

Universidad Autónoma de Baja California

Facultad de Contaduría y Administración.

Maestría en Tecnologías de la Información y la Comunicación.



Título de la Tesis:

EXTRACCIÓN Y PROCESAMIENTO DINÁMICO DE INFORMACIÓN OPERATIVA EN LA EMPRESA
MI TECHNOLOGIES INC.

Para Obtener el Grado de:

Maestro en Tecnologías de la Información y la Comunicación

Presenta:

Faustino Vásquez Limón

Bajo la dirección de:

MC. José Manuel Villegas Izaguirre.

Una firma manuscrita en tinta azul que parece ser la del director de la tesis, José Manuel Villegas Izaguirre. La firma es fluida y difícil de leer, pero se reconoce el nombre.

DEDICATORIA.

A mis padres Faustino Vásquez López (QEPD) e Irma Limón Romero

Por su amor y la vida que me dieron, pero sobre todo por los valores que inculcaron en mí y que me han permitido salir adelante en todas las situaciones que la vida me ha puesto.

A mi esposa Bibiana Tenorio Arriaga.

Por haberme apoyado en todo momento, por su paciencia, sus valores, por la motivación constante que me ha permitido lograr este objetivo, pero más que nada, por su amor.

A mi hijo Sebastián Vásquez Tenorio.

Que es el motor que me obliga a ser mejor cada día mejor, quien es mi inspiración y motivación para alcanzar cada objetivo que me he propuesto y a quien dedico no solo esta tesis si no toda mi vida, te amo hijo.

Hoja en Blanco.

AGRADECIMIENTOS.

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo, en especial al Sr. Amir Tafreshi, CEO de la compañía MI Technologies Inc., quien me ha permitido realizar el proyecto que se expone en este documento. Sus ideas han sido plasmadas en código y han permitido crear un proyecto útil a la compañía.

Quisiera hacer extensiva mi gratitud a los compañeros del Departamento de Sistemas, por su apoyo y trabajo en equipo, que nos ha permitido crecer en conocimientos y destacarnos en el área del área de desarrollo de software.

También quiero dar las gracias al MC. José Manuel Villegas Izaguirre. Por su apoyo en la dirección para el desarrollo de este documento.

Un agradecimiento muy especial merece la comprensión, paciencia y el ánimo recibidos de mi familia y amigos.

A todos ellos, muchas gracias.

RESUMEN

Hoy en día, instituciones, dependencias y empresas privadas, administran excesivos volúmenes de datos. En todos los niveles de la organización existe una necesidad y alta demanda de sistemas que permitan recolectar, procesar y extraer información desde diferentes fuentes, lo cual crea una necesidad hacia dentro de las organizaciones de implementar sistemas computarizados que realicen estas actividades y permitan la toma de decisiones con base en la información obtenida.

El primer capítulo de este documento, describe a grandes rasgos el entorno corporativo, se plantea la problemática, se establece el objetivo general, y los objetivos específicos, para el caso práctico: “Extracción y procesamiento dinámico de información operativa en la empresa Mi Technologies Inc.”, así también se definen sus alcances y límites.

El segundo capítulo, trata el marco teórico y la importancia que tiene adoptar procesos estandarizados, con normativas internacionales en el proceso de administración de proyectos específicos, así como el uso de marcos de trabajo para gestionar el desarrollo y mantenimiento de software. Este capítulo también contiene una síntesis de cada una de las herramientas de software que ayudaron a alcanzar el objetivo general.

En el tercer capítulo, detalla proceso metodológico, los modelos para administrar el desarrollo de software, el plan del proyecto, y los procesos de ingeniería necesarios para la realización del objetivo.

El cuarto capítulo, desarrolla la ingeniería indicada en el proceso metodológico.

En el quinto capítulo, presenta los resultados obtenidos en las mediciones, que la metodología indica, a fin de mostrar en forma cuantitativa, la madurez del proceso que permitió el alcance del objetivo general.

El sexto capítulo, relata una serie de conclusiones a las que se ha llegado, después de los primeros tres meses de operación del sistema EyPDIO.

Capítulo 1 <i>Introducción</i>	13
1.1 <i>Introducción</i>	14
1.2 <i>Antecedentes del problema</i>	14
1.3 <i>Definición del problema</i>	14
1.4 <i>Justificación de la solución</i>	15
1.5 <i>Objetivo general</i>	15
1.6 <i>Objetivos específicos</i>	16
1.7 <i>Alcance del trabajo</i>	16
Capítulo 2 <i>Marco teórico</i>	17
2.1 <i>Introducción</i>	18
2.2 <i>Metodología Ágil</i>	18
2.2.1 <i>El manifiesto Ágil</i>	18
2.3 <i>Scrum</i>	20
2.3.1 <i>Responsabilidades y roles de scrum</i>	21
2.3.2 <i>Elementos de scrum</i>	21
2.3.3 <i>Sprint</i>	22
2.3.4 <i>Valores</i>	24
2.4 <i>MVC (Modelo Vista Controlador)</i>	25
2.4.1 <i>Ventajas de MVC (Pantoja)</i>	26
2.4.2 <i>Desventajas de MVC (Pantoja)</i>	26
2.5 <i>PHP</i>	26
2.5.1 <i>Ventajas de PHP</i>	27
2.5.2 <i>Desventajas de PHP</i>	28
2.6 <i>Bases de datos</i>	28
2.6.1 <i>Bases de datos relacionales</i>	29
2.6.2 <i>Lenguaje de consulta en base de datos</i>	29
2.7 <i>Herramientas web</i>	30
2.7.1 <i>HTML</i>	30
2.7.2 <i>Hojas de estilo en cascada (CSS)</i>	31
2.7.3 <i>JavaScript</i>	31
Capítulo 3 <i>Metodología</i>	33
3.1 <i>Introducción</i>	34
3.2 <i>Etapas del ciclo de vida de un producto</i>	34
3.3 <i>Planeación</i>	34
3.4 <i>Análisis</i>	34
3.5 <i>Arquitectura y diseño</i>	35
3.6 <i>Desarrollo y pruebas</i>	35
3.7 <i>Puesta en marcha</i>	35

3.8	<i>Herramientas a utilizar</i>	36
Capítulo 4 Desarrollo		38
4.1	<i>Introducción</i>	39
4.2	<i>Planeación</i>	39
4.2.1	Definición del proyecto.....	39
4.2.2	Estimación del alcance del proyecto (EDT).....	39
4.2.3	Pila del producto.....	40
4.2.4	Integración del equipo de trabajo.	42
4.2.5	Costo estimado.....	42
4.2.6	Calendario de actividades.....	42
4.2.7	Plan de manejo de riesgos.....	43
4.2.8	Protocolo de entrega.....	44
4.3	<i>Análisis del sistema</i>	44
4.3.1	Requerimientos funcionales.....	44
4.3.2	Requerimientos no funcionales.....	49
4.3.3	Otros requerimientos funcionales.....	50
4.4	<i>Arquitectura y diseño</i>	50
4.4.1	Vista general de la arquitectura del software.....	50
4.4.2	Diseño detallado del sistema.....	55
4.5	<i>Desarrollo y pruebas</i>	56
4.6	<i>Puesta en marcha</i>	58
Capítulo 5 Resultados		59
5.1	<i>Descripción de resultados</i>	60
5.1.1	Resultados a nivel general.....	60
5.1.2	Análisis de los objetivos particulares definidos para el sistema.....	60
Capítulo 6 Conclusiones		67
6.1	<i>Conclusiones</i>	68
Anexo A		70
Anexo B		85

Figura 2-1 Fases de desarrollo de un Sprint.....	22
Figura 2-2 Modelo Vista Controlador.	25
Figura 2-3 Estructura de una tabla relacional.....	29
Figura 3-1 EDT completo para el sistema EyPDIO.....	36
Figura 4-1 Estructura de división, perspectiva del producto.	39
Figura 4-2 Descripción de caso de uso consulta al catálogo de productos.	46
Figura 4-3 Desarrollo del caso de uso acceso al sistema.	46
Figura 4-4 Desarrollo del Caso de uso Consulta al catálogo de productos.....	47
Figura 4-5 Casos de uso del ciclo catálogo.....	49
Figura 4-6 Diagrama de arquitectura de bloques.	51
Figura 4-7 Diagrama de Arquitectura Lógica.	52
Figura 4-8 diagrama de arquitectura Física.	53
Figura 4-9 Diagrama de componentes.....	54
Figura 4-10 Caso de uso 'Consulta al catálogo de productos'.	55
Figura 4-11 Caso de uso 'Datos complementarios'.....	56
Figura 4-12 Caso de uso 'Información detallada del producto'.....	56
Figura 5-1 Estaciones de trabajo con computadora.	60
Figura 5-2 Administración de usuarios en ERP	61
Figura 5-3 Imágenes de productos en catálogo.....	62
Figura 5-4 Costos de Inventario.	62
Figura 5-5 Panel de control con gráficos interactivos.....	63
Figura 5-6 Ventas por mes.	63
Figura 5-7 ventas por mes, proyectores	63
Figura 5-8, Ventas totales por año.....	64
Figura 5-9 Ventas por tienda virtual en un periodo.....	64
Figura 5-10 Ventas por agente de venta.....	64
Figura 5-11 Ventas por Agente de venta en México.....	64
Figura 5-12 Artículos y ordenes x agente de vta.....	64
Figura 5-13 Los diez artículos mas vendidos.....	64
Figura 5-14 Gráficos de status para cada artículo en el inventario.	65
Figura 5-15 Consolidados de venta.....	65
Figura 5-16 Detalle de la orden de venta.....	66
Figura 5-17 Servidor que hospeda al sistema EyPDIO.	66
Figura A-1 Caso de uso "Acceso al sistema".	70
Figura A-2 Caso de uso "Consulta al Catálogo de Productos".	71
Figura A-3 Caso de prueba "Consulta al catálogo de productos".....	72
Figura A-4 Caso de uso "Datos complementarios".....	72
Figura A-5 Caso de uso "Información de producto".	73
Figura A-6 Caso de uso "Campos personalizados".....	73
Figura A-7 Caso de uso "Compatibilidades".	74
Figura A-8 Caso de prueba "Compatibilidades".....	74
Figura A-9 Caso de uso "Imágenes".	75
Figura A-10 Caso de uso "Documentos".	75
Figura A-11 Caso de prueba "Historial".	76
Figura A-12 Caso de uso "Detalles de Inventario".	77

Figura A-13 Caso de uso "Requerimientos de Ensamble"	77
Figura A-14 Caso de uso "Gráficos estadísticos del producto"	78
Figura A-15 Caso de prueba "Gráficos estadísticos del producto"	79
Figura A-16 Caso de uso "Comparativa de ventas por mes y años"	80
Figura A-17 Caso de uso "Ventas totales por año de lámparas de proyector divididas por mes".	80
Figura A-18 Caso de uso "Ventas diarias CSR en los últimos 30 días"	81
Figura A-19 Caso de uso "Total de ventas por anuales"	81
Figura A-20 Caso de uso "Ventas totales por canal de venta"	82
Figura A-21 Caso de uso "Artículos y órdenes, vendidos y enviados por canal de venta"	82
Figura A-22 Caso de uso "Diez productos más vendidos"	83
Figura A-23 Caso de uso "Total de ventas por agente de venta a mayoreo"	83
Figura A-24 Caso de uso "Total de ventas por agente de venta en México"	84
Figura A-25 Caso de uso "Total de artículos y órdenes vendidos por agente de venta en México"	84
Figura B-1 Primera parte del modelo Relacional de la base de datos Inventory	85
Figura B-2 Segunda parte del modelo Relacional de la base de datos Inventory	86
Figura B-3 Tercera parte del modelo Relacional de la base de datos Inventory	87
Figura B-4 Cuarta parte del modelo Relacional de la base de datos Inventory	88
Figura B-5 Desarrollo del caso de uso "Consulta al catálogo de productos"	89
Figura B-6 Diagrama de actividad "Consulta al catálogo de productos"	90
Figura B-7 Vista del caso de uso "Consulta al catálogo de productos" en fase de diseño	91
Figura B-8 Vista del caso de uso "Consulta al catálogo de productos" en fase de desarrollo	91
Figura B-9 Caso de prueba "Consulta al Catálogo de productos"	92

Tabla 2-1 Diferencias entre metodologías ágiles y no ágiles.....	20
Tabla 4-1 Estimación de la duración del proyecto.....	40
Tabla 4-2 Lista detallada de requerimientos.....	41
Tabla 4-3 Equipo de trabajo.....	42
Tabla 4-4 Calendario de actividades para cada iteración.....	43
Tabla 4-5 Análisis cualitativo y cuantitativo del riesgo.....	43
Tabla 4-6 Estrategias seleccionadas en caso de riesgo.....	43
Tabla 4-7 Contramedidas seleccionadas para evitar el riesgo.....	44
Tabla 4-8 Protocolo de Entrega.....	44
Tabla 4-9 Identificación de grupos de usuarios.....	45
Tabla 4-10 Lista de requerimientos actualizada y agrupada en ciclos de desarrollo.....	48
Tabla B-1 Análisis de resultados del proceso de revisión y retroalimentación.....	92

Definiciones, Acrónimos y Abreviaciones

.NET: Framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

AJAX: Asynchronous JavaScript and XML (JavaScript asíncrono y XML).

Amazon: Librería mundial online. Ofrece más de un millón de títulos a través de Internet.

APE: Administración de Proyectos Específicos.

ASP: Active Server Pages (Servidor de Páginas Activas).

C: Lenguaje de programación de propósito general, potente y con campo de acción ilimitado.

C++: Extensión del lenguaje de programación C con mecanismos que permiten la manipulación de objetos.

CAO: Centro de administración de órdenes.

CGI: Common Gateway Interface (Interfaz de entrada común).

ColdFusion: Servidor de aplicaciones y lenguaje de programación usado para desarrollar aplicaciones de Internet, generalmente sitios web generados dinámicamente. Es un producto similar a ASP, JSP o PHP.

CSR: Customer Service Representing (Representante de Atención al Cliente).

Dashboard: Panel de control.

DHTML: Dynamic HTML (HTML Dinámico).

DLP: Digital Light Processing (Procesamiento digital de la luz).

DMS: Desarrollo y Mantenimiento de Software.

DOIR: Days Of Inventory Remaining (Días restantes de inventario).

DOM: Document Object Model (Modelo de Objetos del Documento).

Ebay: Sitio comercial on-line donde cada persona puede comprar o vender mercadería de todo tipo.

ECMA Javascript: Lenguaje se usa ampliamente para scripting del lado del cliente en la web, en la forma de varios dialectos conocidos como JavaScript, JScript.

EDT: Estructura de División de Trabajo.

ERP: Enterprise Resource Planning (Planificación de Recursos Empresariales).

EyeDIO: Extracción y Procesamiento Dinámico de Información Operativa.

Firefox: Navegador web libre y de código abierto, desarrollado para Microsoft Windows, Mac OS X y GNU/Linux coordinado por la Corporación Mozilla y la Fundación Mozilla.

GNU: GNU is Not Unix (GNU No es Unix).

GPL: General Public License (Licencia Pública General de GNU).

IE: Internet Explorer.

Java: Lenguaje de programación de alto nivel orientado a objetos, desarrollado por James Gosling en 1995. Similar a C, Cobol y Visual Basic, pero con modelo de objetos más simple además de eliminar las herramientas de bajo nivel.

JSP: JavaServer Pages (Servidor de páginas Java).

MVC: Model View Controller (Modelo Vista Controlador)

OM: Order Manager (Manejador de Ordenes).

PERL: Lenguaje de programación similar a C o AWK, ampliamente adoptado por su destreza en el procesamiento de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

PMI: Project Management Institute.

Safari: Navegador web de código cerrado desarrollado por Apple Inc. Está disponible para Mac OS X, iOS

SGBDR: Sistema de Gestión de Base de Datos Relacionales.

SPICE: Software Process Improvement and Capability Estimation (Estimación de la capacidad para mejora de los procesos de software).

Sprint: Denomina a una iteración para desarrollo de software que está acotada en el tiempo, usualmente entre 2 y 4 semanas.

W3C: World Wide Web Consortium (consorcio internacional que produce recomendaciones para la World Wide Web.)

XML: eXtensible Markup Language (lenguaje de marcas extensible).

XML-RPC: Protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes (Laurent, Johnston, & Dumbill, 2001).

ZEND: Framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5.

Capítulo 1 **Introducción.**

1.1 Introducción.

Los cambios económicos, tecnológicos, sociales y culturales que se viven a nivel global presentan nuevos retos y también nuevas oportunidades a las organizaciones. Enfocarse en un nicho de mercado requiere de una alta capacidad de innovación que permita dar continuidad a los objetivos de la empresa y no ser simplemente un boom por un tiempo determinado o una opción más. La competencia siempre va en aumento y por esta razón no se debe descuidar la cuota de mercado que se está disputando o la que ya se ha logrado.

Con lo anteriormente dicho, MI Technologies Inc. empresa dedicada al comercio electrónico ha llegado a ser uno de los más grandes distribuidores de partes electrónicas para reparación de televisores de última generación y el más grande distribuidor de todo tipo de lámparas de proyección trasera y frontal, para televisores DLP y proyectores de video.

Como en la mayoría de las empresas dedicadas al comercio electrónico, el software juega un papel demasiado importante y está involucrado en todos los niveles de la organización. Hacia dentro de la empresa el departamento de Informática es el encargado de analizar los requerimientos provenientes de los distintos departamentos y determinar cuáles de estos son viables desarrollos de software.

1.2 Antecedentes del problema.

Mi Technologies Inc. ha mantenido un crecimiento constante en los últimos años, pasando de micro-empresa en 2009 a empresa líder en su ramo con más de 90,000 operaciones por mes en 2012.

Este crecimiento, repercute a todos los niveles de la empresa, creando nuevos puestos de trabajo e inyectando recursos en diferentes áreas, lo cual también genera nuevas necesidades y requerimientos de información, que deben ser evaluados y en caso de ser viables, se deben desarrollar como aplicaciones de software.

Existen dos desarrollos de software importantes, que crean el núcleo y la lógica de negocios de toda la empresa.

- **ERP (*Enterprise Resource Planning*).**
Se ha orientado totalmente hacia la planeación y administración de inventario, además de todos los procesos que de él derivan.
- **OM (*Order Manager*).**
Realiza desde el procesamiento de órdenes, tarjetas de crédito, retornos e intercambios hasta la logística para los procesos de fabricación y envío en cada una de las órdenes de venta autorizada.

Por su naturaleza, ambos sistemas, han presentado una dificultad histórica en cuestiones tales como capacitación, operación y costos de licenciamiento.

1.3 Definición del problema.

Los sistemas ERP-OM generan la información necesaria para los procesos de venta, inventarios, fabricación y envío, ambos sistemas son muy robustos y específicos, requieren de una exhaustiva capacitación para poder operarlos eficientemente, y muy poca personal cuenta con la capacitación necesaria, un estudio realizado por el departamento de producción y manufactura, determino que el

cincuenta por ciento del personal requiere información de ambos sistemas, a fin de poder cumplir en tiempo y forma con sus actividades diarias.

Debido a lo anterior, recolectar información en formatos impresos u hojas de cálculo, se ha convertido en una tarea común, y algunos departamentos hacen uso de esta información, la cual en la mayoría de las veces difiere mucho de la que puede ser extraída de los sistemas ERP-OM.

No existe una manera automatizada, que permita eliminar los controles manuales y generar los reportes necesarios, que permita la interacción con las bases de datos de los sistemas ERP-OM, lo cual ayudaría en gran medida a procesos tales como: determinar la orden de fabricación del día, realizar proyecciones de ventas, logística para mercancías en tránsito, comparativas de venta entre diferentes periodos, logística para importación temporal de productos al país, exportación de producto terminado, estimaciones de costos de inventario, ventas telefónicas, órdenes de compra, o los controles para el manejo de productos que son vendidos en tiendas virtuales como EBay o Amazon.

El tiempo que transcurre entre la recolección de información y el envío de indicaciones a cada puesto de trabajo, ha creado una serie de inconvenientes, debido a que la información no está disponible todo el tiempo, algunos inconvenientes que se pueden mencionar son: falta de materia prima, re-trabajo de materiales, producción fuera de tiempo, problemas en aduana, cancelación y cierre de algunos canales de venta. Todo generado en gran medida por el poco entrenamiento en el uso de estas herramientas de software.

1.4 Justificación de la solución.

Automatizar tareas rutinarias es un concepto común que permite incrementar la productividad y la eficiencia a través de la tecnología, las aplicaciones de extracción, procesamiento y asignación de información clave, a puestos específicos, mejora el desempeño de las actividades, aclara dudas y permite dar fluidez a las actividades diarias dentro de la compañía.

Desde el punto de vista estratégico, táctico y operativo, crear y utilizar herramientas de software en formato WEB representa una ventaja competitiva, cualidades como ubicuidad, usabilidad y funcionalidad, representan aspectos muy importantes en los proyectos de software internos a la empresa.

Debido a lo anterior se justifica la creación de una herramienta de software en formato web que permita la extracción y procesamiento dinámico de información operativa en la empresa Mi Technologies Inc. La cual desde este momento será denominada por las siglas EyPDIO.

1.5 Objetivo general.

Desarrollar un sistema de software en formato Web que permita la extracción y procesamiento dinámico de información operativa, que se integre en todos los departamentos de la empresa, que permita la planeación de actividades y sirva de apoyo a la toma de decisiones a nivel organizacional.

1.6 Objetivos específicos.

1. Desarrollar la interface de software en formato web que permita interactuar con la información de las bases de datos de los sistemas ERP-OM.
2. Analizar y delimitar el universo de soluciones que se pueden satisfacer con la información contenida en las bases de datos de los sistemas ERP-OM.
3. Crear procesos que aseguren la integridad de la información, la cual no debe corromperse o perder utilidad a lo largo de la vida útil del sistema EyPDIO.
4. Utilizar la estructura de jerarquía de usuarios y contraseñas del sistema ERP, para controlar el acceso y delimitar áreas en el sistema EyPDIO.
5. Desarrollar un formato estándar para desplegar resultados, el cual debe ser uniforme a todas las áreas del sistema EyPDIO.
6. Ofrecer información resumida a través de gráficos descriptivos e interactivos, que permitan realizar comparativas de estados financieros en periodos diferentes.
7. Generar información detallada que apoye a la toma de decisiones a nivel directivo.
8. Disponer de acceso al sistema EyPDIO, en todo momento bajo condiciones normales.

1.7 Alcance del trabajo.

Sistematizar requerimientos propuestos por algunos departamentos, como contramedidas a la dificultad en la obtención de información sobre ventas e inventario, estos requerimientos fueron evaluados y elaborados por gerencia en coordinación con los departamentos de Administración e Informática.

La implementación del proyecto EyPDIO, se convirtió en una adición en forma de generador de informes para las aplicaciones ERP y OM, las cuales controlan elementos vitales dentro de la empresa, tales como: Inventarios, ventas, logística, entre otras, este proyecto agilizo el proceso de consulta a información detallada de productos, ventas, órdenes de compra, costos, mediante una gestión de fácil uso y en formato web.

Se considera que para el adecuado funcionamiento del sistema, la información capturada en el programa ERP por el departamento de almacén y compras se encuentre actualizada en todo momento, al igual que la información proveniente de todos los canales de venta hacia el programa OM, lo cual garantiza además del correcto funcionamiento de los procesos, información confiable, que permita retroalimentar a la fuerza productiva y de ventas.

Capítulo 2 Marco teórico.

2.1 Introducción.

En el presente capítulo se describen los fundamentos teóricos y tecnológicos, que son la base para la creación del proyecto de software “extracción y procesamiento dinámico de información operativa en la empresa MI Technologies Inc.”.

2.2 Metodología Ágil.

La década de los 90 presentó un ambiente turbulento para el desarrollo de software, la alta necesidad de que los proyectos llegaran al éxito y obtener un producto de calidad para los clientes, generó grandes cambios en las metodologías adoptadas por los equipos para cumplir sus objetivos. Estas metodologías livianas y maniobrables se conocen colectivamente hoy en día como “metodologías ágiles”.

¿Qué es una metodología ágil?

Ágil se define como la capacidad de responder a los cambios de forma rápida y maximizando los beneficios. Existen múltiples metodologías ágiles, las cuales comparten cuatro postulados definidos en el “Manifiesto Ágil”, que son los principios sobre los que se basan estos métodos. (Palacio, Flexibilidad con Scrum, 2008).

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.

2.2.1 El manifiesto Ágil.

- **A los individuos y su interacción, por encima de los procesos y las herramientas.**
Las personas representan el principal factor de éxito de un proyecto de software. Se puede contar con un buen proceso de desarrollo, pero si el equipo falla, no se puede garantizar el éxito; sin embargo, cuando un equipo funciona, es más fácil lograr el objetivo, aunque no se tenga bien definido el proceso.
- **El software que funciona, por encima de la documentación exhaustiva.**
Un desarrollo de software sin documentación no es un proyecto completo, pero se puede seguir la regla de “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben enfocarse en lo esencial y ser cortos. Si un nuevo elemento se incorpora al equipo de desarrollo, los elementos que le van a servir para ponerse al corriente son el propio código y la interacción con el equipo.
- **La colaboración con el cliente, por encima de la negociación contractual.**
Los plazos y costos preestablecidos en contratos son muy difíciles de determinar, intentar controlar todas las variables que pueden afectar al proyecto de desarrollo desde un inicio puede llevarlo al fracaso, es por ello que se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- **La respuesta al cambio, por encima del seguimiento de un plan.**
La capacidad de responder a los cambios que puedan surgir a lo largo del proyecto (nuevos requerimientos, cambio de tecnología), aumenta el grado de éxito del mismo. Por lo tanto

contar con una planeación flexible, ciclos cortos de desarrollo, permite adaptar a nuevas características o requerimientos sin realizar demasiados cambios a la planeación global.

Los valores anteriores inspiran los doce principios del manifiesto. Son características que diferencian un proceso ágil de uno tradicional. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto metas a seguir y organización del mismo. Los principios son:

- I. Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
- II. Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se doblan al cambio.
- III. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.
- IV. Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
- V. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
- VI. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
- VII. El software que funciona es la principal medida del progreso.
- VIII. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
- IX. La atención continua a la excelencia técnica enaltece la agilidad.
- X. La simplicidad como arte de maximizar la cantidad de trabajo que se hace, es esencial.
- XI. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se auto-organizan.
- XII. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia (Cunningham, 2001).

Existe una gran diferencia entre las metodologías ágiles y las tradicionales (tabla 2-1), la importancia de una metodología robusta que ajustada en un equipo cumpla con sus metas, y satisfaga más allá de las necesidades definidas al inicio del proyecto.

El éxito del producto EyPDIO, depende en gran parte de la metodología seleccionada, experiencias previas basadas en metodologías tradicionales han llegado a buen término, pero los cambios que ha sufrido la empresa, obligan a depender de metodologías más dinámicas, con resultados en periodos cortos, es ahí, donde la selección de una metodología ágil, con las características que la distinguen, permite que los equipos maximicen su potencial, aumenten la calidad del producto con los recursos y tiempos establecidos.

Tabla 2-1 Diferencias entre metodologías ágiles y no ágiles.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

2.3 Scrum.

Scrum es una metodología ágil de desarrollo de proyectos que toma su nombre y principios de los estudios realizados sobre nuevas prácticas de producción por *Hirofuma Takeuchi e Ikujiro Nonaka* a mediados de los 80.

Aunque surgió como modelo en el desarrollo de productos tecnológicos, sus principios son válidos para entornos que trabajan con requisitos inestables, y necesitan agilidad: situaciones frecuentes en el desarrollo de determinados sistemas de software.

Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro, porque la gestión no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. (Palacio, Flexibilidad con Scrum, 2008).

Scrum se basa en los siguientes principios ágiles:

- Privilegiar el valor de la gente sobre el valor de los procesos.
- Entregar software funcional lo más pronto posible.
- Predisposición y respuesta al cambio.
- Fortalecer la comunicación y la colaboración.
- Comunicación verbal directa entre los implicados en el proyecto.
- Simplicidad; supresión de artefactos innecesarios en la gestión del proyecto.

Esencia de scrum. (Palacio, Flexibilidad con Scrum, 2008).

- Más que una metodología de desarrollo es para gestionar proyectos, no contiene definiciones en áreas de ingeniería.
- Con visión de que el trabajo es efectuado por equipos auto-organizados y auto-dirigidos, logrando motivación, responsabilidad y compromiso.
- Está basada en un proceso constructivo iterativo e incremental donde las iteraciones tienen duración fija.
- Contiene definición de roles, prácticas y productos de trabajo escritos de forma simple.
- Está soportada en un conjunto de valores y principios.

2.3.1 Responsabilidades y roles de scrum.

Scrum tiene una estructura muy simple. Todas las responsabilidades del proyecto se reparten en 3 roles: (Palacio & Ruata, Scrum Manager Proyectos - Apuntes de formación, 2009).

- **Propietario del producto.**

Es quien decide en última instancia cómo será el resultado final, y el orden en el que se van construyendo los sucesivos incrementos: qué se pone y qué se quita de la pila del producto, y cuáles la prioridad de las funcionalidades.

Es responsable de la financiación del proyecto, y las decisiones sobre fechas y funcionalidades de las diferentes versiones del producto, y el retorno de la inversión del proyecto.

En los desarrollos internos para la propia empresa, suele asumir este rol el product manager o el responsable de marketing. En desarrollos para clientes externos: el responsable del proceso de adquisición del cliente.

- **El equipo.**

Se recomienda un tamaño de equipo entre 4 y 8 personas. Más allá de 8 resulta más difícil mantener la agilidad en la comunicación directa, y se manifiestan con más intensidad las rigideces habituales de la dinámica de grupos (que comienzan a aparecer a partir de 6 personas). No se trata de un grupo de trabajo formado por un arquitecto, diseñador o analista, programadores, pruebas.

Es un equipo multidisciplinario, en el que todos trabajan de forma conjunta para realizar cada sprint. Las principales responsabilidades, más allá de la auto-organización y uso de tecnologías ágiles son las que se derivan de la diferencia entre “grupo de trabajo” y “equipo”.

- **Scrum Manager (Líder del equipo).**

Es el responsable del funcionamiento de Scrum en el proyecto, cubriendo los aspectos siguientes que la organización necesite según el conocimiento, experiencia con el modelo, o aquellos que no cubra con otras personas con la formación e idoneidad adecuada.

2.3.2 Elementos de scrum.

Este tema describe estos tres elementos. Los dos primeros forman los requisitos del sistema, y el tercero es valor que se le entrega al cliente al final de cada sprint. (Palacio & Ruata, Scrum Manager Proyectos - Apuntes de formación, 2009).

- **Pila del producto (Product Backlog).**

La pila del producto es el inventario de funcionalidades, mejoras, tecnología y corrección de errores que deben incorporarse al producto a través de las sucesivas iteraciones de desarrollo. Representa todo aquello que esperan los clientes, usuarios, y en general los interesados. Todo lo que suponga un trabajo que debe realizar el equipo tiene que estar reflejado en esta pila.

- **Pila Del Sprint (Sprint Backlog).**

La pila del sprint, es la lista que descompone las funcionalidades de la pila del producto en las tareas necesarias para construir un incremento: una parte completa y operativa del producto. Cada tarea de la pila del sprint tiene asignada una persona, y la indicación del tiempo que aún falta para terminarla. Es útil porque descompone el proyecto en unidades de tamaño adecuado para determinar el avance a diario, e identificar riesgos y problemas sin necesidad de procesos complejos de gestión. Es también una herramienta de soporte para la comunicación directa del equipo.

- **Incremento.**

El incremento es la parte de producto producida en un sprint, y tiene como características: que está completamente terminada y operativa, en condiciones de ser entregada al cliente final.

No se trata por tanto de módulos o partes a falta de pruebas, o documentación o, Idealmente en el desarrollo ágil:

- Cada funcionalidad de la pila del producto se refiere a funcionalidades entregables, no a trabajos internas del tipo “diseño de la base de datos”.
- Se produce un “incremento” en cada iteración.

2.3.3 Sprint.

Un Sprint es el periodo de tiempo durante el que se desarrolla un incremento de funcionalidad (figura 2-1). Constituye el núcleo de Scrum, que divide de esta forma el desarrollo de un proyecto en un conjunto de pequeñas “carreras”. (Kniberg & Skaring, 2010) (Palacio, ScrumManager: Gestión de proyectos , 2008).

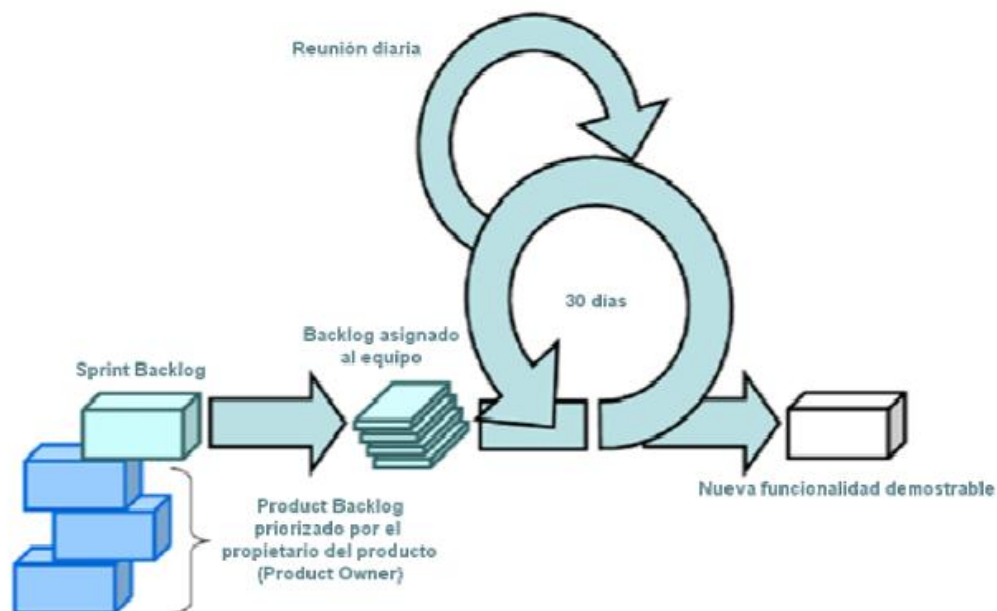


Figura 2-1 Fases de desarrollo de un Sprint.

- Duración máxima del Sprint: 30 días.
- Durante el Sprint no se puede modificar el trabajo que se ha acordado en el Backlog.
- Sólo es posible cambiar el curso de un Sprint, abortándolo, y sólo lo puede hacer el Scrum Master si decide que no es viable por alguna de las razones siguientes:
 - La tecnología acordada no funciona.
 - Las circunstancias del negocio han cambiado.
 - El equipo ha tenido interferencias.

Planificación del Sprint.

Se planifica en detalle el trabajo al inicio de cada Sprint asumiendo que los objetivos no van a cambiar durante el mismo. De esta manera se atenúa el riesgo.

Pre-condiciones:

- ✓ La organización tiene asignados los recursos para llevar a cabo el sprint.
- ✓ El propietario del producto tiene preparada la pila del producto, con su criterio de prioridad para el negocio, y un número suficiente de elementos para desarrollar en el sprint.
- ✓ Siempre que sea posible, el propietario del producto debe haber trabajado antes con el equipo. De esta forma su estimación previa del trabajo que se puede realizar en el sprint será bastante ajustada.
- ✓ El equipo tiene un conocimiento de las tecnologías empleadas, y del negocio del producto suficiente para realizar estimaciones basadas en "juicio de expertos", y para comprender los conceptos del negocio que expone el propietario del producto.

Entradas:

- ✓ La pila del producto.
- ✓ El producto desarrollado hasta la fecha a través de los sucesivos incrementos (excepto si se trata del primer sprint).
- ✓ Circunstancias de las condiciones de negocio del cliente y del escenario tecnológico empleado.

Salidas:

- ✓ Pila del sprint.
- ✓ Duración del sprint y fecha de la reunión de revisión.
- ✓ Objetivo del sprint.

Seguimiento del Sprint.

Reunión diaria breve, de no más de 15 minutos, en la que cada miembro del equipo dice las tareas en las que está trabajando, si se ha encontrado o prevé encontrarse con algún impedimento, y actualiza sobre la pila del sprint las ya terminadas, o los tiempos de trabajo que les quedan.

Entradas:

Pila del sprint y gráfico de avance, actualizados con la información de la reunión anterior. Información de las tareas realizadas por cada componente del equipo.

Salidas:

Pila del sprint y gráfico de avance actualizados. Identificación de necesidades e impedimentos.

Revisión del Sprint.

Reunión realizada al final del sprint en la que, con una duración máxima de 4 horas, el equipo presenta al propietario del producto, clientes, usuarios, gestores, el incremento construido en el sprint.

Objetivo:

- ✓ El propietario del producto obtiene información objetiva del progreso del sistema. Esta reunión marca a intervalos regulares, el ritmo de construcción del sistema y la trayectoria que va tomando la visión del producto.
- ✓ Al ver y probar el incremento, el propietario del producto, y el equipo en general obtienen retroalimentación clave, para evolucionar y dar más valor a la pila del producto.
- ✓ Otros ingenieros y programadores de la empresa también pueden asistir para conocer cómo trabaja la tecnología empleada.
- ✓ El administrador del Scrum obtiene retro-información sobre buenas prácticas y problemas durante el sprint, necesaria para las prácticas de ingeniería de procesos y mejora continua de la implementación.

Estradas:

- ✓ Incremento terminado.

Salidas:

- ✓ Retroalimentación para el propietario del producto: hito de seguimiento de la construcción del sistema, e información para mejorar el valor de la visión del producto.
- ✓ Retroalimentación para el Scrum Manager (o responsable de la gestión de procesos de la organización): buenas prácticas y problemas durante el sprint.
- ✓ Convocatoria de la reunión del siguiente Sprint.

2.3.4 Valores.

- **Foco.**
Los individuos y sus interacciones son más importantes que el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto de software.
- **Comunicación.**
Scrum pone en comunicación directa y continua a clientes y desarrolladores. El cliente se integra en el equipo para establecer prioridades y resolver dudas. De esta forma ve el avance día a día, y es posible ajustar la agenda y las funcionalidades de forma consecuente.
- **Respeto.**
(Palacio & Ruata, Scrum Manager Proyectos - Apuntes de formación, 2009), diferencia claramente entre dos grupos para garantizar que quienes tienen la responsabilidad tienen también la autoridad necesaria para poder lograr el éxito, y que quienes no tienen la responsabilidad, los observadores externos, no produzcan interferencias innecesarias.
- **Coraje.**
El coraje implica saber tomar decisiones difíciles. Reparar errores cuando se detectan. Mejorar el código siempre que la retroalimentación y las sucesivas iteraciones se manifiesten susceptibles de mejora.

2.4 MVC (Modelo Vista Controlador).

(Myer, 2008) El modelo-vista-controlador, es un nombre que se aplica al diseño basado en patrones que permite a los desarrolladores separar limpiamente su código en tres categorías.

- **Modelo.**
Es el objeto que maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.
- **Vista.**
Es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.
- **Controlador.**
Es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Se ha seleccionado el patrón de desarrollo MVC como estructura para el proyecto EyPDIO, las razones para esta selección se basan en el conjunto de ventajas que representa la separación de los elementos principales (Control, Vista, Modelo), lo cual permite involucrar personal calificado en áreas como bases de datos, o diseño web, involucrase en el proyecto, sin afectar la lógica de negocios.

También es necesario comentar que EyPDIO es la primera aplicación en la empresa MI Technologies Inc. en la que el patrón MVC es incluido, y representa un gran reto explotar todas la ventajas.

Debido a MVC separa en tres partes, los desarrolladores pueden crear múltiples vistas y los controladores para cualquier determinado modelo sin forzar los cambios en el diseño del modelo. Esta separación permite un fácil mantenimiento, además de permitir portar y organizar aplicaciones que no se parezcan en nada a este nuevo modelo, la figura 2-2 muestra la relación que existe entre las partes que conforman el patrón de desarrollo MVC.

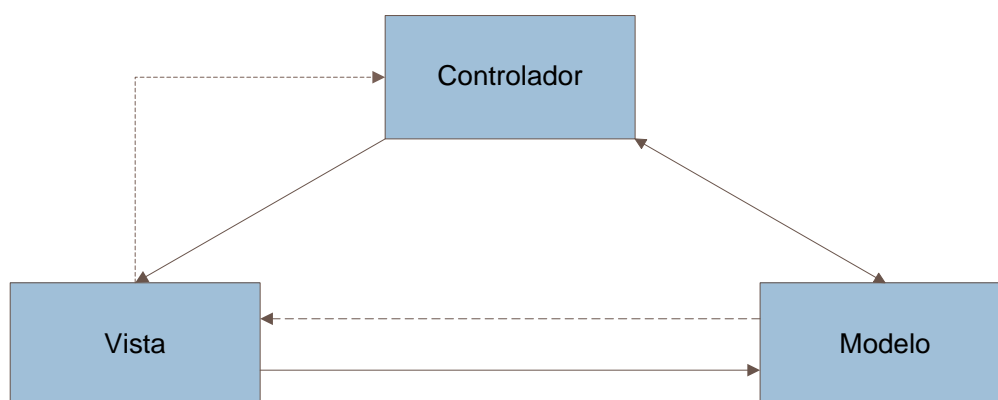


Figura 2-2 Modelo Vista Controlador.

2.4.1 Ventajas de MVC (Pantoja).

- La aplicación está implementada modularmente.
- Sus vistas muestran información actualizada siempre.
- El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
- Si se desea hacer una modificación al modelo del dominio, como aumentar métodos o datos contenidos, sólo debe modificarse el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
- Las modificaciones a las vistas no afectan en absoluto a los otros módulos de la aplicación.
- MVC es bastante utilizado en la actualidad en marcos de aplicación orientados a objeto desarrollados para construir aplicaciones de gran tamaño; Java Swing, Apache Struts, Microsoft ASP.NET, las transformaciones XSL o incluso los documentos LATEX siguen este patrón de diseño.
- MVC está demostrando ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

2.4.2 Desventajas de MVC (Pantoja).

- El tiempo de desarrollo de una aplicación que implementa el patrón de diseño MVC es mayor, al menos en la primera etapa, que el tiempo de desarrollo de una aplicación que no lo implementa, ya que MVC requiere que el programador implemente una mayor cantidad de clases que en un entorno de desarrollo común no son necesarias. Sin embargo, esta desventaja es muy relativa ya que posteriormente, en la etapa de mantenimiento de la aplicación, una aplicación MVC es muchísimo más mantenible, extensible y modificable que una aplicación que no lo implementa.
- MVC requiere la existencia de una arquitectura inicial sobre la que se deben construir clases e interfaces para modificar y comunicar los módulos de una aplicación.
- Esta arquitectura inicial debe incluir, por lo menos: un mecanismo de eventos para poder proporcionar las notificaciones que genera el modelo de aplicación; una clase Modelo, otra clase Vista y una clase Controlador genéricas que realicen todas las tareas de comunicación, notificación y actualización que serán luego transparentes para el desarrollo de la aplicación.
- MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.

2.5 PHP.

(Weeling & Thomson, 2005) nos comenta en su libro que PHP es un lenguaje de secuencia de comandos que trabaja del lado del servidor, está diseñado específicamente para la uso en la Web aunque también se pueden ejecutar programas en línea de comandos. Dentro de una página Web se puede incrustar código PHP que será ejecutado cada vez que la página sea vista en un navegador. El código PHP es interpretado en el servidor y genera código HTML además de otros contenidos que es lo que el visitante ve en su navegador al visitar la página.

PHP fue concebido en 1994 y es fruto del trabajo de un hombre, Rasmus Ledorf. Ha sido adaptado y mejorado a lo largo del tiempo, ha experimentado importantes transformaciones hasta convertirse en el producto actual.

PHP es un producto de código abierto, lo que quiere decir que puede acceder a su código. Puede utilizarlo, modificarlo y redistribuirlo si coste alguno.

Las siglas PHP equivalían originalmente a Personal Home Page (Página de inicio personal) pero se modificaron de acuerdo con la convención de designación de GNU (GNU is not UNIX) y ahora equivalen a PHP Hipertext Preprocessor (Preprocesador de hipertexto PHP). En la actualidad, PHP esta en su versión 5. Esta versión incorpora mejoras importantes en lo que respecta al motor Zend subyacente y al lenguaje que utiliza.

¿ Porque usar PHP?

La razón más importante del porque utilizar PHP es porque con él se puede hacer cualquier cosa que se pueda realizar con un script (CGI). El proyecto EyPDIO requiere de un alto rendimiento, en acceso a información y ejecución de tareas.

Existen otros lenguajes en el mercado que compiten directamente cona PHP se puede citar a Perl, Microsoft Active Server Pages (ASP), Java Server Pages(JSP) y Allaire ColdFusion.

Mi Technologies Inc. ha basado la mayoría de sus desarrollos web en el lenguaje de programación PHP, unque no se descartan algunos otros lenguajes de programación. El motivo por el cual se ha seleccionado este lenguaje, es porque se conocen sus características, potencia, y lo mas importante se cuenta con el personal capacitado y con experiencia en este lenguaje de programación.

En comparación en estos productos, PHP cuenta con más ventajas que desventajas.

2.5.1 Ventajas de PHP.

- **Licenciamiento.**
PHP Como todo en el código abierto y software libre está garantizado por un conjunto de esquemas de licenciamiento, el más famoso es GPL (GNU General Public License) o copyleft. PHP utiliza tanto en la licencia GPL como en su propia licencia y cada usuario tiene la libertad de elegir cualquiera de ellas.
- **Alto rendimiento.**
PHP es muy eficaz. Mediante el uso de un único servidor, puede servir millones de accesos al día. Los comparativos de rendimiento publicados que publica Zend Technologies (<http://www.zend.com>), dejan clara muestra que PHP supera ampliamente a todos sus competidores en esta faceta.
- **Interfaces para diferentes sistemas de base de datos.**
PHP dispone de modulos de conexión propios a todos los sistemas de base de datos. Además de MySQL, puede conectarse también a lass bases de datos de PostgreSQL, mSQL, Oracle, dbm,FilePro, Hiperwave, Informix, InterBase y Sybase, entre otras. PHP5 cuenta también con una interfaz SQL incorporada en formato de archivo plano, denominada SQLite.

- **Bibliotecas incorporadas para muchas tareas Web habituales.**

PHP ha sido diseñado y desarrollado para uso en la Web, incorpora una gran cantidad de funciones integradas para realizar tareas útiles en el web, puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, manejar sesiones, cookies y generar documentos PDF, todo esto con unas cuantas líneas de código .

- **Bajo coste.**

PHP es gratuito. Se puede descargar la última versión de <http://www.php.net>.

- **Facilidad de aprendizaje y uso.**

La curva de aprendizaje para PHP es muy corta si se cuenta con alguna experiencia en lenguajes de programación como C o Perl o algún lenguaje tipo C, C++ o Java.

- **Compatibilidad con el enfoque orientado a objetos.**

2.5.2 Desventajas de PHP.

- **Lenguaje Interpretado.**

PHP es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. El aspecto técnico del lenguaje puede dificultar la lectura del código pero no la impide. Las páginas desarrolladas en PHP pueden resultar en algunas ocasiones un poco lento debido al hecho de que es un lenguaje interpretado.

- **Definición Lenguaje Orientado a Objetos.**

PHP Soporta objetos, sin embargo debido a que es obligatorio el uso de scripts para recibir peticiones de formularios la definición de lenguaje orientado a objetos se puede tornar un tanto ambigua.

- **Tipificación.**

PHP es un lenguaje no tipificado, lo cual dificulta a los diferentes IDEs la asistencia de escritura de código, para lo cual Zend Studio añade un comentario con el tipo a la declaración de la variable.

2.6 Bases de datos.

Las bases de datos sirven para modelar problemas reales y administrar datos dinámicamente, es por esto, que pueden ser manejadas como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

La implementación del proyecto EyPDIO, requiere de un constante acceso la información de las bases de datos de los sistemas ERP-OM, las cuales tienen una estructura y organización diseñada específicamente para su propósito, ambas están basadas en el modelo de bases de datos relacionales.

Entender los conceptos, la estructura y el medio de interacción con los datos es vital en el éxito del proyecto.

2.6.1 Bases de datos relacionales.

“Se basan en el uso de tablas (también se las llama relaciones). Las tablas se representan gráficamente como una estructura rectangular formada por filas y columnas (figura 2-3). Cada columna almacena información sobre una propiedad determinada de la tabla (se le llama también atributo).

Cada fila posee una ocurrencia o ejemplar de la instancia o relación representada por la tabla (a las filas se las llama también tuplas)”, (Sanchez, 2004).

Nombre de la tabla:

Atributo1	Atributo2	Atributo3	...	Atributo n	
Valor 1,1	Valor 1,2	Valor 1,3	...	Valor 1,n	Tupla 1
Valor 2,1	Valor 2,2	Valor 2,3	...	Valor 2,n	Tupla 2
...	
Valor m,1	Valor m,2	Valor m,3	...	Valor m,n	Tupla m

Figura 2-3 Estructura de una tabla relacional

Objetivo de las bases de datos relacionales:

- **Independencia física.** La forma de almacenar los datos, no debe influir en su manipulación lógica.
- **Independencia lógica.** Las aplicaciones que utilizan la base de datos no deben ser modificadas por que se modifiquen elementos de la base de datos.
- **Flexibilidad.** La base de datos ofrece fácilmente distintas vistas en función de los usuarios y aplicaciones.
- **Uniformidad.** Las estructuras lógicas siempre tienen una única forma conceptual (las tablas).
- **Sencillez.**

2.6.2 Lenguaje de consulta en base de datos.

El conocimiento del lenguaje de consulta integrado en las bases de datos de los sistemas ERP-OM, es otro de los requerimientos vitales para el logro del objetivo, debido a que este representa el medio de comunicación entre el sistema EYPDIO y la información.

SQL.

Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre la misma.

Características generales.

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones sobre los mismos.

Es un lenguaje declarativo de alto nivel o de no procedimiento, que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación. De esta forma una sola sentencia puede equivaler a uno o más sentencias de código en un lenguaje de bajo nivel.

Lenguaje de definición de datos (LDD).

Es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Existen cuatro operaciones básicas: CREATE, ALTER, DROP y TRUNCATE.

Lenguaje de manipulación de datos (LMD).

Es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

2.7 Herramientas web.

En este tema se describen las herramientas más comunes en el desarrollo de aplicaciones web desde el punto de vista de la interface de usuarios, el modelo MVC (mencionado anteriormente), permite hacer una separación en tres capas, dos de ellas son internas a la aplicación (Control, Modelo), y otra es la que está expuesta al usuario (Vista), esta última es la presentación del sistema, también sirve de interface con el usuario, y realiza la primer filtro de datos ingresados al sistema.

Existen muchas herramientas destinadas al diseño web, en esta sección se hace una selección de las necesarias para el sistema EyPDIO, el cual además de funcional se requiere que tenga una presentación sencilla, agradable a la vista y funcional a la vez.

2.7.1 HTML.

“El HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). Fue creado en 1986 por el físico nuclear Tim Berners-Lee; el cual tomo dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) el cual permite conectar dos elementos entre si y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas.” (Gauchat, 2012).

¿Para qué sirve?

Html es un lenguaje de marcación de elementos para la creación de documentos hipertexto, muy fácil de aprender, lo que permite que cualquier persona sin experiencia en programación, lo pueda utilizar y enfrentarse a la tarea de crear una página web.

¿Cómo funciona?

HTML no es un lenguaje de programación si no que son sentencias -etiquetas- las cuales indican operaciones a realizar con el texto o con los atributos de esta, estas secuencias son necesarias para el diseño de una página web ya que cada una le indica al navegador la forma en que deben presentar el contenido de la página web.

2.7.2 Hojas de estilo en cascada (CSS).

Consiste en un conjunto de mecanismos simples que permiten especificar la manera en que un documento se presenta en la pantalla, o como se va a imprimir. Esta forma de descripción permite un control total sobre el estilo y la forma de los documentos.

¿Para qué sirve?

“CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.” (w3c, 2012).

¿Cómo funciona?

“CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.” (w3c, 2012).

2.7.3 JavaScript.

Javascript es un lenguaje de programación que permite a los desarrolladores crear acciones en páginas web. Es un lenguaje que puede ser utilizado por profesionales y también quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

Muchos confunden el Javascript con el Java pero ambos lenguajes son diferentes y tienes sus características singulares.

Javascript tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado. Java por su parte tiene como principal característica ser un lenguaje independiente de la plataforma. Se puede crear todo tipo de programa que puede ser ejecutado en cualquier ordenador del mercado: Linux, Windows, Apple, etc. Debido a sus características también es muy utilizado para internet. (Pérez, 2009).

¿Para qué sirve?

JavaScript sirve principalmente para mejorar la gestión de la interfaz cliente/servidor. Un script JavaScript insertado en un documento HTML permite reconocer y tratar localmente, es decir, en el cliente, los eventos generados por el usuario. Estos eventos pueden ser el recorrido del propio documento HTML o la gestión de un formulario.

¿Cómo funciona?

Los clientes Web que soportan JavaScript, pueden interpretar sentencias JavaScript colocadas en un documento HTML. Cuando el cliente Web solicita una página de este tipo, el servidor envía por la red al cliente el contenido completo del documento, incluyendo todos los códigos HTML y las sentencias JavaScript que pudieran existir en éste.

El cliente lee entonces la página de forma secuencial desde el principio hasta el final, representando visualmente los códigos HTML y ejecutando las sentencias JavaScript conforme avanza el proceso de lectura e interpretación.

Las sentencias JavaScript colocadas en una página Web pueden dar respuesta a eventos de usuario, tales como la pulsación de un botón del ratón (clic), la entrada de datos en un formulario y la navegación por una página.

Capítulo 3 Metodología.

3.1 Introducción.

Este capítulo desarrolla el proceso que sirve de complemento a la metodología scrum, funciona como marco de trabajo y evidencia en el desarrollo del sistema EyPDIO.

El proceso Ágil, iterativo que se propone, se refuerza con el tradicional el ciclo de vida de un producto de software, como medida para facilitar la gestión y control del proyecto.

3.2 Etapas del ciclo de vida de un producto.

1. Planeación.
2. Análisis.
3. Diseño.
4. Desarrollo y pruebas.
5. Implementación.

3.3 Planeación.

En esta etapa, se determinan los factores generales que definen al proyecto y sus requisitos.

Actividades:

- Definir el proyecto.
- Estimar de alcance del proyecto.
- Establecer la pila del producto.
- Integrar del equipo de trabajo.
- Estimar el costo del proyecto.
- Fijar el calendario de actividades.
- Establecer el plan de manejo de riesgos.
- Establecer el protocolo de entrega.

Entregable: Plan del proyecto.

El documento realizado, solo es un método formal utilizado como guía para la ejecución y control del proyecto, pero no implica ninguna responsabilidad con la empresa mientras no sea aprobado por los departamentos de administración y gerencia.

3.4 Análisis.

En esta etapa, se desarrollan y detallan, las especificaciones que permitan satisfacer las necesidades de información de los usuarios, además sirve de base para secciones subsecuentes como el diseño y la construcción del sistema.

Actividades:

- Desarrollar de los requerimientos funcionales (Casos de uso).
- Determinar los requerimientos no funcionales.
- Detectar otros requerimientos no funcionales.

Entregable: Análisis de requerimientos.

3.5 Arquitectura y diseño.

En esta etapa, se desarrollan los modelos o especificaciones para el producto o componentes del sistema, basándose en los datos recolectados en la etapa de análisis. El modelo creado en esta etapa debe estar preparado para contemplar modificaciones, crecimiento e incorporación de nuevas funcionalidades.

Actividades:

- Diseñar la vista general de la arquitectura del Software.
 - Realizar el diagrama de Arquitectura de bloques.
 - Realizar el diagrama de arquitectura lógica.
 - Realizar el diagrama de arquitectura física.
 - Realizar el diagrama de componentes.
- Realizar el diseño detallado del sistema.
 - Desarrollo de las vistas de los caso de uso.

Entregable: Arquitectura y diseño.

3.6 Desarrollo y pruebas.

En esta etapa, se utilizan los modelos creados durante la etapa de diseño para dar solución a la iteración o "Sprint", cumpliendo con las especificaciones y requerimientos definidos en los documentos de la etapa de análisis.

Actividades:

- Implementación de la capa de datos, estructura, funciones y procedimientos.
- Implementación de la lógica de negocio o control.
- Implementación de la interface de usuario.
- Integración de sistema y pruebas.
- Revisión y retroalimentación.

Entregable: Entrega del Sprint en funcionamiento.

3.7 Puesta en marcha.

En esta etapa, el producto es puesto a disposición de la empresa.

Actividades:

- Poner en marcha el sistema en un ambiente de producción.
- Capacitar al personal.
- Realizar todas las actividades extra para incorporar el sistema en un ambiente de producción.

Entregables: El sistema en producción con manuales operativos y mantenimiento de procedimientos.

3.8 Herramientas a utilizar.

A continuación, se detallan algunos recursos que se utilizan para realizar y reforzar funciones o trabajos específicos.

Técnicas de relevamiento.

Entrevistas con el cliente y los usuarios; revisión de registros, y observación.

Estructura de división de trabajo (EDT).

Es un organigrama que despliega la estructura de un proyecto y muestra su organización por fases y niveles de detalle (figura 3-1). Cada nivel de descenso representa un aumento en el nivel de detalle de las descripciones de las actividades, sirve de lista de comprobación, y determina el alcance general. Así como también, define los entregables del proyecto. Los entregables pueden ser etapas o procesos del proyecto (plan del proyecto, documentación de diseño, etc.) o partes del producto final (pantallas, ventanas, documentación).

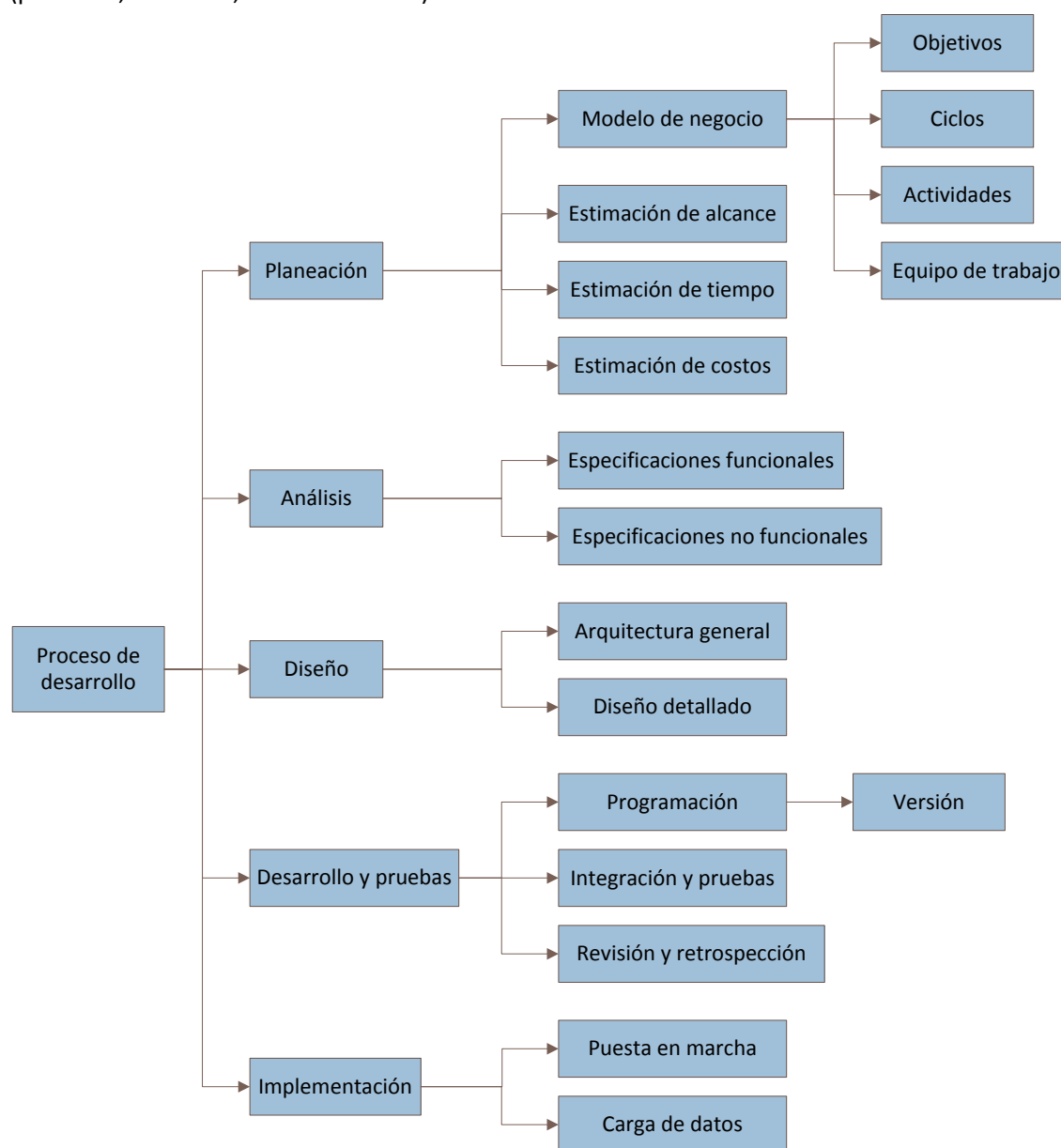


Figura 3-1 EDT completo para el sistema EyPDIO

Tanto el EDT como las técnicas de relevamiento mencionadas anteriormente se utilizan en las dos primeras etapas, o sea para la Planificación y el Análisis.

Casos de uso

Son un método práctico para explorar requerimientos. Ayudan a describir qué es lo que el sistema debe hacer desde el punto de vista del usuario.

Por lo tanto, consideramos que los casos de uso proporcionan un modo claro y preciso de comunicación con el cliente.

Descripciones de casos de uso: detallan los casos de uso, en ellas se explica la forma en cómo interactúa el usuario con el sistema de software.

Tanto los casos de uso como las descripciones de los mismos se utilizan en la etapa del análisis para definir los requisitos.

Capítulo 4 **Desarrollo.**

4.1 Introducción.

Esta sección, describe el proceso de desarrollo del sistema EyPDIO, basándose totalmente en la metodología descrita en el capítulo anterior.

4.2 Planeación.

Esta etapa, permito dar conocer al equipo de trabajo, la situación de la empresa al momento de decidir la creación del sistema EyPDIO. En esta etapa, también se definieron los objetivos, se plantearon las estrategias y cursos de acción necesarios para alcanzarlos, además sirvió de base a las etapas posteriores, permitió detectar oportunidades y amenazas, así como fortalezas y debilidades.

4.2.1 Definición del proyecto.

El objetivo del proyecto así como los objetivos específicos, junto con la justificación de la solución se han definido en el capítulo introducción de este documento, por lo tanto no se incluirán en esta sección.

4.2.2 Estimación del alcance del proyecto (EDT).

La información en un principio era muy limitada, las ideas sobre lo que el sistema debía realizar no eran muy claras. Las reuniones constantes entre los departamentos de administración e informática permitieron definir la estructura jerárquica con los procesos requeridos por los interesados en el producto (figura 4-1). Esta estructura de división de trabajo, definió los límites del sistema y planteó la perspectiva del alcance del proyecto.

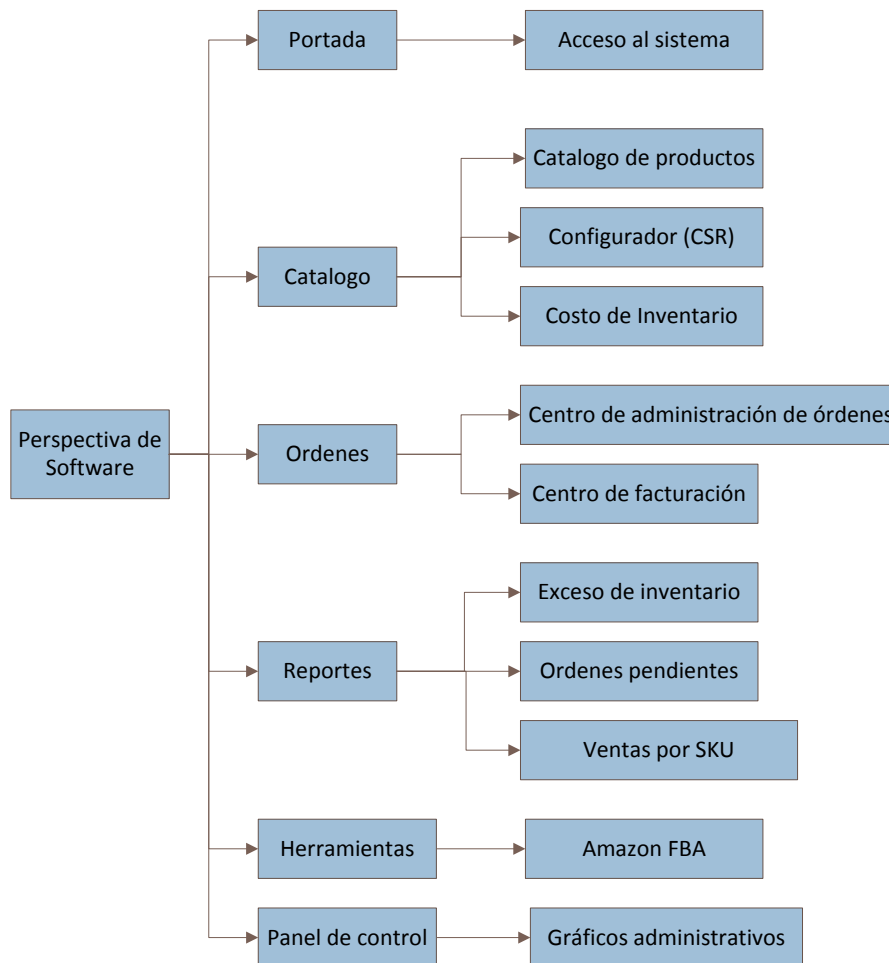


Figura 4-1 Estructura de división, perspectiva del producto.

Plazo de entrega.

Estimar la duración de un proyecto de software no es una tarea sencilla, un porcentaje muy alto de todas las implementaciones de software exceden el tiempo estimado que se ha definido para su desarrollo.

Tomando como base cada una de las fases establecidas en el EDT, “Perspectiva de producto”, y a la experiencia del personal involucrado en el proyecto, se logró establecer que procesos se integrarían en cada fase, la experiencia en desarrollo de software del personal del departamento de informática, permitió determinar el esfuerzo en horas para cada uno de esos procesos, y se estimó que la duración del proyecto EyPDIO sería de 151 horas (tabla 4-1). Esta estimación es poco real, pero planteó un panorama a cerca de la cantidad de recursos y tiempo que se necesitarían incorporar en el desarrollo del proyecto.

Tabla 4-1 Estimación de la duración del proyecto.

Fase en el EDT	Cantidad de procesos	Esfuerzo (hs)	Subtotal (hs)
Acceso al sistema	1	5	5
Catálogo de productos	14	3	42
Configurador (CSR)	1	4	4
Costo de inventario	1	6	6
Centro de administración de órdenes	4	7	28
Centro de facturación	3	6	18
Exceso de inventario	1	5	5
Órdenes pendientes	1	2	2
Ventas por producto	1	2	2
Amazon FBA	3	3	9
Panel de control	10	3	30
Total			151

4.2.3 Pila del producto.

Cada fase establecida en el EDT integra una serie de procesos, los cuales fueron identificados en base a la experiencia del personal involucrado, y documentados como requerimientos al sistema (tabla 4-2).

Esta lista priorizada de requerimientos de usuario se incorporara al producto a través de las iteraciones sucesivas, representa todo aquello que esperan los usuarios y los interesados en el producto, todo lo que aporte valor al usuario, en términos del producto final está representado en esta lista.

Cada uno de estos requerimientos serán analizado posteriormente para determinar cuáles eran realmente necesarios, cuales se debían posponer y cuales se debían eliminar.

Tabla 4-2 Lista detallada de requerimientos.

Lista detallada de requerimientos
Acceso al sistema
Consulta al catálogo de productos
Datos complementarios al catálogo de productos
Información detallada del de producto
Información de campos personalizados
Compatibilidades
Imágenes del producto
Documentos relacionados al producto
Historial de movimientos
Detalle de Inventario
Requerimientos de Ensamble
Gráficos estadísticos del producto
Información del vendedor
Configurador
Costo del inventario
Centro de manejo de órdenes
Información de envío
Detalle de la orden
Centro de Facturación
Información de envío
Detalle de la orden
Consolidación de estados por tienda virtual
Exceso de Inventario
Ordenes Pendientes
Ventas por producto
Costos unitarios por producto
Costos Totales por producto
Productos en Amazon FBA
Ordenes Pendientes Amazon FBA
Ordenes Completas Amazon FBA
Ventas totales por año de lámparas divididas por mes
Ventas totales por año de lámparas de proyector divididas por mes
Ventas diarias CSR en los últimos 30 días.
Total de ventas anuales.
Ventas totales por canal de venta.
Artículos y órdenes vendidos y enviados por canal de venta.
Diez productos más vendidos.
Total de ventas por agente de venta a mayoreo.
Total de ventas por agente de venta en México.
Total de artículos y órdenes vendidos por agente de venta en México.

4.2.4 Integración del equipo de trabajo.

La información recabada hasta este momento, permitió identificar la cantidad de los recursos humanos necesarios para el desarrollo del sistema, el cual debería estar conformado por tres personas, donde cada una de ellas, debía cubrir a menos dos roles en los procesos de administración de proyectos específicos (tabla 4-3).

Las características del proyecto, dieron la pauta para establecer el perfil profesional que debían tener los integrantes del equipo de trabajo:

- Tener una forma de trabajo sencilla de entender y aceptar por el equipo.
- Trabajar de forma eficiente: capacidad de medir el tiempo de desarrollo y tomar decisiones para mejorarlo.
- Cubrir varios roles a la vez.

Tabla 4-3 Equipo de trabajo.

Rol	Responsabilidad	Cantidad
Responsable de Desarrollo y Mantenimiento de Software	Administrar el proyecto y ambiente de trabajo	1
Analista	Analizar los requerimientos y obtener el modelo conceptual.	1
Administrador de bases de datos	Administración de bases de datos SQL SERVER	1
Arquitecto	Construir la arquitectura del sistema, de alto y bajo nivel	1
Programador	Realizar la programación de los componentes reusables y la adecuación de los existentes para el sistema a construir	1
Responsable de Pruebas	Realizar las pruebas de usabilidad y de integración	1
Documentador	Redactar los manuales de usuario y de operación software.	1

4.2.5 Costo estimado.

El análisis realizado por medio de la información recaudada, permitió también, determinar la cantidad de recursos monetarios, que debía erogar la empresa para desarrollar el proyecto de software. La compañía prefirió asignar el proyecto al departamento de Informática, donde únicamente dos personas, serán los responsables de cubrir todos los roles establecidos para el equipo de trabajo, la erogación se incorporó al sueldo que recibe cada uno de los involucrados. Tomando en cuenta que el promedio de horas diarias, que se utilizaron para el desarrollo fueron 6 en conjunto, y el número de horas que se estimaron para el plazo de entrega fueron de 151, se estimó que el costo del proyecto fue el sueldo correspondiente a un solo rol en aproximadamente 22 días de trabajo.

4.2.6 Calendario de actividades.

El proyecto se ejecutó en bloques temporales de cortas y fijas iteraciones. Cada iteración debía proporcionar un resultado completo y un incremento al producto, el cual sería visible y funcional, el calendario de actividades (tabla 4-4), define el periodo de tiempo en que cada iteración se desarrolló.

Tabla 4-4 Calendario de actividades para cada iteración.

Iteraciones	Fecha de Inicio	Fecha Final	Días
Portada	Lun, 2 de julio	Mié, 5 de julio	4
Catálogo	Jue, 6 de julio	Vie, 26 de julio	15
Ordenes	Lun, 27 de julio	Vie, 10 de agosto	11
Reportes	Lun, 13 de agosto	Jue, 23 de agosto	9
Herramientas	Vie, 24 de agosto	Mie, 29 de agosto	4
Panel de control	Jue, 30 de agosto	Mie, 19 de septiembre	15

4.2.7 Plan de manejo de riesgos.

Todos los proyectos, sin excepción tienen implícito algún tipo de riesgo, y este no fue la excepción. Administrar los riesgos (figura 4-5), consiste básicamente en identificarlos y controlarlos de manera efectiva, para ello se identifican los riesgos potenciales, se valora su probabilidad de ocurrencia y su impacto, y se establece una prioridad según su importancia.

Los criterios de puntuación de riesgo, que se definieron para la probabilidad de impacto fueron los siguientes: Muy Bajo: 1, Bajo: 2 Medio: 3 Alto: 4 Muy Alto: 5.

Utilizando los criterios definidos se pudo priorizar cada tipo de riesgo y definir la estrategia a seguir en caso de que sucediera (tabla 4-6).

Tabla 4-5 Análisis cualitativo y cuantitativo del riesgo.

Riesgo	Probabilidad	Impacto	Puntaje	Prioridad
Cliente no comprometido	Muy Bajo	Muy Alto	5	4
Falta de comunicación	Muy Alto	Alto	20	1
Problema en fecha de entrega	Medio	Alto	12	3
Experiencia técnica	Muy Bajo	Muy Alto	5	4
Modificaciones de última hora	Muy Alto	Medio	15	2
Insatisfacción del cliente	Bajo	Muy Alto	5	4

Tabla 4-6 Estrategias seleccionadas en caso de riesgo.

Puntaje	Estrategia
Entre 15 y 25	Evitar
Entre 6 y 14	Reducir
Entre 3 y 5	Aceptar activamente
Entre 1 y 2	Aceptar pasivamente

Una vez que los riesgos se identificaron, calcularon y priorizaron, se concibió un plan de contramedidas para dichos riesgos (tabla 4-7).

Tabla 4-7 Contramedidas seleccionadas para evitar el riesgo.

Riesgo	Contramedida	Acción
Cliente no comprometido	Aceptar Activamente	Establecer compromisos de tiempos de respuesta con el líder del proyecto por parte de cliente.
Falta de comunicación	Evitar	Fijar un cronograma de reuniones entre el responsable del proyecto y el cliente.
Problema en fecha de entrega	Reducir	Priorizar el proyecto e inyectar recursos a fin de cumplir con las fechas de entrega.
Experiencia técnica	Aceptar Activamente	Crear un programa de capacitación en los roles involucrados.
Modificaciones de última hora	Evitar	Establecer como acuerdo que cada modificación al proyecto reestimara el tiempo
Insatisfacción del cliente	Aceptar Activamente	Aumentar la comunicación entre el líder del proyecto y el cliente

4.2.8 Protocolo de entrega.

Las entregas tempranas y continuas de software aportaron un valor adicional al proceso de desarrollo de software. El protocolo de entrega (tabla 4-9), determinó, que actividades debían llevarse a cabo al término de cada una de las fases desarrolladas, la fecha máxima de entrega se estableció que debía ser un día después de terminada la iteración, en caso de no haberse realizado ninguna modificación a los requerimientos durante el ciclo de desarrollo.

Tabla 4-8 Protocolo de Entrega

Entregable	Fase	¿Quién revisa?	¿Quién aprueba?	Medio	Condiciones de entrega
Código Fuente	Final	Área Técnica	Área Técnica	Revisión de Código	Entregar código fuente
Plan de Pruebas y Resultados	Final	Área Técnica	Administración	Revisión de Plan y Resultados	Entregar documentos de pruebas
Programa funcional	Final	Área Técnica	Administración	Sesión de Trabajo	Realizar instalación
Manual de Usuario	Final	Área Técnica	Administración	Revisión de Documento	Manuales

4.3 Análisis del sistema.

En esta etapa, se presenta la implementación del análisis del sistema, que fue realizada de acuerdo a las especificaciones establecidas en el proceso metodológico del capítulo anterior, se definieron las funciones que debía realizar el software y se presentó el comportamiento del sistema en consecuencias y eventos externos.





4.3.1 Requerimientos funcionales.

Describen la interacción entre el sistema y su ambiente, independientemente de su implementación, el ambiente incluye al usuario y cualquier otro elemento externo que interactúa con el sistema.

Identificación de usuarios.

La estructura de usuarios, privilegios y restricciones (tabla 4-9), se encuentra establecida en el sistema ERP, el cual fue creado previamente con la consigna de servir como base a otros sistemas, lo anterior permite, tener una sola estructura de control de usuarios para múltiples aplicaciones de software. La relación entre estos usuarios se basa en herencia.

Tabla 4-9 Identificación de grupos de usuarios.

	General	Usuario con acceso solo a funciones de inventario
	CSR	Usuario con acceso solo a funciones de venta.
	Gerencia	Usuario con acceso a funciones de inventario y venta.
	Administrador	Usuario sin restricciones en las funcionalidades del sistema.

Casos de uso

El diagrama de casos de uso de primer nivel (figuras 4-2 a 4-4), permite tener una visión general de los procesos de la organización, así como también permite mostrar los límites y su entorno. Como muestra, se realiza la descripción y desarrollo de los casos de uso “acceso al sistema” y “consulta al catálogo de productos”.

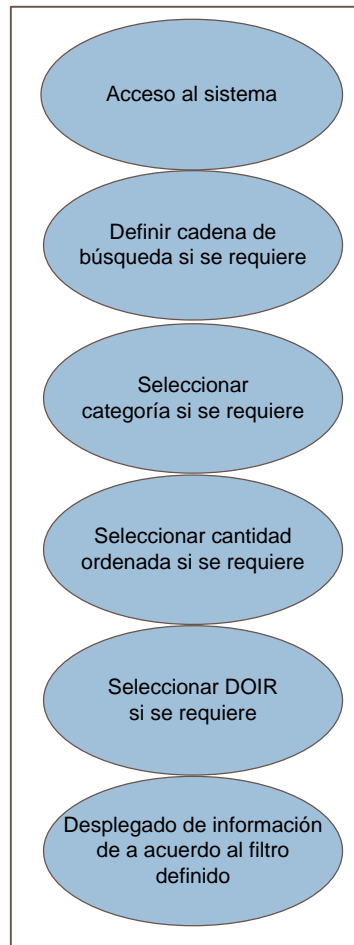


Figura 4-2 Descripción de caso de uso consulta al catálogo de productos.

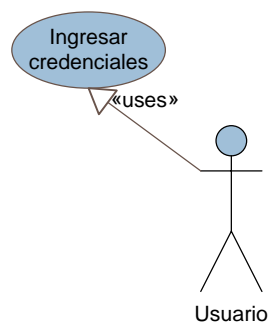


Figura 4-3 Desarrollo del caso de uso acceso al sistema.

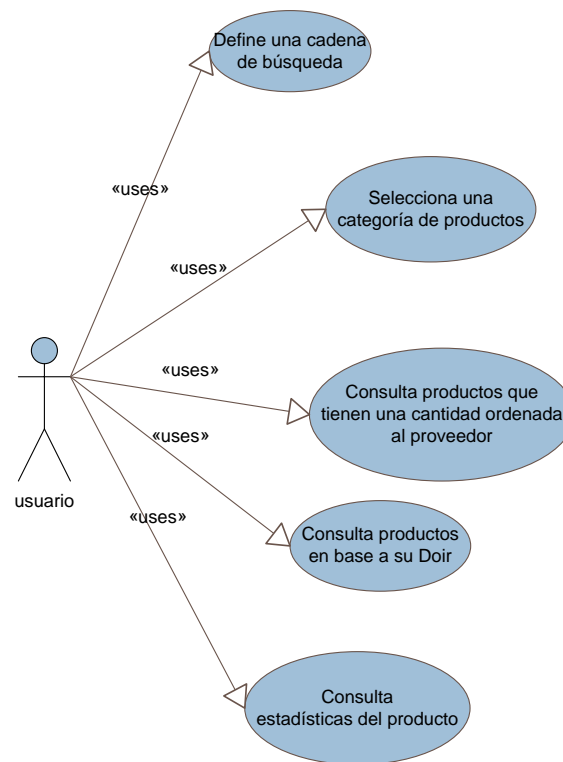


Figura 4-4 Desarrollo del Caso de uso Consulta al catálogo de productos.

Poda de requerimientos.

En la sección de planeación, se definió la pila de producto, la cual contiene los requerimientos propuestos por el dueño del sistema, esta lista debe ajustarse, eliminando redundancias, con el fin de que la especificación de requerimientos sea consistente.

Otras actividades que se realizan en esta sección serán:

- **Determinar en nivel de complejidad para cada requerimiento.**
- **Agrupar requerimientos en iteraciones de desarrollo.**

La lista de requerimientos final (tabla 4.9), fue obtenida después de una discusión entre el equipo de desarrollo, para determinar el nivel de complejidad de cada requerimiento y su correspondencia dentro de las iteraciones previamente descritas en el EDT, perspectiva del producto, cabe hacer notar que este proceso no se realiza de ninguna manera formal es únicamente en base a la experiencia del equipo.

Tabla 4-10 Lista de requerimientos actualizada y agrupada en ciclos de desarrollo

No	Iteraciones	Requerimientos	Complejidad
0	Portada	Acceso al sistema	1
1	Catálogo	Consulta al catálogo de productos	3
		Datos complementarios al catálogo de productos	1
		Información detallada del de producto	1
		Información de campos personalizados	2
		Compatibilidades	4
		Imágenes del producto	1
		Documentos relacionados al producto	1
		Historial de movimientos	2
		Detalle de Inventario	2
		Requerimientos de Ensamble	2
		Gráficos estadísticos del producto	3
		Información del vendedor	2
2	Configurador(CSR)	Configurador	3
3	Costo Inventario	Costo del inventario	3
4	C. M. Ordenes	Centro de manejo de órdenes	3
		Información de envío	3
		Detalle de la orden	3
5	C. Facturación	Centro de Facturación	3
		Información de envío	1
		Detalle de la orden	1
		Consolidación de estados por tienda virtual	1
6	Exceso de inv.	Exceso de Inventario	1
7	Ordenes pendiente	Ordenes Pendientes	1
8	Ventas por producto	Ventas por producto	1
9	Costo de Inv.	Costos unitarios por producto	2
		Costos Totales por producto	2
10	Amazon FBA	Productos en Amazon FBA	3
		Ordenes Pendientes Amazon FBA	3
		Ordenes Completas Amazon FBA	2
11	Panel de control	Ventas totales por año de lámparas divididas por mes	2
		Ventas totales por año de lámparas de proyector divididas por mes	2
		Ventas diarias CSR en los últimos 30 días.	2
		Total de ventas anuales.	2
		Ventas totales por canal de venta.	2
		Artículos y órdenes vendidos y enviados por canal de venta.	2
		Diez productos más vendidos.	2
		Total de ventas por agente de venta a mayoreo.	2
		Total de ventas por agente de venta en México.	2
		Total de artículos y órdenes vendidos por agente de venta en México.	2

Tomando de referencia la lista actualizada con los requerimientos se pudo determinar la correspondencia que existe entre los usuarios, casos de uso y el ciclo de desarrollo (Figura 4-5).

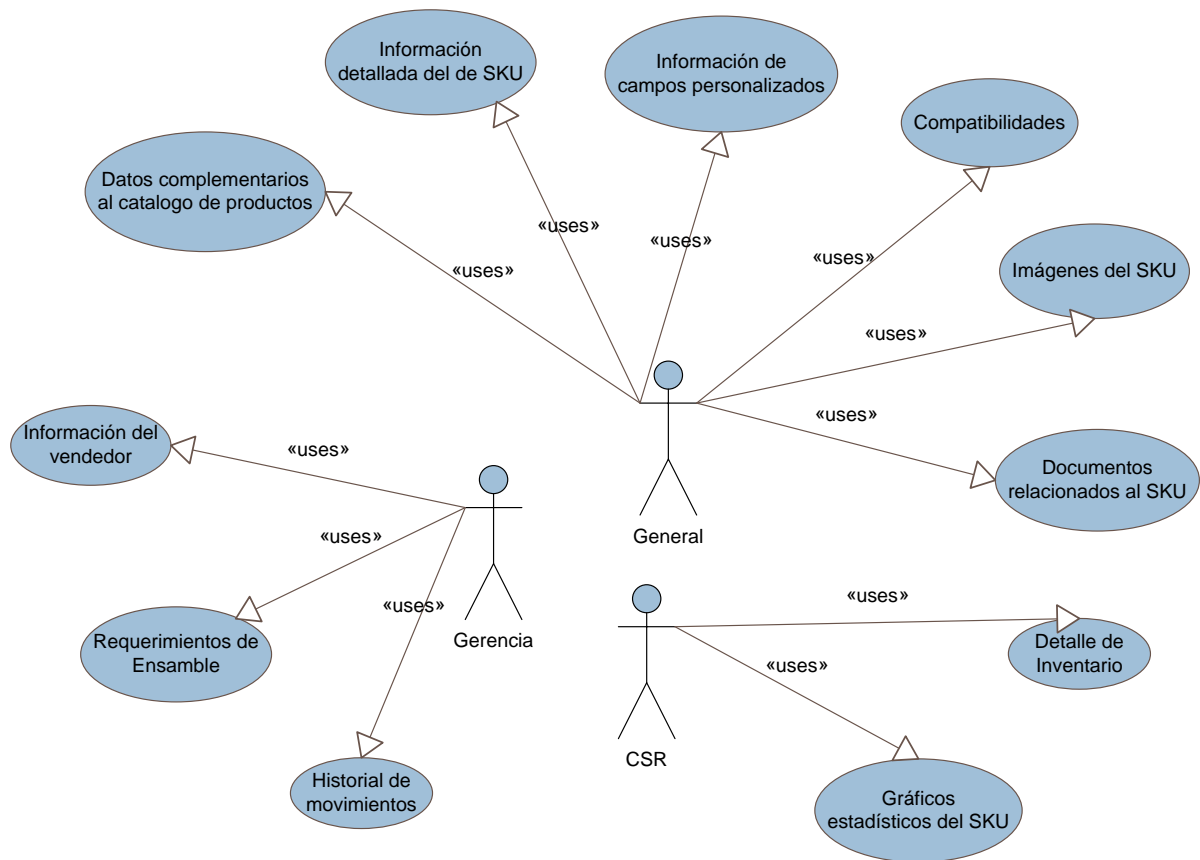


Figura 4-5 Casos de uso del ciclo catálogo.

4.3.2 Requerimientos no funcionales.

Define las restricciones que no pudieron ser negociadas y fueron impuestas por la empresa, como definición para el sistema.

Interface de usuario.

- Todas las páginas web deben mantener uniformidad en colores y forma para todos los objetos que la componen.
- En caso de cambio de resolución, todos los componentes en una página deben ajustarse automáticamente.
- Se maximiza el uso de descripciones emergentes en la presentación de información con gráficos.
- Los elementos de formularios actúan de forma independiente ajustando la información en el momento de sufrir un cambio.

Interface de Hardware.

El sistema es capaz de funcionar correctamente en cualquier dispositivo que cuente con servicio de internet y un navegador web, la calidad de la visualización depende de la resolución gráfica del dispositivo, la cual debe ser mayor o igual a 1024 x 768.

Interfaces de software.

SGBDR.

El sistema donde se encuentran almacenadas las bases de datos de los sistemas ERP-OM es un servidor Windows SQL Server 2008.

Patrón de desarrollo.

El sistema fue implementado bajo el patrón de desarrollo MVC a través del framework CodeIgniter. Los lenguajes de programación que se integraron en el sistema fueron:

- PHP mayor a la versión 5.3.
- Java Script.
- JQuery (Framework para uso de Ajax).
- HTML5.
- CSS.

Operatividad.

El sistema funciona correctamente en los siguientes navegadores:

- IE (Versión 8 y superiores).
- Firefox (Versión 3 y superiores).
- Safari (Versión 5 y superiores).
- Google Chrome(Versión 10 y superiores).

4.3.3 Otros requerimientos funcionales.

Son restricciones a los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo en el proceso de desarrollo. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad.

Desempeño.

- El sistema proporciona capacidad para soportar hasta 400 usuarios simultáneos.
- El sistema está disponible las 24 horas del día.

Seguridad.

Las sesiones creadas al ingresar al sistema, se almacenan en forma de cookies encriptados en la computadora del cliente, con un tiempo de vida de 2 horas.

4.4 Arquitectura y diseño.

A continuación, se presenta el desarrollo de la arquitectura y diseño de acuerdo a la metodología establecida en el capítulo anterior, esta sección identifica los elementos más importantes del sistema así como sus relaciones.

4.4.1 Vista general de la arquitectura del software.

En esta sección, se presentan los diagramas que fueron desarrollados para modelar el sistema , los cuales presentan la idea de la magnitud del proyecto.

Diagrama de arquitectura de bloques:

El diagrama de bloques (figura 4-6), contiene la representación gráfica del funcionamiento interno del sistema EyPDIO, se puede observar el tipo de actividad que se desarrolló en cada bloque y las relaciones entre los componentes. El patrón de desarrollo MVC separa al sistema en tres capas: la capa de datos (Modelo), la capa de presentación (Vista) y la capa de Negocio (Controlador).

El usuario no tiene acceso de ninguna manera a las capas de datos y de negocio, interactúa únicamente con la capa de presentación, y es a través de esta, que el usuario tiene contacto con el sistema, al ser un sistema web, los usuarios pueden acceder a él desde cualquier punto del planeta a través de internet, por medio de un navegador web.

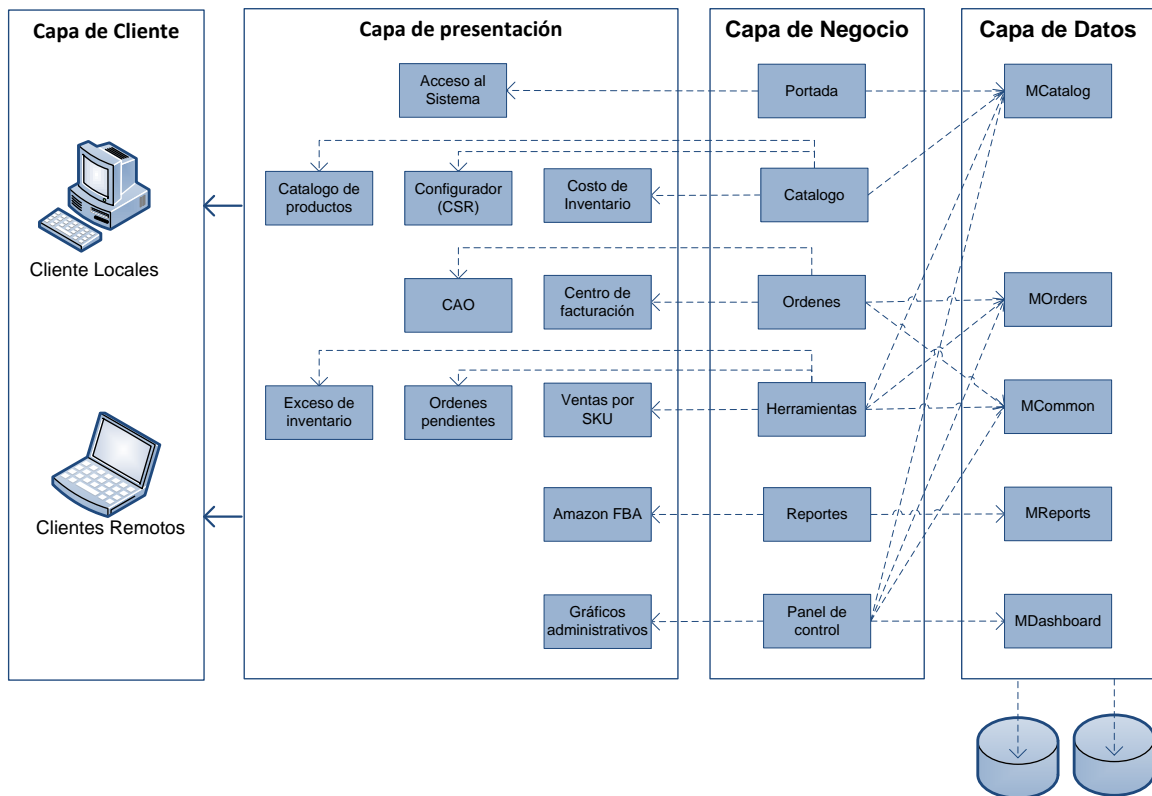


Figura 4-6 Diagrama de arquitectura de bloques.

Diagrama de arquitectura lógica:

El diagrama de arquitectura lógica (Figura 4-7), muestra las regiones en las que el sistema está dividido, por un lado tenemos la división que el patrón MVC impone, y por otro lado tenemos la división del sistema en capas de acuerdo al tipo de servicio que ofrece, cada región puede considerarse como un subsistema o un paquete, cada subsistema contienen una capa de servicios destinados a grupos de usuarios con jerarquías similares.

Dentro de los servicios que el sistema ofrece, tenemos:

- **Servicios de seguridad:** son aquellos que nos permiten cruzar la frontera entre la zona pública y la privada.
- **Servicios de acceso general:** Servicios a los que tienen derecho todos los usuarios.
- **Servicios gerenciales:** Son servicios de recopilación de información específica, que en gran parte tienen que ver con aspectos monetarios, costos o información, y que no se desean generalizar.
- **Servicios de alta dirección:** Por lo general son gráficos interactivos o concentrados de información que le sirven de apoyo a la alta dirección para tomar decisiones en cuanto al rumbo de la empresa.

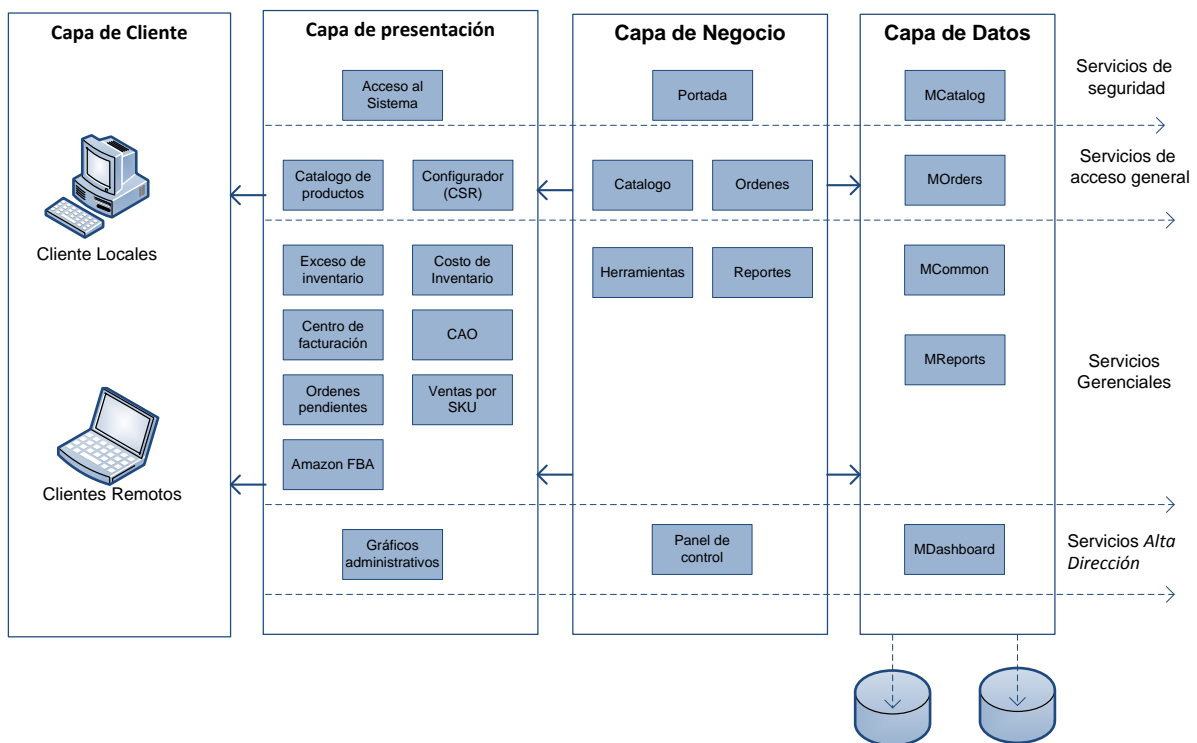


Figura 4-7 Diagrama de Arquitectura Lógica.

Diagrama de arquitectura física:

El Diagrama de arquitectura física (figura 4-8), nos muestra la estructura de elementos físicos que intervienen en el sistema EyPDIO, este diseño está regido por 2 servidores, uno SQL-Server 2008 donde se encuentran las bases de datos de los sistemas ERP y OM, de donde se extraerá la información, el otro es un servidor Linux donde reside la aplicación EyPDIO y el entorno para su ejecución. La otra parte la representa el cliente, este puede ser prácticamente cualquier sistema operativo con un navegador que soporte html5, css3 y JavaScript.

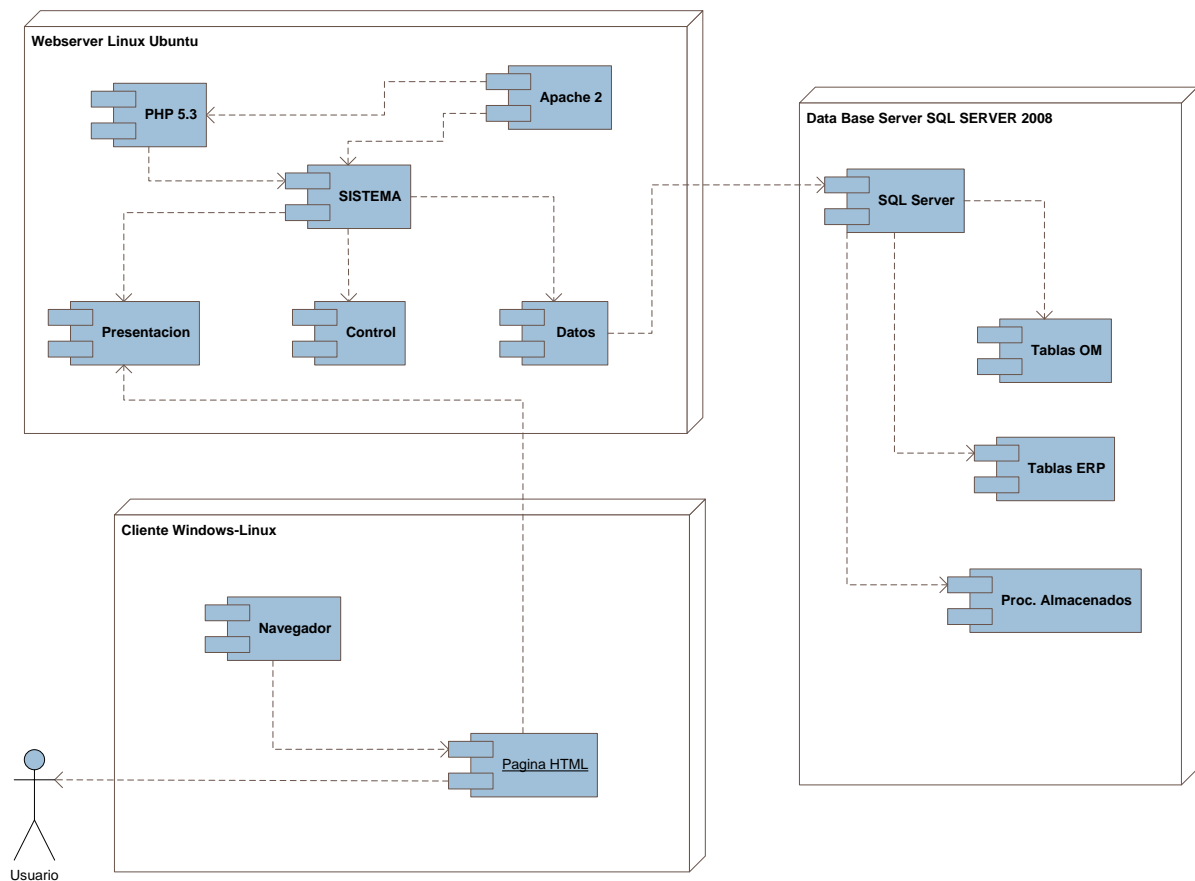


Figura 4-8 diagrama de arquitectura Física.

Diagrama de componentes:

El diagrama de componentes (figura 4-9), describe los aspectos físicos del sistema, modela la implementación estática y los elementos físicos o componentes que residen en un nodo, así como sus dependencias lógicas, comunicación y ubicación en el sistema EyPDIO.

Cada componente representa un bloque de construcción con un nivel de abstracción alta y se representa en este caso con el nombre asignado para el proceso que ejecuta.

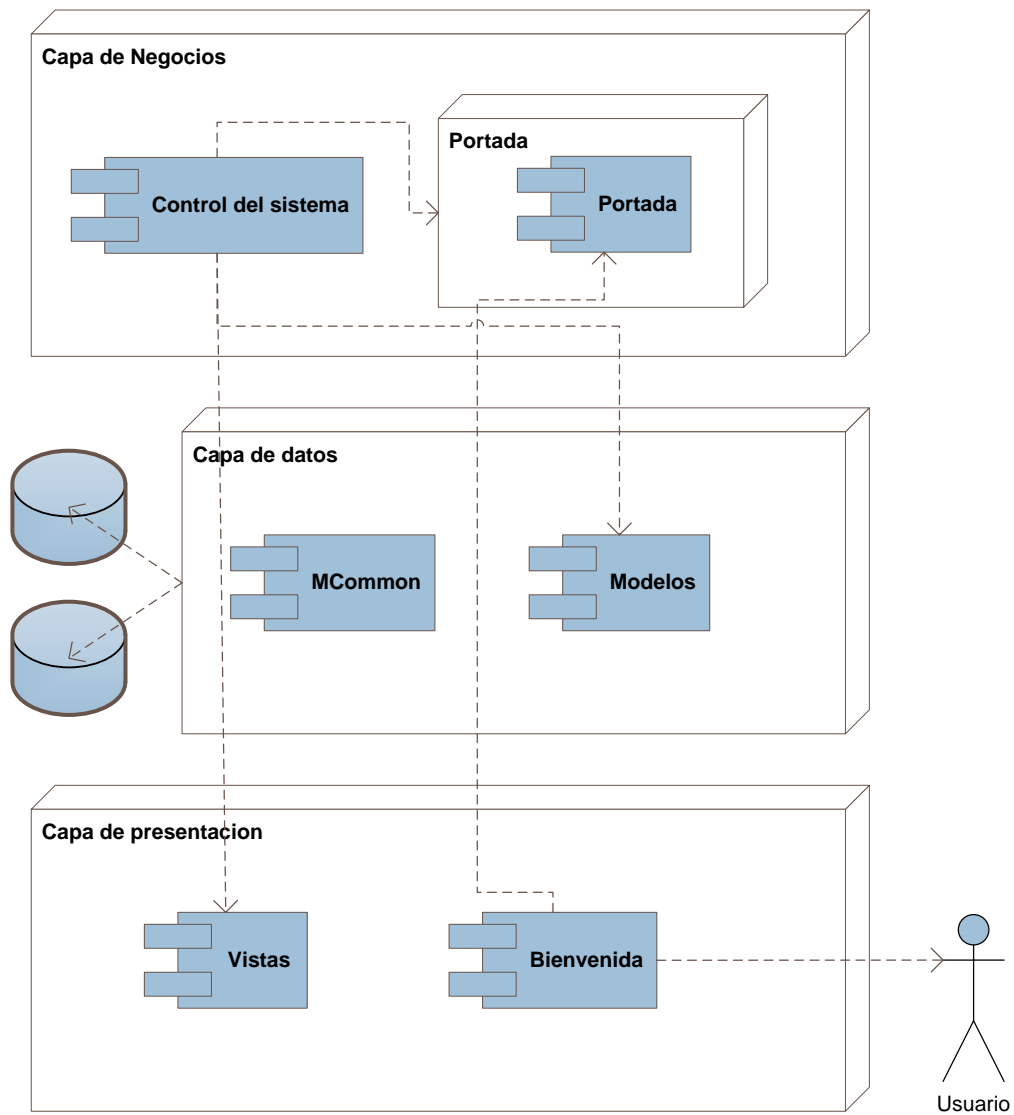


Figura 4-9 Diagrama de componentes.

4.4.2 Diseño detallado del sistema.

En esta etapa, se adecuó el análisis a las características específicas del ambiente de implementación y se completaron las distintas aplicaciones del sistema con los modelos de control, interfaz o comunicación, según el caso.

Vistas del caso de uso.

A manera demostrativa se presentan los tres primeros casos de uso del ciclo “Catálogo”, correspondientes a: Consulta al catálogo de productos (figura 4-10), Datos complementarios al catálogo de productos (figura 4-11), e Información detallada del producto (figura 4-12), el resto de los casos de uso se muestran en el anexo A.

Iteración: Catálogo.

Caso de Uso	Consulta al Catálogo de productos.	Caso de uso Numero: 2
Disparador	El usuario realiza consulta al <i>Catálogo</i> de productos.	
Precondición	Cualquier usuario valido puede tener acceso a esta sección	
Ruta básica	<ol style="list-style-type: none"> 1. El sistema presenta un formulario de búsqueda o con los campos necesarios para realizar consultas por palabra o frase, por categoría, por cantidad ordenada, por <i>DOIR (Days Of Inventory Remaining)</i> o combinaciones de todas estas. 2. El usuario puede digitar una palabra o una frase que considere para la búsqueda. 3. El usuario puede seleccionar una categoría. 4. El usuario puede seleccionar ver los artículos que tienen una Cantidad Ordenada. 5. El usuario puede seleccionar DOIR. 6. El sistema procesa la consulta general. 7. El sistema muestra los resultados en el <i>Grid</i> con campos <i>producto, VendorPN, Manufacturer, Name, QOH, DOIR, ExcessDays, UnitCost, TotalCost, QtyOrdered, Stats.</i> 	
Ruta alternativa	<p>En el paso 2 Si el usuario ingresa una frase:</p> <ol style="list-style-type: none"> (3) El sistema separa la frase en palabras (4) El sistema modifica la consulta global e incluir las condiciones necesarias para la búsqueda de cada palabra en todos los campos de la <i>Tabla Inventory.</i> (5) Enviar el control al paso 6. <p>En el paso 3, Si el usuario selecciona consulta por categorías:</p> <ol style="list-style-type: none"> (4) El sistema crea y presenta una lista con todas las categorías de productos en la base de datos. (5) El usuario selecciona la categoría. (6) El sistema modifica la consulta general integrando las condiciones para la búsqueda por la categoría seleccionada. (7) Retorna al paso 6. <p>En el paso 4 Si el usuario selecciona artículos con cantidad ordenada:</p> <ol style="list-style-type: none"> (4) El sistema modifica la consulta general integrando las condiciones necesarias para realizar el filtro de los registros que tengan un valor en el campo <i>QtyOrdered.</i> (5) Retorna el control al paso 6. <p>En el paso 5, Si el usuario selecciona DOIR:</p> <ol style="list-style-type: none"> (3) El sistema agrega un campo de forma automática al formulario para indicar los días en los que se basa este cálculo, por default tendrá 30. (4) El usuario puede modificar el número de días de base para el cálculo del DOIR. (5) El sistema modifica la consulta general agregando el resultado del procedimiento almacenado <i>DOIR</i> en el campo <i>DOIR</i> y el campo <i>ExcessDays.</i> (6) Retorna el control al paso 6. 	
Post-condición	<ol style="list-style-type: none"> 1. El <i>Grid</i> debe permitir editar los campos <i>TargetDays, QtyOrdered, ETA</i> y actualizar los datos automáticamente. 2. Si <i>DOIR</i> esta seleccionado el <i>Grid</i> debe colorear la columna, el campo se colorea en ROJO si <i>DOIR <= TargetDays</i> en caso contrario se colorea en VERDE. 	
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura 4-10 Caso de uso 'Consulta al catálogo de productos'.

Caso de Uso	Datos complementarios.	Caso de uso Numero: 3
Disparador	El usuario presiona el símbolo de + al lado derecho de cada renglón en el <i>Grid Catalog</i> .	
Precondición	El <i>Grid</i> debe contener datos y una sesión activa.	
Ruta básica	<ol style="list-style-type: none"> 1. El sistema modifica al <i>Grid Catalog</i> para que permita la inclusión de <i>SubGrids</i>. 2. El <i>Grid</i> se modifica y presenta un símbolo de + en el lado derecho. 3. Si el Usuario presiona el símbolo de +. 4. El <i>Grid</i> expande un área justo debajo del renglón seleccionado. 5. El <i>Grid</i> envía el ID del renglón seleccionado al sistema. 6. El <i>Grid</i> integra una serie de <i>Tabs</i> en el área desplegada. 7. Cada <i>Tab</i> contendrá un <i>frame</i> con una página distinta. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> debe permitir exportar el contenido a formato de Excel.	

Figura 4-11 Caso de uso 'Datos complementarios'.

Caso de Uso	Información detallada del <i>producto</i> .	Caso de uso Numero: 4
Disparador	El Usuario Obtiene una sección para presentar información complementaria al <i>producto</i> seleccionado.	
Precondición	<i>SubGrid</i> y <i>Tabs</i> para datos complementarios.	
Ruta básica	<ol style="list-style-type: none"> 1. El usuario presiona el símbolo de + en un renglón de <i>GRID</i>. 2. El <i>Grid</i> envía el ID del renglón al sistema (En este caso lo llamaremos <i>producto</i>) al proceso <i>productoData</i>. 3. El <i>Grid</i> Expande el renglón, incluye los <i>Tabs</i>. 4. El Primer <i>Tab</i> incluye el <i>Frame</i> y hace referencia al proceso <i>productoData</i>. 5. El proceso <i>productoData</i> crea una consulta con todos los campos del <i>Catálogo</i> de productos del <i>producto</i> seleccionado. 6. El sistema ejecuta la consulta. 7. El Proceso <i>productoData</i> muestra la información en forma de <i>Tabla</i>. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura 4-12 Caso de uso 'Información detallada del producto'

4.5 Desarrollo y pruebas.

Esta sección, desarrolla el caso de uso “Consulta al catálogo de productos”, que es el primer caso de uso que tiene un nivel de complejidad de 3, y sirvió como “prueba de control” para el resto del sistema.

Este tipo de proceso de desarrollo permite hacer entregas parciales que se van complementando según avanza el proyecto. De esta manera se reducen los riesgos y a la vez el cliente va experimentando los resultados de su proyecto.

Dado que los proyectos de software son largos es común dividir el trabajo en mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Para ser más efectivas las iteraciones deben ser controladas, es decir deben ser seleccionadas y llevadas a cabo de una forma planeada.

Se ha propuesto un proceso iterativo para garantizar la retroalimentación de información y de requisitos una vez iniciado el desarrollo, así como la validación continua del sistema. Esto

permite que cada iteración contemple ciclos de desarrollos completos y cortos, y obtener así rápidamente, una nueva versión con mejoras sobre las versiones anteriores.

Al entregar partes de la aplicación a tiempo y regularmente, el equipo de desarrollo también gana y establece credibilidad en su entorno y aumenta su auto-estima. A la vez que este tipo de estrategia se enfoca más en las necesidades de los usuarios, incitándolos a identificar sus prioridades en cada etapa del proyecto.

Implementación de la capa de datos, estructura, funciones y procedimientos.

La capa de datos es donde residen los datos y la forma de acceder a ellos:

Se divide en las siguientes áreas:

- Estructura de datos.
- Gestión de información a nivel de base de datos.
- Gestión de datos a nivel de aplicación.

El **Anexo B** muestra a detalle el desarrollo de estas actividades de acuerdo a los requerimientos que se establecieron para cada caso de uso.

Implementación de lógica de negocio o control.

La capa de lógica de negocio es donde residen las aplicaciones, se establecen todas las reglas que el proceso debe cumplir, se reciben las peticiones del usuario y se procesan las respuestas.

El **Anexo B** muestra a detalle el desarrollo de estas actividades, de acuerdo a los requerimientos establecidos en el caso de uso.

Implementación de la interface de usuario.

Es la capa que interactúa con el usuario (también se la denomina "capa de usuario"), presenta el sistema, comprueba errores de formato, comunica datos a la capa de negocio, recibe y despliega información. Como características distintivas tiene el ser entendible y fácil de usar.

El **Anexo B** muestra a detalle el desarrollo de estas actividades de acuerdo a los requerimientos establecidos en cada caso de uso.

Integración de sistema y pruebas.

El objetivo es conectar las tres capas del patrón de desarrollo de manera que el flujo de información recorra todas las áreas, y permita realizar en conjunto la actividad o requerimiento asignado.

Revisión y retroalimentación

- Verificar que los resultados se adapten a los requerimientos funcionales.
- Verificar que los resultados se adapten al caso de uso.
- Verificar que los resultados se adapten a los Requerimientos no funcionales.
- Verificar que los resultados se adapten a los otros Requerimientos no funcionales.

4.6 Puesta en marcha.

Excluir compatibilidades hacia procesos previos, permite que las nuevas funcionalidades no afecten a los módulos anteriormente desarrollados, lo cual permite reemplazar la versión actual, por una con nuevas funcionalidades.

Una vez concluido el proceso de protocolo de entrega, se lleva a cabo la fase de implantación y puesta en marcha de las nuevas funcionalidades, las cuales se dividieron en cuatro etapas debido a las características del sistema EyPDIO:

1. Instalación del software.

Se instaló y configuró el software, junto con todos los requerimientos que el servidor requiere para el funcionamiento correcto del sistema EyPDIO.

2. Instalación de la aplicación.

Migración desde el servidor de desarrollo y pruebas al servidor definitivo.

3. Formación

El responsable del proyecto, preparó la documentación necesaria, y se encargó de capacitar a los usuarios en el uso de las nuevas funcionalidades.

4. Fase de cierre, inicio de la mejora y soporte.

Se dio por finalizado el ciclo, al haberse cubierto el desarrollo de los casos de uso propuestos en él, y entra en vigor el proceso de mejora y soporte. Durante este periodo se encontraron algunos errores, o casos excepcionales, que no fueron contemplados en la fase de análisis, los cuales deben ser estudiados e incluidos como correcciones al sistema.

Capítulo 5 Resultados.

5.1 Descripción de resultados.

Este análisis, se realizó a los tres meses de haber puesto en marcha el sistema de “Extracción y procesamiento dinámico de información operativa”.

5.1.1 Resultados a nivel general.

El objetivo general planteado indica “Desarrollar un sistema de software en formato Web, que permita la extracción y procesamiento dinámico de información operativa, que se integre en todos los departamentos de la empresa, que permita la planeación de actividades y sirva de apoyo a la toma de decisiones a nivel organizacional”.

A los tres meses de funcionamiento del sistema, el 85% de las estaciones de trabajo con computadora dentro de la empresa (figura 5-1), utilizan el sistema EyPDIO como herramienta de trabajo en sus actividades diarias.

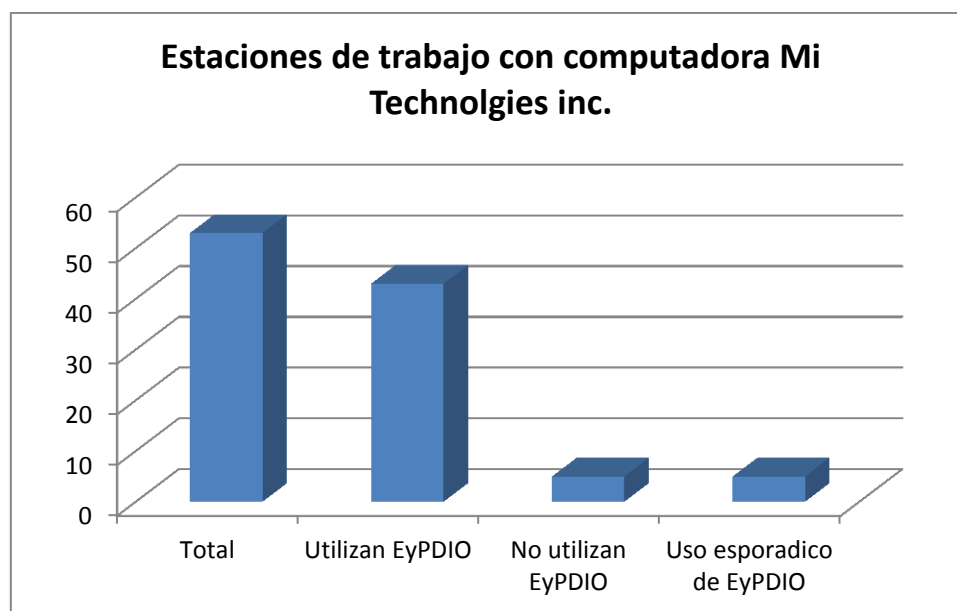


Figura 5-1 Estaciones de trabajo con computadora.

5.1.2 Análisis de los objetivos particulares definidos para el sistema.

Objetivo: “Desarrollar la interface de software en formato web que permita interactuar con la información de las bases de datos de los sistemas ERP-OM”.

A nivel de bases de datos, se logró el desarrollo de una gran cantidad de funciones y procedimientos almacenados. A nivel de aplicación se logró desarrollar dos clases específicas, que realizan la recolección de información de los sistemas ERP-OM, las cuales con modificaciones pequeñas pueden ser utilizadas en algún otro desarrollo de software.

Objetivo: “Analizar y delimitar el universo de soluciones que se pueden satisfacer con la información contenida en las bases de datos de los sistemas ERP-OM”.

Estructurar el proyecto en “casos de uso”, permitió, delimitar el universo de posibles soluciones que se pueden realizar con la información contenida en los sistemas ERP-OM. Cada Caso de uso fue analizado por separado a fin de determinar la viabilidad del mismo, lo cual permitió descartar requerimientos no viables, en la fase de análisis.

Objetivo: “Crear procesos que aseguren la integridad de la información, la cual no debe corromperse o perder utilidad a lo largo de la vida útil del sistema EyPDIO”.

El 98% de las actividades en el sistema EyPDIO, son consultas, solo un 2% de actividades realizan cambios, o modifican información en las bases de datos, el proceso de modificación es utilizado únicamente por usuarios con jerarquía alta. El sistema cuenta, con diversos niveles de control que aseguran la integridad de la información.

Objetivo: “Utilizar la estructura de jerarquía de usuarios y contraseñas del sistema ERP, para controlar el acceso y delimitar áreas en el sistema EyPDIO”.

ERP - EyPDIO cuentan ahora con la misma estructura de usuarios (figura 5-2), siendo una ventaja, que las mismas credenciales de usuario funcionan para ambos sistemas.

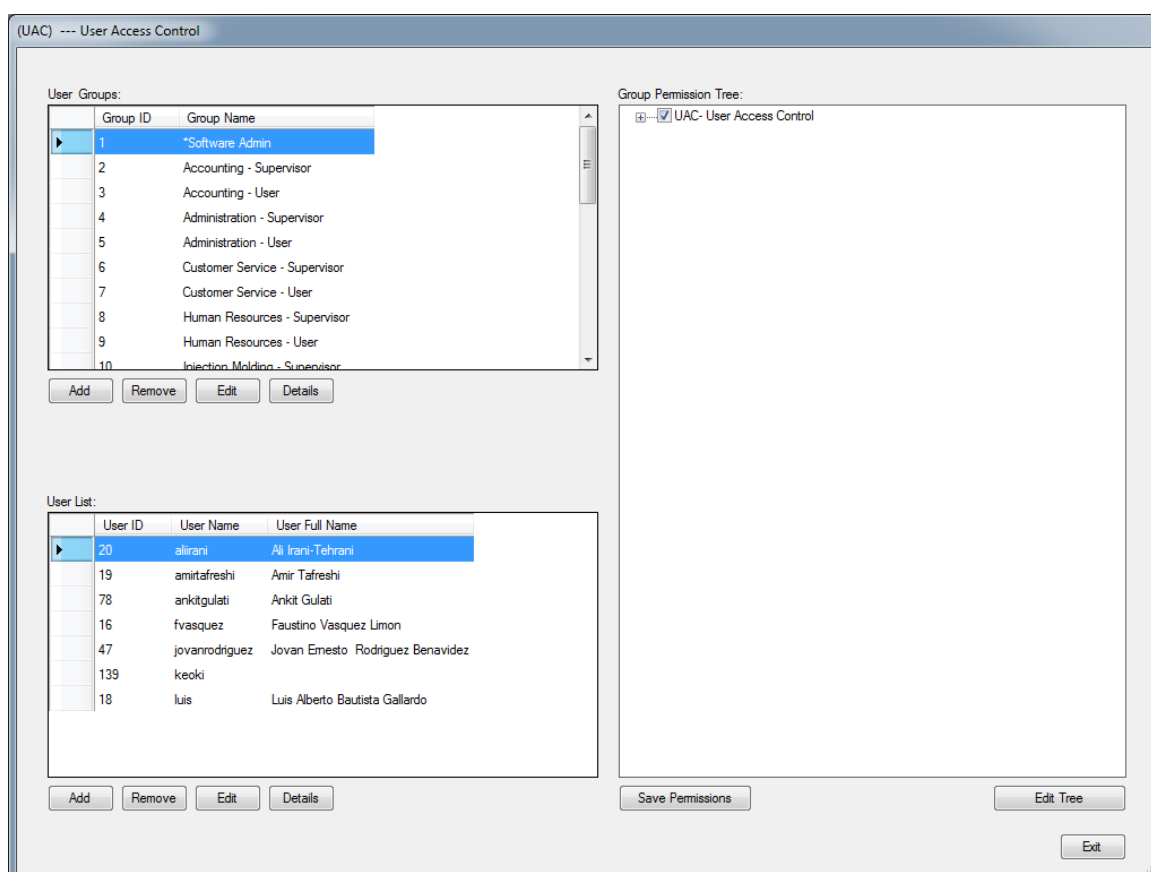


Figura 5-2 Administración de usuarios en ERP

Objetivo: “Desarrollar un formato estándar para desplegar resultados, el cual debe ser uniforme a todas las áreas del sistema EyPDIO”.

Capítulo 5. Resultados.

La calidad de la herramienta que permite el despliegue de información, también permite maximizar la reutilización del software, además mantiene la estética y funcionalidad común a todas las áreas del sistema (figuras 5-3 y 5-4).

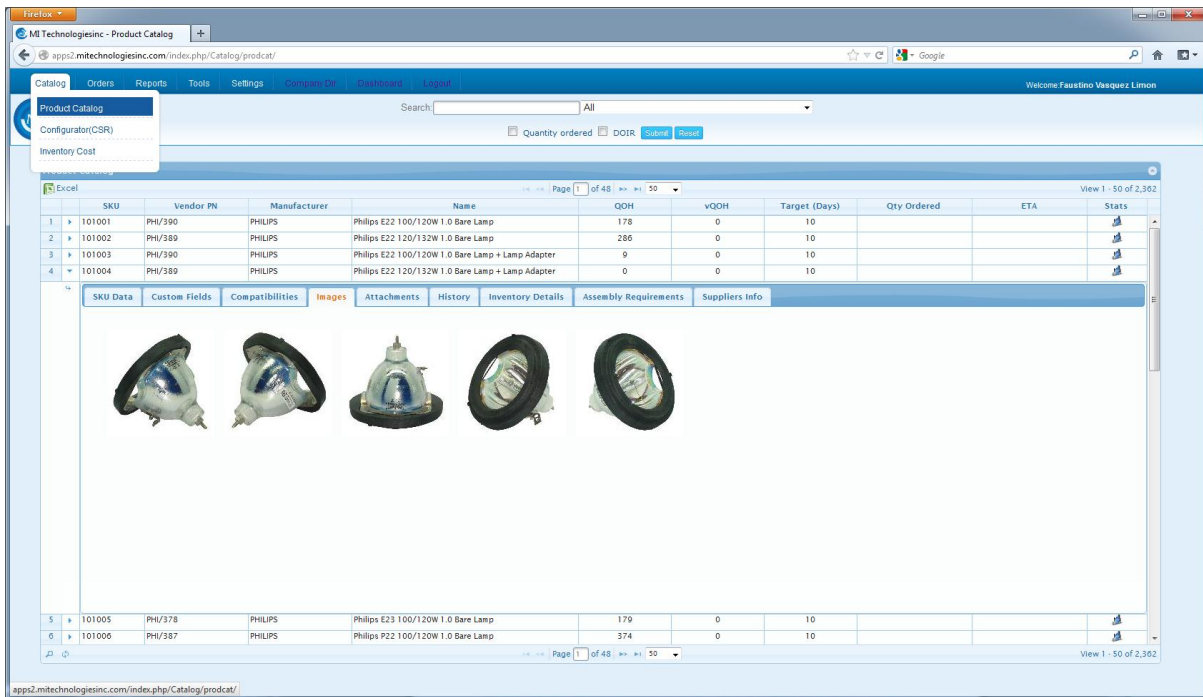


Figura 5-3 Imágenes de productos en catálogo.

El estilo único permite que todas las áreas del sistema presenten la información en un formato común (figura 5-4).

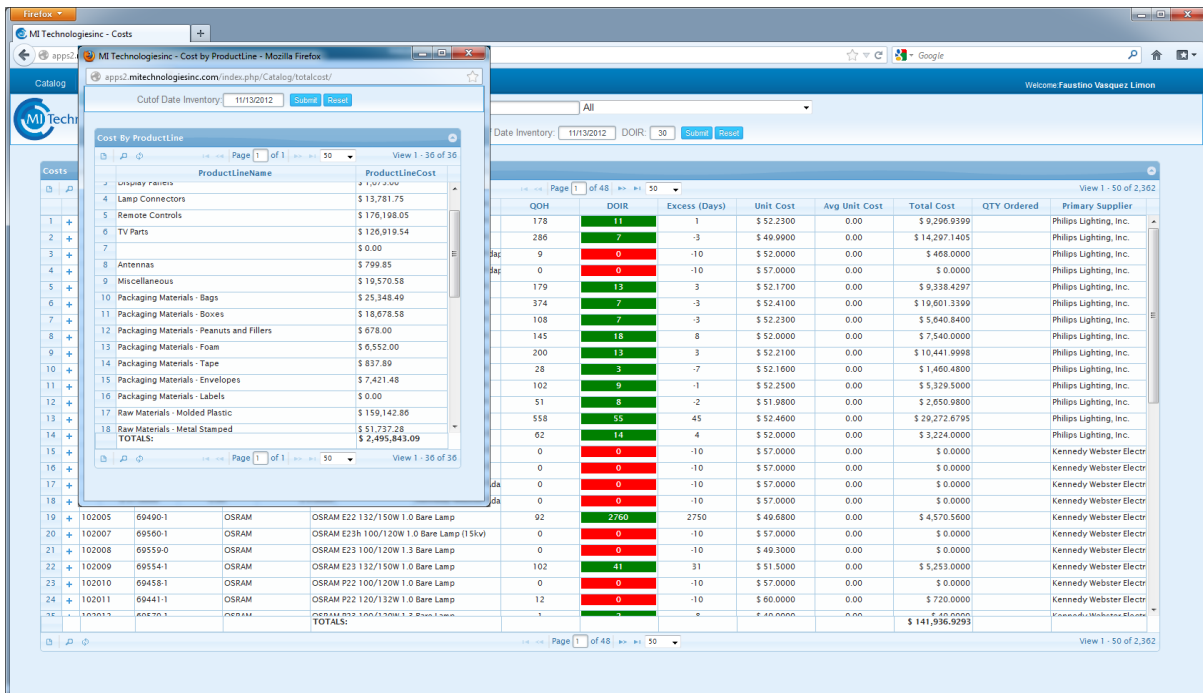


Figura 5-4 Costos de inventario.

Objetivo: “Ofrecer información resumida a través de gráficos descriptivos e interactivos, que permitan realizar comparativas de estados financieros en periodos diferentes. Las ayudas visuales con gráficos interactivos (figuras 5-5), son un gran aporte en el proceso de toma de decisiones.

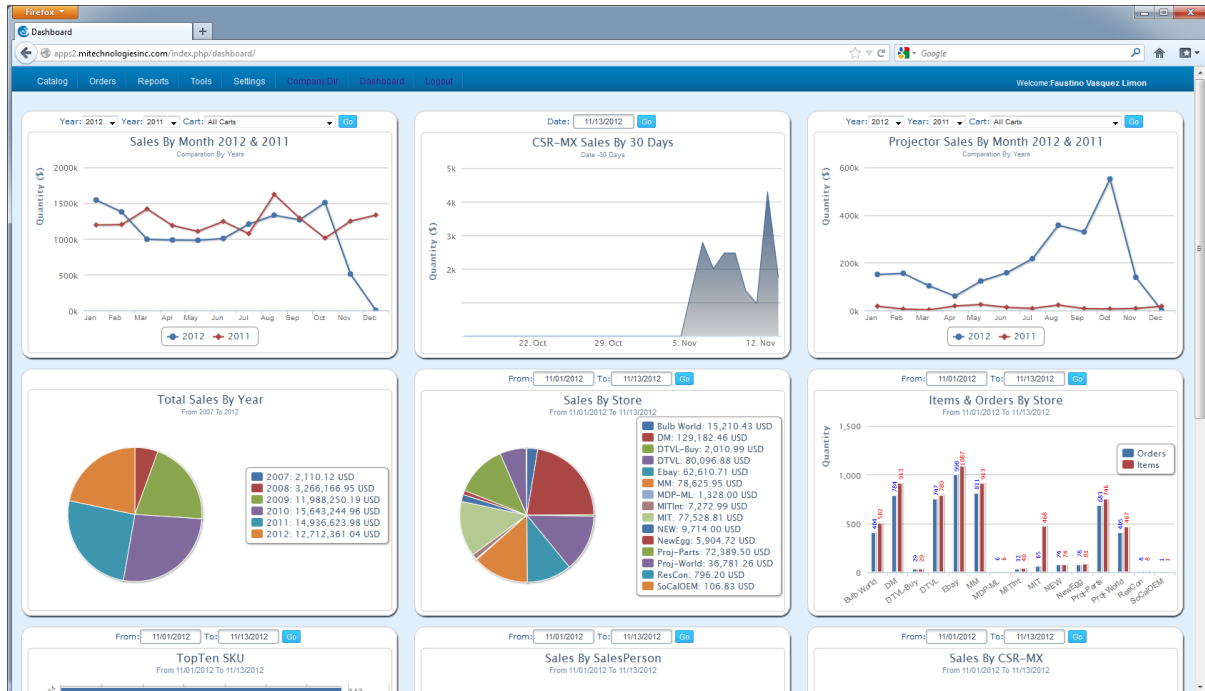


Figura 5-5 Panel de control con gráficos interactivos.

Los requerimientos presentados por los departamentos ejecutivos, se decidió presentarlos en forma de gráficos interactivos, la elección de la forma de presentación dependió enteramente del tipo de información que se debía representar, todas las comparativas de venta se representan con gráficos de línea, cada línea representa un estado diferente (figura 5-6, 5-7).



Figura 5-6 Ventas por mes.



Figura 5-7 ventas por mes, proyectores

Para los volúmenes de venta (figuras 5-8, 5-9, 5-10, 5-11), se decidió presentarlos con gráficos de pastel, esto permite identificar rápidamente que elemento tiene el mayor volumen de ventas.

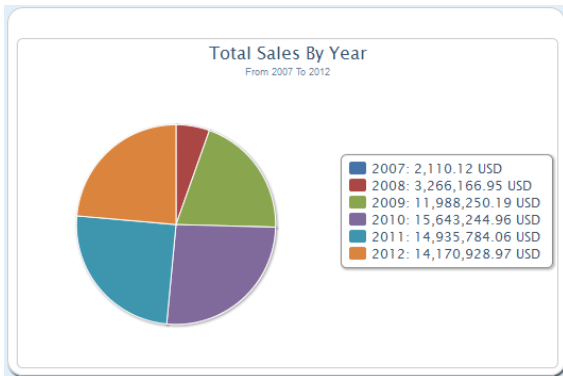


Figura 5-8, Ventas totales por año.

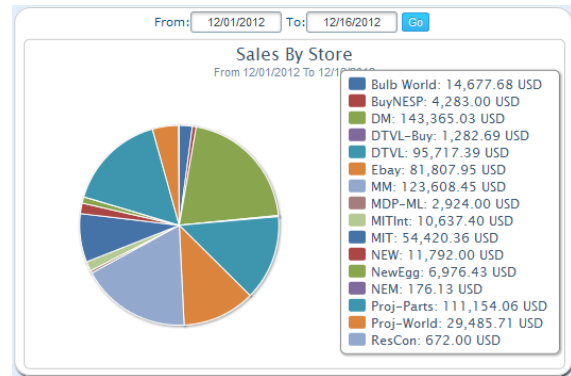


Figura 5-9 Ventas por tienda virtual en un periodo.

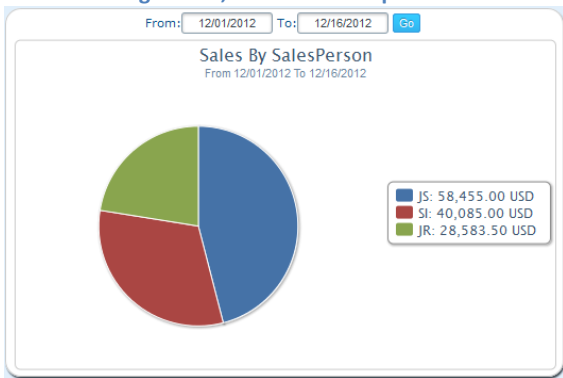


Figura 5-10 Ventas por agente de venta.

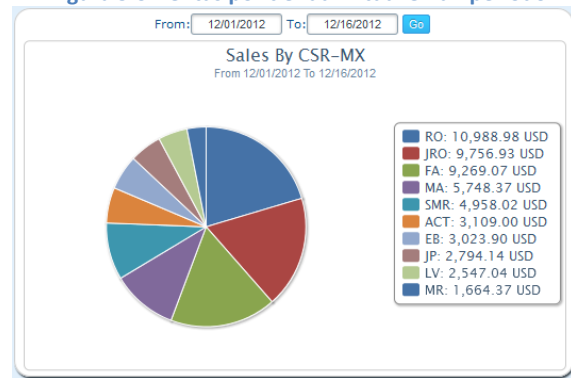


Figura 5-11 Ventas por Agente de venta en México.

Para los artículos y las ordenes (figura 5-12, 5-13), se decidió presentarlos con gráficos de barras, esto permite identificar rápidamente las cantidades representadas en el.

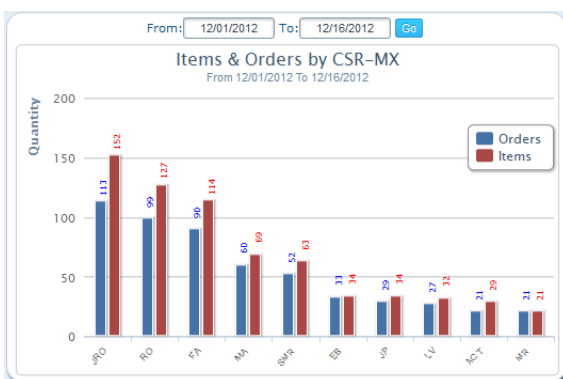


Figura 5-12 Artículos y ordenes x agente de vta.

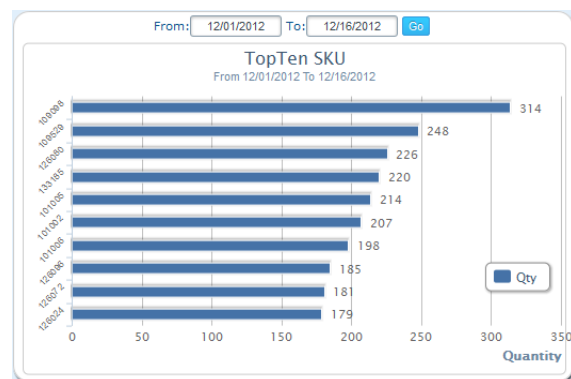


Figura 5-13 Los diez artículos mas vendidos.

Capítulo 5. Resultados.

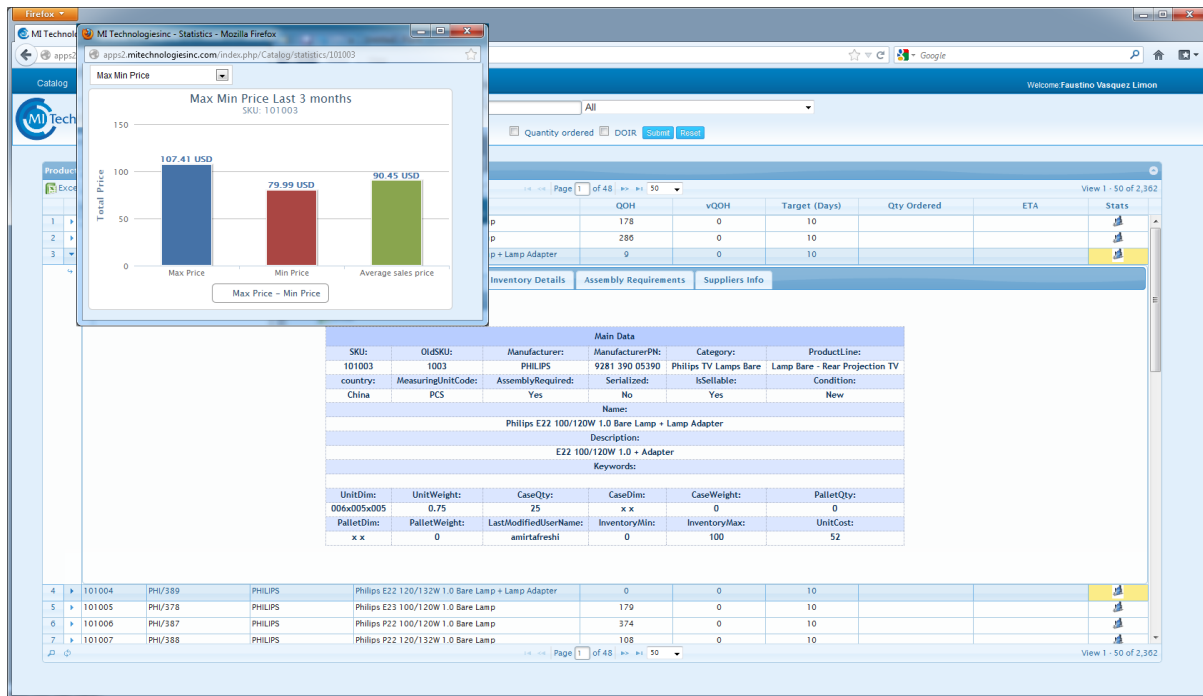


Figura 5-14 Gráficos de status para cada artículo en el inventario.

Objetivo: “Generar información detallada que apoye a la toma de decisiones a nivel directivo”.

La importancia de la información financiera (figura 5-14), que se presenta a los usuarios sirve para que formulen sus conclusiones sobre el desempeño de la empresa. Por medio de esta información y otros elementos de juicio, el usuario podrá evaluar el futuro de la empresa y tomar decisiones de carácter económico sobre la misma.

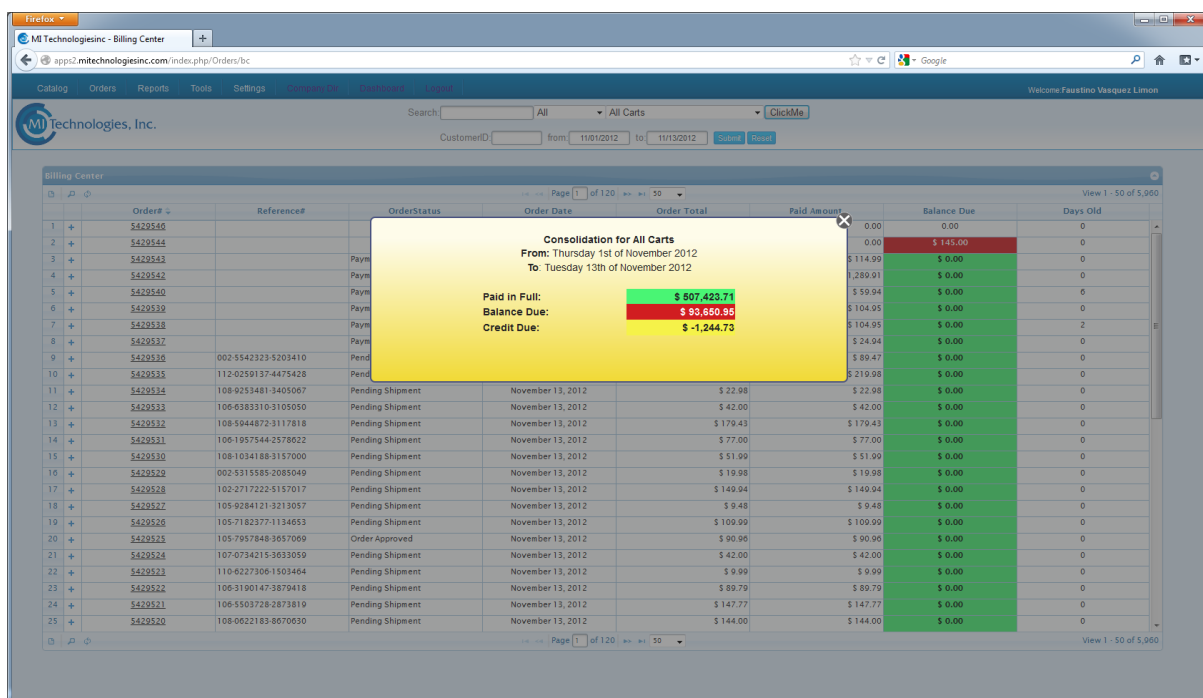


Figura 5-15 Consolidados de venta.

En el proceso comercial, todos los elementos que se integran son de vital importancia. La orden de compra es el elemento que activa toda la operación, es por ello, que ponerla a disposición de los diferentes departamentos, permite identificar errores en cualquiera de las fases del proceso.

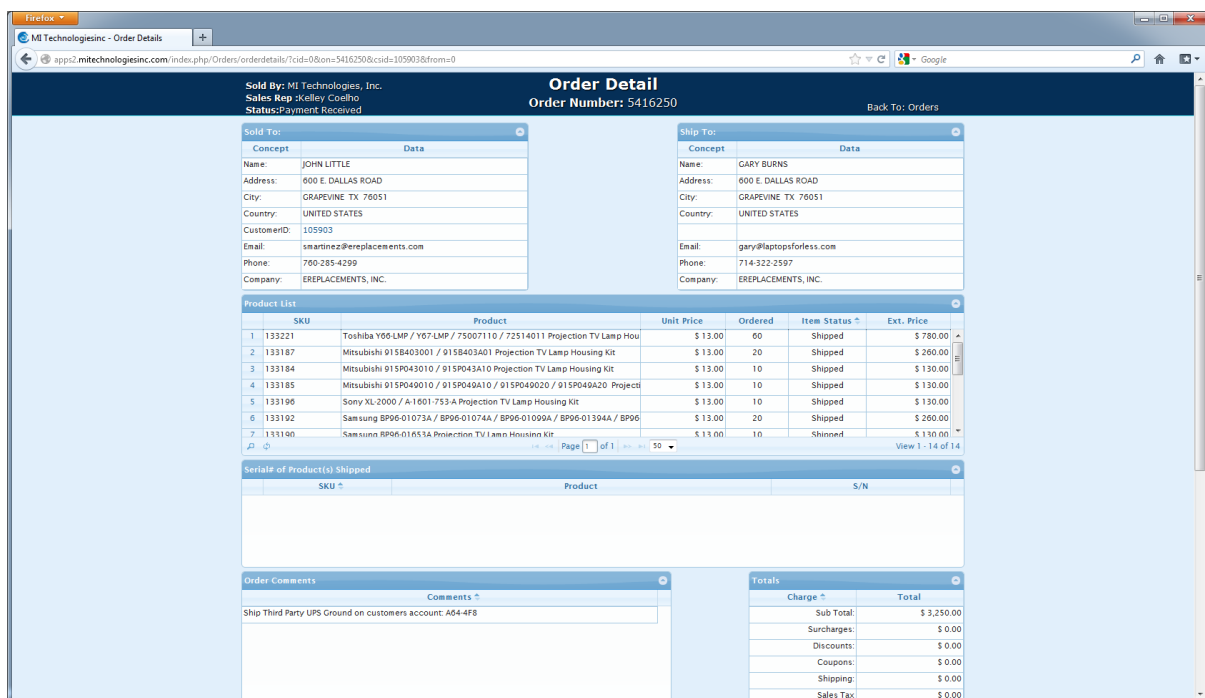


Figura 5-16 Detalle de la orden de venta.

Objetivo: Disponer de acceso al sistema EyPDIO, en todo momento bajo condiciones normales.

Bajo condiciones normales los servidores no se apagan nunca, lo cual garantiza que el sistema funcione todo el tiempo. El servidor que hospeda a la aplicación EyPDIO (figura 5-16), está basado en un sistema operativo Linux Ubuntu 12.04 LTS, el cual a través de su servicio LAMP permite ejecutar el sistema en los clientes remotos.

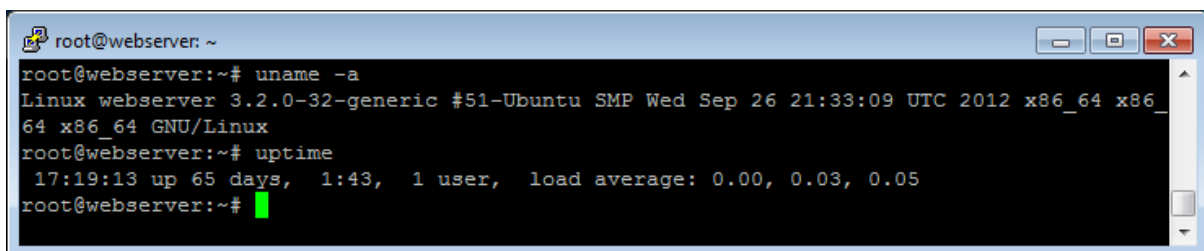


Figura 5-17 Servidor que hospeda al sistema EyPDIO.

Capítulo 6 Conclusiones.

6.1 Conclusiones.

La implementación del sistema para “Extracción y procesamiento dinámico de información operativa”, ha logrado alcanzar el objetivo que se propuso, convirtiéndose en una herramienta de trabajo necesaria en todos los departamentos de la empresa Mi Technologies Inc.

Haciendo un análisis detallado nos encontramos que existe otro resultado, que quizá no se contempló dentro de los objetivos del proyecto, el cual, fue la creación de una estructura o marco de trabajo destinado al desarrollo de proyectos de software, la cual fue adaptada a las características de la empresa y en especial al departamento de Informática.

Haber implementado la metodología Scrum en el desarrollo de un software real, ha sido toda una experiencia, de la cual se ha obtenido un gran aprendizaje en cada etapa del desarrollo, y muchas de las experiencias de este proyecto se trasladarán a otros. El hecho de haber terminado el trabajo obteniendo como resultado un producto funcional, demuestra que los objetivos planteados para el proyecto se cumplieron. Pero aún más importante es el grado de satisfacción que ha logrado el equipo de trabajo, además de la gran cantidad de conocimientos que se han obtenido.

Se utilizó un proceso ágil, iterativo e incremental que respeta las cinco etapas tradicionales de un proyecto, que facilitan su gestión y control; ellas son: planeación, análisis, diseño, desarrollo y pruebas e implementación. Cómo el objetivo final de la metodología es llegar al éxito del proyecto, se definen, en forma clara, los entregables de cada etapa y el alcance global, reflejando estos puntos en la planificación de todas las tareas involucradas.

De la combinación de herramientas utilizadas cabe destacar los casos de uso, resultaron ser una excelente herramienta para hacer el seguimiento diario del proyecto e ir midiendo constantemente la velocidad de desarrollo. Eso sí, para ello hay que ser perseverante y reestimar diariamente el esfuerzo pendiente de las tareas.

El proyecto logró desarrollarse dentro de los tiempos establecidos, nuevas propuestas han surgido y con ello nuevos retos. Existen requerimientos para desarrollar nuevas funcionalidades, como la administración de proyectos o integrar el inventario a tiendas virtuales; actualmente ya se está trabajando en estas funcionalidades que se espera, se liberen en una nueva versión del proyecto.

Con base a los resultados obtenidos, en el presente proyecto, se resumen las siguientes conclusiones:

- El proceso definido con las cinco etapas tradicionales del desarrollo de software se adaptó muy bien a la filosofía ágil. El producto final se llevó adelante sin mayores contratiempos, la comunicación entre el equipo de trabajo y la empresa, permitió darle mucha fluidez al proceso de desarrollo.
- Los beneficios del sistema se mostraron desde la liberación del primer sprint o ciclo, el “catálogo de productos”, el cual tuvo una aceptación inmediata y rápidamente se incorporó como herramienta de trabajo en los distintos departamentos.
- La información agrupada en el sistema, tiene un mayor control, cada requerimiento ha sido estudiado y perfeccionado por los especialistas en el área, lo cual ha permitido un control muy exacto en el tipo de información que el sistema genera.

- El sistema EyPDIO, ha logrado una alta aceptación entre el personal que lo utiliza, la información que genera, se utiliza como referencia para toma de decisiones, en la mayoría de las actividades referentes a ventas e inventario.

Se observa una gran importancia, en los beneficios que el sistema EyPDIO ha traído a la empresa, el cual está siendo utilizado desde los puestos operativos, hasta los altos mandos. El compromiso es alto, decisiones muy importantes para la empresa, se toman en base a la información que el sistema genera, convirtiéndolo en una herramienta estratégica que marcará el rumbo de la empresa Mi Technologies Inc. en los próximos años.

Anexo A

A manera demostrativa se presentan tres de las iteraciones completas, incluyendo casos de uso y casos de prueba.

Estas iteraciones representan los procesos más complejos desarrollados en el sistema EyPDIO.

Iteración: Portada

Acceso al sistema (Figura A-1), caso de uso que permite crear única página pública con el objetivo de recibir y validar credenciales de usuario.

Caso de Uso	Acceso al sistema	Caso de uso Numero: 1
Disparador	Las credenciales proporcionadas por el usuario deben se validan contra la base de datos del programa <i>Inventory</i> del programa ERP.	
Precondición		
Ruta básica	<ol style="list-style-type: none"> 1. El usuario ingresa sus credenciales, nombre de usuario y contraseña. 2. El Sistema evalúa las credenciales contra la <i>Tabla</i> usuario de la base de datos <i>Inventory</i> del sistema ERP. 3. El sistema genera una sesión con el nombre de usuario y su nivel jerárquico. 4. Se envía el control la sección <i>Dashboard</i> del sistema. 	
Ruta alternativa	En el paso 2, Si las credenciales ingresadas son incorrectas <ol style="list-style-type: none"> (1) Se envía un mensaje de error (2) Se retorna el control al paso 2 	
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro		

Figura A-1 Caso de uso "Acceso al sistema".

Iteración: Catálogo.

Esta iteración desarrolla los casos de uso (Figura A-2 a A-15), que presentan al el catálogo de productos y todos los elementos relacionados a este en una sola pantalla.

Caso de Uso	Consulta al Catálogo de productos.	Caso de uso Numero: 2
Disparador	El usuario realiza consulta al <i>Catálogo</i> de productos.	
Precondición	Cualquier usuario valido puede tener acceso a esta sección	
Ruta básica	<ol style="list-style-type: none"> 8. El sistema presenta un formulario de búsqueda o con los campos necesarios para realizar consultas por palabra o frase, por categoría, por cantidad ordenada, por <i>DOIR (Days Of Inventory Remaining)</i> o combinaciones de todas estas. 9. El usuario puede digitar una palabra o una frase que considere para la búsqueda. 10. El usuario puede seleccionar una categoría. 11. El usuario puede seleccionar ver los artículos que tienen una Cantidad Ordenada. 12. El usuario puede seleccionar DOIR. 13. El sistema procesa la consulta general. 14. El sistema muestra los resultados en el <i>Grid</i> con campos <i>producto, VendorPN, Manufacturer, Name, QOH, DOIR, ExcessDays, UnitCost, TotalCost, QtyOrdered, Stats.</i> 	
Ruta alternativa	<p>En el paso 2 Si el usuario ingresa una frase:</p> <ol style="list-style-type: none"> (6) El sistema separa la frase en palabras (7) El sistema modifica la consulta global e incluir las condiciones necesarias para la búsqueda de cada palabra en todos los campos de la <i>Tabla Inventory.</i> (8) Enviar el control al paso 6. <p>En el paso 3, Si el usuario selecciona consulta por categorías:</p> <ol style="list-style-type: none"> (8) El sistema crea y presenta una lista con todas las categorías de productos en la base de datos. (9) El usuario selecciona la categoría. (10) El sistema modifica la consulta general integrando las condiciones para la búsqueda por la categoría seleccionada. (11) Retorna al paso 6. <p>En el paso 4 Si el usuario selecciona artículos con cantidad ordenada:</p> <ol style="list-style-type: none"> (6) El sistema modifica la consulta general integrando las condiciones necesarias para realizar el filtro de los registros que tengan un valor en el campo <i>QtyOrdered.</i> (7) Retorna el control al paso 6. <p>En el paso 5, Si el usuario selecciona DOIR:</p> <ol style="list-style-type: none"> (7) El sistema agrega un campo de forma automática al formulario para indicar los días en los que se basa este cálculo, por default tendrá 30. (8) El usuario puede modificar el número de días de base para el cálculo del DOIR. (9) El sistema modifica la consulta general agregando el resultado del procedimiento almacenado <i>DOIR</i> en el campo <i>DOIR</i> y el campo <i>ExcessDays.</i> (10) Retorna el control al paso 6. 	
Post-condición	<ol style="list-style-type: none"> 3. El <i>Grid</i> debe permitir editar los campos <i>TargetDays, QtyOrdered, ETA</i> y actualizar los datos automáticamente. 4. Si <i>DOIR</i> esta seleccionado el <i>Grid</i> debe colorear la columna, el campo se colorea en ROJO si $DOIR \leq TargetDays$ en caso contrario se colorea en VERDE. 	
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura A-2 Caso de uso "Consulta al Catálogo de Productos".

Caso de prueba	Consulta al <i>Catálogo</i> de productos	Caso de Prueba Numero: 1
Propósito	Realizar consultas al <i>Catálogo</i> de productos	
Prerrequisitos	El usuario debe existir en la aplicación ERP y debe tener una sesión activa y acceso a esta área.	
Datos de prueba	<i>Search</i> = {Cadena, palabra, vacío} <i>Category</i> = {All, Elemento seleccionado} <i>QtyOrdered</i> = {verdadero, falso} <i>DOIR</i> = {verdadero, falso}	
Pasos	Pasos a ejecutar la prueba. <ol style="list-style-type: none"> 1. Teclar cadena de búsqueda, palabra o dejar vacío. 2. Presionar enviar 3. Observar el número de registros encontrados. 4. Seleccionar categoría. 5. Observar que los registros correspondan a la categoría seleccionada 6. Seleccionar <i>QtyOrdered</i>. 7. Observar que todos los registros en la columna <i>QtyOrdered</i> no estén vacíos 8. Seleccionar <i>DOIR</i> 9. Observar que se añaden las columnas <i>DOIR</i> y <i>ExcessDays</i> 10. Realizar combinaciones de todas las opciones anteriores 11. Seleccionar los campos <i>TargetDays</i>, <i>QtyOrdered</i>, <i>ETA</i> , hacer clic en ellos y modificar el contenido y hacer clic fuera del <i>Grid</i>, el campo debe actualizar el valor 	
Notas y preguntas		

Figura A-3 Caso de prueba "Consulta al catálogo de productos".

Caso de Uso	Datos complementarios al catálogo de productos.	Caso de uso Numero: 3
Disparador	El usuario presiona el símbolo de + al lado derecho de cada renglón en el <i>Grid Catalog</i> .	
Precondición	El <i>Grid</i> debe contener datos y una sesión activa.	
Ruta básica	<ol style="list-style-type: none"> 8. El sistema modifica al <i>Grid Catalog</i> para que permita la inclusión de <i>SubGrids</i>. 9. El <i>Grid</i> se modifica y presenta un símbolo de + en el lado derecho. 10. Si el Usuario presiona el símbolo de +. 11. El <i>Grid</i> expande un área justo debajo del renglón seleccionado. 12. El <i>Grid</i> envía el ID del renglón seleccionado al sistema. 13. El <i>Grid</i> integra una serie de <i>Tabs</i> en el área desplegada. 14. Cada <i>Tab</i> contendrá un <i>Frame</i> con una página distinta. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> debe permitir exportar el contenido a formato de Excel.	

Figura A-4 Caso de uso "Datos complementarios".

Caso de Uso	Información detallada del de producto.	Caso de uso Numero: 4
Disparador	El Usuario Obtiene una sección para presentar información complementaria al <i>producto</i> seleccionado.	
Precondición	<i>SubGrid</i> y <i>Tabs</i> para datos complementarios.	
Ruta básica	<ol style="list-style-type: none"> 8. El usuario presiona el símbolo de + en un renglón de <i>GRID</i>. 9. El <i>Grid</i> envía el ID del renglón al sistema (En este caso lo llamaremos <i>producto</i>) al proceso <i>productoData</i>. 10. El <i>Grid</i> Expande el renglón, incluye los <i>Tabs</i>. 11. El Primer <i>Tab</i> incluye el <i>Frame</i> y hace referencia al proceso <i>productoData</i>. 12. El proceso <i>productoData</i> crea una consulta con todos los campos del <i>Catálogo</i> de productos del <i>producto</i> seleccionado. 13. El sistema ejecuta la consulta. 14. El Proceso <i>productoData</i> muestra la información en forma de <i>Tabla</i>. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura A-5 Caso de uso "Información de producto".

Caso de Uso	Información de campos personalizados.	Caso de uso Numero: 5
Disparador	El Usuario Obtiene Toda la información referente al <i>producto</i> seleccionado.	
Precondición	Datos complementarios.	
Ruta básica	<ol style="list-style-type: none"> 1. El usuario presiona el símbolo de + en un renglón de <i>GRID</i>. 2. El <i>Grid</i> envía el ID del renglón al sistema (En este caso lo llamaremos <i>producto</i>) al proceso <i>CustomFiles</i>. 3. El <i>Grid</i> Expande el renglón, incluye los <i>Tabs</i>. 4. El Segundo <i>Tab</i> incluye el <i>Frame</i> y hace referencia al proceso <i>CustomFiles</i>. 5. El proceso <i>CustomFiles</i> crea una consulta para extraer el nombre de los campos personalizados más su valor que tienen relación con el <i>producto</i> seleccionado, estos campos varían de <i>producto</i> a <i>producto</i>. 6. El sistema ejecuta la consulta. 7. El Proceso <i>productoData</i> muestra la información en forma de <i>Tabla</i>. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura A-6 Caso de uso "Campos personalizados".

Caso de Uso	Compatibilidades.	Caso de uso Numero: 6
Disparador	El Usuario Obtiene información sobre las compatibilidades relacionadas al <i>producto</i> seleccionado.	
Precondición	Datos complementarios.	
Ruta básica	<ol style="list-style-type: none"> 1. El usuario presiona el símbolo de + en un renglón de <i>GRID</i>. 2. El <i>Grid</i> envía el ID del renglón al sistema (En este caso lo llamaremos <i>producto</i>) al proceso <i>Compatibilities</i>. 3. El <i>Grid</i> Expande el renglón, incluye los <i>Tabs</i>. 4. El Tercer <i>Tab</i> incluye el <i>Frame</i> y hace referencia al proceso <i>Compatibilities</i>. 5. El Proceso <i>Compatibilities</i> crea una consulta <i>Compatibilities</i> para extraer todos los <i>PartNumber</i>, <i>Manufacturer</i>, <i>ApprovedBy</i>, que tienen relación con el <i>producto</i> seleccionado. 6. El Sistema ejecuta la consulta <i>Compatibilities</i>. 7. El Sistema crea 4 <i>Grids</i>, <i>Compatibilities</i>, <i>CompatibilityDetails</i>, <i>AlternativePN</i>, <i>CompatibleAlternativeModel</i>. 8. El Sistema muestra la información de la consulta, en el <i>Grid Compatibilities</i>. 9. El Usuario puede seleccionar cualquiera de los resultados en el <i>Grid Compatibilities</i> que debe actualizar automáticamente a los <i>Grids Compatibility Details</i> y <i>AlternativePN</i>. 10. El Proceso <i>Compatibilities</i> crea una consulta para extraer todos los <i>Model</i> y <i>ApprovedBy</i> 11. El sistema ejecuta la consulta <i>Compatibilities</i>. 12. El sistema muestra la información de todos los detalles de compatibilidad en el <i>Grid Compatibilities Details</i>. 13. El Proceso <i>Compatibilities</i> crea una consulta para extraer todos los <i>AlternativePN</i> y <i>AddedBy</i>. 14. El sistema ejecuta la consulta <i>Compatibilities</i>. 15. El sistema muestra la información de todos los detalles de compatibilidad en el <i>Grid AlternativePN</i>. 16. El Usuario puede seleccionar cualquier elemento del <i>Grid Compatibilities Details</i> que debe actualizar automáticamente al <i>Grid Compatible Alternative Models</i>. 17. El Proceso <i>Compatibilities</i> crea una consulta para extraer todos los <i>AlternativeModel</i> y <i>ApprovedBy</i>. 18. El sistema ejecuta la consulta <i>Compatibilities</i>. 19. El sistema muestra la información de todos los modelos alternativos compatibles en el <i>Grid Compatible Alternative Model</i>. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura A-7 Caso de uso "Compatibilidades".

Caso de prueba	Compatibilidades.	Caso de Prueba Numero: 2
Propósito	Desplegar la relación de compatibilidades entre Fabricantes, Modelos, Modelos alternativos, Numero de parte y Numero de parte alternativos.	
Prerrequisitos	El usuario debe existir en la aplicación ERP y debe tener una sesión activa y acceso a esta área.	
Datos de prueba	+ = {Clic, NoClic} Seleccionar = {Clic, NoClic}	
Pasos	Pasos a ejecutar la prueba. <ol style="list-style-type: none"> 1. Presionar clic en el botón + a la izquierda del <i>Grid</i>. 2. Ver: Se expande el <i>Grid</i> en la posición del clic. 3. Ver: En el tercer <i>Tab</i> se tienen mostradas las compatibilidades relacionadas al <i>producto</i> Seleccionado. 4. Ver: 4 <i>Grids</i>, <i>Compatibilities</i>, <i>Compatibility Details</i>, <i>AlternativePN</i>, <i>Compatible Alternative Model</i>. 5. Presionar Clic en cualquiera de los resultados del <i>Grid Compatibilities</i> que debe actualizar automáticamente a los <i>Grids Compatibility Details</i> y <i>AlternativePN</i>. 6. Presionar Clic en cualquiera de los Modelos y el sistema debe actualizar automáticamente el <i>Grid AlternativeModel</i> 	
Notas y preguntas		

Figura A-8 Caso de prueba "Compatibilidades".

Caso de Uso	Imágenes del producto.	Caso de uso Numero: 7
Disparador	El Usuario Obtiene las imágenes relacionadas al <i>producto</i> seleccionado.	
Precondición	Datos complementarios.	
Ruta básica	<ol style="list-style-type: none"> 1. El usuario presiona el símbolo de + en un renglón de <i>GRID</i>. 2. El <i>Grid</i> envía el ID del renglón al sistema (En este caso lo llamaremos <i>producto</i>) al proceso <i>Images</i>. 3. El <i>Grid</i> Expande el renglón, incluye los <i>Tabs</i>. 4. El Cuarto <i>Tab</i> incluye el <i>Frame</i> y hace referencia al proceso <i>Images</i>. 5. El proceso <i>Images</i> ejecuta un script <i>index.php</i> en el lado del servidor MitNasMX dentro de la carpeta con el mismo nombre del <i>producto</i>. 6. El proceso <i>Images</i> Muestra las imágenes dentro del <i>Tab Images</i> relacionados al <i>producto</i> seleccionado. 7. El usuario puede hacer clic a cualquiera de las imágenes. 8. El sistema despliega la imagen en una ventana alterna. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura A-9 Caso de uso "Imágenes".

Caso de Uso	Documentos relacionados al producto	Caso de uso Numero: 8
Disparador	El Usuario Obtiene Información sobre documentos relacionados al <i>producto</i> seleccionado.	
Precondición	Datos complementarios.	
Ruta básica	<ol style="list-style-type: none"> 1. El usuario presiona el símbolo de + en un renglón de <i>GRID</i>. 2. El <i>Grid</i> envía el ID del renglón al sistema (En este caso lo llamaremos <i>producto</i>) al proceso <i>Attachments</i>. 3. El <i>Grid</i> Expande el renglón, incluye los <i>Tabs</i>. 4. El Quinto <i>Tab</i> incluye el <i>Frame</i> y hace referencia al proceso <i>Attachments</i>. 5. El proceso <i>Attachments</i> ejecuta un script <i>attachments.php</i> en el lado del servidor MitNasMX dentro de la carpeta con el mismo nombre del <i>producto</i>. 6. El proceso <i>Attachments</i> Muestra todos los documentos relacionados al <i>producto</i> seleccionado. 7. El usuario puede hacer clic a cualquiera de los documentos. 8. El sistema despliega el documento. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel. MitNasMX es un servidor de almacenamiento masivo.	

Figura A-10 Caso de uso "Documentos".

Caso de Uso	Historial de movimientos	Caso de uso Numero: 9
Disparador	El Usuario Obtiene Información sobre movimientos de inventario relacionados al <i>producto</i> seleccionado.	
Precondición	<i>D</i> datos complementarios.	
Ruta básica	<ol style="list-style-type: none"> 1. El usuario presiona el símbolo de + en un renglón de <i>GRID</i>. 2. El <i>Grid</i> envía el ID del renglón al sistema (En este caso lo llamaremos <i>producto</i>) al proceso <i>History</i>. 3. El <i>Grid</i> Expande el renglón, incluye los <i>Tabs</i>. 4. El Sexto <i>Tab</i> incluye el <i>Frame</i> y hace referencia al proceso <i>History</i>. 5. El proceso <i>History</i> crea una consulta <i>History</i> para extraer el historial de los últimos 30 días que ha tenido el <i>producto</i> Seleccionado. 6. El proceso <i>History</i> crea la consulta y debe extraer <i>Date, OriginName, DestinationName, producto, AdjustmentType, Quantity, Remaining UserName y Comments</i>. 7. El sistema ejecuta la consulta. 8. El Proceso <i>History</i> muestra la información en el <i>Grid History</i>. 9. El Proceso <i>History</i> Modifica el <i>Grid History</i> Agregándole campos fecha de Inicio, fecha final, y filtrado por movimiento. 10. El usuario puede modificar la fecha inicial. 11. El usuario puede modificar la fecha final. 12. El usuario puede modificar el filtrado por movimiento. 	
Ruta alternativa	<p>En el paso 9, Si el usuario selecciona modificar la fecha inicial.</p> <ol style="list-style-type: none"> (1) El sistema presenta un calendario justo abajo del campo fecha inicial. (2) El usuario selecciona la fecha del calendario. (3) El valor seleccionado se despliega en el campo fecha inicial en formato mes/día/ año. (4) El sistema crea modifica la consulta <i>History</i> agregando la fecha seleccionada. (5) Retorna al paso 6. <p>En el paso 10, Si el usuario selecciona modificar la fecha final.</p> <ol style="list-style-type: none"> (1) El sistema presenta un calendario justo abajo del campo fecha final. (2) El usuario selecciona la fecha del calendario. (3) El valor seleccionado se despliega en el campo fecha final en formato mes/día/ año. (4) El sistema crea modifica la consulta <i>History</i> agregando la fecha seleccionada. (5) Retorna al paso 6. <p>En el paso 11, Si el usuario selecciona filtrado por movimiento:</p> <ol style="list-style-type: none"> (1) El sistema crea y presenta una lista con todos los movimientos de la <i>Tabla de Movements</i>. (2) El usuario selecciona el movimiento. (3) El sistema modifica la consulta de <i>History</i> integrando las condiciones para la búsqueda por el movimiento. (4) Retorna al paso 6. 	
Post-condición	El <i>Grid History</i> debe permitir exportar el contenido a formato de Excel.	
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura A-11 Caso de prueba "Historial".

Caso de Uso	Detalle de Inventario	Caso de uso Numero: 10
Disparador	El Usuario Obtiene Información acerca de en que almacenes se encuentra el <i>producto</i> seleccionado.	
Precondición	Datos complementarios.	
Ruta básica	<ol style="list-style-type: none"> 1. El usuario presiona el símbolo de + en un renglón de <i>GRID</i>. 2. El <i>Grid</i> envía el ID del renglón al sistema (En este caso lo llamaremos <i>producto</i>) al proceso <i>History</i>. 3. El <i>Grid</i> Expande el renglón, incluye los <i>Tabs</i>. 4. El Séptimo <i>Tab</i> incluye el <i>Frame</i> y hace referencia al proceso <i>InventoryDetails</i>. 5. El proceso <i>InventoryDetails</i> crea una consulta que debe extraer información de en que almacenes y las cantidades de artículos que se tienen inventariadas del <i>producto</i> seleccionado. 6. El proceso <i>InventoryDetails</i> crea la consulta y debe extraer <i>Warehouse</i>, <i>ProductCatalogName</i> y el <i>CurrentStock</i>. 7. El sistema ejecuta la consulta. 8. El sistema muestra la a través de información en <i>InventoryDetails</i> en un <i>Grid</i>. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura A-12 Caso de uso "Detalles de Inventario".

Caso de Uso	Requerimientos de Ensamble.	Caso de uso Numero: 11
Disparador	El Usuario Obtiene Información acerca de los componentes para ensamble del <i>producto</i> seleccionado	
Precondición	Datos complementarios.	
Ruta básica	<ol style="list-style-type: none"> 1. El usuario presiona el símbolo de + en un renglón de <i>GRID</i>. 2. El <i>Grid</i> envía el ID del renglón al sistema (En este caso lo llamaremos <i>producto</i>) al proceso <i>AssemblyRequirements</i>. 3. El <i>Grid</i> Expande el renglón, incluye los <i>Tabs</i>. 4. El Octavo <i>Tab</i> incluye el <i>Frame</i> y hace referencia al proceso <i>AssemblyRequirements</i>. 5. El proceso <i>AssemblyRequirements</i> crea una consulta que debe extraer información de los <i>Subproducto's</i> que se necesitan para ensamblar el <i>producto</i> seleccionado siempre y cuando este tenga la característica de ser <i>AssamblyRequired</i>. 6. El proceso <i>AssemblyRequirements</i> crea la consulta y debe extraer <i>Subproducto</i>, <i>Name</i>, <i>QtyRequired</i>, <i>CurrentStock</i>, <i>VirtualStock</i>, <i>Projection</i>, <i>Notes</i>, <i>ApprovedBy</i>. 7. El sistema ejecuta la consulta. 8. Sistema muestra la información en el <i>AssemblyRequirements</i> en un <i>Grid</i>. 	
Ruta alternativa		
Post-condición	En el paso 5 si el <i>producto</i> no tiene la característica de <i>AssamblyRequired</i> : <ol style="list-style-type: none"> 1. El sistema no muestra información. 	
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura A-13 Caso de uso "Requerimientos de Ensamble".

Caso de Uso	Gráficos Estadísticos del producto.	Caso de uso Numero: 12
Disparador	El Usuario Obtiene Información gráficos con estadísticas del <i>producto</i> Seleccionado	
Precondición	El usuario debe tener jerarquía de Administrador para hacer uso de esta sección.	
Ruta básica	<ol style="list-style-type: none"> 1. El Sistema detecta si la sesión de usuario activa tiene jerarquía de administrador 2. <i>Grid Catalog</i> se modifica y agrega una columna llamada Statistics a la derecha del <i>Grid</i> con un símbolo de un gráfico de barras en cada renglón. 3. El Usuario puede hacer clic en cualquiera de los de los gráficos de barras de la columna <i>Graphs</i>. 4. El <i>Grid</i> envía el ID del renglón al proceso <i>productoGraphs</i>. 5. El Proceso <i>productoGraphs</i> crea una consulta llamada <i>Movements</i> que filtra los movimientos que ha tenido el <i>producto</i> en el año pasado, en último mes, en el mes en cuestión, y obtiene un promedio de por últimos tres meses. 6. El proceso <i>productoGraphs</i> crea una consulta llamada <i>AveragePricingByMonth</i> que obtiene el precio promedio mensual del <i>producto</i> seleccionado, en el periodo de los últimos 12 meses, es decir si nos encontramos en abril el periodo comprendería de mayo del año pasado a abril de este año. 7. El proceso <i>productoGraphs</i> crea una consulta llamada <i>MaxMinPrice</i>, la cual obtiene el precio máximo de venta, el precio mínimo de venta y un precio promedio de venta por los últimos 3 meses. 8. El sistema procesa las consultas. 9. El sistema genera una ventana alterna para presentar las gráficas. 10. El Usuario puede seleccionar <i>StockMovements</i>, <i>AveragePricingByMonth</i> o <i>MaxMinPrice</i> 11. El proceso <i>AssemblyRequirements</i> crea la consulta y debe extraer <i>Subproducto</i>, <i>Name</i>, <i>QtyRequired</i>, <i>CurrentStock</i>, <i>VirtualStock</i>, <i>Projection</i>, <i>Notes</i>, <i>ApprovedBy</i>. 12. El sistema ejecuta la consulta. 13. El sistema genera tres gráficos y campo de selección. 14. Sistema muestra la información en el <i>AssemblyRequirements</i> en un <i>Grid</i>. 	
Ruta alternativa	<p>En el Paso 11 si el Usuario selecciona <i>StockMovements</i>:</p> <ol style="list-style-type: none"> (1) Se muestra una gráfica de barras horizontales que muestran los movimientos de año pasado, el mes pasado, este mes, promedio por los últimos 90 días. <p>En el Paso 11 si el Usuario selecciona <i>Average Pricing By Month</i>:</p> <ol style="list-style-type: none"> (1) Se muestra una gráfica de línea con 12 puntos uno por mes, si se selecciona algún punto debe mostrarse un <i>tooltip</i> que indica el precio máximo de venta en ese mes. <p>En el Paso 11 si el Usuario selecciona <i>MaxMinPrice</i> :</p> <ol style="list-style-type: none"> (1) El sistema muestra una gráfica de barras indicando el precio máximo, el precio mínimo y un promedio de venta por los últimos 3 meses. 	
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura A-14 Caso de uso "Gráficos estadísticos del producto".

Caso de prueba	Gráficos estadísticos del producto	Caso de Prueba Numero: 3
Propósito	Desplegar una serie de gráficos con estadísticas del <i>producto</i> seleccionado.	
Prerrequisitos	El usuario debe existir en la aplicación ERP y debe tener una sesión activa y acceso a esta área.	
Datos de prueba	Stats = {Clic, NoClic} Seleccionar ={Clic,NoClic}	
Pasos	<p>Pasos a ejecutar la prueba.</p> <ol style="list-style-type: none"> 1. Presionar clic en el icono con imagen de grafica en la columna <i>Stats</i> del <i>Grid</i> en cualquier <i>producto</i> 2. Ver: Se abre una ventana extra con un y se muestra un gráfico de barras horizontales con los movimientos del <i>producto</i> <ol style="list-style-type: none"> 1. <i>Last Year</i> 2. <i>Last Month</i> 3. <i>This Month</i> 4. <i>Avg by Month based on last 90 days</i> 3. Ver: Se encuentra un campo de selección con los elementos <ol style="list-style-type: none"> 1. <i>Stock Movements</i> 2. <i>Average Pricing by month</i> 3. <i>Max Min Price</i> 4. Presionar clic en <i>Average Pricing by Month</i> 5. Ver: Se muestra una gráfica de líneas con los últimos 12 meses a partir del mes en cuestión, y cada mes es representado con un punto. 6. Poner el apuntador del ratón en cualquier punto y se debe mostrar el precio de venta en ese mes para el <i>producto</i> seleccionado 7. Presionar clic en <i>Max Min Price</i> 8. Ver: Se muestra una gráfica de barras con los precios: 9. Precio máximo de venta del <i>producto</i> en los últimos 3 meses. 10. Promedio de venta del <i>producto</i> en los últimos 3 meses. 	
Notas y preguntas		

Figura A-15 Caso de prueba "Gráficos estadísticos del producto".

Iteración Panel de control.

Esta iteración desarrolla los casos de uso (Figuras A-16 a A-25), que representan, el estado financiero de la empresa por medio de información en forma de gráficos interactivos, destinados únicamente al área de alta gerencia.

Caso de Uso	Ventas totales por año de lámparas divididas por mes	Caso de uso Numero: 22
Disparador	El usuario obtiene una gráfica comparativa de ventas por mes y años.	
Precondición	El usuario debe ser administrador o tener privilegios para ver esta gráfica.	
Ruta básica	<ol style="list-style-type: none"> 1. El sistema presenta un formulario de búsqueda o con los campos necesarios para realizar comparativa de ventas totales entre dos años y por carro de compra. 2. El sistema muestra una gráfica de líneas que compara el año actual con el inmediato anterior mes por mes indicando el total de venta en cada mes. 3. El usuario selecciona el año que desee comparar. 4. El usuario selecciona el segundo año que desee comparar. 5. El usuario selecciona el carro de compra que desee comparar. 6. El sistema crea la consulta necesaria para la comparativa entre años. 7. El sistema ejecuta la consulta. 8. El sistema presenta al usuario la gráfica correspondiente que se solicitó. 	
Ruta alternativa	<p>En el paso 5, Si el usuario selecciona un carro de compra:</p> <ol style="list-style-type: none"> (1) El sistema crea y presenta una lista con todos los carros de compra en la base de datos. (2) El usuario selecciona el carro de compra. (3) El sistema modifica la consulta general integrando las condiciones para la búsqueda por la categoría carro de compra. (4) Retorna al paso 6. 	
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro		

Figura A-16 Caso de uso "Comparativa de ventas por mes y años".

Caso de Uso	Ventas totales por año de lámparas de proyector divididas por mes.	Caso de uso Numero: 24
Disparador	El usuario obtiene una gráfica comparativa de ventas por mes y años.	
Precondición	El usuario debe ser administrador o tener privilegios para ver esta gráfica.	
Ruta básica	<ol style="list-style-type: none"> 1. El sistema presenta un formulario de búsqueda o con los campos necesarios para realizar comparativa de ventas de proyector entre dos años y por carro de compra. 2. El sistema muestra una gráfica de líneas que compara el año actual con el inmediato anterior mes por mes indicando el total de venta en cada mes. 3. El usuario selecciona el año que desee comparar. 4. El usuario selecciona el segundo año que desee comparar. 5. El usuario selecciona el carro de compra que desee comparar. 6. El sistema crea la consulta necesaria para la comparativa entre años. 7. El sistema ejecuta la consulta. 	
Ruta alternativa	<p>En el paso 5, Si el usuario selecciona un carro de compra:</p> <ol style="list-style-type: none"> (1) El sistema crea y presenta una lista con todos los carros de compra en la base de datos. (2) El usuario selecciona el carro de compra. (3) El sistema modifica la consulta general integrando las condiciones para la búsqueda. (4) por la categoría carro de compra. <p>Retorna al paso 6.</p>	
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro		

Figura A-17 Caso de uso "Ventas totales por año de lámparas de proyector divididas por mes".

Caso de Uso	Ventas diarias CSR en los últimos 30 días.	Caso de uso Numero: 23
Disparador	El usuario obtiene una gráfica con los totales de venta diarios en el departamento de atención a clientes en México.	
Precondición	El usuario debe ser administrador o tener privilegios para ver esta gráfica.	
Ruta básica	<ol style="list-style-type: none"> 2. El sistema presenta un formulario de búsqueda o la fecha del día de hoy. 3. El sistema muestra una gráfica de área con los totales de venta diarios por treinta días. anteriores a la fecha indicada en el formulario. 4. El usuario selecciona la fecha en que desea la comparación. 5. El sistema crea la consulta necesaria para la obtención de los datos. 6. El sistema ejecuta la consulta. 7. El sistema presenta al usuario la gráfica de área correspondiente. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro		

Figura A-18 Caso de uso "Ventas diarias CSR en los últimos 30 días".

Caso de Uso	Total de ventas anuales.	Caso de uso Numero: 25
Disparador	El usuario obtiene una gráfica con los totales de venta de todos los años activos en la empresa.	
Precondición	El usuario debe ser administrador o tener privilegios para ver esta gráfica.	
Ruta básica	<ol style="list-style-type: none"> 1. El sistema crea la consulta necesaria para la obtención de los datos. 2. El sistema ejecuta la consulta. 3. El sistema presenta al usuario la gráfica de pastel con los totales de venta de por cada a año de actividad en la empresa. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro		

Figura A-19 Caso de uso "Total de ventas por anuales".

Caso de Uso	Ventas totales por canal de venta.	Caso de uso Numero: 26
Disparador	El usuario obtiene una gráfica con los totales de venta por cada una de las tiendas virtuales activas en la empresa.	
Precondición	El usuario debe ser administrador o tener privilegios para ver esta gráfica.	
Ruta básica	<ol style="list-style-type: none"> 1. El sistema presenta un formulario con la fecha de inicio y fecha final. 2. El sistema asigna automáticamente las fechas correspondientes al último mes a fecha inicial y fecha final. 3. El sistema presenta una gráfica de pastel con los porcentajes de ventas de cada una de las tiendas activas en la empresa. 4. El usuario puede modificar la fecha de inicio. 5. El usuario puede modificar la fecha final. 6. El sistema crea la consulta necesaria para la obtención de los datos. 7. El sistema ejecuta la consulta. 8. El sistema presenta al usuario la gráfica de pastel con los datos de los totales de venta por cada una de las tienda activas en la empresa. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro		

Figura A-20 Caso de uso "Ventas totales por canal de venta".

Caso de Uso	Artículos y órdenes, vendidos y enviados por canal de venta.	Caso de uso Numero: 27
Disparador	El usuario obtiene una gráfica de barras con los totales de artículos y ordenes por cada una de las tiendas virtuales activas en la empresa.	
Precondición	El usuario debe ser administrador o tener privilegios para ver esta gráfica.	
Ruta básica	<ol style="list-style-type: none"> 1. El sistema presenta un formulario con la fecha de inicio y fecha final. 2. El sistema asigna automáticamente las fechas correspondientes al último mes a fecha inicial y fecha final. 3. El sistema presenta una gráfica de barras con los totales de artículos y ordenes por cada una de las tiendas virtuales activas en la empresa. 4. El usuario puede modificar la fecha de inicio. 5. El usuario puede modificar la fecha final. 6. El sistema crea la consulta necesaria para la obtención de los datos. 7. El sistema ejecuta la consulta. 8. El sistema presenta una gráfica de barras con los totales de artículos y ordenes por cada una de las tiendas virtuales activas en la empresa. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro		

Figura A-21 Caso de uso "Artículos y órdenes, vendidos y enviados por canal de venta".

Caso de Uso	Diez productos más vendidos.	Caso de uso Numero: 28
Disparador	El usuario obtiene una gráfica de barras con los totales de artículos productos más vendidos por la empresa.	
Precondición	El usuario debe ser administrador o tener privilegios para ver esta gráfica.	
Ruta básica	<ol style="list-style-type: none"> 1. El sistema presenta un formulario con la fecha de inicio y fecha final. 2. El sistema asigna automáticamente las fechas correspondientes al último mes a fecha inicial y fecha final. 3. El sistema presenta una gráfica de barras con los totales de artículos productos más vendidos por la empresa. 4. El usuario puede modificar la fecha de inicio. 5. El usuario puede modificar la fecha final. 6. El sistema crea la consulta necesaria para la obtención de los datos. 7. El sistema ejecuta la consulta. 8. El sistema presenta una gráfica de barras con los totales de artículos productos más vendidos por la empresa. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro		

Figura A-22 Caso de uso "Diez productos más vendidos".

Caso de Uso	Total de ventas por agente de venta a mayoreo.	Caso de uso Numero: 29
Disparador	El usuario obtiene una gráfica de pastel con los totales de venta por agente de venta a mayoreo.	
Precondición	El usuario debe ser administrador o tener privilegios para ver esta gráfica.	
Ruta básica	<ol style="list-style-type: none"> 1. El sistema presenta un formulario con la fecha de inicio y fecha final. 2. El sistema asigna automáticamente las fechas correspondientes al último mes a fecha inicial y fecha final. 3. El sistema presenta una gráfica de pastel con los totales de venta por agente de venta a mayoreo. 4. El usuario puede modificar la fecha de inicio. 5. El usuario puede modificar la fecha final. 6. El sistema crea la consulta necesaria para la obtención de los datos. 7. El sistema ejecuta la consulta. 8. El sistema presenta una gráfica de pastel con los totales de venta por agente de venta a mayoreo. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro		

Figura A-23 Caso de uso "Total de ventas por agente de venta a mayoreo".

Caso de Uso	Total de ventas por agente de venta en México.	Caso de uso Numero: 30
Disparador	El usuario obtiene una gráfica de pastel con los totales de venta por agente de venta en México.	
Precondición	El usuario debe ser administrador o tener privilegios para ver esta gráfica.	
Ruta básica	<ol style="list-style-type: none"> 1. El sistema presenta un formulario con la fecha de inicio y fecha final. 2. El sistema asigna automáticamente las fechas correspondientes al último mes a fecha inicial y fecha final. 3. El sistema presenta una gráfica de pastel con los totales de venta por agente de venta en México. 4. El usuario puede modificar la fecha de inicio. 5. El usuario puede modificar la fecha final. 6. El sistema crea la consulta necesaria para la obtención de los datos. 7. El sistema ejecuta la consulta. 8. El sistema presenta una gráfica de pastel con los totales de venta por agente de venta en México. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro		

Figura A-24 Caso de uso "Total de ventas por agente de venta en México".

Caso de Uso	Total de artículos y órdenes vendidos por agente de venta en México.	Caso de uso Numero: 30
Disparador	El usuario obtiene una gráfica de barras con los totales de artículos y órdenes vendidos por agente de venta en México.	
Precondición	El usuario debe ser administrador o tener privilegios para ver esta gráfica.	
Ruta básica	<ol style="list-style-type: none"> 9. El sistema presenta un formulario con la fecha de inicio y fecha final. 10. El sistema asigna automáticamente las fechas correspondientes al último mes a fecha inicial y fecha final. 11. El sistema presenta una gráfica de barras con los totales de artículos y órdenes vendidos por agente de venta en México. 12. El usuario puede modificar la fecha de inicio. 13. El usuario puede modificar la fecha final. 14. El sistema crea la consulta necesaria para la obtención de los datos. 15. El sistema ejecuta la consulta. 16. El sistema presenta una gráfica de barras con los totales de artículos y órdenes vendidos por agente de venta en México. 	
Ruta alternativa		
Post-condición		
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro		

Figura A-25 Caso de uso "Total de artículos y órdenes vendidos por agente de venta en México".

Anexo B

Implementación de la capa de datos.

El modelo relacional (figura B-1 a B-4), es tomado de una sección de la base de datos ERP, muestra las entidades y las relaciones que permiten, el almacenamiento y administración del catálogo de productos así como su información más relevante.

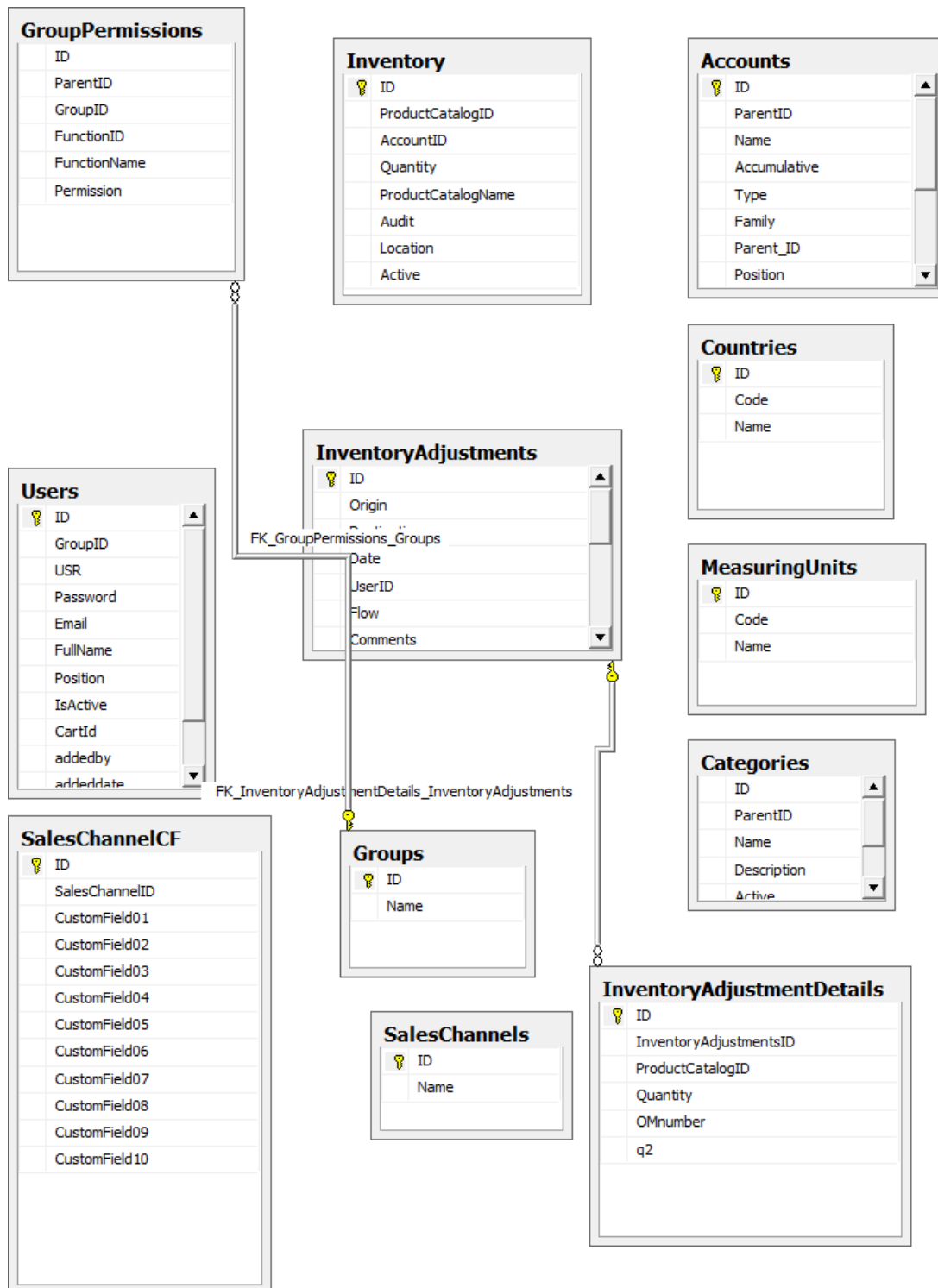


Figura B-1 Primera parte del modelo Relacional de la base de datos Inventory.

ProductCatalog			
Column Name	Data Type	Length	
ID	int	4	
OldSKU	varchar(10)	10	
Manufacturer	varchar(250)	250	
ManufacturerPN	varchar(250)	250	
CategoryID	int	4	
ProductLineID	int	4	
MeasuringUnitCode	varchar(5)	5	
AssemblyRequired	bit	1	
Serialized	bit	1	
NextSerialValue	int	4	
Name	varchar(254)	254	
Description	text	16	
Keywords	text	16	
IsSellable	bit	1	
CountryOfOrigin	varchar(2)	2	
Condition	varchar(20)	20	
InventoryMin	int	4	
InventoryMax	int	4	
LastModifiedUserID	int	4	
CustomField01	varchar(50)	50	
CustomField02	text	16	
CustomField03	text	16	
CustomField04	text	16	
CustomField05	text	16	
CustomField06	text	16	
CustomField07	text	16	
CustomField08	text	16	
CustomField09	text	16	
CustomField10	text	16	
CustomField11	text	16	
CustomField12	text	16	
CustomField13	text	16	
CustomField14	text	16	
CustomField15	text	16	
CustomField16	text	16	
CustomField17	text	16	
CustomField18	text	16	
CustomField19	text	16	
CustomField20	text	16	
UnitCost	real	4	
UnitDim	varchar(15)	15	
UnitWeight	real	4	
CaseQty	real	4	
CaseDim	varchar(15)	15	

ProductLines	
ID	
Name	
CustomField01	
CustomField02	
CustomField03	
CustomField04	
CustomField05	
CustomField06	
CustomField07	
CustomField08	
CustomField09	
CustomField10	
CustomField11	
CustomField12	
CustomField13	
CustomField14	
CustomField15	
CustomField16	
CustomField17	
CustomField18	
CustomField19	
CustomField20	
CompatibilityCF01	
CompatibilityCF02	
CompatibilityCF03	
CompatibilityCF04	
CompatibilityCF05	
CompatibilityCF06	
CompatibilityCF07	
CompatibilityCF08	
CompatibilityCF09	
CompatibilityCF10	
CompatibilityCF11	
CompatibilityCF12	
CompatibilityCF13	
CompatibilityCF14	
CompatibilityCF15	
CompatibilityCF16	
CompatibilityCF17	
CompatibilityCF18	
CompatibilityCF19	
CompatibilityCF20	

Figura B-2 Segunda parte del modelo Relacional de la base de datos Inventory.

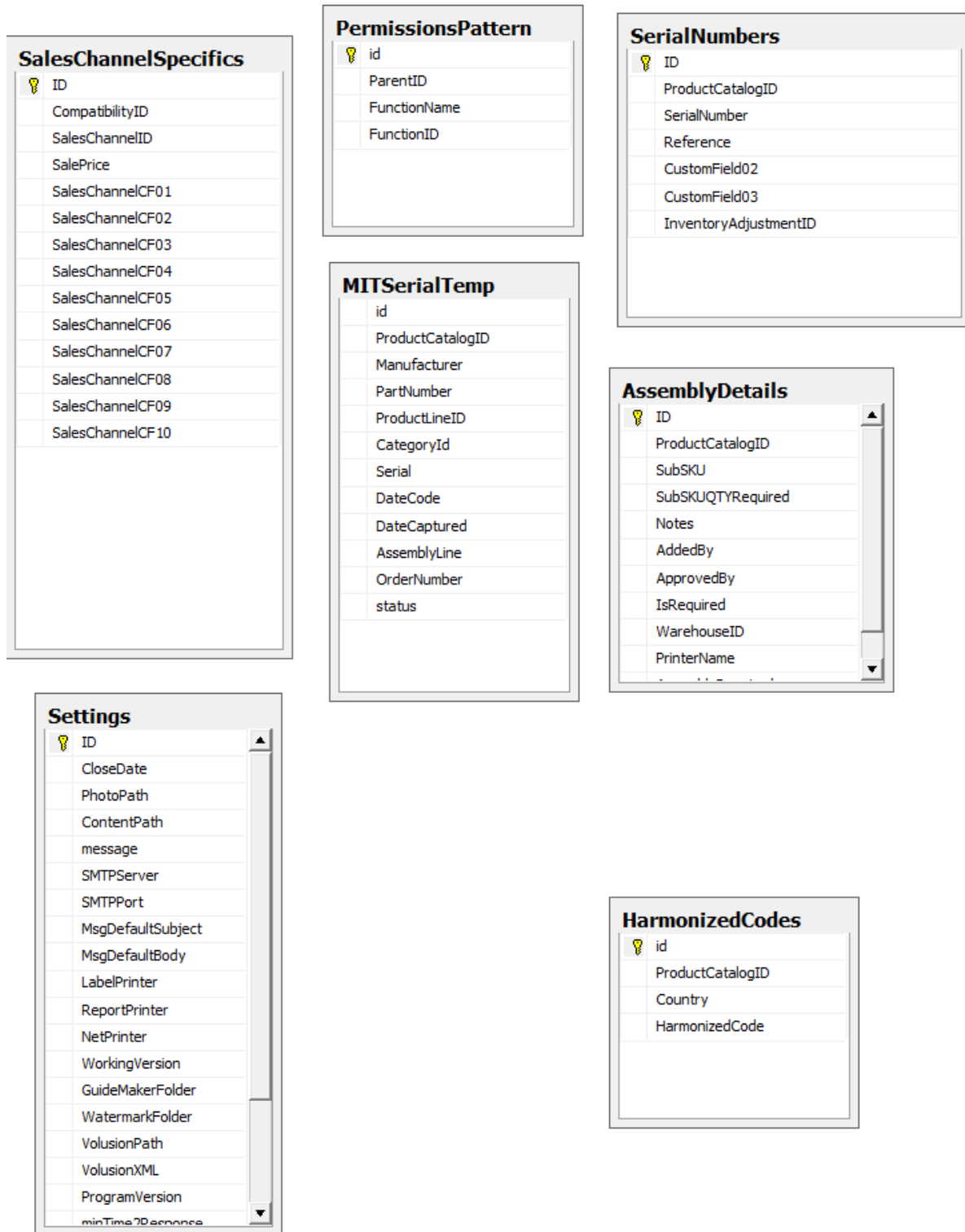


Figura B-3 Tercera parte del modelo Relacional de la base de datos Inventory.

Compatibility		
Column Name	Data Type	Allow ...
id	int	<input type="checkbox"/>
ProductCatalogID	int	<input type="checkbox"/>
PartNumber	varchar(50)	<input type="checkbox"/>
Manufacturer	varchar(50)	<input type="checkbox"/>
Comments	text	<input checked="" type="checkbox"/>
AddedBy	int	<input checked="" type="checkbox"/>
ApprovedBy	int	<input type="checkbox"/>
Publisher	int	<input type="checkbox"/>
html	text	<input checked="" type="checkbox"/>
xml	text	<input checked="" type="checkbox"/>
additional	text	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

CompatibilityDetails		
Column Name	Data Type	Allow ...
id	int	<input type="checkbox"/>
PartNumber	varchar(50)	<input type="checkbox"/>
Model	varchar(30)	<input checked="" type="checkbox"/>
AddedBy	int	<input checked="" type="checkbox"/>
Comments	text	<input checked="" type="checkbox"/>
ApprovedBy	int	<input checked="" type="checkbox"/>
OriginalManufa...	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

CompatibilityAlternativeModels		
Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
OriginalModel	varchar(50)	<input type="checkbox"/>
AlternativeModel	varchar(50)	<input type="checkbox"/>
Comments	text	<input type="checkbox"/>
AddedBy	int	<input type="checkbox"/>
ApprovedBy	int	<input type="checkbox"/>
OriginalManufacturer	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

CompatibilityAlternativePN		
Column Name	Data Type	All...
id	int	<input type="checkbox"/>
CompatibilityID	int	<input type="checkbox"/>
AlternativePN	varchar(50)	<input type="checkbox"/>
Comments	text	<input checked="" type="checkbox"/>
PartNumber	varchar(50)	<input checked="" type="checkbox"/>
ApprovedBy	int	<input checked="" type="checkbox"/>
AddedBy	int	<input checked="" type="checkbox"/>
OriginalManufacturer	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Figura B-4 Cuarta parte del modelo Relacional de la base de datos Inventory.

Implementación de la lógica de control.

Desarrollo de la lógica de control (figuras B-5 a B-8), implementada para el caso de uso "Consulta al Catálogo de productos".

Caso de Uso	Consulta al Catálogo de productos.	Caso de uso Numero: 2
Disparador	El usuario realiza consulta al <i>Catálogo</i> de productos.	
Precondición	Cualquier usuario valido puede tener acceso a esta sección	
Ruta básica	<ol style="list-style-type: none"> 1. El sistema presenta un formulario de búsqueda o con los campos necesarios para realizar consultas por palabra o frase, por categoría, por cantidad ordenada, por <i>DOIR (Days Of Inventory Remaining)</i> o combinaciones de todas estas. 2. El usuario puede digitar una palabra o una frase que considere para la búsqueda. 3. El usuario puede seleccionar una categoría. 4. El usuario puede seleccionar ver los artículos que tienen una Cantidad Ordenada. 5. El usuario puede seleccionar <i>DOIR</i>. 6. El sistema procesa la consulta general. 7. El sistema muestra los resultados en el <i>Grid</i> con campos <i>producto, VendorPN, Manufacturer, Name, QOH, DOIR, ExcessDays, UnitCost, TotalCost, QtyOrdered, Stats</i>. 	
Ruta alternativa	<p>En el paso 2 Si el usuario ingresa una frase:</p> <ol style="list-style-type: none"> (3) El sistema separa la frase en palabras (4) El sistema modifica la consulta global e incluir las condiciones necesarias para la búsqueda de cada palabra en todos los campos de la <i>Tabla Inventory</i>. (5) Enviar el control al paso 6. <p>En el paso 3, Si el usuario selecciona consulta por categorías:</p> <ol style="list-style-type: none"> (4) El sistema crea y presenta una lista con todas las categorías de productos en la base de datos. (5) El usuario selecciona la categoría. (6) El sistema modifica la consulta general integrando las condiciones para la búsqueda por la categoría seleccionada. (7) Retorna al paso 6. <p>En el paso 4 Si el usuario selecciona artículos con cantidad ordenada:</p> <ol style="list-style-type: none"> (4) El sistema modifica la consulta general integrando las condiciones necesarias para realizar el filtro de los registros que tengan un valor en el campo <i>QtyOrdered</i>. (5) Retorna el control al paso 6. <p>En el paso 5, Si el usuario selecciona <i>DOIR</i>:</p> <ol style="list-style-type: none"> (6) El sistema agrega un campo de forma automática al formulario para indicar los días en los que se basa este cálculo, por default tendrá 30. (7) El usuario puede modificar el número de días de base para el cálculo del <i>DOIR</i>. (8) El sistema modifica la consulta general agregando el resultado del procedimiento almacenado <i>DOIR</i> en el campo <i>DOIR</i> y el campo <i>ExcessDays</i>. (9) Retorna el control al paso 6. 	
Post-condición	<ol style="list-style-type: none"> 5. El <i>Grid</i> debe permitir editar los campos <i>TargetDays, QtyOrdered, ETA</i> y actualizar los datos automáticamente. 6. Si <i>DOIR</i> esta seleccionado el <i>Grid</i> debe colorear la columna, el campo se colorea en ROJO si $DOIR \leq TargetDays$ en caso contrario se colorea en VERDE. 	
Excepción	El usuario puede abandonar el sistema en el momento que quiera.	
Otro	El <i>Grid</i> Principal debe permitir exportar el contenido a formato de Excel.	

Figura B-5 Desarrollo del caso de uso "Consulta al catálogo de productos".

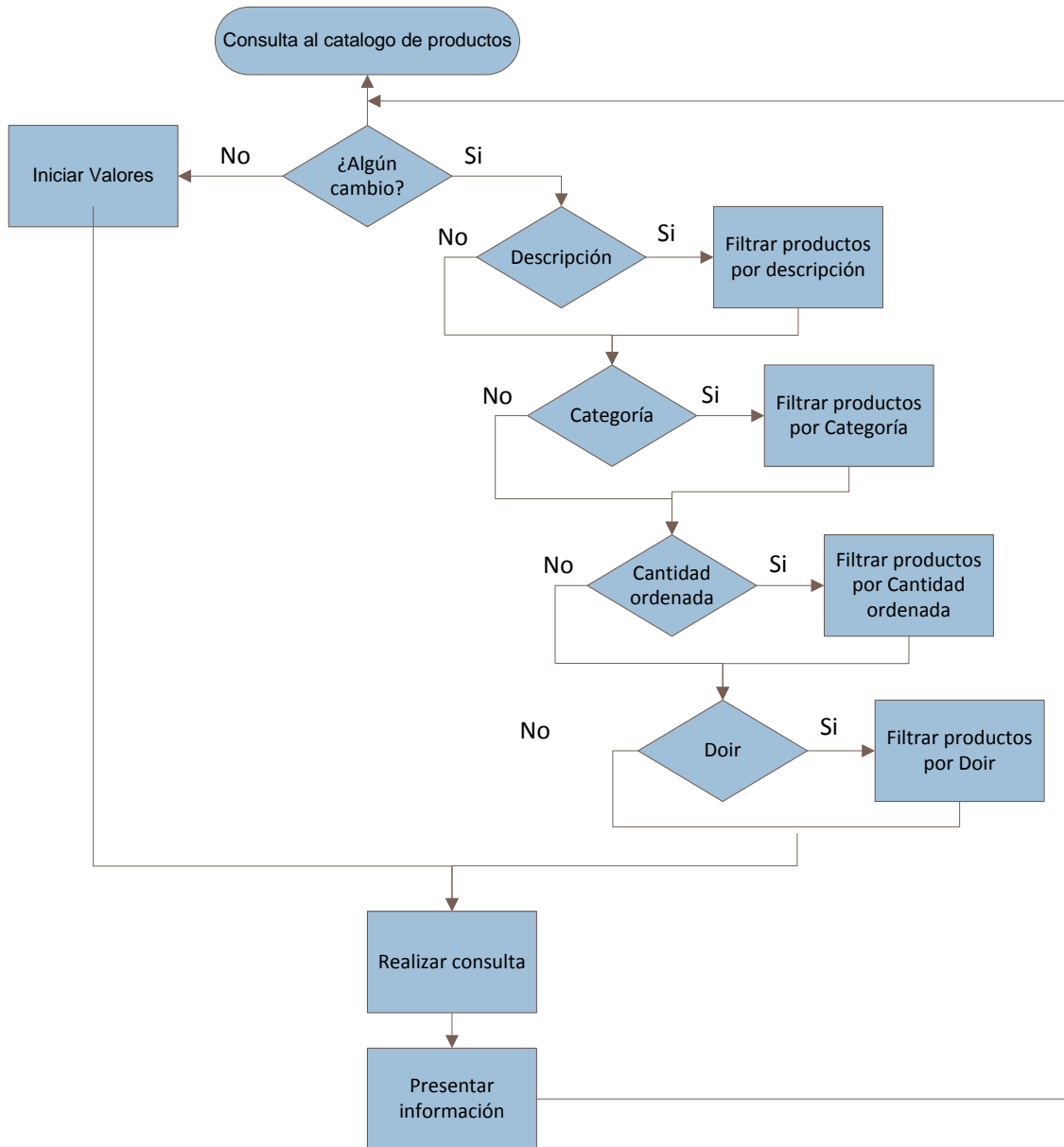


Figura B-6 Diagrama de actividad "Consulta al catálogo de productos".

Implementación de la interface de usuario.

Presentación del diseño de la vista para el caso de uso “Catálogo de productos”, tanto en fase de diseño (figura B-7), como en fase de desarrollo (figura B-8).

Catalog Orders Reports Tools Settings Company Dir Dashboard Logout			
Search <input type="text"/>		All <input type="text"/>	
<input type="checkbox"/> Quantity Ordered	<input type="checkbox"/> DOIR	<input type="button" value="Submit"/>	<input type="button" value="Reset"/>
Name (job title)	Age	Nickname	Employee
Giacomo Guizzoni Founder & CEO	34	Peldi	<input checked="" type="checkbox"/>
Guido Jack Guizzoni	4	The Guids	<input type="checkbox"/>
Marco Botton Tuttofare	31		<input checked="" type="checkbox"/>
Mariah MacLachlan Better Half	35	Patata	<input checked="" type="checkbox"/>
Valerie Liberty COO, WOW! Division		:) Val	<input checked="" type="checkbox"/>

Figura B-7 Vista del caso de uso “Consulta al catálogo de productos” en fase de diseño.

SKU	Vendor PN	Manufacturer	Name	QOH	vQOH	Target (Days)	Qty Ordered	ETA	Stats
101001	PHI/390	PHILIPS	Philips E22 100/120W 1.0 Bare Lamp	238	0	10	144	11/15	
101002	PHI/389	PHILIPS	Philips E22 120/132W 1.0 Bare Lamp	641	0	10	552	11/15	
101003	PHI/390	PHILIPS	Philips E22 100/120W 1.0 Bare Lamp + Lamp Adapte	9	0	10			
101004	PHI/389	PHILIPS	Philips E22 120/132W 1.0 Bare Lamp + Lamp Adapte	0	0	10			
101005	PHI/378	PHILIPS	Philips E23 100/120W 1.0 Bare Lamp	221	0	10	96	11/15	
101006	PHI/387	PHILIPS	Philips P22 100/120W 1.0 Bare Lamp	778	0	10	672	11/15	
101007	PHI/388	PHILIPS	Philips P22 120/132W 1.0 Bare Lamp	242	0	10	192	11/15	
101008	PHI/385	PHILIPS	Philips P23 100W 1.3 Bare Lamp	165	0	10	24	11/15	
101009	PHI/379	PHILIPS	Philips E19.8 100/120W 1.0 Bare Lamp	242	0	10	192	11/15	
101010	PHI/669	PHILIPS	Philips P21 110W 1.0 Bare Lamp	168	0	10	168	11/15	
101011	PHI/670	PHILIPS	Philips E22 132/150W 1.0 Bare Lamp	159	0	10	72	11/15	
101012	PHI/684	PHILIPS	Philips E22 180W/160W 1.0 AC Bare Lamp	113	0	10	96	11/15	
101014	PHI/334	PHILIPS	Philips E22 160/180W 1.0 3KV AC Bare Lamp	489	0	10			
101015	PHI/662	PHILIPS	Philips E23 100/120W 1.3 Bare Lamp	82	0	10	48	11/15	
102001	69561-1	OSRAM	OSRAM E22 100/120W 1.0 Bare Lamp	0	0	10			
102002	69534-1	OSRAM	OSRAM E22 100/132W 1.0 Bare Lamp	0	0	10			
102003	69561-1	OSRAM	OSRAM E22 100/120W 1.0 Bare Lamp + Lamp Adaptr	0	0	10			
102004	69534-1	OSRAM	OSRAM E22 120/132W 1.0 Bare Lamp + Lamp Adaptr	0	0	10			
102005	69490-1	OSRAM	OSRAM E22 132/150W 1.0 Bare Lamp	91	0	10			
102007	69560-1	OSRAM	OSRAM E23h 100/120W 1.0 Bare Lamp (15kv)	0	0	10			
102008	69559-0	OSRAM	OSRAM E23 100/120W 1.3 Bare Lamp	0	0	10	50	ORDERED 11-15	
102009	69554-1	OSRAM	OSRAM E23 132/150W 1.0 Bare Lamp	102	0	10			
102010	69458-1	OSRAM	OSRAM P22 100/120W 1.0 Bare Lamp	0	0	10			
102011	69441-1	OSRAM	OSRAM P22 120/132W 1.0 Bare Lamp	12	0	10			
102012	69570-1	OSRAM	OSRAM P23 100/120W 1.3 Bare Lamp	1	0	10			
102013	69440-1	OSRAM	OSRAM E22 150/180W 1.0 Bare Lamp	18	0	10			

Figura B-8 Vista del caso de uso "Consulta al catálogo de productos" en fase de desarrollo.

Integración de sistema y pruebas.

Las pruebas realizadas a cada caso de uso se desarrollan como casos de prueba (Figura B-9), destinados a validar que todas las funciones integradas en el caso uso, funcionen correctamente.

Caso de prueba	Consulta al <i>Catálogo</i> de productos	Caso de Prueba Numero: 1
Propósito	Realizar consultas al <i>Catálogo</i> de productos	
Prerrequisitos	El usuario debe existir en la aplicación ERP y debe tener una sesión activa y acceso a esta área.	
Datos de prueba	<i>Search</i> = {Cadena, palabra, vacío} <i>Category</i> = {All, Elemento seleccionado} <i>QtyOrdered</i> = {verdadero, falso} <i>DOIR</i> = {verdadero, falso}	
Pasos	Pasos a ejecutar la prueba. <ol style="list-style-type: none"> 1. Teclar cadena de búsqueda, palabra o dejar vacío. 2. Presionar enviar 3. Observar el número de registros encontrados. 4. Seleccionar categoría. 5. Observar que los registros correspondan a la categoría seleccionada 6. Seleccionar <i>QtyOrdered</i>. 7. Observar que todos los registros en la columna <i>QtyOrdered</i> no estén vacíos 8. Seleccionar <i>DOIR</i> 9. Observar que se añaden las columnas <i>DOIR</i> y <i>ExcessDays</i> 10. Realizar combinaciones de todas las opciones anteriores 11. Seleccionar los campos <i>TargetDays</i>, <i>QtyOrdered</i>, <i>ETA</i> , hacer clic en ellos y modificar el contenido y hacer clic fuera del <i>Grid</i>, el campo debe actualizar el valor 	
Notas y preguntas		

Figura B-9 Caso de prueba "Consulta al Catálogo de productos".

Revisión y retroalimentación

El análisis de resultados (Tabla B-1), muestra que todos los requerimientos se ha cumplido en el desarrollo del caso de uso "Consulta al Catálogo de productos".

Tabla B-1 Análisis de resultados del proceso de revisión y retroalimentación.

Requerimiento	Cumplido	Observaciones
Requerimientos funcionales		
• Casos de uso	SI	
Otros Requerimientos funcionales.		
• Interface de usuario	SI	
• Interface de Hardware	SI	
• Interface de Software	SI	
• Patrón de desarrollo	SI	
Requerimientos no funcionales		
• Requerimiento de desempeño	SI	
• Requerimientos de seguridad	SI	

- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Pub Co.
- Castledine, E., & Sharkie, C. (2010). *Jquery Novice to Ninja*. Collingwood: SitePoint ptu. Ltd.
- Comverse, T., Park, J., & Morgan, C. (2004). *PHP5 and Mysql Bible*. Indianapolis, IN: Wiley Publishing, Inc.
- Cunningham, W. (2001). *Manifesto for Agile Software Development*. Retrieved 03 20, 2012, from Manifesto for Agile Software Development: <http://www.agilemanifesto.org/>
- Eriksson, H. E., & Penker, M. (2000). *Business Modeling with UML: Business Patterns at Work*. John Wiley & Sons, Inc.
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y JavaScript*. Barcelona Es: Marcombo,S.A.
- Holzner, S., & Keogh, J. (2009). *Ajax. A Beginner's Gide*. McGraw-Hill.
- Hotek, M. (2009). *Microsofr SQL Server 2008 Step by Step*. Redmond, Washington: Microsoft Press.
- Kniberg, H. (2004). *Scrum y XP desde las trincheras*. C4Media Inc.
- Kniberg, H., & Skaring, M. (2010). *Kanban y Scrum - Obteniendo lo mejor de ambos*. C4Media Inc.
- Laurent, S. S., Johnston, J., & Dumbill, E. (2001). *Programming Web Services with XML-RPC*. O'Reilly. First Edition.
- Lawson, B., & Sharp, R. (2011). *Introduction HTML5*. Berkeley, CA: New Riders.
- Ltd, Z. T. (2010). *zend.com*. Retrieved 11 20, 2012, from <http://www.zend.com/en/>
- Meyer, E. A. (2006). *Cascading Style Sheets The definitive guide*. O'Really.
- Myer, T. (2008). *Profesional CodeIgniter*. Indianapolis, IN.: Wiley Publishing, Inc.
- Newkirk, J., & Martin, R. C. (2002). *La Programación Extrema en la práctica*. Madrid Esp.: Addison-Wesley Iberoamericana.
- Oktaba, A., Alquicira, C., Su, A., Marinez, A., & Quintanilla, G. (2005). *Modelo de Procesos para Industria de Software*. Secretaria de Economia -Mexico.
- Ostenwalder, A., Pigneur, Y., & Tucci, C. (2005). Clarifying Business Models: Origins, Present, and Future of the Concept. *Communications of AIS, Volume 15,*.
- Palacio, J. (2008). *ScrumManager: Gestión de proyectos* . Safe Creative.
- Palacio, J. (2008). *Flexibilidad con Scrum*. Creative Commons.
- Palacio, J., & Ruata, C. (2009). *Scrum Manager Proyectos - Apuntes de formacion*. Safe Creative.
- Pantoja, E. B. (n.d.). <http://www.ucbcba.edu.bo/>. Retrieved 11 02, 2012, from <http://www.ucbcba.edu.bo/Publicaciones/revistas/actanova/documentos/v2n4/v2.n4.bascon.pdf>
- Perez, J. E. (2009). *Introduccion a JavaScript*. Creative Commons.
- Sanchez, J. (2004). *Principios sobre Bases de Datos Relacionales*. Stanford, CA: Creative Commons.
- Team, C. (2011). *CodeIgniter, Guía de Usuario en Español para la Versión 2*. Autoedición.
- Thau, D. (2007). *The Book of JAVASCRIPT 2nd Edition*. San Fracisco, CA.: No Starch Press.

Referencias Bibliográficas.

w3c. (2012, 05 22). <http://www.w3c.es>. Retrieved 11 07, 2012, from <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>

Weeling, L., & Thomson, L. (2005). *PHP and MySQL Web Development 3rd Edition*. Pearson Education, Inc.