Universidad Nacional Autónoma de Nicaragua.

Unan-León

Facultad de Ciencias Puras

Ingeniería en Sistemas de Información

Tema: Actualización del Sistema de Control de Entrada-Salida de la Bodega Central de la Unan-León, utilizando Software Libre.

Elaborado por:

- Br. Luís Manuel Gámez Vallejos.
- Bra. Claudia Neira Moncada.
- Br. Jonathan Zapata Granera.

Tutor:

Msc**.** Francisco Danilo Padilla C.

León, noviembre del 2007**.**

INDICE **PÁGINA**

AGRADECIMIENTO

En esta ocasión quiero expresar mi gratitud y agradecerles de corazón a todas aquellas personas que de una u otra manera me brindaron su ayuda y apoyo durante todo este tiempo.

Con mención especial a:

Dios, el dador de la vida. Por estar ahí en esos momentos cuando solo me sentía y brindarme la sabiduría necesaria para concluir mis estudios.

Mis padres, Concepción Vallejos y Rubén Gámez. Mi mayor agradecimiento para ellos por su incondicional apoyo en todo momento, por sus consejos y educación. Que de mí sin ellos.

Msc. Danilo Padilla, por su ayuda brindada para finalizar este trabajo y cumplir con la meta propuesta.

GRACIAS, muchas gracias.

Luís Manuel Gámez Vallejos

DEDICATORIA

Dedico este trabajo a:

Mis Padres, Rubén Gámez y **Concepción Vallejos** por el apoyo incondicional que me han brindado durante todo este tiempo y por ser mi mayor apoyo e inspiración para salir adelante.

Luís Manuel Gámez Vallejos

AGRADECIMIENTO

En esta ocasión quiero expresar mi gratitud y agradecerles de corazón a todas aquellas personas que de una u otra manera me brindaron su ayuda y apoyo durante todo este tiempo.

Con mención especial a:

Dios, el dador de la vida. Por estar ahí en esos momentos cuando solo me sentía y brindarme la sabiduría necesaria para concluir mis estudios.

Mis padres, Rubel Neira y Angela Moncada . Mi mayor agradecimiento para ellos por su incondicional apoyo en todo momento, por sus consejos y educación.

Msc. Danilo Padilla, por su ayuda brindada para finalizar este trabajo y cumplir con la meta propuesta.

Muchas gracias que dios les bendiga

Llaudia Yveth Neira Moncada

DEDICATORIA

Este trabajo se lo dedico primeramente a Dios, ya que sin el nada podemos hacer.

Mis Padres, Rubel Neira y Angela Moncada por que ellos siempre están aquí en las buenas y en las malas; me educan me aconsejan, me imparten valores para conducirme correctamente y me ofrecen el sabio consejo en el momento oportuno.

Claudia Yvet Neira Moncada

AGRADECIMIENTO

En esta ocasión quiero expresar mi gratitud a todas aquellas personas que de una u otra manera me brindaron su ayuda y apoyo en todo este tiempo.

Con mención especial A:

Dios, el dador de la vida. Por estar ahí en esos momentos cuando solo me sentía y brindarme la sabiduría necesaria para concluir mis estudios.

Mis padres: Guadalupe Granera Rojas y Ervin Zapata López. Mi mayor agradecimiento para ellos por su incondicional apoyo en todo momento, por sus consejos y educación. Que de mí sin ellos.

Msc. Danilo Padilla, por su ayuda brindada para finalizar este trabajo y cumplir con la meta propuesta.

GRACIAS, muchas gracias.

Jonathan Zapata Granera

DEDICATORIA

Dedico este trabajo a:

Quien en vida fuera, **Génesis Ivonne Zapata Granera**, mi hermanita; por ser mi fuente de inspiración para continuar en este mundo."**Vives en mí**".

Jonathan Zapata Granera

I. INTRODUCCIÓN

En la Unan-León existe una oficina que se encarga del control de todo lo que entra y sale de la bodega, esta oficina como tal debe asegurar una excelente administración de los registros e informes de todos los productos que esta adquiere de sus proveedores, de los que entrega a las facultades o unidades administrativas y clientes además de los que esta posee en su inventario.

Dado el avance tecnológico que está viviendo nuestro país es de suma importancia la elaboración de un sistema que permita el avance de la universidad hacia un control automatizado de todas sus dependencias, además les facilite al personal la tarea de realizar los registros de control de solicitud y entrega de productos que esta maneja, razón por la cual nos hemos propuesto llevar a cabo la realización de este trabajo que será de mucha trascendencia para la división administrativa de nuestra universidad.

Este sistema se encarga de llevar el control de los tipos de usuarios especificados en él, al igual que maneja las entradas que se registran en la bodega, como también las salidas actualizando las existencias, permitiendo realizar búsquedas de productos por nombre y código también generar reportes a partir de una fecha determinada siempre y cuando sea el usuario correcto para que dicho sistema lleve un control y funcione de forma más rápida, eficiente y cómoda para el personal, resolviendo así la problemática que presentan estas unidades administrativas de la Unan-León.

Este sistema se ha desarrollado con lenguajes de programación en red, que permite a los usuarios hacer solicitudes de materiales y consultas de todo lo retirado de bodega desde cualquier computadora de la red.

II. ANTECEDENTES

Gracias a la mala situación que esta atravesando nuestro país en el sentido tecnológico y económico muchas empresas deben llevar sus controles laborales, empresariales e institucionales de la manera mas antigua y burda, de forma manual por lo que es urgente la creación de un sistema informático de control automatizado que nos ayude en la administración y nos garantice una información ágil, útil y segura con resultados más rápidos y veraces para el cumplimiento de nuestros objetivos y propósito establecidos.

En el año 1993 el profesor Msc. Francisco Danilo Padilla C. desarrolló un sistema de automatización de control de los procesos de Entrada-Salida de la Bodega Central de la Unan-León el cual se logró implementar.

Luego se propuso llevar a cabo la actualización de este sistema en el año 2003 utilizando Visual Fox Pro V.6 Fox base, este sistema es el que actualmente se esta utilizando en la bodega del cual se han obtenido buenos resultados.

III. JUSTIFICACIÓN

El desarrollo tecnológico hace que las pequeñas empresas e instituciones cambien, obteniendo equipos y software mas actualizado que se acomodan con los usados actualmente en el mercado por la competencia.

Es por ello la necesidad de actualizar el sistema de Control de Entrada-Salida de la Bodega Central de la Unan-León evitando que se sigan utilizando programas que pueden ser actualizados para obtener un mejor rendimiento de ellos.

Sin embargo hoy en día los objetivos han sufrido una serie de cambios y modificaciones que demandan nuevas actualizaciones al sistema utilizado hasta hoy como monousuario para que pueda ser usado y consultado como multiusuario en las diferentes facultades o dependencias.

Todo esto se hará para que el nuevo sistema cumpla con todos los requisitos y demandas propuestas por el usuario.

IV. OBJETIVOS

General:

Elaborar un sistema que lleve el control de las entradas y salidas de una bodega haciendo uso de Software libre y del lenguaje PHP.

Específicos:

1. Diseñar una interfaz que sea de utilidad para el usuario y de fácil manejo.

2. Instalar el servidor Web DEBIAN de manera manual sin uso de asistente.

3. Comprobar que el sistema funcione correctamente.

4. Conocer y manejar correctamente el software y el lenguaje empleado (PHP).

V. MARCO TEÓRICO

Sistema:

Un **sistema** es un conjunto de unidades en interrelación, es una totalidad organizada, hecha de elementos solidarios que no pueden ser definidos más que los unos con relación a los otros en función de su lugar es una colección organizada de hombres, máquinas y métodos necesaria para cumplir un objetivo específico.

Tipos de sistema:

En cuanto a su constitución, pueden ser físicos o abstractos:

- *Sistemas físicos o concretos*: compuestos por equipos, maquinaria, objetos y cosas reales. El hardware.
- *Sistemas abstractos*: compuestos por conceptos, planes, hipótesis e ideas. Muchas veces solo existen en el pensamiento de las personas. Es el software.

En cuanto a su naturaleza, pueden ser cerrados o abiertos:

- *Sistemas cerrados*: Se da el nombre de sistema cerrado a aquellos sistemas cuyo comportamiento es determinístico y programado y que opera con muy pequeño intercambio de energía y materia con el ambiente.
- *Sistemas abiertos*: presentan intercambio con el ambiente, a través de entradas y salidas. Intercambian energía y materia con el ambiente. Son adaptativos para sobrevivir.

Aspectos fundamentales de un Sistema:

- La existencia de elementos diversos e interconectados.
- El carácter de unidad global del conjunto.
- La existencia de objetivos asociados al mismo.

- La integración del conjunto en un entorno.

Inventario

Un inventario representa la existencia de bienes tangibles muebles e inmuebles que tiene la empresa para comerciar con ellos, comprándolos y vendiéndolos en el curso ordinario del negocio tal cual o procesándolos primero antes de venderlos, en un período económico determinado comprenden, además repuestos y accesorios para ser consumidos en la producción de bienes fabricados para la venta o en la prestación de servicios.

Inventario de Mercancías

Lo constituyen todos aquellos bienes que le pertenecen a la empresa bien sea comercial o mercantil, los cuales los compran para luego venderlos sin ser modificados. En esta Cuenta se mostrarán todas las mercancías disponibles para la Venta. Las que tengan otras características y estén sujetas a condiciones particulares se deben mostrar en cuentas separadas, tales como las mercancías en camino (las que han sido compradas y no recibidas aún), las mercancías dadas en consignación o las mercancías ignoradas (aquellas que son propiedad de la empresa pero que han sido dadas a terceros en garantía de valor que ya ha sido recibido en efectivo u otros bienes).

Sistema de Inventario

En el sistema de Inventario el negocio mantiene un registro continuo para cada artículo del inventario. Los registros muestran por lo tanto el inventario disponible todo el tiempo. Los registros son útiles para preparar los estados financieros mensuales, trimestral o provisionalmente. EL negocio puede determinar el costo del inventario final y el costo de las mercancías vendidas directamente de las cuentas sin tener que contabilizar el inventario. El sistema ofrece un alto grado de control, porque los registros de inventario están siempre actualizados.

Software

Se denomina **software** a todos los componentes intangibles de un ordenador o computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica; es el conjunto de instrucciones que permite al hardware de la computadora desempeñar trabajo útil.

Tipos de software

Se puede distinguir al software de la siguiente forma:

- **Software de sistema**: la parte que permite funcionar al hardware. Su objetivo es aislar tanto como sea posible al programador de aplicaciones de los detalles del computador particular que se use, especialmente de las características físicas de la memoria, dispositivos de comunicaciones, impresoras, pantallas, teclados, etcétera. Incluye entre otros:

  - ✓ Sistemas operativos
  - ✓ Controladores de dispositivo
  - ✓ Herramientas de diagnóstico
  - ✓ Servidores
  - ✓ Sistemas de ventanas
  - ✓ Utilidades.

- **Software de programación**: proporciona herramientas que ayudan al programador a escribir programas informáticos y a usar diferentes lenguajes de programación de forma práctica. Incluye entre otros:

  - ✓ Editores de texto
  - ✓ Compiladores
  - ✓ Intérpretes
  - ✓ Enlazadores

- ✓ Depuradores
- ✓ Los entornos integrados de desarrollo (IDE) agrupan estas herramientas de forma que el programador no necesite introducir múltiples comandos para compilar, interpretar, depurar, etcétera, gracias a que habitualmente cuentan con una interfaz gráfica de usuario (GUI) avanzada.

- **Software de aplicación**: permite a los usuarios llevar a cabo una o varias tareas más específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios. Incluye entre otros:

  - ✓ Aplicaciones de automatización industrial.
  - ✓ Aplicaciones ofimáticas.
  - ✓ Software educativo.
  - ✓ Software médico.
  - ✓ Bases de datos.
  - ✓ Videojuegos.

Formas de un software

El software adopta varias formas en distintos momentos de su ciclo de vida:

- **Código fuente**: escrito por programadores. Contiene el conjunto de instrucciones destinadas a la computadora.

- **Código objeto**: resultado del uso de un compilador sobre el código fuente. Consiste en una traducción de éste último. El código objeto no es directamente inteligible por el ser humano, pero tampoco es directamente entendible por la computadora. Se trata de una representación intermedia del código fuente.

- **Código ejecutable**: resultado de enlazar uno o varios fragmentos de código objeto. Constituye un archivo binario con un formato tal que el sistema operativo es capaz de cargarlo en la memoria de un ordenador, y proceder a su ejecución. El código ejecutable es directamente inteligible por la computadora.

Creación de software

El proceso de creación de software es materia de la ingeniería del software. Es un proceso complejo que involucra diversas tareas de gestión y desarrollo. Como resumen de las etapas para la creación de un software, se pueden mencionar:

- ✓ Análisis.
- ✓ Desarrollo.
- ✓ Construcción.
- ✓ Pruebas (unitarias e integradas).
- ✓ Paso a Producción

Ingeniería de software

Es la rama de la ingeniería que crea y mantiene las aplicaciones de software aplicando tecnologías y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos; tiene que ver con el costo y la confiabilidad.

Orígenes de Ingeniería de software

La Ingeniería del Software, término utilizado por primera vez por Fritz Bauer en la primera conferencia sobre desarrollo de software patrocinada por el Comité de Ciencia de la OTAN celebrada en Garmisch, Alemania, en octubre de 1968, puede definirse según Alan Davis como "*la aplicación inteligente de principios probados, técnicas, lenguajes y herramientas para la creación y mantenimiento, dentro de un coste razonable, de software que satisfaga las necesidades de los usuarios*"...

Significados de Ingeniería del Software

El término *ingeniería de software* se usa con una variedad de significados diferentes: Como el término usual contemporáneo de un amplio rango de actividades que se solía llamar programación y análisis de sistemas;

Como un término amplio de todos los aspectos de la *práctica* de la programación de computadoras, en oposición a la *teoría*, que es llamada ciencia computacional o computación;

Pasos del proceso de Ingeniería del Software

La ingeniería de software requiere llevar a cabo muchas tareas, sobre todo las siguientes:

- **Análisis de requisitos:** Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios.
- **Especificación:** Es la tarea de describir detalladamente el software a ser escrito, en una forma matemáticamente rigurosa. En la realidad, la mayoría de las buenas especificaciones han sido escritas para entender y afinar aplicaciones que ya estaban desarrolladas. Las especificaciones son más importantes para las interfaces externas, que deben permanecer estables.
- **Diseño y arquitectura:** Se refiere a determinar como funcionará de forma general sin entrar en detalles. Consiste en incorporar consideraciones de la implementación tecnológica, como el hardware, la red, etc.
- **Programación:** Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no es necesariamente la porción más larga.
- **Prueba:** Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral.
- **Documentación:** Realización del manual de usuario, y posiblemente un manual técnico con el propósito de mantenimiento futuro y ampliaciones al sistema.

- **Mantenimiento:** Mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos.

Software libre

En inglés ***free software*** es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente, suele estar disponible gratuitamente, pero no hay que asociar software libre a software gratuito, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente.

No debe confundirse "software libre" con software de dominio público. Éste último es aquél por el que no es necesario solicitar ninguna licencia y cuyos derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquél cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es dominio público. En resumen, el software de dominio público es la pura definición de la libertad de usufructo de una propiedad intelectual que tiene la humanidad porque así lo ha decidido su autor o la ley tras un plazo contado desde la muerte de éste, habitualmente 70 años.

Historial

Entre los años 60 y 70 del Siglo XX, el software no era considerado un producto sino un añadido que los vendedores de los grandes computadores de la época (los *mainframes*) aportaban a sus clientes para que éstos pudieran usarlos. En dicha cultura, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros. Este comportamiento era particularmente habitual en algunos de los mayores grupos de usuarios de la época, como DECUS (grupo de usuarios de computadoras DEC). A finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de licencia.

Con este antecedente, en 1984 Richard Stallman comenzó a trabajar en el proyecto GNU, y un año más tarde fundó la Free Software Foundation (FSF). Stallman introdujo una definición para *free software* y el concepto de "*copyleft*", el cual desarrolló para dar a los usuarios libertad y para restringir las posibilidades de apropiación del software.

El termino free, traducido al castellano, significa tanto libre como gratis, por eso muchas veces suelen confundirse el freeware con el software libre aunque entre ambos existen notables diferencias.

Hardware

En la Informática se denomina hardware o soporte físico al conjunto de elementos materiales que componen un ordenador. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos de la computadora como: unidades de disco, monitor, teclado, ratón (Mouse), impresora, placas, chips y demás periféricos.

Tipos de hardware

Se clasifica generalmente en básico y complementario, entendiendo por básico todo aquel dispositivo necesario para iniciar el funcionamiento de la computadora, y el complementario como su nombre lo dice sirve para realizar funciones específicas o más allá de las básicas.

Funciones del hardware

Debe tener la capacidad de interpretar y ejecutar comandos programados para operaciones de entrada, salida, cálculo y lógica.

Las computadoras:

- Reciben entradas. La entrada son los datos que se capturan en un sistema de computación para su procesamiento.

- Producen salidas. La salida es la presentación de los resultados del procesamiento.
- Procesan información : Unidad central de procesamiento : Es la computadora real, la "inteligencia" de un sistema de computación donde
- Almacenan información: Memoria y dispositivos de almacenamiento.

Interfaz

En sentido amplio, puede definirse interfaz como el conjunto de comandos y métodos que permiten la intercomunicación del programa con cualquier otro programa o elemento interno o externo. De hecho, los periféricos son controlados por interfaces. Por extensión, se denomina interfaz a cualquier medio que permita la interconexión de dos procesos diferenciados con un único propósito común.

Interfaz de usuario:

Es la parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario. también se habla de interfaz gráfica de usuario, que es un método para facilitar la interacción del usuario con el ordenador o la computadora a través de la utilización de un conjunto de imágenes y objetos pictóricos (iconos, ventanas..) además de texto.

Características de Interfaz

Al implementar el diseño de una interfaz de usuario, se deben tener en cuenta -entre varios otros- los siguientes puntos:

- ✓ Tipo y características del usuario para quien se diseña.
- ✓ Necesidades y objetivos de la/s tarea/s.
- ✓ Requerimientos del sistema para cumplir la tarea.
- ✓ Métodos de ergonomía e ingeniería de usabilidad (facilidad de aprendizaje y de uso).

Base da datos

Una **base** o **banco de datos** es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática.

Estructura Cliente-Servidor.

En las redes basadas en estructuras cliente-servidor, los servidores ponen a disposición de sus clientes recursos, servicios y aplicaciones.

Dependiendo de que recursos ofrezca el servidor y cuales se mantienen en los clientes se pueden hacer distinciones entre distintas estructuras cliente-servidor.

En estas estructuras se diferencia:

- Donde se encuentran los datos.
- Donde se encuentran los programas de aplicación.
- Donde se presentan los datos.

La World Wide Web (w.w.w):

La **Web** o WWW, es un sistema de navegador para extraer elementos de información llamados "documentos" o "páginas Web". Puede referirse a "una Web" como una página, sitio o conjunto de sitios que proveen información por los medios descritos, o a "la Web", que es la enorme e interconectada red disponible prácticamente en todos los

sitios de Internet. Ésta es parte de Internet, siendo la World Wide Web uno de los muchos servicios ofertados en la red Internet.

Paginas dinámicas:

Las páginas dinámicas son páginas HTML generadas a partir de lenguajes de programación (scripts) que son ejecutados en el propio servidor web. A diferencia de otros scripts, como el JavaScript, que se ejecutan en el propio navegador del usuario, los 'Server Side' scripts generan un código HTML desde el propio servidor web.

Este código HTML puede ser modificado (por ejemplo) en función de una petición realizada por el usuario en una Base de Datos. Dependiendo de los resultados de la consulta en la Base de Datos, se generará un código HTML u otro, mostrando diferentes contenidos.

PHP

PHP (Hypertext Preprocessor) es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor; es un lenguaje script (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de Open Source (código abierto), es el más popular y extendido en la web.

PHP es capaz de realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas escritos en un lenguaje distinto al HTML. Esto se debe a que PHP ofrece un extenso conjunto de funciones para la explotación de bases de datos sin complicaciones. Es por esto, que levanta un mayor interés con respecto a los lenguajes pensados para los CGI.

Ejemplo:

```
<html>
   <head>
      <title>EjemploPHP</title>
   </head>
   <body>
      <?phpecho "Hola, este es un ejemplo con PHP!";?>
   </body>
</html>
```

Podemos ver que no es lo mismo que un script CGI escrito en otro lenguaje de programación como Perl o C – En vez de escribir un programa con muchos comandos para crear una salida en HTML, escribimos el código HTML concierto código PHP embebido (introducido) en el mismo, que producirá cierta salida (en nuestro ejemplo, producir un texto). El código PHP se incluye entre etiquetas especiales de comienzo y final que nos permitirán entrar y salir del modo PHP.

Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si tuviésemos un script similar al de nuestro ejemplo en nuestro servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los ficheros HTML con PHP.

Algunas características de PHP

Al nivel más básico, PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies. Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir

un interfaz vía Web para una base de datos es una tarea simple con PHP. Las siguientes bases de datos están soportadas actualmente:

Adabas D	Ingres	Oracle(OCI7andOCI8)
dBase	Internase	PostgreSQL
Empress	FrontBase	Solid
FilePro	mSQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unixdbm

PHP también soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados. También se pueden abrir sockets de red directos (rawsockets) e interactuar con otros protocolos.

Corta historia de PHP

PHP fue concebido en otoño de 1994 por Rasmus Lerdorf. Las primeras versiones no distribuidas al público fueron usadas en sus páginas web para mantener un control sobre quien consultaba su currículum. La primera versión disponible para el público a principios de 1995 fue conocida como "Herramientas para paginas web personales. Consistían en un analizador sintáctico muy simple que solo entendía unas cuantas macros y una serie de utilidades comunes en las páginas web de entonces, un libro de visitas, un contador y otras pequeñas cosas. El analizador sintáctico fue reescrito a mediados de 1995 y fue nombrado PHP/FI versión 2.

FI viene de otro programa que Rasmus había escrito y que procesaba los datos de formularios. Así que combinó las "Herramientas 45 para páginas web personales", el "intérprete de formularios", añadió soporte para mySQL y PHP/FI vio la luz. PHP/FI creció a gran velocidad y la gente empezó a contribuir en el código.

A mediados de 1997 el desarrollo del proyecto sufrió un profundo cambio, dejó de ser un proyecto personal de Rasmus, al cual habían ayudado un grupo de usuarios y se convirtió en un proyecto de grupo mucho más organizado. El analizador sintáctico se reescribió desde el principio por ZeevSuraski y Andi Gutmans y este nuevo analizador estableció las bases para PHP versión 3. Gran cantidad de código de PHP/FI fue portado a PHP3 y otra gran cantidad fue escrita completamente de nueva.

A finales de 1999, tanto PHP/FI como PHP3 se distribuyen en un gran número de productos comerciales tales como el servidor web "C2's StrongHold" y Redhat Linux. Una estimación conservativa basada en estadísticas de NetCraft, es que más de 1.000.000 de servidores alrededor del mundo usan PHP. Para hacernos una idea, este número es mayor que el número de servidores que utilizan el "Netscape's Enterprise server" en Internet.

A la vez que todo esto está pasando, el trabajo de desarrollo de la próxima generación de PHP está en marcha. Esta versión utiliza el potente motor de scripts Zend para proporcionar altas prestaciones, así como soporta otros servidores web, además de apache, que corren PHP como módulo nativo.

Servidor Apache

La historia de Apache se remonta a febrero de 1995, donde empieza el proyecto del grupo Apache, el cual esta basado en el servidor Apache httpd de la aplicación original de NCSA. El desarrollo de esta aplicación original se estancó por algún tiempo tras la marcha de Rob McCool por lo que varios Web master siguieron creando sus parches para sus servidores Web hasta que se contactaron vía e-mail para seguir en conjunto el mantenimiento del servidor Web, fue ahí cuando formaron el grupo Apache.

Fueron Brian Behlendorf y Cliff Skolnick quienes a través de una lista de correo coordinaron el trabajo y lograron establecer un espacio compartido de libre acceso para los desarrolladores.

Apache surgió a partir del servidor de HTTP más famoso y difundido en su época: NCSA. Desde entonces se convirtió en un poderoso rival de todos los servidores Unix utilizados hasta la fecha por su eficiencia, funcionalidad y rapidez. Es por ello que se conoce como el rey de los servidores Web. Se desarrolla de forma estable y segura gracias a la cooperación y los esfuerzos de un grupo de personas conocidas como grupo Apache (Apache Group), los cuales se comunican a través de Internet y del Web. Juntos se dedican a perfeccionar el servidor y su documentación regidos por la ASF (Apache Software Foundation).

Apache es el servidor Web líder en el mercado. Su costo es gratuito, gran fiabilidad y extensibilidad le convierten en una herramienta potente y muy configurable.

¿Qué es el servidor Apache?

El servidor Apache (http://www.apache.org) es el servicio que se encarga de resolver las peticiones de páginas de Internet de los clientes utilizando el protocolo de Internet http. En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios.

El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor Web, este término podría referirse a la máquina que almacena y maneja los sitios

Web, y en este sentido es utilizada por las compañías que ofrecen hospedaje. Alternativamente, el servidor Web podría referirse al software, como el servidor de HTTP de Apache, que funciona en la máquina y maneja la entrega de los componentes de las páginas Web como respuesta a peticiones de los navegadores de los clientes.

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Como su nombre implica, un servidor sirve información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor. En la WWW, un servidor Web es un ordenador que usa el protocolo http para enviar páginas Web al ordenador de un usuario, cuando el usuario las solicita.

Los servidores Web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar el Internet. Algunos servidores manejan solamente correo o solamente archivos, mientras que otros hacen más de un trabajo, ya que un mismo ordenador puede tener diferentes programas de servidor funcionando al mismo tiempo. Los servidores se conectan a la red mediante una interfaz.

Características del servidor Apache

Dentro de las características de Apache tenemos las siguientes.

1. Es un servidor Web potente, flexible y ajustado al HTTP/1.1
2. Es altamente configurable y extensible.
3. Puede ser ajustado a través de la definición de módulos empleando su propio API (Aplication Programming Interface).

4. Provee todo su código fuente de forma libre y se distribuye bajo una licencia no restrictiva.
5. Se ejecuta en diversas plataformas operativas tales como: Windows 9x/NT, Macintosh, Novell NetWare, OS/2, Linux y la mayoría de los Unix existentes: IRIX, Solaris, HPUX, SCO, FreeBSD, NetBSD, AIX, Digital Unix, etc.
6. Se desarrolla de forma acelerada estimulando la retroalimentación desde sus usuarios a través de nuevas ideas, reportes de errores y parches.
7. Apache significa ``A PAtCHy sErver'', o sea se basa en un código y un conjunto de ficheros ``parches''. Otros desarrolladores relacionan su nombre con el de las tribus nativas americanas de Apaches.
8. Bases de datos DBM para autenticación.
9. Permiten establecer fácilmente la protección de documentos a través de passwords para una gran cantidad de usuarios sin dañar el funcionamiento del servidor.
10. Respuestas adaptables a los errores o problemas.
11. Se pueden definir ficheros o scripts de tipo CGI11.3 que respondan ante la ocurrencia de errores internos o en las solicitudes realizadas.
12. Directiva para definir múltiples índices.
13. Se utiliza cuando se solicitan directorios por parte de los clientes a partir de lo cual se puede buscar en estos y devolver un documento índice cuyo nombre puede ser por ejemplo: index.HTML, index.cgi o default.HTML.
14. Ilimitadas y flexibles posibilidades de redireccionamiento y definición de alias para los URLs.
15. Apache no tiene un límite establecido para definir alias y redireccionamientos que pueden ser declarados en sus ficheros de configuración.
16. Negociación del contenido de las respuestas.

17. Apache es capaz de ofrecer la mejor representación de la información accedida de acuerdo con las capacidades del cliente solicitante.
18. Soporte de hosts virtuales.
19. Es la habilidad del servidor de distinguir entre los pedidos hechos a diferentes direcciones IP o nombres de dominio definidos en la misma máquina.
20. Configuración flexible de las trazas generadas.
21. Es posible adaptar el formato de las trazas obtenidas así como redireccionarlas a través de tuberías (Unix) en aras de filtrarlas. De esta forma se puede lograr por ejemplo dividir dinámicamente las trazas de los hosts virtuales en distintos ficheros.

Ventajas del uso del Servidor Apache

Algunas de las ventajas son las siguientes:

1. Se pueden extender las características de Apache hasta donde nuestra imaginación y conocimientos lleguen.
2. El Apache soporta Dinamic Shared Object (DSO). Gracias a ello se pueden construir módulos que le den nuevas funcionalidades que son cargadas en tiempos de ejecución.
3. Apache puede facilitar información en varios formatos para que un determinado cliente pueda interpretarla.
4. Apache permite la creación de sitios Web dinámicos mediante:
   - El uso de CGI's.
   - El uso de Server Side Includes (SSI).
   - El uso de lenguajes de Scripting como PHP, JavaScript, Python.
   - El uso de Java y páginas jsp.

Javascript

JavaScript es una de las múltiples aplicaciones que han surgido para extender las capacidades del Lenguaje HTML. JavaScript es un lenguaje script orientado a documento. No sirve para crear programas, sino para agregarles características a nuestra Web.

Las normas para poder escribir cualquier código de JavaScript se basan en algunos puntos básicos y que debemos cumplir siempre. Estas normas son las siguientes:

1. Todo el código (sentencias) esta dentro de funciones.
2. Las funciones se desarrollan entre las etiquetas <script> y </script>.
3. Las etiquetas “<script>” deben colocarse entre las etiquetas <head> y </head>.
4. Las etiquetas “<title>” no pueden estar colocadas entre las de “<script>”.
5. La llamada a la función se hace a través de un evento de un elemento del documento.

Las funciones son un conjunto de sentencias (bloque de código) que especifica al programa las operaciones a realizar. Son útiles para evitar la repetición de líneas y modular el código. Para trabajar con ellas hay que desarrollarlas y llamarlas cuando lo necesitemos.

En el primero de los casos la llamada se realiza desde un elemento del documento. En el segundo caso la llamada se realiza desde el interior de otra función que también es posible.

Eclipse

Eclipse es un proyecto de desarrollo de software de código fuente abierto (open source) cuyo objetivo es la construcción de herramientas integradas para el desarrollo

de aplicaciones. También sobre el se pueden montar herramientas de desarrollo para cualquier lenguaje, mediante la implementación de los plugins adecuados.

La arquitectura de plugins de Eclipse permite, demás de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo como: herramientas UML, editores visuales de interfaces, ayuda en línea para librerías, etc.

El Consorcio Eclipse

En su origen, el Proyecto Eclipse era un proyecto de desarrollo *OpenSource*, soportado y mantenido en su totalidad por IBM. Bajo la dirección de IBM, se fundó el *Consorcio Eclipse* al cual se unieron algunas empresas importantes como Rational, HP o Borland.

Desde el día 2 de febrero de 2004, el *Consorcio Eclipse* es independiente de IBM y entre otras, está formado por las empresas: HP, QNX, IBM, Intel, SAP, Fujitsu, Hitachi, Novell, Oracle, Palm, Ericsson y RedHat, además de algunas universidades e institutos tecnológicos.

Obtener e instalar Eclipse IDE.

El IDE Eclipse se puede obtener bajándolo directamente del sitio web oficial del *Proyecto Eclipse* - www.eclipse.org - o de cualquier “mirror” autorizado. Existen versiones instalables para cualquier plataforma que soporte la librería SWT, descargas que incluyen el código fuente y descargas que incluyen los plugins más habituales.

Como Eclipse está escrito en Java, en necesario, para su ejecución, que exista un JRE (Java Runtime Environment) instalado previamente en el sistema.

La instalación de Eclipse, es tan sencilla como descomprimir el archivo descargado en el directorio que se estime conveniente.

Ejecutar Eclipse

Las versiones que se pueden descargar del sitio web de Eclipse vienen con un ejecutable que permite lanzar directamente el IDE Eclipse. Antes de ejecutar Eclipse es importante verificar que se tienen permisos de escritura en el directorio, ya que, la primera vez que se ejecuta, Eclipse tiene que crear las carpetas en las que guardará información sobre workspaces, logs, etc.

Postgres

Postgres ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema

- Clase
- Herencia
- Tipos
- Funciones

Otras características aportan potencia y flexibilidad adicional:

- Restricciones (Constraints)
- Disparadores (triggers)
- Reglas (rules)
- Integridad transaccional

PostgreSQL

En 1996, se hizo evidente que el nombre "Postgres95" no resistiría el paso del tiempo. Elegimos un nuevo nombre, PostgreSQL, para reflejar la relación entre el Postgres original y las versiones más recientes con capacidades SQL. Al mismo tiempo, hicimos que los números de versión partieran de la 6.0, volviendo a la secuencia seguida originalmente por el proyecto Postgres.

Durante el desarrollo de Postgres95 se hizo hincapié en identificar y entender los problemas en el código del motor de datos. Con PostgreSQL, el énfasis ha pasado a aumentar características y capacidades, aunque el trabajo continúa en todas las áreas.

Las principales mejoras en PostgreSQL incluyen:

- Los bloqueos de tabla han sido sustituidos por el control de concurrencia multi-versión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde **pg_dump** mientras la base de datos permanece disponible para consultas.

- Se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).

- Se han añadido características adicionales que cumplen el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipos cadenas literales, conversión de tipos y entrada de enteros binarios y hexadecimales.

- Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.

- La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20-40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

Los sistemas de gestión de base de datos con soporte SQL más utilizados son:

- DB2
- Oracle
- SQL Server
- Sybase ASE
- MySQL
- PostgreSQL
- Firebird

VI. DISEÑO METODOLÓGICO

Para el desarrollo de este trabajo usaremos el enfoque metodológico desarrollo en casada, también llamado modelo en cascada, que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar la finalización de la anterior.

Tomando en cuenta que el ciclo de vida son todas las etapas que asocian una serie de tareas que se realizaran incluyendo los documentos que generaran, y que servirán de entrada a las siguientes fases; por las cuales pasa el software, desde que el proyecto es concebido hasta que este es dejado de usarse.

La metodología de desarrollo en cascada puede contemplar los siguientes pasos:

Paso 1: *Análisis de requerimientos.*
Paso 2*: Diseño del sistema.*
Paso 3*: Diseño del programa.*
Paso 4*: Codificación*
Paso 5*: Pruebas*
Paso 6*: Implantación*
Paso 7*: Mantenimiento.*

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nuevo programación del código afectado, aumentando los costes del desarrollo.

Análisis de requerimientos

Se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (Documento de

Especificación de Requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.

Es importante señalar que en esta etapa se deben consensuar todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del software.

Diseño del sistema

Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

Diseño del programa

Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber que herramientas usar en la etapa de Codificación.

Codificación

Es la fase de programación propiamente dicha. Aquí se desarrolla el código_fuente, haciendo uso de prototipos así como pruebas y ensayos para corregir errores.

Dependiendo del lenguaje de programación y su versión se crean las librerías y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

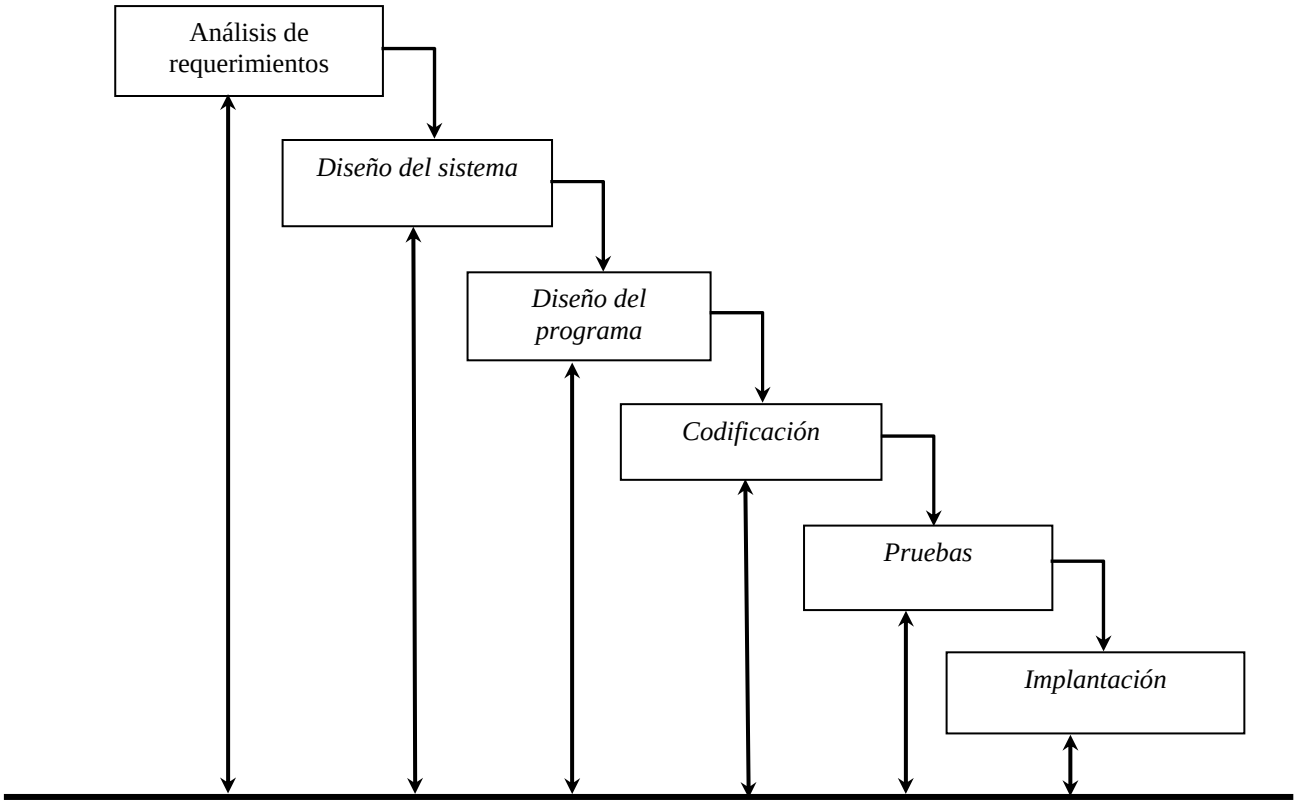
Pruebas

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente antes de ser puesto en explotación.

Implantación

El software obtenido se pone en producción. Se implementan los niveles software y hardware que componen el proyecto. La implantación es la fase con más duración y con más cambios en el ciclo de elaboración de un proyecto. Es una de las fases finales del proyecto

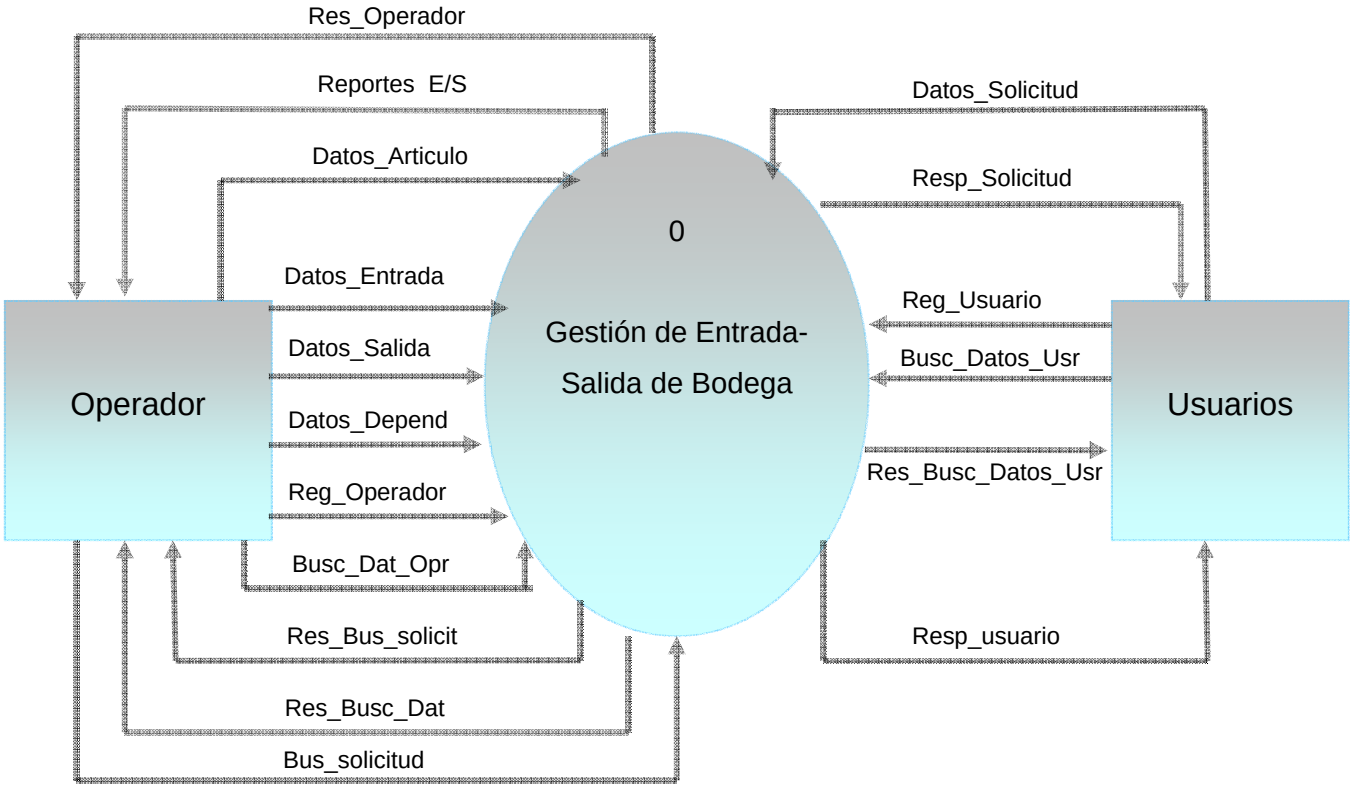
Durante la explotación del sistema software pueden surgir cambios, bien para corregir errores o bien para introducir mejoras. Todo ello se recoge en los Documentos de Cambios.



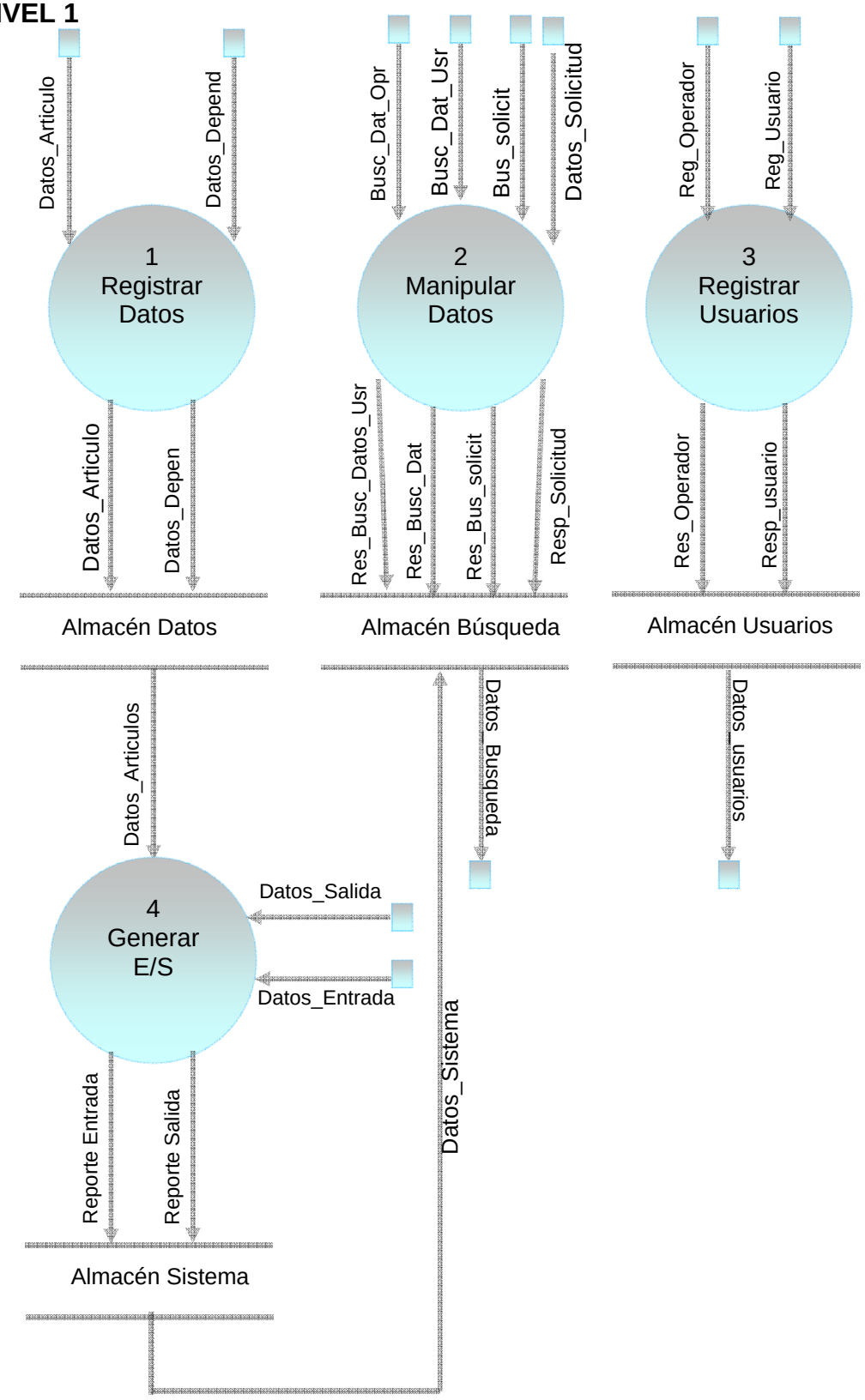
VII. ANÁLISIS

Diagrama de Flujo de Datos

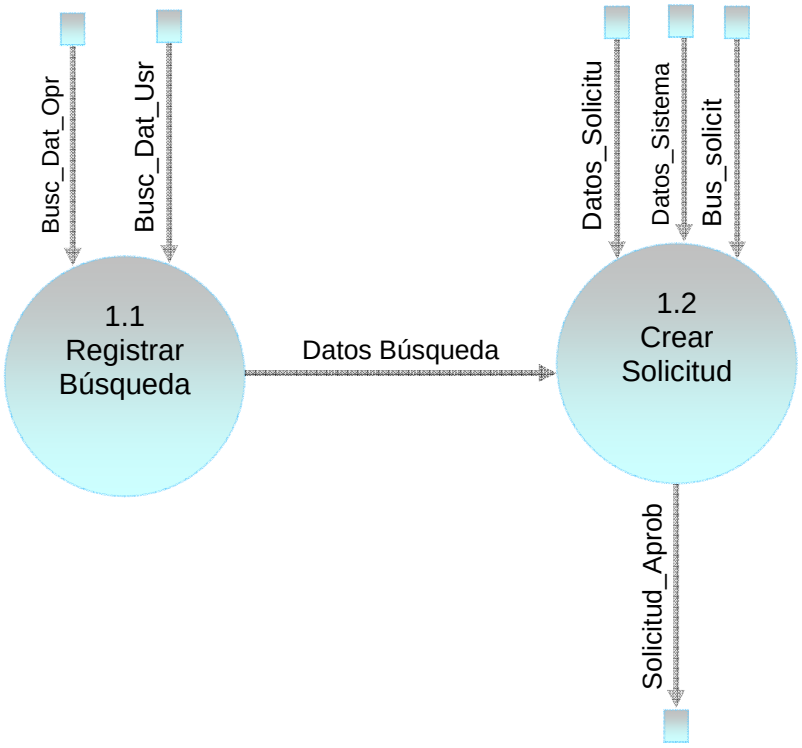
Nivel 0



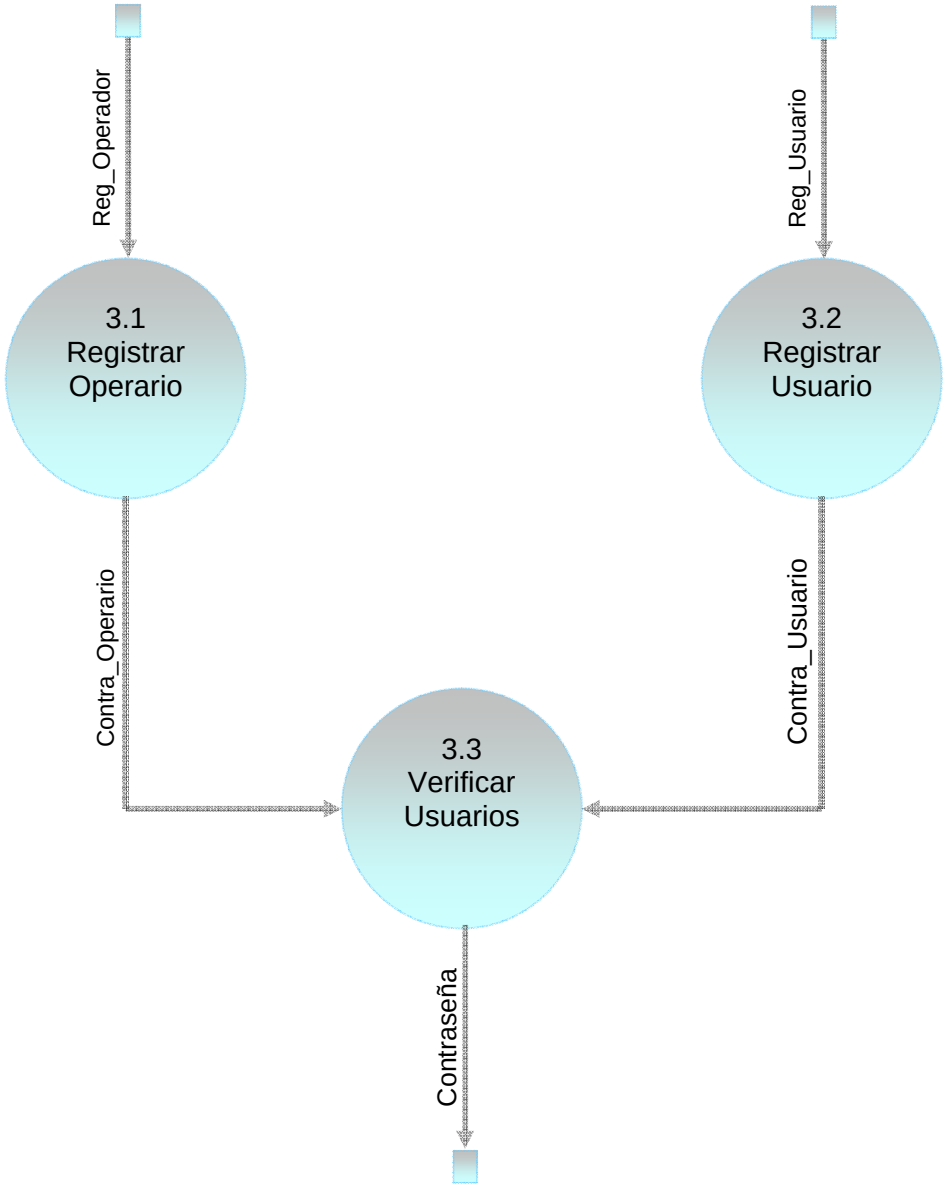
NIVEL 1


Datos_Articulo
Datos_Depend
1
Registrar
Datos
Datos_Articulo
Datos_Depen
Almacén Datos
Busc_Dat_Opr
Busc_Dat_Usr
Bus_solicit
Datos_Solicitud
2
Manipular
Datos
Res_Busc_Datos_Usr
Res_Busc_Dat
Res_Bus_solicit
Resp_Solicitud
Almacén Búsqueda
Reg_Operador
Reg_Usuario
3
Registrar
Usuarios
Res_Operador
Resp_usuario
Almacén Usuarios
Datos_Articulos
4
Generar
E/S
Datos_Salida
Datos_Entrada
Reporte Entrada
Reporte Salida
Almacén Sistema
Datos_Busqueda
Datos_usuarios
Datos_Sistema

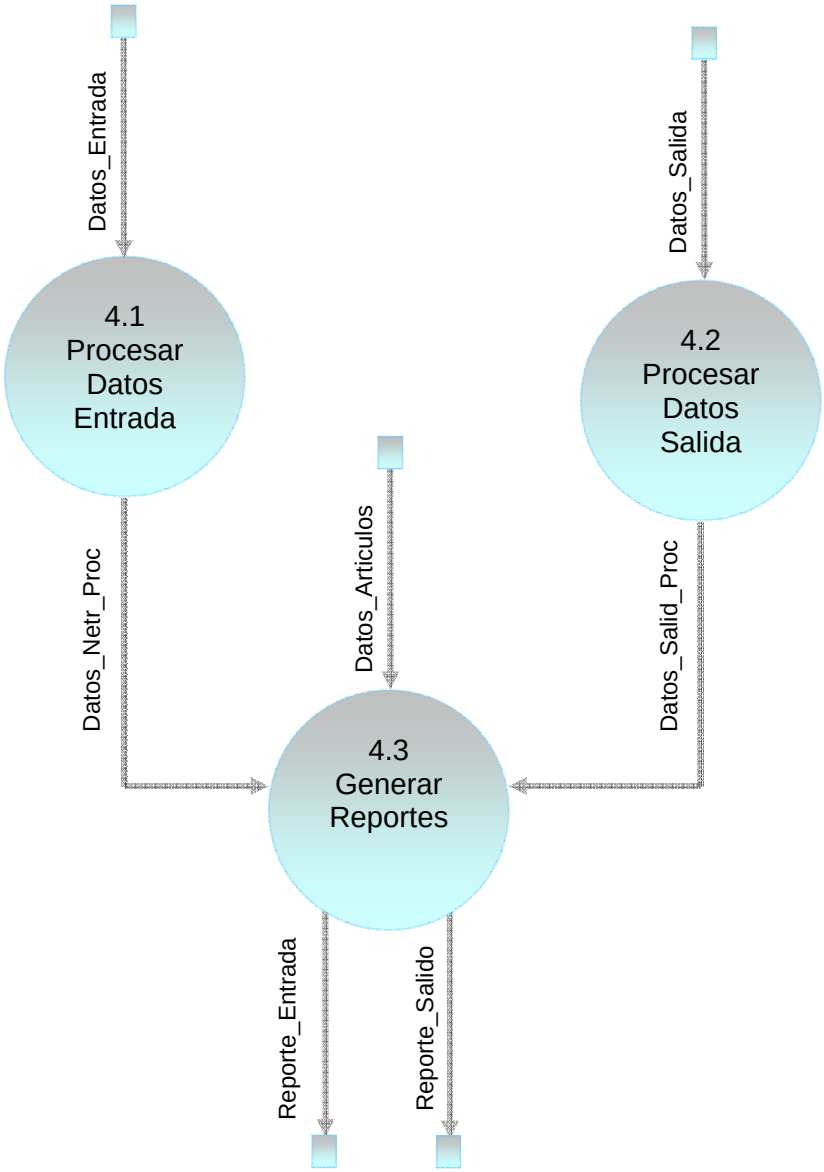
Nivel 2, Diagrama 1



Nivel 2, Diagrama 3



Nivel 2, Diagrama 4



Flujos de Datos

Datos_Articulo: codart + nombreart + umedida + cantidad + valor_unit

Datos_Entrada: cod_trans + fecha + nota_entrada + procedencia + codart + cantidad + valor_unit + valor_total + observaciones.

Datos_Salida: requisa + iden_salida + fecha + cantidad + valor_unit + valor_tota + codigo_dependencia + observaciones + codart

Datos_Depend: codigo_dependencia + nombre_dependencia + monto

Reg_Operador: nombusr + password

Busc_Dat_Opr: Datos_Articulo

Datos_solicitud: numero + nombre_articulo + cantidad1 + unidad1 + cantidad2 + unidad2 + cantidad3 + unidad3 + valor_unit + valor_total + codigo_art + observaciones + nombre_dependencia + fecha

Datos_Sistema: Almacen sistema.

Bus_solicitud: Datos_Solicitud

Busc_Datos_Usr: Datos_Articulo

Reg_usuario: nombusr + password

Reportes_ E/S: datos_entrada + Datos_Salida

Resp_Solicitud: datos_Solicitud

Almacenes de Datos

Almacen Datos: Datos_Articulo + Datos_Depen

Almacén Búsqueda: Res_Busc_Datos_Usr + Res_Busc_Dat + Res_Bus_solicit + Resp_Solicitud + Datos_Sistema

Almacén Usuarios: Res_Operador + Resp_usuario

Almacén Sistema: Reporte Entrada + Reporte Salida

Procesos de Datos

Registrar datos: Este proceso tiene como entradas datos referentes a artículos y dependencias digitados por la parte del operador, teniendo como función registrar los datos en tablas.

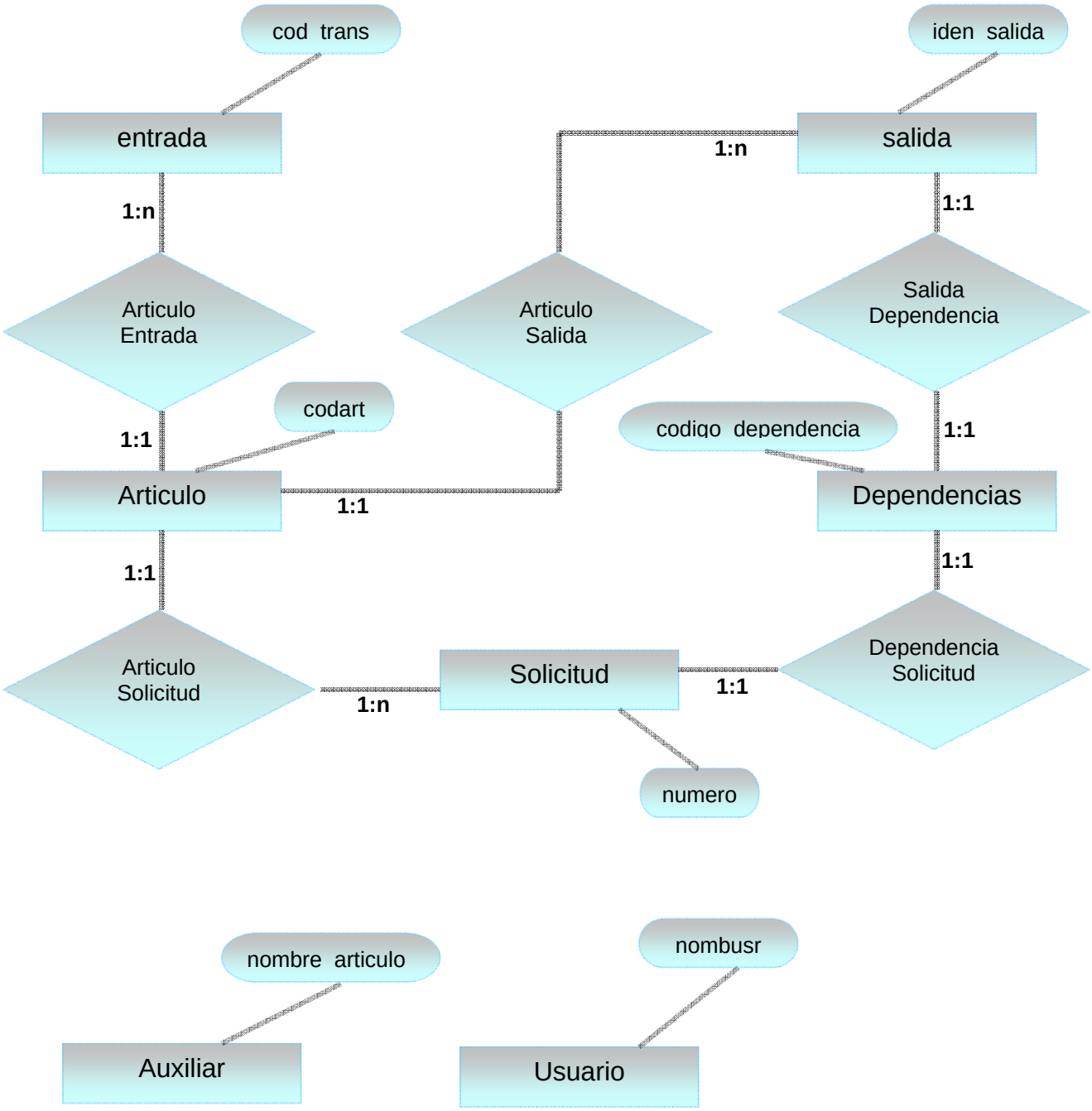
Manipular datos: Una vez almacenados los datos, este proceso hace uso de ellos para que puedan ser manipulados tanto por parte de los usuarios como del operador, específicamente para almacenar lo que es una solicitud, buscar una solicitud.

Registrar Usuarios: Este proceso registra únicamente lo que son los usuarios del sistema, tanto usuarios comunes como el operador.

Generar E/S: Este proceso tiene como finalidad, la de generar reportes de las entradas como de las salidas realizadas al sistema tomando como entradas los datos referentes a artículos, datos de entrada y datos de salida.

Verificar usuarios: Este proceso tiene como función la verificar los usuarios registrados, comprueba si el nombre de usuario y contraseña introducidos son los correctos.

Diagrama Entidad Relación



1. Introducción

1.1 Propósito

Definición del conjunto de especificaciones de requisitos software que debe cumplir la aplicación de **Automatización de Entradas Salidas de la Bodega de la Universidad Nacional Autónoma de Nicaragua (Unan - León)** el cual consiste en la mecanización de proceso de las entradas y salidas del producto de la bodega a las diferentes facultades de la universidad.

1.2 Alcance

El nombre con el que se conocerá dicha aplicación será: **Automatización de Entradas Salidas de la Bodega de la Universidad Nacional Autónoma de Nicaragua (Unan - León)**

El sistema realiza las siguientes funciones:

- Introducir un nuevo producto.
- Anotación de la entrada del producto.
- Anotación de la salida del producto.
- Buscar un determinado artículo.
- Crear y modificar dependencias.
- Generar informes de salidas, entradas y dependencias de los productos.
- Verificar destino.
- Permitir que los usuarios realicen su propia solicitud.

1.3 Definiciones, Acrónimos y Abreviaturas

- **Interfaz:** Conjunto de comandos y métodos que permiten la intercomunicación del programa con cualquier otro programa o elemento interno o externo con un único propósito común.

- **Sistema:** Colección organizada de hombres, máquinas y métodos necesaria para cumplir un objetivo específico.

- **Articulo:** Producto que será introducido a la base de datos para verificar su existencia y tener su control.

- **Entrada:** Cantidad y tipo de producto que deberá ser introducido a la base de datos unas ves adquirido de su procedencia.

- **Salida:** Cantidad de producto que deberá sacarse de la base de datos conociendo su destino y su dependencia.

- **Destino:** Lugar hacia donde van dirigidos los producto.

- **Reporte:** Información generada a partir de un campo específico ya sea de la entrada o la salida como puede ser la fecha.
- **Requisa:** Salidas de productos de la bodega concedidos a las dependencias.
- **Nota de Entrada:** Identifican los productos que entran a la bodega.
- **Unidad Administrativa:** Son los clientes o usuarios que hacen uso del sistema.
- **Productos:** Nombres de objetos o materiales que entran a la Bodega.

1.4 Referencias

- Software de sistema anterior, escrito en FoxPro realizado en el año 1996 por Marcos Morán Rodríguez Y Francisco Danilo Padilla.

- Consultas realizadas al responsable de la bodega indagando a cerca de las entradas y como funciona el sistema de la bodega en realidad para tener mejor noción en la realización del sistema.

1.5 Visión general

El sistema se encarga de facilitar el trabajo al encargado de la bodega al momento de visualizar las entradas y las salidas de los productos (artículos) llevando un mejor control de las existencias.

2. Descripción general

2.1. Relaciones del producto

Nuestro sistema interactúa de forma independiente

El equipo en el que se desarrolló el sistema:

- Emachines
- Pentium IV 2.6 GHz
- 256 Mb RAM
- 40 GB HD

El equipo en el que se implementara:

- Butterfly
- Pentium IV 2.8 GHz
- 256 Mb RAM
- 40 GB HD

2.2. Funciones del producto

El sistema debe contener todas las tareas que realiza manualmente el personal de la bodega de forma diaria. Estas son:

1. Verificar si el nombre de usuario y password son correctos mediante la función ***sesion().***

2. Cuando llegue un nuevo producto este debe ser creado en el sistema mediante la función ***insertar()***, el cual se creará automáticamente en la tabla **Articulo**.

3. Al realizar las entradas de los productos capturarlos con la función ***insertar_entrada(),*** el cual almacenará y modificara automáticamente la tabla **Entrada.**

4. Al realizar las salidas de los productos capturarlos con la función ***registrar_salida(),*** el cual modificara automáticamente la tabla **Salida.**

5. Permite crear dependencias mediante la función ***insertar_dependencia()*****,** lo cual permitirá modificar la tabla **Dependencia.**

6. Modificar las dependencias existentes mediante la función ***modificar_dependencia()*****,** lo cual permitirá actualizar la tabla **Dependencia.**

7. Realizar una búsqueda de los productos existentes en la base de datos por medio de la función ***buscar()*** ya sea por código o por nombre de articulo (producto).

8. Tendrá la opción de llenar solicitudes mediante la función ***solicitud***() que se especifica en la pagina principal.

9. Podrá realizar la búsqueda de una determinada solicitud por medio de la función ***ver_solicitud***() que se especifica en el menú de la pagina principal.

10. Generación de reportes de las entradas y las salidas de los artículos de la bodega por medio de la función ***generar_reporte()*** ya sea por entradas o salidas.

11. También podrá abandonar el sitio cerrando la sesión solamente llamando a la función ***session_destroy()***.

2.3. Características del usuario

Los usuarios finales de nuestro sistema son personas cuyo conocimiento y experiencias informáticas son escasos, por lo cual el sistema fue diseño con una interfaz fácil de usar para un sencillo manejo.

2.4. Restricciones generales

El lenguaje a utilizar será el PHP y se deberá implementar bajo la plataforma Debian_linux y seguir los estándares de la programación estructurada y la orientada a objeto.

2.5. Suposiciones y dependencias

Durante las consultas realizadas a los encargados y a los desarrolladores del sistema se pretende realizar o desarrollar mejoras o aportes que permitan que dicho sistema sea más eficiente y sin menos complicaciones.

3. Requisitos Específicos

3.1 Requisitos funcionales

3.1.1 INTRODUCIR UN NUEVO PRODUCTO

3.1.1.1 Especificación

3.1.1.1.1 Introducción

Este proceso deberá crear un nuevo producto que no se encuentra en la base de datos teniendo presente las características de dicho producto.

3.1.1.1.2 Entradas

Por pantalla: datos para crear el nuevo artículo:

- Código del artículo.
- Nombre del artículo.
- Unidad de medida.

3.1.1.1.3 Proceso

Se mostrará por pantalla el formulario de un nuevo artículo. Donde los datos necesarios a introducir serán:

- ✓ **Código del artículo:** es un dato obligatorio. Este dato todavía no existe en el fichero maestro.
- ✓ **Nombre del artículo**: es un dato obligatorio. Este dato todavía no existe en el fichero maestro**.**
- ✓ **Unidad de medida:** es un dato obligatorio.

3.1.1.1.4 Salidas

Con los datos mencionados se almacenará el nuevo artículo en la base de datos del sistema.

3.1.1.1.5 Interfaces externas

- **Interfaces de usuario.**

La captura de datos del nuevo artículo se realizará de forma interactiva por pantalla.

- **Interfaces hardware.**

  Se podrá utilizar cualquier terminal conectado al ordenador principal

- **Interfaces software.**

  El proceso interactúa con la base de datos ya mencionada.

- **Interfaces de comunicaciones.**

  Uso de la Internet y de intranet que poco se utiliza.

3.1.2 ENTRADAS DE PRODUCTOS A LA BODEGA

3.1.2.1 Especificación

3.1.2.1.1 Introducción

Este proceso se encargará de registrar las entradas de los artículos a la bodega con todos sus datos correspondientes.

3.1.2.1.2 Entradas

Por pantalla: datos para codificar las entradas

- Código de transacción
- Código del articulo
- Fecha
- Nota de entrada
- Cantidad
- Costo unitario
- Procedencia
- Observaciones

Datos proporcionados por el sistema:

Referente al artículo:

- Nombre del articulo

- Unidad de medida

3.1.2.1.3 Proceso

Se mostrará por pantalla el formulario de entrada de datos.

Donde los datos necesarios a introducir serán:

- ✓ **Código de transacción:** es un dato automático.
- ✓ **Código del artículo:** es un dato obligatorio. De be existir en el fichero maestro de articulo o si no indicara error.
- ✓ **Fecha:** es un dato obligatorio.
- ✓ **Nota de entrada:** dato automático.
- ✓ **Cantidad:** dato obligatorio.
- ✓ **Costo unitario:** dato obligatorio.
- ✓ **Procedencia:** dato obligatorio.
- ✓ **Observaciones:** dato opcional.

A partir de estos datos se calculará las entradas de los artículos automáticamente. También se actualizarán las existencias agregándole estas nuevas entradas.

3.1.2.1.4 Salidas

Con los datos antes mencionados se almacenaran las entradas en la base de datos del sistema y se registrará los artículos correspondientes a las entradas.

3.1.2.1.5 Interfaces externas

- **Interfaces de usuario.**

La captura de los datos de las entradas se realizará de forma interactiva por pantalla.

- **Interfaces hardware.**

  Se podrá utilizar cualquier terminal conectado al ordenador principal.

- **Interfaces software**

  El proceso interactúa con la base de datos ya mencionada.

- **Interfaces de comunicaciones**

  Uso de la Internet y de intranet que poco se utiliza.

3.1.3 BÚSQUEDA DE UN DETERMINADO PRODUCTO

3.1.3.1 Especificación

3.1.3.1.1 Introducción

Este proceso se encargará de buscar un artículo ya sea por código o por nombre.

3.1.3.1.2 Entradas

Por pantalla: datos para codificar la búsqueda:

- Código del articulo
- Nombre del articulo

3.1.3.1.3 Proceso

Se mostrará por pantalla el formulario de búsqueda de datos. Donde los datos necesarios a introducir serán:

- ✓ **Código del artículo:** es un dato obligatorio. De be existir en el fichero maestro de articulo o si no indicara error.
- ✓ **Nombre del artículo:** es un dato obligatorio. De be existir en el fichero maestro de articulo o si no indicara error.

A partir de uno de estos datos se realizara la búsqueda del artículo automáticamente.

3.1.3.1.4 Salidas

Con los datos antes mencionados se realizará la búsqueda en la base de datos del sistema y se visualizarán los artículos correspondientes a las entradas de lo contrario enviará un mensaje de error.

3.1.3.1.5 Interfaces externas

- **Interfaces de usuario.**

  La captura de los datos de la búsqueda se realizará de forma interactiva por pantalla.

- **Interfaces hardware.**

  Se podrá utilizar cualquier terminal conectado al ordenador principal

- **Interfaces software**

  El proceso interactúa con la base de datos ya mencionada.

- **Interfaces de comunicaciones**

  Uso de la Internet y de intranet que poco se utiliza.

3.1.4 CREAR DEPENDENCIAS A LA BODEGA

3.1.4.1 Especificación

3.1.4.1.1 Introducción

Este proceso se encargará de registrar las dependencias existentes de la bodega con todos sus datos correspondientes.

3.1.4.1.2 Entradas

Por pantalla: datos para codificar las entradas

- Código _ dependencia
- Nombre _dependencia
- Monto

Datos proporcionados por el sistema:

Referente a dependencia:

- Nombre _dependencia.

3.1.4.1.3 Proceso

Se mostrará por pantalla el formulario de dependencias. Donde los datos necesarios a introducir serán:

- ✓ **Código de la dependencia:** es un dato obligatorio.
- ✓ **Nombre de la dependencia:** es un dato obligatorio. De be existir en el fichero maestro de articulo o si no indicara error.
- ✓ **Monto:** es un dato obligatorio.

A partir de estos datos se crearan las dependencias de la bodega automáticamente.

3.1.4.1.4 Salidas

Con los datos antes mencionados se registrarán las dependencias en la base de datos del sistema y se modificarán los datos.

3.1.4.1.5 Interfaces externas

- **Interfaces de usuario.**

  La captura de los datos de las dependencias se realizará de forma interactiva por pantalla.

- **Interfaces hardware.**

  Se podrá utilizar cualquier terminal conectado al ordenador principal.

- **Interfaces software**

  El proceso interactúa con la base de datos ya mencionada.

- **Interfaces de comunicaciones**

  Uso de la Internet y de intranet que poco se utiliza.

3.1.5 MODIFICAR DEPENDENCIAS DE LA BODEGA

3.1.4.1 Especificación

3.1.4.1.1 Introducción

Este proceso se encargará de actualizar las dependencias existentes de la bodega con todos sus datos correspondientes.

3.1.4.1.2 Entradas

Por pantalla: datos para codificar las entradas

- monto

Datos proporcionados por el sistema:

Referente a dependencia:

- nombre _dependencia.
- codigo_dependencia.
- monto

3.1.4.1.3 Proceso

Se mostrará por pantalla el formulario de dependencias. Donde los datos necesarios a introducir serán:

- ✓ **Monto:** es un dato obligatorio.

A partir de este dato se modificaran las dependencias de la bodega automáticamente.

3.1.4.1.4 Salidas

Con el dato antes mencionado se actualizara la dependencia en la base de datos del sistema y se modificarán los datos.

3.1.4.1.5 Interfaces externas

- **Interfaces de usuario.**

  La captura de los datos de las dependencias se realizará de forma interactiva por pantalla.

- **Interfaces hardware.**

  Se podrá utilizar cualquier terminal conectado al ordenador principal.

- **Interfaces software**

  El proceso interactúa con la base de datos ya mencionada.

- **Interfaces de comunicaciones**

  Uso de la Internet y de intranet que poco se utiliza.

3.1.6 GENERACIÓN DE REPORTE DE ENTRADA

3.1.5.1Especificación

3.1.5.1.1 Introducción

Este proceso se encargará de generar los reportes de las entradas de los artículos a la bodega con todos sus datos correspondientes.

3.1.6.1.2 Entradas

Por pantalla: datos para codificar las entradas

- Fecha

Datos proporcionados por el sistema:

Referente al artículo:

- Nombre del articulo
- Unidad de medida

Referente a entrada:

- Fecha
- Nota de entrada

- Cantidad
- Costo unitario
- Procedencia

Referente a dependencia:

- Nombre dependencia

Referente a salida:

- Fecha
- Requisa

3.1.6.1.3 Proceso

Se mostrará por pantalla el formulario de reportes de datos. Donde los datos necesarios a introducir serán:

- ✓ **fecha:** dato obligatorio.

3.1.6.1.4 Salidas

A partir de este dato se generará el reporte de las entradas de artículos automáticamente.

3.1.6.1.5 Interfaces externas

- **Interfaces de usuario.**

  La captura de los datos de las entradas se realizará de forma interactiva por pantalla.

- **Interfaces hardware.**

  Se podrá utilizar cualquier terminal conectado al ordenador principal

- **Interfaces software**

  El proceso interactúa con la base de datos ya mencionada.

- **Interfaces de comunicaciones**

  Uso de la Internet y de intranet que poco se utiliza.

3.1.7 GENERACIÓN DE REPORTES DE SALIDA

3.1.5.1Especificación

3.1.5.1.1 Introducción

Este proceso se encargará de generar los reportes de las salidas de los artículos a la bodega con todos sus datos correspondientes.

3.1.7.1.2 Entradas

Por pantalla: datos para codificar las salidas

- Fecha
- Requisa
- cantidad

Datos proporcionados por el sistema:

Referente al artículo:

- Nombre del articulo
- Unidad de medida

Referente a dependencia:

- Nombre dependencia

3.1.7.1.3 Proceso

Se mostrará por pantalla el formulario de reportes de datos. Donde los datos necesarios a introducir serán:

- ✓ **fecha:** dato obligatorio.

3.1.7.1.4 Salidas

A partir de este dato se generará el reporte de salida de los artículos automáticamente.

3.1.7.1.5 Interfaces externas

- **Interfaces de usuario.**

  La captura de los datos de las salidas se realizará de forma interactiva por pantalla.

- **Interfaces hardware.**

  Se podrá utilizar cualquier terminal conectado al ordenador principal

- **Interfaces software**

  El proceso interactúa con la base de datos ya mencionada.

- **Interfaces de comunicaciones**

  Uso de la Internet y de intranet que poco se utiliza.

3.1.8 GENERACIÓN DE REPORTES POR DEPENDENCIA.

3.1.5.1Especificación

3.1.5.1.1 Introducción

Este proceso se encargará de generar los reportes de las dependencias de la bodega con todos sus datos correspondientes.

3.1.8.1.2 Entradas

Por pantalla: datos para codificar las entradas

- Codigo dependencia.

Datos proporcionados por el sistema:

Referente al artículo:

- Nombre del articulo
- Unidad de medida

Referente a entrada:

- Fecha
- Nota de entrada
- Cantidad
- Costo unitario
- Procedencia

Referente a salida:

- Fecha
- Requisa

3.1.8.1.3 Proceso

Se mostrará por pantalla el formulario de reportes de datos. Donde los datos necesarios a introducir serán:

.

- ✓ **Código de dependencia:** es un dato obligatorio. De be existir en el fichero maestro dependencia o si no indicara error.

3.1.8.1.4 Salidas

A partir de este dato se generará los reportes de las dependencias automáticamente.

3.1.8.1.5 Interfaces externas

- **Interfaces de usuario.**

  La captura de los datos de las entradas y las salidas se realizará de forma interactiva por pantalla.

- **Interfaces hardware.**

  Se podrá utilizar cualquier terminal conectado al ordenador principal

- **Interfaces software**

  El proceso interactúa con la base de datos ya mencionada.

- **Interfaces de comunicaciones**

  Uso de la Internet y de intranet que poco se utiliza.

3.1.9 REALIZAR UNA SOLICITUD A LA BODEGA

3.1.9.1 Especificación

3.1.6.1.1 Introducción

Este proceso le permitirá al usuario generar su propia solicitud de productos ala bodega con todos sus datos correspondientes.

3.1.6.1.2 Entradas

Por pantalla: datos para codificar las entradas

- cantidad
- Nombre del articulo
- Unidad de medida

Datos proporcionados por el sistema:

Referente a dependencia:

- Nombre _dependencia.

3.1.6.1.3 Proceso

Se mostrará por pantalla el formulario de solicitud. Donde los datos necesarios a introducir serán:

- ✓ **Cantidad:** es un dato obligatorio.
- ✓ **Unidad de medida:** es un dato obligatorio. De be existir en el fichero maestro de articulo o si no indicara error.
- ✓ **Nombre del artículo:** es un dato obligatorio.

A partir de estos datos se creara la solicitud de la bodega automáticamente.

3.1.6.1.4 Salidas

Con los datos antes mencionados se registrara la solicitud en la base de datos del sistema y verificara si hay en existencias.

3.1.6.1.5 Interfaces externas

- **Interfaces de usuario.**

  La captura de los datos de la solicitud se realizará de forma interactiva por pantalla.

- **Interfaces hardware.**

  Se podrá utilizar cualquier terminal conectado al ordenador principal.

- **Interfaces software**

  El proceso interactúa con la base de datos ya mencionada.

- **Interfaces de comunicaciones**

  Uso de la Internet y de intranet que poco se utiliza.

3.1.10 SALIDAS DE PRODUCTOS DE LA BODEGA

3.1.10.1 Especificación

3.1.10.1.1 Introducción

Este proceso se encargará de registrar las salidas de los artículos a la bodega con todos sus datos correspondientes.

3.1.10.1.2 Entradas

Por pantalla: datos para codificar las entradas

- Identificador de salida.
- Código del articulo

- Fecha
- Requisa
- Cantidad
- Costo unitario
- Código dependencia
- Observaciones

Datos proporcionados por el sistema:

Referente al artículo:

- Nombre del articulo
- Unidad de medida.
- Valor unitario.

Referente a dependencia:

- Código dependencia.
- Monto.

Referente a salida:

- Identificador salida.
- Requisa.

Referente a solicitud:

- Cantidad
- Observación.

A partir de estos datos se calculará las entradas de los artículos automáticamente. También se actualizarán las existencias agregándole estas nuevas entradas.

3.1.10.1.3 Proceso

Se mostrará por pantalla un combo donde se pide seleccionar el número de solicitud para luego generar una salida a partir de esa solicitud.

3.1.10.1.4 Salidas

Con los datos proporcionados por el sistema se generaran y almacenaran las salidas en la base de datos del sistema y se actualizaran los artículos y dependencias relacionados con la solicitud.

3.1.10.1.5 Interfaces externas

- **Interfaces de usuario.**

  La captura de los datos de las salidas se realizará de forma interactiva por pantalla.

- **Interfaces hardware.**

  Se podrá utilizar cualquier terminal conectado al ordenador principal.

- **Interfaces software**

  El proceso interactúa con la base de datos ya mencionada.

- **Interfaces de comunicaciones**

  Uso de la Internet y de intranet que poco se utiliza.

4. Requisitos de funcionamiento

- **Requisitos estáticos**

  No existe ninguna restricción sobre el número de terminales y usuarios que estén trabajando simultáneamente con el sistema, también el número de ficheros y tamaño de estos varía.

- **Requisitos dinámicos**

  Es importante que el tiempo de respuesta no aumente exponencialmente con el número de usuarios y de ficheros que estén trabajando simultáneamente.

5. Restricciones de diseño

El formato de pantalla y listados del sistema deberá contener información acerca del nombre de la empresa, el nombre del usuario que realiza el trabajo, la fecha y la hora.

6. Atributos

- **Seguridad**

  El sistema esta protegido mediante autorización de uso.

- **Mantenibilidad**

  Cualquier modificación que afecte estos requisitos deberá ser reflejada en el mismo, así como la documentación obtenida en las fases de análisis, diseño y programación.

Otros requisitos

Base de datos

El almacenamiento de información se realizara por medio de una base de datos relacional.

Operaciones.

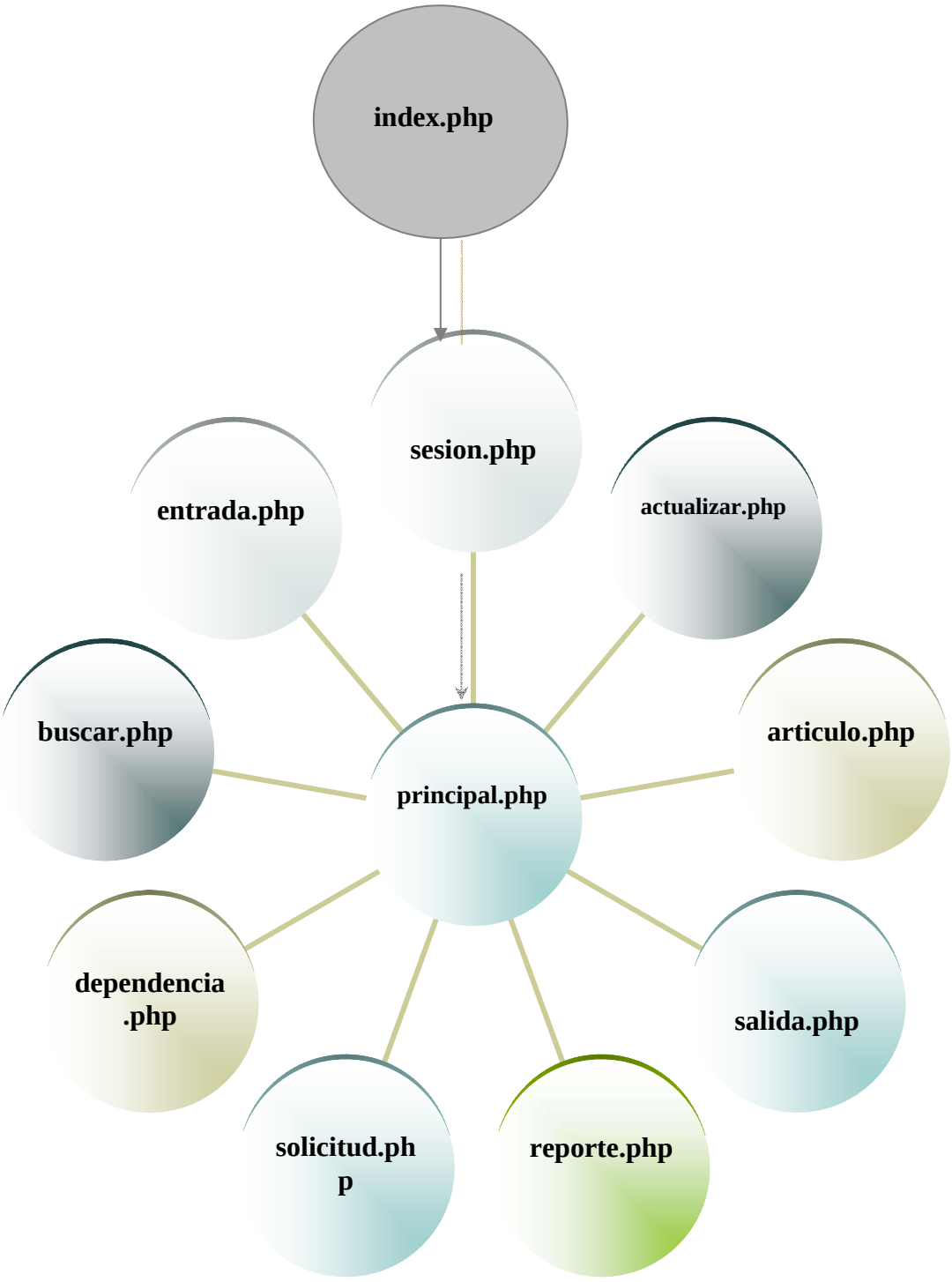
Todas las operaciones sobre la base de datos se realizarán siguiendo lo mencionado en el subapartado de seguridad.

VIII. DISEÑO

Diseño Arquitectónico

Articulo	Actualizar	Dependen	Buscar	Solicitud	Reportes	Salir
Nuevo	Entradas	Crear	Nom/Cód	Crear	Entrada	Cerrar sesión
Entrada	Salidas	Modificar		Buscar	Salida	
Salida					Depend	
					Dep_Usr	

Mapa de Sitio



Diseño de Datos

1. Tabla Articulo		
Nombre del atributo	**Tipo del articulo**	**Tamaño del atributo**
codart	carácter	10
nombreart	varchar	50
umedida	varchar	15
cantidad	integer	
valor_unit	float	

2. Tabla usuario		
Nombre del atributo	**Tipo del articulo**	**Tamaño del atributo**
nombreusr	varchar	35
password	varchar	35

3. Tabla dependencias		
Nombre del atributo	**Tipo del articulo**	**Tamaño del atributo**
codigo_dependencia	varchar	20
nombre_dependencia	varchar	50

4. Tabla entrada		
Nombre del atributo	**Tipo del articulo**	**Tamaño del atributo**
cod_trans	integer	
fecha	varchar	20
nota_entrada	smalling	
procedencia	varchar	50
codart	varchar	
cantidad	integer	
valor_unit	float	
valor_total	float	
observaciones	varchar	30

5. Tabla salida		
Nombre del atributo	**Tipo del articulo**	**Tamaño del atributo**
requisa	smallint	
iden_salida	integer	
fecha	varchar	25
cantidad	integer	50
valor_unit	float	
valor_total	float	
codigo_dependencia	varchar	20
observaciones	varchar	50
codart	carácter	10

6. Tabla auxiliar		
Nombre del atributo	**Tipo del articulo**	**Tamaño del atributo**
numero	varchar	25
nombre_ articulo	varchar	25
cantidad1	varchar	20
unidad1	varchar	25
cantidad2	varchar	20
unidad2	varchar	25
cantidad3	varchar	20
unidad3	varchar	25
valor_unit	float	
valor_total	float	
codigo_art	varchar	20
observaciones	varchar	50

7. Tabla solicitud		
Nombre del atributo	**Tipo del articulo**	**Tamaño del atributo**
numero	varchar	25
nombre_ articulo	varchar	25
cantidad1	varchar	20
unidad1	varchar	25
cantidad2	varchar	20
unidad2	varchar	25
cantidad3	varchar	20
unidad3	varchar	25
valor_unit	float	
valor_total	float	
codigo_art	varchar	20
observaciones	varchar	50

IX. CONCLUSIONES

Al finalizar nuestro trabajo monográfico consideramos que hemos cumplido con los objetivos propuestos, al actualizar de forma eficiente un sistema de información automatizado, que facilitara el ccontrol de Entradas-Salidas de la Bodega Central de la Unan-León utilizando “Software Libre” como herramienta de desarrollo para la aplicación.

En el desarrollo de nuestro sistema utilizando Linux Debian (Sistema Operativo), PHP (Lenguaje de Programación), PosgreSQL (Sistema Gestor de Base de Datos), Apache (Servidor Web) y Eclipse (Entorno de desarrollo), hemos desarrollado nuevos conocimientos y podemos concluir que son herramientas potentes y de fácil uso en el desarrollo de Sistemas de Información.

En este trabajo observamos la importancia que tienen los sistemas automatizados de información en red para cualquier institución que desee llevar un mejor manejo y control de información.

X. RECOMENDACIONES

Todo sistema esta compuesto por una serie de funciones y apartados que con el tiempo lo hace fiable y también permite el desarrollo de la institución que lo utiliza por lo que creemos que seria necesario llevar a cabo las siguientes recomendaciones:

- La capacitación de las personas que van hacer uso o estén en contacto directo con el sistema.

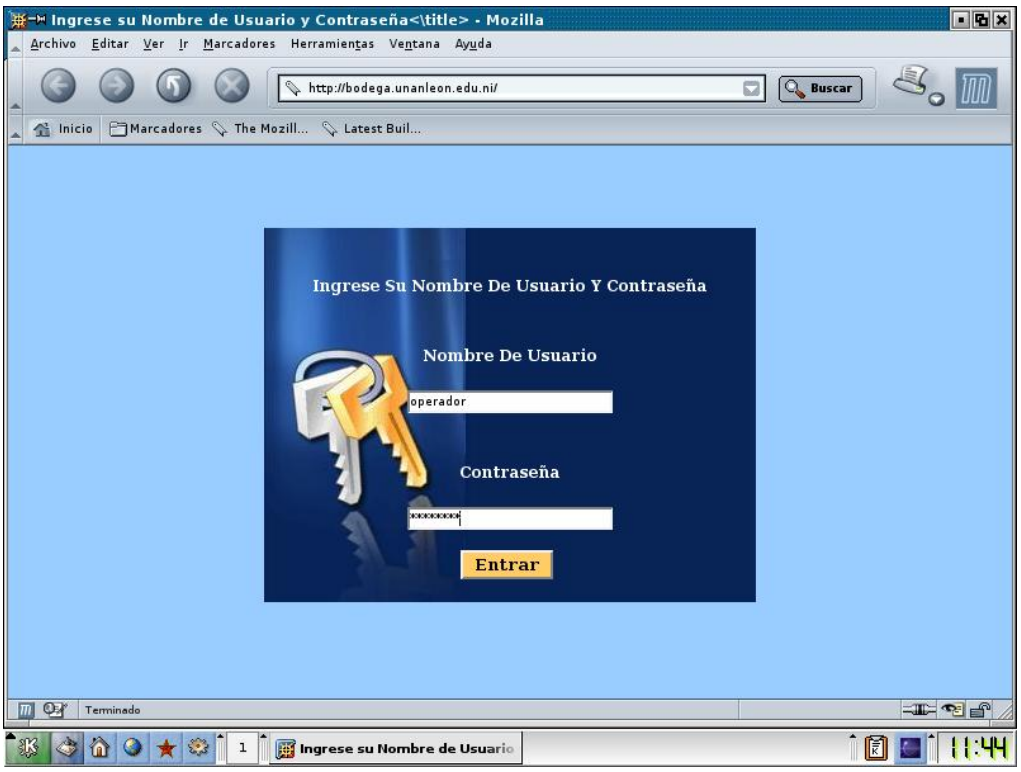
- Estar en contacto con las nuevas actualizaciones que sean necesarias en dicha aplicación de acuerdo a los requerimientos de los usuarios para que nuestro sistema evolucione.

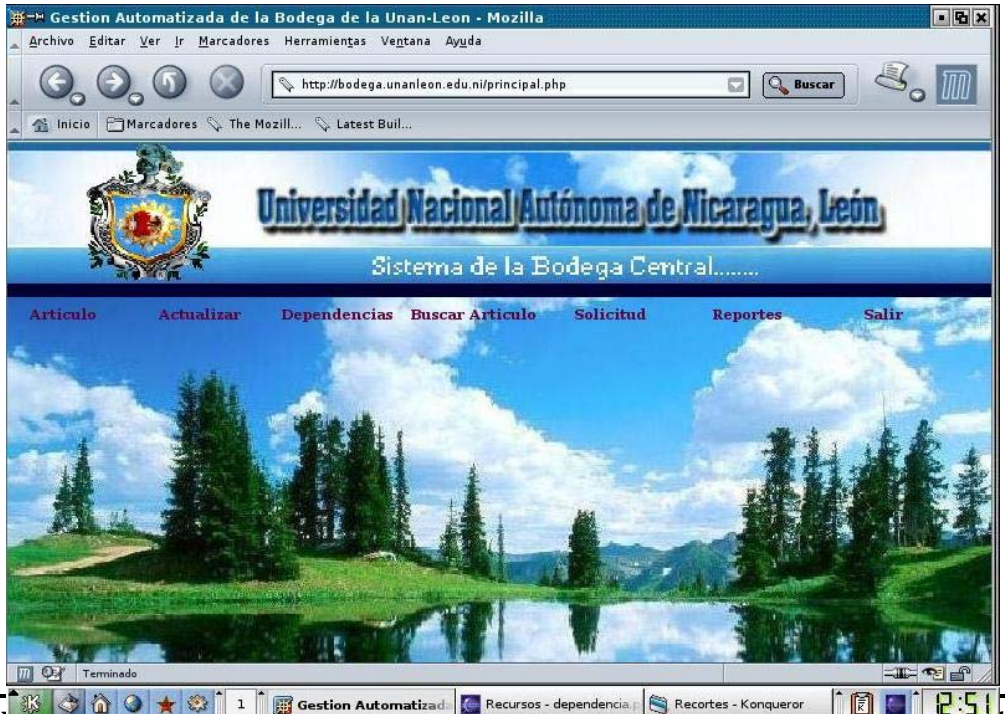
XI. BIBLIOGRAFÍA

1. **Internet**
    - **www.monografias.com**
    - **www.desarrolloweb.com**
    - **www.php.net**
    - **www.bulma.com**

2. **Manuales acerca de postgresql y php.**

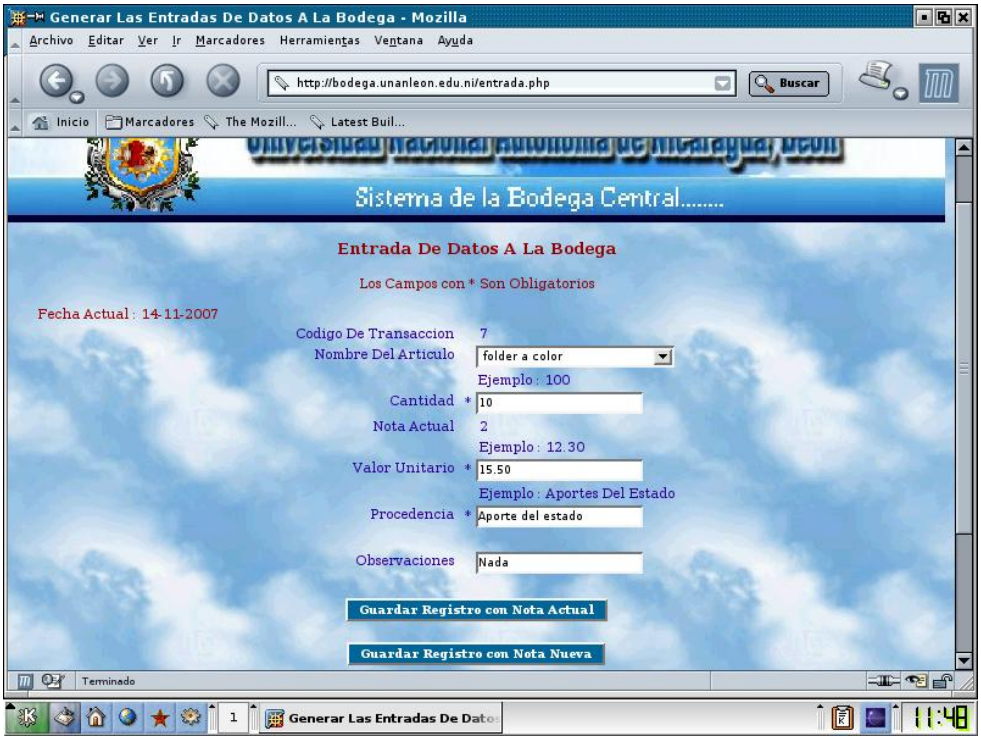
XII . ANEXOS


Ingrese su Nombre de Usuario y Contraseña<\title> - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Ingrese Su Nombre De Usuario Y Contraseña
Nombre De Usuario
operador
Contraseña
Entrar
Terminado
Ingrese su Nombre de Usuario
11:44


Gestion Automatizada de la Bodega de la Unan-Leon - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/principal.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Universidad Nacional Autónoma de Nicaragua, León
Sistema de la Bodega Central.....
Articulo Actualizar Dependencias Buscar Articulo Solicitud Reportes Salir
Terminado
Gestion Automatizad
Recursos - dependencia...
Recortes - Konqueror
12:51


Crear Un Articulo - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/articulo.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Universidad Nacional Autónoma de Nicaragua, León
Sistema de la Bodega Central.....
Crear Nuevos Articulos
Los Campos Con * Son Obligatorios
Buscar Ultimo Codigo Asignado
Buscar Codigo
Ejemplo : 326598
Codigo *
341210
Ejemplo : PAPEL BOND
Nombre *
Aceite
Unidad Medida *
Litro
Guardar Registro
Limpiar
Terminado
Crear Un Articulo - Mozilla
11:46


Crear Un Articulo - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/articulo.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Universidad Nacional Autónoma de Nicaragua, León
Sistema de la Bodega Central.....
[JavaScript Application]
Registro guardado
Aceptar
Buscar Ultimo Codigo As
Buscar Codigo
Ejemplo : 326598
Codigo *
Ejemplo : PAPEL BOND
Nombre *
Unidad Medida *
Elija Unidad de Medida
Guardar Registro
Limpiar
Transfiriendo datos desde bodega.unanleon.edu.ni
Crear Un Articulo - Mozilla
11:46


Generar Las Entradas De Datos A La Bodega - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/entrada.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Sistema de la Bodega Central.....
Entrada De Datos A La Bodega
Los Campos con * Son Obligatorios
Fecha Actual : 14-11-2007
Codigo De Transaccion 7
Nombre Del Articulo folder a color
Ejemplo : 100
Cantidad * 10
Nota Actual 2
Ejemplo : 12.30
Valor Unitario * 15.50
Ejemplo : Aportes Del Estado
Procedencia * Aporte del estado
Observaciones Nada
Guardar Registro con Nota Actual
Guardar Registro con Nota Nueva
Terminado
Generar Las Entradas De Dato
11:48


Generar Las Entradas De Datos A La Bodega - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/entrada.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Universidad Nacional Autónoma de Nicaragua, León
Sistema de la Bodega Central.....
[JavaScript Application]
Entrada Guardada
Aceptar
Fecha Actual : 14-11-2007
Nombre Del Articulo papel bond tamaNo carta
Ejemplo : 100
Cantidad *
Nota Actual 2
Ejemplo : 12.30
Valor Unitario *
Ejemplo : Aportes Del Estado
Procedencia *
Observaciones
Transfiriendo datos desde bodega.unanleon.edu.ni
Generar Las Entradas De Dato
11:49


Generar Salida de Datos - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/salida.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Universidad Nacional Autónoma de Nicaragua, León
Sistema de la Bodega Central.....
Buscar Numero De Solicitud
Fecha Actual : 14-11-2007
3
Buscar Solicitud
Pagina Principal
Terminado
Generar Salida de Datos - Mozi
11:50

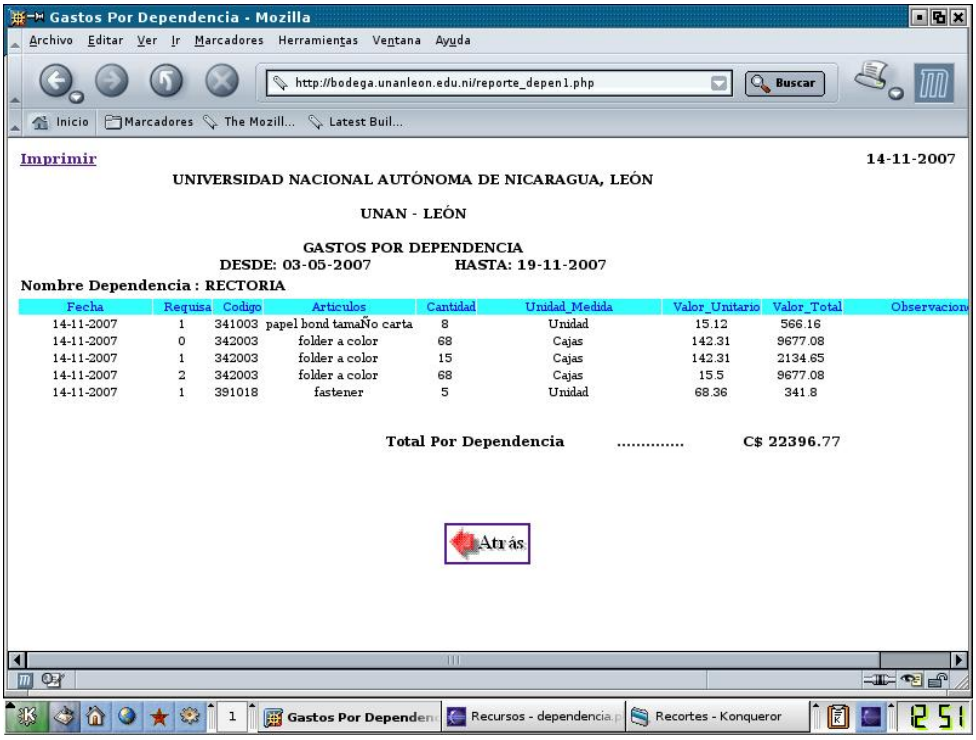

Generar Salida de Datos - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/sa2.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Universidad Nacional Autónoma de Nicaragua, León
Sistema de la Bodega Central.....
[JavaScript Application]
Salida Registrada
Aceptar
Requiza De Materiales No. 3
Descripcion Del Articulo
Codigo
Entregado
Unidad
P. Unitario
Sub-Total
Observaciones
folder a color
342003
Unidad
142.31
9677.08
TOTAL REQUIZADO
9677.08
Requiza Realizada Por RECTORIA el 13-11-2007
Guardar
Transfiriendo datos desde bodega.unanleon.edu.ni
Generar Salida de Datos - Mozi
11:50


Buscar Un Articulo En La Base De Datos De Bodega - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/buscar.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Universidad Nacional Autónoma de Nicaragua, León
Sistema de la Bodega Central.....
Busqueda De Datos En La Bodega
Los Campos con * Son Obligatorios
Introduzca Nombre O Codigo
Ejemplo :391018 / fastener
Codigo / Nombre * fastener
Buscar Registro
Datos existentes
Codigo Articulo | Nombre Articulo | Cantidad | Valor Unitario
341003 | papel bond tamaÑo carta | 69 | 15.12
341007 | papel bond tamaÑo legal | 100 | 89.81
342003 | folder a color | -149 | 15.5
342004 | folder a color tamaÑo legal | 480 | 159.79
Terminado
Buscar Un Articulo En La Base
Recursos - reporte_depen.php - Pla
2:40


Crear Las Dependencias En La Base De Datos De Bodega - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/dependencia.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Universidad Nacional Autónoma de Nicaragua, León
Sistema de la Bodega Central.....
Añadir Nuevas Dependencias
Los Campos Con * Son Obligatorios
Ejemplo : A01
Codigo Dependencia * B02
Ejemplo : RECTORIA
Nombre Dependencia * INTERNADO PREPARATORIA
Ejemplo : 15000.50
Monto Disponible * 25000.50
Guardar Registro
Limpiar
Dependencia Existentes
Terminado
Crear Las Dependenci
Recursos - reporte_depen
Recortes - Konqueror
2:42


Crear Las Dependencias En La Base De Datos De Bodega - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/dependencia.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Universidad Nacional Autónoma de Nicaragua, León
Sistema de la Bodega Central.....
[JavaScript Application]
Dependencia guardada....
Aceptar
Nombre Dependencia *
Ejemplo : 15000.50
Monto Disponible *
Guardar Registro
Limpiar
Transfiriendo datos desde bodega.unanleon.edu.ni
Crear Las Dependenci
Recursos - reporte_depen
Recortes - Konqueror
2:49


Reporte Por Dependencias De Articulos De La Bodega - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/reporte_depen.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Universidad Nacional Autónoma de Nicaragua, León
Sistema de la Bodega Central.....
Generar Reporte Por Costo De Dependencias
Los Campos con * Son Obligatorios
Debe Introducir Dia, Mes y Año
Ej.= 18-10-2007
Fecha Inicial: * 03-05-2007
Ej.= 02-05-2007
Fecha Final: * 19-11-2007
Nombre Dependencia * RECTORIA
Generar Reporte
Pagina Principal
Terminado
Reporte Por Depende
Recursos - dependencia.p
Recortes - Konqueror
2:50


Gastos Por Dependencia - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/reporte_depen1.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Imprimir
14-11-2007
UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, LEÓN
UNAN - LEÓN
GASTOS POR DEPENDENCIA
DESDE: 03-05-2007 HASTA: 19-11-2007
Nombre Dependencia : RECTORIA
Fecha Requisa Codigo Articulos Cantidad Unidad_Medida Valor_Unitario Valor_Total Observacion
14-11-2007 1 341003 papel bond tamaÑo carta 8 Unidad 15.12 566.16
14-11-2007 0 342003 folder a color 68 Cajas 142.31 9677.08
14-11-2007 1 342003 folder a color 15 Cajas 142.31 2134.65
14-11-2007 2 342003 folder a color 68 Cajas 15.5 9677.08
14-11-2007 1 391018 fastener 5 Unidad 68.36 341.8
Total Por Dependencia ..... C$ 22396.77
Atrás
Gastos Por Dependen
Recursos - dependencia
Recortes - Konqueror
2:51


Gestion Automatizada de la Bodega de la Unan-Leon - Mozilla
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
http://bodega.unanleon.edu.ni/Cerrar.php
Buscar
Inicio Marcadores The Mozill... Latest Buil...
Universidad Nacional Autónoma de Nicaragua, León
Sistema de la Bodega Central.....
[JavaScript Application]
Su Sesion Ha Sido Cerrada Correctamente
Aceptar
Articulo
Actualizar
Reportes
Salir
Cerrar Sesion
Transfiriendo datos desde bodega.unanleon.edu.ni
Gestion Automatizad
Recursos - dependencia
Recortes - Konqueror
2:51

Codificación

//Index

```
<?php
              session_register("n_usuario");
              $_SESSION['n_usuario'] = "";
?>

<html>
<head>
  <title> Ingrese su Nombre de Usuario y  Contrase&ntilde;a<\title>

</style>
  <link rel = "stylesheet" type="text/css" href="./estilo/estilo3_1.css">
  <style type = "text/css">object{visibility:hidden;}</style>
  <style type = "text/css">object{visibility:visible;}</style>
</head>
<br>
<br><br>
<table  border = "0" width="50%" height = "10%"cellpadding="0" cellspacing="1"
align="center" background = "./imagenes/key.jpg" align = "center">
<tr>
<td>

<table width="90%" height = "10%" cellpadding="0" cellspacing="1"
align="center"  border="0">

<body bgcolor = 99CCFF>
<form action = 'sesion.php' method = 'post'>
<tr>
  <td align = "center"><br><h5>Ingrese Su Nombre De Usuario Y
Contrase&ntilde;a</h5>
</td>
</tr>

<table width="20%" height = "10%" cellpadding="1" cellspacing="1"
align="center"  border="0" >
<tr>
  <td align = "center"><br><h5>Nombre De Usuario</h5>   <input type = 'text'
name = 'usr' id = 'usr' size=30 maxlength=80>
</tr>
<tr>
  <td align = "center"><br><h5>Contraseña</h5>          <input type =
'password' name = 'contr' id = 'contr' size=30 maxlength=80>
</tr>

</table>
```

```
<p align="center">
  <button type = 'submit'  size = "10" name = 'enviar' value =
'entrar'>Entrar</button>
<br><br>

</form>
</table>
</td></tr></table>

</td></tr></table>
</body>
</html>
```

//Principal

```
  <?php
  include_once("verificar.php");
  session_start();

  if($_SESSION['n_usuario'] != "operador" && $_SESSION['n_usuario'] !=
  "usuario")
  {
        echo "<script type 'text/javascript'>
                alert ('Acceso Denegado en principal');
                parent.location = 'index.php';
                </script> ";
              exit;
  }
  else
  {
  ?>
  <html>
  <title>Gestion Automatizada de la Bodega de la Unan-Leon</title>
  <head>
  <style type=text/css>.noright {
        MARGIN-LEFT: 5px
  }
  .noleft {
        MARGIN-RIGHT: 5px
  }
  .week {
        WIDTH: 350px
  }
  </style>

        <link href="./estilo/estilo2.css" type=text/css rel=stylesheet>
        <link href="./menu/menu.css" type=text/css rel=stylesheet>
        <link rel="stylesheet" type="text/css" href="./estilo/estilos.css">
        <style type="text/css">object{visibility:hidden;}</style>
        <style type="text/css">object{visibility:visible;}</style>
  </head>

  <body background = "./imagenes/b.jpg" >
  <table width="100%" align=center cellpadding='0' cellspacing='0'>
```

```
<tr>
      <td height="126"  bgcolor="#000033">
      <img src="imagenes/logo.jpg" width="100%" height="115"
align="absbottom">
      </td>
</tr>
</table>

<br>
<div id=menu>
<ul id=nav>

<li>
      Articulo
<ul>
<li>
      <a href="http:./articulo.php">Nuevo</a>
</li>
<li>
      <a href="http:./entrada.php">Entrada</a>
</li>
<li>
      <a href="http:./salida.php">Salida</a>
</li>
</ul>
</li>

<li>
      Actualizar
<ul>
<li>
      <a href="http:./mod_entrada.php">Entrada</a>
</li>
<li>
      <a href="http:./mod_salida.php">Salida</a>
</li>
</ul>
</li>
<li>
      Dependencias
<ul>
<li>
      <a href="http:./dependencia.php">Dependencias</a>
</li>
<li>
      <a href="http:./mod_depend.php">Modificar</a>
</li>
</ul>
</li>

<li>
      Buscar Articulo
<ul>
<li>
      <a href="http:./buscar.php">Nombre/Codigo </a>
</li>
```

```
</ul>
</li>

<li>
      Solicitud
<ul>
<li>
      <a href="http:./solicitud.php">Crear</a>
</li>
<li>
      <a href="http:./ver_solicitud.php">Buscar</a>
</li>
</ul>
</li>

<li>
      Reportes
<ul>
<li>
      <a href="http:./reporte_entrada.php">Entradas </a>
</li>
<li>
      <a href="http:./reporte_salida.php">Salidas</a>
</li>
<li>
      <a href="http:./reporte_depen.php">Dependencia</a>
</li>
<li>
      <a href="http:./reporte_depen_usuario.php">Dependencia Usuario</a>
</li>
</ul>
</li>

<li>
      Salir
<ul>
<li>
      <a href="http:./Cerrar.php">Cerrar Sesion</a>
</li>
</ul>
</div>

</body>
</html>

<?php
}
?>
```

//Articulo

```
<?php
include_once("verificar.php");

session_start();
```

```
if($_SESSION['n_usuario'] != "operador")

{
      echo "<script type 'text/javascript'>
              alert ('Acceso Denegado');
              history.back();
              </script> ";
      exit;
}
else
{
      $conexion = Conexion();
      if($conexion == false)
      {
            echo "No pudo abrirse la conexion a $database";
            exit (-1);
      }
?>
<html>
<head>
      <title> Crear Un Articulo</title>
      <link rel="stylesheet" type="text/css"
href="./estilo/estilo2_1.css">

<script language = "javascript" type = "text/javascript">
function validar(fr)
{
      if(fr.codart.value.search(/^\d{6}$/) == -1)
      {
            alert("Error. Codigo Articulo Debe Contener 6 Numeros");
            fr.codart.focus();
            return false;
      }
      if(fr.nombreart.value.search(/(\D+\s\D+)|(\D+)$/) == -1)
            {
                  alert("Error. Nombre Articulo Debe Contener Solo
Letras");
                  fr.nombreart.focus();
                  return false;
            }

      if(confirm("¿Desea Guardar el Registro?") == true)
      {
            return true;
      }
      else
      {
            return false;
      }
}
</script>
</head>
<body leftmargin='0' topmargin='0' background =
"./imagenes/nubecilla.gif">

<table width="100%" align="center" cellpadding='0' cellspacing='0'>
```

```
<tr>
      <td height="126"  bgcolor="#000033">
      <img src="imagenes/logo.jpg" width="100%" height="120"
align="absbottom">
      </td>
</tr>
</table>

<h3><p align =  "center">Crear Nuevos Articulos</p></h3>
<p align =  "center"> Los Campos Con * Son Obligatorios </p>
<center>

<table width = "70%" cellpadding = "1" cellspacing = "1" aling = "center"
border = "0">
<form method = "POST" name = "formu_insercion1" action =" articulo.php">

<tr>
      <td align = "center">
      Buscar Ultimo Codigo Asignado
      </td>

      <td align = "left">   
      <input type = 'text' name = 'ultimo'
size='25'>          &nb
sp;    
      <button type = 'submit'  size = "10" name = 'b_ultimo' value =
'Buscar Codigo'>Buscar Codigo</button>
      <input type = hidden name = 'pagina' value = '1'/>
      </td>
</tr>
<tr>
      <td rowspan='1' colspan='2'>
      <br><br>
      </td>
</tr>

</form>

<form method = "POST" name = "formu_insercion2" action =" articulo.php"
onsubmit =  "return validar(this)">

<tr>
      <td align = "center" colspan = '2'>Ejemplo : 326598</td>
</tr>

<tr>
      <td align = "right">Codigo</td>
      <td align = "left">  *<input type='text' name='codart'
size='25'></td>
</tr>

<tr>
      <td align = "center" colspan =
'2'>       
                                                Ejemplo : PAPEL
BOND</td>
```

```
</tr>

<tr>
      <td align = "right">Nombre</td>
      <td align = "left">  *<input type='text' name='nombreart'
size='25'></td>
</tr>

<tr>
      <td align = "center" colspan = '2'><br></td>
</tr>

<?php
$unidad =
array("Arrobas","Block","Bolsas","Cajas","Centimetros","Cubetas","Docena"
,"Galon","Libras","Litro",

"Metro","Paquete","Pliego","Pulgadas","Quintal","Resmas","Rollo","Unidad"
,"Yardas");
?>

<tr>
      <td align = "right">Unidad Medida</td>
      <td align = "left">  *<select name = "umedida">

<?php
      echo" <option selected>Elija Unidad de Medida</option>";
      for($i=0;$i<19;$i++)
            echo" <option> $unidad[$i]</option>"
?>
   </td>
</tr>

<tr>
      <td rowspan='1' colspan='2'>
      <br><br>
      </td>
</tr>

<tr>
      <td align="center" rowspan='1' colspan='2'>
      <button type = 'submit'  size = "15" name = 'guardar' value =
'Guardar Registro'>Guardar Registro</button>   
      <input type = hidden name = 'pagina' value = '2'/>
      <button type = 'reset'  size = "15" name = 'Limpiar' value =
'Limpiar'>Limpiar</button>
      </td>
</tr>

<tr>
      <td rowspan='1' colspan='2'>
      <br>
      </td>
</tr>

</form>
```

```
<form method="POST" name = "formu_insercion3" action="articulo.php">
<tr>
<table width = "65%" bordercolor = red cellpadding = "1" cellspacing =
"1" aling = "center"  border = "2">
      <td align="center" rowspan='1' colspan='2'>
      <h3><p align =  "center">Borrar Articulo</p></h3>
      <p align =  "center">Elija El Articulo Que Desea Borrar Y Presione
Borrar Registro</p>
      </td>
</tr>

<tr>
      <td align = "center" rowspan = '1' colspan = '2'>
<?php
      $sentsql= "select * from articulo order by codart";
      $ResultadoBusqueda = pg_query($conexion,$sentsql) or die("Error en
la consulta: $sentsql" . pg_last_error($conexion));

      echo"     ";
      if($ResultadoBusqueda)
      {
            echo "<select name = 'nombrearticulo'>";
            echo "<option value = '-1' selected> Elija Articulo A
Borrar</option>";
            while($arr=pg_fetch_array($ResultadoBusqueda))
            {
              echo "<option>$arr[nombreart]</option>";
            }
            echo "</select> ";
      }
?>
               
      <button type = 'submit'  size = "10" name = 'borrar' value =
'Borrar Registro'>Borrar Registro</button>
    <input type = hidden name = 'pagina' value = '3'/>
      </td>
</table>
</tr>

<tr>
      <td rowspan='1' colspan='2'>
      <br><br>
      </td>
</tr>

</form>
</table>
</center>

<p align = "center">
      <a href="principal.php"><img src = "./lg/p2.gif"></a>
</p>

<p align = "center">
      <a href="entrada.php"><img src = "./lg/e2.gif"></a>
```

```
</p>

<p align = "center">
      <a href="buscar.php"><img src = "./lg/b2.gif"></a>
</p>

<?php
function buscar()
{
      global $conexion;
      global $b_ultimo, $ultimo;

      if(empty($_POST[ultimo]))
      {
            echo ("     <script language='javascript'>
                    alert('Introdusca un valor...');
                    parent.location='articulo.php';
                  </script>");
            exit (-1);
      }
      else
      {
            $consulta_b = pg_query($conexion,"SELECT * from articulo
where codart like '$ultimo%' order by codart");

            $numfilas_b = pg_num_rows($consulta_b);
            if ($numfilas_b == 0)
            {
                  echo ("     <script language='javascript'>
                          alert('No existen codigos...');
                          parent.location='articulo.php';
                    </script>");
                exit (-1);
          }
          else
          {
                  $numfilas_b = pg_num_rows($consulta_b);
                  $i = $numfilas_b - 1;
            $fila_b = pg_fetch_array($consulta_b,$i);

            echo ("     <script language='javascript'>
                          alert('El ultimo Codigo introducido fue:
$fila_b[codart]   ');
                          parent.location='articulo.php';
                    </script>");
                exit (-1);
            }
      }
}

function insertar()
{
      global $conexion;
      global $nombreart, $codart, $umedida, $nue_nombre;
```

```
      if(empty($_POST[codart]) || empty($_POST[nombreart]) ||
empty($_POST[umedida]))
      {
            return false;
      }
      else
      {
            echo $nue_nombre = strtolower($_POST[nombreart]);
            $consulta = pg_exec($conexion,"insert into articulo
values('$_POST[codart]','$nue_nombre','$_POST[umedida]','0','0')");
            if (!$consulta)
          {
               echo ("Hubo algun problema !!!");
              exit;
            }
            else
            {
                  echo ("     <script language='javascript'>
                         alert('Registro guardado');
                         parent.location = 'articulo.php';
                         </script>");
            }
      }
           $codart = "";
           $nombreart =  "";
           $umedida = "";
}

function borrar()
{
      global $conexion;
      global $nombrearticulo;

      $articulo = $_POST[nombrearticulo];
      $con1= "select codart from articulo where nombreart = '$articulo'";
      $consult1=pg_query($conexion,$con1);
      $arreglo1=pg_fetch_array($consult1);
      $codigoarticulo=$arreglo1[codart];

      $bor="DELETE from articulo where codart='$codigoarticulo'";
      $consulta = pg_query($conexion,$bor);

      if($consulta)
      {
            echo ("<script language='javascript'>
                alert('Registro Borrado');
                    parent.location = 'articulo.php';
                </script>");
      }
      else
      {
          echo ("<script language='javascript'>
                alert('No Se Puede Borrar');
                </script>");
      }
}
```

```
switch($pagina)
{
      case 0:
           break;
      case 1:
             buscar();
           break;
      case 2:
             insertar();
           break;
      case 3:
             borrar();
           break;
}
?>

</body>
</html>

<?php
}
?>
```

//Entrada

```
<?php
include_once("verificar.php");

session_start();
if($_SESSION['n_usuario'] != "operador")
{
      echo "<script type 'text/javascript'>
              alert ('Acceso Denegado');
                  history.back();
              </script> ";
            exit;
}
else
{
      $conexion = Conexion();
      if($conexion == false)
      {
            echo "No pudo abrirse la conexion a $database";
            exit (-1);
      }

function insertar()
{
      global $conexion, $guardar, $QueNota;
      global $dia, $cantidad_ingre,
$costounit,$procedencia,$observaciones,$codart,$nombrearticulo;
      global $pagina, $Ultima_transaccion, $Ultima_Nota,
$Nombre_Btn,$valor_total;
```

```
      if($Nombre_Btn == "Guardar Registro con Nota Nueva")
      {
            $Ultima_Nota = $Ultima_Nota + 1;
      }

      if(empty($_POST[cantidad_ingre])||empty($_POST[costounit])||empty($
_POST[procedencia]))
      {
            if($_POST[costounit] == 0)
                  echo ("     <script language='javascript'>
                           alert('Verifique datos');
                           parent.location = 'entrada.php';
                    </script>");
      }
      else
      {
            $conexion = Conexion();
            if($conexion == false)
            {
                  echo "No pudo abrirse la conexion a $database";
                  exit (-1);
            }

            $articulo = $_POST[nombrearticulo];
            $con1= "select * from articulo where nombreart =
'$articulo'";
            $consult1=pg_query($conexion,$con1);
            $arreglo1=pg_fetch_array($consult1);
            $codigoarticulo=$arreglo1[codart];
            $nota_act = 1;
            $cantidad_ant=$arreglo1[cantidad];
            $cantidad_act= $cantidad_ant + $_POST[cantidad_ingre];
            $valor_total= $_POST[cantidad_ingre]* $_POST[costounit];

            echo $Ultima_transaccion;echo"<br>";
            echo $dia;echo"<br>";
            echo $Ultima_Nota;echo"<br>";
            echo $_POST[procedencia];echo"<br>";
            echo $codigoarticulo;echo"<br>";
            echo $_POST[cantidad_ingre];echo"<br>";
            echo $_POST[costounit];echo"<br>";
            echo $valor_total;echo"<br>";
            echo $_POST[observaciones];echo"<br>";

            $sentencia = "insert into entrada
values('$Ultima_transaccion','$dia','$Ultima_Nota','$_POST[procedencia]',
'$codigoarticulo','$_POST[cantidad_ingre]','$_POST[costounit]','$valor_to
tal','$_POST[observaciones]')";
            $consulta = pg_query($conexion,$sentencia) or die ("Error en
la consulta : $sentencia");
            $aux0 = pg_affected_rows($consulta);
            $consulta1 = pg_exec($conexion,"update articulo set
cantidad='$cantidad_act', valor_unit ='$_POST[costounit]'  where codart =
'$codigoarticulo'");
```

```
            if (!$aux0)
            {
                  echo ("Hubo Algun Problema !!!");
                  exit;
            }
            else
            {
                  echo ("<script language='javascript'>
                          alert('Entrada Guardada ');
                          parent.location='entrada.php';
                    </script>");
            }
      }
}
?>
<html>
<head>
<title> Generar Las Entradas De Datos A La Bodega</title>
<link rel="stylesheet" type="text/css" href="./estilo/estilo2_1.css">
<style type="text/css">object{visibility:hidden;}</style>
<style type="text/css">object{visibility:visible;}</style>

<script language = "javascript" type = "text/javascript">
function validar(fr)
{
            if(fr.cantidad_ingre.value.search(/^\d{1,3}$/) == -1)
            {
                  alert("Error. Cantidad Esta Entre 1 y 999");
                  fr.cantidad_ingre.focus();
                  return false;
            }

            if(fr.costounit.value.search(/^\d{1,3}.\d{1,2}$/) == -1)
            {
                  alert("Error. Valor Unitario Debe Contener Solo
Numeros");
                  fr.costounit.focus();
                  return false;
            }

            if(fr.procedencia.value.search(/(\D+\s\D+)|(\D+)$/) == -1)
            {
                  alert("Error. Procedencia Debe Contener Solo Letras");
                  fr.procedencia.focus();
                  return false;
            }
      if(confirm("¿Desea Guardar el Registro?") == true)
      {
            return true;
      }
      else
      {
            return false;
      }
}
</script>
```

```
</head>

<body leftmargin='0' topmargin='0' background =
"./imagenes/nubecilla.gif">

<table width="100%" align="center" cellpadding='0' cellspacing='0'>
<tr>
      <td height="126"  bgcolor="#000033">
      <img src="imagenes/logo.jpg" width="105%" height="120"
align="absbottom">
      </td>
</tr>
</table>

<h3><p align =  "center">Entrada De Datos A La Bodega</p></h3>
<p align =  "center"> Los Campos con * Son Obligatorios </p>

<?php
$dia= date("d-m-Y");
echo "        Fecha Actual : $dia";
?>

<center>
<table width="50%" cellpadding="1" cellspacing="1" align="center"
border="0">
<form method="POST" action="entrada.php" onsubmit = "return
validar(this)">

<?php
$Ultima_transaccion = 0;
$Sql1= "select cod_trans from entrada order by cod_trans";
$ResultadoSql1 = pg_query($conexion,$Sql1);
$FilasEntrada = pg_num_rows($ResultadoSql1);

if($FilasEntrada !=0)
{
    $FilasEntrada = $FilasEntrada - 1;
    $DatoCod = pg_fetch_array($ResultadoSql1, $FilasEntrada);
    $Ultima_transaccion = $DatoCod[cod_trans] + 1;
}

echo"
<tr>
      <td align = 'right'>      Codigo De Transaccion
      </td
      <td align = 'left'>      $Ultima_transaccion
      <input type='hidden' name='Transaccion'
value='$Ultima_transaccion'/>
      </td>
</tr>";
?>

<tr>
      <td align = "right">Nombre Del Articulo
      </td>
      <td align = "left">
```

```
<?php
      $sentsql= "select * from articulo order by codart";
      $ResultadoBusqueda = pg_query($conexion,$sentsql) or die("Error en
la consulta: $sentsql" . pg_last_error($conexion));

      echo"     ";
      if($ResultadoBusqueda)
      {
            echo "<select name = 'nombrearticulo'>";
            while($arr=pg_fetch_array($ResultadoBusqueda))
            {
              echo "<option>$arr[nombreart]</option>";
            }
            echo "</select> ";
    }
?>
</td>
</tr>

<tr>
      <td align = "center" colspan =
'2'>           &nb
sp;    

              
                                                            Ejemplo :
100<br></td>
</tr>

<tr>
      <td align = "right">Cantidad</td>
      <td align = "left">  * <input type='text' name='cantidad_ingre'
size='25'></td>
</tr>

<?php
$Ultima_Nota = 0;
$Sql2= "select nota_entrada from entrada order by nota_entrada";
$ResultadoSql2 = pg_query($conexion,$Sql2);
$FilasEntradaNota = pg_num_rows($ResultadoSql2);

if($FilasEntradaNota !=0)
{
      $FilasEntradaNota = $FilasEntradaNota - 1;
      $DatoNota = pg_fetch_array($ResultadoSql2, $FilasEntradaNota);
      $Ultima_Nota = $DatoNota[nota_entrada];
}
else
      $Ultima_Nota = 1;

    echo"
      <tr align = 'center'>
      <td align = 'right'>Nota Actual<input type='hidden' name='Nota'
value='$Ultima_Nota'/>
      </td>
```

```
            <td align = 'left'>     
$Ultima_Nota</td>
    </tr>";
?>

<tr>
      <td align = "center" colspan =
'2'>           &nb
sp;  

            &
nbsp; 
                                                        Ejemplo :
12.30<br></td>
</tr>

<tr align = "center"">
      <td align = "right">Valor Unitario</td>
      <td align = "left">  * <input type='text' name='costounit'
size='25'></td>
</tr>
<tr>
      <td align = "center" colspan =
'2'>           &nb
sp;           &nbs
p;            
;            
   
Ejemplo : Aportes Del Estado<br></td>
</tr>
<tr align = "center">
      <td align = "right">Procedencia</td>
      <td  align = "left">  * <input type='text' name='procedencia'
size='25'></td>
</tr>

<tr>
      <td align = "center" colspan = '2'><br></td>
</tr>
<tr align = "center">
      <td align = "right">Observaciones</td>
      <td align = "left">     <input type='text'
name='observaciones' size='25' maxlenght = '80'></td>
</tr>

<tr align = "center">
      <td rowspan='1' colspan='2'>
      <br>
      </td>
</tr>

<tr align = "center">
      <td rowspan='1' colspan='2'>
      <button type = 'submit'  size = "10" name = 'Nombre_Btn' value =
'Guardar Registro con Nota Actual'>Guardar Registro con Nota
Actual</button>
```

```
      <input type=hidden name='pagina' value='1'>
      </td>
</tr>

<tr align = "center">
      <td rowspan='1' colspan='2'>
      <br>
      <button type = 'submit'  size = "10" name = 'Nombre_Btn' value =
'Guardar Registro con Nota Nueva'>Guardar Registro con Nota
Nueva</button>
      <input type=hidden name='pagina' value='2'>
      </td>
</tr>

</form>

<tr align = "center">
      <td rowspan='1' colspan='2'>
      <br>
      <button type = 'reset'  size = "10" name = 'Limpiar' value =
'Limpiar'>Limpiar</button>
      </td>
</tr>

</table>
</center>

<?php
switch($pagina)
{
    case 0:
         break;
    case 1:
         insertar();
         break;
    case 2:
         insertar();
         break;
}
?>
<br><br>
<p align = "center">
      <a href="principal.php"><img src = "./lg/p2.gif"></a>
</p>

<p align = "center">
      <a href="articulo.php"><img src = "./lg/a2.gif"></a>
</p>

</body>
</html>

<?php
}
```

```
?>
```

//Salida

```
<?php
include_once("verificar.php");

session_start();
if($_SESSION['n_usuario'] != "operador")
{
      echo "<script type 'text/javascript'>
              alert ('Acceso Denegado');
                  history.back();
              </script> ";
            exit;
}
else
{
      $conexion = Conexion();
      if($conexion == false)
      {
            echo "No pudo abrirse la conexion a $database";
            exit (-1);
      }
?>
<html>
<head>
    <title> Generar Salida de Datos </title>
      <link rel="stylesheet" type="text/css"
href="./estilo/estilo2_1.css">
      <style type="text/css">object{visibility:hidden;}</style>
      <style type="text/css">object{visibility:visible;}</style>
</head>

<body leftmargin='0' topmargin='0' background =
"./imagenes/nubecilla.gif">
<table width="100%" align=center cellpadding='0' cellspacing='0'>
<tr>
      <td height="125"  bgcolor="#000033">
      <img src="imagenes/logo.jpg" width="100%" height="120"
align="absbottom">
      </td>
</tr>
</table>

<h3><p align =  "center"> Buscar Numero De Solicitud </h3>

<?php
$dia= date("d-m-Y");
echo "        Fecha Actual : $dia";
?>

<form method = "POST"  action = "sa1.php">
<p align = "center">
```

```
<table width = "60%" cellpadding = "1" cellspacing = "1" aling = "center"
border = "0" class = "border">
<tr>
      <td align = "center" rowspan = '1' colspan = '2'>

<?php
      $conexion = Conexion();
      if($conexion == false)
      {
            echo "No pudo abrirse la conexion a $database";
            exit (-1);
      }

      echo"<center>";
      $sentsql = "select numero from solicitud group by numero";
      $ResultadoBusqueda = pg_query($conexion,$sentsql) or die("Error en
la consulta: $sentsql" . pg_last_error($conexion));

      echo"     ";
      if($ResultadoBusqueda)
      {
              echo "<select name = 'combo_numero'>";
              echo"<option value='-1' selected>Selecione Un Numero De
Solicitud</option>";

              while($areglo_numero = pg_fetch_array($ResultadoBusqueda))
              {
                        echo "<option>$areglo_numero[numero]</option>";

              }
              echo "</select> ";
      }
      echo"</center>";
?>

<tr>
      <td rowspan='1' colspan='3'> <br> </td>
</tr>

<tr>
      <td colspan = '2' align = 'center'>
      <button type = 'submit' align = 'center' size = "10" name =
'buscar' value = 'Buscar Solicitud'>Buscar Solicitud</button>
</tr>

<tr
      <td>
      <br>
      <p align = "center">
      <a href="principal.php"><img src = "./lg/p2.gif"></a>
      </p>
      </td>
</tr>

</tbody>
</table>
```

```
</p>
</body>
</form>
</html>

<?php
}
?>
```

//Buscar

```
<?php
include_once("verificar.php");

session_start();
if($_SESSION['n_usuario'] != "operador" && $_SESSION['n_usuario'] !=
"usuario")
{
      echo "<script type 'text/javascript'>
              alert ('Acceso Denegado');
                  history.back();
              </script> ";
            exit;
}
else
{
      $conexion = Conexion();
      if($conexion == false)
      {
            echo "No pudo abrirse la conexion a $database";
            exit (-1);
      }
      global $nombre,$codigo;

      function buscar()
      {
            global $datos, $conexion;

            if(empty($_POST[datos]))
              {
                  return false;
              }
         else
                   {
                      $conv=strtolower($_POST[datos]);
                      $consulta1 = pg_query($conexion,"SELECT * from
articulo where codart like '$conv%' or nombreart like '$conv%'");
                      $numfilas = pg_num_rows($consulta1);
                       if ($numfilas == 0)
                              {
                                    echo ("     <script
language='javascript'>
                                                alert('No hay
coincidencias');
```

```
      parent.location='buscar.php';
                                          </script>");
                                    exit (-1);
                              }

                              else
                                    {
                                          echo"<h2><p align =
'center'>Datos existentes</p></h2>
                                          <table bordercolor = red border
= 4 align = center>
                                                <tr>
                                                      <td> <h4>
    Codigo Articulo     </h4> </td>
                                                      <td> <h4>
    Nombre Articulo     </h4> </td>
                                                      <td> <h4>
    Cantidad     </h4> </td>
                                                      <td> <h4>
    Valor Unitario     </h4> </td>
                                                </tr>";
                        for ($i=0;$i<$numfilas;$i++)
                              {
                              $fila=pg_fetch_array($consulta1,$i);
                               echo"<tr>
                                          <td align = 'center'>
$fila[codart]</td>
                                    <td align = 'center'>
$fila[nombreart]</td>
                                    <td align = 'right' > $fila[cantidad]
        </td>
                                    <td align = 'right' >
$fila[valor_unit]         </td></tr>";
                              }
                        echo"</table>";
                  }
               }
      }
      function mostrar()
      {
            global $conexion;
        if ($buscar != "Buscar Registro")
              {
                $consulta1 = pg_query($conexion,"SELECT * from articulo
order by codart");
                  $numfilas = pg_num_rows($consulta1);
                  if ($numfilas != 0)
                    {
                        echo"<h2><p align = 'center'>Datos
existentes</p></h2>
                              <table bordercolor = red border = 4 align =
center>
                                    <tr>
                                          <td> <h4>     Codigo
Articulo     </h4> </td>
```

```
                                        <td> <h4>     Nombre
Articulo     </h4> </td>
                                        <td> <h4>    
Cantidad     </h4> </td>
                                        <td> <h4>     Valor
Unitario     </h4> </td>
                                    </tr>";
                          for ($i=0;$i<$numfilas;$i++)
                                {
                                $fila=pg_fetch_array($consulta1,$i);
                                echo"<tr>
                                        <td align = 'center'>
$fila[codart]</td>
                                    <td align = 'center'>
$fila[nombreart]</td>
                                    <td align = 'right' > $fila[cantidad]
        </td>
                                    <td align = 'right' >
$fila[valor_unit]         </td></tr>";
                                }
                          echo"</table>";
                      }
                    }
      }
 ?>
      <html>
        <head>
          <title> Buscar Un Articulo En La Base De Datos De
Bodega</title>
                    <link rel = "stylesheet" type = "text/css" href =
"./estilo/estilo2.css">
                    <style type = "text/css">
object{visibility:hidden;}</style>
                    <style type = "text/css">
object{visibility:visible;}</style>

            </head>

            <body leftmargin = '0' topmargin = '0' background =
"./imagenes/nubecilla.gif">

            <table width = "100%" align = center cellpadding = '0'
cellspacing = '0'>
                    <tr>
                    <td height = "125"  bgcolor = "#000033"> <img src =
"imagenes/logo.jpg" width = "100%" height = "120" align = "absbottom">
                    </td>
            </tr>
          </table>

              <h3><p align =  "center"> Busqueda De Datos En La
Bodega</h3>
              <h4><p align =  "center"> Los Campos con * Son Obligatorios
</p></h4>
            <form method = "POST"  action = "buscar.php ">
                    <p align = "center">
```

```
                    <table width = "60%" cellpadding = "1" cellspacing =
"1" aling = "center"  border = "0" class = "border">
                  <tr
                              <td rowspan = '1' colspan = '2' align =
"center"> <h4> Introduzca Nombre O Codigo</h4>
                              </td>

                        </tr>

                        <tr align = "center">
                        <td rowspan = '1' colspan = '2' align = "center" >
              
              
              
              

Ejemplo :391018 / fastener<br> </td>
                </tr>

                        <tr

                              <td align = "right"> <h4>
              
                                     Codigo /
Nombre   </h4> </td>

                              <td align = "left">    * <input type
= 'text' name = 'datos' id = 'datos' size = '25'></td>
                        </tr>

                        <tr align = "center">
                        <td rowspan = '1' colspan = '2'> <br> </td>
                </tr>
                        <tr
                              <td rowspan = '1' colspan = '2' align =
"center"
                              <button type = 'submit'  size = "10" name =
'buscar' value = 'Buscar Registro'>Buscar Registro</button>
                              <!--input type = 'submit' name = 'buscar'
value = 'Buscar Registro'-->
                              </td>
                        </tr>
                  </tbody>
            </table>
        </p>

<?php
 buscar();
 mostrar();
 ?>
</form>
<br><br>
<p align = "center">
<a href="principal.php"><img src = "./lg/p2.gif"></a>
</p>
```

```
<p align = "center">
<a href="articulo.php"><img src = "./lg/a2.gif"></a>
</p>

</body>
</html>
<?php
}
 ?>
```

//Dependencia

```
<?php
include_once("verificar.php");

session_start();
if($_SESSION['n_usuario'] != "operador")
{
      echo "<script type 'text/javascript'>
              alert ('Acceso Denegado');
                  history.back();
              </script> ";
            exit;
}
else
{
      $conexion = Conexion();
      if($conexion == false)
      {
            echo "No pudo abrirse la conexion a $database";
            exit (-1);
      }
function insertar()
{
      global $conexion;
      global $nombre_dependencia, $codigo_dependencia,$monto;

      if(empty($_POST[codigo_dependencia]) ||
empty($_POST[nombre_dependencia])|| empty($_POST[monto]))
      {
            return false;
      }
      else
      {
             $consulta = pg_query($conexion,"insert into dependencias
values('$_POST[codigo_dependencia]','$_POST[nombre_dependencia]','$_POST[
monto]')");

          if (!$consulta)
          {
                  echo ("Hubo algun problema !!!");
                  exit (-1);
          }
            else
```

```
            {
                        echo ("     <script language='javascript'>
                                alert('Dependencia guardada....');
                                parent.location='dependencia.php';
                          </script>");
           }
        }
 }

function mostrar()
{
      global $conexion;

      if ($conexion == false)
      {
      echo "no pudo abrirse la conexion a $database";
      exit (-1);
      }
      $resultado = pg_query($conexion, "SELECT * from dependencias order
by codigo_dependencia asc");
      $numfilas=pg_num_rows($resultado);

      if (!$numfilas)
      {
      echo "<h2>Aún no hay Dependencias en la base de datos</h2>";
      }
      else
      {
           echo "<h3>
             <p align = 'center'>Dependencia  Existentes</p>
             </h3>
           <table bordercolor = red border = 1 align = center>
                           <tr><th>    Codigo Dependencia
   </th><th>     Nombre Dependencia
    </th><th>     Monto    
</th></tr>";
                           for ($i=0;$i<$numfilas;$i++)
                           {
                               $fila=pg_fetch_array($resultado,$i);
                      echo "<tr>
                                                  <td align='left'>
            
$fila[codigo_dependencia] </td>
                                                  <td align='left'>
       $fila[nombre_dependencia] </td>
                                                      <td align='left'>
          $fila[monto] </td>
                                  </tr>";
                           }
                 echo "</table>";
      }
  }

?>
<html>
<head>
```

```
<title> Crear Las Dependencias En La Base De Datos De Bodega </title>

<script language = "javascript" type = "text/javascript">
function validar(fr)
{
            if(fr.codigo_dependencia.value.search(/^\D\d{2}$/) == -1)
            {
                  alert("Error. Codigo Dependencia Debe Ser Igual a
C04");
                  fr.codigo_dependencia.focus();
                  return false;
            }

            if(fr.nombre_dependencia.value.search(/(\D+\s\D+)|(\D+)$/) ==
-1)
            {
                  alert("Error. Nombre Dependencia Debe Contener Solo
Letras");
                  fr.nombre_dependencia.focus();
                  return false;
            }

            if(fr.monto.value.search(/^\d{1,5}$/) == -1)
            {
                  alert("Error. Monto Debe Contener Solo Numeros");
                  fr.monto.focus();
                  return false;
            }

      if(confirm("¿Desea Guardar el Registro?") == true)
      {
            return true;
      }
      else
      {
            return false;
      }
}
</script>
        </head>

</style>
<link rel="stylesheet" type="text/css" href="./estilo/estilo2.css">
<style type="text/css">object{visibility:hidden;}</style>
<style type="text/css">object{visibility:visible;}</style>

<body leftmargin='0' topmargin='0' background =
"./imagenes/nubecilla.gif">

<table width="100%" align=center cellpadding='0' cellspacing='0'>
<tr>
      <td height="126"  bgcolor="#000033">
      <img src="imagenes/logo.jpg" width="100%" height="120"
align="absbottom">
      </td>
      </tr>
```

```
</table>

            <h3>
        <p align =  "center">Añadir Nuevas Dependencias</p>
        </h3>
        <p align =  "center"> Los Campos Con * Son Obligatorios </p>


<form method = POST name = formu_insercion action = dependencia.php
onsubmit = "return validar(this)">
<p align = "center">
<table width = "60%" cellpadding = "1" cellspacing = "1" aling = "center"
border = "0">
<tbody>
<tr>
      <td align = "center" colspan =
'2'>           

                 
                                                  Ejemplo : A01 </td>
</tr>
<tr>
            <td align = "right">Codigo
Dependencia     </td>
            <td align = "left">   * <input type='text'
name='codigo_dependencia' size='15'>
</td></tr>
<tr>
      <td align = "center" colspan =
'2'>      

                
                
             Ejemplo : RECTORIA</td>
</tr>

<tr>
            <td align = "right">Nombre
Dependencia     </td>
            <td align = "left">   * <input type='text'
name='nombre_dependencia' size='25'>
            </td></tr>
<tr>
      <td align = "center" colspan =
'2'>           &nb
sp;          
  Ejemplo : 15000</td>
</tr>

       <tr>
            <td align = "right">Monto
Disponible     </td>
            <td align = "left">   * <input type='text'
name='monto' size='25'>
</tr></tr>
```

```
<tr>
    <td rowspan='1' colspan='2'>
      <br><br>
    </td>
</tr>

      <td align = "right">
      <button type = 'submit'  size = "10" name = 'guardar' value =
'Guardar Registro'>Guardar Registro</button>

    <td align = "left">    
    <button type = 'reset'  size = "10" name = 'Limpiar' value =
'Limpiar'>Limpiar</button>

      </tbody>
      </table>
      </p>
      <br><br
 <?php
      insertar();
      mostrar();
?>

</form>
<p align = "center">
      <a href="principal.php"><img src = "./lg/p2.gif"></a>
</p>

<p align = "center">
      <a href="salida.php"><img src = "./lg/s2.gif"></a>
</p>

</body>
</html>

 <?php
}
?>
```

//Reporte por dependencia

```
<?php

include_once("verificar.php");

session_start();
if($_SESSION['n_usuario'] != "operador")

{
      echo "<script type 'text/javascript'>
              alert ('Acceso Denegado');
                  history.back();
              </script> ";
            exit;
```

```
}
else
{
      $conexion = Conexion();
      if($conexion == false)
      {
            echo "No pudo abrirse la conexion a $database";
            exit (-1);
      }
?>
<html>
<head>
    <title> Reporte Por Dependencias De Articulos De La Bodega </title>
      <link rel = "stylesheet" type = "text/css" href =
"./estilo/estilo2.css">
      <style type = "text/css"> object{visibility:hidden;}</style>
      <style type = "text/css"> object{visibility:visible;}</style>

<script language = "javascript" type = "text/javascript">
function validar(fr)
{
            if(fr.datos1.value.search(/^([012][1-9]|[123][01])(-)(0[1-
9]|1[012])\2(\d{4})$/) == -1)
            {
                  alert("Error. La Fecha de Inicio de Mes debe ser
compatible con el formato 01-10-2007");
                  fr.datos1.focus();
                  return false;
            }

            if(fr.datos2.value.search(/^([012][1-9]|[123][01])(-)(0[1-
9]|1[012])\2(\d{4})$/) == -1)
            {
                  alert("Error. La Fecha de Fin de Mes debe ser
compatible con el formato 01-10-2007");
                  fr.datos2.focus();
                  return false;
            }
}
</script>
</head>

<body leftmargin = '0' topmargin = '0' background =
"./imagenes/nubecilla.gif">
      <table width = "100%" align = center cellpadding = '0' cellspacing
= '0'>
            <tr>
            <td height = "125"  bgcolor = "#000033"> <img src =
"imagenes/logo.jpg" width = "100%" height = "120" align =
"absbottom"></td>
          </tr>
    </table>

      <h3> <p align = "center"> Generar Reporte Por Costo De
Dependencias</p> </h3>
```

```
      <h4><p align =  "center"> Los Campos con * Son Obligatorios
</p></h4>
      <form method = "POST"  action = "reporte_depen1.php" onsubmit =
"return validar(this)">
      <p align = "center">

      <table width = "90%" cellpadding = "1" cellspacing = "1" aling =
"center"  border = "0" class = "border">
                              <tr align = "center">
                  <td rowspan='1' colspan='2'>
                        <br>
                  </td>
                </tr>
            <tr>
                        <td align = 'right'> <h4>
         Debe Introducir Dia, Mes y Año
    
                        </h4> </td> <td> <br>     Ej.= 18-
10-2007              

                   &n
bsp         Ej.= 02-05-2007</td>
                  </tr>

            <tr>
                        <td align = 'right'> <h4>
      Fecha Inicial:      </h4>
</td><td align = 'left'> <h4>   *
                              <input type = 'text' name = 'datos1' id =
'datos1' size = '15'>    Fecha Final:   *
                              <input type = 'text' name = 'datos2' id =
'datos2' size = '15'> </h4>
                        </td>
                  </tr>

                  <tr>
                              <td align = "right"><h4><br>Nombre
Dependencia     

                              <?php
                                    $sentsql= "select * from dependencias
order by codigo_dependencia";
                                    $ResultadoBusqueda =
pg_query($conexion,$sentsql) or die("Error en la consulta: $sentsql" .
pg_last_error($conexion));

echo"<td>     ";
                                    echo"<br>    * ";
                                     if($ResultadoBusqueda)
                                          {
                                          echo "<select name =
'nombre_dependencia'>";

      while($arr=pg_fetch_array($ResultadoBusqueda))
                                                {
```

```
                                                                    echo "<option
selected>$arr[nombre_dependencia]</option>";
                                                                }
                                                        echo "</select>";
                                                }

                                                ?>
                                                </h4>
                                                </td>
                                                </tr>

                        <tr>
                                <td rowspan='1' colspan='3'> <br> </td>
                    </tr>
                                <tr>
                                <td rowspan = '1' colspan = '2' align = 'center'>
                                <button type = 'submit'  size = "10" name =
'buscar' value = 'Buscar Entrada'>Generar Reporte</button>
                                <!--input type = 'submit' name = 'buscar' value =
'Buscar Entrada'--></td>
                                </tr>
                    </tbody>
            </table>
            </p>

</form>
<br>
<br>

<p align = "center">
      <a href = "principal.php"> <img src = "./lg/p2.gif"> </a>
</p>

</body>
</html>

<?php
}
?>
```

//Funcion Para Mostrar el Reporte

```
<?php

include_once("verificar.php");

$conexion = Conexion();
if(!$conexion)          //verificar si se establecio la conexion
{
      echo "No Se Conecto con La Base De Datos Bodega/Pongase En Contacto
Con El Administrador".pg_last_error();
}

function mensaje($mensaje)
{
```

```
      echo"
                        <script language = 'javascript'>
                        window.alert('$mensaje');
                        </script>";
}

$temp = $_POST[datos1];
$temp1 = $_POST[datos2];

            $dependencia = $_POST[nombre_dependencia];
            $con1 = "select * from dependencias where nombre_dependencia
= '$dependencia'";
            $consult1 = pg_query($conexion,$con1);
            $arreglo1 = pg_fetch_array($consult1);
            $codigo_dependencia = $arreglo1[codigo_dependencia];

                        $datos = pg_exec($conexion,"select
fecha,requisa,codart,articulo.nombreart,cantidad,articulo.umedida,
                                    valor_unit,valor_total,
salida.observaciones from salida
                                    where salida.codart = articulo.codart
and salida.codigo_dependencia = '$codigo_dependencia' and
                                    fecha >= '$temp' and fecha <= '$temp1'
order by codart"); //seleccion de registros de la base de datos
                        $filas = pg_NumRows($datos); //numero de filas

                        if(!$filas)
                        {
                              echo "<script type 'text/javascript'>
                                          alert ('No hay datos en el
rango de Fechas definido...');
                                          history.back();
                                    </script> ";
                              exit;
                        }

?>
<html>
<head>
<title>Gastos Por Dependencia

<style type="text/css">

          BODY {background: #ffffff ; color: black;}
            td.info {background: #33FFFF; font-family: arial; font-size:
10px; color: blue;}
            td.info1 {background: #ffffff; font-family: arial; font-size:
10px; color: blue;}
            table.border {background:#D8DCFF; color: blue;}

            td.subcat {background:#ffffff; color:black; font-family:
"arial"; font-size: 10px;}

</style>
</head>
```

```
<body>
<tr>
<td>
<?php
      $dia_repor = date("d-m-Y");
?>
<table width="150%" align="center" cellspacing='1' border='0'>
<tr>

<td align  = 'left'>
<h6>
<A onclick = 'window.print();return false' href =
'solicitud.php'>Imprimir</A>
</h6>
</td>

<td align ='right'>
<h6>
<?php
      echo $dia_repor;
?>
</h6>
</td>
</tr>

<tr>
<td align ='center'>
<h6>UNIVERSIDAD NACIONAL AUT&Oacute;NOMA DE NICARAGUA,
LE&Oacute;N<br><br>
UNAN - LE&Oacute;N<br><br>
GASTOS POR DEPENDENCIA<br>
DESDE:
<?php
      echo $temp;
?>
            &
nbsp;   
HASTA:
<?php
      echo $temp1;
?>
<br>
</h6>
</td>
</tr>
<tr>
<td align ='left'>
<h6>
Nombre Dependencia :
<?php
      echo $dependencia;
 ?>
 </h6>
</td>
</tr>
```

```
<tr>
<table border="0" width="80%" cellspacing="1" cellpadding="0" align =
"center">
<tr>
      <td class='info'><blockquote>Fecha</td>
      <td class='info'>   Requisa</td>
      <td class='info'>   Codigo</td>
      <td class='info'><blockquote>Articulos</td>
      <td class='info'>   Cantidad</td>
      <td class='info'><blockquote>Unidad_Medida</td>
      <td class='info'>   Valor_Unitario</td>
      <td class='info'>   Valor_Total</td>
      <td class='info'><blockquote>Observaciones</td>
</tr>

<tr>
      <td class="subcat" align='center'>
<?php
      $aux3 = 0;
      for($f=0; $f<$filas;$f++)    //copiar datos en el archivo
      {
       echo pg_result($datos,$f,0);
?>
      </font>
      <td class="subcat" align='center'>
<?php
      echo pg_result($datos,$f,1);

?>
      </td>
      <td class="subcat" align='center'>
<?php
      echo pg_result($datos,$f,2);

?>
      </td>
      <td class="subcat" align='center' >
<?php
      echo pg_result($datos,$f,3);

?>
      </td>
      <td class="subcat" align='center'>
<?php
      echo pg_result($datos,$f,4);

?>
      </td>
      <td class="subcat" align='center'>
<?php
      echo pg_result($datos,$f,5);

?>
      </td>
      <td class="subcat" align="center" >
<?php
```

```
      echo pg_result($datos,$f,6);

?>
      </td>
      <td class="subcat" align='center'>

<?php
      echo pg_result($datos,$f,7);
      $aux3 += pg_result($datos,$f,7);

?>
      </td>
      <td class="subcat" align='center'>

<?php
      echo pg_result($datos,$f,8);

?>
      </td>

<?php
      if($f == $filas-1)
            break;
?>

<tr>
      <td class="subcat" align='center'>
<?php
}
?>
</td>
</tr>
</font>
</td>
</tr>
<tr>
</table>
</tr>

</table>
<h6>           &nb
sp;       
            &
nbsp;
            &
nbsp;       
            &
nbsp;       
Total Por Dependencia
          .............
.    
       C$
 <?php
      echo $aux3;
 ?></h6>
```

```
<tr>
<br>
<p align = "center">
      <a href = "reporte_depen.php"> <img src = "./lg/Atras2.GIF"> </a>
</p>

</tr>

</td>
</tr>

</body>
</html>
```

//Cerrar

```
<?php
session_register("n_usuario");
$_SESSION['n_usuario'] = "";

session_start();
session_destroy();
echo "<script type 'text/javascript'>
              alert ('Su Sesion Ha Sido Cerrada Correctamente');
                  parent.location='index.php';
              </script> ";
?>
```