

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR  
FACULTAD DE INGENIERÍA  
ESCUELA DE SISTEMAS



DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN SISTEMAS

“DESARROLLO, IMPLEMENTACIÓN E IMPLANTACIÓN DEL  
SISTEMA DE ADMINISTRACIÓN DE CLIENTES,  
PROVEEDORES, PRODUCTOS Y GESTIÓN DE CONTRATOS  
PARA EMPRESA INTERMEDIARIA DE PRODUCTOS  
QUÍMICOS, QUITO ECUADOR”

SANTIAGO ANDRÉS BENÍTEZ AGUILAR

DIRECTOR: ING. FABIÁN DE LA CRUZ

QUITO, 2013

Esta disertación de grado está dedicada especialmente a mis padres y mis hermanas quienes me han apoyado en todo momento.

## Índice

1. MARCO TEÓRICO .....	5
1.1 Extreme Programming .....	5
1.1.1 Introducción a Extreme Programming .....	5
1.1.2 Valores y Principios del Extreme Programming .....	6
1.1.3 Actividades del Extreme Programming (2) .....	7
1.2 Patrón Modelo Vista Controlador .....	10
1.2.1 Modelo .....	10
1.2.2 Vista .....	10
1.3 Aplicaciones Web .....	11
1.3.1 HTML .....	12
1.3.2 CSS .....	12
1.3.3 Navegadores Web .....	13
1.4 Herramientas .....	15
1.4.1 PHP .....	15
1.4.2 Javascript .....	15
1.4.3 MySQL .....	17
1.4.4 Symfony .....	18
1.4.5 Apache .....	19
1.4.6 Netbeans PHP .....	19
1.4.7 PowerDesigner .....	20
1.4.8 phpMyAdmin .....	21
2. DISEÑO .....	22
2.1 Descripción de la aplicación .....	22
2.2 Levantamiento de requerimientos .....	23
2.3 Diagrama de Procesos .....	24
2.3.1 Pedidos .....	24
2.3.2 Compras .....	26
2.3.3 Créditos .....	27
2.3.4 Pagos .....	28
2.4 Diagrama de Clases .....	29

2.5 Casos de uso .....	30
2.5.1 Diagrama de casos de uso a nivel general.....	30
2.5.2 Diagramas de casos de uso a detalle.....	31
3. BASE DE DATOS .....	43
3.1 Diccionario de Datos.....	45
4. PRUEBAS E IMPLANTACIÓN .....	49
4.1 Fase de Pruebas.....	49
4.2 Fase de Implantación .....	55
4.2.1 Configuración del servidor .....	56
4.2.2 Configuración de la base de datos del servidor y seguridades .....	56
4.2.3 Sitio de Pruebas y Producción .....	57
4.2.4 Tips de soporte del sistema.....	58
5. CONCLUSIONES Y RECOMENDACIONES .....	60
5.1 Conclusiones.....	60
5.2 Recomendaciones .....	61
Glosario .....	64
Trabajos citados .....	68
Agradecimientos .....	69

## **1. MARCO TEÓRICO**

Este capítulo presenta los conceptos fundamentales que se ven involucrados directamente al desarrollar una aplicación web. Conceptos, metodologías, mecanismos, patrones y herramientas que serán explicados para tener una idea clara de como se desarrolló el proyecto.

### **1.1 Extreme Programming**

#### **1.1.1 Introducción a Extreme Programming**

El Extreme Programming es una metodología con enfoque de ingeniería de software<sup>1</sup>, con el propósito de optimizar y mejorar los procesos de desarrollo de programas. Esta metodología nació debido a la necesidad de tener una guía y normas, que a través de la retroalimentación, permita optimizar tiempos, costos y recursos en general, de tal manera que el producto final sea de la mejor calidad posible. “La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad” (1).

“Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos” (1). Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Es por esto que esta metodología es considerada una de las más destacadas de los procesos ágiles de desarrollo de software<sup>2</sup>, y es la recomendada cuando se estima que el proyecto está sujeto a posibles cambios durante la fase de desarrollo.

Para ser más específicos, usando XP se consiguen productos usables más rápido debido a que el proceso de integración es continuo. Esto permite que el esfuerzo para

---

<sup>1</sup> Ingeniería de Software es aquella que a través de técnicas ordenadas...

<sup>2</sup> Procesos ágiles de desarrollo de software son aquellos que a través de un marco de trabajo...

consolidar cada uno de los módulos y fases del proyecto sea casi nulo y que los requerimientos del usuario sean atendidos con mayor exactitud.

## **1.1.2 Valores y Principios del Extreme Programming**

El extreme programming se enfoca en mejorar un proyecto en 4 formas esenciales: comunicación, simplicidad, retroalimentación y coraje, las cuales se establecen como valores.

### ***1.1.2.1 Comunicación***

La comunicación prevalece durante todo el proceso del extreme programming, usando modelos simples, documentos comunes y principalmente comunicación verbal frecuente. De tal manera que tanto los desarrolladores como los usuarios tengan una visión compartida del sistema que se está realizando. Esto permite que los cambios necesarios sean ágiles, que la colaboración entre los actores mejore y que se tenga una retroalimentación constante en cada fase del proyecto.

### ***1.1.2.2 Simplicidad***

“XP fomenta el iniciar con la solución más simple y desarrollar a partir de esa, versiones mejores” (2). En otras palabras, siempre iniciar desde el nivel más general, enfocándose en las necesidades y requerimientos básicos que se deben solventar, para luego encargarse de niveles más detallados. Es por eso que plantea que el riesgo de codificar sin considerar cambios futuros compensa el trabajo de invertir en posibles futuros requerimientos que podrían cambiar antes de ser relevantes.

### ***1.1.2.3 Retroalimentación***

La retroalimentación se da en varios niveles. Primero internamente, al ejecutar periódicamente pruebas unitarias e integrales. De esta manera los programadores se darán cuenta de los cambios y errores que se han dado durante el desarrollo de la aplicación.

“También existe retroalimentación cuando se realizan pruebas con el cliente. Gracias a que con esta metodología se realizan presentaciones del programa al cliente periódicamente, éste puede constatar por sí mismo como se está desarrollando el sistema y si va acorde a sus expectativas. Y de igual manera, el equipo desarrollo puede planificar si el cliente solicita algún cambio o un nuevo requerimiento” (2)

#### ***1.1.2.4 Coraje***

El último valor y quizás el más importante, se aplica en varios aspectos. Es fundamental seguir cierta disciplina y no desviarse de lo que exige el XP, empezando por planificar y codificar para las necesidades actuales y no para las futuras. También se debe ser estricto y seguro al rato de tomar decisiones como descartar cierto código que simplemente dejó de servir o no afectar otros procesos al realizar un cambio para uno en específico. Y lo más importante, siempre ser persistente al rato de buscar la solución de un problema.

#### **1.1.3 Actividades del Extreme Programming (2)**

Al rato de implementar un sistema usando XP es necesario entender y conocer las cuatro actividades que se deben realizar, que son la base de esta metodología. Estas actividades se las realiza en cada una de las iteraciones y sirven para llegar al objetivo principal que es optimizar tiempo, recursos y mejorar la calidad del producto final.

##### ***1.1.3.1 Codificar***

XP argumenta que lo más importante para el desarrollo del producto de software es codificar, actividad que definen en un concepto más amplio que la definición tradicional. Codificar en XP es hacer diagramas que generarán código, scripts para una aplicación web o código para una aplicación que necesita compilarse. Cuando se presentan varias alternativas a un problema, deben codificarse todas para elegir la mejor con pruebas automatizadas. Comunicar soluciones codificándolas es más claro y conciso ya que no está sujeto a interpretaciones.

### ***1.1.3.2 Probar***

No se puede asegurar que algo funcione a menos que se pruebe, las pruebas unitarias automatizadas aseguran que el desarrollador elimine esta incertidumbre si todas las pruebas son satisfactorias. Las pruebas de aceptación aseguran que el cliente tiene el producto que necesita de acuerdo a los requerimientos planteados. Y puede constatar en cada una de las presentaciones planificadas.

### ***1.1.3.3 Escuchar***

Los programadores no necesariamente conocen el negocio del cual el sistema es una solución, la funcionalidad es definida por los expertos del negocio, y para entender como debe ser planteada y construida la solución, los programadores deben escuchar a los expertos del negocio. Es necesario hacer esto cada vez que se tiene una reunión con el cliente.

### ***1.1.3.4 Diseñar***

En la práctica, restringirse solo a codificar, probar y escuchar no basta para crear una buena solución. Un buen diseño ayuda a evitar un sistema con muchas dependencias, complejo y difícil de mantener, facilitando de esta forma los cambios incrementales. XP fomenta a cambiar fácilmente el diseño conforme se vayan presentando los cambios en los requerimientos.

## **1.1.4 Ciclo de Vida del Extreme Programming**

El ciclo de vida de un proyecto XP incluye, entender lo que el cliente necesita, estimar el esfuerzo, crear la solución y entregar el producto final al cliente. Sin embargo, XP propone un ciclo de vida dinámico, donde se admite expresamente que, en muchos casos, los clientes no son capaces de especificar sus requerimientos al comienzo de un proyecto.

Por ésto, se trata de realizar ciclos de desarrollo cortos con módulos funcionales al finalizar cada ciclo. En cada iteración se realiza un ciclo completo de análisis, diseño, desarrollo y pruebas, pero utilizando un conjunto de reglas y prácticas que caracterizan

a XP, y lo diferencian del resto de modelos. Típicamente un proyecto con XP lleva 10 a 15 ciclos o iteraciones y si bien el ciclo de vida de un proyecto XP es muy dinámico, se puede separar en fases.

#### ***1.1.4.1 Fase de Exploración***

Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita y sus requerimientos. Los programadores estiman los tiempos de desarrollo en base a esta información. “Debe quedar claro que las estimaciones realizadas en esta fase son primarias, ya que estarán basadas en datos de muy alto nivel, y podrían variar cuando se analicen más en detalle en cada iteración. Esta fase dura típicamente un par de semanas, y el resultado es una visión general del sistema, y un plazo total estimado” (2).

#### ***1.1.4.2 Fase de Planificación***

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas<sup>3</sup> en donde se tiene un esquema y una guía para poder empezar el proyecto.

#### ***1.1.4.3 Fase Iteraciones***

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa los requerimientos del usuario asignadas a cada iteración. Como los requerimientos en un inicio no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo siendo ésta, una característica importante del XP. Las iteraciones son también utilizadas para medir el

---

<sup>3</sup> El plan de entrega es un cronograma definido en donde están especificadas...

progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.

#### ***1.1.4.4 Fase de Puesta en Producción***

“Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta que no se tenga la funcionalidad completa. En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste.” (2)

## **1.2 Patrón Modelo Vista Controlador**

“Modelo Vista Controlador (MVC) es un patrón de Arquitectura de Software<sup>4</sup> que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista.” (1) Sus grandes ventajas radican en que facilita la funcionalidad, mantenimiento y escalabilidad del sistema, de forma simple y sencilla, a la vez que permite no mezclar lenguajes de programación en el mismo código.

### **1.2.1 Modelo**

El modelo principalmente representa la lógica del negocio y la funcionalidad del sistema. Es el encargado de tener acceso de forma directa a la capa de almacenamiento de datos. Esta puede ser una base de datos, o cualquier otro tipo de almacenamiento de datos.

### **1.2.2 Vista**

La vista es la encargada de interactuar con el usuario y presentar los datos del modelo de forma gráfica. Pueden existir varias vistas bajo un mismo modelo, permitiendo que

---

<sup>4</sup> La Arquitectura de Software es la forma en la que se organizan los componentes de un sistema...

se puedan crear nuevas vistas sin tener que crear un nuevo modelo. Además es la encargada de instanciar al controlador una vez que el usuario realiza una acción.

### **1.2.3 Controlador**

El controlador es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario. Es el encargado de controlar los eventos y acciones realizadas por el usuario e interactuar con la lógica del negocio (el modelo).

## **1.3 Aplicaciones Web**

“Son aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. Es decir, es una aplicación de software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. Las aplicaciones web son populares debido a la facilidad para actualizar y mantenerlas sin tener que distribuir e instalar software a miles de usuarios. Algunos ejemplos son los webmails, weblog o tiendas en línea. Una página Web puede contener elementos que permiten la comunicación activa entre el usuario y la información, accediendo a los datos de modo interactivo, como rellenar y enviar formularios, participar en juegos, etc.” (3)

Existen algunos componentes que integran toda la arquitectura de una aplicación web bien definida y compleja, y que permiten que pueda funcionar correctamente. Primero se necesita un navegador con el que vamos a llamar a la aplicación y en donde se va a desplegar la misma. El navegador interpreta órdenes recibidas en formato HTML y las convierte en páginas que son el resultado de dicha orden. Para hacer ésto, el navegador (cliente) establece una conexión para extraer información de un servidor y la interpreta para luego desplegar dicha información. De esta manera, se tiene lenguajes de programación del lado del servidor, que son ejecutados e interpretados por el mismo servidor y que se envían al cliente en un formato comprensible para éste. Por otro lado, los lenguajes de lado del cliente son aquellos que son interpretados directamente por el navegador por lo que pueden ser incluidos en el código HTML.

Aparte del servidor y los lenguajes, una aplicación web debe tener alguna manera de almacenar información y tener acceso a ella. Para ésto, en la gran mayoría de los casos se

necesita una base de datos que cumpla esta función. Por otro lado, también es necesaria alguna manera de poder generar un ambiente gráfico que sea lo más amigable y agradable para el usuario. Las hojas de estilo son las que se encargan de aquello, permiten personalizar la parte gráfica dentro de una aplicación web. Finalmente se necesita alguna manera de administrar todos estos componentes de tal modo que sea fácil acceder a ellos y también mantenerlos bajo cierto esquema de organización. Por eso actualmente es recomendable usar un framework que nos ayude a hacer todo eso.

### **1.3.1 HTML**

HTML (*HyperText Mark-Up Language*) es lo que se conoce como lenguaje de marcado, cuya función es preparar documentos escritos aplicando etiquetas de formato. Las etiquetas indican cómo se presenta el documento y cómo se vincula a otros documentos. HTML se usa también para la lectura de documentos en Internet desde diferentes equipos gracias al protocolo HTTP, que permite a los usuarios acceder, de forma remota, a documentos almacenados en una dirección específica de la red, denominada dirección URL<sup>5</sup>.

Básicamente el lenguaje HTML es el que nos permite observar los contenidos de una página web de forma estructurada. Es una herramienta para que el navegador conectado a Internet interprete cómo visualizar el documento. Es por todo esto que es una pieza clave para poder desarrollar aplicaciones web. Y quizás ya tenga varios años de evolución pero sin duda alguna su nueva versión HTML5<sup>6</sup>, revolucionará y facilitará el desarrollo de aplicativos web.

### **1.3.2 CSS**

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su

---

<sup>5</sup> El URL (Uniform Resource Locator - Localizador Uniforme de Recursos)...

<sup>6</sup> HTML5 es la nueva versión de HTML que viene con una serie de mejoras...

presentación y es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

### **1.3.3 Navegadores Web**

“Un navegador o navegador web, es una aplicación que opera a través de Internet, interpretando la información de archivos y sitios web.” (1) Estos pueden encontrarse alojados en un servidor dentro de la Web o en un servidor local. El navegador interpreta el código, HTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos, es decir un servidor web. Tales documentos, poseen hipervínculos que enlazan una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen. El seguimiento de enlaces de una página a otra, ubicada en cualquier computadora conectada a la Internet, se llama navegación, de donde se origina el nombre navegador.

Los navegadores web son una herramienta esencial al rato de construir una aplicación web, ya que nos sirve para interpretar el código y mostrarlo visualmente al usuario. Al momento existe un gran número de navegadores web, pero los más usados son: Internet Explorer de Microsoft, Firefox de Mozilla, Chrome de Google, Safari de Apple y Opera de Opera Software. Existe un gran debate por cuál de estos es el mejor como herramienta para desarrollar aplicaciones y páginas web.

Internet Explorer es quizás el más popular a nivel mundial, pero debido a su bajo rendimiento y a que las versiones anteriores no cumplen con la mayoría de estándares, diseñar y desarrollar aplicaciones que funcionen correctamente en este navegador es bastante complicado. Google Chrome por otro lado, va ganando cada vez más popularidad ya que es muy amigable y de excelente rendimiento. Y no solo eso, cada vez posee más herramientas auxiliares para desarrolladores. Pero sin duda alguna, el favorito por los desarrolladores es Firefox. Su alta velocidad y rendimiento y sobre todo por poseer una inmensa cantidad de herramientas auxiliares que facilitan el desarrollo web, hacen de Firefox el navegador más recomendable para los programadores.

### ***1.3.3.1 Herramientas de Firefox***

La ventaja principal de Firefox respecto al resto de navegadores, es su gran cantidad de herramientas para facilitar el desarrollo. Estas herramientas son conocidas como complementos y se las puede instalar dependiendo la funcionalidad que se necesite. Es increíble la cantidad de tiempo que se puede ahorrar al tenerlas incluidas dentro del mismo navegador, ya que no es necesario abrir otras aplicaciones y estar intercambiando entre dichas aplicaciones y el navegador. Además optimiza recursos de la máquina y permite interactuar directamente con las páginas web.

La más popular con casi 3 millones de usuarios y una de las más útiles de todas es Firebug. Este complemento despliega una barra completa de herramientas para el desarrollo que permite editar, monitorear y probar el código de cualquier sitio web. Es muy útil ya que ayuda al rato de diseñar los estilos, a identificar y corregir errores rápidamente, a manipular el código y los elementos HTML, entre otras ventajas. Además Firebug posee una consola para Javascript que permite monitorear paso a paso lo que va sucediendo mientras se está dentro de un sitio web.

Además de este complemento, existen otros bastante útiles como Colorzilla, que te permite capturar y manipular los colores de las páginas web. También está MeasureIt, con el que se puede medir los componentes a través de una regla virtual graduada en píxeles. Una muy provechosa es Html Validator, valida y muestra los errores HTML

de una página web a través de iconos. Para probar la velocidad y performance existe YSlow que te permite identificar las razones por las cuales alguna página web en particular no tiene un rendimiento óptimo. En definitiva, para un desarrollador que esté familiarizado con la programación para sitios o aplicaciones web, encontrará estas herramientas complementarias muy útiles.

## **1.4 Herramientas**

### **1.4.1 PHP**

PHP es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor web. El significado de sus siglas es un acrónimo recursivo que significa PHP HyperText Preprocessor. Entre sus principales características cabe destacar su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos.

“Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante dominio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones. “ (1)

### **1.4.2 Javascript**

Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los

navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Javascript se puede crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso con que cuenta este lenguaje es el propio navegador.

Javascript es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez y ligereza. Incluso las personas que no tienen una experiencia previa en la programación podrían aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica. Existen dos acciones principales que se pueden realizar con Javascript. Por un lado los efectos especiales sobre páginas web. Esto es, crear contenidos dinámicos y elementos de la página que tengan movimiento o que cambien de color o cualquier otro dinamismo. Por el otro, Javascript permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se puede crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

#### ***1.4.2.1 jQuery***

jQuery es una librería de Javascript que facilita la manera de interactuar dentro de una aplicación Web. Simplifica la manipulación de HTML, eventos generados, el desarrollo de animaciones y agrega interacción con la técnica AJAX<sup>7</sup>. Su lema es “Escribir menos y hacer más”, esto quiere decir que está enfocada en simplificar lo más posible el código, de tal manera que con pocas líneas se consigan grandes resultados.

Las ventajas de JQuery radican justamente en la simplicidad de su código y su flexibilidad al rato de manipular los objetos de un ambiente web. Es un producto estable, serio y muy bien documentado por lo que no es para nada complicado aprenderlo y mantenerse actualizado respecto a las nuevas versiones. Otra de las ventajas es la inmensa cantidad de plugins y componentes generados, lo cual permite hallar fácilmente una solución ya creada. Todo esto hace que hoy por hoy, JQuery sea la librería más popular y recomendable entre los desarrolladores de aplicaciones web.

---

<sup>7</sup> AJAX (Asynchronous Javascript And Xml), es una técnica de desarrollo web...

### 1.4.3 MySQL

MySQL es un sistema de bases de datos relacional<sup>8</sup> con más de seis millones de instalaciones. Una de las razones es que generalmente está ligado a las aplicaciones hechas en PHP, y en general es muy recomendado su uso para aplicaciones WEB ya que no tienen una alta concurrencia de modificación. A pesar de su popularidad, este sistema todavía no tiene algunas de las características que poseen otros sistemas comerciales. Sin embargo, estas diferencias son cada vez más pequeñas por lo que cada vez existen más usuarios que la prefieren.

“MySQL es software de fuente abierta<sup>9</sup>. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones. Esto permite a los usuarios tener la libertad de usar este sistema de bases de datos para adaptarlo a sus necesidades de manera gratuita sin ningún inconveniente.” (4)

MySQL tiene una gran cantidad de ventajas y si la aplicación que se desea implementar se acopla a estas características, obviamente lo aconsejable sería escogerla por sobre otras opciones. Primero como ya se lo mencionó antes, es software libre lo que permite usarla y distribuir la aplicación gratuitamente. Segundo, tiene un excelente rendimiento y velocidad, lo que le hace uno de los gestores con mayor rendimiento. Además posee un bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema. Y si fuera poco, tiene poca probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está. En resumen, su conectividad, velocidad, fácil instalación, soporte en casi todos los sistemas operativos y seguridad, hacen de MySQL una de las mejores opciones para usar al momento de construir aplicaciones.

---

<sup>8</sup> Una base de datos relacional es un conjunto ordenado de datos...

<sup>9</sup> El Software de fuente abierta o Software Libre es un tipo particular de software...

#### **1.4.4 Symfony**

Symfony es un framework<sup>10</sup> PHP que facilita el desarrollo de las aplicaciones web. Symfony se encarga de todos los aspectos comunes de las aplicaciones web, dejando que el programador se dedique a aportar valor desarrollando las características únicas de cada proyecto. Symfony aumenta exponencialmente la productividad y ayuda a mejorar la calidad de las aplicaciones web aplicando todas las buenas prácticas y patrones de diseño que se han definido para la web. Symfony es además el framework más documentado del mundo, ya que cuenta con miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales.

Symfony posee algunas características que lo convierten en una de las mejores opciones si se desea desarrollar un sistema usando PHP. Primero, es fácil de instalar en sistemas Windows, Linux y Mac, y funciona con todas las bases de datos comunes. Es enfocado a aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada organización. Además es software libre, por lo que su utilización no tiene costo alguno. Y como si fuera poco, es muy flexible y extensible, ya que posee un completo mecanismo de plugins.

En cuanto a su funcionalidad, Symfony posee una manera ágil y sencilla de generar y manipular componentes que se adaptan fácilmente al modelo inicial de una aplicación, a través de una poderosa consola y Doctrine. Toda esta arquitectura está basada en el patrón MVC por lo que es fácil interactuar con los elementos generados. Y todos sus archivos de configuración están generados en código YAML<sup>11</sup> por lo que es muy fácil cambiar esta configuración para que se adapte con la lógica del negocio.<sup>12</sup>

##### **1.4.4.1 Doctrine**

Doctrine es un ORM (Object Relational Mapper)<sup>13</sup> realizado en PHP, que permite abstraer la lógica del modelo de una base de datos en objetos codificados en PHP. Una característica de Doctrine es el bajo nivel de configuración que necesita para empezar

---

<sup>10</sup> Un framework es una estructura de soporte definida en la cual otro proyecto de software...

<sup>11</sup> YAML es un formato de serialización de datos legible por humanos inspirado en lenguajes...

<sup>12</sup> Más información visitar <http://www.symfony.es/>

<sup>13</sup> Un ORM (Object Relation Mapper) es una técnica de programación...

un proyecto. Doctrine puede generar clases a partir de una base de datos existente y después el programador puede especificar relaciones y añadir funcionalidad extra a las clases autogeneradas. No es necesario generar o mantener complejos esquemas XML de base de datos como en otros frameworks.

Una característica importante de Doctrine es la posibilidad de escribir consultas de base de datos utilizando un dialecto de SQL denominado DQL (Doctrine Query Language). A través de DQL es posible realizar consultas a la base de datos usando objetos directamente, tanto al rato de realizar la consulta como al rato de recibir la respuesta. Además Doctrine tiene soporte para datos jerárquicos y herencia por lo que es posible recibir las respuestas de la base de datos incluso con objetos hijos en caso de haberlos.

#### **1.4.5 Apache**

Apache es un servidor del protocolo http<sup>14</sup>, comúnmente llamado servidor web pues es la mayor utilidad para dicho protocolo. Un servidor como Apache puede alojar varios sitios, y pueden coexistir varios servidores Apache en un sólo equipo. Básicamente Apache lee un directorio con todo el contenido posible a enviar y los navegadores piden las páginas o recursos para luego, mostrarlos en pantalla. Es el funcionamiento más básico de un servidor, sin embargo, los servidores actuales realizan muchas tareas complejas. Un ejemplo sería modificar el recurso para personalizarlo y luego enviarlo. O ejecutar un programa y que la salida de este programa devuelva el recurso a enviar. A estos programas se llaman scripts y se tienden a escribir en lenguajes que fueron creados para ese propósito, como lo es PHP.

#### **1.4.6 Netbeans PHP**

Netbeans es un IDE<sup>15</sup> libre fundado por Sun Microsystems, hecho principalmente para el lenguaje de programación JAVA. Debido a su excelente rendimiento y simpleza, posteriormente se desarrolló versiones para PHP y Python. NetBeans provee una estructura para los proyectos que se puede crear junto a este IDE, propone un esqueleto

---

<sup>14</sup> Http significa “Protocolo de Transferencia de Hiper Texto”...

<sup>15</sup> Un entorno de desarrollo integrado, llamado también IDE, es un programa informático...

para organizar nuestro código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS. Además NetBeans posee un sistema para examinar todo los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación.

Otra de las ventajas es que permite la integración con Symfony y Zend Framework, por lo que si se va a utilizar cualquiera de estos 2 frameworks, Netbeans va a facilitar mucho el desarrollo. Además tiene un robusto editor de texto que facilita mucho la codificación. En cuanto a las pruebas unitarias<sup>16</sup>, es posible utilizar PHPUnit<sup>17</sup> para diferentes funciones, luego realizar la comprobación y finalmente ver todos los resultados. NetBeans también posee una integración completa en términos de administración básica y avanzada de MySQL, y todo desde el mismo entorno. Y para las versiones, una integración con sistemas de control de versiones como SVN, CVS, Mercurial y Git. Lo que permite desde el mismo editor manipular libre y directamente las versiones.

#### **1.4.7 PowerDesigner**

PowerDesigner es una herramienta desarrollada por Sybase para modelamiento de datos y procesos de negocios. En ella es posible construir los modelos UML<sup>18</sup> que se realizan en la fase de diseño de una aplicación, así como el diseño de la base de datos. PowerDesigner brinda un enfoque basado en modelos, el cual permite alinear al negocio con la tecnología de información, facilitando la implementación de arquitecturas efectivas de información empresarial. Brinda potentes técnicas de análisis, diseño y gestión para el desarrollo de una aplicación.

PowerDesigner combina varias técnicas estándar de modelamiento con herramientas líder de desarrollo, como .NET, Sybase WorkSpace, Sybase PowerBuilder, Java y Eclipse. Además trabaja con más de 60 bases de datos relacionales lo cual la convierte en una herramienta sumamente robusta. Otra de las ventajas de PowerDesigner es su ambiente gráfico, es muy fácil manipular y modificar los modelos, siendo esta

---

<sup>16</sup> En programación, una prueba unitaria es una forma de probar el correcto funcionamiento...

<sup>17</sup> PHPUnit es un entorno para realizar pruebas unitarias en el lenguaje de programación PHP.

<sup>18</sup> Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language)...

herramienta muy amigable para el usuario. Además reduce significativamente el costo y tiempo al implementar cualquier cambio a través de una vista exacta, bi-direccional y multi-modelo para análisis de impacto que integra todos los modelos de requerimientos, análisis, base de datos y aplicación.

#### **1.4.8 phpMyAdmin**

phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 62 idiomas.

Una de sus ventajas es que es software libre, por lo que su utilización no tiene ningún costo. Además es sumamente liviana y fácil de instalar, lo que la hace muy popular dentro de los desarrolladores. Y ya que se la usa dentro de un navegador, facilita mucho al rato de querer hacer consultas rápidas a la base de datos. También permite obtener fácilmente respaldo de la base de datos, así como crear y cargar scripts de datos.

## **2. DISEÑO**

En este capítulo se hará una descripción detallada del problema y de la aplicación. Se presentará un listado de cada uno de los requerimientos y sus características. También la definición de los procesos que se ven involucrados en el aplicativo, con un gráfico descriptivo de cada uno de ellos. Finalmente se mostrará los casos de uso existentes, tanto a nivel general como a detalle.

### **2.1 Descripción de la aplicación**

El ingeniero químico Carlos Argudo empezó hace ya algunos años, un negocio como intermediario en la importación de productos químicos tanto a nivel local como internacional. La funcionalidad principal de la empresa es facilitar la negociación, distribución y búsqueda de productos entre proveedores y clientes. Al momento estos procesos se los realiza con herramientas como Excel y Word para almacenar y gestionar la información.

Debido a que el negocio ha ido creciendo, lo que ha llevado a un aumento considerable, tanto de clientes como proveedores, se ha visto la necesidad de contar con un sistema para facilitar y agilizar los procesos que realiza la empresa. Esto permitiría llevar un mejor control de las gestiones y negociaciones que se realizan entre clientes y proveedores, ahorrando tiempo y evitando posibles errores.

El negocio funciona de la siguiente manera. Primero un cliente solicita cierto producto al intermediario y el intermediario busca el mejor proveedor para el producto solicitado. Para hacerlo, realiza un análisis de disponibilidad, calidad y precio, para posteriormente enviar un pedido de cotización. El proveedor entonces envía una oferta al intermediario y éste se la envía al cliente. El cliente aprueba la oferta y procede a realizar la compra, en ese momento el intermediario envía al proveedor la orden de compra y el proveedor responde enviando la factura. Una vez que el cliente realiza su pago, el proveedor paga una comisión por el proceso realizado.

Los módulos que se requieren para el funcionamiento del sistema abarcan administraciones, procesos y reportes. Administraciones de clientes, proveedores, productos, tipos de producto, alertas y usuarios. Por el lado de procesos tenemos: pedidos, compras, créditos, pagos. Y se necesita reportes de ventas por cliente, ventas por proveedor, comisiones por cliente y

comisiones por proveedor. Todos estos módulos estarán integrados para obtener los resultados deseados.

Basándose en las características del sistema y el tamaño de la empresa se ha optado por construir la aplicación usando la metodología ágil Extreme Programming y bajo un patrón de arquitectura Modelo Vista Controlador. El lenguaje en el que se construirá el programa será PHP y se lo montará en un servidor Apache.

## **2.2 Levantamiento de requerimientos**

Para el levantamiento de los requerimientos se usó la técnica de la entrevista, a través de la cual y durante 3 sesiones se logró establecer un conjunto de requerimiento sobre el sistema. Posteriormente se llevó a cabo la clasificación de las necesidades a incorporar y se eliminó algunos requerimientos innecesarios. Finalmente se presentó este conjunto de requerimientos y de manera conjunta se aprobó la implementación. A continuación el listado de los mismos:

- F1: El sistema permitirá una administración de clientes.<sup>19</sup>
- F2: El sistema permitirá una administración de proveedores.
- F3: El sistema permitirá una administración de tipos de productos.
- F4: El sistema permitirá una administración de productos.
- F5: El sistema permitirá una administración de alertas.
- F6: El sistema permitirá una administración de usuarios.
- F7: El sistema contará con una autenticación de usuario para poder ingresar a la aplicación.
- F8: El sistema permitirá registrar pedidos por parte de los clientes.
- F9: El sistema permitirá registrar compras una vez que los pedidos sean aprobados.
- F10: El sistema permitirá asignar créditos a los clientes (financiamiento).
- F11: El sistema permitirá registrar pagos.
- F12: El sistema presentará alertas de pedidos y ordenes de compra.

---

<sup>19</sup> Entiéndase por administración las operaciones de insertar, consultar y editar sobre cada una de las entidades. Se llevó a cabo la clasificación de las necesidades a incorporar.

- F13: El sistema permitirá generar reporte de ventas.
- F14: El sistema permitirá generar reporte de comisiones.

## **2.3 Diagrama de Procesos**

Se ha identificado los siguientes procesos.

### **2.3.1 Pedidos**

Para los pedidos, primero el cliente realiza un pedido de cotización de ciertos productos que desea comprar. El intermediario analiza usando una lista de inventario, si dichos productos están disponibles en alguno de sus proveedores. En el caso de que no, se envía una respuesta al cliente vía correo electrónico indicando que no podrá realizar la compra. Caso contrario se envía un documento al proveedor con un pedido de cotización. Si el proveedor tiene disponibles los productos requeridos entonces envía una oferta al intermediario y este a su vez la envía al cliente. Una vez analizada la oferta, el cliente está en la potestad de aceptar o no dicha oferta. Si la acepta, se genera una orden de compra y se procede a crear un contrato, caso contrario se procede a anular el pedido.

#### **2.3.1.1 Entradas**

- Pedido de cotización por parte del cliente.

#### **2.3.1.2 Actividades**

- Registrar pedido de cotización del cliente.
- Seleccionar el mejor proveedor para los productos requeridos.
- Enviar el pedido al proveedor.
- Registrar la oferta o respuesta del proveedor.
- Enviar la oferta o respuesta del proveedor al cliente.
- Registrar la respuesta del cliente una vez recibida la oferta del proveedor.

#### **2.3.1.3 Salidas**

- Orden de compra

- Anulación de pedido

### 2.3.1.4 Gráfico

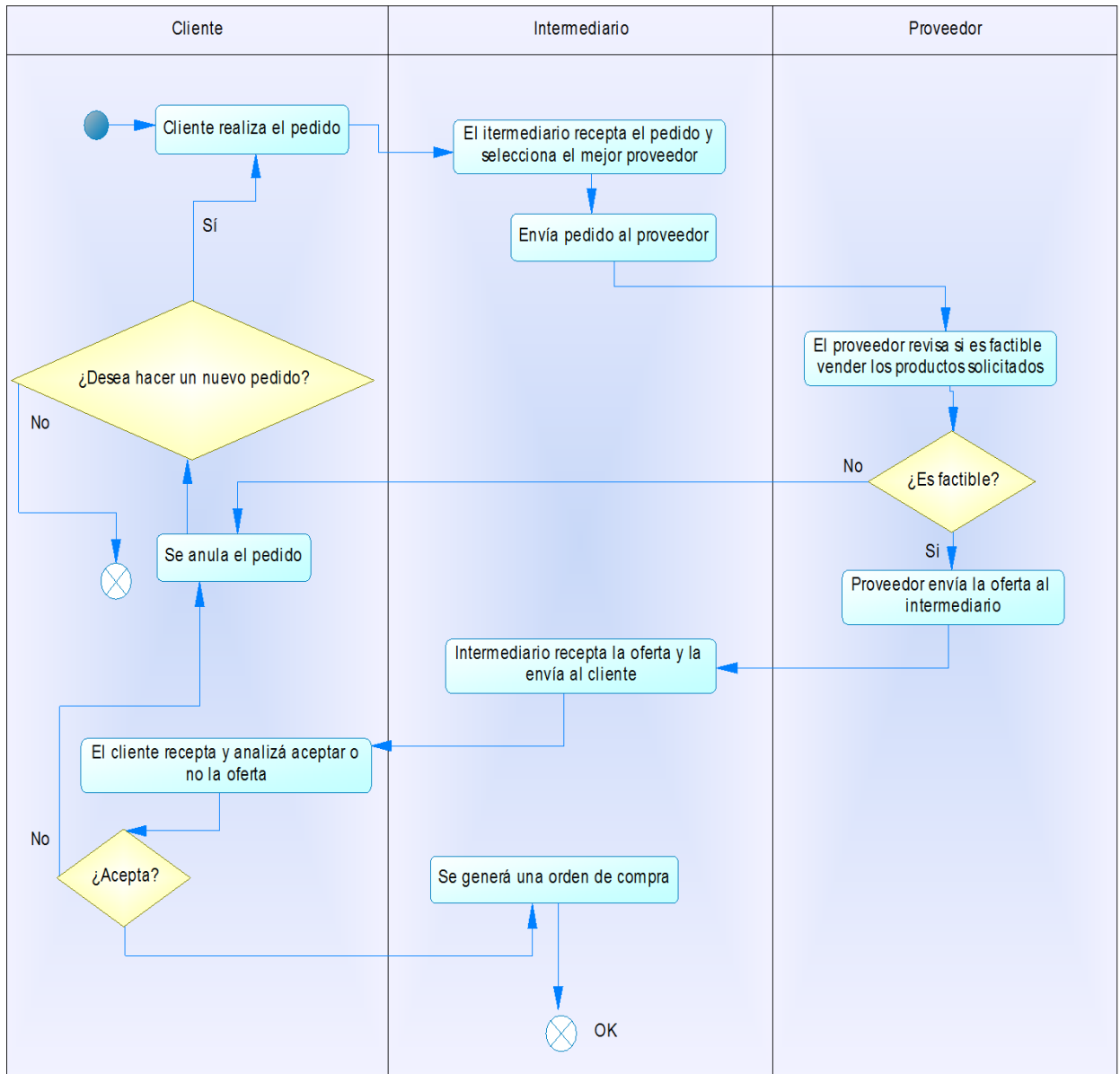


Ilustración 1 Diagrama de Procesos de Pedidos (5)

### 2.3.2 Compras

Una vez que se genera una orden de compra por parte del cliente, el intermediario procede a enviársela al proveedor. El proveedor entonces envía los datos de la factura y se registra la compra.

#### 2.3.2.1 Entradas

- Orden de compra por parte del cliente.

#### 2.3.2.2 Actividades

- Registrar la orden de compra.
- Enviar la orden al proveedor.
- Registrar los datos de la factura.
- Registrar la fecha de pago de la comisión.

#### 2.3.2.3 Salidas

- Compra efectiva.

#### 2.3.2.4 Gráfico

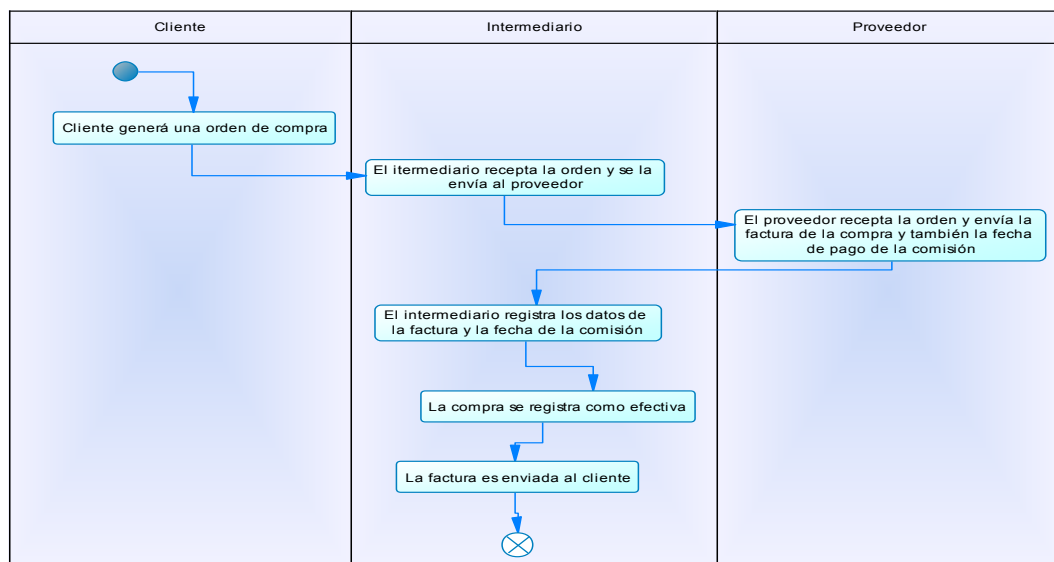


Ilustración 2 Diagrama de Procesos de Ordenes de Compra (5)

### 2.3.3 Créditos

El cliente puede solicitar créditos a los proveedores para pagar las compras realizadas. El proveedor analiza si le da o no el crédito al cliente y el valor del mismo. Si se le otorga, entonces el crédito queda registrado.

#### 2.3.3.1 Entradas

- Solicitud de crédito por parte del cliente.

#### 2.3.3.2 Actividades

- Enviar la solicitud de crédito al proveedor.
- Si la acepta, registrar el crédito y el valor del mismo.

#### 2.3.3.3 Salidas

- Crédito.

#### 2.3.3.4 Gráfico

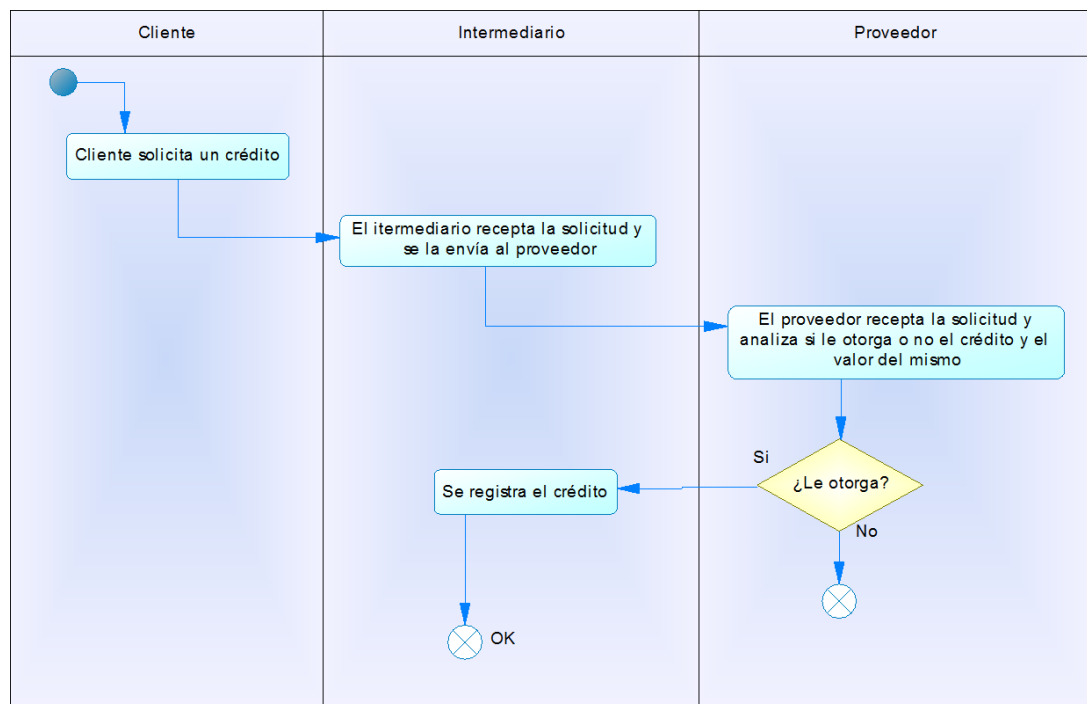


Ilustración 3 Diagrama de Procesos de Créditos (5)

### 2.3.4 Pagos

Una vez que el cliente realiza una compra, se definen los plazos para los respectivos pagos que cubran el costo de los productos adquiridos. Una vez que el cliente ejecuta el pago, éste queda registrado en el sistema y se genera un comprobante de pago.

#### 2.3.4.1 Entradas

- Compra

#### 2.3.4.2 Actividades

- Registrar el pago realizado por el cliente.
- Generar comprobante de pago.
- En caso de que la cuota esté pagada en su totalidad, registrar la compra pagada.

#### 2.3.4.3 Salidas

- Comprobante de pago

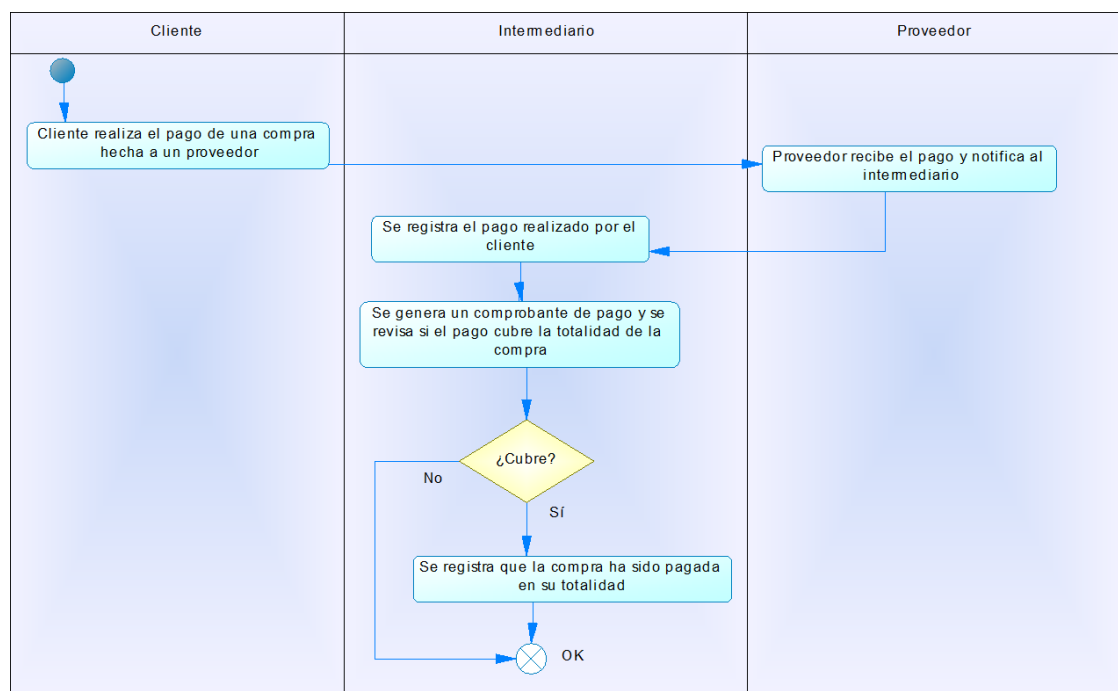
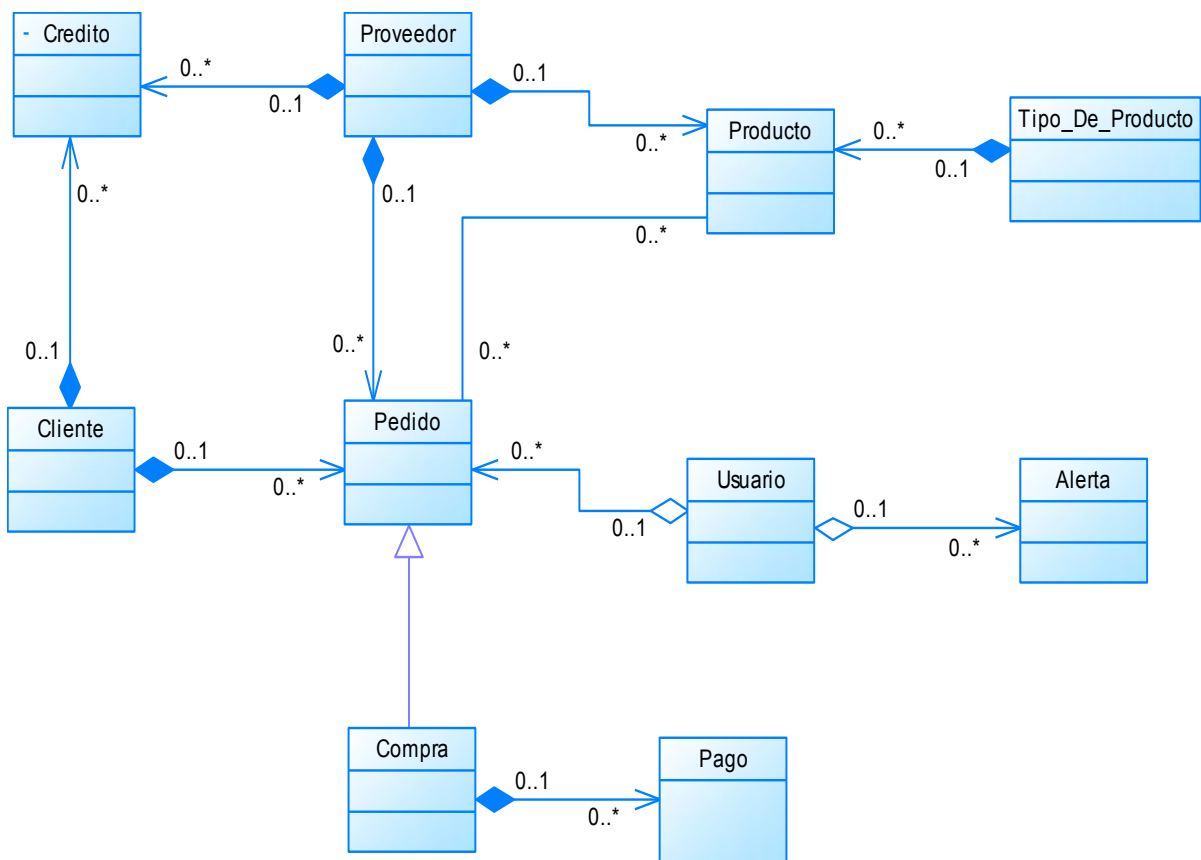


Ilustración 4 Diagrama de Procesos de Pagos

## 2.4 Diagrama de Clases

Una vez realizado el análisis y levantamiento de requerimientos, así como el levantamiento de procesos, se ha podido establecer el siguiente diagrama general de clases que lo componen ciertas entidades tal y como se muestra en el siguiente diagrama.



## 2.5 Casos de uso

### 2.5.1 Diagrama de casos de uso a nivel general

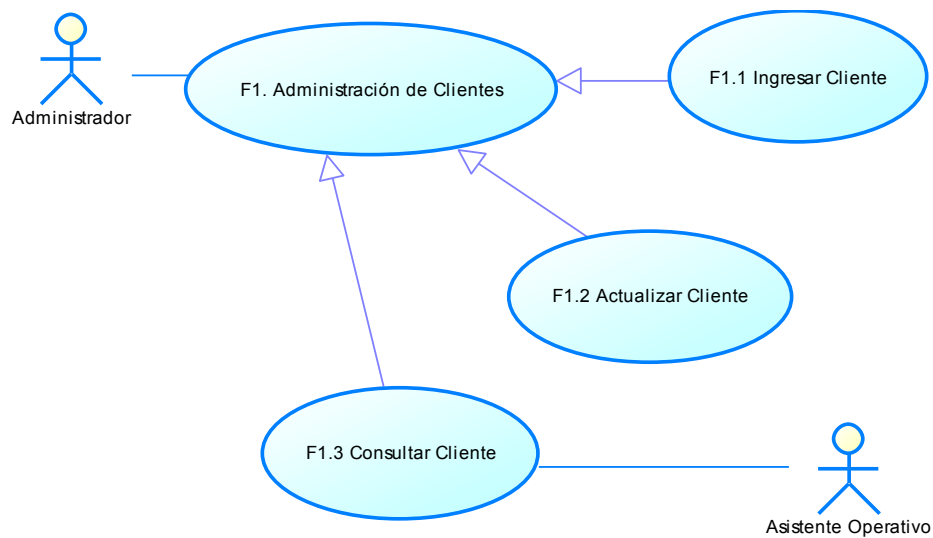
A continuación se indica a nivel general los casos de uso del presente trabajo.



## 2.5.2 Diagramas de casos de uso a detalle

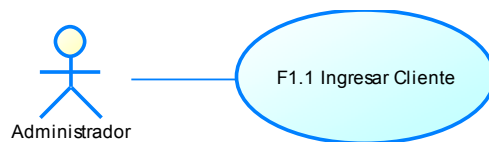
A continuación se presenta cada uno de los casos de uso a detalle. Debido a que la lógica entre las administraciones y procesos no varía mucho, se mostrarán solo los casos de uso más representativos. Los restantes pueden ser consultados en los anexos.

### *F1. Administración de clientes*



### **DETALLE:**

#### ***F1.1 INGRESAR CLIENTE***



#### ***Actores***

Administrador

#### ***Descripción***

A través de este caso de uso el administrador ingresa un nuevo cliente al sistema.

### *Flujo Principal*

Paso	Actor	Paso	Sistema	Excep.
1	El administrador escoge la opción de ingresar un nuevo cliente.	2	El sistema despliega la vista para ingresar un nuevo cliente.	
3	El administrador llena los datos del cliente.			
4	El administrador presiona el botón de guardar.	5	El sistema despliega un mensaje indicando que se ha guardado con éxito.	E1, E2, E3, E4

### *Excepciones*

Código	Descripción	Alternativa
E1	La identificación del cliente ingresada ya existe.	Consultar el cliente ya existente para verificar que los datos estén correctos.
E2	Quedaron campos obligatorios vacíos.	Llenar los campos restantes.
E3	Datos ingresados incorrectos.	Corregir los datos incorrectos y volver a intentar.
E4	Error en la base de datos	Volver a intentar más tarde.

### *F1.2 ACTUALIZAR CLIENTE*



### **Actores**

Administrador

### **Descripción**

A través de este caso de uso el administrador actualiza los datos de un cliente.

### **Flujo Principal**

<b>Paso</b>	<b>Actor</b>	<b>Paso</b>	<b>Sistema</b>	<b>Excep.</b>
1	El administrador selecciona la opción de administrar clientes.	2	El sistema despliega una tabla con todos los clientes y su información más importante.	E4
3	El administrador utiliza el campo para filtrar y encontrar el cliente deseado.	4	El sistema disminuye los clientes de la tabla de acuerdo a lo ingresado en el campo para filtrar.	
5	El administrador presiona la opción de editar en el cliente que desea actualizar.	6	El sistema despliega una vista para editar los datos del cliente.	
7	El administrador modifica los campos que desea actualizar.			
8	El administrador presiona el botón de guardar.	9	El sistema despliega un mensaje indicando que se ha guardado con éxito.	E1, E2, E3, E4

### Excepciones

Código	Descripción	Alternativa
E1	La identificación del cliente ingresada ya existe.	Consultar el cliente ya existente para verificar que los datos estén correctos.
E2	Quedaron campos obligatorios vacíos.	Llenar los campos restantes.
E3	Datos ingresados incorrectos.	Corregir los datos incorrectos y volver a intentar.
E4	Error en la base de datos	Volver a intentar más tarde.

### F1.3 CONSULTAR CLIENTE



### Actores

Administrador, Asistente Operativo

### Descripción

A través de este caso de uso el administrador o el asistente operativo consulta los datos de un cliente.

### Flujo Principal

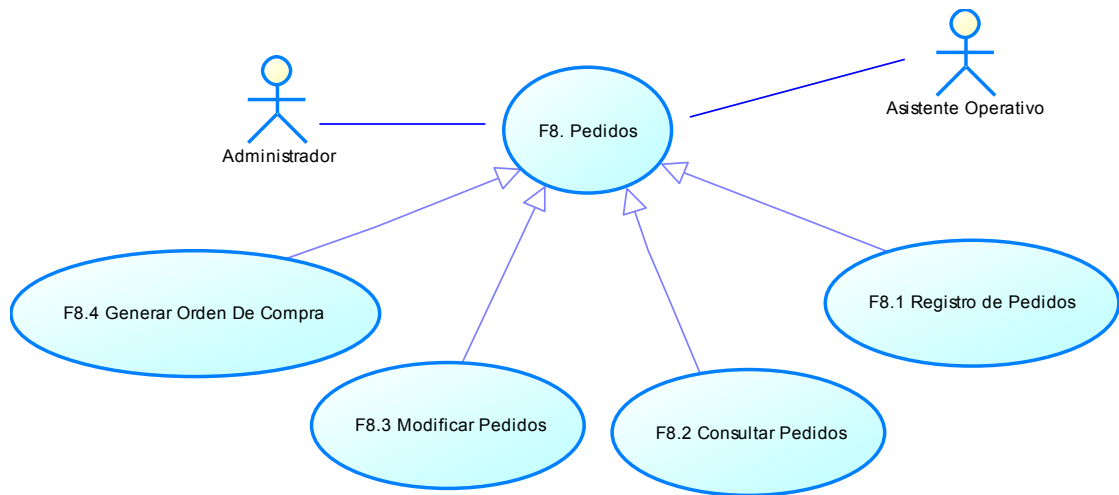
Paso	Actor	Paso	Sistema	Excep.
1	El administrador o asistente operativo selecciona la opción de	2	El sistema despliega una tabla con todos los clientes y su información	E1

	administrar clientes.		más importante.	
3	El administrador o asistente operativo utiliza el campo para filtrar y encontrar el cliente deseado.	4	El sistema disminuye los clientes de la tabla de acuerdo a lo ingresado en el campo para filtrar.	
5	El administrador presiona la opción de consultar en el cliente que desea.	6	El sistema despliega una vista con todos los datos del cliente a detalle.	

***Excepciones***

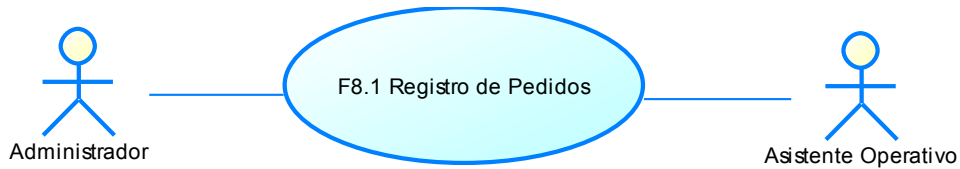
<b>Código</b>	<b>Descripción</b>	<b>Alternativa</b>
E1	Error en la base de datos	Volver a intentar más tarde.

**F8. Pedidos**



**DETALLE:**

**F8.1 REGISTRO DE PEDIDOS**



**Actores**

Administrador, Asistente Operativo

**Descripción**

A través de este caso de uso el administrador o el asistente operativo puede registrar pedidos.

**Flujo Principal**

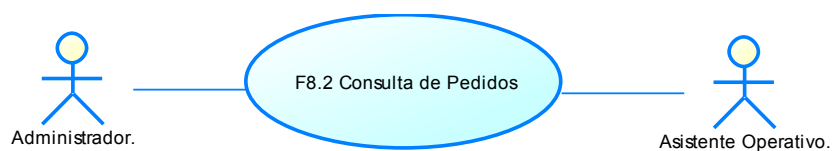
Paso	Actor	Paso	Sistema	Excep.
1	El administrador o asistente operativo	2	El sistema despliega la vista para registrar	

	selecciona la opción de registrar pedidos.		pedidos.	
3	El usuario llena los datos del pedido			
4	El usuario presiona el botón de “Registrar”.	5	El sistema despliega un mensaje indicando que el pedido se ha registrado exitosamente.	E1, E2, E3, E4

### *Excepciones*

<b>Código</b>	<b>Descripción</b>	<b>Alternativa</b>
E1	Quedaron campos obligatorios vacíos.	Llenar los campos restantes.
E2	Datos ingresados incorrectos.	Corregir los datos incorrectos y volver a intentar.
E3	Error en la base de datos	Volver a intentar más tarde.
E4	El código ingresado ya existe	Ingresar un código de pedido que no exista todavía

### **F8.2 CONSULTA DE PEDIDOS**



### **Actores**

Administrador, Asistente Operativo

### ***Descripción***

A través de este caso de uso el administrador o el asistente operativo consulta los datos de un pedido.

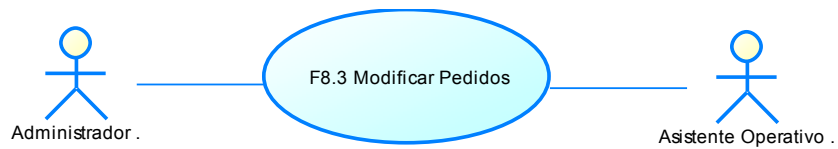
### ***Flujo Principal***

<b>Paso</b>	<b>Actor</b>	<b>Paso</b>	<b>Sistema</b>	<b>Excep.</b>
1	El administrador o asistente operativo selecciona la opción de pedidos.	2	El sistema despliega una tabla con todos los pedidos y su información más importante.	E1
3	El administrador o asistente operativo utiliza los campos para filtrar y encontrar el pedido deseado.	4	El sistema disminuye los usuarios de la tabla de acuerdo a lo ingresado en los campos para filtrar.	E2
5	El administrador presiona la opción de consultar en el pedido que desea.	6	El sistema despliega una vista con todos los datos del pedido a detalle.	

### ***Excepciones***

<b>Código</b>	<b>Descripción</b>	<b>Alternativa</b>
E1	Error en la base de datos	Volver a intentar más tarde.
E2	No ingreso correctamente el rango de fechas.	Ingresar correctamente el rango de fechas.

### **F8.3 MODIFICAR PEDIDOS**



#### **Actores**

Administrador, Asistente Operativo

#### **Descripción**

A través de este caso de uso el administrador o asistente operativo modifica los datos de un pedido.

#### **Flujo Principal**

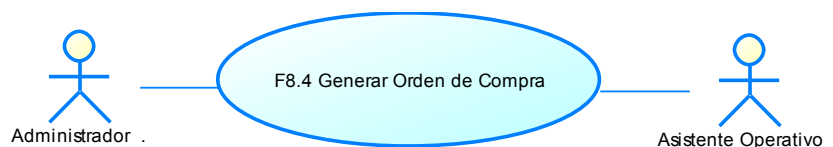
<b>Paso</b>	<b>Actor</b>	<b>Paso</b>	<b>Sistema</b>	<b>Excep.</b>
1	El administrador o asistente operativo selecciona la opción de pedidos.	2	El sistema despliega una tabla con todos los pedidos y su información más importante.	E4
3	El administrador o asistente operativo utiliza los campos para filtrar y encontrar el pedido deseado.	4	El sistema disminuye los pedidos de la tabla de acuerdo a lo ingresado en los campos para filtrar.	
5	El administrador o asistente operativo presiona la opción de editar en el pedido que desea modificar datos.	6	El sistema despliega una vista para editar los datos del pedido.	

7	El administrador o asistente operativo modifica los campos que desea actualizar.			
8	El administrador o asistente operativo presiona el botón de guardar.	9	El sistema despliega un mensaje indicando que se ha guardado con éxito.	E1, E2, E3, E4

### *Excepciones*

<b>Código</b>	<b>Descripción</b>	<b>Alternativa</b>
E1	El código del pedido ya existe.	Ingresar un código de pedido que no exista todavía.
E2	Quedaron campos obligatorios vacíos.	Llenar los campos restantes.
E3	Datos ingresados incorrectos.	Corregir los datos incorrectos y volver a intentar.
E4	Error en la base de datos	Volver a intentar más tarde.

### **F8.4 GENERAR ORDEN DE COMPRA**



### **Actores**

Administrador, Asistente Operativo

### ***Descripción***

A través de este caso de uso el administrador o asistente operativo genera una orden de compra correspondiente a un pedido.

### ***Flujo Principal***

<b>Paso</b>	<b>Actor</b>	<b>Paso</b>	<b>Sistema</b>	<b>Excep.</b>
1	El administrador o asistente operativo selecciona la opción de pedidos.	2	El sistema despliega una tabla con todos los pedidos y su información más importante.	E2
3	El administrador o asistente operativo utiliza los campos para filtrar y encontrar el pedido deseado.	4	El sistema disminuye los pedidos de la tabla de acuerdo a lo ingresado en los campos para filtrar.	
5	El administrador o asistente operativo presiona la opción de generar orden de compra en el pedido.	6	El sistema despliega una ventana para que el administrador o asistente operativo confirmen la operación.	
7	El administrador o asistente operativo presiona el botón de aceptar.	8	El sistema despliega un mensaje indicando que se generó exitosamente la orden de compra	E1, E2

### ***Excepciones***

<b>Código</b>	<b>Descripción</b>	<b>Alternativa</b>
E1	El administrador o asistente	

	operativo presiono el botón de cancelar.	
E2	Error en la base de datos	Volver a intentar más tarde.

### 3. BASE DE DATOS

En este capítulo se muestra un extracto del modelo conceptual que se utilizó para generar la base de datos del sistema, el modelo físico se lo generó a partir del conceptual, y a ambos se los puede ver en la sección de anexos. Estos modelos son necesarios para tener un esquema, o en otras palabras, un mapa de la base de datos. De esta manera es más fácil entender, ya que si es una base de datos relacional, resultaría muy difícil captar la información solo leyendo el archivo. También está incluido el diccionario de datos, que es un listado de todas las tablas y atributos con la descripción de cada uno de ellos. Sirve principalmente para que si una persona ajena a la que construyó la base de datos, quiere trabajar en ella, tenga una referencia de para que sirve cada una de las tablas y atributos respectivos.

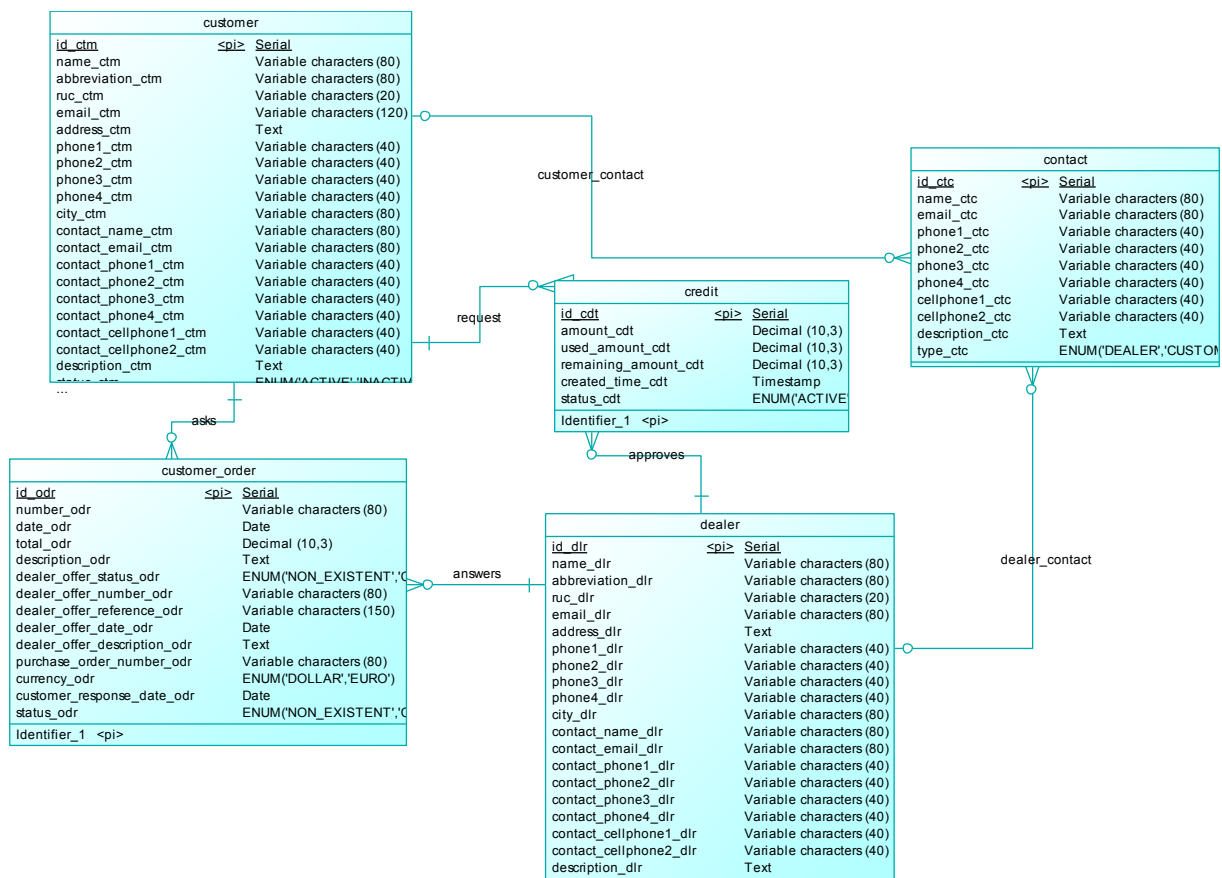


Ilustración 5 Extracto del Diagrama Conceptual de la Base de Datos (5)

En la ilustración 5, se tiene una parte de lo que es el diagrama conceptual de la base de datos. El cuál, ayudará a explicar mejor como se diseño el diagrama. Lo primero que se hizo, es seleccionar el idioma inglés para construir el modelo. Ya que es uno de los idiomas más hablados en el mundo y un requisito fundamental para nuestra carrera. La ventaja de hacer la base en inglés, es que si alguna persona que sea extranjera, desea realizar cambios en la base de datos, es más probable que este sepa inglés que español. Además gracias a esto es más factible vender la base de datos a otros países en caso de que así se requiera.

En cuanto a las relaciones utilizadas en la ilustración, se hizo un análisis basado en los requerimientos explicados en el capítulo 2 para elaborarlas. Empecemos por la tabla “contact” y las relaciones con “customer” y “dealer”. Ambas son relaciones de cero a muchos ya que tanto los clientes como proveedores pueden como no pueden tener contactos. Y los contactos pueden pertenecer a un cliente o a un proveedor pero no a ambos. En la tabla “credit” y “customer\_order”, se tiene relaciones con las tablas “customer” y “dealer”, ambas con relación de uno a muchos. Esto se debe a que no necesariamente los clientes y proveedores tienen que tener asignados créditos u órdenes de compra. Además, si se asigna un crédito u orden de compra, es obligatorio designarles un cliente y un proveedor.

file		
<u>id_file</u>	<pi>	Serial
module_file		ENUM
module_id_file		Intege
name_file		Variab
original_name_file		Variab
dir_file		Variab
Identifier_1	<pi>	

Ilustración 6 Diagrama Conceptual de la Tabla File

La tabla “file”, es la única que no tiene relación con otras tablas. Esto se debe a que su función es almacenar datos de archivos que se suben a la aplicación desde distintos módulos.

### 3.1 Diccionario de Datos

A continuación se presenta un diccionario de datos con la descripción de las tablas más representativas que componen la base de datos y sus respectivos atributos.

<b>TABLA: customer (1/2)</b>		
<b>Descripción:</b>	En esta tabla se registran los clientes y sus datos principales como razón social, ruc, correo electrónico, dirección, teléfonos, etc.	
<b>Nombre</b>	<b>Descripción</b>	<b>Tipo de Dato</b>
id_ctm	Identificador y clave primaria de la tabla.	int
id_cnt	Identificador de la tabla país, sirve para presentar el país al que pertenece el cliente.	int
name_ctm	Razón social del cliente.	varchar(80)
abbreviation_ctm	Abreviación de la razón social del cliente.	varchar(80)
ruc_ctm	Número de RUC.	varchar(20)
address_ctm	Dirección.	text
phone1_ctm	Teléfono 1.	varchar(40)
phone2_ctm	Teléfono 2.	varchar(40)
phone3_ctm	Teléfono 3.	varchar(40)
phone4_ctm	Teléfono 4.	varchar(40)
city_ctm	Ciudad	varchar(80)
contact_name_ctm	Nombre del contacto principal del cliente.	varchar(80)
contact_phone1_ctm	Teléfono 1 del contacto principal.	varchar(40)
contact_phone2_ctm	Teléfono 2 del contacto principal.	varchar(40)
contact_phone3_ctm	Teléfono 3 del contacto principal.	varchar(40)

<b>TABLA: customer (2/2)</b>		
contact_phone4_ctm	Teléfono 4 del contacto principal.	varchar(40)
contact_cellphone1_ctm	Celular 1 del contacto principal.	varchar(40)
contact_cellphone2_ctm	Celular 2 del contacto principal.	varchar(40)
description_ctm	Se registra una descripción, o cualquier nota referente al cliente.	text
status_ctm	El estado del cliente, es posible desactivar cierto cliente para que el mismo no pueda ser usado en créditos, pedidos y órdenes de compra. Debido a que éste puede ser momentáneo, es posible volverlo a activar. Los valores pueden ser: activo, inactivo.	enum('ACTIVE', 'INACTIVE')

<b>TABLA: customer_order (1/3)</b>		
<b>Descripción:</b>	En esta tabla se registra el proceso de pedido que un cliente solicita a un proveedor.	
<b>Nombre</b>	<b>Descripción</b>	<b>Tipo de Dato</b>
id_odr	Identificador y clave primaria de la tabla.	int
id_usr	Identificador de la tabla usuario, sirve para tener registrado que usuario fue el que registró el pedido.	int
id_ctm	Identificador de la tabla cliente, sirve para saber que cliente solicitó el pedido.	int
id_dlr	Identificador de la tabla cliente, sirve para saber a qué proveedor se le solicitó.	int

<b>TABLA: customer_order (2/3)</b>		
number_odr	Número de pedido.	varchar(80)
date_odr	Fecha en la que se realizó el pedido.	date
total_odr	Valor total que se registra una vez recibida la oferta	decimal(10,3)
description_odr	Se registra una descripción, o cualquier nota referente al pedido.	text
dealer_offer_status_odr	El estado del pedido, el proveedor confirma o no si puede entregar el pedido que se solicita. Los valores posibles son: no existente, confirmado, cancelado.	enum('NON_EXISTENT', 'CONFIRMED', 'CANCELED')
dealer_offer_number_odr	El número de oferta proporcionado por el proveedor	varchar(80)
dealer_offer_reference_odr	La referencia que gestionó la oferta enviada por el proveedor, puede ser el nombre de una persona o una organización.	varchar(150)
dealer_offer_date_odr	Fecha de envío de la oferta.	date
dealer_offer_description_odr	Se registra una descripción, o cualquier nota referente a la oferta.	text
purchase_order_number_odr	Número de orden de compra en caso de que la oferta haya sido aceptada.	varchar(80)
currency_odr	Moneda en la que se manejarán los valores tanto del pedido como de la orden de compra.	enum('DOLLAR', 'EURO')

<b>TABLA: customer_order (3/3)</b>		
customer_response_date_odr	Fecha de respuesta del cliente respecto a la oferta.	date
status_odr	El estado final del pedido, se ingresa al principio como no existente y luego dependiendo si el cliente acepta o no la oferta se cambia el estado a confirmada o cancelada. Los posibles valores son: no existente, confirmado, cancelado	enum('NON_EXISTENT', 'CONFIRMED', 'CANCELED')

## 4. PRUEBAS E IMPLANTACIÓN

En este capítulo se explicará detalladamente la fase de pruebas y la de implantación. En la de pruebas, se expondrá el mecanismo que se utilizó y el alcance de las mismas. Y también las pruebas como tal de los módulos más relevantes de la aplicación, el resto se las podrá consultar en los anexos. Luego, se explicará como se realizó el procedimiento de implantación, cuales fueron los pasos y cuál fue la impresión del cliente respecto al sistema.

### 4.1 Fase de Pruebas

Para el presente proyecto se ha optado por armar un conjunto de listas de verificación que validen con el usuario el funcionamiento de la aplicación. Para lo cual, se utilizará como base los casos de uso generados anteriormente. En la definición de estos, se definió un flujo principal en el caso de que no existan errores y también flujos alternos o excepciones. Se utilizarán estos flujos para generar escenarios posibles que serán utilizados como base para crear las pruebas. Adicionalmente a cada escenario se lo someterá a uno o más casos de prueba que son las condiciones y valores requeridos para la ejecución de los distintos escenarios. Es necesario indicar que solo se presentarán las pruebas de los más importantes.

Para un mejor entendimiento, se utilizará la siguiente tabla de códigos:

Código	Descripción
V	Valor válido
I	Valor inválido
Vacío	No se indica ningún valor
N/A	El valor que tenga es irrelevante
E #	Ver en las excepciones descritas en la definición de los casos de uso a detalle

### ***F1.1 Ingresar cliente***

#### ***Requerimientos:***

- Autenticación e ingreso al sistema con perfil administrador

#### ***Lista de Escenarios***

<b>ID</b>	<b>Escenario de caso de uso</b>	<b>Descripción</b>	<b>Excepción</b>
1	Ingreso correcto	El usuario ingresa los todos los datos correctamente.	
2	Error en la base de datos	Al guardar existe un error en la base de datos.	E4
3	Cliente ya existe.	El usuario ingresa un cliente ya existente.	E1
4	Campos obligatorios vacíos.	El usuario no llena campos que son obligatorios.	E2
5	E-mail inválido	El usuario ingresa un e-mail inválido.	E3

#### ***Matriz de casos de prueba***

<b>ID</b>	<b>Escenario</b>	<b>Datos de Entrada</b>	<b>Resultado Esperado</b>
1	Ingreso correcto	V -- Razón Social: ACOSA	Cliente creado. Se abre la página de lista de clientes.
2	Ingreso correcto	V -- Razón Social: UBESA V -- RUC: 20259551022	Cliente creado. Se abre la página de lista de clientes.
3	Ingreso correcto	V -- Razón Social: SUPAN S.A. V -- RUC: 1790005739001 V -- E-mail: info@supan.com	Cliente creado. Se abre la página de lista de clientes.
4	Error en la base de datos	N/A	Cliente no creado. Se abre la página de lista de clientes y se presenta un mensaje de error.
5	Cliente ya existe.	V -- Razón Social: ACOSA	Cliente no creado. Se despliega un mensaje indicando que ese

			cliente ya existe.
6	Cliente ya existe.	V -- RUC: 1790005739001	Cliente no creado. Se despliega un mensaje indicando que ya existe un cliente con ese RUC.
7	Campos obligatorios vacíos.	V -- Razón Social: Vacío	Cliente no creado. Se despliega un mensaje indicando que el campo es obligatorio.
8	E-mail inválido	I -- E-mail: info.com	Cliente no creado. Se despliega un mensaje indicando que el e-mail ingresado no es válido.

### ***F1.2 Actualizar cliente***

#### ***Requerimientos:***

- Autenticación e ingreso al sistema con perfil administrador

#### ***Lista de Escenarios***

<b>ID</b>	<b>Escenario de caso de uso</b>	<b>Descripción</b>	<b>Excepción</b>
1	Actualización correcta	El usuario ingresa los todos los datos correctamente.	
2	Error en la base de datos	Al guardar existe un error en la base de datos.	E4
3	Identificación del cliente ya existe.	El usuario ingresa la identificación de un cliente ya existente.	E1
4	Campos obligatorios vacíos.	El usuario no llena campos que son obligatorios.	E2
5	E-mail inválido	El usuario ingresa un e-mail inválido.	E3

#### ***Matriz de casos de prueba***

<b>ID</b>	<b>Escenario</b>	<b>Datos de Entrada</b>	<b>Resultado Esperado</b>
-----------	------------------	-------------------------	---------------------------

1	Actualización correcta	V -- Razón Social: ACOSA	Cliente actualizado. Se abre la página de lista de clientes.
2	Actualización correcta	V -- Razón Social: UBESA V -- RUC: 20259551022	Cliente actualizado. Se abre la página de lista de clientes.
3	Actualización correcta	V -- Razón Social: SUPAN S.A. V -- RUC: 1790005739001 V -- E-mail: info@supan.com	Cliente actualizado. Se abre la página de lista de clientes.
4	Error en la base de datos	N/A	Cliente no actualizado. Se abre la página de lista de clientes y se presenta un mensaje de error.
5	Cliente ya existe.	V -- Razón Social: ACOSA	Cliente no actualizado. Se despliega un mensaje indicando que ese cliente ya existe.
6	Cliente ya existe.	V -- RUC: 1790005739001	Cliente no actualizado. Se despliega un mensaje indicando que ya existe un cliente con ese RUC.
7	Campos obligatorios vacíos.	V -- Razón Social: Vacío	Cliente no actualizado. Se despliega un mensaje indicando que el campo es obligatorio.
8	E-mail inválido	I -- E-mail: info.com	Cliente no actualizado. Se despliega un mensaje indicando que el e-mail ingresado no es válido.

### ***F8.1 Registro de Pedidos***

#### ***Requerimientos:***

- Autenticación e ingreso al sistema con perfil administrador o asistente operativo

- Haber ingresado antes clientes, proveedores y productos

***Lista de Escenarios***

<b>ID</b>	<b>Escenario de caso de uso</b>	<b>Descripción</b>	<b>Excepción</b>
1	Registro correcto	El usuario ingresa los todos los datos correctamente.	
2	Error en la base de datos	Al guardar existe un error en la base de datos.	E3
3	Pedido ya existe.	Se ingresa un pedido ya existente.	E4
4	Campos obligatorios vacíos.	El usuario no llena campos que son obligatorios.	E1
5	Valor total inválido	El usuario ingresa un valor total inválido.	E2

***Matriz de casos de prueba***

<b>ID</b>	<b>Escenario</b>	<b>Datos de Entrada</b>	<b>Resultado Esperado</b>
1	Ingreso correcto	V -- # de Pedido: 001 V -- Proveedor: GO IN V -- Cliente: ACOSA V -- Valor total: \$3000	Pedido creado. Se abre la página de lista de pedidos.
2	Error en la base de datos	N/A	Pedido no creado. Se abre la página de lista de pedidos y se presenta un mensaje de error.
3	Pedido ya Existe	V -- # de Pedido: 001 V -- Proveedor: N/A V -- Cliente: N/A V -- Valor total: N/A	Pedido no creado. Se despliega un mensaje indicando que ese # de pedido ya existe.
4	Campos obligatorios vacíos.	V -- # de Pedido: Vacío V -- Proveedor: Vacío V -- Cliente: Vacío	Pedido no creado. Se despliega un mensaje indicando que los campos son obligatorios.
5	Valor total inválido	V -- # de Pedido: N/A	Pedido no creado. Se despliega un mensaje

		V -- Proveedor: N/A V -- Cliente: N/A I -- Valor total: s56	indicando que el valor total acepta solo números.
6	Valor total inválido	V -- # de Pedido: N/A V -- Proveedor: N/A V -- Cliente: N/A I -- Valor total: -5	Pedido no creado. Se despliega un mensaje indicando que el valor total acepta solo números mayores o iguales a cero.

### ***F8.3 Modificar Pedido***

#### ***Requerimientos:***

- Autenticación e ingreso al sistema con perfil administrador o asistente operativo
- Haber ingresado antes clientes, proveedores y productos

#### ***Lista de Escenarios***

<b>ID</b>	<b>Escenario de caso de uso</b>	<b>Descripción</b>	<b>Excepción</b>
1	Registro correcto	El usuario ingresa los todos los datos correctamente.	
2	Error en la base de datos	Al guardar existe un error en la base de datos.	E3
3	Pedido ya existe.	Se ingresa un pedido ya existente.	E4
4	Campos obligatorios vacíos.	El usuario no llena campos que son obligatorios.	E1
5	Valor total inválido	El usuario ingresa un valor total inválido.	E2

#### ***Matriz de casos de prueba***

<b>ID</b>	<b>Escenario</b>	<b>Datos de Entrada</b>	<b>Resultado Esperado</b>
1	Actualización correcta	V -- # de Pedido: 001 V -- Proveedor: GO IN	Pedido actualizado. Se abre la página de lista de pedidos.

		V -- Cliente: ACOSA V -- Valor total: \$3000	
2	Error en la base de datos	N/A	Pedido no actualizado. Se abre la página de lista de pedidos y se presenta un mensaje de error.
3	Pedido ya Existe	V -- # de Pedido: 001 V -- Proveedor: N/A V -- Cliente: N/A V -- Valor total: N/A	Pedido no actualizado. Se despliega un mensaje indicando que ese # de pedido ya existe.
4	Campos obligatorios vacíos.	V -- # de Pedido: Vacío V -- Proveedor: Vacío V -- Cliente: Vacío	Pedido no actualizado. Se despliega un mensaje indicando que los campos son obligatorios.
5	Valor total inválido	V -- # de Pedido: N/A V -- Proveedor: N/A V -- Cliente: N/A I -- Valor total: s56	Pedido no actualizado. Se despliega un mensaje indicando que el valor total acepta solo números.
6	Valor total inválido	V -- # de Pedido: N/A V -- Proveedor: N/A V -- Cliente: N/A I -- Valor total: -5	Pedido no actualizado. Se despliega un mensaje indicando que el valor total acepta solo números mayores o iguales a cero.

## 4.2 Fase de Implantación

En ésta fase se describe como fue el proceso de implantación del sistema. Para hacerlo, fueron necesarias tres etapas. La primera fue configurar el servidor y todo lo que eso conlleva. En la segunda se procedió a configurar la base de datos en el servidor y las seguridades. Posteriormente se subió la aplicación al sitio de pruebas para hacer la presentación al cliente. Finalmente una vez que el cliente dio el visto bueno a la aplicación, se subió la aplicación al sitio de producción.

#### **4.2.1 Configuración del servidor**

Lo primero que se hizo en esta etapa fue seleccionar el alojamiento web, debido a que los requerimientos demandaban una aplicación web, se seleccionó un servidor web privado provisionado por Dreamhost<sup>20</sup>. Este proveedor cuenta con todas las herramientas necesarias para poder instalar una aplicación hecha con PHP y MySQL. Además cuenta con un soporte inmediato que facilita el mantenimiento de las aplicaciones que se deseen subir.

Posteriormente se procedió a revisar que el servidor cuente con todo lo necesario para poder montar la aplicación. Se revisó que tenga instalado MySQL, PHP, Apache y después de verificar que no tenía instalado un acelerador de PHP se pidió al soporte de Dreamhost que lo instalen. Debido a que la aplicación debía contar con servidor de correo electrónico, se configuro una cuenta para el dominio de prueba.

#### **4.2.2 Configuración de la base de datos del servidor y seguridades**

Para la configuración de la base de datos, se comenzó con crear un usuario administrador para gestionar las bases del servidor, con una clave lo segura para que no pueda ser descubierta fácilmente. Luego, haciendo uso de PuTTY<sup>21</sup> se creó la base de datos para que esté lista para cargar el script con toda la definición de la misma. Para eso, se generó un script con la estructura de la base y otro con los datos iniciales como los de la tabla de países y súper administrador del sistema.

Una vez cargada la base y realizado las pruebas respectivas, se analizó la seguridad del servidor. Se hicieron pruebas básicas para sacar a la luz las posibles vulnerabilidades que se podrían presentar. Se realizaron pruebas de Cross Site Scripting<sup>22</sup>, SQL Injection<sup>23</sup> y Cross Site Request Foregery<sup>24</sup>. Hasta que finalmente se concluyo que el servidor era lo suficientemente seguro para que lo accedan hackers o cualquier otra persona que no sean los administradores del sitio.

---

<sup>20</sup> Dreamhost es un proveedor de alojamiento web y registro de dominios con base en Los Ángeles

<sup>21</sup> PuTTY es un cliente SSH que sirve para acceder al servidor haciendo uso de líneas de comando.

<sup>22</sup> XSS (Cross Site Scripting) es un tipo de ataque informático, que permite a una tercera parte...

<sup>23</sup> Sql Injection es un método de infiltración de código intruso que se vale de una...

<sup>24</sup> CSRF (Cross Site Request Foregery) o en español falsificación de petición en sitios cruzados...

### **4.2.3 Sitio de Pruebas y Producción**

Se definió el sitio de pruebas en el dominio [www.santiagobenitez.com](http://www.santiagobenitez.com). Como estaba planificado, se subió la aplicación con el procedimiento que recomienda Symfony (ver en anexos). Para esto se hizo uso de WinSCP, que es un cliente SFTP gráfico para Windows. Se cambió los archivos de configuración de la base de datos para que coincidan con la base del servidor. Luego se subió la aplicación de tal manera que el dominio apunte solo a los archivos del lado del cliente que son las hojas de estilos, archivos JavaScript e imágenes. De ésta manera, todos los archivos de configuración y de lógica del negocio se encuentran seguros y no son accesibles a través del internet.

Una vez que se tuvo el servidor y la base de datos configurados, la aplicación subida y el ambiente listo para ser probado, se realizó la presentación final al cliente. Se mostró cada uno de los módulos ya funcionando en la web y cada uno de los detalles que contienen. Una vez finalizada la presentación el cliente se vio muy complacido ya que este satisfacía todas las necesidades del negocio. Además le gustó mucho la interfaz gráfica por lo fácil de usar, la decoración y los efectos de los componentes. Por todo esto, el cliente aceptó sin ningún problema la aplicación y de esta manera subió el sistema a producción para empezar a usarla.

Ya con el visto bueno del cliente, se definió un nombre para el dominio y se lo compró por un año. Luego se alojó este dominio igualmente en el servidor de Dreamhost y se subió la aplicación de la misma manera que se lo subió al sitio de pruebas. Todo esto correspondiente a los estándares que recomienda Symfony para el desarrollo de aplicaciones web (Ver en anexos).

En cuanto al respaldo de la base de datos, haciendo uso de comandos, se configuró para sacar un script con toda la estructura y datos de la base cada hora. En la última hora del día, automáticamente se copia el script correspondiente para tener un respaldo diario. Y en el último día de la semana se genera un script correspondiente a ese día para tener un respaldo semanal. Con esto, se tendría respaldos por hora de las últimas 24 horas, respaldos diarios de la última semana y respaldos semanales permanentes. Adicionalmente a esto, para el caso del respaldo semanal, automáticamente se envía un correo electrónico notificando que el respaldo ha sido generado exitosamente.

#### 4.2.4 Tips de soporte del sistema

- Usuarios e ingreso al sistema:
  - Solo los usuarios con perfil “Administrador” pueden administrar usuarios.
  - Una vez que se crea un usuario, se le envía al correo electrónico las instrucciones para configurar una contraseña y poder ingresar al sistema.
  - Para reenviar un correo con las instrucciones para configurar la contraseña, se debe ingresar a la pantalla de lista de usuarios y presionar sobre reenviar correo en el usuario deseado.
  - En caso de olvidarse la contraseña, es posible recuperarla presionando sobre “Contraseña Olvidada” e ingresando la dirección de correo electrónico.
- Pedidos y Ordenes de Compra:
  - Es posible editar los pedidos y ordenes de compra en cualquier momento.
  - Para crear una orden de compra, es necesario cambiar el estado del pedido a “confirmado”, en ese caso se generará automáticamente la orden de compra y el estado del pedido ya no podrá ser modificado.
  - Una vez agregados los plazos, se calculará automáticamente la fecha de expiración de la factura, basándose en la fecha más nueva de los plazos.
  - Una vez que los pagos cubran el total a pagar, automáticamente la factura pasará ha estado “Pagada”.
  - Tanto los pedidos como órdenes de compra, poseen una sección de filtros para facilitar la búsqueda. Para habilitar estos filtros se necesita presionar sobre “+ Filtros” en la parte superior de la lista.
- Alertas:
  - Las alertas pueden ser configuradas por el usuario, las opciones seleccionadas servirán automáticamente como filtro para acceder a los pedidos y ordenes de compra requeridas.

- Es posible asignar prioridad a las alertas dependiendo su importancia, cada alerta será presentada con el color correspondiente a su prioridad.
- En caso de no existir coincidencias para una alerta determinada, ésta aparecerá de color gris.
- Clientes y Proveedores:
  - Es posible agregar cualquier número de contactos, tanto a clientes como proveedores. Para esto es necesario ingresar a la pantalla de lista de clientes o lista de proveedores y presionar sobre “Contactos” en el cliente que se desee agregar contactos.
  - Es posible acceder a los créditos tanto de los clientes como proveedores y configurarlos. Para esto es necesario ingresar a la pantalla de lista de clientes o lista de proveedores y presionar sobre “Contactos” en el cliente que se desee agregar contactos.
- General:
  - En las administraciones de clientes, proveedores, pedidos, ordenes de compra y productos, es posible adjuntar archivos y luego descargarlos sin ningún problema.
  - Un usuario sin las credenciales necesarias, no podrá acceder de ninguna manera a módulos que no le correspondan.

## **5. CONCLUSIONES Y RECOMENDACIONES**

En este capítulo se detallaran todas las conclusiones y recomendaciones una vez finalizado el proyecto.

### **5.1 Conclusiones**

En la actualidad las aplicaciones web se han constituido como uno de las principales estándares de desarrollo de todos los países, e influyen directamente en la economía y desarrollo social. La mayoría de los sistemas de automatización dependen o son controlados por software. Hoy en día, para tener un negocio estable, controlado y con visión de crecimiento, es necesaria la ayuda de una o varias aplicaciones. Esto obliga a utilizar técnicas y procedimientos que permitan obtener un software que sea funcional, confiable y de calidad, para satisfacer las necesidades del negocio.

Al haber crecido la industria del software, y teniendo en cuenta que las tecnologías y herramientas son cada vez más sofisticadas y fáciles de usar, es necesario todo el tiempo estar actualizando dichas herramientas y tecnologías para no quedarse atrás respecto a otros desarrolladores. Por lo que se concluye que la fase de selección de tecnologías previo al desarrollo de la aplicación es muy importante.

También es necesario darse cuenta que los principios de diseño que se utilizarán, varían dependiendo de algunos factores. Primero, de la aplicación a desarrollarse y su alcance. Segundo de las tecnologías y lenguajes de programación a utilizarse. Y finalmente, del equipo que va a desarrollar la aplicación. En definitiva, no existe un modelo general establecido para el desarrollo de software, únicamente se definen principios que deberán ser acogidos según las necesidades y requerimientos del proyecto.

Es importante establecer los ámbitos y objetivos que tendrá el proyecto, considerar soluciones alternativas e identificar restricciones técnicas. Sin tener esta información clara, es imposible desarrollar un plan de proyecto realista y que proporcione una guía para realizar el proyecto de una manera adecuada.

Tomando esto en cuenta, las primeras etapas son cruciales en la consecución de que el producto final sea de calidad, y se concluya en el tiempo establecido y concorde a los costos

establecidos. Si estas etapas no se les da la importancia debida, luego resulta mucho más difícil mantener el software y hacer modificaciones para adecuarse a las nuevas necesidades que vayan surgiendo. De igual manera, una mala planificación inicial puede tener consecuencias desastrosas en cuanto a costes y plazos de entrega.

Cuando se cambian los requerimientos en la mitad o al finalizar el proyecto de software, se producen demoras que pueden llegar a ser igual o mayor al tiempo que ya se ha venido trabajando. Por eso es muy importante la etapa de levantamiento y especificación de requerimientos y como las metodologías ágiles sugieren, estar lo más posible en contacto con el cliente. Para esto, incluir en la planificación una presentación al cliente una vez concluida cada una de las fases es muy importante. Para facilitar el entendimiento de lo que necesita el cliente, resulta de mucha utilidad hacer un buen levantamiento de procesos del negocio. Teniendo en cuenta también los procesos ajenos a la aplicación, pero que sirven para un mejor entendimiento de la solución.

Si para la aplicación que se va a construir, se estima que tendrá un gran número de distintas funcionalidades, es necesario dedicar una gran cantidad de tiempo para ver si ya existen librerías o extensiones que realizan esa funcionalidad. Como mucha gente sugiere, no es necesario reinventar la rueda. Para Symfony y JQuery existe un gran número de Plug-ins que agilitan de gran manera la programación y son muy fáciles de usar.

En cuanto a las pruebas, si no se las realiza siguiendo un esquema y una metodología, el producto final no tendrá la calidad adecuada. Esto se debe a que si se sigue un esquema, se analizan todos los escenarios posibles y es más fácil que no se escape ningún error. Es por eso que cada escenario que se presenta en las pruebas, es una posible falla. Gracias a las pruebas, en el presente proyecto, se analizó una serie de escenarios que no estaban contemplados y que pudieron resultar en errores.

## **5.2 Recomendaciones**

Una vez concluido el proyecto, quedaron a la vista una serie de recomendaciones que servirán para futuros proyectos con un enfoque similar. Las recomendaciones van relacionadas con todos los ámbitos tratados en el proyecto. Desde la planificación hasta el desarrollo, así como

también respecto a las técnicas, metodologías utilizadas, tecnologías y herramientas más apropiadas.

Primeramente, es recomendable para realizar un proyecto de software, utilizar alguna metodología ágil de desarrollo como Extreme Programming. Ayuda notablemente para sacar un proyecto en el menor tiempo posible y además se adapta con facilidad a los cambios que se van suscitando durante el proyecto.

Para recolectar la información, la metodología puede variar dependiendo la aplicación que se desee desarrollar. Si se requiere una aplicación para un negocio ya definido y el cliente tiene bastante claro la funcionalidad que le quiere dar, lo más recomendable es usar el método de la entrevista ya que es la manera más fácil de recolectar información para obtener la solución final.

Los modelos UML por otro lado, dan un gran aporte al proyecto y son esenciales para llevar a cabo un trabajo bien elaborado. Ya que sirven para especificar, construir, documentar y planificar un sistema de software. Haciendo uso de estos modelos, es mucho más sencillo desarrollar las aplicaciones. Así como un ingeniero civil tiene planos para construir un edificio o una casa, los ingenieros en sistemas hacen uso de estos modelos para guiarse y no desviarse de la lógica del negocio y de los requerimientos y necesidades del mismo.

Quizás una de las partes más importantes de una aplicación, es el modelamiento de la base de datos. Es el núcleo de todo sistema y de donde parte toda la información. Por lo que es fundamental analizar muy bien cada una de las tablas y relaciones al momento de diseñarla. Una base mal diseñada causa efectos fatales al momento de desarrollar la aplicación. MySQL es un motor de bases de datos, quizás no tan poderoso como Oracle o SQL Server, pero es el más adecuado para aplicaciones pequeñas y medianas. Es robusto, muy rápido, seguro y es software libre por lo que no tiene costo.

En cuanto a las tecnologías, es importante darse el mayor tiempo posible para seleccionarlas. Estas deben ir lo más acorde a la solución y además tienen que dentro de lo posible ser dominadas por el equipo de desarrollo. También deben ser lo más actuales posibles, ya que tecnologías antiguas harían que la aplicación que se está desarrollando quede obsoleta en menos tiempo. En este caso se utilizó PHP, que es un lenguaje muy recomendable para

construir aplicaciones web por su agilidad, seguridad, curva de aprendizaje relativamente rápida, documentación y adaptabilidad a HTML y javascript.

Por el lado del framework, realizar una aplicación sin uno de ellos es contraproducente, ya que éste ayuda a estructurar de una mejor manera los archivos dentro del proyecto, contiene módulos prefabricados y ayuda a mantener el sistema más seguro. Más aún si utiliza el patrón de arquitectura de software adecuado como es el caso de Symfony, que utiliza el modelo MVC. Symfony es muy aconsejable para proyectos grandes, ya que te facilita la generación de módulos y es compatible con Doctrine que es un ORM de los más poderosos que existen actualmente. Además provee la funcionalidad de trabajar sobre diferentes ambientes, de desarrollo, pruebas, producción. También incluye librerías para realizar pruebas unitarias e incluso un módulo para el ingreso al sistema y perfiles.

Otra recomendación importante, ya entrando al ámbito de la programación, es basarse en las buenas prácticas. Esto se refiere, a seguir los estándares recomendados por el lenguaje de programación y framework que se esté utilizando, comentar muy bien el código, programar de manera ordenada, no utilizar código innecesario, simplificar lo más posible, entre otras buenas prácticas.

Una de las más importantes y que actualmente se utiliza muy a menudo y lógicamente se utilizo en este proyecto, es la de no dar a cada uno de los componentes de la aplicación, más de una responsabilidad. De esta manera cada módulo es un componente de la aplicación y así dependa de otros, es muy fácil corregir e identificar rápidamente un error. Ya que si falla una funcionalidad en específico, se sabe exactamente que entidad es la responsable de la misma. Esto permite una excelente adaptabilidad a cambios y ayuda un mejor entendimiento.

Hablando de la Escuela de Sistemas de la Facultad de Ingeniería de la Pontificia Universidad Católica del Ecuador, tengo algunas recomendaciones que las mencionaré a continuación. Creo que sería importante considerar actualizar el Pensum de semestre a semestre, de esta manera los alumnos estarían siempre a la par con las últimas tecnologías y no tomarían materias que actualmente se encuentran obsoletas. También sería interesante si se agregan materias opcionales de ciertos temas en específico que sean de interés de los estudiantes y que estén relacionadas a lo que los estudiantes desean para especializarse.

## **Glosario**

### **A**

**AJAX (Asynchronous Javascript And Xml):** Ajax es una técnica de desarrollo web para crear aplicaciones interactivas. Esto significa que es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

**Arquitectura de Software:** La Arquitectura de Software es la forma en la que se organizan los componentes de un sistema, interactúan y se relacionan entre sí y con el contexto, aplicando normas y principios de diseño y calidad.

### **B**

**Base de datos relacional:** Una base de datos relacional es un conjunto ordenado de datos, los cuales normalmente están almacenados en uno o más archivos. Los datos están estructurados en tablas, y estas pueden tener referencias cruzadas. La existencia de estas referencias o relaciones es lo que da el nombre a este tipo de bases de datos.

### **C**

**Cross Site Scripting:** XSS (Cross Site Scripting) es un tipo de ataque informático, que permite a una tercera parte inyectar en páginas web código javascript o en otro lenguaje script similar.

**Cross Site Request Forgery:** CSRF (Cross Site Request Foregery) o en español falsificación de petición en sitios cruzados. Es un tipo de ataque malicioso de un sitio web en el que comandos no autorizados son transmitidos por un usuario en el cual el sitio web confía.

### **F**

**Framework:** Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

## H

**HTML5:** HTML5 es la nueva versión de HTML que viene con una serie de mejoras respecto a su predecesor. Posee nuevas etiquetas, entre las más importantes, etiquetas de canvas, audio, y de video. Y lo más importante es que permitirá independizarse de una serie de lenguajes que antes eran necesarios para poder realizar algunas funciones específicas.

**HTTP:** HTTP significa “Protocolo de Transferencia de Hiper Texto”, y es gracias a este que nuestro ordenador es capaz de recibir y enviar ficheros de hipertexto de WWW, más comúnmente conocidos como páginas web. Utilizando este protocolo, diferentes tipos de ordenadores y sistemas operativos pueden visualizar de forma correcta la misma información.

## I

**IDE (Integrated Development Environment):** IDE es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

**Ingeniería de Software:** Ingeniería de Software es aquella que a través de técnicas ordenadas y metodologías de desarrollo y mantenimiento, se obtiene un producto de software de calidad

## O

**ORM (Object Relation Mapper):** Un ORM es una técnica de programación que permite convertir datos entre el sistema utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

## P

**Plan de entrega:** El plan de entrega es un cronograma definido en donde están especificadas las fechas de finalización de cada una de las iteraciones utilizando un modelo XP. Sirve principalmente como guía y para ir viendo si el avance del proyecto va acorde a lo planificado.

**Procesos ágiles de desarrollo de software:** Procesos ágiles de desarrollo de software son aquellos que a través de un marco de trabajo conceptual de la ingeniería de software, promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto

**Pruebas Unitarias:** En programación, una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego, con las Pruebas de Integración, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

## S

**Software Libre:** El Software de fuente abierta o Software Libre es un tipo particular de software que le permite al usuario el ejercicio de cuatro libertades básicas: ejecutarlo con cualquier propósito, estudiar como funciona y adaptarlo a sus necesidades, distribuir copias, mejorarlo, y liberar esas mejoras al público.

**SQL Injection:** Sql Injection es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar consultas a una base de datos.

## U

**URL (Uniform Resource Locator - Localizador Uniforme de Recursos):** Url es la cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en el internet.

**UML (Unified Modeling Language):** Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

## **Y**

**YAML:** YAML es un formato de serialización de datos legible por humanos inspirado en lenguajes como XML, C, Python, Perl, así como el formato para correos electrónicos especificado por el RFC.

## Trabajos citados

1. <http://es.wikipedia.org>. [En línea]
2. **Joskowicks, José.** *Reglas y Prácticas en eXtreme Programming*. 2008.
3. **Fernandez, Jaime.** *Introducción a las aplicaciones WEB*. México DF : 2008.
4. **Varios.** *Administración de Bases de Datos MySQL 5.1*. Girona. s.l. : Universidad de Girona, 2009.
5. **Benítez, Santiago.** *Disertación de grado titulada: Desarrollo, implementación e implantación del sistema de administración de clientes, proveedores, productos y gestión de contratos para empresa intermediaria de productos químicos, Quito, Ecuador*. s.l. : P.U.C.E., 2013.

## **Agradecimientos**

Expreso mis más sinceros agradecimientos, primeramente a la Pontificia Universidad Católica del Ecuador y todos los que la conforman, que en estos años de estudio me han permitido forjarme y consolidarme como un profesional de principios y valores. A mis correctores Ing. Alfredo Calderón e Ing. Javier Condor que han brindado parte de su tiempo para poder conseguir un trabajo de calidad. Especialmente a mi director Ing. Fabián de la Cruz, que siempre con muy buena predisposición y buenos consejos me permitió concluir esta etapa de mi carrera de la mejor manera posible.

También agradezco a mis amigos y compañeros que siempre estuvieron colaborando afuera y adentro de las aulas de clase. Y finalmente muchas gracias a las personas más importantes de mi vida. A mis hermanas Trinidad y Alejandra, a mi madre Rocío y a mi padre Santiago, que son mi mayor inspiración y que con mucha paciencia estuvieron siempre apoyándome sin importar las circunstancias.