

# UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



*Excelencia que trasciende*

**DELVALLE**  
GRUPO EDUCATIVO

Sistema de compras programadas y estadística de ventas  
en línea para FPK electrónicos

Erick Francisco Marroquín Rodríguez

Guatemala

2011



Sistema de compras programadas y estadística de  
ventas en línea para FPK electrónicos

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

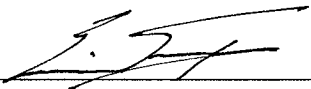
Sistema de compras programadas y estadística de ventas  
en línea para FPK electrónicos

Trabajo de graduación en modalidad de prácticas supervisadas  
presentado por Erick Francisco Marroquín Rodríguez para optar al  
grado académico de Licenciado en Ciencias de la computación

Guatemala

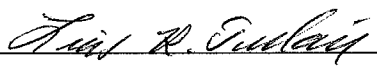
2011

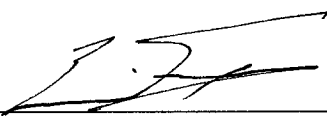
Vo. Bo.:

(f)   
Ingeniero Erwin Flohr

Tribunal Examinador:

(f)   
Ingeniera Maria Mercedes Zagui

(f)   
Ingeniero Luis Roberto Furlán

(f)   
Ingeniero Erwin Flohr

Fecha de aprobación de examen de graduación:

Guatemala, 7 de diciembre de 2011

## **PREFACIO**

La idea del proyecto surge con la necesidad de estandarizar y regular la forma en que la empresa realiza sus compras y maneja a sus clientes. Personalmente éste fue un proyecto bastante gratificante en el que pude poner en práctica muchos de los conocimientos estudiados en la universidad junto con la creatividad y la experiencia aprendida en la vida laboral.

Este trabajo me llena de orgullo ya que en él se plasman meses de arduo trabajo, desvelo, planificación y desarrollo. Es un proyecto que empecé de cero y, poco a poco, con dedicación y trabajo duro lo fui moldeando y construyendo para darle la forma final.

Quisiera aprovechar esta oportunidad para agradecer a las maravillosas personas en mi vida. Sin el apoyo de muchos ustedes no estaría aquí en este momento.

Quisiera agradecer a mi padre el Arquitecto Omar Marroquín ya que sin su apoyo incondicional y sacrificio no hubiera tenido acceso a educación superior de la calidad que recibí, pero más que nada le agradezco haber estado ahí para mí cuando lo necesité, no sólo durante los años de la carrera universitaria sino en los momentos más adversos de mi vida.

A mi hermana Sofía, quisiera agradecerle por ser el mejor ejemplo tanto de estudiante, de mujer y de persona. Me enorgullece saber que es una persona íntegra,

completa y ambiciosa por lo que le deseo todo el éxito, que yo sé que puede lograr y el éxito que se merece, pero sobre todo le deseo felicidad en todo lo que haga en su vida.

A mis tías, Ana, Thelma, Magda y Sonia les agradezco su esfuerzo y dedicación a nosotros. Ellas son uno de los mejores ejemplos que puedo tener y son las que me han enseñado el verdadero significado de la familia. Gracias por adoptarnos y darnos todo sin remordimientos y sin esperar nada a cambio. Todas son unas grandes mujeres y me enorgullece mucho ser parte de sus corazones.

A mi asesor de tesis Erwin Flohr quiero agradecerla la enorme oportunidad que me ha dado para poder llevar a cabo este trabajo en su empresa. En él he encontrado un profesional íntegro y respetable que me ha guiado y me ha dado la confianza y responsabilidad de trabajar con él en este proyecto.

Finalmente y, no por eso menos importante, quiero dedicarle este trabajo a mi madre Edna Rodríguez, que a pesar de no estar conmigo físicamente guía mis pasos y es la fuerza que me hace despertar todas las mañanas y querer ser excelente en todo lo que hago como ella lo fue. A ti madre te debo todo lo que soy porque sin ti como ejemplo nunca hubiera tenido los deseos de sobrepasar mis expectativas días tras día.

## Contenido

<b>PREFACIO .....</b>	<b>vi</b>
<b>LISTA DE CUADROS .....</b>	<b>xi</b>
<b>RESUMEN.....</b>	<b>xv</b>
<b>I. INTRODUCCIÓN.....</b>	<b>1</b>
<b>II. OBJETIVOS .....</b>	<b>2</b>
<b>A. Generales .....</b>	<b>2</b>
<b>B. Específicos.....</b>	<b>2</b>
<b>III. MARCO TEÓRICO.....</b>	<b>3</b>
<b>A. Patrón modelo-vista-controlador.....</b>	<b>3</b>
1. Definición.....	3
2. Funcionamiento.....	3
a. Modelo .....	4
b. Vista .....	5
c. Controlador.....	5
3. Comunicación.....	6
4. Flujo de control.....	7
a. Modelo pasivo .....	7
b. Modelo activo.....	8
5. Framweorks .....	9
a. Symfony .....	10
<b>B. Base de datos.....</b>	<b>10</b>
B. Modelo relacional.....	10
C. Normalización .....	11
a. Llaves.....	12
<b>C. Estadística.....</b>	<b>12</b>
1. Crecimiento vegetativo.....	12
<b>IV. MARCO METODOLÓGICO.....</b>	<b>14</b>
<b>A. Requerimientos .....</b>	<b>14</b>
<b>B. Herramientas .....</b>	<b>15</b>
<b>V. MÓDULO DE BASE DE DATOS.....</b>	<b>17</b>
<b>A. Descripción.....</b>	<b>17</b>
<b>B. Requerimientos .....</b>	<b>17</b>
<b>C. Diseño.....</b>	<b>18</b>

1. Diagrama.....	18
2. Diccionario de datos.....	20
a. access_log.....	20
b. backend_modules.....	20
c. backend_user.....	21
d. clients.....	21
e. company.....	22
f. error_log.....	22
g. event_log.....	23
h. module_permission.....	23
i. order_notification.....	24
j. permission.....	24
k. product_category.....	25
l. product_relation.....	25
m. products.....	25
n. role.....	26
o. role_module_permission.....	26
p. sale_detail.....	27
q. sale_history.....	27
r. sale_status.....	27
s. sales.....	28
t. tmp_cart.....	28
u. user_status.....	29
3. Permiso de roles.....	29
<b>VI. MÓDULO DE ADMINISTRACIÓN INTERNA.....</b>	<b>33</b>
<b>A. Descripción.....</b>	<b>33</b>
<b>B. Requerimientos.....</b>	<b>34</b>
<b>C. Diseño.....</b>	<b>35</b>
1. Roles.....	35
2. UML.....	37
3. Interfaces.....	41
<b>VII. MÓDULO DE VENTA EN LÍNEA.....</b>	<b>45</b>
<b>A. Descripción.....</b>	<b>45</b>
<b>B. Requerimientos.....</b>	<b>46</b>
<b>C. Diseño.....</b>	<b>47</b>
1. Roles.....	47
2. UML.....	47
3. Interfaces.....	48

<b>VIII. MÓDULO DE REPORTES Y ESTADÍSTICA .....</b>	<b>53</b>
<b>1. Descripción .....</b>	<b>53</b>
<b>2. Requerimientos .....</b>	<b>54</b>
<b>3. Diseño.....</b>	<b>54</b>
1. Comportamiento de productos .....	54
2. Reportes de ventas .....	55
3. Records de ventas .....	56
4. Sugeridos de compra .....	57
a. Pedidos realizados .....	57
b. Sugeridos de compra .....	57
c. Proyecciones de ventas .....	58
<b>5. Pruebas de efectividad .....</b>	<b>61</b>
a. Descripción de la prueba .....	62
b. Resultados de la prueba .....	62
c. Interpretación de resultados .....	63
<b>IX. DISCUSIÓN .....</b>	<b>65</b>
<b>X. CONCLUSIONES .....</b>	<b>67</b>
<b>XI. RECOMENDACIONES .....</b>	<b>69</b>
<b>XII. REFERENCIAS BIBLIOGRÁFICA .....</b>	<b>70</b>

## LISTA DE CUADROS

Tabla 1 – Campos de la tabla access_log .....	20
Tabla 2 – Campos de la tabla backend_modules .....	21
Tabla 3 – Campos de la tabla backend_user.....	21
Tabla 4 – Campos de la tabla clients .....	22
Tabla 5 – Campos de la tabla company.....	22
Tabla 6 – Campos de la tabla error_log.....	23
Tabla 7 – Campos de la tabla event_log.....	23
Tabla 8 – Campos de la tabla module_permission .....	24
Tabla 9 – Campos de la tabla order_notification.....	24
Tabla 10 – Campos de la tabla permission.....	24
Tabla 11 – Campos de la tabla product_category .....	25
Tabla 12 – Campos de la tabla product_relation .....	25
Tabla 13 – Campos de la tabla products .....	26
Tabla 14 – Campos de la tabla role .....	26
Tabla 15 – Campos de la tabla role_module_permission .....	27
Tabla 16 – Campos de la tabla sale_detail.....	27
Tabla 17 – Campos de la tabla sale_history .....	27
Tabla 18 – Campos de la tabla sale_status.....	28
Tabla 19 – Campos de la tabla sales.....	28
Tabla 20 – Campos de la tabla tmp_cart .....	28
Tabla 21 – Campos de la tabla user_status.....	29
Tabla 22 – Permisos de roles principales sobre las tablas.....	30

Tabla 23 – Permisos de roles secundarios sobre las tablas.....	31
Tabla 24 - Resultados de prueba estadística.....	63

## LISTA DE FIGURAS

Figura 1 – Diagrama de comunicación del patrón MVC .....	6
Figura 2 – Flujo de control del modelo pasivo, patrón MVC .....	8
Figura 3 – Flujo de control del modelo activo, patrón MVC .....	9
Figura 4 – Cálculo de la tasa de crecimiento vegetativo.....	13
Figura 5 – Cálculo de la proyección de población mediante el uso de la tasa de crecimiento vegetativo .....	13
Figura 6 – Diagrama de flujo del proceso de autenticación de usuario .....	36
Figura 7 – Diagrama de casos de uso del rol Administrador .....	37
Figura 8– Diagrama de casos de uso del rol operativo.....	38
Figura 9 – Diagrama de casos de uso del rol cliente .....	39
Figura 10 – Diagrama de casos de uso del rol reports y estadística .....	40
Figura 11 – Interfaz de login backend.....	42
Figura 12 – Interfaz de catálogo de productos.....	43
Figura 13 – Forma para ingreso de datos.....	44
Figura 14 – Diagrama de casos de uso del rol cliente en el frontend .....	48
Figura 15 – Vista de la carretilla de compra.....	49
Figura 16 – Vista de producto individual de la carretilla.....	50
Figura 17 – Vista de la carreta de compra de cada usuario .....	51
Figura 18 – Vista del reporte de compras disponible para los clientes.....	52
Figura 19 – Vista del módulo de estadística y sugeridos de compra.....	59
Figura 20 – Vista del reporte de comportamiento de producto .....	60

Figura 21 – Vista del reporte de productos más vendidos..... 61

## RESUMEN

El objetivo de la práctica supervisada es desarrollar un sistema en línea que permita a FPK electrónicos brindarle a sus clientes mayoristas un portal de pedidos en línea para que estos realicen sus compras de forma anticipada, cómoda y segura. Junto con este portal es necesario brindarle a la administración las herramientas necesarias para cumplir con la demanda de sus clientes. Para poder llegar a hacer esto se desarrolló un sistema de reportes y estadística que ayuda a los administradores a tomar la mejor decisión de compra a los distribuidores.

Por medio del sistema de estadística se brindan las herramientas necesarias para que las compras que realiza FPK a sus proveedores sean más exactas y con un margen de error menor al actual. Para lograr esto se utilizan proyecciones estadísticas que evalúan las ventas de los meses anteriores y con base al movimiento de los productos en el mercado sugieren que tanto producto se debe comprar.

Junto con las proyecciones se maneja una red sugerida de productos con base en relaciones de los mismos. Esto surge con la necesidad de comprar productos similares en cantidades similares para surtir la demanda que se tiene por parte de los clientes tanto mayoristas como de tienda. Esta herramienta le permite a los encargados de inventario crear relaciones de producto que al momento de realizar un pedido muestran alertas de los productos similares para que los encargados de las compras tenga una idea de lo que necesitan comprar.

Finalmente la posibilidad de tener un sistema de reportes sobre todas las compras realizadas ayuda a la gerencia de FPK a tener una noción de cómo es el movimiento de los productos y el comportamiento de los mismos en el mercado. Para lograr esto se creó un módulo de reportes que, de una forma gráfica y ordenada muestra el comportamiento de los productos y de los clientes.

Junto con las necesidades de los portales nace la necesidad de construir un administrador del sitio. Este administrador le brinda al encargado del mantenimiento de los portales las herramientas necesarias para poder administrar el contenido de la tienda, las categorías de productos y las relaciones de los mismos. El módulo de administración interna surte estas necesidades y pone a disposición de FPK todas estas herramientas y hace que la administración del sitio sea una tarea fácil y que se lleve a cabo de una forma amigable.

# I. INTRODUCCIÓN

Para el desarrollo de esta aplicación se utilizaron herramientas para programación web. El proyecto en su totalidad es un conjunto de sitios para la administración de un portal de venta en línea como un portal para ver reportes y estadística concernientes a las ventas. El objetivo principal del proyecto fue construir un sistema completo que permita a los clientes mayoristas realizar pedidos por medio de un portal web. Luego estos pedidos son presentados en una forma gráfica y ordenada para que los administradores de FPK Electrónicos puedan utilizarlos como base para realizar las compras del período seleccionado.

La necesidad del proyecto nace cuando la administración de FPK se percató que sus compras no tienen una concordancia con la demanda de sus clientes por lo que muchas veces no logra llegar a surtir sus necesidades, lo que representa pérdida de ingresos y descontento en sus clientes. Teniendo un sistema de pedidos se puede determinar exactamente cuánto producto debe comprarse ya que los números mostrados ya son productos vendidos. Al tener solucionada la parte de los clientes mayoristas se le puede dar más importancia a los clientes de retail que son más inestables y dependen de factores externos que no pueden ser controlados por la empresa.

El proyecto consistió en crear estos portales, tanto el de venta en línea como el de estadística y reportes, cumpliendo así con la delimitación del proyecto y llegando a cumplir la necesidad de poder monitorear de una forma más exacta la compra de productos por medio de los clientes mayoristas.

## II. OBJETIVOS

### A. Generales

1. Investigar qué herramienta, framework o biblioteca se puede utilizar para facilitar el desarrollo del sistema propuesto.
2. Diseñar una herramienta que FPK pueda dar a sus clientes mayoristas para que estos puedan reservar los productos que requieran.
3. Diseñar un sistema administrativo con el cual los trabajadores de FPK puedan tener el control del funcionamiento del sistema.
4. Investigar los diferentes modelos estadísticos para determinar cuál es que mejor se adecua a las necesidades del sistema.

### B. Específicos

1. Profundizar en el uso del framework Symfony para el desarrollo de aplicaciones web.
2. Diseñar un portal con módulos de venta en línea y estadística de compras en el que los clientes de FPK puedan realizar pedidos y monitorear sus ordenes.
3. Diseñar un backend para administrar completamente el sitio, desde manejo de inventario hasta control de usuarios.
4. Utilizar el modelo de proyecciones de ventas empleando una tasa de crecimiento vegetativo para poder predecir las ventas de productos basándose en las ventas pasadas.

### III. MARCO TEÓRICO

#### A. Patrón modelo-vista-controlador

**1. Definición.** El modelo vista controlador es un patrón de arquitectura para el desarrollo de software. Entre sus ventajas está la separación de la lógica de la aplicación con la interfaz para la interacción con el usuario. Esto permite individualizar y separar la tarea de desarrollo de ambas partes manteniendo la facilidad para hacer la integración de las mismas. Otro beneficio del patrón modelo vista controlador es la capacidad de reutilizar código lo que optimiza las aplicaciones. (Pavón, 2009)

Actualmente existen diversas implementaciones del patrón MVC. La ventaja del mismo es la versatilidad para implementación que posee, es decir ya que es un modelo de arquitectura teórico puede implementarse independientemente del lenguaje y si se utiliza alguna herramienta de apoyo o si se desarrolla de la nada. Es por esto que el patrón ha ganado popularidad ya que muchos desarrolladores pueden adaptar este patrón en el desarrollo de sus aplicaciones en .NET, Java, HTML y PHP.

**2. Funcionamiento.** El patrón modelo vista controlador consta de tres elementos básicos que, como su nombre lo indica son: el modelo, las vistas y el controlador. En una aplicación construida con las bases de esta arquitectura se tiene un modelo único, muchas vistas y muchos controladores. Esta organización brinda al desarrollador mayor libertad de organizar el flujo de desarrollo de la aplicación. (Pavón, 2009)

**a. Modelo.** El modelo se refiere a los datos en sí, ya sea almacenados en una base de datos o guardadas en archivos. Generalmente es una base de datos que contiene toda la data que se manipulará y se desplegará dentro de la aplicación. El modelo es la entidad que se comunica directa y exclusivamente con los datos. Para poder comunicarse con los datos es necesario que también cuente con todos los procesos y toda la información para establecer una conexión entre la aplicación y los datos físicos. (Pavón, 2009)

El modelo es el encargado de regular las reglas de comunicación entre los usuarios y los datos; es por eso que para cada necesidad de ya sea guardar datos nuevos, editar datos existentes o de borrar datos el modelo debe constar con una definición de estos procedimientos para poder llevarlos a cabo. Es necesario que en el modelo se detallen todas las acciones que estarán a disposición del usuario ya que si no existe esta definición no se podrá crear una comunicación entre el usuario y la data ya que esta es la función exclusiva del modelo. (Pavón, 2009)

No solamente se puede dar el caso en el que la comunicación sea entre el modelo y una base de datos. Por lo general tiende a confundirse que el modelo solamente almacena procedimientos que serán ejecutados sobre una base de datos. La mejor forma de entender este concepto es llegando a un nivel de abstracción más alto en el que se puede determinar que los datos, almacenados en la forma en que estén deben siempre comunicarse con los usuarios por medio de un modelo. Ya sea que éste almacene las consultas y procedimientos en SQL para una base de datos, si contiene la información necesaria para comunicarse con un web service remoto o si simplemente guarda el árbol de directorio del archivo en el que se debe leer y escribir. (n.a., 2009)

**b. Vista.** Las vistas se refieren a las partes de la aplicación que tienen una comunicación directa con el usuario. Debido a que este patrón no depende de ningún lenguaje de programación establecido, las vistas pueden variar y tomar la forma de varias entidades que se comunican con un usuario y poseen un GUI. Los mejores ejemplos de una vista son páginas web en HTML o PHP, Applets de Java o interfaces de .NET. (Pavón, 2009)

Es en la vista que el usuario puede darle formato a los datos recibidos por el modelo, aunque esta comunicación no es en forma directa. Toda la interacción del usuario con los datos se lleva a cabo en la vista. Los datos ingresados nuevos para ser almacenados son primero recibidos en una lista y se procesan por medio del controlador y se almacenan físicamente utilizando el modelo. (n.a., 2009)

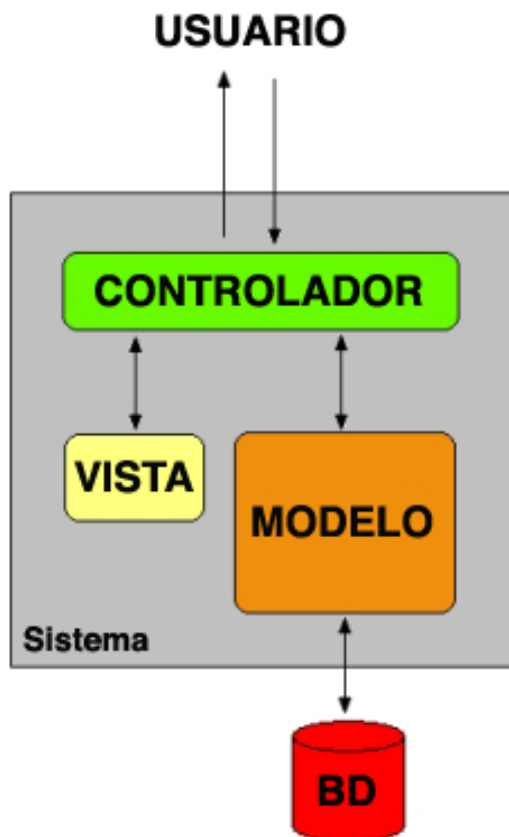
Es importante no confundir la vista con cualquier tipo de GUI. Las vistas muestran el resultado de una acción ya sea que se involucre data almacenada o calculada en ese instante mientras que un GUI, por ejemplo una página web puede reutilizar código como el encabezado o el pie de página. En el uso del MVC en un ambiente de PHP las vistas son renderizadas dentro de las plantillas, que en este caso serían el GUI. (n.a., 2009)

**c. Controlador.** Los controladores son el medio de comunicación entre las vistas y el modelo. Es en los controladores en donde se llevan a cabo el procesamiento dinámico de los datos y se generan las estructuras de datos con los resultados de las consultas a la data física. Todas las acciones que se realizan dentro de una vista son manejadas por el controlador que finalmente realiza la acción deseada y regresa el resultado por medio de la misma u otra vista. (Pavón, 2009)

Por motivos de seguridad y para mantener la autonomía de cada entidad del patrón el controlador es el único que puede mantener una relación con el modelo. Es por eso que cuando una vista solicita una acción que requiera data, el controlador debe establecer una conexión con el modelo, solicitar la acción deseada, obtener el resultado y enviarlo de regreso a la vista. (n.a., 2009)

**3. Comunicación.** Para que el patrón MVC se cumpla en su perfección existen normas en cómo se debe llevar a cabo la comunicación de las entidades involucradas. El siguiente diagrama muestra cómo debe ser la comunicación:

**Figura 1 – Diagrama de comunicación del patrón MVC**



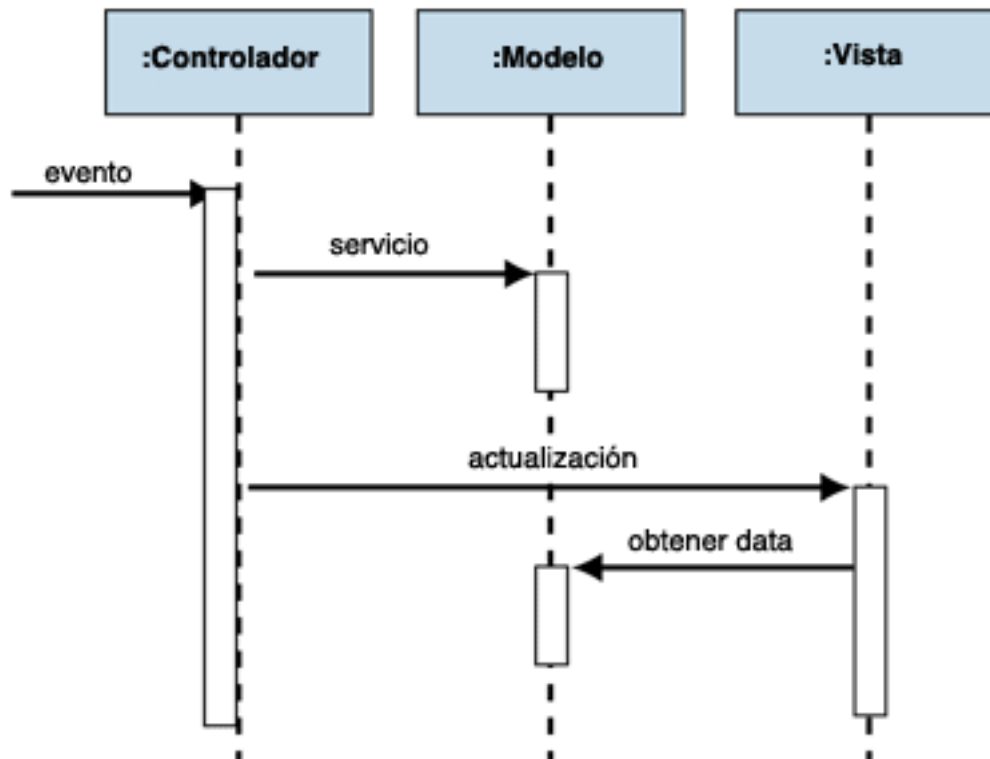
Como es posible observar en el diagrama el usuario interactúa de una forma directa con el controlador por medio de una vista. Dependiendo de la complejidad o la necesidad del usuario así va a ser la necesidad de conectarse a una base de datos por medio del modelo. (n.a., 2009)

Utilizar este modelo garantiza que el desarrollo de la aplicación sea lo más ordenado posible ya que fuerza a tener en cada entidad solo la información necesaria para el funcionamiento del sistema. Junto con el orden se evita correr riesgo de tener conexiones abiertas y accesos a base de datos en el mismo archivo que el usuario está manipulando, si los sistemas están aislados será más difícil la tarea de acceder a información restringida y/o privada.

**4. Flujo de control.** El flujo de control está determinado por qué tipo de patrón MVC se esté implementando. Existen dos patrones diferentes del patrón MVC y cada uno de ellos tiene su propio flujo de control definido y diferente al otro. Estos dos patrones son los siguientes:

**a. Modelo pasivo.** El modelo pasivo del patrón MVC se utiliza cuando el controlador manipula exclusivamente la comunicación con el controlador. En este patrón el controlador se comunica con los datos por medio del modelo y luego informa a las vistas del resultado obtenido. Al utilizar este patrón se garantiza que el modelo estará completamente independiente de las vistas y los controladores. (Microsoft, MSDN, 2011)

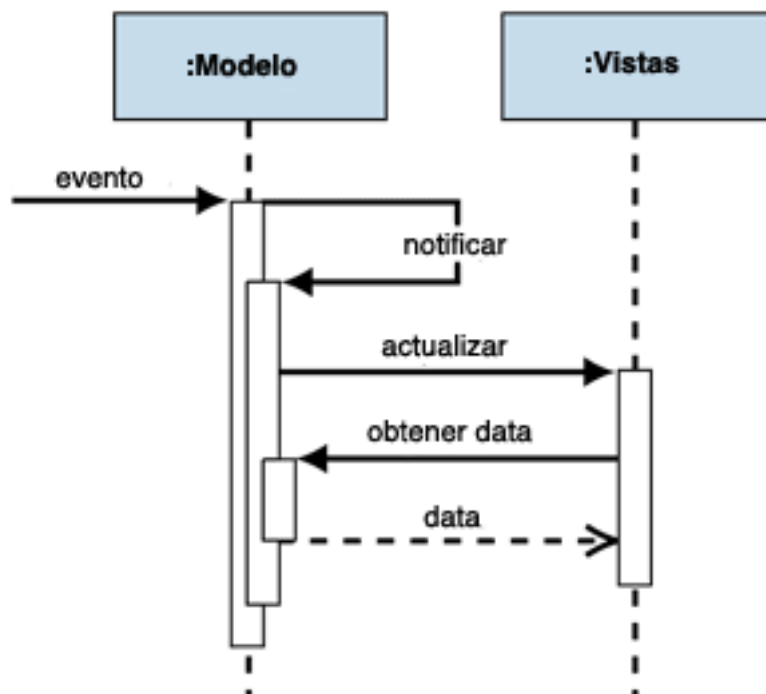
Figura 2 – Flujo de control del modelo pasivo, patrón MVC



El flujo de control de modelo pasivo, como se muestra en el diagrama involucra a las tres entidades que participan en el patrón MVC. El controlador recibe un evento que se da en el sistema. Este evento puede ser el clic de un botón que marca una forma para que sea procesada o cualquier evento del teclado. Cuando el controlador recibe el evento le solicita al modelo la ejecución de un servicio. El modelo realiza el servicio y envía la data a la vista para que esta despliegue el resultado mediante un proceso de actualización que se da desde el controlador. (Microsoft, MSDN, 2011)

**b. Modelo activo.** A diferencia del modelo pasivo en el que la comunicación de las vistas y el modelo deber hacerse por medio del controlador, en el modelo obvia la necesidad de constar con un intermediario para llevar a cabo la comunicación. (Microsoft, MSDN, 2011)

Figura 3 – Flujo de control del modelo activo, patrón MVC



El flujo de control del modelo activo en el patrón MVC distribuye las funciones del controlador dentro del modelo por lo que el sistema se convierte en un modelo de dos entidades que se comunican entre sí.

**5. Frameworks.** Un framework es una colección de librerías, clases y/o recursos que forman una base para la construcción de software. Actualmente se ha tendido a utilizar frameworks de desarrollo que implementen el patrón modelo-vista-controlador.

En el desarrollo web existe una gran variedad para poder escoger en cuestión de frameworks de desarrollo. Algunos son más completos que otros y están desde los más simples que solamente brindan una estructura para desarrollar hasta los más complejos que poseen librerías más completas para trabajar.

**a. *Symfony.*** Symfony es un framework desarrollado en PHP y para crear aplicaciones en PHP. Actualmente está lanzando su versión número 2 a la que le iniciarán a brindar soporte total en el año 2012. (SensioLabs, 2009)

Este framework cuenta con un set de librerías listas para usarse como lo es la integración con AJAX, también posee una poderosa línea de comandos para automatizar la generación de proyectos y desarrollo de aplicaciones. Una de las ventajas del framework es la abstracción que tiene con los datos ya que permite mantener completamente separadas la capa de data y la aplicación en sí. (SensioLabs, 2009)

La comunidad del framework es bastante extensa por lo que la documentación es lo suficientemente completa para solucionar cualquier problema que pueda presentarse. Es esta misma comunidad que mantiene una amplia librería de plugins que hacen más robusta las aplicaciones creadas. (SensioLabs, 2009)

## **B. Base de datos**

**1. Modelo relacional.** El modelo relacional es un modelo para el almacenamiento de datos dinámicamente en una base de datos. Su principal característica es que modela relaciones en tablas creadas por filas y columnas. Cada una de esta relación se da entre las entidades que representan la estructura de la base de datos. (Quiroz, 2003)

La teoría de conjuntos juega un papel clave en el desarrollo del modelo relacional ya que las relaciones del mismo son análogas a las relaciones de los conjuntos. Es por esto que los datos son almacenados de una forma en la que reamente no importa el orden de los datos como lo sería un modelo de jerarquías. (Quiroz, 2003)

El valor de implementar este modelo es que por medio del mismo se pueden representar datos de una forma más intuitiva y simple ya que las relaciones pueden ser explicadas de una forma más natural. Junto con la estructura fácil de entender el modelo cuenta con información sobre la estructura de la base de datos llamada metadata. La metadata ayuda a entender la estructura, facilitar los cambios sobre la misma y minimiza los posibles conflictos con las estructuras ya creadas. (n.a., El modelo relacional, 2007)

**2. Normalización.** La normalización se refiere al proceso de aplicar las reglas de normalización a una base de datos que cumple con el modelo relacional. Como principal objetivo es construir bases de datos más robustas, libres de errores, integras y no redundantes. (n.a., El modelo relacional, 2007)

Es mediante la normalización que se determinan ciertas dependencias para categorizar las relaciones de las entidades. La principal dependencia es la funcional que establece la existencia de una dependencia entre uno o más atributos que permiten determinar más información en base a estos. (n.a., El modelo relacional, 2007)

La dependencia funcional se puede dividir en tres tipos diferentes que permiten un mejor entendimiento de las reglas de la normalización. La primera es la dependencia funcional reflexiva la cual determina que partiendo de un atributo siempre se puede deducir el mismo. (Microsoft, Fundamentos de la normalización en base de datos, 2007)

La segunda forma de la dependencia funcional es la transitiva que funciona exactamente igual que la transitividad matemática en la que si un atributo deduce a otro y a su vez el segundo atributo deduce a un tercero entonces el primer atributo también deduce al tercero. (Microsoft, Fundamentos de la normalización en base de datos, 2007)

La tercera y última propiedad es la aumentativa. La dependencia funcional aumentativa introduce una variable a la deducción por lo que si una variable se multiplica por un atributo, entonces el parámetro que afecta al atributo si es aplicado a todos los atributos que el primero deduce se dice que existe equivalencia en la deducción. (Microsoft, Fundamentos de la normalización en base de datos, 2007)

**a. Llaves.** Las llaves son elementos de una tabla que permiten determinar el comportamiento de las relaciones. Existen diferentes tipos de llaves y estas se detallan a continuación:

- Llave primaria: es un identificador único de cada fila de una tabla.
- Llave candidata: llaves únicas que pueden ser utilizadas como llaves primarias.
- Llave foránea: es el identificador de una llave primaria de otra tabla ajena a la en la que se encuentra.
- Llave alternativa: es una llave candidata que no fue seleccionada como llave primaria.
- Llave compuesta: es una llave que se crea a partir de más de una columna.

## C. Estadística

**1. Crecimiento vegetativo.** El crecimiento natural o vegetativo se refiere a la razón del cambio entre lo que entra a un sistema y lo que sale del mismo. En ciencias de la vida puede determinarse como la cantidad de individuos que nacen en un lugar en comparación a los que mueren. En el mundo de las ventas puede representarse por la cantidad de compras en comparación a las ventas. (n.a., Crecimiento Vegetativo, 2000)

El crecimiento vegetativo está dado por medio de la tasa de crecimiento vegetativo que se calcula empleando la siguiente fórmula.

**Figura 4 – Cálculo de la tasa de crecimiento vegetativo**

$$r = \frac{1}{n} * \ln\left(\frac{v_2}{v_1}\right)$$

*r = tasa de crecimiento vegetativo*  
*n = diferencia en tiempo entre los dos períodos dados*  
*v<sub>1</sub> = el valor del primero de los dos períodos*  
*v<sub>2</sub> = el valor del segundo de los dos períodos*

Empleando la tasa anteriormente encontrada es posible realizar predicciones o proyecciones de los valores de una población que se puedan dar mediante el cálculo de proyecciones con crecimiento natural. (n.a., Crecimiento Vegetativo, 2000)

A continuación se describe la fórmula para hacer esta proyección:

**Figura 5 – Cálculo de la proyección de población mediante el uso de la tasa de crecimiento vegetativo**

$$p = v_2 * \ln(1)^{(r*d)}$$

*p = proyección*  
*v<sub>2</sub> = el último valor registrado (empleado para calcular r)*  
*r = tasa de crecimiento vegetativo*  
*d = diferencia de tiempo entre el período a proyectar y v<sub>2</sub>*

## **IV. MARCO METODOLÓGICO**

### **A. Requerimientos**

Para mejorar los procesos de compra mayorista tanto de FPK hacia sus proveedores como para los clientes de FPK fue necesario comprender los procesos actuales, determinar las fallas del mismo y plantear soluciones factibles a la problemática actual. Compenetrarse del tema es el primer paso para desarrollar un proyecto de software exitoso que cumpla las necesidades del cliente y que permita al desarrollador emplear todas las herramientas de su conocimiento para que la solución planteada sea la más eficiente.

Como primer paso fue necesario reunirse con la administración de FPK, asistir a reuniones y estar presente cuando se llevaron a cabo los procesos. Esto se hace para entender qué es lo que se está haciendo y cómo se hace. Es en esta etapa en donde es necesario preguntar y entender todas las variables y todas las acciones que se dan dentro del sistema.

Para el presente proyecto fueron necesarias reuniones administrativas en las que se discutieron los procesos de compra de producto. Se observó como la empresa compra sus productos y como los distribuye a sus clientes. Paralelamente a estas reuniones fue necesario contar con el apoyo de la alta Gerencia para determinar el enfoque que se le quería dar al proyecto. Mediante la observación directa de los procesos y del funcionamiento actual del sistema se recabó suficiente información para determinar la problemática actual, con esta definida se puede plantear una solución y

finalmente con base en este planteamiento se puede iniciar el proceso de diseño y planificación para desarrollar un sistema que cumpla las necesidades de la empresa.

## **B. Herramientas**

Las herramientas que se utilizaron para el desarrollo del sistema en su totalidad fueron los siguientes:

1. Como sistema operativo se utilizó Mac OS X 10.6 Snow Leopard. Todas las herramientas de software son compatibles con este sistema operativo.
2. Como lenguaje de desarrollo se empleó PHP ya que el proyecto es WEB en su totalidad. Para tener un ambiente de PHP se utilizó MAMP, que consiste de las herramientas básicas para desarrollo de aplicaciones WEB. MAMP es un acrónimo de Mac OS X, Apache, MySQL y PHP.
3. Se utilizó Symfony como Framework para desarrollo de aplicaciones de PHP en su versión 1.4.6.
4. Se utilizó la biblioteca de JavaScript, jQuery para ayudar en la mejora de interfaces y hacer más “user friendly” las aplicaciones.
5. Se utilizó Prototype como framework de JavaScript para facilitar el desarrollo de páginas dinámicas y para la integración con AJAX.
6. Como manejador de base de datos se utilizó MySQL. En software se utilizó la herramienta llamada Sequel Pro con el fin de facilitar las consultas a la base de datos ya que esta herramienta permite el manejo de operaciones en la base de datos de una forma gráfica.
7. Como IDE de desarrollo se utilizó Espresso en su versión 1.1.2. Esta herramienta es compatible con PHP, HTML, JavaScript y SQL por lo que fue empleada para el desarrollo de todo el proyecto.

8. Como herramienta de diagramado de base de datos se utilizó SQL Editor en su versión 1.7.6.
9. Como herramienta de diagramado de UML se utilizó OmniGraffle en su versión 5.2.2.

Es necesario mencionar que todas las herramientas de software son de software libre o se utilizaron con su licencia de prueba a excepción del sistema operativo Mac OS X cuya licencia sí fue comprada.

## **V. MÓDULO DE BASE DE DATOS**

### **A. Descripción**

El diseño de la base de datos del proyecto es una de las tareas más importantes. Esto es debido a que una buena arquitectura y diseño de la base de datos garantiza que los datos necesarios para el funcionamiento del proyecto estén disponibles para los usuarios. El funcionamiento es lo más importante, pero también se deben considerar factores como eficiencia, seguridad y consistencia de los datos.

Para el proyecto es necesario diseñar una base de datos que se adapte a un modelo de negocios de venta en línea, tomando en cuenta es necesario contar con un modelo de estadística y reportería lo suficientemente complejo para surtir las necesidades de los administradores de la empresa en la toma de decisiones al momento de hacer las compras.

### **B. Requerimientos**

Debido a la necesidad de constar con una base de datos bastante robusta en manejo de cantidad de datos es requerimiento esencial del proyecto contar con un manejador de base de datos flexible y confiable. Para cumplir esta necesidad y aprovechando las demás herramientas con las que se está desarrollando el proyecto se decidió utilizar el manejador de base de datos open source MySQL. Este manejador tiene las ventajas de ser gratuito, tener documentación completa y disponible al público y se adapta de una forma fácil con PHP y Apache.

Es necesario contar con una base de datos que maneje un volumen de transacciones considerablemente alto, en el módulo de compras debido

a que los clientes son mayoristas y sus pedidos son numerosos. Junto con esto es importante considerar que la finalidad del proyecto es generar la suficiente estadística y reportería para que la administración de FPK pueda tomar sus decisiones de una mejor forma. Esto significa llevar a cabo reportería constante sobre las ventas, el sistema debe tener el diseño adecuado para que el rendimiento tanto del sitio de administración interna como el de compra en línea no se vea afectado por la carga de consultas y transacciones constantes.

## **C. Diseño**

Considerando los requerimientos del proyecto con respecto a la base de datos el modelo relacional es el más indicado para crear la misma. Utilizar este modelo garantiza que no habrá duplicidad de datos y facilita la tarea de hacer asignaciones de permisos de acciones sobre cada tabla, que es otro factor necesario.

Restringir las acciones que cada tipo de usuario puede llevar a cabo sobre la base de datos es muy importante ya en esta se guarda desde información de ventas hasta la administración interna de ambos sitios.

**1. Diagrama.** El diagrama de la base de datos representa de una forma gráfica todas las tablas y relaciones de la misma. El diagrama es el primer paso de diseño ya que más adelante servirá como un mapa que ayuda al desarrollador identificando como se deben comportar las entradas de datos y como debe ser almacenada la información.

backend_modules						
id	INT(11)	U	A	N	P	
name	VARCHAR(100)	N				
description	TEXT	N				
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

permission						
id	INT(11)	U	A	N	P	
name	VARCHAR(100)	N				
description	TEXT	N				
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

module_permission						
id	INT(11)	U	A	N	P	
module_id	INT(11)	I	F			
permission_id	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

role						
id	INT(11)	U	A	N	P	
name	VARCHAR(100)	N				
description	TEXT	N				
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

role_module_permission						
id	INT(11)	U	A	N	P	
role_id	INT(11)	I	F			
module_permission_id	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

access_log						
id	INT(11)	U	A	N	P	
user_id	INT(11)	I	F			
access_time	DATETIME					
logout_time	DATETIME					
ip_address	VARCHAR(15)					
created_at	DATETIME					
updated_at	DATETIME					

backend_user						
id	INT(11)	U	A	N	P	
first_name	VARCHAR(150)	N				
last_name	VARCHAR(150)	N				
email	VARCHAR(150)	N				
password	VARCHAR(100)	N				
status_id	INT(11)	I	F			
role_id	INT(11)	I	F			
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

event_log						
id	INT(11)	U	A	N	P	
action	VARCHAR(100)					
description	TEXT					
module	VARCHAR(100)					
created_by	INT(11)	I	F			
created_at	DATETIME					

error_log						
id	INT(11)	U	A	N	P	
action	VARCHAR(100)					
description	TEXT					
module	VARCHAR(100)					
tags	VARCHAR(255)					
created_by	INT(11)	I	F			
created_at	DATETIME					

product_relation						
id	INT(11)	U	A	N	P	
product_one	INT(11)	I	F			
product_two	INT(11)	I	F			
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

products						
id	INT(11)	U	A	N	P	
name	VARCHAR(255)	N				
code	VARCHAR(100)	U	N			
description	TEXT					
category_id	INT(11)	I	F			
price	DECIMAL(8,2)	N				
image	VARCHAR(255)					
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

clients						
id	INT(11)	U	A	N	P	
first_name	VARCHAR(150)	N				
last_name	VARCHAR(150)	N				
email	VARCHAR(150)	N				
password	VARCHAR(100)	N				
mobile	VARCHAR(20)					
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					
company_id	INT(11)	I	F			

company						
id	INT(11)	U	A	N	P	
name	VARCHAR(255)	N				
phone	VARCHAR(20)					
address	VARCHAR(200)					
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

product_category						
id	INT(11)	U	A	N	P	
name	VARCHAR(100)	N				
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

sales						
id	INT(11)	U	A	N	P	
client_id	INT(11)	I	F			
status_id	INT(11)	I	F			
total	DECIMAL(8,2)	N				
sale_date	DATETIME					
created_at	DATETIME					
updated_at	DATETIME					

sale_history						
id	INT(11)	A	N	P		
product_id	INT(11)	I	F			
sale_total	INT(11)					
shop_total	INT(11)					
month	INT(11)					
year	INT(11)					

tmp_cart						
id	INT(11)	A	N	P		
client_id	INT(11)	I	F			
product_id	INT(11)	I	F			
quantity	INT(11)					
created_at	DATETIME					
updated_at	DATETIME					

sale_detail						
id	INT(11)	U	A	N	P	
sale_id	INT(11)	I	F			
item_id	INT(11)	I	F			
quantity	INT(11)					

order_notification						
id	INT(11)	A	N	P		
first_name	VARCHAR(150)					
last_name	VARCHAR(150)					
email	VARCHAR(150)					
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

sale_status						
id	INT(11)	U	A	N	P	
name	VARCHAR(100)	N				
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

user_status						
id	INT(11)	U	A	N	P	
name	VARCHAR(50)	N				
created_by	INT(11)	I	F			
updated_by	INT(11)	I	F			
created_at	DATETIME					
updated_at	DATETIME					

Basándose en el diagrama anterior la base de datos del proyecto consta de 21 tablas y las relaciones que existen entre sí. En el diccionario de datos se hace una descripción de cada tabla y todos los campos de la misma.

**2. Diccionario de datos.** El diccionario de datos es la descripción de la estructura y campos de cada una de las tablas de la base de datos. Para facilitar la escritura de las tablas del diccionario se ha utilizado la siguiente abreviatura. PK = llave primaria por sus siglas en inglés y FK = llave foránea por sus siglas en inglés.

**a. access\_log.** Almacena la información de acceso al sistema de administración interna del sistema.

**Tabla 1 – Campos de la tabla access\_log**

Campo	Descripción	Tipo	PK	FK
id	Identificador único.	INT(11)		
user_id	Identificador del usuario que está accediendo al sistema	INT(11)		
access_time	Fecha y hora de acceso	DATETIME		
logout_time	Fecha y hora de cierre de sesión	DATETIME		
ip_address	Dirección IP del usuario que está accediendo.	VARCHAR(15)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

**b. backend\_modules.** Almacena la lista de módulos del proyecto para utilizarlos en el sistema de seguridad y restricción de acceso.

**Tabla 2 – Campos de la tabla backend\_modules**

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
id	Identificador único.	INT(11)		
name	Nombre que recibe el módulo	VARCHAR(100)		
description	Descripción del módulo	TEXT		
created_by	Identificador del usuario que creo el registro	INT(11)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro			
updated_at	Fecha y hora de actualización del registro			

**c. backend\_user.** Almacena la lista de usuarios que pueden acceder al sistema de administración.

**Tabla 3 – Campos de la tabla backend\_user**

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
id	Identificador único.	INT(11)		
first_name	Primer nombre del usuario.	VARCHAR(150)		
last_name	Apellido del usuario.	VARCHAR(150)		
email	Dirección de correo electrónico. Será utilizado como nombre de usuario.	VARCHAR(150)		
password	Contraseña que le da acceso al sistema.	VARCHAR(150)		
status_id	Identificador del estado del usuario.	INT(11)		
role_id	Identificador del rol del usuario.	INT(11)		
created_by	Identificador del usuario que creo el registro	INT(11)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

**d. clients.** Almacena la información de los clientes que pueden acceder al sistema de venta en línea.

**Tabla 4 – Campos de la tabla clients**

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
id	Identificador único.	INT(11)		
first_name	Primer nombre del cliente.	VARCHAR(150)		
last_name	Apellido del cliente.	VARCHAR(150)		
email	Dirección de correo electrónico. Será utilizado como nombre de usuario.	VARCHAR(150)		
password	Contraseña que le da acceso al sistema.	VARCHAR(150)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		
company_id	Identificador de la compañía que respalda al cliente.	INT(11)		
mobile	Teléfono celular del cliente	VARCHAR(20)		

**e. company.** Almacena las diferentes empresas que compran con FPK.

Es utilizada para definirle a un cliente una empresa que lo respalda.

**Tabla 5 – Campos de la tabla company**

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
id	Identificador único.	INT(11)		
name	Nombre de la compañía.	VARHAR(20)		
phone	Teléfono de la empresa	VARCHAR(20)		
address	Dirección de la empresa	VARCHAR(20)		
created_by	Identificador del usuario que creó el registro	INT(11)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

**f. error\_log.** Almacena la lista de errores que se dan tanto en base de datos como en PHP del sistema.

**Tabla 6 – Campos de la tabla error\_log**

Campo	Descripción	Tipo	PK	FK
id	Identificador único.	INT(11)		
action	Acción que se estaba llevando a cabo cuando se dio el error.	VARCHAR(100)		
description	Breve descripción del error causado	TEXT		
module	Módulo en donde se dio el error	VARCHAR(100)		
tags	Información del error. Texto que da la excepción de PHP o el error de MySql	VARCHAR(255)		
created_by	Identificador del usuario que creó el registro	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		

**g. event\_log.** Almacena la lista de eventos que se llevan a cabo en el sistema de administración.

**Tabla 7 – Campos de la tabla event\_log**

Campo	Descripción	Tipo	PK	FK
id	Identificador único.	INT(11)		
action	Acción que se estaba llevando a cabo cuando se dio el error.	VARCHAR(100)		
description	Breve descripción del evento causado	TEXT		
module	Módulo en donde se dio el evento	VARCHAR(100)		
created_by	Identificador del usuario que creó el registro	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		

**h. module\_permission.** Almacena los datos de permiso y módulo para determinar que permisos pertenecen a qué módulo. (tabla de cruce).

**Tabla 8 – Campos de la tabla module\_permission**

Campo	Descripción	Tipo	PK	FK
id	Identificador único.	INT(11)		
module_id	Identificador del módulo	INT(11)		
permission_id	Identificador del permiso	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

*i. order\_notification.* Almacena la información de notificaciones administrativas, es decir quién va a recibir notificaciones de los usuarios administrativos.

**Tabla 9 – Campos de la tabla order\_notification**

Campo	Descripción	Tipo	PK	FK
id	Identificador único.	INT(11)		
first_name	Primer nombre del destinatario.	VARCHAR(150)		
last_name	Apellido del destinatario.	VARCHAR(150)		
email	Dirección de correo electrónico.	VARCHAR(150)		
created_by	Identificador del usuario que creó el registro	INT(11)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

*j. permission.* Almacena la lista de permisos disponibles en el sistema.

**Tabla 10 – Campos de la tabla permission**

Campo	Descripción	Tipo	PK	FK
id	Identificador único.	INT(11)		
name	Nombre que recibe el permiso	VARCHAR(100)		
description	Descripción del permiso	TEXT		
created_by	Identificador del usuario que creó el registro	INT(11)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro			
updated_at	Fecha y hora de actualización del registro			

**k. product\_category.** Almacena todas las diferentes categorías de productos disponibles para venta.

**Tabla 11 – Campos de la tabla product\_category**

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
id	Identificador único.	INT(11)		
name	Nombre que recibe la categoría de producto.	VARCHAR(100)		
created_by	Identificador del usuario que creó el registro	INT(11)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro			
updated_at	Fecha y hora de actualización del registro			

**l. product\_relation.** Almacena la relación de productos para los sugeridos de compras.

**Tabla 12 – Campos de la tabla product\_relation**

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
id	Identificador único.	INT(11)		
product_one	Identificador del primer producto	INT(11)		
product_two	Identificador del segundo producto	INT(11)		
created_by	Identificador del usuario que creó el registro	INT(11)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

**m. products.** Almacena los diferentes productos disponibles para venta.

Tabla 13 – Campos de la tabla products

Campo	Descripción	Tipo	PK	FK
id	Identificador único.	INT(11)		
name	Nombre del producto	VARHAR(255)		
code	Código del producto	VARHAR(100)		
description	Descripción del producto	TEXT		
category_id	Identificador de la categoría a la cual pertenece el producto	INT(11)		
created_by	Identificador del usuario que creó el registro	INT(11)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

**n. role.** Almacena los diferentes roles que se le pueden asignar a cada usuario.

Tabla 14 – Campos de la tabla role

Campo	Descripción	Tipo	PK	FK
id	Identificador único.	INT(11)		
name	Nombre que recibe el rol	VARCHAR(100)		
description	Descripción del rol	TEXT		
created_by	Identificador del usuario que creó el registro	INT(11)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro			
updated_at	Fecha y hora de actualización del registro			

**o. role\_module\_permission.** Almacena los datos de rol y permiso para determinar que permisos posee cada rol. (tabla de cruce).

**Tabla 15 – Campos de la tabla role\_module\_permission**

Campo	Descripción	Tipo	PK	FK
id	Identificador único.	INT(11)		
role_id	Identificador del rol	INT(11)		
module_permission_id	Identificador de la tabla module_permission.	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

**p. sale\_detail.** Almacena el detalle de cada orden realizada.

**Tabla 16 – Campos de la tabla sale\_detail**

Campo	Descripción	Tipo	PK	FK
id	Identificador único.	INT(11)		
sale_id	Identificador de la venta	INT(11)		
item_id	Identificador del producto comprado	INT(11)		
quantity	Cantidad de producto a ser comprado.	INT(11)		

**q. sale\_history.** Almacena los históricos de compras de productos para utilizarlos como referencia en el módulo estadístico.

**Tabla 17 – Campos de la tabla sale\_history**

Campo	Descripción	Tipo	PK	FK
id	Identificador único.	INT(11)		
product_id	Identificador de producto	INT(11)		
sale_total	Total de ventas	INT(11)		
shop_total	Total de compras (por parte de FPK)	INT(11)		
month	Mes en que se llevó a cabo la transacción	INT(11)		
year	Año en que se llevó a cabo la transacción	INT(11)		

**r. sale\_status.** Almacena los diferentes estados que puede tomar cada orden.

**Tabla 18 – Campos de la tabla sale\_status**

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
id	Identificador único.	INT(11)		
name	Nombre del estado de orden	VARCHAR(100)		
created_by	Identificador del usuario que creó el registro	INT(11)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

**s. sales.** Almacena todas las ventas que se han realizado.

**Tabla 19 – Campos de la tabla sales**

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
id	Identificador único.	INT(11)		
client_id	Identificado del cliente que hace la compra.	INT(11)		
status_by	Identificador del estado de la orden.	INT(11)		
total	Total de la compra	DECIMAL(8,2)		
sale_date	Fecha y hora en la que se realizó la obra	DATETIME		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

**t. tmp\_cart.** Almacena la carretilla de compra de cada usuario.

**Tabla 20 – Campos de la tabla tmp\_cart**

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
id	Identificador único.	INT(11)		
client_id	Identificado del cliente que hace la compra.	INT(11)		
product_id	Identificador del producto en carretilla.	INT(11)		
quantity	Cantidad de producto en carretilla	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

**u. user\_status.** Almacena los diferentes estados que se le pueden asignar a cada usuario.

**Tabla 21 – Campos de la tabla user\_status**

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
id	Identificador único.	INT(11)		
name	Nombre del estado de usuario	VARCHAR(100)		
created_by	Identificador del usuario que creó el registro	INT(11)		
updated_by	Identificador del usuario que editó por último el registro	INT(11)		
created_at	Fecha y hora de creación del registro	DATETIME		
updated_at	Fecha y hora de actualización del registro	DATETIME		

**3. Permiso de roles.** Debido a que existen diferentes roles que un usuario puede tomar en el sistema es necesario detallar los permisos que cada rol tendrá sobre las diferentes tablas de la base de datos. La siguiente tabla detalla las operaciones que cada rol principal puede ejecutar en cada una de las tablas.

Para facilitar la escritura de la tabla se utilizó las siguientes abreviaturas: C = ingresar datos por sus siglas en inglés (create), R = lectura por sus siglas en inglés (read), U = actualizar por sus siglas en inglés (update) y D = borrar por sus siglas en inglés. Los roles, definidos en el capítulo anterior fueron abreviados de la siguiente manera: D = desarrollador y A = administrador.







## VI. MÓDULO DE ADMINISTRACIÓN INTERNA

### A. Descripción

La administración interna se refiere al manejo completo del sitio que el cliente utiliza para realizar sus compras. Las tareas que están disponibles en el módulo son las siguientes:

1. Crear roles y asignarles permisos.
2. Crear usuarios y asignarles un rol.
3. Crear fichas de clientes.
4. Crear fichas de empresas.
5. Crear categorías de productos.
6. Crear catálogos de productos.
7. Relacionar productos entre sí.
8. Crear estados de órdenes.
9. Crear usuarios para notificaciones de compras.
10. Acceder a reportería sobre el comportamiento de productos.
11. Acceder a reportería sobre el total de ventas.
12. Acceder a los productos más vendidos y lo menos vendidos.
13. Acceder al módulo de estadística para obtener sugeridos de compra.

Es desde este módulo por el que la administración de FPK podrá poner a disponibilidad de sus clientes todos sus productos, recomendar productos con base en lo que se está comprando y le permitirá ver reportería simple sobre sus compras realizadas.

Por medio de este sitio el administrador debe alimentar la base de datos, tanto de clientes y las empresas que los respaldan, como de sus productos. Debido a que para lograr esto se debe tener acceso directo a la base de datos es necesario implementar medidas de seguridad para restringir el acceso de intrusos o personal ajeno a la empresa.

## **B. Requerimientos**

Como requerimiento primordial está la necesidad de aislar el sistema para que solamente los usuarios debidamente autenticados puedan acceder. Esta necesidad surge debido a que por medio de este módulo los administradores tienen acceso a la mayoría de operaciones permitidas sobre las bases de datos y en manos erróneas se pueden causar serios daños a la integridad de los datos. La privacidad es otro factor importante ya que es posible consultar todo el historial de ventas lo que es un dato bastante privado y reservado para la alta Gerencia.

Junto con la seguridad, el diseño de una interfaz clara, entendible y amigable es un requerimiento importante. Es necesario tomar en cuenta la cantidad de usuarios administrativos que el sistema puede llegar a tener. Es importante que en su mayoría estos usuarios puedan navegar por el sitio y realizar sus tareas cotidianas de una forma intuitiva y sin la mayor explicación posible. La meta es crear un sistema cuya curva de aprendizaje sea bastante baja y con una simple inducción de la herramienta los usuarios puedan aprender a usarla.

El tercer requerimiento importante es desarrollar un sistema de administración que valla un paso más lejos de simplemente crear catálogos y administrar el sitio. Es importante también obtener retroalimentación del comportamiento de los clientes. Es por

eso que se hizo necesario crear los módulos de reportería y estadística. Para que en cualquier momento y bajo cualquier circunstancia se pueda consultar el desempeño de los productos en el mercado, el movimiento de los productos en tienda y la popularidad de los mismos.

## C. Diseño

**1. Roles.** La solución propuesta para garantizar la seguridad y el acceso privado a datos se desarrollo un sistema de roles. Un rol es simplemente el papel que cada usuario va a jugar dentro del sistema. Esto facilita la tarea de asignar permisos a cada usuario individualmente ya que se pueden crear los roles necesarios, asignarle permiso a estos y luego signarle un rol ya sea a uno o muchos usuarios.

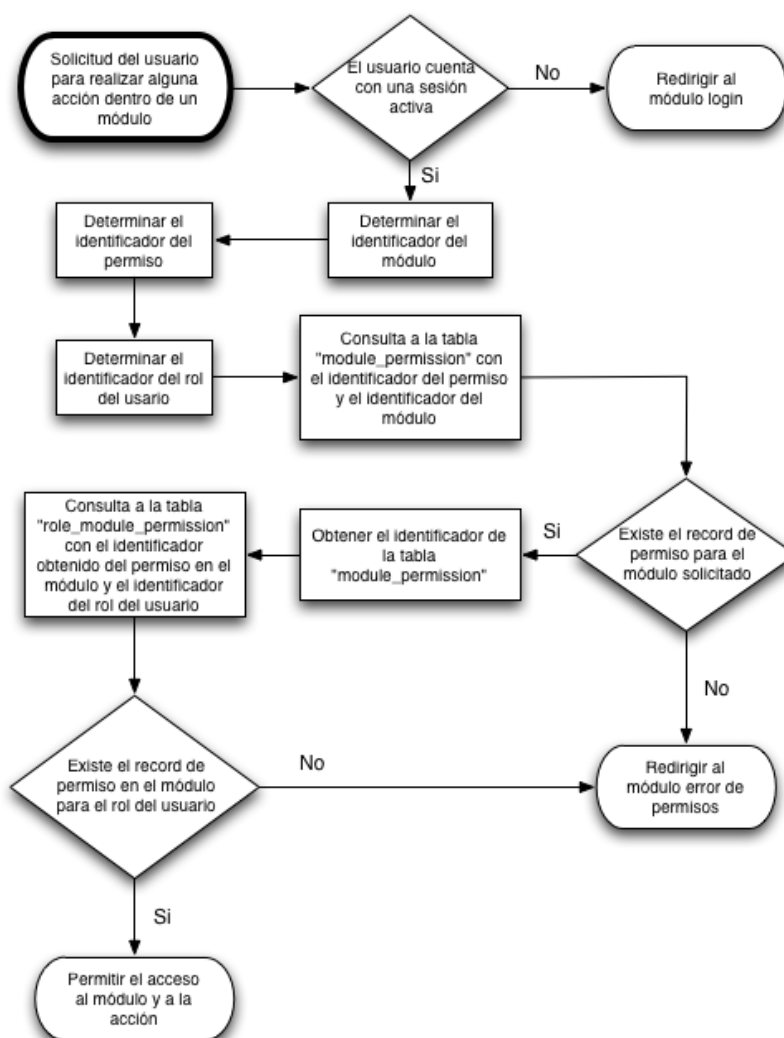
Para este sistema de roles es necesario el uso de varias tablas de la base de datos. Su funcionamiento no es complejo pero requiere de una buena definición de que tipos de usuarios se desea incluir. Con una definición de usuarios creada es necesario establecer una lista de permisos que un rol puede tener en el sistema. Estos permisos son almacenados en una tabla en la base de datos. La data recabada al momento permite ingresar registros a la tabla de cruce entre módulos y permisos. Es en esta tabla que se asocia un permiso a cierto módulo. Es necesario hacer está asociación ya que no todos los permisos estarán disponibles para todos los módulos.

La creación del rol no requiere de mucho trabajo ya que simplemente es generar un registro en una tabla de la base de datos. Con el rol ya creado es necesario otorgarle permisos de realizar diferentes acciones en cada módulo. Para hacer esto se hace un cruce entre el rol creado y la tabla de cruce definida por los módulos y permisos. Al hacer este cruce obtenemos una tabla en la que se guarda el identificador del rol y el

identificador del cruce entre un módulo y un permiso. Ahora la tarea de autenticación se hizo relativamente simple ya que solo se debe consultar la base de datos para determinar si el rol del usuario posee los permisos de las acciones que está solicitando realizar dentro del módulo en el que se encuentra.

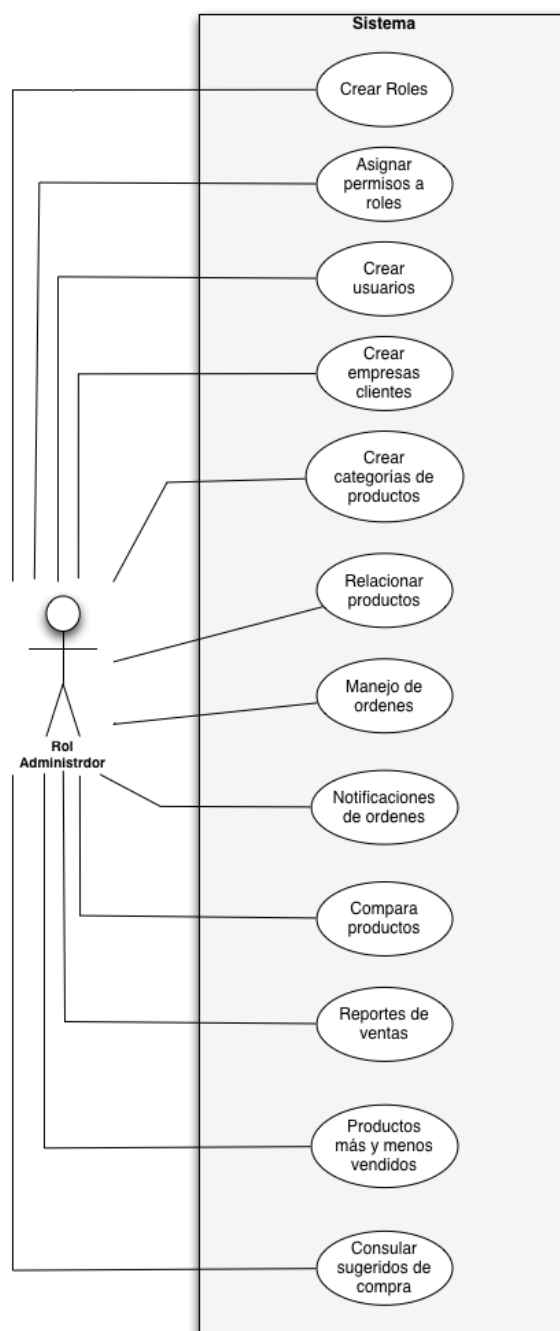
El siguiente diagrama de flujo muestra el comportamiento del sistema y los pasos que este realiza para determinar si un usuario tiene permisos de realizar las acciones solicitadas.

**Figura 6 – Diagrama de flujo del proceso de autenticación de usuario**



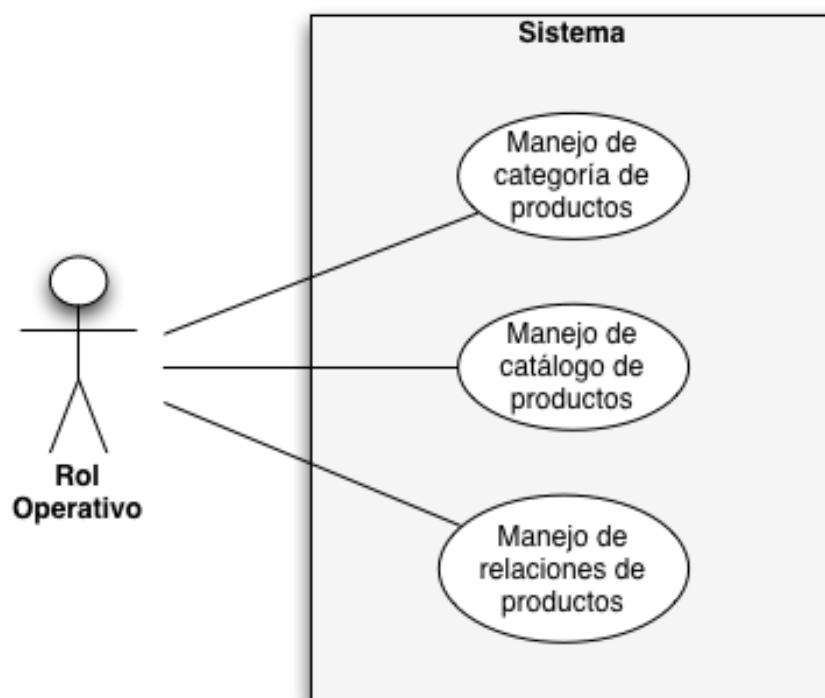
**2. UML.** Basándose en el diseño de la base de datos se procedió a diseñar las acciones del sistema. Fue necesario definir los procesos que se deben llevar a cabo dentro del módulo. Para realizar esta definición se emplearon diagramas de casos de uso de las acciones posibles de cada uno de los roles.

**Figura 7 – Diagrama de casos de uso del rol Administrador**

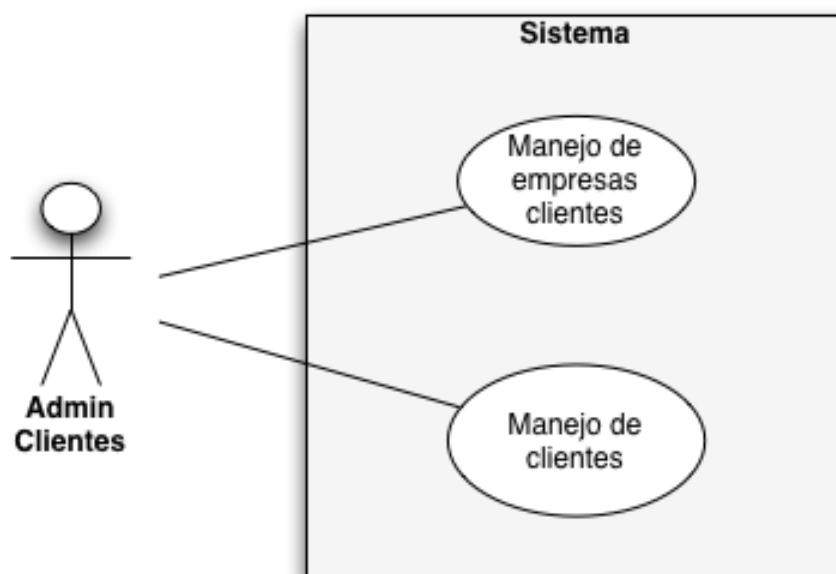


El rol de administrador es el más completo ya que puede llevar a cabo todas las operaciones del módulo de administración.

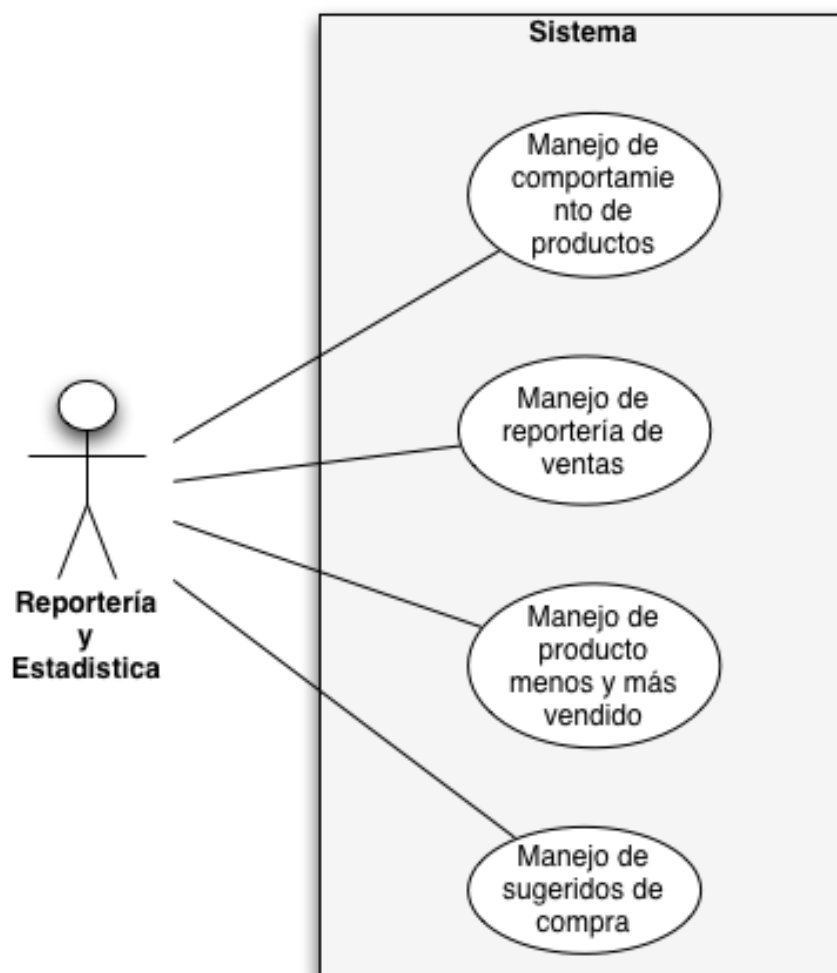
**Figura 8– Diagrama de casos de uso del rol operativo**



El rol operativo fue creado con el propósito de tener un grupo de usuarios dedicados al manejo de los catálogos de productos y mantener el inventario actualizado. A pesar de lo limitado que son sus permisos el rol operativo juega un papel crucial ya que es éste el que alimenta los productos que los clientes pueden comprar en el módulo de tienda.

**Figura 9 – Diagrama de casos de uso del rol cliente**

El rol de manejo de clientes es el más limitado. Sus únicas funciones son enteramente del manejo de clientes. La importancia de este rol es la habilidad de poder crear usuarios que accedan al módulo de tienda ya que es necesario determinar que clientes cumplen el perfil definido para interactuar con el sistema.

**Figura 10 – Diagrama de casos de uso del rol reports y estadística**

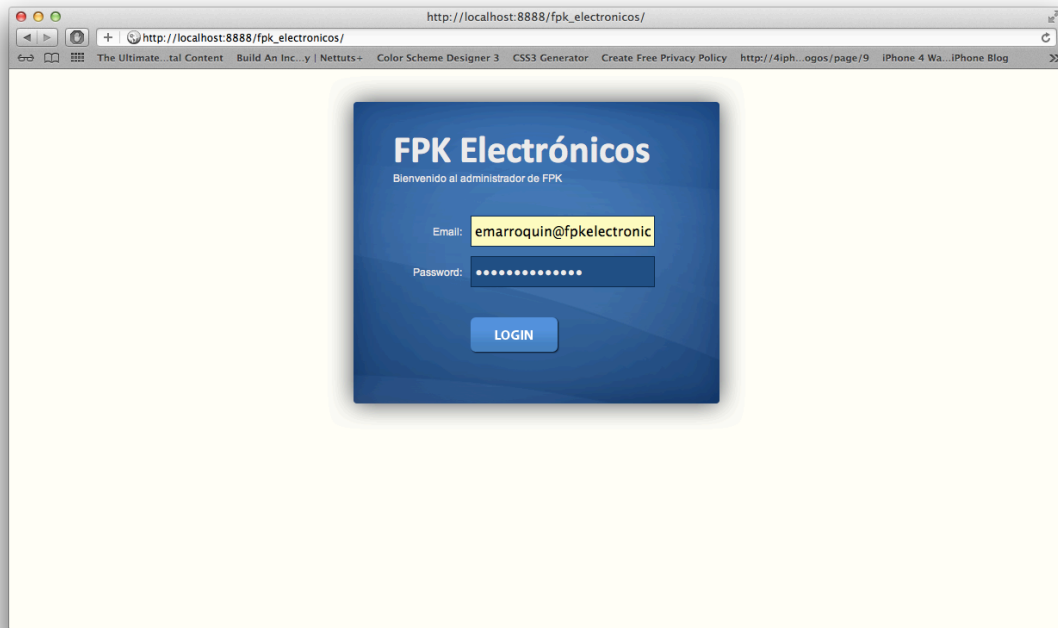
El rol de reportería y estadística es de los roles más importantes ya que los usuarios que juegan este rol dentro del sistema tienen a su disponibilidad toda la información en forma de reportes y todas las proyecciones estadísticas para poder tomar las decisiones para realizar compras a los proveedores, determinar qué productos se mueven más y como se han movido las ventas.

El rol de desarrollador no fue definido ya que éste puede desarrollar todas las acciones del módulo de administrador.

**3. Interfaces.** Las interfaces son una parte importante del desarrollo ya que son la línea directa con el usuario final y el sistema. Una mala interfaz genera problemas de usabilidad que perjudican el rendimiento de los usuarios y puede llegar a causar mal funcionamiento de la aplicación.

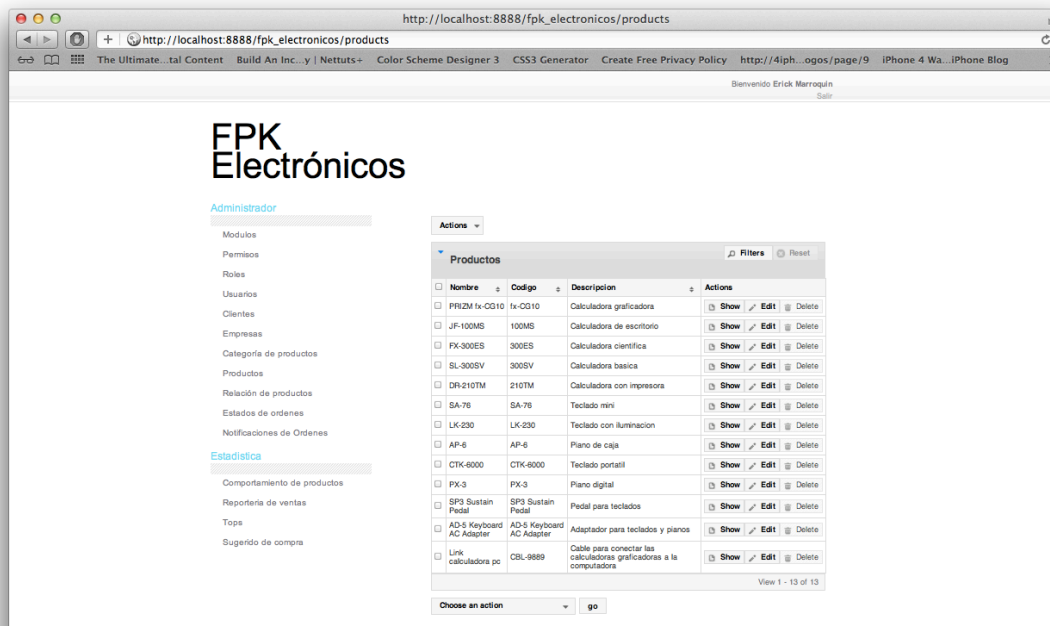
Junto con un diseño amigable es necesario implementar validaciones del sistema para que este sea funcional, robusto y no falle. Uno de los retos más grandes es validar y corregir al usuario en una forma entendible, usando mensajes claros y explícitos, no ambiguos para no confundir a los usuarios. Para lograr esto fue necesario integrar la seguridad y validaciones en el diseño para que la integración con el mismo sea lo más plano posible y que el usuario no sienta que las validaciones estorban y son molestas. La meta es que el usuario considere las validaciones como ayudas y no como obstáculos.

A continuación se muestran las diferentes interfaces de este módulo.

**Figura 11 – Interfaz de login backend**

La autenticación de los usuarios administrativos es el primer paso para acceder al sistema. Si los usuarios no son válidos el sistema rechaza al mismo y no le permite el acceso. Al momento de hacer la autenticación y que esta sea de un usuario válido se procede a cargar los permisos del mismo.

Figura 12 – Interfaz de catálogo de productos

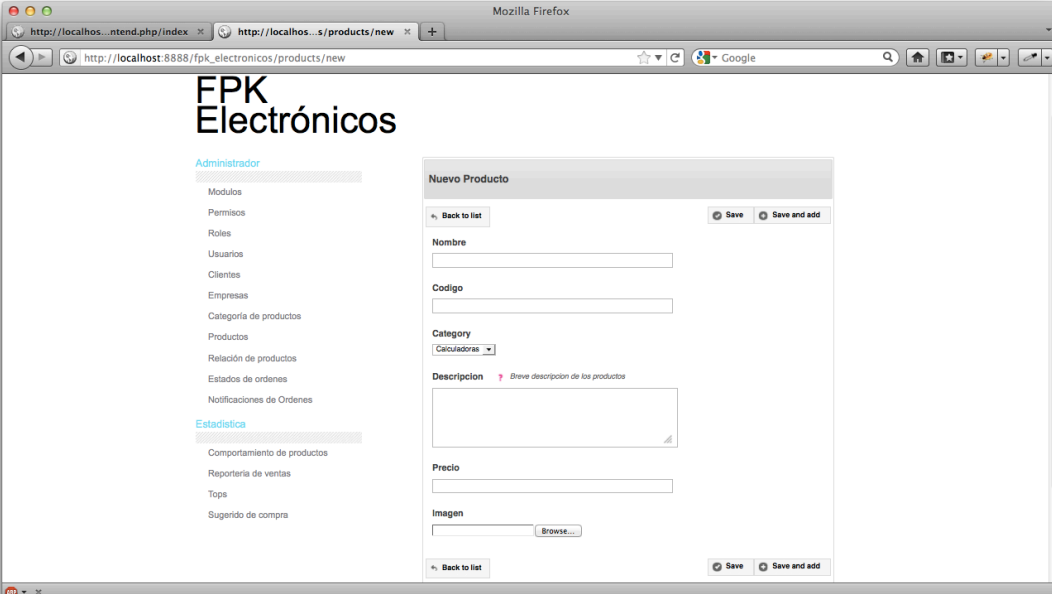


The screenshot displays the FPK Electrónicos product catalog interface. The page title is "FPK Electrónicos" and the user is logged in as "Bienvenido Erick Marroquín". The interface includes a navigation menu on the left with sections like "Administrador" and "Estadística". The main content area shows a table of products with the following data:

Nombre	Código	Descripción	Actions
PRIZM fx-CG10	fx-CG10	Calculadora graficadora	Show Edit Delete
JF-100MS	100MS	Calculadora de escritorio	Show Edit Delete
FX-300ES	300ES	Calculadora científica	Show Edit Delete
SL-300SV	300SV	Calculadora básica	Show Edit Delete
DR-210TM	210TM	Calculadora con impresora	Show Edit Delete
SA-76	SA-76	Teclado mini	Show Edit Delete
LK-230	LK-230	Teclado con iluminación	Show Edit Delete
AP-6	AP-6	Piano de caja	Show Edit Delete
CTK-6000	CTK-6000	Teclado portátil	Show Edit Delete
PX-3	PX-3	Piano digital	Show Edit Delete
SP3 Sustain Pedal	SP3 Sustain Pedal	Pedal para teclados	Show Edit Delete
AD-5 Keyboard AC Adapter	AD-5 Keyboard AC Adapter	Adaptador para teclados y pianos	Show Edit Delete
Link calculadora pc	CSL-9889	Cable para conectar las calculadoras graficadoras a la computadora	Show Edit Delete

The interface also includes a "Choose an action" dropdown and a "go" button at the bottom of the table.

El catálogo de productos lista todos los productos que FPK Electrónicos tiene disponible para sus clientes. En este catálogo es posible insertar campos, eliminar campos, editar campos y ver campos.

**Figura 13 – Forma para ingreso de datos**

The image shows a screenshot of a web browser window displaying a form for adding a new product to an inventory system. The browser is Mozilla Firefox, and the address bar shows the URL `http://localhost:8888/fpk_electronicos/products/new`. The page title is "FPK Electrónicos".

The form is titled "Nuevo Producto" and contains the following fields and controls:

- Nombre:** A text input field.
- Código:** A text input field.
- Category:** A dropdown menu with "Calculadoras" selected.
- Descripción:** A text area with a placeholder text "Breve descripción de los productos".
- Precio:** A text input field.
- Imagen:** A text input field with a "Browse..." button next to it.

At the top of the form, there are two buttons: "Back to list" and "Save". At the bottom of the form, there are two buttons: "Save" and "Save and add".

On the left side of the page, there is a navigation menu with the following items:

- Administrador
- Modulos
- Permisos
- Roles
- Usuarios
- Clientes
- Empresas
- Categoría de productos
- Productos
- Relación de productos
- Estados de ordenes
- Notificaciones de Ordenes
- Estadística
- Comportamiento de productos
- Reporteria de ventas
- Tops
- Sugerido de compra

Éste es un ejemplo de las formas para ingresar contenido a la página. En este caso es la forma para crear un producto nuevo al inventario.

## VII. MÓDULO DE VENTA EN LÍNEA

### A. Descripción

El módulo de venta en línea es el “frontend” del proyecto. Esta es la cara a los clientes. En este módulo los clientes mayoristas pueden realizar sus pedidos de mercadería a FPK, teniendo el catálogo de productos a su disposición. En su mayoría las configuraciones del módulo de administración interna se ven reflejadas en este sitio.

En esencia el módulo de venta en línea es una carretilla de pedidos en la que los usuarios pueden ver y realizar pedidos sobre el inventario pero con la funcionalidad extra de poder ver reportería básica de sus pedidos previos, así como el de los pedidos actuales.

Las acciones disponibles en este módulo deben ser las siguientes:

- a. Ver el catálogo de productos.
- b. Realizar pedidos de productos.
- c. Ver estado de órdenes.
- d. Ver detalles de órdenes pendientes.
- e. Ver detalle de órdenes históricas.
- f. Ver reportería de compras.

Es de este módulo que se producen todas las entradas de data, que será recopilada y presentada como estadística al administrador.

## **B. Requerimientos**

Como requerimiento primordial está la posibilidad del cliente a realizar sus pedidos en un sitio seguro, amigable y fácil de utilizar. Es de suma importancia constar con este portal ya que es aquí en donde se alimentan las compras que posteriormente son analizadas por la administración. Como requerimiento extra se pondrá a disposición de los clientes un reporte de todos los pedidos que ha realizado junto con un detalle de los mismos.

Un diseño amigable y fácil de usar hace que los clientes se sientan cómodos utilizando el sistema y esto los lleva a usarlo y adaptarse al mismo, en vez de que FPK tenga que adaptarse a como cada cliente realiza sus compras. Es importante tomar en cuenta que tanto los clientes como FPK se benefician del uso del sistema ya que los clientes pueden realizar los pedidos de los productos que estos necesiten con la seguridad de que los van a recibir no importando el volumen de las mismas. FPK se ve beneficiado ya que puede venderles más a sus clientes y genera una base de datos que posteriormente puede traducirse en datos estadísticos para mejor toma de decisiones.

Un requerimiento que ayudará a que el sistema sea más apetecible para los clientes es la reportería de productos. La reportería es una subsección de este módulo que le brinda una visión clara y detallada de todas las compras que este ha realizado y un detalle de cada una de las mismas. Teniendo este control los clientes pueden determinar cómo han invertido su dinero en compra de productos y como han variado la compra de los productos con respecto al tiempo. Otro servicio de reportería es la capacidad de ver los pedidos pendientes y el estado de los mismos. Los clientes podrán ver el estado de sus pedidos y la descripción completa de los mismos.

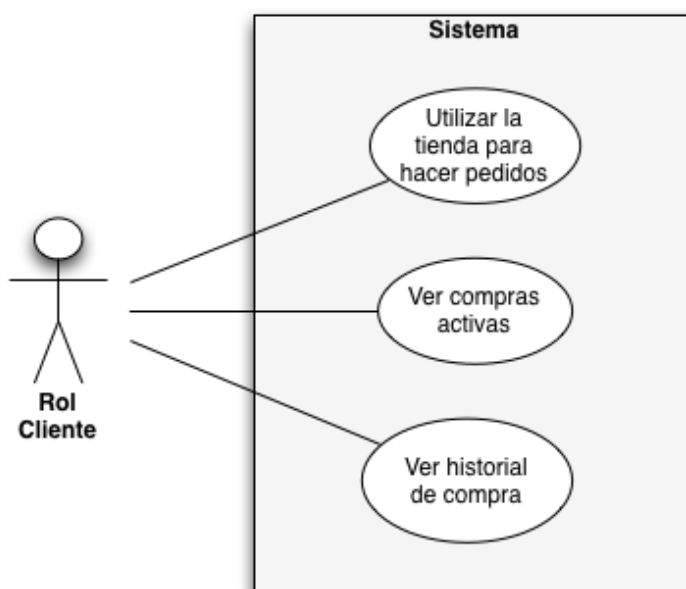
Las notificaciones vía email son un requerimiento básico ya que ayudan al cliente a mantener un control y saber al momento exacto en el que sus pedidos han cambiado de estado. Este servicio brinda una forma más eficiente de acercar al cliente al proceso de compra ya que este se siente incluido y participa en todo momento del proceso de la compra.

## **C. Diseño**

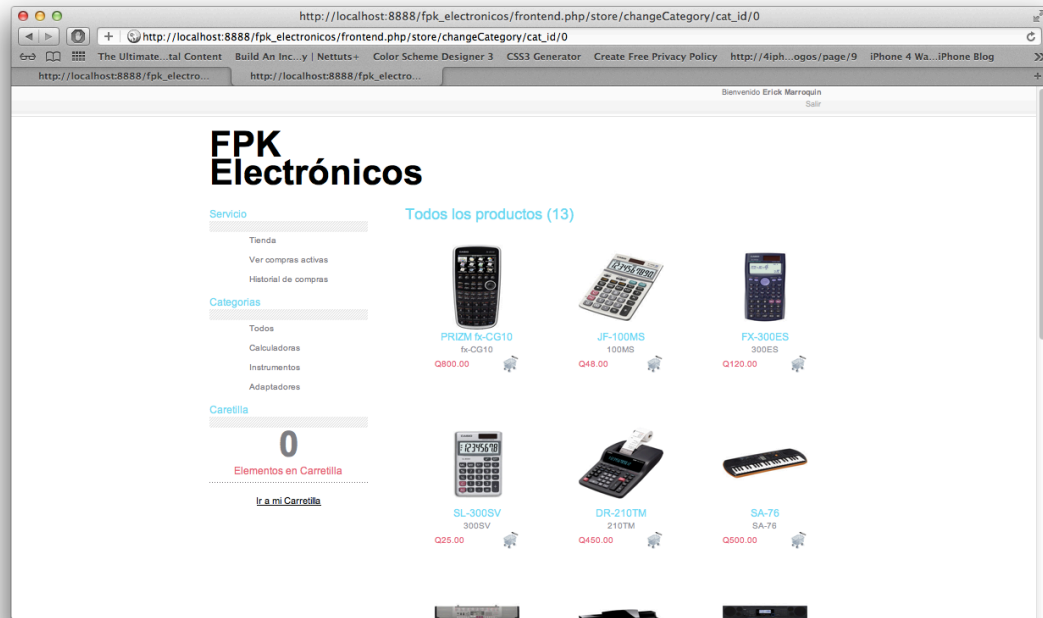
**1. Roles.** Para la venta en línea fue necesario implementar el rol de cliente definido con anterioridad. A diferencia de los roles administrativos el rol de cliente ya tiene especificadas las acciones que podrá realizar y debido a que el acceso de los clientes es en otro portal se garantiza que ningún cliente pueda acceder al sistema de administración.

**2. UML.** Basándose en el diseño de la base de datos se procedió a diseñar las acciones del sistema. Fue necesario definir los procesos que se deben llevar a cabo dentro del módulo. Para realizar esta definición se empleó un diagrama de casos de uso en el que se muestra las acciones posibles que los usuarios de este sistema pueden realizar.

**Figura 14 – Diagrama de casos de uso del rol cliente en el frontend**

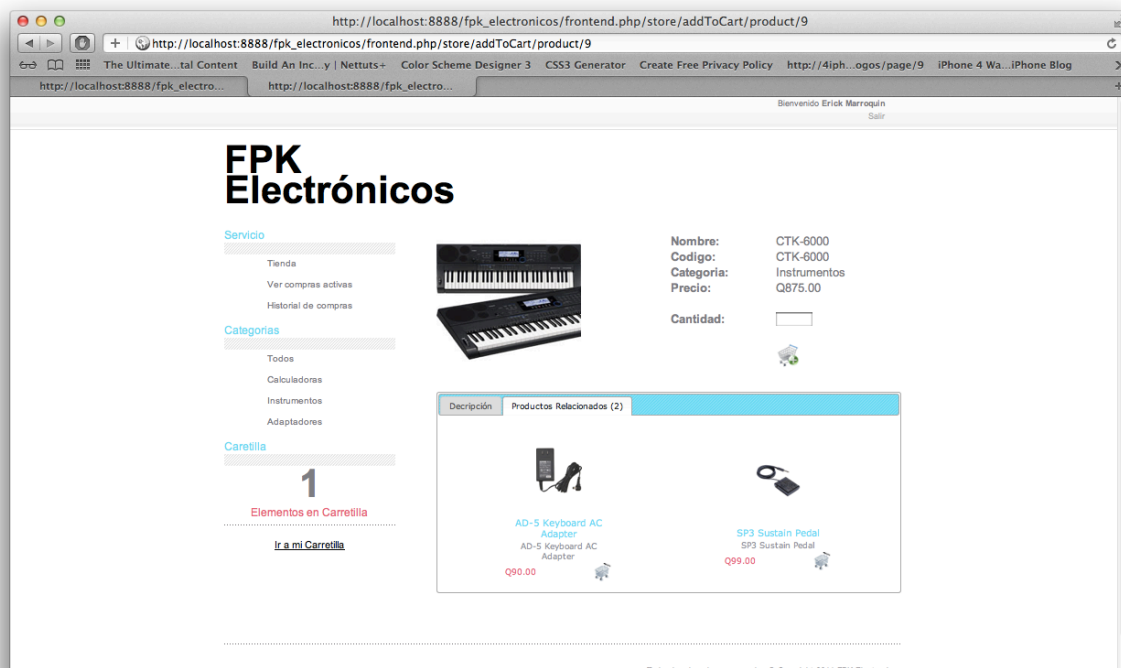


**3. Interfaces.** El diseño de interfaces amigables fue una prioridad al momento de diseñar el módulo de venta en línea. Los diagramas a continuación muestran como fue implementado el diseño del presente módulo.

**Figura 15 – Vista de la carretilla de compra**

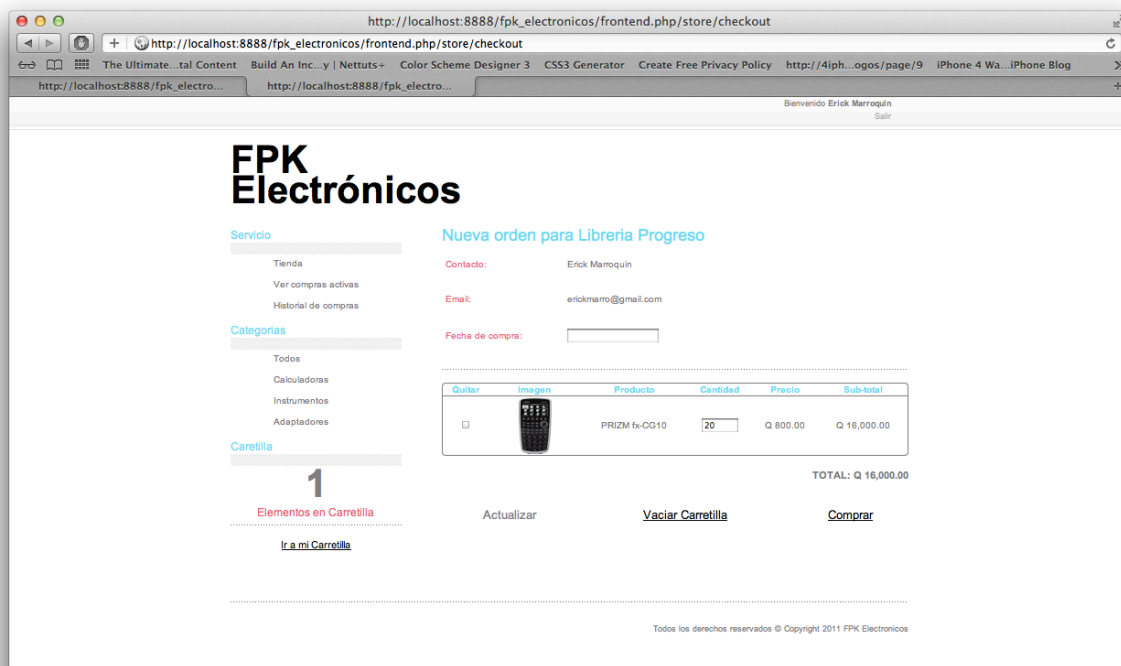
El inventario de los productos es desplegado en la tienda virtual con la cual los clientes pueden ver las fotos, leer las descripciones, ver los precios y hacer sus pedidos.

Figura 16 – Vista de producto individual de la carretilla




Los clientes pueden ver la información completa de cada uno de los productos en el inventario. También está disponible una descripción completa del producto junto con una sección de los productos relacionados.

Figura 17 – Vista de la carreta de compra de cada usuario



The screenshot shows a web browser window displaying the checkout page for FPK Electrónicos. The page title is "Nueva orden para Librería Progreso". The user is logged in as "Erick Marroquin". The page features a navigation menu on the left with sections for "Servicio", "Categorías", and "Carretilla". The "Carretilla" section shows "1 Elementos en Carretilla" and a link to "Ir a mi Carretilla". The main content area displays the shopping cart with one item: "PRIZM fx-CG10" with a quantity of 20, a unit price of Q 800.00, and a subtotal of Q 16,000.00. The total amount is also Q 16,000.00. Below the cart, there are buttons for "Actualizar", "Vaciar Carretilla", and "Comprar".

Quitar	Imagen	Producto	Cantidad	Precio	Sub-total
<input type="checkbox"/>		PRIZM fx-CG10	20	Q 800.00	Q 16,000.00

TOTAL: Q 16,000.00

Actualizar      Vaciar Carretilla      Comprar

Todos los derechos reservados © Copyright 2011 FPK Electronicos

Los clientes pueden visualizar todos los productos en su carretilla de compras junto con el detalle de la venta total. En la carretilla se muestra el ítem que se ha agregado, el precio unitario, la cantidad, un subtotal por ítem y el total de la compra.

**Figura 18 – Vista del reporte de compras disponible para los clientes**

http://localhost:8888/fpk\_electronicos/frontend.php/store/viewActiveOrders

FPK Electrónicos

Servicio


- Tienda
- Ver compras activas
- Historial de compras

Ordenes en proceso

Compra programada para: **2011-12-20**

Numero de orden: 201100001  
Fecha de compra: 2011-10-18  
Estado de orden: En Proceso

Total: Q 16,000.00 [Ver Detalle](#)

Producto	Precio u.	Cantidad	Sub-Total
 PRIZM tv-CG10	Q 800.00	20	Q 16000

Todos los derechos reservados © Copyright 2011 FPK Electronicos

El detalle de las órdenes muestra la fecha de la compra, el número de orden, el estado de la misma y el total. Los usuarios también pueden ver el detalle completo de sus pedidos.

## VIII. MÓDULO DE REPORTES Y ESTADÍSTICA

### A. Descripción

El módulo de reportes y estadística es la colección de herramientas puestas a la disposición de los administradores de la empresa para que estos puedan ayudarse a tomar las mejores decisiones sobre la cantidad de productos que se deben comprar basándose en el estado actual de las ventas y proyecciones hechas tomando en cuenta ventas pasadas. Éste es el módulo más importante del proyecto ya que en este se pueden ver los resultados de las acciones hechas en los demás módulos.

Las tareas disponibles en este módulo son las siguientes:

1. Visualizar de una forma gráfica y detallada el comportamiento de cada producto del catálogo.
2. Visualizar de una forma ordenada un reporte de ordenes junto con el detalle de las mismas.
3. Visualizar de una forma gráfica y detallada los productos más vendidos.
4. Visualizar de una forma gráfica y detallada los productos menos vendidos.
5. Visualizar una lista de los pedidos realizados para cierto período.
6. Visualizar una lista de los productos sugeridos en base a la relación de los mismos con pedidos realizados.
7. Visualizar proyecciones de ventas para los productos seleccionados.

## **B. Requerimientos**

Como requerimiento primordial para este módulo está el mostrar la información de una forma clara, concisa y ordenada. El propósito del proyecto es ayudar a la gerencia a tomar mejores decisiones sobre sus compras y es por eso que este módulo es de suma importancia. Por medio de este módulo los usuarios de FPK podrán visualizar de una forma gráfica y esquematizada cómo se han comportado las órdenes, que productos tienen más movimiento, cómo compran los clientes y cómo estarán las ventas en el futuro basándose en ventas pasadas.

La claridad y orden de la presentación de los reportes es de suma importancia y es por eso que emplear herramientas gráficas y tablas de resumen es un requerimiento clave para presentar la información recopilada de una forma ordenada y amigable. Es de suma importancia poder filtrar la data para poder visualizar solamente lo necesario en base al contexto y a la situación que se esté evaluando.

Finalmente como requerimiento se tiene la optimización del manejo de los datos. Debido a que la cantidad de información que se estará consultando es amplia y ya que el crecimiento de los datos es lo suficiente agresivo para considerar emplear métodos de optimización al momento de realizar consultas a la base de datos, construir tablas y realizar gráficas. Es necesario poder acceder a los datos de una forma rápida sin arriesgar la integridad de los datos.

## **C. Diseño**

**1. Comportamiento de productos.** Para poder mostrar el comportamiento de los productos es necesario que el usuario seleccione el producto que desea consultar y con base en eso generar un reporte completo de cómo éste se ha

comportado en las ventas según el sistema. Para hacer esto de una forma amigable es necesario organizar los datos de ventas organizadas por número de orden y catalogadas por cliente siempre mostrando la cantidad que es lo más importante.

Graficar el comportamiento de los productos ayuda a tener un entendimiento más rápido y directo de cómo se ha comportado el producto al pasar del tiempo. A diferencia de una tabla en donde se muestran los valores de cantidad por orden, en una tabla se pueden ver las ventas con respecto al tiempo por lo que el comportamiento del producto es más evidente.

Para poder graficar es necesario utilizar bibliotecas de JavaScript para poder dibujar en un documento de HTML. Para el proyecto de FPK, en todas sus graficas se utilizaron las bibliotecas de Highcharts. Estas bibliotecas brindan una forma práctica de dibujar data obtenida de una base de datos, lo que es ideal para el presente proyecto. Highcharts fue seleccionada como biblioteca principal para graficar ya que es bastante poderosa, es muy amigable y es gratis.

**2. Reportes de ventas.** Los reportes de ventas deben reflejar de una forma clara como se han llevado a cabo las ventas de todos los clientes mayoristas. Para esto es necesario permitir al usuario filtrar la información que desea visualizar. Los filtros son necesarios debido a que la cantidad de ventas es sumamente grande y no es amigable mostrar la información de todas las ventas registradas en el sistema.

Los filtros implementados en los reportes de ventas fueron seleccionados en base al criterio del encargado actual de las ventas y le permiten visualizar la data ordenada como actualmente es requerida para el funcionamiento del departamento pertinente. El sistema cuenta con los filtros de:

- a. Cliente: Permite determinar las ventas de un cliente en específico. Es requerido para poder determinar las compras de un cliente determinado.
- b. Estado: Permite determinar el estado de las ordenes a ser desplegadas. Debido a que es posible constar con órdenes de diferente estado es importante poder determinar el estado que se desea visualizar.
- c. Rango de fechas: El rango de fechas en la que la compra fue realizada. Una necesidad es poder mostrar las ventas realizadas en un rango de fechas específico y es por eso que se implemento el presente filtro.

**3. Récorde de ventas.** Los récorde de ventas se refiere a los productos más vendidos y los menos vendidos. Para este reporte se tomaron los 20 productos más vendidos y los 20 menos vendidos. Es necesario mostrar los productos más vendidos para poder tomar estrategias de ventas más agresivas y explotarlos al máximo pero también es necesario mostrar los menos vendidos ya que esto ayudar a tomar mejores decisiones de mercadeo y decidir como poder abordar el problema de las pocas ventas.

Para esta sección la información se despliega en una tabla ordenada para cada récord. Junto con el nombre del producto se muestra la cantidad de ventas que se han registrado para el mismo. Esto se hace tanto para los más vendidos como para los menos vendidos.

La tabla es una forma ordenada de visualizar los resultados pero una gráfica es más concisa y brinda la información de una forma más clara y directa. Es por eso que se decidió graficar el resultado empleando las mismas bibliotecas mencionadas con anterioridad.

**4. Sugeridos de compra.** Los sugeridos de venta son herramientas estadísticas que ayudan a poder predecir como se comportarán los productos en base a experiencias pasadas. El objetivo principal de esta sección es brindarle a los administradores un reporte completo de lo que se debe comprar para cierto período seleccionado por el mismo. Para lograr esto se debe utilizar una mezcla de data y herramientas definidas.

El sugerido de compra consta de tres partes diferentes que muestran una visión más clara de lo que es necesario comprar a los proveedores. Estas partes son las siguientes:

**a. Pedidos realizados.** Los pedidos realizados son todos aquellos pedidos que se han hecho por los clientes en la fecha estipulada por el usuario. Estos datos son reales ya que brindan la demanda explícita solicitada por los clientes. Esta recopilación de datos reales muestra las ventas reales que se deben hacer para el periodo establecido. Es de suma importancia tomar en cuenta que hacer una compra a los proveedores por una cantidad menor de producto que la estipulada en esta sección repercute en no satisfacer la demanda de los clientes.

Este dato es el más valorado por los administradores ya que estas son ventas seguras a diferencia de las otras dos secciones que se basan más en la estadística y en supuestos del sistema.

**b. Sugeridos de compra.** La segunda sección es la de sugeridos de compra. En esta sección el sistema sugiere a los administradores los productos que él considera se deben comprar basándose en las relaciones de productos previamente establecidas. Esto se hace con el propósito de tratar de mejorar la posible demanda

futura de producto altamente relacionado a los productos que ya han sido comprados por los clientes y que están listados en la sección anterior.

El mejor ejemplo de esto son los teclados marca Casio y los cargadores de los mismos. Durante las reuniones con FPK se llego a la conclusión que es necesario que el sistema advierta cuando se está comprando un producto que está altamente relacionado con otro. Siguiendo el ejemplo si se compran n teclados es necesario comprar n o más cargadores por lo que si no se ha solicitado por el cliente explícitamente es tarea del sistema alertar que va a ser necesario comprar los cargadores.

**c. Proyecciones de ventas.** La tercera sección es la de proyecciones de venta. Ésta consta de procesos estadísticos que, con base al comportamiento de los productos en los mismos períodos pero de años diferentes. Esta sección muestra los períodos de dos años atrás junto con cuánto producto se compró, cuánto se vendió y de acuerdo a esos dos datos se calcula la efectividad de venta del producto. Ambos datos son empleados para calcular la tasa de crecimiento normal o vegetativo. La tasa de crecimiento normal es simplemente la razón de crecimiento o decaimiento en las ventas del producto. La tasa por sí sola no es de mucha ayuda por lo que se convierte en un dato inútil, pero al combinarla con los datos históricos de ventas se obtiene una proyección de venta de producto para el período actual.

Esta sección ayuda a los administradores a determinar de una forma más precisa cómo será el comportamiento del producto. Esto se hace con el propósito de que el administrador pueda observar qué puede pasar si varía los montos de producto a comprar lo que puede ayudar a comprar más y lograr vender más producto o ser más

reservado y tímido con las compras ya que los históricos muestran que el producto determinado no se ha vendido bien en años anteriores.

Es importante recalcar que la proyección calculada posee un 95% de efectividad y certeza. Este dato es de suma importancia ya que respalda la veracidad y la integridad del valor proyectado. Se escogió este estadístico ya que su margen de error es lo suficientemente bajo para despreciarlo.

**Figura 19 – Vista del módulo de estadística y sugeridos de compra**

**Fecha de Compra:** Diciembre ▾ 2011 ▾ 

---

**Pedidos Realizados**

Producto	Codigo	Precio	Cantidad	Sub Total
PRIZM fx-CG10	fx-CG10	800.00	20	16,000
CTK-6000	CTK-6000	875.00	5	4,375

**Sugeridos de Compra**

Producto	Codigo	Precio	Cantidad	Sub Total
AD-5 Keyboard AC Adapter	AD-5 Keyboard AC Adapter	90.00	5	450
SP3 Sustain Pedal	SP3 Sustain Pedal	99.00	5	495

**Proyecciones**

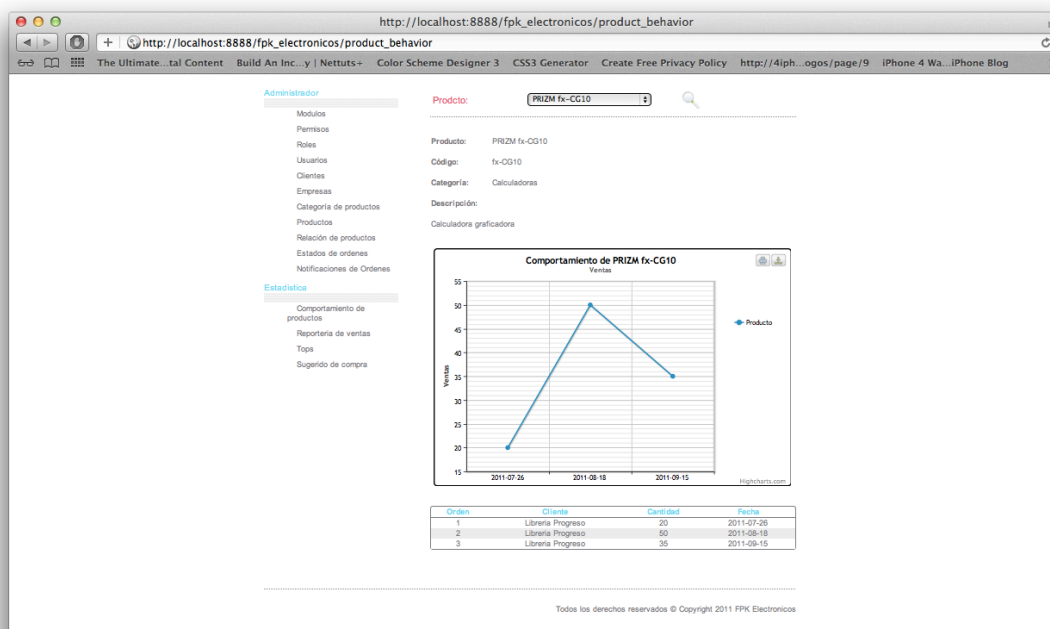
PRIZM fx-CG10  
 JF-100MS  
 FX-300ES  
 SL-300SV  
 DR-210TM  
 SA-76  
 LK-230  
 AP-6  
 CTK-6000

[Generar Proyecciones](#) 

Producto	Diciembre 2009			Diciembre 2010			Tasa	Proyeccion Diciembre 2011
	Ventas	Compras	Efectividad	Ventas	Compras	Efectividad		
PRIZM fx-CG10	50	75	66.67%	62	90	68.89%	0.2151	<b>77</b>
CTK-6000	10	39	25.64%	25	30	83.33%	0.9163	<b>63</b>

Los administradores pueden tener acceso a determinar que productos se han comprado en cierto período lo que les brinda un dato exacto de cuanto deben comprar para suplir la necesidad de sus clientes mayoristas. La segunda tabla muestra los sugeridos de producto en base a los productos comprados. La tercera sección muestra los productos para que los administradores proyecten las ventas. La última tabla muestra las proyecciones de dos productos seleccionados por el administrador. En esta se muestra cantidad de producto comprado y vendido en los dos últimos períodos. Estos valores son utilizados para calcular la efectividad del producto en el mercado. Los últimos dos valores son la tasa de crecimiento calculada para cada producto y el valor estimado para el período seleccionado.

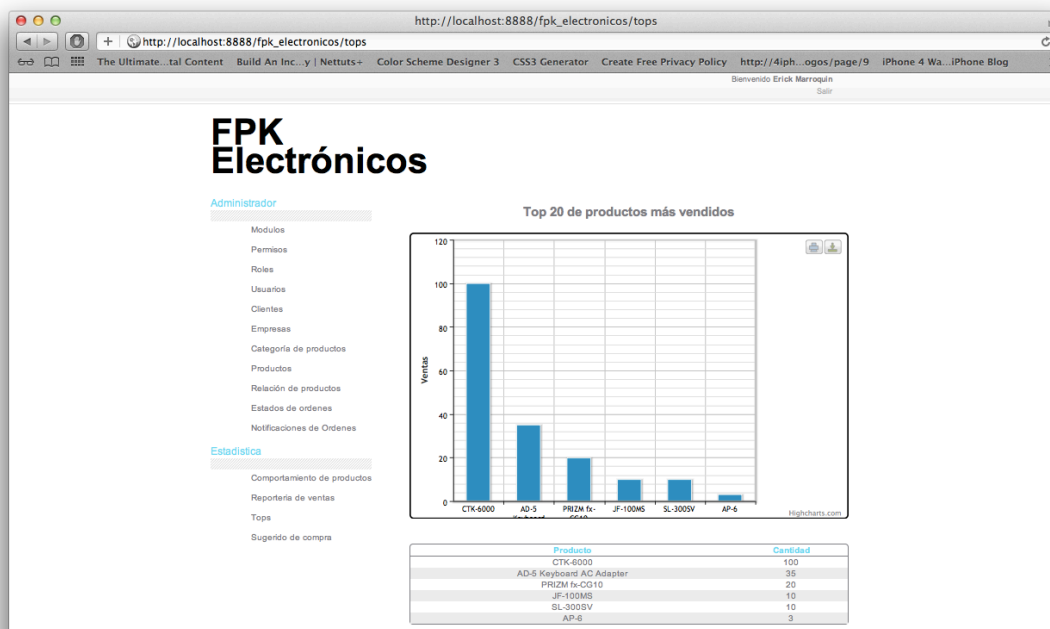
**Figura 20 – Vista del reporte de comportamiento de producto**



El comportamiento de producto muestra de una forma gráfica como se ha comportado un producto en el tiempo. Se muestran las ventas ordenadas por fechas. La

gráfica es complementada con una tabla que muestra estos datos de una forma ordenada.

**Figura 21 – Vista del reporte de productos más vendidos**



Las gráficas de productos más vendidos reflejan los 20 productos que más se han vendido y los 20 productos que menos se han vendido en una gráfica de barras bastante amigable.

**5. Pruebas de efectividad.** Es de suma importancia para el proyecto determinar la efectividad del módulo estadístico de proyectados de compra sea exacto y realmente refleje valores que concuerden en el escenario en el que FPK se está desarrollando. Es por este motivo que se diseñó una prueba para validar los resultados obtenidos por el software y determinar su credibilidad en el ámbito de compras de FPK.

**a. Descripción de la prueba.** La prueba consiste en utilizar la experiencia y el conocimiento del mercado que tienen los vendedores de FPK. Siendo éste el único punto fidedigno de referencia con el que se cuenta, la prueba debe validarse con los resultados que los vendedores indiquen.

Debido a que ésta debe ser una prueba comparativa en la que se comparan los resultados de los vendedores y el resultado del software fue necesario crear un escenario en el que se tuvieran las mismas condiciones e información para que ambas partes pudieran brindar respuestas informadas. En la prueba se seleccionaba un producto real del inventario de FPK, este debía ser un producto que todos los vendedores conocieran bien y que todos hubieran vendido con antelación. Luego se les presentaba un escenario en el que por dos períodos de tiempo distintos se les incluía la información de cuantas unidades del producto seleccionado se habían comprado para inventario y cuantas de esas unidades de habían vendido. Finalmente la tarea de los vendedores era simplemente anotar su predicción de lo que ellos creían que se fuera a vender para el tercer período.

Con la predicción de los vendedores y el escenario planteado se procedió a generar el mismo escenario en la base de datos del software para que este hiciera la predicción que, posteriormente se validaría con los resultados de los vendedores.

**b. Resultados de la prueba.** Los resultados tabulados de la prueba presentada a los vendedores como el resultado que el software generó se presentan en la tabla siguiente.

**Tabla 24 - Resultados de prueba estadística**

Vendedor	Proyección Vendedor	Proyección Software	Diferencia
1	76	77	1
2	70		7
3	75		2
4	80		3

**c. Interpretación de resultados.** Para obtener un resultado mucho más exacto y apegado a la realidad de ventas de FPK se procedió a calcular el promedio ponderado de los datos proyectados por los vendedores. Para calcular el promedio ponderado es necesario sumar todos los valores obtenidos. A esa sumatoria se le debe adicionar el resultado de multiplicar 4 y el valor más alto obtenido. Para el siguiente caso el primer paso para calcular el promedio sería:  $76 + 70 + 75 + 80 + (4 * 80) = 621$ .

El segundo paso es dividir el resultado obtenido por las sumatorias dentro de una constante, que por la fórmula de promedio ponderado, es 8. En el caso planteado el resultado es 77.63. Con el cálculo del promedio ponderado se procede a obtener la varianza entre el promedio calculado de las proyecciones de los vendedores y el promedio calculado de la proyección del software que, debido a que siempre es el mismo valor para este escenario, va a ser 77.

La varianza es obtenida al restar los dos promedios por lo que se obtiene una varianza de 0.63%. Estadísticamente una varianza menor al 1% es altamente aceptada

ya que se asemeja mucho a la realidad apegándose al modelo de ventas que los vendedores han desarrollado en base a su experiencia y conocimiento del mercado.

## IX. DISCUSIÓN

El uso del framework Symfony ayudó a construir ambas aplicaciones del proyecto de una forma modular por lo que se logró independizar de una forma eficiente y práctica cada uno de los módulos que conforman las aplicaciones. El modular ayuda para la reducción de errores ya que estos no se propagan a los otros módulos causando más problemas. Junto con la reducción de errores se garantiza que la construcción de soluciones adicionales a las del alcance del proyecto pueda hacerse de una forma fácil sin la necesidad de cambiar todo el proyecto o la aplicación a la cual pertenece el módulo.

El buen diseño de una base de datos que se apega al modelo relacional permite que la comunicación entre la misma y las aplicaciones sea más eficiente. Debido al modelo modular del proyecto la integración con la base de datos relacional es sumamente simple y casi transparente para el desarrollador ya que ambos modelos trabajan muy bien juntos. Es muy importante resaltar que la base de datos es compartida por ambas aplicaciones lo que pone una gran responsabilidad sobre el diseño de la misma ya que esta debe ser lo suficientemente robusta para soportar la carga de ambos sitios.

El uso del modelo vista controlador favorece en garantizar el orden de la comunicación entre estas tres entidades lo que brinda una mayor seguridad, abstracción entre cada entidad y facilita la creación de diferentes ambientes ya sea para desarrollo, mantenimiento o pruebas del sistema. Esto se da ya que se restringe la comunicación directa entre ciertos modelos y se fuerzan a comunicarse de una forma que se presta a que el proyecto cuente con una mayor abstracción. Para el proyecto es altamente

beneficioso poder crear diferentes ambientes ya que se puede desarrollar mientras se realizan pruebas de funcionamiento y si en un futuro es necesario implementar más módulos se puede hacer sin poner en alto toda la aplicación ejecutándose en ese momento.

El uso de la tasa de crecimiento vegetativo o normal permite realizar proyecciones de ventas ya que se toman en cuenta las ventas pasadas y se aplican identificadores estadísticos como la eficiencia de venta y la razón de cambio entre ventas de diferente período para genera una tasa de crecimiento empleada para la proyección de ventas para un período seleccionado. Este estadístico fue seleccionado ya que es el que mejor se apega a la necesidad de poder predecir las ventas que se llevarán a cabo. Es necesario mencionar que, a pesar de ser el indicador necesario no garantiza que exista algún margen de error que puede ser ocasionado por fluctuaciones en la economía y/o causas externas y de fuerza mayor que no fueron contempladas y que son impredecibles.

## X. CONCLUSIONES

1. La variación real calculada entre la proyección de los vendedores y la proyección del software que se obtuvo en las pruebas de eficiencia es menor al 1% lo que es válido y aceptable en un escenario estadístico. Esto indica que el software realizado se asemeja en sus proyecciones a las de los vendedores brindándole un soporte estadístico a los sugeridos de compra y no solo un elemento empírico.
2. El módulo de estadística genera suficientes reportes para que los administradores de FPK Electrónicos puedan realizar compras más precisas conociendo a fondo el comportamiento de los productos en el mercado.
3. El desarrollo del módulo de control interno cumple la necesidad de poseer sitios completamente administrables en los que se maneje el inventario, las relaciones de productos, los reportes de venta, las proyecciones estadísticas y el manejo de todo lo referente a usuarios tanto administradores como clientes mayoristas.
4. El uso del framework Symfony permitió hacer el proyecto de una forma completamente modular y garantizar la división en las dos aplicaciones del proyecto que son el backend, sitio de administración interna y el frontend, sitio de venta y reportes para clientes mayoristas.
5. El uso de la estadística, más específicamente la tasa de crecimiento normal o vegetativo, conjuntamente con los datos históricos de las ventas previamente realizadas permitió calcular proyecciones de ventas para predecir el comportamiento del producto en el mercado para un período de tiempo determinado.

6. El uso de una base de datos que se apega al modelo relacional y que cumple con las reglas de la normalización ayudó a eficientar los procesos de consultas, manejo de datos y permitió poder integrar las dos aplicaciones del proyecto dentro de una misma estructura.
  
7. El uso del modelo vista controlador permitió crear diferentes ambientes para desarrollo y pruebas lo que permite tener un sistema al que se le puede desarrollar y dar mantenimiento al mismo tiempo haciendo más eficiente el trabajo en el proyecto.

## **XI. RECOMENDACIONES**

Se recomienda fuertemente mantener una base de datos limpia y ordenada. Es decir sin registros vacíos o data insignificante ya que esto puede afectar directamente el desempeño de las consultas a la misma y hacer lento el sistema. El mantenimiento a la base de datos se recomienda hacerse cada año en el cierre del año fiscal para así iniciar un nuevo régimen con una base de datos rápida y eficiente. Para lograr esto es necesario tomar en cuenta el volumen de datos a ingresarse y, dependiendo del mismo se pueden implementar rutinas de segmentación de tablas e indexación de las mismas para que las consultas se hagan sobre tablas con menor información y el tiempo de consulta baje considerablemente.

Se recomienda utilizar un servidor dedicado o virtualmente dedicado con sistema operativo Linux debido a que el uso del framework solicita la autenticación y permisos de administrador y debido a que el proyecto sólo fue probado bajo estas circunstancias. Es por esto que no se garantiza que pueda correr sobre un servidor con otro sistema operativo y sin acceso de administrador como lo sería un servidor compartido. El sistema operativo brinda una mayor flexibilidad por lo que aunque exista una versión compatible para un sistema operativo diferente, la versatilidad y estabilidad de un servidor Linux es mucho más confiable.

Si fuese necesario desarrollar más módulos estadísticos se recomienda hacer análisis de mercados para poder hacer predicciones no sólo basándose en las ventas anteriores sino también en el estado actual del mercado. Esto aproximará de una mejor manera las proyecciones brindando un dato más exacto para la mejor toma de decisiones en la administración.

## XII. REFERENCIAS BIBLIOGRÁFICA

Microsoft. (2011). *MSDN*. Retrieved 20 de 8 de 2011 from Model View Controller:  
<http://msdn.microsoft.com/en-us/library/ff649643.aspx>

Microsoft. (22 de 2 de 2007). *Fundamentos de la normalización en base de datos*. Retrieved 2 de 10 de 2011 from Microsoft Support:  
<http://support.microsoft.com/kb/283878/es>

n.a. (10 de 08 de 2009). *Model View Controller(MVC) in PHP*. Retrieved 2 de 09 de 2011 from php-html: <http://php-html.net/tutorials/model-view-controller-in-php/>

n.a. (2000). *Crecimiento Vegetativo*. Retrieved 12 de 08 de 2011 from GEA:  
[http://www.encyclopedia-aragonesa.com/voz.asp?voz\\_id=4382](http://www.encyclopedia-aragonesa.com/voz.asp?voz_id=4382)

n.a. (2007). *El modelo relacional*. Retrieved 28 de 9 de 2011 from Universidad del Azay:  
<http://www.uazuay.edu.ec/analisis/El%20modelo%20relacional.pdf>

Pavón, J. (2009). *Facultad de Informática Universidad Complutence de Madrid*. Retrieved 12 de 8 de 2011 from El Patrón Modelo-Vista-Controlador (MVC):  
<http://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>

Quiroz, J. (2003). *Instituto nacional de estadística y geografía*. Retrieved 25 de 09 de 2011 from El modelo relaciona de base de datos:  
<http://www.inegi.gob.mx/inegi/contenidos/espanol/prensa/Contenidos/Articulos/tecnologia/relacional.pdf>

SensioLabs. (2009). *Symfony Proyect*. Retrieved 20 de 09 de 2011 from About Symfony:  
<http://www.symfony-project.org/about>