

UNIVERSIDAD TECNICA DE BABAHOYO



FACULTAD DE ADMINISTRACION FINANZAS E INFORMATICAS

ESCUELA DE SISTEMAS

TESIS DE GRADO PREVIO A LA OBTENCION DEL TITULO DE

INGENIERO EN SISTEMAS

TEMA: “IMPLEMENTACION DE UN SOFTWARE PARA EL CONTROL, RETIRO Y SEGUIMIENTO DE MATERIALES DE OFICINA E INSUMOS MEDICOS EN EL MINISTERIO DE SALUD PUBLICA DEL CANTON BABAHOYO”

PRESENTADO POR: CARLOS EDISON SANCHEZ BAJAÑA

AÑO 2012

BABAHOYO, LOS RIOS

AGRADECIMIENTO

Agradezco a Dios por permitirme alcanzar una meta más en mi vida, a mis padres, mis hermanos por apoyarme siempre en todos los momentos más críticos de mi vida, y que gracias a sus consejos y confianza depositados en mí soy por hoy una persona que anda por el buen camino.

Y ante todo un agradecimiento muy pero muy especial a mi madre que con su apoyo constante e incondicional me ayudado día a día a salir adelante a pesar de los múltiples obstáculos que se me presentaron en pos de realizar este trabajo de tesis.

A mi Director de Tesis Ing. José Mejía Viteri y mi lectora de la misma la Ing. María Gonzales por su apoyo constante y su optimismo en todas las etapas de este proyecto.

A mis amigos que de una u otra manera siempre estuvieron ahí presentes en los momentos buenos y malos y también a quienes me juzgaron sin saberlo.

DEDICATORIA

Dedico este trabajo de tesis a nuestro padre celestial creador del universo que nos permite vivir día a día ya que sin él no fuéramos nada, a mis padres, Sr. Alfonso Sánchez, mi madre la Sra. Jenny Bajaña Naranjo, mis hermanos Paola, Ariel, Joselo, Jaime, Kerly, Cindy y Junior por apoyarme en esta gran batalla en la que se convirtió realizar este trabajo de tesis, y por estar siempre ahí cuando más los necesitaba.

DECLARACION DE AUTORIA

La Responsabilidad por los hechos, ideas y doctrinas expuestos en este trabajo, me corresponden exclusivamente, y el patrimonio intelectual de la misma a la Facultad de Administración Finanzas e Informática de la Escuela de Sistemas de la Universidad Técnica de Babahoyo.

FIRMAS DE RESPONSABILIDAD

El jurado calificador de la Escuela de Sistema – Facultad de Administración Finanzas e Informática de la Universidad Técnica de Babahoyo le da al siguiente proyecto de tesis.

La calificación de:

Equivalente a:

Fecha:

Firman para corroborar su veracidad:

Presidente Tribunal de Defensa

Director de Tesis

Lector Tesis

Secretario

| ÍNDICE DE CONTENIDOS | PÁGINA |
|---|---------------|
| CAPITULO I | 11 |
| 1. EL PROBLEMA | 12 |
| 1.1 PLANTEAMIENTO DEL PROBLEMA | 12 |
| 1.2 FORMULACION DEL PROBLEMA | 12 |
| 1.3 DELIMITACIÓN | 12 |
| 1.4 OBJETIVOS | 13 |
| 1.4.1 OBJETIVO GENERAL | 13 |
| 1.4.2 OBJETIVO ESPECÍFICOS | 13 |
| 1.5 JUSTIFICACION | 13 |
| CAPITULO II | 15 |
| 2. MARCO TEORICO | 16 |
| 2.1 ANTECEDENTES DE LA INVESTIGACION | 16 |
| 2.2 FUNDAMENTACIÓN CIENTIFICA | 17 |
| 2.2.1 HISTORIA DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR | 17 |
| 2.2.2 QUE ES EL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR | 20 |
| 2.2.3 MISIÓN DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR | 21 |
| 2.2.4 VISIÓN DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR | 21 |
| 2.2.5 OBJETIVOS DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR | 21 |
| 2.2.6 PROGRAMAS DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR | 22 |
| 2.2.6.1 DENTRO DE LOS PROGRAMAS TENEMOS LOS SIGUIENTES | 22 |
| 2.2.7 AUTORIDADES DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR | 23 |
| 2.2.8 QUE SON LOS SISTEMAS INFORMÁTICOS | 25 |
| 2.2.8.1 DESARROLLO DE SISTEMAS INFORMÁTICOS | 25 |
| 2.2.8.2 CLASIFICACIÓN DE LOS SISTEMAS INFORMATICOS | 25 |
| 2.2.8.3 VENTAJAS DE LOS SISTEMAS INFORMÁTICOS | 26 |
| 2.2.8.4 DESVENTAJAS DE LOS SISTEMAS INFORMÁTICOS | 27 |
| 2.2.9 POR QUE IMPLEMENTAR LOS SISTEMAS INFORMÁTICOS | 28 |
| 2.2.9.1 CARACTERÍSTICAS DE LOS SISTEMAS INFORMÁTICOS | 28 |
| 2.2.10 QUE SON LOS SISTEMAS WEB | 29 |
| 2.2.10.1 VENTAJAS DE LOS SERVICIOS WEB | 30 |
| 2.2.10.2 DESVENTAJAS DE LOS SERVICIOS WEB | 30 |

| | |
|---|----|
| 2.2.11 QUE ES PHP | 32 |
| 2.2.11.1 VISIÓN GENERAL DE PHP | 33 |
| 2.2.11.2 HISTORIA DE PHP | 34 |
| 2.2.11.3 HISTORIAL DE LANZAMIENTO DE PHP | 35 |
| 2.2.11.4 CARACTERÍSTICAS DE PHP | 38 |
| 2.2.11.5 VENTAJAS DE PHP | 38 |
| 2.2.11.6 DESVENTAJAS DE PHP | 39 |
| 2.2.12 BASES DE DATOS | 40 |
| 2.2.12.1 PARTES QUE COMPONEN UNA BASE DE DATOS | 40 |
| 2.2.12.2 OBJETIVOS DE LOS SISTEMAS DE BASES DE DATOS | 41 |
| 2.2.13 SISTEMA GESTOR DE BASE DE DATOS | 42 |
| 2.2.13.1 FUNCIONES PRINCIPALES DE UN SGBD | 43 |
| 2.2.13.2 FUNCIONES PRINCIPALES DE UN ABD | 43 |
| 2.2.14 MODELO ENTIDAD-RELACIÓN | 44 |
| 2.2.14.1 CARACTERÍSTICAS DE UN MODELO E-R | 44 |
| 2.2.15 ENTIDADES FUERTES Y DÉBILES | 46 |
| 2.2.16 GRADO DE UNA RELACIÓN | 46 |
| 2.2.17 CARDINALIDAD DE LAS RELACIONES. | 46 |
| 2.2.18 PASO DEL MODELO E-R AL MODELO RELACIONAL | 47 |
| 2.2.19 HISTORIA DE SQL | 48 |
| 2.2.19.1 ORÍGENES Y EVOLUCIÓN DE SQL | 49 |
| 2.2.19.2 CARACTERÍSTICAS GENERALES DEL SQL | 50 |
| 2.2.20 LENGUAJE DE DEFINICIÓN DE DATOS (DDL) | 51 |
| 2.2.21 LENGUAJE DE MANIPULACIÓN DE DATOS (DML) | 53 |
| 2.2.22 QUE ES SQL | 58 |
| 2.2.23 QUE ES MYSQL | 59 |
| 2.2.23.1 MOTORES DE ALMACENAMIENTO DE MYSQL Y TIPOS DE TABLAS | 59 |
| 2.2.23.2 MYISAM | 60 |
| 2.2.23.2.1 CARACTERÍSTICAS DE MYISAM | 60 |
| 2.2.23.2.2 OPCIONES DE ARRANQUE DE MYISAM | 62 |
| 2.2.23.3 MERGE | 64 |
| 2.2.23.3.1 DESVENTAJAS DE LAS TABLAS MERGE | 67 |
| 2.2.23.4 MEMORY (HEAP) | 68 |
| 2.2.23.4.1 CARACTERÍSTICAS DE MEMORY | 68 |

| | |
|--|-----------|
| 2.2.23.5 BDB (BERKELEYDB) | 71 |
| 2.2.23.5.1 SISTEMAS OPERATIVOS QUE SOPORTA BDB | 71 |
| 2.2.23.5.2 INSTALACIÓN DE BDB | 72 |
| 2.2.23.5.3 OPCIONES DE ARRANQUE DE BDB | 72 |
| 2.2.23.5.4 CARACTERÍSTICAS DE LAS TABLAS BDB | 74 |
| 2.2.23.6 EXAMPLE | 74 |
| 2.2.23.7 FEDERATED | 75 |
| 2.2.23.7.1 DESCRIPCIÓN DEL MOTOR DE ALMACENAMIENTO FEDERATED | 75 |
| 2.2.23.7.2 CÓMO USAR LAS TABLAS FEDERATED | 76 |
| 2.2.23.8 ARCHIVE | 79 |
| 2.2.23.9 CSV | 79 |
| 2.2.24 QUE ES WAMP SERVER | 80 |
| 2.2.25 INSTALACIÓN DEL WAMP SERVER 2 | 81 |
| 2.2.26 QUÉ SON LAS PÁGINAS DINÁMICAS | 85 |
| 2.2.27 HIPOTESIS Y VARIABLES | 85 |
| 2.2.27.1 HIPOTESIS | 85 |
| 2.2.27.2 VARIABLES | 85 |
| CAPITULO III | 86 |
| 3. MARCO METODOLOGICO | 87 |
| 3.1 MODALIDAD DE LA INVESTIGACION | 87 |
| 3.2 TIPOS DE INVESTIGACIÓN | 87 |
| 3.3 POBLACIÓN Y MUESTRA | 88 |
| 3.4 MÉTODOS, TÉCNICAS E INSTRUMENTOS | 89 |
| 3.5 TABULACION DE RESULTADOS | 92 |
| 3.6 CONCLUSIONES | 102 |
| 3.7 RECOMENDACIONES | 102 |

| | |
|--|-----|
| CAPITULO IV | 103 |
| 4. DESARROLLO TÉCNICO DE LA INVESTIGACION | 104 |
| 4.1 INTRODUCCION | 104 |
| 4.2 OBJETIVO DE LA PROPUESTA | 104 |
| 4.2.1 OBJETIVO GENERAL | 104 |
| 4.2.2 OBJETIVOS ESPECIFICOS | 104 |
| 4.3 METODOLOGIA DE DESARROLLO UTILIZADA | 104 |
| 4.4 ANALISIS PREVIO | 105 |
| 4.5 DISEÑO | 105 |
| 4.5.1 LISTADO DE REQUERIMIENTOS Y FUNCIONES QUE TENDRÁ EL SOFTWARE | 105 |
| 4.5.2 BASE DE DATOS (MODELO CONCEPTUAL Y MODELO FÍSICO) | 107 |
| 4.5.2.1 MODELO CONCEPTUAL | 107 |
| 4.5.2.2 MODELO FÍSICO | 108 |
| 4.5.3 DICCIONARIO DE DATOS | 109 |
| 4.5.4 SCRIPT DE LA BASE DE DATOS | 113 |
| 4.5.5 DIAGRAMA DE CASO DE USO | 116 |
| 4.5.6 DIAGRAMA DE SECUENCIA | 116 |
| 4.5.7 DIAGRAMA DE ACTIVIDADES | 117 |
| 4.5.8 DIAGRAMA DE DESPLIEGUE | 117 |

INDICE DE GRAFICAS

| | |
|---|----|
| 2.2.11 GRAFICO 1 ESQUEMA DE LAS PÁGINAS PHP | 32 |
| 2.2.16 GRAFICO 1 MODELO ENTIDAD-RELACIÓN | 45 |
| 2.2.16 GRAFICO 2 MODELO ENTIDAD-RELACIÓN | 47 |
| 2.2.24 GRAFICO 1 INSTALACIÓN DEL WAMPSEVER2 | 81 |
| 2.2.24 GRAFICO 2 INSTALACIÓN DEL WAMPSEVER2 | 82 |
| 2.2.24 GRAFICO 3 INSTALACIÓN DEL WAMPSEVER2 | 82 |
| 2.2.24 GRAFICO 4 INSTALACIÓN DEL WAMPSEVER2 | 83 |
| 2.2.24 GRAFICO 5 INSTALACIÓN DEL WAMPSEVER2 | 83 |
| 2.2.24 GRAFICO 6 INSTALACIÓN DEL WAMPSEVER2 | 84 |
| 2.2.24 GRAFICO 7 INSTALACIÓN DEL WAMPSEVER2 | 84 |
| 3.5 GRAFICO 1 TABULACION DE RESULTADOS | 92 |

| | |
|---|-----|
| 3.5 GRAFICO 2 TABULACION DE RESULTADOS | 93 |
| 3.5 GRAFICO 3 TABULACION DE RESULTADOS | 94 |
| 3.5 GRAFICO 4 TABULACION DE RESULTADOS | 95 |
| 3.5 GRAFICO 5 TABULACION DE RESULTADOS | 96 |
| 3.5 GRAFICO 6 TABULACION DE RESULTADOS | 97 |
| 3.5 GRAFICO 7 TABULACION DE RESULTADOS | 98 |
| 3.5 GRAFICO 8 TABULACION DE RESULTADOS | 99 |
| 3.5 GRAFICO 9 TABULACION DE RESULTADOS | 100 |
| 3.5 GRAFICO 10 TABULACION DE RESULTADOS | 101 |

INDICE DE TABLAS

| | |
|--|-----|
| 2.2.13 TABLA 1 HISTORIAL DE LANZAMIENTO DE PHP | 37 |
| 2.2.16 TABLA 1 MODELO ENTIDAD-RELACIÓN | 48 |
| 2.2.18 TABLA 1 ORÍGENES Y EVOLUCIÓN DE SQL | 49 |
| 3.3 TABLA 1 POBLACIÓN Y MUESTRA | 88 |
| 3.3 TABLA 1 TABULACION DE RESULTADOS | 92 |
| 3.3 TABLA 2 TABULACION DE RESULTADOS | 93 |
| 3.3 TABLA 3 TABULACION DE RESULTADOS | 94 |
| 3.3 TABLA 4 TABULACION DE RESULTADOS | 95 |
| 3.3 TABLA 5 TABULACION DE RESULTADOS | 96 |
| 3.3 TABLA 6 TABULACION DE RESULTADOS | 97 |
| 3.3 TABLA 7 TABULACION DE RESULTADOS | 98 |
| 3.3 TABLA 8 TABULACION DE RESULTADOS | 99 |
| 3.3 TABLA 9 TABULACION DE RESULTADOS | 100 |
| 3.3 TABLA 10 TABULACION DE RESULTADOS | 101 |

CAPÍTULO I

1. EL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

En el Ministerio de Salud Pública de la Provincia de los Ríos en los procesos de retiro de materiales de oficina así como los pedidos de insumos médicos se dificulta debido a la falta de una clasificación y un orden en pasos de retiro, esto conlleva un inadecuado control de los procesos de organización, entrega y aprobación, de los pedidos necesarios para cada departamento esto hace de una situación caótica al momento de realizar un seguimiento.

En la actualidad todos los pedidos de materiales de oficina e insumos médicos que se realizan en la Institución no constan con un historial de los departamentos y sus necesidades, debido a la falta de información y todas estas dificultades mencionadas anteriormente, provocaban la imposibilidad de realizar la ejecución y entrega de los pedidos.

En ocasiones falta requisitos necesarios para llevar a cabo los procesos de entrega de materiales de oficina e insumos médicos, los beneficiarios se ven afectados al no contar con un seguimiento del mismo, generando pérdida de tiempo al momento de la entrega.

1.2 FORMULACION DEL PROBLEMA

¿Cómo administrar de manera adecuada los procesos de entrega y recepción de materiales de oficina e insumos médicos del Ministerio de Salud Pública de la Provincia de los Ríos?

1.3 DELIMITACIÓN

Objetos de Estudio: Ingeniería de Sistemas

Campo de Acción: Ingeniería del Software

Lugar: "Ministerio de Salud Pública de la Provincia de Los Ríos"

Departamento: Bodega

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Diseñar una Aplicación Web que permita controlar los Materiales de la Bodega del Ministerio de Salud Pública de la Provincia de los Ríos.

1.4.2 OBJETIVO ESPECÍFICOS

- Fundamentar la base teórica y científica que permita el desarrollo de la investigación
- Analizar y preparar información para conocer las mejores soluciones.
- Validar la investigación y resultados con la ayuda de un experto.

1.5 JUSTIFICACION

Del planteamiento del problema se deduce que este existe, y que incide negativamente en el avance y progreso de la institución en el área de bodega de materiales. Se considera que de persistir esta situación problemática, puede acarrear serias dificultades a la institución en desmedro de los afiliados.

Por el contrario si se incrementa una solución al problema; es decir, se instala el sistema, este afectará positivamente en el control de materiales de bodega del “Ministerio de Salud Pública”.

El Ministerio de Salud Publica en su proceso de optimización los procesos de inventarios se han convertido en una necesidad, pues su búsqueda central radica en satisfacer las necesidades y expectativas de los empleados, asegurando un equilibrio ideal entre el mayor nivel de servicio y el menos nivel de inventario posibles.

La correcta administración del inventario es mantener la cantidad suficiente para que no se presenten ni faltantes (stockouts) ni exceso de inventarios (overstock), si un producto no está disponible en el momento en que el empleado lo solicita se mantendrá en espera, esto permitirá hacer ahorros atraves de una óptima gestión de inventarios, es una prioridad inaplazable para el Ministerio de Salud Pública hoy más que nunca los problemas en el control y gestión de inventarios se originan por una deficiente gestión logística y administrativa de la entidad.

Debido a esta gran necesidad de cambio, se desarrollará la automatización de las actividades que se realizan en los diferentes procesos del departamento de servicio general como es el registro de entradas y salidas del inventario de almacén, requisiciones de compras y/o servicios, solicitud y entrega de bienes materiales y suministros.

Al poder realizar rápidamente los procesos, el personal de las diferentes áreas de la institución podrá desenvolverse de manera adecuada.

Técnicamente este proyecto es viable, ya que se cuenta con el apoyo de la institución en el sentido de brindar toda la información necesaria para la elaboración de un Portal Web Dinámico.

CAPÍTULO II

2. MARCO TEORICO

2.1 ANTECEDENTES DE LA INVESTIGACION

Al presente trabajo de investigación, no le antecede proyecto similar luego de buscar las referencias en tesis en la Facultad de Administración Finanzas e Informática (FAFI) de la Universidad Técnica de Babahoyo, motivos por el cual este proyecto surge en base al estudio realizado del control de materiales de oficina e insumos médicos, del Ministerio de Salud Pública de la Provincia de Los Ríos.

Durante este seguimiento se observó que todos los pedidos de materiales de oficina e insumos médicos no constan con un historial de los departamentos, debido a la falta de información de los mismos.

Motivos por el cual la situación se vuelve caótica en la bodega del Ministerio de Salud al momento de la entrega de los materiales e insumos para medicina.

En vista de que no hay un antecedente de investigación, este trabajo tiene toda la pertinencia e importancia de un estudio científico, para de esta manera mejorar positivamente en el control de materiales de bodega del Ministerio de Salud Pública de la Provincia de Los Ríos.

2.2 FUNDAMENTACIÓN CIENTIFICA

2.2.1 HISTORIA DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR

El 16 de junio de 1967, la Asamblea Nacional Constituyente creó el Ministerio de Salud Pública que se encargaría de “Atender las ramas de sanidad, asistencia social y demás que se relacionan con la salud en general”.

Con tal objeto emitió el Decreto N° 84, para cuyo cumplimiento el Ministerio de Previsión Social y Trabajo designó una comisión presidida por el titular de la Subsecretaría de ese entonces, para elaborar el Reglamento organizativo. La Comisión utilizó todos los informes que existían sobre el sector salud y trató de acondicionar una estructura y un reglamento que permitan al nuevo Ministerio iniciar sus actividades en la vida político-administrativa del país. El resultado de este trabajo fue entregado a la Presidencia de la República, con la debida oportunidad y que fue sancionado mediante Resolución 684-A, del 26 de junio de 1967. En la estructura del Ministerio se consideraron tres niveles básicos: Directivo, Operativo y Asesor.

A partir de esa fecha se han sucedido una serie de acontecimientos con el propósito de darle mayor estructura orgánica al nuevo Ministerio, de los cuales señalamos como los más notables, la Integración de las Juntas de Asistencia Social al país a la estructura orgánico funcional del Ministerio (abril de 1972), igual que la Liga Ecuatoriana Antituberculosa (L.E.A.) y la creación de la Subsecretaría de Saneamiento Ambiental, con la adscripción del Instituto Ecuatoriano de Obras Sanitarias (IEOS) en 1975. El principio fundamental de la nueva organización fue: Centralización Normativa y Descentralización Ejecutiva. El Ministerio de Salud Pública se constituyó por consiguiente, en el organismo que en representación de la Función Ejecutiva, formularía y ejecutará la política integral de salud del país. Con el tripe sentido de aumentarla, defenderla y restaurarla como deber del Estado y como derecho inalienable del pueblo ecuatoriano.

Desde entonces, se creó el Sector Salud formado por un conjunto de entidades, organismos e instituciones públicas y privadas, que realizan acciones de salud, tanto como producen servicios, formados de capital social básico o como colaboradores de las autoridades nacionales de salud.

El Sector Salud quedó de esta manera integrado sobre la base del Subsector Público y del Subsector Privado, el primero formado por: Gobierno, entidades descentralizadas y Seguro Social; el segundo por el Subsector Privado organizado con fines de lucro en otros, por el Subsector Liberal.

El organismo rector del sector es el Ministerio de Salud Pública, que fue creado respondiendo a uno de los objetivos del Plan General de Desarrollo, de tal manera que le corresponden actividades de dirección, coordinación, evaluación, fijación y ejecución de esas políticas en el sector salud.

Al recordar la creación del Ministerio de Salud, es importante aportar algunos elementos históricos de juicio para reflexionar sobre la situación actual de la salud pública y de la atención y enseñanza médicas nacionales.

Las realizaciones y logros alcanzados a favor de la Salud Pública bajo la rectoría estatal en estos últimos años son indiscutibles, sin embargo habrá que enmarcarlos dentro de procesos históricos de la realidad nacional, latinoamericana y mundial; procesos modelados por ideologías, intereses políticos, sociales, económicos y culturales. Nuestra sociedad está estructurada bajo un modelo productivo capitalista agro exportador, dependiente de un sistema hegemónico global, que presiona enormemente sobre el diseño y ejecución de políticas; sistema que actualmente se encuentra en crisis y lleno de contradicciones.

En esta sociedad, sólo un pequeño grupo ha estado tradicionalmente incorporado al sistema; en cambio la mayoría, el grupo más débil, producto de su atraso y marginación permanece fuera; y por tanto, sus condiciones de salud y calidad de vida han sido y son muy vulnerables, padeciendo los más severos sufrimientos por enfermedad y muerte.

En 1967, las tasas nacionales de mortalidad infantil y general eran de 93 y 13 por mil, respectivamente, que reflejaban la difícil situación de salud, cuyas causas principales eran varios trastornos prevenibles con mejoramiento del medio, inmunizaciones y alimentación adecuada y sana.

El sistema de servicios de salud atomizado, pobre y auto competitivo estuvo en manos de la Dirección Nacional de Salud, Asistencia Social, Sanidad Militar, algunas municipalidades e instituciones autónomas como la Junta de Beneficencia de Guayaquil, LEA, SOLCA Y SNEM; y los dependientes de la medicina liberal.

Para 1998 pese a los esfuerzos institucionales, los indicadores de salud aún reflejan una situación y calidad de vida deficientes; así, la tasa de mortalidad infantil

se ubica en 44 por mil nacidos vivos, que es 250% más alta que la de Chile; con una distribución del riesgo para enfermar y morir, más alto en zonas urbano-marginales y rurales, en especial aquellas con alta concentración de población indígena. Desde la década de los 80's, el perfil epidemiológico experimenta un notable incremento de patologías de la "modernización" (crónico degenerativas, cardio y cerebro vasculares, violencia, accidentes, cáncer, etc.) asociado a un limitado acceso al saneamiento básico, así como a una severa crisis en la cobertura y calidad de los servicios de salud. Sin embargo, cabe destacar algunos logros alcanzados por el Ministerio de Salud desde su creación: durante el 5to. Velasquismo (1968-1972) se destaca la expedición del Código de Salud para regular las acciones de control sanitario y la implementación del Plan Nacional de Salud Rural. Se erradica la viruela. En la dictadura de Guillermo Rodríguez Lara (1972-1976) se incrementa la oferta de servicios estatales de salud (expansión de cobertura) que continúa durante el Triunvirato Militar (1976-1979) con la construcción generalizada de unidades de baja y mediana complejidad a nivel nacional, así como de infraestructura sanitaria. En este último período se establece el primer programa de capacitación de Colaboradoras Voluntarias Rurales(CVR). En la etapa de retorno a la democracia con el abogado Jaime Roldós Aguilera (1979-1981) y el Dr. Oswaldo Hurtado (1981-1984) se establece la gratuidad de los servicios estatales de salud y se crea la Subsecretaría de Salud de la Región II en Guayaquil, así como el Consejo Nacional de Salud (CONASA. Fueron Capacitados 400 Promotores de Salud y se desencadena un incremento incontrolado de la burocracia.

Durante el neoliberalismo de Febres Cordero (1984-1988) se reorganiza la administración del MSP, se fortalece al IEOS y la infraestructura hospitalaria. Destaca el Programa de Reducción de la Morbi-mortalidad Infantil (PREMI) a cargo de INFA, optimizando inmunizaciones y los programas de control de enfermedades diarreicas respiratorias. Se crea el Centro Estatal de Medicamentos e Insumos Médicos, CEMEIM, para boticas y medicamentos populares y su gratuidad para niños a través del Programa MEGRAME.

La socialdemocracia de Borja Cevallos (1988-1992) reestructura el Plan Nacional de Salud orientado al fortalecimiento de la Atención Primaria de Salud (APS), los Sistemas Locales de Salud (SILOS), la Regionalización y la Descentralización de los Servicios. Se enfrenta y combate eficientemente la primera epidemia nacional

de cólera. Por iniciativa de la representación de OPS/OMS, se propone crear un Sistema Nacional de Salud fortaleciendo al CONASA. El Banco Mundial aprueba el financiamiento del Proyecto FASBASE (Fortalecimiento y Ampliación de los Servicios Básicos de Salud en el Ecuador), con participación familiar y comunitaria. Durante el gobierno del arquitecto Sixto Durán Ballén 1992-1996, por falta de recursos, los servicios primarios se deterioran y se propone buscar apoyo privado. Se materializa el Proyecto FASBASE, y el de Micronutrientes para la corrección del déficit alimenticio. El CONASA elabora un plan para reformar el Sector Salud.

El corto período del abogado Abdalá Bucarán (1996-1997) no logra plasmar acciones concretas en salud. Durante el interinazgo de Fabián Alarcón (1997-1998) la Cartera de Salud enfrenta y resuelve serios conflictos laborales y fortalece su rectoría apoyando la Reforma del Sector y la propuesta de crear un Sistema Nacional de Salud desde el CONASA.

Durante el último período de Jamil Mahuad (1998-2000) y del Dr. Gustavo Noboa, el Ministerio asume las nuevas disposiciones de la Constitución vigente: solidificar el proceso de Reforma del Sector Salud, elaborando una bien estructurada propuesta de Ley del Sistema Nacional de Salud, que fue enviada inicialmente al Ejecutivo discutida en primer debate por el Congreso Nacional y luego aprobada.

Tomado del libro “25 Años de Vida Institucional”, Dr. Julio Larrea y de la revista institucional Info Salud, artículo “35 Años, Ministerio de Salud Pública” del Dr. Antonio Crespo Burgos¹

2.2.2 QUE ES EL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR

Es una institución encargada de velar por el mejoramiento permanente del nivel de salud y bienestar de la población, coadyuvando a la elevación de su calidad de vida que propenda al desarrollo e implantación del Sistema Nacional de Salud. Normalizar, regular y controlar las actividades vinculadas de salud de las personas y el medio ambiente, realizados por entidades públicas y privadas. Promover, desarrollar y difundir actividades de investigación en salud en los ámbitos: básicos, clínicos epidemiológico, ambiental y operativo, que den todas las soluciones de los problemas de salud en el país.

[1]<http://www.msp.gob.ec/index.php/Historia/>

2.2.3 MISIÓN DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR

El Ministerio de Salud Pública del Ecuador, como autoridad sanitaria, ejerce la rectoría, regulación, planificación, gestión, coordinación y control de la salud pública ecuatoriana a través de la vigilancia y control sanitario, atención integral a personas, promoción y prevención, investigación y desarrollo de la ciencia y tecnología, articulación de los actores del sistema, con el fin de garantizar el derecho del pueblo ecuatoriano a la salud.

2.2.4 VISIÓN DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR

El Ministerio de Salud Pública, ejercerá plenamente la gobernanza del Sistema Nacional de Salud, con un modelo referencial en Latinoamérica que priorice la promoción de la salud y la prevención de enfermedades, con altos niveles de atención de calidad, con calidez, garantizando la salud integral de la población y el acceso universal a una red de servicios, con la participación coordinada de organizaciones públicas, privadas y de la comunidad.

2.2.5 OBJETIVOS DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR

- Erradicar la pobreza extrema y el hambre
- Alcanzar la educación primaria universal
- Promover la igualdad de género y empoderar a las mujeres
- Reducir la mortalidad infantil
- Mejorar la salud materna
- Combatir el VIH/SIDA, malaria y otras enfermedades
- Asegurar la sostenibilidad ambiental
- Promover la asociación global para el desarrollo²

[2]<http://www.msp.gob.ec/index.php/Historia/historia-del-ministerio-de-salud-publica.html>

2.2.6 PROGRAMAS DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR

Los programas del Ministerio de Salud Pública están enfocados en el apoyo social de la población ecuatoriana, estando siempre al servicio de la comunidad y de las personas más vulnerables.

Se realizan un conjunto de acciones implementadas por un gobierno con el objetivo de mejorar las condiciones de salud de la población. De esta forma, las autoridades promueven campañas de prevención y garantizan el acceso democrático y masivo a los centros de atención.

Un programa de salud consta de diversas partes. En principio, se plantea una introducción, con los antecedentes y la misión que cumplirá el programa. Después se realiza un diagnóstico de la situación actual, que puede incluir una síntesis de evaluación de planes similares que se hayan desarrollado con anterioridad.

Tras el diagnóstico, se presenta el plan (con la programación de actividades) y, en ocasiones, también se detallan las conclusiones respecto a los resultados que se esperan conseguir. Por lo tanto, el programa de salud es un instrumento para operacionalizar las políticas de salud a través de la planeación, ejecución y evaluación de acciones de promoción, prevención, tratamiento y recuperación de la salud.

2.2.6.1 DENTRO DE LOS PROGRAMAS TENEMOS LOS SIGUIENTES

- Programa Ampliado de Inmunizaciones PAI
- Nutrición
- Tuberculosis
- VIH-SIDA
- Salud intercultural
- Modelo de Atención Integral de Salud
- Atención Integral a Adolescente
- Adulto Mayor
- Maternidad gratuita y atención a la infancia
- Enfermedades dermatológicas
- Control de la Zoonosis
- Mitigación, prevención y atención a emergencias

- Salud Ambiental
- Salud del adulto-enfermedades crónicas no transmisibles
- Salud Mental
- Programa De Control y Vigilancia De Los DDI
- Programa de Enfermedades Catastróficas
- Farmacovigilancia³

2.2.7 AUTORIDADES DEL MINISTERIO DE SALUD PÚBLICA DEL ECUADOR

Dr. David Chiriboga Allnutt

MINISTRO DE SALUD PÚBLICA

Dra. Ximena Abarca Durán

SUBSECRETARIA GENERAL DE SALUD

Dra. Fátima Franco

SUBSECRETARIA REGIONAL COSTA INSULAR

Dr. Xavier Solórzano Salazar

SUBSECRETARIO DE EXTENSIÓN DE LA PROTECCIÓN SOCIAL EN SALUD

* MACRO PROCESO HABILITANTES DE ASESORIA

Eco. Sara Beatriz Naranjo Bonilla

SUBSECRETARIA DE PLANIFICACION

Dra. Vilma Freire

ASESORA MINISTERIAL

Dra. Lorena Gómez Albán

DIRECTORA DEL PROCESO DE DIPLASESE

Dr. Raúl Zambrano Figueroa

DIRECTOR DEL PROCESO DE ASESORIA JURIDICA

Ing. Gustavo Giler Alarcón

COORDINADOR DEL SUBPROCESO DE COOPERACIÓN INTERNACIONAL

Dr. Ramiro López Pulles

DIRECTOR DEL PROCESO DE INVESTIGACIÓN Y DESARROLLO

CPA. Diego Humberto Chávez Ruiz

DIRECTOR DEL PROCESO DE AUDITORIA INTERNA

[3]<http://www.msp.gob.ec/index.php/introduccion.html>

Lcdo. Paolo Muñoz Vásquez

DIRECTOR DEL PROCESO DE COMUNICACIÓN SOCIAL

Dr. Ernesto Chamorro

DIRECTOR DEL PROCESO DE GESTION DE RECURSOS HUMANOS

Eco. Pedro Francisco Núñez Gómez

DIRECTOR DEL PROCESO DE GESTION ADMINISTRATIVA

Ing. Vinicio Alarcón

DIRECTOR PROCESO DE GESTION FINANCIERA

* MACRO PROCESOS AGREGADORES DE VALOR

Dra. María del Carmen Laspina Arellano

DIRECTORA GENERAL DE SALUD

Dr. Xavier Toledo Rodríguez

DIRECTOR DEL PROCESO DE CALIFICACIÓN DE LA DEMANDA Y OFERTA NACIONAL EN SALUD

Dr. Gioconda Gavilanes

DIRECTORA DEL PROCESO DE NORMATIZACION DEL SISTEMA NACIONAL DE SALUD

Dra. Miriam del Rocío Conejo Maldonado

DIRECTOR DEL SUBPROCESO DE MEDICINA CULTURAL

Dr. Juan Martin Moreira Viteri

DIRECTOR DEL PROCESO DE CONTROL Y MEJORAMIENTO EN SALUD PÚBLICA

Dra. Elva Judith Romero Romero

DIRECTORA DEL PROCESO DE CONTROL Y MEJORAMIENTO EN GESTIÓN SERVICIOS DE SALUD

Dra. Lorena Ruiz Abril

DIRECTORA DEL PROCESO DE CONTROL Y MEJORAMIENTO EN VIGILANCIA SANITARIA.

Dra. Nancy Vascones Galarza

RESPONSABLE DEL PROGRAMA AMPLIADO DE INMUNIZACIONES PAI

Dr. Pablo Zambrano⁴

[4]Gestión de Recursos Humanos MSP

2.2.8 QUE SON LOS SISTEMAS INFORMÁTICOS

Un sistema informático como todo sistema, es el conjunto de partes interrelacionadas hardware software y de recurso humano(humanware) que permite almacenar y procesar información. El hardware incluye computadoras o cualquier otro tipo de dispositivo electrónico inteligente, que consisten en procesadores, memoria, sistema de almacenamiento externo, etc. El software incluye al sistema operativo, firmware y aplicaciones, siendo especialmente importante los sistemas de gestión de bases de datos. Por último el soporte humano incluye al personal técnico que crean y mantienen el sistema (analistas, programadores, operarios, etc.) y a los usuarios que lo utilizan.

2.2.8.1 DESARROLLO DE SISTEMAS INFORMÁTICOS

Los sistemas informáticos pasan por diferentes fases en su ciclo de vida, desde la captura de requisitos hasta el mantenimiento. En la actualidad se emplean numerosos sistemas informáticos en la administración pública.

ESTRUCTURAS

Los sistemas informáticos suelen estructurarse en Subsistemas.

- Subsistema físico: asociado al hardware. Incluye entre otros elementos la CPU memoria principal, la placa base, periféricos de entrada y salida, etc.
- Subsistema lógico: asociado al software y la arquitectura. Incluye al sistema operativo, el firmware, las aplicaciones y las bases de datos.

2.2.8.2 CLASIFICACIÓN DE LOS SISTEMAS INFORMATICOS

Los S.I. pueden clasificarse en base a numerosos criterios. Por supuesto las clasificaciones no son estancas y es común encontrar sistemas híbridos que no encajen en una única categoría.

POR SU USO PUEDEN SER:

- De uso general
- De uso específico

EL PARALELISMO DE LOS PROCESADORES, QUE PUEDE SER:

- SISD: Single Instruction Single Data
- SIMD: Single Instruction Multiple Data
- MIMD: Multiple Instruction Multiple Data

POR EL TIPO DE ORDENADOR UTILIZADO EN EL SISTEMA:

- Estaciones de trabajo (Workstations)
- Terminales ligeros (Thinclients)
- Microordenadores (por ejemplo ordenadores personales)
- Miniordenadores (servidores pequeños)
- Macro ordenadores (servidores de gran capacidad)
- Superordenadores

POR LA ARQUITECTURA:

- Sistema aislado
- Arquitectura cliente-servidor
- Arquitectura de 3 capas
- Arquitectura de n capas
- Servidor de aplicaciones
- Monitor de teleproceso o servidor de transacciones⁵

2.2.8.3 VENTAJAS DE LOS SISTEMAS INFORMÁTICOS

- Nos ayuda a realizar los trabajos de una forma más rápida
- Nos permite llevar un mejor control de los datos
- Ganamos eficiencia y capacidad de respuestas y ello se debe a que la información está disponible al instante, donde y cuando se necesite, incluso fuera de la oficina o en el extranjero.
- Integridad de la información. Hace referencia a que la información se encuentra en óptimas condiciones al estar en un medio informático seguro y diseñado especialmente para el fin establecido.
- Almacenamiento ordenado de información.
- La información es guardada según una estructura bien diseñada con el fin de tener acceso fácil y rápido a ella en cualquier momento.
- Centralización de la información
- El sistema informático será la fuente de información principal, haciendo que documentos, recibos y demás información sea manejada por él.

[5]http://es.wikipedia.org/wiki/Sistema_inform%C3%A1tico

- Claridad en los procesos.
Los procesos alrededor del sistema de información se tornarán más claros pues este será un gestor que sigue unas reglas claras y ágiles.
- Confidencialidad de la información y seguridad.
Solo personal autorizado tiene acceso a determinada información.
- Rapidez en la generación de reportes.
Un sistema informático debe estar en capacidad de generar reportes tanto externos (para otras entidades), como internos de forma rápida pues tiene una velocidad alta de procesamiento.
- Disponibilidad de datos estadísticos.
Al contar con información debidamente organizada.

2.2.8.4 DESVENTAJAS DE LOS SISTEMAS INFORMÁTICOS

- Sin el ser humano los sistemas informáticos quedarían inservibles en una semana.
- Gran capital
- Decremento severo en la flexibilidad
- Problemas con la seguridad de los datos si no se manejan con confidencialidad
- Negativa de ciertas áreas de las empresas a compartir información con otras
- Incremento en la dependencia del mantenimiento y reparación⁶.

[6][http://www.buenastareas.com/ensayos/Ventajas-De-Los-Sistemas- Inform%C3%A1ticos/12404](http://www.buenastareas.com/ensayos/Ventajas-De-Los-Sistemas-Inform%C3%A1ticos/12404)

2.2.9 POR QUE IMPLEMENTAR LOS SISTEMAS INFORMÁTICOS

Hoy en día la mayoría de las tareas de oficina en empresas, o incluso en lugares tales como bancos, comercios, inmobiliarias, agencias de viajes, etc. Son realizadas y controladas por los diferentes sistemas informáticos que nos ayudan a facilitar dichos deberes, y lo cierto es que, los mismos son más que útiles si consideramos la cantidad de oportunidades que se nos presentan con estos sistemas. En primer lugar uno de los aspectos más importantes de los mismos, es que nos permiten crear una base de dato realmente muy amplia en la cual podemos almacenar todo tipo de archivos, y precisamente esta es la utilidad más común que suele dársele a los sistemas informático.

De hecho, incluso en nuestra vivienda disponemos de estas herramientas. No obstante, los sistemas informáticos no se limitan únicamente a las bases de datos, sino que podemos aprovecharlos en muchos otros sentidos, especialmente en una empresa. En primer lugar haciendo referencia a esto debemos decir que los sistemas suelen estar conectados entre sí a través de una red, mediante la cual viaja todo tipo de información. Justamente por esta razón es que los sistemas informáticos son fundamentales para algunas instituciones, ya que es una manera ideal de agilizar mucho el trabajo que se tenga que hacer

2.2.9.1 CARACTERÍSTICAS DE LOS SISTEMAS INFORMÁTICOS

- Los sistemas Informáticos son manipulables.
- Podemos determinar el tipo de instrucción a seguir.
- Se puede visualizar y transferir información.
- Realizar instrucciones y supervisión de las operaciones que se están realizando.⁷

[7] <http://www.cursosdeofimatica.net/sistemas-informaticos.html>

2.2.10 QUE SON LOS SISTEMAS WEB

Un servicio web (en inglés, Web service) es una pieza de software que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

ESTÁNDARES EMPLEADOS

Web Services Protocol Stack: Así se denomina al conjunto de servicios y protocolos de los servicios Web que ofrece jota.

XML (Extensible Markup Language): Es el formato estándar para los datos que se vayan a intercambiar.

SOAP (Simple Object Access Protocol) o XML-RPC (XML Remote Procedure Call): Protocolos sobre los que se establece el intercambio.

Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP, FTP, o SMTP.

WSDL (Web Services Description Language): Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.

UDDI (Universal Description Discovery and Integration): Protocolo para publicar la información de los servicios Web. Permite comprobar qué servicios web están disponibles.

WS-Security (Web Service Security): Protocolo de seguridad aceptado como estándar por OASIS (Organization for the Advancement of Structured Information Standards). Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

2.2.10.1 VENTAJAS DE LOS SERVICIOS WEB

Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.

Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.

Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

2.2.10.2 DESVENTAJAS DE LOS SERVICIOS WEB

Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida como CORBA (Common Object Request Broker Architecture).

Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como RMI (Remote Method Invocation), CORBA o DCOM (Distributed Component Object Model). Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de XML no se encuentra la concisión ni la eficacia de procesamiento.

Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.

RAZONES PARA CREAR SERVICIOS WEB

La principal razón para usar servicios Web es que se pueden utilizar con HTTP sobre TCP (Transmisión Control Protocol) en el puerto 80. Dado que las organizaciones protegen sus redes mediante firewalls -que filtran y bloquean gran parte del tráfico de Internet-, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios Web utilizan este puerto, por la simple razón de que no resultan bloqueados. Es importante señalar que los servicios web se pueden utilizar sobre cualquier protocolo, sin embargo, TCP es el más común.

Otra razón es que, antes de que existiera SOAP, no había buenas interfaces para acceder a las funcionalidades de otros ordenadores en red. Las que había eran ad

hoc y poco conocidas, tales como EDI (Electronic Data Interchange), RPC (Remote Procedure Call) u otras APIS.

Una tercera razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada.

Se espera que para los próximos años mejoren la calidad y cantidad de servicios ofrecidos basados en los nuevos estándares.

PLATAFORMAS

- Servidores de aplicaciones para servicios Web:
- JBoss servidor de aplicaciones J2EE Open Source de Red Hat inc.
- Oracle Fusion Middleware
- IBM Lotus Domino a partir de la versión 7.0
- Axis y el servidor JakartaTomcat (de Apache)
- ColdFusion MX de [[Macromedia]httpd]
- Java Web Services Development Pack (JWSDP) de Sun Microsystems (basado en JakartaTomcat)
- JOnAS (parte de Object Web una iniciativa de código abierto)
- Microsoft .NET
- Novell exteNd (basado en la plataforma J2EE)
- WebLogic
- WebSphere
- JAX-WS con GlassFish
- Zope es un servidor de aplicaciones Web orientado a objetos desarrollado en el lenguaje de programación Python
- VERASTREAM de Attachmate WRQ para modernizar o integrar aplicaciones host IBM y VT
- PHP⁸

[8]http://es.wikipedia.org/wiki/Servicio_web

2.2.11 QUE ES PHP

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+⁹.

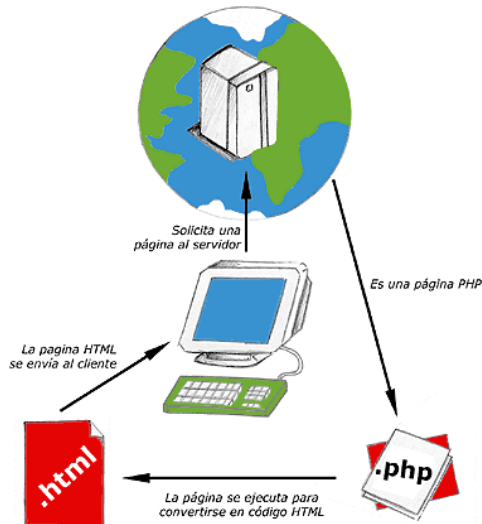


Grafico #1. Esquema del funcionamiento de las páginas PHP

Grafico[1]:<http://www.google.com.ec>
[9]<http://es.wikipedia.org/wiki/PHP>

2.2.11.1 VISIÓN GENERAL DE PHP

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente *PHP Tools*, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

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

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

Aunque todo en su diseño está orientado a facilitar la creación de sitios webs, es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de órdenes, de la misma manera como Perl o Python pueden hacerlo; a esta versión de PHP se la llama PHP-CLI (Command Line Interface).

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos).

2.2.11.2 HISTORIA DE PHP

Fue originalmente diseñado en Perl, con base en la escritura de un grupo de CGI binarios escritos en el lenguaje C por el programador danés-canadiense Rasmus Lerdorf en el año 1994 para mostrar su currículum vitae y guardar ciertos datos, como la cantidad de tráfico que su página web recibía. El 8 de junio de 1995 fue publicado "Personal Home Page Tools" después de que Lerdorf lo combinara con su propio Form Interpreter para crear PHP/FI.

Dos programadores israelíes del Technion, ZeevSuraski y AndiGutmans, reescribieron el analizador sintáctico (*parser* en inglés) en el año 1997 y crearon la base del PHP3, cambiando el nombre del lenguaje a la forma actual. Inmediatamente comenzaron experimentaciones públicas de PHP3 y fue publicado oficialmente en junio de 1998.

Para 1999, Suraski y Gutmans reescribieron el código de PHP, produciendo lo que hoy se conoce como motor Zend. También fundaron Zend Technologies en RamatGan, Israel.

En mayo de 2000 PHP 4 fue lanzado bajo el poder del motor ZendEngine 1.0. El día 13 de julio de 2007 se anunció la suspensión del soporte y desarrollo de la versión 4 de PHP,² a pesar de lo anunciado se ha liberado una nueva versión con mejoras de seguridad, la 4.4.8 publicada el 13 de enero del 2008 y posteriormente la versión 4.4.9 publicada el 7 de agosto de 2008. Según esta noticia se dará soporte a fallos críticos hasta el 9 de agosto de 2008.

El 13 de julio de 2004, fue lanzado PHP 5, utilizando el motor ZendEngine 2.0 (o ZendEngine 2). Incluye todas las ventajas que provee el nuevo ZendEngine 2 con. Mejor soporte para la programación orientada a objetos, que en versiones anteriores era extremadamente rudimentario.

Mejoras de rendimiento.

Mejor soporte para MySQL con extensión completamente reescrita.

Mejor soporte a XML (XPath, DOM, etc.).

Soporte nativo para SQLite.

Soporte integrado para SOAP.

Iteradores de datos.

Manejo de excepciones.

Mejoras con la implementación con Oracle.

2.2.11.3 HISTORIAL DE LANZAMIENTO DE PHP

| | |
|-------|--------------------|
| | Significado |
| Rojo | Sin soporte |
| Verde | Con soporte |
| Azul | Lanzamiento futuro |

| Versión mayor | Versión menor | Fecha de lanzamiento | Notas |
|---------------|---------------|----------------------|---|
| 1 | 1.0.0 | 1995-06-08 | Llamada oficialmente "Personal Home Page Tools (PHP Tools)". Es la primera versión en usar el nombre "PHP". ⁴ |
| 2 | 2.0.0 | 1997-11-01 | Considerada por sus creadores la "herramienta más rápida y sencilla" para crear páginas web dinámicas. ⁴ |
| 3 | 3.0.0 | 1998-06-06 | ZeevSuraski y Andi Gutmans reescribieron la base para esta versión. ⁴ |
| 4 | 4.0.0 | 2000-05-22 | Se añadió un sistema de parsing de dos fases llamado motor Zend. ⁵ |
| | 4.1.0 | 2001-12-10 | Se introducen 'super globales' (\$_GET, \$_POST, \$_SESSION, etc.) ⁵ |
| | 4.2.0 | 2002-04-22 | Se deshabilitan register_globals por defecto. Datos recibidos por la red no son insertados en el espacio de nombres global, cerrando posibles agujeros de seguridad en las aplicaciones. ⁵ |
| | 4.3.0 | 2002-12-27 | Se introduce <u>CLI</u> , y CGI. ⁵⁶ |
| | 4.4.0 | 2005-07-11 | Se añaden páginas man para phpize y php-config. ⁵ |
| | 4.4.9 | 2008-08-07 | Mejoras de seguridad y arreglo de bugs. |

| | | | |
|---|--------|------------|---|
| | | | Último lanzamiento de la serie PHP 4.4. ⁷⁸ |
| 5 | 5.0.0 | 2004-07-13 | Motor Zend II con un nuevo modelo de objetos. ⁹ |
| | 5.1.0 | 2005-11-24 | Mejoras de rendimiento con la introducción de variables de compilador para el nuevo motor de PHP. ⁹ |
| | 5.2.0 | 2006-11-02 | Actividad extensión de filtro por defecto. Soporte <u>JSON</u> nativo. ⁹ |
| | 5.2.17 | 2011-01-06 | Arreglo de vulnerabilidad crítica conectada a punto flotante. |
| | 5.3.0 | 2009-06-30 | Soporte para espacios de nombres ; etiqueta de salto (goto limitada), <u>Clausuras</u> nativas, Soporte nativo para archivos PHP (phar), recolección de basura para referencias circulares, soporte mejorado para <u>Windows</u> , <u>sqlite3</u> , <u>mysqlnd</u> como reemplazo a <u>libmysql</u> como librería para extensiones que funcionan con <u>MySQL</u> , <u>fileinfo</u> como reemplazo de <u>mime_magic</u> para mejor soporte <u>MIME</u> , extensión de <u>Internacionalización</u> , y <u>etiqueraereg</u> obsoleta. |
| | 5.3.1 | 2009-11-19 | Cerca de 100 <u>bug fixes</u> . |
| | 5.3.2 | 2010-03-04 | Gran número de bug fixes. |
| | 5.3.3 | 2010-07-22 | Mayoritariamente arreglo de errores; <u>FPM SAPI</u> . |

| | | | |
|---|-------------|-------------|--|
| | 5.3.4 | 2010-12-10 | Mejoras a <u>FPM</u> SAPI. |
| | 5.3.5 | 2011-01-06 | Arreglo de vulnerabilidad crítica relacionada a punto flotante. |
| | 5.3.6 | 2011-03-10 | Cerca de 60 bug fixes. |
| | 5.3.7 | 2011-08-18 | Esta versión se enfoca en la estabilidad con cerca de 90 bug fixes, algunos de los cuales tienen implicaciones a la seguridad. |
| | 5.3.8 | 2011-08-23 | Esta versión arregla dos errores introducidos por la versión 5.3.7. |
| | 5.4.0alpha3 | 2011-08-03 | Items removidos: register_globals, safe_mode, allow_call_time_pass_reference, session_register(), session_unregister() y session_is_registered(). Mejoras a características existentes. |
| 6 | ?.? | No date set | El desarrollo de PHP 6 ha sido retrasado porque los desarrolladores decidieron que el enfoque actual para tratar cadenas Unicode no es correcto, y están considerando formas alternas para la siguiente versión. |

TABLA# 1 2.2.11.3 HISTORIAL DE LANZAMIENTO DE PHP

2.2.11.4 CARACTERÍSTICAS DE PHP

- Más Rápido que ASP.
- Lenguaje más fácil y potente.
- Integración perfecta más de ocho servidores HTTP.
- Diseño Modular de fácil ampliación.
- Licencia Abierta.
- Acceso a 20 tipos de Bases de Datos.

2.2.11.5 VENTAJAS DE PHP

- Es un lenguaje multiplataforma.
- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).

- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño. Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

2.2.11.6 DESVENTAJAS DE PHP

- No posee una abstracción de base de datos estándar, sino bibliotecas especializadas para cada motor (a veces más de una para el mismo motor).
- No posee adecuado manejo de internacionalización, unicode, etc.
- Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar.
- Por sus características favorece la creación de código desordenado y complejo de mantener.
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable.
- Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (o MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes¹⁰.

[10]<http://es.wikipedia.org/wiki/PHP>

2.2.12 BASES DE DATOS

INTRODUCCIÓN

Como sabemos una base de datos o banco de datos, es un conjunto de datos pertenecientes a un mismo contexto y 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 la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

En la actualidad existen programas denominados sistemas gestores de bases de datos, abreviado SGBD, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Un objetivo importante de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de los datos, es decir, esconder ciertos detalles de cómo se almacenan y mantienen los datos. Sin embargo para que el sistema sea manejable, los datos se deben extraer eficientemente.

Los sistemas de base de datos se diseñan para manejar grandes cantidades de información. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

2.2.12.1 PARTES QUE COMPONEN UNA BASE DE DATOS

Una base de datos está compuesta por:

- Hardware: equipo informático
- Software: El SGBD (Sistema Gestor de Bases de Datos) es la parte más importante del software de un sistema de base de datos. Es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Por ejemplo, Access es un SGBD, que actúa como interfaz entre los datos y los usuarios finales. Más adelante se detallarán aspectos relevantes de un SGBD.

- Usuarios: Podemos definir a los usuarios como toda persona que tenga todo tipo de contacto con el sistema de base de datos, desde quién la diseña, elabora, termina, hasta los usuarios finales.

2.2.12.2 OBJETIVOS DE LOS SISTEMAS DE BASES DE DATOS

Los objetivos principales de un sistema de base de datos son:

Disminución de la redundancia y eliminación de la inconsistencia de datos.

Si no se controla detalladamente el almacenamiento, se pueda originar un duplicado de información, es decir, que la misma información esté más de una vez en un dispositivo de almacenamiento. Esto aumenta los costos de almacenamiento y acceso a los datos, además de que puede originar la inconsistencia de los datos (diversas copias de un mismo dato no concuerdan entre sí).

Minimizar la dificultad para tener acceso a los datos.

Un sistema de base de datos debe contemplar un entorno de datos que le facilite al usuario el manejo de los mismos. Supóngase un banco, y que uno de los gerentes necesita averiguar los nombres de todos los clientes que viven dentro del código postal 48733. El gerente pide al departamento de procesamiento de datos que genere la lista correspondiente. Puesto que esta situación no fue prevista en el diseño del sistema, no existe ninguna aplicación de consulta que permita este tipo de solicitud, y esto ocasiona una deficiencia del sistema.

Evitar anomalías del acceso concurrente.

Para mejorar el funcionamiento global del sistema y obtener un tiempo de respuesta más rápido, muchos sistemas permiten que múltiples usuarios actualicen los datos simultáneamente. En un entorno así, la interacción de actualizaciones concurrentes puede dar por resultado datos inconsistentes. Para prevenir esta posibilidad debe mantenerse alguna forma de supervisión en el sistema.

Permitir restricciones de seguridad.

La información de toda empresa es importante, aunque unos datos lo son más que otros, por tal motivo se debe considerar el control de acceso a los mismos, no todos los usuarios pueden visualizar alguna información, por tal motivo para que un

sistema de base de datos sea confiable debe mantener un grado de seguridad que garantice la autenticación y protección de los datos.

Asegurar el mantenimiento de integridad.

Los valores de datos almacenados en la base de datos deben ser correctos.

Pueden ser incorrectos debido a:

Errores en la entrada de datos (Prtugalete en vez de Portugalete)

Errores en el programa de aplicación (Cálculo incorrecto del IVA soportado)

Falsificación deliberada

(A nivel de programación, el uso de cuadros combinados y listas facilitan la integridad).

El nivel interno depende del SGBD elegido. Y será este el que implemente (realice) esta tarea. Cuando usamos Access, no sabemos cómo almacena físicamente los datos ni dónde. Access es en el intermediario entre usuarios y el SO(Windows). Los otros dos niveles (Conceptual y Externo) son tarea del Administrador de Bases de Datos (ABD), que es la persona encargada y con el control total sobre el sistema de base de datos.¹¹

2.2.13 SISTEMA GESTOR DE BASE DE DATOS

El SGBD es la porción más importante del software de un sistema de base de datos.

El objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos.

La figura muestra el SGBD como interfaz (intermediario) entre la base de datos física y las peticiones del usuario.

El SGBD interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria (discos) y la memoria principal.

En sí, un SGBD es el corazón de la base de datos ya que se encarga del control total de los posibles aspectos que la puedan afectar.

[11]http://es.wikipedia.org/wiki/Base_de_datos

2.2.13.1 FUNCIONES PRINCIPALES DE UN SGBD

- Crear y organizar la Base de Datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accedidos rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Responsable del verdadero almacenamiento de los datos.
- Respaldo y recuperación.
- Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallos en el sistema de base de datos.
- Control de concurrencia. Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- Seguridad e integridad.

2.2.13.2 FUNCIONES PRINCIPALES DE UN ABD

- Definición de esquema (diseño de la BD).
- Definición de la estructura de almacenamiento según el SGBD elegido
- (En Access ^ Tablas y relaciones).
- Concesión de autorización para el acceso a los datos.
- Especificación de límites de integridad.
- Aunque es un tarea de propia de usuarios informáticos, también puede encargarse de:
- Programación del interfaz de la BD con los usuarios finales (Consultas, Formularios e Informes, Módulos de código)
- Para realizar el Nivel Conceptual, un ABD se servirá de un modelo de datos, que es una representación de la realidad mediante una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, la semántica asociada a los datos y las restricciones de consistencia.
- Existen diferentes modelos de datos, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad-Relación. La realización de un modelo Entidad-Relación supone siempre el paso previo al futuro diseño de una BD en cualquiera de los modelos existentes (Relacional Jerárquico, en Red,...)¹²

2.2.14 MODELO ENTIDAD-RELACIÓN

Denominado por sus siglas como E-R, este modelo representa la realidad a través de una representación gráfica que incorpora información relativa a los datos y las relaciones entre ellos.

Se utiliza como punto de partida en el diseño de una BD, para después transformarlo al modelo elegido (en nuestro caso el Relacional).

2.2.14.1 CARACTERÍSTICAS DE UN MODELO E-R

- Refleja tan sólo la existencia de datos, no lo que se hacen con ellos.
- Se incluyen todos los datos del sistema en estudio, y por tanto, no está orientado a aplicaciones particulares.
- Es independiente de la base de datos y del sistema operativo elegido.
- No se tienen en cuenta restricciones de espacio, almacenamiento ni tiempo de ejecución.
- Está abierto a la evolución del sistema.
- El modelo E-R se basa en la percepción del mundo real, que consiste en un conjunto de objetos básicos llamados Entidades así como de las Relaciones entre ellos.

ENTIDADES. OCURRENCIAS. ATRIBUTOS. CLAVES Y RELACIONES.

Una Entidad es una cosa u objeto concreto o abstracto que existe, que puede distinguirse de otros y del que se desea almacenar información.

Los alumnos de una escuela forman la Entidad Alumnos.

Juan Alberdi, de Santurtzi y con DNI 11.635.326P, es una Ocurrencia de la Entidad Alumnos, ya que identifica de forma única a una persona dentro de ese universo.

Las características de las Ocurrencias de las Entidades se llaman Atributos. Por ejemplo, el Nombre, DNI y Ciudad son Atributos de Alumnos.

Los Atributos de una entidad pueden tomar un conjunto de valores permitidos al que se le conoce como Dominio del atributo.

[12]html.rincondelvago.com/bases-de-datos_3.html

Una Clave es un conjunto de 1 o más atributos, que permiten identificar de forma única a una ocurrencia dentro de una entidad. Además añadir que ese conjunto de atributos debe ser mínimo, es decir, que ningún subconjunto de atributos pueda también funcionar como clave

Una Clave se especifica gráficamente en el modelo E-R con una línea debajo del nombre del atributo.

Se denomina Clave Secundaria o Extranjera al conjunto de atributos de una entidad que son Clave Primaria en otra.

Entidad Alumnos Ocurrencias

Juan Alberdi Santurtzi 11.635.326P

Luis Alfageme Sestao 41.485.256H .

Rosa Ma Sanz Portugalete 15.123.563K .

Amparo Rivas Santurtzi 23.236.896N .

Roberto Pozas Sestao 15.365.452F1

Atributos

Una Relación es una asociación entre varias Entidades. Pueden contener también Atributos, que representan características propias de dicha asociación.



Grafico #12.2.14 MODELO ENTIDAD-RELACIÓN

Formas Normales de Bases de Datos

Las formas normales son aplicadas a las tablas de una base de datos. Decir que una base de datos está en la forma normal **N** es decir que todas sus tablas están en la forma normal **N**.



Diagrama de inclusión de todas las formas normales.

En general, las primeras tres formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos. El creador de estas 3 primeras formas normales (o reglas) fue Edgar F Codd.

Primera Forma Normal (1FN)

Una tabla está en Primera Forma Normal si:

- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son indivisibles, mínimos.
- La tabla contiene una llave primaria única.
- La llave primaria no contiene atributos nulos.
- No debe existir variación en el número de columnas.
- Los Campos no llave deben identificarse por la llave (Dependencia Funcional)
- Debe Existir una independencia del orden tanto de las filas como de las columnas, es decir, si los datos cambian de orden no deben cambiar sus significados

- Una tabla no puede tener múltiples valores en cada columna.
- Los datos son atómicos (a cada valor de X le pertenece un valor de Y y viceversa).

Esta forma normal elimina los valores repetidos dentro de una BD

Segunda Forma Normal (2FN).

Dependencia Funcional. Una relación está en 2FN si está en 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir que no existen dependencias parciales. (Todos los atributos que no son clave principal deben depender únicamente de la clave principal).

En otras palabras podríamos decir que la segunda forma normal está basada en el concepto de dependencia completamente funcional. Una dependencia funcional $x \rightarrow y$ es completamente funcional si al eliminar los atributos A de X significa que la dependencia no es mantenida, esto es que $A \in X, X - \{A\} \not\rightarrow Y$. Una dependencia funcional $x \rightarrow y$ es una dependencia parcial si hay algunos atributos $A \in X$ que pueden ser eliminados de X y la dependencia todavía se mantiene, esto es $A \in X, X - \{A\} \rightarrow Y$.

Por ejemplo $\{\text{DNI}, \text{ID_PROYECTO}\} \rightarrow \text{HORAS_TRABAJO}$ (con el DNI de un empleado y el ID de un proyecto sabemos cuántas horas de trabajo por semana trabaja un empleado en dicho proyecto) es completamente dependiente dado que ni $\text{DNI} \rightarrow \text{HORAS_TRABAJO}$ ni $\text{ID_PROYECTO} \rightarrow \text{HORAS_TRABAJO}$ mantienen la dependencia. Sin embargo $\{\text{DNI}, \text{ID_PROYECTO}\} \rightarrow \text{NOMBRE_EMPLEADO}$ es parcialmente dependiente dado que $\text{DNI} \rightarrow \text{NOMBRE_EMPLEADO}$ mantiene la dependencia.

Tercera Forma Normal (3FN).

La tabla se encuentra en 3FN si es 2FN y si no existe ninguna dependencia funcional transitiva entre los atributos que no son clave.

Un ejemplo de este concepto sería que, una dependencia funcional $X \rightarrow Y$ en un esquema de relación R es una dependencia transitiva si hay un conjunto de atributos Z que no es un subconjunto de alguna clave de R , donde se mantiene $X \rightarrow Z$ y $Z \rightarrow Y$.

Por ejemplo, la dependencia $SSN \rightarrow DMGRSSN$ es una dependencia transitiva en EMP_DEPT de la siguiente figura. Decimos que la dependencia de $DMGRSSN$ el atributo clave SSN es transitiva vía $DNUMBER$ porque las dependencias $SSN \rightarrow DNUMBER$ y $DNUMBER \rightarrow DMGRSSN$ son mantenidas, y $DNUMBER$ no es un subconjunto de la clave de EMP_DEPT . Intuitivamente, podemos ver que la dependencia de $DMGRSSN$ sobre $DNUMBER$ es indeseable en EMP_DEPT dado que $DNUMBER$ no es una clave de EMP_DEPT .

Formalmente, un esquema de relación R está en 3 Forma Normal Elmasri-Navathe, si para toda dependencia funcional $X \rightarrow A$, se cumple al menos una de las siguientes condiciones:

1. X es súper llave o clave.
2. A es atributo primo de R ; esto es, si es miembro de alguna clave en R .

Además el esquema debe cumplir necesariamente, con las condiciones de segunda forma normal.

2.2.15 ENTIDADES FUERTES Y DÉBILES

En función de las claves, las entidades pueden clasificarse en:

Entidades fuertes: Entidades con una clave primaria.

Entidades débiles: Entidades que no poseen los atributos necesarios para definir una clave primaria ya que dependen de una entidad fuerte. No obstante es necesario dotar a la entidad de una clave primaria. Esta estará formada por la Clave Primaria de la entidad fuerte más 1 o más atributos de la entidad débil (a dichos atributos de la entidad débil se le llama Discriminador). A veces es posible crear una Clave primaria en la entidad débil con un nuevo atributo más identificativo.

Nota: Esta es la definición que el modelo E-R hace de las entidades fuertes y débiles. En el modelo relacional se ofrece otra que más adelante se detallará.

2.2.16 GRADO DE UNA RELACIÓN.

La cantidad de entidades que intervienen en una relación determina el Grado de la relación.

Por ejemplo la relación NOTAS del ejemplo anterior es de grado 2, ya que intervienen la entidad ALUMNOS y la entidad ASIGNATURAS.

Aunque el modelo E-R permite relaciones de cualquier grado, la mayoría de las aplicaciones del modelo sólo consideran relaciones del grado 2. Cuando son de tal tipo, se denominan Relaciones Binarias.

Un ejemplo de relación ternaria:

PADRES

MADRES

2.2.17 CARDINALIDAD DE LAS RELACIONES.

Otra de las características importantes que hay que tener en cuenta en este modelo es la cardinalidad de cada extremo en una relación. La cardinalidad expresa cuántas del conjunto de entidades de un extremo de la relación están relacionadas con cuántas entidades del conjunto del otro extremo. Pueden ser ``uno a uno'', ``uno a varios'' o ``varios a varios''. Por ejemplo, un artículo puede ser escrito por un solo autor o por varios, pero nunca por ninguno; un autor puede pertenecer a exactamente una institución (no para cero o varias); un artículo puede tener cero, uno o varios experimentos. Finalmente, un autor puede escribir muchos artículos, o ninguno. Observe que las cardinalidades en algunos casos dependen

de restricciones arbitrarias: se podría decidir aceptar sólo aquellos autores que han escrito al menos un artículo (y con esto cambiaría la última regla mencionada); hemos decidido considerar sólo la institución primaria para la cual un determinado autor trabaja (y esto ha determinado nuestra segunda regla).

Hay varias maneras de mostrar las cardinalidades en el diagrama. Una de ellas es poner etiquetas en las líneas que unen las relaciones con las entidades.

La etiqueta consiste de un mínimo y un máximo, cada uno de los cuales contiene un cero, un uno o una letra n ("varios"). Si la cardinalidad es exactamente uno, se pone sólo el uno. En el caso de una relación varios a varios, lo usual es poner una m en un extremo y una n en el otro.

2.2.18 PASO DEL MODELO E-R AL MODELO RELACIONAL

Para pasar a tablas todos los datos sin dejarnos nada y que las tablas tengan sentido por si solas tenemos que seguir unos pasos:

Toda entidad se transforma en una tabla

todo atributo se transforma en una columna dentro de la tabla a la que pertenece

El identificador de la entidad se convierte en la clave primaria de la tabla

Toda relación N:M se convierte en una tabla que tendrá como clave primaria las dos claves primarias de las entidades que se asocian

En las relaciones 1:N la clave primaria de la entidad con cardinalidad 1 pasa a la tabla de la entidad cuya cardinalidad es N

en las relaciones N:M existen tres posibilidades: Si la cardinalidad es (0,1) en ambas entidades, se crea tabla. Mientras que si la cardinalidad de una es (0,1) y de la otra es (1,1) se suele pasar la clave primaria de (1,1) a la de (0,1). Si la cardinalidad de ambas es (1,1) se pasa la clave de cualquiera de ellas a la otra.

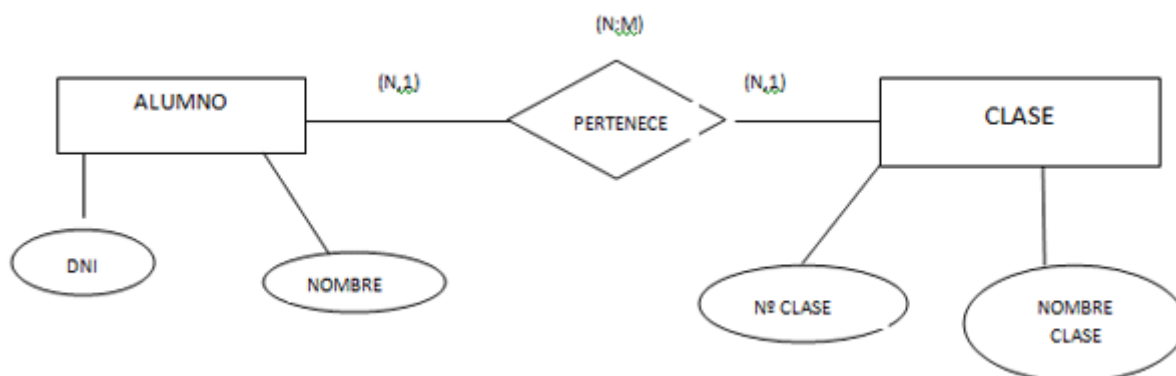
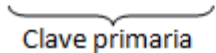


Grafico #2 2.2.18 MODELO ENTIDAD-RELACIÓN

Para este modelo de entidad-relación el paso a tablas quedaría de la siguiente forma¹³:

| | | |
|-----------------|---------------------------|--------------------------|
| Tabla alumno | DNI(clave primaria) | nombre |
| Tabla clase | Nº clase (clave primaria) | Nombre clase |
| Tabla pertenece | DNI (clave foránea) | Nº clase (clave foránea) |



Clave primaria

Tabla #1 2.2.18 MODELO ENTIDAD-RELACIÓN

2.2.19 HISTORIA DE SQL

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en estas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre ella. El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones sobre los mismos.

En la actualidad el SQL es el estándar de la inmensa mayoría de los SGBD comerciales (MySQL, Postgre, DB2, Informix, Microsoft SQL, Access, Oracle, SyBase, etc). Y aunque la diversidad de añadidos particulares que incluyen las distintas implementaciones comerciales del lenguaje es amplia, el soporte al estándar SQL-92 es general y muy amplio.

El SQL es un lenguaje declarativo de “alto nivel” o “de no procedimiento” (como lo son C o Basic) , que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación y la orientación a objetos. De esta forma una sola sentencia puede equivaler a uno o más programas que utilizas en un lenguaje de bajo nivel orientado a registro.

[13]http://html.rincondelvago.com/bases-de-datos_3.html

2.2.19.1 ORÍGENES Y EVOLUCIÓN DE SQL

| Año | Nombre | Alias | Comentarios |
|------|----------|---------|---|
| 1986 | SQL-86 | SQL-87 | Primera publicación hecha por ANSI. Confirmada por ISO en 1987. |
| 1989 | SQL-89 | | Revisión menor. |
| 1992 | SQL-92 | SQL2 | Revisión mayor. |
| 1999 | SQL:1999 | SQL2000 | Se agregaron expresiones regulares, consultas recursivas (para relaciones jerárquicas), triggers y algunas características orientadas a objetos. |
| 2003 | SQL:2003 | | Introduce algunas características de XML, cambios en las funciones, estandarización del objeto sequence y de las columnas autonumericas. (Ver Eisenberg et al.: SQL:2003 Has BeenPublished.) |
| 2006 | SQL:2006 | | ISO/IEC 9075-14:2006 Define las maneras en las cuales el SQL se puede utilizar conjuntamente con XML. Define maneras de importar y guardar datos XML en una base de datos SQL, manipulándolos dentro de la base de datos y publicando el XML y los datos SQL convencionales en forma XML. Además, proporciona facilidades que permiten a las aplicaciones integrar dentro de su código SQL el uso de XQuery, lenguaje de consulta XML publicado por el W3C (World Wide Web Consortium) para acceso concurrente a datos ordinarios SQL y documentos XML. |
| 2008 | SQL:2008 | | Permite el uso de la cláusula ORDER BY fuera de las definiciones de los cursores. Incluye los disparadores del tipo INSTEAD OF. Añade la sentencia TRUNCATE. (Ver [1].) |

Tabla # 1 2.2.19.1 ORÍGENES Y EVOLUCIÓN DE SQL

Los orígenes del SQL están ligados a los de las bases de datos relacionales. En 1970 E. F. Codd propone el modelo relacional y asociado a este un sublenguaje de acceso a los datos basado en el cálculo de predicados. Basándose en estas ideas, los laboratorios de IBM definen el lenguaje SEQUEL (Structured English Query Language) que más tarde sería ampliamente implementado por el sistema de gestión de bases de datos (SGBD) experimental System R, desarrollado en 1977 también por IBM. Sin embargo, fue Oracle quien lo introdujo por primera vez en 1979 en un programa comercial.

El SEQUEL terminaría siendo el predecesor de SQL, siendo este una versión evolucionada del primero. El SQL pasa a ser el lenguaje por excelencia de los diversos sistemas de gestión de bases de datos relacionales surgidos en los años siguientes y es por fin estandarizado en 1986 por el ANSI, dando lugar a la primera versión estándar de este lenguaje, el "SQL-86" o "SQL1". Al año siguiente este estándar es también adoptado por la ISO.

Sin embargo, este primer estándar no cubre todas las necesidades de los desarrolladores e incluye funcionalidades de definición de almacenamiento que se consideraron suprimir. Así que en 1992 se lanza un nuevo estándar ampliado y revisado del SQL llamado "SQL-92" o "SQL2".

En la actualidad el SQL es el estándar *de facto* de la inmensa mayoría de los SGBD comerciales. Y, aunque la diversidad de añadidos particulares que incluyen las distintas implementaciones comerciales del lenguaje es amplia, el soporte al estándar SQL-92 es general y muy amplio.

El ANSI SQL sufrió varias revisiones y agregados a lo largo del tiempo:

2.2.19.2 CARACTERÍSTICAS GENERALES DEL SQL

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

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

OPTIMIZACIÓN

Como ya se dijo arriba, y suele ser común en los lenguajes de acceso a bases de datos de alto nivel, el SQL es un lenguaje declarativo. O sea, que especifica qué es lo que se quiere y no cómo conseguirlo, por lo que una sentencia no establece explícitamente un orden de ejecución.

El orden de ejecución interno de una sentencia puede afectar gravemente a la eficiencia del SGBD, por lo que se hace necesario que éste lleve a cabo una optimización antes de su ejecución. Muchas veces, el uso de índices acelera una instrucción de consulta, pero ralentiza la actualización de los datos. Dependiendo del uso de la aplicación, se priorizará el acceso indexado o una rápida actualización de la información. La optimización difiere sensiblemente en cada motor de base de datos y depende de muchos factores.

Existe una ampliación de SQL conocida como FSQL (Fuzzy SQL, SQL difuso) que permite el acceso a bases de datos difusas, usando la lógica difusa. Este lenguaje ha sido implementado a nivel experimental y está evolucionando rápidamente.

2.2.20 LENGUAJE DE DEFINICIÓN DE DATOS (DDL)

El lenguaje de definición de datos (en inglés Data Definition Language, o DDL), es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Incluye órdenes para modificar, borrar o definir las tablas en las que se almacenan los datos de la base de datos. Existen cuatro operaciones básicas: CREATE, ALTER, DROP y TRUNCATE.

CREATE

Este comando crea un objeto dentro de la base de datos. Puede ser una tabla, `[[Vista 'CAMPO 2' STRING </source>`

Ejemplo (crear una función)

```
CREATE OR REPLACE FUNCTION 'NOMBRE FUNCION'('PARAMETROS')
RETURNS 'TIPO RETORNO' AS
$BODY$
BEGIN
'INSTRUCCIÓN SQL'
--por Ejemplo:
DELETE FROM con empleado WHERE id empleado= 'ANY' (ids);
END;
```

\$BODY\$

```
LANGUAGE 'plpgsql';
```

ALTER

Este comando permite modificar la estructura de un objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc.

Ejemplo (agregar columna a una tabla)

```
ALTER TABLE 'TABLA_NOMBRE' (  
    ADD NUEVO_CAMPO INT UNSIGNED meel  
)
```

DROP

Este comando elimina un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte. Se puede combinar con la sentencia ALTER.

Ejemplo

```
DROP TABLE 'TABLA_NOMBRE'
```

TRUNCATE

Este comando trunca todo el contenido de una tabla. La ventaja sobre el comando DROP, es que si se quiere borrar todo el contenido de la tabla, es mucho más rápido, especialmente si la tabla es muy grande. La desventaja es que TRUNCATE sólo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula WHERE. Si bien, en un principio, esta sentencia parecería ser DML (Lenguaje de Manipulación de Datos), es en realidad una DDL, ya que internamente, el comando TRUNCATE borra la tabla y la vuelve a crear y no ejecuta ninguna transacción.

Ejemplo

```
TRUNCATE TABLE "TABLA_NOMBRE1"
```

2.2.21 LENGUAJE DE MANIPULACIÓN DE DATOS (DML)

Un lenguaje de manipulación de datos (Data Manipulation Language, o DML en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional.

INSERT

Una sentencia *INSERT* de SQL agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional.

Forma básica

```
INSERT INTO "tabla" ("columna1",["columna2,..."]) VALUES ("valor1",["valor2,..."])
```

Las cantidades de columnas y valores deben ser iguales. Si una columna no se especifica, le será asignado el valor por omisión. Los valores especificados (o implícitos) por la sentencia *INSERT* deberán satisfacer todas las restricciones aplicables. Si ocurre un error de sintaxis o si alguna de las restricciones es violada, no se agrega la fila y se devuelve un error.

Ejemplo

```
INSERT INTO agenda_telefonica (nombre,numero) VALUES ('Roberto  
Jeldrez',4886850);
```

Cuando se especifican todos los valores de una tabla, se puede utilizar la sentencia acortada:

```
INSERT INTO "VALUES ("valor1",["valor2,..."])
```

Ejemplo (asumiendo que 'nombre' y 'número' son las únicas columnas de la tabla 'agenda_telefonica'):

```
INSERT INTO agenda_telefonica VALUES ('Roberto Jeldrez', 080473968);
```

Formas avanzadas

Una característica de SQL (desde SQL-92) es el uso de constructores de filas para insertar múltiples filas a la vez, con una sola sentencia SQL:

```
INSERT INTO "tabla" ("columna1",["columna2,... "])  
VALUES ("valor1a",["valor1b,..."]),("value2a",["value2b,..."]),...
```

Esta característica es soportada por DB2, PostgreSQL (desde la versión 8.2), MySQL, y H2.

Ejemplo (asumiendo que 'nombre' y 'número' son las únicas columnas en la tabla 'agenda_telefonica'):

```
INSERT INTO agenda_telefonica VALUES ('Roberto Fernández', '4886850'),('Alejandro Sosa', '4556550');
```

Que podía haber sido realizado por las sentencias

```
INSERT INTO agenda_telefonica VALUES ('Roberto Fernández', '4886850');
```

```
INSERT INTO agenda_telefonica VALUES ('Alejandro Sosa', '4556550');
```

Notar que las sentencias separadas pueden tener semántica diferente (especialmente con respecto a los triggers), y puede tener diferente rendimiento que la sentencia de inserción múltiple.

Para insertar varias filas en MS SQL puede utilizar esa construcción:

```
INSERT INTO phone_book  
SELECT 'John Doe', '555-1212'  
UNION ALL  
SELECT 'Peter Doe', '555-2323';
```

Tenga en cuenta que no se trata de una sentencia SQL válida de acuerdo con el estándar SQL (SQL: 2003), debido a la cláusula subselect incompleta.

Para hacer lo mismo en Oracle se usa DUAL TABLE, siempre que se trate de solo una simple fila:

```
INSERT INTO phone_book  
SELECT 'John Doe', '555-1212' FROM DUAL  
UNION ALL  
SELECT 'Peter Doe', '555-2323' FROM DUAL
```

Una implementación conforme al estándar de esta lógica se muestra el siguiente ejemplo, o como se muestra arriba (no aplica en Oracle):

```
INSERT INTO phone_book  
SELECT 'John Doe', '555-1212' FROM LATERAL ( VALUES(1) ) AS t(c)  
UNION ALL  
SELECT 'Peter Doe', '555-2323' FROM LATERAL ( VALUES(1) ) AS t(c)
```

Copia de filas de otras tablas

Un INSERT también puede utilizarse para recuperar datos de otros, modificarla si es necesario e insertarla directamente en la tabla. Todo esto se hace en una sola sentencia SQL que no implica ningún procesamiento intermedio en la aplicación cliente. Un SUBSELECT se utiliza en lugar de la cláusula VALUES. El

SUBSELECT puede contener JOIN, llamadas a funciones, y puede incluso consultar en la misma TABLA los datos que se inserta. Lógicamente, el SELECT se evalúa antes que la operación INSERT esté iniciada. Un ejemplo se da a continuación.

```
INSERT INTO phone_book2
```

```
SELECT *  
FROM phone_book  
WHERE name IN ('John Doe', 'Peter Doe')
```

Una variación es necesaria cuando algunos de los datos de la tabla fuente se está insertando en la nueva tabla, pero no todo el registro. (O cuando los esquemas de las tablas no son iguales.)

```
INSERT INTO phone_book2 ([name], [phoneNumber])
```

```
SELECT [name], [phoneNumber]  
FROM phone_book  
WHERE name IN ('John Doe', 'Peter Doe')
```

El SELECT produce una tabla (temporal), y el esquema de la tabla temporal debe coincidir con el esquema de la tabla donde los datos son insertados.

UPDATE

Una sentencia UPDATE de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla.

Ejemplo

```
UPDATE My_table SET field1 = 'updated value asd' WHERE field2 = 'N';
```

DELETE

Una sentencia *DELETE* de SQL borra uno o más registros existentes en una tabla.

Forma básica

```
SQL="DELETE FROM "tabla" WHERE "columna1" = "valor1""
```

Ejemplo

```
DELETE FROM My_table WHERE field2 = 'N';
```

Recuperación de clave

Los diseñadores de base de datos que usan una clave suplente como la clave principal para cada tabla, se ejecutará en el ocasional escenario en el que es necesario recuperar automáticamente la base de datos, generando una clave primaria de una sentencia SQL INSERT para su uso en otras sentencias SQL. La mayoría de los sistemas no permiten sentencias SQL INSERT para retornar fila de datos. Por lo tanto, se hace necesario aplicar una solución en tales escenarios.

Implementaciones comunes incluyen:

- Utilizando un procedimiento almacenado específico de base de datos que genera la clave suplente, realice la operación INSERT, y finalmente devuelve la clave generada.
- Utilizando una sentencia SELECT específica de base de datos, sobre una tabla temporal que contiene la última fila insertada. DB2 implementa esta característica de la siguiente manera:

```
SELECT *
```

```
FROM NEW TABLE ( INSERT INTO phone_book VALUES ( 'Cristobal Jeldrez','0426.817.10.30' )) AS t
```

- Utilizando una sentencia SELECT después de la sentencia INSERT con función específica de base de datos, que devuelve la clave primaria generada por el registro insertado más recientemente.
- Utilizando una combinación única de elementos del original SQL INSERT en una posterior sentencia SELECT.
- Utilizando un GUID en la sentencia SQL INSERT y la recupera en una sentencia SELECT.
- Utilizando la función de PHPmysql_insert_id() de MySQL después de la sentencia INSERT.
- Utilizando un INSERT con la cláusula RETURNING para Oracle, que sólo se puede utilizar dentro de un PL/SQL bloque, en el caso de PostgreSQL se puede usar también tanto con SQL como con PL/SQL.

```
INSERT INTO phone_book VALUES ( 'Cristobal Jeldrez','0426.817.10.30' )  
RETURNING phone_book_id INTO v_pb_id
```

- En el caso de MS SQL se puede utilizar la siguiente instrucción:
SET NoCount ON;

```
INSERT INTO phone_book VALUES ( 'Cristobal Jeldrez','0426.817.10.30' );  
SELECT @@IDENTITY AS ID
```

Disparadores

Los disparadores, también conocidos como desencadenantes (*triggers* en inglés) son definidos sobre la tabla en la que opera la sentencia INSERT, los desencadenantes son evaluados en el contexto de la operación. Desencadenantes BEFORE INSERT permiten la modificación de los valores que se insertará en la tabla. Desencadenantes AFTER INSERT no puede modificar los datos de ahora en adelante, pero se puede utilizar para iniciar acciones en otras tablas, por ejemplo para aplicar mecanismos de auditoría.

Sistemas de gestión de base de datos

Los sistemas de gestión de base de datos con soporte SQL más utilizados son, por orden alfabético:

- DB2
- Firebird
- Informix
- Interbase
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL
- PervasiveSQL
- SQLite
- SQL Server
- Sybase ASE¹⁴

[14]<http://es.wikipedia.org/wiki/SQL>

2.2.22 QUE ES SQL

SQL (Lenguaje de Consulta Estructurado) fue comercializado por primera vez en 1981 por IBM, el cual fue presentado a ANSI y desde entonces ha sido considerado como un estándar para las bases de datos relacionales. Desde 1986, el estándar SQL ha aparecido en diferentes versiones como por ejemplo: SQL:92, SQL:99, SQL:2003. MySQL es una idea originaria de la empresa open source MySQL AB establecida inicialmente en Suecia en 1995 y cuyos fundadores son David Axmark, Allan Larsson, y Michael "Monty" Widenius. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad.

Michael "Monty" Widenius en la década de los 90 trató de usar mSQL para conectar las tablas usando rutinas de bajo nivel ISAM, sin embargo, mSQL no era rápido y flexible para sus necesidades. Esto lo llevó a crear una API SQL denominada MySQL para bases de datos muy similar a la de mSQL pero más portable.

El nombre de MySQL procede de la combinación de My, hija del cofundador Michael "Monty" Widenius, con el acrónimo SQL (según la documentación de la última versión en inglés[2]). Por otra parte, el directorio base y muchas de las bibliotecas usadas por los desarrolladores tenían el prefijo My.

El nombre del delfín de MySQL es Sakila y fue seleccionado por los fundadores de MySQL AB en el concurso "NameTheDolphin". Este nombre fue enviado por AmbroseTwebaze, un desarrollador de software de código abierto africano, derivado del idioma SiSwate, el idioma local de Swazilandia y corresponde al nombre de una ciudad en Arusha, Tanzania, cerca de Uganda la ciudad origen de Ambrose¹⁵.

[15]<http://es.wikipedia.org/wiki/SQL>

2.2.23 QUE ES MYSQL

MySQL es un sistema de gestión de bases de datos Brelacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius¹⁶.

2.2.23.1 MOTORES DE ALMACENAMIENTO DE MYSQL Y TIPOS DE TABLAS

MySQL soporta varios motores de almacenamiento que tratan con distintos tipos de tabla. Los motores de almacenamiento de MySQL incluyen algunos que tratan con tablas transaccionales y otros que no lo hacen:

[16]<http://es.wikipedia.org/wiki/MySQL>

2.2.23.2 MYISAM

Es el motor de almacenamiento por defecto. Se basa en el código ISAM pero tiene muchas extensiones útiles. (Tenga en cuenta que MySQL 5.0 no soporta ISAM.)

Cada tabla MyISAM se almacena en disco en tres ficheros. Los ficheros tienen nombres que comienzan con el nombre de tabla y tienen una extensión para indicar el tipo de fichero. Un fichero .frm almacena la definición de tabla. El fichero de datos tiene una extensión .MYD (MYData) . El fichero índice tiene una extensión .MYI (MYIndex) . Para especificar explícitamente que quiere una tabla MyISAM, indíquelo con una opción ENGINE:

```
CREATE TABLE t (i INT) ENGINE = MYISAM;
```

(Nota: Antiguas versiones de MySQL usaban TYPE en lugar de ENGINE (por ejemplo: TYPE = MYISAM). MySQL 5.0 soporta esta sintaxis para compatibilidad con versiones anteriores pero TYPE está obsoleto y ahora se usa ENGINE .)

Normalmente, la opción ENGINE no es necesaria; MyISAM es el motor de almacenamiento por defecto a no ser que se cambie.

2.2.23.2.1 CARACTERÍSTICAS DE MYISAM

- Todos los datos se almacenan con el byte menor primero. Esto hace que sean independientes de la máquina y el sistema operativo. El único requerimiento para portabilidad binaria es que la máquina use enteros con signo en complemento a dos (como todas las máquinas en los últimos 20 años) y formato en coma flotante IEEE (también dominante en todas las máquinas). La única área de máquinas que pueden no soportar compatibilidad binaria son sistemas empotrados, que a veces tienen procesadores peculiares.
- No hay penalización de velocidad al almacenar el byte menor primero; los bytes en un registro de tabla normalmente no están alineados y no es un problema leer un byte no alineado en orden normal o inverso. Además, el código en el servidor que escoge los valores de las columnas no es crítico respecto a otro código en cuanto a velocidad.
- Ficheros grandes (hasta longitud de 63 bits) se soportan en sistemas de ficheros y sistemas operativos que soportan ficheros grandes.
- Registros de tamaño dinámico se fragmentan mucho menos cuando se mezclan borrados con actualizaciones e inserciones. Esto se hace

combinando automáticamente bloques borrados adyacentes y extendiendo bloques si el siguiente bloque se borra.

- El máximo número de índices por tabla MyISAM en MySQL 5.0 es 64. Esto puede cambiarse recompilando. El máximo número de columnas por índice es 16.
- La longitud máxima de clave es 1000 bytes. Esto puede cambiarse recompilando. En caso de clave mayor a 250 bytes, se usa un tamaño de bloque mayor, de 1024 bytes.
- Las columnas BLOB y TEXT pueden indexarse.
- Valores NULL se permiten en columnas indexadas. Esto ocupa 0-1 bytes por clave.
- Todos los valores de clave numérico se almacenan con el byte mayor primero para mejor compresión de índice.
- Cuando se insertan registros en orden (como al usar columnas AUTO_INCREMENT), el árbol índice se divide de forma que el nodo mayor sólo contenga una clave. Esto mejora la utilización de espacio en el árbol índice.
- El tratamiento internode una columna AUTO_INCREMENT portable. MyISAM actualiza automáticamente esta columna para operaciones INSERT y UPDATE.
- Esto hace las columnas AUTO_INCREMENT más rápidas (al menos 10%). Los valores iniciales de la secuencia no se reusan tras ser borrados.
- Si una tabla no tiene bloques libres en medio del fichero de datos, puede INSERT nuevos registros a la vez que otros flujos leen de la tabla. (Esto se conoce como inserciones concurrentes.) Un bloque libre puede ser resultado de borrar o actualizar registros de longitud dinámica con más datos que su contenido. Cuando todos los bloques libres se usan (se rellenan), las inserciones futuras vuelven a ser concurrentes.
- Puede tener el fichero de datos e índice en directorios distintos para obtener más velocidad con las opciones DATA DIRECTORY y INDEX DIRECTORY para CREATE TABLE.
- Cada columna de caracteres puede tener distintos conjuntos de caracteres.

- Hay un flag en el fichero índice MyISAM que indica si la tabla se ha cerrado correctamente.
- Si mysqld se arranca con la opción `--myisam-recover` , las tablas MyISAM se chequean automáticamente al abrirse, y se reparan si la tabla no se cierra correctamente.
- `Myisamchk` marca las tablas como chequeadas si se ejecuta con la opción `--update-state` .`myisamchk --fast` cheque sólo las tablas que no tienen esta marca.
- `Myisamchk --analyze` almacena estadísticas para partes de las claves, así como para las claves enteras.
- `Myisampack` puede comprimir columnas BLOB y VARCHAR.

2.2.23.2 OPCIONES DE ARRANQUE DE MYISAM

Las siguientes opciones de mysqld pueden usarse para cambiar el comportamiento de tablas MyISAM :

`--myisam-recover=mode`

Cambia el modo para recuperación automática para tablas MyISAM .

`--delay-key-write=ALL`

No vuelca buffers de clave entre escrituras para cualquier tabla MyISAM .

Nota: Si hace esto, no debe usar tablas MyISAM desde otro programa (como otro servidor MySQL server o con `myisamchk`) cuando la tabla está en uso. Hacerlo provoca corrupción de índice.

Usar `--external-locking` no ayuda para tablas que usan `--delay-key-write`.

Las siguientes variables de sistema afectan al comportamiento de tablas MyISAM :

`bulk_insert_buffer_size`

Tamaño del árbol de caché usado en optimización de inserciones. *Nota:* Este es el límite *por flujo!*

`myisam_max_extra_sort_file_size`

Usado para ayudar a MySQL a decidir cuándo usar el método de creación de índice de clave caché lento pero seguro *Nota:* Este parámetro se daba en bytes antes de MySQL 5.0.6, cuando se eliminó.

`myisam_max_sort_file_size`

No usa el método de ordenación de índice rápido para crear un índice si el fichero temporal será más grande a este tamaño. *Nota:* En MySQL 5.0, este parámetro se da en bytes.

`myisam_sort_buffer_size`

Cambia el tamaño del búffer usado al recuperar tablas.

La recuperación automática se activa si arranca mysqld con la opción `--myisam-recover`. En ese caso, cuando el servidor abre una tabla MyISAM, chequea si la tabla está marcada como mal cerrada o si el contador de veces que se ha abierto la tabla no es 0 y está ejecutando el servidor con `--skip-external-locking`. Si alguna de estas condiciones es cierta, ocurre lo siguiente:

Se chequea la tabla para errores.

Si el servidor encuentra un error, trata de hacer una reparación de tabla rápida (ordenando y sin recrear el fichero de datos).

Si falla la reparación debido a un error en el fichero de datos (por ejemplo, error de clave duplicada), el servidor lo intenta otra vez, esta vez recreando el fichero de datos.

Si sigue fallando, el servidor trata una vez más con el método de reparación antiguo (escrito registro a registro sin ordenar). Este método debe ser capaz de reparar cualquier clase de error y tiene requerimientos de espacio bajos.

Si la recuperación no fuera capaz de recuperar todos los registros de un comando previamente completado y no ha especificado `FORCE` en la opción `--myisam-recover` la recuperación automática aborta con un mensaje de error en el log de errores:

```
Error: Couldn't repair table: test.g00pages
```

Si especifica `FORCE`, se escribe una advertencia como esta:

```
Warning: Found 344 of 354 rows when repairing ./test/g00pages
```

Tenga en cuenta aquí el valor de recuperación automático incluye `BACKUP`, el proceso de recuperación crea ficheros con nombres de la forma `tbl_name-datetime.BAK`. Debe tener un script cron que mueva estos ficheros automáticamente del directorio de base de datos al dispositivo de copia de seguridad.

Cuánto espacio necesitan las claves

Tablas MyISAM usan índices B-tree. Puede calcular el tamaño del fichero índice mediante $(key_length+4)/0.67$, sumado sobre todas las claves. Este es el peor caso

en que todas las claves se insertan en orden ordenado y la tabla no tienen ninguna clave comprimida.

Los índices de cadenas de caracteres están comprimidos en espacio. Si la primera parte del índice es una cadena de caracteres, también tiene el prefijo comprimido. La compresión de espacio hace que el fichero índice sea menor que el peor caso si la columna de la cadena de caracteres tiene muchos espacios finales o es una columna VARCHAR que no se usa siempre con la longitud total. La compresión de prefijo se usa en claves que comienzan con una cadena de caracteres. La compresión de prefijo ayuda si hay muchas cadenas de caracteres con un prefijo idéntico.

Entablas MyISAM puede comprimir números prefijo:

Especificando `PACK_KEYS=1` cuando crea la tabla. Esto ayuda cuando tiene, muchas claves enteras con un prefijo idéntico cuando los números se almacenan con el byte mayor primero.

En este caso, se puede usar `CHECK TABLE`, aunque puede obtener una advertencia de otros servidores. Sin embargo, `REPAIR TABLE` debe evitarse ya que cuando un servidor reemplaza el fichero de datos con uno nuevo, no se envía a los otros servidores¹⁷.

2.2.23.3 MERGE

El motor de almacenamiento MERGE también conocido como MRG_MyISAM es una colección de tablas MyISAM idénticas que pueden usarse como una "Idéntica" significa que todas las tablas tienen información de columna e índice idéntica. No puede mezclar tablas en que las columnas se listen en orden distinto, no tengan exactamente las mismas columnas, o tengan los índices en orden distinto. Sin embargo, alguna o todas las tablas pueden comprimirse con `myisampack`.

Cuando crea una tabla MERGE MySQL crea dos ficheros en disco. Los ficheros tienen nombres que comienzan con el nombre de la tabla y tienen una extensión para indicar el tipo de fichero, Un fichero.frm almacena la definición de tabla y un fichero.MRG contiene los nombres de las tablas que deben usarse como una. Las tablas no tienen que estar en la misma base de datos que la tabla MERGE misma.

[17]<http://dev.mysql.com/doc/refman/5.0/es/myisam-storage-engine.html>

Puede usar SELECT, DELETE, UPDATE, y INSERT en la colección de tablas, Debe tener permisos de SELECT, UPDATE, y DELETE en las tablas que mapea a una tabla MERGE.

Si hace un DROP de la tabla MERGE sólo borra la especificación MERGE . Las tablas subyacentes no se ven afectadas.

Cuando crea una tabla MERGE debe especificar una cláusula UNION=(*list-of-tables*) que indica qué tablas quiere usar como una. Puede especificar opcionalmente una opción INSERT_METHOD si quiere que las inserciones en la tabla MERGE se realicen en la primera o última tabla de la lista UNION.

El siguiente ejemplo muestra cómo crear una tabla MERGE :

```
mysql> CREATE TABLE t1 (  
  ->a INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  -> message CHAR(20));  
mysql> CREATE TABLE t2 (  
  ->a INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  -> message CHAR(20));  
mysql> INSERT INTO t1 (message) VALUES ('Testing'),('table'),('t1');  
mysql> INSERT INTO t2 (message) VALUES ('Testing'),('table'),('t2');  
mysql> CREATE TABLE total (  
  ->a INT NOT NULL AUTO_INCREMENT,  
  -> message CHAR(20), INDEX(a)  
  -> TYPE=MERGE UNION=(t1,t2) INSERT_METHOD=LAST;
```

Tenga en cuenta que la columna A está indexada en la tabla MERGE pero no está declarada como PRIMARY KEY como lo está en las tablas MyISAM subyacente. Esto es necesario ya que una tabla MERGE no puede forzar unicidad en un conjunto de tablas subyacentes.

Tras crear la tabla MERGE, puede realizar consultas que operen en el grupo de tablas como unidad:

```
mysql> SELECT * FROM total;
```

Tenga en cuenta que puede manipular el fichero .MRG directamente desde fuera del servidor MySQL :

```
shell> cd /mysql-data-directory/current-database
```

```
shell>ls -l t1 t2 >total.MRG
```

```
shell>mysqladminflush-tables
```

Para reparar una tabla MERGE a una colección diferente de tablas MyISAM, puede realizar una de las siguientes opciones:

DROP la tabla MERGE y recrearla.

Use ALTER TABLE *tbl_name* UNION=(...) para cambiar la lista de tablas subyacentes.

Cambie el fichero .MRG y realice un comando FLUSH TABLE para la tabla MERGE y todas las tablas subyacentes para forzar al motor de almacenamiento a leer el nuevo fichero de definición.

Las tablas MERGE pueden ayudarle a arreglar los siguientes problemas:

Administrar fácilmente un largo conjunto de tablas. Por ejemplo, puede poner datos de meses distintos en tablas separadas, comprimir algunos de ellos con myisampack, y luego crear una tabla MERGE para usarlas como una.

Obtener más velocidad. Puede dividir una tabla grande de sólo lectura basándose en algún criterio, y luego poner las tablas individuales en distintos discos. Una tabla MERGE puede ser mucho más rápida que usar una tabla grande.

Realizar búsquedas más eficientes. Si conoce exactamente lo que busca, puede buscar en sólo una de las tablas divididas y usar una tabla MERGE para las otras. Puede tener distintas tablas MERGE que usen conjuntos no disjuntos de tablas.

Realizar reparaciones más eficientes. Es más fácil de reparar tablas individuales mapeadas en una tabla MERGE que reparar una única tabla grande.

Mapear instantáneamente varias tablas como una. Una tabla MERGE no necesita mantener un índice propio ya que usa los índices de las tablas individuales. Como resultado las colecciones tablas MERGE son muy rápidas de crear o remapear.

(Tenga en cuenta que debe especificar las definiciones de índices cuando crea una tabla MERGE incluso cuando no se crean índices.)

Si tiene un conjunto de tablas que une como una tabla grande bajo demanda o batch, debe crear una tabla MERGE sobre ellas bajo demanda. Esto es mucho más rápido y ahorra mucho espacio de disco.

Excede el límite de tamaño del sistema operativo. Cada tabla MyISAM está ligada a este límite, pero una colección de tablas MyISAM no lo está.

2.2.23.3.1 DESVENTAJAS DE LAS TABLAS MERGE

Sólo puede usar tablas MyISAM idénticas para una tabla MERGE.

No puede usar un número de características MyISAM en tablas MERGE . Por ejemplo, no puede crear índices FULLTEXT en tablas MERGE . (Puede crear índices FULLTEXT en las tablas MyISAM subyacentes, pero no puede buscar en la tabla MERGE con búsquedas full-text.)

Si la tabla MERGE es temporal, las tablas MyISAM pueden ser cualquier mezcla de tablas temporales y no temporales.

Las tablas MERGE usan más descriptores de fichero. Si 10 clientes usan una tabla MERGE que mapee a 10 tablas, el servidor usa $(10 \times 10) + 10$ descriptores de fichero. (10 descriptores de ficheros de datos para cada uno de los 10 clientes, y 10 descriptores de ficheros de índice compartidos entre los clientes.)

Las lecturas de claves son más lentas. Cuando lee una clave, el motor de almacenamiento MERGE necesita leer en todas las tablas subyacentes para chequear cuál se parece más a la clave dada. Si luego lee el siguiente, el motor MERGE necesita buscar en los buffers de lectura para buscar la siguiente clave. Sólo cuando se usa un búffer de claves el motor necesita leer el siguiente bloque de claves.

Esto hace que las claves MERGE sean mucho más lentas en búsquedas eq_ref pero no mucho más lentas en búsquedas ref.

El orden de índices en la tabla MERGE y sus tablas subyacentes debe ser el mismo. Si usa ALTER TABLE para añadir un índice UNIQUE a una tabla usada en una tabla MERGE y después usa ALTER TABLE para añadir un índice no único en la tabla MERGE, la ordenación de índice es distinta para las tablas si ya hay un índice no único en la tabla subyacente. (Esto es porque ALTER TABLE pone los índices UNIQUE antes de los índices no únicos para facilitar detección rápida de claves duplicadas.) Consecuentemente, las consultas en tablas con tales índices pueden retornar resultados no esperados¹⁸.

[18]<http://dev.mysql.com/doc/refman/5.0/es/merge-storage-engine.html>

2.2.23.4 MEMORY (HEAP)

El motor de almacenamiento MEMORY crea tablas con contenidos que se almacenan en memoria. Éstas se conocían previamente como HEAP. En MySQL 5.0, MEMORY es el término preferido, aunque HEAP se soporta para compatibilidad con versiones anteriores.

Cada tabla MEMORY está asociada con un fichero de disco. El nombre de fichero comienza con el nombre de la tabla y tiene una extensión de frm para indicar que almacena la definición de la tabla.

Para especificar explícitamente que quiere una tabla MEMORY, indíquelo con una opción ENGINE:

```
CREATE TABLE t (i INT) ENGINE = MEMORY;
```

Como indica su nombre, las tablas MEMORY se almacenan en memoria y usan índices hash por defecto. Esto las hace muy rápidas, y muy útiles para crear tablas temporales. Sin embargo, cuando se apaga el servidor, todos los datos almacenados en las tablas MEMORY se pierden. Las tablas por sí mismas continúan existiendo ya que sus definiciones se almacenan en ficheros .frm en disco, pero están vacías cuando reinicia el servidor.

Este ejemplo muestra cómo puede crear, usar , y borrar una tabla MEMORY :

```
mysql> CREATE TABLE test ENGINE=MEMORY
-> SELECT ip,SUM(downloads) AS down
-> FROM log_table GROUP BY ip;
mysql> SELECT COUNT(ip),AVG(down) FROM test;
mysql> DROP TABLE test;
```

2.2.23.4.1 CARACTERÍSTICAS DE MEMORY

El espacio para tablas MEMORY se reserva en pequeños bloques. Las tablas usan el 100% del hashing dinámico para inserciones. No se necesita área de desbordamiento o espacio extra para claves. No se necesita espacio extra para listas libres. Los registros borrados se ponen en una lista encadenada y se rehúsan cuando inserta nuevos datos en la tabla. Las tablas MEMORY no tienen ninguno de los problemas asociados con borrados más inserciones en tablas hashadas.

Las tablas MEMORY pueden tener hasta 32 índices por tabla, 16 columnas por índice y una clave de longitud máxima de 500 bytes.

En MySQL 5.0, el motor MEMORY implementa índices HASH y BTREE . Puede especificar uno u otro para cada índice añadiendo una cláusula USING tal y como se muestra:

```
CREATE TABLE lookup  
  (id INT, INDEX USING HASH (id))  
  ENGINE = MEMORY;
```

```
CREATE TABLE lookup  
  (id INT, INDEX USING BTREE (id))  
  ENGINE = MEMORY;
```

Las características generales de B-trees e índices hash se describen en “Cómo utiliza MySQL los índices”.

Puede tener claves no únicas en una tabla MEMORY. (Esta es una característica no común de implementaciones de índices hash.)

En MySQL 5.0, puede usar INSERT DELAYED con tablas MEMORY ..

Si tiene un índice hash en una tabla MEMORY que tenga un alto índice de duplicación de claves (muchas entradas de índice con el mismo valor), las actualizaciones a la tabla que afecten valores claves y todos los borrados son significativamente más lentos. El rango de esta ralentización es proporcional al rango de duplicación (o inversamente proporcional al grado cardinalidad). Puede usar un índice BTREE para evitar este problema.

Las tablas MEMORY usan una longitud de registro fija.

MEMORY no soporta columnas BLOB o TEXT .

MEMORY en MySQL 5.0 incluye soporte para columnas AUTO_INCREMENT e índices en columnas que contengan valores NULL .Las tablas MEMORY se comparten entre todos los clientes (como cualquier otra tabla no-TEMPORARY).

Los contenidos de las tablas MEMORY se almacenan en memoria , lo que es una propiedad que las tablas MEMORY comparten con las tablas internas que el servidor va creando al procesar consultas. Sin embargo, los dos tipos de tablas difieren en que las tablas MEMORY no están sujetas a conversión de almacenamiento, mientras que las tablas internas sí:

Si una tabla interna llega a ser demasiado grande, el servidor la convierte automáticamente a una tabla en disco. El límite de tamaño lo determina la variable de sistema tmp_table_size .

Las tablas MEMORY nunca se convierten en tablas de disco. Para asegurar que no comete un error accidentalmente, puede cambiar la variable de sistema `max_heap_table_size` para que imponga un tamaño máximo de tablas MEMORY. Para tablas individuales, puede especificar la opción de tabla `MAX_ROWS` en el comando `CREATE TABLE`.

El servidor necesita suficiente memoria para mantener todas las tablas MEMORY en uso a la vez.

Para liberar memoria usada por una tabla MEMORY cuando no se requiere su contenido, debe ejecutar `DELETE FROM` o `TRUNCATE TABLE`, o borrar toda la tabla con `DROP TABLE`.

Si quiere rellenar una tabla MEMORY cuando arranca el servidor MySQL, puede usar la opción `--init-file`. Por ejemplo, puede usar comandos como `INSERT INTO SELECT` o `LOAD DATA INFILE` en este fichero para cargar la tabla de una fuente de datos persistente.

Si usa replicación, las tablas MEMORY del servidor maestro se vacían cuando se apaga y reinicia. Sin embargo, un esclavo no es consciente que se vacían estas tablas, así que retorna contenido desfasado si selecciona datos del mismo. En MySQL 5.0, cuando se usa una tabla MEMORY en el maestro por primera vez desde que arrancó el maestro, se escribe un comando `DELETE FROM` en el log binario del maestro automáticamente, re sincronizando el maestro y el esclavo otra vez. Tenga en cuenta que incluso con esta estrategia, el esclavo tiene datos desfasados en la tabla en el intervalo entre el reinicio del maestro y el primer uso de la tabla. Sin embargo, si usa la opción `--init-file` para rellenar la tabla MEMORY al arrancar el maestro, se asegura que este intervalo sea cero.

La memoria necesaria por un registro en una tabla MEMORY se calcula con la siguiente expresión:

```
SUM_OVER_ALL_BTREE_KEYS(max_length_of_key + sizeof(char*) * 4)
+ SUM_OVER_ALL_HASH_KEYS(sizeof(char*) * 2)
+ ALIGN(length_of_row+1, sizeof(char*))
```

`ALIGN()` representa un factor de redondeo para que la longitud del registro sea un múltiplo exacto del tamaño del puntero `char`. `sizeof(char*)` es 4 en máquinas de 32-bit y 8 en máquinas de 64-bit¹⁹.

[19]<http://dev.mysql.com/doc/refman/5.0/es/memory-storage-engine.html>

2.2.23.5 BDB (BERKELEYDB)

Sleep y cat Software ha proporcionado a MySQL el motor de almacenamiento transaccional Berkeley DB. A este motor de almacenamiento se le llama tradicionalmente BDB. Se incluye soporte para BDB en distribuciones fuentes de MySQL y en distribuciones binarias MySQL-Max

Las tablas BDB pueden tener una gran probabilidad de sobrevivir a fallos del sistema y ser capaces de realizar COMMIT y ROLLBACK en transacciones. La distribución fuente MySQL incluye una distribución BDB preparada para funcionar con MySQL. No puede usar una versión de BDB no preparada para funcionar con MySQL. En MySQL AB trabajamos codo a codo con Sleep y cat para mantener la calidad de la interfaz MySQL/BDB. (Incluso aunque Berkeley DB está muy testeado y es muy fiable, la interfaz MySQL se considera de calidad gamma. Continuamos mejorando y optimizándola.)

Cuando hay algún problema con tablas BDB, ayudamos a nuestros usuarios a localizar el problema y reproducir casos de test. Cualquiera de estos casos de test se envía a Sleep y cat, que nos ayuda a encontrar y arreglar el problema. Con esta operación de dos fases, cualquier problema con tablas BDB puede tardar un poco más en ser resuelto que con nuestros motores de almacenamiento. Sin embargo, no anticipamos dificultades significativas con este procedimiento ya que el código Berkeley DB se usa en muchas aplicaciones distintas a MySQL.

2.2.23.5.1 SISTEMAS OPERATIVOS QUE SOPORTA BDB

Actualmente, sabemos que el motor de almacenamiento BDB funciona con los siguientes sistemas operativos:

- Linux 2.x Intel
- Sun Solaris (SPARC y x86)
- FreeBSD 4.x/5.x (x86, sparc64)
- IBM AIX 4.3.x
- SCO OpenServer
- SCO UnixWare 7.1.x
- Windows NT/2000/XP

BDB *no* funciona con los siguientes sistemas operativos:

- Linux 2.x Alpha
- Linux 2.x AMD64
- Linux 2.x IA-64
- Linux 2.x s390

Nota: La lista precedente no está completa. La actualizamos cuando tenemos más información.

Si compila MySQL de las fuentes con soporte para tablas BDB , pero ocurre el siguiente error cuando arranca mysqld, significa que BDB no está soportado por su arquitectura:

```
bdb: architecture lacks fast mutexes: applications cannot be threaded  
Can't init data bases
```

En este caso, debe recompilar MySQL sin soporte para tablas BDB o arrancar el servidor con la opción `--skip-bdb` .

2.2.23.5.2 INSTALACIÓN DE BDB

Si ha bajado una versión binaria de MySQL que incluya soporte para Berkeley DB, simplemente siga las instrucciones de instalación usuales. (Distribuciones MySQL-Max incluyen soporte BDB)

Si compila MySQL de las fuentes, puede activar:

Soporte BDB ejecutando configure con la opción `--with-berkeley-db` además de cualquier otra distribución que use normalmente. Descargue una distribución MySQL5.0 cambie la localización en su directorio más alto, y ejecute este comando:

```
shell> ./configure --with-berkeley-db [other-options]
```

2.2.23.5.3 OPCIONES DE ARRANQUE DE BDB

Las siguientes opciones de mysql pueden usarse para cambiar el comportamiento del motor de almacenamiento BDB:

`--bdb-home=path`

Directorio base para tablas BDB . Debe ser el mismo directorio que use para `--datadir`.

`--bdb-lock-detect=method`

El método de bloqueo BDB. El valor debe ser DEFAULT, OLDEST, RANDOM, o YOUNGEST.

`--bdb-logdir=path`

El directorio del fichero de log BDB.

`--bdb-no-recover`

No arranca BerkeleyDB en modo recuperación

No vuelca síncronamente los logs BDB. Esta opción está obsoleta; use `--skip-sync-bdb-logs` en su lugar (consulte la descripción de `--sync-bdb-logs`).

`--bdb-shared-data`

Arranca Berkeley DB en modo multi proceso. (No use `DB_PRIVATE` al inicializar Berkeley DB.)

`--bdb-tmpdir=path`

El directorio de ficheros temporales de BDB.

`--skip-bdb`

Desactiva el motor de almacenamiento BDB.

`--sync-bdb-logs`

Vuelca síncronamente los logs BDB. Esta opción está activada por defecto; use `--skip-sync-bdb-logs` para desactivarla.

Si usa la opción `--skip-bdb`, MySQL no inicializa la biblioteca Berkeley DB library y esto ahorra mucha memoria. Sin embargo, si usa esta opción, no puede usar tablas BDB. Si trata de crear una tabla BDB, MySQL crea una tabla MyISAM en su lugar.

Normalmente, debe arrancar `mysqld` sin la opción `--bdb-no-recover` si quiere usar tablas BDB. Sin embargo, esto puede causar problemas cuando trata de arrancar `mysqld` si los ficheros de log de BDB están corruptos.

Con la variable `bdb_max_lock`, puede especificar el máximo número de bloqueos que pueden estar activos en una tabla BDB. Por defecto son 10,000. Debe incrementar esto si ocurren errores como el siguiente cuando realiza transacciones largas o cuando `mysql` tiene que examinar muchos registros para ejecutar una consulta:

```
bdb: Lock table is out of available locks
```

```
Got error 12 from
```

Puede cambiar las variables `binlog_cache_size` y `max_binlog_cache_size` si usa transacciones de varios comandos muy largas.

2.2.23.5.4 CARACTERÍSTICAS DE LAS TABLAS BDB

Cada tabla BDB se almacena en disco en dos ficheros. Los ficheros tienen nombres que comienzan con el nombre de la tabla y tienen una extensión para indicar el tipo de fichero. Un fichero frm almacena la definición de tabla, y un fichero db contiene los datos de tabla e índices.

Para especificar explícitamente que quiere una tabla BDB, indíquelo con una opción de tabla ENGINE o TYPE:

- CREATE TABLE t (i INT) ENGINE = BDB;
- CREATE TABLE t (i INT) TYPE = BDB;

BerkeleyDB es sinónimo de BDB en la opción ENGINE o TYPE .

El motor de almacenamiento BDB proporciona tablas transaccionales. La forma de usar estas tablas depende del modo autocommit:

Si está ejecutando con autocommit activado (por defecto), los cambios en las tablas BDB se efectúan inmediatamente y no pueden deshacerse.

Si está ejecutando con autocommit desactivado, los cambios no son permanentes hasta que ejecuta un comando COMMIT . En lugar de hacer un commit puede ejecutar ROLLBACK para olvidar los cambios.

Puede comenzar una transacción con el comando BEGIN WORK para suspender autocommit, o con SET AUTOCOMMIT=0 para desactivar autocommit explícitamente.

“Sintaxis de START TRANSACTION, COMMIT y ROLLBACK”.²⁰

2.2.23.6 EXAMPLE

El motor de almacenamiento EXAMPLE es un motor de pruebas que no hace nada. Su propósito es servir como ejemplo en el código fuente MySQL para ilustrar cómo empezar a escribir nuevos motores de almacenamiento. Como tal, tiene interés principalmente para desarrolladores.

Para examinar la fuente del motor EXAMPLE consulte el directorio sql/examples de distribuciones fuentes MySQL 5.0.

Para activar este motor de almacenamiento, use la opción --with-example-storage-engine con configure al compilar MySQL.

[20]<http://dev.mysql.com/doc/refman/5.0/es/bdb-storage-engine.html>

Cuando crea una tabla EXAMPLE, el servidor crea un fichero de definición de tabla en el directorio de base de datos. El fichero comienza con el nombre de tabla y tiene una extensión .frm . No se crean más ficheros. No puede almacenarse ni recuperarse datos de la tabla.

```
mysql> CREATE TABLE test (i INT) ENGINE = EXAMPLE;
```

```
Query OK, 0 rows affected (0.78 sec)
```

```
mysql> INSERT INTO test VALUES(1),(2),(3);
```

```
ERROR 1031 (HY000): Table storage engine for 'test' doesn't have this option
```

```
mysql> SELECT * FROM test;
```

```
Empty set (0.31 sec)
```

El motor EXAMPLE no soporta indexación²¹.

2.2.23.7 FEDERATED

El motor FEDERATED está disponible desde MySQL 5.0.3. Es un motor que accede a datos en tablas de bases de datos remotas en lugar de tablas locales.

Para examinar la fuente para el motor FEDERATED, consulte el directorio sql de una distribución fuente de MySQL 5.0.3 o posterior.

Instalación del motor de almacenamiento FEDERATED

Para activar este motor, use la opción

--with-federated-storage-engine con configure al compilar MySQL.

2.2.23.7.1 DESCRIPCIÓN DEL MOTOR DE ALMACENAMIENTO

FEDERATED

Cuando crea una tabla FEDERATED, el servidor crea un fichero de definición de tabla en el directorio de base de datos. El fichero comienza con el nombre de tabla y tiene extensión .frm . No se crean más ficheros, ya que los datos reales están en la base de datos remota. Esto difiere de cómo funcionan los motores con tablas locales.

Para tablas de bases de datos locales, los ficheros de datos son locales. Por ejemplo, si crea una tabla MyISAM llamada users, el handler MyISAM crea un fichero de datos llamado users.MYD. Un handler para tablas locales lee, inserta, borra y actualiza datos en ficheros de datos locales, y los registros se guardan en un formato particular del handler.

[21]<http://dev.mysql.com/doc/refman/5.0/es/EXAMPLE-storage-engine.html>

Para leer registros, el handler debe pasar los datos en columnas. Para escribir registros, los valores de la columna deben convertirse al formato de registro usado por el handler y escribirse en el fichero de datos local.

Con el motor MySQL FEDERATED no hay ficheros de datos locales para una tabla (por ejemplo, no hay fichero .MYD). En su lugar, una base de datos remota almacena los datos que normalmente estarían en la tabla. Esto necesita el uso de la API del cliente MySQL para leer, borrar, actualizar e insertar datos. La recuperación de datos se inicia via un comando `SELECT * FROM tbl_name`. Para leer el resultado, los registros se tratan uno a uno usando la función de la API C `mysql_fetch_row()` y luego se convierten desde las columnas del conjunto de resultados `SELECT` al formato que el handler FEDERATED espera.

El flujo básico es el siguiente:

- Llamadas SQL efectuadas localmente
- API del handlerMySQL (datos en formato del handler)
- API del cliente MySQL (datos convertidos a llamadas SQL)
- Base de datos remota -> API del cliente MySQL
- Convierte el conjunto de resultados al formato del handler
- API del handler -> registros resultado o conteo de registros afectados a local

2.2.23.7.2 CÓMO USAR LAS TABLAS FEDERATED

El procedimiento para usar tablas FEDERATED es muy simple. Normalmente, tiene dos servidores en ejecución, en la misma máquina o en distintas. (También es posible para una tabla FEDERATED usar otra tabla administrada por el mismo servidor, aunque no tiene mucho sentido.)

Primero, tiene que tener una tabla en el servidor remoto que quiera acceder con la tabla FEDERATED. Suponga que la tabla remota está en la base de datos federated y se define así:

```
CREATE TABLE test_table (  
  idint(20) NOT NULL auto_increment,  
  namevarchar(32) NOT NULL default "",  
  other int(20) NOT NULL default '0',  
  PRIMARY KEY (id),
```

```
KEY name (name),  
KEY other_key (other)  
)
```

```
ENGINE=MyISAM
```

```
DEFAULT CHARSET=latin1;
```

La opción de tabla ENGINE puede nombrar cualquier motor de almacenamiento. La tabla no tiene porqué ser MyISAM.

A continuación, cree una tabla FEDERATED en el servidor local para acceder a la tabla remota:

```
CREATE TABLE federated_table (  
idint(20) NOT NULL auto_increment,  
namevarchar(32) NOT NULL default "",  
other int(20) NOT NULL default '0',  
PRIMARY KEY (id),  
KEY name (name),  
KEY other_key (other)  
)
```

```
ENGINE=FEDERATED
```

```
DEFAULT CHARSET=latin1
```

```
COMMENT='mysql://root@remote_host:9306/federated/test_table';
```

La estructura de esta tabla debe ser exactamente la misma que la de la tabla remota, excepto que la opción de tabla ENGINE debe ser FEDERATED y la opción de tabla COMMENT es una cadena de conexión que indica al motor FEDERATED cómo conectar al servidor remoto.

El motor FEDERATED crea sólo el fichero test_table.frm en la base de datos federated.

La información del equipo remoto indica el servidor remoto al que se conecta el servidor local, y la información de base de datos y tabla indica la tabla remota a usar como fichero de datos. En este ejemplo, el servidor remoto está indicado para ser remote_host corriendo en el puerto 9306, así que puede arrancar ese servidor para que escuche en el puerto 9306.

La forma general de la cadena de conexión en la opción COMMENT es la siguiente:

```
scheme://user_name[:password]@host_name[:port_num]/db_name/tbl_name
```


Sólo se soporta mysql como *scheme* en este punto; la contraseña y el número de puerto son opcionales.

Aquí hay algunas cadenas de conexión de ejemplo:

```
COMMENT='mysql://username:password@hostname:port/database/tablename'
```

```
COMMENT='mysql://username@hostname/database/tablename'
```

```
COMMENT='mysql://username:password@hostname/database/tablename'
```

El uso de COMMENT para especificar la cadena de conexión no es óptima y posiblemente cambiará en MySQL 5.1. Téngalo en cuenta al usar tablas FEDERATED , ya que significa que habrá que hacer modificaciones cuando ocurra.

Como cualquier contraseña usada se almacena en la cadena de conexión como texto plano, puede ser leído por cualquier usuario que pueda usar SHOW CREATE TABLE o SHOW TABLE STATUS para la tabla FEDERATED , o consultar la tabla TABLES en la base de datos INFORMATION_SCHEMA .

Limitaciones del motor de almacenamiento FEDERATED

Lo que el motor FEDERATED soporta y lo que no:

En la primera versión, el servidor remoto debe ser un servidor MySQL. El soporte de FEDERATED para otros motores de bases de datos se añadirá en el futuro.

La tabla remota a la que apunta una tabla FEDERATED *debe* existir antes de intentar acceder a ella.

Es posible para una tabla FEDERATED apuntar a otra, pero debe tener cuidado de no crear un bucle.

No hay soporte para transacciones.

No hay forma que el motor FEDERATED sepa si la tabla remota ha cambiado. La razón es que la tabla debe funcionar como un fichero de datos que nunca se escribirá para hacer algo distinto a la base de datos. La integridad de los datos en la tabla local debe comprobarse si hay algún cambio en la base de datos remota.

El motor FEDERATED soporta SELECT, INSERT, UPDATE, DELETE, e índices.

No soporta ALTER TABLE, DROP TABLE, o cualquier otro comando Data Definition Language. La implementación actual no usa comandos preparados.

La implementación usa SELECT, INSERT, UPDATE, y DELETE, pero no HANDLE.

Las tablas FEDERATED no funcionan con la caché de consultas²².

[22]<http://dev.mysql.com/doc/refman/5.0/es/federated-storage-engine.htm>

2.2.23.8 ARCHIVE

El motor de almacenamiento ARCHIVE se usa para guardar grandes cantidades de datos sin índices con una huella relativamente pequeña.

Para activar este motor, use la opción `--with-archive-storage-engine` configure al Compilar MySQL.

Cuando crea una tabla ARCHIVE, el servidor crea un fichero de definición de tabla en el directorio de base de datos. El fichero comienza con el nombre de tabla y tiene una extensión de `.ARZ` y `.ARM`. Un fichero `.ARN` puede aparecer durante operaciones de optimización.

El motor ARCHIVE soporta sólo INSERT y SELECT. Esto significa que no puede ejecutar DELETEREPLACE, o update. Un SELECT realiza un escaneo de tabla completo. Los registros se comprimen al insertarse. OPTIMIZE TABLE puede usarse para comprimir la tabla.

El motor ARCHIVE usa bloqueo a nivel de registro²³.

2.2.23.9 CSV

El motor de almacenamiento CSV almacena datos en ficheros de texto usando valores separados por comas.

Para activar este motor de almacenamiento, use la opción `--with-csv-storage-engine` configure al compilar MySQL.

Cuando crea una tabla CSV, el servidor crea un fichero de definición de tabla en el directorio de base de datos. El fichero comienza con el nombre de tabla y tienen una extensión `.frm`. El motor de almacenamiento crea un fichero de datos. Su nombre comienza con el nombre de tabla y tiene extensión `.CSV`. El fichero de datos es un fichero de texto. Cuando almacena datos en la tabla, el motor la guarda en el fichero de datos en formato CVS.

[23]<http://dev.mysql.com/doc/refman/5.0/es/archive-storage-engine.html>

```
mysql> CREATE TABLE test(i INT, c CHAR(10)) ENGINE = CSV;
Query OK, 0 rows affected (0.12 sec)
mysql> INSERT INTO test VALUES(1,'record one'),(2,'record two');
Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT * FROM test;
2 rows in set (0.00 sec)
```

Si examina el fichero test.CSV en el directorio de base de datos creado al ejecutar los comandos precedentes, su contenido debe ser como este:

```
"1","record one"
"2","record two"
```

El motor de almacenamiento CSV no soporta indexación²⁴.

2.2.24 QUE ES WAMP SERVER

Es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Windows, como sistema operativo;
- Apache, como servidor web;
- MySQL, como gestor de bases de datos;
- PHP (generalmente) Perl, o Python, como lenguaje.

El uso de un WAMP permite servir páginas html a internet, además de poder gestionar datos en ellas, al mismo tiempo un WAMP, proporciona lenguajes de programación para desarrollar aplicaciones web.

LAMP es el sistema análogo que corre bajo ambiente Linux

WAMP es el sistema análogo que corre bajo ambiente Windows

MAMP es el sistema análogo que corre bajo ambiente Macintosh²⁵.

[24]<http://dev.mysql.com/doc/refman/5.0/es/csv-storage-engine.html>

[25]<http://es.wikipedia.org/wiki/WAMP>

2.2.25 INSTALACIÓN DEL WAMPSEVER2

WAMP es un sistema que se instala en un solo paquete (Apache, PHP y MySQL, además de un gestor de bases de datos como php MyAdmin) que será todo lo que necesitemos para tener nuestro servidor local, en el que realizar nuestras pruebas, o desarrollar nuestros scripts, con la comodidad de no tener que subirlos a un servidor web online.

Una vez descargado lo ejecutamos y veremos la siguiente pantalla:



GRAFICO #1 2.2.25 Instalación Del Wampserver2

Pulsamos sobre next y nos saldrá una ventana con la licencia: Aceptamos la licencia y volvemos a pulsar sobre siguiente y nos saldrá una ventana donde nos preguntará donde queremos instalar wamp:



GRAFICO #2 2.2.25 Instalación Del Wampserver2

Elegimos la ruta en la que queremos instalar wamp, yo lo dejo tal y como viene por defecto en c:\wamp y pulsamos sobre next de nuevo

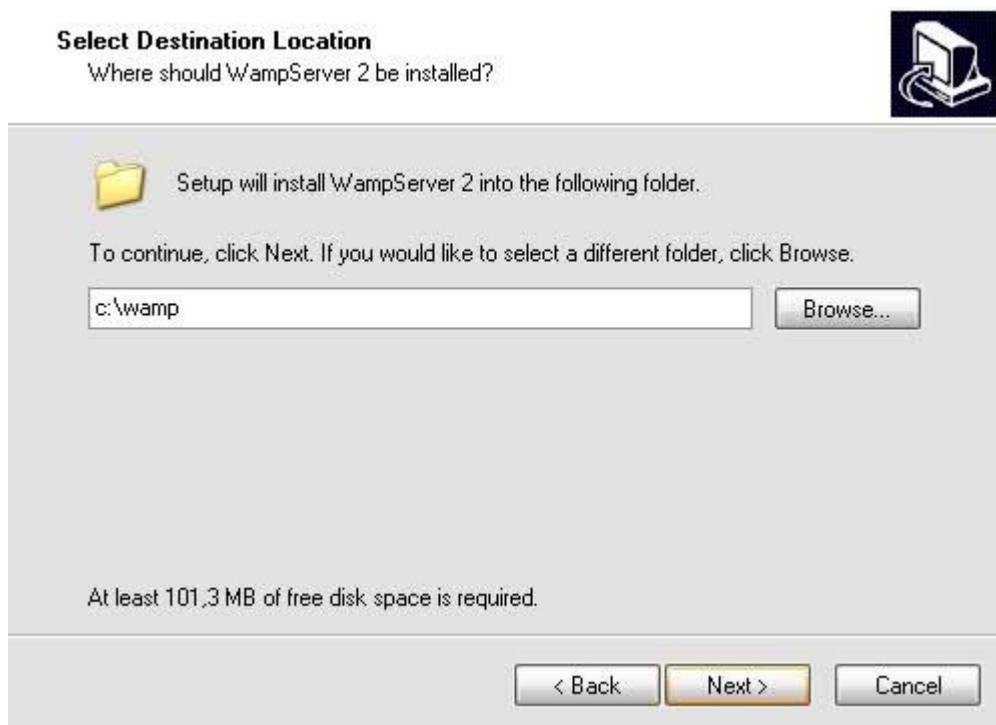


GRAFICO #3 2.2.25 Instalación Del Wampserver2

A continuación nos preguntará si queremos crear un acceso de inicio rápido en el menú de inicio (que es el que está marcado en la captura) y un acceso en el escritorio, como vemos en la siguiente imagen:

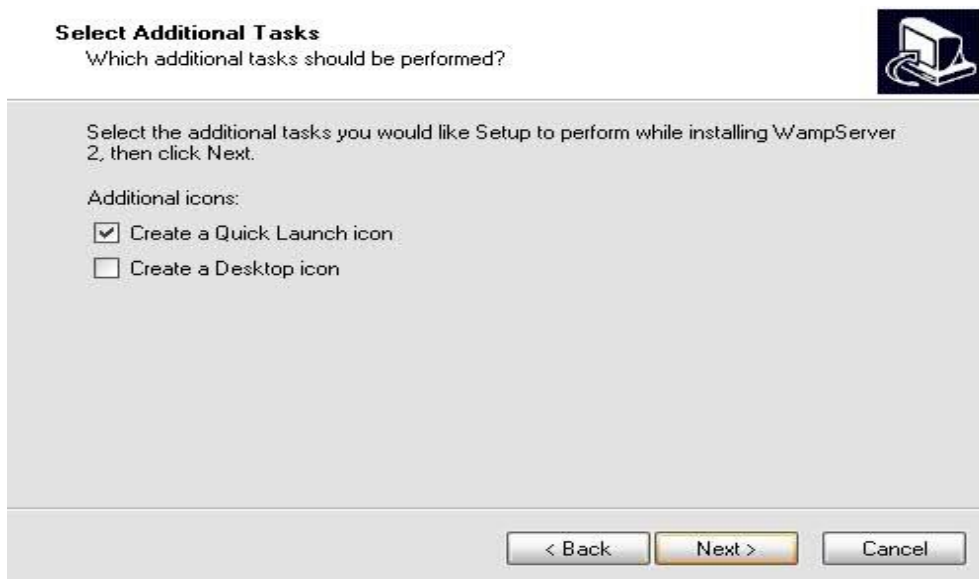


GRAFICO #42.2.25 Instalación Del Wampserver2

Escogemos los accesos que queramos y pulsamos sobre next, nos saldrá una ventana que nos pregunta si queremos instalar.

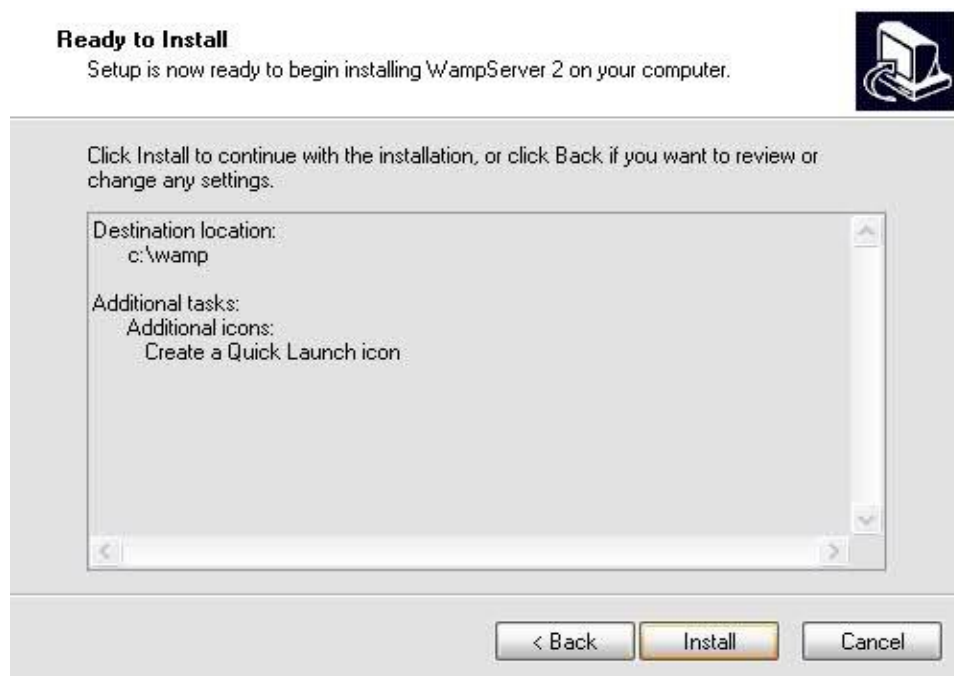


GRAFICO #5 2.2.25 Instalación Del Wampserver2

Pulsamos sobre install y en la siguiente ventana lo podemos dejar como está, a no ser que tengamos un servidor local de correo instalado.

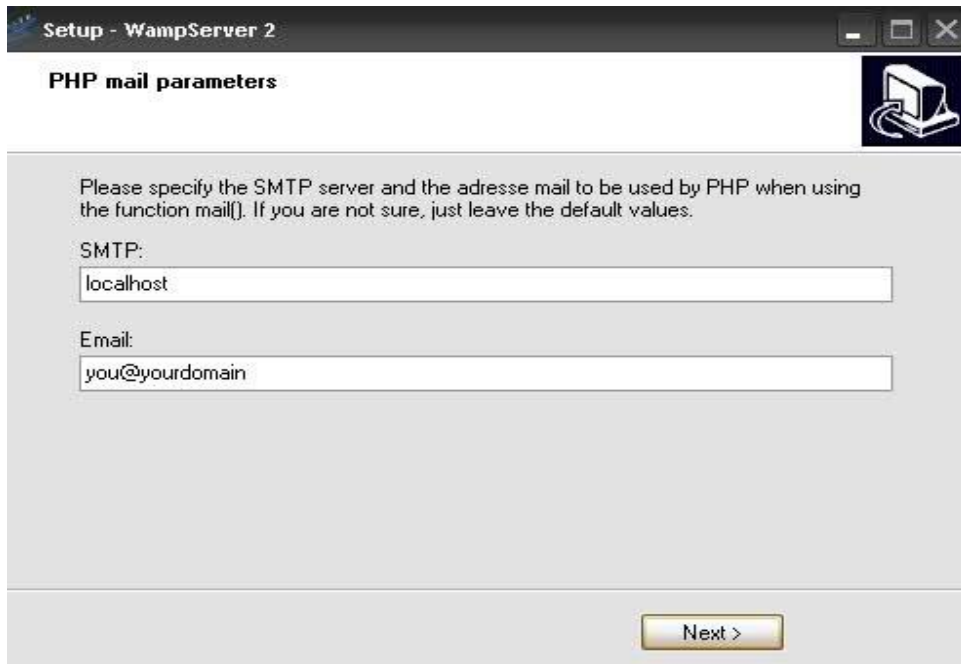


GRAFICO #6 2.2.25 Instalación Del Wampserver2

Volvemos a pulsar sobre next y ya está completa la instalación...al final nos pregunta si queremos ejecutar ahora wamp



GRAFICO #7 2.2.25 Instalación Del Wampserver2

Si queremos lo ejecutamos y se iniciaran los servicios. Es importante ejecutarlo antes de usarlo, sino no cargará el servidor, si no lo queremos ejecutar ahora podremos hacerlo desde los accesos anteriormente creados.

Todos los proyectos debemos colocarlos en la carpeta c:\wamp\www y a los cuales podremos acceder con el navegador desde <http://localhost/> o <http://127.0.0.1/> y para acceder a gestionar nuestras bases de datos accederemos desde <http://localhost/phpmyadmin>.

2.2.26 QUÉ SON LAS PÁGINAS 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 Java Script, 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 contenidos²⁶.

2.2.27 HIPOTESIS Y VARIABLES

2.2.27.1 HIPOTESIS

Con el desarrollo de un sistema se podrá tener un control más detallado de todos los materiales que salen e ingresan a la bodega del Ministerio de Salud Pública del cantón Babahoyo.

2.2.27.2 VARIABLES

Variable Independiente: Calculo de materiales e insumos

Variables Dependiente: Sistema de Control

[26]<http://google.dirson.com/posicionamiento.net/paginas-dinamica>

CAPÍTULO III

3. MARCO METODOLOGICO

Durante el transcurso de este trabajo se aplicó el método científico para el desarrollo del conocimiento en relación al objeto de estudio.

En la cual he realizado una búsqueda detallada en cada uno de los departamentos que conforman el Ministerio de Salud, para de esta manera poder llegar al fondo de la problemática dentro de esta institución pública y por ende llegar a una solución tanto para la entidad como para los usuarios.

Además se empleara los métodos cualitativos y cuantitativos porque se partirá de descripciones detalladas de situaciones, para buscar soluciones y se utilizaran las técnicas experimentales aleatorias, cuasi-experimentales, test “objetivos” de lápiz y papel, estudios de muestra, etc.

3.1 MODALIDAD DE LA INVESTIGACION

Se utilizaran 2 tipos de métodos:

Método Cualitativo.-Esta investigación trata de comprender, explicar, argumentar el objeto de estudio, considerando su contexto histórico, tecnológico y socioeconómico. Se aplicó directamente esta modalidad para evaluar aspectos de comportamiento humano, es decir se investigó los procesos decisionales de la Institución y su influencia en el proceso administrativo.

Método Cuantitativo.-Esta modalidad de investigación trata de describir, contextualizar o explicar con técnicas estadísticas el objeto de estudio. Parte de las técnicas inductivas (razonamientos que se desarrollan de los casos particulares hasta la generalización). En mi caso las muestras han sido representativas con relación al Universo y han permitido la evaluación del proceso.

3.2 TIPOS DE INVESTIGACIÓN

Los tipos de investigación a utilizar son:

Bibliográfica.- Se realizó en base a libros, revistas e internet, permite fundamentar científicamente las variables del presente trabajo de tesis y se utiliza para elaborar el marco teórico.

De Campo.- En cambio con este tipo de investigación se ha podido valorar la problemática relacionada con la toma de decisiones y ratificarla, luego en base a esto nos orientaremos a la propuesta de solución.

Descriptiva.- Como su nombre mismo lo dice aquí se recogen los datos en base a una hipótesis o resultado a fin de extraer generalizaciones significativas que contribuyan al conocimiento. Es decir estudiamos la información de manera cuidadosa para de esta manera mejorar lo conocido.

3.3 POBLACIÓN Y MUESTRA

N= Tamaño de la Muestra

Z= Valor de Confianza

P= Población

$$N = \frac{Z \cdot P}{(P-1)(Z^2/2^2) + Z}$$

| | |
|----------|-------------|
| N | 65 |
| Z | 0.05 |
| P | 340 |

TABLA # 1 3.3 POBLACIÓN Y MUESTRA

$$N = \frac{0.05 \cdot 340}{((340-1)((0.05)^2 / (2)^2) + 0.05)}$$

$$N = \frac{17}{(339(0.0025/4) + 0.05)}$$

$$N = \frac{17}{(339(0.000625) + 0.05)}$$

17

$$N = \frac{17}{(0.211875 + 0.05)}$$

17

$$N = \frac{17}{0.261875}$$

N= 64.91646778

N=65

3.4 MÉTODOS, TÉCNICAS E INSTRUMENTOS

MÉTODOS

MÉTODO CIENTIFICO

Durante el transcurso de este trabajo se aplicó el método científico para el desarrollo del conocimiento en relación al objeto de estudio. En la cual he realizado una búsqueda detallada en cada uno de los departamentos que conforman el Ministerio de Salud, para de esta manera poder llegar al fondo de la problemática dentro de esta institución pública y por ende llegar a una solución tanto para la entidad como para los usuarios.

Además se emplearon los métodos cualitativos y cuantitativos porque se partirá de descripciones detalladas de situaciones, para buscar soluciones y se utilizarán las técnicas experimentales aleatorias, cuasi-experimentales, test “objetivos” de lápiz y papel, estudios de muestra, etc.

TECNICAS

Dentro de las técnicas, e instrumentos se necesitara recopilar información para luego clasificarla, agruparla y finalmente presentarla.

ENCUESTA

Como técnica de obtención de información he utilizado la encuesta ya que es una manera más directa y cómoda para saber las necesidades de los usuarios que laboran en el Ministerio de Salud Pública de Babahoyo en Bodega.

ENTREVISTA

Al realizar las entrevistas a los usuarios podremos analizar las necesidades y el tipo de trabajo que se realiza en el Ministerio de Salud Pública De Babahoyo en Bodega. Las preguntas que se aplicaran serán Abiertas y cerradas.

INSTRUMENTO

Mediante un cuestionario de preguntas que se realizaran de forma interna se podrá establecer cuáles son las fallas en el Ministerio de Salud Pública De Babahoyo en Bodega.

INSTRUMENTO (ENCUESTAS)

1.- ¿Anteriormente ha existido algún tipo de control de inventario de materiales e insumos?

Sí

No

2.- ¿Cree usted que es necesario la implementación de un sistema para la gestión de inventarios de los materiales de oficina e insumos médicos?

Sí

No

3.- ¿Considera usted que los pedidos de materiales e insumos médicos por parte de la institución son generados a tiempo de tal forma que la bodega este abastecida de ellos?

Sí

No

4.- ¿Usted cree que todos los pedidos de materiales e insumos médicos consten de un historial?

Sí

No

5.- ¿Considera usted importante la capacitación a las personas que vayan a utilizar el sistema?

Sí

No

6.- ¿Al implementar una aplicación web las entregas de materiales e insumos médicos se agilizaran?

Sí

No

7.- ¿Está de acuerdo que al momento de realizar rápidamente los procesos, el personal de las diferentes áreas estarán más motivados?

Sí

No

8.- ¿Usted cree primordial que la institución maneje un proceso de inventario para obtener información relacionada con los materiales e insumos médicos?

Sí

No

9.- ¿En el departamento de bodega se guarda la información de manera manual?

Sí

No

10.- ¿Usted estaría dispuesto a capacitarse para optimizar el manejo del sistema para la gestión del inventario de la institución?

Sí

No

3.5 TABULACION DE RESULTADOS

Con un total de mi población de 340 y una muestra de 65

1.- ¿Anteriormente ha existido algún tipo de control de materiales e insumos médicos?

| | Muestra | Porcentaje |
|--------------|---------|------------|
| Si | 0 | 0% |
| No | 65 | 100% |
| Total | 65 | 100% |
| | | |

TABLA # 1 3.5 TABULACION DE RESULTADOS

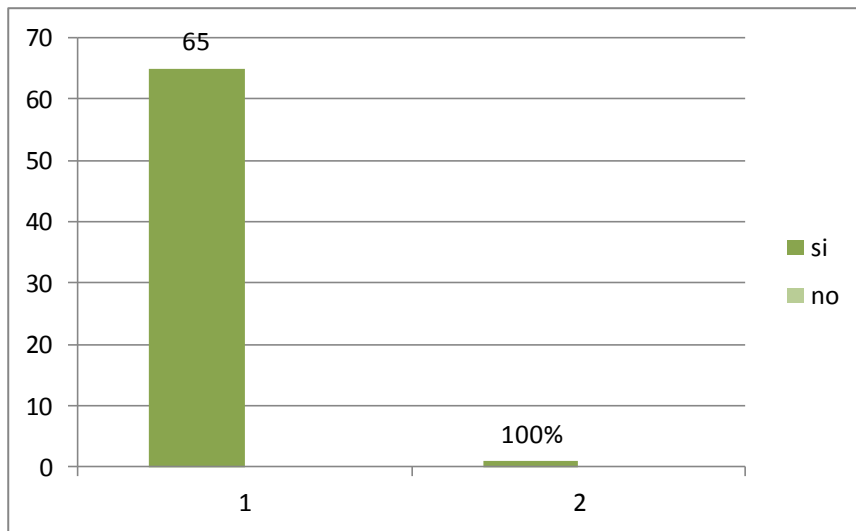


GRAFICO # 1 3.5 TABULACION DE RESULTADOS

Como podemos observar anteriormente en la institución no ha existido un control adecuado de los materiales de oficina e insumos médicos, por tales motivos la implementación de un control de inventarios sería de mucha ayuda en la entidad.

2.- ¿Cree usted que es necesario la implementación de un sistema de control de materiales de oficina e insumos médicos?

| | Muestra | Porcentaje |
|--------------|---------|------------|
| Si | 50 | 77% |
| No | 15 | 23% |
| Total | 65 | 100% |

TABLA # 2 3.5 TABULACION DE RESULTADOS

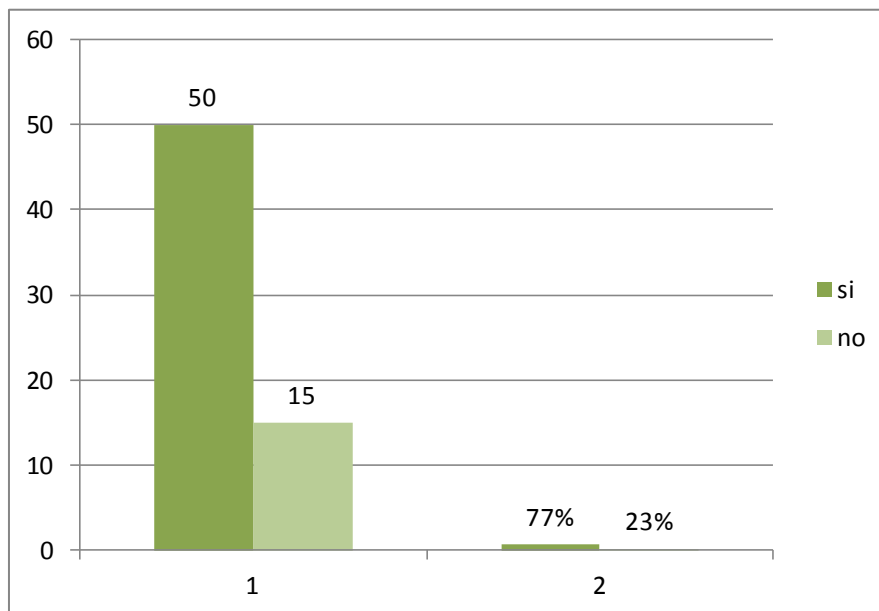


GRAFICO # 2 3.5 TABULACION DE RESULTADOS

Con respecto a los encuestados la mayor parte, considera como una alternativa que existiese un control, donde se le facilite al usuario hacer su solicitud de pedido directamente al departamento de bodega, pero el otro grupo menor, considera que no es necesario, quizás porque ellos no necesita un contacto directo o están satisfechos con la eficiencia.

3.- ¿Considera usted que las decisiones sobre pedidos de materiales e insumos médicos por parte de la institución sean las necesarias?

| | Muestra | Porcentaje |
|--------------|---------|------------|
| Si | 40 | 62% |
| No | 25 | 38% |
| Total | 65 | 100% |

TABLA # 3 3.5 TABULACION DE RESULTADOS

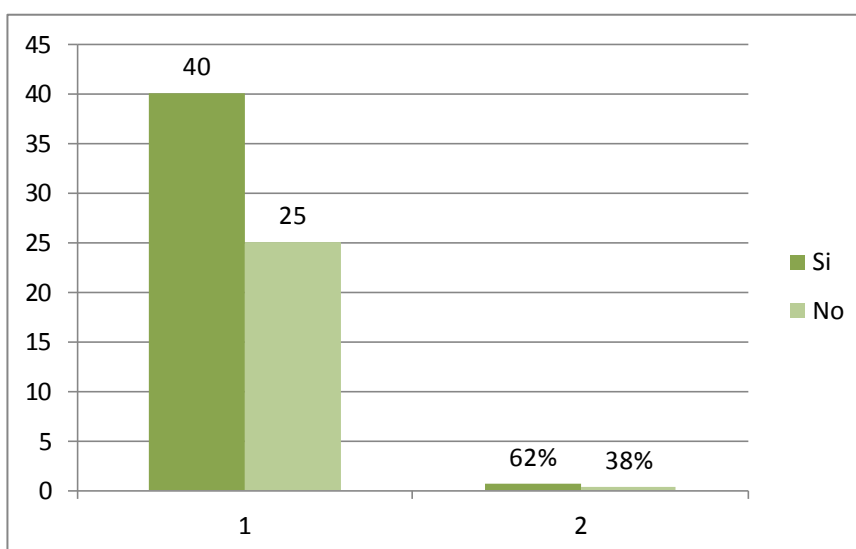


GRAFICO # 3 3.5 TABULACION DE RESULTADOS

El 62% de los encuestados dijo que los pedidos se realizan bien y con anticipación, pero el 38% de los encuestados considero que la institución muchas veces al momento de realizar los pedidos no son los necesarios ya que al momento de que algún usuario va a pedir un insumo o material muchas veces no lo tienen, esto se da a la falta de control existente en la misma.

4.- ¿Usted cree que todos los pedidos de materiales e insumos médicos consten de un historial?

| | Muestra | Porcentaje |
|--------------|---------|------------|
| Si | 35 | 54% |
| No | 30 | 46% |
| Total | 65 | 100% |

TABLA # 4 3.5 TABULACION DE RESULTADOS

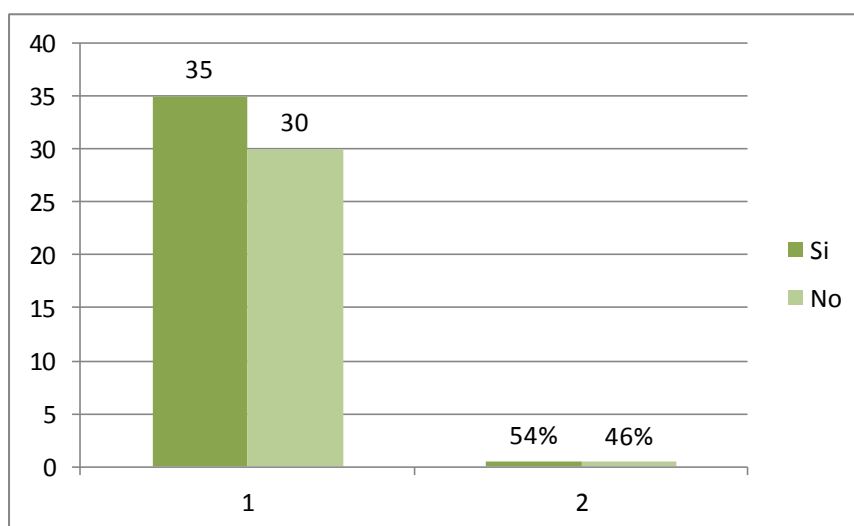


GRAFICO # 4 3.5 TABULACION DE RESULTADOS

Como se observa el 54% de las personas encuestadas dijo que dentro del departamento de bodega, en la forma manual se lleva un historial de todos los pedidos que entran y salen de la misma pero el 46% opinó lo contrario ya que muchas veces no se sabe que se ha pedido más durante el mes o año.

5.- ¿Considera usted importante la capacitación a las personas que vayan a utilizar el sistema?

| | Muestra | Porcentaje |
|--------------|---------|------------|
| Si | 65 | 100% |
| No | 0 | 0% |
| Total | 65 | 100% |

TABLA # 5 3.5 TABULACION DE RESULTADOS

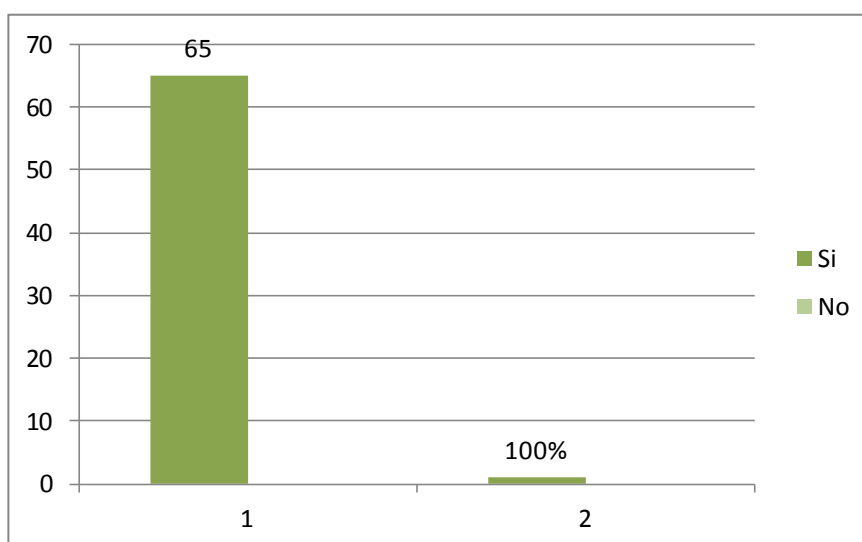


GRAFICO # 5 3.5 TABULACION DE RESULTADOS

Con el resultado que se obtuvo podemos decir que el personal de la institución está totalmente de acuerdo a que la persona o las personas que vayan a manejar el sistema se tengan que capacitar para que de esta manera lograr resultados óptimos al momento de realizar los pedidos de materiales e insumos médicos.

6.- ¿Al implementar una aplicación web las entregas de materiales e insumos médicos se agilizaran?

| | Muestra | Porcentaje |
|--------------|---------|------------|
| Si | 55 | 85% |
| No | 10 | 15% |
| Total | 65 | 100% |

TABLA # 6 3.5 TABULACION DE RESULTADOS

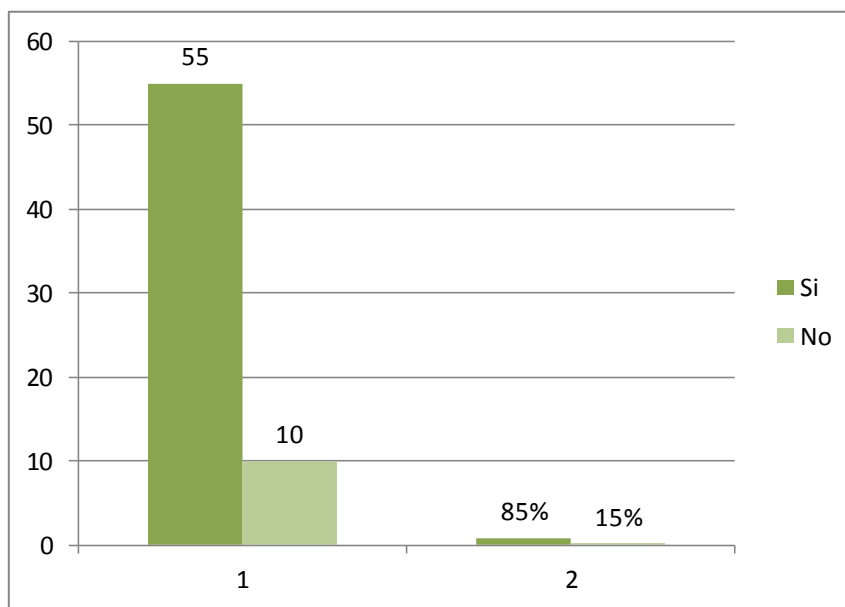


GRAFICO # 6 3.5 TABULACION DE RESULTADOS

Como nos damos cuenta casi la mayoría de las personas encuestadas están de acuerdo que de implementarse una aplicación las tareas se le agilizarían y muy aparte de eso ellos se sentirían más cómodos en sus diferentes departamentos, solo un 15% de los encuestados dijeron que no, debido a que ellos se sienten mucho mejor al hacerlo de manera manual.

7.- ¿Está de acuerdo que al momento de realizar rápidamente los procesos, el personal de las diferentes áreas estarán más motivados?

| | Muestra | Porcentaje |
|--------------|---------|------------|
| Si | 65 | 100% |
| No | 0 | 0% |
| Total | 65 | 100% |

TABLA # 7 3.5 TABULACION DE RESULTADOS

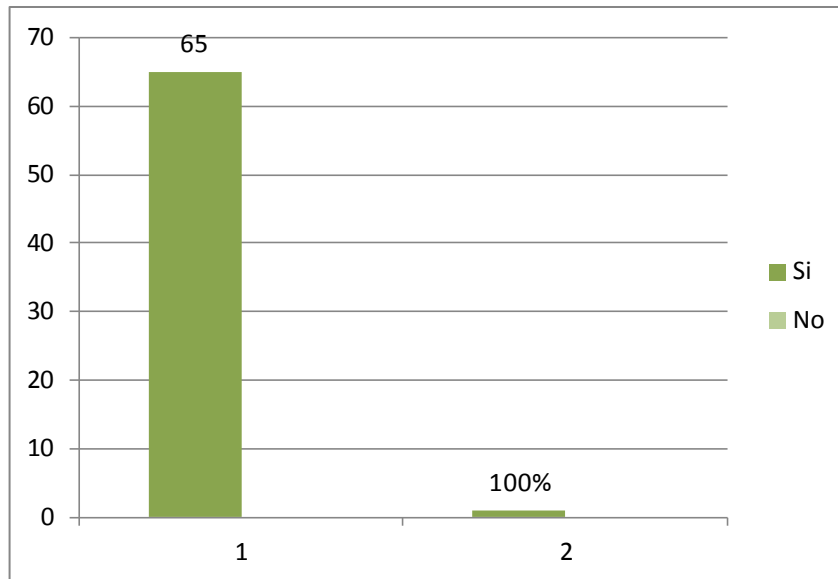


GRAFICO # 7 3.5 TABULACION DE RESULTADOS

Como vemos, todos los encuestados dijeron que si, ya que si los procesos se realizan de una forma rápida ellos no tendrían ningún problema al momento realizar los pedidos ya sean estos de materiales de oficina o insumos médicos.

8.- ¿Usted cree primordial que la institución maneje un proceso de inventario para obtener información relacionada con los materiales e insumos médicos?

| | Muestra | Porcentaje |
|--------------|---------|------------|
| Si | 33 | 51% |
| No | 32 | 49% |
| Total | 65 | 100% |

TABLA # 8 3.5 TABULACION DE RESULTADOS

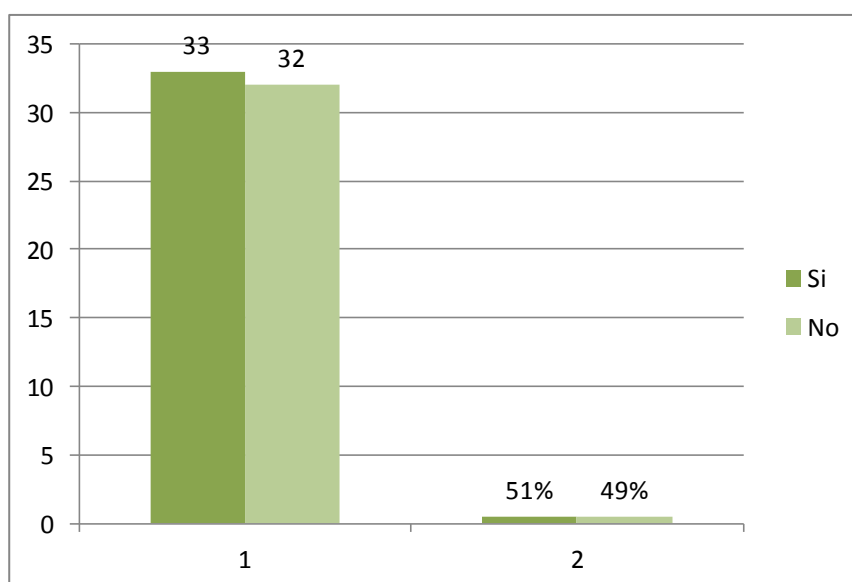


GRAFICO # 8 3.5 TABULACION DE RESULTADOS

De acuerdo al análisis realizado no todos los encuestados están de acuerdo con la implementación de un control de inventarios, para ellos sería mucho más fácil manejar la información de una manera manual ya que por medio de algún sistema se les complicaría la situación, unos por falta de conocimientos en computación otros porque sencillamente no les gusta.

9.- ¿En el departamento de bodega se guarda la información de manera manual?

| | Muestra | Porcentaje |
|--------------|---------|------------|
| Si | 40 | 62% |
| No | 25 | 38% |
| Total | 65 | 100% |

TABLA # 9 3.5 TABULACION DE RESULTADOS

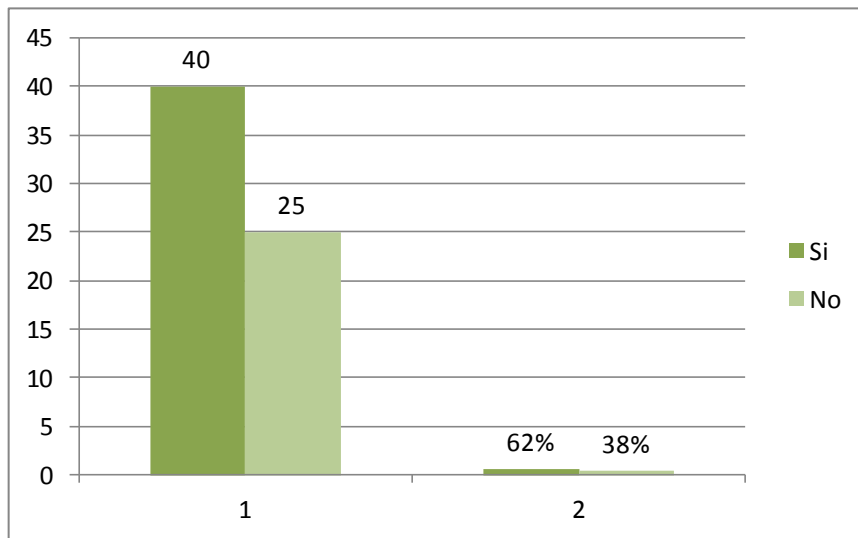


GRAFICO # 9 3.5 TABULACION DE RESULTADOS

En el análisis que se realizó, se concluyó en que el grupo menor dice que la información no se la guarda de manera manual. Pero con respecto al 62% de los encuestados nos dimos cuenta que dicho departamento aún no cuenta con la suficiente tecnología para almacenar sus pedidos de una manera actualizada.

10.- ¿Usted estaría dispuesto a capacitarse para optimizar el manejo del sistema para la gestión del inventario de la institución?

| | Muestra | Porcentaje |
|--------------|---------|------------|
| Si | 5 | 8% |
| No | 60 | 92% |
| Total | 65 | 100% |

TABLA # 10 3.5 TABULACION DE RESULTADOS

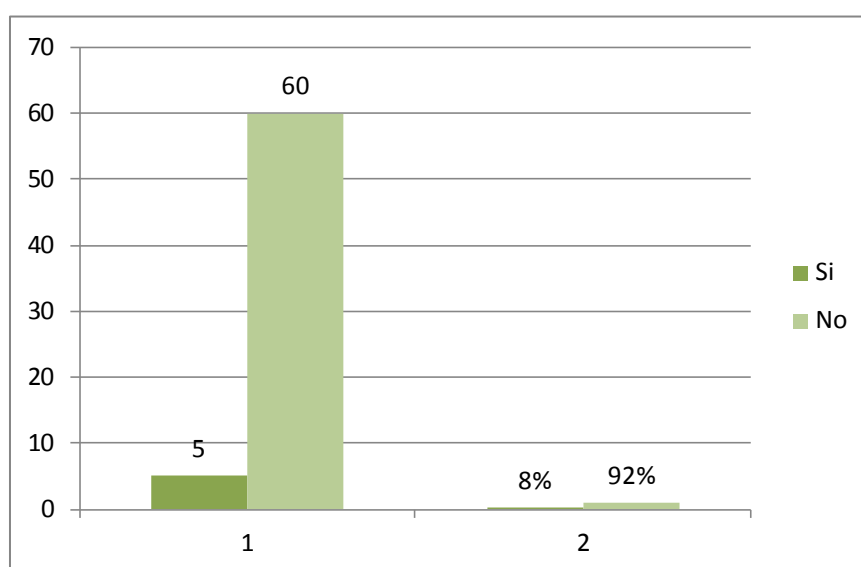


GRAFICO # 10 3.5 TABULACION DE RESULTADOS

El 92% de los encuestados no les gustaría capacitarse, mientras que a una minoría si por lo que observamos que por estos motivos se da el problema de que los pedidos no lleven un historial ya sean estos insumos o materiales de oficina.

3.6 CONCLUSIONES

De este estudio concluyo que el departamento de bodega no tiene un control adecuado de los pedidos que a ellos les hacen por lo que afecta al momento de realizar la adquisición de los materiales e insumos médicos, debido a que no tienen un historial de lo que manejan.

Este departamento aún no cuenta con la suficiente tecnología para almacenar sus pedidos de una manera actualizada, aunque trabajen de una manera manual, pero no es suficiente para brindar un buen servicio a cada uno de los usuarios y departamentos.

Con respecto a todo lo antes mencionado se llega a la conclusión de que este departamento necesita un sistema Informático, donde se les facilite a los departamentos hacer su solicitud de pedido directamente al departamento de bodega y llevar un control de datos actualizado ya no manual, estos y otros beneficios obtendría este departamento con dicho sistema.

3.7 RECOMENDACIONES

De este estudio e investigación, podemos recomendar que:

El Ministerio de Salud Pública de la Provincial de Los Ríos debería comprar o mandar a desarrollar un sistema para mejorar los procesos de pedidos de materiales de oficina e insumos médicos dentro del departamento de bodega, es decir donde los usuarios de diferentes departamentos tenga comunicación directa con el o los encargados de bodega, donde se puedan hacer los pedidos y además de poder llevar un registro de bodega, es decir el ingreso y egreso de los materiales e insumos.

Se le recomienda al administrador del departamento de bodega que capacite a la persona o a las personas que utilizara el sistema para que pueda manejarlo y de esta manera ellos puedan brindar un mejor servicio a cada departamento.

CAPITULO IV

4. DESARROLLO TÉCNICO DE LA INVESTIGACION

4.1 INTRODUCCION

Este Sistema permitirá apoyar el control de los movimientos y distribución de los materiales de oficina e insumos médicos, este software es especialmente construido para instalaciones de todo tipo de entornos de trabajo permitiendo el ahorro de recursos.

4.2 OBJETIVO DE LA PROPUESTA

4.2.1 OBJETIVO GENERAL

Realizar una Aplicación que permita mejorar el manejo de los Materiales de Oficina e Insumos Médicos en la Bodega del Ministerio de Salud Pública de la Provincia de los Ríos.

4.2.2 OBJETIVOS ESPECIFICOS

- Automatizar las operaciones manuales y la confiabilidad de los pedidos en la entrega de materiales de oficina e insumos médicos.
- Diseñar un sistema con un ambiente de trabajo amigable, de tal manera que cualquier persona con conocimientos básicos de computación pueda manejarlo.
- Probar que la aplicación tenga aceptación de los usuarios, para que en cualquier momento pueda adicionársele módulos que permitan el control de otros departamentos.

4.3 METODOLOGIA DE DESARROLLO UTILIZADA

El software está desarrollado en tecnología de vanguardia Cliente Servidor utilizando metodología de análisis y técnicas de desarrollo web

Esta aplicación tendrá una cómoda interacción entre el programa y su operador, por las adaptaciones que cuenta el programa y un diseño de pantalla amigable proporcionara un fácil manejo del software para los usuarios.

Este programa cuenta con una estructura de desarrollo pensado en expansiones futuras, de tal manera que en cualquier momento puede adicionársele nuevos módulos que permitan el control de otras Áreas o Departamentos.

El programa cuenta con gestión absoluta de la Seguridad de usuarios del sistema, proporciona un control de accesos a diferentes niveles para grupos de usuarios y garantizara la seguridad de la información del sistema.

4.4 ANALISIS PREVIO

4.5 DISEÑO

4.5.1 LISTADO DE REQUERIMIENTOS Y FUNCIONES QUE TENDRÁ EL SOFTWARE

REQUERIMIENTOS DEL SOFTWARE

- MYSQL
- WAMP SERVER
- SISTEMA OPERATIVO QUE SOPORTE APACHE

FUNCIONES DEL SOFTWARE

- Movimientos
- Inventario Inicial
- Movimiento de Ingreso
- Movimiento de Egreso

Consultas

- Inventario Físico
- Consulta de Movimiento por Departamento
- Movimiento de Productos
- Certificado de Existencia
- Reporte de Productos
- Listado de Proveedores y Productos
- Reporte de productos por Categoría
- Reporte de Kardex

Funciones de Mantenición

Mantenición de Clasificación Artículos

Mantenición de Entradas

Mantenición de Proveedores

Mantenición de Áreas

Mantenición de Empleados (Internos y Externos)

Control de Usuarios

Control de Periodos

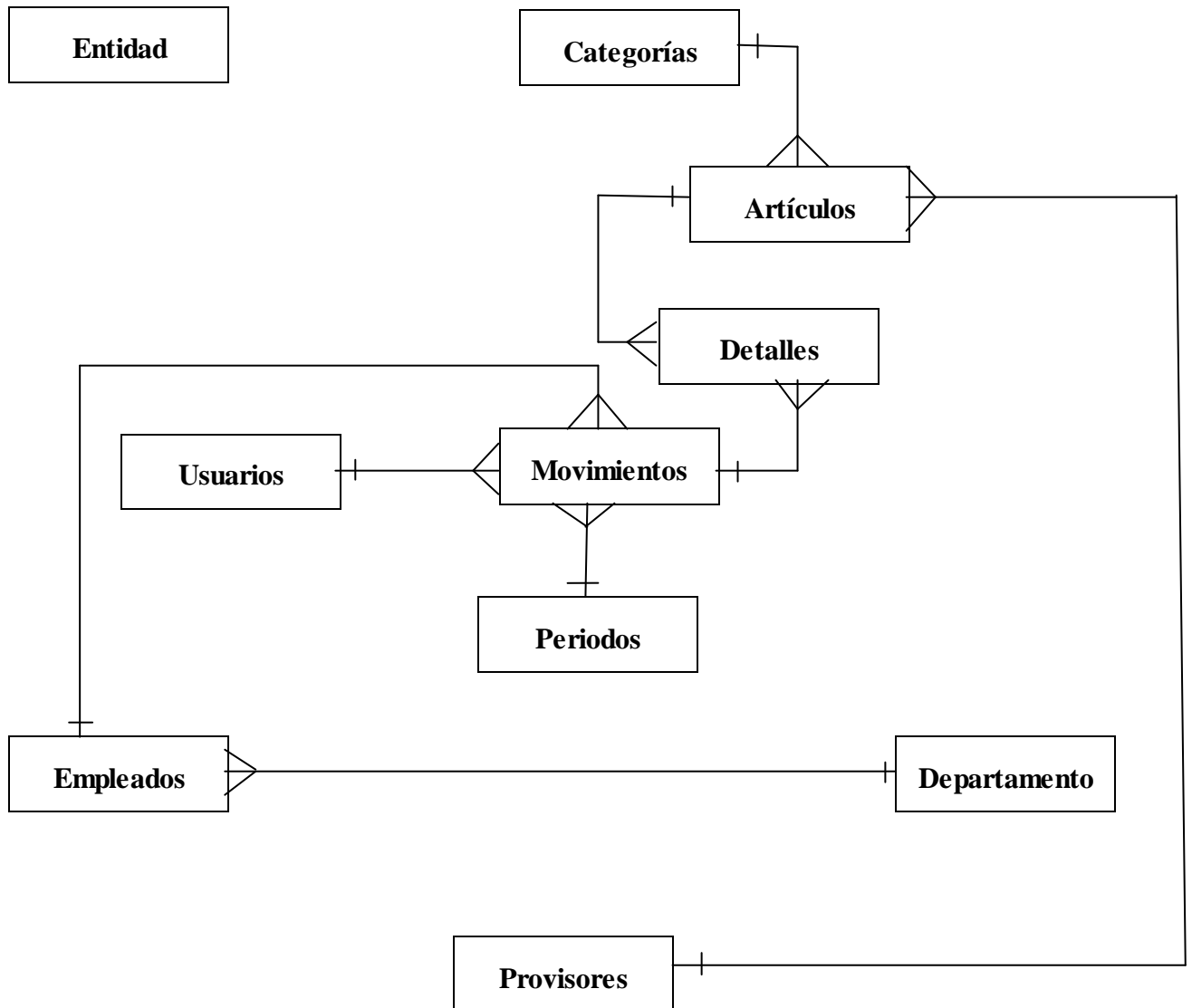
Control de Sesiones

Sesión Administrador

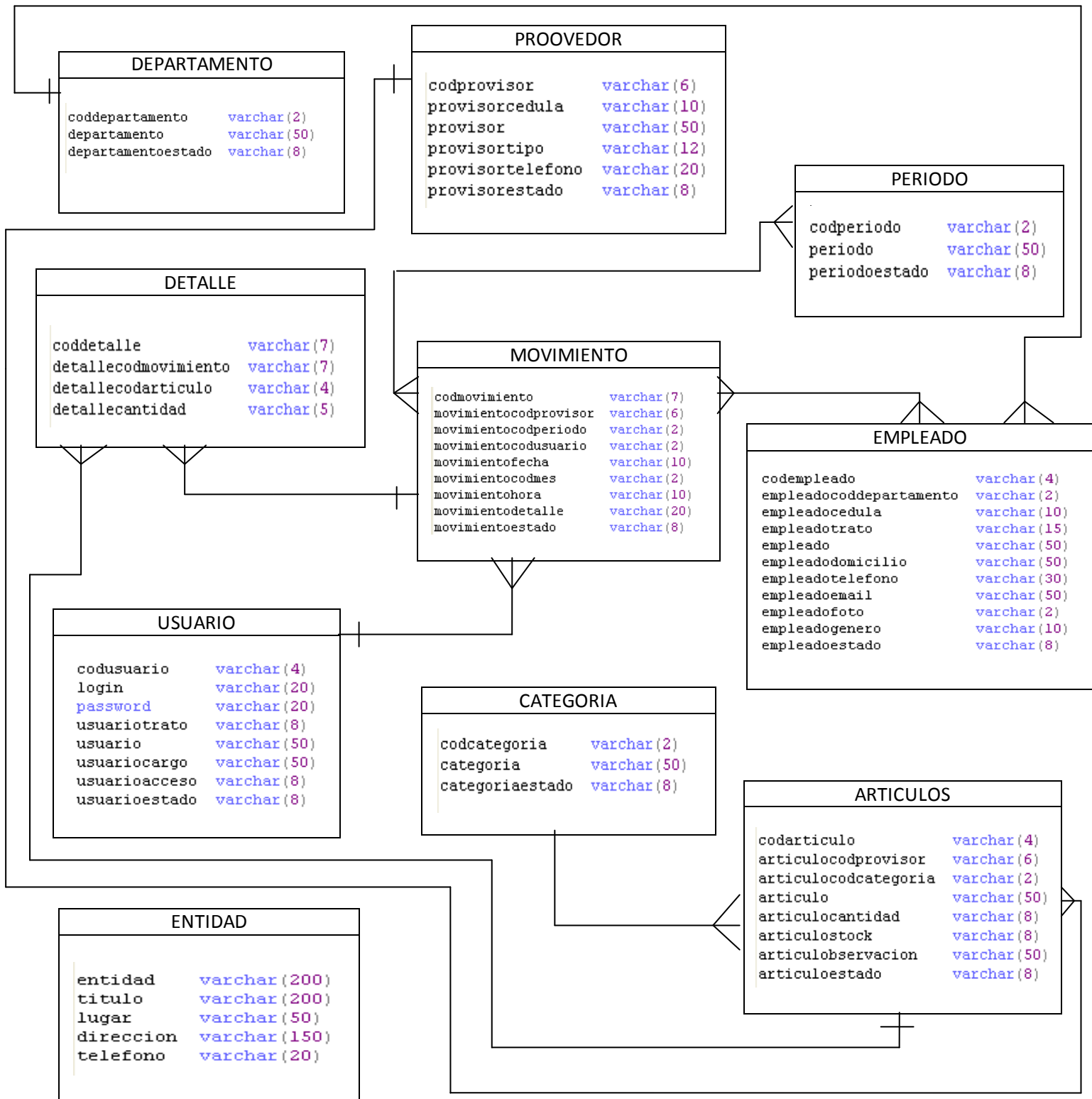
Sesión usuario

4.5.2 BASE DE DATOS (MODELO CONCEPTUAL Y MODELO FÍSICO)

4.5.2.1 MODELO CONCEPTUAL



4.5.2.2 MODELO FÍSICO



4.5.3 DICCIONARIO DE DATOS

Tabla 1 Usuarios

| Campo | Tipo | Null | Clave | Comentario |
|---------------|-------------|------|-------|--------------------------|
| codusuario | Varchar(6) | NO | PRI | Clave primaria |
| Login | varchar(10) | NO | | Alias del Usuario |
| password | varchar(50) | NO | | Contraseña de Usuario |
| usuariotrato | varchar(20) | NO | | Trato del Usuario |
| usuario | varchar(20) | NO | | Nombre de Usuario |
| usuariocargo | varchar(20) | NO | | Cargo del Usuario |
| usuarioacceso | varchar(20) | NO | | Seguridad de Acceso |
| usuarioestado | varchar(8) | NO | | Estado (Activo/Inactivo) |

Tabla 2 Proveedor

| Campo | Tipo | Null | Clave | Comentario |
|-------------------|-------------|------|-------|--------------------------|
| codproveedor | Varchar(6) | NO | PRI | Clave primaria |
| proveedorcedula | varchar(10) | NO | | Cedula del Proveedor |
| proveedor | varchar(50) | NO | | Nombre del Proveedor |
| proveedortelefono | varchar(20) | NO | | Telefono del Proveedor |
| proveedorestado | varchar(8) | NO | | Estado (Activo/Inactivo) |

Tabla 3 Movimientos:

| Campo | Tipo | Null | Clave | Comentario |
|------------------------------|-------------|-------------|--------------|--------------------------|
| Codmovimiento | varchar(7) | NO | PRI | Clave primaria |
| Movimientocodfactura | varchar(7) | NO | | Código de factura |
| movimientocodempleado | varchar(4) | NO | | Código de empleado |
| Movimientocodperiodo | varchar(3) | NO | MUL | Código de periodo |
| Movimientocodusuario | varchar(2) | NO | | Código de usuario |
| movimientofecha | varchar(10) | NO | | Fecha de ingreso |
| Movimientocodmes | varchar(2) | NO | | Código de mes |
| Movimientohora | varchar(10) | NO | | Hora |
| Movimientodetalle | varchar(20) | NO | | Detalle |
| Movimientoestado | varchar(8) | NO | | Estado (Activo/Inactivo) |

Tabla 4 Artículos:

| Campo | Tipo | Null | Clave | Comentario |
|-----------------------------|-------------|-------------|--------------|------------------------|
| codarticulo | varchar(4) | NO | PRI | Clave primaria |
| articulocodcategoria | varchar(2) | NO | | Código de categoría |
| Articulo | varchar(50) | NO | | Nombre del articulo |
| Articulocantidad | varchar(8) | NO | | Cantidad del articulo |
| Articulostock | varchar(8) | NO | | Stock del articulo |
| Articuloestado | varchar(8) | NO | | Estado activo/inactivo |

Tabla 5 Categoría:

| Campo | Tipo | Null | Clave | Comentario |
|------------------------|-------------|-------------|--------------|------------------------|
| Codcategoria | varchar(2) | NO | PRI | Clave primaria |
| Categoría | varchar(50) | NO | | Nombre de categoría |
| Categoriaestado | varchar(8) | NO | | Estado activo/inactivo |

Tabla 6 Departamentos:

| Campo | Tipo | Null | Clave | Comentario |
|--------------------------------|-------------|------|-------|------------------------|
| coddepartamento | varchar(2) | NO | PRI | Clave primaria |
| departamento | varchar(50) | NO | | Nombredel departamento |
| Departamentoesta do | varchar(8) | NO | | Estado activo/inactivo |

Tabla 7 Movimiento detalle:

| Campo | Tipo | Null | Clave | Comentario |
|-----------------------------|------------|------|-------|----------------------|
| Coddetalle | varchar(7) | NO | | Código de detalle |
| Detallecodmovimiento | varchar(7) | NO | | Código de movimiento |
| Detallecodarticulo | varchar(4) | NO | | Código de articulo |
| Detallecantidad | varchar(5) | NO | | Cantidad |

Tabla 8 Empleados:

| Campo | Tipo | Null | Clave | Comentario |
|--------------------------------|-----------------|------|-------|------------------------|
| codempleado | varchar(4) | NO | PRI | Clave primaria |
| empleadocoddepartamento | varchar(2) | NO | | Código de departamento |
| Cedula | varchar(10) | NO | | Numero de cedula |
| Trato | varchar(15) | NO | | Trato de Empleado |
| Empleado | varchar(50) | NO | | Nombre de empleado |
| Domicilio | varchar(50) | NO | | Domicilio |
| Teléfono | varchar(30) | NO | | N° Teléfono |
| Email | varchar(50) | NO | | Correo electrónico |

| | | | | |
|-----------------------|-----------------|----|--|-----------------------------|
| Genero | varchar(10) | NO | | Genero |
| Empleadoestado | varchar(8) | NO | | Estado (Activo/Inactivo) |
| Empleadotipo | varchar(8) | NO | | Tipo de Empleado |

Tabla 9 Entidad:

| Campo | Tipo | Null | Clave | Comentario |
|------------------|--------------|-------------|--------------|-------------------|
| Entidad | varchar(200) | NO | PRI | Clave primaria |
| Titulo | varchar(200) | NO | | Titulo |
| Lugar | varchar(50) | NO | | Lugar |
| Dirección | varchar(150) | NO | | Dirección |
| Teléfono | varchar(20) | NO | | N° Teléfono |

Tabla 10 Periodos:

| Campo | Tipo | Null | Clave | Comentario |
|----------------------|-------------|-------------|--------------|--------------------------|
| Codperiodo | varchar(3) | NO | PRI | Clave primaria |
| Periodo | varchar(50) | NO | | Nombre de Periodos |
| Periodoestado | varchar(8) | NO | | Estado (Activo/Inactivo) |

4.5.4 SCRIPT DE LA BASE DE DATOS

```
CREATE TABLE `articulos` (  
  `codarticulo` varchar(4) NOT NULL,  
  `articulocodprovisor` varchar(6) NOT NULL,  
  `articulocodcategoria` varchar(2) NOT NULL,  
  `articulo` varchar(50) NOT NULL,  
  `articulocantidad` varchar(8) NOT NULL,  
  `articulosstock` varchar(8) NOT NULL,  
  `articulosobservacion` varchar(50) NOT NULL,  
  `articuloestado` varchar(8) NOT NULL,  
  PRIMARY KEY (`codarticulo`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `categorias` (  
  `codcategoria` varchar(2) NOT NULL,  
  `categoria` varchar(50) NOT NULL,  
  `categoriaestado` varchar(8) NOT NULL,  
  PRIMARY KEY (`codcategoria`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `departamentos` (  
  `coddepartamento` varchar(2) NOT NULL,  
  `departamento` varchar(50) NOT NULL,  
  `departamentoestado` varchar(8) NOT NULL,  
  PRIMARY KEY (`coddepartamento`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `empleados` (  
  `codempleado` varchar(4) NOT NULL,  
  `empleadocoddepartamento` varchar(2) NOT NULL,  
  `empleadocedula` varchar(10) NOT NULL,  
  `empleadotrato` varchar(15) NOT NULL,  
  `empleado` varchar(50) NOT NULL,
```

```
`empleadodomicilio` varchar(50) NOT NULL,  
`empleadotelefono` varchar(30) NOT NULL,  
`empleadoemail` varchar(50) NOT NULL,  
`empleadofoto` varchar(2) NOT NULL,  
`empleadogenero` varchar(10) NOT NULL,  
`empleadoestado` varchar(8) NOT NULL,  
PRIMARY KEY (`codempleado`)
```

```
CREATE TABLE `entidad` (  
`entidad` varchar(200) NOT NULL,  
`titulo` varchar(200) NOT NULL,  
`lugar` varchar(50) NOT NULL,  
`direccion` varchar(150) NOT NULL,  
`telefono` varchar(20) NOT NULL,  
PRIMARY KEY (`entidad`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `movimientodetalle` (  
`coddetalle` varchar(7) NOT NULL,  
`detallecodmovimiento` varchar(7) NOT NULL,  
`detallecodarticulo` varchar(4) NOT NULL,  
`detallecantidad` varchar(5) NOT NULL
```

```
CREATE TABLE `movimientos` (  
`codmovimiento` varchar(7) NOT NULL,  
`movimientocodprovisor` varchar(6) NOT NULL,  
`movimientocodperiodo` varchar(2) NOT NULL,  
`movimientocodusuario` varchar(2) NOT NULL,  
`movimientofecha` varchar(10) NOT NULL,  
`movimientocodmes` varchar(2) NOT NULL,  
`movimientohora` varchar(10) NOT NULL,  
`movimientodetalle` varchar(20) NOT NULL,  
`movimientoestado` varchar(8) NOT NULL,  
PRIMARY KEY (`codmovimiento`),
```

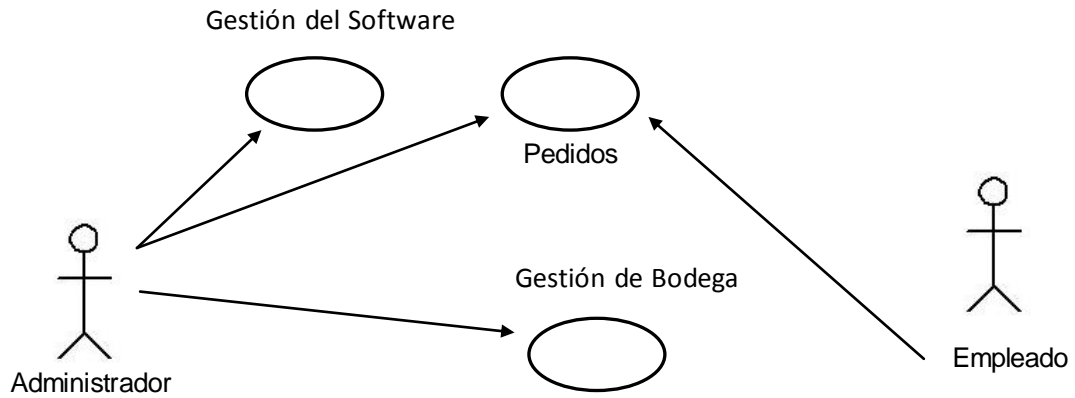
```
KEY `FK_matriculas` (`movimientocodperiodo`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `periodos` (  
  `codperiodo` varchar(2) NOT NULL,  
  `periodo` varchar(50) NOT NULL,  
  `periodoestado` varchar(8) NOT NULL,  
  PRIMARY KEY (`codperiodo`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

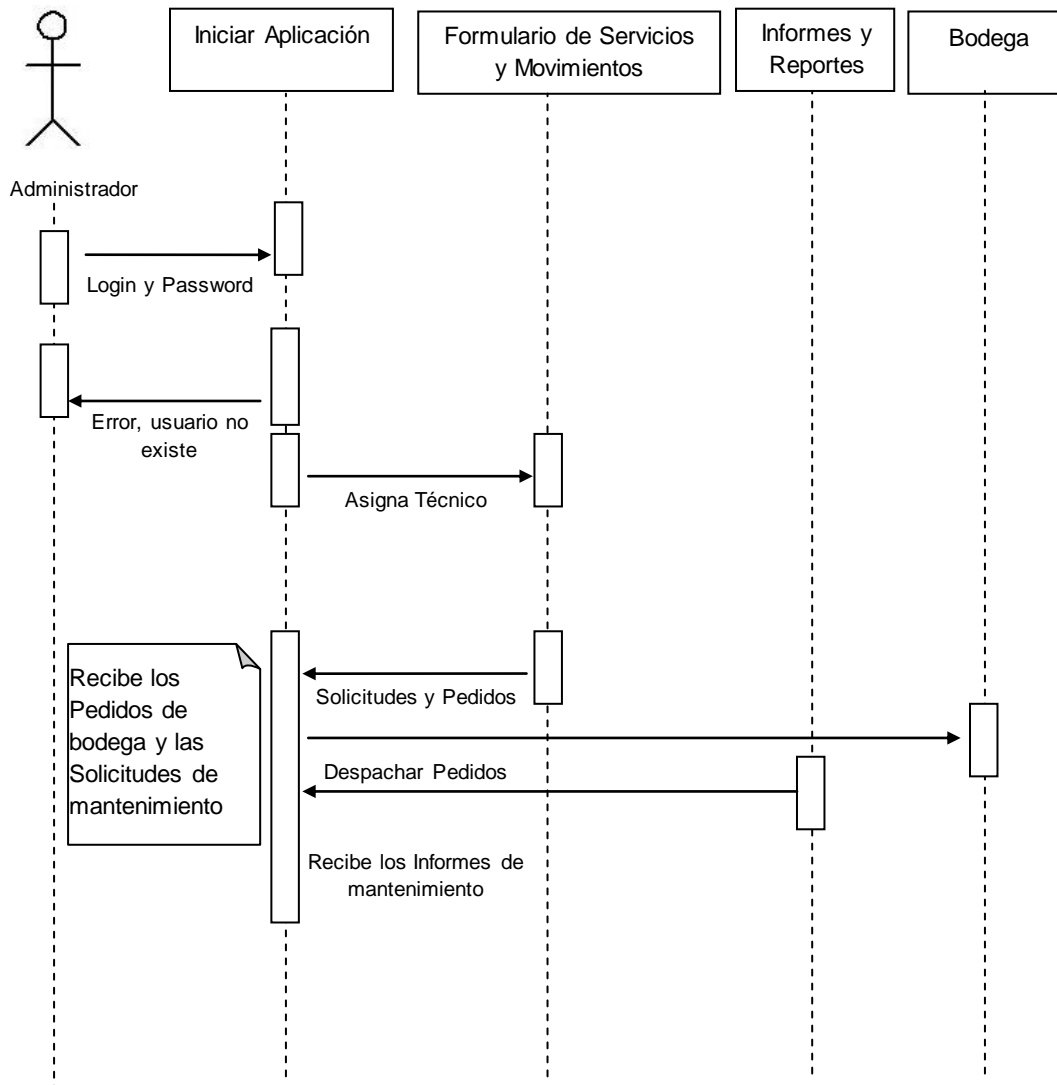
```
CREATE TABLE `provisores` (  
  `codprovisor` varchar(6) NOT NULL,  
  `provisorcedula` varchar(10) NOT NULL,  
  `provisor` varchar(50) NOT NULL,  
  `provisortipo` varchar(12) NOT NULL,  
  `provisortelefono` varchar(20) NOT NULL,  
  `provisorestado` varchar(8) NOT NULL,  
  PRIMARY KEY (`codprovisor`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `usuarios` (  
  `codusuario` varchar(4) NOT NULL,  
  `login` varchar(20) NOT NULL,  
  `password` varchar(20) NOT NULL,  
  `usuariorato` varchar(8) NOT NULL,  
  `usuario` varchar(50) NOT NULL,  
  `usuariocargo` varchar(50) NOT NULL,  
  `usuarioacceso` varchar(8) NOT NULL,  
  `usuarioestado` varchar(8) NOT NULL,  
  PRIMARY KEY (`codusuario`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

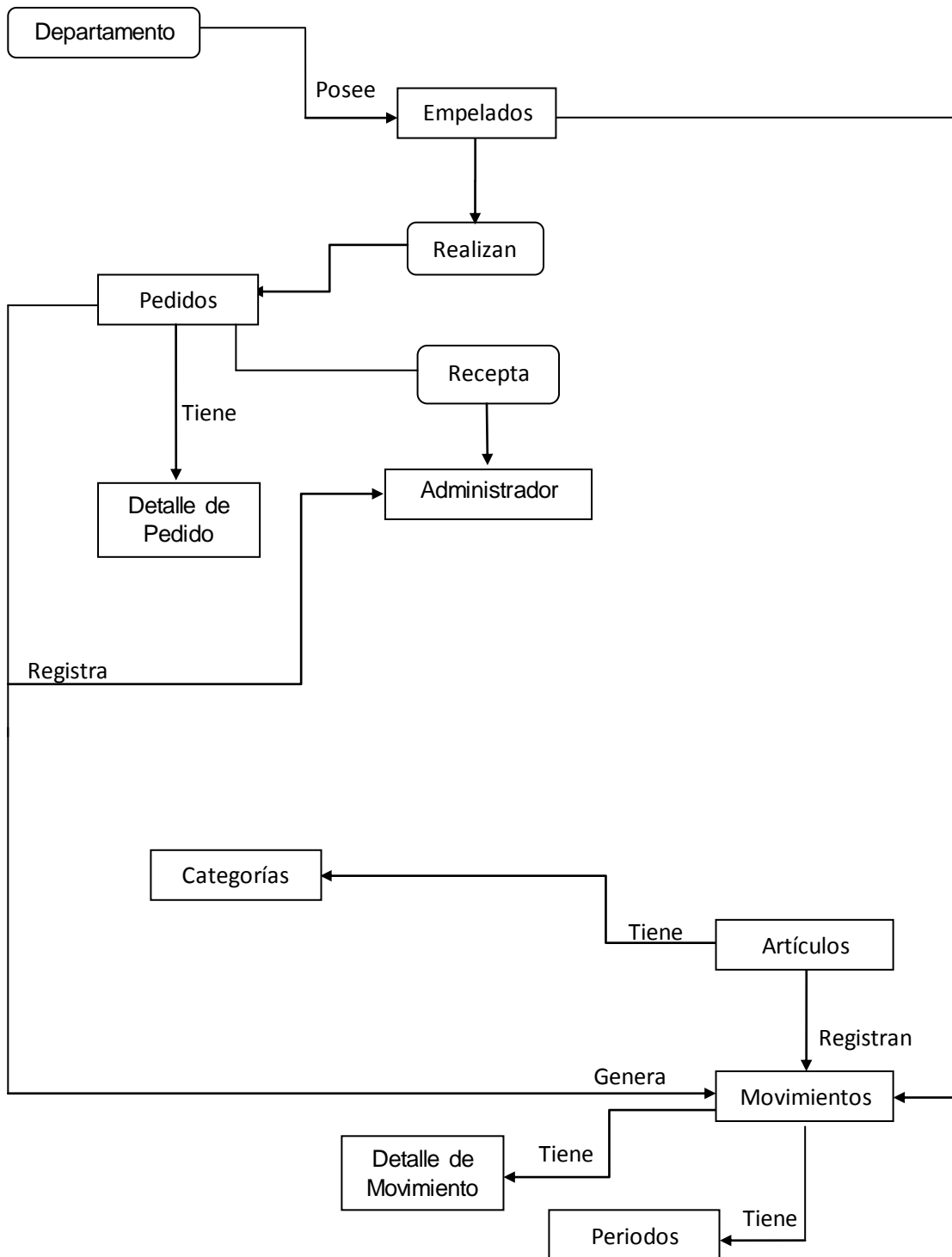
4.5.5 DIAGRAMA DE CASO DE USO



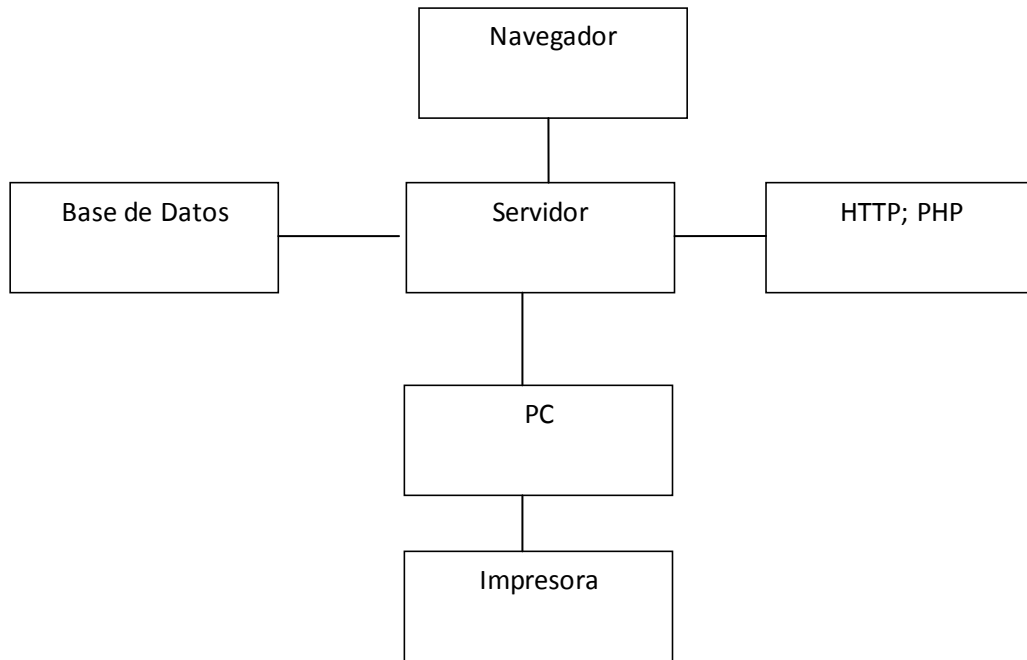
4.5.6 DIAGRAMA DE SECUENCIA



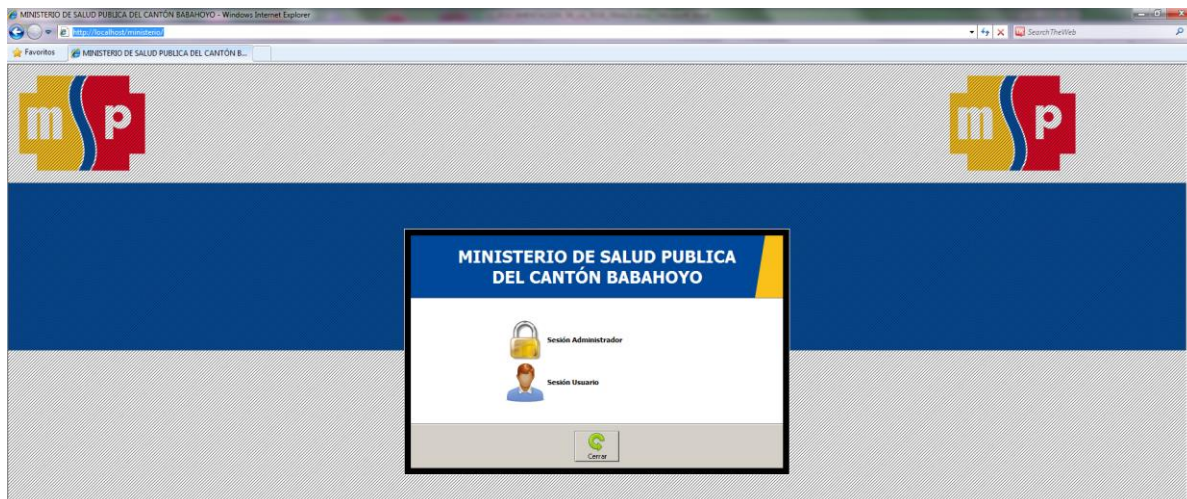
4.5.7 DIAGRAMA DE ACTIVIDADES



4.5.8 DIAGRAMA DE DESPLIEGUE



4.5.9 DISEÑO DE INTERFASES



Como podemos observar el sistema cuenta con dos formas de acceder al mismo, al momento de empezar sesión tanto administrador como usuario tendremos que llenar un login y un password como veremos a continuación. También tendremos un botón para cerrar el sistema.



Este formulario es para modo administrador como para usuario.

Al momento de acceder como administrador tendrá acceso a todo el sistema mientras que cuando lo haga como usuario no podrá acceder a todo.

Como usuario no podrá acceder a todo el sistema tendrá sus respectivas restricciones no podrá ingresar a (administrar, entidad, usuario, módulos y periodos) por ende los formularios se familiarizan.

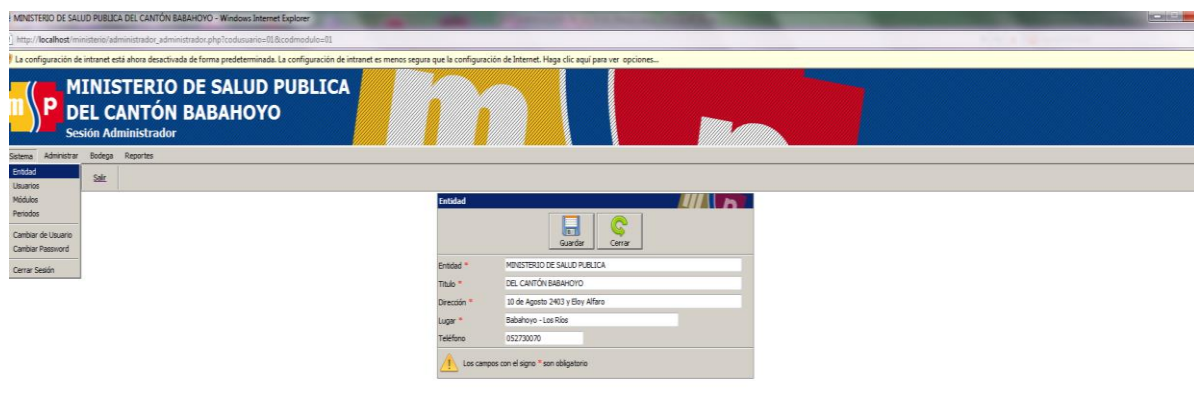
Si los datos fueron bien ingresados tanto para administrador como usuario veremos la siguiente página.

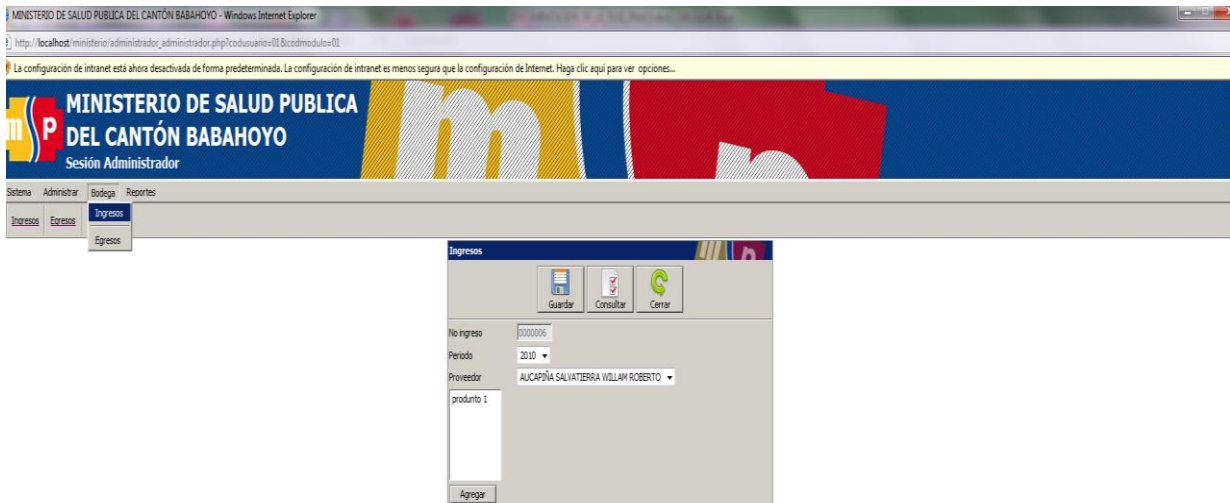
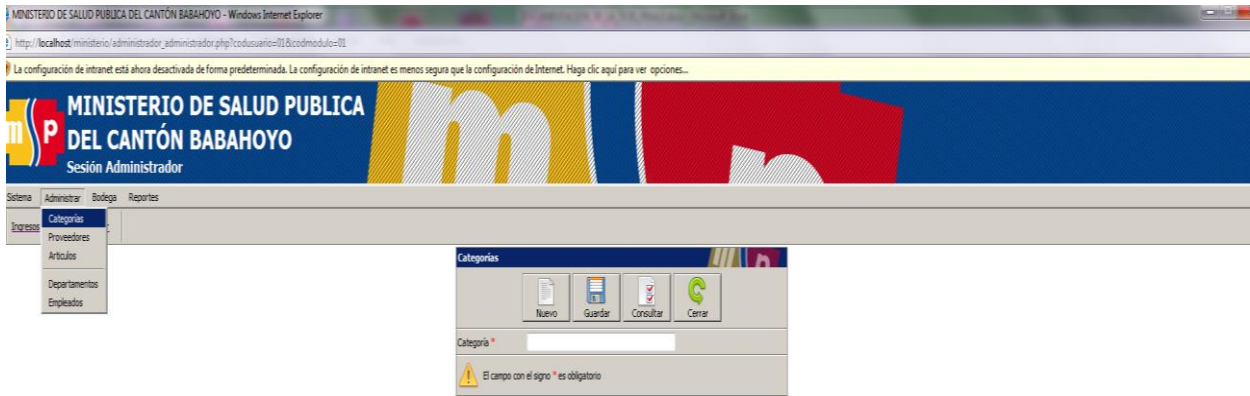
Y aquí tendremos el menú administrador que consta de Sistema, Administrar, Bodega, Reportes, también consta de Ingresos, Egresos y un botón para salir.



Por consiguiente al dar click en Sistema, Administrar, Bodega, Reportes. Hay diferentes formularios para cada submenú.

Cada página llama a otra página para ir desarrollando todas las situaciones planteadas por parte del sistema. Como se ve a continuación.

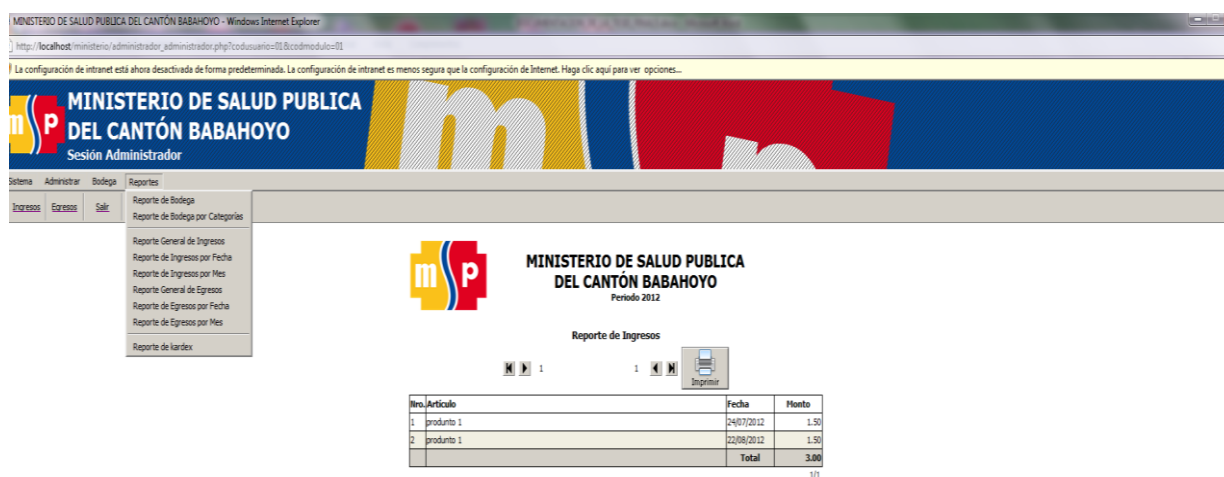
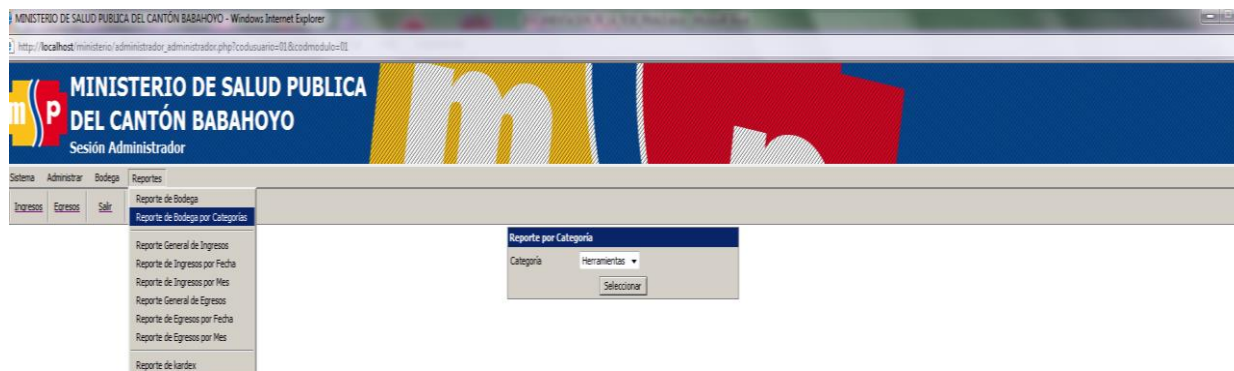




4.5.10 DISEÑO DE SALIDAS

Tendremos como salidas los siguientes reportes tanto por mes, fecha, egresos por categoría, ingresos:





4.6 DESARROLLO

A continuación se detalla el proceso de desarrollo de la aplicación.

4.7 PRUEBAS

4.7.1 IMPLEMENTACION DEL SISTEMA

El sistema funcionará en el departamento de BODEGA del Ministerio de Salud Pública del cantón Babahoyo bajo la responsabilidad de personas encargadas en su manejo. Los usuarios para poder utilizar el sistema deben contar con un usuario y una contraseña de acceso, una vez identificados en el sistemas, contarán con todas las opciones que brinda el sistema, realizarán los procesos necesarios y para finalizar deberán cerrar sesión.

4.7.1.1 REQUERIMIENTO DE HARDWARE

| Cantidad | Característica |
|--------------|--|
| 1 computador | procesador Pentium IV intel 3.2 ghz Disco duro 120gb Memoria ram DDR 512 mb, tarjeta de video 32 mb. Puerto paralelo, puerto serial, puerto usb, monitor Samsung "17", CD write 52x32x52 Fax modem 56 kpps Tarjeta red10/100 tarjeta de serial Mouse PS-2,teclado Una impresora |

4.7.1.2 REQUERIMIENTO DE SOFTWARE

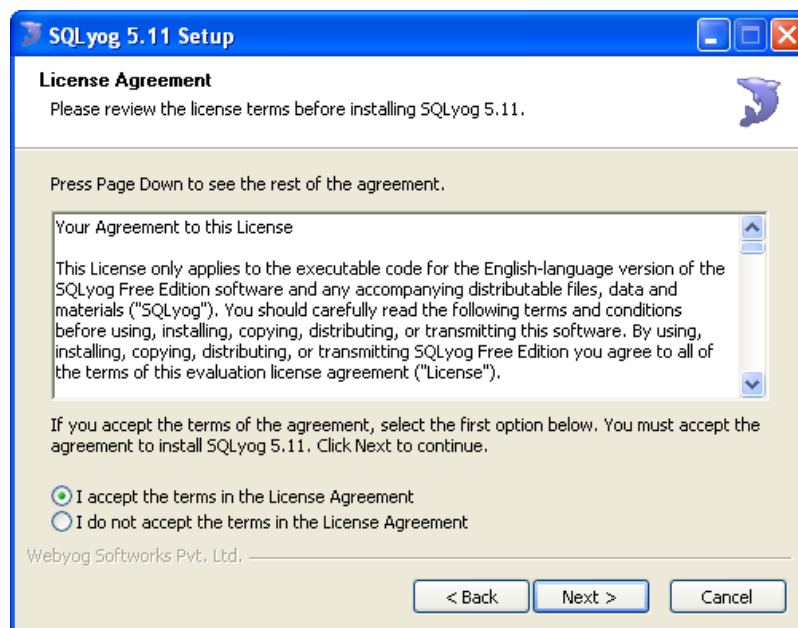
- MYSQL
- WAMP SERVER
- WINDOWS XP

4.7.1.3 PROCESO DE INSTALACION

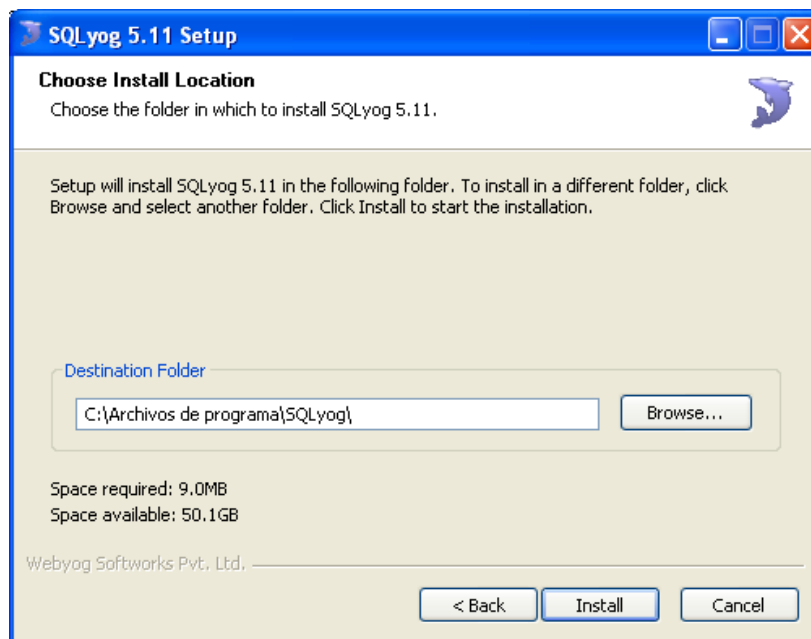
Instalación del programa SQLyog 5.11



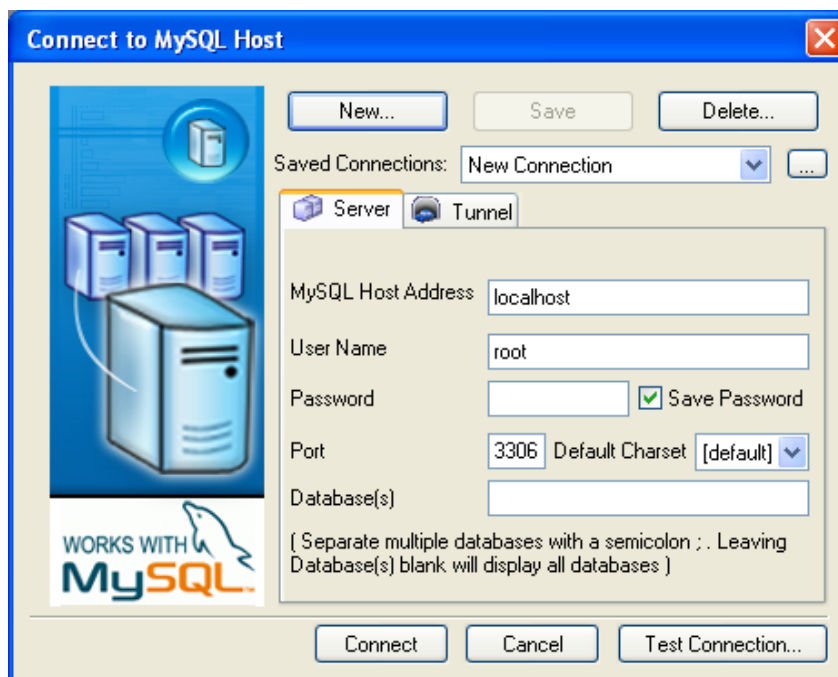
1.- Presione **Next**

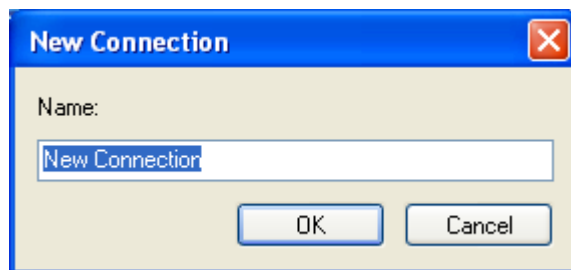


2.- Seleccione la primera opción y luego presione **Next**



3.- Presione **Install**





Presione OK

4.7.1.4 SEGURIDADES

Este sistema está dividido en dos módulos, que son Administrador y Usuario. Cada uno de ellos cumple una función en el sistema, la cual se la define de la siguiente manera:

A través de usos de sesiones ingresando (Login y Password) solo tendrán acceso al Software el o los usuarios calificados, evitando el acceso de personas ajenas al uso del programa.

Administrador: Sera la persona encargada de manipular la información que este contiene mediante la clave de acceso, lo que aquí se manipula es Bodega, en donde se lleva el registro del ingreso y egreso de los materiales de oficina e insumos médicos del ministerio de salud pública del cantón Babahoyo, las solicitudes o pedidos que el Usuario envía y los reportes de los registros.

Usuario: Sera la persona asignada por el administrador del software, este tendrá restricciones al momento de acceder, será el encargado de registrar los pedidos que realicen cada uno de los trabajadores de la institución la solicitud o pedido dirigidos al administrador.

4.8 CONCLUSIONES Y RECOMENDACIONES PARA UNA EFICIENTE IMPLEMENTACION DEL SOFTWARE

RECOMENDACIONES

- Los procesos de recolección de información se los realice diariamente para evitar confusión de fechas de los movimientos de Ingresos y Egresos, manteniendo así de la base de dato del servidor Web actualizada.
- Que el personal encargado de implementar el sistema tenga una completa capacitación del sistema para un óptimo manejo de todas las aplicaciones
- Para sistemas basadas en la Web se recomienda el uso de la plataforma que sean compatible con MySQL y PHP.

CONCLUSIONES

- Los datos implementados en esta tesis generara información de tipo Gestión de inventario permitiendo la elaboración de reportes detallados que establezcan un control de cada uno de los movimiento de Insumos Médicos y materiales de Oficina del Ministerio de Salud Pública del Cantón Babahoyo
- Los módulos desarrollados (Usuario y Administrador) restringen los accesos a persona no calificado para el uso del Software.
- La información de los reportes se presenta de una manera más comprensible y amigable para que el usuario encargado de manejar el software obtenga un mejor resultado. El desarrollo de este tema concluimos que es posible presentar información de bodega y sus movimientos en un informe final.

BIBLIOGRAFIA

- JOEL MANRRIQUE CHÁVEZ, Lenguaje de Programación PHP, Lima, 2011
- Dynamic HTML Reference and Software Development Kit
- Análisis y Diseño de sistemas de Información. Autor (James A. Senn.)
- SQLyog
- PHP con base de datos

LINKOGRAFIA

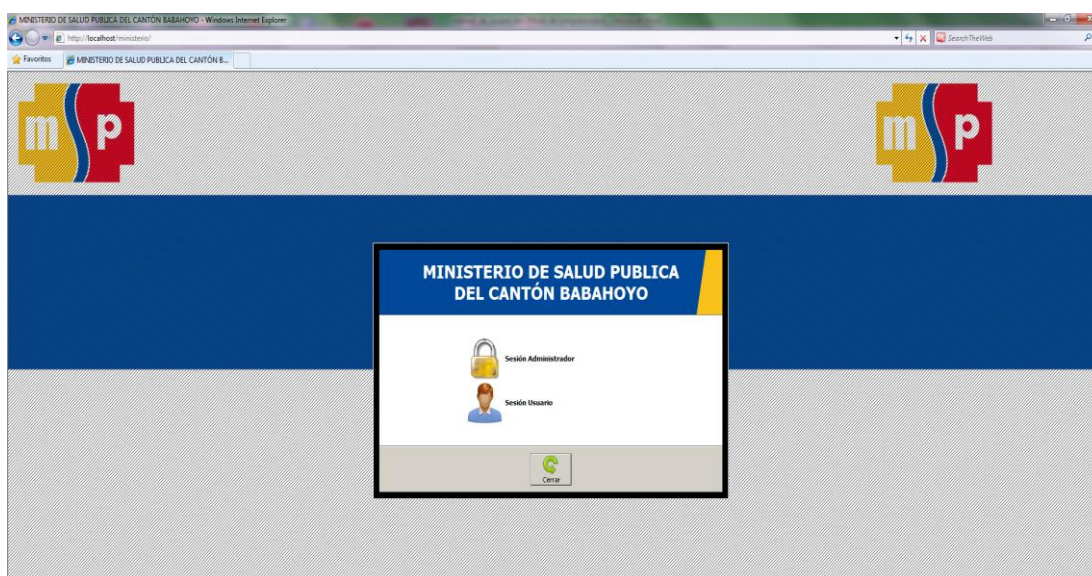
- <http://html.rincondelvago.com/concepto-de-base-de-datos.html>
- <http://ola.icmyl.unam.mx/biblio/Tesis-Bus.php>
- <http://www.webestilo.com/foros/mensaje.php>
- <http://www.programacion.com/foros/5/msg/25774/>
- http://www.fvet.uba.ar/biblioteca/como_hacer.htm
- <http://www.programacion.com/php/>
- <http://www.br.uipr.edu/caiweb/default2.asp?tree=625>
- <http://www.asesoriatesis.com/cgi-bin/default.php>

ANEXOS

MANUAL DE USUARIO

A continuación les presento el manual de usuario en donde se detalla el funcionamiento del Sistema de Control, Retiro y Seguimiento de Materiales de oficina e insumos médicos para el departamento de BODEGA del Ministerio de Salud Pública del cantón Babahoyo.

Primero tenemos la pantalla principal del sistema, donde nos permite escoger entre 2 usuarios (ADMINISTRADOR, USUARIO).



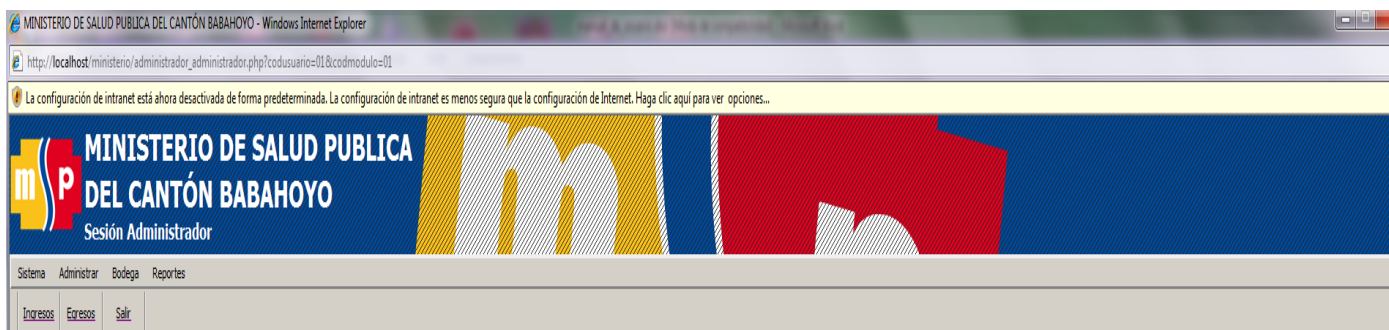
Para acceder al sistema se lo realiza ingresando ya sea para modo administrador como para usuario un Login (Nombre de Usuario) y el Password (Contraseña) asignada.

Veremos la siguiente pantalla:






Administración de la barra de menú

A continuación se muestra el manejo de las diferentes opciones de la barra de menú que ofrece el sistema.



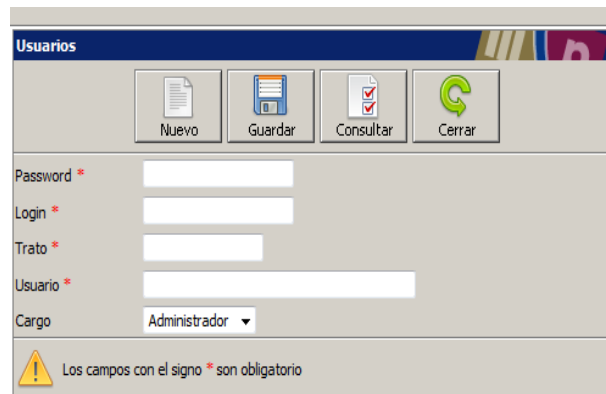
Sistema.- Esta opción del menú permitirá el acceso a los siguientes ítems.

Entidad.- Formulario donde se ingresara los datos de la Institución

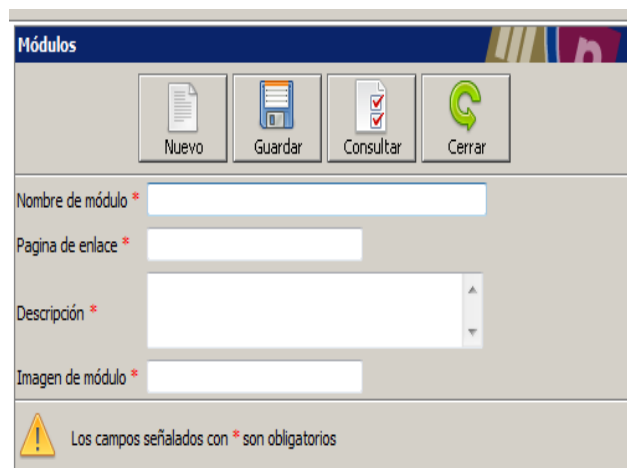
| Entidad | |
|---|---|
| |   |
| Entidad * | MINISTERIO DE SALUD PUBLICA |
| Título * | DEL CANTÓN BABAHOYO |
| Dirección * | Bolivar entre 5 de Junio y General Barona |
| Lugar * | Babahoyo - Los Ríos |
| Teléfono | 052730070 |
|  Los campos con el signo * son obligatorio | |

| | |
|-------------------------|-----------------------------|
| Unidad Educativa | Nombre de la Institución |
| Título | Nombre del Sistema |
| Dirección | Ubicación de la Institución |
| Lugar | Ciudad y País |
| Teléfono | Números telefónicos |

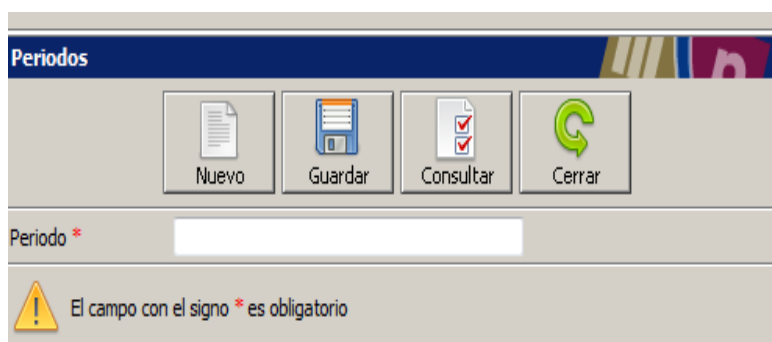
Usuarios.- Aquí podrán cambiar tanto la contraseña como el nombre del usuario para ambos.



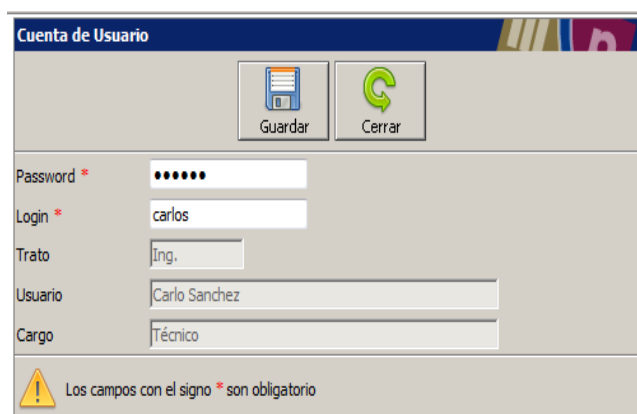
Módulos.- Este formulario permitirá agregar nuevos procesos que se necesiten para el buen funcionamiento del sistema.



Periodos.- Este nos permitirá ingresar un nuevo periodo de acuerdo a las necesidades que se presenten.

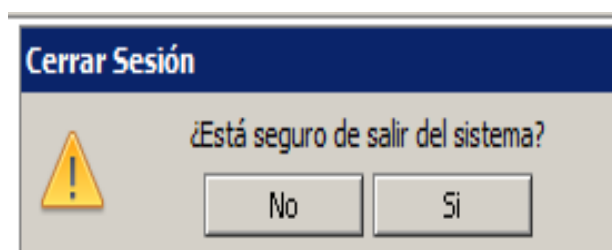


Cambiar Password.- Este formulario permite modificar el Login y el Password y actualizar sus datos si el usuario no posee cuenta de usuario.



The screenshot shows a web form titled "Cuenta de Usuario". At the top, there are two buttons: "Guardar" (Save) and "Cerrar" (Close). Below these are several input fields: "Password *" with masked characters, "Login *" with the value "carlos", "Trato" with "Ing.", "Usuario" with "Carlo Sanchez", and "Cargo" with "Técnico". A warning icon and text at the bottom state: "Los campos con el signo * son obligatorio" (Fields with the asterisk * are mandatory).

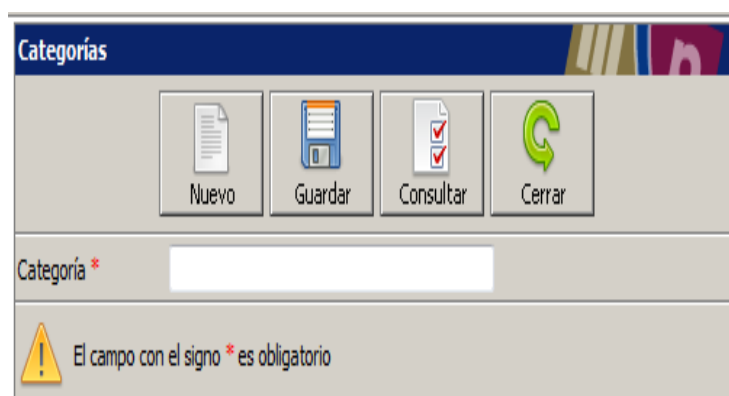
Cerrar Sesión.- Esta opción termina una sesión de un usuario específico, finalizando así el uso del sistema.



The screenshot shows a dialog box titled "Cerrar Sesión". It contains a warning icon and the question "¿Está seguro de salir del sistema?". There are two buttons: "No" and "Si" (Yes).

Administrar.- En este ítems contara con los siguientes submenús.

Categoría.- Nos permitirá realizar el ingreso como su nombre mismo lo indica por la categoría del artículo, ya sea este material de oficina o insumos médicos



The screenshot shows a web form titled "Categorías". At the top, there are four buttons: "Nuevo" (New), "Guardar" (Save), "Consultar" (Consult), and "Cerrar" (Close). Below these is a single input field labeled "Categoría *". A warning icon and text at the bottom state: "El campo con el signo * es obligatorio" (The field with the asterisk * is mandatory).

Proveedores.- Ingresaremos todos los tipos de proveedores que abastecen a la institución para el departamento de bodega.

Proveedores

Nuevo Guardar Consultar Cerrar

Cedula *

Proveedor *

Teléfono

! Los campos con el signo * son obligatorio

Artículos.- Permitirá el ingreso de los diferentes artículos

Artículos

Nuevo Guardar Consultar Cerrar

Proveedor AUCAPIÑA SALVATIERRA WILLAM ROBERTO

Categoría Herramientas

Artículo *

Cantidad *

Precio de Compra *

Observación

! Los campos con el signo * son obligatorio

Departamento.- Aquí se registrara todos los departamentos que formen parte de la institución.

Departamentos

Nuevo Guardar Consultar Cerrar

Departamento *

! El campo con el signo * es obligatorio

Empleados.- Permitira registrar a todos los empleados que trabajen dentro de la institucion. Ademas contara con botones que permitiran consultar, guardar, nuevo y cerrar.

Empleados

Nuevo Guardar Consultar Cerrar

Departamento Departamento de Financiero

Cedula *

Password *

Login *

Trato *

Empleado *

Cargo *

Los campos con el signo * son obligatorio

Bodega.- Contara con dos items el uno sera para el ingreso y el otro para egreso de los articulos

Ingresos

Guardar Consultar Cerrar

No ingreso 0000004

Periodo 2010

Proveedor AUCAPIÑA SALVATIERRA WILLAM ROBERTO

guantes
resmas de papel

Agregar

Reportes.- Tendremos algunas formas de realizar reportes de bodega, por categorías, general de ingreso, por fecha, por mes, egresos general, por fecha, por mes y un kardex



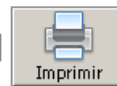
**MINISTERIO DE SALUD PUBLICA
DEL CANTÓN BABAHOYO**

Periodo 2012

Reporte de Bodega

1

1



| Nro. | Artículo | Categoría | Cantidad | Mínimo |
|------|-----------------|-----------------------|----------|--------|
| 1 | resmas de papel | materiales de oficina | 100 | 90 |

1/1

Este formulario muestra un reporte por categorías



**MINISTERIO DE SALUD PUBLICA
DEL CANTÓN BABAHOYO**

Periodo 2012

Reporte de Bodega por materiales de oficina

1

1



| Nro. | Artículo | Cantidad | Mínimo |
|------|-----------------|----------|--------|
| 1 | resmas de papel | 100 | 90 |

1/1

Nos muestra un reporte de ingreso general y a su vez nos mostrara reportes, tanto por fecha, como mes.



**MINISTERIO DE SALUD PUBLICA
DEL CANTÓN BABAHOYO**
Periodo 2012

Reporte de Ingresos




| Nro. | Articulo | Fecha | Monto |
|--------------|------------|------------|--------------|
| 1 | producto 1 | 24/07/2012 | 1.50 |
| 2 | producto 1 | 22/08/2012 | 1.50 |
| 3 | guantes | 31/08/2012 | 20.00 |
| Total | | | 23.00 |

1/2

Seleccionamos la fecha que queremos hacer la busqueda y damos un enter en BUSCAR.

Reporte de Ingresos diario

Seleccionar fecha 

- 22/08/2012
- 24/07/2012
- 31/08/2012

Automáticamente el sistema nos mostrara un kardex por fecha



**MINISTERIO DE SALUD PUBLICA
DEL CANTÓN BABAHOYO**
Periodo 2012

Reporte de Ingresos del 31/08/2012



| Nro. | Articulo | Fecha | Precio | Cantidad | Total |
|--------------|----------|------------|--------|----------|--------------|
| 1 | guantes | 31/08/2012 | 2.00 | 10 | 20.00 |
| Total | | | | | 20.00 |

1/1

Lo mismo sucederá para cuando queramos realizar un reporte por mes.

Reporte de Ingresos por mes

Seleccionar Mes Julio ▼

Julio
Agosto

Seleccionar



**MINISTERIO DE SALUD PUBLICA
DEL CANTÓN BABAHOYO**
Periodo 2012

Reporte de Ingresos del mes de Julio

1

1



| Nro. | Artículo | Fecha | Precio | Cantidad | Total |
|------|------------|------------|--------|--------------|-------------|
| 1 | producto 1 | 24/07/2012 | 1.50 | 1 | 1.50 |
| | | | | Total | 1.50 |

1/1



**MINISTERIO DE SALUD PUBLICA
DEL CANTÓN BABAHOYO**
Periodo 2012

Reporte de Egresos

1

1



| Nro. | Artículo | Fecha | Precio | Cantidad | Total |
|------|-----------------|------------|--------|--------------|--------------|
| 1 | producto 1 | 24/07/2012 | 1.50 | 1 | 1.50 |
| 2 | producto 1 | 22/08/2012 | 1.50 | 4 | 6.00 |
| 3 | resmas de papel | 31/08/2012 | 1.00 | 10 | 10.00 |
| 4 | guantes | 31/08/2012 | 2.00 | 30 | 60.00 |
| | | | | Total | 77.50 |

1/1

Reporte de Egresos diario

Seleccionar fecha ▼

- 22/08/2012
- 24/07/2012
- 31/08/2012

ar



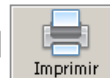
MINISTERIO DE SALUD PUBLICA DEL CANTÓN BABAHOYO

Periodo 2012

Reporte de Egresos del 22/08/2012

◀ ▶ 1

1 ▶ ▶



| Nro. | Artículo | Fecha | Precio | Cantidad | Total |
|------|------------|------------|--------|--------------|-------------|
| 1 | producto 1 | 22/08/2012 | 1.50 | 4 | 6.00 |
| | | | | Total | 6.00 |

1/1

Reporte de Egresos por mes

Seleccionar Mes ▼

- Julio
- Agosto

ccionar



MINISTERIO DE SALUD PUBLICA DEL CANTÓN BABAHOYO

Periodo 2012

Reporte de Egresos del mes de Agosto

◀ ▶ 1

1 ▶ ▶




| Nro. | Artículo | Fecha | Precio | Cantidad | Total |
|------|-----------------|------------|--------|--------------|--------------|
| 1 | producto 1 | 22/08/2012 | 1.50 | 4 | 6.00 |
| 2 | resmas de papel | 31/08/2012 | 1.00 | 10 | 10.00 |
| 3 | guantes | 31/08/2012 | 2.00 | 30 | 60.00 |
| | | | | Total | 76.00 |

1/1

Reporte de Kardex

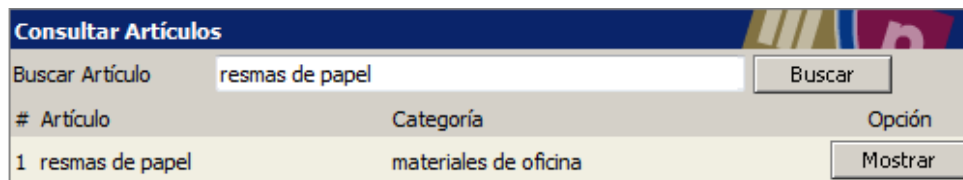
Primero escribimos el nombre del artículo a buscar sea este de un material o de un insumo.



Consultar Artículos

Buscar Artículo

Luego le damos click en el botón BUSCAR

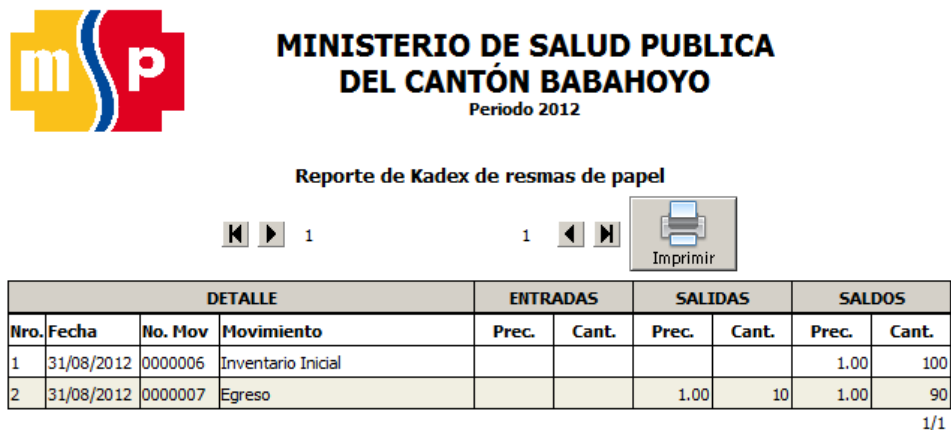



Consultar Artículos

Buscar Artículo

| # Artículo | Categoría | Opción |
|-------------------|-----------------------|--|
| 1 resmas de papel | materiales de oficina | <input type="button" value="Mostrar"/> |

Finalmente nos mostrara el Kardex del artículo a buscar.



 **MINISTERIO DE SALUD PÚBLICA
DEL CANTÓN BABAHOYO**
Periodo 2012

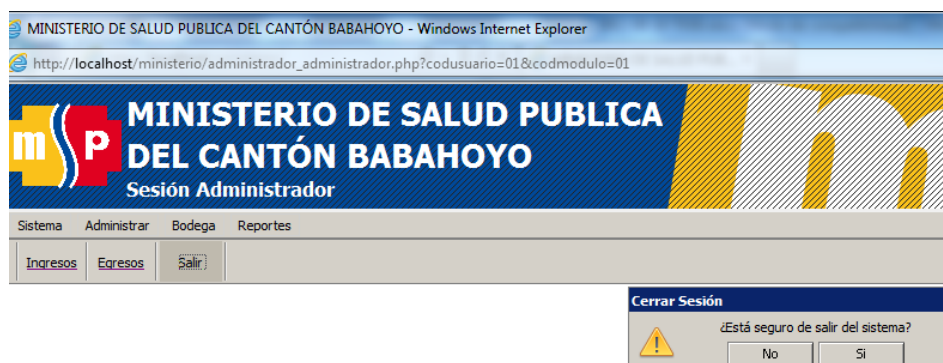
Reporte de Kadex de resmas de papel

1

| DETALLE | | | | ENTRADAS | | SALIDAS | | SALDOS | |
|---------|------------|---------|--------------------|----------|-------|---------|-------|--------|-------|
| Nro. | Fecha | No. Mov | Movimiento | Prec. | Cant. | Prec. | Cant. | Prec. | Cant. |
| 1 | 31/08/2012 | 0000006 | Inventario Inicial | | | | | 1.00 | 100 |
| 2 | 31/08/2012 | 0000007 | Egreso | | | 1.00 | 10 | 1.00 | 90 |

1/1

Por último el sistema cuenta con un BOTON para salir de la aplicación.



MINISTERIO DE SALUD PÚBLICA DEL CANTÓN BABAHOYO - Windows Internet Explorer

http://localhost/ministerio/administrador_administrador.php?codusuario=01&codmodulo=01

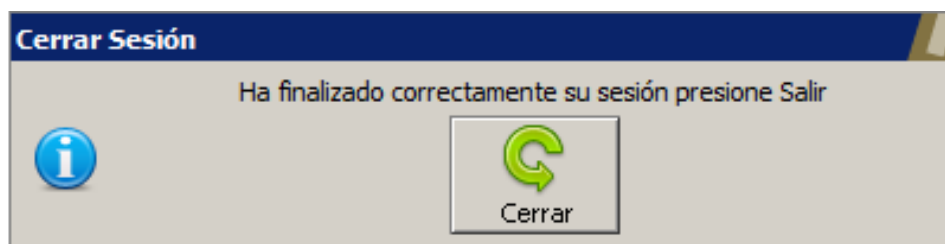
**MINISTERIO DE SALUD PÚBLICA
DEL CANTÓN BABAHOYO**
Sesión Administrador

Sistema Administrar Bodega Reportes

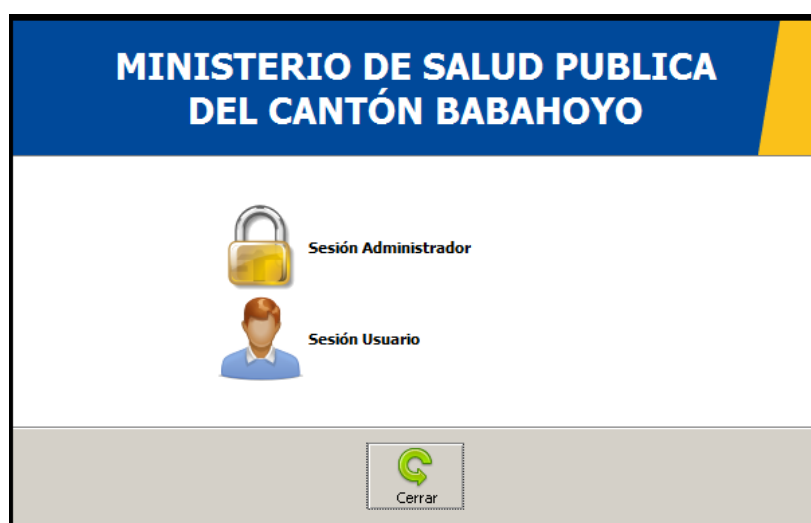
Ingresos Egresos

Cerrar Sesión
¿Está seguro de salir del sistema?

Al momento de presionar **SI** veremos la siguiente pantalla.



Al dar clip regresara a la pantalla principal donde podrá ingresar con otro usuario, lógicamente asignado.



Y si quiere definitivamente salir tendrá que dar clip en el BOTON cerrar y le aparecerá un mensaje. Donde le preguntara si desea cerrarla.

