



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION

**UN SISTEMA WEB DE INFORMACION GEOGRAFICA, DE APOYO A
LA COMERCIALIZACION DE PRODUCTOS ORGANICOS DE UNA
COOPERATIVA DE PEQUEÑOS PRODUCTORES DE ECUADOR**

**TESIS PARA OPTAR AL GRADO DE MAGISTER EN
TECNOLOGIAS DE INFORMACION**

ALEX SANTIAGO CEVALLOS CULQUI

PROFESOR GUIA:
SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISION:
NANCY HITSCHFELD KAHLER
PABLO GONZALEZ JURE
MARCOS SEPULVEDA FERNANDEZ

SANTIAGO DE CHILE
DICIEMBRE 2011

Resumen Ejecutivo

En los últimos años se ha incrementado la tendencia al consumo de alimentos orgánicos, tanto por motivos de salud como ecológicos. La mayor parte de los países latinoamericanos que se han caracterizado por su biodiversidad y su forma de producción tradicional, se han ido convirtiendo en proveedores de productos orgánicos. Muchos de estos proveedores son pequeños productores que intentan colocar sus productos en el mercado. Lamentablemente no es simple conectar la oferta con la demanda, debido a que los productores no siempre están cerca de centros urbanos.

Para tratar de sortear esta limitante, este trabajo de tesis diseñó e implementó una solución integral, usable y escalable. Dicha solución está basada en una aplicación Web de información geográfica, la cual busca fortalecer el canal de comunicación y comercio entre productores y clientes. Las necesidades de estos actores se han determinado en base a la realidad de una cooperativa de pequeños productores ecuatorianos, dedicados al comercio de productos orgánicos.

La solución desarrollada involucra cuatro ámbitos del proceso de comercialización: (1) el registro de clientes, (2) la integración de los productores en unidades más grandes (por ejemplo en cooperativas), (3) la venta de los productos, y (4) el pago por las transacciones comerciales realizadas. Respectivamente estos servicios permiten, el registro tanto tradicional como geográfico de clientes actuales y potenciales, la integración de la producción individual de los productores, la búsqueda, selección y administración de productos (es decir, el shopping cart), y el pago online de una orden de compra.

La metodología de trabajo para la implementación del sistema se basó en un proceso iterativo, el cual nos permitió mejorar las funcionalidades de la aplicación en forma paulatina. En lo que respecta a su estructura, la aplicación está compuesta por múltiples capas (se utilizó el patrón Modelo-Vista-Controlador), donde su capa de lógica de negocio tiene un diseño orientado a objetos.

Dentro de los logros alcanzados por la aplicación tenemos, la recopilación y publicación de información de tipo hipertexto y geográfica, tanto del cliente como del productor. Esto ha aportado al intercambio de información de ubicación entre el productor y cliente, que es importante para la localizar un producto o el lugar de entrega de una orden de venta. Se ha logrado que la cooperativa de productores involucrada en este trabajo, disponga de un nuevo canal de comercio para sus productos, por medio del cual no sólo clientes actuales, sino también clientes potenciales pueden hacer su pedido y pagarlo de forma online.

En el futuro inmediato se buscará obtener mayor fluidez en este canal de comercio alternativo, incrementando el número de registro de clientes y órdenes de venta para la cooperativa. Una vez consolidado el entorno de trabajo con esta primera cooperativa, se buscará replicar el canal de comercio con otras cooperativas de productores con características similares.

Indice de Contenidos

1.	Antecedentes	8
1.1.	Introducción	8
1.2.	Problema a Resolver.....	9
1.3.	Justificación del Trabajo.....	11
1.4.	Objetivos de la Tesis.....	13
1.5.	Estructura del Documento.....	13
2.	Iniciativas Relacionadas	15
2.1.	Propuestas a Nivel Mundial	15
2.2.	Propuestas a Nivel América Latina	16
2.3.	Propuestas a Nivel Ecuador.....	18
3.	Análisis de la Solución.....	20
3.1.	Visión General.....	20
3.2.	Subsistemas de Negocio	21
3.3.	Requisitos Funcionales	23
3.4.	Casos de Usos.....	23
3.4.1.	Actores.....	23
3.4.2.	Subsistema de Registro.....	25
3.4.3.	Subsistema de Ventas	26
3.4.4.	Subsistema de Integración.....	27
3.4.5.	Subsistema de Pago	27
3.5.	Restricciones.....	28
3.6.	Requisitos de Calidad	29
4.	Diseño de la Solución.....	30
4.1.	Descripción General.....	30
4.2.	Ambiente Operacional.....	30
4.3.	Diseño Arquitectónico	32
4.3.1.	Diseño Estructural.....	32
4.3.2.	Modelo de Control.....	35
4.4.	Diseño Detallado.....	36
4.4.1.	Subsistema de Registro	37
4.4.2.	Subsistema de Venta	39
4.4.3.	Subsistema de Pago	42
4.4.4.	Subsistema de Integración.....	45
4.4.5.	Modelo de Clases	47

5.	Implementación de la Solución.....	51
5.1.	Herramientas y Contexto de Implementación	51
5.2.	Metodología de Desarrollo	52
5.3.	Subsistemas desarrollados	53
5.3.1.	Subsistema de Registro.....	53
5.3.2.	Subsistema de Venta.....	56
5.3.3.	Subsistema de Pago.....	58
5.3.4.	Subsistema de Integración.....	61
5.4.	Estructura de Paquetes.....	62
5.5.	Estructura de Clases y Vistas	64
5.6.	Funcionamiento de SGPO	65
5.6.1.	Instalación y Configuración	66
5.6.2.	Forma de Usar los Subsistemas	66
6.	Pruebas y Resultados Preliminares.....	76
6.1.	Pruebas por Subsistema.....	77
6.1.1.	Inspección 1 - Subsistema de Registro.....	78
6.1.2.	Inspección 2 - Subsistema de Venta.....	81
6.1.3.	Inspección 3 - Subsistema de Pago.....	85
6.1.4.	Inspección 4 - Subsistema de Integración	87
6.2.	Pruebas Globales.....	89
6.2.1.	Plan de Pruebas Funcionales	90
6.3.	Pruebas de Rendimiento.....	94
6.4.	Resultados Obtenidos.....	98
6.4.1.	Subsistema de Registro.....	98
6.4.2.	Subsistema de Venta.....	98
6.4.3.	Subsistema de Pago.....	99
6.4.4.	Subsistema de Integración.....	99
7.	Conclusiones y Trabajo a Futuro.....	100
8.	Bibliografía y Referencias.....	102
	ANEXO 1: Detalle de los Casos de Uso del Subsistema de Registro	103
	ANEXO 2: Detalle de los Casos de Uso del Subsistema de Venta	105
	ANEXO 3: Detalle de los Casos de Uso del Subsistema de Integración.....	108
	ANEXO 4: Detalle de los Casos de Uso del Subsistema de Pago.....	109
	ANEXO 5: Productores Ecuatorianos de la Cooperativa Probio	111
	ANEXO 6: Listado de Necesidades y Funcionalidades de SGPO.....	112
	ANEXO 7: Productos de la Cooperativa Probio	114

ANEXO 8: Artefacto de Control de Cambios	116
ANEXO 9: Plantilla de Plan de Pruebas	119

Indice de Tablas

Tabla 2.1 Evaluación de Sitios Web Ambito Internacional.....	15
Tabla 2.2 Evaluación de Sitios Web Ambito Regional	16
Tabla 2.3 Evaluación de Sitios Web Ambito Ecuatoriano	18
Tabla 4.1 Estados de la Orden de Venta.....	42
Tabla 6.1 Usuarios para Pruebas de Primera Instancia	78
Tabla 6.2 Lista de Pruebas para Inspección de Registro	79
Tabla 6.3 Resultado de Pruebas Inspección de Registro.....	80
Tabla 6.4 Tiempo de Pruebas Inspección de Registro	81
Tabla 6.5 Lista de Pruebas Inspección de Venta	81
Tabla 6.6 Resultados de Pruebas Inspección de Venta	83
Tabla 6.7 Tiempo de Pruebas Inspección de Registro	85
Tabla 6.8 Lista de Pruebas Inspección de Pago	85
Tabla 6.9 Resultado de Pruebas Inspección de Pago	86
Tabla 6.10 Tiempo de Pruebas Inspección de Pago	87
Tabla 6.11 Lista de Pruebas Inspección de Integración	87
Tabla 6.12 Resultado de Pruebas Inspección de Integración.....	88
Tabla 6.13 Tiempo de Pruebas Inspección de Integración.....	89
Tabla 6.14 Usuarios para Pruebas Inspección Global.....	90
Tabla 6.15 Lista de Pruebas Inspección Global	91
Tabla 6.16 Tiempo de Pruebas Inspección Global	94
Tabla 6.17 URLs para Pruebas de Rendimiento	95
Tabla 6.18 URLs y su Tiempo de Respuesta	95

Índice de Figuras

Figura 3.1 Capas de Presentación de SGPO	20
Figura 3.2 Entorno de Negocio de la Aplicación	21
Figura 3.3 Subsistemas de Negocio de SGPO	22
Figura 3.4 Actores de SGPO	24
Figura 3.5 Relación de los Subsistemas de Negocio	25
Figura 3.6 Casos de Uso del Subsistema de Registro	26
Figura 3.7 Casos de Uso del Subsistema de Venta	27
Figura 3.8 Casos de Uso del Subsistema de Integración	27
Figura 3.9 Casos de Uso del Subsistema de Pago	28
Figura 4.1 Ambiente Operacional de SGPO	31
Figura 4.2 Diseño estructural de SGPO	33
Figura 4.3 Arquitectura del Modelo de Control	35
Figura 4.4 Modelo de Dominio	36
Figura 4.5 Subsistemas de la Lógica de Negocio	37
Figura 4.6 Controladores y Funciones del Subsistema de Registro	38
Figura 4.7 Modelo de Clases del Subsistema de Registro	39
Figura 4.8 Controladores y Funciones del Subsistema de Venta	40
Figura 4.9 Modelo de Clases del Subsistema de Venta	41
Figura 4.10 Controladores y Funciones del Subsistema de Pago	43
Figura 4.11 Flujo de Proceso de Pago con PayPal	44
Figura 4.12 Modelo de Clases del Subsistema de Pago	45
Figura 4.13 Controladores y Funciones del Subsistema de Integración	46
Figura 4.14 Modelo de Clases del Subsistema de Integración	47
Figura 4.15 Modelo de Clases del Sistema	48
Figura 5.1 Paquetes de la Capa Lógica de Negocio	62
Figura 5.2 Organización de Archivos en SGPO	64
Figura 5.3 Clases de las Capas de Negocio	65
Figura 5.4 Secciones de la Interfaz de SGPO	67
Figura 5.5 Interfaz Inicial de SGPO	68
Figura 5.6 Interfaz de Registro de Ubicación	68
Figura 5.7 Interfaz de Ingreso de Usuario	69
Figura 5.8 Interfaz de Registro de Usuario	69
Figura 5.9 Interfaz de Dirección Almacenada	70
Figura 5.10 Sección de Interfaz de Selección de Productos	71
Figura 5.11 Búsqueda con Información Geográfica (producto)	71
Figura 5.12 Búsqueda con Información Geográfica (productor)	72
Figura 5.13 Interfaz de Ingreso de Ítems	72
Figura 5.14 Interfaz de Administración de Productos	73
Figura 5.15 Interfaz de Actualización de Productos	73
Figura 5.16 Interfaz de Confirmación de Orden de Venta	74
Figura 5.17 Interfaz de Pago con PayPal	74
Figura 5.18 Interfaz de Finalización de Pago	75
Figura 5.19 Interfaz de Integración de Producción	75
Figura 6.1 Flujo del Proceso de Pruebas	77
Figura 6.2 Estadísticas de Tiempo de Respuesta (10 usuarios)	96
Figura 6.3 Estadísticas de Tiempo de Respuesta (50 usuarios)	96
Figura 6.4 Estadísticas de Tiempo de Respuesta (100 usuarios)	97

1. Antecedentes

1.1. Introducción

La revolución industrial fue el hito que marcaría celeridad en el ritmo de producción de nuestras sociedades actuales, incentivando el mayor consumo. Esta época trajo consigo técnicas de producción que fortalecieron las economías a escala, así se adoptó el concepto de “más por menos”. Según los años pasaban, muchas áreas productivas se industrializaron, y en la época de los cincuenta bajo la metáfora “revolución verde, combatir el hambre” fue el turno del campo, así la producción agrícola se triplica.

Este incremento de producción, mejora el comercio internacional y acelera el proceso de globalización. Este fenómeno hace que en Ecuador y en el mundo, el mercado agrario sea abastecido por dos tipos de producción: el convencional y el orgánico. La globalización golpea fuertemente a la agricultura familiar-orgánica, debido a la competencia dispareja con la agroindustria. Una producción con alta tecnología es difícilmente igualable para aquellos que usan sólo sus manos para trabajar.

La preocupación de la producción en las últimas décadas ha estado enfocada en mejorar la cantidad de productos producidos por unidad de área. Sin embargo, con el pasar del tiempo nos hemos dado cuenta que esta producción enfocada a la cantidad (convencional) acelera la erosión de la tierra, impacta nuestros ecosistemas y nuestra capacidad de producción agrícola hacia delante, siendo éstos algunos de los factores incidentes en el cambio climático que hoy en día es más palpable¹.

Estos hechos junto con la falta de salubridad de productos de consumo masivo, empiezan a ser valorados por la sociedad, y así vuelve a recuperarse nuevamente el concepto de agricultura orgánica. Si bien la agricultura orgánica es aún poco conocida, va ganando adeptos todos los días. Por ejemplo, estos productos ya aparecen en las góndolas de los supermercados, y la gente está demostrando estar dispuesta a consumirlos, lo cual es un gran primer paso.

Para mejorar la inserción de la producción orgánica en los actuales mercados y buscar nuevos canales de comercio, es importante la organización de los productores. Sólo juntos podrán cubrir la demanda y competitividad exigida por la globalización. De su organización dependerá: la mejor visibilidad de sus productos, la creación de nuevos canales de comercio y el control de la intermediación.

En Ecuador hay varios intentos de promoción de los productos orgánicos, principalmente a través de pequeñas ferias que buscan insertarse en la vida diaria de nuestra sociedad. Esto ha sido fruto de la organización de agricultores que han gestionado con municipalidades, nuevos espacios para promocionar su trabajo y alcanzar nuevos mercados. Las TICs pueden ayudar a alcanzar estas metas. Particularmente la Web con sus nuevas carreteras de información ha fortalecido

¹ Servicios Web Ministerio de Medio Ambiente España 2011, Ministerio de Medio Ambiente España, último acceso 18 de Agosto de 2011 <http://www.mapa.es/ministerio/pags/biblioteca/revistas/pdf_reas/r090_01.pdf>.

enormemente los negocios electrónicos. Hoy en día algunos estratos productivos han empezado a potenciar sus ventas apoyándose en el e-commerce².

Cada vez son más los cibernautas que realizan transacciones online, y las estadísticas muestran que sus usuarios están en un continuo incremento. Se estima que de una población mundial de 6.768 millones de personas, el 26.6% tiene acceso a Internet³ y que de ellas el 84% compran online⁴.

En los últimos años el mercado electrónico ha brindado grandes muestras de innovación; hoy incluso brinda la posibilidad de comercializar frutas y legumbres. Su propuesta comercial basada en TICs rompe el paradigma de venta tradicional; ya que ahora el proveedor no tiene la necesidad de presentar al comprador su mercadería en un espacio físico real.

Este cambio promovido por la masificación de Internet y sus tecnologías asociadas, ha abierto un enorme nicho de mercado, donde el dinero en efectivo se convierte en dinero digital y todo gira en torno a una tarjeta de crédito/débito.

El área de las aplicaciones Web brinda facilidades de comunicación que están siendo constantemente fortalecidas con nuevas y robustas aplicaciones tecnológicas. Una de estas aplicaciones que promete tener un rol protagónico son los sistemas de información geográfica (GIS) [Far99, Bol05]. A través del uso de GIS muchos rubros del comercio y servicio han incrementado considerablemente su demanda. Por ejemplo, hoy en día casi todos los hoteles que cuentan con página Web, están geo-referenciados a través de GoogleMaps u otro GIS. De esa manera, los potenciales clientes pueden identificar el lugar exacto donde se encuentra el hotel, también conocer las atracciones que eventualmente hay en los alrededores, los medios de transporte a los que pueden acceder, etc. Si se combina este tipo de aplicaciones con tecnología móvil, se logra una solución que brinda ubicuidad de acceso a la información a los usuarios móviles. Algunos de los dominios de aplicación donde los GIS están tomando mucho protagonismo son los siguientes: turismo, hotelería, transporte, alimentario (restaurantes) y comercio, entre otros.

1.2. Problema a Resolver

Hoy en día el método de cultivo para la obtención de un producto agrícola empieza a tener importancia, comercialmente los productos orgánicos tienden a valorizarse y diferenciarse de los convencionales. Es decir, la producción orgánica empieza a tener importancia, pero lamentablemente, este tipo de producción enfrenta ciertas limitaciones como: la falta de organización de los productores, el restringido uso de tecnología por parte de estas personas y los limitados canales de comercio a los que ellos tienen acceso. Motivos como éstos provocan que los agricultores orgánicos se encuentren en una situación relativamente débil para comercializar sus productos,

² Servicios Web Kirsir 2005, Kirsir Internet Solutions, último acceso 29 Agosto 2011 <http://www.kirsir.ca/ecommerce/stats_and_trends.php>

³ Servicios Web Internet World Stats 2009, Internet World Stats, último acceso 29 Agosto 2011 <<http://www.Internetworldstats.com/stats7.htm>>.

⁴ Servicios Web Nielsen 2010, Global Trends in Online Shopping, Nielsen, último acceso 29 Agosto 2011 <http://en-us.nielsen.com/content/nielsen/en_us/insights/reports.HTML>.

obteniendo por lo tanto términos contractuales desventajosos y precios bajos⁵. Muchos de estos proveedores son pequeños productores, que intentan colocar sus productos en el mercado, pero lamentablemente no es simple conectar la oferta con la demanda, debido a que los productores no siempre están cerca de centros urbanos.

Para tratar de sortear esta limitante, el presente trabajo de tesis ha diseñado e implementado una solución integral, basada en una aplicación Web de información geográfica, la cual busca fortalecer el canal de comunicación y comercio entre productores y clientes. Como una forma de delimitar el alcance de la aplicación, la solución está enfocada a las particularidades de la cooperativa **Probio**. Los productores asociados a través de esta cooperativa, aprovechando la fortaleza del incremento del consumo de productos orgánicos y avizorando la amenaza de los intermediarios, se han integrado para dedicarse al comercio de hortalizas y frutas orgánicas. Su núcleo son 9 productores orgánicos ecuatorianos (ver Anexo 5) que estarán colaborando con este trabajo de tesis. A futuro se buscará la forma de extender el sistema para dar apoyo a cooperativas con otras realidades.

Para poder llevar a cabo este trabajo se hizo un análisis en terreno de los productos y procesos de comercialización de Probio durante el verano de 2010. Esta cooperativa se encuentra ubicada en la parroquia Toacaso, provincia de Cotopaxi, a una hora de la capital ecuatoriana Quito. La cooperativa posee líderes con estudios superiores que impulsan este tipo de agricultura. Ellos están abiertos al uso de nuevas tecnologías. Un ejemplo de su apertura es su actual recepción de pedidos vía mail. Sus mercados actuales son los siguientes:

1. En algunos casos, ferias generales dependiendo si el productor tiene acceso a ellas.
2. Entregas directas de canastas familiares o pedidos al detalle a restaurantes y familias.

Para el segundo caso, el medio de comunicación para la recepción de pedidos se ha ido fortaleciendo en el uso del teléfono y el mail. Ésta es una muestra de que la cooperativa está abierta al uso de nuevas tecnologías que le ayuden a mejorar su entorno de negocio. Considerando el escenario general y el particular de esta cooperativa, se identifican los siguientes problemas:

- P1.** Falta de disponibilidad de datos centralizados de productores orgánicos y clientes, que aporten a la organización de la cooperativa.
- P2.** Falta de visibilidad de sus productos en las zonas urbanas.
- P3.** Falta de fortalecimiento de los canales de comercio directo entre productor y cliente⁶.

⁵ Servicios Web IFAD 2010, Sobre las limitaciones que enfrentan los pequeños productores orgánicos, IFAD, último acceso 29 Agosto 2011 <http://www.ifad.org/evaluation/public_HTML/eksyst/doc/thematic/pl/organic_s.htm>.

⁶ Servicios Web IFAD 2010, Sobre la comercialización de la producción, IFAD, último acceso 29 Agosto 2011 <http://www.ifad.org/evaluation/public_HTML/eksyst/doc/thematic/pl/organic_s.htm>.

- P4.** Limitantes de integración y organización entre productores de una cooperativa o entre cooperativas, en lo que concierne a saber su cantidad de producción individual y/o agrupada.

1.3. Justificación del Trabajo

Las soluciones informáticas no sólo deben adaptarse a las particularidades de cada comunidad, sino también deben progresar en la dirección que evolucionan las comunidades. Puede ser que en un cierto instante de tiempo una solución implementada para una comunidad le sirva a otra. Lo que sí es difícil de creer es que una solución genérica que brinde servicio a más de una comunidad, evolucione como un producto único en la misma dirección que cada comunidad.

Esto hace que cada comunidad necesite contar con una solución que le sirva hoy, y que además ésta se pueda ajustar en función de la evolución natural de dicha comunidad. Este trabajo pretende mejorar la venta de productos en la cooperativa Probio cuyo núcleo son 9 productores ecuatorianos. Posteriormente, se verá la forma de que el servicio se pueda ajustar a las necesidades de otras comunidades.

El comercio electrónico de verduras y frutas es una estrategia de negocio joven. En el “primer mundo” existen casos de madurez y en América Latina recién hay algunos indicios de este modelo (ver Capítulo 2). Del análisis de los mejores casos se busca receptor como aprendizaje, las mejores prácticas y funcionalidades que éstos tengan con el fin de implementarlos en nuestra aplicación. En el caso particular de Ecuador:

- Este servicio está muy poco explotado. En general las soluciones existentes están enfocadas a nichos y comunidades que sólo pretenden solucionar sus problemas. Productores que no pertenecen a esa comunidad no tienen oportunidad de usar dichas herramientas.
- No se conoce en qué zonas existe producción orgánica y particularmente de qué productos, por lo tanto es difícil que la demanda encuentre al productor directo.

Crear una solución que involucre el uso de las TICs permitiría:

- Diseñar una tienda virtual que no requiere un gran esfuerzo para funcionar, que además estará visible para un número importante de clientes potenciales (familias ejecutivas urbanas). Éste sería un canal más de venta de los productos, el cuál probablemente mejorará e incrementará la demanda de lo que cultivan los productores.
- La representación de información de manera visual, respondiendo a las demandas de un usuario. Esto es algo a lo que la mayoría de la gente se ha acostumbrado en el último tiempo. Por lo tanto, el uso de una solución de TI sería natural para la mayoría de los potenciales clientes.

- Almacenar, procesar y automatizar datos de cooperativas de productores, clientes, ofertas y ventas de productos orgánicos, promocionando productos frescos que vienen directo del campo sin haber transitado por procesos de refrigeración, con una entrega a domicilio y a un mejor precio.
- Usar las carreteras de Internet, para crear una canal de comunicación directo entre cooperativa de productores y cliente, así se recortaría la brecha existente entre ellos.
- Mejorar comunicación y vínculos entre los productores, en término de cantidad de medios a través de los cuales su comunidad puede estar comunicada.

Se busca el uso de información geográfica ya que no existe información pública acerca de las zonas de cultivos orgánicos. De contarse con dicha información se fortalecería el comercio de la tienda virtual⁷ y los clientes potenciales podrían hacer uso de la ubicuidad para:

- Visitar a los productores, con el objetivo de ver el tipo de producto que tienen, y en base a eso establecer una relación comercial de más largo plazo entre ellos.
- Saber dónde encontrar un producto (o el productor más cercano), cuando un cliente está buscando un producto en especial.
- Saber dónde hacer una compra directa en el caso de encontrarse circulando por el lugar, por ejemplo en un viaje de turismo.

La geografía de la zona sobre la cual se va a trabajar es irregular, con caminos que generalmente no son triviales de encontrar. La información acerca de la ubicación física del productor va a permitir que los clientes puedan llegar con mayor facilidad. Típicamente una persona se va a sentir más cómoda comprando a través de un portal, si es que ésta ha tenido contacto con alguno de los productores. Desde ese punto de vista, la ubicación geográfica del productor cumple un papel muy importante.

Para superar estos desafíos se utilizará una buena parte del conocimiento adquirido en el transcurso de éstos dos últimos años de estudio. Desde la planificación y gestión del proyecto, hasta la ingeniería y calidad del software. El desafío es reflejar el fortalecimiento humanista y técnico que, como profesional, he adquirido durante el programa de Magíster.

En el aspecto humanista, el aprendizaje más relevante ha sido la indagación de la verdadera motivación personal (Liderazgo), la búsqueda de una actividad animada y no forzada. Esto me ha estimulado a la planificación del presente proyecto cuyos servicios se alinean con mis expectativas personales y valores.

El trabajo hará uso de las buenas prácticas (también aprendidas en el programa) para llevar a cabo el proceso de desarrollo de software, desde su especificación de

⁷ Servicios Web The Geography of Transport System 2011, Trade and Commercial Geography, último acceso 29 Agosto 2011
<<http://people.hofstra.edu/geotrans/eng/ch7en/conc7en/ch7c2en.HTML>>

requisitos hasta su prueba. El hecho de profundizar y practicar cada una de sus fases en talleres demostrativos, es una buena base para la replicación en nuestra aplicación. Además, se ha adquirido conocimiento importante en el ámbito de las arquitecturas de software, que ha permitido el diseño de una solución escalable y de fácil mantención. Esto con la perspectiva de que con el tiempo podamos ir fortaleciendo las cualidades del software, para que se pueda convertir en un sustento económico no sólo para los clientes del mismo, sino también para el proponente.

Finalmente, se utilizarán conceptos de programación orientada a objetos y herramientas orientadas a lenguajes de alto nivel. Estos componentes serán usados para brindar, a la parte interna de la aplicación, la confiabilidad y seguridad necesaria al momento de procesar la información.

1.4. Objetivos de la Tesis

El objetivo general del proyecto de tesis es diseñar e implementar una aplicación Web, que embeba un sistema de información geográfica, y que permita a pequeños agricultores orgánicos de Ecuador, tener una mejor visibilidad de su ubicación y producción agrícola. Se espera que esta solución les ayude a superar las limitaciones antes enunciadas. Se espera también que esta herramienta permita la integración de agricultores y clientes con el fin de crear una red laboral dedicada al comercio agrario orgánico. Los objetivos específicos que se desprenden del objetivo principal son los siguientes:

- Diseñar una solución tecnológica capaz de almacenar, recuperar y representar información tanto tradicional como geográfica, en forma eficiente.
- Desarrollar un portal Web que permita compartir información entre productores y clientes en relación con oferta y demanda de productos orgánicos.
- Implementar una tienda virtual que permita la búsqueda, selección y venta de productos orgánicos con pago online.

1.5. Estructura del Documento

Este documento de Tesis está estructurado de la siguiente manera:

Capítulo 1: Antecedentes: Motivación, planteamiento del problema y objetivos.

Capítulo 2: Iniciativas relacionadas: Apreciación de aplicaciones con similar enfoque, en el ámbito internacional, regional y ecuatoriano.

Capítulo 3: Análisis de la solución: Contexto operacional, definición, distribución y descripción de las actividades de los requisitos funcionales.

Capítulo 4: Diseño de la solución: Levantamiento de la arquitectura estructural y detallada de la aplicación.

Capítulo 5: Implementación de la solución: Descripción de la forma de elaboración y uso de la aplicación.

Capítulo 6: Pruebas y resultados preliminares: Planificación, verificación y evaluación de casos de prueba.

Capítulo 7: Conclusiones y trabajo futuro: Inferencias obtenidas con el desarrollo del presente trabajo y perspectivas posteriores.

Capítulo 8: Bibliografía y referencias: Textos de apoyo.

Anexos: Documentos con información complementaria a lo expuesto en el cuerpo de la tesis.

2. Iniciativas Relacionadas

En esta sección se presenta un listado de aplicaciones con una visión relacionada a la propuesta de tesis. Las mismas poseen un adecuado nivel de estabilidad y madurez. Se ha evaluado el ámbito internacional, regional y ecuatoriano, para analizar las ventajas y desventajas de las experiencias de determinados casos, así se obtiene un aprendizaje que es considerado en nuestro trabajo.

2.1. Propuestas a Nivel Mundial

La Tabla 2.1 presenta la evaluación de dos aplicaciones del ámbito internacional que se dedican a la comercialización de productos orgánicos.

Tabla 2.1 Evaluación de Sitios Web Ambito Internacional

ENTIDAD	PAIS	VENTAJAS	DESVENTAJAS
Greenling www.greenling.com	USA	<ul style="list-style-type: none">• Poseen un representativo número de clientes.• Para la entrega del producto usan el ZIPCODE del cliente.• Sus productos tienen buena reputación por su exquisitez.	<ul style="list-style-type: none">• Disponible en algunos estados.
RiverFood www.riverford.co.uk	Inglaterra	<ul style="list-style-type: none">• El precio del producto incluye costos de entrega.• Entrega inmediata en 24 horas.• Buena sincronización con servicio de entrega Express.• Reconocimiento en medios de comunicación.	<ul style="list-style-type: none">• Alto costo de transporte.

GREENLING

Análisis:

Ésta es una empresa norteamericana cuya historia cuenta que fue creada por un joven Ingeniero Químico. Su sensibilidad por la naturaleza lo llevo a idear un sistema de comercio de alimentos saludables. Éste consistía en unificar el uso de la tecnología con productos frescos de altos nutrientes y ecológicos. La forma de trabajo de su aplicación Web parte con el ingreso del zipcode del cliente, así se valida si su ubicación está al alcance de los recorridos de entrega.

Una vez que ingresa el cliente, una interfaz amigable le permite administrar su canasta de compras o conocer sus productores. El uso de ventanas emergentes le brinda facilidad al usuario al momento de elegir sus productos. Adicionalmente tiene un innovador calendario de entregas, el cual le permite al cliente elegir las fechas que desea recibir su canasta de compras. La empresa posee su propio transporte lo que da lugar a un servicio integral.

Aprendizaje:

- Es importante conocer inicialmente la ubicación del cliente potencial para definir si el servicio tiene cobertura en determinada área.
- El uso de ventanas emergentes brindan al usuario una mejor usabilidad de la administración de su canasta de compras.

RIVERFORD

Análisis:

Riverford es una empresa inglesa que fue creada por un agricultor que empezó haciendo entregas de vegetales a treinta amigos cercanos. Con el pasar del tiempo los pedidos se incrementaron y actualmente poseen entregas de 47000 cajas semanales alrededor de Inglaterra. Su demanda es cubierta a través de una alianza de granjas. Su aplicación Web, al igual que Greenling, posee un servicio similar de ubicación por zipcode, pero en este portal no es necesario ingresarlo para proceder a la administración de la canasta de compras. Esto debido a que la cobertura del servicio de esta empresa es mucho más amplia. Además de la posibilidad que tiene el usuario de elegir los productos que desee agregar a su canasta de compras. El portal oferta los mismos productos en cajas de presentación prediseñadas de acuerdo al tipo de producto o cantidad de personas. El portal posee otros servicios de valor agregado como información de recetas de cocina. También notificaciones de eventos en las granjas e invitaciones para fomentar sus visitas.

Aprendizaje:

- Cuando la cobertura del servicio es amplia, no es necesario partir conociendo la ubicación del cliente potencial, algo que en nuestra situación no sucede.
- El diseño de cajas de presentación son una alternativa de compra más rápida.

2.2. Propuestas a Nivel América Latina

La tabla 2.2 presenta la evaluación de dos aplicaciones del ámbito regional que se dedican a la comercialización de productos orgánicos.

Tabla 2.2 Evaluación de Sitios Web Ambito Regional

ENTIDAD	PAIS	VENTAJAS	DESVENTAJAS
Directo del Campo www.directodelcampo.com	España, Argentina, Colombia, Chile, México.	<ul style="list-style-type: none">• Respaldada por robustos inversionistas en el mercado digital.• Conocer la valoración y opiniones del productor.	<ul style="list-style-type: none">• Alto costo para la entrega del producto.• Red de productores en América Latina en formación.
Vegetales y Frutas www.vegetalesyfrutas.cl	Chile	<ul style="list-style-type: none">• Acumulación de puntos por compra realizada.• Rango de tiempo para entrega, 24 horas.	<ul style="list-style-type: none">• Usabilidad del sistema.• Pago por transferencia a cuenta corriente.

DIRECTO DEL CAMPO

Análisis:

Éste es un proyecto de la empresa española GroupIntercom, la cual se dedica a la creación de negocios por Internet. Su aplicación Web está disponible en España y en varios países americanos. Los servicios que ofrece están focalizados a promover el contacto cliente-productor. Su catálogo de productos se va fortaleciendo con el registro y contacto de nuevos productores que deciden ajustarse a las políticas de Directo del Campo.

Cuando un usuario cliente efectúa una compra en el sitio Web. El productor implicado es notificado a través de un mensaje en su teléfono móvil. Para la entrega del producto, Directo del campo revisa el acuerdo de entrega y forma de pago establecido con el productor.

Finalizada la compra el cliente puede exponer su opinión sobre el productor, así este último va adquiriendo una reputación en la aplicación. Los sitios Web de España y Argentina se encuentran más consolidados, ya que registra una red más grande de productores.

Aprendizaje:

- Generar un ambiente de reputación del productor con registro de opiniones y una calificación, ayuda a brindar mayor confiabilidad al cliente al momento de comprar.
- Desvincular al sitio Web del efecto del pago, permite una relación más directa entre cliente y productor, sin embargo esto incrementa el tiempo de finalización de la transacción.

VEGETALES Y FRUTAS

Análisis:

Vegetales y Frutas es una micro-empresa chilena emprendida por un Ingeniero Agrónomo y un Ingeniero Comercial. Esta empresa se dedica a la comercialización de productos orgánicos en la región metropolitana. Su servicio no está focalizado en entregar información de sus productores, sólo receptan órdenes de compra y las despachan.

De lunes a viernes la entrega de la compra es gratis y su cobertura alcanza a casi todas las comunas de Santiago. Si solicita una entrega fuera de los horarios definidos ésta será facturada. El pago se realiza a través de transferencias electrónicas a una cuenta corriente.

Este portal motiva el consumo del cliente, ofreciendo premios y descuentos a través de la acumulación de puntos por cada compra realizada. Ésta es una política para evaluar y mantener la fidelidad de sus clientes.

Aprendizaje:

- Realizar un cobro adicional por la compra, cuando el cliente solicita que se entregue su producto en un horario fuera de los establecidos.
- La oferta de puntos por compra realizada, motiva al cliente por los descuentos recibidos, de esa manera el proveedor mantiene la fidelidad del cliente.

2.3. Propuestas a Nivel Ecuador

La tabla 2.3 presenta la evaluación de dos aplicaciones del ámbito ecuatoriano que se dedican a la comercialización de productos orgánicos.

Tabla 2.3 Evaluación de Sitios Web Ambito Ecuatoriano

ENTIDAD	PAIS	VENTAJAS	DESVENTAJAS
De la Mata a la Olla delamataalaolla.wordpress.com	Ecuador	<ul style="list-style-type: none">• Considerable red de productores, aledaña a Quito.• Buenas políticas de trabajo comunitario y entrega de productos.	<ul style="list-style-type: none">• Tecnología usada para la canasta de compras no posee buena usabilidad.
Zapallo Verde www.zapalloverde.com	Ecuador	<ul style="list-style-type: none">• Buenas relaciones personales en la cooperativa, las que fortalecen el comercio.• Variado catálogo de productos caseros.	<ul style="list-style-type: none">• No permite el registro online de usuarios clientes.• Pago por transferencia electrónica.

DE LA MATA A LA OLLA

Análisis:

Ésta es una micro-empresa ecuatoriana, conformada por jóvenes profesionales oriundos de la ciudad de Sangolquí-Quito. Inicialmente se dedicaban al levantamiento de proyectos de desarrollo.

Con esta experiencia y la perspectiva de comercializar productos libres de químicos conjuntamente con productores de la zona, inician el cultivo de hortalizas en un huerto de su propiedad. Poco a poco sus clientes se han incrementado y han conseguido procesar productos elaborados con insumos orgánicos.

Los servicios de su sitio Web están implementados a través de un navegador de contenidos, llamado WordPress. Sus contenidos son amigables y modificables, pero al momento de efectuar acciones dinámicas (como las de una canasta de compras), el portal tiene sus limitantes. El manejo de su canasta de compras es engorroso, requiere ejecutar un hipervínculo por cada producto que el usuario desee.

Sin embargo su experiencia y fortaleza están centradas en su buena logística de relación y capacitación con los productores. También las políticas de la forma de entrega del producto.

Aprendizaje:

- La implementación de una canasta de compras con WordPress no brinda una buena usabilidad al cliente.
- Con una amplia gama de clientes actuales que ofrece a la micro-empresa fidelidad de consumo, las limitantes de tecnología pasan desapercibidas.

ZAPALLO VERDE

Análisis:

Esta micro-empresa ecuatoriana, está conformada por socios comerciales de las zonas urbanas y rurales de la ciudad de Quito. Su visión es establecer relaciones de mercado, pero sobre todo personales, en torno a proveedores y consumidores de productos saludables.

Los servicios de su página de pedidos no poseen una sección de registro de clientes. El sitio Web está dirigido únicamente a aquellas personas que de forma presencial acuden a su local a comprar. Después de establecer una relación personal, la cooperativa le entrega al cliente un usuario y password, para que pueda hacer su compra de forma online si así lo desea.

El sitio Web posee un amplio catálogo de productos. La forma de pago es por transferencia electrónica a una cuenta corriente.

Aprendizaje:

- Además del enlace virtual, para esta micro-empresa es importante el contacto personal con productores y clientes. Por esta razón fomenta la organización de eventos de participación entre ellos, como por ejemplo mingas, ferias y/o foros.
- La incorporación de productos con valor agregado a la tienda virtual. Mejor aún si éstos son preparados por la misma organización. Por ejemplo, en este caso se preparan productos con recetas caseras. Los mismos que posteriormente son ofertados en su portal Web.

3. Análisis de la Solución

En esta sección se presenta una visión general del contexto de la aplicación. El punto de partida fue la definición de necesidades que presenta la cooperativa Probio (ver Anexo 6). De la lista de necesidades se obtuvieron los requisitos funcionales del sistema.

En la clasificación de las funcionalidades, surgió la necesidad de organizarlas y distribuirlas en distintos subsistemas de negocio. Finalmente se definieron las actividades (casos de uso) involucradas en cada uno de los subsistemas de negocio.

3.1. Visión General

El sistema Web de información geográfica de apoyo a la comercialización de Productos Orgánicos (SGPO), brinda un acercamiento entre campo y ciudad, a través de un canal de comercio directo entre productores y clientes. Así, la cooperativa cambia el contexto de su comercio digital actual, basado en la recepción de pedidos de sus clientes por vía telefónica o mail, hacia el uso de un sistema Web que entrega los servicios planteados en los objetivos de esta tesis. Es importante mencionar que la cooperativa mantiene su canal de comercio a través de las ferias libres.

SGPO ha sido implementado en su totalidad usando herramientas OpenSource, que han permitido el manejo de aplicaciones geográficas y la implementación de la lógica de negocio. Para el almacenamiento de datos geográficos, se ha hecho uso del motor de base de datos PostgreSQL⁸ con su extensión PostGis⁹.

La estructura del sistema está basada en un Modelo-Vista-Controlador, para lo cual se ha hecho uso del Framework Spring¹⁰, que posee las capas presentadas en la Figura 3.1:

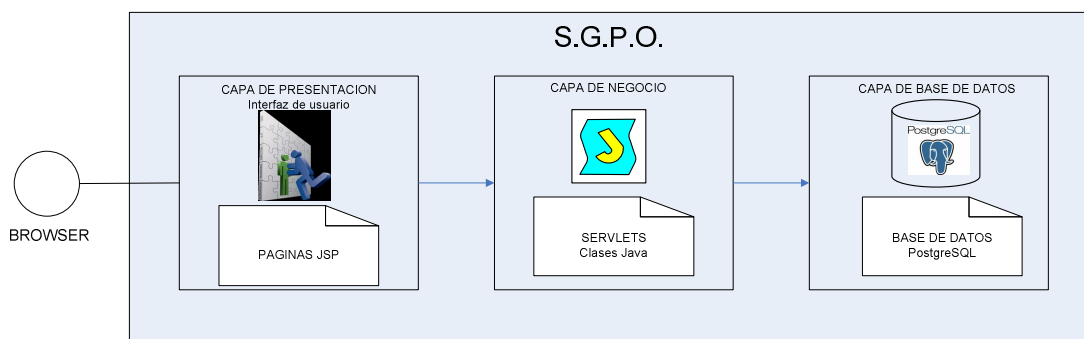


Figura 3.1 Capas de Presentación de SGPO

⁸ Servicios Web PostgreSQL 2011, PostgreSQL Global Development Group, último acceso 05 Septiembre 2011 <<http://www.postgresql.org>>

⁹ Servicios Web PostGIS 2011, OSGeo Project, último acceso 05 Septiembre 2011 <<http://postgis.refractor.net/>>

¹⁰ Servicios Web Spring Source 2011, Spring Source Community, último acceso 05 Septiembre 2011 <<http://www.springsource.org>>

- 1) *Capa de presentación*: Implementada usando tecnología Java Server Pages (JSP).
- 2) *Capa de negocio*: Contiene los paquetes de clases java para el control de acciones que la interfaz solicita.
- 3) *Capa de Base de datos*: Contiene tablas físicas en PostgreSQL, que poseen persistencia con las clases java, por medio de Hibernate¹¹.

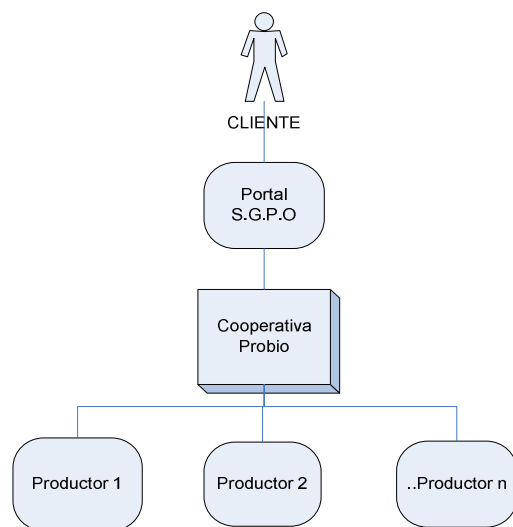


Figura 3.2 Entorno de Negocio de la Aplicación

El portal es de acceso público y permite el registro de datos de cuenta y posición geográfica del usuario cliente. Toda venta que se realiza, asocia un cliente registrado con uno o varios productores en el caso de que la compra supere la capacidad independiente de los agricultores o falte variedad de productos.

Así, SGPO integra la producción total de toda la cooperativa, permitiendo que el usuario cliente administre la compra de productos de diferentes agricultores en una canasta de compras. La venta relaciona un cliente con uno o varios productores. Cuando esta información es obtenida, la cooperativa es la responsable de la preparación y entrega del pedido. También, es la que recibe el dinero de la compra en una cuenta que se ha asignado para dicho fin. Con este contexto, la aplicación contempla el esquema de trabajo de la Figura 3.2.

3.2. Subsistemas de Negocio

Con la visión expuesta y definidas las necesidades de la cooperativa (ver anexo 6), se procedió a distribuir las características y funcionalidades consideradas en la aplicación. Es así como SGPO es estructurado con cuatro subsistemas de negocio fundamentales (ver Figura 3.3): subsistema de registro, subsistema de venta, subsistema de integración y subsistema de pago. Cada uno de éstos se describe brevemente a continuación.

¹¹ Servicios Web Hibernate 2011, Hibernate Jboss Community, último acceso 05 Septiembre 2011 <<http://www.hibernate.org>>

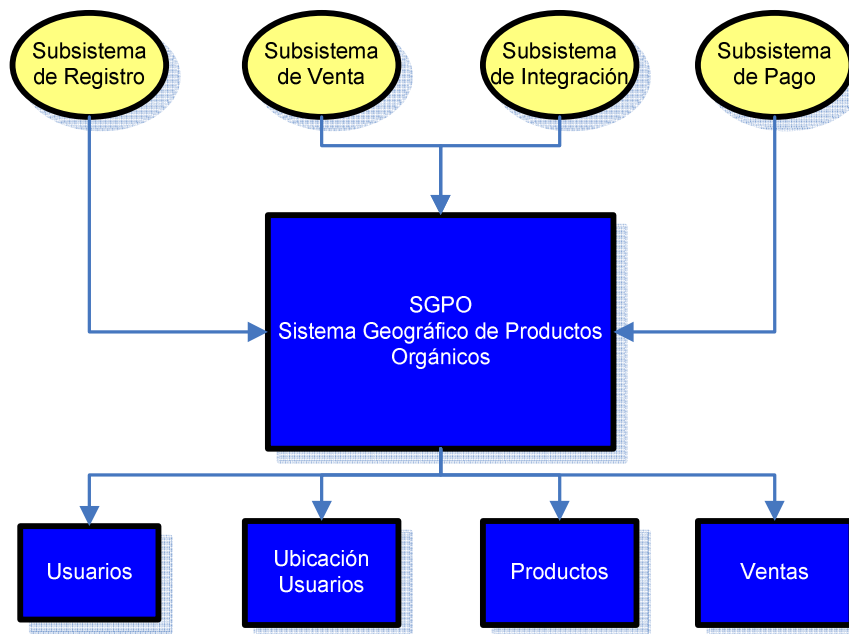


Figura 3.3 Subsistemas de Negocio de SGPO

- *Subsistema de Registro (S1)*: Este subsistema administra el registro de datos de cuenta y ubicación geográfica del cliente actual o potencial. De este subsistema se obtiene datos de Usuarios y Ubicación de Usuarios.
- *Subsistema de Venta (S2)*: Este subsistema parte de la búsqueda geográfica de productos por la categoría: productor, fruta u hortaliza. La búsqueda da lugar a la presentación de productos de uno o varios productores. Del resultado de la búsqueda o independiente a ella, el usuario puede administrar la selección de productos en una canasta de compras. De este subsistema se obtiene una orden de compra en estado inicial, la cual finalizará cuando la orden sea pagada.
- *Subsistema de Integración (S3)*: En este subsistema se unifica la producción individual de cada uno de los productores pertenecientes a la cooperativa. Con el fin de satisfacer pedidos que superen la capacidad individual de los productores.
- *Subsistema de Pago (S4)*: Una vez confirmada la canasta de compras, se procede a su pago online por medio de PayPal¹². Registrados los datos de pago y confirmada la transacción, se procede al control de stock de productos y el cambio de la orden de venta a su estado final.

Los subsistemas S1 y S2 implican la interacción del usuario con la cartografía de la aplicación geográfica. El fin es obtener datos relacionados a su cuenta, así como también sus datos geográficos de ubicación.

¹² Servicios Web PayPal 2011, PayPal Pte. Ltd., último acceso 05 Septiembre 2011 <<https://www.paypal.com/>>

3.3. Requisitos Funcionales

Cada uno de los subsistemas establecidos tiene las siguientes funcionalidades.

S1. Subsistema de Registro:

- Registrar datos de cuenta de un usuario.
- Identificar datos de cuenta de un usuario registrado, a través de un usuario y password. Después de validar sus datos se genera una sesión.
- Registrar datos de ubicación de un usuario, después de validar una sesión activa.
- Administrar cuenta de usuario.
- Administrar ubicación de usuario.

S2. Subsistema de Venta:

- Buscar geográficamente por las siguientes categorías: clientes, productores, frutas y/o hortalizas.
- Recibir información de ubicación, de la categoría de búsqueda aplicada.
- Administrar la selección de productos en una canasta de compras. La orden de venta puede estar relacionada a uno o varios productores. La administración implica añadir y retirar la cantidad de productos que se desee de la canasta de compras.
- Guardar y confirmar la orden de venta.
- Reducir el tiempo de administración de la canasta de compras, con la recuperación de la última canasta de compras del usuario cliente.

S3. Subsistema de Integración:

- Integrar la producción individual de los productores de la cooperativa.
- Presentar el catálogo de productos, con datos de producción que representan el valor acumulado de la producción individual de los productores.

S4. Subsistema de Pago:

- Conectar la orden de venta con el servicio de pago online de PayPal.
- Ejecutar el pago de la orden de venta.
- Confirmar y validar el pago de la orden.
- Actualizar el stock de productos y el estado de la orden de venta.

3.4. Casos de Usos

3.4.1. Actores

En la Figura 3.4, se muestra los actores que interactúan con SGPO. A continuación una breve descripción de ellos.

Cliente: Usuarios con afinidad al consumo de productos orgánicos. Pueden ser vigentes consumidores (clientes actuales) de los productos de la cooperativa o cibernautas (clientes potenciales) que localizaron el portal. Se registran en el portal con el fin de buscar y demandar la compra de productos de la cooperativa.

Productor: Integrante de la cooperativa. Oferta sus productos a través del portal y revisa pedidos efectuados por clientes para entregarlos.

Administrador de Cooperativa: Es un representante de la cooperativa que tiene igual actividad que el productor, pero adicionalmente tiene la responsabilidad de fijarse en los contenidos que la cooperativa oferta.

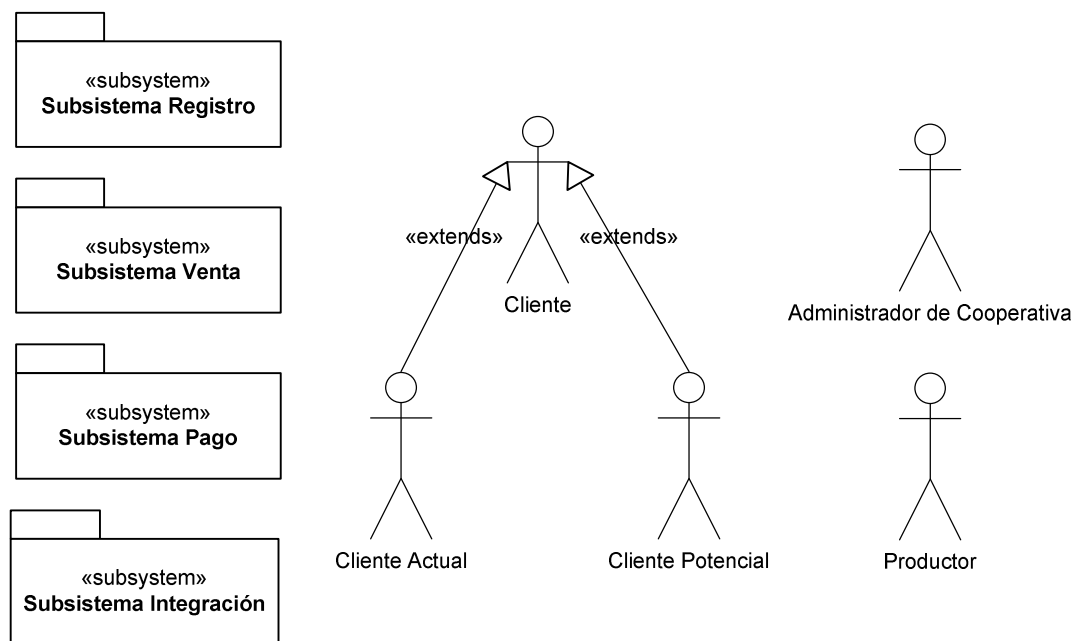


Figura 3.4 Actores de SGPO

En el diagrama de la Figura 3.5 se presenta todos los subsistemas definidos y la relación existente entre cada uno de ellos. Adicional se presenta las actividades de los subsistemas planteados a nivel general.

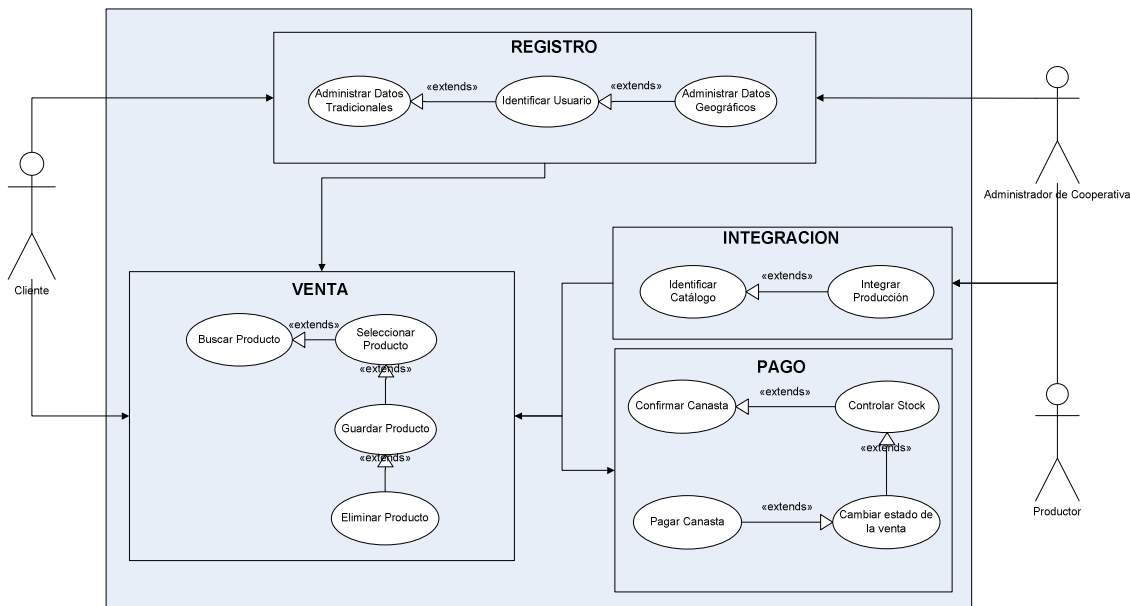


Figura 3.5 Relación de los Subsistemas de Negocio

El flujo normal de interacción del cliente con los subsistemas tiene la siguiente secuencia: registro, venta, integración (opcional) y pago. En el caso de que el cliente esté registrado puede hacer uso de cualquiera de los subsistemas de forma independiente. En el subsistema de registro se ha segmentado el tipo de registro tradicional (datos comunes) y geográfico (datos geográficos), ya que para almacenar una ubicación se requiere que un usuario esté previamente registrado. El subsistema de venta posee actividades de mantenimiento de un carro de compras, desde la búsqueda de un producto hasta la adición, modificación y eliminación del mismo en una orden de compra.

En el caso de que el stock de productos no satisfaga la necesidad del cliente, el subsistema de integración es capaz de identificar todo el catálogo de productos de la cooperativa y sumar toda la producción que tenga cada productor del listado de productos. Finalmente con una orden de compra preparada y confirmada, el subsistema de pago nos permite establecer una conexión con los servicios de PayPal, para entregarle los datos de la orden y proceder al pago de la misma, si ésta se confirma, el sistema actualiza el stock de los productos.

3.4.2. Subsistema de Registro

A continuación se presenta la clasificación de los casos de uso considerados en el sistema:

Administrar datos tradicionales:

- CU1 - Ingresar datos usuario.
- CU2 - Modificar datos usuario.

Identificar usuario:

- CU3 - Identificar usuario.
- CU4 - Iniciar sesión usuario.

Administrar datos geográficos:

Los siguientes casos de uso inician cuando el usuario tiene su sesión activa, es decir cuando el usuario ya se ha registrado y/o identificado.

- CU5 - Ingresar ubicación usuario.
- CU6 - Modificar ubicación usuario.

En la figura 3.6 se muestra los casos de uso correspondientes a este subsistema y el detalle de sus actividades se presenta en el anexo 1.

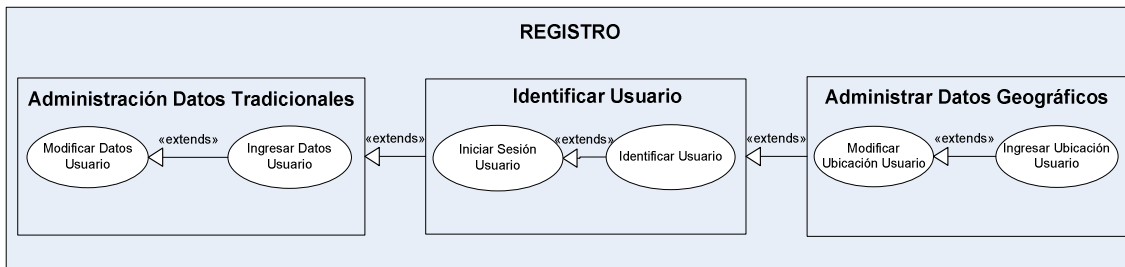


Figura 3.6 Casos de Uso del Subsistema de Registro

3.4.3. Subsistema de Ventas

A continuación se presenta la clasificación de los casos de uso considerados en el sistema:

Buscar Productos:

- CU7 - Identificar categoría de búsqueda (productores, frutas, hortalizas, organización, clientes).
- CU8 - Buscar geográficamente categoría.
- CU9 - Obtener productores y sus ofertas acorde a categoría.

Administrar canasta de compras:

- CU10 - Añadir o seleccionar producto a canasta de compra.
- CU11 - Guardar producto.
- CU12 - Retirar producto de la canasta de compra

En la figura 3.7 se muestra los casos de uso correspondientes a este subsistema y el detalle de sus actividades se presenta en el anexo 2.

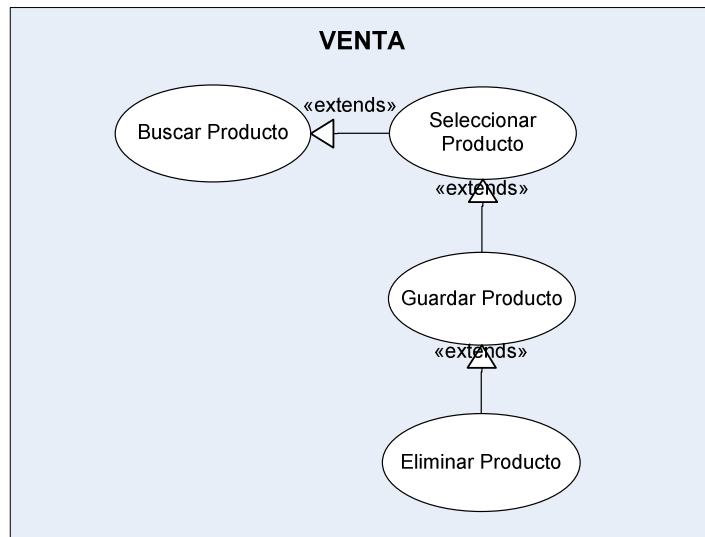


Figura 3.7 Casos de Uso del Subsistema de Venta

3.4.4. Subsistema de Integración

A continuación se presenta la clasificación de los casos de uso considerados en el sistema:

Identificar Catálogo:

- CU13 - Identificar catálogo de productos (frutas, hortalizas).

Integrar Producción:

- CU14 - Sumar la cantidad de stock de productos de todos los productores.

En la figura 3.8 se muestra los casos de uso correspondientes a este subsistema y el detalle de sus actividades se presenta en el anexo 3.

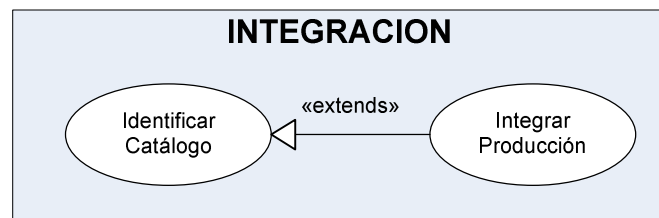


Figura 3.8 Casos de Uso del Subsistema de Integración

3.4.5. Subsistema de Pago

A continuación se presenta la clasificación de los casos de uso considerados en el sistema:

Pagar canasta:

- CU15 - Confirmar orden de venta.
- CU16 - Pagar orden de venta.
- CU17 - Controlar stock.
- CU18 - Cambiar estado de la orden de venta.

En la figura 3.9 se muestra los casos de uso correspondientes a este subsistema y el detalle de sus actividades se presenta en el anexo 4.

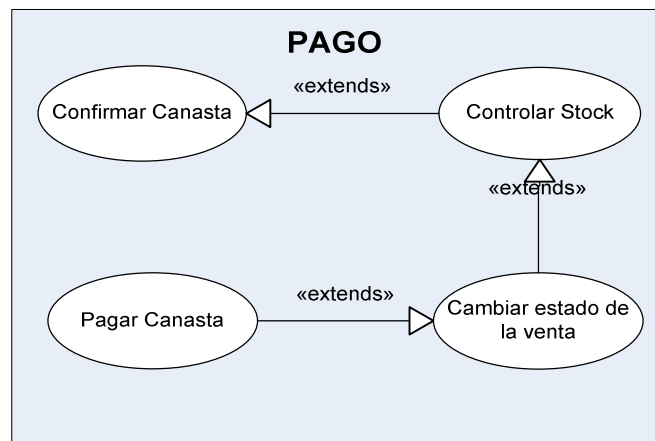


Figura 3.9 Casos de Uso del Subsistema de Pago

3.5. Restricciones

A continuación presentamos las restricciones acerca de las normativas, estándares y tecnología consideradas en el proceso de desarrollo y producto desarrollado.

Normativas:

- La ubicación geográfica de usuarios cliente, no es pública, es información que únicamente puede ser observada por la cooperativa.
- El stock de productos que considera el sistema es el resultado de la planificación semanal del productor. La cantidad de esta planificación es constante. Esta información fue obtenida de la cooperativa y es usada para publicar la estructura del catálogo de productos de SGPO.
- Está fuera del alcance del portal, la forma de cooperación física de productos entre los productores. Así como también la forma de entrega de la orden de compra al cliente. Esto forma parte de acuerdos y normativas de la cooperativa. SGPO entrega la información de la venta, es decir el identificador del cliente y productor(es) con los que está relacionada la venta.
- Sólo clientes con registro de cuenta podrán administrar la canasta de compras y efectuar su compra.

- El pago será permitido únicamente por medio de tarjeta de crédito, a través de PayPal. La comisión que se atribuye por pago es el 5% del valor de la transacción.

Estándares:

- *Lenguaje de modelado:* Todo tipo de documentación entre el autor del sistema y la cooperativa está basado en UML.
- *Aplicaciones geográficas:* La cartografía está manejada con OSM (OpenStreetMaps¹³) y Openlayers¹⁴.

Tecnología:

- El desarrollo del sistema ha sido realizado en la versión 1.6 de Java.
- El motor de BD usado es PostgreSQL y Postgis, por su soporte para datos geográficos.

3.6. Requisitos de Calidad

Una vez planteado el contexto de la aplicación, con el fin de brindar calidad y continuidad a la aplicación para su evolución y uso, se definieron los siguientes requisitos transversales:

Usabilidad:

- Facilidades de uso para la interacción del usuario con la cartografía.
- Facilidades de uso en la administración de la canasta de compras.

Confiabilidad:

- Las acciones de los subsistemas de venta y pago: guardar, confirmar, pagar canasta y controlar stock, son críticos para la cooperativa y poseen un alto control de tolerancia a fallos.
- El subsistema de pago se ejecuta bajo un protocolo de seguridad (HTTPS).

Funcionalidad:

- En el caso de registro de ubicación del usuario (CU3), se permite la interacción del usuario con cartografía para fijar su ubicación.
- Recuperación de la orden de compra de la última adquisición realizada por el cliente, con el fin de agilizar su próxima compra.

¹³ Servicios Web OpenStreetMap 2011, The Free Wiki World Map, último acceso 05 Septiembre 2011 <<http://www.openstreetmap.org/>>

¹⁴ Servicios Web OpenLayers 2011, Open Source Geospatial Foundation, último acceso 05 Septiembre 2011 <<http://openlayers.org/>>

4. Diseño de la Solución

En el presente capítulo se muestra el diseño de la arquitectura de SGPO, para lo cual se consideró la globalidad de las funcionalidades planteadas en el anterior capítulo. Así, se elaboró los siguientes esquemas de presentación: *ambiente operacional*, *arquitectónico*, *diseño detallado* y *modelo de clases*. El aporte de estos diseños ha sido soportar el procesamiento de información que requiere la aplicación.

4.1. Descripción General

El diseño parte con la definición de la frontera del sistema, así se estableció el ambiente operacional sobre el cual se desempeña SGPO. Posteriormente se procedió con el diseño de la arquitectura estructural y detallada del sistema.

El diseño de estos esquemas se fue mejorando de forma iterativa. Esta metodología permitió que los esquemas resultantes sean producto de una maduración paulatina. Cada esquema se ajustó a las características que requería el sistema.

Primero se diseñó los esquemas de la arquitectura estructural y de control. Posteriormente se efectuó descomposición modular y se trabajó en el diseño detallado de los subsistemas que conforman la aplicación.

Para el diseño detallado de la lógica de negocio de los subsistemas planteados se definieron prioridades. El diseño de aquellas funcionalidades críticas se fue definiendo en las primeras iteraciones. El detalle del nivel de criticidad se lo presenta en el *Anexo 6*.

Para fortalecer los atributos de calidad y cumplir con las metas de confiabilidad, usabilidad y performance del sistema, se ha hecho uso del Framework Spring que soporta tecnología J2EE¹⁵. Sus cualidades aportan con particularidades como: persistencia, estructura MVC (Modelo-Vista-Controlador) y el uso de varios controladores que receptan eventos para evitar cuellos de botella (ver Figura 4.3).

4.2. Ambiente Operacional

El contexto con el cual interactúa SGPO, se lo ha definido en torno a cinco segmentos (ver Figura 4.1). Cada uno de ellos, es un complemento para la aplicación, cuya comunicación está basada en la entrega y/o recepción de datos.

¹⁵ Servicios Web Java Sun 2011, Sun Microsystems, último acceso 05 Septiembre 2011, <<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/>>

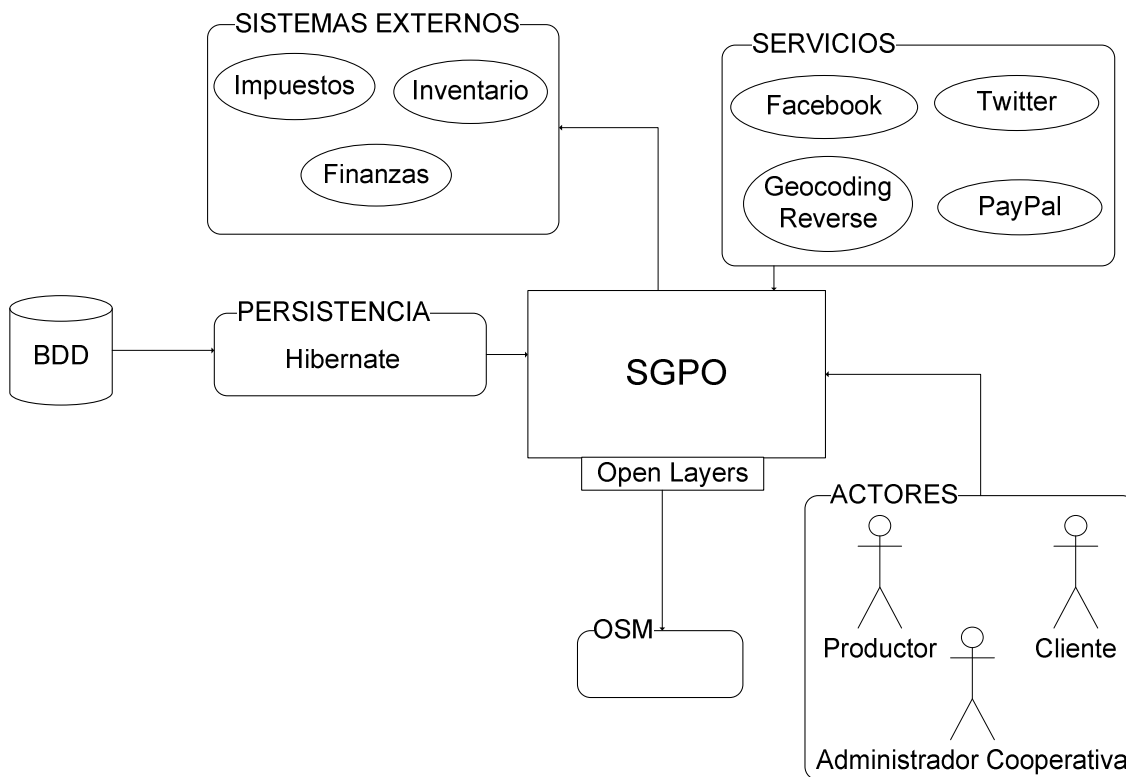


Figura 4.1 Ambiente Operacional de SGPO

SERVICIOS: Este segmento posee servicios Web y de comunicación que alimentan a SGPO, con datos que son usados en su capa de presentación o lógica de negocio. *Geocoding Reverse* es un servicio Web, que después de recibir coordenadas geográficas, entrega datos de nombres de calles, avenidas, sector, provincia, región, país.

Paypal posee un servicio Web, que receipta la orden de venta generada en SGPO. Receiptada la orden, permite el pago online con el uso de una cuenta PayPal. Finalmente nuestra aplicación recibe datos, indicando si el pago ha sido confirmado o cancelado. Por otro lado, se hace uso de las cualidades de las redes sociales (Facebook y Twitter) para fortalecer la comunicación y publicidad.

OSM: Open Street Maps, posee una considerable cartografía de calles y avenidas a nivel mundial, que es alimentada por comunidades voluntarias. Además posee una aplicación online, que permite ingresar rutas adicionales que un cibernauta posea.

La conexión de SGPO con los mapas de OSM, se la efectuó a través de una librería JavaScript, denominada Openlayers. Esta librería es la que ha permitido diseñar una interfaz de fácil usabilidad, en el sentido de la interacción usuario-cartografía.

ACTORES: Son los usuarios que por medio de un registro previo de sus datos, adquieren una cuenta de acceso. Por medio de esta cuenta, pueden solicitar y entregar información a la aplicación, para que ésta sea procesada.

SISTEMAS EXTERNOS: Inventario, finanzas e impuestos son módulos que pueden abarcar un amplio nivel de operación y son complementos importantes para la visión del sistema.

Estos módulos se los independiza para que en una próxima etapa sean profundizados. Al momento estas aplicaciones se representan en clases, que instanciadas, son objetos que proporcionan la información necesaria a SGPO.

PERSISTENCIA: Es una capa que preserva los objetos generados por la aplicación. A través de una configuración de mapeo entre las clases y las tablas de la base de datos, los valores de los objetos son guardados en el almacén de datos. Las cualidades que se obtiene con esta configuración son la portabilidad y la seguridad.

4.3. Diseño Arquitectónico

Luego de haber analizado el entorno operacional de SGPO, se procedió a diseñar la arquitectura interna de la aplicación. Para esto se consideró dos entornos: por un lado el diseño estructural y por otro el diseño de control. Para el primer diseño se ha considerado un modelo por capas y para el segundo, un modelo de manejador de eventos.

4.3.1. Diseño Estructural

La estructura de SGPO fue definida en las tres capas (ver figura 4.2) siguientes: presentación, lógica de negocio y persistencia. Estas capas apoyan el flujo de información, desde la interfaz de usuario hasta el almacén de datos.

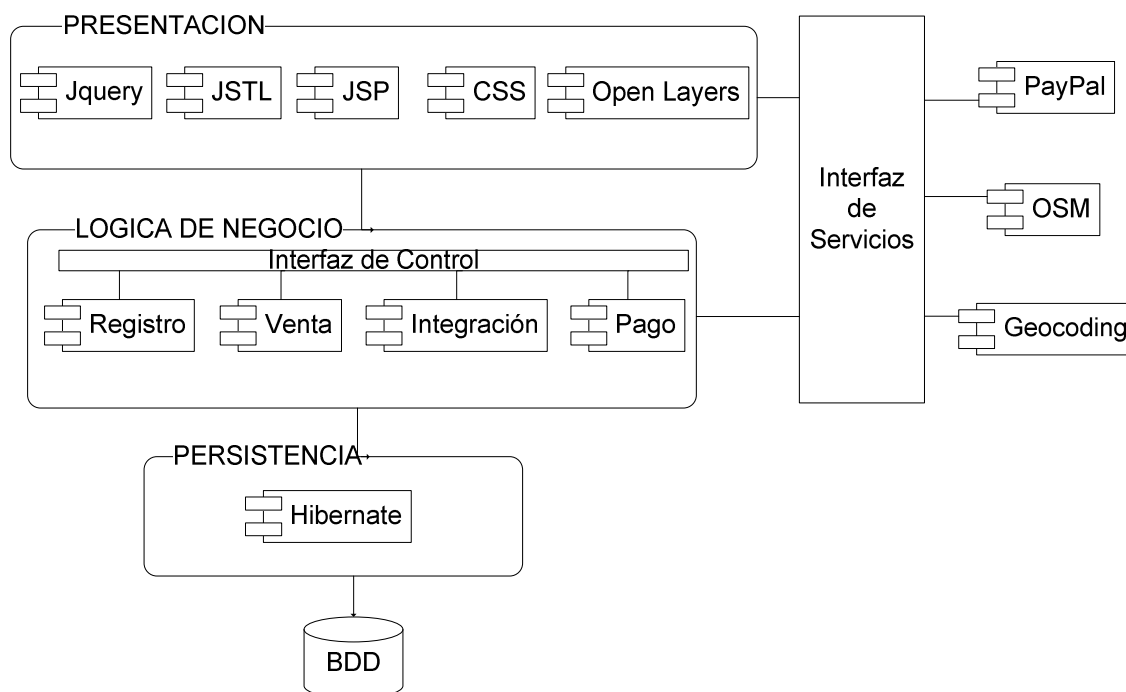


Figura 4.2 Diseño estructural de SGPO

Las funcionalidades del sistema están almacenadas en la capa de lógica de negocio. Éstas fueron distribuidas en los subsistemas de registro, venta, pago e integración. También, se dispone de una interfaz de servicios, que posee funcionalidades de enlace con los siguientes servicios Web: OSM (OpenStreetsMaps), Geocoding y PayPal.

El flujo de información a través de la estructura, inicia cuando el usuario gatilla una funcionalidad deseada por medio de la capa de presentación. Si la petición lo amerita la aplicación, utiliza una interfaz de control para conectarse a la lógica de negocio y acceder al subsistema que haya solicitado. Caso contrario, la petición puede ser conducida hacia una interfaz de servicios para conectarse con los servicios Web antes mencionados.

En la capa de lógica de negocio, la interfaz de control dispone de varios controladores para recibir y conducir las peticiones del usuario. De acuerdo a la acción gatillada, se procesan los datos y se instancian los objetos necesarios. Procesados los datos y generados los objetos, éstos se preservan en la capa de persistencia, para finalmente sincronizar con el modelo de datos.

a) Capa de Presentación

Esta capa posee librerías, hojas de estilo y tags necesarios para presentar el diseño de la interfaz de usuario. Éstos han sido los componentes para brindarle al usuario una mejor usabilidad de la aplicación. A continuación se detalla la función de los componentes empleados:

OPENLAYERS: Es una librería Java Script, encargada de la conexión con OSM (OpenStreetMaps). Su uso permitió que el usuario pueda avizorar en nuestra aplicación la cartografía de OSM. También posee eventos que permite al usuario interactuar con la cartografía.

JQUERY: Son librerías de diseño de presentación, que han mejorado la usabilidad de la aplicación. También, brindan un mejor control de los eventos a los componentes usados en la interfaz de usuario.

JSTL: Es una biblioteca de etiquetas que se usa en la capa de presentación. Nos ha permitido tener un control limitado de aquellos objetos instanciados en memoria por la lógica de negocio.

JSP: Es la tecnología que permite el uso de lenguaje Java en código HTML. Con esta cualidad se obtiene una interfaz de presentación con información dinámica.

CSS: Representan hojas de estilo que almacenan propiedades de diseño. En este caso almacena la estructura de las páginas HTML de nuestra la aplicación.

b) Capa de Lógica de Negocio

Esta capa es de relevante importancia ya que contiene el modelo de control de toda la estructura establecida. Este modelo es el encargado de conducir el flujo de trayectoria de la información, en su contexto interno y externo (enlace con las otras capas).

El modelo de control está basado en el Framework Spring que soporta tecnología J2EE. Se lo ha configurado para que posea varios controladores y funcione como un manejador de eventos.

Estos eventos son gatillados por el usuario a través de la capa de presentación. Los controladores establecidos receptan la petición y la distribuyen al subsistema de la capa de negocio que corresponda.

SUBSISTEMA DE REGISTRO: Este subsistema es el encargado de recibir datos personales y de ubicación del usuario. Los datos son entregados a través de la capa de presentación. Esta acción es receptada a través de sus controladores en la capa de lógica de negocio. Posteriormente se procesan los datos y se instancian los objetos necesarios. Finalmente la capa de persistencia preserva y almacena la información.

SUBSISTEMA DE VENTA: El desempeño de este subsistema gira en torno a la generación y administración de una orden de venta. Esta orden está relacionada con un usuario. Para lo cual es indispensable tener su sesión activa. El flujo de información (entre capas) de este subsistema tiene la misma trayectoria que el módulo anterior.

SUSBSISTEMA DE PAGO: Este subsistema permite comunicarse con los servicios Web de PayPal para que el usuario pueda hacer el pago online de su orden de compra. La trayectoria del flujo de información es canalizada por la interfaz de servicios. SGPO

entrega los datos de la orden de venta y recibe la confirmación del pago realizado (a través de una cuenta PayPal).

SUBSISTEMA DE INTEGRACION: Este subsistema permite acumular la producción individual de cada productor. Así, se obtiene un catálogo de productos con la cantidad de producción que puede generar todos los integrantes de la cooperativa.

c) Capa de Persistencia

En esta capa se define el enlace entre las clases de la capa de lógica de negocio y las tablas del modelo de datos. La información almacenada en objetos (instancias de clases) es sincronizada y actualizada con el modelo de nuestra base de datos.

Nuestra tecnología de persistencia empleada es Hibernate. Ésta usa sentencias de lenguaje HQL, para realizar las tareas anteriormente mencionadas. Esta capa proporciona confiabilidad a la base de datos, ya que las sentencias usadas para la extracción de información, trabajan únicamente con objetos que se encuentran en memoria.

La configuración de esta capa se la efectuó en un archivo XML, denominado hibernate.cfg.xml. Aquí se definió la conexión a la base de datos, el puerto de enlace y nombre de la instancia. Pero lo más importante es la lista de mapeo, que contiene los nombres de las clases y tablas que mantienen relación entre sí.

4.3.2. Modelo de Control

Como parte del diseño arquitectónico de la aplicación se implanta el diseño de un modelo de control (ver figura 4.3). Éste define la trayectoria de la información entre las capas de la estructura establecida (contexto externo) y los subsistemas de la lógica de negocio (contexto interno).

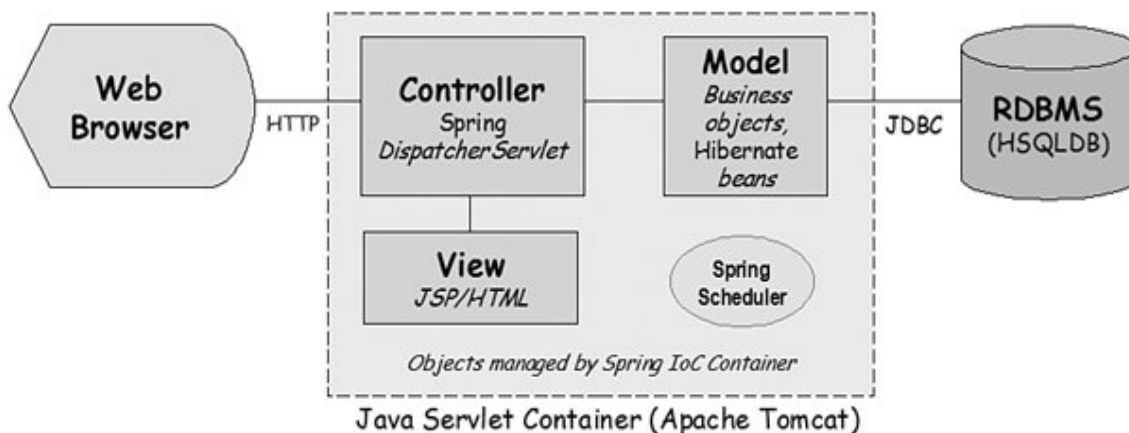


Figura 4.3 Arquitectura del Modelo de Control¹⁶

¹⁶ Hemrajani Anil, Agile Java™ Development With Spring 2006, Free-Form Architecture Diagram, May 2006.

La estructura del modelo de control ha sido llevada a cabo haciendo uso del patrón de diseño MVC (Modelo-Vista-Controlador). Este patrón permite el flujo de las peticiones del usuario desde la capa de presentación hasta el almacenamiento de datos o viceversa.

Internamente en la capa de lógica de negocio, el modelo de control trabaja bajo el esquema *Manejador de Eventos*, pues de acuerdo al evento que el usuario gatille, existen controladores configurados para recibir esta petición. La misma que es procesada y dirigida al almacén de datos para extraer o guardar información. Finalmente el resultado del proceso es devuelto a una vista, que puede ser un archivo HTML o JSP.

Cada subsistema establecido en la lógica de negocio tiene uno o más controladores. La configuración de los controladores se la efectúa en el archivo `mercadorganico-servlet.xml`, ubicado en la capa de negocio.

4.4. Diseño Detallado

Una vez desarrolladas las funcionalidades del sistema y la estructura del ambiente de trabajo, llegó el momento de diseñar el detalle de cada uno de los subsistemas que se habían definido. Para lo cual se utilizó la misma metodología de trabajo del diseño arquitectónico; es decir, el diseño detallado se fue afinando en cada nueva iteración.

Primero se definió un modelo de dominio (ver figura 4.4) con el fin de tener un soporte para percibir la globalidad del sistema. Así consideramos las clases necesarias para crear una lógica de negocio que ejecute las funcionalidades que se nos hemos propuesto.

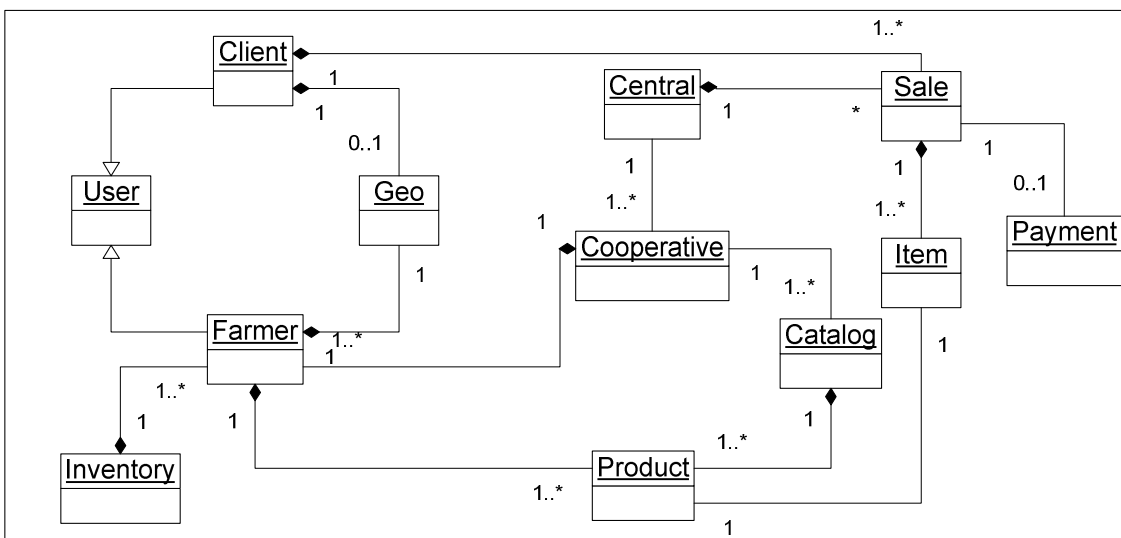


Figura 4.4 Modelo de Dominio

Posteriormente, se considera los subsistemas establecidos en la capa de lógica de negocio (ver Figura 4.5) y por cada uno de ellos se elaboró una descomposición modular. En este nivel de granularidad se definen las clases, atributos y métodos necesarios para soportar dichos subsistemas. Las clases definidas fueron la base inicial para la implementación del sistema.

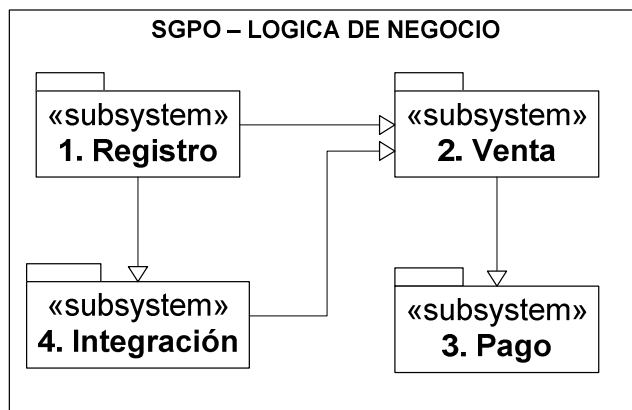


Figura 4.5 Subsistemas de la Lógica de Negocio

El orden de presentación de los subsistema se lo ha hecho considerando su calificación de prioridad (ver Anexo 6). Producto del diseño detallado, se logró obtener un diagrama de clases final para SGPO, éste es presentado en la sección 4.4.5.

4.4.1. Subsistema de Registro

Considerado un punto crítico debido a la importancia que tiene el registro de la ubicación y dirección del usuario-cliente, su diseño ha sido trabajado con el fin de ofrecer la mejor usabilidad posible.

Para el levantamiento de su estructura ha sido importante investigar las necesidades técnicas que requiere el subsistema. El aspecto técnico fue muy importante para desarrollar estas funcionalidades.

Para obtener una ubicación geográfica ha sido necesario el uso de la cartografía del Sistema OSM (Open Streets Maps) y los servicios de Geocoding de Nominatim. Las cualidades de ambas alimentan nuestra aplicación a través de la librería OpenLayers. Establecida esta conexión, se intercambian datos geográficos que son conducidos por el flujo del subsistema de acuerdo a la actividad del usuario.

El subsistema recibe las peticiones que el usuario haya realizado en la capa de presentación, a través de dos controladores: RegisterController y SigInController (ver Figura 4.6). Dependiendo del controlador elegido, seis funciones llevarán a cabo la ejecución de la tarea solicitada.

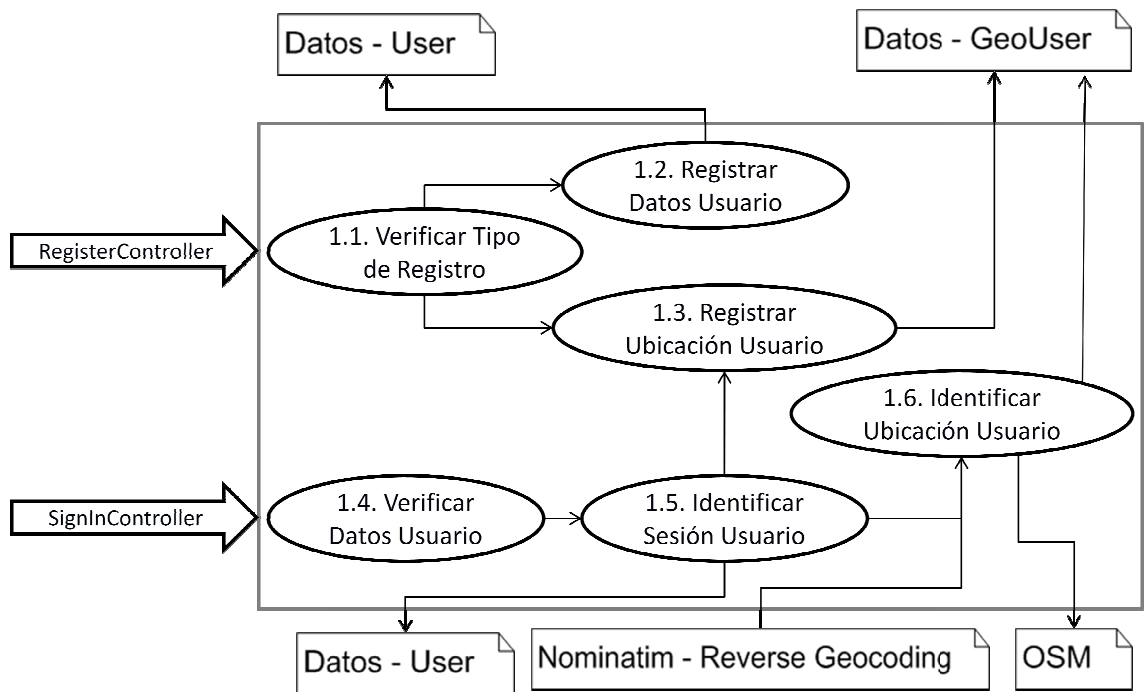


Figura 4.6 Controladores y Funciones del Subsistema de Registro

Si la petición del usuario es de registro, ésta ingresa por RegisterController. Aquí se verifica si el registro es de datos tradicionales o geográficos. Si son tradicionales ingresa por la función *Registrar Datos Usuario*, identifica una sesión para el usuario registrado y brinda disponibilidad para registrar su ubicación.

En caso de que la acción sea de Login, ésta ingresa por SignInController. La función inicial a ejecutarse es *Verificar Datos Usuario*. Posteriormente identifica la sesión del usuario para que éste pueda empezar a preparar su orden de compra.

Establecidas las funciones necesarias, se definió un modelo de clases que soporte el procesamiento de datos (ver figura 4.7). Dos son los tipos de usuarios que el sistema soporta, éstos son Client y Farmer, ambos son una herencia de la clase User. En cuanto a la clase Geo, ésta almacena el dato de posición geográfica del usuario y es una dependencia para Client y Farmer.

Los atributos establecidos son únicamente la información más relevante del usuario, esto con el fin de obtener un rápido registro. Estas clases son las que se instancian para generar objetos que se preservan en la capa de persistencia. Cuando se recibe la petición de almacenamiento, estos objetos sincronizan con el almacén de datos.

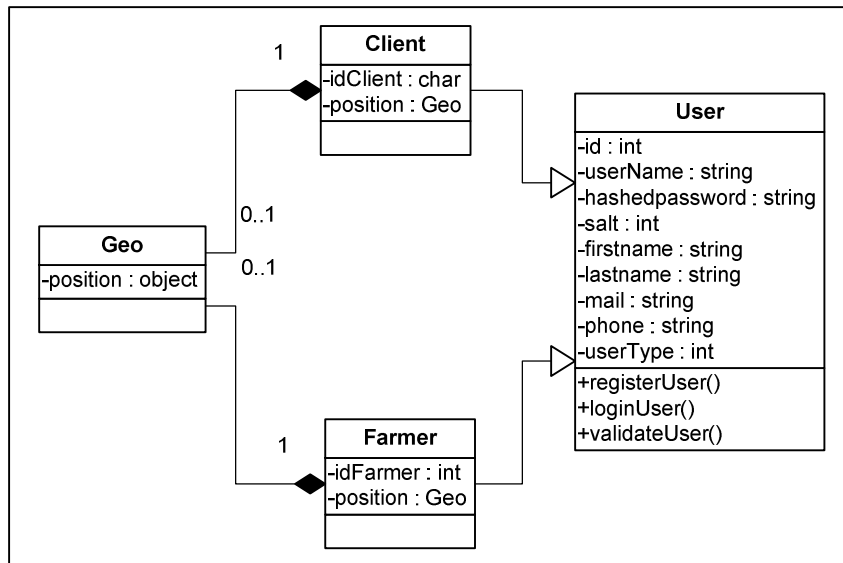


Figura 4.7 Modelo de Clases del Subsistema de Registro

4.4.2. Subsistema de Venta

Como segundo orden se consideró el diseño del subsistema de venta. La importancia de éste radica en la estructuración de un esquema que soporte el mantenimiento de una orden de venta a través de una canasta de compras.

Su diseño se ha centrado en ofrecer al usuario consistencia en sus datos, así como una buena usabilidad a nivel de interfaz. La consistencia que SGPO ofrece en su diseño de lógica de negocio, es que la orden de compra que el usuario prepara, va siendo actualizada constantemente con la base de datos.

Este escenario aporta robustez, en caso de que el usuario por alguna eventualidad pierda conexión con la aplicación. Si esto llegara a suceder la orden es recuperada un su siguiente login.

En cuanto a la usabilidad, en la capa de presentación se hace uso de técnicas y bibliotecas como JQuery y Ajax¹⁷. Herramientas que aportan a que el usuario pueda manipular y administrar su canasta de compras de una manera ágil.

Este subsistema recibe las peticiones del usuario a través de cinco controladores: SearchListController, AddItemController, UpdateItemController, DeleteItemController y ConfirmCartController. Dependiendo del controlador solicitado diez funciones ejecutan el procesamiento de datos, para la generación de los objetos necesarios (ver Figura 4.8).

¹⁷ Servicios Web JQuery 2010, The jQuery Project, último acceso 06 Septiembre 2011 <<http://api.jquery.com/category/ajax/>>.

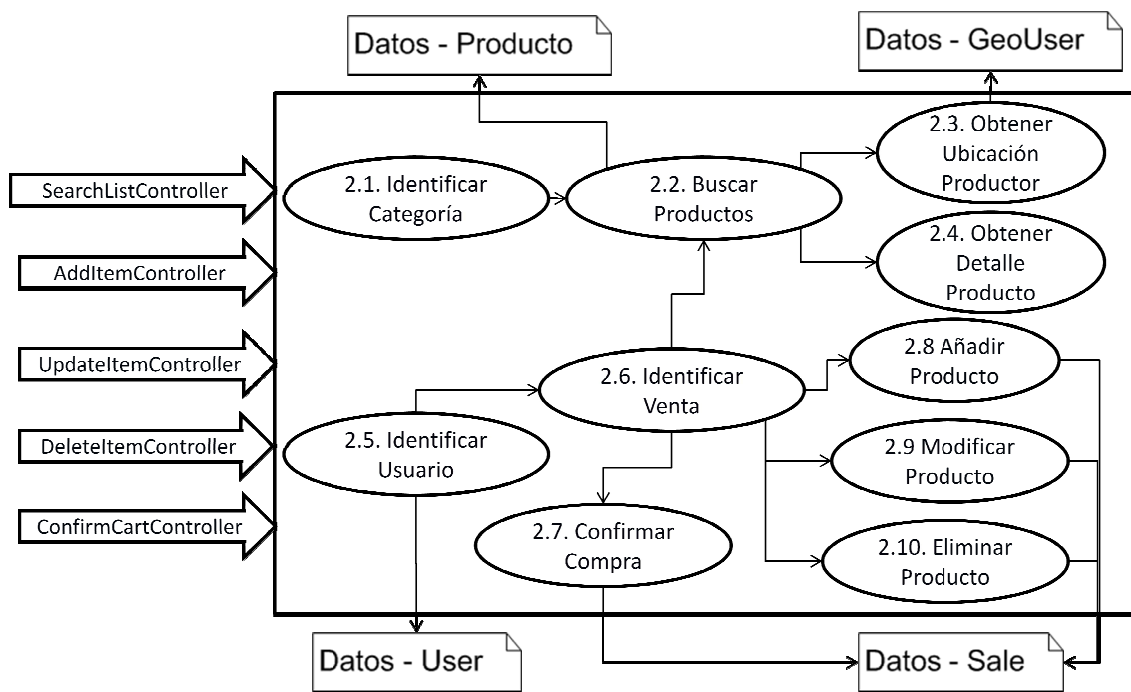


Figura 4.8 Controladores y Funciones del Subsistema de Venta

Todos los controladores exceptuando SearchListController inician su flujo con una sesión de usuario activa y válida. Esto es necesario para que la orden de compra que se genere sea anexada a un usuario.

Si la petición es de búsqueda, ésta ingresa por SearchListController. En este caso no es necesaria una sesión activa, ya que listar o buscar los productos disponibles, es algo que puede realizar cualquier navegante. La búsqueda inicia con la función *Identificar Categoría*, ésta determina si se desea buscar un productor, fruta y/o hortaliza.

Posteriormente se dirige a *Buscar Productos*, esta función obtiene los datos resultantes de la búsqueda, que generalmente son los datos de ubicación del productor y el detalle de sus productos. Si el usuario no tiene activa su sesión, únicamente podrá ver el detalle del producto o apreciar la ubicación del productor.

Si el usuario desea administrar (ingreso, modificación o retiro de productos) su canasta de compras debe tener su sesión activa. Es decir, inicialmente el usuario debió haber ejecutado la función *Identificar Usuario*. Posteriormente identifica la venta, en caso de que el usuario ya tenga una abierta, y la aplicación recupera los datos de esta orden de compra.

Dependiendo del tipo de petición del usuario, ésta ingresa por los controladores: AddItemController, UpdateItemController o DeleteItemController. De acuerdo a la petición receptada, el sistema ejecuta la alteración solicitada en la orden de venta.

Finalmente si el usuario decide confirmar la orden de compra, esta acción es recibida por ConfirmCartController. Su flujo inicia con el uso de las funciones *Identificar Usuario*

e *Identificar Venta*. Una vez que obtiene la venta del usuario implicado, la función confirma la compra y cambia la orden de venta a un estado confirmado.

Con estas funciones establecidas, se procedió al diseño del modelo de clases del subsistema de venta (ver Figura 4.9). Para este modelo se hizo uso de una clase lógica denominada Central.

Se la denominó lógica ya que únicamente es usada en la capa de lógica de negocio para mantener el enlace entre las clases restantes. Es decir Central no será configurada en la capa de persistencia para sincronizar su información con el almacén de datos.

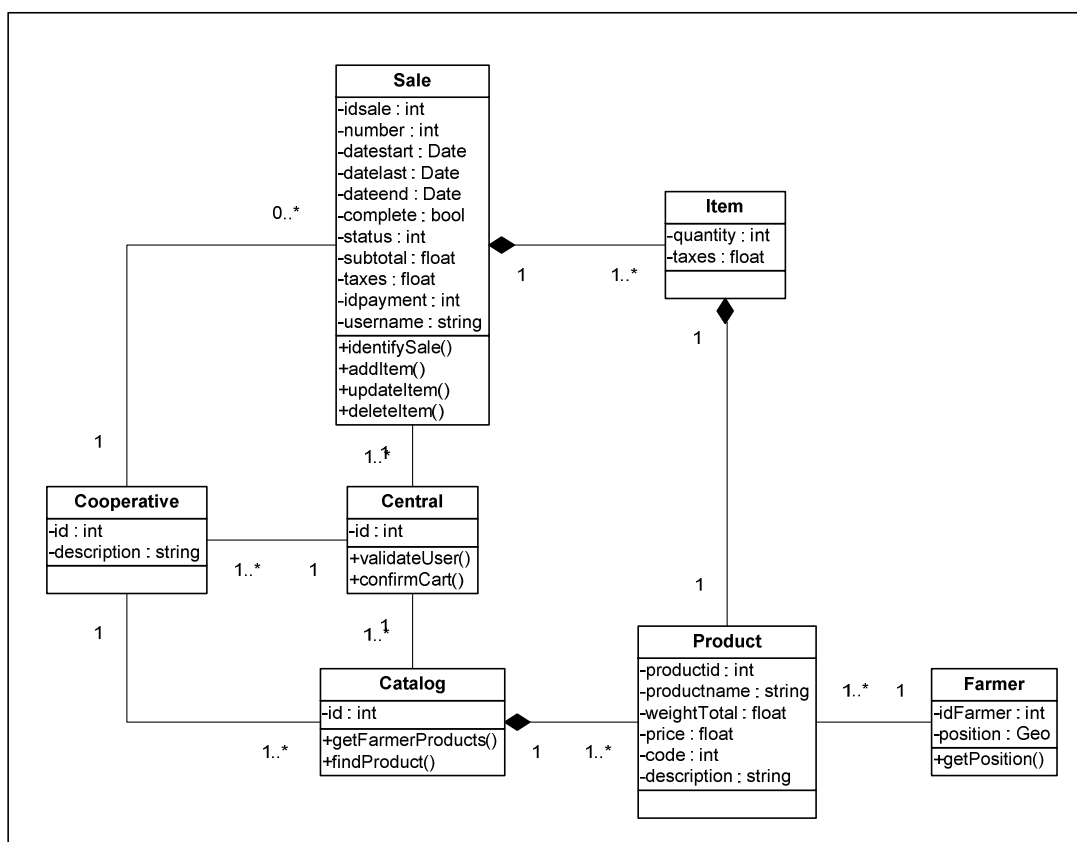


Figura 4.9 Modelo de Clases del Subsistema de Venta

A la clase Central se la considera como un control interno del subsistema de venta. Todas las funciones que implican el mantenimiento de la orden de venta parten de la instancia de esta clase.

La clase Cooperative posee la información de la cooperativa a la que se le efectúa la compra. Actualmente se trabaja con una, pero se consideró la instancia, para que la aplicación soporte otras cooperativas. También el modelo permite que la cooperativa tenga uno o varios catálogos de productos, según su planificación y forma de presentarlos.

Cada objeto generado con la clase Sale está ligado a una cooperativa. Esta clase posee atributos de la orden de venta y adicional contiene una composición de uno o varios Ítems, que son los productos que el usuario elije.

A la clase Sale la aplicación añade, modifica o elimina ítems de la canasta de compras. Sale tiene un atributo importante que es su Status, este dato muestra las condiciones en que se encuentra la orden de venta y para esto se ha definido los estados presentados en la Tabla 4.1.

Tabla 4.1 Estados de la Orden de Venta

Estado de Orden de Venta	Descripción
1	Creada
2	Confirmada
3	Pagada

Las clases mencionadas son las que se instancian en este subsistema y los objetos creados se preservan en la capa de persistencia. Exceptuando la clase lógica Central todas las demás sincronizan con el almacén de datos, cuando se solicita esta acción.

4.4.3. Subsistema de Pago

Estructurados los subsistemas de registro y venta, se procede a diseñar el flujo de datos para el pago de la orden de compra que el usuario ha preparado. Para su diseño ha sido importante considerar el factor de confiabilidad, ya que la acción implica una transacción económica.

Para su solución técnica se ha investigado sobre la vinculación de nuestra aplicación con PayPal. Ésta ofrece servicios de pagos online a través de una tarjeta de crédito asociada a una cuenta PayPal.

Por medio de un Webservice, PayPal puede recibir el valor de la orden de venta de nuestra aplicación. Si este valor es pagado por una cuenta PayPal del usuario, nuestra aplicación recibe una confirmación de que el pago ha sido realizado.

La reputación de este servicio es altamente calificado a nivel mundial. Su confiabilidad radica: en el manejo de los datos de transacción con un protocolo de seguridad (HTTPS) y en las certificaciones que lo respaldan.

El Subsistema recibe la petición de pago del usuario a través del controlador: PaymentController (ver Figura 4.10). Seis son las actividades necesarias, para la ejecución de la tarea solicitada.

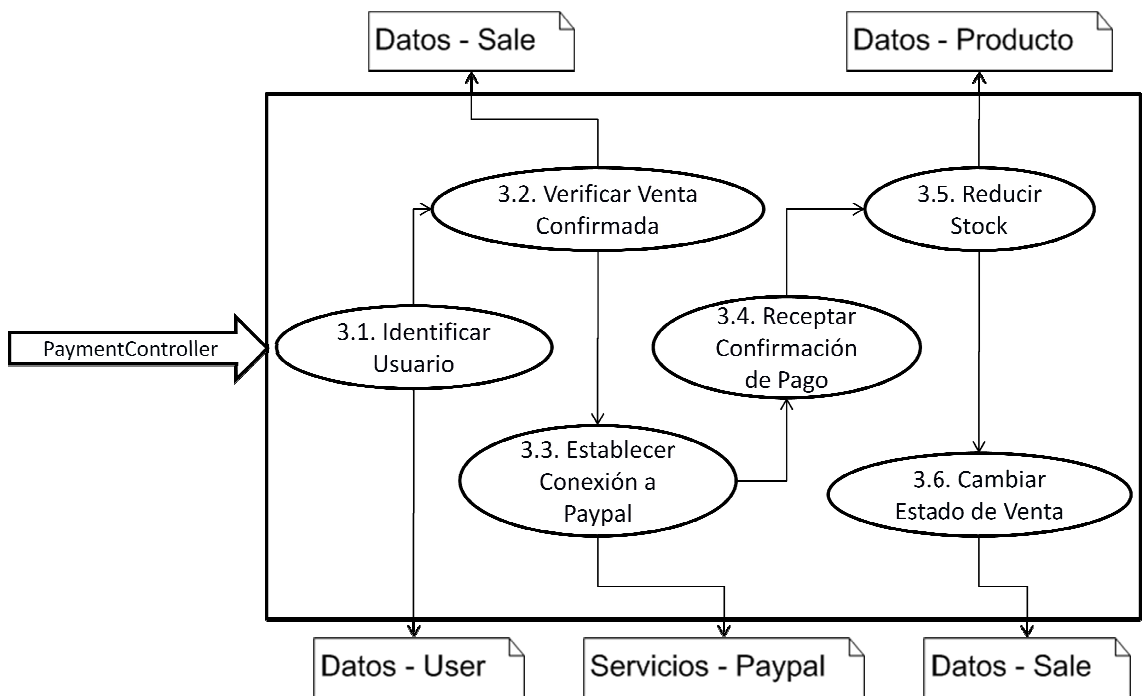


Figura 4.10 Controladores y Funciones del Subsistema de Pago

Al ser gatillada la petición de pago por el usuario. Lo primero que se hace es identificar y validar la sesión activa del usuario. Posteriormente, con la función *Verificar Venta Confirmada*, se valida que la orden de compra del usuario se encuentre en estado confirmada.

De ser así, la aplicación procede a establecer una conexión con los servicios Web de PayPal. Se entrega los datos de la orden de venta y se espera por la confirmación del Pago, cuando los servicios PayPal envían la confirmación de pago de la transacción, dos son las funciones que finalizan el proceso de venta: *Cambiar Estado de Venta* y *Reducir Stock*.

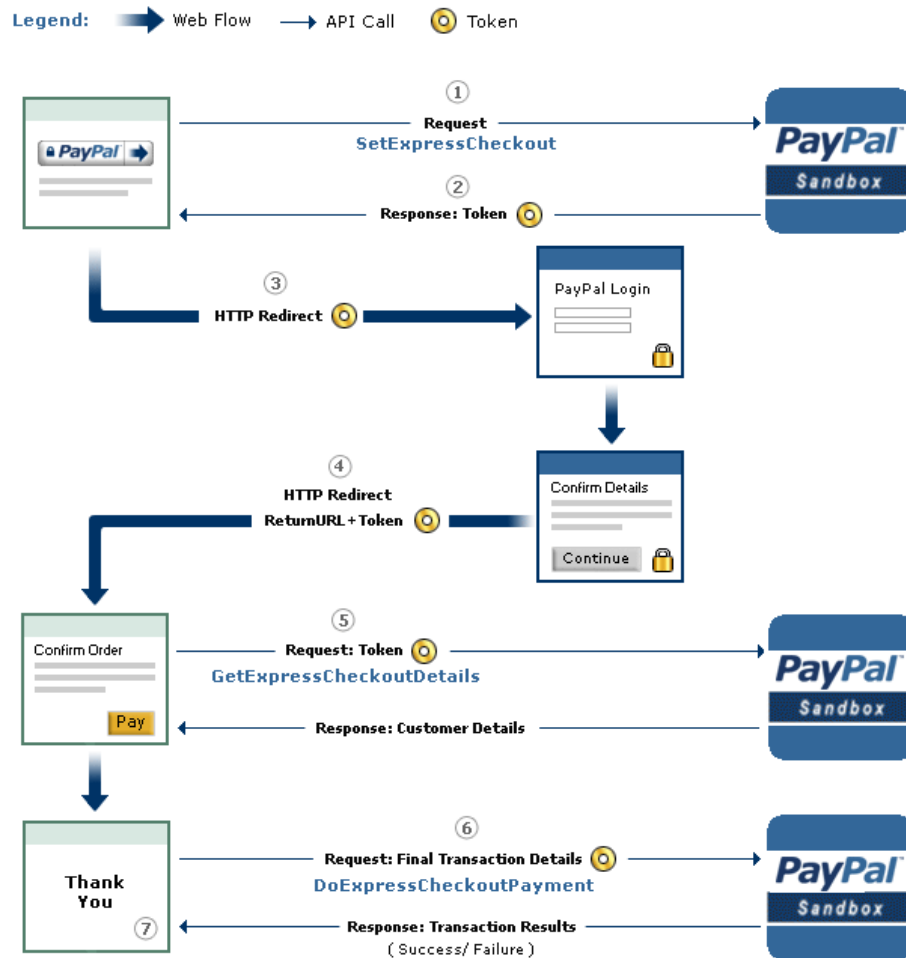


Figura 4.11 Flujo de Proceso de Pago con PayPal¹⁸

En la figura 4.11 se presenta el detalle del flujo de la realización de pago en PayPal, desde la entrega de los datos de la orden hasta la recepción de la confirmación de pago. En el paso uno se envía la petición de conexión con los servicios de PayPal y en el paso dos se recibe la respuesta. Si el enlace es satisfactorio se procede al paso tres, en el cual el usuario puede ingresar sus datos de cuenta y editar detalles de su forma de pago. En el paso cuatro se recibe estos datos de cuenta para que el usuario confirme la realización del pago. El paso cinco envía la confirmación o cancelación de la compra, mientras que en el paso seis se recibe en SGPO esta selección del usuario. Finalmente con el resultado de la transacción se direcciona a la interfaz pertinente.

¹⁸ Servicios Web PayPal 2011, Getting Started With Express Checkout, último acceso 18 Julio 2011 <https://cms.paypal.com/us/cgi-bin/?cmd=_render-content&content_ID=developer/e_howto_api_ECSimpleIntegration >

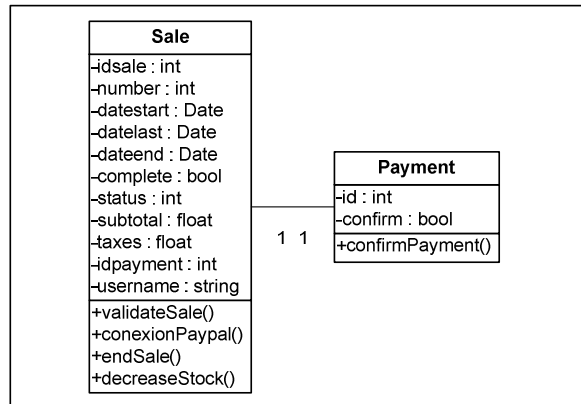


Figura 4.12 Modelo de Clases del Subsistema de Pago

Definidas las funciones del proceso de pago, se procede al diseño del modelo de clases que soporta este flujo de datos (ver Figura 4.12). La clase **Sale**, es la que posee todos los atributos necesarios para que el servicio de PayPal reciba la orden de venta y proceda a ejecutar la transacción.

Importante es el atributo **Status** de **Sale**, aquí se muestra el estado final de la orden de venta. Si del servicio PayPal se confirma el pago de la transacción, la instancia **Payment** adquiere un valor booleano de verdadero que corrobora la finalización de la venta. Las clases presentadas son las que se instancian y los objetos creados se preservan en la capa de persistencia, para finalmente sincronizar con el almacén de datos.

4.4.4. Subsistema de Integración

Finalmente se define el diseño del subsistema que permite obtener la suma total de la cantidad de producción individual de cada productor. El valor de esta acumulación representa la oferta que puede hacer la cooperativa.

Para el levantamiento de este proceso fue necesario conocer y definir la cantidad de producción semanal de cada productor. Cada uno de sus productos tiene una cantidad planificada semanalmente, y este valor representa el stock por productor en la aplicación. Terminada la semana, este stock de productos vuelve a tomar su valor inicial.

No es un subsistema crítico, su servicio es de valor añadido. Entrega información complementaria, como el límite máximo que la cooperativa puede ofrecer en su catálogo de productos. El subsistema recibe la petición de integración a través del controlador **IntegrateController** (ver Figura 4.13), y luego tres son las funciones que atienden esta petición.

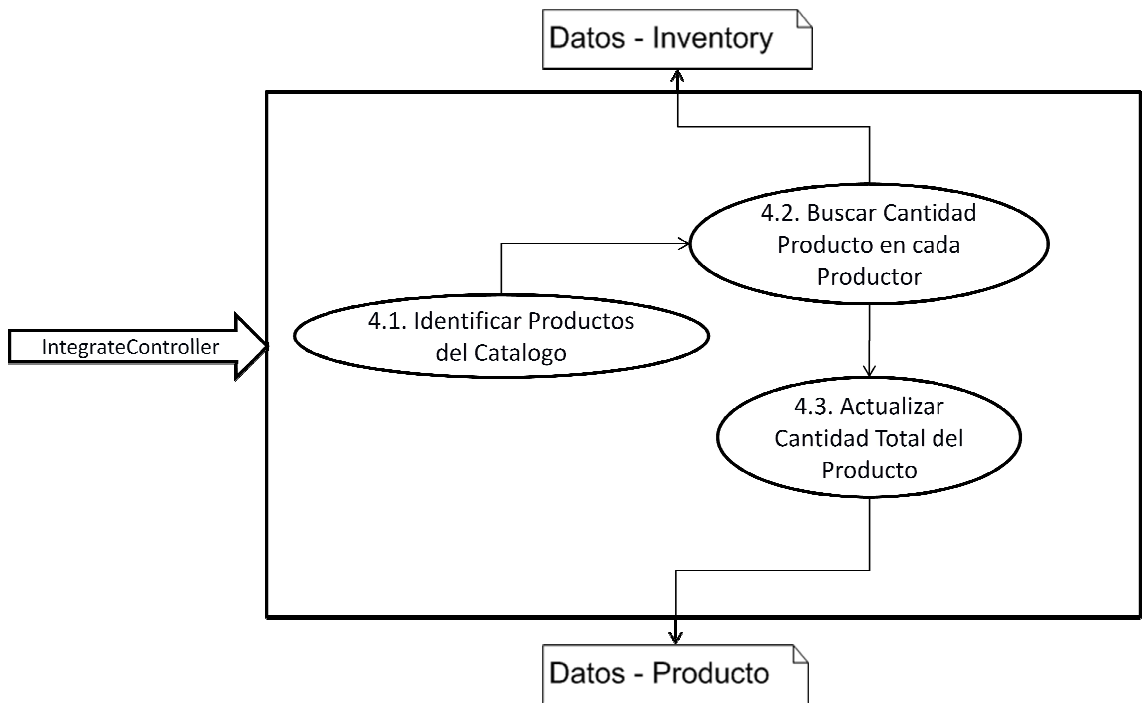


Figura 4.13 Controladores y Funciones del Subsistema de Integración

Inicialmente se identifica el catálogo de productos que posee la cooperativa. Validados los productos del catálogo, se procede a buscar la cantidad de producción en cada productor, valor que se va almacenando en un acumulador.

Después de obtener el total, este valor es actualizado en la clase Product. Esta instancia es la que almacena el dato de producción total en uno de sus atributos. Este subsistema no requiere la validación de una sesión activa para su uso.

Definidas las actividades necesarias en el subsistema, se procedió a estructurar su modelo de clases (ver Figura 4.14). Las clases Farmer e Inventory han sido necesarias, para almacenar los atributos de identificación del productor con su listado de productos, precios y cantidades.

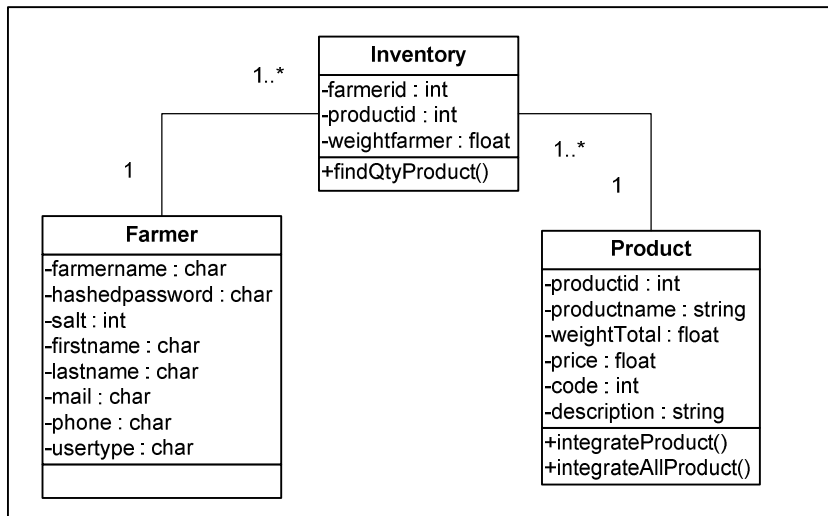


Figura 4.14 Modelo de Clases del Subsistema de Integración

La relación de las clases Farmer-Inventory ha sido determinada con la lógica de que cada Farmer posee un listado de productos. Esta relación permite almacenar el stock de la producción que tiene cada productor con determinado producto.

Mientras que en la clase Product se almacena el total de producción acumulado por todos los Farmer. Al igual que los subsistemas anteriores, estas clases son las que se instancian y los objetos generados se preservan en la capa de persistencia, para sincronizar con el almacén de datos.

4.4.5. Modelo de Clases

Recapitulando la globalidad del sistema y considerando los modelos de clases de cada uno de los subsistemas de la capa lógica de negocio. Se unifica los segmentos en un solo esquema, el mismo que representa el modelo de clases de SGPO (ver Figura 4.15).

Tres clases son externas a SGPO, pues en el transcurso se las puede considerar como aplicaciones a profundizar. Actualmente son clases que aportan a la funcionalidad del sistema.

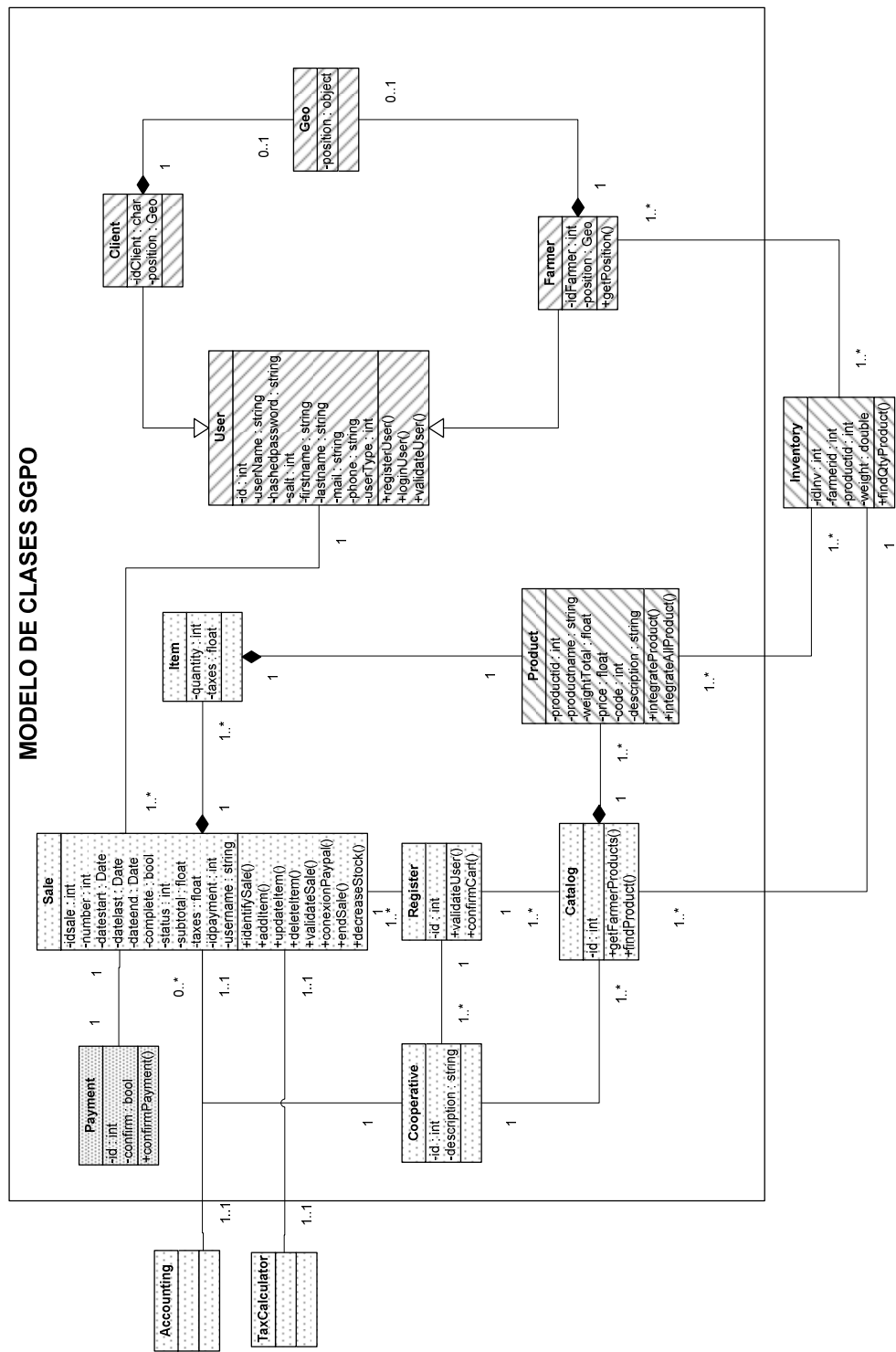


Figura 4.15 Modelo de Clases del Sistema

Las clases sombreadas con líneas oblicuas hacia la derecha, soportan la data del subsistema de registro. De la clase base User se derivan Client y Farmer, ambas clases poseen como dependencia Geo que es la clase que almacena la posición geográfica.

Las clases sombreadas con puntos separados, soportan la data del subsistema de venta. Central es una clase medular, que relaciona a todas las entidades para el procesamiento de las peticiones dirigidas a la orden de venta. Es decir controla las solicitudes de ingreso, modificación, retiro de ítems o cambios de estado en la orden de venta.

La clase sombreada con mayor densidad de puntos, soporta la data del subsistema de pago. La clase Payment almacena el código enviado por PayPal de aquellas órdenes de pago que han sido ejecutadas y confirmadas.

Las clases sombreadas con líneas oblicuas hacia la izquierda, soportan la data del subsistema de integración. La clase Farmer establece relación con Inventory y ésta a su vez con Product. Así, se pueden generar instancias que vinculen diferentes Farmer con diferentes productos. La cantidad de producción individual del Farmer se almacena en Inventory y la acumulada en Product.

Haciendo referencia a las variables y métodos de las clases que conforman el modelo establecido, a continuación describimos sus responsabilidades.

Geo.- Posee los datos de posición geográfica de clientes y productores.

Client.- Posee la identificación del cliente y su posición geográfica, obtenida por la relación de composición con la clase Geo.

Farmer.- Posee la identificación del productor y su posición geográfica, obtenida por la relación de composición con la clase Geo.

User.- Es la clase base de la herencia que existe con las clases Client y Farmer, almacena datos como: el nombre del usuario (userName), password encriptado (hashedPassword), código complemento para seguridad del password generado por el algoritmo de encriptación (salt), nombre del usuario (firstname), apellido del usuario (lastname), mail de contacto (mail), teléfono de contacto (phone), tipo de usuario (userType) campo necesario para identificar la clase derivada del usuario ya sea ésta cliente o productor.

Adicionalmente esta clase posee las responsabilidades de: registrar los datos de un usuario nuevo (registerUser), validar los datos que haya ingresado el usuario para ingresar al sistema (validateUser), iniciar la sesión del usuario (loginUser).

Sale.- Sus variables poseen datos de: identificación de la orden de venta (idSale), número de compra en la cooperativa (number), fecha en que el usuario inicio la orden de compra (datestart), fecha de su última actualización (datelast), fecha de finalización de la orden de venta (dateend), el estado de finalización de la orden (complete), el

estado de la orden de venta ya sea creada-confirmada-pagada (status), la suma total del precio de los productos que tiene la orden (subtotal), el valor de los impuestos por el subtotal de la compra (taxes), la identificación del pago de la orden de venta si ésta ha sido realizada (idpayment), el usuario que ha efectuado la compra (username).

Adicionalmente esta clase tiene las responsabilidades de: identificar la orden de venta cada vez que el usuario inicia su sesión (identifySale), agregar un nuevo producto a la orden (addItem), modificar un producto de la orden (updateItem), eliminar un producto de la orden (deleteItem), calcular el subtotal de los productos elegidos en la orden de venta con sus respectivos impuestos (endSale), reducir el stock de productos cada vez que una orden de venta finaliza su ciclo (decreaseStock).

Item.- Posee la cantidad de unidades que el usuario ha elegido de determinado producto (quantity) y los impuestos que suma dicha selección (taxes).

Central.- Posee datos de identificación de la cooperativa, receptora de transacciones (id). Adicionalmente esta clase tiene las responsabilidades de: validar los datos del usuario cuando se ha confirmado la compra (validateUser), cambiar el estado de la orden de venta después de que el usuario haya confirmado la compra (confirmCart).

Cooperative.- Posee el identificador de la cooperativa (id) y la descripción de la misma (description).

Catalog.- Posee un identificador para el catalogo (id) y tiene responsabilidades de: obtener los productos que posee cada productor de la cooperativa (getFarmerProduct) y buscar un producto en específico (findProduct).

Product.- Posee datos de: identificación del producto (productid), nombre del producto (productname), peso acumulado entre todos los productores de un producto en específico (weightTotal), precio del producto (price), código del producto (code), descripción de las propiedades del producto (description).

Adicionalmente esta clase tiene las responsabilidades de: integrar la cantidad de producción de todos los productores de un producto en específico (integrateProduct) o todos los productos en general (integrateAllProduct).

Inventory.- Posee datos de: identificación del inventario (idinv), identificación del productor que tiene stock de determinado producto (farmerid), identificación del producto (productid), cantidad de producción del productor en un producto en específico (weight). Adicionalmente esta clase tiene la responsabilidad de buscar la cantidad de producción total de un producto en específico (findQtyProduct).

Payment.- Posee datos de: identificación del pago (id), el estado de confirmación del pago de la orden de venta (confirm). Adicionalmente esta clase tiene la responsabilidad de cambiar el estado de la orden de venta cuando el pago ha sido confirmado (confirmPayment).

5. Implementación de la Solución

Esta sección presenta cómo se efectuó el levantamiento de las funcionalidades de cada uno de los subsistemas propuestos. También presenta, la forma en que nuestra aplicación fue evaluada con la perspectiva de verificar su consistencia y mejorar el desempeño de sus funcionalidades.

Inicialmente se presenta el contexto y las consideraciones de desarrollo usadas para llevar a cabo esta etapa. Posteriormente, se menciona la metodología de trabajo para el levantamiento de la aplicación. Definidos estos lineamientos, se detalla la forma en que fue madurando cada uno de los subsistemas acorde a las perspectivas de su funcionalidad.

También se presenta la forma de organización de la implementación. Esto en cuanto a la estructura de paquetes y clases que fueron utilizadas en la aplicación. Finalmente se detalla la forma de uso de cada uno de los subsistemas.

5.1. Herramientas y Contexto de Implementación

Con la necesidad de generar un adecuado ambiente de desarrollo, para la ejecución de la implementación, y después de una investigación y pruebas efectuadas con varias herramientas (se detallará en sección 5.3), se adquiere un contexto final de implementación. El mismo que está conformado de las siguientes herramientas:

- *Eclipse IDE for Java EE Developer*: Entorno de desarrollo que ha permitido la creación y manipulación de clases JAVA, así como también archivos de configuración necesarios para estructurar la aplicación. Entre sus fortalezas tenemos: el manejo de programación orientada a objetos y la refactorización (actualización) de código en todas las relaciones de su esquema.
- *Spring 2.0.8 o superior*: Framework que permite implementar nuestra aplicación Web bajo la estructura tecnológica MVC (Modelo-Vista-Controlador). Posee una amplia comunidad de soporte y está abierta a la integración con otros frameworks como por ejemplo Struts.
- *Aptana*: Módulo complementario de Eclipse, que permite un control de los lenguajes Jsp, Javascript y JQuery.
- *PostgreSQL 8.4 o superior*: Motor de Base de Datos relacional. De fácil integración con módulos complementarios de manejo de datos geográficos.
- *PostGis 1.4 o superior*: Módulo que añade soporte de objetos geográficos a la base de datos PostgreSQL. Su cualidad es la apertura que tiene para enlazarse con servidores de mapas (proveedores de cartografía) de diferentes tecnologías.

- *Hibernate*: Herramienta que por medio de una configuración de mapeo-objeto-relacional (ORM), permite el enlace entre la capa lógica y la base de datos. Hibernate posee un lenguaje HQL, para consultar datos almacenados en objetos.
- *OpenLayers*: Librería que permite el manejo de eventos sobre cartografía. En el caso de nuestra aplicación se ha hecho uso de cartografía de OSM (Open Street Maps).

5.2. Metodología de Desarrollo

Los cuatro subsistemas establecidos en la aplicación se los implementó considerando la secuencia de usabilidad (Registro, Venta, Integración, Pago) que el usuario tiene con la aplicación. Las razones para considerar la secuencia de los subsistemas fueron dos: primero, mitigar los riesgos de mayor envergadura en la parte inicial y segundo, obtener una funcionalidad paulatina que pueda seguir siendo probada.

Ya considerando la globalidad del desarrollo de la aplicación, se buscaba estructurar una metodología de trabajo, la misma que fue planteada en dos instancias. En la primera, la implementación de cada subsistema en forma individual e incremental.

La segunda, un refinamiento de todos los subsistemas en forma conjunta e incremental. Cada incremento y cada instancia fueron añadiendo valor a las funcionalidades de la aplicación. Con un ambiente de trabajo establecido y primordiales consideraciones definidas. Se procede a presentar las dos instancias de implementación de la aplicación:

a) Primera Instancia

En esta etapa se implementó cada uno de los subsistemas en forma individual e incremental. El efecto de cada uno de ellos tuvo tres iteraciones, con el fin de obtener un modelo que pueda ser usado y probado de forma individual por un administrador de la cooperativa.

- *Iteración 1*: Centrado en la estructura de la funcionalidad y su alineamiento con los objetivos del subsistema. La iteración buscó la reducción de riesgos en cuanto a la tecnología a usar y la forma de usarla.
- *Iteración 2*: Orientada a satisfacer atributos de calidad. También, ajusta la organización de las clases empleadas en paquetes, y en caso de ser necesario, segmenta la estructura inicial del subsistema.
- *Iteración 3*: Se focalizó en mejorar el diseño de la interfaz de presentación. Para ofrecer una mejor usabilidad al usuario.

Al final de las iteraciones se evaluaba que cada subsistema se encuentre lo suficientemente maduro, para proceder a las pruebas internas con el Administrador de la Cooperativa.

b) Segunda Instancia

Esta instancia recepta como entrada los cuatro subsistemas de la aplicación en un estado considerable de uso. Éstos fueron sometidos a un fortalecimiento de su integración.

En esta instancia se pudo someter a la aplicación a un set de pruebas global, con el propósito de hallar, retroalimentar y afinar paulatinamente, falencias en las funcionalidades de todos los subsistemas. Así, de forma conjunta e incremental se brindó mayor madurez a SGPO. Con esta perspectiva se definieron tres iteraciones.

- *Iteración 1:* Se definió un plan de pruebas para la validación de cada subsistema.
- *Iteración 2:* Las pruebas de funcionalidad preparadas son ejecutadas con la ayuda de un focus group. Por su parte, las pruebas de rendimiento se apoyaron en el uso de un software que examina el nivel de stress de la aplicación.
- *Iteración 3:* Registro de observaciones y realización de correcciones.

5.3. Subsistemas desarrollados

Definidas las iteraciones que establecen el camino de cada uno de los subsistemas. En esta etapa presentamos el detalle de su trayectoria hacia la obtención de un nivel de madurez considerable, acorde a las características del destino deseado.

5.3.1. Subsistema de Registro

Este subsistema ha sido implementado para que el usuario pueda:

- Registrar sus datos de cuenta de usuario.
- Generar una sesión a través de la identificación y validación de sus datos de cuenta (user y password) registrados.
- Registrar sus datos de ubicación geográfica, a través de su participación con la cartografía presentada.

El resultado final del subsistema es obtener una sesión que guarde la identificación del usuario que se ha registrado.

a) Iteracion1

Tomando en cuenta que este subsistema era el punto de partida de la implementación de SGPO y que su desarrollo implica el uso de aplicaciones Geográficas. Se avizoraba dos considerables riesgos que debían empezar a ser mitigados.

El primer riesgo, de impacto transversal estaba alineado con el tipo de estructura tecnológica Web de la aplicación. La decisión sobre el Framework elegido sería la que prevalecería en los subsistemas siguientes.

En esta iteración se trabaja con el Framework Struts¹⁹. La investigación y pruebas realizadas estaban orientadas al acoplamiento del sistema con el uso de aplicaciones geográficas.

El segundo riesgo muy ligado al primero por la necesidad de acoplamiento, correspondía al tipo de tecnología geográfica a usar. Inicialmente se buscaba, dotarse de una cartografía independiente, a razón de disponer de shapes de la zona de ubicación de la cooperativa.

Estos shapes fueron tratados con GVSIG²⁰ (aplicación de GIS), para generar capas geométricas. Estas capas exportadas a nuestra base de datos de PostgreSQL con PostGis, podían ser usadas a través de un servidor de datos geográficos, como lo es GEOSERVER²¹.

Por un lado, un registro de datos tradicionales bajo la estructura Modelo-Vista-Controlador de Struts. Por otro lado, un esquema de presentación de cartografía con un tiempo de respuesta alto y con pobre presentación.

Fueron dos contextos que distaban mucho de ser fusionados. La rigidez que brindaban ambos entornos no generaba un buen ambiente de trabajo. Sin embargo, las clases fueron establecidas de acuerdo a su diseño inicial. Las mismas que fueron configuradas con su capa de persistencia a través de Hibernate. Con el enlace establecido, se precedió a la creación de las tablas necesarias en el motor de base de datos.

b) Iteración 2

En esta iteración se continuó investigando acerca de la estructura tecnológica Web de la aplicación, el uso de aplicaciones geográficas y su acoplamiento. En esta ocasión se probó las ventajas del Framework SPRING y DJANGO²². Este último caracterizado por su fácil conectividad con PostGis y el API de GoogleMaps.

Se analizó que el nivel de madurez del Framework Spring es más alto que el de Struts. Este último, había lanzado su última versión con un esquema completamente diferente a su predecesora. Hecho que no ofrecía muestras de confiabilidad, mientras que el esquema de Spring se mostraba menos rígido y brindaba servicios de integración. Como por ejemplo, la fusión con el mismo Framework Struts.

Se efectuaron pruebas del Framework Spring con nuestro modelo de clases diseñado. Se determinó que una de sus bondades funcionales está en la forma de recibir las peticiones del usuario.

¹⁹ Servicios Web Apache 2011, Apache Software Foundation-Struts, último acceso 05 Septiembre 2011 <<http://struts.apache.org/>>

²⁰ Servicios Web gvSIG 2011, Grupo gvSIG, último acceso 05 Septiembre 2011 <<http://www.gvsig.org/web/>>

²¹ Servicios Web Geoserver 2011, GeoServer by OpenGeo, último acceso 05 Septiembre 2011 <<http://geoserver.org/display/GEOS/Welcome/>>

²² Servicios Web Django 2011, Django Software Foundation, último acceso 05 Septiembre 2011 <<https://www.djangoproject.com/>>

Un modelo de control en la lógica de negocio, permite distribuir estas peticiones de la forma que el implementador requiera. Este hecho permitió un equilibrio en la carga de trabajo de la aplicación, para evitar cuellos de botella.

En cuanto a DJANGO (tecnología diseñada con python), su mayor virtud fue la integración directa entre los datos geográficos almacenados en PostGis y el API de GoogleMaps. Éste no requería de un enlace intermedio, como es el caso de OSM que requiere del uso de la librería OpenLayers.

El inconveniente de continuar usando DJANGO fue que su tecnología y estructura era completamente incompatible con Spring o Struts, no se encontró punto de integración. El acoplamiento que se buscaba entre la tecnología de estructura Web y aplicaciones geográficas, tampoco se logra obtener en esta iteración. Sin embargo se conoce las ventajas y desventajas de los Framework Spring y Struts. Por el lado estructural se corrigieron pequeños detalles de la estructura de clases, llegando a definir su diseño final.

c) Iteración 3

En esta iteración se decidió el uso del Framework Spring como tecnología de estructura Web. Principalmente por sus cualidades de integración y su esquema de recepción de acciones a través de controladores.

En cuanto a la estructura geográfica, en esta iteración se probó las siguientes tecnologías: OSM (OpenStreetMaps), OpenLayers (Biblioteca JavaScript) y CloudMade²³ (Api de servicios geográficos). Durante la investigación de la herramienta OSM, se encontraron varias virtudes que aportaban a nuestras necesidades:

- Alto nivel de madurez en la información de vías en zonas rurales (en determinadas zonas posee mayor información que googlemaps).
- La actualización de cartografía es soportada por comunidades a nivel mundial. Disponen de APIs online para la edición de esta cartografía. El registro para formar parte de la comunidad está abierto y la información geográfica que se aporte es validada antes de ser publicada.
- Ofrece integración a estructuras Web y control de eventos en cartografía. Esto con el soporte de bibliotecas externas y de enlace como OpenLayers y/o CloudenMade.
- Posee servicios de geocoding y geocoding reverse. Es decir servicios que permiten obtener nombres de calles, avenidas y otros en base a coordenadas geográficas.
- Cartografía OpenSource, con disponibilidad de descargarla y configurarla en servidores propios.

²³ Servicios Web CloudMade 2011, CloudMade. Map data, último acceso 05 Septiembre 2011 <<http://cloudmade.com/>>

Para elegir la herramienta intermediaria que permita el enlace de OSM con nuestra aplicación, teníamos dos opciones: OpenLayers y CloudMade. Se eligió OpenLayers para seguir la misma línea OpenSource.

CloudenMade tenía un buen desempeño, sin embargo su servicio era manejado por cuotas, es decir si sobrepasaba un número determinado de peticiones, las siguientes tenían un costo. OpenLayers permite el control de eventos o acciones hechas por el usuario en cartografía como son Click, DobleClick, DragDrop.

También brinda un adecuado control de las características visuales de OSM, y sobre todo brinda facilidades de integración con la estructura Web de Spring. Con este contexto finalmente se logró obtener el acoplamiento deseado. La obtención de esta fusión fue uno de los más importantes logros para el autor. Puesto que el diseño obtenido era una definición transversal para la aplicación. Fue la base de la estructura para la implementación de los subsistemas siguientes.

Además se estabilizó el ambiente de desarrollo, quedando estructurado como se lo había planificado en un entorno multicapas (presentación, lógica de negocio, datos). Adicionalmente se afinó el diseño de la interfaz en las plantillas de registro de datos tradicionales del usuario y en su interactividad con la cartografía.

Se buscaba que el usuario pueda fijar su ubicación en cartografía de una manera amigable y que sus datos geográficos se registren en nuestra aplicación. El tipo de interfaz que se decidió usar para una mejor presentación y usabilidad, fueron los tipos de ventanas float (ver Figura 5.8)

5.3.2. Subsistema de Venta

Este subsistema recibe como entrada una sesión que contiene datos de identificación del usuario. El subsistema ha sido implementado para que el usuario pueda efectuar las siguientes actividades:

- Buscar información tradicional y geográfica de clientes, productores, frutas y/o hortalizas.
- Administrar la selección de productos en una canasta de compras. La orden de venta puede estar relacionada a uno o varios productores. La administración implica añadir, modificar y retirar la cantidad de productos que se desee de la canasta de compras.

Para obtener estos incrementos en nuestra aplicación fue importante tener en cuenta las siguientes consideraciones: el diseño de una estructura de datos que permita soportar un catálogo de productos por productor y que conjuntamente represente un catálogo general para la cooperativa.

La consideración de una sesión activa para la administración de selección de productos en una canasta de compras. El resultado final de este subsistema fue obtener una orden de compra que contenga el listado y cantidad de productos que el usuario desea.

a) Iteración 1

Con estas metas definidas, se planteó un modelo de clases que soporte sus funcionalidades. Así, se avizoraron dos considerables riesgos que debían ser reducidos en el diseño de la estructura de clases.

Primero, se buscó un modelo de datos que soporte un catálogo de productos, cuyos elementos y cantidad disponible, puedan ser identificados individualmente por productor. Pero también que puedan ser identificados de forma conjunta, representando el catálogo general de la Cooperativa. Así mismo se debía dejar abierta la posibilidad de que el modelo soporte el anexo de otras cooperativas.

Segundo, se necesitaba que el modelo permita controlar y administrar la orden de venta, que la sesión activa poseía en el sistema. Con estas consideraciones se definió la estructura de clases presentada en la Figura 4.9. Una vez que estas dos secciones de esquema estaban definidos. Se creó una clase virtual (no fue necesario crearla en la BD) denominada Central, ésta fue la encargada de enlazar el primer modelo con el segundo.

Posteriormente, se usa el diseño establecido para crear cada uno de los DataBeans Java (clases con métodos get/set). Los mismos que también fueron configurados en la capa de persistencia para que puedan establecer enlace con las tablas del motor de base de datos.

Una vez que se levantó el enlace se procedió a crear las tablas necesarias. Así como también a cargar los datos recopilados de los productores de la cooperativa conjuntamente con su producción (ver Anexo 7). Estos datos corresponden a su producción semanal y es importante señalar que es una producción constante durante todo el año.

La tecnología de las herramientas de desarrollo no representaba un riesgo significativo. Pues con el incremento resultante del subsistema anterior, este factor se encontraba estable. Sin embargo las herramientas que se definen para la implementación de este subsistema son el Ide Eclipse y OpenLayers.

b) Iteración 2

Las modificaciones en la estructura del modelo de clases fueron leves, únicamente se afinaron detalles de campos y tipos de dato. Junto con estas modificaciones se organizaba a las clases en paquetes que representen su acoplamiento.

También, se preparó la capa de presentación para que despliegue los datos cargados en la BD. Esta interfaz de presentación es un reflejo de la visión planteada en el modelo de clases diseñado en la primera iteración.

La interfaz de la sección del catálogo (ver Figura 5.10) presenta en su parte superior el listado de los productores de la cooperativa, y en su parte inferior el catálogo de productos del productor que esté seleccionado.

Posteriormente se prepararon los Queries HQL (Hibernate Query Language) que fueron usados para la búsqueda de información tradicional y geográfica de clientes, productores, frutas y/o hortalizas. Estos queries forman parte de la capa de lógica de negocio e interactúan con la capa de persistencia, para extraer en forma de objetos la información requerida.

Se preparó a la capa de presentación para que reciba peticiones de búsqueda. Las mismas que interactuaran con los queries pertinentes, para que éstos, extraigan la data necesaria. Obtenidos los resultados de búsqueda se adecua la capa de presentación, para que en la sección del catálogo y ubicación presente la información respectiva. Para desplegar datos geográficos en cartografía se hace uso de la librería OpenLayers.

c) Iteración 3

En esta iteración se afinó los queries HQL de búsqueda con el fin de mejorar el tiempo de respuesta. Por otro lado, se trabaja en el diseño de una ventana float genérica (ver Figura 5.13) que permita examinar el detalle del producto y mantener (añadir, modificar, retirar) el mismo en una canasta de compras.

Con la ventana genérica lista, se procedió al diseño de una canasta de productos amigable y de fácil uso, que permita anexar y retirar productos de una forma ágil. Junto con este diseño, se tuvo que preparar a la capa de negocio para que tenga la capacidad de recibir peticiones de mantenimiento en la canasta de compras (añadir, modificar y/o retirar). Al momento de añadir un producto por primera vez se debía crear una orden de venta en estado inicial.

Esta orden está vinculada a la sesión activa del usuario. Cada actualización en su canasta de compras será almacenada a nivel de base datos inmediatamente. Esto con el fin de tener contingencia en situaciones de pérdida de conexión. Cuando el usuario se vuelve a conectar, recupera el último estado de su canasta de compras.

5.3.3. Subsistema de Pago

Recibe como entrada una orden de compra que contiene el listado y cantidad de productos que el usuario pretende adquirir. Este subsistema ha sido implementado para que el usuario pueda realizar las siguientes actividades:

- Confirmar su canasta de compras y presentar el detalle de su orden de compra.
- Pagar de forma online su orden de compra, por medio de una cuenta PayPal.

Para obtener estos incrementos en nuestra aplicación fue importante establecer estados (creada, confirmada, pagada) a la orden de venta. También fue necesario establecer una conexión con los servicios Web de PayPal. Como resultado de este subsistema se obtiene una confirmación o revocación del pago de la orden.

a) Iteración 1

Con estas metas definidas, el principal riesgo que se avizoraba era el establecimiento de una conexión con PayPal. Se decidió trabajar con PayPal por su reconocimiento internacional en cuanto a seguridad (Protocolo SSL) y servicios de pago online.

Sin embargo previo al levantamiento de la conexión para el efecto del pago online. Se habilitó una opción para que el usuario confirme su canasta de compras. Con esta confirmación lo que se buscaba es cambiar el estado de la orden de compra y que el usuario pueda apreciar el costo y estado final de su orden. En este detalle de compra se fijó el botón de pago, para que con su ejecución se levante la conexión a PayPal y se pueda efectuar la transacción.

La acción resultante del botón pago era el principal riesgo que debía ser mitigado. Así, se investigó acerca de la estructura de conexión con los servicios Web de PayPal. Adquirido este conocimiento, se tuvo que preparar la capa de negocio para hacer el levantamiento de este enlace. Varios son los tipos de pago y estructuras de conexión que ofrece PayPal, entre ellas:

- *Direct Payment*: El usuario efectúa el pago de la transacción en el Website del vendedor. La conexión a PayPal es únicamente para cargar en la cuenta del vendedor el valor pagado de la orden de compra. Es decir el Website tiene la responsabilidad de generar el ambiente de seguridad necesario, para los datos de la tarjeta de crédito
- *Express Checkout*: El usuario efectúa el pago de la orden de compra en el ambiente de seguridad de PayPal con el uso de su cuenta. La conexión a Paypal es para enviar el total o detalle de la orden de compra que se ha generado en el Website del vendedor.
- *Adaptive Payments*: Es el tipo de pago más completo y configurable de los propuestos por PayPal. Permite configurar y estructurar modelos de pago tomando en consideración una o varias cuentas de vendedores.
- *Mobile Express*: Forma de pago dirigida a aplicaciones móviles. Posee una estructura de pago similar a Express Checkout. Soporta pagos online en plataformas Android.

Con este análisis dos eran las estructuras que encajaban con las necesidades de SGPO. Éstas fueron: Adaptive Payments y Express Checkout. Inicialmente se trató de configurar la estructura Adaptive Payments, para lo cual se efectuó los siguientes pasos:

- Generar las credenciales de conexión (user, password, signature) con nuestra cuenta PayPal. Estas credenciales fueron esenciales para levantar la conexión desde la capa de negocio de nuestra aplicación hasta los servicios de PayPal.
- Preparar una clase en la capa de negocio para que reciba la petición de pago, hecha por el usuario en la capa de presentación. En esta clase se fue preparando las funciones para levantar la conexión con PayPal y enviar la orden de compra que haya generado el usuario.

El nivel de robustez y servicios que poseía este tipo de pago, tornaba a la aplicación compleja. Es en este momento cuando se evalúa su detalle y se aprecia cualidades que no eran necesarias para nuestra funcionalidad.

Por ejemplo, el flujo de este tipo de pago exigía obtener una aprobación en PayPal de la orden de compra que se busca pagar. Siendo éste un demandante de tiempo y recursos que no se ajustaban a nuestra necesidad.

Sin embargo se adquirió experiencia sobre las estructuras de los tipos de pago existentes. También se logró establecer la conexión inicial de nuestra capa de negocio con los servicios Web de PayPal.

b) Iteración 2

En esta iteración se continuó afinando la estructura de conexión con PayPal. También, se hizo pruebas de uso con la forma de pago Express Checkout. Ya con la conexión establecida, los pasos que se requería en este tipo de pago fueron:

- Validación de datos complementarios de la orden de venta. Con el fin de ahorrar tiempo en la entrega de estos datos. Este modelo empieza a reutilizar la información de dirección y contactos que previamente había ingresado el usuario.
- Enlace con el API de PayPal y envío de la orden de compra. En esta API es en donde el usuario puede loguearse con su cuenta PayPal, para hacer el pago de su orden. Como el API también está direccionado a la cuenta del vendedor, el pago también se registra en su cuenta.
- En el anterior paso el usuario tiene dos posibilidades hacer el pago o cancelar la compra. En el caso de que se confirme el pago, el usuario es dirigido a la página inicial de nuestro Website. Adicionalmente se receipta un KeyPayment que es un código que representa la confirmación del pago de la orden.
- Para recibir el KeyPayment que representa la confirmación de pago, se procede a diseñar una nueva clase en nuestra capa de lógica de negocios.

Finalmente en esta iteración se ha logrado mitigar el riesgo de conexión y trabajo con PayPal. Adicional se precedió a organizar en paquetes las clases y funciones que se fueron preparando para el desempeño de esta funcionalidad.

c) Iteración 3

En esta iteración se implementa los detalles que posee la clase *PaymentConfirmController*. Esta clase recibe el valor del *KeyPayment*, para finalmente proceder a cerrar la orden de venta ejecutando los siguientes pasos:

- Validar que el *KeyPayment* corresponda a una orden cuyo pago haya finalizado con éxito.
- Cambiar la orden de venta a su estado final.
- Reducir el stock de los productos adquiridos. El stock se lo reduce de acuerdo al productor del cual se haya seleccionado el producto.

Por otro lado se afina detalles de diseño, en dos instancias: primero en la interfaz de presentación al momento de enlazarse con PayPal, y segundo en la interfaz de recepción del *KeyPayment* (cierre de la orden de venta).

5.3.4. Subsistema de Integración

Para este subsistema no fue importante ningún tipo de información de entrada. Este subsistema permite al usuario:

- Integrar la producción individual de los productores de la cooperativa. En el caso de que la demanda supere la cantidad de producción individual de los productores.
- Administrar la selección de productos considerando el stock integrado de la producción.

Este subsistema no estuvo centrado en la solución de un riesgo de gran envergadura, sino que representa un servicio de valor agregado de la aplicación.

a) Iteración 1

La funcionalidad de este modelo no representaba mayor riesgo para el desempeño de la globalidad de la aplicación puesto que es un servicio complementario para SGPO. Sin embargo, se analizó el modelo de clases y los atributos necesarios para su implementación. Se definió que la clase que almacena el total de esta integración sea el campo *Weight* de la clase *Product*. Los datos de donde se toma la cantidad de producción, es *Inventory*. Que es la clase que posee la cantidad de producción del producto en relación al *Farmer*.

b) Iteración 2

En esta iteración se diseñó un procedimiento, que considerando cada producto del catálogo filtre la cantidad de producción que posee cada productor en relación a

determinado producto. El total de esta integración obtenida se almacenó en el campo definido en la iteración anterior.

La ejecución de este procedimiento inicia en el momento que el usuario elige la solapa *Integrar* del menú central derecho (ver Figura 5.19). Se prepara la capa de negocio, para que reciba esta petición y se enlace con el procedimiento implementado. En el transcurso se va afinando el procedimiento para obtener un buen tiempo de respuesta en la obtención de la información integrada.

c) Iteración 3

Se preparó a la capa de presentación, para que despliegue el catálogo de productos con su cantidad de producción integrada. Para esta presentación se utilizó el mismo diseño del catálogo utilizado en el subsistema de venta.

En esta interfaz también se habilitó las acciones de mantenimiento de la canasta de compras. Es decir el usuario puede añadir, modificar y/o retirar la cantidad de productos que desee.

5.4. Estructura de Paquetes

Para mantener la organización de las clases implementadas, fue necesario acoplarlas en paquetes que representen su rol funcional y estructural (ver Figura 5.1). Estos paquetes también representan el Modelo-Vista-Controlador que se ha logrado preparar con el Framework Spring.

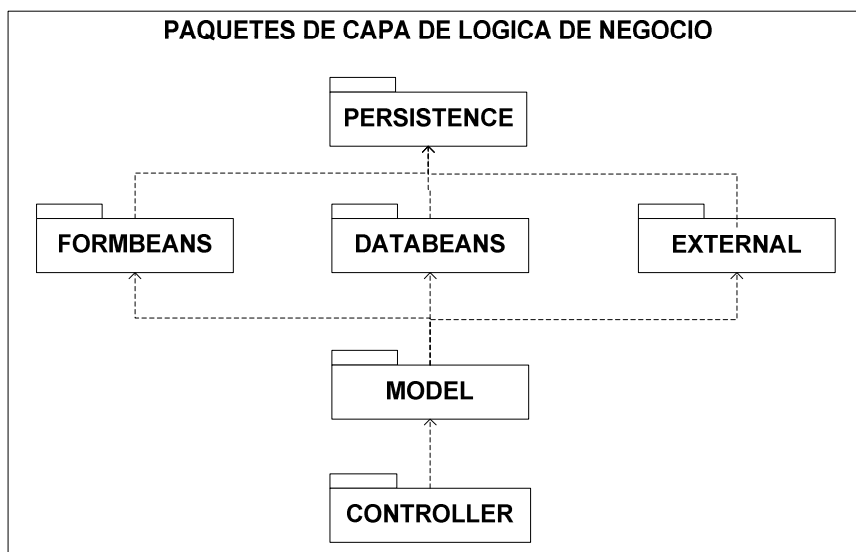


Figura 5.1 Paquetes de la Capa Lógica de Negocio

- **CONTROLLER**: Posee las clases controladoras de los subsistemas: Registro, Venta, Pago e Integración. Estos controladores son los que reciben las peticiones que el usuario ejecuta en la capa de presentación.

- *DATABEANS*: Corresponde a las clases java que han sido implementadas, en base a nuestro modelo de clases establecido. Cada atributo de la clase posee sus métodos get and set.
- *MODEL*: Posee clases que establecen el enlace de los databeans con los controladores. Aquí es en donde se prepara las sentencias HQL, para la consulta o modificación de datos. Cada controlador hace uso de una o varias clases model, esto de acuerdo al tipo de actividad que tenga el controlador.
- *FORMBEANS*: Son clases que permite la lectura de los datos de formularios Web diseñados en la capa de presentación. De igual forma que un databean sus atributos poseen métodos get and set.
- *EXTERNAL*: Posee databeans que en el diseño se los consideró como clases externas a la aplicación SGPO. A razón de que su funcionalidad puede ser profundizada y manejada como un sistema más.
- *PERSISTENCE*: Posee archivos hbm.xml que guardan la configuración de enlace de cada uno de los databeans con las tablas de nuestra base de datos.

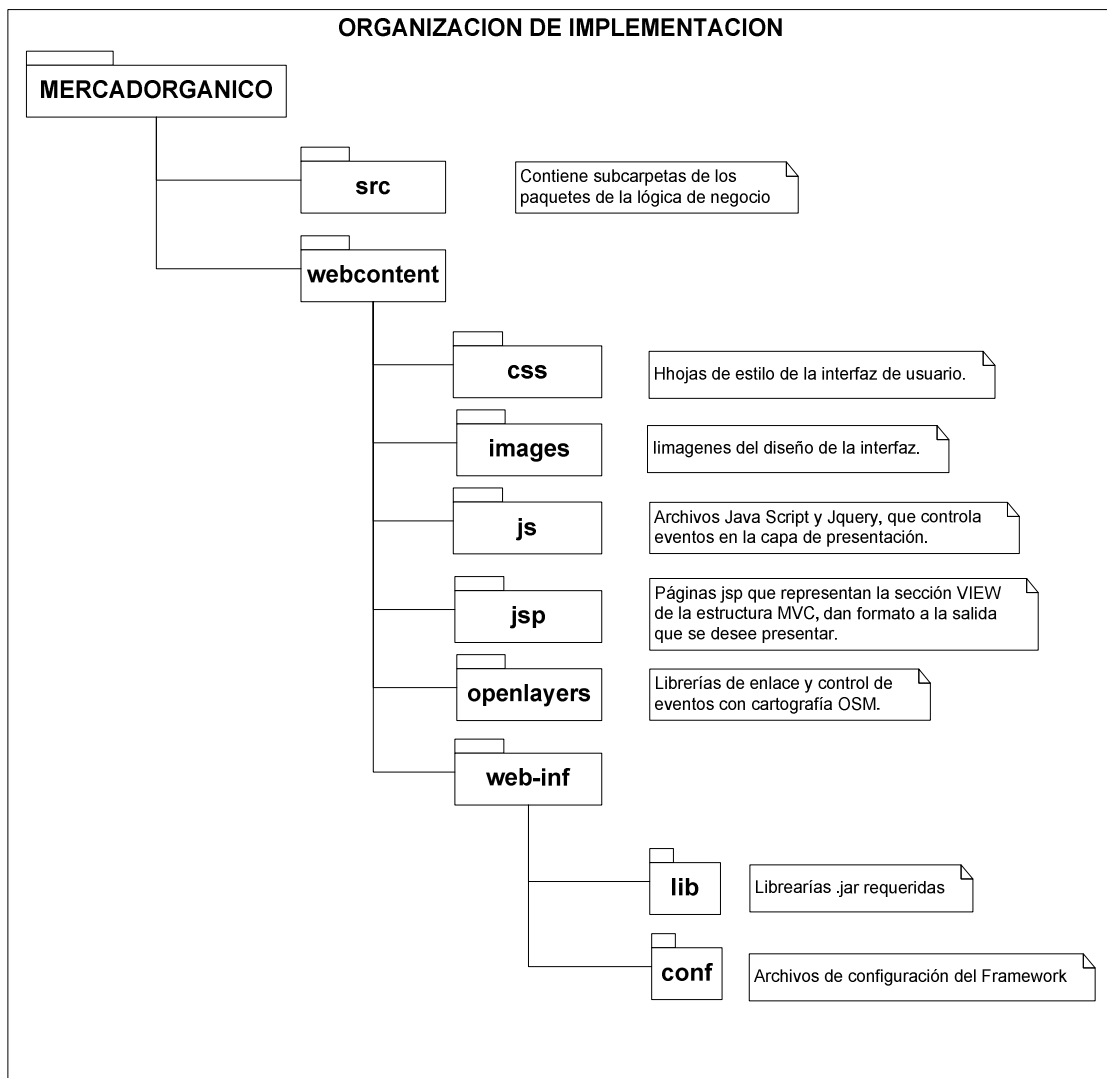


Figura 5.2 Organización de Archivos en SGPO

Después de haber organizado las clases implementadas en nuestra capa de negocios, se procede a organizar los archivos y librerías que complementan nuestra aplicación. Para esto se hace uso de la estructura de directorios que generalmente posee un proyecto Web dinámico J2EE (ver figura 5.2).

En el directorio *src* se almacena todas las clases de la capa de lógica de negocio, así como también los archivos de configuración de hibernate para el enlace de almacenamiento de datos. Mientras que en el directorio *webcontent* se almacena todos los archivos correspondientes a la capa de presentación: imágenes, hojas de estilo, efectos y librerías.

5.5. Estructura de Clases y Vistas

Las páginas JSP son las vistas (front-end) de la estructura MVC de nuestra aplicación. Por medio de ellas el usuario emite o recibe acciones de la capa lógica de negocios.

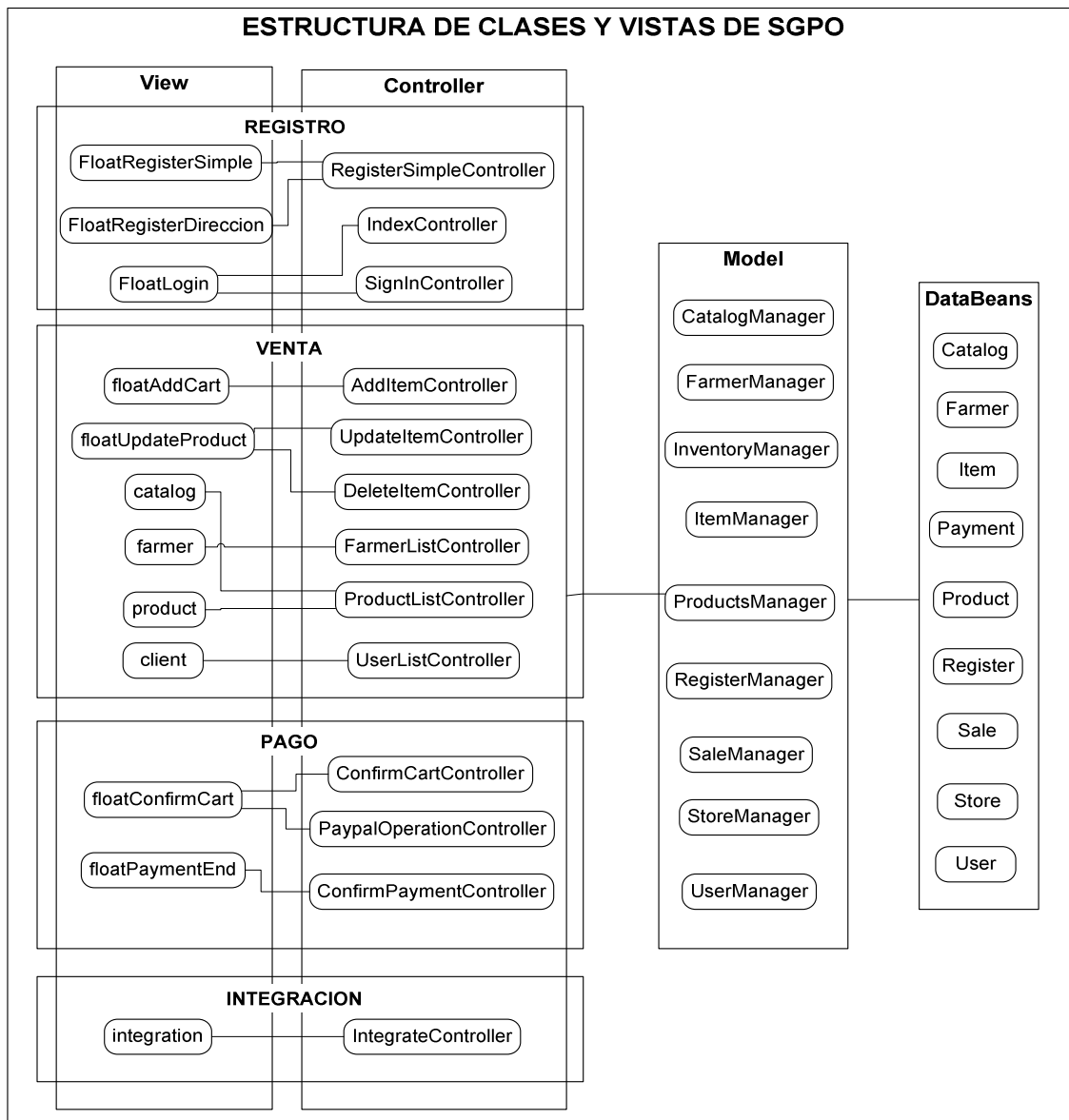


Figura 5.3 Clases de las Capas de Negocio

En la figura 5.3 se presenta las principales vistas y clases que se implementaron en SGPO. Las peticiones del usuario se reciben a través de controladores, los mismos que analizan el requerimiento y lo entregan a las clases model, para que las procese haciendo uso de los databeans que corresponda.

5.6. Funcionamiento de SGPO

En la siguiente sección se presenta la forma de instalación y configuración de la aplicación. Así también, la forma de usar las particularidades que tiene cada uno de los subsistemas que conforman SGPO.

5.6.1. Instalación y Configuración

Para que la aplicación pueda ser ubicada en un servidor, no se requiere de acciones de gran complejidad, únicamente considerar los siguientes prerequisites y pasos de instalación.

Prerequisites:

- *JDK(Java Development Kit) 1.6 o superior.* Producto de SUN que posee una colección de herramientas de programación.
- *Tomcat 6 o superior.* Servidor Web que soporta objetos y páginas JSP.
- *Postgres 8.4 o superior.* Motor de base de datos.
- *Postgis 1.4 o superior.* Modulo complementario de Postgres que permite el almacenamiento de datos geométricos.
- *Mercadorganico.war.* Archivo compactado que contiene todas las clases, archivos jsp y librerías de SGPO.

Instalación:

- Ubicar al archivo *mercadorganico.war* en el path de aplicaciones del servidor Web Tomcat.
- Reiniciar el servidor Web para que el archivo haga deploy y pueda ser usado.
- Configuración del archivo *hibernate.cfg.xml*, de acuerdo a los datos de conexión a la base de datos.

5.6.2. Forma de Usar los Subsistemas

Del subsistema de implementación, cuatro son los subsistemas resultantes. A continuación se detalla la forma de uso de cada uno de ellos. Partimos especificando las seis secciones que conforman la interfaz de nuestra aplicación, en la Figura 5.4 se puede apreciar la ubicación de cada una de ellas.



Figura 5.4 Secciones de la Interfaz de SGPO

1. *Sección Mapas*: Zona donde se despliega la información de cartografía para el usuario, ya sea para determinar la ubicación de un productor o la suya propia.
2. *Sección Cabecera*: Posee un menú de opciones vinculadas a dos actividades: el registro (derecha) y la búsqueda (izquierda). Cuando no existe una sesión activa el usuario puede registrarse o loguearse si dispone de sus datos de cuenta. También puede buscar información geográfica de acuerdo a las categorías: clientes, productores, frutas, hortalizas, cooperativa, provincia.
3. *Sección Catálogo*: Zona donde se despliega el listado de productos disponibles de la cooperativa, clasificándolos por productor. Es aquí donde el usuario puede seleccionar los productos deseados para preparar su canasta de compras.
4. *Sección Canasta*: Lugar donde se van acumulando los productos seleccionados. Aquí el usuario puede modificar el contenido de su canasta. Esta sección también posee el link de confirmación de la canasta. Que se ejecuta cuando el usuario haya decidido el contenido de su compra.
5. *Sección Estado*: Aquí se presenta el estado de la compra con respecto a un usuario. Cuatro son los estados por los cuales el usuario transita hasta finalizar su compra: registro, selección, confirmación y pago.
6. *Sección Flotante*: Son ventanas float que presentan información complementaria para cada uno de los subsistemas. Generalmente serán formularios que obtienen información de los usuarios y la entregan al sistema.

- *Subsistema de Registro*

La aplicación parte con la presentación de una ventana flotante que posee el mapa de Ecuador y sus provincias (ver Figura 5.5). Al momento la única provincia habilitada es Pichincha, correspondiente a la capital Quito. Esto a razón de que los pedidos de compra se focalizan en la zona mencionada debido a la ubicación de la cooperativa.



Figura 5.5 Interfaz Inicial de SGPO

Al seleccionar la provincia en la sección mapas se carga la cartografía de la ciudad de Quito. El cursor del usuario tiene la gráfica de una casa, para que fije la posición de su dirección. Este ícono se asienta en la cartografía y puede ser movilizado hasta ubicar la posición que el usuario desee (ver Figura 5.6).



Figura 5.6 Interfaz de Registro de Ubicación

Junto con el ícono de ubicación, se presenta una ventana en la que el usuario puede añadir detalles adicionales de su dirección. Para guardar la ubicación, si el usuario no está logueado la aplicación, le pide hacer un login con su cuenta (ver Figura 5.7) o un registro de sus datos (ver Figura 5.8).



Figura 5.7 Interfaz de Ingreso de Usuario

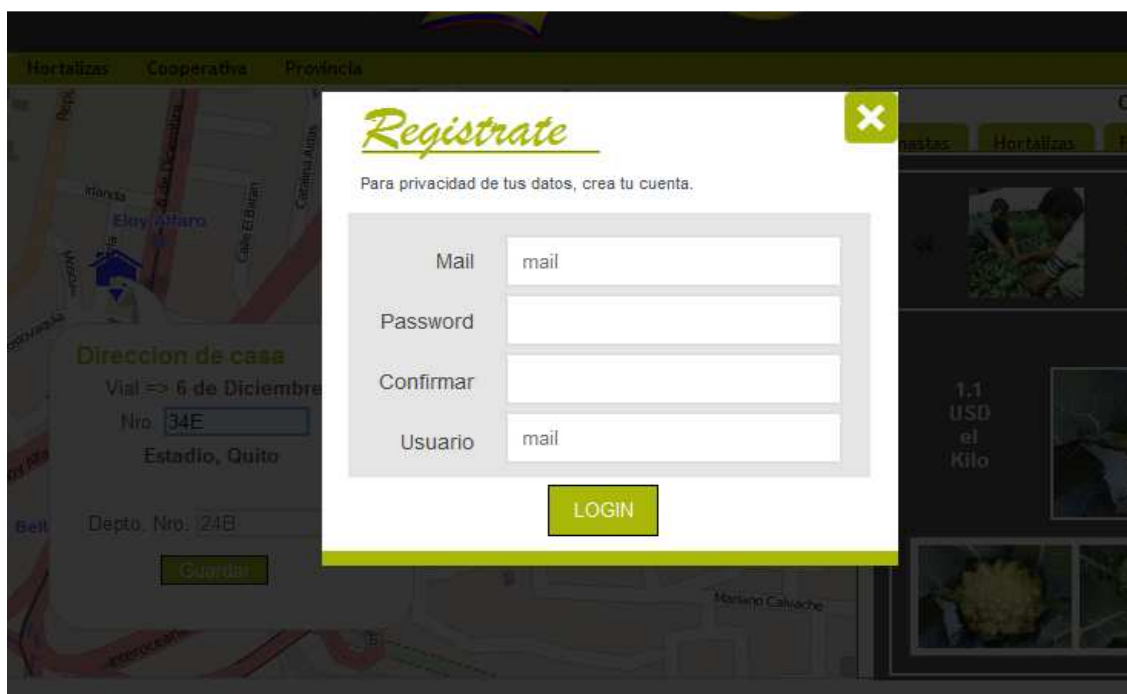


Figura 5.8 Interfaz de Registro de Usuario

Registrado y/o logueado el usuario podrá guardar su ubicación (ver Figura 5.9). Como resultado de estas acciones obtenemos una sesión activa con los datos de identificación del usuario.



Figura 5.9 Interfaz de Dirección Almacenada

- *Subsistema de Venta*

La mayor parte de las actividades de este módulo se las efectúa en las secciones catálogo y canasta. Inicialmente por defecto el sistema despliega en la sección catálogo todos los productos que posee la cooperativa, clasificándolos por productor (ver Figura 5.10). El usuario puede revisar y seleccionar el productor que desee, recorriendo los listados a su derecha o izquierda.

Las imágenes de la parte superior poseen el listado de productores, las imágenes de la parte inferior poseen el listado de productos. Éstos se vuelven a cargar cada vez que el usuario selecciona un productor. En la parte central se presenta el producto seleccionado con información de su precio y el stock disponible.



productos Integrar Videos noticias



Figura 5.10 Sección de Interfaz de Selección de Productos

En el caso de que se desee buscar un producto en específico, el usuario debe hacer uso del menú de la sección cabecera. Así, la sección catálogo carga información de acuerdo a la información buscada, mientras que la sección mapas, presenta la ubicación del lugar donde poseen el producto buscado (ver Figura 5.11).

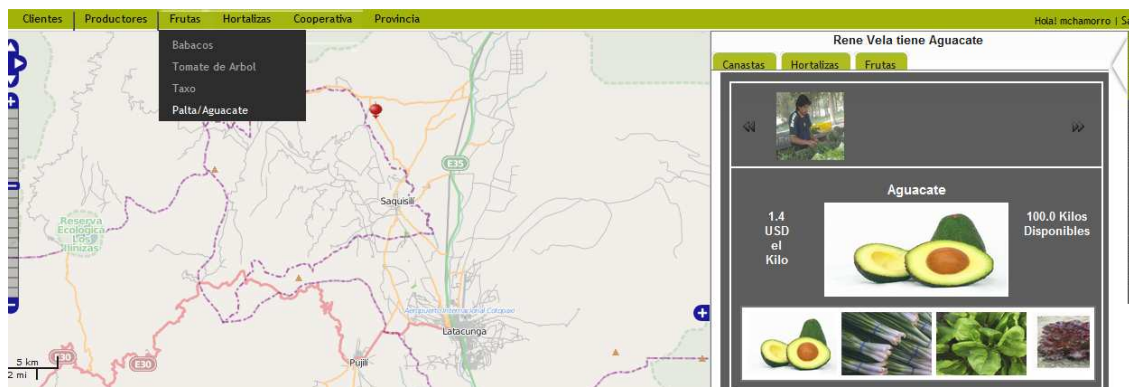


Figura 5.11 Búsqueda con Información Geográfica (producto)

Existen también otras categorías de búsqueda de información geográfica (ver figura 5.12), aquellas que estén vinculadas con información de productos, hacen que la sección catálogo sea actualizada.

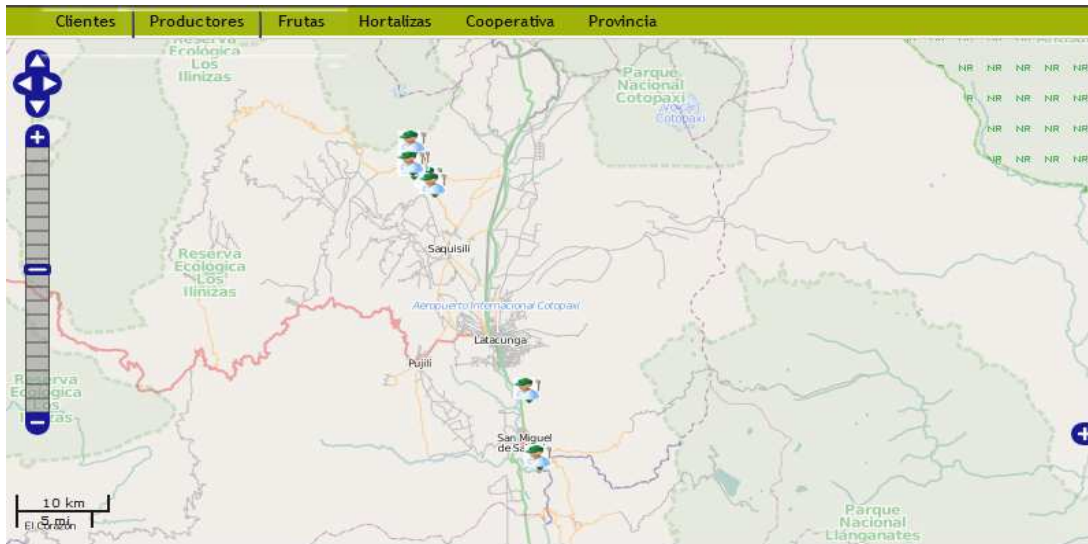


Figura 5.12 Búsqueda con Información Geográfica (productor)

La búsqueda será un servicio adicional para que el usuario prepare su canasta de compras. Continuando con el subsistema de venta, el detalle del producto (ver Figura 5.13) se lo puede apreciar cuando el usuario hace un click sobre la imagen central.

En esta ventana float se puede especificar la cantidad del producto y añadirlo a la canasta de compra. Es importante mencionar que para el efecto de las tareas de agregar, modificar y/o retirar, el usuario debe estar logueado.

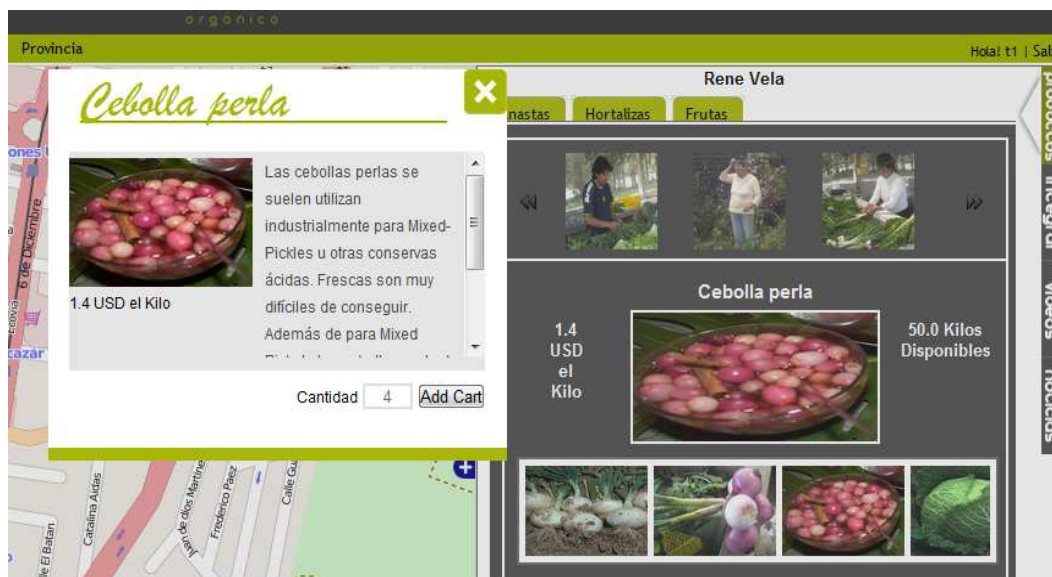


Figura 5.13 Interfaz de Ingreso de Ítems

La canasta de compras puede abrirse o cerrarse de forma dinámica, aquí se puede apreciar el detalle de los productos seleccionados por el usuario. En caso que lo desee, el usuario puede modificar o retirar la cantidad de productos (ver Figura 5.14)

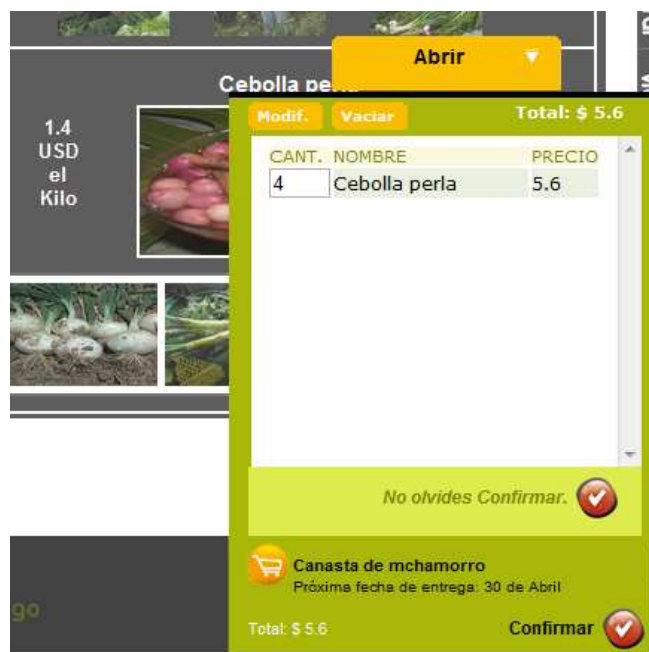


Figura 5.14 Interfaz de Administración de Productos

La edición o retiro de productos de la canasta de compras, se la efectúa en una ventana float con las mismas características que la ventana de agregación (ver Figura 5.15).

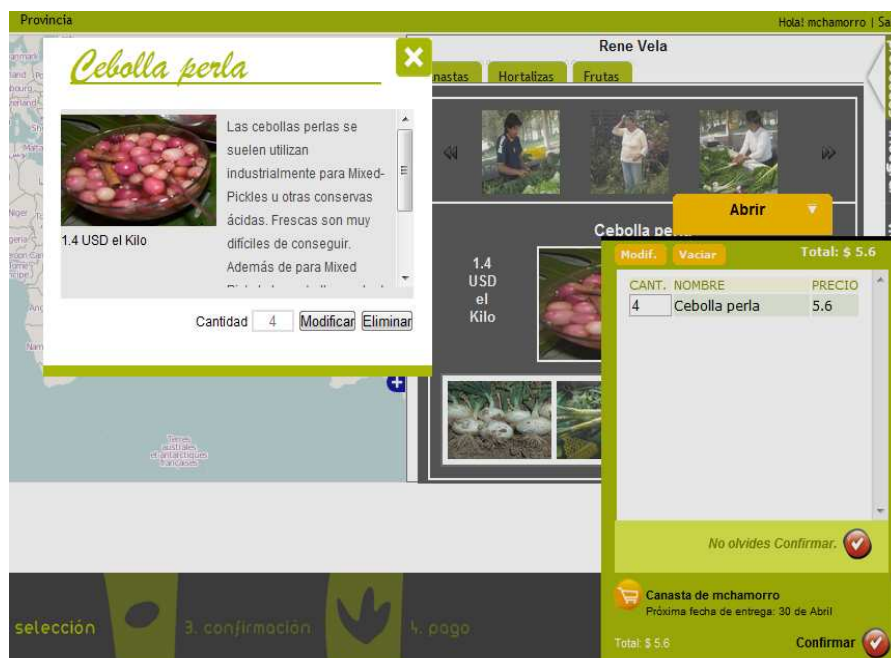


Figura 5.15 Interfaz de Actualización de Productos

- *Subsistema de Pago*

Cuando el usuario decida que su canasta de compras está lista, puede confirmar su compra con el botón rojo ubicado en la parte inferior derecha de la sección canasta. Al confirmar, su orden de compra se despliega en una ventana flotante (ver figura 5.16).



Figura 5.16 Interfaz de Confirmación de Orden de Venta

Si está de acuerdo con la orden de compra, procede hacer el pago haciendo click en el botón PayPal. Esta acción lo enlazará a una interfaz de PayPal, en la cual el usuario aprecia el resumen de su compra (ver figura 5.17). En el caso de que el usuario decida hacer el pago, puede hacer uso de su cuenta PayPal. Caso contrario si cancela su orden será direccionado a la interfaz inicial de Mercado Orgánico.

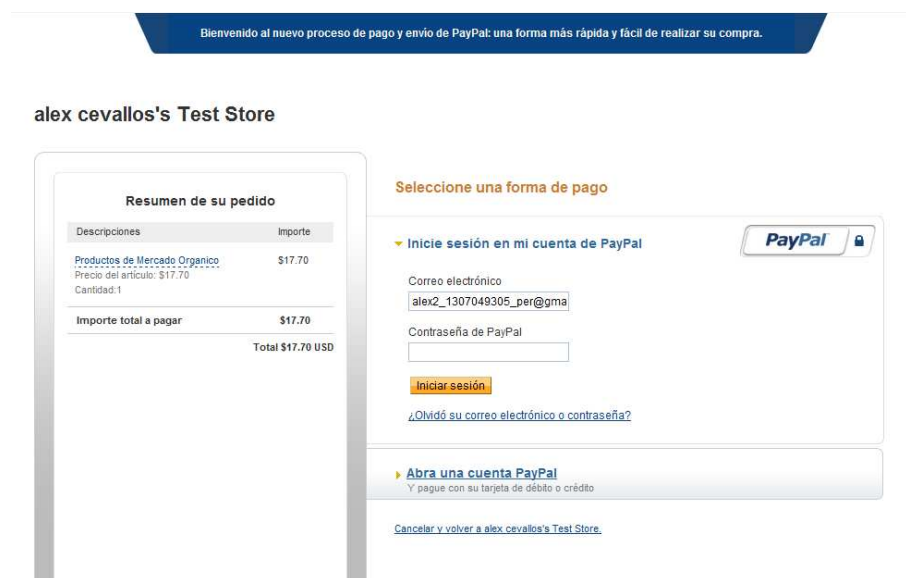


Figura 5.17 Interfaz de Pago con PayPal

Después de ejecutarse el pago, el usuario es dirigido a una interfaz de mercado orgánico, la cual presenta un aviso de que su pedido ha sido aceptado.

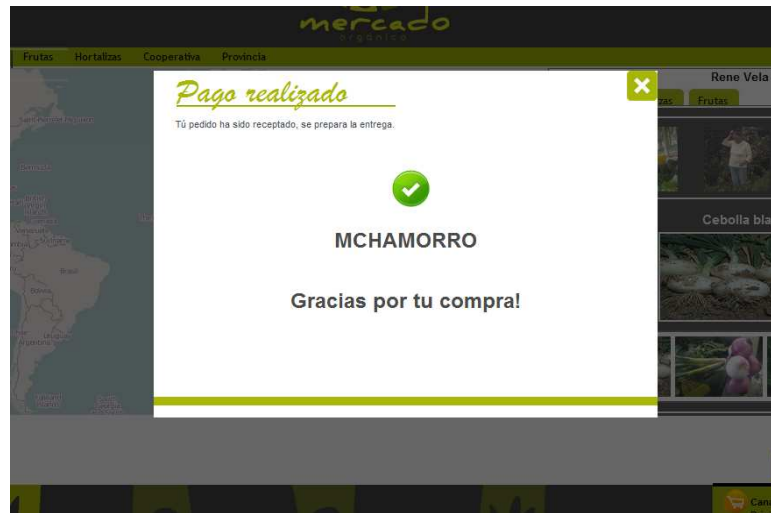


Figura 5.18 Interfaz de Finalización de Pago

- *Subsistema de Integración*

Si durante la selección de productos, el usuario requiere de una cantidad, cuyo valor supera la capacidad individual de un productor. El sistema por medio de la opción integrar, ubicada en la parte vertical derecha de la sección catálogo (ver Figura 5.19), permite la integración de la producción de todos los productores. La presentación tendrá un esquema similar al usado en la presentación del catálogo de productos de forma independiente.



Figura 5.19 Interfaz de Integración de Producción

6. Pruebas y Resultados Preliminares

En este capítulo se presenta el ambiente y plan de pruebas que se preparó, para que la aplicación implementada sea probada. En la sección 5.2, se definió que la aplicación tendría dos instancias de prueba: la primera involucraría a cada subsistema de forma individual, y posteriormente se probaría la aplicación con sus subsistemas en forma conjunta.

Con estas consideraciones se planificó un set de pruebas para cada subsistema. Cada set está alineado con los casos de uso definidos en la sección 3.4. Con las pruebas de usabilidad listas, el siguiente paso fue formar un grupo de usuarios productores y clientes con los cuales se ejecuta la inspección de los subsistemas.

Abierta esta necesidad, se solicitó a la cooperativa que defina un representante que sea el encargado del seguimiento y validación de las funcionalidades del sitio Web. Por el lado del usuario cliente, se trabajó con clientes actuales de confianza de la cooperativa y clientes potenciales que tienen interés en usar el canal de comercio.

En la primera instancia de evaluación, las pruebas individuales de cada subsistema fueron realizadas con un número de usuarios controlado (ver Tabla 6.1). Este plan de pruebas permitió obtener paulatinamente los primeros defectos que se fueron registrando en el artefacto de control de cambios (ver Anexo 8).

Las observaciones y defectos obtenidos en la primera instancia, sirvieron para brindarle mayor madurez a la aplicación, también para afinar el nuevo plan de pruebas que consideraba la globalidad del sistema. Las pruebas globales se las dirigió a un FocusGroup (ver Tabla 6.14); los defectos y observaciones que se obtuvieron, al igual que en la instancia anterior se fueron registrando en el artefacto de control de cambios.

Estas actividades definen el flujo de proceso (ver Figura 6.1) que se ha llevado a cabo para la ejecución de pruebas de SGPO. Con este esquema se buscó brindarle mayor madurez y calidad a las funcionalidades deseadas en cada uno de los subsistemas. El proceso deja un camino abierto para la planificación de nuevas iteraciones de pruebas, que proporcionen nuevos incrementos a las características de SGPO. Cada plan de pruebas generado con el formato presentado en el Anexo 9 permite registrar observaciones que aportan a mejorar la aplicación.

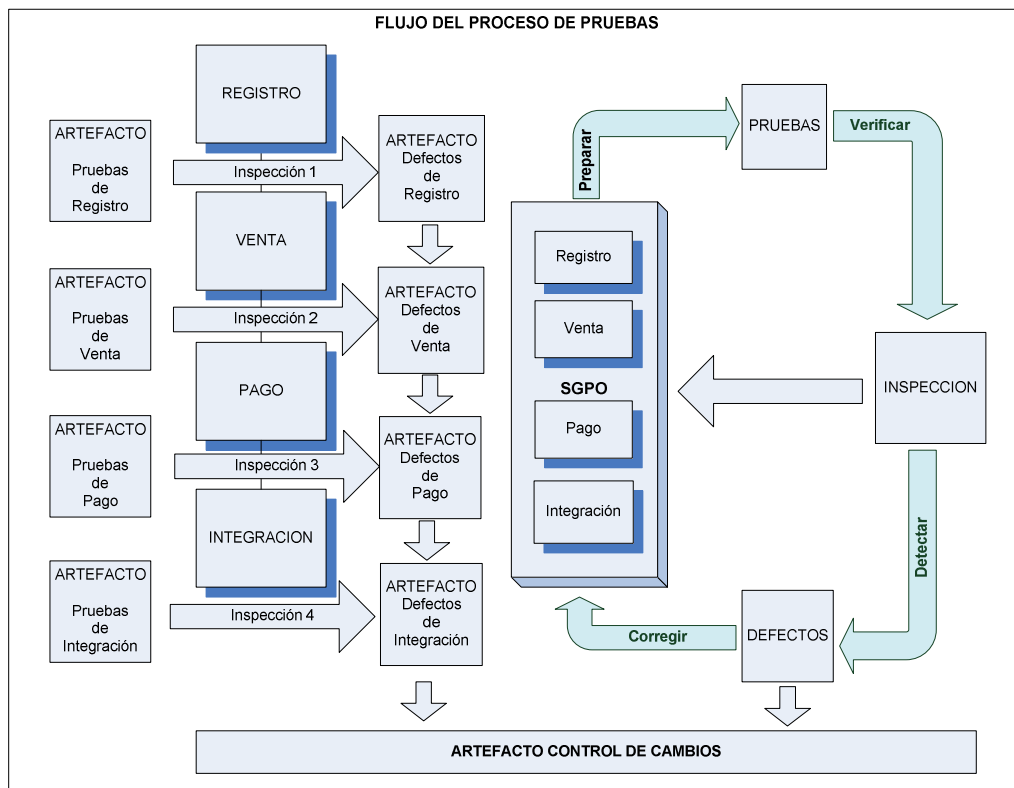


Figura 6.1 Flujo del Proceso de Pruebas

Posteriormente se sometió la aplicación a pruebas de rendimiento, con el fin de conocer el tiempo de respuesta que tiene SGPO, en comparación con un Website estable. También buscamos evaluar el comportamiento de la aplicación, cuando las peticiones de usuarios se incrementan. Para la ejecución de estas pruebas de usabilidad y rendimiento se publicó el Website en uno de los servidores de la Universidad de Chile cuya URL es: <http://anakena.dcc.uchile.cl:8080/mo/> (disponible hasta la finalización del periodo académico 2011). Los resultados de estas experiencias de prueba, fueron retroalimentado las funcionalidades de SGPO hasta obtener los resultados que se presentan en la sección 6.4.

6.1. Pruebas por Subsistema

La planificación y ejecución de estas pruebas buscó identificar la mayor cantidad de defectos en cada uno de los subsistemas, así como también inspeccionar el alineamiento de sus funcionalidades con las metas propuestas. Estas tareas aportaron al proceso de desarrollo de software, para que de forma incremental, se mejore el nivel de madurez de los subsistemas.

Para cada subsistema se preparó un listado de pruebas, cada una posee un código de identificación. Para clasificar y ordenar dichos códigos se definió la siguiente nomenclatura TXY-Z:

X= El número de iteración de prueba. En este caso, las pruebas por subsistema corresponden a la primera iteración.

Y= El tipo de subsistema Registro (R), Venta (V), Pago (P) o Integración (I).

Z= El número de prueba en el subsistema.

Posteriormente se determinó los usuarios encargados de ejecutar el set de pruebas (ver Tabla 6.1). Paulatinamente se mantuvo reuniones con cada uno de ellos, receptando y registrando las observaciones emitidas.

Tabla 6.1 Usuarios para Pruebas de Primera Instancia

ID	USUARIO	TIPO	DESCRIPCION
U1	Camilo Viera	Cliente de Cooperativa	Profesión: Turismo. Usuario con habilidades de Internet, nivel avanzado.
U2	Norma Marcalla	Representante de Cooperativa	Profesión: Agrónoma. Usuario con habilidades de Internet, nivel medio.

Las observaciones y/o defectos que se fueron obteniendo, se registraron en el artefacto de control de cambios (ver Anexo 8), éstas fueron aprovechadas para corregir la aplicación y también para afinar el plan de pruebas de la próxima iteración.

6.1.1. Inspección 1 - Subsistema de Registro

La inspección del subsistema de registro fue planificada considerando los casos de uso definidos para dicho subsistema (ver sección 3.4.2). Adicionalmente se consideró las funcionalidades implementadas hasta ese momento. El set de pruebas que se definió se lo presenta en la Tabla 6.2 y contiene su código de asignación para el segmento de prueba y un listado de acciones para el usuario.

Tabla 6.2 Lista de Pruebas para Inspección de Registro

INSPECCION REGISTRO		
CODIGO	PRUEBA	ACCIONES
T1R-1	Registrar Ubicación	Seleccionar provincia
		Buscar dirección en cartografía
		Fijar ubicación en cartografía
		Ingresar datos complementarios
		Registrar usuario o ingresar con datos de cuenta
		Guardar dirección
T1R-2	Registrar Usuario	Ingresar datos
		Guardar datos
T1R-3	Login Usuario	Ingresar datos
		Ejecutar datos

Se establecieron las primeras reuniones con cada uno de los usuarios mencionados anteriormente y con el efecto de las pruebas realizadas se consiguieron los resultados presentados en la Tabla 6.3.

Tabla 6.3 Resultado de Pruebas Inspección de Registro

CODIGO Prueba	RESULTADOS		DEFECTOS y/o OBSERVACIONES		
	Esperados	Obtenidos U1	Obtenidos U2	Usuario 1 Usuario 2	
T1R-1	Datos de dirección de usuario registrados.	<ul style="list-style-type: none"> Intento1: Sin éxito, olvida datos guardarlos. Intento2: Exitoso. 	<ul style="list-style-type: none"> Intento1: Sin éxito, olvida datos guardarlos. Intento2: Exitoso. 	<ul style="list-style-type: none"> Alto tiempo de respuesta al cargar tiles de cartografía, la primera vez. Después de ubicar e ingresar los datos de ubicación, el sistema solicita loguearse o registrarse, el usuario realiza esta acción pero olvida guardar la dirección inicialmente ubicada e ingresada. Si no fija la dirección inicialmente luego el usuario no tiene una opción donde pueda volver a fijar su dirección. No puede editar su ubicación. 	<ul style="list-style-type: none"> Con navegador Firefox, no se puede ejecutar la prueba. Pierde la ubicación de la casa en el mapa. No reconoce el DragDrop inmediatamente. Olvida guardar la dirección, después de haberle pedido el sistema registrarse. Tiles de cartografía tarda mucho en cargar la primera vez.
T1R-2	Datos de usuario registrados y sesión activa	Intento1: Exitoso.	Intento1: Exitoso.	<ul style="list-style-type: none"> En la confirmación del password no se válida que su valor sea igual al valor inicial. 	
T1R-3	Sesión activa	Intento1: Exitoso.	Intento1: Exitoso.	<ul style="list-style-type: none"> Luego de registrar sus datos de cuenta y posición ya no es necesario la ventana float de mapa. No puede consultar su ubicación, la consulta presenta todos los clientes. 	

Haciendo hincapié en los resultados relevantes, se detectó que la usabilidad del almacenamiento de dirección no era la adecuada, ya que el usuario olvidaba guardarla, a razón de interrumpir el proceso por el registro o login de su cuenta. Por otro lado se detectó problemas en el desempeño de la aplicación con el uso del navegador Firefox. Todos estos defectos encontrados, fueron registrados y clasificados en el artefacto de control de cambios. Este documento ha sido el que ha aportado al seguimiento y corrección de los defectos y observaciones que se han ido obteniendo con las pruebas.

Tabla 6.4 Tiempo de Pruebas Inspección de Registro

TIEMPO		
Estimado	Realizado U1	Realizado U2
3.5 min	5 min	9min

En cuanto al tiempo estimado de pruebas y el realizado, se obtuvieron los resultados presentados en la Tabla 6.4. Es importante mencionar que el nivel de habilidad de los usuarios en el manejo de Internet impactó directamente en el tiempo registrado. El usuario dos tenía menos costumbre en el uso de aplicaciones web y por esta razón el tiempo empleado para su registro se incrementó.

6.1.2. Inspección 2 - Subsistema de Venta

La inspección del subsistema de venta, fue planificada considerando los casos de uso definidos para dicho subsistema (ver sección 3.4.3). El set de pruebas que se definió se lo presenta en la Tabla 6.5. El listado de pruebas posee actividades con las cuales se puso a prueba la administración de la canasta de compras virtual de la aplicación, partiendo de la búsqueda del producto que el usuario desee.

Tabla 6.5 Lista de Pruebas Inspección de Venta

INSPECCION VENTA		
CODIGO	PRUEBA	ACCIONES
T1V-1	Buscar información geográfica	Buscar clientes
		Buscar productores
		Buscar frutas
		Buscar hortalizas
		Buscar cooperativa
T1V-2	Añadir producto (carro compras)	Seleccionar producto
		Ingresar cantidad deseada
		Aceptar selección
T1V-3	Modificar producto	Seleccionar producto
		Modificar cantidad deseada
		Aceptar modificación
T1V-4	Eliminar producto	Seleccionar producto
		Eliminar producto

Se realizó reuniones con cada uno de los usuarios implicados y con el efecto de las pruebas realizadas se obtuvieron los resultados presentados en la Tabla 6.6.

Tabla 6.6 Resultados de Pruebas Inspección de Venta

CODIGO Prueba	RESULTADOS			DEFECTOS y/o OBSERVACIONES	
	Esperados	Obtenidos U1	Obtenidos U2	Usuario 1	Usuario 2
T1V-1	<ul style="list-style-type: none"> Ubicación de clientes en cartografía. Ubicación de productores y catálogo de productos. Ubicación de productor(es) e información de sus detalles en la sección catálogo. Ubicación de productor(es) e información de sus detalles en la sección catálogo. Ubicación de productores y catálogo de productos. Ubicación de productores y catálogo de productos. 	Intento1: Exitoso	Intento1: Exitoso	<ul style="list-style-type: none"> En los resultados de la búsqueda, los markers no contienen información del cliente. En los resultados de la búsqueda, los markers no contienen información del productor. Buscar los productores más cercanos de acuerdo a la ubicación del usuario que haya realizado la búsqueda. Textos que contienen tildes no son presentados adecuadamente. 	<ul style="list-style-type: none"> En el resultado de una búsqueda al elegir un marker actualizar la información del catálogo con el productor correspondiente. Hacer notar que en el catálogo de productos, el usuario puede movilizar las imágenes a la izquierda o derecha. Reducir la velocidad en el cambio de imágenes de productos. Dar más realce a la posibilidad de selección de productores, para la carga de un nuevo catálogo de productos.
T1V-2	<ul style="list-style-type: none"> Presentación del detalle de producto. Generación de orden de venta. Nuevo ítem añadido a la canasta de compras (orden de venta) 	Intento1: Exitoso	Intento1: Exitoso	<ul style="list-style-type: none"> Dar formato de dos decimales a los números float que presentan información del precio del producto en la canasta de compras. Controlar que no permita añadir una cantidad de productos que supere el stock existente. No permitir abrir canasta de compras cuando esté logueado. 	<ul style="list-style-type: none"> Dar formato de dos decimales a los números float que presentan la información del precio del producto en la canasta de compras. Presentar precio con USD, el tipo de moneda es confuso. Dar más realce a la acción de abrir en la canasta, para que el usuario note que tiene la posibilidad de abrirla.

CODIGO	RESULTADOS			DEFECTOS y/o OBSERVACIONES	
	Esperados	Obtenidos U1	Obtenidos U2	Usuario 1	Usuario 2
T1V-3	Ítem modificado	Intento 1: Exitoso	Intento 1: Requiere de ayuda para encontrar la acción modificar. Intento2: Exitoso.	<ul style="list-style-type: none"> Mantener la canasta abierta al modificar la cantidad de un producto. Controlar el no ingreso de números negativos en el valor de la cantidad de productos a añadir o modificar, en la canasta de compras. Ordenar el listado de productos de la canasta de compras, de acuerdo al título que haya recibido un click. 	<ul style="list-style-type: none"> Habilitar la opción modificar en la misma canasta de compras. Dar más realce al link del nombre del producto, para que el usuario note que al cliquear puede modificar o eliminar.
T1V-4	Ítem eliminado	Intento 1: Exitoso	Intento 1: Exitoso		

Haciendo hincapié en los resultados relevantes, se evaluó como primordial la necesidad de incorporar información del productor en los markers resultantes de la búsqueda de información geográfica. En lo que respecta al mantenimiento de la canasta de compras, los resultados exigían correcciones de presentación y validación.

Los defectos encontrados, fueron registrados y clasificados en el artefacto de control de cambios. En cuanto al tiempo estimado de pruebas y el realizado, se obtuvieron los resultados presentados en la Tabla 6.7.

Tabla 6.7 Tiempo de Pruebas Inspección de Registro

TIEMPO		
Estimado	Realizado U1	Realizado U2
5 min	12 min	13 min

6.1.3. Inspección 3 - Subsistema de Pago

La inspección del subsistema de pago, fue planificada considerando los casos de uso definidos para dicho subsistema (ver sección 3.4.5). El set de pruebas que se definió se lo presenta en la Tabla 6.8.

Tabla 6.8 Lista de Pruebas Inspección de Pago

INSPECCION PAGO		
CODIGO	PRUEBA	ACCIONES
T1P-1	Confirmar canasta de compra	Ejecutar botón confirmación
T1P-2	Pagar compra	Ejecutar botón de pago PayPal
		Loguearse con usuario de prueba PayPal
		Aceptar el Pago

Se realizaron reuniones con cada uno de los usuarios implicados y con el efecto de las pruebas realizadas se obtuvieron los resultados presentados en la Tabla 6.9.

Tabla 6.9 Resultado de Pruebas Inspección de Pago

CODIGO Prueba	RESULTADOS		DEFECTOS y/o OBSERVACIONES	
	Esperados	Obtenidos U1	Obtenidos U2	Usuario 1 Usuario 2
T1P-1	Confirmación y detalle de la orden de venta	Intento1: Exitoso	Intento1: Exitoso	<ul style="list-style-type: none"> Registrar información de mi contacto.
T1P-2	<ul style="list-style-type: none"> Resumen de la orden de venta en interfaz PayPal. Confirmación en SGPO de pago realizado en PayPal. Control de Stock 	Intento1: Exitoso	<ul style="list-style-type: none"> Intento 1: Problemas al finalizar el pago, para presentar el mensaje de finalización de compra. Intento 2: Se refresca la página y se termina con éxito. 	<ul style="list-style-type: none"> Evitar la pantalla en blanco después haber hecho el pago.

Haciendo hincapié en los resultados, se valoró la necesidad de información del contacto del cliente, esta observación también hizo notar que antes de hacer el pago, era necesario permitir que el usuario valide su ubicación registrada.

Los defectos encontrados, fueron registrados y clasificados en el artefacto de control de cambios. En cuanto al tiempo estimado de pruebas y el realizado, se obtuvieron los resultados presentados en la Tabla 6.10.

Tabla 6.10 Tiempo de Pruebas Inspección de Pago

TIEMPO		
Estimado	Realizado U1	Realizado U2
3 min	5 min	7 min

6.1.4. Inspección 4 - Subsistema de Integración

La inspección del subsistema de pago, fue planificada considerando los casos de uso definidos para dicho subsistema (ver sección 3.4.4). El set de pruebas que se definió se lo presenta en la Tabla 6.11.

Tabla 6.11 Lista de Pruebas Inspección de Integración

INSPECCION INTEGRACION		
CODIGO	PRUEBA	ACCIONES
T1I-1	Integrar producción	Ejecutar botón integrar
T1I-2	Añadir producto de catálogo integrado	Seleccionar producto
		Ingresar cantidad deseada
		Aceptar selección
T1I-3	Modificar producto de catálogo integrado	Seleccionar producto
		Modificar cantidad deseada
		Aceptar modificación
T1I-4	Eliminar producto de catálogo integrado	Seleccionar producto
		Eliminar producto

Se realizaron reuniones con cada uno de los usuarios implicados y con el efecto de las pruebas realizadas se obtuvieron los resultados presentados en la Tabla 6.12.

Tabla 6.12 Resultado de Pruebas Inspección de Integración

CODIGO Prueba	RESULTADOS			DEFECTOS y/o OBSERVACIONES	
	Esperados	Obtenidos U1	Obtenidos U2	Usuario 1	Usuario 2
T11-1	<ul style="list-style-type: none"> Catálogo con cantidad de productos integrada. 	Intento1: Exitoso	Intento1: Exitoso	<ul style="list-style-type: none"> Permitir efectuar comparación entre cooperativas así como se puede hacer comparación entre productores. 	
T11-2	<ul style="list-style-type: none"> Presentación del detalle de producto. Generación de orden de compra. Nuevo ítem añadido a la canasta de compras (orden de compra) 	Intento1: Exitoso	Intento1: Exitoso		
T11-3	Ítem modificado	Intento1: Exitoso	Intento1: Exitoso		
T11-4	Ítem eliminado	Intento1: Exitoso	Intento1: Exitoso		

Los resultados de esta inspección dieron muestra de ser el subsistema más estable, se registró una observación con el fin de que sea considerada cuando el sistema recopile información de otra cooperativa.

Las observaciones encontradas, fueron registradas y clasificadas en el artefacto de control de cambios. En cuanto al tiempo estimado de pruebas y el realizado, se obtuvieron los resultados presentados en la Tabla 6.13.

Tabla 6.13 Tiempo de Pruebas Inspección de Integración

TIEMPO		
Estimado	Realizado U1	Realizado U2
6 min	10 min	12 min

6.2. Pruebas Globales

Después de la corrección de los defectos encontrados en las primeras inspecciones de las pruebas por subsistema, se procedió a la planificación del efecto de las pruebas de sistema. Para esto se consideró la experiencia de la preparación de los sets de pruebas que fueron usados en la inspección individual de cada subsistema. También se tomó en cuenta la práctica obtenida en el progreso de dichas pruebas, refinando los aciertos y falencias que se suscitaron.

Con esta retroalimentación se estableció el plan de pruebas para esta segunda instancia. Con este plan se buscó brindarle mayor madurez de la aplicación con un nuevo incremento.

Cada prueba posee un código de identificación, cuya nomenclatura (TXY-Z) es la misma que fue usada en las pruebas anteriores. En este caso X tomará el valor de segunda iteración.

Posteriormente se definió un FocusGroup (ver Tabla 6.14), para la ejecución del set de pruebas. Paulatinamente se mantuvo reuniones con cada uno de ellos, receptando y registrando las observaciones emitidas.

Tabla 6.14 Usuarios para Pruebas Inspección Global

ID	USUARIO	TIPO	DESCRIPCION
U1	Camilo Viera	Cliente de la Cooperativa	Profesión: Turismo. Usuario con habilidades de Internet, nivel avanzado.
U2	Norma Marcalla	Representante de Cooperativa	Profesión: Agrónoma. Usuario con habilidades de Internet, nivel medio.
U3	René Vela	Directivo de la Cooperativa	Profesión: Agrónomo. Usuario con habilidades de Internet, nivel medio.
U4	Andrea Idrovo	Cliente de la Cooperativa	Profesión: Comerciante Usuario con habilidades de Internet, nivel avanzado
U5	Jenny Zambrano	Cliente de la Cooperativa	Profesión: Parvularia Usuario con habilidades de Internet, nivel avanzado
U6	Bryan Vergara	Cliente de la Cooperativa	Profesión: Arquitecto Usuario con habilidades de Internet, nivel avanzado

Las observaciones y/o defectos que se fueron obteniendo, se registraron en el artefacto de control de cambios (ver Anexo 8), los registros fueron aprovechados para corregir la aplicación. Los cambios ejecutados iban brindando mejor usabilidad y calidad a la aplicación. A partir de esta iteración el proceso está abierto a la preparación y refinamiento de nuevos planes de pruebas, para continuar mejorando las funcionalidades de SGPO.

6.2.1. Plan de Pruebas Funcionales

La inspección del sistema en su globalidad, fue planificada considerando los planes de prueba de la anterior iteración. Estos planes fueron mejorados tomando en cuenta las falencias y necesidades que se identificaron en la primera instancia.

El set de pruebas que se definió se lo presenta en la Tabla 6.15. Contiene su código de asignación para el segmento de prueba, el subsistema al que pertenece, una secuencia de acciones para el usuario y variantes de acciones si la prueba lo amerita.

Tabla 6.15 Lista de Pruebas Inspección Global

CODIGO	SUBSISTEMA	PRUEBA	INSPECCION GLOBAL			
			ACCIONES VARIANTE 1	ACCIONES VARIANTE 2	ACCIONES VARIANTE 3	ACCIONES VARIANTE 4
T2R-1	REGISTRO	REGISTRAR USUARIO	Registrar Ubicación (definida)	Registrar datos inválidos de Usuario	Login Usuario	Con navegador Firefox
			Registrar datos de Usuario	Registrar Ubicación	Registrar Ubicación	
T2R-2	REGISTRO	MODIFICAR USUARIO	Logear Usuario	Logear Usuario		
			Modificar datos de Usuario	Modificar Ubicación		
T2V-1	VENTA	BUSCAR INFORMACION GEOGRAFICA (Clientes)	Buscar clientes			
			Revisar información clientes			
T2V-2	VENTA	BUSCAR INFORMACION GEOGRAFICA (productores)	Buscar productores			
			Revisar información productores			
			Elegir productores y revisar información en catálogo			
T2V-3	VENTA	BUSCAR INFORMACION GEOGRAFICA (frutas)	Buscar frutas			
			Revisar información productores			
			Elegir productores y revisar información en catálogo			
T2V-4	VENTA	BUSCAR INFORMACION GEOGRAFICA (hortalizas)	Buscar hortalizas			
			Revisar información productores			
			Elegir productores y revisar información en catálogo			

INSPECCION GLOBAL						
CODIGO	SUBSISTEMA	PRUEBA	ACCIONES VARIANTE 1	ACCIONES VARIANTE 2	ACCIONES VARIANTE 3	ACCIONES VARIANTE 4
T2V-5	VENTA	BUSCAR INFORMACION GEOGRAFICA (cooperativa)	Buscar cooperativa			
			Revisar información productores			
			Elegir productores y revisar información en catálogo			
T2V-6	VENTA	BUSCAR INFORMACION GEOGRAFICA (provincia)	Buscar provincia			
			Revisar información productores			
			Elegir productores y revisar información en catálogo			
T2V-7	VENTA	MANTENIMIENTO DE CANASTA DE COMPRAS	Añadir producto	Añadir producto con cantidad negativa	Añadir producto con cantidad que supere stock	Añadir producto de diferente productor
			Modificar producto	Modificar producto con cantidad negativa	Modificar producto con cantidad que supere stock	Modificar producto con cantidad que supere stock
			Eliminar producto	Eliminar producto	Eliminar producto	Eliminar producto
T2P-1	PAGO	PAGAR COMPRA	Confirmar canasta de compras			
			Pagar con PayPal			
			Loguearse con usuario de prueba paypal			
			Acceptar el Pago			

INSPECCION GLOBAL						
CODIGO	SUBSISTEMA	PRUEBA	ACCIONES VARIANTE 1	ACCIONES VARIANTE 2	ACCIONES VARIANTE 3	ACCIONES VARIANTE 4
T2I-1	INTEGRACION	INTEGRAR PRODUCCION	Ejecutar botón de integración			
T2I-2	INTEGRACION	MANTENIMIENTO DE CANASTA DE COMPRAS	Añadir producto	Añadir producto con cantidad negativa	Añadir producto con cantidad que supere stock	
			Modificar producto	Modificar producto con cantidad negativa	Modificar producto con cantidad que supere stock	
			Eliminar producto	Eliminar producto	Eliminar producto	

Se establecieron reuniones con cada uno de los usuarios definidos. Para el efecto de las pruebas realizadas, se hizo uso de la plantilla de trabajo presentada en el Anexo 9.

Analizando los resultados obtenidos y haciendo énfasis en los relevantes, se ve que después de las primeras iteraciones de pruebas el número de defectos, se reduce. Los subsistemas de registro y venta muestran mayor estabilidad en los resultados de pruebas.

Aparecen observaciones que implican la adhesión de nuevas actividades importantes, como por ejemplo: permitir la alternativa de búsqueda por texto, validar datos de ubicación y contactos antes de confirmar la orden de venta, reportar las ventas registradas de la semana, volver al stock inicial finalizada la semana.

Todos los defectos mencionados y demás encontrados, fueron registrados y clasificados en el artefacto de control de cambios.

Tabla 6.16 Tiempo de Pruebas Inspección Global

TIEMPO						
Estimado	Realizado U1	Realizado U2	Realizado U3	Realizado U4	Realizado U5	Realizado U6
25 min	30 min	40 min	40 min	35 min	45 min	30 min

En cuanto al tiempo estimado de pruebas y el realizado por los usuarios, se obtuvieron los resultados presentados en la Tabla 6.16. Es importante mencionar que el nivel de habilidad de los usuarios en el manejo de Internet impactó directamente en el tiempo registrado.

6.3. Pruebas de Rendimiento

El efecto de las pruebas de rendimiento buscaba analizar y medir el tiempo de respuesta que ofrece la estructura de la aplicación, a las acciones solicitadas por el usuario en la interfaz de presentación.

Para la obtención de esta información se utilizó el software Webserver Stress Tool 7, esta aplicación permite la simulación de un determinado número de usuarios solicitando, por protocolo HTTP una acción en una URL definida.

Como resultados relevantes se obtiene un informe de los errores encontrados, el tiempo de respuesta en milisegundos y la cantidad de bytes transmitidos en cada solicitud. Para la ejecución de esta prueba se eligieron dos URLs, la primera correspondiente a la aplicación SGPO y la segunda, a una aplicación estable y en producción, que serviría como punto de referencia (ver Tabla 6.17).

La selección de nuestra URL se la hizo, tomando en cuenta aquella acción que implica mayor cantidad de recursos, es así que se eligió una búsqueda de información geográfica que requiere consultar en tablas relacionadas.

Tabla 6.17 URLs para Pruebas de Rendimiento

SISTEMA	URL	DESCRIPCION
SGPO	http://anakena.dcc.uchile.cl:8080/mo/fruit.htm	Búsqueda de información geográfica
BBC	http://www.bbc.co.uk/news/technology/	Noticias internacionales

Para determinar el tiempo de respuesta promedio de la petición de una URL en la red, se precedió a registrar los tiempos de carga de cinco Websites diferentes (ver Tabla 6.18). Con estos resultados consideramos que el tiempo de respuesta de una URL es aceptable cuando la presentación de su información oscila entre tres y cinco segundos. Después de diez segundos de espera el usuario empieza a pensar en otras cosas y su cerebro tiene dificultad en retomar la actividad solicitada²⁴.

Tabla 6.18 URLs y su Tiempo de Respuesta

SISTEMA	URL	TIEMPO DE RESPUESTA (S)
BBC	http://www.bbc.co.uk/news/technology/	4,55
Ride City	http://ridethecity.com/	2,58
El Mercurio	http://diario.elmercurio.com/2011/07/12/_portada/index.htm	4,49
El Comercio	http://www.elcomercio.com.ec/	3,74
Flixxy	http://www.flixxy.com/	5,60
PROMEDIO		4,19

Se eligió la URL de la BBC, para que sea el punto de comparación del rendimiento de nuestra aplicación. Las pruebas iniciales fueron con diez usuarios haciendo peticiones constantes durante un minuto (ver Figura 6.2).

²⁴ Servicios Web USEIT 2010, Website Response Time Limits, USEIT, último acceso 02 Noviembre 2011 <<http://www.useit.com/alertbox/response-times.html>>

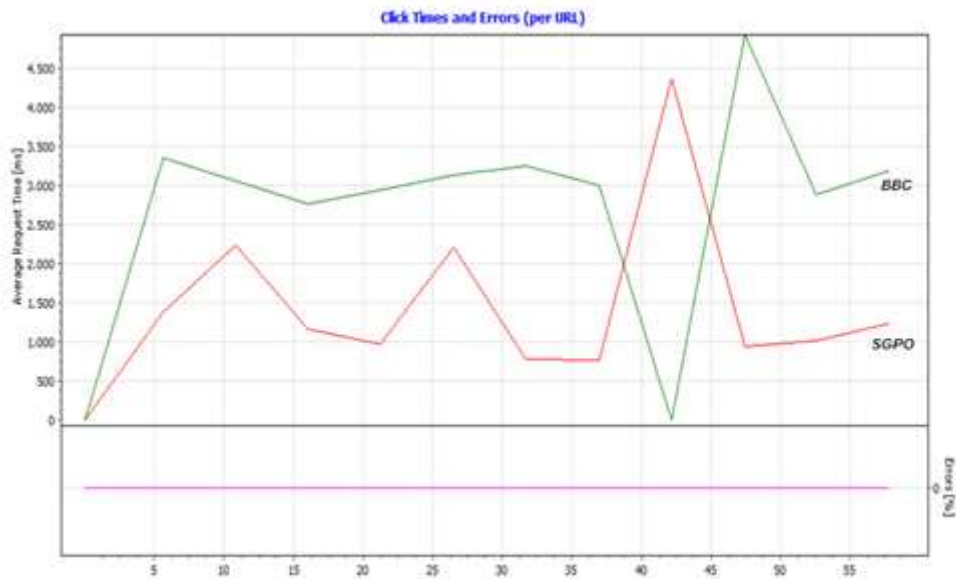


Figura 6.2 Estadísticas de Tiempo de Respuesta (10 usuarios)

Los resultados muestran que el rango de respuesta de ambas URLs está entre los tres y cinco segundos. Con un promedio de respuesta de 2.5 segundos para SGPO y 3.5 segundos para BBC, sin registro de errores, lo cual son tiempos aceptables.

El siguiente paso fue incrementar el número de usuarios que incurren en la solicitud de una acción en las URLs definidas. Con cincuenta usuarios haciendo peticiones constantes durante un minuto, los resultados en cuanto a su tiempo de respuesta fueron los presentados en la Figura 6.3.



Figura 6.3 Estadísticas de Tiempo de Respuesta (50 usuarios)

Haciendo interpretación de la gráfica, el eje de las X corresponde al transcurso del tiempo definido para las pruebas (un minuto), en el eje de las Y corresponde al tiempo de respuesta de la solicitud (milisegundos).

Los resultados muestran que el rango de respuesta de ambas URLs está entre los tres y cinco segundos. Con un promedio de respuesta de 3.5 segundos para SGPO y 3.8 segundos para BBC, sin registro de errores, lo cual son tiempos aceptables.

Finalmente se ejecutan las pruebas con un número de usuarios en el cual el rendimiento de SGPO ya no tiene un escenario aceptable. El resultado de las pruebas realizadas con la concurrencia de cien usuarios durante un minuto a las URLs definidas, fueron los presentados en la Figura 6.4.

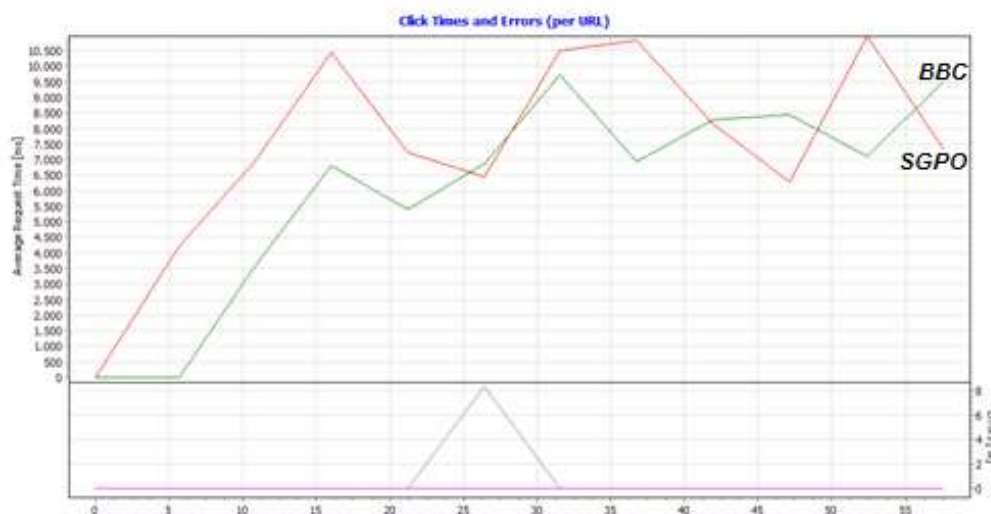


Figura 6.4 Estadísticas de Tiempo de Respuesta (100 usuarios)

Los resultados presentan que el rango de respuesta de ambas URLs está entre los seis y once segundos. Con un promedio de respuesta de 9,5 segundos para SGPO y 9 segundos para la BBC, sin registro de errores, lo cual son tiempos de respuesta poco aceptables para el usuario. Sin embargo se registra que la URL de la BBC tiene un mejor rendimiento que nuestra aplicación.

Es importante mencionar que para este análisis no se tomó en cuenta factores como la robustez del servidor de los Websites y la demanda externa a las URLs en el momento de las pruebas. Con este análisis se concluye que la aplicación presenta un comportamiento aceptable, con un número determinado de usuarios.

Sin embargo el incremento paulatino de los mismos podría dar origen a una amenaza para la aplicación. Así que es importante considerar el hardware y equilibrio de la carga de trabajo en la aplicación. Cuando se esté llegando a los límites advertidos, se deberá hacer un tuning de los queries usados en la capa de persistencia y/o refinar el procesamiento de las solicitudes con el uso threads.

6.4. Resultados Obtenidos

El mejoramiento de los defectos y observaciones encontrados en la aplicación, producto de la ejecución del plan de pruebas preparado, dieron lugar a un sistema con un nivel de madurez aceptable en sus funcionalidades.

Es importante mencionar que el flujo de pruebas preparado aporta al proceso de desarrollo de software, dejando abierta la posibilidad de ejecución de nuevas iteraciones que continúen mejorando la calidad del software.

Las metas funcionales de los subsistemas fueron establecidas en la sección 3.3. Después de las fases de diseño, implementación y pruebas las funcionalidades obtenidas han sido las siguientes:

6.4.1. Subsistema de Registro

Permite la recepción de datos de productores y clientes, información que se encuentra a la disponibilidad de la cooperativa. Las funcionalidades habilitadas en este subsistema son:

- *Administración de datos tradicionales de usuarios:* Consiente la creación y modificación de cuentas de usuario que poseen información de productores y clientes (actuales y potenciales).
- *Administración de datos geográficos de usuarios:* Permite la fijación y edición (en cartografía) de la posición geográfica del usuario; información que aporta a la obtención de su domicilio.
- *Identificación de cuenta:* El usuario accede al sistema usando el nombre y password con el que se registró, se valida los datos de cuenta y se genera una sesión de trabajo en la aplicación.

6.4.2. Subsistema de Venta

Permite que usuarios ejecuten búsquedas, dirigidas a encontrar información sobre la producción que posee la cooperativa. También establece un canal de comunicación entre el cliente y la cooperativa, por el cual se entrega información de una canasta de compras deseada. Las funcionalidades habilitadas son:

- *Búsqueda de información de la cooperativa:* Permite la búsqueda de información en las siguientes categorías: clientes, productores, frutas y hortalizas. Haciendo uso de OSM (OpenStreetMaps), a los resultados de la búsqueda se adiciona información geográfica.

- *Administración de canasta de compras:* Se permite la inserción, modificación y eliminación de productos que se ligan a una orden de compra de un usuario, ésta puede estar relacionada a uno o más productores.

6.4.3. Subsistema de Pago

Permite que usuarios que han preparado una canasta de compras, puedan hacer el pago online de la misma. Las funcionalidades habilitadas son las siguientes:

- *Confirmación de orden de compra:* El usuario puede revisar y confirmar su canasta de compras.
- *Pago Online con PayPal:* Verificada y confirmada la orden de compra, el usuario puede hacer el pago de forma online con una cuenta PayPal.

6.4.4. Subsistema de Integración

Permite acumular la cantidad de producción individual de los productores pertenecientes a la cooperativa. El usuario puede administrar su canasta de compras, en base a cantidades de stock acumuladas. Las funcionalidades habilitadas son:

- *Calcular la producción total de la cooperativa:* Se puede totalizar la cantidad de producción que pueden acumular los productores de la cooperativa.
- *Administración de canasta de compras:* Con cantidades integradas que superan el stock individual de un productor. Se permite la inserción, modificación y eliminación de productos vinculados a una orden de compra de un usuario.

7. Conclusiones y Trabajo a Futuro

Se ha implementado una aplicación Web que recopila información de clientes y producción orgánica de los productores de la Cooperativa ecuatoriana Probio. El sistema presenta la información con el apoyo de una aplicación geográfica y permite el comercio electrónico de los productos de la cooperativa. Por lo tanto, los objetivos planteados han sido alcanzados. SGPO posee una estructura Modelo-Vista-Controlador que soporta el almacén de información tipo hipertexto y geográfica, que de acuerdo a las pruebas de rendimiento se presentan al usuario con un tiempo de respuesta aceptable. El cliente y el productor entregan información a la aplicación que es usada en un contexto de tienda virtual, en el cual el cliente puede preparar una orden de compra y pagar la misma de forma online, a través de su cuenta PayPal.

Las funcionalidades de SGPO están distribuidas en subsistemas. Inicialmente se propuso elaborar un módulo independiente dedicado al fortalecimiento de la comunicación cliente-productor. Dos factores dificultaron esta implementación, el limitado tiempo y la perspectiva de la gran acogida de las redes sociales para potenciar un negocio. Se acuerda con la cooperativa el uso de estas herramientas, con el fin de promocionar los productos y URL de la aplicación en una red de contactos de amigos y familiares, así como también tener un espacio de atención al cliente, para recibir inquietudes y comentarios de clientes actuales o potenciales. Sin embargo, los subsistemas implementados en la aplicación poseen una comunicación cliente-productor implícita, ésta es la oferta y demanda de productos a través de la generación de una orden de venta en el sistema.

Con estas consideraciones y en base al análisis de las necesidades de la cooperativa, se acuerda la definición de los requisitos funcionales para cada subsistema. Cuatro son los subsistemas establecidos: Registro, Venta, Pago e Integración.

El plan de pruebas ejecutado se convierte en una parte importante del desarrollo de software, cada inspección realizada corresponde a una nueva iteración y sus resultados generan un nuevo incremento que mejora la madurez y calidad en la aplicación. Los defectos, observaciones y nuevas ideas, están registrados en el documento de control de cambios. Por razones de tiempo no se ha logrado cerrar todos los casos en su totalidad, pero son parte de nuestro trabajo a futuro.

Se acuerda con la cooperativa algunas definiciones de negocio importantes, por ejemplo la entrega de las compras registradas en SGPO no será inmediata, se las hará una vez por semana. El stock de cada producto corresponde a un valor constante semanal que el productor lo obtiene con su planificación de cultivo. Después de la entrega semanal, el stock de los productos en la aplicación adopta su estado inicial.

Para la recepción de los pagos online confirmados en PayPal, el acuerdo ha sido usar datos de una cuenta designada por la cooperativa, la cual es la responsable de la administración de los ingresos.

En cuanto al significado que tiene el presente trabajo para la cooperativa, éste radica en el fortalecimiento que la aplicación brinda a su canal de comercio. Ya que al momento sus ventas están dirigidas únicamente a clientes actuales.

El hecho de tener una aplicación tecnológica de compra en un medio público como es Internet. La cooperativa tiene mayor apertura de dirigirse a nuevos clientes potenciales con el fin de incrementar sus ventas. Considerando a la aplicación como una herramienta de promoción y marketing de sus productos.

Tiene también valor para la cooperativa, el hecho de tener una forma de pago online. Esto es considerado como una alternativa y comodidad para su negocio.

Por el lado del cliente, la importancia que tiene la aplicación, está en el tiempo que un usuario emplea para hacer sus compras en la tienda virtual. Al momento el promedio es de veinte minutos. En este tiempo el usuario puede registrarse, seleccionar y pagar productos del catalogo de nuestra aplicación, considerando que tenga una cuenta PayPal activa. Este tiempo es aceptable considerando el tiempo y esfuerzo que implica hacer la misma compra en una feria orgánica o supermercado.

También es importante para el cliente conocer la ubicación del origen de los productos. Este detalle brinda mayor confiabilidad para su adquisición, en el sentido de saber dónde está el vendedor.

En lo que respecta al trabajo a futuro, está la puesta en producción de SGPO. Buscamos usar un dominio comercial que pueda ser promocionado y comunicado. La puesta en producción se ha convertido en un trabajo a futuro, debido a que el procesamiento de la información que se reciba a través de la aplicación (registros de proveedores y clientes, órdenes de compra), requiere una sincronización con tareas complementarias de la cooperativa. Por ejemplo, es necesario coordinar el acopio y entrega de productos al cliente.

Para ello la cooperativa está armando un equipo de trabajo con el fin de mantener e implementar nuevas funcionalidades en SGPO. La visión de trabajo estará dirigida a obtener información del usuario, obtener y mejorar las siguientes métricas:

- El número de usuarios registrados por motivos de referencias de otros clientes y/o medios de comunicación.
- El número de ventas mensuales. En lo que respecta a las entregas directas, actualmente la cooperativa tiene un promedio de 30 a 40 pedidos semanales. Que la cooperativa pueda incrementar estos pedidos en un 70% o 100%, será considerado como un factor de éxito para nuestra aplicación.
- El número de ventas mensuales efectuadas con el subsistema de integración.

Luego de conseguir un desempeño de uso estable de SGPO, con usuarios de la cooperativa Probio y el FocusGroup definido, se buscará implantar la solución en otras cooperativas con características similares a Probio.

8. Bibliografía y Referencias

- [Bol05] Bolstad, Peter. "GIS Fundamentals: A first text on Geographic Information Systems", Second Edition. White Bear Lake, MN: Eider Press. 2005.
- [Esc02] Escalona, María José. "Ingeniería de Requisitos en Aplicaciones para la Web". Universidad de Sevilla, España, Diciembre 2002.
- [Far99] Farley James. "Building Enterprise Government Using OpenGIS Technology". Proc. of the Open GIS Seminar. Geospatial Information and Technology Association. 1999.
- [Har05] Havey, Michael. "Modelado de Procesos Esenciales del Negocio". O'Reilly. 2005.
- [Och09] Ochoa, Sergio. "Arquitecturas de Software Genéricas". Material del Curso "CC57J: Arquitectura de Software". Departamento de Ciencias de la Computación. Universidad de Chile, Chile, Junio 2009.
- [Ney03] Yefim, Natis. "Escenarios para Arquitecturas Orientadas a Servicios". Gartner, Inc., Estados Unidos, 2003.
- [Ros02] Rosenfeld, Louis, Morville, Peter. Information Architecture for the World Wide Web, Second Edition, O'Reilly Media, August 2002.
- [Som05] Sommerville, Ian. "Ingeniería de Software 7 Ed.". Pearson Education, S.A., Madrid, 2005.

ANEXO 1: Detalle de los Casos de Uso del Subsistema de Registro

En esta sección se presenta el detalle de los casos de uso incluidos en el Subsistema de Registro.

CU1 - Ingresar Datos de Usuario:

NOMBRE	INGRESAR DATOS DE USUARIO (CU1)
ACTORES	Cliente
ACTIVIDADES	Hacer Registro, Validar registro, Confirmar registro
SINOPSIS	Este CU comienza cuando el usuario elige registrarse. El sistema le presenta los datos que debe llenar. El sistema valida que los datos estén correctos. Si todo está correcto y el usuario confirma, los datos se guardan.
CURSO TIPICO DE EVENTOS	1. Usuario ingresa datos personales. 2. Sistema valida datos ingresados. 3. Usuario confirma registro
EXTENSIONES	2a. Datos existentes
	1. Mensaje de dato existente
	2. Retoma 1
	2b. Inconsistencia de Datos
	1. Mensaje de dato erróneo
	2. Retoma 1
	2c. Datos faltantes
	1. Mensaje de dato faltante
	2. Retoma 1

CU3 - Identificar Usuario:

NOMBRE	IDENTIFICAR USUARIO (CU3)
ACTORES	Usuario
ACTIVIDADES	N/A
SINOPSIS	Localiza un cliente registrado.
CURSO TIPICO DE EVENTOS	1. Usuario ingresa datos personales. 2. Sistema valida datos provistos. 3. Sistema localiza usuario y crea sesión
EXTENSIONES	2a. Sistema no encuentra usuarios que coincidan con datos provistos
	1. Sistema notifica el error
	2. Retoma 1

CU5 - Ingresar Ubicación de Usuario:

NOMBRE	INGRESAR UBICACION DE USUARIO(CU5)
ACTORES	Usuario
ACTIVIDADES	Hacer Registro, Validar registro, Confirmar registro
SINOPSIS	Este CU comienza cuando el usuario elige ubicarse. El sistema le presenta la cartografía para elegir su ubicación y le pide confirmar su posición. Los datos se guardan. Para el efecto de este caso de uso el usuario debe estar logueado.
CURSO TIPICO DE EVENTOS	1. Usuario elige posición
	2. Sistema pide confirmar posición.
	3. Sistema pide identificación de Usuario (CU3)
	4. Usuario se loguea
	5. Usuario confirma posición
EXTENSIONES	2a. No confirma
	1. Vuelve a cartografía
	2. Retoma 1
	4a. Usuario no registrado
	1. Incluir Ingresar Datos de Usuario (CU1)

ANEXO 2: Detalle de los Casos de Uso del Subsistema de Venta

En esta sección se presenta el detalle de los casos de uso incluidos en el Subsistema de Venta.

CU7 - Identificar categoría de búsqueda:

NOMBRE	IDENTIFICAR CATEGORIA DE BUSQUEDA
ACTORES	Productor, Cliente, Cooperativa
ACTIVIDADES	Sistema presenta categorías, elige categoría/tipo y obtiene código de categoría de búsqueda.
SINOPSIS	Este CU comienza cuando el usuario desea elegir una categoría de búsqueda. El sistema le presenta las categorías y tipos de productos. El sistema acepta la opción elegida por el usuario.
CURSO TIPICO DE EVENTOS	1. Usuario elige categoría y tipo. 2. Sistema recibe código de búsqueda seleccionado.

CU8 – Buscar geográficamente categoría:

NOMBRE	BUSCAR GEOGRAFICAMENTE CATEGORIA
ACTORES	Cliente, Productor, Cooperativa
ACTIVIDADES	Recibir código de búsqueda, filtrar datos de búsqueda, buscar las posiciones geográficas, presentar posiciones seleccionadas.
SINOPSIS	Este CU comienza cuando el usuario ha elegido la categoría de búsqueda. De acuerdo al código el sistema filtra los datos necesarios. Busca las posiciones geográficas de los datos filtrados. Finalmente presenta la posición en cartografía.
CURSO TIPICO DE EVENTOS	1. Sistema acepta código de búsqueda. 2. Sistema filtra los datos necesarios 3. Sistema busca posiciones geográficas de los datos 4. Sistema presenta datos en cartografía.
EXTENSIONES	2a. Datos inexistentes 1. Mensaje de dato inexistente (Retoma 1)

CU9 – Obtener productores y ofertas de acuerdo a categoría:

NOMBRE	OBTENER PRODUCTORES DE ACUERDO A CATEGORIA
ACTORES	Cliente, Productor, Cooperativa
ACTIVIDADES	Determinar si es una oferta. Por productor obtener la cantidad de productos de la categoría.
SINOPSIS	Este CU comienza cuando el usuario ha elegido la categoría de búsqueda. El sistema receipta el código de búsqueda. El sistema filtra los datos necesarios. Por productor busca la cantidad de productos de la categoría. Presenta datos.
CURSO TIPICO DE EVENTOS	1. Sistema receipta código de búsqueda
	2. Sistema filtra los datos necesarios
	3. Sistema busca cantidad de producción por categoría de los productores
	4. Sistema presenta datos
EXTENSIONES	2a. Datos inexistentes
	1. Mensaje de dato inexistente
	2. Retoma 1

CU10 – Añadir producto a canasta de compras:

NOMBRE	AÑADIR PRODUCTO A CANASTA DE COMPRAS
ACTORES	Cliente
ACTIVIDADES	Seleccionar Categoría, seleccionar producto, ingresar cantidad, validar stock, añadir a la canasta de compras.
SINOPSIS	Este CU comienza cuando el usuario elige la opción AÑADIR CARRO. El sistema valida la disponibilidad de stock del producto seleccionado en la categoría correspondiente. El sistema añade la cantidad del producto a la canasta de compras.
CURSO TIPICO DE EVENTOS	1. Usuario ingresa cantidad de producto.
	2. Sistema valida disponibilidad de stock.
	3. Sistema añade el producto a la canasta de compras.
EXTENSIONES	2a. Datos existentes
	1. Mensaje de dato existente
	2. Retoma 1
	2b. Inconsistencia de Datos
	1. Mensaje de dato erróneo
	2. Retoma 1
	2c. Datos faltantes
	1. Mensaje de dato faltante
	2. Retoma 1

CU12 – Retirar producto de canasta de compras:

NOMBRE	RETIRAR PRODUCTO DE CANASTA DE COMPRAS
ACTORES	Cliente
ACTIVIDADES	Seleccionar producto, eliminar producto de canasta, actualizar canasta de compras.
SINOPSIS	Este CU comienza cuando un usuario selecciona un producto y elige la opción Eliminar. El sistema retira el producto de la orden de compra y actualiza la canasta.
CURSO TIPICO DE EVENTOS	1. Usuario selecciona producto.
	2. Usuario elimina producto.
	3. Sistema elimina producto de la orden de compra.

ANEXO 3: Detalle de los Casos de Uso del Subsistema de Integración

En esta sección se presenta el detalle de los casos de uso incluidos en el Subsistema de Integración.

CU14 – Sumar producción de productores:

NOMBRE	SUMAR PRODUCCION DE PRODUCTORES
ACTORES	Productor, Cooperativa, Cliente
ACTIVIDADES	Identificar catálogo de productos, obtener la cantidad de producción individual de cada productor, acumular y totalizar la cantidad de cada producto que la cooperativa puede obtener.
SINOPSIS	Este CU comienza cuando el usuario elige la opción Integrar. El sistema identifica el catalogo de productos de la cooperativa. El sistema, por cada producto obtiene la cantidad que posee del mismo cada productor de la cooperativa. Los valores acumulados son entregados a la interfaz de usuario.
CURSO TIPICO DE EVENTOS	1. Usuario elige integrar.
	2. Sistema identifica catalogo de productos.
	3. Sistema obtiene cantidad de producción de cada productor.
	4. Sistema acumula y entrega total.

ANEXO 4: Detalle de los Casos de Uso del Subsistema de Pago

En esta sección se presenta el detalle de los casos de uso incluidos en el Subsistema de Pago.

CU15 - Confirmar orden de venta:

NOMBRE	CONFIRMAR ORDEN DE VENTA
ACTORES	Cliente
ACTIVIDADES	Confirmar canasta de compras, cambiar estado de la orden de compra, presentar el resumen de la compra.
SINOPSIS	Este CU comienza cuando el usuario elige la opción Confirmar Orden de Venta. El sistema identifica la orden de venta y cambia su estado a confirmada. El sistema presenta un resumen de la orden de compra.
CURSO TIPICO DE EVENTOS	1. Usuario elige Confirmar Orden de venta. 2. Sistema identifica el código de la venta y cambia de estado. 3. Sistema entrega toda la información de la orden de venta.
EXTENSIONES	n/a

CU16 – Pagar orden de venta:

NOMBRE	PAGAR ORDEN DE VENTA
ACTORES	Cliente, Servicios PayPal
ACTIVIDADES	Confirmar pago PayPal, establecer conexión con servicios PayPal, enviar orden de compra, efectuar pago, recibir confirmación de pago.
SINOPSIS	Este CU comienza cuando el usuario ha ejecutado la acción de pago con PayPal. Receptada la petición, el sistema establece conexión con los servicios web de PayPal. Con un enlace abierto se procede al envío de los datos de la orden de compra. El usuario efectúa el pago con su tarjeta de crédito. Al finalizar el sistema recepta la confirmación del pago.
CURSO TIPICO DE EVENTOS	1. Usuario confirma pago con PayPal. 2. Sistema establece conexión con servicios web PayPal 3. Sistema envía los datos de la orden venta. 4. Usuario efectúa pago de la orden. 5. Sistema recibe confirmación de pago.
EXTENSIONES	4a. Pago no realizado 1. Se redirección a la interfaz principal 2. Retoma 1

CU17 – Controlar stock:

NOMBRE	CONTROLAR STOCK
ACTORES	Servicios PayPal
ACTIVIDADES	Recibir confirmación de pago, cambiar estado de la orden de compra, reducir stock de productos.
SINOPSIS	Este CU comienza cuando el sistema ha receptado la confirmación de pago de la orden. Luego el sistema identifica los productos y su cantidad, para el control de stock correspondiente.
CURSO TIPICO DE EVENTOS	1. Sistema recepta confirmación de pago. 2. Sistema identifica la cantidad de los productos de la orden 3. Sistema reduce el stock en el catálogo de productos.
EXTENSIONES	n/a

ANEXO 5: Productores Ecuatorianos de la Cooperativa Probio

En esta sección se presenta el listado de los productores y su información de contacto.

PRODUCTOR	EDAD	CONTACTO	MAIL
José Jiménez (JJ)	55	32266757	n/a
Edison Jacome (EJ)	32	84743630	n/a
René Vela (RV)	39	32716214	kurisisa9@yahoo.com
Luis Marcalla (LM)	42	32716014	n/a
Hilda Sánchez (HS)	50	32716204	analucia_garcia@yahoo.es
Patricia Pallasca (PP)	32	32716344	luisgomezalbert@hotmail.com
Martha Quizahuano (MQ)	55	32716339	maryisabel-marc@hotmail.com
Carmen Amelia Tigse (CT)	40	98336042	n/a
Carlos Pallasca (CP)	48	96097807	n/a

ANEXO 6: Listado de Necesidades y Funcionalidades de SGPO

En esta sección se presenta el análisis de las necesidades de la cooperativa, conjuntamente con la definición de sus prioridades.

a) Lista de necesidades ordenadas de acuerdo a su prioridad:

N°	NECESIDAD/ATRIBUTO	RELACION	BENEFICIO	ESFUERZO	RIESGO	IMPACTO ARQUITECTONICO	CALIF.
1	Buscar, localizar, seleccionar, comprar y pagar productos orgánicos	1	A	A	A	A	12
2	Cooperación entre productores, a través de una red social laboral	5	A	A	A	A	12
3	Compartir información entre productores y clientes en relación con oferta y demanda de productos orgánicos	5	A	A	A	A	12
4	Localizar geográficamente a los productores	1	A	M	A	M	11
5	Registrar usuarios y permitir su interacción con los mapas	3	M	A	A	M	11
6	Integrar y organizar a los productores, para el efecto de toma de decisiones	5	M	A	A	A	11
7	Calcular la cantidad de producción individual y agrupada	5	M	A	A	A	11
8	Registrar su ubicación geográfica	3	A	M	M	M	9
9	Publicar la demanda que tenga el cliente	6	M	M	M	M	8
10	Conocer demandas de clientes finales	6	M	M	M	M	8
11	Dar visibilidad a los productos orgánicos	1	M	M	B	M	6
12	Contactar al productor	2	M	B	B	B	5
13	Obtener contactos del productor	2	M	B	B	B	5
14	Publicar contactos del cliente	4	M	B	B	B	5
15	Conocer los productores de su comunidad	5	B	M	B	M	5
16	Contactar clientes	4	M	B	B	B	5
17	Obtener contactos del cliente	4	M	B	B	B	5
18	Obtener contactos del productor	2	B	B	B	B	4

CLIENTES
COOPERATIVA
TODOS

B	BAJO=1
M	MEDIO=2
A	ALTO=3

b) Lista de necesidades filtradas

N°	NECESIDAD/ATRIBUTO	USUARIO	BENEFICIO	ESFUERZO	RIESGO	IMPACTO ARQUITECTONICO	CALIF.
1	Buscar, localizar, comprar y pagar productos orgánicos	Cliente	A	A	A	A	12
2	Registrar y permitir fijar su ubicación geográfica por si mismo	Cliente/Cooperativa	M	A	A	M	11
3	Calcular la cantidad de producción individual y agrupada	Cliente/Cooperativa	M	A	A	A	11
4	Conocer demandas de clientes finales	Cooperativa	M	M	M	M	8
5	Contactar al productor	Cliente/Cooperativa	M	B	B	B	5
6	Contactar clientes	Cooperativa	M	B	B	B	5
7	Conocer los productores de su comunidad	Cooperativa	B	M	B	M	5

c) Lista de de funcionalidades por subsistema

PC°	PAQUETES DE CARACTERISTICAS	SUBSISTEMA	BENEFICIO	ESFUERZO	RIESGO	IMPACTO ARQUITECTONICO	CALIF.	TOTAL
1	Interacción de clientes con cartografía.	Subsistema de Registro	A	A	M	M	10	
2	Administración geográfica de productores y clientes	Subsistema de Registro	A	A	M	M	10	27
3	Administración tradicional de productores y clientes	Subsistema de Registro	M	A	B	M	7	
4	Búsqueda de productos	Subsistema de Venta	A	A	A	A	12	
5	Administración de Canasta de Compras	Subsistema de Venta	A	A	A	A	12	24
6	Pago de canasta	Subsistema de Pago	A	A	A	B	12	12
7	Unificación de producción	Subsistema de Integración	M	A	B	B	7	7

	CLIENTES
	COOPERATIVA
	TODOS

B	BAJO=1
M	MEDIO=2
A	ALTO=3

ANEXO 7: Productos de la Cooperativa Probio

Listado de productos comercializados por la Cooperativa Probio.

PRODUCTO	CANT. (JJ)	CANT. (EJ)	CANT. (RV)	CANT. (LM)	CANT. (HS)	CANT. (PP)	CANT. (MQ)	CANT. (CP)	CANT. (CT)	TOTAL (K)
Aguacate	0	0	100	0	0	0	0	0	0	100
Babacos	90	0	0	0	0	0	0	0	0	90
Broccoli	0	0	100	40	10	10	5	5	25	195
Cebolla blanca	50	0	50	40	5	10	0	10	8	173
Cebolla Paitaña	50	0	50	10	4	5	5	10	8	142
Cebolla perla	0	0	50	10	4	5	5	10	8	92
Cebollín	0	0	10	2	0	0	0	1	0	13
Col Bruselas	0	0	10	0	0	0	5	4	6	25
Col Milán	0	0	50	0	0	5	5	7	8	75
Col morada	0	0	50	0	5	10	5	0	20	90
Col verde	0	0	50	0	3	10	7	0	20	90
Coliflor	0	0	100	40	10	10	0	5	20	185
Espinaca	0	0	50	10	3	3	0	5	5	76
Lechuga crespa roja maravilla de verano	0	0	100	70	0	0	0	0	0	170
Lechuga crespa roja red salad bowl	0	0	50	0	0	0	7	10	0	67
Lechuga crespa verde salad bowl	0	0	100	0	0	0	0	0	0	100
Lechuga crespa verde simpson	0	0	100	0	5	15	7	10	15	152
Lechuga criolla	70	0	300	70	10	0	7	7	8	472
Lechuga romana	0	0	20	70	0	0	7	0	15	112
Nabo chino	50	0	100	20	10	5	6	4	17	212
Papa nabo	0	0	50	0	10	0	10	5	0	75
Rábano blanco	0	0	30	0	0	0	0	6	5	41
Rábano rojo	10	0	100	70	5	5	7	6	10	213
Remolacha	20	0	50	40	10	7	3	10	10	150
Romanesco	0	0	50	20	5	10	5	5	25	120
Taxo	5	0	0	0	6	0	0	0	0	11
Tomate de árbol	0	900	0	0	10	0	0	0	0	910

PRODUCTO	CANT. (JJ)	CANT. (EJ)	CANT. (RV)	CANT. (LM)	CANT. (HS)	CANT. (PP)	CANT. (MQ)	CANT. (CP)	CANT. (CT)	TOTAL (K)
Tomate riñón	0	0	500	0	0	0	0	0	0	500
Zanahoria fina	4	0	300	50	0	20	4	5	25	408
Zuquini	0	0	0	5	10	0	0	0	30	45

PRODUCTOR
José Jimenez (JJ)
Edison Jacome (EJ)
Rene Vela (RV)
Luis Marcalla (LM)
Hilda Sanchez (HS)
Patricia Pallasca (PP)
Martha Quizahuano (MQ)
Carlos Pallasca (CP)
Carmen Amelia Tigse (CT)

ANEXO 8: Artefacto de Control de Cambios

Listado de defectos encontrados en la fase de pruebas (instancia uno e instancia dos).

DEFECTOS						
COD.	COD. PRUEBA	DESCRIPCION	TIPO	NIVEL	ESTADO	USUARIO
D1	T1R-1	Alto tiempo de respuesta al cargar por primera vez, los tiles de cartografía.	P	M	R	U1, U2
D2	T1R-1	Después de ubicar e ingresar los datos de ubicación, el sistema solicita loguearse o registrarse, el usuario realiza esta acción, pero olvida guardar los datos de dirección inicialmente entregados.	U	A	R	U1, U2
D3	T1R-1	Si no fija la dirección inicialmente luego el usuario no tiene una opción donde pueda volver a fijar su dirección	U	M	R	U1
D4	T1R-1	No puede editar su ubicación.	F	M	R	U1
D5	T1R-2	En la confirmación del password no válida que el Password inicial sea igual al de confirmación.	E	M	R	U1
D6	T1R-3	Luego de registrar sus datos de cuenta y posición ya no es necesaria la ventana float de mapa.	U	B	R	U1
D7	T1R-3	No puede consultar su ubicación, la consulta presenta todos los clientes.	F	B	R	U1
D8	T1R-1	Con navegador Firefox, no se puede ejecutar la prueba.	E	A	R	U2
D9	T1R-1	Usuario pierde la ubicación de la casa en el mapa.	U	M	R	U2
D10	T1R-1	No reconoce el DragDrop inmediatamente.	U	M	R	U2
D11	T1V-1	En los resultados de la búsqueda, los markers no contienen información del cliente.	U	A	R	U1
D12	T1V-1	En los resultados de la búsqueda, los markers no contienen información del productor.	U	A	R	U1

COD.	COD. PRUEBA	DESCRIPCION	TIPO	NIVEL	ESTADO	USUARIO
D13	T1V-1	Buscar los productores más cercanos de acuerdo a la ubicación del usuario que haya realizado la búsqueda.	F	B	R	U1
D14	T1V-1	Textos que contienen tildes no son presentados adecuadamente.	D	A	R	U1
D15	T1V-2	Dar formato de dos decimales a los números float que presentan la información del precio del producto en la canasta de compras.	D	M	R	U1, U2
D16	T1V-2	Controlar que no permita añadir una cantidad de productos que supere el stock existente.	E	A	R	U1
D17	T1V-2	No permitir abrir canasta de compras cuando no esté logueado	U	B	R	U1
D18	T1V-3	Mantener la canasta abierta al modificar la cantidad de un producto.	U	M	R	U1
D19	T1V-3	Validar que no ingrese números negativos en la cantidad de productos a añadir o modificar, en la canasta de compras.	E	A	R	U1
D20	T1V-3	Ordenar el listado de productos de la canasta de compras, de acuerdo al título que haya recibido un click.	F	B	R	U1
D21	T1V-1	En el resultado de una búsqueda al elegir un marker actualizar la información del catálogo con el productor correspondiente.	U	M	R	U2
D22	T1V-1	Hacer notar que en el catálogo de productos, el usuario puede movilizar las imágenes a la izquierda o derecha.	U	B	R	U2
D23	T1V-1	Reducir la velocidad en el cambio de imágenes de productos.	U	B	R	U2
D24	T1V-1	Dar más realce a la posibilidad de selección de productores, para la carga de un nuevo catálogo de productos.	U	A	R	U2
D26	T1V-2	Presentar precio con USD, el tipo de moneda es confuso.	D	B	R	U2

COD.	COD. PRUEBA	DESCRIPCION	TIPO	NIVEL	ESTADO	USUARIO
D27	T1V-2	Dar más realce a la acción de abrir en la canasta, para que el usuario note que tiene la posibilidad de abrirla.	U	B	R	U2
D28	T1V-3	Habilitar la opción modificar en la misma canasta de compras.	U	M	R	U2
D29	T1V-3	Dar más realce al link del nombre del producto, para que el usuario note que al clickear puede modificar o eliminar.	D	A	R	U2
D30	T1P-1	Evitar la pantalla en blanco después haber hecho el pago.	U	M	R	U2
D31	T1I-1	Permitir efectuar comparación entre cooperativas así como se puede hacer comparación entre productores.	U	M	R	U1

Simbología:

TIPO DE DEFECTO	DESCRIPCION
P	Performance
U	Usabilidad
F	Funcionalidad Extra
E	Error de validación

ESTADO DE CORRECCION	DESCRIPCION
R	Registrado
A	Analizando
C	Corregido

NIVEL DE SEVERIDAD	DESCRIPCION
A	Alta
M	Media
B	Baja

ANEXO 9: Plantilla de Plan de Pruebas

Formato del plan de pruebas utilizado en la segunda instancia de la fase de evaluación. Este documento puede ser refinado para ser usado en otras iteraciones.

USUARIO:	FECHA:		Prueba #:			
	Hora inicio:		Hora fin:			
RESULTADOS						
CODIGO PRUEBA	Esperados	Obtenidos (Variante 1)	Obtenidos (Variante 2)	Obtenidos (Variante 3)	Obtenidos (Variante 4)	DEFECTOS
T2R-1	<ul style="list-style-type: none"> Datos del usuario y su ubicación guardados en Base de Datos. Sesión activa en aplicación. 					
T2R-2	<ul style="list-style-type: none"> Datos de usuario y/o ubicación modificados en Base de Datos 					
T2V-1	<ul style="list-style-type: none"> Ubicación de clientes en cartografía. 					
T2V-2	<ul style="list-style-type: none"> Ubicación de productor(es). Información de sus detalles en la sección catálogo. 					

T2V-3	<ul style="list-style-type: none"> • Ubicación de productor(es) • Información de sus detalles en la sección catálogo. 						
T2V-4	<ul style="list-style-type: none"> • Ubicación de productor(es). • Información de sus detalles en la sección catálogo. 						
T2V-5	<ul style="list-style-type: none"> • Ubicación de productor(es). • Información de sus detalles en la sección catálogo. 						
T2V-6	<ul style="list-style-type: none"> • Ubicación de productor(es). • Información de sus detalles en la sección catálogo. 						

T2I-2	<ul style="list-style-type: none"> • Generación de orden de compra. • Nuevo ítem añadido a la canasta de compras (orden de compra) • Ítem modificado • Ítem Eliminado 					
-------	---	--	--	--	--	--