de la aplicación del presente proyecto se utilizará el IDE eclipse INDIGO versión 3.7, ya que es una aplicación open source multiplataforma para desarrollo de aplicaciones usando el lenguaje de programación Java.

Además eclipse contiene componentes que permiten la integración con el servidor de aplicaciones Jboss de tal manera que la aplicación web desarrollada puede ser instalada directamente en el servidor y realizar pruebas de la funcionalidad.

## **CAPÍTULO II: DESARROLLO DEL SISTEMA**

### **2.1 ANÁLISIS DEL SISTEMA**

Dentro del proceso ICONIX la fase de análisis consiste en la elaboración del modelo de dominio y el análisis de los casos de uso.

Al inicio esta fase de análisis se realiza la captura de los requerimientos funcionales, y se realiza las primeras interfaces como prototipos de lo que tendrá el sistema.

La siguiente fase de ICONIX es la de diseño. El vínculo entre el análisis y el diseño es el análisis de robustez que involucra relacionar los casos de uso con los objetos que serán utilizados en el diseño. [19]

#### **2.1.1 REQUERIMIENTOS FUNCIONALES DEL SISTEMA**

Los requerimientos funcionales sirven para definir lo que el sistema será capaz de hacer.

En base al análisis realizado en el capítulo I sobre el dominio del problema, se ha tomado varios requerimientos funcionales que pueden ser considerados comunes para cualquier empresa que desee implementar un sistema que provea la posibilidad de realizar pedidos para su entrega a domicilio.

Los requerimientos para realizar la aplicación para venta de comida a domicilio son:

1. La aplicación debe ser basada en web, se mostrará las categorías de los productos y los respectivos productos que son vendidos en los locales de comida. Para cada producto se mostrará información como su nombre, imagen, descripción, precio, stock.

---

[19]Agile Development ICONIX.

2. El sistema debe tener la capacidad para vender los productos mediante órdenes aceptadas por internet.
3. En el sistema se deben mostrar los productos que estén en stock, para que puedan ser solicitados por los clientes.
4. El usuario debe tener la capacidad de realizar búsquedas de productos en el sistema con varios criterios: por nombre, en promoción, por categoría, y luego mostrar los detalles de los productos que cumplen con el criterio de búsqueda.
5. El sistema deberá tener la funcionalidad de carro de compras y permitir:
  - 5.1. que se añadan productos al carro de compras.
  - 5.2. que se eliminen productos al carro de compras.
  - 5.3. que se actualicen las cantidades del carro de compras.
  - 5.4. ver un resumen de los productos y cantidades en el carro de compras.
6. El sistema permitirá que se realice el pedido del carro de compras, implementando las seguridades respectivas para el pago.
  - 6.1. Para realizar el pedido, el usuario deberá tener una cuenta en el sistema y además logearse.
  - 6.2. El sistema permitirá el pago de las órdenes en efectivo o tarjeta de crédito.
  - 6.3. El sistema deberá encriptar la información sensible de los clientes mientras se realice alguna transacción por internet.
  - 6.4. El sistema permitirá que el pedido realizado por el cliente tenga cualquier dirección de entrega.
  - 6.5. En el caso no haber la cantidad suficiente de productos del pedido, la dirección de entrega no es correcta, el cliente no es confiable, el proveedor podrá cancelar el pedido a través del usuario cajero.
7. El sistema permitirá la creación de cuentas de usuario donde se puedan almacenar datos tales como: nombres, apellidos, cédula, dirección, teléfonos.
8. El sistema deberá conservar una lista de todas las cuentas de usuario, en una base de datos.

9. Cuando un usuario quiera logearse en el sistema, deberá ingresar su nombre de usuario y una contraseña que será validada contra la información de la base de datos.
10. El sistema deberá proveer una sección para el administrador del sistema que:
  - 10.1. Tendrá una pantalla donde pueda ingresar los proveedores en el sistema, así como el tiempo que estarán activos en el sistema.
  - 10.2. La sección del administrador del sistema deberá ser asegurada mediante mecanismos de autenticación y autorización.
11. El sistema deberá proveer una sección para el administrador de cada proveedor, en donde:
  - 11.1. Se podrá ingresar las categorías y los productos de su empresa.
  - 11.2. Se podrá ingresar los empleados de su empresa, tales como cajero y repartidor. Para el repartidor se ingresara datos como nombres, apellidos, cedula, celular, y para los cajeros se tendrá que crear una cuenta de usuario en el sistema.
  - 11.3. Se podrá definir los datos de los horarios de los trabajadores de la empresa. Y se podrá asignar a los empleados a un horario de trabajo.
  - 11.4. Se podrá definir el número máximo de pedidos por hora, y valor mínimo por pedido para la empresa proveedora.
  - 11.5. Se podrá escoger los productos que se tendrán mayor publicidad en el sistema.
  - 11.6. Se tendrá una página en la que verá los pedidos que se han realizado a su empresa junto con las fechas en que fueron realizados.
  - 11.7. se tendrá una página en donde se verá los productos de los que dispone el proveedor, en el cuál se detalla el stock del producto.
12. El sistema deberá proveer una sección para un cajero de cada empresa proveedora, en donde:
  - 12.1. Se tendrá una pantalla en la cual se muestran todos los pedidos pendientes que tenga su empresa.

- 12.2. Se podrá asignar un repartidor que entregará el pedido al cliente, también se podrá asignar el tiempo medio de demora en la entrega.
- 12.3. Se podrá enviar un correo al cliente con los datos del pedido, indicando la confirmación del pedido del cliente.
- 12.4. Se podrá definir un pedido como entregado una vez que el repartidor entregue el pedido, y se pondrá al repartidor como disponible para nuevos pedidos.

### **2.1.2 MODELO DE DOMINIO**

La elaboración del modelo de dominio consiste en la construcción del glosario del proyecto, o un diccionario de términos que serán usados en el proyecto. Su propósito es asegurar que el equipo de desarrollo entienda el problema para el cual se está desarrollando el sistema. De la misma forma el modelo de dominio para un proyecto define el alcance y es la parte inicial sobre la cual se describirán los casos de uso. El modelo de dominio también provee de una nomenclatura común para tener una buena comunicación entre los miembros del equipo de desarrollo. [20]

El modelo de dominio también muestra gráficamente como se relacionan los diferentes objetos identificados durante la etapa de análisis de requerimientos para el sistema.

Para la elaboración del modelo de dominio se debe identificar los objetos del mundo real involucrados en el problema para el cual se está desarrollando el sistema. Además es recomendable que se elabore el modelo de dominio antes que los casos de uso para evitar ambigüedad en los nombres de los objetos.

---

[20]Apress Use Case Driven Object Modeling with UML.

Un aspecto importante es que el modelo de dominio sirve de apoyo durante todo el desarrollo, y además es actualizado a través del proyecto mientras se va comprendiendo de mejor forma el problema.

Para el desarrollo del sistema se ha elaborado el modelo de dominio representado en la Figura 2.1.

Este modelo de dominio contiene los objetos básicos que han sido identificados. Los objetos identificados son Pedido, Producto, Proveedor, la Cuenta de usuario, el empleado repartidor, mismo que se encuentran involucrados con el proceso realización de pedidos y de entrega de comida a domicilio.

Este modelo, de acuerdo a la metodología, será actualizado con el diseño del diagrama de clases. Al final se completará el modelo de dominio teniendo identificados los objetos faltantes, los atributos y métodos de los objetos.

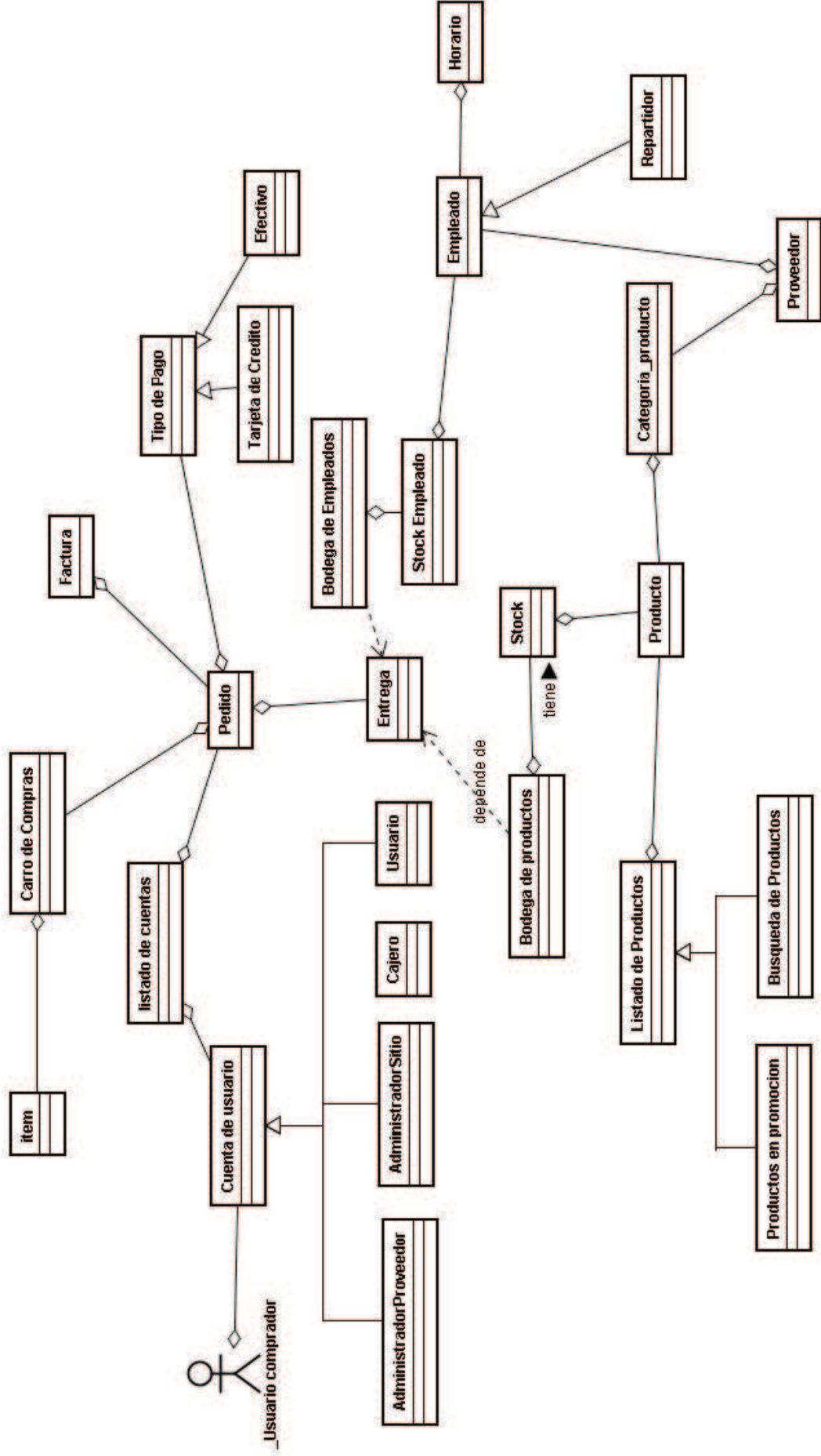


Figura 2. 1 Modelo de dominio



### 2.1.3 CASOS DE USO

Para la elaboración de los diagramas de casos de uso se han identificado 5 actores: administrador del sistema, administrador de cada proveedor, cajero del proveedor, cliente, y visitante. A cada uno de estos actores le corresponden varios casos de uso los cuales se detallan a continuación.

#### 2.1.3.1 Casos de uso actor Administrador del sistema

El administrador del sistema es el usuario encargado de ingresar las empresas proveedoras para el sistema así como el usuario que administrará los datos de dicha empresa. El diagrama de casos de uso se lo puede ver en la Figura 2.2.

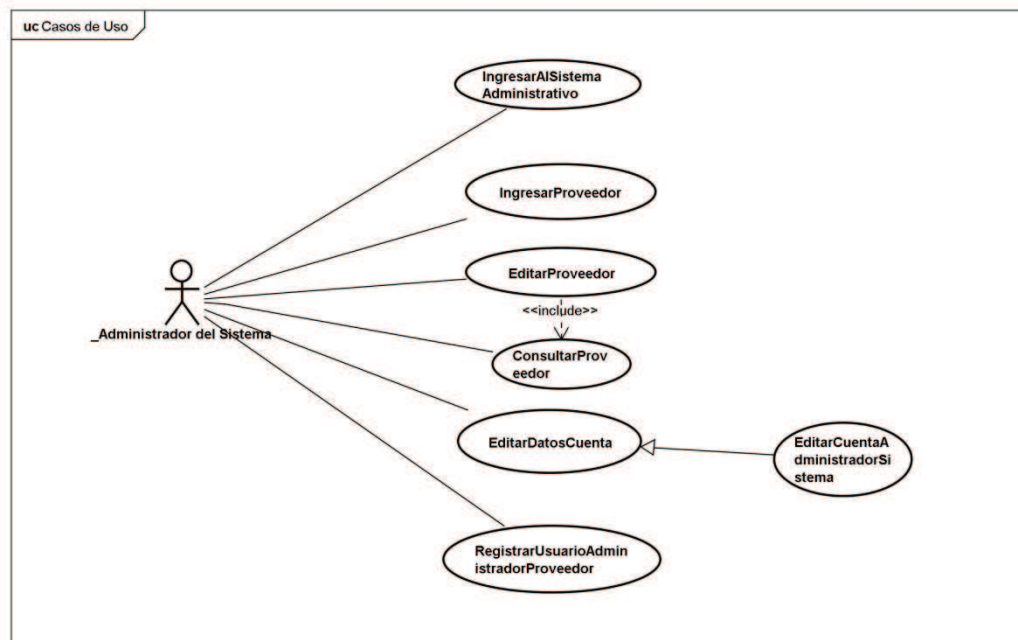


Figura 2. 2 Diagrama de casos de uso para el administrador del sistema

#### 2.1.3.2 Casos de uso actor Administrador del proveedor

El administrador de cada proveedor es el usuario encargado de ingresar información de la empresa proveedora para el sistema; esta información incluye los productos, categorías, empleados repartidores, así como el usuario cajero que gestionará los pedidos pendientes para dicha empresa. Los casos de uso para el administrador proveedor se los puede ver en la Figura 2.3.

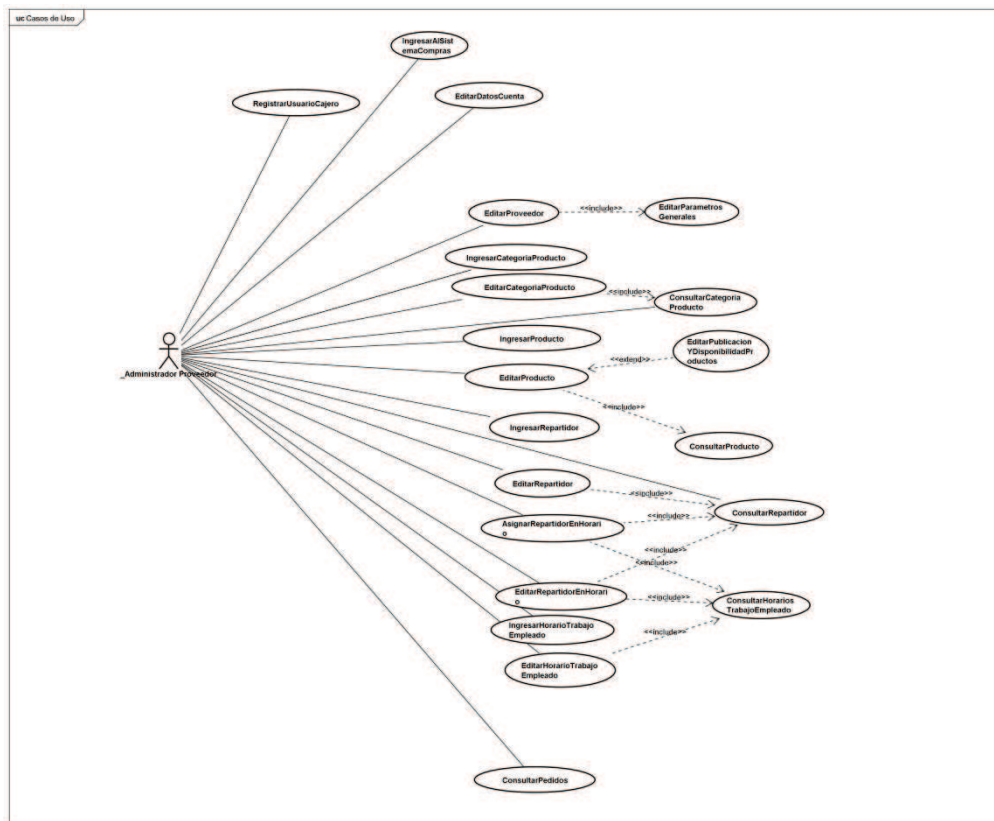


Figura 2. 3 Diagrama de casos de uso para el administrador del sistema

### 2.1.3.3 Casos de uso actor Cajero

El cajero de cada proveedor es el usuario encargado de manejar los pedidos realizados por los usuarios del sistema; este actor se encargará de definir el empleado repartidor de la empresa que entregará el pedido, y la hora de entrega. También deberá enviar un mensaje de confirmación de su pedido al correo del usuario comprador. El diagrama de casos de uso se lo puede ver en la Figura 2.4.

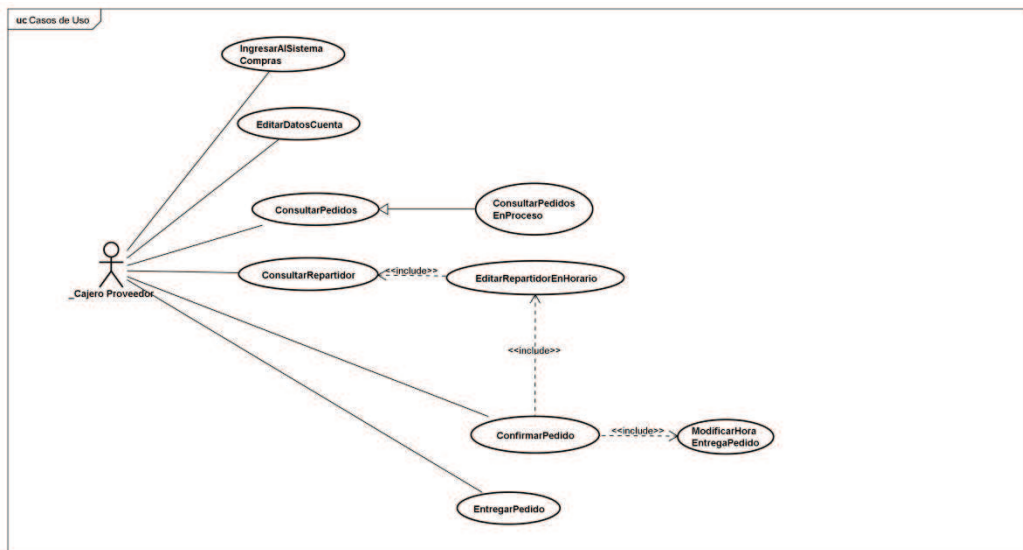


Figura 2. 4 Diagrama de casos de uso para el cajero del proveedor

#### 2.1.3.4 Casos de uso actor Usuario Visitante y cliente.

El usuario visitante es aquel que puede ingresar a la página del sistema y ver las características de los productos. También este usuario puede realizar su carro de compras y modificarlo, añadiendo o eliminando productos. El usuario visitante debe registrarse en el sistema para poder realizar el pedido de su carro de compras.

El usuario cliente es aquel usuario que se encuentra registrado en el sistema. Este usuario tiene las mismas opciones que el usuario visitante, solo que además puede realizar el pedido de su carro de compras. Cuando el usuario cliente este realizando pedido, puede definir el tipo de pago, la dirección de entrega y sus datos de facturación. El diagrama de casos de uso para el usuario visitante y usuario cliente se lo puede ver en la Figura 2.5.

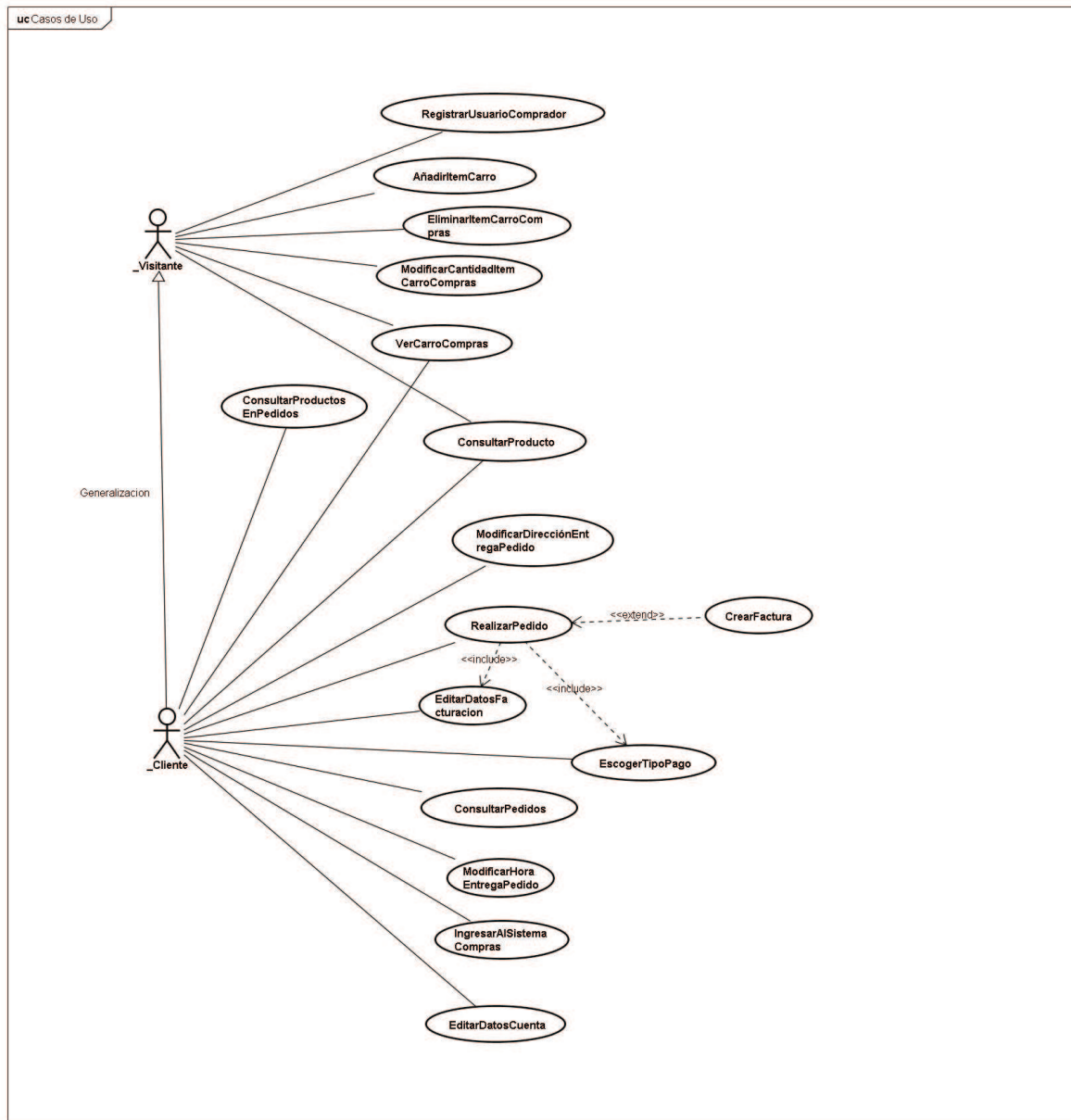


Figura 2. 5 Diagrama de casos de uso para el cajero del proveedor

## 2.1.4 DESCRIPCIÓN DE CASOS DE USO

A continuación se describirán 5 casos de uso desde las tablas 2.1 a la 2.5, y las demás descripciones se encuentran en el documento completo del CD anexo.

### 2.1.4.1 Caso de uso Registrar usuario administrador Proveedor

<b>Nombre Caso de Uso</b>	RegistrarUsuarioAdministradorProveedor
<b>Descripción</b>	El usuario administrador del sistema se

	encargará de crear las cuentas de los administradores de cada proveedor de comida a domicilio.
<b>Actor</b>	Administrador del Sistema
<b>Precondición</b>	El administrador del sistema se ha logeado en el sistema y existe el proveedor del usuario a registrar.
<b>Postcondición</b>	Se crea una cuenta de administrador proveedor para el usuario que manejará los datos de la empresa proveedora
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El administrador selecciona la opción crear nuevo administrador proveedor.</li> <li>2. El sistema carga la lista de los proveedores registrados.</li> <li>3. El usuario ingresa los datos del administrador proveedor.</li> <li>3. El usuario escoge la opción guardar.</li> <li>4. El sistema verifica si los datos son válidos.</li> <li>5. Si son válidos los datos, sistema verifica si existe email registrado.</li> <li>6. Si no existe ese email registrado, sistema genera contraseña.</li> <li>7. Sistema guarda registro de usuario con perfil de usuario Administrador Proveedor y Perfil Usuario.</li> <li>8. Sistema envía email al correo del dueño de la cuenta con los datos de acceso.</li> <li>9. Sistema emite mensaje de creación de usuario ha sido exitosa.</li> </ol>
<b>Flujo(s) alternativo(s)</b>	<p>5a. Si los datos no son válidos, sistema emite mensaje que los datos ingresados son incorrectos.</p> <p>6a. Si existe email registrado, sistema emite mensaje que el correo ya está registrado en una cuenta, que puede editar roles en Cuentas de Usuario.</p>
<b>Nota</b>	

Tabla 2. 1 Caso de uso RegistrarUsuarioAdministradorProveedor

### 2.1.4.2 Caso de uso Realizar pedido

<b>Nombre Caso de Uso</b>	RealizarPedido
<b>Descripción</b>	El usuario podrá enviar el pedido de su compra a la empresa proveedora para que se le entregue la comida en su domicilio.
<b>Actor</b>	Cliente
<b>Precondición</b>	Usuario tenga creado un carro de compras con lista de productos a comprar.
<b>Postcondición</b>	Usuario recibe mensaje de que su pedido está siendo procesado por la empresa proveedora.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. Usuario elige la Opción Realizar Pedido.</li> <li>2. Sistema verifica si usuario esta autenticado.</li> <li>3. Si usuario esta autenticado, sistema muestra carro de compras y carga datos de Facturación y dirección de entrega.</li> <li>4. Usuario elige Comprar.</li> <li>5. Sistema verifica si pedido cumple los requisitos del proveedor (mínimo de la compra) y para cada producto (cantidad mínima de compra, cantidad máxima de compra).</li> <li>6. Si pedido cumple, Sistema incluye (EditarDatosFacturacion).</li> <li>7. Sistema incluye (EscogerTipoPago).</li> <li>8. Sistema verifica si tipo de pago es efectivo.</li> <li>9. Si tipo de pago es efectivo, sistema verifica si existen todos los productos en las cantidades indicadas.</li> <li>10. Si existen todos los productos y cantidades, Sistema guarda registro de pedido.</li> <li>11. Sistema envía correo a usuario con el pedido y que está en estado pendiente.</li> </ol>

	<p>12. Sistema muestra mensaje que su pedido está siendo procesado y que pronto se le indicará la hora de entrega.</p>
<p><b>Flujo(s) alternativo(s)</b></p>	<p>3a. Si usuario no está autenticado, sistema incluye (IngresarAlSistema).</p> <p>6a. Si no cumple, sistema emite mensaje que el pedido debe cumplir con un mínimo y detalla que productos no están en las cantidades aceptables.</p> <p>7. Sistema regresa a paso 3.</p> <p>9a. Si tipo de pago es crédito, sistema pide datos de la tarjeta.</p> <p>10. Usuario ingresa los datos de la tarjeta.</p> <p>11. Usuario elige Continuar.</p> <p>12. Sistema verifica si los datos de la tarjeta son válidos.</p> <p>13. Si los datos de la tarjeta son válidos, sistema guarda registro de tarjeta.</p> <p>14. Sistema verifica si existen todos los productos en las cantidades indicadas.</p> <p>15. Si existen todos los productos y cantidades, Sistema guarda registro de pedido.</p> <p>16. Sistema envía correo a usuario con el pedido y que está en estado pendiente.</p> <p>17. Sistema muestra mensaje que su pedido está siendo procesado y que pronto se le indicará la hora de entrega.</p> <p>13a. Si los datos de la tarjeta no son válidos, sistema emite mensaje que los datos de la tarjeta no son válidos, y regresa a 9a.</p> <p>10a. Si no existen todos los productos y</p>

	<p>en las cantidades especificadas, sistema calcula nuevo pedido.</p> <ol style="list-style-type: none"> <li>11. Sistema muestra usuario nuevo pedido y pide confirmación.</li> <li>12. Usuario elige confirmar.</li> <li>13. Sistema guarda registro de pedido.</li> <li>14. Sistema envía correo a usuario con el pedido y que está en estado pendiente.</li> <li>15. Sistema muestra mensaje que su pedido está siendo procesado y que pronto se le indicará la hora de entrega.</li> </ol>
<b>Nota</b>	

Tabla 2. 2 Caso de uso RealizarPedido

#### 2.1.4.3 Caso de uso ver Carro de compras

<b>Nombre Caso de Uso</b>	VerCarroCompras
<b>Descripción</b>	El usuario podrá visualizar la lista de los productos que tiene en el carro de compras.
<b>Actor</b>	Visitante Cliente
<b>Precondición</b>	Usuario ha añadido productos al carro de compras.
<b>Postcondición</b>	Usuario obtiene la lista de los productos que ha añadido al carro de compras, con la suma total.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. Usuario elige la opción Ver Carro de Compras.</li> <li>2. Sistema verifica si existe carro de compras.</li> <li>3. Si existe carro de compras, sistema verifica si existe al menos un producto en el carro de compras.</li> <li>4. Si existe al menos un producto, sistema lista productos del carro de compras.</li> </ol>



	5. Sistema muestra la lista de los productos del carro de compras.
<b>Flujo(s) alternativo(s)</b>	3a. Si no existe carro de compras, sistema muestra a usuario mensaje que no tiene ningún producto en el carro de compras.  4a. Si no existe al menos un producto, sistema muestra a usuario mensaje que no tiene ningún producto en el carro de compras.
<b>Nota</b>	

Tabla 2. 3 Caso de uso VerCarroCompras

#### 2.1.4.4 Caso de uso Editar Parámetros generales Proveedor

<b>Nombre Caso de Uso</b>	EditarParametrosGeneralesProveedor
<b>Descripción</b>	El usuario administrador proveedor podrá editar los datos del registro del proveedor como son: número máximo de pedidos por hora, valor mínimo por pedido, y correo electrónico.
<b>Actor</b>	Administrador Proveedor
<b>Precondición</b>	El usuario administrador proveedor se ha logeado en el sistema.
<b>Postcondición</b>	Se guarda en la base de datos las modificaciones realizadas al proveedor.
<b>Flujo principal</b>	1. el usuario selecciona la opción modificar para los parámetros generales de un proveedor. 2. el sistema presenta un formulario con los datos generales de un proveedor. 3. el usuario edita los datos del proveedor. 4. el usuario escoge la opción guardar. 5. el sistema almacena en la base de datos los cambios realizados en el horario de trabajo.

<b>Flujo(s) alternativo(s)</b>	<p>3a. No se ingresa todos los datos obligatorios del proveedor.  4. El sistema emite una alerta solicitando los campos obligatorios  5. El proveedor no se guarda.</p> <p>3b. El usuario ingresa datos erróneos en los campos.  4. El sistema emite una alerta  5. El proveedor no se guarda.</p> <p>5a. El usuario selecciona la opción cancelar.  6. El proveedor no se guarda en el sistema.</p>
<b>Nota</b>	

Tabla 2. 4 Caso de uso EditarParametrosGeneralesProveedor

#### 2.1.4.5 Caso de uso ingresar Producto

<b>Nombre Caso de Uso</b>	IngresarProducto
<b>Descripción</b>	el usuario podrá ingresar un producto en el sistema siempre y cuando este no este registrado anteriormente
<b>Actor</b>	Administrador Proveedor
<b>Precondición</b>	<p>Usuario este logeado.</p> <p>El producto no exista.</p> <p>Ya existen categorías de productos.</p>
<b>Postcondición</b>	Se crea un registro de un nuevo producto en el sistema
<b>Flujo principal</b>	<p>1. el usuario selecciona la opción nuevo producto.</p> <p>2. el sistema retorna un formulario para la creación de un nuevo producto.</p> <p>2. el usuario ingresa los datos del producto.</p> <p>3. el usuario selecciona una categoría para el nuevo producto.</p>

	<p>4. el usuario selecciona la opción guardar</p> <p>5. el sistema almacena un registro del producto.</p>
<b>Flujo(s) alternativo(s)</b>	<p>3a. no se ingresa todos los datos obligatorios del producto.</p> <p>4. el sistema emite una alerta solicitando los campos obligatorios</p> <p>5. el producto no se guarda.</p> <p>3b. el usuario ingresa datos erróneos en los campos.</p> <p>4. el sistema emite una alerta</p> <p>5. el producto no se guarda.</p> <p>5a. el usuario selecciona la opción cancelar.</p> <p>6. el producto no se guarda en el sistema.</p>
<b>Nota</b>	

Tabla 2. 5 Caso de uso IngresarProducto

### 2.1.5 PROTOTIPO DE INTERFACES

En base a los requerimientos funcionales para el presente proyecto, se ha realizado un diseño preliminar de las pantallas a desarrollar, algunos diseños se muestran en las Figuras de la 2.6 a la 2.22.



The image shows a login interface for 'food shopping'. At the top, the logo 'food shopping' is displayed in red and black text. Below the logo, a yellow banner contains the text 'Bienvenido al mayor sitio de pedidos a domicilio'. Underneath the banner, there are two input fields: 'Email:' and 'Contraseña'. Below these fields is a white button with the text 'INGRESAR'.

Figura 2. 6 Pantalla de Ingreso al sistema

## Prototipos para el usuario visitante



Figura 2. 7 Prototipo página principal



Figura 2. 8 Pantalla para Registrarse en el sistema

**food shopping**

### EDITAR DATOS CUENTA

Email:   
 Nombres :   
 Apellidos:   
 Dirección:   
 Cedula:   
 Contraseña:   
 Confirmar:

Figura 2. 9 Prototipo Pantalla para Editar Cuenta Usuario

## Prototipos para el usuario cliente

### Carrito de Compras

Nombre	Cantidad	PU	Total	Eliminar
<b>Arroz con camarón</b>	<input type="text" value="1"/>	<b>6.00</b>	<b>6.0</b>	<input type="button" value="X"/>
<b>Pollo asado</b>	<input type="text" value="1"/>	<b>12.00</b>	<b>12.0</b>	<input type="button" value="X"/>

**Total : 18.0**

Figura 2. 10 Prototipo vista carrito de compras

**food shopping**

### Pedido

Cantidad	Detalle	V.U	V.T
1	Arroz con camarón	6.00	6.00
1	Ceviche	7.00	7.00
2	colas	1.00	2.00
		SUBTOTAL:	15.00
		IVA 12%:	1.80
		TOTAL:	16.80

**COMPRAR**

### DATOS DE FACTURACIÓN

Nombre:

Dirección:

Teléfono:

RUC / CI:

**CAMBIAR DATOS**

### DIRECCIÓN DE ENTREGA

**CAMBIAR DIRECCIÓN DE ENTREGA**

Figura 2. 11 Prototipo pantalla para realizar el pedido.

(Escoger el tipo de pago, dirección, datos de facturación)

## Prototipos para el usuario cajero Proveedor

**food shopping**

HORA: 09:48

### Pedidos Pendientes

Id Pedido	Cliente	Direccion Entrega	Hora de Entrega	Repartidor	Editar
1	Carlos González	Real Audiencia	10 : 30	Sin asignar	
2	Sebastian Morales	Cotocollao	10 : 50	Sin asignar	
3	Ana Grijalva	Av. La Prensa	11 : 40	Sin asignar	
4	Gabriela Jimenez	Av. Los Cedros	12 : 00	Sin asignar	

Figura 2. 12 Prototipo pantalla de listado de pedidos pendientes.

HORA: 09:48

**food shopping**

### Pedido Pendiente

Cantidad	Detalle	V.U	V.T	Pedido No 16	
1	Arroz con camarón	6.00	6.00	Cliente:	Carlos González
1	Ceviche	7.00	7.00	Dirección de Entrega:	Av. La Prensa y Bellavista
2	colas	1.00	2.00	Forma de Pago:	Crédito
SUBTOTAL:			15.00	Repartidor:	Juan Perez
IVA 12%:			1.80	Hora de Entrega:	10:30
TOTAL:			16.80		

**GUARDAR**

Figura 2. 13 Prototipo pantalla de confirmación del pedido. Asignar repartidor y hora de entrega.

HORA: 09:48

**food shopping**

### Pedidos

Fecha: 10-06-2012      Usuario: Elena      Estado: Pendiente

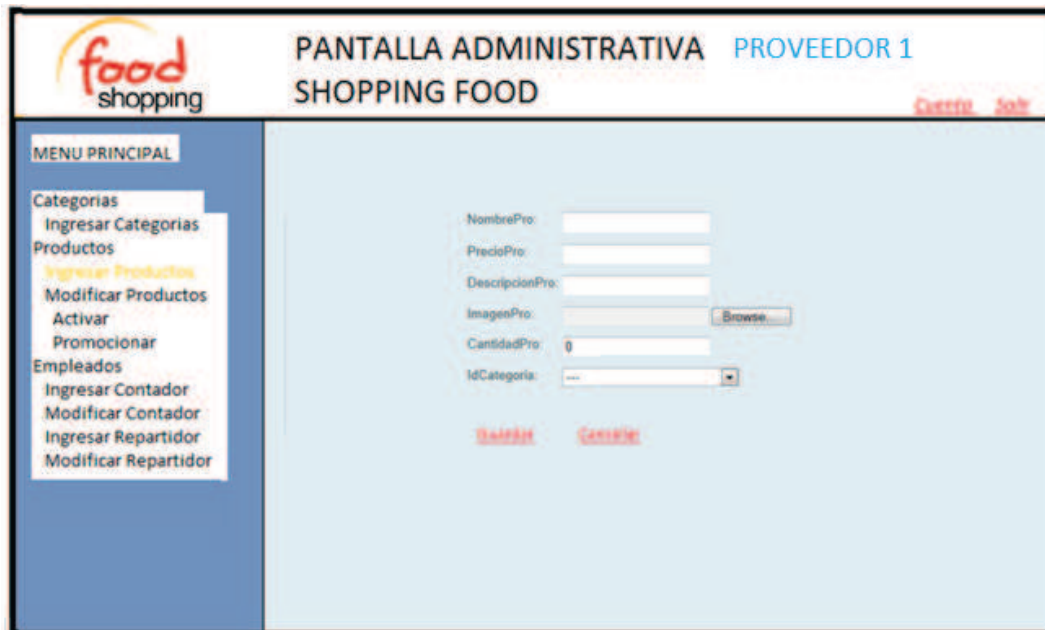
Fecha-Fin: 17-06-2012

**GUARDAR**

Id Pedido	Cliente	Total	Estado
1	Carlos González	\$20.00	Finalizado
2	Sebastian Morales	\$15.60	Cancelado
3	Ana Grijalva	\$8.00	Pendiente
4	Gabriela Jimenez	\$10.50	Finalizado

Figura 2. 14 Prototipo pantalla Consultar Pedidos

## Prototipos para el usuario Administrador proveedor



food shopping

PANTALLA ADMINISTRATIVA PROVEEDOR 1  
SHOPPING FOOD

Crear Job

MENU PRINCIPAL

- Categorías
  - Ingresar Categorías
- Productos
  - Ingresar Productos
  - Modificar Productos
  - Activar
  - Promocionar
- Empleados
  - Ingresar Contador
  - Modificar Contador
  - Ingresar Repartidor
  - Modificar Repartidor

NombrePro:

PrecioPro:

DescripcionPro:

ImagenPro:

CantidadPro:

IdCategoria:

Figura 2. 15 Prototipo de pantalla para ingresar un producto



food shopping

PANTALLA ADMINISTRATIVA PROVEEDOR 1  
SHOPPING FOOD

Crear Job

MENU PRINCIPAL

- Categorías
  - Ingresar Categorías
- Productos
  - Ingresar Productos
  - Modificar Productos
  - Activar
  - Promocionar
- Empleados
  - Ingresar Contador
  - Modificar Contador
  - Ingresar Repartidor
  - Modificar Repartidor

CATEGORIAS

IdCategoria	NombreCat	IdProv	
1	Entrada	1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
2	Plato fuerte	1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>
3	Postre	1	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[Nueva Categoría](#)

NombreCat:

Figura 2. 16 Prototipo de pantalla para ingresar una nueva categoría



**food shopping** PANTALLA ADMINISTRATIVA **PROVEEDOR 1** Cuenta [Salir](#)

**INGRESO DE EMPLEADOS**

CIEmp:

NombresEmp:

ApellidosEmp:

TelefonoEmp:

CelularEmp:

Tipo Empleado:

Correo empleado:

[Volver](#) [Cancelar](#)

Figura 2. 17 Prototipo de pantalla para ingresar empleados

**food shopping** PANTALLA ADMINISTRATIVA **PROVEEDOR 1** Cuenta [Salir](#)

**MODIFICAR PRODUCTOS**

NOMBRE	CATEGORIA	PRECIO	STOCK	ACTIVO	EN PUBLICIDAD	
Camarones Apanados	ENTRADA	\$2.00	12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Modificar</a>
Empanadas de morocho	ENTRADA	\$3.00	22	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">Modificar</a>
Plato de arroz con Pollo Cordobues	PLATO PRINCIPAL	\$5.00	32	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Modificar</a>
Plato de arroz con Carne a la milanesa	PLATO PRINCIPAL	\$8.00	32	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">Modificar</a>
Bola de Helado	POSTRE	\$3.00	33	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Modificar</a>
Pedazo de Pastel de chocolate	POSTRE	\$2.00	44	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">Modificar</a>

[Volver](#) [Cancelar](#)

Figura 2. 18 Prototipo de pantalla para Modificar estado de disponibilidad y publicidad de productos

food shopping

PANTALLA ADMINISTRATIVA PROVEEDOR 1  
SHOPPING FOOD

Cuenta Salir

MENU PRINCIPAL

Categorías  
Ingresar Categorías

Productos  
Ingresar Productos  
Modificar Productos  
Activar  
Promocionar

Empleados  
Ingresar Contador  
Modificar Contador  
Ingresar Repartidor  
Modificar Repartidor

ASIGNAR HORARIO A REPARTIDOR

NOMBRE	HORARIOS	DISPONIBLE	
Repartidor 1	Elija un horario 8:00- 14:00 14:00- 18:00	<input type="checkbox"/>	Guardar
Repartidor 2	Elija un horario	<input type="checkbox"/>	Guardar
Repartidor 3	Elija un horario	<input type="checkbox"/>	Guardar
Repartidor 4	Elija un horario	<input type="checkbox"/>	Guardar
Repartidor 5	Elija un horario	<input type="checkbox"/>	Guardar
Repartidor 6	Elija un horario	<input type="checkbox"/>	Guardar

Figura 2. 19 Prototipo de pantalla para asignar horario de trabajo a repartidores.

## Prototipos para el usuario Administrador del sitio

food shopping

ADMINISTRADOR DEL SITIO  
SHOPPING FOOD

Cuenta Salir

MENU PRINCIPAL

Proveedor  
Ingresar Proveedor  
Crear administrador  
Proveedor  
Editar Proveedor

INGRESAR PROVEEDOR

NombreProv:

CorreoProv:

ValorMin:

PedidosHora:

Fecha de caducidad Proveedor

junio de 2012

do	lu	ma	mi	ju	vi	sá
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Guardar Cancelar

Figura 2. 20 Prototipo de pantalla para Ingresar un nuevo proveedor.

food shopping ADMINISTRADOR DEL SITIO SHOPPING FOOD Cuenta Salir

MENU PRINCIPAL

- Proveedor
  - Ingresar Proveedor
  - Crear administrador Proveedor
  - Editar Proveedor

CUENTA DE ADMINISTRADOR PARA EL PROVEEDOR **PROVEEDOR 1**

Nombres

Apellidos

Direccion

Cedula

Correo

[Guardar](#) [Cancelar](#)

Figura 2. 21 Prototipo de pantalla para crear una cuenta de administrador proveedor.

food shopping ADMINISTRADOR DEL SITIO SHOPPING FOOD Cuenta Salir

MENU PRINCIPAL

- Proveedor
  - Ingresar Proveedor
  - Crear administrador Proveedor
  - Editar Proveedor

MODIFICAR PROVEEDOR

NombreProv:  [BUSCAR](#)

NOMBRE	VIGENTE HASTA	
Repartidor1	22/11/2012	<a href="#">Modificar</a>
Repartidor2	22/11/2012	<a href="#">Modificar</a>
Repartidor3	22/12/2012	<a href="#">Modificar</a>
Repartidor4	22/12/2013	<a href="#">Modificar</a>
Repartidor5	22/12/2013	<a href="#">Modificar</a>
Repartidor6	22/12/2013	<a href="#">Modificar</a>

Fecha de caducidad Proveedor

junio de 2012

do	lu	ma	mi	ju	vi	sá
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

[Guardar](#) [Cancelar](#)

Figura 2. 22 Prototipo de pantalla para editar la vigencia para un proveedor en el sistema.

## 2.1.6 REVISIÓN DE REQUERIMIENTOS

La revisión de requerimientos es un aspecto importante en el desarrollo del sistema, ya que ayuda a asegurar que los requerimientos están claramente comprendidos por el equipo de desarrollo.

En el sistema se ha validado que se tomen en cuenta los objetos importantes en el modelo de dominio, y también que se describan los escenarios básicos y alternativos de los casos de uso.

## 2.2 DISEÑO DEL SISTEMA

El diseño del sistema consiste en diseñar los diagramas de casos de uso en diagramas de robustez y los diagramas de secuencia para el actual proyecto.

### 2.2.1 DISEÑO PRELIMINAR

El diseño preliminar permitirá tener una visión inicial del sistema, el cuál irá afinándose según se avanza con los diagramas de Robustez para los casos de uso.

#### 2.2.1.1 Análisis de Robustez para el Caso de Uso

A continuación se muestran 5 diagramas de robustez desde la figura 2.23 a la 2.27, y los demás se encuentran en el documento completo del CD anexo.

#### Diagrama de robustez Caso de uso registrar usuario administrador proveedor

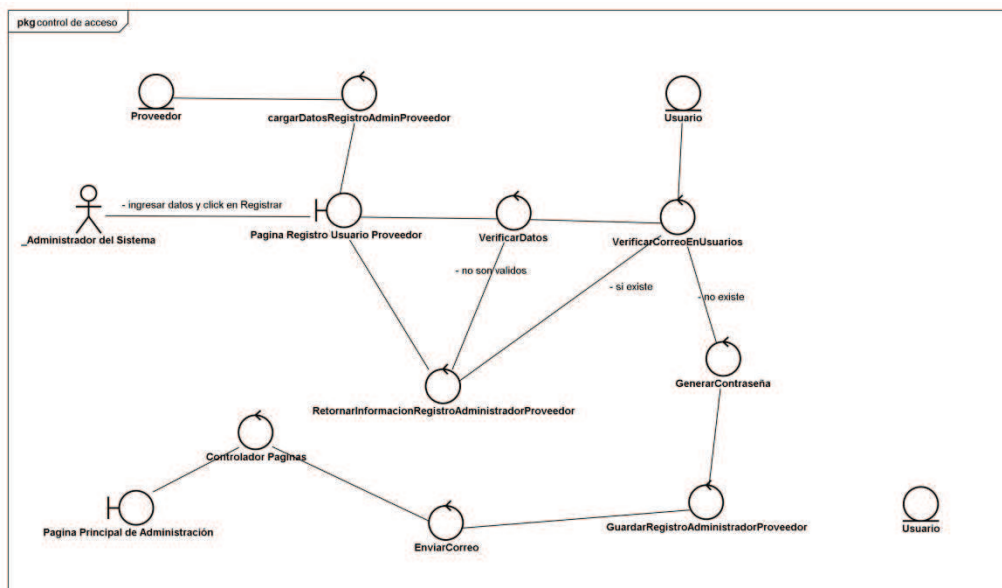


Figura 2. 23 Análisis de Robustez para el Caso de Uso registrarUsuarioAdministradorProveedor

## Diagrama de robustez Caso de uso Realizar pedido

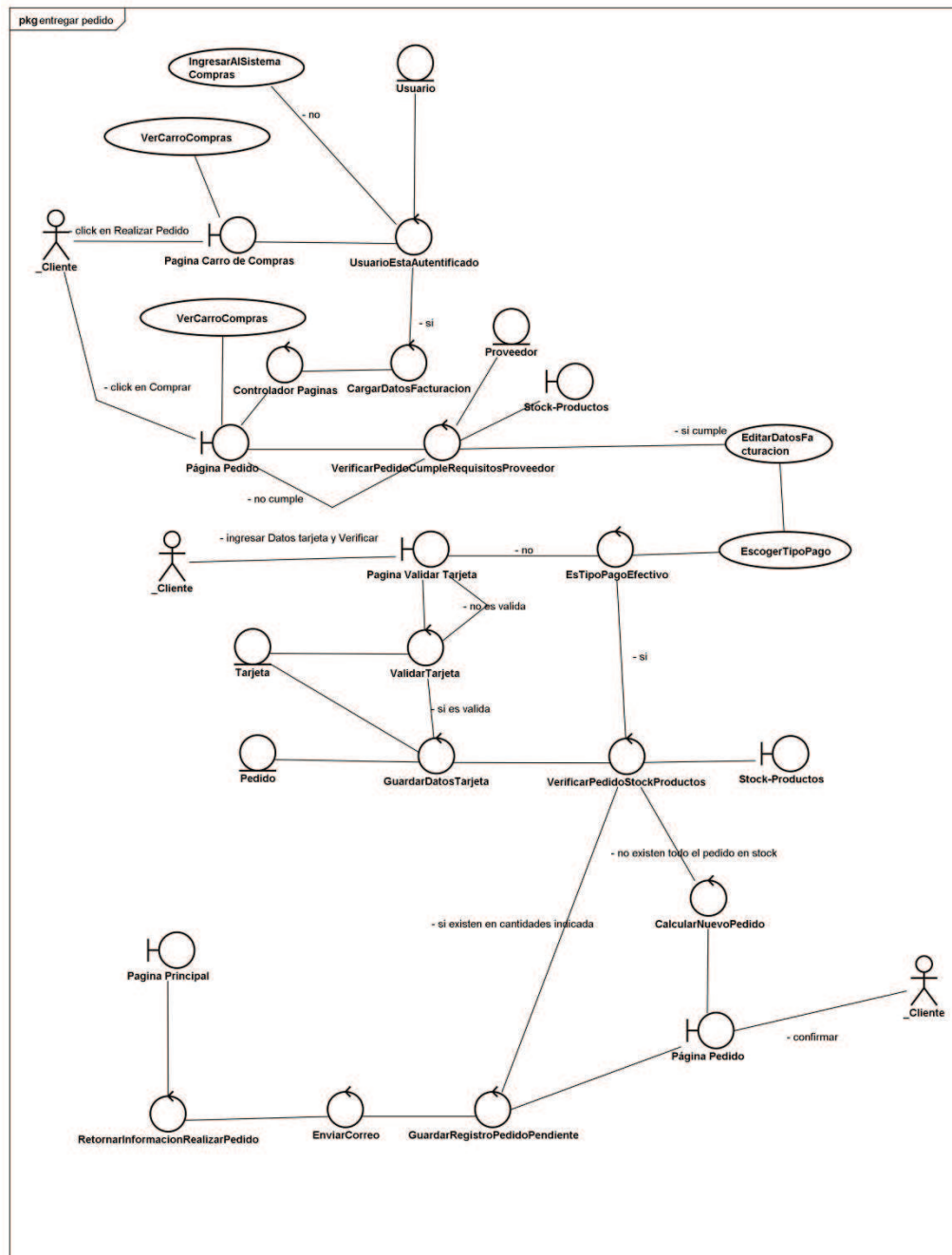


Figura 2. 24 Análisis de Robustez para el Caso de Uso RealizarPedido

## Diagrama de robustez Caso de uso ver Carro de compras

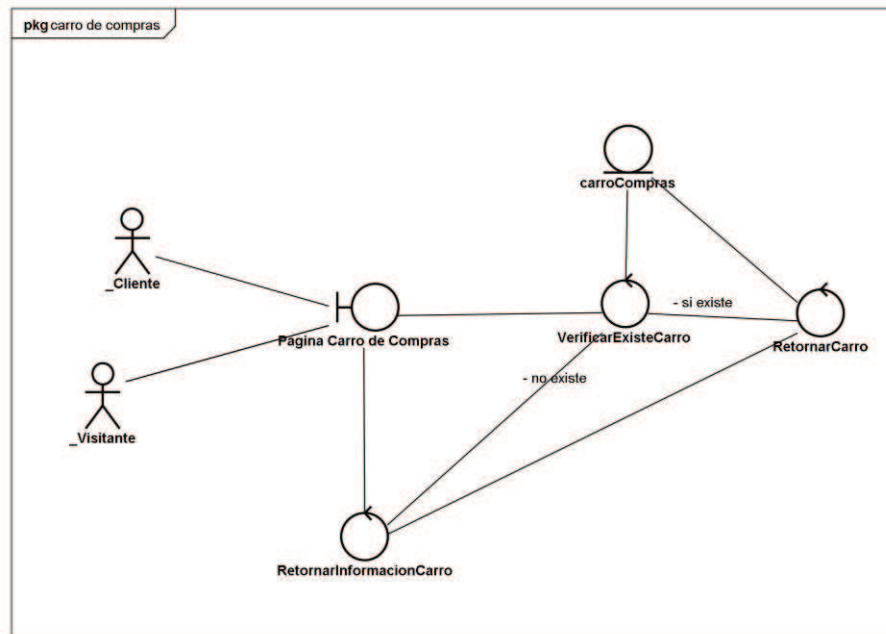


Figura 2. 25 Análisis de robustez para el caso de uso VerCarroCompras

## Diagrama de robustez Caso de uso editar parámetros generales Proveedor

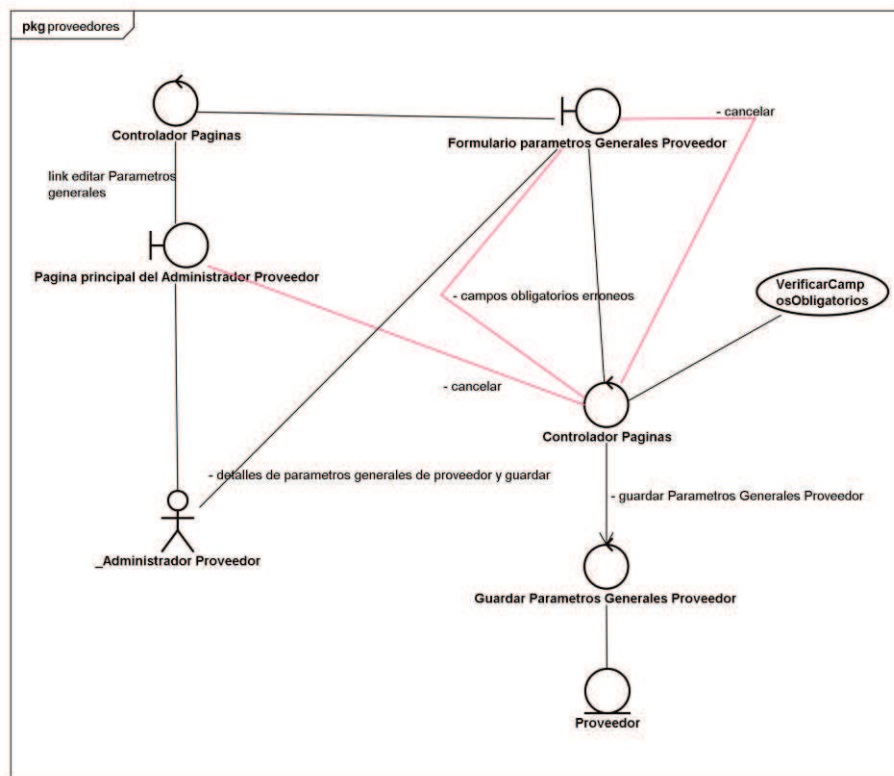


Figura 2. 26 Análisis de Robustez para el Caso de Uso EditarParametroGeneralesProveedor

## Diagrama de robustez Caso de uso ingresar Producto

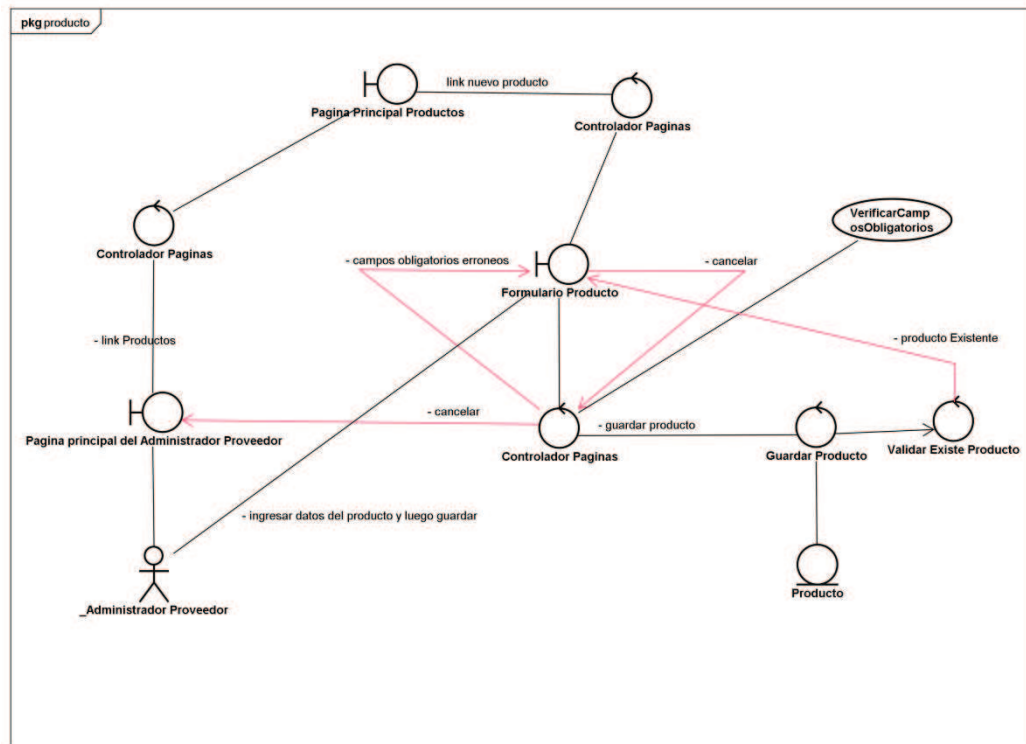


Figura 2. 27 Análisis de Robustez para el Caso de Uso IngresarProducto

### 2.2.1.2 Arquitectura de la aplicación

Con ayuda de las descripciones de los casos de uso y la realización de los diagramas de robustez, se diseñó la arquitectura de la aplicación. Se ha seleccionado una arquitectura multicapas, basada en el patrón MVC (Modelo-Vista-Controlador), en el que se incorporaran varias tecnologías como se muestra en la Figura 2.28:

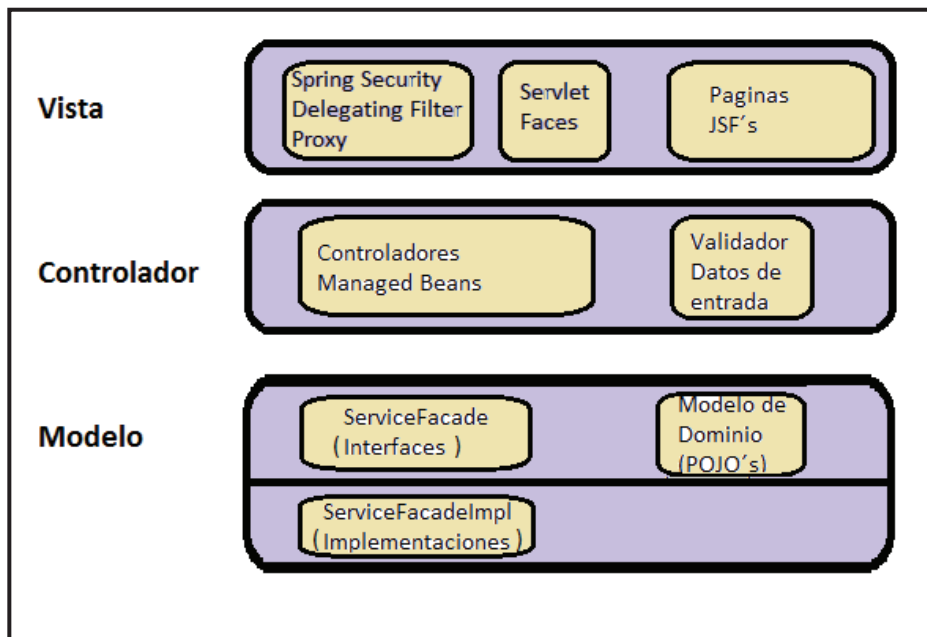


Figura 2. 28 Arquitectura de la aplicación shopping food

#### Capa de Vista

*Java Servlet Faces JSF:* La tecnología JSF provee de los componentes HTML necesarios para la realización de las interfaces con las que interactuará el usuario.

*Spring:* El uso de Spring en esta capa, es fundamental, ya que ayuda en la implementación de los mecanismos de seguridad tales como autenticación y autorización de acceso a los recursos de la aplicación.



## **Capa del Controlador**

*JSF Managed Beans:* Los componentes gráficos de JSF son controlados a través de objetos llamados managed beans, que se encargan de procesar las peticiones de los clientes. Además los managed Beans se utilizan para establecer la navegación entre las páginas de la aplicación.

## **Capa del Modelo**

*Enterprise Java Bean EJB:* Los beans enterprise implementan los métodos de la lógica del negocio, que pueden ser invocados por clientes remotos para acceder a los servicios proporcionados para la aplicación.

*EJB JPA (Java Persistence API):* JPA en la arquitectura adquirida ayuda con la abstracción del mapeo objeto-relacional, permitiendo trabajar con objetos en la aplicación para luego transformarlos a registros de la base de datos relacional. Se encarga de proveer las funciones para realizar las operaciones de lectura, inserción, actualización y borrado de los datos almacenados en la base de datos.

### **2.2.1.3 Revisión del diseño preliminar**

Se ha verificado la congruencia de los diagramas de robustez con la descripción de los casos de uso, además también se ha actualizado el modelo de dominio. Con los cambios realizados, el modelo de dominio queda como se ve en la Figura 2.29.

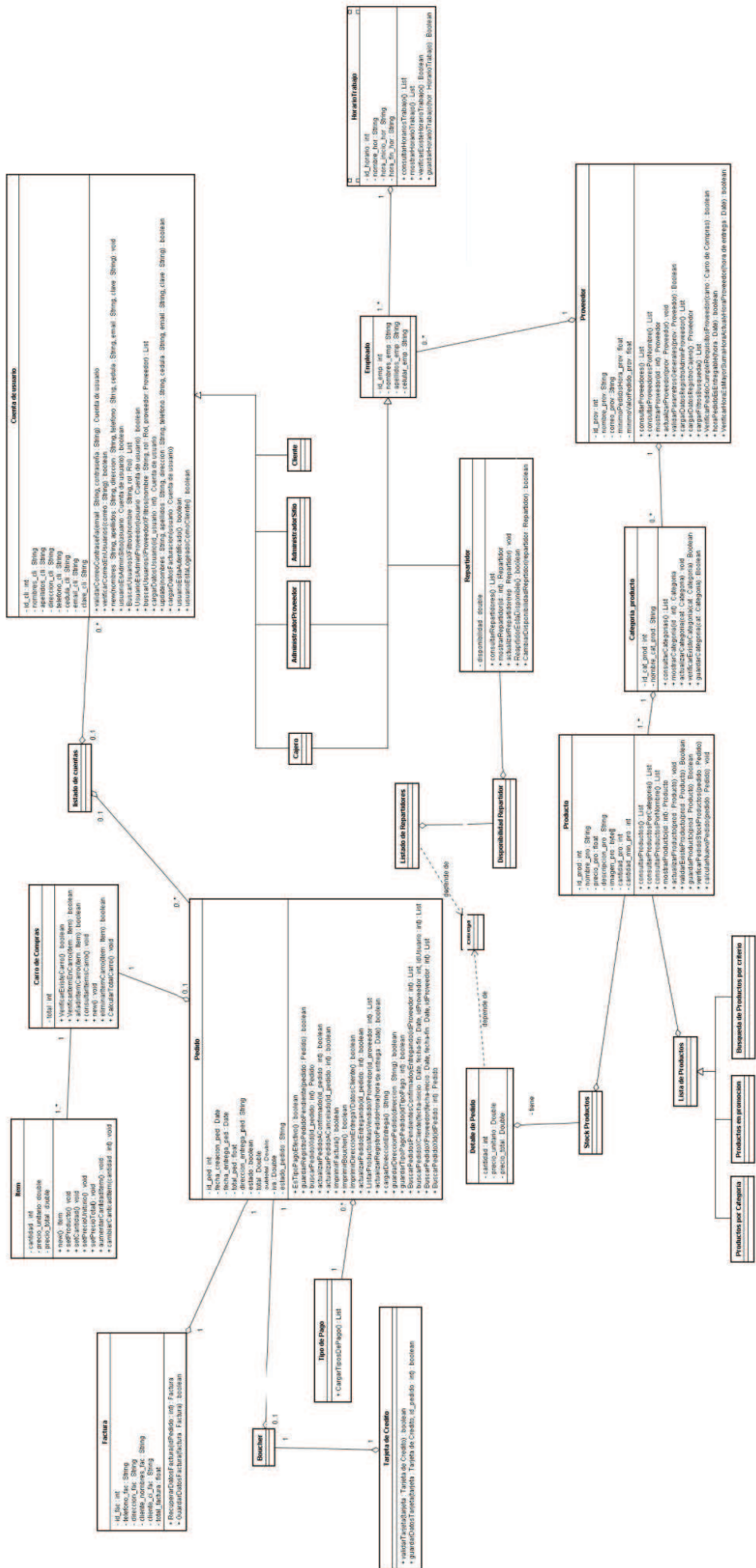


Figura 2. 29 Revisión del diseño preliminar del modelo de dominio

## 2.2.2 DISEÑO DETALLADO

En esta fase se realizará los diagramas de secuencia con el fin de identificar los métodos de las clases del modelo de dominio.

### 2.2.2.1 Diagramas de secuencia

A continuación se muestran 5 diagramas de secuencia desde la figura 2.30 a la 2.34, y los demás se encuentran en el documento completo del CD anexo.

#### Diagrama de robustez Caso de uso registrar usuario administrador proveedor

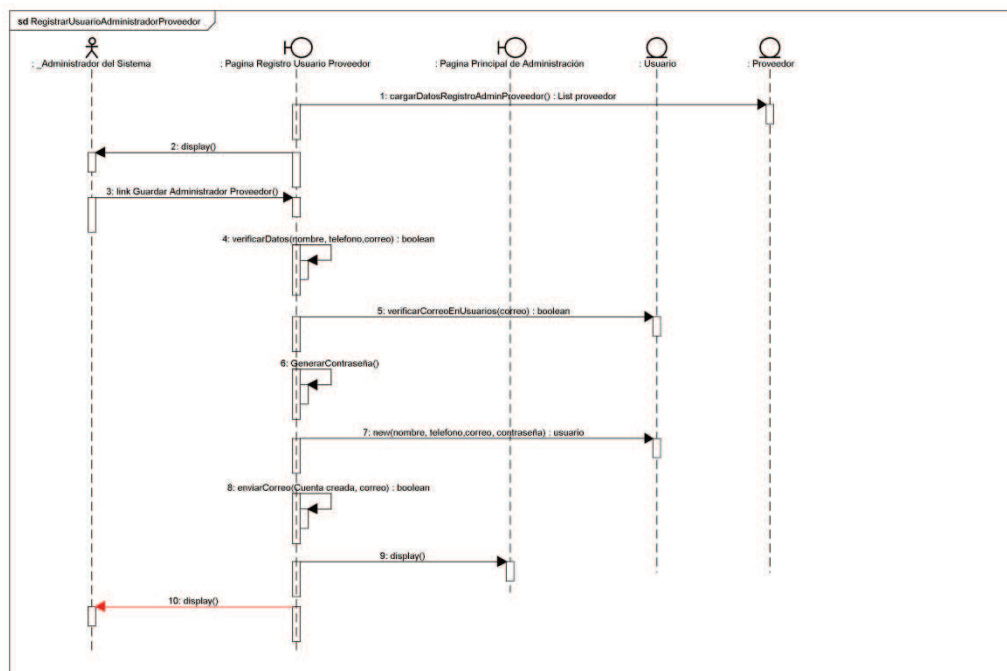


Figura 2. 30 Diagrama de secuencia para el caso de uso RegistrarUsuarioAdministradorProveedor

## Diagrama de secuencia Caso de uso realizar Pedido

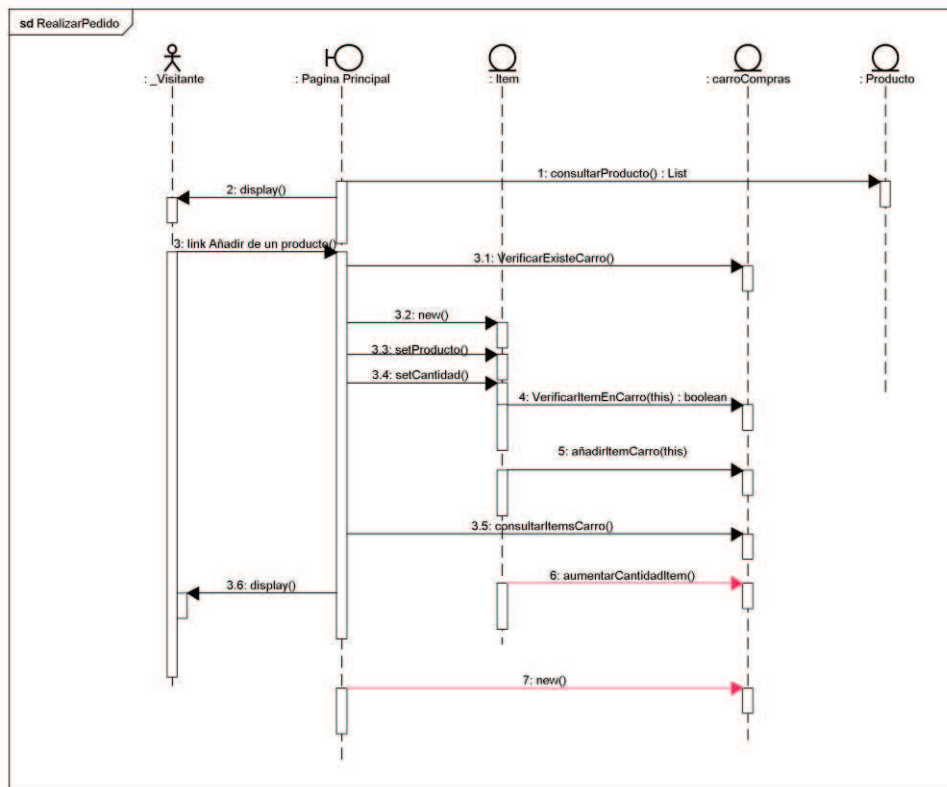


Figura 2. 31 Diagrama de secuencia para el caso de uso RealizarPedido

## Diagrama de secuencia Caso de uso ver Carro de compras

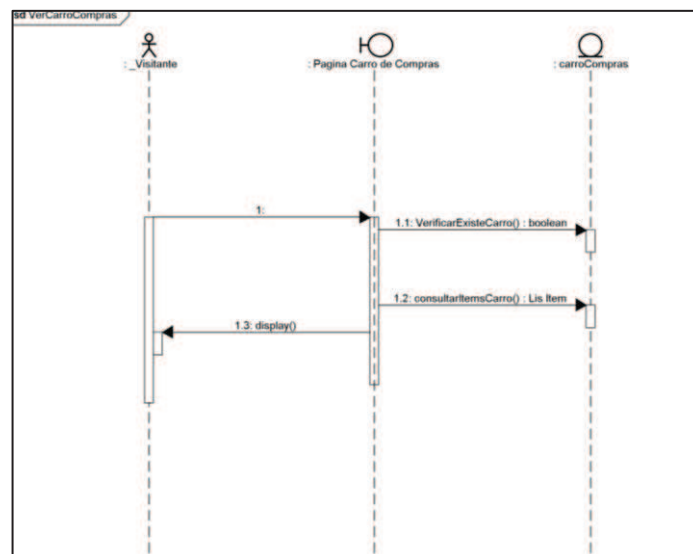


Figura 2. 32 Diagramas de secuencia para el caso de uso VerCarroCompras

## Diagrama de secuencia Caso de uso editar parámetros generales Proveedor

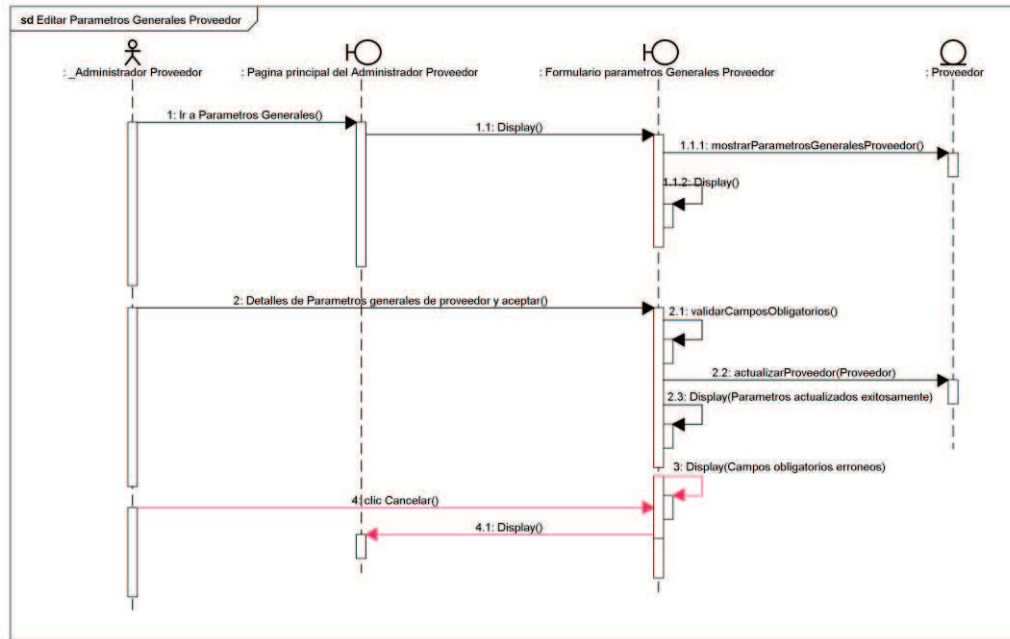


Figura 2. 33 Diagrama de secuencia para el caso de uso Editar Parámetros Generales Proveedor

## Diagrama de secuencia Caso de uso ingresar Producto

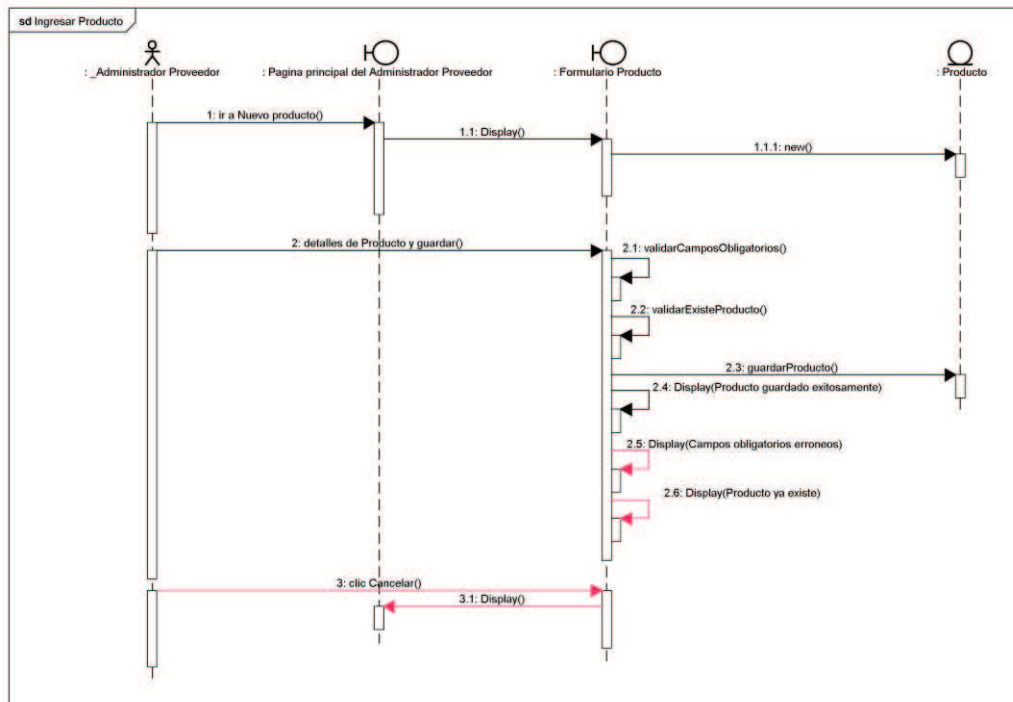


Figura 2. 34 Diagrama de secuencia para el caso de uso Ingresar Producto

### **2.2.2.2 Diseño del Diagrama de Clases**

Una vez que se han realizado los diagramas de secuencia se identificarán los métodos junto con los atributos y objetos faltantes en el modelo de dominio que se describe en la sección 2.1.2. Con las mejoras realizadas se ha elaborado el diagrama de clases expuesto en la Figura 2.35.



## **2.3 IMPLEMENTACIÓN DEL SISTEMA**

Finalizado el análisis del sistema que se realizó en la sección 2.2, en este apartado se debe continuar con la parte técnica del proyecto, es decir elaborar el sistema en base a todo lo que se ha diagramado en la etapa de diseño.

### **2.3.1 CÓDIGO DEL SISTEMA**

En base a la arquitectura definida para la aplicación se ha codificado los objetos de la base de datos, interfaces y estilos web, y el código de los servicios de negocio.

#### **2.3.1.1 Esquema de Base de Datos.**

En la Figura 2.36 se ha definido el Esquema Entidad/Relación de la Base de Datos, donde se puede observar cada una de las Entidades y sus atributos con su respectivo dominio. Además que se puede identificar la clave primaria de cada Entidad y las relaciones entre ellas.





### 2.3.1.2 Código de Interfaces.

Para el desarrollo de las interfaces de la aplicación se usó la implementación del estándar JSF 2.0 Primefaces versión 3.0. Además se ha realizado un diseño web en la generación de las interfaces, dando como resultado templates, páginas y componentes.

#### *Templates*

Se ha utilizado 2 templates para la realización de las interfaces, una para la presentación de los productos de los restaurantes (UserShoppingFood.xhtml) y la otra para páginas alternas (indexRegistrarse.xhtml).

#### *Páginas*

Las páginas web que se utilizaron para la presentación son las siguientes:

- index.xhtml
- restaurante.xhtml
- login.xhtml
- registrarse.xhtml
- cuenta\_recuperar.xhtml
- forma\_pago.xhtml

Todo el código de las páginas web que contiene la aplicación se encuentra disponible en el CD Anexo.

#### *Estilos CSS.*

Para la elaboración de las páginas se ha creado 3 estilos que se utilizarán para las páginas del administrador del sitio, del administrador proveedor, del cajero proveedor y para las páginas de pedidos de productos. El nombre de los estilos se los muestra en la tabla 2.6.

Páginas	Nombre del estilo
administrador del sitio	AdminSiteStyle.css
administrador proveedor	AdminSiteStyle.css
cajero proveedor	AdminSiteStyle.css
usuario registrado y visitante	UserShoppingFood.css y registro.css

Tabla 2. 6 Definición de estilos para las páginas del sistema

Todo el código de los estilos para las páginas web que contiene la aplicación se encuentra disponible en el CD Anexo.

### 2.3.1.3 Código servicios de Negocio

Para codificar los servicios de negocio que se utilizarán en el sistema se debe crear los objetos del modelo de dominio que representan las tablas del esquema de base de datos. También se debe crear para cada objeto una clase donde se implemente la lógica de negocio y que será expuesta a través de EJB's que serán invocados desde las páginas JSF.

*Entidades:* Son las clases que representan las Entidades del Negocio. Las Entidades tienen una conexión directa con las tablas de la base de Datos.

Las Entidades que se utilizaron en el presente proyecto son las siguientes:

- Categoria
- CuentaEmp
- CuentaUsuario
- DetallePedido
- EmpleadoEnHorario
- EmpleadoRepartidor
- Factura
- Pedido
- Producto
- Proveedor
- Rol

- RolCuenta
- ServicioCorreo
- ServicioPaypal
- TipoPago

Todo el código de las entidades que contiene la aplicación se encuentra disponible en el CD Anexo.

*Clases Facade:* Son las Clases Java que contienen la lógica de negocio ya sea para crear, seleccionar, actualizar, borrar u otro servicio que necesite un objeto.

Las Clases Facade que se utilizaron en el presente proyecto son las siguientes:

- AbstractFacade
- CategoriaFacade
- CuentaEmpFacade
- CuentaUsuarioFacade
- DetallePedidoFacade
- EmpleadoEnHorarioFacade
- EmpleadoRepartidorFacade
- FacturaFacade
- PedidoFacade
- ProductoFacade
- ProveedorFacade
- RolFacade
- RolCuentaFacade
- ServicioCorreoFacade
- ServicioPaypalFacade
- TipoPagoFacade

Todo el código de las clases Facade que contiene la aplicación se encuentra disponible en el CD Anexo.

*Clases Controller:* Son las Clases Java que permiten actuar como controladores en una Arquitectura MVC. Contienen métodos que darán una respuesta a las peticiones web y que se comunicarán con los servicios Facade de la aplicación.

Las Clases Controller que se utilizaron en el presente proyecto son las siguientes:

- AuthenticationBean
- CarController
- AbstractController
- CategoriaController
- CuentaEmpController
- CuentaUsuarioController
- DetallePedidoController
- EmpleadoEnHorarioController
- EmpleadoRepartidorController
- FacturaController
- PedidoController
- ProductoController
- ProveedorController
- RolController
- RolCuentaController
- ServicioCorreoController
- ServicioPaypalController
- TipoPagoController

Todo el código de las clases Controller que contiene la aplicación se encuentra disponible en el CD Anexo. A continuación se muestra como ejemplo el código del controlador del carro de compras de la aplicación.

*CarController*

```
package ec.edu.epn.fis.shoppingfood.controller;
```

```
import java.io.Serializable;
```

```
import java.util.Map;
```

```

import javax.faces.bean.ManagedBean;
import javax.faces.bean.SessionScoped;
import javax.faces.context.FacesContext;
import javax.faces.event.ActionEvent;
import ec.edu.epn.fis.shoppingfood.entities.Cart;
import ec.edu.epn.fis.shoppingfood.entities.CartItem;
import ec.edu.epn.fis.shoppingfood.entities.Producto;

@ManagedBean(name = "carController")
@SessionScoped
public class CarController implements Serializable {

    /**
     * representa el carrito de compras
     */
    private Cart cart;

    private Producto currentProd;

    private CartItem currentItem;
    private int numeroItems;

    public CartItem getCurrentItem() {
        return currentItem;
    }

    public Producto getCurrentProd() {
        return currentProd;
    }

    public CarController() {
        cart = new Cart();
    }

    public Cart getCart() {
        return cart;
    }

    public void setCart(Cart cart) {
        this.cart = cart;
    }

    public void addToCartAction(ActionEvent event) {
        // pararecuperar un objeto que ha sido enviado como f:attribute
        currentProd = (Producto)
            event.getComponent().getAttributes().get("selectedProd");
        getCart().addToCart(new CartItem(currentProd, 1));
    }

    public String addToCartAction(Producto producto) {
        currentProd = (Producto) producto;
        getCart().addToCart(new CartItem(currentProd, 1));
        return null;
    }
}

```

```
public void deleteItemAction(ActionEvent event) {
    currentItem = (CartItem)
        event.getComponent().getAttributes().get("selectedItem");
    getCart().removeFromCart(currentItem);
}

public void deleteAllItemsAction(ActionEvent event) {
    System.out.println("entree: eliminar todos "+event.toString());

    getCart().cleanCart();
}

    public int getNumeroItems() {

        numeroItems = cart.getNumeroItems();
        return numeroItems;
    }

    public void setNumeroItems(int numeroItems) {
        this.numeroItems = numeroItems;
    }

}
```

#### 2.3.1.4 Seguridad de la Información en la aplicación

“Seguridad de la información es el proceso de protección de la disponibilidad, la privacidad y la integridad de los datos. Ningún sistema de seguridad es infalible, pero es crítico tomar pasos básicos y prácticos para una buena seguridad de la información”. [22]

En el presente proyecto se han implementado algunos lineamientos para proteger la información sensible del negocio, recordando que el monitoreo para la seguridad de la información debe ser un proceso continuo.

##### *Seguridad a nivel de Transporte HTTPS.*

Hyper Text Transfer Protocol Secure (HTTPS) es una versión segura del protocolo de transferencia de hipertexto (http). HTTPS permite transacciones seguras de comercio electrónico, como banca en línea. Cuando un usuario se conecta a un sitio web a través de HTTPS, el sitio web encripta la sesión con un certificado digital. Un usuario puede saber si están conectados a una página web segura si la URL del sitio web comienza con https:// en lugar de http://. [23]

Para el presente proyecto se ha utilizado el Servidor de aplicaciones JBoss, por lo tanto para que la aplicación utilice un canal seguro (HTTPS) se deben realizar dos configuraciones:

1. Habilitar HTTPS en JBOSS 7.

Para habilitar la seguridad en el servidor de aplicaciones modificar el archivo **standalone.xml** que se encuentra en la ruta:  
**\$JBOSS\_HOME\standalone\configuration\**

---

[22] <http://www.wisegEEK.com/what-is-information-security.htm>

[23] <http://www.instantssl.com/ssl-certificate-products/https.html>



1.1. Como muestra la Figura 2.37 ubicarse en la sección:

```
<subsystem xmlns="urn:jboss:domain:web:1.1">
```

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host" native="false">
  <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
  <virtual-server name="default-host" enable-welcome-root="true">
    <alias name="localhost"/>
    <alias name="example.com"/>
  </virtual-server>
</subsystem>
```

Figura 2. 37 Sección urn:jboss:domain:web:1.1 del archivo Standalone.xml

Aquí se deberá agregar un nuevo conector que permitirá habilitar HTTPS en el servidor de aplicaciones.

Definir los repositorios de certificados digitales keystore y truststore con los cuales se establecerán las comunicaciones por HTTPS con JBOSS. Ver Figura 2.38.

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host" native="false">
  <connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
  <connector name="https" protocol="HTTP/1.1" scheme="https" socket-binding="https" secure="true">
    <ssl name="https" password="XXXXXXXX" certificate-key-file="C:\imagenTest\keystore.jks"
      protocol="TLSv1" ca-certificate-file="C:\imagenTest\truststore.jks"/>
  </connector>
  <virtual-server name="default-host" enable-welcome-root="true">
    <alias name="localhost"/>
    <alias name="example.com"/>
  </virtual-server>
</subsystem>
```

Figura 2. 38 Configuración del Conector https en Jboss

1.2. Adicionalmente establecer en la sección <socket-binding-group name="standard-sockets" el puerto por el cual correrá la aplicación por HTTPS, por defecto está el puerto 8443 como se observa en la Figura 2.39.

```
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="{jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management" port="{jboss.management.native.port:9999}"/>
  <socket-binding name="management-http" interface="management" port="{jboss.management.http.port:9990}"/>
  <socket-binding name="management-https" interface="management" port="{jboss.management.https.port:9443}"/>
  <socket-binding name="ajp" port="8009"/>
  <socket-binding name="http" port="8080"/>
  <socket-binding name="https" port="8443"/>
  <socket-binding name="osgi-http" interface="management" port="8090"/>
  <socket-binding name="remoting" port="4447"/>
  <socket-binding name="txn-recovery-environment" port="4712"/>
  <socket-binding name="txn-status-manager" port="4713"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="25"/>
  </outbound-socket-binding>
</socket-binding-group>
```

Figura 2. 39 Definición del Puerto https para Jboss

2. Habilitar HTTPS en aplicación web con Spring

Se puede habilitar la seguridad usando HTTPS para diferentes secciones de la aplicación web. Para esto es indispensable que se haya habilitado HTTPS en el servidor de aplicaciones donde se instalará la aplicación.

Spring Security permite realizar la protección de ciertos recursos de la aplicación mediante la configuración llamada **intercept-url**. Esta propiedad además permite definir a que usuarios se les concederá el acceso a ciertos recursos. Esta actividad se denomina autorización.

En la propiedad **intercept-url**, existe un atributo denominado **requires-channel**, que se encarga de definir qué recursos necesitarán HTTPS o HTTP o que pueden estar disponibles en ambos protocolos.

Para la aplicación el módulo de administración de shopping food se accederá usando HTTPS, el módulo de servicio al cliente funcionará sobre HTTP y la sección de confirmación del pedido por HTTPS.

### *Autenticación y autorización con Spring Security.*

Spring Security es un framework que proporciona servicios de seguridad para aplicaciones J2EE.

Spring Security proporciona un mecanismo de seguridad declarativo independiente del entorno donde se despliega la aplicación, brindando una solución completa para los dos requisitos más importantes de seguridad: Autenticación y Autorización. [24]

Entre las características más importantes están:

- Solución madura (versiones estables desde 2003)
- No intrusivo

---

[24]<http://wiki.fluxit.com.ar/display/PUBLIC/Spring+Security>

- Facilidad de configuración y parametrización
- Totalmente extensible
- Gran cantidad de documentación y soporte
- Integración tanto con aplicaciones web como aplicaciones Desktop

### *Autenticación*

La autenticación es el proceso por el que se comprueba la identidad de alguien o algo, para ver si es lo que dice ser. Ese "alguien" o "algo" se denomina principal. La autenticación requiere pruebas de identidad, denominadas credenciales. Por ejemplo, una aplicación cliente puede presentar una contraseña como sus credenciales. Si la aplicación cliente presenta las credenciales correctas, se asume que es quien dice ser. [25]

### *Autenticación con Spring Security*

El principal componente del proceso de autenticación es el AuthenticationManager. A continuación se detallan los componentes compartidos que son centrales para la autenticación:

- *SecurityContextHolder*: Almacena los datos del Principal y sus roles dentro del contexto de seguridad. Por defecto el contexto utiliza ThreadLocal para almacenar la información lo que significa que el contexto está siempre disponible desde cualquier método del mismo Thread.
- *Authentication*: Representación del Principal con sus respectivos roles. Es el objeto que se guarda dentro del SecurityContextHolder.
- *UserDetailsService*: el objeto AuthenticationManager delega en el objeto UserDetails el trabajo de obtener la información del usuario desde la fuente de datos especificada. Por ejemplo, si el repositorio de usuarios, credenciales y

---

[25] [http://elenet.ws/art%C3%ADculos/8973-seguridad\\_en\\_servicios\\_web.\\_autenticaci%C3%B3n\\_y\\_authorized.\\_interoperabilidad.html](http://elenet.ws/art%C3%ADculos/8973-seguridad_en_servicios_web._autenticaci%C3%B3n_y_authorized._interoperabilidad.html)

roles de la aplicación está en una Base de Datos, el objeto UserDetails será el encargado de realizar las consultas necesarias para obtener la información del Principal a partir de dicha Base de Datos. [25]

### *Implementación de Autenticación con Spring Security.*

La autenticación de la aplicación se ha implementado mediante el ingreso de un nombre de usuario y una contraseña.

Las contraseñas de los usuarios se han encriptado usando el algoritmo SHA-512, mismo que es un algoritmo de una sola vía es decir que no se puede recuperar el valor encriptado inicialmente.

El flujo de autenticación es el siguiente:

1. El usuario ingresa su nombre de usuario y contraseña
2. El sistema encripta la contraseña usando SHA
3. El sistema recupera la contraseña del usuario ingresado
4. Si los valores coinciden el sistema retorna autenticación exitosa, caso contrario el sistema retorna autenticación fallida

### *Autorización*

“La autorización, el proceso que permite determinar si se tiene o no acceso al recurso que se está solicitando, no está determinada por quién se es, sino por qué se es (lo que ocurre es que habitualmente, se necesita saber quién soy para saber qué soy). La pertenencia a un determinado grupo o categoría, o el hecho de estar en el ejercicio de un determinado cargo; el perfil al que se pertenece, en definitiva, determina el conjunto de privilegios de acceso que se tienen como usuario.” [26]

---

[26]<http://appsec.infonova.es/Seguridadintegrada/Desarrollodeaplicaciones/Autenticaci%C3%B3nautorizaci%C3%B3nyauditor%C3%ADa.aspx>

Existen tres mecanismos diferentes de autorización:

- Autorización basada en solicitudes web
- Autorización basada en AOP
- Autorización por dominio

En la Figura 2.40 se muestran los componentes involucrados en el proceso de Autorización de Spring Security:

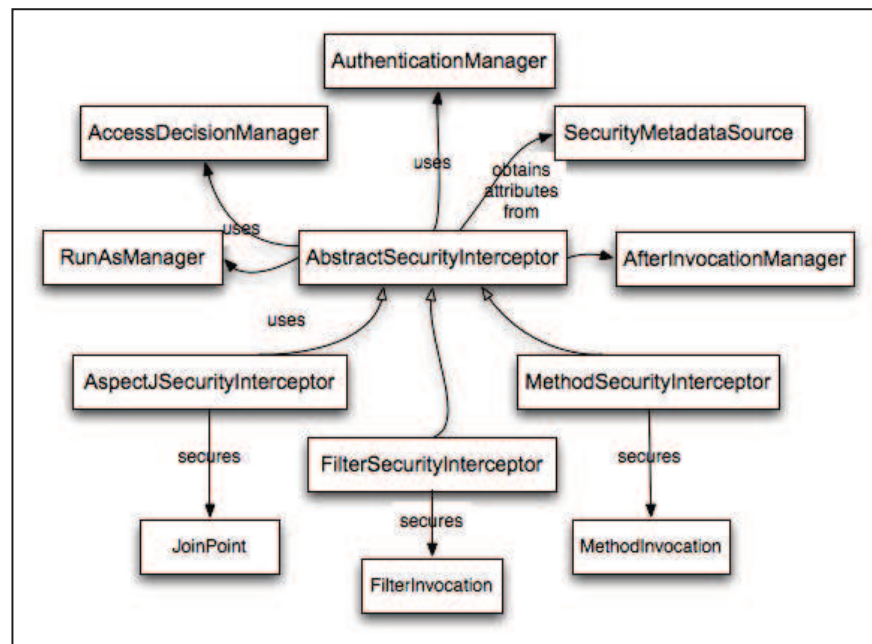


Figura 2. 40 Autorización de Spring Security

El principal componente dentro del modelo de autorización es `AbstractSecurityInterceptor`, el cual provee la funcionalidad básica de autorización.

Como se puede apreciar en la Figura 2.40, los objetos que se pueden asegurar con Spring Security son peticiones HTTP a través del `FilterSecurityInterceptor` e invocaciones a métodos mediante `MethodSecurityInterceptor` y `AspectJSecurityInterceptor`.

---

El `AbstractSecurityInterceptor` delega la decisión de autorización final en el `AccessDecisionManager`. Este componente implementa un método el cual recibe un objeto `Authentication`, un recurso seguro (url o ejecución de método) y una lista de autorizaciones. Con estos parámetros decide si se puede acceder al recurso seguro o no.

Spring Security define tres implementaciones de `AccessDecisionManager`:

- `AffirmativeBased`: Es necesario recibir solamente un voto positivo para dar acceso al recurso protegido.
- `ConsensusBased`: Es necesario que haya más votos positivos que negativos para dar acceso al recurso protegido.
- `UnanimousBased`: Se necesitan que todos los votos sean positivos para dar acceso al recurso protegido.

Para la decisión de los votos, el `AccessDecisionManager` utiliza un `AccessDecisionVoter`. Por defecto vienen dos implementaciones de `AccessDecisionVoter`:

- `RoleVoter`: Básicamente lo que realiza este componente es comprobar si el `Principal` contiene un rol para el recurso protegido, en tal caso emite un voto positivo, caso contrario emite un voto negativo.
- `AuthenticatedVoter`: Con este componente se puede diferenciar entre un acceso anónimo a un recurso, un acceso autenticado o un acceso mediante el mecanismo `remember-me`. Este componente es generalmente utilizado para recursos que necesitan tener un comportamiento diferente de acuerdo al tipo de acceso. Así por ejemplo se permite que un usuario anónimo pueda acceder a la página de login.

[24]

---

[24]<http://wiki.fluxit.com.ar/display/PUBLIC/Spring+Security>

### *Implementación de Autorización con Spring Security*

La autorización de acceso a los recursos de la aplicación se la realiza usando el componente <intercept-url> mismo que permite establecer mediante la definición de roles, la concesión de acceso.

Para la aplicación se ha definido las siguientes reglas de autorización:

- Acceso a datos de cuenta de usuario y realización de pedidos, rol ROL\_USER.
- Acceso a la página de inicio y creación de carro de compras, todos los usuarios.
- Acceso a páginas de administración de la aplicación, rol ROL\_ADMIN\_SITE.
- Acceso a páginas de administración de cada proveedor, rol ROL\_ADMIN.
- Acceso a páginas de los cajeros de cada proveedor, rol ROL\_CAJERO.

### *Establecer contraseñas seguras en los usuarios*

La principal manera de autenticación en aplicaciones web es a través del uso de credenciales de usuario y contraseña. La dificultad con este sistema de autenticación radica en que una persona autorizada solo necesita saber su usuario y su contraseña, de la misma forma cualquier otra persona puede ingresar con saber los mismos datos, sin importar como los obtuvo o si tiene autorización para su uso. Existen varios mecanismos para obtener contraseñas sin la necesidad de adivinarlas, el más popular se basa en usar palabras de diccionario o generar caracteres en secuencia hasta que se obtiene la contraseña de acceso deseada. La mejor forma de asegurarnos en contra de estos ataques es usando contraseñas complejas, al azar y tan largas como sea posible. [27]

---

[27]<http://www.isocron.net/blog/2011/08/02/la-importancia-de-usar-contrasenas-seguras>

Como parte de la autenticación y autorización se ha establecido que las contraseñas deben ser robustas, por lo cual los usuarios deben seguir los siguientes lineamientos al generar una contraseña: [28]

1. Mezclar caracteres alfabéticos (mayúsculas y minúsculas)
2. Al menos una letra mayúscula
3. Al menos un número
4. Deben ser largas, de 8 caracteres mínimo

### *Seguridad en pago con el uso de PAYPAL*

“PayPal permite a las empresas o consumidores que dispongan de correo electrónico enviar y recibir pagos en Internet de forma segura, cómoda y rentable. La red de PayPal se basa en la infraestructura financiera existente de cuentas bancarias y tarjetas de crédito para crear una solución global de pago en tiempo real. Le ofrecemos un servicio especialmente pensado para pequeñas empresas, vendedores por Internet, particulares y otros a los que no satisfacen los mecanismos de pago tradicionales”. [29]

PayPal permite generar botones de compra estáticos, únicamente generando el código fuente que puede ser pegado en una página HTML. Sin embargo, si se tiene bastantes productos que vender, no tiene sentido crear un botón por cada uno de ellos y se debería generar los botones dinámicamente.

En la generación dinámica de botones de compra, PayPal permite generar botones que estarán cifrados y firmados digitalmente para evitar posibles intromisiones. [30].

---

[28]<http://www.segu-info.com.ar/proteccion/clavesseguras.htm>

[29]<https://www.paypal.com/es/webapps/mpp/about>

[30]<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=botonPaypalCifrado>



Todo proceso de configuración para el uso de PayPal en la aplicación se encuentra disponible en el CD Anexo.

## **2.4 PRUEBAS**

Una vez finalizada la implementación de los servicios de negocio se debe proceder con las pruebas del desarrollo realizado.

### **2.4.1 PLAN DE PRUEBAS**

Para la aplicación se realizará las pruebas funcionales para las diferentes clases que implementan la lógica de negocio, y se realizarán pruebas de navegación de las páginas Web donde se verificará la indexación correcta de los enlaces.

Para realizar las pruebas funcionales se usará JUnit que es una librería java que facilita la realización de pruebas unitarias de los servicios. Para las pruebas de navegación se usará W3 Validator.

Las pruebas funcionales permiten validar que se ha realizado una implementación adecuada de las reglas de negocio.

#### **2.4.1.1 Pruebas unitarias**

Para la realización de las pruebas unitarias del presente proyecto se ha utilizado JUnit que es una herramienta de Java que permite automatizar las pruebas unitarias.

***JUnit.***- JUnit fue creado por Erich Gamma y Kent Beck y es utilizado en programación para hacer pruebas unitarias de aplicaciones Java. **[31]**

---

**[31]**<http://elverdaderoblogdelaob.wordpress.com/2009/07/04/pruebas-unitarias-en-java-junit/>

JUnit es un conjunto de clases (framework) que ayuda a realizar evaluaciones sobre el funcionamiento de los métodos de la clase, para probar si se comporta como se espera. Para evaluar un método en la prueba se envía un valor esperado y el resultado, en caso de que el resultado sea igual al valor esperado, la prueba es exitosa, caso contrario la prueba se considera como fallida.

*Listado de las pruebas unitarias de los métodos de negocio*

En la Tabla 2.7 se muestra los métodos de cada clase del negocio y el resultado de las pruebas obtenidas, cada clase tiene los 4 métodos fundamentales que son Crear (Create), Seleccionar (Find), Editar (Edit) y Borrar (Remove), además cada clase tiene métodos que permiten desarrollar la lógica del negocio.

CLASE	METODO	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
<b>Categoría</b>	Create	Categoría Creada	Categoría Creada	EXITOSO
	FindAll	Listado de 6 Categorías	Listado de 6 Categorías	EXITOSO
	Find	Categoría de id 1	Categoría de id 1	EXITOSO
	Count	6	6	EXITOSO
	Edit	Categoría de id 1 Editada	Categoría de id 1 Editada	EXITOSO
	Remove	Categoría de id 1 eliminada	Categoría de id 1 eliminada	EXITOSO
	FindByProv	3 Categorías de Proveedor de id=1	3 Categorías de Proveedor de id=1	EXITOSO
<b>CuentaEmp</b>	Create	Un registro de CuentaEmp creado	Un registro de CuentaEmp creado	EXITOSO
	FindAll	Listado de 4 CuentaEmp	Listado de 4 CuentaEmp	EXITOSO
	Find	CuentaEmp de id 1	CuentaEmp de id 1	EXITOSO
	Count	4	4	EXITOSO
	Edit	CuentaEmp de id 6 Editada	CuentaEmp de id 6 Editada	EXITOSO
	Remove	CuentaEmp de id 7 eliminada	CuentaEmp de id 7 eliminada	EXITOSO
	findByEmail	CuentaEmp de email = fredy888_san@hotmail.com	CuentaEmp de email = fredy888_san@hotmail.com	EXITOSO
<b>CuentaUsuario</b>	Create	Un registro de CuentaUsuario creado	Un registro de CuentaUsuario creado	EXITOSO
	FindAll	Listado de 6 CuentaUsuarios	Listado de 6 CuentaUsuarios	EXITOSO
	Find	CuentaUsuario de id 28	CuentaUsuario de id 28	EXITOSO
	Count	5	5	EXITOSO

	Edit	CuentaUsuario de id 28 Editada	CuentaUsuario de id 28 Editada	EXITOSO
	Remove	CuentaUsuario de id 28 eliminada	CuentaUsuario de id 28 eliminada	EXITOSO
	findByEmail	CuentaUsuario de email = esteban@hotmail.com	CuentaUsuario de email = esteban@hotmail.com	EXITOSO
<b>DetallePedido</b>	Create	Un registro de DetallePedido creado	Un registro de DetallePedido creado	EXITOSO
	FindAll	Listado de 105 DetallePedidos	Listado de 105 DetallePedidos	EXITOSO
	Find	DetallePedido de id 84	DetallePedido de id 84	EXITOSO
	Count	105	105	EXITOSO
	Edit	DetallePedido de id 84 Editado	DetallePedido de id 84 Editado	EXITOSO
	Remove	DetallePedido de id 84 eliminado	DetallePedido de id 84 eliminado	EXITOSO
	findByIdPedido	Listado de 3 DetallePedidos de idPedido=63	Listado de 3 DetallePedidos de idPedido=63	EXITOSO
<b>EmpleadoEnHorario</b>	Create	Un registro de EmpleadoEnHorario creado	Un registro de EmpleadoEnHorario creado	EXITOSO
	FindAll	Listado de 2 EmpleadoEnHorarios	Listado de 2 EmpleadoEnHorarios	EXITOSO
	Find	EmpleadoEnHorario de id 2	EmpleadoEnHorario de id 2	EXITOSO
	Count	2	2	EXITOSO
	Edit	EmpleadoEnHorario de id 2 Editada	EmpleadoEnHorario de id 2 Editada	EXITOSO
	Remove	EmpleadoEnHorario de id 2 eliminada	EmpleadoEnHorario de id 2 eliminada	EXITOSO
	findEmpByIdProv Est	Listado de 1 EmpleadoEnHorario de Proveedor = 1 y Estado = L	Listado de 1 EmpleadoEnHorario de Proveedor = 1 y Estado = L	EXITOSO
	findUnassignedEmp	Listado de 0 EmpleadoEnHorario de Proveedor = 1	Listado de 0 EmpleadoEnHorario de Proveedor = 1	EXITOSO
<b>EmpleadoRepartidor</b>	Create	Un registro de EmpleadoRepartidor creado	Un registro de EmpleadoRepartidor creado	EXITOSO
	FindAll	Listado de 6 EmpleadoRepartidores	Listado de 6 EmpleadoRepartidores	EXITOSO
	Find	EmpleadoRepartidor de id 1	EmpleadoRepartidor de id 1	EXITOSO
	Count	6	6	EXITOSO

	Edit	EmpleadoRepartidor de id 1 Editada	EmpleadoRepartidor de id 1 Editada	EXITOSO
	Remove	EmpleadoRepartidor de id 1 eliminada	EmpleadoRepartidor de id 1 eliminada	EXITOSO
	findByldProv	Listado de 2 Repartidores de Proveedor = 1	Listado de 2 Repartidores de Proveedor = 1	EXITOSO
<b>Factura</b>	Create	Un registro de Factura creado	Un registro de Factura creado	EXITOSO
	FindAll	Listado de 6 Facturas	Listado de 6 Facturas	EXITOSO
	Find	Factura de id 1	Factura de id 1	EXITOSO
	Count	6	6	EXITOSO
	Edit	Factura de id 1 Editada	Factura de id 1 Editada	EXITOSO
	Remove	Factura de id 1 eliminada	Factura de id 1 eliminada	EXITOSO
<b>Pedido</b>	Create	Un registro de Pedido creado	Un registro de Pedido creado	EXITOSO
	FindAll	Listado de 6 Pedidos	Listado de 6 Pedidos	EXITOSO
	Find	Pedido de id 1	Pedido de id 1	EXITOSO
	Count	35	35	EXITOSO
	Edit	Pedido de id 1 Editada	Pedido de id 1 Editada	EXITOSO
	Remove	Pedido de id 1 eliminada	Pedido de id 1 eliminada	EXITOSO
	countByCliente	10	10	EXITOSO
	proveedorByPedido	Banquetes Norte	Nombre del Proveedor del Pedido = 74	EXITOSO
getAllByldProv	Listado de 10 Pedidos de Proveedor = 1	Listado de 10 Pedidos de Proveedor = 1	EXITOSO	
<b>Producto</b>	Create	Un registro de Producto creado	Un registro de Producto creado	EXITOSO
	FindAll	Listado de 6 Productos	Listado de 6 Productos	EXITOSO
	Find	Producto de id 1	Producto de id 1	EXITOSO
	Count	6	6	EXITOSO
	Edit	Producto de id 1 Editada	Producto de id 1 Editada	EXITOSO
	Remove	Producto de id 1 eliminada	Producto de id 1 eliminada	EXITOSO
	findByldProv	Producto de id 1	Producto de id 1	EXITOSO
	findProductoVendidosByldProv	Listado de 5 Productos de Proveedor = 1	Listado de 5 Productos de Proveedor = 1	EXITOSO
	findByCategoria	Listado de 4 Productos de Categoria = 5	Listado de 4 Productos de Categoria = 5	EXITOSO
	findByCategoriaProveedor	Listado de 4 Productos de Categoria = 5 y Proveedor = 2	Listado de 4 Productos de Categoria = 5 y Proveedor = 2	EXITOSO
findListProductByCategory	Listado de 4 Productos de Categoria = 5	Listado de 4 Productos de Categoria = 5	EXITOSO	

	findByProvNombre	Listado de 3 Productos de Proveedor = 1 y nombre = cama	Listado de 3 Productos de Proveedor = 1 y nombre = cama	EXITOSO
	findByProvPromocion	Listado de 5 Productos en Promoción de Proveedor = 1	Listado de 5 Productos en Promoción de Proveedor = 1	EXITOSO
<b>Proveedor</b>	Create	Un registro de Proveedor creado	Un registro de Proveedor creado	EXITOSO
	FindAll	Listado de 3 Proveedores	Listado de 3 Proveedores	EXITOSO
	Find	Proveedor de id 3	Proveedor de id 3	EXITOSO
	Count	3	3	EXITOSO
	Edit	Proveedor de id 3 Editada	Proveedor de id 3 Editada	EXITOSO
	Remove	Proveedor de id 3 eliminada	Proveedor de id 3 eliminada	EXITOSO
	findByProv	Proveedor de id 3	Proveedor de id 3	EXITOSO
<b>Rol</b>	Create	Un registro de Rol creado	Un registro de Rol creado	EXITOSO
	FindAll	Listado de 4 Roles	Listado de 4 Roles	EXITOSO
	Find	Rol de id 1	Rol de id 1	EXITOSO
	Count	4	4	EXITOSO
	Edit	Rol de id 1 Editada	Rol de id 1 Editada	EXITOSO
	Remove	Rol de id 1 eliminada	Rol de id 1 eliminada	EXITOSO
<b>RolCuenta</b>	Create	Un registro de RolCuenta creado	Un registro de RolCuenta creado	EXITOSO
	FindAll	Listado de 7 RolCuentas	Listado de 7 RolCuentas	EXITOSO
	Find	RolCuenta de id 1	RolCuenta de id 1	EXITOSO
	Count	7	7	EXITOSO
	Edit	RolCuenta de id 7 Editada	RolCuenta de id 7 Editada	EXITOSO
	Remove	RolCuenta de id 7 eliminada	RolCuenta de id 7 eliminada	EXITOSO
	findByNombreRol	Lista de 1 Rol_cuenta de Rol = ROL_ADMIN	Lista de 1 Rol_cuenta de Rol = ROL_ADMIN	EXITOSO
findByNombreRol	Lista de 1 Rol_cuenta de Rol = ROL_ADMIN y ProveedorId = 3	Lista de 1 Rol_cuenta de Rol = ROL_ADMIN y ProveedorId = 3	EXITOSO	
<b>TipoPago</b>	Create	Un registro de TipoPago creado	Un registro de TipoPago creado	EXITOSO
	FindAll	Listado de 2 TipoPagos	Listado de 2 TipoPagos	EXITOSO
	Find	TipoPago de id 1	TipoPago de id 1	EXITOSO
	Count	2	2	EXITOSO
	Edit	TipoPago de id 1 Editada	TipoPago de id 1 Editada	EXITOSO
	Remove	TipoPago de id 1 eliminada	TipoPago de id 1 eliminada	EXITOSO

**Tabla 2. 7 Pruebas unitarias de los métodos de negocio**

Cómo se puede visualizar en la Tabla 2.7 las pruebas son exitosas y para cumplir esto se ha realizado modificaciones en ciertos métodos, por lo tanto las funciones básicas de la lógica del negocio se realizan de forma eficaz y correcta.

### *Capturas con los resultados*

Tomando como muestra las 2 clases más relevantes del negocio (Pedido y Producto) se ha realizado capturas de pantalla con los resultados de las pruebas unitarias con JUnit.

### *Pedido*

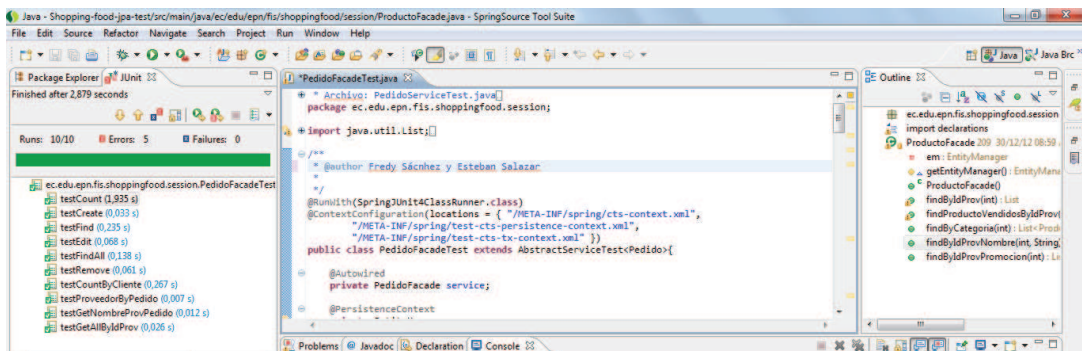


Figura 2. 41 Pruebas Unitarias de Pedido con JUnit

Para el caso de Pedido tiene 10 métodos de los cuáles como se puede ver en la Figura 2.41 las pruebas unitarias sobre los 10 métodos han sido exitosas.

### *Producto*

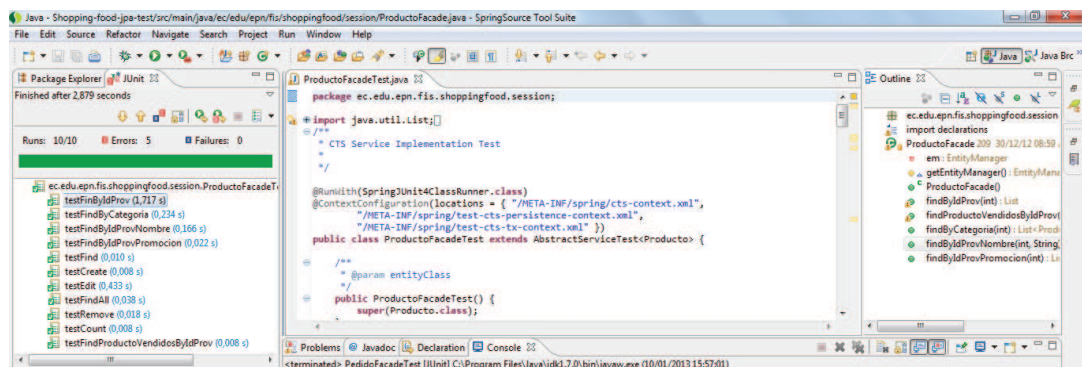


Figura 2. 42 Pruebas Unitarias de Producto con JUnit

Para el caso de Producto tiene 11 métodos, y como resultado de las pruebas unitarias como se ve en la Figura 2.42 han sido exitosas.

#### 2.4.1.2 Pruebas de Páginas Web

Las pruebas de Páginas web se realizan para comprobar que el código de las interfaces esté escrito según los estándares de la web, para lo cual se ha utilizado el componente Validator de la W3.

*Validator W3.*- Es una herramienta de la W3 (Consortio World Wide Web) que es una comunidad internacional que desarrolla estándares abiertos para asegurar el crecimiento a largo plazo de la Web. [32]

Es una herramienta online disponible en <http://validator.w3.org/> que comprueba la validez de marcado de documentos Web en HTML, XHTML, SMIL, MathML, etc. [33]

#### *Desarrollo de las Pruebas*

Todo el proceso para la realización de las pruebas de las páginas de la aplicación se encuentra disponible en el CD Anexo.

Las páginas más representativas sobre las cuáles se realizó las pruebas son:

- <http://localhost:8080/ShoppingFood-WAR/index.jsf>
- <http://localhost:8080/ShoppingFood-WAR/restaurante.xhtml>
- <http://localhost:8080/ShoppingFood-WAR/login.xhtml>
- <http://localhost:8080/ShoppingFood-WAR/secciones/secure/pedido.xhtml>

---

[32] Y [33] <http://validator.w3.org/>

En la Tabla 2.8 se tabularon los resultados de las pruebas realizadas.

Página	N° errores	Tipo de Error There is no attribute:			
		role	aria-haspopup	data-ri	Value
<a href="http://localhost:8080/ShoppingFood-WAR/index.jsf">http://localhost:8080/ShoppingFood-WAR/index.jsf</a>	10	7	1	1	1
<a href="http://localhost:8080/ShoppingFood-WAR/restaurante.xhtml">http://localhost:8080/ShoppingFood-WAR/restaurante.xhtml</a>	4	3	1	-	-
<a href="http://localhost:8080/ShoppingFood-WAR/login.xhtml">http://localhost:8080/ShoppingFood-WAR/login.xhtml</a>	4	3	1	-	-
<a href="http://localhost:8080/ShoppingFood-WAR/secciones/secure/pedido.xhtml">http://localhost:8080/ShoppingFood-WAR/secciones/secure/pedido.xhtml</a>	9	7	1	1	-

**Tabla 2. 8 Pruebas de Validación de páginas web**

Como se puede verificar en la Tabla 2.8 los 2 errores más frecuentes en la validación de las páginas web son:

- there is no attribute "role" y
- there is no attribute "aria-haspopup"

Que tienen que ver con el código generado por los componentes de Primefaces. Estos incidentes se deben a que los componentes de Primefaces usan atributos de WAI ARIA. [34]. Estos atributos son aceptados y reconocidos por los validadores cuando la página web está escrita en formato HTML5 (el más reciente).

En el caso del presente proyecto al utilizar Primefaces y JSF las páginas web son generadas en formato XHTML 1.0 que por el momento no contiene estas nuevas especificaciones, pero según la W3 en algún momento se implementarán luego de una revisión de las especificaciones. [34]



## **CAPÍTULO III: EVALUACIÓN DEL SISTEMA CON UN CASO DE ESTUDIO.**

### **3.1 RECOLECCIÓN DE INFORMACIÓN SOBRE EL CASO DE ESTUDIO**

La Empresa que ha permitido poner en práctica el presente proyecto es el Restaurante “Piguari”. Esta empresa se dedica a la venta de comida y además brinda el servicio a domicilio, por lo cual es una empresa en que se podrá probar la aplicación de servicio de comida a domicilio.

#### **3.1.1 APRESTAMIENTO EMPRESARIAL**

##### **3.1.1.1 Misión**

El Restaurante “Piguari” no tiene establecida su misión, pero de acuerdo con la información obtenida del establecimiento por parte de sus propietarios y su competencia se podría decir que la misión es:

Ofrecer a nuestros clientes productos de calidad a precios razonables y brindarles un buen servicio y atención oportuna por un personal comprometido con los valores empresariales, creando un vínculo comercial que sea de beneficio mutuo para la sociedad y nosotros. Generar empleo y mantener los valores tradicionales de nuestra cultura, así como también preservar nuestro entorno a través del cumplimiento de políticas ambientales.

##### **3.1.1.2 Visión**

La visión del Restaurante “Piguari” es:

Ser una empresa altamente productiva, innovadora, competitiva y dedicada a la satisfacción plena de nuestros clientes y lograr el crecimiento a través de la creación de sucursales en la ciudad de Quito.

Al igual que la misión, todos los miembros de la empresa deben conocer la visión ya que representa el direccionamiento de la empresa hacia el futuro.

### 3.1.1.3 Objetivos estratégicos

El Restaurante “Piguari” tiene por objetivos empresariales:

- Satisfacer al cliente en sus gustos y necesidades
- Ser eficaces y eficientes en las áreas productivas y administrativas
- Cumplir con las normas de higiene y seguridad alimentaria
- Tener excelente relación con los clientes, proveedores y empleados

### 3.1.1.4 Organigrama funcional

La figura 3.1 muestra el organigrama funcional del restaurante Piguari.

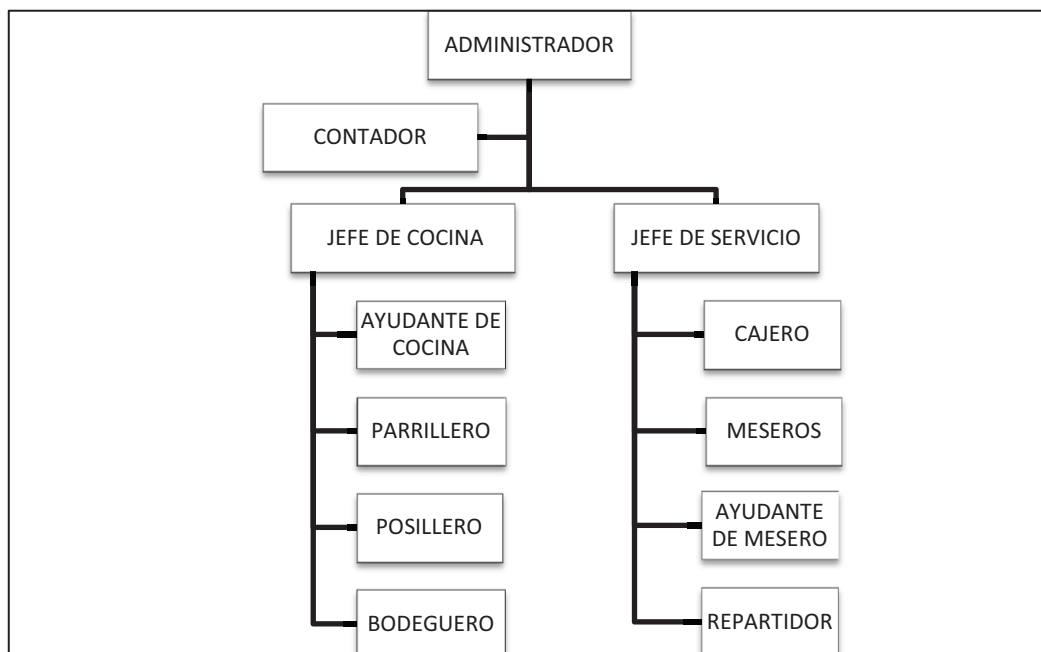


Figura 3. 1 Organigrama funcional Restaurante “Piguari”

## *DESCRIPCIÓN DE FUNCIONES*

### *ADMINISTRACIÓN*

#### ADMINISTRADOR

- Administra el restaurante
- Hace cumplir las normas y políticas establecidas

- Realiza las compras semanales de vegetales y carnes
- Resuelve problemas con los empleados y clientes
- Efectúa las promociones y publicidad de la empresa
- Realiza el cuadro de caja todos los días

## AREA DE COCINA

### JEFE DE COCINA

- Administra la cocina
- Preparar el almuerzo en días ordinarios
- Prepara las bandejas especiales
- Innova recetas
- Realiza el inventario de cocina

### AYUDANTE DE COCINA

- Ayuda en la preparación de los platillos
- Se encarga de hacer los preparativos para el siguiente día
- Mantiene la limpieza de la cocina

### POSILLERO

- Lava los platos y demás menaje de la cocina
- Mantiene la limpieza del área de producción

### PARRILLERO

- Encargado de asar carnes, chuleta y pollo y freír papas
- Tomar los pedidos que son para llevar
- Mantener la limpieza de su área

### BODEGUERO

- Embodega todos los alimentos perecibles y no perecibles
- Porcionar y limpiar todos los productos
- Almacena suministros del restaurante y de oficina

- Mantiene la limpieza de las áreas
- Realiza el inventario de producto existente

#### *AREA DE SERVICIO*

##### CAJERO

- Toma el pedido y cobra a los clientes
- Realiza la facturación detallada
- Se le delega la responsabilidad de la caja chica de la empresa

##### MESEROS

- Atención a los comensales
- Atención de las sugerencias y reclamos del cliente
- Verificar la limpieza del restaurante y baños y otros desperfectos

##### AYUDANTE DE MESERO

- Es el encargado de realizar la limpieza del restaurante y baños
- Retira la vajilla, aderezos, botellas vacías al final del servicio

##### REPARTIDOR

- Es el encargado de entregar los pedidos del servicio a domicilio
- Se encarga de cobrar el pedido o hacer firmar los vouchers del cliente

#### **3.1.1.5 Mapa de procesos**

La empresa no cuenta con documentación acerca de un mapa de procesos. Pero se ha identificado los procesos de atención al cliente, de distribución del producto y de marketing que serán descritos a continuación.

#### **3.1.2 PROCESO DE ATENCIÓN AL CLIENTE**

La principal estrategia de venta es personal, la cual se la realiza directamente en el punto de venta, contando con el personal de servicio y caja, quienes se

encargan de atender al cliente, mostrarle las especialidades, dar sugerencias de consumo, brindando asesoramiento al cliente.

### 3.1.3 PROCESO DE DISTRIBUCIÓN DEL PRODUCTO

**Servicio a domicilio:** El cliente realiza su pedido por teléfono, y el pedido se lo lleva a la comodidad de su casa. Dicho servicio tiene un costo adicional.

**Punto de venta:** El servicio cumple una triple actividad, se atiende al cliente, distribuye el producto y se cobra la factura.

### 3.1.4 PROCESO DE MARKETING

En el establecimiento no existe un departamento independiente de marketing en el que se realice las funciones y estrategias de comercialización y promoción del producto. Así como tampoco existe personal calificado para la realización de dichas funciones. Es en realidad el dueño del establecimiento quien implementa la publicidad y promoción de acuerdo a su conocimiento y mediante la contratación de este servicio a distintas empresas de publicidad que le brindan asesoría.

En la parte superior del local se muestra un rótulo con el logotipo de la empresa, números de teléfono, dirección y la oferta de sus productos principales.

Existen fotos y la descripción de los platos ubicados en el área de caja (Ver Figuras 3.2 y 3.3), del self-service, ventanas y parte frontal del restaurante facilitando al cliente el escoger un producto para servirse.



Figura 3. 2 Publicidad de promociones restaurante “Piguari”



Figura 3. 3 Publicidad de productos de mayor acogida restaurante "Piguari"

### 3.1.5 PRECIOS

El propietario ha establecido los precios de acuerdo a los costos tanto de la materia prima y mano de obra y de acuerdo a la competencia.

En la tabla 3.1 se especifican los productos, en sus respectivas categorías y con sus precios unitarios.

PLATOS ESPECIALES		MENU		BEBIDAS	
CHULETA CON MENESTRA	4,20	DESAYUNO NORMAL	1,90	AGUA SIN GAS	0,45
CARNE CON MENESTRA	4,20	DESAYUNO COMPLETO	2,90	AGUA CON GAS	0,50
APANADO	4,20	SANDUCHES	0,50	COLA VASO 12oz	0,60
CHURRASCO	4,20	CAFÉ	0,70	COLA VASO 16oz	0,70
BISTECK	4,20	HUMITAS	0,50	COLA LIGHT	1,00
CHAULAFAN	4,20	AGUA AROMATICA / TE	0,50	COLA FAMILIAR 3ltrs	2,20
CARNE CON POLLO	4,60	ALMUERZO	2,50	V220	1,10
CHULETA CON POLLO	4,60	SEGUNDOS	1,60	JUGOS	0,90
CHULETA CON CARNE	4,60	JUGO DE ALMUERZO	0,50	JARRAS DE JUGO	2,50
YAPINGACHO	4,80			BATIDOS	1,10
PARRILLADA	6,50	<b>COMBOS</b>		LIMONADA EN BOTELLA	0,60
		1/8 POLLO	2,70	CERVEZA PILSENER	1,20
<b>CUARTOS</b>		CARNE ASADA	2,70	CONO DE HELADO	0,50
1/4 COMPLETO	3,60	CHULETA	2,70		
1/4 CON PAPAS	3,40	CARNE APANADA	2,70	<b>PORCIONES</b>	
1/4 CON MENESTRA	3,60	SECO DE POLLO	2,70	ARROZ	1,15
1/4 SOLO	2,60	CORVINA	2,90	MENESTRA	0,90
1/8 SOLO	1,30	1/2 PARGO	2,90	ENSALADA	0,90
		ARROZ CON POLLO	2,70	CANGUIL	0,30
<b>MEDIOS</b>		YAPINGACHO	4,60	MADURO	0,60
1/2 COMPLETO	6,60			PATACONES	0,90
1/2 CON PAPAS	5,50	<b>CALDOS</b>		CARNE ASADA	2,20
1/2 SOLO	4,90	CALDO DE POLLO	2,30	CHULETA ASADA	2,20

		CONSOME NORMAL	0,60	CARNE APANADA	2,20
<b>POLLOS</b>		CONSOME EXTRA	1,00	SALCHICHA SOLA	0,25
POLLO COMPLETO	12,10	SOPA DEL DIA	1,00	HUEVO FRITO	0,25
POLLO CON PAPAS	10,20	CONSOME GRANDE	1,00	TORTILLA DE PAPA	0,50
POLLO SOLO	9,20				
		<b>PAPAS</b>			
<b>MARISCOS</b>		PAPI POLLO	2,20		
PARGO FRITO	4,80	PAPI CARNE	2,20		
CORVINA FRITA	4,80	SALCHIPAPA	1,35		
CORVINA APANADA	4,80	PAPA FRITA	1,15		
CORVINA CON MENESTRA	4,80	SALCHIPAPA + COLA	1,65		
ARROZ CON CAMARONES	4,80				
CAMARONES APANADOS	4,80				
CEVICHE DE CAMARON	4,50				
ENCEBOLLADO	2,20				

**Tabla 3. 1 Menú Restaurante "Piguari"**

## 3.2 PROPUESTA DE IMPLANTACIÓN DE LA APLICACIÓN.

### 3.2.1 INSTALACIÓN DEL SISTEMA

Para la instalación del sistema en un servidor son necesarios ciertos requerimientos de HW y SW que a continuación se describirán.

#### 3.2.1.1 Hardware Requerido

Para la instalación de la aplicación web es necesario un equipo con los requerimientos de hardware listados en la tabla 3.2.

Dispositivo	Descripción
Procesador	Core i3 2.5 GHz o superior
Memoria RAM	4 GB o superior
Disco Duro	500GB o superior
Tarjeta de Red	10/100 Mbps

Tabla 3. 2 Requerimientos de HW

#### 3.2.1.2 Software Requerido

Para la instalación de la aplicación es necesario un equipo con los requerimientos de software que se muestran en la tabla 3.3.

Software	Descripción - Versión
Sistema Operativo	Centos 6.2 de 64 bits
Java Runtime Enviroment	1.7
JBoss	7.1.1
PostgreSQL	9.2

Tabla 3. 3 Requerimientos de SW

Para cumplir con los requerimientos de software de la aplicación se ha preparado guías de instalación para Java Runtime Enviroment 1.7, JBoss 7.1.1 y Postgresql 9.2. Estas guías de instalación de los pre-requisitos de software se describen en el documento de instalación que se encuentra en el CD Anexo.



En el CD anexo se encuentra también el documento de instalación de la aplicación shopping food y el documento del Manual de usuario.

### 3.2.2 CARGA DE DATOS

Para realizar la carga de datos se va a usar el módulo administrativo de la aplicación shopping food.

Para la carga de datos se utilizará el menú de productos que fue proporcionado por el restaurante.

Se partirá de la aplicación shopping food sin ningún dato y se realizará el ingreso del usuario que administrará el restaurante. Luego el administrador del restaurante se encargará de los ingresos de las categorías y productos, así como de sus empleados cajeros y repartidores.

#### Creación de usuario administrador del sitio.

Se debe ingresar en el módulo de administración de shopping food con el rol Administrador del sitio como se puede ver en la Figura 3.4. Aquí el administrador se encargará de registrar un proveedor para la aplicación.

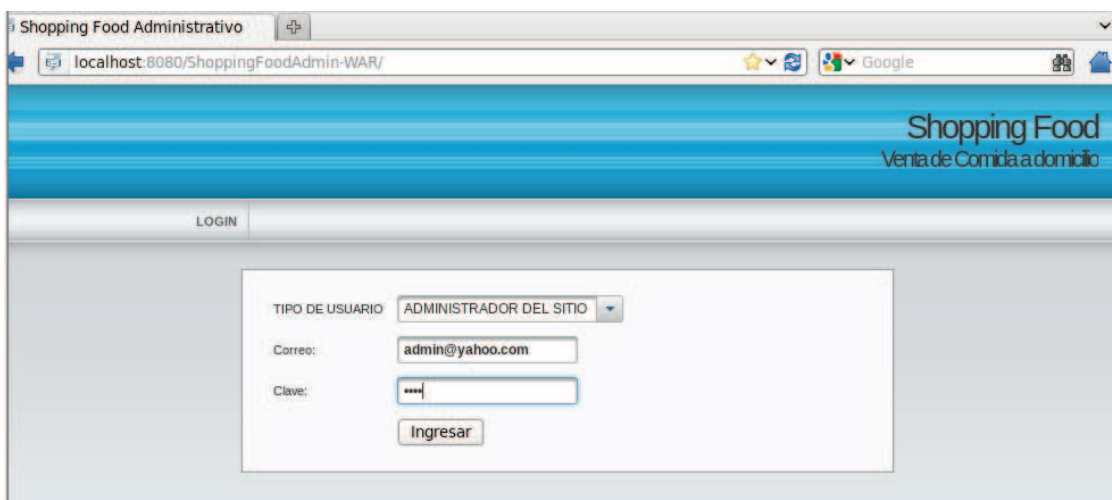


Figura 3. 4 Login en la aplicación como administrador del sitio

Una vez que logeado aparecerá una pantalla indicando que el certificado de seguridad no es válido como se puede ver en la Figura 3.5. Esto se presenta

debido a que el certificado es autofirmado y el explorador no lo reconoce como un certificado de confianza.

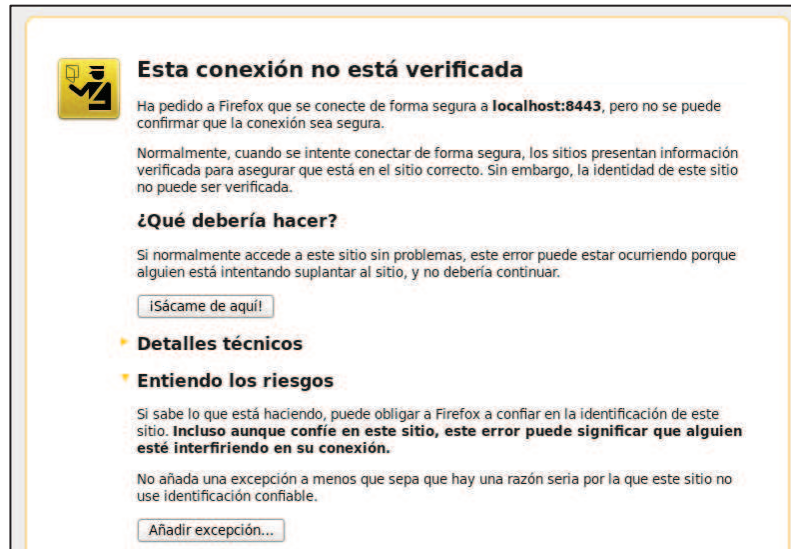


Figura 3. 5 Advertencia del browser del certificado para HTTPS

Agregar el certificado al browser y la advertencia no volverá a aparecer, esto se lo puede apreciar en la Figura 3.6.



Figura 3. 6 Agregación del certificado de la aplicación al browser.

Con ello se ingresará a la pantalla principal para el administrador del sitio. Para registrar el restaurante “Piguari” se deberá ingresar el nombre del Proveedor y la fecha límite hasta donde estará vigente en la aplicación tal como se puede ver en la Figura 3.7.

Shopping Food  
Venta de Comida a domicilio - Administrador Site  
Mi Cuenta

INDEX PROVEEDORES

Menu Administrativo

- Proveedores
- Cuentas
- Salir

NombreProv: PIGUARI

CorreoProv: restaurantepiguari@hotmail

Direccion: Avenida Real Audiencia

Telefono: 3413567

Fecha de Vencimiento cuenta: 4/10/13

Imagen: + Seleccione Foto (max.500Kb)

Guardar

Cancelar

ADMIN

Figura 3. 7 Creación del proveedor en el administrador del sitio

También el administrador del sitio deberá crear una cuenta con rol administrador proveedor para que se administre el restaurante “Piguari”. Se puede ver en la Figura 3.8 el formulario de creación de un usuario administrador proveedor.

Shopping Food  
Venta de Comida a domicilio - Administrador Site  
Mi Cuenta

INDEX PROVEEDORES

Menu Administrativo

- Proveedores
- Cuentas
- Salir

EmailCli: admin@piguari.com

NombresCli: Andrea

ApellidosCli: Santacruz

DireccionCli: EL bosque

CICI: 1209223312

IdProv: PIGUARI


ClaveCli: \*\*\*\*

Guardar

ADMIN

Figura 3. 8 Creación del usuario administrador proveedor del restaurante “Piguari”.

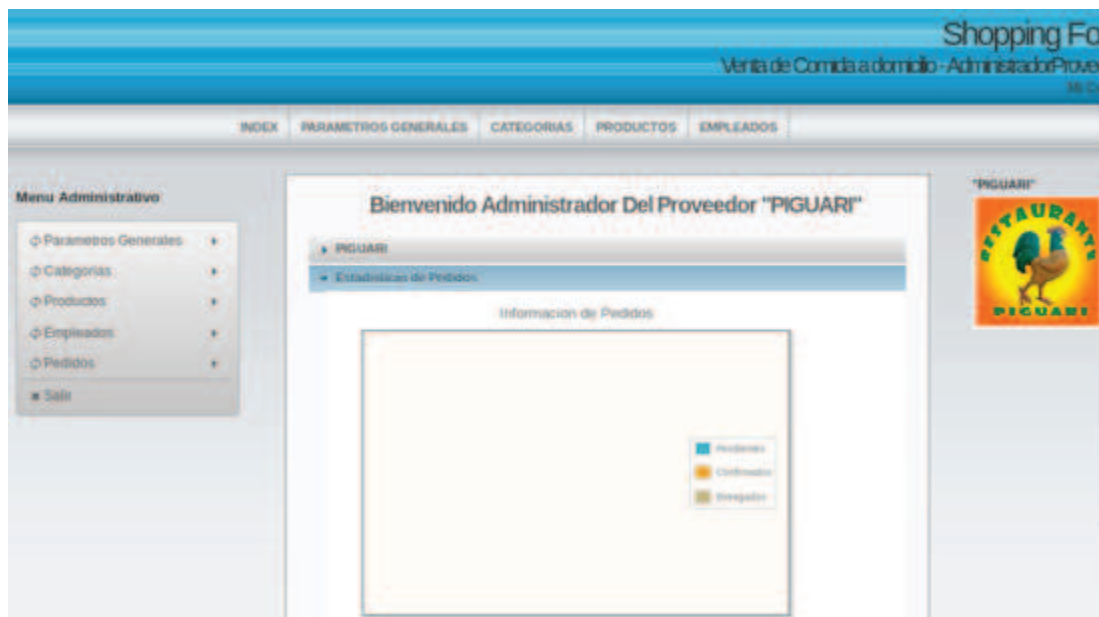
Se finaliza la sesión del usuario administrador del sitio y con la cuenta de administrador proveedor creada en el paso anterior, se ingresa al sistema como se puede ver en la Figura 3.9. Luego de esto se procede con el ingreso de los datos del restaurante.



The screenshot shows the login interface for 'Shopping Food'. At the top right, the logo 'Shopping Food' is displayed with the tagline 'Venta de Comida a domicilio'. Below the logo is a 'LOGIN' tab. The main content area contains a login form with the following fields: 'TIPO DE USUARIO' (a dropdown menu set to 'ADMIN PROVEEDOR'), 'Correo:' (a text box containing 'admin@piguari.com'), and 'Clave:' (a password field with masked characters). An 'Ingresar' button is located below the password field. At the bottom of the page, there is a footer: '© Shopping Food | Design by Esteban Salazar and Fredy Sanchez | Web Hosting'.

**Figura 3. 9 Login como usuario administrador proveedor**

Si el usuario administrador proveedor se ha logeado correctamente se podrá observar la pantalla principal del Restaurante "Piguari" como se aprecia en la Figura 3.10.



**Figura 3. 10 Pantalla principal del administrador del restaurante "Piguari".**

Configurar primero los parámetros generales como el valor mínimo para realizar un pedido, número máximo de pedidos por hora y los datos para el pago usando PayPal. La Figura 3.11 muestra el formulario para ingreso de parámetros generales de un proveedor.

The screenshot shows the 'Parámetros Generales' form for the restaurant 'Piguari'. The form is part of a web application titled 'Shopping For Venta de Comida a domicilio - Administración Proveedores'. The navigation menu includes 'INDICE', 'PARAMETROS GENERALES', 'CATEGORIAS', 'PRODUCTOS', and 'EMPLEADOS'. The 'Menu Administrativo' on the left lists 'Parámetros Generales', 'Categorías', 'Productos', 'Empleados', 'Pedidos', and 'Salir'. The form fields are as follows:

Parámetros Generales	
IdProv:	<input type="text"/>
NombreProv:	<input type="text" value="PIGUARI"/>
CorreoProv:	<input type="text" value="restaurantepiguar@hotmail.com"/>
Imagen:	<input type="button" value="Selecione Foto (max.500Kb)"/>
ValorMinPedido:	<input type="text" value="15.0"/>
ValorMinPedidosHora:	<input type="text" value="10"/>
Configuración PayPal	
Valor Business:	<input type="text" value="13486059@hotmail.com"/>
Id Certificado PayPal:	<input type="text" value="FF8TUJVFTRVG"/>
<input type="button" value="Guardar"/>	
<input type="button" value="Cancelar"/>	

On the right side, there is a logo for 'PIGUARI RESTAURANTE' featuring a rooster.

Figura 3. 11 Actualización de los parámetros generales del restaurante “Piguari”.

Luego se debe crear las categorías de los productos en base al menú de opciones. La Figura 3.12 muestra el formulario de creación de una categoría.

The screenshot shows the 'Categorías' form for the restaurant 'Piguari'. The navigation menu includes 'INDICE', 'PARAMETROS GENERALES', 'CATEGORIAS', 'PRODUCTOS', and 'EMPLEADOS'. The 'Menu Administrativo' on the left lists 'Parámetros Generales', 'Categorías', 'Productos', 'Empleados', 'Pedidos', and 'Salir'. The form displays a success message: 'Categoría was successfully created.' Below the message is a text input field for 'NombreCat:'. There are buttons for 'Guardar' and 'Mostrar todos los objetos Categoría Items'. The 'PIGUARI RESTAURANTE' logo is visible on the right.

Figura 3. 12 Creación de las categorías del restaurante “Piguari”.

Ingresar todas las categorías del proveedor. Con esto luego se podrá listar todas las categorías tal como se puede ver en la Figura 3.13.

IdCategoria	NombreCat	
1	Platos Especiales	Ver Editar Eliminar
2	Desayunos	Ver Editar Eliminar
3	Bebidas	Ver Editar Eliminar
4	Combos	Ver Editar Eliminar
5	Cuartos	Ver Editar Eliminar
6	Porciones	Ver Editar Eliminar

Crear Nuevo Objeto Categoria

Inicio




Figura 3. 13 Listado de categorías del proveedor “Piguari”.

Luego ingresar los productos que pertenecen a cada categoría definiendo sus características como precio y cantidad disponible en el inventario. La Figura 3.14 muestra un ejemplo de registro de un producto.

Shopping Fo  
Venta de Comida a domicilio - Administrador Proveedores

INDEX PARAMETROS GENERALES CATEGORIAS PRODUCTOS EMPLEADOS

Menu Administrativo

- Parametros Generales
- Categorias
- Productos
- Empleados
- Pedidos
- Sair

NombrePro:

PrecioPro:

DescripcionPro:

CantidadPro:

CantidadMinPro:

ImagenPro:

IdCategoria:

Guardar

Mostrar todos los objetos Producto Items

Inicio




Figura 3. 14 Formulario de creación de Productos

Se debe definir una cuenta para el empleado cajero del restaurante. Para ello en el menú principal de la aplicación acceder a la sección empleados y luego escoger la opción Nuevo Repartidor tal como se ve en la Figura 3.15.

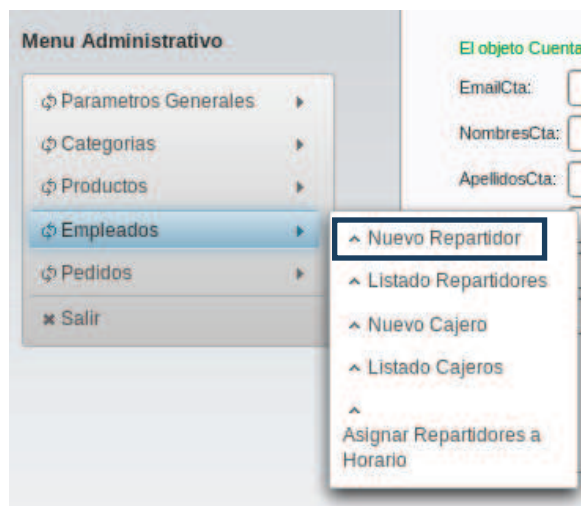


Figura 3. 15 Menú de creación de empleados del restaurante “Piguari”.

La aplicación mostrará un formulario para el ingreso de un repartidor. Ingresar los datos del repartidor y luego guardar el registro tal como se aprecia en la Figura 3.16.

Figura 3. 16 Formulario de creación de empleado Cajero.

También en el menú principal, en la sección empleados escoger la opción Nuevo Repartidor y con ello se podrá crear el repartidor restaurante. La Figura 3.17 muestra el formulario para creación de un repartidor.

Figura 3. 17 Formulario de creación de empleado Repartidor.

Para poder asignar un repartidor a un pedido, este debe tener un horario de trabajo. Se debe asignar un horario al repartidor y para ello ir al menú principal luego Asignar Repartidores a Horario de trabajo. Ahí se mostrará una lista de todos los repartidores que no tienen asignado un horario de trabajo. Seleccionar el enlace (+) y con ello aparecerá el formulario que se ve en la Figura 3.18 y que permite el registro de un horario para un empleado.

Figura 3. 18 Formulario de asignación de horario a empleado.



Con esto en el módulo de atención al cliente se mostrará el proveedor creado y dentro de este sus categorías y productos tal como se puede observar en las Figuras 3.19 y 3.20.



Figura 3. 19 Pantalla principal del módulo de atención al cliente de shopping food.

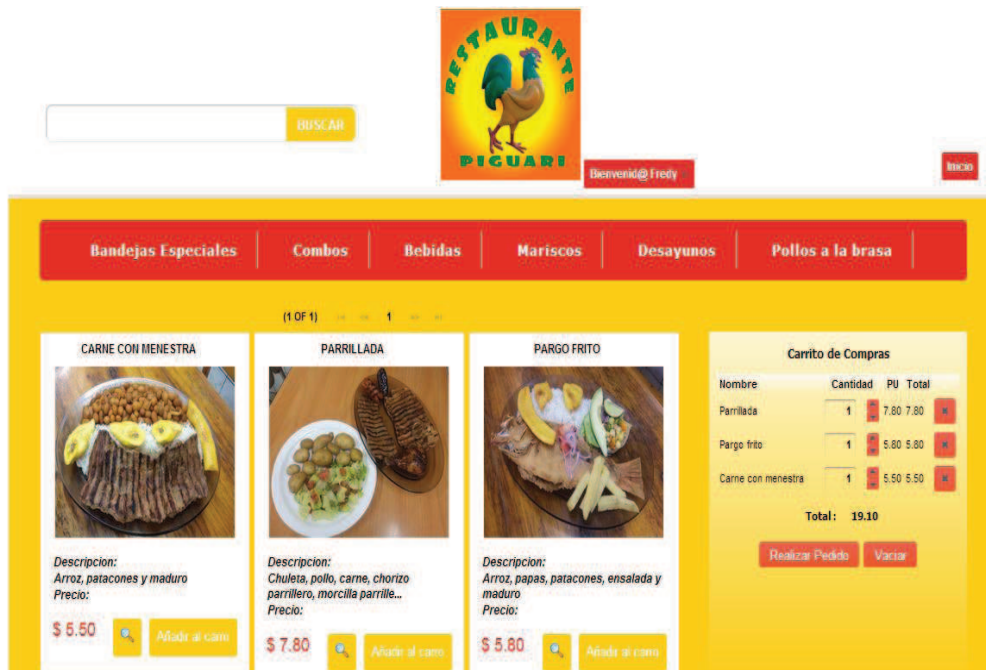


Figura 3. 20 Pantalla principal de atención al cliente para el restaurante Figuari.

### 3.3 PRUEBAS DE CARGA Y STRESS

Las pruebas de carga y stress forman parte de un conjunto de pruebas cuyo objetivo es evaluar el rendimiento de una aplicación tomando en cuenta varios aspectos como la velocidad de procesamiento de una petición, cantidad de memoria utilizada y uso del procesador.

Durante la ejecución de estas pruebas para el sistema, se utilizarán herramientas de monitoreo que permiten determinar los valores entre los cuales la aplicación tiene un rendimiento aceptable, así como permiten definir los valores máximos sobre los cuales la aplicación comienza a bajar su rendimiento.

Otro aspecto importante en este tipo de pruebas es que se debe utilizar valores que permitan simular el comportamiento de la aplicación en producción.

#### 3.3.1 PRUEBAS DE CARGA

El propósito principal de las pruebas de carga es simular el acceso concurrente a una aplicación realizando un número de peticiones determinado.

En una prueba de carga se puede realizar varias acciones como simular la apertura de múltiples conexiones a un servidor y también enviar varias solicitudes HTTP, con el fin de determinar cuellos de botella que puedan existir en la aplicación. [35]

Con los resultados de esta prueba se puede evaluar los tiempos de respuesta de las transacciones importantes que procesará la aplicación.

---

[35] <http://www.buzzle.com/articles/software-testing-stress-testing.html>

### 3.3.2 PRUEBAS DE STRESS

Las pruebas de Stress sobre un sistema software sirven para determinar la efectividad del rendimiento del sistema cuando se presentan condiciones extremas de carga de procesamiento superior a las normales.

El objetivo principal de esta prueba es determinar el grado de robustez del rendimiento de un sistema cuando se ha sobrepasado los valores normales de operación. [35]

Algunas de las acciones que se realizan en las pruebas de Stress incluyen bombardear un sitio web con un gran número de peticiones, realizar varios intentos de login, etc.

### 3.3.3 PROCESO DE PRUEBAS

Para poder probar la aplicación se usará la herramienta JMeter.

JMeter es una aplicación basada en Java, 100% open source cuyo objetivo es ejecutar pruebas funcionales y medir el rendimiento. [36]

Para la elaboración de estas pruebas se ha utilizado una funcionalidad de JMeter denominada proxy la cual permite grabar un flujo de navegación web a través de la aplicación para luego poder repetir este flujo y simular la interacción de un usuario.

#### 3.3.3.1 Escenarios a probar

Para realizar las pruebas de carga y Stress en la aplicación se han definido 3 escenarios que serán los más comunes en ser usados en la aplicación.

Estos escenarios son:

---

[36]<http://jmeter.apache.org/>

1. Realizar pedido de carro de compras como usuario registrado
2. Consulta de pedidos pendientes para usuario registrado
3. Realizar carro de compras como usuario visitante

Todo el proceso que involucra replicar los escenarios para realizar las pruebas para la aplicación, se encuentra disponible en documento completo del CD Anexo. A continuación se muestra las actividades que se realizaron para las pruebas del escenario 3.

### Escenario 3. Realizar carro de compras como usuario visitante.

En este escenario un usuario no registrado visitará la aplicación y realizará un carrito de compras con los productos que él desee.

Primero se debe definir un proxy con la herramienta JMeter. Luego iniciar el proxy para luego seguir con la navegación para grabar el flujo. Las tareas del flujo se las detalla a continuación.

Escoger un proveedor tal como se muestra en la Figura 3.21.

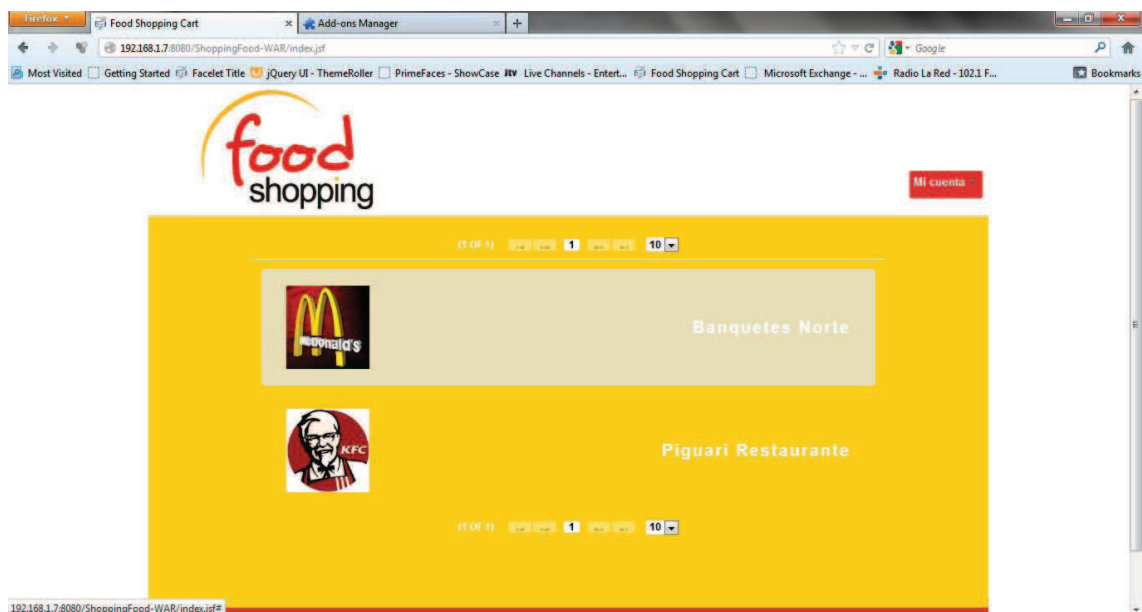


Figura 3. 21 Primer Paso para grabación en el Proxy escenario 3

Añadir productos al carro de compras, como se puede observar en la Figura 3.22.

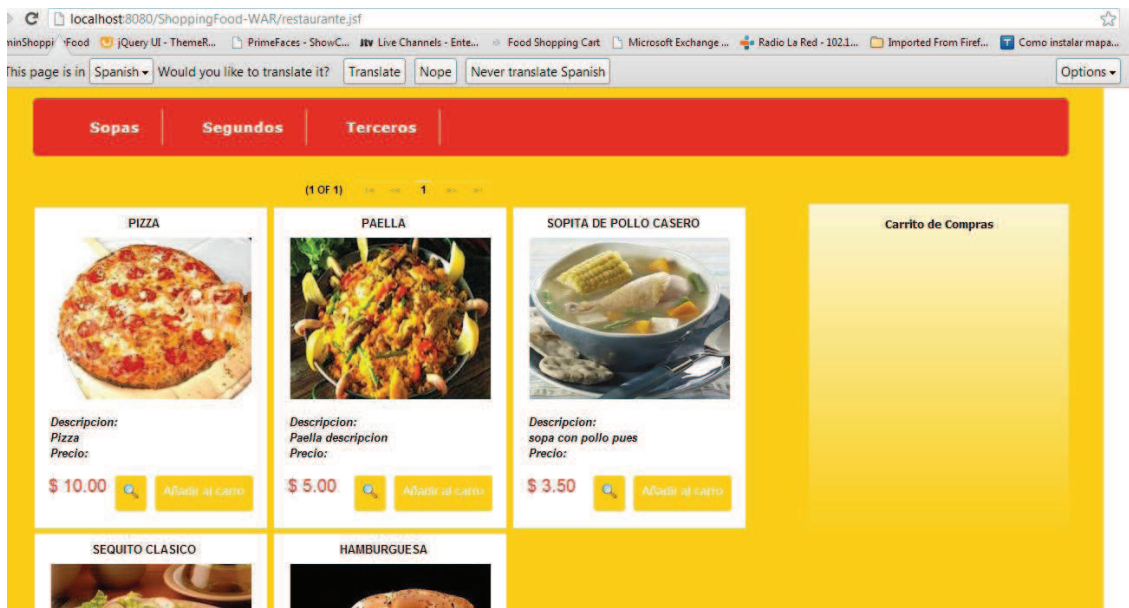


Figura 3. 22 Segundo Paso para grabación en el Proxy

Luego detener el proxy de JMeter y verificar las peticiones HTTP grabadas para este escenario. Ver Figura 3.23.

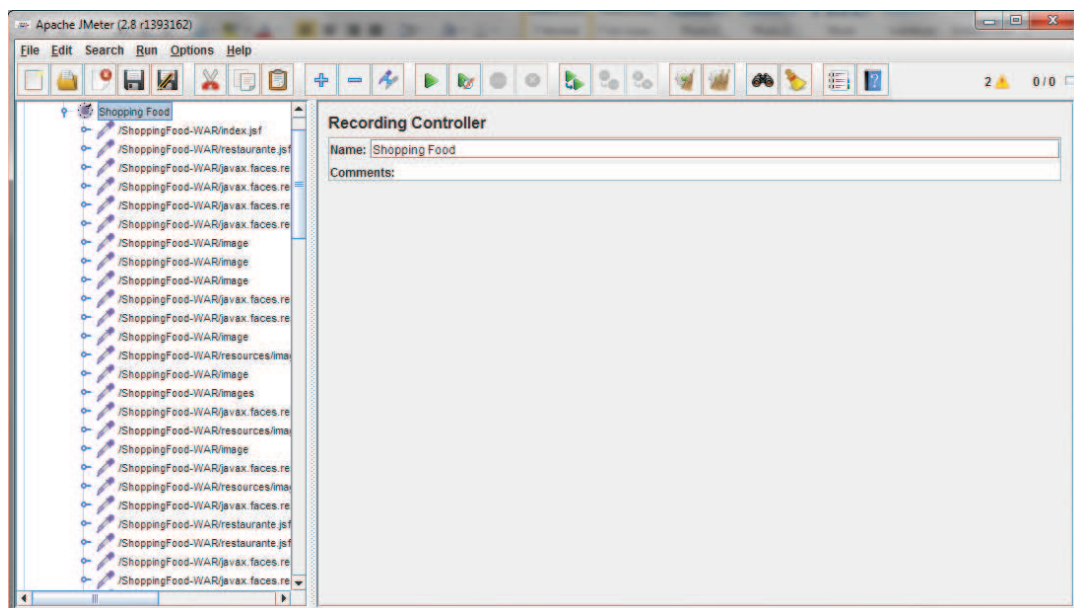


Figura 3. 23 Resultados de grabación del escenario 3.

### **3.3.4 EJECUCIÓN DE LAS PRUEBAS DE CARGA**

Una vez que se han grabado los flujos de ejecución que se realizaron en los escenarios a probar, es necesario definir el número de usuarios y el retardo entre pruebas de los usuarios para simular el flujo.

En las pruebas se va a simular 25 y 50 usuarios y el retardo entre cada petición será de 5 segundos. Se realizará la prueba 5 veces para tener más valores con los cuales se pueda sacar el promedio. Con esto el sistema recibirá 125 peticiones y 250 peticiones para 25 y 50 usuarios respectivamente.

Para la ejecución de la prueba se recomienda que la herramienta JMeter se ejecute en un equipo diferente al servidor que se está probando. Esto se lo hace para tratar de obtener datos solo de la aplicación ya que pueden afectar en los valores del servidor a probar.

### **3.3.5 EJECUCIÓN DE LAS PRUEBAS DE STRESS**

Para las pruebas de Stress se usarán los mismos escenarios de las pruebas de carga pero se aumentarán los valores de los usuarios concurrentes hasta que el rendimiento de los recursos de memoria y CPU se degrade. Con esto se verá cuál será la carga máxima que soportará la aplicación.

### 3.4 ANÁLISIS DE RESULTADOS

Para analizar los resultados se ha tomado algunos objetos y contadores para las respectivas pruebas, mismos se muestran en la tabla 3.4.

Objeto	Contadores	Valores Umbrales
<b>Servidor</b>	Tiempo de respuesta	1s [37]
	Carga procesada por segundo (Páginas/segundo)	Mayor es mejor
<b>Memoria</b>	Bytes disponibles.	Mínimo 4MB
<b>Procesador</b>	% Uso de procesador.	85%

Tabla 3. 4 Objetos, contadores y valores umbrales del servidor [38]

#### 3.4.1 ANÁLISIS DE LAS PRUEBAS DE CARGA

**Escenario 1. Realizar pedido de carro de compras como usuario registrado.**

Número de usuarios: 25

---

[38] Naranjo, Jaime. Información obtenida del material de la asignatura Auditoría y Evaluación de Sistemas Computacionales y analizado por los autores.

[37] [http://www.linti.unlp.edu.ar/uploads/docs/usando\\_jmeter\\_para\\_pruebas\\_de\\_rendimiento.pdf](http://www.linti.unlp.edu.ar/uploads/docs/usando_jmeter_para_pruebas_de_rendimiento.pdf)

## Tiempo de respuesta

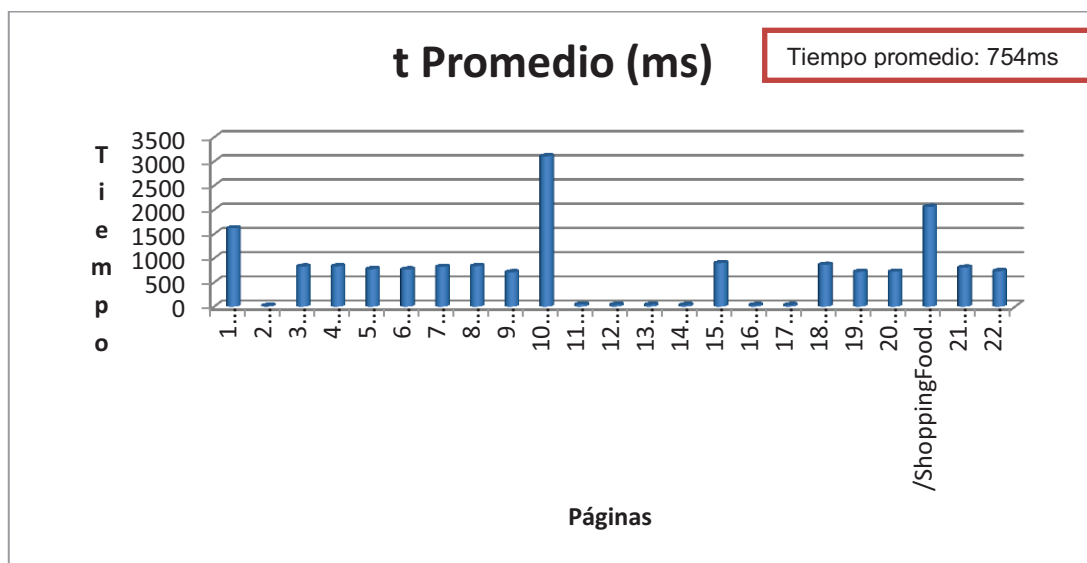


Figura 3. 24 Tiempo de Respuesta escenario Realizar Pedido para 25 Usuarios

Como se puede observar en la Figura 3.24 el tiempo de respuesta de algunas páginas supera el valor óptimo (1000ms). Pero el valor promedio general es de 754ms que se encuentra debajo del valor óptimo, por lo tanto el tiempo de respuesta del servidor web está en un nivel aceptable.

## Carga procesada por segundo

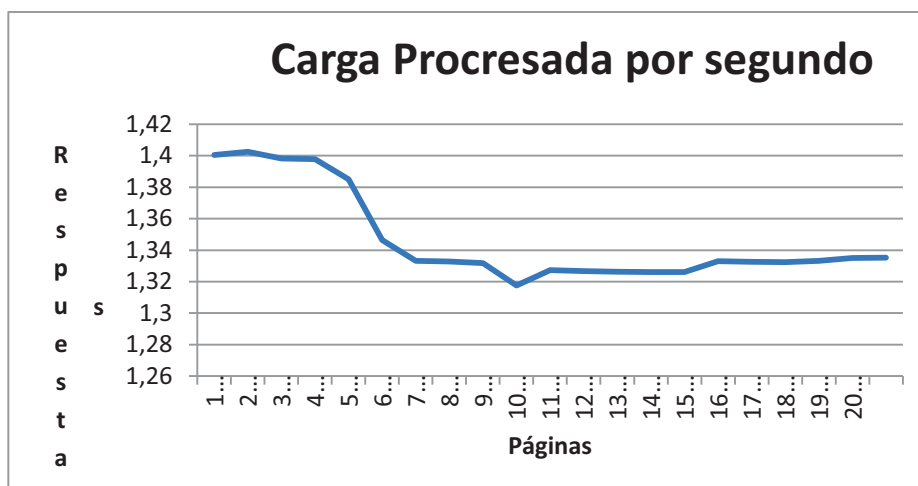
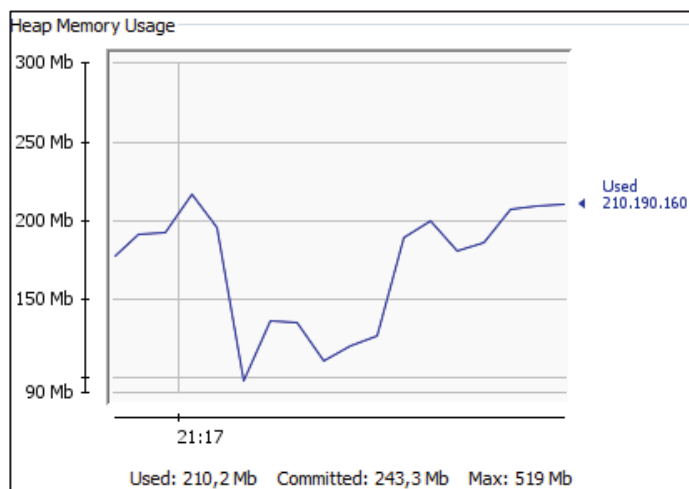


Figura 3. 25 Carga procesada por segundo escenario Realizar Pedido para 25 Usuarios

La carga procesada por segundo como se ve en la Figura 3.25 por lo menos es de 1,32 páginas por segundo, que está dentro de un valor aceptable.



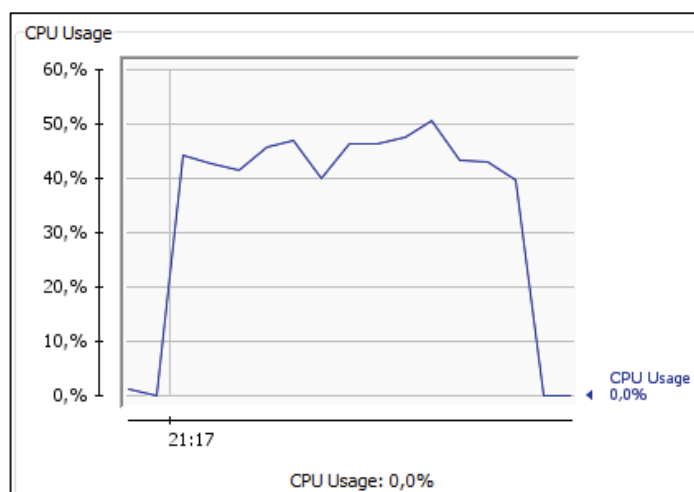
## Uso de Memoria



**Figura 3. 26** Uso de memoria escenario Realizar Pedido para 25 Usuarios

El uso de la RAM como se puede observar en la Figura 3.26 oscila entre 95MB a 220MB, por lo tanto los Bytes disponibles son mas de 300MB es durante estas pruebas de 25 usuarios. Este indicador de Bytes disponibles es mayor a los 4MB del umbral.

## Uso de CPU



**Figura 3. 27** Uso del CPU escenario Realizar Pedido para 25 Usuarios

En la Figura 3.27 los datos muestran que el procesador está sobredimensionado para 25 usuarios en el servidor de aplicaciones web, al estar muy por debajo de los valores umbrales, 85% para uso del procesador.

Número de usuarios: 50

Tiempo de respuesta

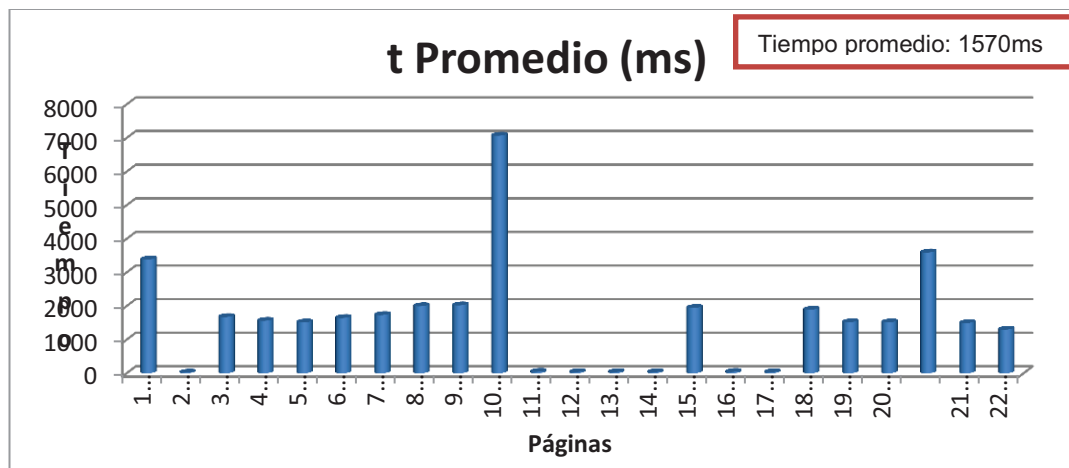


Figura 3. 28 Tiempo de Respuesta escenario Realizar Pedido para 50 Usuarios

Como se puede observar en la Figura 3.28 el tiempo de respuesta de la mayoría de las páginas supera el valor óptimo (1000ms). Además el valor promedio general es de 1670ms que se encuentra por encima del valor recomendable, por lo tanto el tiempo de respuesta del servidor web empieza a tener una deficiencia cuando la carga de usuarios es de 50.

Carga procesada por segundo

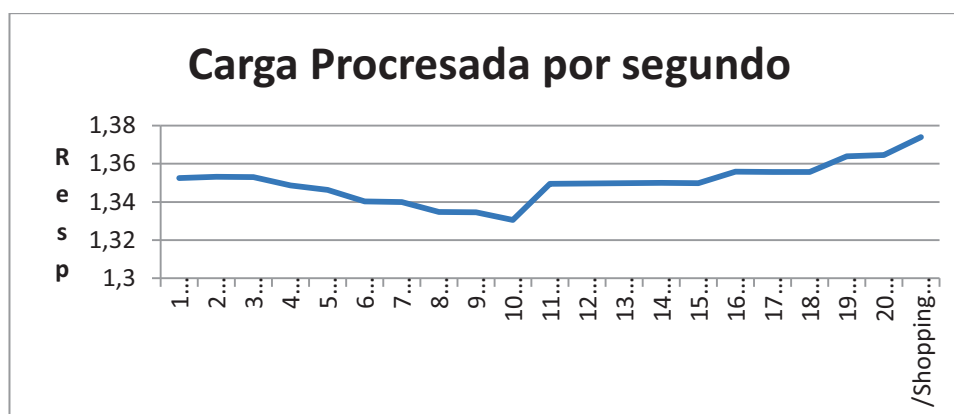
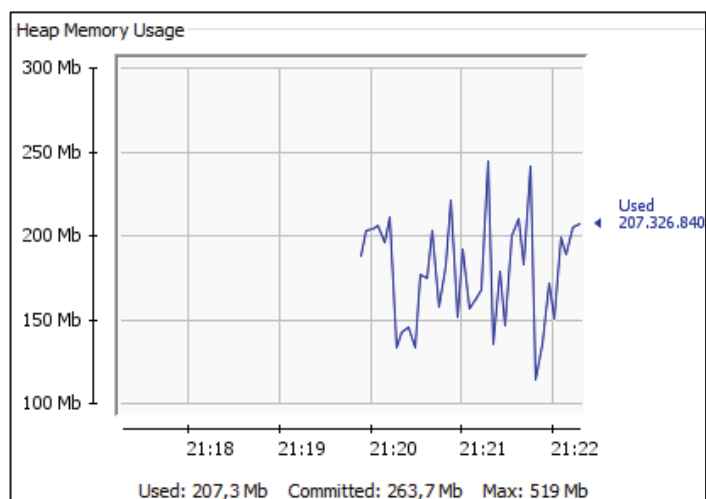


Figura 3. 29 Carga procesada por segundo escenario Realizar Pedido para 50 Usuarios

Como se puede apreciar en la Figura 3.29 la carga procesada por segundo se encuentra sobre 1,33 página/segundo, que representa que se encuentra en un rango tolerable.

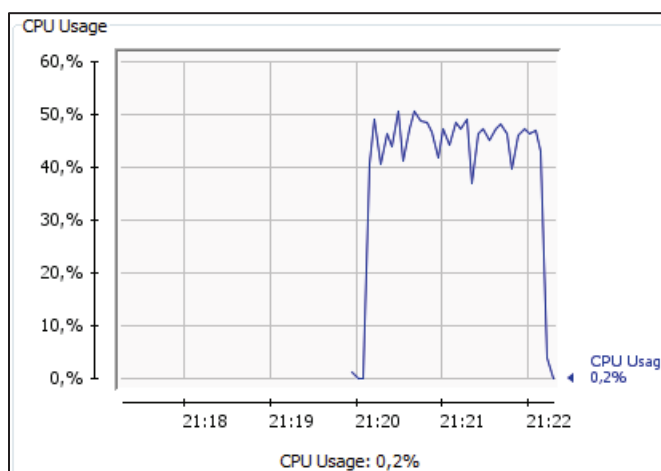
### Uso de Memoria



**Figura 3. 30 Uso de memoria escenario Realizar Pedido para 50 Usuarios**

El uso de la RAM como se puede observar en la Figura 3.30 oscila entre 105MB a 245MB, por lo tanto los Bytes Disponibles son 256MB durante estas pruebas. Este indicador de Bytes Disponibles es mayor a los 4MB del umbral.

### Uso de CPU



**Figura 3. 31 Uso del CPU escenario Realizar Pedido para 50 Usuarios**

En la Figura 3.31 los datos muestran que el procesador está sobredimensionado para 50 usuarios en el servidor de aplicaciones web, al estar muy por debajo de los valores umbrales, 85% para uso del procesador.

**Escenario 2. Consulta de pedidos pendientes para usuario registrado.**

Número de usuarios 25

Tiempo de respuesta

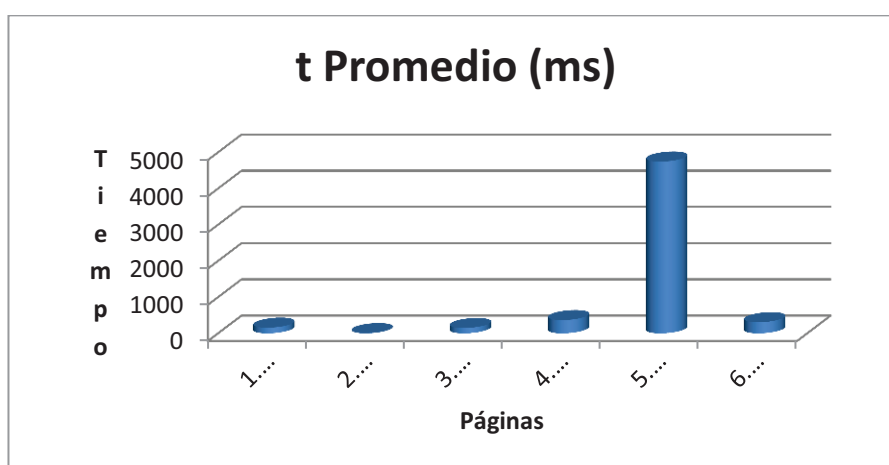


Figura 3. 32 Tiempo de respuesta escenario consultar pedidos pendientes 25 usuarios

En la Figura 3.32 se puede observar que existe una petición que toma algo más de 4 segundos en responder. Esta página es la que se encarga de mostrar los pedidos de un cliente. Esto hace que el tiempo que demora la consulta de los pedidos de un usuario sea de 1 segundo.

## Carga procesada por segundo

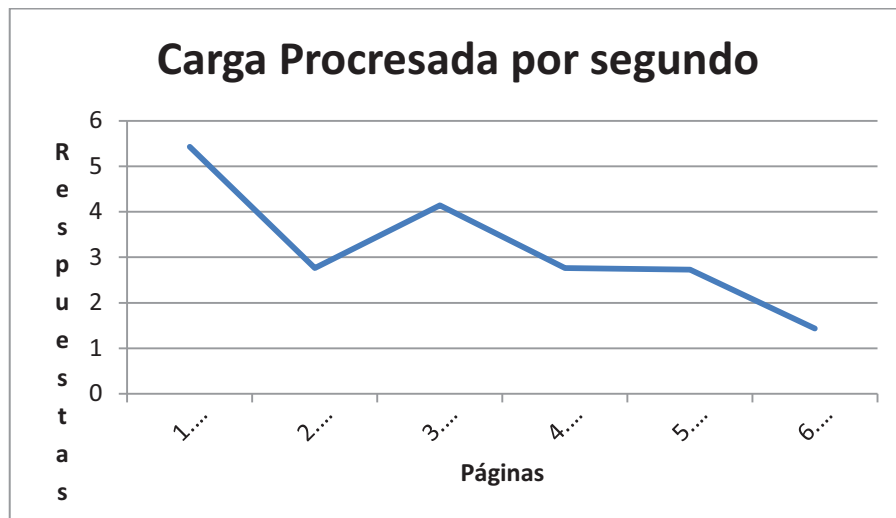


Figura 3. 33 Carga procesada escenario consultar pedidos pendientes 25 usuarios

En la Figura 3.33 se puede observar que la página que fue más solicitada fue la de la primera petición y esto se debe a que luego de la consulta se finaliza la sesión y el flujo vuelve a la página principal. Con eso la carga procesada en la prueba para 25 usuarios en promedio de todo el flujo es de 18 repeticiones por segundo.

## Uso de Memoria

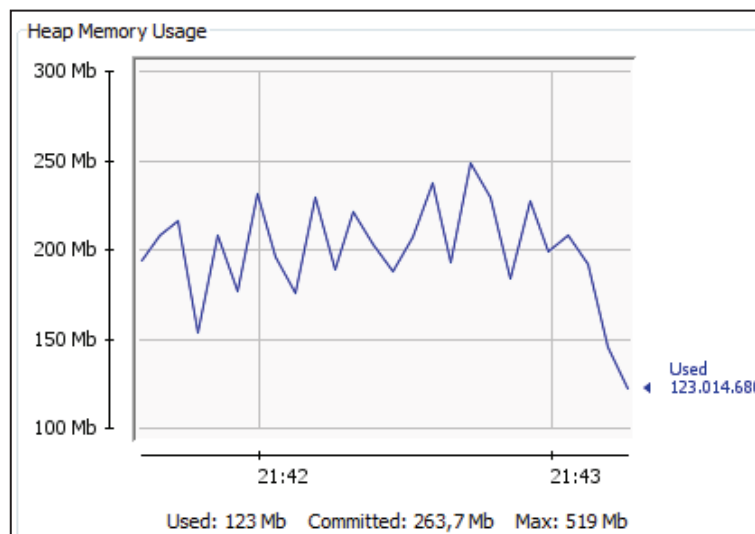


Figura 3. 34 Uso de memoria escenario consultar pedidos pendientes 25 usuarios

En la Figura 3.34 se puede observar que el proceso de prueba duró algo mas de 1 segundo. Durante este tiempo el consumo de memoria se mantuvo en valores debajo de la mitad de la memoria reservada para el servidor de aplicaciones.

### Uso de CPU

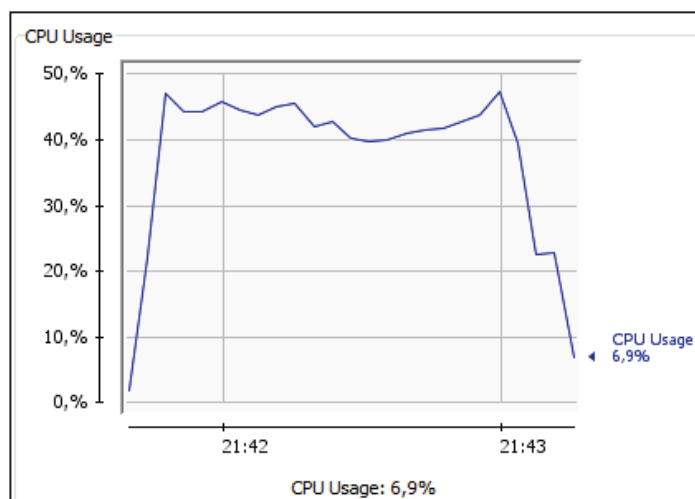


Figura 3. 35 Uso de CPU escenario consultar pedidos pendientes 25 usuarios

En la Figura 3.35 se puede observar que para las pruebas del escenario, el consumo de memoria subió casi al 50%, valor que se considera aceptable para el uso de CPU de un servidor de aplicaciones.

### Número de usuarios 50

### Tiempo de respuesta

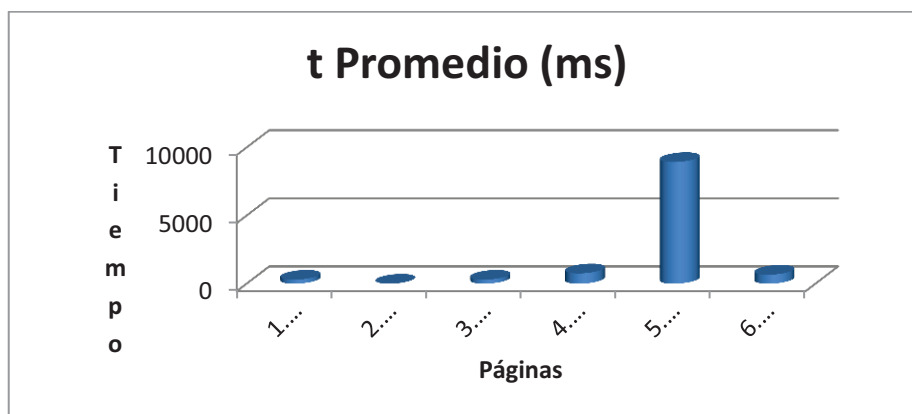


Figura 3. 36 Tiempo de respuesta escenario consultar pedidos pendientes 50 usuarios

El tiempo de respuesta como se puede ver en la figura 3.36 muestra información similar a la que se podía observar con la prueba del escenario con 25 usuarios. La diferencia radica en que el tiempo promedio de la quinta petición se llega a duplicar y con esto el tiempo total para cumplir el flujo también se incrementa.

Carga procesada por segundo

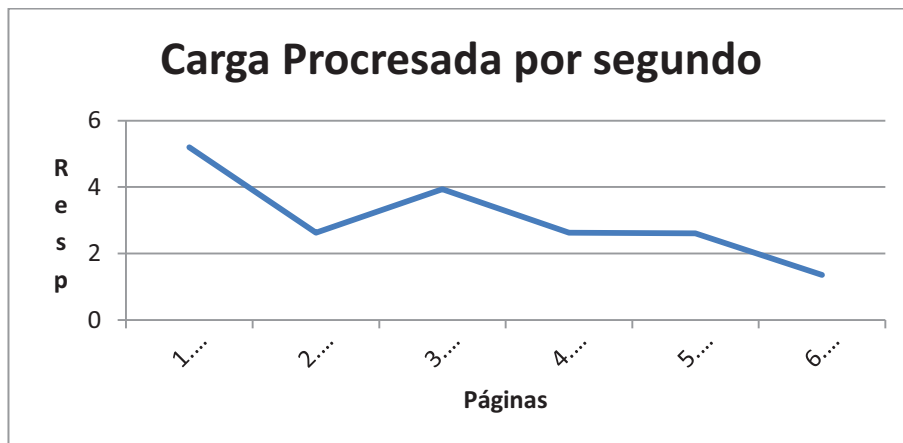


Figura 3. 37 Carga procesada escenario consultar pedidos pendientes 50 usuarios

En la figura 3.37 se puede apreciar que la carga procesada por segundo de las peticiones de este flujo no cambia demasiado para 25 y 50 usuarios concurrentes. Es decir que la primera petición del flujo fue la de mayor rendimiento en el tiempo.

Uso de Memoria

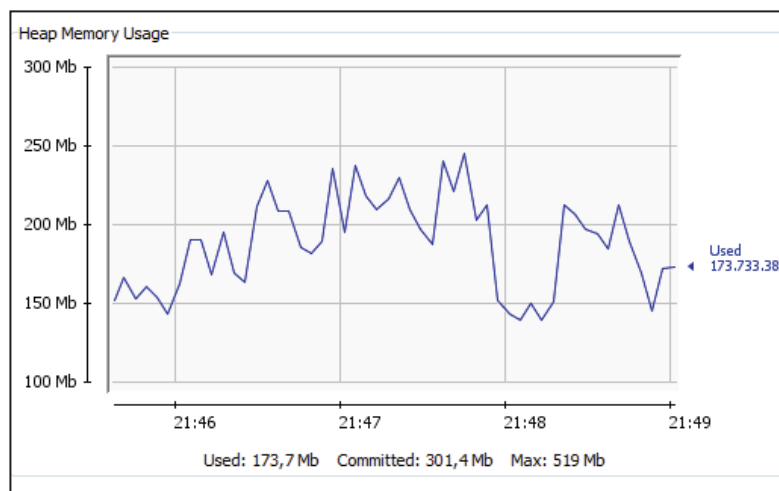


Figura 3. 38 Uso de memoria escenario consultar pedidos pendientes 50 usuarios

En la Figura 3.38 se puede notar que tampoco se sobrepasa el consumo del 50% de memoria asignado para el servidor de aplicaciones. El proceso de prueba para el escenario tomó 2.7 segundos que es casi el doble que tomo para 25 usuarios.

### Uso de CPU



**Figura 3. 39 Uso de CPU escenario consultar pedidos pendientes 50 usuarios**

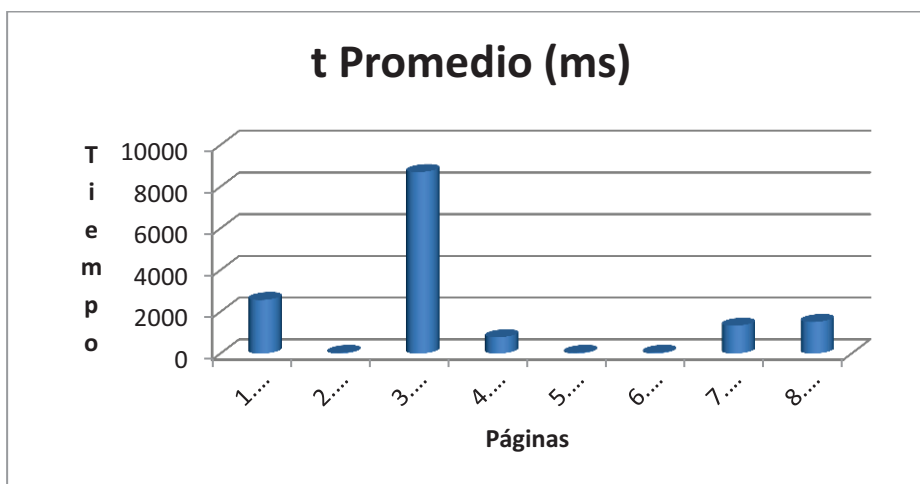
En la Figura 3.39 se puede observar que el consumo de CPU para esta prueba se mantuvo en promedio en el 43% de uso. Este es un valor manejable para un servidor Web de aplicaciones.



**Escenario 3. Realizar carro de compras como usuario visitante.**

Numero usuarios: 25.

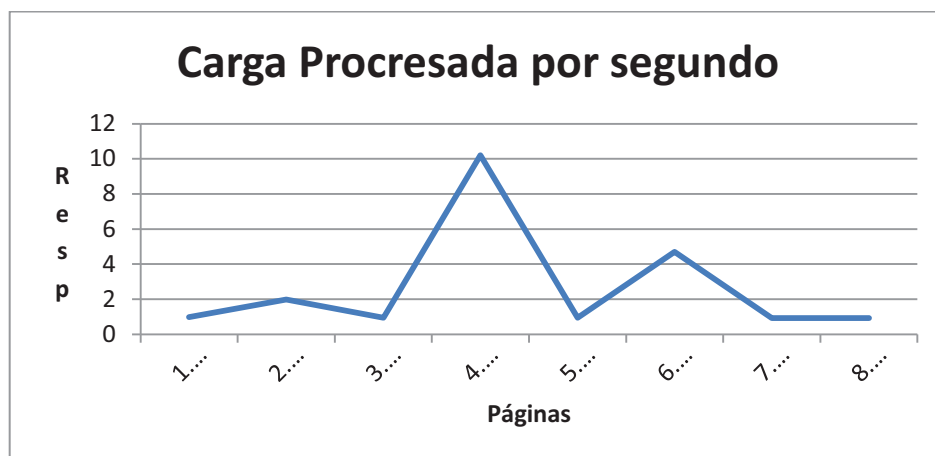
Tiempo de Respuesta



**Figura 3. 40 Tiempo de respuesta escenario realizar carro de compras visitante 25 usuarios**

Como se puede apreciar en la Figura 3.40, cuando se probó el escenario 3 hubo una petición del flujo que se demoró en promedio 8,7 segundos. Cabe mencionar que este valor de la tercera petición se debe a que esta página se encarga de recuperar las imágenes de los productos y la primera vez esta petición requiere bastante tiempo. Una vez cargados estas imágenes en el cliente, las siguientes peticiones toman menos de 1 segundo. Lo que hace que el tiempo promedio total para realizar una prueba tome cerca de un segundo.

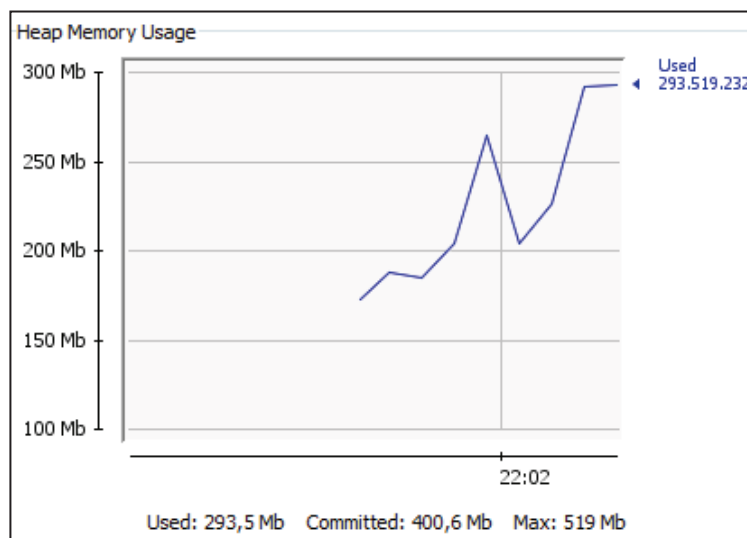
### Carga procesada por segundo



**Figura 3. 41 Carga procesada escenario realizar carro de compras visitante 25 usuarios**

El gráfico de la Figura 3.41 muestra que al realizar la prueba del escenario, la cuarta petición es que mayor carga procesada por segundo tiene, esto se da debido a que fue la petición que más realizó el cliente en el flujo y además el tiempo de respuesta de esta petición fue bajo.

### Uso de Memoria



**Figura 3. 42 Uso de Memoria escenario realizar carro de compras visitante 25 usuarios**

En la Figura 3.42 se puede observar que el consumo de memoria subió hasta 302 MB, que es el 58 % de la cantidad de memoria que tiene el servidor de

aplicaciones para la JVM de JBoss. Se considera un valor aceptable para el servidor.

### Uso de CPU

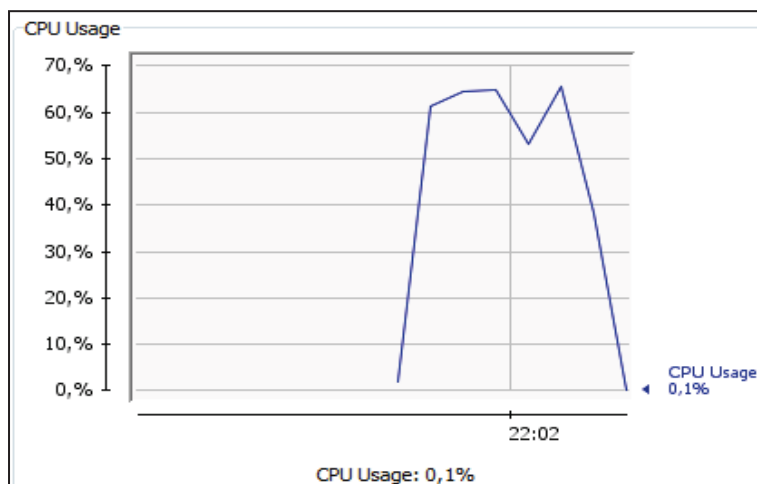


Figura 3. 43 Uso de CPU escenario realizar carro de compras visitante 25 usuarios

En la Figura 3.43 se puede observar que durante la prueba se alcanzó un consumo de CPU del 65.5%. Este porcentaje es considerado aceptable para un servidor web de aplicaciones.

Numero usuarios: 50.

### Tiempo de Respuesta

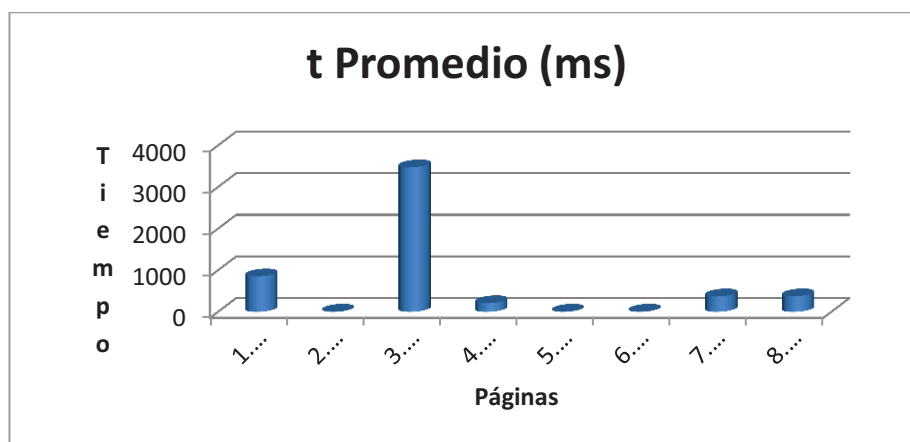


Figura 3. 44 Tiempo de respuesta escenario realizar carro de compras visitante 50 usuarios

En la Figura 3.44 se puede apreciar que como se indicó en las pruebas de este flujo para 25 usuarios, el tiempo para ejecutar la tercera petición tiene un promedio alto. El tiempo promedio de la prueba fue de 1s.

Carga procesada por segundo

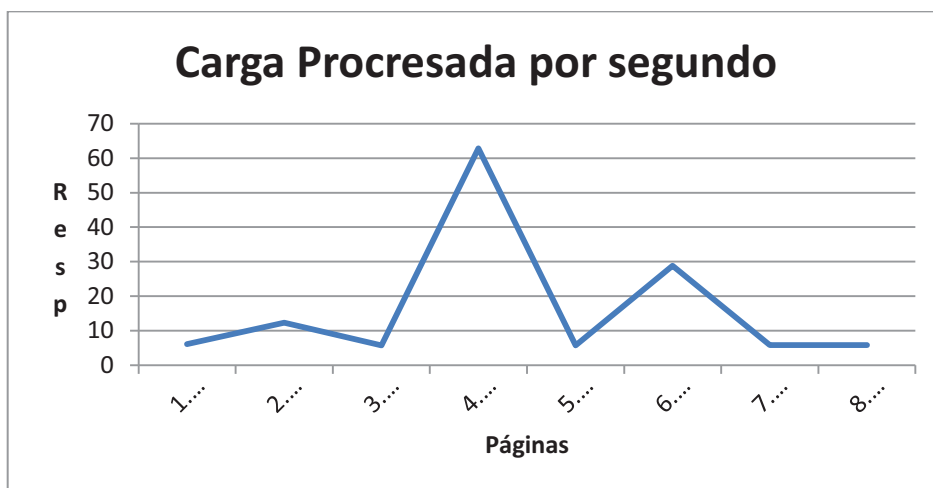


Figura 3. 45 Carga procesada escenario realizar carro de compras visitante 50 usuarios

La Figura 3.45 se asemeja mucho a la figura que se tenía para la prueba con 25 usuarios (Figura 3.41), y con ello se puede ver que la cuarta petición es la que ha procesado más requerimientos por segundo.

Uso de Memoria

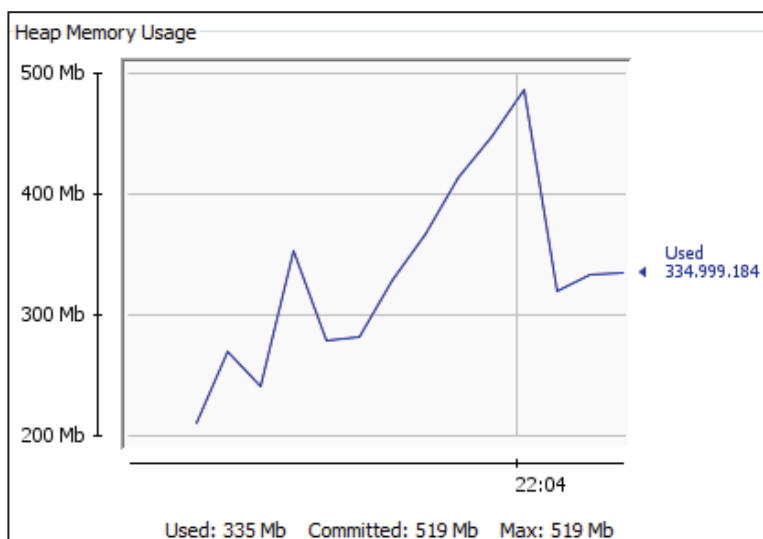


Figura 3. 46 Uso de Memoria escenario realizar carro de compras visitante 50 usuarios

En la Figura 3.46 algo muy importante a notar aquí es que el consumo de memoria incrementa para cada repetición de la prueba llegando a un valor de 448 MB valor que representa el 86%. Aquí se debería tomar precauciones para evitar el colapso de la aplicación por falta de memoria.

### Uso de CPU

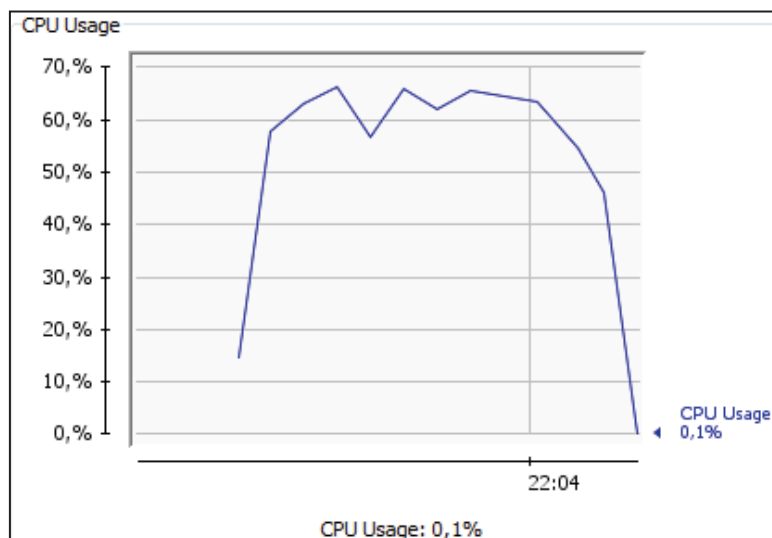


Figura 3. 47 Uso de CPU escenario realizar carro de compras visitante 50 usuarios

En la Figura 3.47 se puede notar que el consumo de memoria se mantiene entre el 60% y 70 %. Estos valores se consideran como aceptables para el rendimiento de CPU en un servidor de web de aplicaciones.

### 3.4.2 ANÁLISIS DE LAS PRUEBAS DE STRESS

Para la realización de las pruebas de Stress, se aumentó el tamaño de memoria asignado para el servidor de aplicaciones a 1024MB. Esto se lo hace debido a que con las pruebas de carga anteriores para 50 usuarios ya se presentaban problemas por el tamaño asignado.

Para aumentarla se debe editar la configuración de inicio de JBoss en el archivo: JBOSS\_HOME\bin\standalone.conf.bat, como se muestra en la figura 3.48.

```
rem # JVM memory allocation pool parameters - modify as appropriate.
set "JAVA_OPTS=-Xms128M -Xmx1024M -XX:MaxPermSize=512M"
```

Figura 3. 48 Configuración de tamaño de memoria asignado a JBoss

Luego de esto iniciar el servidor de aplicaciones y realizar las pruebas.

### Escenario 1. Realizar pedido de carro de compras como usuario registrado.

Para esta prueba se utilizó 175 usuarios con 5 repeticiones por cada usuario. En total se crearon 839 pedidos de los 875 esperados. El uso de memoria alcanzó el 70% tal como se puede ver en figura 3.49 mientras que el procesador alcanzó un valor del 74% tal como se aprecia en la figura 3.50.

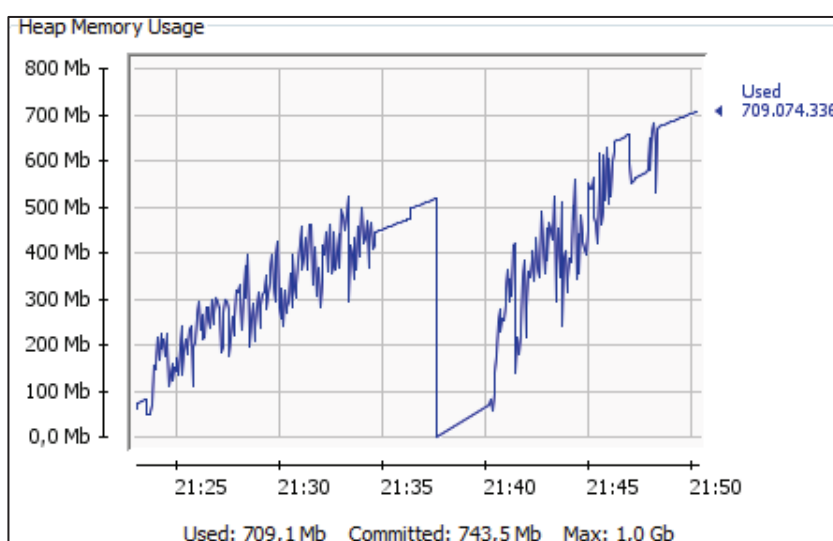


Figura 3. 49 Uso de Memoria prueba de Stress escenario realizar pedido



Figura 3. 50 Uso de CPU prueba de Stress escenario realizar pedido

## Escenario 2.Consulta de pedidos pendientes para usuario registrado.

Este escenario se llegó a probar hasta con 200 usuarios concurrentes y ahí se alcanzó en memoria el consumo del 80% como se aprecia en la Figura 3.51. El consumo de CPU estuvo en promedio en el 38% con picos a tomar en cuenta del 50% como se puede apreciar en la figura 3.52.

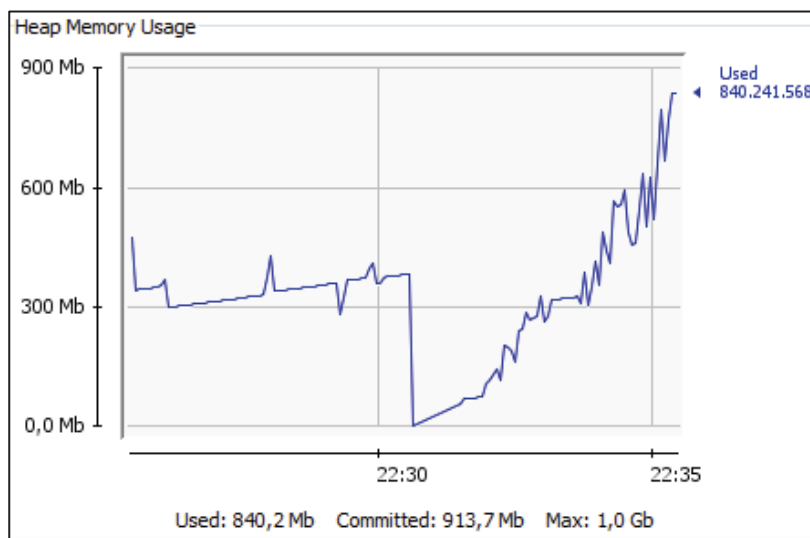


Figura 3. 51 Uso de Memoria prueba de Stress consultar pedidos pendientes.

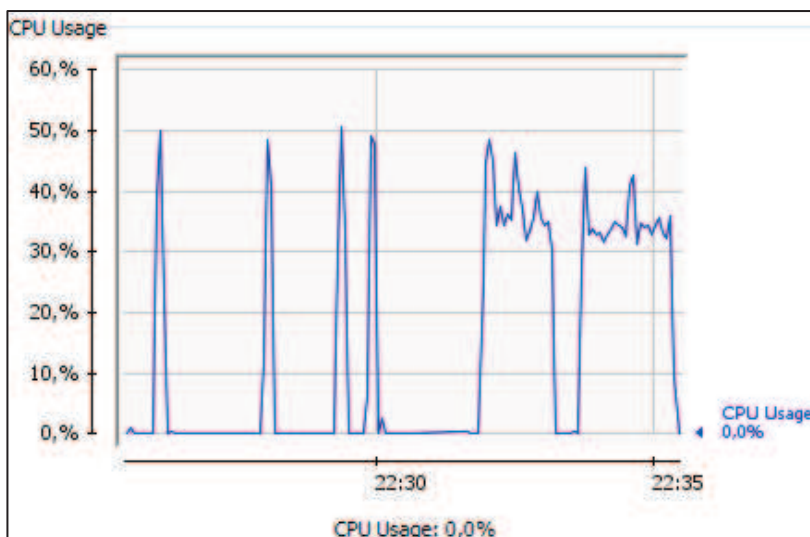


Figura 3. 52 Uso de CPU prueba de Stress consultar pedidos pendientes.

### Escenario 3. Realizar carro de compras como usuario visitante.

Se ha realizado la prueba con 125 usuarios concurrentes 5 repeticiones, y el tiempo promedio que se demoró cada prueba fue de 2.23 segundos. Alcanzándose un consumo de memoria del 86,9% como se puede apreciar en la Figura 3.53.

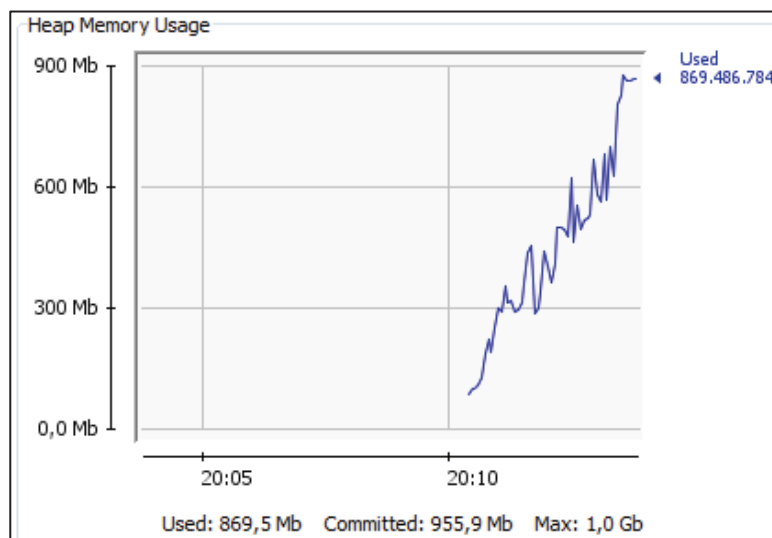


Figura 3. 53 Uso de Memoria prueba de Stress escenario Realizar Carro de Compras.

En cuanto al consumo de CPU se obtuvo un porcentaje promedio de 38% de uso con picos de 48%, como se lo puede apreciar en la Figura 3.54.

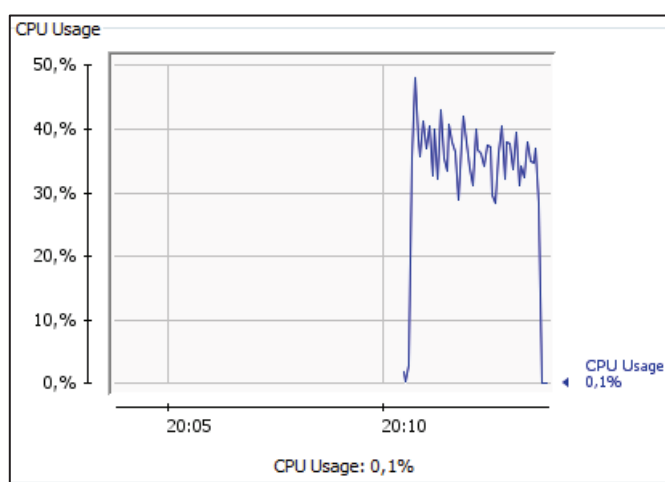


Figura 3. 54 Uso de CPU prueba de Stress escenario Realizar Carro de Compras.



## **CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES.**

Una vez finalizado este proyecto se han obtenido las siguientes conclusiones y recomendaciones:

### **4.1 CONCLUSIONES**

Mediante el desarrollo que involucró el presente proyecto, se pudo apreciar el beneficio que supone para una empresa la posibilidad de ofrecer sus productos en el Internet. Gracias a este sistema, el restaurante Piguari puede acceder a un nuevo mercado a través del Internet.

Durante el desarrollo, el uso del framework Primefaces sirvió de soporte en la realización y validación de interfaces web invirtiendo tiempos menores con respecto a otras herramientas HTML gracias a ello se adquirió el conocimiento de JSF y de su implementación.

La aplicación de cobros por tarjeta usando PayPal es segura, probada y certificada a nivel internacional. Integrarla a la aplicación del presente proyecto permitió despreocuparse de las implicaciones de seguridad del comercio electrónico.

En las Pruebas de carga se pudo constatar que al momento de aumentar de 25 usuarios concurrentes a 50 usuarios concurrentes el rendimiento del servidor se vio afectado, dando como resultado un tiempo de respuesta al cliente mayor a 1s que es el tiempo óptimo de respuesta.

SpringSecurity es un componente web que intercepta las peticiones web y verifica los controles de acceso y autorización implementados, de una forma fácil ya que no es necesario el desarrollo de todo un módulo de control de acceso. Al usar un componente probado y seguro se ahorra tiempo de desarrollo.

Las seguridades implementadas como canal seguro HTTPS, autenticación y autorización, certificados para pagos con PayPal y claves seguras no son suficientes sin un monitoreo constante a la aplicación para detectar posibles

problemas que no puedan ser cubiertos con las medidas de seguridad utilizadas en este proyecto.

El uso de la metodología ICONIX permitió que se desarrolle ágilmente la aplicación de tal manera que se satisface las necesidades que el cliente tiene para su negocio.

El uso de plataformas libres para el desarrollo este proyecto evitó el costo en licencias de software. También se pudo realizar la implementación con rapidez gracias a la documentación que se dispuso de las herramientas.

## 4.2 RECOMENDACIONES

Seguir todos los pasos en el orden que propone la metodología ICONIX, ya que con esto se garantiza un alto porcentaje en el éxito del proyecto.

Invertir un tiempo prudencial en la revisión de los requerimientos y actualización del modelo de dominio durante la etapa de diseño, ya que esto otorga un alto porcentaje de entendimiento del problema para satisfacer los requerimientos de los clientes.

Usar las facilidades de PayPal en los proyectos web inherentes al comercio electrónico con el fin de que los aspectos de seguridad en el pago sean manejados por PayPal y con ello preocuparse de la funcionalidad de aplicación propia.

Monitorear constantemente la aplicación para detectar nuevos problemas que no han sido cubiertos con las seguridades implementadas y para controlar el rendimiento del servidor de acuerdo a la escalabilidad de los usuarios.

Desarrollar aplicaciones multicapa ya que esto permite diferenciar las funcionalidades de los sistemas en capas. Con esto la aplicación gana en escalabilidad.

Realizar pruebas de carga y Stress ya que con esto se puede simular la posible carga de procesamiento que tendrá el servidor y la aplicación. Con esta información se podría realizar las correcciones necesarias para obtener el rendimiento deseado.

Con respecto al hardware utilizado para la ejecución de las pruebas de Stress, se evidencia que es necesario aumentar la cantidad de memoria ya que se alcanzó los valores umbrales en los cuales un servidor web disminuye su rendimiento.

## BIBLIOGRAFÍA

- [1] Comida a domicilio. [Online].  
[http://es.wikipedia.org/wiki/Comida\\_a\\_domicilio](http://es.wikipedia.org/wiki/Comida_a_domicilio)
- [2] Sammy Burgess. food-delivery-fast-food. [Online].  
<http://www.squidoo.com/food-delivery-fast-food>
- [3] Horace Tait. Food Delivery Services - A History. [Online].  
<http://ezinearticles.com/?Food-Delivery-Services---A-History&id=4767478>
- [4] J. hompson Arthur y Strickland A., *Administración Estratégica Conceptos y Casos*, III ed.: Mc Graw Hill, 2001.
- [5] [Online]. <http://mcdonalds.com.ec/>
- [6] Ivan Thompson. Definición de Misión. [Online].  
<http://www.promonegocios.net/mercadotecnia/mision-definicion.html>
- [7] [Online]. [www.dominos.com.ec/](http://www.dominos.com.ec/)
- [8] Fred Nichols,. Three Forms of Strategy: Corporate, Competition and Strategy in general. 2000
- [9] Lluís Codó Pla. gestionrestaurantes. [Online].  
[http://www.gestionrestaurantes.com/llegir\\_article.php?article=262](http://www.gestionrestaurantes.com/llegir_article.php?article=262)
- [10] [Online]. <http://www.iconixweb.com/>
- [11] [Online].  
<http://www.portalhuarpe.com.ar/Seminario09/archivos/MetodologiaCONIX.pdf>
- [12] Jhon Crupi, Dan Malks Deepak Allur, *CORE J2EE Best practices and Design Strategies*.
- [13] [Online]. <http://ccia.ei.uvigo.es/docencia/SCS/Tema5-1.pdf>
- [14] Angel Cob, *Diseño y programación de bases de datos*.
- [15] [Online]. <http://www.postgresql.org/docs/9.0/interactive/intro-what-is.html>
- [16] [Online].  
[http://soporte.tiendalinux.com/portal/Portfolio/postgresql\\_ventajas\\_html](http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html)

- [17] *JBoss Application Server STANDARDS BASED INFRASTRUCTURE FOR THE ENTERPRISE.*
- [18] Y.V. Natis, *CIO Update: Enterprise Java Lock-in Will Increase Through 2006 IGG-11122003-02.*, 2003.
- [19] *Agile Development ICONIX.*
- [20] *Apress Use Case Driven Object Modeling with UML.*
- [21] [Online]. <http://www.emagister.com/curso-diseno-base-datos-relacionales/diseno-logico-bases-datos>
- [22] [Online]. <http://www.wisegeek.com/what-is-information-security.htm>
- [23] [Online]. <http://www.instantssl.com/ssl-certificate-products/https.html>
- [24] [Online]. <http://wiki.fluxit.com.ar/display/PUBLIC/Spring+Security>
- [25] [Online]. <http://elenet.ws/art%C3%ADculos/8973-seguridad-en-servicios-web-autenticaci%C3%B3n-y-autorizaci%C3%B3n-interoperabilidad.html>
- [26] [Online]. <http://appsec.infonova.es/Seguridadintegrada/Desarrollodeaplicaciones/Autenticaci%C3%B3nautorizaci%C3%B3nyauditor%C3%ADa.aspx>
- [27] [Online]. <http://www.isocron.net/blog/2011/08/02/la-importancia-de-usar-contrasenas-seguras>
- [28] [Online]. <http://www.segu-info.com.ar/proteccion/clavesseguras.htm>
- [29] [Online]. <https://www.paypal.com/es/webapps/mpp/about>
- [30] [Online]. <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?página=botonPaypalCifrado>
- [31] [Online]. <http://elverdaderoblogdelaob.wordpress.com/2009/07/04/pruebas-unitarias-en-java-junit/>
- [32] [Online]. <http://www.w3.org/>
- [33] [Online]. <http://validator.w3.org/>
- [34] [Online]. <http://blog.powermapper.com/blog/post/W3-ARIA-why-doesnt-it-validate.aspx>

- [35] [Online]. [http://msdn.microsoft.com/es-es/library/ms182562\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/ms182562(v=vs.80).aspx)
- [36] [Online]. <http://jmeter.apache.org/>
- [37] [Online]. [http://www.linti.unlp.edu.ar/uploads/docs/usando\\_jmeter\\_para\\_pruebas\\_de\\_rendimiento.pdf](http://www.linti.unlp.edu.ar/uploads/docs/usando_jmeter_para_pruebas_de_rendimiento.pdf)
- [38] Jaime Naranjo, Información obtenida del material de la asignatura Auditoría y Evaluación de Sistemas Computacionales y analizado por los autores.
- [39] Philippe Kruchten, *The rational unified process: an introduction* , 2nd ed.: Addison Wesley Longman, Inc, 2000.
- [40] Philippe Kruchten Per Kroll, *Rational Unified Process Made Easy.*: Addison Wesley, 2003.
- [41] Schwabe Daniel , Olsina Luis, Pastor Oscar Rossi Gustavo, *Web engineering: modelling and implementing web applications.*: Springer, 2007.
- [42] Andreas Kraus Nora Koch. (2007) scientificcommons. [Online]. <http://en.scientificcommons.org/42681520>
- [43] www.areaordenadores.com. [Online]. <http://www.areaordenadores.com/Metodologias-Web.html>
- [44] Bárbara Mercerat Darío Andrés Silva. www.lifia.info.unlp.edu.ar. [Online]. <http://www.lifia.info.unlp.edu.ar/papers/2001/Silva2001.pdf>
- [45] María José Escalona Cuaresma. Departamento de Lenguajes y Sistemas Informaticos. [Online]. <http://www.lsi.us.es/docs/informes/EstadoActual.pdf>
- [46] [Online]. <http://www.asp.net/get-started>
- [47] [Online]. [www.php.net](http://www.php.net)
- [48] [Online]. <http://es.scribd.com/doc/64933584/45/Ventajas-de-J2EE>
- [49] [Online]. [http://es.wikipedia.org/wiki/C\\_Sharp](http://es.wikipedia.org/wiki/C_Sharp).
- [50] [Online]. [http://msdn.microsoft.com/es-es/library/fa1h9d0d\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/fa1h9d0d(v=vs.80).aspx)
- [51] [Online]. <http://www.danysoft.com/free/aspnet.pdf>
- [52] [Online]. <http://www.vaslibre.org.ve/publicaciones/phpflisol2006.pdf>
- [53] [Online]. <http://www.tizag.com/phpT/>

- [54] [Online]. <http://www.php.net/manual/es/>
- [55] [Online]. <http://st-curriculum.oracle.com/tutorial/DBXETutorial/index.htm>
- [56] [Online]. <http://www.oracle.com/technetwork/es/database/enterprise-edition/documentation/comparacion-ediciones-base-de-datos-1375494-esa.pdf>
- [57] [Online]. <http://www.postgresonline.com/journal/archives/51-Cross-Compare-of-SQL-Server,-MySQL,-and-PostgreSQL.html>
- [58] [Online]. <http://dspace.icesi.edu.co/dspace/bitstream/item/391/1/ljimenez-javaypontonet.pdf>.
- [59] [Online].  
<http://ftp.ceces.upr.edu.cu/centro/repositorio/Informaticos/Docencia/PIV/2do%20Semestre%203ero/Programacion%20IV/lo%20del%20servidor/Material%20de%20Consulta/LP/Java/Java%20vs.%20C%23.pdf>.
- [60] [Online]. [http://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server)
- [61] [Online]. <http://www.microsoft.com/sqlserver/en/us/product-info/overview-capabilities.aspx>
- [62] [Online].  
<http://www.microsoft.com/spain/sql/productinfo/features/top30features.msp>  
[x](#)
- [63] [Online]. [http://www.ehow.com/list\\_6458544\\_microsoft-sql-server-advantages.htm](http://www.ehow.com/list_6458544_microsoft-sql-server-advantages.htm)
- [64] [Online].  
<https://wikis.oracle.com/display/GlassFish/GlassFish+Server+Open+Source+Edition+3.1+Documentation+Reviews>
- [65] [Online].  
<http://svn2.assembla.com/svn/pgrado/Documentaci%C3%B3n%20Final/Documentaci%C3%B3n%20a%20Entregar/Anexos%20Investigaci%C3%B3n/1.%20Decisiones%20Tomadas.doc>
- [66] [Online]. [http://es.wikipedia.org/wiki/Internet\\_Information\\_Services](http://es.wikipedia.org/wiki/Internet_Information_Services)
- [67] [Online]. <http://www.iis.net/overview>

## ANEXOS

- Anexo 1 Descripción de casos de Uso.
- Anexo 2 Diagramas de robustez.
- Anexo 3 Diagramas de secuencia.
- Anexo 4 Código de las Interfaces.
- Anexo 5 Código de los Servicios de Negocio.
- Anexo 6 Script para creación de la base de Datos.
- Anexo 7 Configuración de Seguridad de pago con PayPal.
- Anexo 8 Proceso de Pruebas de validación del código de las interfaces.
- Anexo 9 Manual de Instalación del sistema.
- Anexo 10 Manual de Usuario del sistema.
- Anexo 11 Proceso de Pruebas de carga.
- Anexo 12 Documento completo del proyecto de titulación.