

Universidad de las Ciencias Informáticas

Título: Implementación del módulo Distribución de productos.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Daymel Machado Cabrera

Tutor: Ing.Tte. Pedro Julio Rodríguez Paredes

Co-Tutor: Ing. Eduardo Octavio León Alen

Ciudad de la Habana. Enero de 2010

“Año del 52 Aniversario del Triunfo de la Revolución”

FACULTAD 1



Declaración de autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

<Nombre autor>

<nombre tutor>

Firma del Autor

Firma del Tutor

Datos de Contacto

Tutor: Ing.Tte. Pedro Julio Rodríguez Paredes

UCID, Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: pjrodriguez@uci.cu

Agradecimientos

A mis padres, a mi hermana y a mi familia por el apoyo, el amor y los consejos que me han brindado para poder llegar a donde estoy.

A mi novia Dayanis y su familia por estar a mi lado, dándome confianza y apoyo siempre que la necesité.

A mi tutor por toda la ayuda brindada en estos meses de trabajo.

A mis amigos que siempre han estado ahí para ayudarme siempre que tuve un problema.

A todas las personas que de una forma u otra han contribuido en mi desarrollo profesional.

A Fidel, a Raúl y la Revolución

A todos gracias

¡Muchas Gracias!

Dedicatoria

A mis padres, a mi hermana, a mis familiares y a mi novia Dayanis y su familia: por todos los valores y consejos que me han brindado y por haber confiado en mí todos estos años.

Resumen

La Universidad de las Ciencias Informáticas tiene entre sus principales metas apoyar el proyecto de informatización del país, así como desarrollar la industria del software para contribuir al desarrollo económico del mismo. Para ello se ha iniciado un proceso de automatización de los servicios y actividades que realizan algunas de las empresas encargadas de la distribución de productos en Cuba. Actualmente la situación general de los procesos de gestión de distribución de productos que realizan las entidades presupuestadas y empresariales a escala nacional está afectada por la existencia de sistemas informáticos de software propietario, que no cumplen con la política de Cuba de migrar todos los sistemas informáticos a software libre, para así proteger los recursos de las empresas, y llegar a la exactitud y confiabilidad de la información económica y contable. A raíz de esto se hizo necesario elaborar un software que cumpliera con estas características para mejorar el proceso de distribución de productos, utilizando plataformas confiables y eficientes, donde de esta manera surge el módulo de Distribución de productos del Sistema Integral de Gestión CEDRUX.

Con el desarrollo de éste módulo se deberá lograr optimizar el proceso de distribución de productos, con ello disminuir los costos totales de las operaciones que se realizan actualmente con software propietario, ya que representa una pérdida para el país. Una vez finalizado el módulo se ratifica la solución propuesta a partir de la utilización pruebas de caja negra, para verificar que el sistema funcione de manera correcta.

Palabras claves: informatización, distribución de productos, recursos, gestión de distribución.

Índice

Resumen	5
Introducción.....	8
Capítulo 1. Fundamentación Teórica.....	12
1.1 Introducción	12
1.2 Conceptos Generales	12
1.2.1 ¿Qué es un ERP?.....	12
1.2.2 ¿Qué es un Sistema de Gestión de Distribución?	13
1.3 Sistemas automatizados existentes.	14
1.3.1 Sistemas Internacionales.	14
1.3.2 Sistemas Utilizados en Cuba.....	21
1.4 Proceso de Desarrollo y Gestión de Proyectos de Software.....	22
1.5 Herramientas y tecnologías utilizadas	25
1.5.1 Herramientas CASE	25
1.5.2 Base de datos	27
1.5.3 IDE de desarrollo	28
1.6 Tecnologías Web.....	29
1.6.1 Lenguajes de programación del lado del cliente.	29
1.6.2 Lenguaje de programación del lado del servidor.....	32
1.6.3 Tipo de arquitectura.....	33
1.6.4 Frameworks	37
1.6.5 Navegador.....	42
1.7 Control de Versiones.....	43
1.9 Conclusiones	44
Capítulo 2: Características del Sistema.....	45

2.1	Introducción	45
2.2	Flujo de procesos.	45
2.3	Objeto de automatización.	47
2.4	Información que se maneja.	48
2.5	Propuesta de sistema.	48
2.6	Especificación de los requisitos de software.	49
2.6.1	Requerimientos Funcionales.	49
2.7	Conclusiones Parciales.	52
Capítulo 3: Implementación y prueba.		53
3.1	Introducción	53
3.2	Implementación.....	53
3.2.1	Diagrama de Componentes.....	53
3.2.1.1	Matriz de integración.....	54
3.3	Pruebas.....	55
3.3.1	Pruebas de caja negra	56
3.3.1	Diseños de casos de prueba	57
3.4	Conclusiones.....	60
Conclusiones generales		61
Recomendaciones		62
Bibliografía.....		63
Referencias Bibliográficas		64
Anexos		68
Glosario de Términos		68

Introducción.

En la actualidad, el desarrollo de los sistemas informáticos para la gestión empresarial, está muy ligado al desarrollo de la sociedad. Esto es gracias a la generalización de las nuevas tecnologías de información y las comunicaciones (T.I.C.). Estos sistemas nombrados ERP (Planificación de Recursos Empresariales) se han logrado vincular a las principales ramas de la sociedad como lo son la salud, el deporte, la economía, la educación, entre otros. Una de las empresas que se ha beneficiado con este desarrollo informático, son las empresas encargadas de la distribución de productos.

En el siglo pasado este tipo de empresa experimentó un gran desarrollo, de tal forma que a nivel mundial se convirtieron en un negocio fabuloso y se concentran principalmente en potencias como Estados Unidos, Europa y Asia. Esto se debe a que los principales procesos que se llevan a cabo en estas empresas se encuentran informatizados. Cuba no se encuentra fuera de este desarrollo, pero las empresas cubanas no tienen semejanzas con la del mundo, pues sus fines no son de rentabilidad, sino que resuelvan las necesidades del pueblo.

Para que los productos y, más importante, sus beneficios, lleguen al pueblo es preciso que existan métodos de distribución adecuados. La planificación de la distribución de los productos que circulan en Cuba se efectúa de forma manual y a través de software propietarios, lo cual no brinda una solución completa a las actividades planificadas y reguladas para garantizar que la calidad de los productos se mantenga durante las etapas de recepción, almacenamiento, transportación y dispensación, lo que trae consigo que se realice un ineficiente proceso de distribución de productos. De ahí la necesidad de crear un sistema propio, capaz de gestionar los procesos de distribución.

Debido a la importancia que representa la gestión de distribución de productos para las empresas cubanas, el equipo de desarrollo debe entender el negocio al cual se va a enfrentar y conocer con detalles las funcionalidades que debe realizar el sistema a implementar.

Este trabajo se centra precisamente en implementar los requisitos obtenidos por el equipo de análisis para el Módulo de Distribución por lo que se puede identificar como **problema** a resolver:

¿Cómo llevar a cabo la automatización de los requisitos obtenidos por el equipo de análisis, para el desarrollo del proceso de Distribución?

Por lo anteriormente expuesto se definió el **objeto de estudio** los Procesos de cadena de distribución logística y como **campo de acción** los Procesos de distribución de productos sin pedidos.

Para solucionar el problema planteado se propone como **Objetivo General**: Implementar las funcionalidades para el módulo de distribución del Sistema Integral de Gestión de Entidades CedruX.

Para darle cumplimiento al objetivo general se plantearon los siguientes **objetivos específicos**:

- Realizar diseño del marco teórico de la investigación.
- Diseñar diagrama de componentes a partir del análisis de los componentes que intervienen.
- Realizar la matriz de integración de componentes interna.
- Implementar el componente Configuración.
- Implementar el componente Distribución sin pedidos.
- Implementar las interfaces a partir del prototipo entregado por los analistas.
- Implementar las validaciones y excepciones.
- Implementar los servicios incluidos dentro de sus responsabilidades que se necesiten para la implementación de otros módulos y componentes.
- Validar el sistema a través de pruebas de aceptación.
- Realizar la integración del subsistema.

Para la realización de esta investigación se ha planteado la siguiente **idea a defender**:

Si se realiza la implementación de las funcionalidades para el módulo distribución, se obtendrá un producto funcional que permitirá una mejor organización, planificación y control del proceso de distribución de productos.

Como **posibles resultados** se esperan:

Una herramienta que gestione la información referente al proceso de distribución con pedidos y sin pedidos de forma automatizada, a partir de los requerimientos analizados y priorizados por los usuarios funcionales del mismo.

Los **métodos teóricos** sustentan la investigación son:

Histórico-Lógico: Su empleo permitió el desarrollo evolutivo y coherente en el estudio de la metodología orientada a objetos, patrones de diseño, herramientas de desarrollo de software, sistemas ERP y sistemas de distribución para el desarrollo de los artefactos que proponen los flujos estudiados.

Analítico-Sintético: Permitted integrar y descomponer el conocimiento, descubriendo las relaciones para la utilización de artefactos propuestos por constituir este método una unidad dialéctica, determinando los aspectos esenciales y el arribo a conclusiones prácticas y teóricas.

Modelación: Su utilización permitió crear abstracciones que explican la realidad, por ejemplo, todos los modelos y diagramas presentados.

El **aporte práctico** esperado del trabajo de diploma es: La obtención del módulo distribución.

El presente documento se encuentra conformado por tres capítulos que incluyen todo lo relacionado con el trabajo investigativo y con la implementación del módulo.

En el **Capítulo 1** se muestra el estado del arte, mostrando los sistemas vinculados al campo de acción. Al mismo tiempo, se describe el objeto de estudio. También se detallan tendencias y tecnologías actuales

utilizadas para el desarrollo del subsistema y el motivo de su utilización. Se realiza un análisis del modelo de desarrollo utilizado, así como de la herramienta utilizada para el desarrollo del Módulo Distribución.

En el **Capítulo 2** se realizará una valoración crítica del diseño propuesto por los analistas, las posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados, estrategias de integración, se realizará la descripción de las principales clases que se implementaron.

En el **Capítulo 3** se realizará la validación de la solución propuesta, esto incluye el diseño, descripción teniendo los objetivos, alcance, tipo y detalles de las pruebas, además se hará una descripción de los valores utilizados para los mismos, se evaluará su ejecución y los resultados obtenidos.

Capítulo 1. Fundamentación Teórica

1.1 Introducción

En estos tiempos el ambiente competitivo del mundo empresarial se ha incrementado indudablemente, debido a que las entidades requieren promover procesos y entidades de negocio que brinden ventajas ante la competencia o que simplemente proporcionen ahorro y eficiencia económica. Hoy más que nunca se hace necesaria la utilización de herramientas que proporcionen un mejor control y centralización de la información con el fin de tomar las mejores decisiones para los procesos y estrategias del negocio.

En el resto del mundo se han desarrollado software relacionados con la Gestión de distribución de productos, cuyo estudio ha servido para tener en cuenta las ventajas y defectos que tienen. A menudo, surgen dudas sobre cuál usar, debido a que actualmente aparecen nuevas herramientas con mejoras, que se van imponiendo para el desarrollo de software. Es por esto, que surge la necesidad de hacer un estudio de los sistemas existentes y las herramientas a utilizar, buscando obtener la información más actual de las mismas.

1.2 Conceptos Generales

1.2.1 ¿Qué es un ERP?

El ERP es un sistema integral de gestión empresarial que está diseñado para modelar y automatizar la mayoría de procesos en la empresa (área de finanzas, comercial, logística, producción, etc.). Su misión es facilitar la planificación de todos los recursos de la empresa (2).

Por otro lado, definen al Enterprise Resource Planning (ERP) como "paquetes de sistemas configurables de información dentro de los cuales se integra la información a través de áreas funcionales de la organización". Estos sistemas son extremadamente costosos, y una vez que se implantan con éxito trae una serie de beneficios importantes para las empresas. (5)

También se definen a los sistemas de planificación de recursos empresariales (ERP) como un sistema que permite coleccionar y consolidar la información a través de la Empresa. (6)

Los ERP son una evolución de los sistemas MRP (Planificación de Recursos Materiales), los cuales estaban enfocados únicamente a la planificación de materiales y capacidades productivas. Los ERP disponen de herramientas para efectuar la planificación de los trabajos en planta. Esta planificación se efectúa enfrentando los requerimientos de materiales y capacidad de los productos a fabricar contra las existencias y capacidades sin asignar. Los ERP más completos ofrecen módulos para planificar a capacidad finita. Son el núcleo de otras aplicaciones como pueden ser el CRM (Gestión de las relaciones con los clientes), Data Mining (Conversión de datos en información útil), etc. (2)

En resumen, los sistemas ERP unifican información de las diferentes áreas (finanzas, recursos humanos, ventas, manufacturación, etc.) de la empresa en un solo lugar, haciendo más fácil la toma de decisiones dentro de la empresa. El software planea y automatiza muchos procesos con la meta de integrar información a lo largo de la empresa y elimina los complejos enlaces entre los sistemas de las diferentes áreas del negocio.

1.2.2 ¿Qué es un Sistema de Gestión de Distribución?

Los Sistemas de Gestión de Distribución permiten a las empresas que distribuyen productos gestionar de manera eficiente y eficaz la red de distribución, y mantener un estricto control de los productos que se están distribuyendo.

1.2.3 Situación actual de los procesos.

En la actualidad en Cuba no existe una herramienta automatizada capaz de cumplir con los requerimientos de funcionalidad que le den una solución completa a los procesos de distribución, de manera que pueda ser utilizado como una herramienta para mejorar el cumplimiento de las tareas, en las empresas que se ocupan de la distribución de productos en el país.

1.3 Sistemas automatizados existentes.

Actualmente en el mundo existen gran variedad de sistemas para la gestión de distribución pero la mayoría pertenecen a softwares propietarios que en ocasiones no satisfacen las necesidades que solicitan nuestras empresas. A continuación, mostraremos algunos de estos softwares y sus principales características.

1.3.1 Sistemas Internacionales.

Axos Visual

Axos Visual es una herramienta de gestión integral pensada y desarrollada para la pequeña, mediana y gran empresa. Axos Visual es una solución empresarial capaz de gestionar diversas las áreas de negocio de la empresa de forma integrada, y mejorar la eficiencia de los recursos de la misma, incorporando la última tecnología.

Se ha desarrollado desde la filosofía del dato único es decir, toda la información que existe en el seno de la organización es única, vista desde los prismas de los diferentes usuarios o departamentos. El cliente siempre es único, independientemente que se analice desde el departamento comercial, financiero o logístico. Cada operador tiene acceso a la parte de la información que precisa. No existe duplicidad de datos ni tediosos procesos de intercambio de información / trasposos entre los diferentes módulos de la aplicación.

Su fácil manejo e intuitivo diseño del interfaz de trabajo, permiten al usuario adaptarse de forma rápida y cómoda a su utilización. La agilidad y funcionalidad de Axos Visual cubre el 100% de las necesidades de las empresas dentro de sectores de distribución y servicios, permitiendo al usuario la realización de complejos procesos con un simple “clic” del ratón. Es decir, la gran variedad de procesos ejecutables desde el sistema, se adaptan y parametrizan para que la aplicación y funcionalidades sean completamente transparentes al usuario, indistintamente de la cualificación o perfil que este posea.

La plataforma cliente servidor en la cual está desarrollado Axos Visual permite una perfecta conectividad entre los diferentes centros de trabajo, independientemente de la ubicación geográfica de los mismos, y permite a cualquier usuario trabajar a cualquier hora del día, desde cualquier parte del mundo, disponiendo siempre de toda la información en tiempo real.

Características y funcionalidades

- Control constante de los stocks¹ en sus distintos estados: inventariable (o contable), real, disponible y futuro.
- Rápido acceso a la información referente a stocks en la generación de todos los documentos.
- Multialmacén y multiubicación con stock mínimo manual o bajo punto de pedido (automático) para la generación automática de propuestas de compra por almacén.
- Generación de automática de propuestas traspaso de mercancía entre almacenes.
- Función Composición/Fabricación: para el control de stocks de componentes y artículos compuestos.
- Generación de repartos por repartidor/agencia o rutas de reparto en base a carga de trabajo.
- Estandarización de costes de envío según códigos postales o importe mínimo de pedidos.
- Repercusión de Costes logísticos de Importación.
- Generación de inventarios totales o rotativos. A valor actual o a una fecha determinada.

¹ **Stocks:** se usa con el sentido de existencias o reserva de algún producto disponible para un uso futuro.

Características generales

Los requerimientos de hardware son mínimos, pudiéndose utilizar en casi cualquier equipo, los sistemas operativos soportados son según la necesidad del cliente. (8).

Cumulus Software

Cumulus Software para Control de Almacenes, Pedidos y Distribución es una solución poderosa y económica para Almacenadoras, Centros de Distribución y empresas que manejan su propio almacén.

Cumulus cuenta con 2 versiones que le ayudarán a maximizar la eficacia y productividad en Almacenes y Centros de Distribución. Estas versiones son la Estándar y la Profesional con Radiofrecuencia (WiFi). Ambas versiones constan de funciones básicas:

- **Control de Inventarios**
- **Administración de Inventarios,**
- **Control de Pedidos**
- **Control de Distribución,**
- **Administración del Espacio**
- **Recibo de Productos**
- **Surtido de Ordenes.**

La versión Profesional, además le incluye funciones como Mejora Continua en el Nivel de Servicio de su Operación (Medición de la Confiabilidad del Inventario vs Cumulus, % de Pedidos Bien Surtidos, % de Pedidos Cargados Correctamente en el Vehículo Asignado y Tiempos de Surtido, Transporte y Entrega de Evidencias), Áreas dedicadas del Almacén por Cliente, Línea de Surtido, Administración de Back order y Re-abasto de Áreas de Surtido. (9)

Características generales

Los requerimientos de hardware son Pentium II, 256 Mb de RAM, Pantalla de 15", Resolución mínima 1024x768, 16-bit color video, mouse, 25 Mb de espacio en disco duro para Cumulus y tarjeta de red Ethernet 10/100, los sistemas operativos soportados son Windows 98 ó XP.

JR SOFTWARE

JR SOFTWARE para distribuidoras de alimentos, sistemas para distribuidores de alimentos, programas para distribuidoras de productos alimenticios, distribución de alimentos, software de distribución de productos alimenticios, software para la distribución de artículos alimenticios, sistema para venta de alimentos, software para la toma de pedidos de alimentos, distribución de bebidas y comidas, Software distribuidor de alimentos.

Características y funcionalidades

- **Alícuotas de IVA:** registra la alícuota de I.V.A. vigente en su país.
- **Vendedores:** registra los datos de los vendedores o preventistas.
- **Vehículos:** se deben cargar los datos de los vehículos que repartes los pedidos.
- **Fleteros:** se ingresan los datos personales de los fleteros o choferes.
- **Zonas:** se deben cargar las zonas hasta donde llegan todos los repartos.
- **Precios (Nombres):** breve descripción del precio para hacer referencia al tipo de precio que utilizará en sus ventas, se permiten ingresar hasta 5 precios diferentes por artículo.
- **Envases:** cargar todos los tipos de envases en que se presentan sus productos, por ejemplo (Pack, Caja, Bidones, Cajón, Bultos, etc.).

- **Rubros:** se deben cargar los distintos rubros en los cuales encuadran sus artículos, por ejemplo (Comestibles, Lácteos y Quesos, Bebidas varias, etc.).
- **Marcas:** ingresar las distintas marcas de los productos.
- **Lista de Precios:** debe cargar los artículos con sus respectivos precios, se permiten hasta 5 precios por artículos.
- **Ajustes:** se utiliza para ajustar stock, ingresa o saca mercadería de depósito detallando el motivo.
- **Clientes:** registra los datos de los clientes, a cada cliente deberá asignarle una zona y un vendedor.
- **Cuenta Corriente:** muestra el estado de cuentas de los clientes y general.
- **Pedidos:** registra los pedidos de los clientes.
- **Repartos:** genera el reparto mostrando una hoja de ruta y la planilla de carga, esta última agrupa los artículos a cargar en cada vehículo, por ejemplo si en 3 pedidos hay 2 del art.1, 3 del art. 1 y 1 del art. 1 muestra art.1 cantidad a cargar....6, etc. Por vehículo mostrando un total de bultos y el cargamento valorizado.
- **Recibos:** emisión de recibos que afectan a la cuenta corriente del cliente.
- **Plan de Cuentas:** plan de cuentas que se utiliza para asignar una cuenta a las compras y gastos.
- **Proveedores:** registra los datos de los proveedores.
- **Compras:** se deben ingresar los comprobantes de compras, automáticamente incrementará su stock y actualizará sus precios si el costo de la mercadería es distinto del anterior.
- **Cuenta Corriente Proveedor:** estado de cuenta corriente de sus proveedores y general.

- **Pago a Proveedores:** emisión de órdenes de pagos a proveedores.
- **Sub-Diario I.V.A. Compras:** libro de I.V.A. compras.
- **Cartera de Cheques:** muestra los cheques en cartera.
- **Cuentas Bancarias:** registra los movimientos de sus cuentas bancarias.
- **Caja Diaria:** muestra la caja del día.
- **Ingreso directo a Caja:** ingresa dinero en caja por pago de clientes o ventas contado.
- **Egreso directo a Caja:** saca dinero de caja que no sea por pago a proveedores o compras contado.
- **Configuración:** módulo especialmente para configurar el sistema. Aquí también se encuentran los módulos necesarios para su configuración en red local si usted adquirió licencias adicionales.
- **Usuarios:** alta de los usuarios del sistema bajo claves.

Características generales

Los requerimientos de hardware son mínimos, pudiéndose utilizar en casi cualquier equipo, los sistemas operativos soportados son Windows® 98, 98Se, ME, XP, 2000, 2003, Windows® Vista y Windows® 7 (10)

EcoSoftCS.net

ecoSoftCS.net es un producto especialmente diseñado para facilitar todas las labores de introducción de datos, haciendo posible que con el menor número de pasos a realizar pueda cargar su pedido y partiendo de éste, terminar el ciclo de compra, gestión de stock y etiquetado al completo. Por supuesto, contempla todas las posibilidades de facturación de proveedores, tales como agrupación de albaranes, de abono de mercancía, efectos comerciales a pagar. Podrá exportar toda esta información a su contabilidad

automáticamente. Igualmente dispone de las mismas herramientas en facturación de ventas, además de sistema de clonación de documentos, pedidos a proveedores en función del stock y presupuestos aceptados, consultas de precios, stock por almacenes.

Genera las normativas bancarias más utilizadas para evitar trabajar en varios programas, con ecoSoftCS.net puede realizar todas las operaciones de su empresa. Además, puede incorporar procesos a medida diseñados especialmente para su empresa de forma automática y totalmente transparente para los usuarios que se encuentran trabajando en la aplicación en el momento de este tipo de actualizaciones. Sin duda un gran adelanto para adaptar el programa a las necesidades cambiantes de su empresa.

Características y funcionalidades

- **Control de inventario**
- **Sistema de ubicaciones**
- **Gestión de compras**
- **Gestión de precios y tarifas**
- **Envío de mercancía**
- **Control del stock**
- **Configuración**

Características generales

Los requerimientos de hardware son mínimos, pudiéndose utilizar en casi cualquier equipo, los sistemas operativos soportados son Windows® 98, 98Se, ME, XP, 2000, 2003, Windows® Vista y Windows® 7.
(11)

1.3.2 Sistemas Utilizados en Cuba.

Stock Mistral

El objetivo fundamental es la disminución de los niveles de stock para reducir los recursos inmovilizados e incrementar la liquidez en la empresa. Es una herramienta imprescindible para facilitar la toma de decisiones. Para entidades con alcance nacional, posee herramientas de comunicaciones que crean, en tiempo real, almacenes virtuales a diferentes niveles, con las existencias y movimientos de sus dependencias, permitiendo organizar el proceso de compras, distribución y pedidos. La consulta de esta información se realiza sobre sitios web ajustándose a los procedimientos internos de cada cliente.

Se le incorporaron nuevas prestaciones como el completo control de la consignación comercial, generación de ofertas, conciliación con proveedores, así como el riguroso tratamiento de los lotes y sus vencimientos (caducidad), logrando su total aplicación para productos perecederos como alimentos y medicamentos.

Valoración Crítica

Desde el punto de vista de la solución estos sistemas podrían ser capaces de resolver el problema del proceso de distribución de productos debido a su alto nivel de configuración y servicios que proveen, pero tienen la desventaja de que algunos de ellos utilizan tecnologías que no son accesibles a Cuba debido a las restricciones impuestas por Estados Unidos. Como otra desventaja aparece que el diseño de estos Sistemas de Gestión ha sido para empresas capitalistas que tienen un modelo de gestión y de procesos muy diferentes de las empresas cubanas donde la economía es centralizada y operan otros mecanismos. Por último, todos estos softwares propietarios no constituyen una opción viable pues representan gastos muy elevados al país por conceptos de licencias y mantenimiento. El sistema utilizado en Cuba, se concluye que no resulta una solución factible para las entidades cubanas debido a que fue desarrollado sobre plataforma de software propietario, es un producto que se caracteriza por abordar solamente parte del problema del proceso de distribución de productos de la empresa o la unidad presupuestada.

1.4 Proceso de Desarrollo y Gestión de Proyectos de Software.

Diversas tendencias y metodologías de desarrollo de software han aparecido en años recientes, buscando resolver los problemas que proyectos más tradicionales, no han conseguido enfrentar. Entre ellas están los frameworks de proyectos, las metodologías ágiles y los modelos de medición de madurez. Junto con estos marcos de trabajo, ciertas estrategias específicas han permitido a los equipos de desarrollo producir software más robusto, predecible, reutilizable y de fácil mantenimiento. (33)

1.4.1 Fases del Ciclo de vida.

Inicio: Se logra una visión preliminar de la problemática a resolver, se identifica el alcance preliminar del proyecto, se especifican los involucrados y las líneas de desarrollo ejecutoras del proyecto constituyéndose el equipo de desarrollo y se estiman los recursos necesarios que deberán ser asignados al mismo.

Modelación: Se definen los procesos del dominio del problema, se estiman los principales riesgos que presenta el proyecto y se especifica la forma de mitigarlos, se identifican las necesidades del usuario de las que se derivan los requerimientos del producto a desarrollar, se determina la factibilidad operativa, técnica y/o económica de continuar el proyecto. Se define la arquitectura del software, se valida y establece para disponer de cimientos sólidos sobre los que se basará el grueso del esfuerzo durante la fase de Construcción. Se realiza la planificación detallada de la siguiente fase.

Los principales hitos de esta fase son:

- Análisis de las necesidades del usuario.
- Especificación de la Arquitectura.
- Planificación de la construcción del software.

Construcción: En esta fase todas las características, componentes y requerimientos deben ser integrados, implementados y probados en su totalidad, obteniendo una versión estable del producto comúnmente llamada versión beta. Se realiza la planificación detallada de la siguiente fase.

Los principales hitos de esta fase son:

- Versión Funcional del producto (Versión Beta).
- Manuales de usuario y de instalación.
- Liberación de calidad del producto.
- Planeación de la Explotación Experimental.

Explotación experimental: Incluye las pruebas del producto como parte de su preparación para ser entregado, y la realización de ajustes en respuesta a la retroalimentación recibida de los usuarios. En este punto del ciclo de vida la retroalimentación de los usuarios debe enfocarse fundamentalmente en ajustes específicos y de corto alcance al producto junto a otros temas como configuración, instalación, y usabilidad.

Los principales hitos de esta fase son:

- Solución Estable.
- Planeamiento para el despliegue.
- Aceptación de los resultados.

Despliegue: Se realiza la generalización del producto en las entidades y órganos según lo aprobado en el Cronograma de implantación. Durante el proceso de implantación por lo general no es necesaria la participación de los integrantes del equipo de desarrollo.



Figura 1. Esquema de las fases del Proceso de Desarrollo y Gestión de Proyectos de Software.

1.4.2 Modelo de desarrollo de software

El modelo de desarrollo de software propuesto describe la secuencia de actividades de alto nivel para la construcción y desarrollo de soluciones. Se logra con la combinación entre los modelos basado en Componentes, el Iterativo y el Incremental. Se emplearán las técnicas de prototipo, si son requeridas, para los requerimientos del usuario de los que no existe una visión clara por parte de estos, con el objetivo de desarrollar una definición mejorada de los requisitos del usuario para el sistema.

Desarrollo iterativo e incremental: Es un enfoque en el que el ciclo de vida está compuesto por iteraciones, estas son pequeños procesos compuestos de varias actividades cuyo objetivo es entregar una parte del sistema parcialmente completo, probado, integrado y estable. Todo el software es integrado en cada entrega de cada iteración hasta obtener el producto de software completo en la última iteración. En cada iteración se obtiene como resultado un incremento.

Desarrollo basado en componentes: Nos lleva a alcanzar un mayor nivel de reutilización de software, aún en contextos distintos de aquellos para los que fue diseñado. Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados. Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema. Dado que un componente puede ser construido y luego mejorado continuamente, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

1.5 Herramientas y tecnologías utilizadas

1.5.1 Herramientas CASE

Las herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Sistema de software que intenta proporcionar ayuda automatizada a las actividades del proceso de software. Los sistemas CASE a menudo se utilizan como apoyo al método. (34)

Visual Paradigm for UML 6.1

Visual Paradigm para UML² es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un

² **UML:** (Unified Modeling Language) Lenguaje Unificado de Modelado

menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Visual Paradigm es una herramienta CASE creada para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas entre otras opciones. Constituye una herramienta de software libre de gran utilidad para el analista. Dentro de sus características se aprecia que soporta BPMN y UML versión 2.1. Muestra también (15).

- Diagramas de Procesos de Negocio.
- Modelado colaborativo con CVS y Subversión.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Dibujo de diagramas UML con plantillas (stencils) de MS Visio.
- Editor de figuras.

UML (Lenguaje Unificado de Modelado)

UML (Unified Modeling Lenguaje) o Lenguaje Unificado de Modelado es un lenguaje basado en una notación gráfica la cual permite: especificar, construir, visualizar y documentar los objetos de un sistema programado. UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones, ha sido ampliamente aceptado debido al prestigio de sus creadores. Hay que tener en cuenta que el estándar UML no es un proceso, no es una metodología de desarrollo, sino una notación, un lenguaje (16).

De forma general las principales características son:

- Lenguaje unificado para la modelación de sistemas.

- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.
- Facilita a los integrantes de un equipo multidisciplinario participar e intercomunicarse fácilmente, estos integrantes siendo los analistas, diseñadores, especialistas de área y desde luego los programadores.

1.5.2 Base de datos

Una base de datos es 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.

PostgreSQL 8.3

PostgreSQL es un sistema de gestión de bases de datos, objeto-relacional (ORDBMS), basado en el proyecto POSTGRES de la universidad de Berkeley. Está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo, publicado bajo la licencia BSD.

A continuación se enumeran las principales características de este gestor de bases de datos:

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora una estructura de datos array.

- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta queries complejos, incluyendo subselects, integridad referencial (Foreign Keys), triggers, vistas (Views), integridad transaccional (ACID), control de versionado concurrente (MVCC). (17)

1.5.3 Herramientas de base de datos

PgAdmin III

PgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL entre ellas: Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres. Incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP³ o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL⁴ para mayor seguridad (18).

1.5.3 IDE de desarrollo

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de *integrated development environment*), es un programa informático compuesto por un conjunto de herramientas de programación o puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. (35)

³ **TCP/IP:** (Transmission Control Protocol/Internet Protocol) Protocolo de control de transmisión/Protocolo de Internet

⁴ **SSL:** (Secure Sockets Layer) Protocolo de Capa de Conexión Segura

Zend Studio for Eclipse

Zend Studio para Eclipse posee las capacidades para crear poderosas herramientas web. Permite a los desarrolladores crear estrategias de integración más eficientes. Al proporcionar potentes capacidades de acción con el lenguaje PHP, mejoras de soporte con JavaScript y profunda integración a Zend Framework, el desarrollo de aplicaciones se realiza en un tiempo récord. Incluye un poderoso editor de código con soporte para todos los formatos web. Posee un fuerte manejo de la arquitectura Cliente/Servidor, excelente depuración, elaboración de perfiles y un código de cobertura. Zend estudio tiene todas las herramientas que un desarrollador necesita para asegurarse de que el código es correcto y empezar a diagnosticar problemas con antelación. Tiene características de depuración avanzadas, incluyendo: condiciones límites, visualización de errores, variables y buffer de salida. Asegura la protección máxima de ubicaciones de proyectos o en Internet con depuradores remotos (19).

1.6 Tecnologías Web.

Las tecnologías web son un conjunto de herramientas que nos facilitarán lograr mejores resultados a la hora del desarrollo de un sitio web.

1.6.1 Lenguajes de programación del lado del cliente.

HTML

No es más que un conjunto de etiquetas o comandos, complementados en la mayoría de los casos por extensiones que permiten dar formato a un archivo, con el objetivo básico de crear un documento que pueda ser visualizado en ambiente Internet en forma de Página Web y que esta, además, pueda, por medio de dichas etiquetas, tener la estructura o forma deseada por quien la diseñó. Las etiquetas HTML⁵ tienen, por lo general, una etiqueta de apertura y una de cierre; aunque existen algunas excepciones en las que solo basta con colocar la de apertura. (20)

⁵ **HTML:** (Hyper Text Markup Language) Lenguaje de Marcas de Hipertexto

JavaScript

JavaScript es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web, puede ser utilizado por profesionales y para 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. (21).

Ajax

AJAX, acrónimo de Asynchronous JavaScript And XML⁶ (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma, es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

AJAX es una combinación de tres tecnologías ya existentes:

XHTML (o HTML) y hojas de estilos en cascada (CSS⁷) para el diseño que acompaña a la información.

DOM⁸ accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y Script, para mostrar e interactuar dinámicamente con la información presentada.

El objeto XMLHttpRequest⁹ para intercambiar datos asincrónicamente con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.

⁶ **XML:** (Extensible Markup Language) Lenguaje de Etiquetado Extensible

⁷ **CSS:** (Cascading Style Sheets) Hojas de Estilo en Cascada

⁸ **DOM:** (Document Object Model) Modelo en Objetos para la representación de Documentos

XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML pre formateado, texto plano y JSON.

AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente (22).

Json

JSON ¹⁰ es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto se conoce como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

El formato JSON es el más adecuado para la respuesta del servidor cuando la acción Ajax debe devolver una estructura de datos a la página que realizó la llamada de forma que se pueda procesar con JavaScript. Este mecanismo es útil por ejemplo cuando una sola petición Ajax debe actualizar varios elementos en la página.

⁹ **XMLHttpRequest:** Interfaz empleada para realizar peticiones HTTP y HTTPS a servidores WEB.

¹⁰ **JSON:** (JavaScript Object Notation) Notación de Objetos de JavaScript

JSON se ha convertido en un estándar en el desarrollo de aplicaciones web. Los servicios web proponen la utilización de JSON en vez de XML para permitir la integración de servicios en el navegador del usuario en vez de en el servidor. El formato JSON es seguramente la mejor opción para el intercambio de información entre el servidor y las funciones JavaScript (23).

CSS

CSS es la tecnología desarrollada por el World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación, permite crear páginas web de una manera más exacta, gracias a las CSS el desarrollador es mucho más dueño de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas. (24)

Entre los beneficios concretos de CSS encontramos:

- control de la presentación de muchos documentos desde una única hoja de estilo.
- control más preciso de la presentación.
- aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla, impresión, etc.).
- numerosas técnicas avanzadas y sofisticadas.

1.6.2 Lenguaje de programación del lado del servidor

PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar. Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web.

Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante. Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como MySQL, mSQL, Oracle, Informix, y ODBC¹¹, por ejemplo. Incluye funciones para el envío de correo electrónico, upload de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas: Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader), analizar código XML. Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes. (25)

1.6.3 Tipo de arquitectura

Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. (26)

Patrón Modelo-Vista-Controlador (MVC)

MVC es un patrón de diseño de arquitectura de software usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de los conceptos para que el desarrollo este estructurado de una mejor manera, facilitando la programación en diferentes capas de manera paralela e independiente. MVC sugiere la separación del software en 3 estratos.

Modelo: Es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros que puestos en un contexto del sistema proveen de información al usuario o a la aplicación misma.

¹¹ **ODBC:** (Open Database Connectivity) Estándar de acceso a Bases de datos.

Vista: Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación web la "Vista" es la página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones.

Controlador: Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario.

El **Modelo** es el responsable de:

Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.

Definir las reglas de negocio (la funcionalidad del sistema).

Llevar un registro de las vistas y controladores del sistema.

Si se encuentra ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

El **Controlador** es responsable de:

Recibir los eventos de entrada.

Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas.

Las **Vistas** son responsables de:

Recibir datos del modelo y lo muestra al usuario.

Tienen un registro de su controlador asociado (normalmente porque además lo instancia).

Pueden dar el servicio de "Actualización", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

En la siguiente figura se muestra la estructura del patrón Modelo-Vista-Controlador.

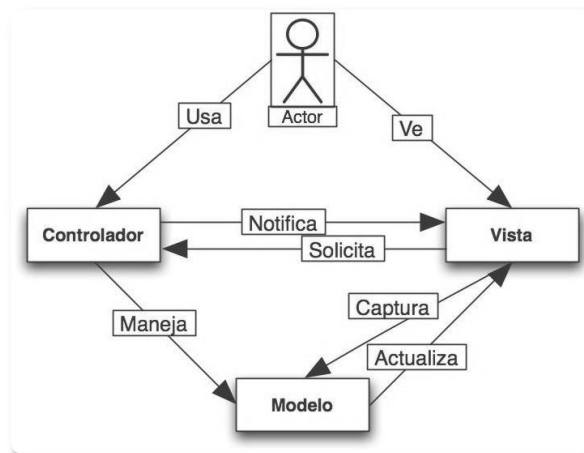


Figura 2. Estructura Modelo-Vista-Controlador

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual.

Ventajas:

Soporte de múltiples vistas: Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación web pueden utilizar el mismo modelo de objetos mostrado de maneras diferentes.

Adaptación al cambio: Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celulares o PDAs. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Desventaja:

Costo de actualizaciones frecuentes: Si el modelo experimenta cambios frecuentes, por ejemplo, podrían desbordar las vistas con una lluvia de requerimientos de actualización. (26)

Servidor Web Apache

Apache es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

Entre sus principales características se encuentran:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita de código fuente abierta. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber, sin ningún secreto, sin ninguna puerta trasera).
- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que los instalemos cuando los necesitemos. Otra cosa importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.

- Tiene una alta configurabilidad en la creación y gestión de logs¹². Este permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.(27)

1.6.4 Frameworks

Un framework es una estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. (28)

Zend Frameworks

Es un framework para desarrollo de aplicaciones Web y servicios Web con PHP. Brinda soluciones para construir sitios web modernos, robustos y seguros. Además, es código abierto y trabaja con PHP 5. (28)

Presenta entre otras las siguientes características:

- Simplifica la gestión de archivos de configuración.
- Proporciona los componentes que forma la infraestructura del patrón Modelo-Vista-Controlador.
- Proporciona una capa de acceso a base de datos, construida sobre PDO¹³ pero ampliándola con diferentes características.

¹² **Logs:** Registro de errores.

- Cuenta con módulos para manejar archivos PDF, canales RSS, Web Services (Amazon, Flickr, Yahoo)
- Proporciona mecanismos de filtrado y validación de entradas de datos.
- Clientes para servicios web, incluidos Google Data APIs y Strikelron.

En la siguiente figura se muestra la estructura de Zend Framework.

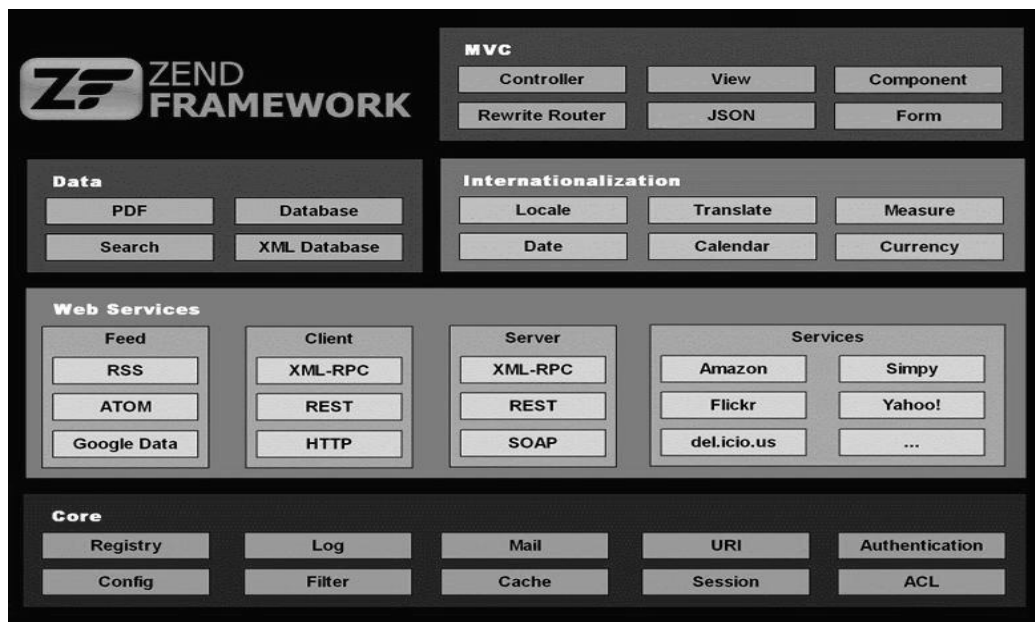


Figura 3. Estructura de Zend Framework.

Doctrine Framework

Doctrine es un ORM para PHP 5.2.3 y posterior. Además de todas las ventajas que conlleva un ORM, uno de sus puntos fuertes es su lenguaje DQL (Doctrine Query Language) inspirado en el HQL de Hibernate.

¹³ PDO: (PHP Data Objects) Capa de abstracción de acceso a datos para PHP

Cuando trabajamos con Doctrine, necesitamos informar a su motor interno de cual es el modelo de nuestra aplicación, para ello podemos hacer ingeniería inversa de nuestra base de datos existente, o si empezamos la aplicación desde 0, crear el modelo en la sintaxis específica que nos propone Doctrine y luego generar toda la base de datos.

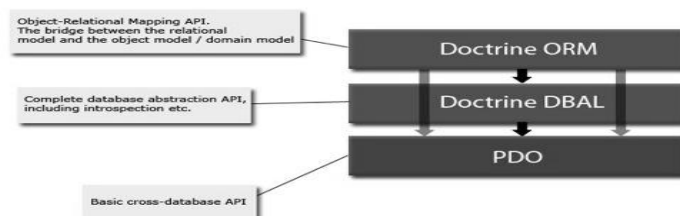


Figura 4 Estructura de Doctrine ORM.

Un ORM o (Object Relation Mapper) es una técnica de programación que nos permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de nuestra base de datos pasan a ser clases y los registros objetos que podemos manejar con facilidad.

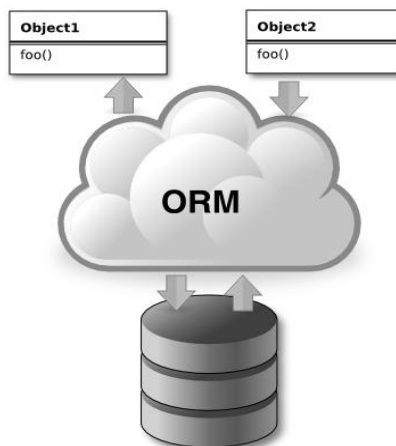


Figura 5 Estructura de ORM.

Utilizar un ORM tiene una serie de ventajas que nos facilitan enormemente tareas comunes y de mantenimiento:

- **Reutilización:** La principal ventaja que aporta un ORM es la reutilización permitiendo llamar a los métodos de un objeto de datos desde distintas partes de la aplicación e incluso desde diferentes aplicaciones.
- **Encapsulación:** La capa ORM encapsula la lógica de los datos pudiendo hacer cambios que afectan a toda la aplicación únicamente modificando una función.
- **Portabilidad:** Utilizar una capa de abstracción nos permite cambiar en mitad de un proyecto de una base de datos MySQL a una Oracle sin ningún tipo de complicación. Esto es debido a que no utilizamos una sintaxis MySQL, Oracle o SQLite para acceder a nuestro modelo, sino una sintaxis propia del ORM utilizado que es capaz de traducir a diferentes tipos de bases de datos.
- **Seguridad:** Los ORM suelen implementar mecanismos de seguridad que protegen nuestra aplicación de los ataques más comunes como SQL Injections.
- **Mantenimiento del código:** Gracias a la correcta ordenación de la capa de datos, modificar y mantener nuestro código es una tarea sencilla. (29).

ExtJs Framework

Es una librería construida con JavaScript que proporciona una interfaz a las famosas librerías de Yahoo, jQuery, y Prototype + Scriptaculous, su potencia radica en la rica colección de componentes para el diseño de GUI's del lado del cliente haciendo uso extensivo de Ajax.

ExtJS es neutral al lenguaje que se use en el servidor. Siempre que el resultado se envíe a la página en el formato adecuado, ExtJS no se preocupará de lo que pase en el servidor. Hay docenas de widgets a escoger en ExtJS, incluyendo composiciones automáticas de páginas, pestañas, menús, barras de herramientas, diálogos, vistas en árbol. Proporciona un selector de nodos DOM extremadamente

poderoso llamado DomQuery (puede usarse como una librería independiente, pero en el contexto de ExtJS se usará para seleccionar elementos para poder interactuar con ellos a través de la interfaz Element, contiene mucho de los métodos y propiedades de DOM que se necesitará proporcionando una interfaz conveniente, unificada y multinavegador).

Entre los componentes que esta librería ofrece encontramos cuadros de diálogo, menús, tablas editables, layouts, paneles, pestañas y todo lo necesario para construir atractivos desarrollos al estilo de Web 2.0.

Ventajas:

- La orientación a objetos intensa hará modular todos los scripts.
- El diseño está completamente separado de la funcionalidad.
- Funciones comunes como validación, comboboxes editables, ventanas arrastrables (con minimizar y maximizar), grillas editables, son muy fáciles de implementar.
- Buena y amplia documentación, así como también su comunidad.

Desventajas:

Crear un sistema serio con esta herramienta requiere un previo uso prolongado, ya que se dificulta el manejo de los nuevos objetos en su extensa y bien documentada API. El tiempo de aprendizaje puede llegar a compararse con aprender a programar en un lenguaje nuevo.

- Al estar todo el sitio en JS, no podrá ser accesible para los buscadores, limitando su uso a sistemas y no sitios web.
- Si se desea algún objeto y no está, se debe asumir la compleja tarea de crear un nuevo objeto (sólo apto para programadores JS avanzados). (30)

UCID Framework

Es el Framework encargado del trabajo con las vistas. Abarca la integración de ExtJs Framework con el sistema incluyendo el integrador de interfaz, el generador de interfaz dinámica y la impresión de documentos. Integra la iconografía, los diferentes temas de escritorio de la aplicación, el multilinguaje.

1.6.5 Navegador

Mozilla Firefox 3.0

Es el nuevo e innovador navegador open source acerca del cual todo el mundo habla. Firefox ha sido creado por el proyecto Mozilla, un esfuerzo open source sin ánimo de lucro que incluye a miles de voluntarios alrededor del mundo. La misión del proyecto Mozilla es preservar la elección y la innovación en Internet. El apoyo organizativo del proyecto Mozilla es proporcionado por Mozilla Foundation (en los Estados Unidos de América), Mozilla Europe y Mozilla Japón.

Por nombrar algunas de las posibilidades que ofrece Firefox y que no ofrece IE¹⁴, están:

- Es Software libre.
- En Firefox no existen la cantidad de bugs que posee el catastrófico IE, inmediatamente se encuentra un bug en el producto es notificado al Proyecto Mozilla para que sea reparado el problema.
- Navegación por tabs: Esta es una de las principales características que tiene Firefox.
- También existen excelentes extensiones de fácil instalación, estos mejoran la usabilidad y el aspecto del navegador, cosa que no se logra en IE el cual siempre permanece con los mismos colores.(31)

¹⁴ IE: Internet Explorer

1.7 Control de Versiones

Es la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas.

Subversion 1.4.5

Subversión es un controlador de versiones empleado en la administración de archivos utilizados en el desarrollo de software o contenido. CVS¹⁵ considerado su antecesor como uno de los controladores de versiones más utilizados en proyectos de software libre, sin embargo, a pesar de su amplio uso, el mismo diseño de CVS resultó ineficiente para diversos grupos de usuarios, y ante estas inconformidades se dio inicio al proyecto que hoy es conocido como Subversión, el mismo que ha empezado a socavar el dominio de CVS.

Ventajas

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
- La creación de ramas y etiquetas es una operación más eficiente; Tiene costo de complejidad constante ($O(1)$) y no lineal ($O(n)$) como en CVS.
- Se envían sólo las diferencias en ambas direcciones (en CVS siempre se envían al servidor archivos completos).
- Maneja eficientemente archivos binarios (a diferencia de CVS que los trata internamente como si fueran de texto).

¹⁵ CVS: Concurrent Versions System

- Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos (SQL, LDAP¹⁶, PAM¹⁷).
- Existen varias interfaces a Subversión, ya sea programas individuales como interfaces que lo integran en entornos de desarrollo.
- TortoiseSVN. Provee integración con el explorador de Windows. Es la interfaz más popular en este sistema operativo.
- Subclipse. "Plugin" que integra Subversión al entorno Eclipse.
- Subversive. "Plugin" alternativo para Eclipse.
- ViewVC. Interfaz web, que también trabaja delante de CVS.
- Para Mac, pueden emplearse los interfaces SvnX, RapidSVN y Zigversion.(32)

Valoración de las Herramientas y tecnologías utilizadas.

Estas herramientas fueron seleccionadas por el equipo de desarrollo, debido a que pertenecen a software libre, lo que proporciona libertad tecnológica a nivel colectivo y nacional. Lo que suministra al país una solución mucho más eficaz y segura, al menor costo monetario posible.

1.9 Conclusiones

En el presente capítulo se trata de forma resumida los diferentes sistemas que abarcan la gestión de distribución de productos, determinando las ventajas, desventajas y factibilidad de su utilización en las entidades cubanas; además se realizó un estudio de las herramientas, tecnologías y la metodología propuesta por la dirección de arquitectura del proyecto, permitiendo sentar las bases para el desarrollo.

¹⁶ **LDAP:** (Lightweight Directory Access Protocol) Protocolo Ligero de Acceso a Directorios

¹⁷ **PAM:** Pluggable Authentication Modules

Capítulo 2: Características del Sistema

2.1 Introducción

En este capítulo se realizará un estudio y análisis del problema y la situación problemática. Se identificarán los procesos que requieren de automatización para su descripción. Así como también estudiaremos los sistemas automatizados que poseen las distintas empresas, para trazar una propuesta del sistema y modelar los procesos de negocio especificando los requisitos del software y describiendo los requisitos funcionales y no funcionales.

2.2 Flujo de procesos.

2.2.1 Valoración crítica de los artefactos propuestos por los analistas.

La obtención de la especificación de requisitos propuesto por los analistas, resultó de gran importancia, pues permitió una mejor comprensión de los aspectos relacionados con los requisitos funcionales y componentes reutilizables. Se pretende mostrar cómo está aprovechada la arquitectura y las posibilidades que proporcionan los marcos de trabajo utilizados en la programación de la aplicación, con el objetivo de facilitar la comprensión del funcionamiento de los componentes implementados. Se muestra el diagrama de componentes propuesto por el equipo de arquitectura.

La principal fuente de información que brinda el analista es la especificación de requisitos donde estos no son más según la IEEE¹⁸ Standard Glossary of Software Engineering Terminology:

- Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
- Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

¹⁸ **IEEE:** (Institute of Electrical and Electronics Engineers) Instituto de Ingenieros Eléctricos y Electrónicos.

- Una representación documentada de una condición o capacidad.

El propósito de la especificación de requerimientos es reunir en un documento escrito los requisitos de todo el sistema de software o parte de él. Esto con la finalidad de plasmar qué es el sistema y cuál es su alcance.

Los requisitos se pueden clasificar en funcionales y no funcionales.

Requerimientos funcionales: Son capacidades o condiciones que el sistema debe cumplir.

Requerimientos no funcionales: Son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta, aunque no limitan a la definición de otros.

- Requerimientos de Software
- Requerimientos de Hardware
- Restricciones en el diseño y la implementación
- Requerimientos de apariencia o interfaz externa
- Requerimientos de Usabilidad
- Requerimientos de Soporte
- Requerimientos Legales
- Requerimientos de confiabilidad

- Requerimientos de interfaz Interna
- Requerimientos de Seguridad

2.3 Objeto de automatización.

El proceso de distribución de productos se realiza de dos formas, distribución de productos por pedidos unificados y distribución de productos sin pedido. La distribución por pedido se realiza cuando el cliente solicita que productos que desea que se le distribuyan, el ciclo o grupo de clientes a distribuir según su plan de consumo, se realiza una revisión de la existencia de los productos en el almacén y se realiza una propuesta de distribución, esta se envía al almacén, en caso de no existir los productos requeridos queda pendiente a distribuir y se decide si se le da seguimiento o no, en caso de existir los productos, se procede a realizar la distribución. En la distribución de productos sin pedidos se gestiona la distribución de cada uno de los productos seleccionados al azar según el plan de consumo de los clientes, se muestra la propuesta de distribución de productos para el circuito o grupo de clientes, y se realiza la propuesta de distribución, en caso de estar correcta se procede con la distribución, si no se repite nuevamente el proceso. En estos procesos de distribución intervienen una serie de fórmulas que ayudan a realizar estas operaciones.

Fórmulas utilizadas

Cobertura del cliente (cobertura del cliente = Existencia del producto en el cliente + Cantidad del producto en tránsito + Cantidad del producto facturado + Cantidad del producto distribuido) / Consumo)

Pendiente de distribución (Cantidad pendiente de distribución = Existencia en inventario - cantidad distribuida)

Cobertura (Cobertura = Cantidad distribuida al cliente / Consumo)

Cobertura total (Cobertura total = Cobertura + Cobertura del cliente)

Pendiente desglose (Pendiente desglose = cantidad distribuida - cantidad del lote seleccionado)

Días abastecidos para el cliente (Días abastecidos para el cliente = Existencia en el almacén del cliente / Plan de consumo)

Cantidad pendiente del pedido (Cantidad pendiente del pedido = Plan del período del cliente - Cantidad entregada del pedido - Cantidad en proceso del pedido)

Coeficiente de distribución (Coeficiente de distribución = Cantidad pendiente del pedido / \sum Cantidad pendiente del pedido, de todos los clientes del ciclo)

Cobertura del producto (Cobertura del producto = (Existencia en inventario / (Plan / días del periodo)))

Cobertura lote (Cobertura lote = Cantidad distribuida por lote / Consumo)

Cobertura total (Cobertura total = \sum Cobertura lote + Cobertura del cliente)

Resto de la cantidad distribuida (Resto de la cantidad distribuida = Cantidad distribuir - (capacidad del bulto * cantidad de bultos¹⁹)

Cantidad distribuida (Cantidad distribuida = Cantidad de bultos * Capacidad del bulto)

2.4 Información que se maneja.

- Plan de Distribución.
- Plan de Consumo.

2.5 Propuesta de sistema.

El sistema sea capaz de realizar los distintos procesos de distribución de productos que se llevan a cabo en el país. El usuario tendrá que loguearse en la aplicación, y el sistema verificara sus privilegios,

¹⁹ **Bulto:** Caja donde van los productos que se recibe del proveedor y se entregan al cliente.

permitiéndole acceder solo a las áreas que tenga acceso, así como las funcionalidades y acciones que le están permitidas según su rol. Para poder realizar el proceso de distribución de productos es necesario que se configure que tipo de distribución se va a realizar, distribución por pedidos unificados o distribución sin pedidos.

La distribución de productos sin pedidos, el sistema debe ser capaz de sin captar el pedido del cliente, poder realizar la propuesta de distribución. El sistema debe conocer del proveedor la existencia de productos que tiene en su almacén y la política de cobertura; de los clientes los planes de consumo o planes contratados, la cantidad de productos que se encuentran en tránsito y conocer la existencia de productos de los clientes, realizar los cálculos de las fórmulas que intervienen en la distribución de productos para luego realizar la propuesta de distribución de productos por ciclo o grupo de clientes, emitir un certificado de la distribución en proceso, realizar un despacho físico del producto, emitir una factura o transferencia y la guía de expedición.

El sistema debe permitir realizar, modificar, y eliminar las distribuciones. Debe ser capaz de listar todas las distribuciones realizadas hasta el momento, mostrar los productos de las distribuciones, certificar las distribuciones, imprimir y buscar distribuciones anteriormente realizadas.

2.6 Especificación de los requisitos de software.

2.6.1 Requerimientos Funcionales.

- RF1: Configurar tipo de distribución.

Se configura que tipo de distribución de productos se desea realizar, (distribución de productos por pedido unificado, distribución de productos sin pedido o ambas).

- RF2: Configurar la forma de distribuir.

Se configura la forma por la cual se va a distribuir, (prioridad de clientes, equitativa, mixta o decidir en el momento de la distribución de productos).

- RF3: Configurar si la entidad va a permitir modificar consumos y existencias de los productos que se importan desde los clientes.
- RF4: Configurar si la entidad va a permitir modificar los planes de máximos y mínimos de los productos de los clientes.
- RF5: Configurar si se va a dar seguimiento al Plan.
- RF6: Configurar si el seguimiento del plan va limitar o no la cantidad a distribuir propuesta por el sistema.
- RF7: Configurar si se va a dar seguimiento al Pedido.
- RF8: Configurar si la entidad va a reservar o no en las distribuciones para los demás circuitos.
- RF9: Configurar, en Distribución con Pedidos, si se va a distribuir a completar máximos o no.
- RF10 Configurar, en Distribución sin Pedidos, si se va a distribuir a completar políticas de coberturas o máximos.
- RF11: Configurar si en la distribución con Pedidos la entidad va a distribuir por Prioridad de clientes, Equitativa o Mixta. O si el usuario puede seleccionar una de las 3 a voluntad.
- RF12: Configurar en la distribución sin Pedidos que la entidad va a distribuir solamente de forma Equitativa para sus clientes. En una versión superior se incluirá la de por Prioridad y la Mixta.
- RF13: Adicionar distribución sin pedido.

2.6.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Apariencia o interfaz externa: El sistema debe contar con una interfaz fácil, amigable, sencilla, permitiendo que los usuarios finales del mismo sean capaces de interactuar con este aún teniendo conocimientos básicos. Se empleará imágenes y colores identificando la funcionalidad.

Usabilidad: La herramienta podrá ser usada por personas que cuenten con un conocimiento básico de informática. Se emplearán barras de progreso para indicar el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable por los usuarios. El software tendrá siempre visible la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de sus funcionalidades.

Rendimiento: Los tiempos de respuesta y velocidad de procesamiento de la información serán no tan rápidos, con una espera de 20 a 30 segundos para representar el mapa y de 10 a 20 segundos para la representación de la información.

Soporte: Se necesita un servidor de base de datos que soporte grandes volúmenes de información.

Portabilidad: La herramienta multiplataforma principalmente (Linux y Windows).

Políticos-culturales: La herramienta será utilizada en nuestro país y la podrán utilizar entidades autorizadas por el MINFAR.

Legales: El sistema debe registre por las leyes.

Confiabilidad: Debe existir mecanismos de restablecimiento de la herramienta en caso de fallos o problemas de comunicación. Deben instalarse sistemas de respaldo eléctrico en los servidores para el mantenimiento de los servicios.

Ayuda y documentación en línea: Estará disponible en cada una de las interfaces, la ayuda correspondiente a dicha interfaz para que así los usuarios tengan un conocimiento de las funcionalidades de la misma y puedan hacer un mejor uso de estas.

Requerimientos de software

Para el cliente: Navegador Mozilla Firefox y cualquiera de los sistemas operativos en los que se puede ejecutar Mozilla Firefox (Windows o Linux).

Para el servidor: Sistema operativo Windows o Linux. Un servidor WWW Apache v2.0 o superior con módulo PHP5 disponible, este debe estar configurado con las extensiones PDO y PDO_pgsql. Para servidor de base de datos PostgreSQL v4.1.0.1 o superior.

Requerimientos de hardware

Para el cliente: Requerimientos mínimos: Procesador Pentium IV a 3.00Ghz con 512 MB de memoria RAM. Tarjeta de red e impresora.

Para el servidor: Requerimientos mínimos: Procesador Pentium IV a 1GHz y 1Gb de memoria RAM. Tarjeta de red.

Restricciones en el diseño y la implementación: Como lenguaje del lado del cliente se usó el Frameworks EXTJS con JavaScripts y del lado del servidor PHP v5.0 o superior. Para el servidor de base de datos PostgreSQL v4.1.0.1 o superior y servidor web Apache v2.0 o superior.

2.7 Conclusiones Parciales.

En este capítulo se identificaron los principales procesos que serán objeto de automatización. Se describe la situación problemática para una perspectiva de cómo se encuentra la situación actual de los procesos. Se ha descrito la propuesta de solución para dar un enfoque general de los objetivos que el software debe cumplir, así como sus requisitos funcionales y no funcionales.

Capítulo 3: Implementación y prueba.

3.1 Introducción

En el presente capítulo, a través de la representación del diagrama de componentes y la matriz de integración se exhibe la interrelación interna de los componentes del sistema de manera general. Se desarrolla una búsqueda de las pruebas de aceptación que permitan validar la solución propuesta, se realiza una descripción del mismo teniendo en cuenta: tipo de pruebas, objetivo de la prueba y alcance, se describen los valores utilizados y se evalúa la ejecución de la prueba y los resultados obtenidos.

3.2 Implementación.

3.2.1 Diagrama de Componentes

Este diagrama representa la forma en que un sistema de software se divide en componentes mostrando así las dependencias entre dichos componentes. Los componentes físicos incluyen archivos, cabeceras, librerías compartidas, módulos, ejecutables y paquetes.

En las siguientes figuras se muestran los diagramas de componente del componente Distribución.



Figura 5. Paquete componente Distribución.

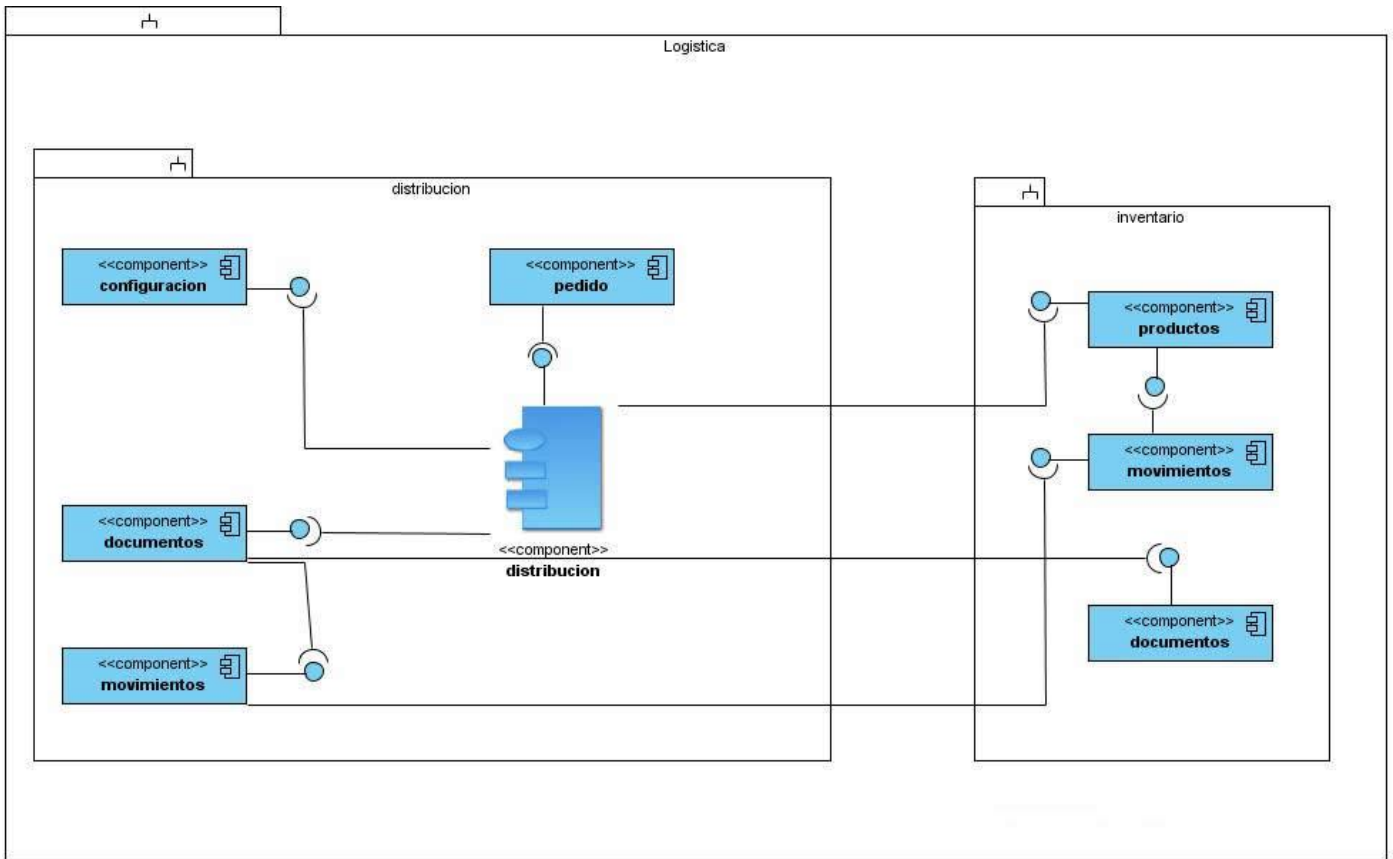


Figura 6. Diagrama del componente Distribución.

3.2.1.1 Matriz de integración

La matriz de integración de componentes contiene todos los componentes definidos en el subsistema, de forma matricial. En las intercepciones se especifican los servicios que consume el componente con respecto a su correspondiente. Existen dos matrices de integración de componentes, Interna y Externa.

Matriz de Integración de Componentes Interna.

Componentes Internos	Componentes Internos			
	Configuración(Distribución)	Documento	Movimiento	Producto (Inv.)
Distribución	devolverCircuitosClientes, devolverReserva, delvolverPrioridadPorCliente, salvarPrioridad, devolverConfPorEntidad, devolverPoliticadeCobertura, devolverstockmaxymin	adicionarDocumento, modificarDoc, devolverDocs, eliminarDoc	adicionarMovimientos, actualizarMov,eliminarMov	cargarProductos, devolverProductosPorDoc

Como se puede apreciar, la matriz muestra cómo se relacionan los componentes internos del sistema donde en la segunda fila se encuentran todos los componentes que brindan servicios o funcionalidades en la primera columna los componentes internos que consumen o precisan del servicio brindado.

3.3 Pruebas

Para poder validar el sistema se propone la realización de pruebas de aceptación. El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

Las pruebas de aceptación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al usuario.

La validación del sistema se consigue mediante la realización de pruebas de caja negra que demuestran la conformidad con los requisitos y que se recogen en el plan de pruebas, el cual define las verificaciones a realizar y los casos de prueba asociados. Dicho plan está diseñado para asegurar que se satisfacen todos los requisitos funcionales especificados por el usuario teniendo en cuenta también los requisitos no funcionales relacionados con el rendimiento, seguridad de acceso al sistema, a los datos y procesos, así como a los distintos recursos del sistema.

3.3.1 Pruebas de caja negra

Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software.

La prueba de caja negra se centra principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

La prueba de caja negra no es una alternativa a las técnicas de prueba de la caja blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la caja blanca, por lo que se puede decir, que estas no son excluyentes, sino complementarias.

Entre las principales características que presenta la prueba de caja negra están que:

- Verifican las especificaciones funcionales y no consideran la estructura interna del programa.
- Es hecha sin el conocimiento interno del producto.
- No validan funciones ocultas por tanto los errores asociados a ellas no serán encontrados.

En otras palabras, la prueba de la caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas se encuentran:

Técnica de la Partición de Equivalencia: Técnica que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica.

Técnica del Análisis de Valores Límites: El análisis de valores límite (AVL) es una técnica de diseño de casos de prueba que completa a la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la elección de casos de prueba en los extremos de la clase. En lugar de centrarse solamente en las condiciones de entrada, el AVL obtiene casos de prueba también para el campo de salida.

3.3.1 Diseños de casos de prueba

Los casos de prueba son un conjunto de condiciones o variables bajo las cuales el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

La prueba es una actividad en la cual, un sistema o componente, es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Configurar distribución	El requisito tiene 2 escenarios y es el encargado de configurar la de distribución de productos.	Cargar configuración	
		Guardar configuración	Guardar

Tabla 1 Diseños de caso de prueba Configurar distribución.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Gestionar Distribución	El requisito tiene 4 escenarios y es el encargado de gestionar la distribución de productos.	Listar distribución	
		Adicionar distribución	Aceptar, Aplicar y Cancelar
		Modificar distribución	Aceptar y Cancelar
		Eliminar distribución	Aceptar y Cancelar

Tabla 2 Diseños de caso de prueba Gestionar distribución.

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
1.1	Cargar configuración correctamente.	Se muestra la configuración para la el módulo de distribución.	Satisfactorio
1.2	Cargar configuración incorrectamente.	Se muestra un mensaje de error indicando que no se ha podido cargar la configuración.	Satisfactorio

Tabla 1 Juego de datos del escenario Cargar configuración.

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
2.1	Guardar configuración correctamente.	Se guardan los cambios y se muestra mensaje de confirmación.	Satisfactorio
2.2	Guardar configuración	Se muestra un mensaje de	Satisfactorio

	incorrectamente.	error indicando que no se ha podido guardar la configuración.	
--	------------------	---	--

Tabla 4 Juego de datos del escenario Guardar configuración.

Id del escenario	Escenario	Código-descripción	Observaciones	Respuesta del sistema	Resultado de la prueba
3.1	Adicionar documento de distribución.	V(nuevo)	V(en elaboración)	Se adiciona el documento y se muestra el mensaje de confirmación.	Satisfactorio
3.2	Adicionar documento de distribución incorrectamente	V(vacío)	V(en elaboración)	Se muestra un mensaje de error indicando que hay campos requeridos vacíos.	Satisfactorio
3.3	Cancelar operación.	NA	NA	El sistema cancela las operaciones.	Satisfactorio

Tabla 5 Juego de datos del escenario Adicionar documento de distribución.

Id del escenario	Escenario	Código-descripción	Observaciones	Respuesta del sistema	Resultado de la prueba
4.1	Modificar documento de distribución.	V(Circuito 1)	V(completo)	Se modifica el documento y se muestra el mensaje de confirmación.	Satisfactorio
4.2	Modificar documento de distribución incorrectamente.	V(vacío)	V(completo)	Se muestra un mensaje de error indicando que hay campos requeridos vacíos.	Satisfactorio
4.3	Cancelar operación.	NA	NA	El sistema cancela las operaciones.	Satisfactorio

Tabla 6 Juego de datos del escenario Modificar documento de distribución.

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
5.1	Eliminar documento de distribución.	Se elimina el documento y se muestra el mensaje de confirmación.	Satisfactorio
5.2	Eliminar documento de distribución.	Se muestra un mensaje de error indicando que no hay seleccionado ningún documento de distribución.	Satisfactorio
3.3	Cancelar operación.	El sistema cancela las operaciones	Satisfactorio

Tabla 7 Juego de datos del escenario Eliminar documento de distribución.

Restantes escenarios ver anexos.

3.4 Conclusiones

Con la realización de este capítulo se especificó la propuesta del sistema donde se describe el proceso de implementación del módulo distribución y así facilitar la comprensión de la implementación realizada. Se comentan los diagramas de componentes, la matriz de integración interna. En el diagrama de componentes se determinaron las relaciones existentes entre componentes del sistema, la matriz de integración nos permite ver las funcionalidades que necesitamos de otros componentes. Se realiza la validación del producto a través de pruebas de aceptación que permitieron validar la solución propuesta, se realizaron descripciones del mismo teniendo en cuenta: tipo de prueba, objetivo de la prueba y alcance. Se describieron los valores utilizados y se evaluó la ejecución de la prueba así como los resultados obtenidos.

Conclusiones generales

Con la culminación de la presente investigación realizada, queda confirmada la necesidad de la implementación de un módulo de distribución de productos dentro del subsistema de logística que permita configurar los procesos de distribución de productos en una entidad. La implementación del módulo permitió el ahorro de una serie de proceso engorroso que consumían tiempo y esfuerzos y que daban un alto margen a realizar un mal trabajo, a demás de que se ayuda a resolver de forma eficiente uno de los problemas que más afecta a Cuba en la actualidad, con los temas relacionados con la gestión de las distribuciones de productos, así como la adaptación de la solución a las nuevas concepciones de informatización del país, de manera general se concluye con lo siguiente:

Con la elaboración de este trabajo se logró:

Realizar un estudio del estado del arte de sistemas similares al propuesto donde se llegó a la conclusión que sus características no eran las apropiadas y deseadas para aplicar a la solución descrita en este trabajo.

Dar cumplimiento de forma satisfactoria al objetivo general, implementándose las funcionalidades para el módulo distribución de productos.

La realización de un sistema potencialmente navegable, confiable y eficiente.

Utilizar las herramientas, lenguajes y proceso de desarrollo necesario para implementar la aplicación con la calidad requerida, haciendo uso de software libre.

Recomendaciones

Con el objetivo de mejorar la solución planteada se agregan las siguientes recomendaciones:

Se recomienda que este trabajo sea estudiado para que puedan agregársele nuevas funcionalidades que permitan una mayor flexibilidad del sistema de manera que pueda ser empleado no solamente en la industria farmacéutica y entidades presupuestadas de las FAR.

Revisar la implementación de esta aplicación para incorporar modificaciones que garanticen la mayor fiabilidad posible.

Seguir perfeccionando la aplicación de acuerdo con las nuevas necesidades que van surgiendo en el transcurso del tiempo.

Realizar la implementación de la distribución de productos con pedido.

Se recomienda realizarle al sistema pruebas unitarias.

Bibliografía

1. **Pressman, R. S.** 1998. Ingeniería de software. Un enfoque práctico. 1998.
2. **UCID.** Estándar para el diseño de interfaces v1.1. 2008.
3. **Leopoldo Magaña, Carlos.** Zend Framework, una introducción. [En línea] 2005-2008. [Citado el: 29 de marzo de 2010.] [http://www.carlosleopoldo.com/post/zend-framework-una-introduccion/..](http://www.carlosleopoldo.com/post/zend-framework-una-introduccion/)
4. **UCID.** Proceso de Desarrollo y Gestión v1.0.2010.
5. **Celis, Ismael.** ESTADOBETA desarrollo web con estándares. *Active Record*. [En línea] 2005. [Citado el: 6 de abril de 2010.] [http://www.estadobeta.com/2006/05/02/active-record/.](http://www.estadobeta.com/2006/05/02/active-record/)
6. **UCID.** *Ayuda al usuario del caso de estudio en el nuevo marco de trabajo.* 2008.
7. **Proenza Y.** *Revista Atix. Diseavanzado de aplicaciones web. ExtJS-Zend Framework-Doctrine.* 2009

Referencias Bibliográficas

- 1 El impacto de un ERP en la Empresa, InfoJobs [Online] 2008. [Citado: 21 de enero del 2010]
http://www.infojobs.net/noticias_frame.cfm?id=187450103
- 2 ERP Sistema de Gestión, Adpime.Es ERP [Online] 2007. [Citado: 21 de enero del 2010]
http://www.adpime.com/ERP/Es_ERP_intro.htm
- 3 ERP Software Benefits, ChozamSoftware [Online] 2008. [Citado: 21 de enero del 2010]
http://www.chozamsoftware.com/software/business/businessproc/erp/erp_software.html
- 4 ERP Solutions, Daugherty. Services [Online] 2007. [Citado: 21 de enero del 2010]
[www.daugherty.com/ services_erp_solutions.asp](http://www.daugherty.com/services_erp_solutions.asp)
- 5 Kumar, K., y Hillegersberg, J. v. (2006). Enterprise resource planning: Introduction. Communications of the ACM, 43(4), 22-26. [Online] 2005. [Citado: 21 de enero del 2010]
<http://0-delivery.acm.org.millennium.itesm.mx/10.1145/340000/332063/p22-kumar.pdf?key1=332063&key2=8889879211&coll=portal&dl=ACM&CFID=55975134&CFTOKEN=20158783>
- 6 Orton y Marlene (2004). Summit: ERP software. Summit, [Online] 2005. [Citado: 21 de enero del 2010]
http://www.findarticles.com/p/articles/mi_qa3993/is_200409/ai_n9449287
- 7 Wailgum, Thomas. 2008. ABC: An introduction to ERP. [Online] 2008. [Citado el: 21 de enero de 2010.]
[http://www.cio.com/research/erp/edit/erpbasics.html.](http://www.cio.com/research/erp/edit/erpbasics.html)
- 8 AxosVisual, ERP para Distribución [Online] 2007 [Citado el: 7 de abril de 2010.]
<http://www.axosvisual.com/empresa.php>

- 9 Cumulus, Software para Control de Almacenes, Pedidos y Distribución.. [Documento] 2005. [Citado el: 8 de abril de 2010.] <http://www.supplychain-software.com/index.html>
- 10 JR Software, JR Software Distribuidores. [Online] 2006. [Citado el: 8 de abril de 2010.] <http://www.softwarejr.com.ar/software-distribuidora-alimentos.htm>
- 11 ecoSoftCS.net. ecosoftconsulting. [Online] 2007. [Citado: 8 de abril de 2010.] <http://www.ecosoftconsulting.net/programasparapymes.aspx>
- 12 vance Soluciones Informáticas. SGTaller. [Online] 2009. [Citado: 9 de febrero del 2010.] <http://www.sgtaller.com>.
- 13 Assets S.A. Sistema de Gestión Integral. [Online] 2007. [Citado:9 de febrero del 2010, 2008.] <http://assets.co.cu/taller.asp>.
- 14 CUJAE. Macwin. Instituto Superior Politécnico José Antonio Echeverría. [Online] 2006. [Citado: 9 de febrero del 2010.] <http://www.cujae.edu.cu/centros/ceim/macwin.htm>
- 15 freedownloadmanager.org. 2008. Visual paradigm. [Online] 2008. [Citado el: 11 de febrero de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_147_20_p](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_147_20_p)
- 16 José Carlos y González Brito, Henry Raúl. 2009. Documento Visión. Proyecto ERP-Cuba. [Documento] La Habana: Universidad de las Ciencias Informáticas. [Citado el: 11 de febrero de 2010].
- 17 Netpecos.org. PostgreSql. [Online] 2008. [Citado el: 11 de febrero de 2009.] http://www.netpecos.org/docs/mysql_postgres/x15.html.
- 18 Sqlmanager.net. Sqlmanager para PostgreSQL. [Online] 2007. [Citado el: 11 de febrero de 2010.] <http://sqlmanager.net/products/postgresql/manager>.

- 19 Zend.com. Zend Studio for Eclipse. [Online] 2007. [Citado el: 11 de febrero de 2010.]
<http://www.zend.com/products/studio/>.
- 20 Cova, Rubén Darío. Html. [Online] 2007. [Citado el: 11 de febrero de 2010.]
<http://www.geocities.com/covag/defhtml.html>.
- 21 Maestrosdelweb.com. Que es javascript. [Online] 2007. [Citado el: 16 de febrero de 2010.]
<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
- 22 Webexperto.com. AJAX. [Online] 2006. [Citado el: 16 de febrero de 2010.]
<http://www.webexperto.com/articulos/articulo.php?cod=223>.
- 23 Json.org. Sitio oficial de JSON. [Online] 2007. [Citado el: 16 de febrero de 2010.]
<http://www.json.org/json-es.html>.
- 24 Desarrolloweb.com. CSS. [Online] 2008. [Citado el: 16 de febrero de 2010.]
<http://www.desarrolloweb.com/articulos/26.php>.
- 25 Desarrolloweb.com. Qué es PHP. [Online] 2009. [Citado el: 16 de febrero de 2010.]
<http://www.desarrolloweb.com/articulos/392.php>.
- 26 Desarrolloweb.com. Arquitectura Cliente Servidor. [Online] 2008. [Citado el: 16 de febrero de 2010.]
<http://www.desarrolloweb.com/articulos/26.php>.
- 27 Ciberaula.com. Introducción a Apache. [Online] 2008. [Citado el: 16 de febrero de 2010.]
http://linux.ciberaula.com/articulo/linux_apache_intro/.
- 28 Echarte, Patxi. Frameworks de Zend para el desarrollo de aplicaciones PHP. [Online] 2007. [Citado el: 16 de febrero de 2010.]
<http://www.eslomas.com/index.php/archives/2007/07/03/framework-de-zend-para-el-desarrollo-de-aplicaciones-php/>.

- 29 Doctrine-Project.org. [Online] 2008. [Citado el: 16 de febrero de 2010.]
<http://www.doctrine-project.org>.
- 30 Wordpress.com. Librería ExtJs. [Online] 2007. [Citado el: 16 del febrero de 2010.]
<http://vargasti.wordpress.com/2007/08/06/libreria-extjs/>.
- 31 Firefox, Mozilla. [Online] 2009. [Citado el: 16 de febrero de 2010.]
<http://www.getfirefox.es/firefox-features>.
- 32 Osmosislatina.com. Subversion. [Online] 2009. [Citado el: 16 del febrero de 2010.]
<http://www.osmosislatina.com/subversion/basico.htm>.
- 33 Centro de Soluciones de Gestión. 2008. Modelo de desarrollo orientado a componentes. [Documento]
La Habana: Universidad de las Ciencias Informáticas, 2008.
- 34 Cyta. Herramientas Case. [Online] 2008. [Citado el: 12 de abril de 2010.]
<http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>
- 35 Desarrolloweb.com. IDE desarrollo. [Online] 2008. [Citado el: 12 de abril de 2010.]
<http://www.desarrolloweb.com/articulos/26.php>.

Anexos

Glosario de Términos

[C]

CedruX: Vocablo formado por la unión de las palabras “Cedro” (fortaleza, resistencia) y “Linux” (tecnología libre).

CSS: (Cascading Style Sheets) Hojas de Estilo en Cascada.

CVS: (Concurrent Versions System) Aplicación informática que implementa un sistema de control de versiones.

Componente: Un componente es una parte no trivial, casi independiente, y reemplazable de un subsistema que llena claramente una funcionalidad dentro de un contexto en una arquitectura bien definida. Un componente se conforma y provee la realización física por medio de un conjunto de interfaces.

[D]

DOM: (Document Object Model) Modelo en Objetos para la representación de Documentos.

[F]

Framework: Denota la infraestructura sobre la cual se reúnen un conjunto de lenguajes, herramientas y servicios que simplifican el desarrollo de aplicaciones en entorno de ejecución distribuido.

[H]

HTML: (Hyper Text Markup Language) Lenguaje de Marcas de Hipertexto.

Herencia: Las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que re-implementar su comportamiento. Esto suele hacerse habitualmente agrupando los objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple; esta característica no está soportada por algunos lenguajes (como Java).

[I]

IE: Internet Explorer.

[J]

JSON: (JavaScript Object Notation) Notación de Objetos de JavaScript.

[L]

LDAP: (Lightweight Directory Access Protocol) Protocolo Ligero de Acceso a Directorios.

Logs: Registro de errores.

[O]

ODBC: (Open Database Connectivity) Estándar de acceso a Bases de datos.

[P]

PDO: (PHP Data Objects) Capa de abstracción de acceso a datos para PHP.

PAM: (Pluggable Authentication Modules) mecanismo que proporciona una interfaz entre las aplicaciones de usuario y diferentes métodos de autenticación.

Proceso: Conjunto de actividades que guían los esfuerzos de las personas implicadas para el cumplimiento de un objetivo.

[S]

SSL: (Secure Sockets Layer) Protocolo de Capa de Conexión Segura.

Software: Es un programa o aplicación de que permite a los usuarios el control o realización de varias tareas y que hacen que el trabajo sea más cómodo, rápido y eficiente.

Subsistema: Son las partes que forman un sistema. Cada sistema está compuesto de subsistemas, los cuales a su vez son parte de otros subsistemas; cada subsistema es delineado por sus límites.

Servidor: En informática, un servidor es una computadora que, formando parte de una red, provee servicios a otras computadoras denominadas clientes.

[T]

TCP/IP: (Transmission Control Protocol/Internet Protocol) Protocolo de control de transmisión/Protocolo de Internet.

[U]

UML: (Unified Modeling Language) Lenguaje Unificado de Modelado.

[X]

XMLHttpRequest: Interfaz empleada para realizar peticiones HTTP y HTTPS a servidores WEB.

XML: (Extensible Markup Language) Lenguaje de Etiquetado Extensible.

[W]

Web: En informática, la World Wide Web, cuya traducción podría ser *Red Global Mundial* o "Red de Amplitud Mundial", es un sistema de documentos de hipertexto enlazados y accesibles a través de Internet.